International Journal of Industrial Engineering & Production Research December 2022 Vol. 33, No. 4: 1-10
DOI: 10.22068/ijiepr.33.4.12

RESEARCH PAPER

Optimal Drilling Sequences for Rectangular Hole Matrices Using Modified Shuffled Frog Leaping Algorithm

Amol M. Dalavi*[1], Padmakar J. Pawar[2], Tejinder P. Singh[3] & Alyssa Gomes[4]

Received 22 July 2021; Revised 19 June 2022; Accepted 12 September 2022;

ABSTRACT

Several industrial products, including engine blocks, moulds, automotive parts, dies, etc., require many holes to be machined into them. Similarly, applications like processing separators in the food industry, printed circuit boards, drum and trammel screens, boilerplates, etc., also contain a large matrix of holes. Many machining operations, such as drilling, tapping, and reaming, are needed to achieve the required hole diameters, which various operational sequences can produce. One of the biggest hassles with hole-making operations is optimizing the sequence of operations to achieve minimal tool travel time. This is essential in reducing the overall processing costs associated with hole-making operations. The present work attempts to minimize the total tool travel time for hole-drilling operations by employing a relatively new nature-inspired optimization algorithm - the modified shuffled frog leaping algorithm (SFLA). The modifications in SFLA are introduced using three positive parameters to broaden its search capacity. This paper considers three case studies of a rectangular matrix of holes to explain the proposed procedure. The outcomes of the optimization using modified SFLA are compared to those obtained using a genetic algorithm and the ant colony algorithm. Additionally, a higher dimensional problem of a 20x20 rectangular matrix of holes is considered in this work.

KEYWORDS: *Hole-drilling; Injection moulding; Modified shuffled frog leaping algorithm; Advanced optimization techniques; Tool path planning.*

1. Introduction

The manufacturing business is continually striving to cut the price of production to meet the requirements dictated by an unprecedented level of competition and regular improvements in product design & new product introduction. Technology plays a significant role in achieving this objective of cost-cutting. Manufacturers employ multiple advanced machines using computer numerical control and other technologies. These advanced machines require the use of computer programming for the optimization of time and resources spent on carrying out operations. Hence, optimizing operational processes is crucial to achieving success in this business.

Hole-making operations are combinatorial problems in which the correct sequence of drilling holes needs to be worked out. In such a combinatorial model, minimization of tool time and cost is the objective function, and the operational limitations serve as the constraints. Hence, an adequate mathematical model can be developed, which can be solved to determine the best order of drilling operations. With the aid of local and global optima, metaheuristic algorithms iteratively advance from a less optimal solution to an ideal solution. Many metaheuristic algorithms have already been employed to solve hole-making problems. These algorithms emulate processes that occur in nature, like the process of natural selection in the genetic algorithm and the foraging behaviour of animals in swarm-inspired algorithms like the ant colony optimizer, the shuffled frog-leaping process, and so on.

* **Corresponding author:** ***Amol M. Dalavi***
amol.dalavi@sitpune.edu.in

1. *Department of Mechanical Engineering, Symbiosis Institute of Technology, Symbiosis International University, Gram Lavale, Mulshi, Pune, India 412115.*
2. *Department of Production Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik, India*
3. *Department of Mechanical Engineering, Symbiosis Institute of Technology, Symbiosis International University, Gram Lavale, Mulshi, Pune, India 412115.*
4. *Department of Mechanical Engineering, Symbiosis Institute of Technology, Symbiosis International University, Gram Lavale, Mulshi, Pune, India 412115.*

2. Literature Review

Optimizing drilling sequences for computer numerically controlled (CNC) machines is vital to reduce non-productive machining time such as tool travel and changeover time and its associated costs. On average, 70% of the total manufacturing time is wasted on tool and part movements [1].

Previously, the ant colony optimization (ACO) algorithm and genetic algorithm (GA) were used to optimize a drilling path for a rectangular matrix of holes [2, 3]. Different non-traditional optimization methods, such as GA, ACO, and simulated annealing (SA) were used to obtain the ideal cutting parameters for a multi-tool CNC system [4]. Another case study employed particle swarm optimization (PSO) to obtain the optimal drilling path [5]. The shortest overall machine processing time for an injection mould was determined using a GA, and the results were compared with those obtained using SA [6]. A GA was employed to generate the shortest machine tool cutting path. [7]. Another case study demonstrated the use of the ACO to obtain the best path for cutting holes in a common industrial item. Initially, the ACO algorithm was validated using six benchmark functions and compared with the results obtained via dynamic programming (DP) [8]. An immunity-based evolutionary method (IA) was used to optimize a machining process by reducing non-productive tool travel and switch time. The findings of this study were then compared with those obtained using ACO and PSO [9]. Similarly, another evolutionary optimization algorithm based on the geographic classification of biological organisms was applied to a case study wherein the goal was to optimize tool travel and switching time during hole-making operations. The algorithm's performance was validated using test functions from literature [10].

An open vehicle routing problem was solved using the bumble bee mating optimization algorithm, to reduce the travel distance. The algorithm was validated using two benchmark problems [11]. The path sequence for a multi-drilling operation for concentric circular patterns was optimized by employing six popular metaheuristic optimization algorithms [12]. A novel hybrid combinatorial meta-heuristic method was utilized to solve a multi-hole drill sequencing problem [13].

The problem of obtaining optimal tool travel in hole-making operations can be compared to the travelling salesman (TSP) problem. Here, each hole must be visited by the tool, followed by the drilling operation, and finally, after all the holes have been drilled, the tool has to return to its original position. For hole-making operations, the problem of optimality arises when the number of holes is increased. Therefore, an essential step in such manufacturing processes is determining the shortest drill sequence to reduce valuable non-productive tool travel time and, consequently, manufacturing costs.

Tabu Search, Bumble Bees Mating Optimization Algorithm, and Biogeography Based Optimization (BBO) algorithm have also been attempted in previous literature to optimize hole-making operations. However, most previously attempted methods have certain limitations, including constricted search capabilities, multiple tunable parameters, intensive computation requirements, and slow convergence, among others. For example, the pure Tabu Search technique might ignore some areas of the search space which could potentially contain the ideal solution. Similarly, GA can achieve near-optimal solutions for complex problems [14]. However, for GA and IA, there are several parameter requirements [15]. The ACO algorithm's rate of convergence is slow because of pheromone evaporation, which drastically increases the CPU process time [15]. The honey bee mating optimization algorithm may provide a near-optimum solution instead of the optimum when the runtime is limited [16]. The biogeography-based optimization algorithm cannot effectively exploit the optimal solutions since it cannot select the best solution at each generation [17]. Hence, there is a need for a non-traditional optimization technique that is robust and provides optimal solutions for complex problems and search spaces [14]. By modifying the previously established Shuffled Frog Leaping Algorithm, this study examines how it may be possible to lower the overall tool travel distance in drilling operations for a rectangular matrix of holes [2].

3. Modified Shuffled Frog Leaping
1.3. Algorithm

Eusuff and Lansey developed a meta-heuristic algorithm called the Shuffled Frog Leaping Algorithm (SFLA). The algorithm is based on the behaviour of a group of frogs hunting for food in a pond. The group of frogs tries to locate the spots in the pond with the most amount of food. [18]. In SFLA, the intial "population" is made up of a randomised set of frogs which is further divided into smaller subgroups called "memeplexes." Each frog carries out two search mechanisms, the 'local' and 'global', to locate the optimum solution. This means that each frog's behaviour is affected by its neighbours' behaviour. After the local and global search steps are complete, the memeplexes are shuffled, and the same search process is repeated till the convergence criteria are met [19].

SFLA has been used to solve discrete optimization problems [18] and engineering problems such as economic load dispatch problems [20], multi-objective optimal power flows [21], project management optimization problems [22], and the travelling salesman problem [23]. SFLA's speedy convergence capabilities are one of its most well-known benefits [15]. The reason for this fast convergence is because it incorporates the benefits of the memetic algorithm (MA) and the PSO algorithm (which is based on social behaviour) [24, 25]. The original SFLA consists of three evolutionary steps: local search, global search, and random search. The proposed modified SFLA incorporates three search parameters in addition to the existing mechanisms. These parameters are C_1, C_2, and w [26]. Introducing these parameters widens the algorithm's search capability and prevents premature convergence. This modified algorithm is as follows:

1. Generation of a randomized virtual frog population p.
2. Evaluation of the fitness of the population.
3. Sorting of the population in descending order of fitness values.
4. Division of the population into m number of memeplexes.
5. Here, the 'i'th frog is expressed as $X_i = (X_{i1}, X_{i2}, \ldots X_{is})$, Where s represents different variables.
6. Next, the worst frog X_w and best frog X_b are selected within each memeplex.
7. Then the global best frog X_g is selected among the entire population.
8. Local search is applied for each new generation as shown in Eq.(1)
$$X_{i+1} = w \times Xi + C_1 \times r \times (X_b - X_w) \quad (1)$$
9. If the fitness of the new frog generated by local search is superior to the previous frog, then it is replaced by the new frog. If not, apply Eq.(2)

The change in the position of the worst frog X_w is negligible when the difference between the worst and best frogs (X_b - X_w) becomes minute. This might trap the algorithm in a local optimum, resulting in premature convergence. Hence, a positive search acceleration factor (C_1) is introduced in the modified SFLA on the right-hand side of Eq. (1) [22]. Similarly, C_2 is introduced on the right-hand side of Eq. (2).

10. Apply Eq.(2) :
$$X_{i+1} = w \times X_i + C_2 \times r \times (X_g - X_w) \quad (2)$$

Where,

X_{i+1} = Newly generated position of the i^{th} frog
X_i = Previous position of the i^{th} frog
r = Randomly chosen value between 0 and 1
X_b = Position of the best frog in the memeplex
X_w = Position of the worst frog in the memeplex
X_g = Position of best frog in theentire population (global best frog)
w = Inertia weight
C_1 and C_2 = Positive search acceleration factors If the fitness value of the new frog generated using Eq. (2) is better than that of the previous frog, then the new frog replaces the old one; else, the worst frog is generated randomly.

11. Repeat the above steps and the frog shuffling process until convergence is achieved.

Pseudocode for modified SFLA is as follows:

Begin;
Set the frog population size (***p***);
Set the number of memeplexes (***m***);
Set the number of iterations;
Randomly generate the initial frog population ***p***;
For each individual, i ∈ p: Evaluate the fitness of the population;
Sort the frog population in terms of descending order of fitness;
Divide the frog population into n number of memeplexes;

For i=1 to the number of generations
For each memeplex;
Determine the X_w. X_b. X_g from the population;
Improve the position of the frog as Eq. (1) & (2) and random selection approach;
End;
Combine memeplexes;
Sort population p in terms of descending order of fitness;
Check if convergence criteria = true;
End;
End;

The flowchart of the modified SFLA is shown in Fig. 1. In the next section, three benchmark hole-drilling case studies have been solved using the modified SFLA.

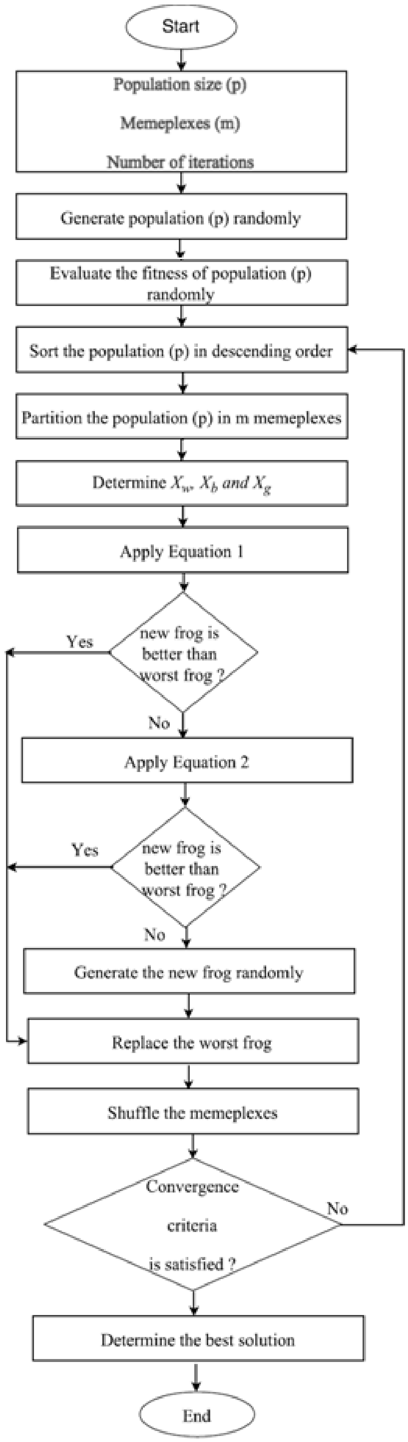

Fig. 1. Flowchart of modified SFLA.

4. Case Study

The case studies of a rectangular matrix of holes, namely, 4x5, 5x5, and 11x11 matrixes [2] are considered. The input details of the three matrixes are mentioned in Table 1.

Tab. 1. Input details of the case study [2].

	Case study		
	1	2	3
Rectangular matrix of holes	4x5	5x5	11x11
Total number of holes in the matrix	20	25	121
X-direction pitch in mm	100	100	100
Y-direction pitch in mm	50	50	50

The distance between any two holes in the rectangular matrix is calculated using Eq. (3):

$$D_{ij} = \sqrt{(U_i - U_j)^2 + (V_i - V_j)^2} \qquad (3)$$

Where,
D_{ij}=Distance between hole *i* and hole *j*
U_i= X direction co-ordinates of hole *i*
U_j= X direction co-ordinates of hole *j*
V_i = Y direction co-ordinates of hole *i*
V_j = Y direction co-ordinates of hole *j*

5. Results and Discussion

In this section the results obtained using the modified SFLA is compared with those obtained using GA and ACO [2] for the three case studies attempted.

The modified SFLA was coded in C++ using *Code Blocks* and run on a Windows 8 PC with an Intel Core i3 processor (1.90 GHz). The repair approach was used to handle constraints for all the case studies. The constraint employed was that the tool would visit each hole once, drill, and return to its initial position.

After conducting appropriate computational experiments, the following parameter values were selected for the 4x5 matrix:
C_1 = 0.85,
C_2 = 0.95,
w = 0.05,
Total frog population = 25
Number of memeplexes = 5
Number of frogs per memeplex = 5

A comparison of the results obtained for the 4x5 rectangular matrix is shown in Table 2.

Tab. 2. Comparison of results for the 4x5 rectangular matrix

	Case study of 4x5 Matrix holes		
Method	GA [2]	ACO [2]	Modified SFLA
Optimum path in mm	1300	1300	1300
No. of iterations	100	80	50
Population size	80	60	25

The optimal paths obtained using ACO and modified SFLA for the 4x5 matrix [2] are shown in Figures 2 (a) and 2 (b).

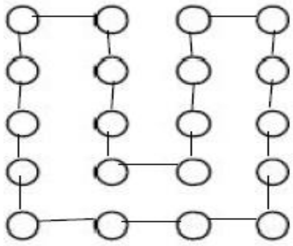

Fig. 2 (a). Optimal path obtained using ACO for 4x5 matrix [2].

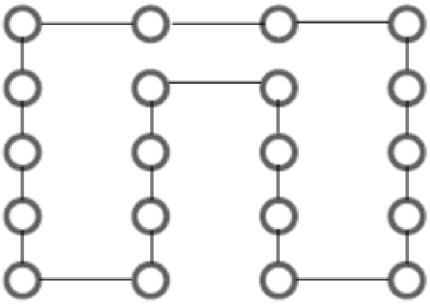

Fig. 2 (b). Optimal path obtained using modified SFLA for 4x5 matrix.

Similarly, computational experiments were carried out to obtain the algorithm-specific parameters for the modified SFLA for the matrix of 5x5.
C_1 = 0.05,
C_2 = 0.95,
w = 1.0,
Total frog population = 100
Number of memeplexes = 50
Number of frogs per memeplex = 2

A comparison of the results for the 5x5 rectangular matrix is shown in Table 3.

Tab. 3. Comparison of results for the 5x5 rectangular matrix

	Case study of 5x5 Matrix holes		
Method	GA [2]	ACO [2]	Modified SFLA
Optimum path in mm	1977	1730	1822
No. of iterations	100	80	48
Population size	80	60	100

The optimal paths determined using ACO and Modified SFLA for the 5x5 matrix [2] are shown in Figures 3 (a) and 3 (b).

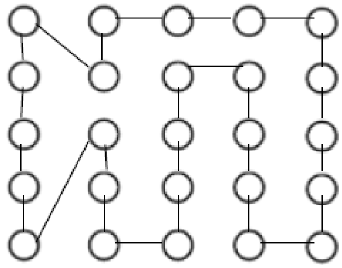

Fig. 3 (a). Optimal path obtained using ACO for 5X5 matrix [2]

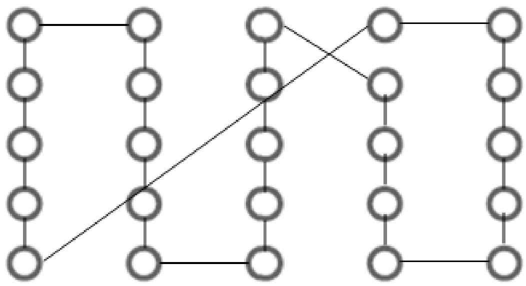

Fig. 3 (b). Optimal path obtained using modified SFLA for 5x5 matrix.

The following algorithm-specific parameters for the modified SFLA were obtained through various computational experiments for the 11x11 matrix.
$C_1 = 1.0$,
$C_2 = 0.95$,
$w = 0.05$,
Total frog population = 100
Number of memeplexes = 10
Number of frogs per memeplex =10
A comparison of the results obtained for the 11x11 rectangular matrix is shown in Table 4.

Tab. 4. Comparison of results for the 11x11 rectangular matrix.

	Case study of 11x11 Matrix holes		
Method	GA [2]	ACO [2]	Modified SFLA
Optimum path in mm	26074	11555	7618
No. of iterations	200	100	100
Population size	120	100	80

The optimal paths found by ACO and Modified SFLA for the 11x11 matrix [2] are shown in Figures 4 (a) and 4 (b).

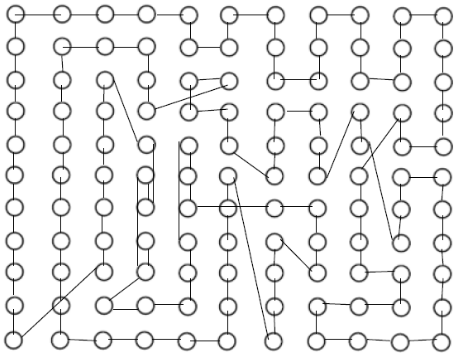

Fig. 4 (a). Optimal path obtained using ACO for 11X11 matrix [2].

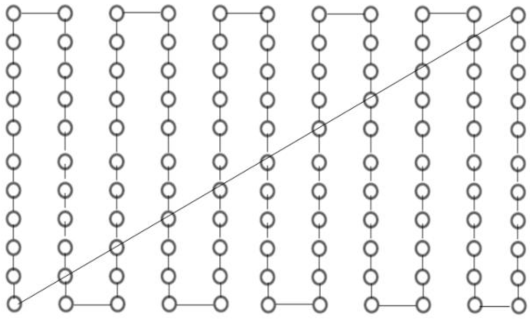

Fig. 4 (b). Optimal path obtained using modified SFLA for 11x11 matrix.

The convergence graphs for the 4x5, 5x5, and 11x11 rectangular matrices obtained using a modified SFLA are shown in Figures 5 (a - c).

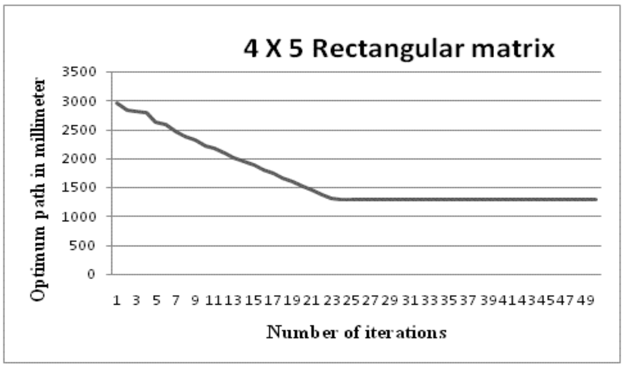

Fig. 5. a. Convergence graph for 4x5 rectangular matrix obtained using modified SFLA.

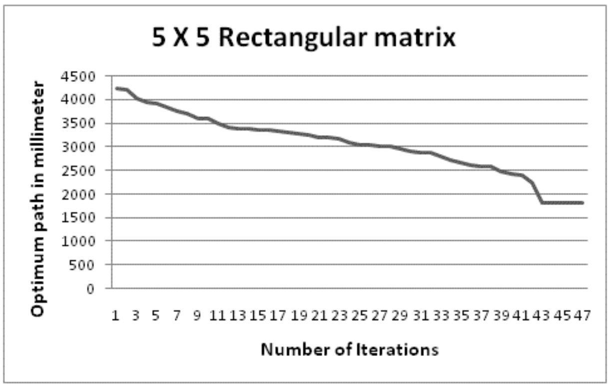

Fig. 5. b. Convergence graph for 5x5 rectangular matrix obtained using modified SFLA.

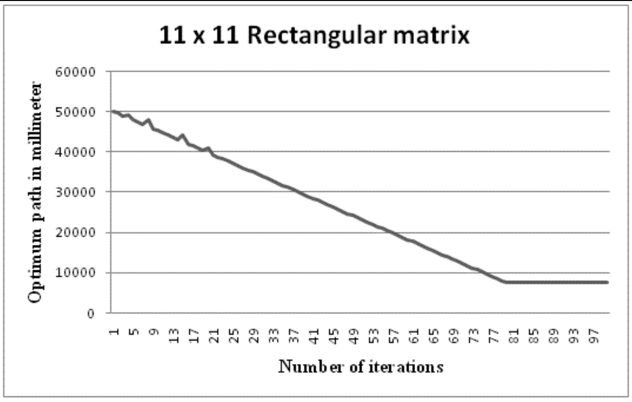

Fig. 5. c. Convergence graph for 11x11 rectangular matrix obtained modified SFLA.

The summary of the best paths for the 4x5, 5x5, and 11x11 rectangular matrices obtained using GA [2], ACO [2], and modified SFLA are shown in Figure 6.

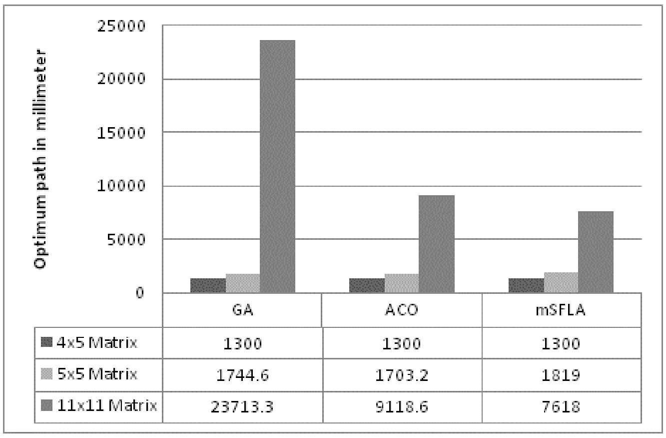

Fig. 6. Summary of best path obtained using GA [2], ACO [2], and modified SFLA.

A higher-dimensional problem of a 20x20 rectangular matrix of holes was attempted. The X and Y direction pitch is as given in table number 1. The various computational experiments carried out using modified SFLA as shown in Table 5.

Tab. 5. 20x20 matrix experiment using Modified SFLA.

w	C_1	C_2	Number of frogs	Number of frogs per meme-plex	Number of memeplexes	Number of iterations	Fitness function value $f(x)$
1.0	0.95	0.05	25	5	5	200	39900
0.05	0.95	1	25	5	5	200	22800

From Table 5, the minimum fitness function value obtained is 22800.

6. Conclusion

For hole-making process optimization involving many machine tool sequences depending on the hole's location, it is essential to follow an optimal sequence of machining operations so that the total tool travel distance is minimized. This work proposes a modified shuffled frog leaping algorithm to solve this problem.

Three parameters (C_1, C_2, and w) were introduced into the original SFLA so that premature convergence of the algorithm is avoided, whilst also widening its search capability. The proposed algorithm was applied to three case studies of a 4x5, 5x5, and 11x11 rectangular matrix of holes. The objective in all three cases is to reduce the overall tool travel distance during the drilling operations. The results obtained using modified SFLA were compared with those obtained using GA & ACO in previous literature.

According to these results, modified SFLA achieved superior results for the 4x5 and 11x11 rectangular matrices compared to GA & ACO [2]. However, for the 5x5 rectangular matrix, the results are slightly inferior to ACO [2] but superior to GA [2]. Additionally, a higher-dimensional problem of a 20x20 rectangular matrix of holes was attempted using modified SFLA. It was observed that for higher-dimensional problems, the modified algorithm works well. This signifies the potential of the modified SFLA in solving real-life industrial problems like toolpath optimization for hole-making operations. A similar study can be carried out in the future to optimize the machining time for hole-making operations.

References

[1] R.L.Merchant, World trends and prospects in manufacturing technology, *International Journal for Vehicle Design*, Vol. 6, (1985), pp. 121-138.
https://doi.org/10.1504/IJVD.1985.061334

[2] T. A. Adel, F. A.Mohamed, H. Karim, Optimum drilling path planning for a rectangular matrix of holes using ant colony optimization: International Journal of Production Research, Vol. 49, No. 19, (2011), pp. 5877-5891.
https://doi.org/10.1080/00207543.2010.507608

[3] A.T.Abbas, S.M. Megahed, A general algorithm for drilling holes lying in a matrix, *Robotics and Computer-Integrated Manufacturing*, Vol. 21, (2005), pp. 235-239.
https://doi.org/10.1016/j.rcim.2004.08.001

[4] S.Satishkumar, P. Asokan, Selection of optimal conditions for CNC multi-tool drilling system using non-traditional techniques, *Int. J. Machining and Machinability of Materials*, Vol. 3, Nos. 1-2, (2008), pp. 190-207.
https://doi.org/10.1504/IJMMM.2008.017633

[5] Y.Z.Guang, B.Z. Wei, Drilling path optimization by the particle swarm optimization algorithm with global convergence characteristics, *International Journal of Production Research*, Vol. 46, No. 8, (2008), pp. 2299-2311.
https://doi.org/10.1080/00207540601042480

[6] M.R.Alam, K.S Lee, M.Rahman, Process planning optimization for the manufacture of injection moulds using a genetic algorithm, *Int J Computer Integrated Manufacturing* Vol. 16, No. 3, (2003), pp. 181-191.
https://doi.org/10.1080/0951192021000025742

[7] J.A.Qudeiri, Y.Hidehiko, Optimization of Operation Sequence in CNC Machine Tools Using Genetic Algorithm, *J Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 1, No. 2, (2007), pp. 272-282.
https://doi.org/10.1299/jamdsm.1.272

[8] H.Ghaiebi, M. Solimanpur, An ant algorithm for optimization of hole-making operations, *Computers, and Industrial Eng*, Vol. 52, No. 2, (2007), pp. 308-319.
https://doi.org/10.1016/j.cie.2007.01.001

[9] Y.C.Hsieh, Y.C. Lee, P.S.You, Using an Effective Immune Based Evolutionary Approach for the Optimal Operation Sequence of Hole-Making with Multiple Tools, *J of Compt Information Syst*, Vol. 7, No. 2, (2011), pp. 411-418.

[10] M.Tamjidy, P.Shahla, Biogeography-based optimization (BBO) algorithm to minimize the non-productive time during hole-making process, Int J Production Research, Vol. 53, No. 6, (2015), pp. 1880-1894.
https://doi.org/10.1080/00207543.2014.965356

[11] Y.Marinakis, A.Marinaki, Bumble Bees Mating Optimization algorithm for the Open Vehicle Routing Problem, *Swarm and Evolutionary Comput*, (2014), pp. 1580-94.
https://doi.org/10.1016/j.swevo.2013.12.003

[12] S.Diyaley, A. Aditya, S. Chakraborty, Optimization of the multi-hole drilling path sequence for concentric circular patterns. *OPSEARCH*, Vol. 57, (2020), pp. 746-764.
https://doi.org/10.1007/s12597-020-00441-0

[13] A.Karthikeyan, R. Venkatesh, Machine learning in optimization of multi-hole drilling using a hybrid combinatorial IGSA algorithm. *Concurrent Engineering*, Vol. 28, No. 2, (2020), pp. 130-141.
https://doi.org/10.1177/1063293X20908318

[14] R.Venkata Rao, Modeling and optimization of Modern Machining

processes, Springer series in advanced manufacturing, (2011).
https://doi:10.1007/978-0-85729-015-1

[15] E.Elbeltagi, H. Tarek, G. Donal, Comparison among five evolutionary-based optimization algorithms, *Advanced Engineering Informatics*, Vol. 19, (2005), pp. 43-53.
https://doi.org/10.1016/j.aei.2005.01.004

[16] T.Niknam,H.D. Mojarrad, H.Z. Meymand, B.B.Firouzi, A new honey bee mating optimization algorithm for non-smooth economic dispatch, *Energy*, Vol. 36, No. 2, (2011), pp. 896-908.
https://doi.org/10.1016/j.energy.2010.12.021

[17] P.K.Ammu, K.C. Sivakumar,R. Rejimoan, Biogeography-Based Optimization - A Survey, *International Journal of Electronics and Computer Science Engineering*, Vol. 2, No. 1, (2012), pp. 154-160.

[18] M.M.Eusuff, K.E. Lansey, F. Pasha, Shuffled frog-leaping algorithm, a memetic metaheuristic for discrete optimization, *Eng. Optim*, Vol. 38, No. 2, (2006), pp. 129-154.
https://doi.org/10.1080/03052150500384759

[19] P.Luo, L.U. Qinang, W.U. Chenxi, Modified shuffled frog leaping algorithm based on new searching strategy, Seventh International Conference on Natural computation (2011).
https://doi. 10.1109/ICNC.2011.6022273

[20] P.Roy, Roy, Pritam, A. Chakrabarti, Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect, *Applied Soft Computing*, Vol. 13, (2013), pp. 4244-4252.
https://doi.org/10.1016/j.asoc.2013.07.006

[21] T.Niknam, M.R.Narimani, M. Jabbari, A.R.Malekpour, A modified shuffled frog leaping algorithm for multi-objective optimal power flow, *Energy*, Vol. 36, (2011), pp. 6420-6432.
https://doi.org/10.1016/j.energy.2011.09.027

[22] E.Elbeltagi,H. Tarek, G. Donald, A modified shuffled frog-leaping optimization algorithm, applications to project management, *Structure and Infrastructure Engineering*, Vol. 3, No. 1, (2007), pp. 53 -60.
https://doi.org/10.1080/15732470500254535

[23] X.H.Luo, Y. Yang, X. Li, Solving TSP with shuffled frog-leaping algorithm, *Proc. ISDA*, Vol. 3, (2008), pp. 228-232.
https://doi. 10.1109/ISDA.2008.346

[24] J.Kennedy, R.C. Eberhart, Particle swarm optimization, Proc. IEEE Conf. Neural Network, Vol. 4, (1995), pp.1942-1948.
https://doi. 10.1109/ICNN.1995.488968

[25] T.H.Huynh, A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers, In, Proceedings of the ICIT, (2008),pp. 1-6.
https://doi. 10.1109/ICIT.2008.4608439

[26] A.M.Dalavi, P.J.Pawar, T.P.Singh, Tool path planning of hole-making operations in ejector plate of injection mould using modified shuffled frog leaping algorithm, *Journal of Computational Design and Engineering*, Vol. 3, No. 3, (2016), pp. 266-273.
https://doi.org/10.1016/j.jcde.2016.04.001