DOI: 10.31534/engmod.2019.2-4.ri.01v
Original scientific paper
Received: 04.06.2019.

Determination of optimal tool path in drilling operation using Modified Shuffled Frog Leaping Algorithm

A.M. Dalavi[(1)], **P.J. Pawar**[(2)], **T.P. Singh**[(1)]

[(1)] *Department of Mechanical Engineering, Symbiosis Institute of Technology, Symbiosis International University, Gram Lavale, Mulshi, Pune, INDIA*
e-mail: amol.dalavi83@gmail.com

[(2)] *Department of Production Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik, INDIA*
e-mail: pjpawar1@rediffmail.com

SUMMARY

Applications like boilerplates, food-industry processing separator, printed circuit boards, drum and trammel screens, etc. consists of a matrix of a large number of holes. The primary issue involved in hole-making operations is a tool travel time. It is often necessary to find the optimal sequence of operations so that the total processing cost of hole-making operations can be minimized. In this work, therefore an attempt is made to reduce the total tool travel of hole-making operations by applying a relatively new optimization algorithm known as modified shuffled frog leaping for determining the optimal sequence of operations. Modification is made in the existing shuffled frog-leaping algorithm by introducing three parameters with their positive values to widen the search capability of existing algorithms. A case study of the printed circuit board is considered in this work to demonstrate the proposed approach. Obtained results of optimization using modified shuffled frog leaping algorithm are compared with those obtained using particle swarm optimization, firefly algorithm and shortest path search algorithm.

KEY WORDS: *Modified Shuffled Frog Leaping Algorithm; tool path; hole-making operations.*

1. INTRODUCTION

Growing revenues and cutting costs is often a long term goal of any industry, and it is no surprise that successful manufacturers focus on the tried and tested basics in operations; but without committing to continuous research and product innovation it is difficult to expand business in a vastly growing market. In this scenario, the optimization of operational processes becomes the key to success in the manufacturing industry. Specifically, as the manufacturing industry accounts for a significant share of the industrial sector in developed countries, the importance of optimization is evident and need not be overemphasized.

The typical manufacturing industry is engaged in activities based on the fabrication, processing, or preparation of products from raw materials and products. Out of several operations that may be performed on the shop floor, hole-making is one very frequently used operation. Optimization of hole-making operations in manufacturing industries, therefore, plays a vital role.

2. LITERATURE REVIEW

Sequence optimization of drilling operations in computer numerically controlled (CNC) machine is essential. It is required to minimize the total non-productive tool traveling cost of hole-making operations. Typically, *70%* of the overall time in manufacturing processes is spent on the tool and part movements [1]. Genetic algorithm (GA) was used for drilling holes in the matrix [2]. A methodology has been presented to determine the optimum cutting parameters of CNC multi-tool drilling system using non-traditional optimization techniques such as genetic algorithm, simulated annealing, and ant colony optimization algorithm (ACO) [3]. Particle swarm optimization (PSO) was used for finding an optimal drilling path. Ant colony optimization algorithm was used for finding optimal drilling operations paths of a rectangular matrix of holes [4].

Case study of injection mould was presented with the aim of achieving minimum total processing time of machining using genetic algorithm (GA) and compared with the results obtained using GA with simulated annealing (SA) [5]. Genetic algorithm was used to obtain a concise, cutting tool path for machine operations [6]. Case study presented the application of the ant colony optimization (ACO) algorithm for achieving optimal path of machining holes in a typical industrial part. Six benchmark problems were attempted in order to validate the performance of their ACO algorithm and compare ACO results with dynamic programming (DP) [7]. Optimal sequence of hole-making operations where various tools were required to obtain the desired size of hole, minimized the non-productive tool travel time and tool switch time, using an immune based evolutionary approach (IA) and its results were compared with results obtained using ACO and PSO [8]. The evolutionary optimization algorithm was used to reduce the tool travel and tool switching time during hole-making operations based on geographic classification of biological organisms. The performance of their proposed algorithm was validated based on the test functions adopted from in literature [9]. Bumble bees mating optimization algorithm was used for the open vehicle routing problem, i.e. to reduce vehicle travel distance. Two benchmark problems were considered in order to validate their proposed algorithm [10]. Minimization of tool travel in hole-making operations is similar to the travelling salesmen's (TSP) problems, in which tool must inspect each hole location, perform drilling operations and return to its original location. Problem of optimization of hole-making operations becomes more difficult as the increase in number of holes. Hence it is necessary to determine the optimum sequence of drilling operations, in order to reduce the non-productive tool travel time and thus reduce its cost.

In the previous literature related to this area the non-traditional optimization methods such as Tabu search, genetic algorithm, particle swarm optimization, ant colony algorithm, immune algorithm, bumble bees mating optimization algorithm and biogeography based optimization (BBO) algorithm etc. were used in optimization of hole-making operations. For instance, pure tabu search uses only one solution, ignoring some promising areas of the search space, and as a result, they may not find the optimal solution. Genetic Algorithm gives near optimal solution for complex problems [11]. GA as well as IA requires several parameters [12]. GA requires crossover, mutation and population etc. parameters and IA requires population size, the maximum number of generations (iterations), threshold value, probability of mutation rate and the total number of antibodies etc. parameters.

In ACO algorithm convergence rate is slow due to pheromone evaporation and CPU time requirement is very high [12]. Honey Bees Mating Optimization algorithm may miss the optimum and provide a near optimum solution in a limited runtime [13]. Biogeography-based optimization (BBO) is poor in exploiting the solutions. In addition, there is no provision for selecting the best members from each generation [14]. It is therefore necessary to use non-traditional optimization algorithm, which is robust and gives the correct solution for complex problems [11]. This paper discusses the possibility of reducing the total tool travel time in drilling operations on printed circuit board by implementing modifications in the recently developed Shuffled Frog Leaping algorithm.

3. MODIFIED SHUFFLED FROG LEAPING ALGORITHM (mSFLA)

The Shuffled Frog Leaping Algorithm is a meta-heuristic optimization technique originated by Eusuff and Lansey, similar to the conduct of a group of frogs while searching for the maximum amount of food sites [15]. It consists of a random number of frogs called 'population', which is further divided into different parts called 'memeplexes'. Each individual frog performs two mechanisms called 'local' and 'global' search mechanisms to get optimum solution. Through these two mechanisms, behavior of an individual frog is influenced by the neighbouring one in order to obtain the best solution. Thereafter, the frog population is shuffled and the local and global search mechanisms are carried out until the convergence criteria are achieved [16].

Shuffled Frog Leaping Algorithm can be used for discrete optimization problems [15]. It has been successfully applied in several engineering optimization problems such as the economic load dispatch problem [17], multi-objective optimal power flow [18], project management [19] and traveling salesman problem [20]. The most well-known benefit of SFL algorithm is its fast convergence speed [15]. It combines advantages of both, the genetic-based Memetic Algorithm (MA) and the social behavior-based PSO algorithm [21], [22]. Existing SFL algorithm consists of three evolution steps, namely, local search, global search and random search. In the proposed modified SFL algorithm, along with these three evolution steps, three additional search parameters, i.e., w, C_1 and C_2 are introduced. This actually widens search capability and avoids premature convergence. This modified algorithm is shown below:

- Generate randomly virtual frog population called p.
- Estimate the fitness of the population.
- Group the population in a descending manner.
- Divide the population in m memeplexes.
- Frogs i is expressed as X_i = (X_{i1}, X_{i2}, ...X_{is}), where 's' stands for the number of variables.
- Select the worst frog X_w and the best frog X_b within each memeplexes.
- Select the global best frog X_g in the entire population.
- Apply the local search for new generations by following Eq. (1):

$$X_{i+1} = w \cdot X_i + C_1 \cdot r \cdot \left(X_b - X_w\right) \tag{1}$$

If fitness of new frog generated by above Eq. (1) is better than previous frog then replace it with the new frog. If not, apply the Eq. (2).

When the difference between the worst frog X_w and the best frog X_b becomes small, change in frog X_w's position will be very small, hence it might stuck in a local optimum and result into premature convergence. To avoid such an event, in the right hand side of Eq. (1) search acceleration factor with positive values of C_1 is introduced (Eusuff et al. [15]). Similarly, C_2 is introduced in the right hand side of Eq. (2):

$$X_{i+1} = w \cdot X_i + C_2 \cdot r \cdot \left(X_g - X_w \right) \tag{2}$$

Where:

X_{i+1} - new position of the frog,

X_i - previous position of frog,

r - random number values between 0 and 1,

X_b - position of best frog among the memeplexes,

X_w - position of worst frog among the memeplexes,

X_g - position of global best frog in search space which is the best among all frogs,

w - inertia weight,

C_1 - search acceleration factor for local search with positive values,

C_2 - search acceleration factor for global search with positive values.

If fitness of new frog generated by above Eq. (2) is better than previous frog then replace it with the new frog, else replace randomly the worst frog.

Continue search and the frog shuffling process until the convergence criteria are satisfied.

The optimum selection of the algorithm operating parameters like acceleration constants 'C_1 and 'C_2' and inertia coefficient 'w', is essential for the convergence of the algorithm. Bergh and Engelbrecht [23] provided the effect of parameters w, C_1 and C_2 on convergence for standard numerical benchmark functions. To ensure the convergence of the algorithm, the following Equation must be satisfied [23]:

$$max\left(|\lambda_1|, |\lambda_2| \right) < 1 \tag{3}$$

Where:

$$\lambda_1 = \frac{1 + w - \phi_1 - \phi_2 + \gamma}{2} \tag{4}$$

$$\lambda_2 = \frac{1 + w - \phi_1 - \phi_2 - \gamma}{2} \tag{5}$$

Where: $\gamma = \sqrt{\left(1 + w - \phi_1 - \phi_2 \right)^2 - 4w}$, $\phi_1 = C_1 \cdot r_1$ and $\phi_2 = C_2 \cdot r_2$.

As the feasible range for inertia factor 'w' is 0-1 and for C_1 and C_2 is 0-1, the selected values of w, C_1 and C_2 should be such that the Eq. (3) is satisfied for all possible values of random numbers r_1 and r_2 in the range 0-1.

Initially the population (frogs) is generated randomly, say e.g., number of frog=*100*. Product of number of memeplex and subfrogs should equal to the total population of frogs. Computational experiments with total population of *100* and various parameter values are presented in Table 1.

The flow chart of modified Shuffled Frog Leaping Algorithm is shown in Figure 1.

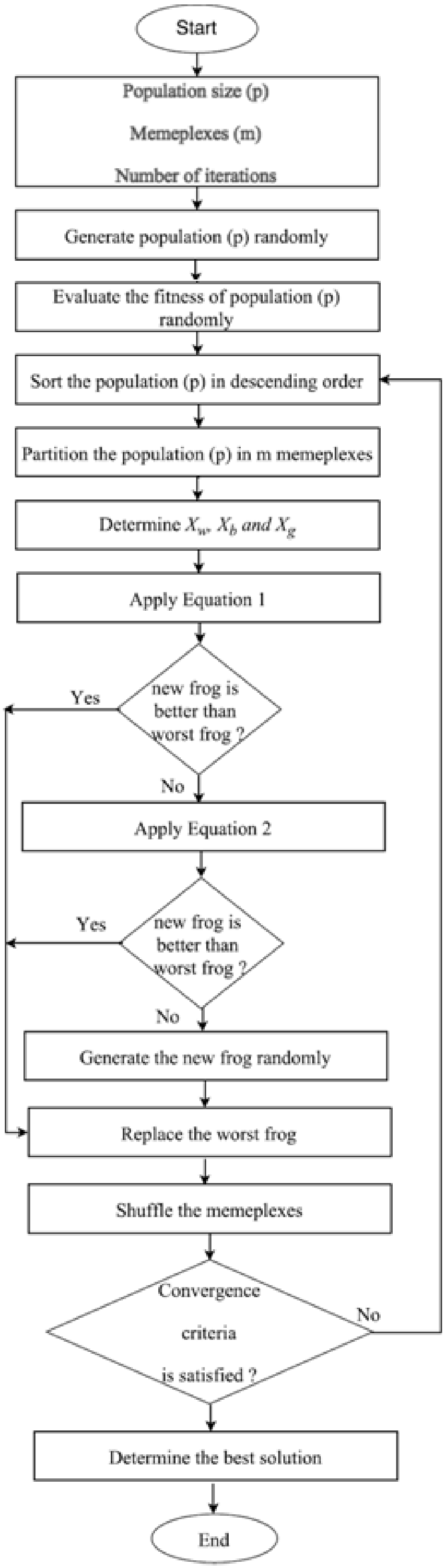
Start
Population size (p)
Memeplexes (m)
Number of iterations
Generate population (p) randomly
Evaluate the fitness of population (p) randomly
Sort the population (p) in descending order
Partition the population (p) in m memeplexes
Determine X_w, X_b and X_g
Apply Equation 1
new frog is better than worst frog ?
Yes
No
Apply Equation 2
new frog is better than worst frog ?
Yes
No
Generate the new frog randomly
Replace the worst frog
Shuffle the memeplexes
Convergence criteria is satisfied ?
No
Determine the best solution
End

Fig. 1 *Flowchart of modified shuffled frog leaping algorithm*

4. CASE STUDY

Case study of printed circuit board consisting of 14 holes of *3 mm* diameter is considered [24]. It is assumed that a single tool is required to perform the drilling operation. The details of holes location are shown in Figure 2.

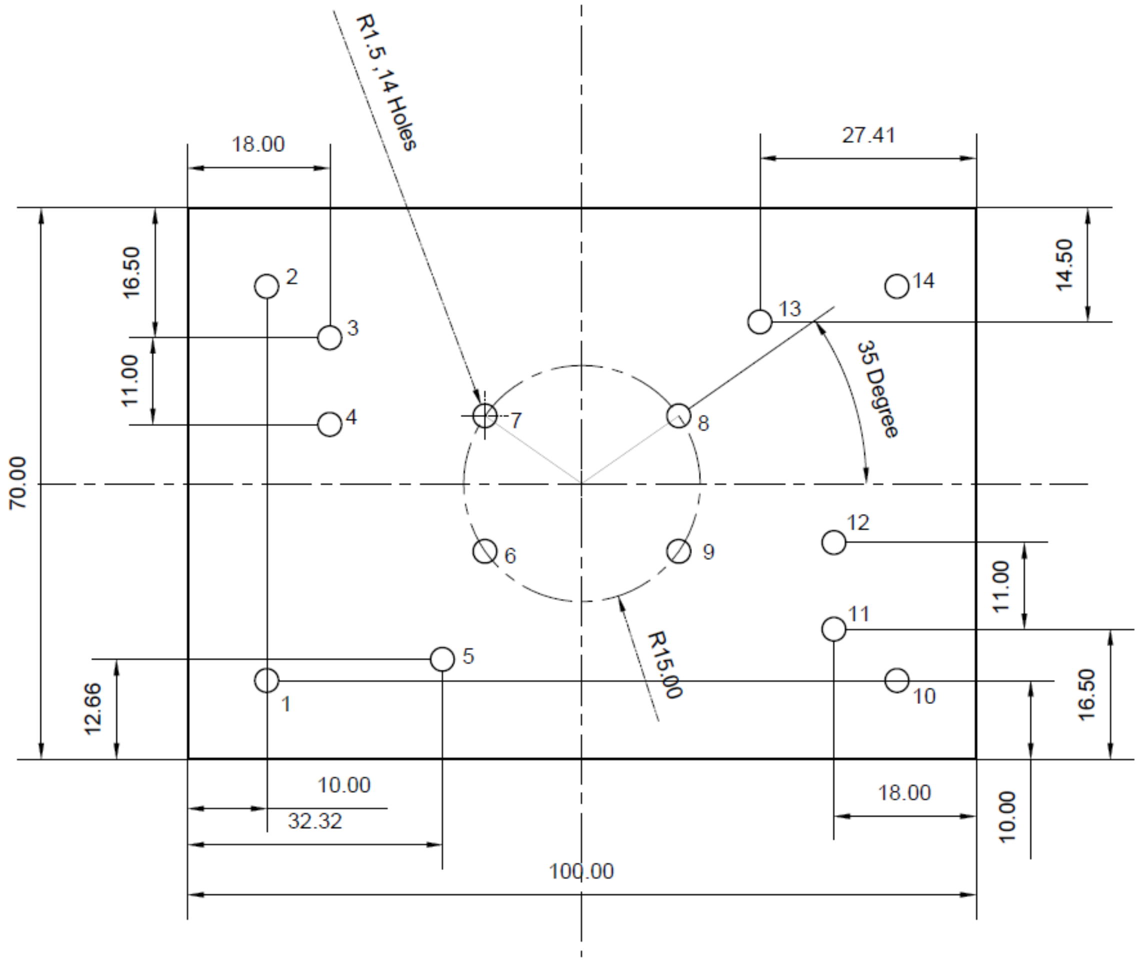

Fig. 2 *Layout of printed circuit board*

Sequence optimization of drilling operations of holes of printed circuit board is similar to the traveling salesmen problem. This problem takes in consideration that the tool has to visit each hole once and finally return to its original position.

The distance between any two holes can be calculated using the following Eq. (6):

$$D_{ij} = \sqrt{\left(U_i - U_j\right)^2 + \left(V_i - V_j\right)^2} \quad \textbf{(6)}$$

Where:

D_{ij} = Distance between hole *i* and hole *j,*

U_i and U_j = *X* direction co-ordinates of hole *i* and hole *j,*

V_i and V_j = *Y* direction co-ordinates of hole *i* and *j.*

Optimal drilling path for the case study using mSFLA is obtained using a number of frogs equal to *100*. Computational experiments are carried out using various parameter values and the optimization results are shown in Table 1.

5. RESULTS AND DISCUSSION

In this section obtained results of optimization of the modified Shuffled Frog Leaping algorithm are compared with those obtained using particle swarm optimization, firefly algorithm and shortest path search algorithm [24] for the case studies presented in the above section.

The modified Shuffled Frog Leaping algorithm was coded by the constraint that the tool will visit each hole once, perform the drilling operation, and return to its original position. In order to obtain the optimal sequence of drilling operations of printed circuit board are as shown in Figure 2 using Code blocks C++ and run on a windows 8 PC with intel core i3 CPU @ 1.90 GHz. While carrying out sequence optimization for the case study, constraint handling is performed by using the repair approach of constraint handling. Table 1 shows the results of optimization using mSFLA.

Objective function Eq. (7) can be used for above computational experiments being as follows:

$$Min\,D = \sum_{i=1}^{m} D_{ij} \tag{7}$$

Where:

D = Optimal distance to complete the drilling of all holes on PCB,

m = Number of possible operations in sequence.

From Table 2, it is seen that modified shuffled frog leaping algorithm results are better than other algorithms considered for this case study. Optimal sequence for the printed circuit board considered is *2, 3, 7, 4, 1, 5, 6, 9, 8, 13, 14, 12, 11, 10* and corresponding optimal length is *250* for the parameters shown in Table 1. Three parameters C_1, C_2, and w help to widen the search capability of the modified algorithm and hence possibility of achieving better solution.

Table 1 *Computational experiments using mSFLA*

Number of frogs	*Number of memeplexes*	*Number of sub frogs*	w	C_1	C_2	*Number of iterations*	*Optimal path, mm*
100	10	10	1	0.85	1	40	282
100	50	2	1	0.85	1	100	257
100	2	50	1	0.85	1	40	279
100	25	4	1	0.85	1	35	256
100	4	25	1	0.85	1	25	275
100	20	5	1	0.85	1	20	265
100	5	20	1	0.85	1	40	290
100	10	10	1	0.85	0.05	100	257
100	50	2	1	0.85	0.05	70	262
100	2	50	1	0.85	0.05	100	268
100	25	4	1	0.85	0.05	100	252
100	4	25	1	0.85	0.05	100	273
100	20	5	1	0.85	0.05	150	252
100	5	20	1	0.85	0.05	250	250
100	10	10	0.05	0.85	1	50	283
100	50	2	0.05	0.85	1	50	309
100	10	10	1	0.95	1	40	277
100	50	2	1	0.95	1	70	254
100	2	50	1	0.95	1	40	290
100	25	4	1	0.95	1	50	263
100	10	10	1	0.95	0.05	250	262
100	50	2	1	0.95	0.05	100	262
100	2	50	1	0.95	0.05	100	263
100	25	4	1	0.95	0.05	100	250
100	4	25	1	0.95	0.05	100	267
100	20	5	1	0.95	0.05	100	250
100	5	20	1	0.95	0.05	250	250
100	10	10	0.05	0.95	1	10	329
100	50	2	0.05	0.95	1	20	283
100	2	50	0.05	0.95	1	40	328
100	25	4	0.05	0.95	1	50	309

Convergence graph for iterations in Table 1 is as shown in Figure 3.

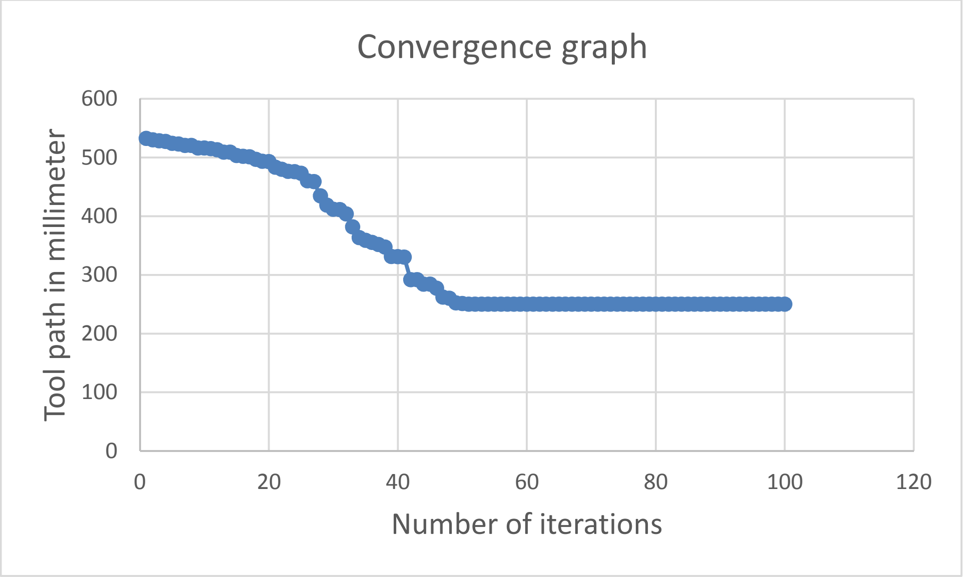

Fig. 3 *Convergence graph*

In this paper, case study of PCB is used [24]. Same case study of PCB has been used by authors mentioned in Table 2, using different algorithm.

Results obtained by modified SFLA for the case study of PCB is compared with results obtained by Zhu [24], Adam et al. [5], Othman et al. [27] and Mohd Muzafer et al. [26] for the same case study in Table 2.

Table 2 *Comparison of results*

	Swarm Intelligent Algorithm [24]	*PSO [5]*	*Binary PSO [27]*	*Magnetic optimization algorithm [26]*	*mSFLA*
Length of optimal solution	*280*	*280*	*280*	*280*	*250*

6. CONCLUSION

In the optimization of hole-making operations which involves large number of machining tool sequences to complete the hole-making operations depending on location of the hole on the part, it is necessary to achieve optimal sequence of machining operations to minimize the total tool travel distance. This paper proposes recently developed modified shuffled frog leaping algorithm. In existing shuffled frog leaping algorithm three parameters C_1, C_2, w are introduced to avoid premature convergence and widen the search capability. In this paper, the proposed modified SFL algorithm is applied on case study of printed circuit board to reduce the overall tool travel distance in drilling operations. The obtained result using modified shuffled frog leaping algorithm is compared with results obtained using particle swarm optimization, firefly algorithm, and shortest path search algorithm [24]. It can be seen that the result of optimization achieved by modified Shuffled Frog Leaping algorithm is *10.71%* better than particle swarm optimization, firefly algorithm and shortest path search algorithm. The improvement obtained by the modified Shuffled Frog Leaping algorithm is thus significant, and clearly indicates the potential of this method to solve real life problems related to hole-making operations in various industrial applications.

7. REFERENCES

[1] R.L. Merchant, World trends and prospects in manufacturing technology, *International Journal for Vehicle Design*, Vol. 6, pp. 121–138, 1985.

[2] A.T. Abbas, S.M. Megahed, A general algorithm for drilling holes lying in a matrix, *Robotics and Computer-Integrated Manufacturing*, Vol. 21, pp. 235–239, 2005.

https://doi.org/10.1016/j.rcim.2004.08.001

[3] S. Satishkumar, P. Asokan, Selection of optimal conditions for CNC multitool drilling system using non-traditional techniques, *Int. J. Machining and Machinability of Materials*, Vol. 3, No. 1-2, pp. 190-207, 2008. https://doi.org/10.1504/IJMMM.2008.017633

[4] 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, pp. 5877-5891, 2011.

https://doi.org/10.1080/00207543.2010.507608

[5] A. Adam, A.F.Z. Abidin, Z. Ibrahim, A.R. Husain, Z.M. Yusof, I.A. Ibrahim, A particle swarm optimization approach to Robotic Drill route optimization, *In 2010 fourth Asia international conference on mathematical/analytical modeling and computer simulation,* pp. 60-64, IEEE, 2010. https://doi.org/10.1109/AMS.2010.25

[6] A.J. Qudeiri ,H. Yamamoto, R. Ramli, Optimization of operation sequence in CNC machine tools using genetic algorithm, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, Vol. 1, No. 2, pp. 272-282, 2007.

https://doi.org/10.1299/jamdsm.1.272

[7] H. Ghaiebi, M. Solimanpur, An ant algorithm for optimization of hole-making operations, *Computers & Industrial Engineering*, Vol. 52, No. 2, pp. 308-319, 2007.

https://doi.org/10.1016/j.cie.2007.01.001

[8] 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, *Journal of Computational Information Systems*, Vol. 7, No. 2, pp. 411-418, 2011.

[9] M. Tamjidy, S. Paslar, B.T. Hang Tuah Baharudin, T.S. Hong, M.K.A. Ariffin Biogeography based optimization (BBO) algorithm to minimise non-productive time during hole-making process, *International Journal of Production Research,* Vol. 53, No. 6, pp. 1880-1894, 2015. https://doi.org/10.1080/00207543.2014.965356

[10] Y. Marinakis, Marinaki, A Bumble Bees Mating Optimization algorithm for the Open Vehicle Routing Problem, *Swarm and Evolutionary Computation*, Vol. 15, pp. 80–94, 2014. https://doi.org/10.1016/j.swevo.2013.12.003

[11] V. Rao, Modeling and Optimization of Machining Processes, *In Advanced Modeling and Optimization of Manufacturing Processes*; Springer Series in Advanced Manufacturing, Chapter 2, Springer: London, UK, 2011. https://doi.org/10.1007/978-0-85729-015-1

[12] E. Elbeltagi, T. Hegazy, D. Grierson, Comparison among five evolutionary-based optimization algorithms, *Advanced engineering informatics*, Vol. 19, No. 1, pp. 43-53, 2005. https://doi.org/10.1016/j.aei.2005.01.004

[13] 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, pp. 896–908, 2011. https://doi.org/10.1016/j.energy.2010.12.021

[14] 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, pp. 154-160, 2012.

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

[16] P. Luo, L.U. Qinang, Wu. Chenxi, Modified shuffled frog leaping algorithm based on new searching strategy, *Seventh International Conference on Natural computation*, 2011.

https://doi.org/10.1109/ICNC.2011.6022273

[17] P. Roy, P. Roy, 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, pp. 4244–4252, 2013.

https://doi.org/10.1016/j.asoc.2013.07.006

[18] T. Niknam, M.R. Narimani, M. Jabbari, A.R. Malekpour, A modified shuffle frog leaping algorithm for multi-objective optimal power flow, *Energy*, Vol. 36, pp. 6420-6432, 2011.

https://doi.org/10.1016/j.energy.2011.09.027

[19] 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, pp. 53–60, 2007. https://doi.org/10.1080/15732470500254535

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

[21] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948, 1995.

[22] T.H. Huynh, A modified shuffled frog leaping algorithm for optimal tuning of multivariable PID controllers, *In Proceedings of the ICIT*, pp. 1–6, 2008.

[23] F. Bergh, A.P. Engelbrecht, A study of particle swarm optimisation particle Trajectories, *Information sciences*, Vol. 176, No. 8, pp. 937-971, 2006.

https://doi.org/10.1016/j.ins.2005.02.003

[24] G.Y. Zhu, Drilling path optimization based on swarm intelligent algorithm, *In IEEE International Conference on Robotics and Biomimetic*, pp. 193-196, 2006.

https://doi.org/10.1109/ROBIO.2006.340357

[25] 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, pp. 2299–2311, 2008.

https://doi.org/10.1080/00207540601042480

[26] I. Mohd Muzafar, O. Mohd Azlishah, S. Hamzah Asyrani, M.S. Maizatul Alice, M. Mohd Harris, R. Ridza Azri, O. Khairuddin, Route planning analysis in holes drilling process using magnetic optimization algorithm for electronic manufacturing sector, *World Applied Sciences Journal 21 (Special Issue of Engineering and Technology),* pp. 91-97, 2013.

[27] M.H. Othman, A.F.Z. Abidin, A. Adam, Z.M Yusof, Z. Ibrahim, S.M. Mustaza, L.Y. Yang, A binary particle swarm optimization approach for routing in PCB holes drilling process, *In First International Conference on Robotic Automation System*, pp. 201-206, 2011.