OPTIMAL SEQUENCE OF HOLE-MAKING OPERATIONS USING PARTICLE SWARM OPTIMISATION AND SHUFFLED FROG LEAPING ALGORITHM

Amol M. Dalavi[1*] - Padmakar J. Pawar[2] - Tejinder P. Singh[1]

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

ARTICLE INFO

***Article history*:**
Received: 14.5.2015.
Received in revised form: 28.11.2015.
Accepted: 1.12.2015.

Keywords:
Injection mould
Hole-making operations
Particle swarm optimisation
Shuffled frog leaping algorithm

Abstract:

Tool travel and tool switch scheduling are two major issues in hole-making operations. It is necessary to find the optimal sequence of operations to reduce the total processing cost of hole-making operations. In this work therefore, an attempt is made to use both a recently developed particle swarm optimisation algorithm and a shuffled frog leaping algorithm demonstrating in this way an example of plastic injection mould. The exact value of the minimum total processing cost is obtained by considering all possible combinations of sequences. The results obtained using particle swarm optimisation and shuffled frog leaping algorithm are compared with the minimum total processing cost results obtained by considering all possible combinations of sequences. It is observed that the results obtained using particle swarm optimisation and shuffled frog leaping algorithm are closer to the results of the minimum total processing cost obtained by considering all possible combinations of sequences presented in this work. This clearly shows that particle swarm optimisation and shuffled frog leaping algorithm can be effectively used in optimisation of large scale injection mould hole-making operations.

1 Introduction

In the process of machining several industrial parts such as dies and moulds, operations like drilling, reaming and tapping account for a huge segment of processing. Usually a part, for e.g., a plastic injection mould, may have several holes of different diameters, surface finishes, and maybe different depths. Different combinations of tools can be used to drill a hole, which consists of a pilot tool, one or more intermediate tools and a final tool to achieve the final hole size. E.g., for drilling the hole H_3, shown in Fig. 1, there could be four different combinations of tools; $\{T_1, T_2, T_3\}$, $\{T_1, T_3\}$, $\{T_2, T_3\}$, and $\{T_3\}$. Tool travel, tool switch and tooling & machining cost is directly influenced by combinations of tools used for machining a hole [1]. Tool travel takes a considerable

* Corresponding author. Tel.: +91-9921517645
E-mail address: amol.dalavi83@gmail.com,

amount of time, as a result of the point-to-point machining aspect in hole-making.

Kolahan and Liang [1] report a tabu-search (TS) approach to reduce the total machining cost for hole-making operations. In order to reduce the total processing cost, the correct sequence of operations and the related machining speeds used to carry out every operation are important.

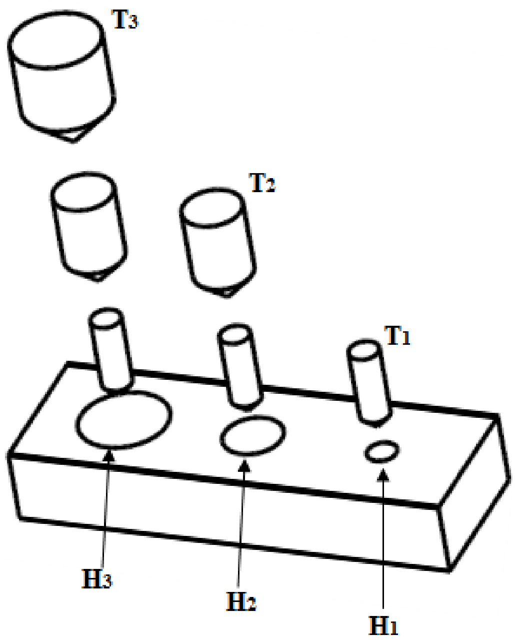

Figure 1. Diagram showing various combinations of tool used for machining the holes [1].

In machining processes, it takes more machining time for tool switching and table movement from one position to another. . Current industry scenario for machining a hole is to use the same tool required for all possible holes, which increases the tool travel cost. On the other side, Carrying out all drilling, enlarging and if required tapping or reaming operations on each hole at a time increases tool switch cost. Luong and Spedding [2] report process scheduling in hole-making operations using a generic knowledge based method. Qudeiri and Hidehiko [3] introduced a genetic algorithm (GA) to achieve the least cutting path possible. Ghaiebi et al. [4] applied the proposed ant colony optimisation (ACO) algorithm for optimizing the sequence of hole-making operations in an industrial part. Hsieh et al. [5] investigated the optimal sequence of hole-making operations using an immune based evolutionary approach. Alam et al. [6] presented a practical application of a computer-aided process planning method to reduce the overall machining time of injection moulds using genetic algorithm. Tamjidy et al. [7] presented an evolutionary algorithm to reduce tool travel and tool switching time during hole-making operations based on geographic distribution of a biological organism. Rajkumar and Annamalai [8] investigated assembly fixture design cost using the genetic algorithm. Dalavi et al. [9] used particle swarm optimisation to optimise hole-making operations for plastic injection moulding of upper holder. Srivatsava et al. [10] presented a firefly algorithm (FA) for achieving optimal test sequence generation. Marinakis Y and Marinaki M [11] used bumble bees mating optimisation (BBMO) algorithm for the open vehicle routing problem. Narooei et al. [12] used ACO algorithm for optimizing the tool path of case study involving multiple holes. Oscar et al. [13] presented a methodology to optimize the manufacturing time by using ACO. Liu et al. [14] used ACO algorithm for process planning optimisation of hole-making operations. Lim et al. [15] used a hybrid cuckoo search-genetic algorithm (CS-GA) for hole-making sequence optimisation. Lim et al. [16] used Cuckoo Search (CS) algorithm for optimisation of sequence in PCB Holes drilling process.

It is found in the literature related to this area that the advanced optimisation techniques such as tabu-search, genetic algorithm, ant colony optimisation, firefly algorithm, and immune based evolutionary approach were used in finding the optimal path of drilling operations. A frequently used optimisation method is genetic algorithm (GA) which requires more parameters [17]. Convergence of ACO algorithms is slow due to pheromone evaporation and due to high CPU time availability requirement [17]. Immune based evolutionary approach requires more parameters for solving optimisation problem and it is perhaps difficult to deal with multi-objective optimization problem. Hence, it is required to use an algorithm which gives more correct results [18]. Therefore, to reduce the total processing cost of an application example considered, an attempt is made by using particle swarm optimisation [19] and a shuffled frog leaping algorithm [20, 21]. The next section briefly describes formulation of an optimisation model.

2 Formulation of an optimisation model

With the objective of reducing the total processing cost of hole-making operations, the following optimisation model is formulated based on analysis given by Kolahan and Liang considering following components of total costs [1]:

a) Tool travel cost: It occurs when tool travels from one place to another place.
b) Tool switch cost: It occurs whenever a different tool is used for next operation. If tool type required for operation is not available on spindle, then the required tool must be loaded on the spindle prior to performing operation.

c) Tooling and machining costs: It includes the new tool cost and the cost of machine down time required to replace the tool. Machining cost comprises the operating cost and the machine overhead cost. The combined tooling and machining costs when tool type *m* is used on hole *n* can be expressed as Eq. (1):

$$Z_{mn} = \frac{t_{mn}}{T_{mn}} * Y_m + Z * t_{mn} \tag{1}$$

Where,
m, tool type index in ascending order according to the tool diameters, $m=1,...,M;$
n hole index, $n=1,...,N;$
mn, index for the previous tool to be used on hole *n;*
$Z_{mn,}$ collective tooling and machining costs when tool type *m* is used on hole *n;*
$t_{mn,}$ machining time necessary by tool *m* for hole *n;*
T_{mn} life of tool type *m* related with cutting operation on hole *n;*
$Y_{m,}$ cost of tool type *m;*
Z, machining cost per unit time.
Machining time, t_{mn}, is determined by Eq. (2):

$$t_{mn} = \frac{\pi * d_m * L_n}{1000 * U_{mn} * f_m} \tag{2}$$

Where,
$d_{m,}$ diameter of tool *m;*
$L_{n,}$ depth of hole *n*, with the clearance;
$U_{mn,}$ cutting speed of tool *m* related with an operation on hole *n;*
f_m ,suggested feed rate for tool type *m;*
In normal practice, depth of cut and feed rate of cutting speed in drilling operations is kept fixed and constant. Hence the optimum cutting speed, U_{mn}, for the constant feed rate can be obtained by solving the following differential Eq. (3):

$$\frac{dZ_{mn}}{dU_{mn}} = 0 \tag{3}$$

The cutting speed obtained from Eq. (3) reduces the sum of tooling and machining costs for a single operation.
Considering all aspects mentioned above, the final optimisation model can be expressed as given by Eqs. (4)- (6).

$$Min\ \ G(s) = Min \sum_{m\in M_n} \sum_{m'\in M_n} \sum_{m''\in M} \sum_{l=1}^{N} \sum_{n=1}^{N} \sum_{k=1}^{N} x_{mm'm''l\,nk} \left[a * \left(\frac{P_{ln} + p_{nk}}{2} \right) + b * q_{mm''n} + \frac{t_{mm'n}}{T_{mm'n}} * Y_m + t_{mm'n} * Z \right] \tag{4}$$

Subject to:

$$\sum_{m\in M_n} \sum_{m''\in M} \sum_{l=1}^{N} \sum_{k=1}^{N} x_{m_n m' n'' l\,nk} = 1 \forall n \tag{5}$$

$$\begin{gathered} x_{mm'm''l\,nk} + x_{mm'm''knl} \le 1; \\ \{l, n, k, m, m', m''\} \\ l \ne n, k \ne n, m \in M_n, m' \in M_n, m'' \in M_j \end{gathered} \tag{6}$$

Where,
s order index, indicating a specific permutation of operations;
G(*s*), total processing cost related with operations in order *s;*
k,l, hole index, $k=1,...,N$ $l=1,...,N;$
a, cost per unit non-productive travelling distance;

b, cost per unit tool switch time;
M_n, set of tools that can be used to drill hole n to its final size;
P_{nk}, non-productive travelling distance between current hole n and following hole k;
P_{ln}, non-productive travelling distance between current hole l and following hole n;
$q_{mm'n}$, tool switch time between current tool type, m', and tool m required by hole n;
$t_{mm'n}$ machining time required by current tool type, m', and tool m required by hole n;
$T_{mm'n}$ life of current tool type, m', and tool m associated with cutting operation on hole j;
$x_{mm'm''lnk}$, a 0-1 integer variable, $x_{mm'm''lnk}$=1 if tool m changes tool m'' to drill hole n which is situated in the path between previous hole l and following hole k and has been drilled by tool m'; 0, otherwise. Similarly $x_{mm'm''knl}$, a 0-1 integer variable, $x_{mm'm''knl}$=1 if tool m changes tool m'' to drill hole n which is situated in the path between previous hole k and following hole l and has been drilled by tool m'; 0, otherwise. The indices m, m', m'' in the 0-1 decision variable are used to determine the proper tool switch order during operation. The 0-1 decision variables, $x_{mm'm''lnk}$ simultaneously decide the order of holes to be processed as well as the order of tools to be used to process each hole. The function of this variable is to make sure the each operation should be carried out once. This particular condition has been taken care by constraint Eq. (5). Eq. (6) makes sure that, the backward movement of spindle is not allowed unless the tool switch is required [1].

Mathematical model given in Eqs. (4)- (6), which requires large amount of computational time due to higher number of 0-1 decision variables in order to minimise the total processing cost of hole-making operations. Hence, to solve this model, two efficient solution procedures, namely, particle swarm optimisation algorithm and shuffled frog leaping algorithm are proposed. Next section discusses particle swarm optimisation algorithm.

3 Particle swarm optimisation (PSO) algorithm

Particle swarm optimisation (PSO) is an evolutionary computation method developed by Kennedy and Eberhart [19]. This method starts with initialization of population of random solutions called 'particles'. Optimal solution is obtained by updating generations by Eqs. (7)- (8) [19]. Particle searches the optimum solution through the problem space by comparing the current optimum particles. This algorithm consists of two 'best' values. First is 'p_{best}' – the best fitness values of individual particles achieved so far. Second one is 'g_{best}' which are the best values between all particles. In PSO, velocity of particles is changed at every generation towards the 'p_{best}' and 'g_{best}'. Velocity and position of individual particles are obtained using following Eqs. (7)- (8):

$$V_{i+1} = w * V_i + C_1 * r_1 * (p_{besti} - X_i) + \\ + C_2 * r_2 * (g_{besti} - X_i) \tag{7}$$

$$X_{i+1} = X_i + V_{i+1} \tag{8}$$

Where,
V_{i+1}= New velocity of each particle using Eq. (7);
X_{i+1}= New position of each particle using Eq. (8);
C_1 = cognitive parameter of each particle;
C_2 = social parameter of each particle;
W = Inertia weight.
This process of iterations using Eqs. (7) - (8) continues till convergence criteria are satisfied.

4 Shuffled frog leaping algorithm

The shuffled frog leaping algorithm is a meta-heuristic optimisation technique, originally developed by Eusuff and Lansey in 2003, which is based on the conduct of a group of frogs while searching for the maximum amount of food site [20]. The most well-known benefit of shuffled frog leaping algorithm is its fast convergence speed [17].
The various steps in SFL algorithm with modification are as follows [20]:

1. Generate virtual frog randomly called population p;

2. Evaluate the fitness of population;

3. Sort the population in descending order;

4. Partition the population in m memeplexes;

5. Frogs i is expressed as $X_i = (X_{i1}, X_{i2}, \ldots X_{is})$ where S represents number of variables;

6. Identify the worst 'X_w' and the best frog 'X_b' within each memeplexes;

7. Identify the global best frog 'X_g' in entire population;

8. Apply the local search for new positions (X_{i+1}) by following equations:

$$X_{i+1} = X_i + r \times (X_b - X_w) \quad (9)$$

If fitness of new frog generated by above Eq.(9) is better than previous frog, then replace it with new frog, where:
X_{i+1} = New position of frog;
X_i = Previous position of frog;
r = Random number values between 0 to 1;
X_b = Position of best frog among the memeplexes;
X_w = Position of worst frog among the memeplexes.

9. If not, apply Eq.(10) to obtain better position:

$$X_{i+1} = X_i + r \times (X_g - X_w) \quad (10)$$

10. If fitness of new frog generated by Eq.(10), is better than previous frog, then replace it with new frog, else replace the worst frog randomly, where
X_g = Position of best frog among the memeplexes.

Table 1. Distance between each holes in mm.

Hole	1	2	3	4	5	6	7	8
1	0	30	80	70	90	110	150	190
2	30	0	60	110	65	80	170	180
3	80	60	0	160	40	70	200	150
4	70	110	160	0	40	90	30	100
5	90	65	40	40	0	60	60	70
6	110	80	70	90	60	0	140	50
7	150	170	200	30	60	140	0	90
8	190	180	150	100	70	50	90	0

The next section briefly describes injection mould example.

5 Injection mould example

The particle swarm optimisation & shuffled frog leaping algorithm are applied to determine the optimal sequence of operations and corresponding cutting speeds for the upper base of industrial plastic injection mould as shown in Fig. 2, consisting of overall 8 holes.

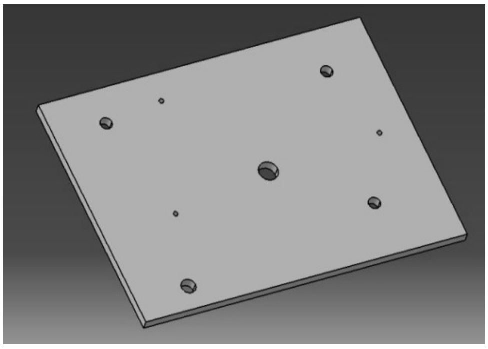

Figure 2. Plastic injection mould upper hole plate.

Distances between each hole are given in Tab. 1. Tab. 2 shows the tool switch times.

Table 2. Tool switch times in minutes.

Tool	Predecessor tool			
Successor tool	1	2	3	4
1	0	0.5	0.5	0.5
2	0.5	0	0.5	0.5
3	0.5	0.5	0	0.5
4	0.5	0.5	0.5	0

In this work it is assumed that the hole can be machined using single individual tool.
Information of diameters of holes and tools required to machine these holes are as shown in Fig. 3 and they are as follows:
H_2=H_4=H_6= Ø 4, Tool 1 is required,
H_1=H_3=H_8=Ø10=Tool 2 is required,
H_7= Ø 12=Tool 3 is required,
H_5= Ø 16=Tool 4 is required.

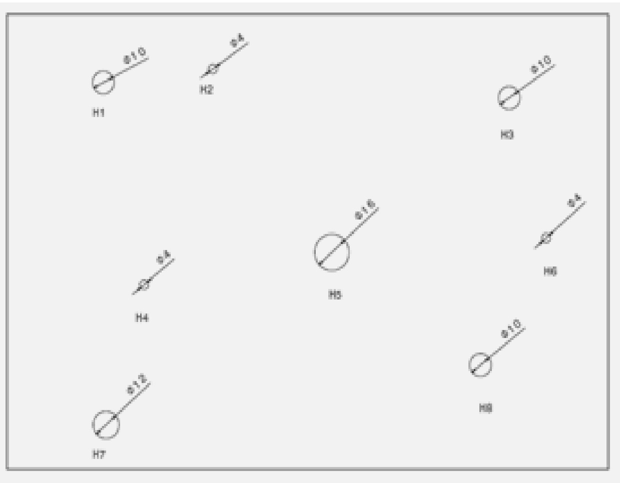

Figure 3. Data of hole diameters.

Tab. 3 shows data related to feed rate, diameter of tool & machining cost.

Table 3. Data related to feed rate, diameter of tool and machining cost.

Tool	f_m, (mm/rev.)	d_m, mm	Y_m, €
1	0.11	4	0.267
2	0.12	10	0.4
3	0.13	12	0.533
4	0.14	16	0.667

6 Results & discussion

The proposed particle swarm optimisation algorithm and shuffled frog leaping algorithm were coded in code blocks C++ and run on a Windows 8 PC with Intel core i3 CPU @ 1.90 GHz to determine the optimum sequence of operations for a workpiece shown in Fig. 3.
Possible number of sequences by considering all combinations of sequences for above example can be obtained using following Eq. (11) [4]:

$$\frac{\sum_{i=1}^{l}(n_i)!}{\prod_{i=1}^{l}(n_i)!} \tag{11}$$

Possible numbers of sequences = 40320, where, n_i is a total number of operations required for machining. 'i' stands for single machining operation. 'l' stands for total number of operations required for entire part.
To carry out this analysis, initially 40320 sequences of operations as given in Eq. (11) are generated. Then total processing cost of each sequence is obtained as per the mathematical model given in Eqs. (4)-(6) by coding it in code blocks C++ software and the sequence corresponding to the minimum value of total processing cost obtained between all sequences is considered for comparison with results of PSO & SFLA with modification.
In next paragraph the results of PSO and SFLA with modification are discussed.
The following constants 'a' & 'b' are used during computational experiments given in Eq. (4), in order to determine the total processing cost of hole-making operations.
a= € 0.00053/min;
b= € 0.666/min;
Z_{mn}= €.2.3335;

Z, machining cost per unit time= € 1/min

Collective machining and tooling cost of all holes on injection mould are calculated using Eq. (1). To calculate collective machining and tooling cost Z_{mn}, tool life and optimum cutting speeds are obtained from Eq. (12) and Eq. (13), respectively.

The tool life expression for these operations [22] is as follows in Eq. (12), for drilling a new hole:

$$T_{mn} = \left(\frac{8 d_m^{\;0.4}}{U_{mn} \times f_m^{\;0.7}}\right)^5 \qquad (12)$$

Optimum cutting speed as expressed below in Eq.(13), can be achieved [1] by solving differential Eq. (3) with above tool life Eq. (12), for drilling a new hole:

$$U_{mn} = 6 \times \sqrt[5]{\frac{Y d_m^{\;2}}{Z_m f_m^{\;3.5}}} \qquad (3)$$

Following algorithm specific parameters for particle swarm optimisation algorithm are obtained through various computational experiments. The effect of *w*, C_1 and C_2 on convergence for standard numerical benchmark functions was provided by Bergh and Engelbrecht [23]. The optimum selection of operating parameters of the algorithm like acceleration constants C_1 and C_2 as well as inertia coefficient *w* is essential for the convergence of the algorithm. To ensure the convergence of the PSO algorithm, the condition specified by the following Eq. (14) must be satisfied [23]:

$$max\,(|\lambda 1|, |\lambda 2|) < 1 \qquad (14)$$

where:

$$\lambda_1 = \frac{1 + w - \phi_1 - \phi_2 + \gamma}{2} \qquad (15)$$

$$\lambda_2 = \frac{1 + w - \phi_1 - \phi_2 - \gamma}{2} \qquad (16)$$

Where, $\gamma = \sqrt{(1 + w - \phi_1 - \phi_2)^2 - 4w}$, $\phi_1 = c_1 \times r_1$ and $\phi_2 = c_2 \times r_2$.

As the feasible range for '*w*' is 0-1 and for C_1 and C_2 is 0-2, the selected values of *w*, C_1 and C_2 should be such that the Eq. (14) is satisfied for all possible values of random numbers 'r_1' and 'r_2' in the range 0-1.

C_1=1.75;
C_2=1.65;
w=0.615;
Number of variables: 8;
Range of variables: 1 to 8;
Number of iterations: 500;
Number of particles: 100.

Following algorithm specific parameters for shuffled frog leaping algorithm are obtained through various computational experiments.

C_1=0.95;
C_2=1.0;
w=1.0;
Number of variables = 8;
Range of variables = 1 to 8;
Number of frogs = 25;
Number of memeplexes = 5;
Number of subfrogs = 5;
Numbers of iterations =15.

Tab. 4 shows comparison of total processing cost of hole-making operations for best optimal sequence of each proposed method.

Obtained results of the example discussed in section 5 using particle swarm optimisation, shuffled frog leaping algorithm and minimum value of total processing cost obtained by considering all possible combinations of sequences are shown in Tab. 5.

7 Conclusion

Optimisation of hole-making operations involves a large number of possible sequences depending upon the hole location and tool sequence to be followed on part in order to minimise the total machining cost. An application example of injection mould is attempted by using particle swarm optimisation algorithm and shuffled frog leaping algorithm. The exact value of minimum total processing cost is obtained by considering all possible combinations of sequences. It is observed that the minimum value of total processing cost obtained by using particle swarm optimisation and shuffled frog leaping algorithm are almost the same as those obtained by considering all possible combinations of sequences. This clearly indicates the potential of the presented algorithms to solve the complex problem of determining optimal sequence of hole-making operation. Although, the numbers of code lines for particle swarm optimisation algorithm and shuffled frog leaping algorithm are higher than those required for obtaining minimum total processing cost, considering all possible combinations of sequences approach, the computational time required for particle swarm optimisation algorithm and shuffled frog leaping algorithm is shorter than that required for obtaining minimum total processing cost if we consider all possible combinations of sequences approach. Small error in results of PSO and SFLA compared to minimum value of total processing cost, when considering all possible combinations of sequences, is due to probabilistic nature of these algorithms. Moreover, if more numbers of holes with different diameters, depths, tolerances, surface finish requirements along with tapping and reaming operations are considered in this problem, the complexity of the problem involved makes it harder to solve considering all possible combinations of sequences approach. This clearly shows that particle swarm optimisation and shuffled frog leaping algorithm can be effectively used in optimisation of large scale injection mould hole-making operations. Future work for these two proposed algorithms can be applied to an array of industrial injection mould applications involving holes with different diameters and depths.

Table 4. Comparison of total processing cost of hole-making operations for best optimal sequence

Method	Z_{mn}, €	Tool switch travel cost, €	Non-productive tool travel cost, €	Total processing cost, €
Minimum value of total processing cost by considering all combinations of sequences	2.3335	0.99	0.2756	3.6081
PSO	2.3335	0.99	0.2809	3.6134
SFLA	2.3335	0.99	0.3074	3.6399

Table 5. Results comparison of best optimal order of hole-making operations

Method	Best possible sequence	Total processing cost in €	Number of code lines & Execution time
Minimum value of total processing cost by considering all combinations of sequences	{8,3,1,2,6,4,7,5}	3.6081	258 &4 minutes
PSO	{8,3,1,2,6,4,5,7}	3.6134	843 lines & 56 seconds
SFLA	{1,3,8,5,6,2,4,7}	3.6399	1210 &1 min

REFERENCES

[1] Kolahan, F., Liang, M,: *Optimisation of hole-making operations: a tabu-search approach*, International Journal of Machine Tools & Manufacture, 40 (2000), 1735–1753.

[2] Luong, L.H.S., Spedding, T.: *An integrated system for process planning and cost estimation in hole-making,* International Journal of Manufacturing Technology, 10 (1995), 411–415.

[3] Qudeiri, J. A., Yamamoto, H.: *Optimisation of Operation Sequence in CNC Machine Tools Using Genetic Algorithm*, Journal of Advanced Mechanical Design, Systems, and Manufacturing, 1 (2007) 2.

[4] Ghaiebi, H., & Solimanpur, M.: *An ant algorithm for optimisation of hole-making operations,* Computers and Industrial Engineering, 52 (2007) 2, 308-319.

[5] 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*, Journal of Computational Information Systems, 7 (2011) 2, 411-418.

[6] Alam, M. R., Lee, K. S., Rahman, M., Zhang, Y. F.: *Process planning optimisation for the manufacture of injection moulds using a genetic algorithm,* International Journal of Computer Integrated Manufacturing, 16 (2003) 3, 181-191.

[7] Tamjidy, M, Paslar, S., Hang, B. T., Baharudin, T., Hong, T. S., Ariffin, M. K. A.: *Biogeography based optimisation (BBO) algorithm to reduce non-productive time during hole-making process,* International Journal of Production Research, (2014).

[8] Rajkumar, E., Annamalai, K.: *Investigation into optimal fixturing cost of an assembly using genetic algorithm,* Engineering Review, 34 (2014) 2, 85-91.

[9] Dalavi, A. M., Pawar, P.J., Singh, T.P.: *Optimisation of hole-making operations for injection mould using particle swarm optimisation algorithm,* International Journal of Industrial Engineering Computations, 6 (2014), 433-444.

[10] Srivatsava, P. R. et al:, *Optimal test sequence generation using firefly algorithm*, Swarm and Evolutionary Computation, 8 (2013), 44–53.

[11] Marinakis, Y., Marinaki, A.: *Bumble Bees Mating Optimisation algorithm for the Open Vehicle Routing Problem*, Swarm and Evolutionary Computation, (2014), 1580–1594.

[12] Narooei, K. M., Ramli, R., Rahman, M.Z., Iberahim,F., Qudeiri, J. A.: *Tool Routing Path Optimisation for Multi-Hole Drilling Based on Ant Colony Optimisation*, World Applied Sciences Journal, 32 (2014) 9,1894-1898.

[13] Montiel-Ross, O., Rodríguez, N., Sepúlveda, R, Melin, P.: *Methodology to Optimize Manufacturing Time for a CNC Using a High Performance Implementation of ACO*, International Journal of Advanced Robotic Systems, 9 (2012), 121.

[14] Liu, X., Hong, Y., Ni, Z., Qi, J., Qiu X.: *Process planning optimisation of hole-making operations using ant colony algorithm*, The International Journal of Advanced Manufacturing Technology, 69 (2013) 1-4,753-769.

[15] Lim, W. C. E., Kanagaraj,G., Ponnambalam, S. G.: *A hybrid cuckoo search-genetic algorithm for hole-making sequence optimisation*, Journal of Intelligent Manufacturing, (2014).

[16] Lim W. C. E., Kanagaraj, G., Ponnambalam, S. G.: *Cuckoo Search Algorithm for Optimisation of Sequence in PCB Holes Drilling Process*, Emerging Trends in Science, Engineering and Technology, Lecture Notes in Mechanical Engineering, (2012), 207-216.

[17] Elbeltagi, E., Tarek, H., Donal, G.: *Comparison among five evolutionary based optimisation algorithms,* Advanced Engineering Informatics, 19 (2005), 43-53.

[18] Rao, R. V.: (2011) *Modeling and optimisation of Modern Machining processes.* Springer series in advanced manufacturing.

[19] Kennedy, J., Eberhart. R.: Particle swarm optimisation proceedings of IEEE International Conference on Neural Networks, 4 (1995), 1942-1948.

[20] Eusuff, M.M., Lansey, K.E., Pasha, F.: *Shuffled frog-leaping algorithm: a memetic metaheuristic for discrete optimisation,* Engineering Optimization 38 (2006) 2, 129–154.

[21] Elbeltagi, E., Tarek, H., Donald, G.: *A modified shuffled frog-leaping optimisation algorithm: applications to project management.* Structure

and Infrastructure Engineering, 3 (2007) 1, 53 – 60.

[22] Zhao, R.: (1992). *Handbook for machinists*. Shanghai Science and Technology Press, China.

[23] Bergh, F.; & Engelbrecht, A.P.: *A study of particle swarm optimisation particle Trajectories,* Information sciences, 176 (2006), 937-971.