Genetic Algorithm For Multi-Objective Optimization in Sculptured Dies-Cavity Roughing

Ineu Widaningsih
Faculty of Industrial Engineering
Bandung Institute of Technology
Bandung, Indonesia
ineu_widaningsih@president.ac.id

Suprayogi
Faculty of Industrial Engineering
Bandung Institute of Technology
Bandung, Indonesia
yogi@mail.ti.itb.ac.id

Anas Ma'ruf
Faculty of Industrial Engineering
Bandung Institute of Technology
Bandung, Indonesia
maruf@mail.ti.itb.ac.id

Dradjad Irianto
Faculty of Industrial Engineering
Bandung Institute of Technology
Bandung, Indonesia
dradjad@mail.ti.itb.ac.id

***Abstract*— Sculptured-dies Cavity Roughing (SDCR) problem is a multi-dimensional problem. In XY-single cutter problem, the decision variables consist of layering and tooling selection problem by maximizing the efficiency of roughing, which consider finishing efficiency. The previous research approach to this problem shows that the dynamic programming approach. However, it is effective in searching solutions for the time-to-volume coefficient minimization (TVC) problem, and empirically shows 10% improvement compared to machining time minimization objective. The pre-processing procedures in dynamic programming approach are quite complex time-consuming. Applying a genetic algorithm procedure for the multi-dimensional problem (GAMD) guarantees the merging process's feasibility, these pre-processing procedures can be eliminated, and significantly faster computational time. In the 7-3-3 problem chosen in this research, the computational time is reduced from about 2 hours to 30 seconds.**

Keywords*— *Sculptured-dies Cavity Roughing, genetic algorithm, tool selection, Cutting layer determination, time-to-volume coefficient.

I. INTRODUCTION

Sculptured-dies Cavity Roughing (SDCR) problem is a multidimensional problem because it takes more than one decision variables and can have more than one objective functions [1]. There are decision variables that are machining parameters such as feedrate, cutting force and others. Some other decision variables are not machining parameters such as selecting cutting tools and cutting height or cutting plane. Likewise, there are various objective functions in SDCR, such as minimizing cutting path length, minimizing roughing time, minimizing energy consumption and others as described in [1]. The study of [2] examined that one of the crucial problems found from the 168 papers on sculpture surface milling is the optimization of machining time. Some research in machining optimization considers more than one objective and a multi-objective optimization problem [3-7].

In the SDCR and XY-single cutter problem, the cutting plane determination, and the selection of tool, one tool for each cutting plane is crucial to optimize the process [8-11]. The concept of merging the hunting layers was introduced by [8] as the procedure for cutting plane determination which should be optimized simultaneously with the decision of tool allocation. The combinations of alternative tools and hunting layers make the problem a complex combinatorial problem. To solve that problem, both [9, 10] have applied dynamic programming (DP) approach, with single objective [9] and multiple objectives [10]. Although the approach produces optimum solution and fast searching time, the requirement of the pre-processing procedure is complex and not easy. [9] has applied a genetic algorithm (GA) approach to solve the problem and show that the solution is acceptable [12].

Section II presents the result of GA application into SDCR problem with 2 objectives considered: minimizing machining time and residual volume. Aggregate measurement, Time-to-Volume Coefficient (TVC), is used to trade-off between both objectives, as applied in [10]. The application of GA is expected to eliminate the complex pre-processing procedure when DP applied as in [10]. Table I explains the difference between [10,11] and this research.

TABLE I. SDCR OPTIMIZATION

Researcher	Method	Objective (Min)
[10]	Dynammic Programming and Genetic Algorithm	Roughing time
[11]	Dynammic Programming	Time-to-Volume Coefficient
This paper	Genetic Algorithm	

Fig 1, as presented in [10], represents XY-Single cutter problem, with four candidates of cutting layers or hunting layers. A maximum of 4 cutting layers can be cut. If there is/are merging some hunting layers, this cavity will be cut by less than four cutting layers. Only one cutting tool is allocated for each cutting layer. The searching of both cutting layers and cutting tools should be done simultaneously to find the optimum TVC.

The relationship between TVC, machining time and residual volume is shown by equation 1 as presented in [10].

$TVC_l^{p,m}$ is a function of machining time T_l^p, residual volume R_l^m, and total cavity volume to be machined V_l^m. Where l is representing hunting layer (l=1..L/A,B,C…), p is representing alternative tool (p=1..P), and m is representing number of merged layer (m=1..M).

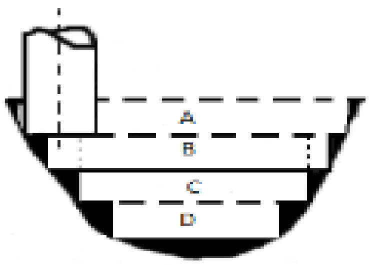

Fig. 1. Layer-by-layer sculptured-die cavity machining [10]

$$TVC_l^{p,m} = \frac{T_l^p}{\frac{V_l^m - R_l^m}{V_l^m}} \text{ x } \frac{R_l^m}{V_l^m} \qquad (1)$$

Section III is the literature review of hybrid evolutionary approaches in CNC machining and section IV is the discussion of application of hybrid evolutionary approaches in SDCR problem for the next research. Section V is the conclusion of GA application in SDCR problem and the recommendation for hybrid evolutionary approaches in SDCR problem.

II. Genetic Algorithm Application For SDCR Problem To Minimimize Time-to-Volume Coefficient

A. *Procedure*

Fig. 2 is GA procedure for SDCR which is started by setting the parameter value for L,M,P, population size (B), Number of iteration (I), Cross-over probability (ProbC), Mutation probability (ProbM), and number of elite chromosomes. The chromosome is the binary coding as presented in [11]. The number of consecutives zeros in a chromosome is limited by the maximum merging values. This is guaranteed by the feasibility checking in the algorithm.

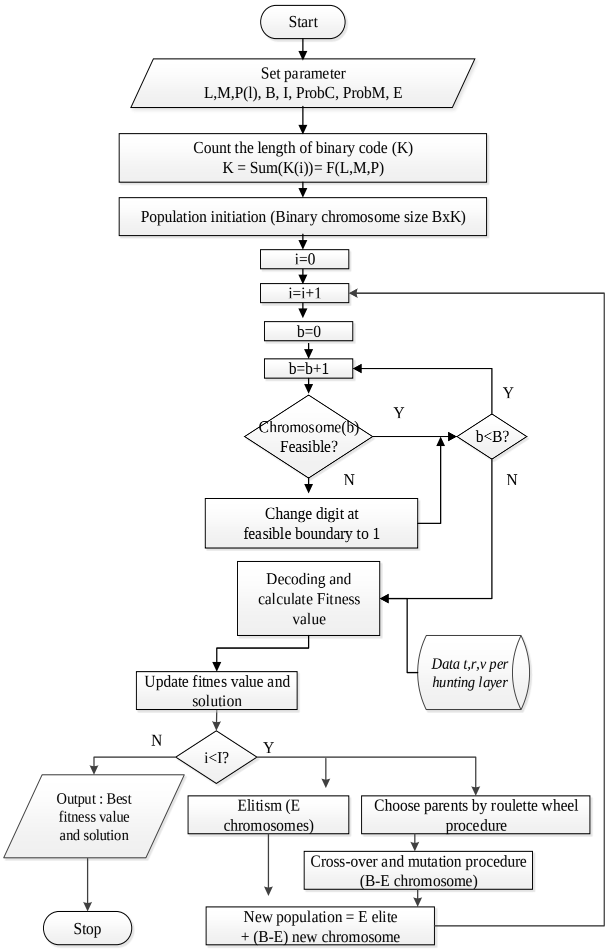

Fig. 2. GA procedure for SDCR

The following is an example of an explanation of the steps in the genetic algorithm for the problem of determining the cutting layer and tool selection in DCM. The case discussed is the L-P-M value of 7-3-3 with 20 data sets and two feasible tool conditions (3332211 and 3333333).

Step 1: Generate a matrix of binary numbers with size BxK randomly.]

For example, B=100, K=12, where B is number of chromosomes and K represents the number of feasible alternative tools for each cutting layer. The case of L-P-M = 7-3-3 and the number of appropriate tools per hunting layer = 3,3,3,2,2,1,1 so that k=2,2,2,2,1,1,1, then K=Sum(k)=12.

Fig. 3 shows an example of 6 out of 100 chromosomes that are generated. This matrix is then checked for feasibility using the procedure in step 2.

init =

1	0	1	1	1	1	1	1	1	0	1	1
1	1	0	1	1	0	0	1	1	0	0	0
0	1	1	1	1	0	1	0	0	1	1	0
1	1	0	0	0	0	0	0	1	0	1	0
1	0	0	1	1	0	0	0	1	1	1	1
0	1	1	1	0	1	0	1	0	0	1	0

Fig. 3. An example of matrix of 6 x 12

Step 2: Check the maximum merging procedure

Columns 1 to 10 are declared eligible if there are 0 in front of a maximum of 6 consecutive columns. Columns 11 to 12 are declared eligible if there are a maximum of 5 successive 0 in front of them. In the example. The matrix generated in step 1 (see Fig, 3) has an impropriety in the 4th row and 8th column, so the binary number matrix is revised as seen in Fig.4.

```
init =
   1   0   1   1   1   1   1   1   1   0   1   1
   1   1   0   1   1   0   0   1   1   0   0   0
   0   1   1   1   1   0   1   0   0   1   1   0
   1   1   0   0   0   0   0   1   1   0   1   0
   1   0   0   1   1   0   0   0   1   1   1   1
   0   1   1   1   0   1   0   1   0   0   1   0
```

Fig. 4. Revised matrix of 6 x 12

Step 3: Convert the matrix of size BxK to matrix of size BxL

It is necessary to calculate the fitness value for each chromosome to determine the parent chromosome. For calculating the fitness value, each binary number group needs to be returned to its actual number so that the matrix size becomes BxL. In this example, a 6x12 binary number matrix is translated into a 6x7 real number matrix, according to the number of hunting layers and the number of chromosomes. The converted matrix can be seen in Fig. 5.

```
dek =
   2   3   3   2   1   1   1
   3   1   2   1   1   0   0
   1   3   2   1   1   1   0
   3   0   0   1   1   1   0
   2   1   2   0   2   1   1
   1   3   1   1   0   1   0
```

Fig. 5. Converted matrix of 6 x 7

Step 4: Decoding to tool code numbers and merging (kpm)

The result of step 4 can be seen in Fig.6.

```
kpm =
   2100   3200   3300   2400   1500   1600   1700
   3100   1200   2300   1400   1500      0   1670
   1100   3200   2300   1400   1500   1600   1700
   3100      0      0   1234   1500   1600   1700
   2100   1200   2300      0   2450   1600   1700
   1100   3200   1300   1400      0   1560   1700
```

Fig. 6. Decoding to tool code numbers and merging

Step 5: Calculation of Fitness value

The fitness value is computed using (2), because the objective function is to minimize (3), where TVC is Time-to-Volume Coefficient, t is roughing machining time, r is residual volume (cut-off), and v is volume of machining area (roughing+cut-off). The result of fitness value can be seen in Table II.

$$Fitness\ Value = \frac{1}{TVC} \tag{2}$$

$$TVC = \frac{t.r}{(v-r)} \tag{3}$$

TABLE II. FITNESS VALUE

Chromosome	Fitness Value
001001101111	1.2639
110010011111	0.9929
001101100110	1.2541
100101111110	1.1212
010101001110	1.0670
101011100100	1.0244

Step 6: Selected two chromosomes with the best fitness value to be stored as an elite group.

Subsequently, 98 chromosomes were generated with a random binary number of size L. Then, cross-over operations and mutations were performed on the selected chromosomes. The first and third chromosomes have the largest fitness values in the following example, so they become elite chromosomes (see Table II).

B. *Numerical Example*

In the application of the genetic algorithm, 7-3-3 cases were selected with 20 data sets with the following explanation:

1. The twenty data sets used to represent the shape of the mould cavity having the same height, with the total number of feasible cutting tools being the same for each data set. The difference between each data set is the inclination of the mould cavity wall, which is slightly different from one data set to another. The data from A to T represent mould cavities with different cavity wall slopes and change gradually from cavity A to cavity T.

2. There are seven candidate cutting layers for each mould cavity, also known as hunting layers. The number of appropriate chisels is different for each hunting layer. Each alternative decision combination (merging and tool allocation) will be represented by chromosomes, a collection of binary numbers. Although the number of feasible tools is different for each hunting layer, they can be exchanged in the cross-over process because binary numbers represent them.

The cross-over and mutation procedure uses a single-point crossover and a low mutation probability. By following the procedure in [11] for the development of chromosome with binary coding, for the case of 7 hunting layers (L=7), 3 maximum merging layers (M=3), and 3 alternative cutting tools (P=3), there are 12 digits of binary coding (K=12). Parameter settings of B, I, ProbC, ProbM, E are 100, 50, 0.8, 0.05 and 2 consecutively. Single cross-over is applied.

The following explains the steps in the application of genetic algorithms. The first step begins with the determination of the chromosomes that will represent the expected solution, namely the cutting layer and the selected cutting tool.

The binary numbers are used to represent the two decisions. The cutting layer decision is the result of the merging decision of several hunting layers, which is represented by a 0. For example, two zeros before 1 means that two other hunting layers will merge with the 3rd hunting layer.

For example, chromosome 001100001011 is a matrix of binary numbers with 12 digits representing seven hunting layers with the number of appropriate tools in a row from hunting layers 1 to 7 is 3,3,3,2,2,1,1. The number of suitable tools for each hunting layer is represented by the number of binary digits. The number of 3 suitable tools and will be represented by 2 digits (11,10,01, or 00) which states the selected tool number, except 00 which means that a merging process occurs with hunting, next layer. The interpretation of chromosome 001100001011 into a process code is shown in Table III. T1AB0 means hunting layers A and B will be using cutting tool number 3 (T1). The last zero indicates that the maximum merging is 2. T1AB0 is represented by the binary number 0011.

TABLE III. INTERPRETATION OF BINARY CODE ON CHROMOSOMES

Hunting layer	**HL1**		**HL2**		**HL3**		**HL4**		**HL5**		**HL6**	**HL7**
Binary	0	0	1	1	0	0	0	0	1	0	1	1
Merging	2 hunting layer				3 hunting layer							
Cutting tool			Tool 3 (T1)						Tool 2 (T2)		Tool 1 (T3)	Tool 1 (T3)
Real No	0		3		0		0		2		1	1
Process Code	T1AB0				T2CDE						T3F 00	T3G 00

C. *Cases : 3 different classes of merging*

The data (cases) chosen for GA application for SDCR is the same data presented in [11]. Twenty data sets in Table IV represent different cavities with slightly different inclinations in each sculpture wall. Based on that experiment, it is concluded that three different merging classes can be represented by 3 data sets to be chosen in the GA application: data set 1, 2 and 3. The first 3 rows in Table IV are the optimum solution for those sets of data (cavity 1-3).

Optimum TVC for data set-1 (cavity-1) is achieved when the roughing process is T1AB-T2CD-T2E-T3F-T3G. That means hunting layers A and B are merged and cut together using cutting tool T1. Hunting layers C and D are merged and cut together using cutting tool T2.

Optimum TVC for data set-2 (cavity-2) is achieved when the roughing process is T1ABC-T2DE-T3FG. Optimum TVC for data set-3 (cavity-3) is achieved when the roughing process is T1A-T1B-T1C-T2D-T2E-T3F-T3G, or no merging for any of those hunting layers.

Since those three sets of data (3 cavities) represent overall merging classification, then those data are used to test the GA application for SDCR.

TABLE IV. OPTIMUM RESULT FOR EACH DATA SET

Data Set#	**1/TVC**	**Optimum Merging and Tool Allocation**						
		1	***2***	***3***	***4***	***5***	***6***	***7***
1	1.3819	T1AB		T2CD		T2E	T3F	T3G
2	1.4396	T1ABC			T2DE		T3FG	
3	1.0575	T1A	T1B	T1C	T2D	T2E	T3F	T3G
4	1.1161	T1A	T1B	T1C	T2D	T2E	T3F	T3G
5	1.0081	T1A	T1B	T1C	T2D	T2E	T3F	T3G
6	0.9183	T1A	T1B	T1C	T2D	T2E	T3F	T3G
7	0.8432	T1A	T1B	T1C	T2D	T2E	T3F	T3G
8	0.7788	T1A	T1B	T1C	T2D	T2E	T3F	T3G
9	0.7241	T1A	T1B	T1C	T2D	T2E	T3F	T3G
10	0.6780	T1A	T1B	T1C	T2D	T2E	T3F	T3G
11	0.6386	T1A	T1B	T1C	T2D	T2E	T3F	T3G
12	0.6057	T1A	T1B	T1C	T2D	T2E	T3F	T3G
13	0.5780	T1A	T1B	T1C	T2D	T2E	T3F	T3G
14	0.5537	T1A	T1B	T1C	T2D	T2E	T3F	T3G
15	0.5348	T1A	T1B	T1C	T2D	T2E	T3F	T3G
16	0.5297	T1ABC			T2DE		T3FG	
17	0.5230	T1ABC			T2DE		T3FG	
18	0.5247	T1ABC			T2DE		T3FG	
19	0.5345	T1ABC			T2DE		T3FG	
20	0.5571	T1ABC			T2DE		T3FG	

Each procedure is run 14 times for each set of data. Table V to Table VII summarize the distribution of the best fitness values for each run. The GA result's frequency percentage representing the optimum solution is quite good. Consecutively the percentages are 64.3, 71.4, and 64.3. The remaining values are close enough to the optimum values.

TABLE V. DISTRIBUTION OF FITNESS VALUE ON 14 RUNS (DATA SET-1)

Fitness Value	**Frequency**	**%**
1.3720	1	
1.3770	2	
1.3778	2	
1.3819 (optimum)	9	64.3

TABLE VI. DISTRIBUTION OF FITNESS VALUE ON 14 RUNS (DATA SET-2)

Fitness Value	**Frequency**	**%**
1.2636	1	
1.4251	2	
1.4306	1	
1.4396 (optimum)	10	71.4

TABLE VII. DISTRIBUTION OF FITNESS VALUE ON 14 RUNS (DATA SET-3)

Fitness Value	**Frequency**	**%**
1.0552	1	
1.0545	3	
1.0575 (optimum)	10	64.3

III. DISCUSSION

It is concluded that the proposed GA procedures are showing a good result. 15 out of 20 experiments conducted by using GA approach produce optimum results. The TVC values are good enough means that very high chance to get the optimum, and in the case not optimum, it is close to optimum. The computation time and searching time of GA procedures is fast, less than 30 seconds. Compare the multi-objective dynamic programming (MODP) method. This GA procedure does not require the pre-processing stage, MODP requires complex pre-processing procedure [9].

Fig.7 are some graphics showing the GA searching for better fitness values. The convergence of the searching process is quite good. However, the improvement of the values during the searching time shows that at some point at the beginning of searching time, the values are stuck in local optimum values. It can be improved using Hybrid Evolutionary Approach [13, 14].

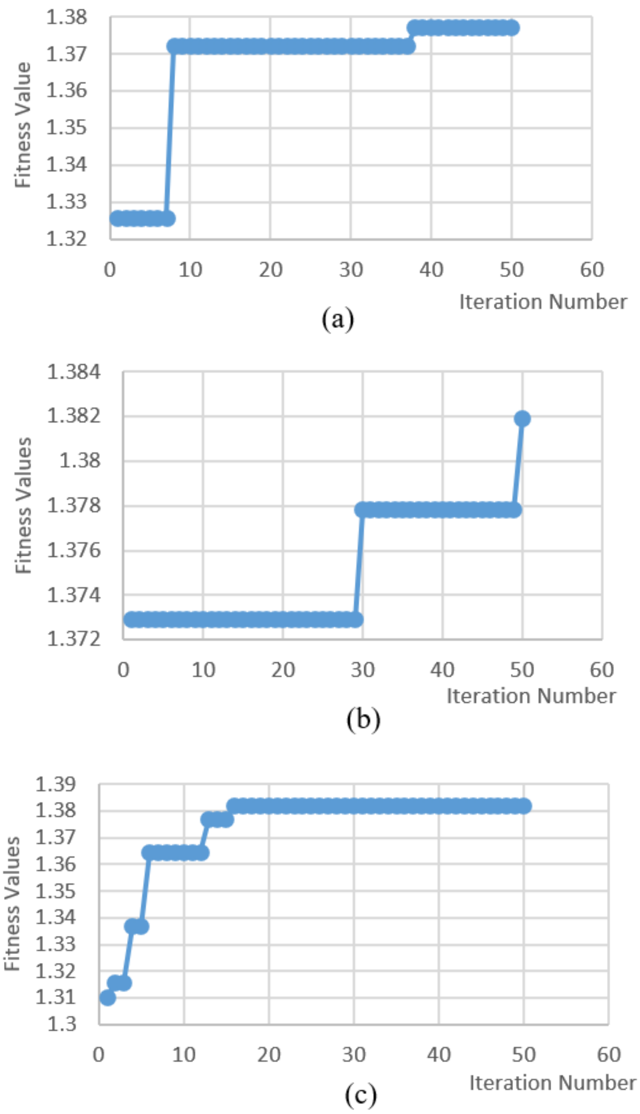

Fig. 7. GA Searching process to better fitness values for Data Set 1 (a) Run #1; (b) Run #4; (c) Run #8.

Since each data set represents different cavities with slightly different inclination in each sculpture wall, the simulation results for the machining time and residual volume obtained from the simulator will also be different. So the best solution is different for each data set.

The problem of determining cutting tool layers and selecting tools for each layer simultaneously is a complex combinatorial problem. The application of dynamic programming algorithms with the Dijkstra algorithm with three dimensions has shown promising results: the same solution as the optimal solution. Through Matlab programming, the computation time required is short, which is only 19 seconds on average for each data set.

The same data set, namely, the data set with scenario 7-3-3, which produces 4095 combinations of solutions for layering decisions (merging) and cutting tool allocation decisions, takes 4 hours of computation time for each data set enumeration process for each combination. Thus, there is a significant reduction in computation time. The search for solutions using dynamic programming only takes an average of 19 seconds. However, a reasonably complex pre-processing process is required to become impractical.

It takes a more practical approach without a complex pre-processing process, but with a short computation time. The metaheuristic approach, in this case, the genetic algorithm, is a promising approach. The genetic algorithm was chosen to simplify the solution search procedure. This section will discuss the steps of the algorithm and the results of applying the algorithm both from the resulting solution and from GA's ability to produce optimal or near-optimal solutions.

IV. Conclusion

Using a Genetic Algorithm for the multi-dimensional problem (multi decision variables) in SDCR problem has proven to be effective in searching the optimum solution and can eliminate the pre-processing procedures such as in the DP approach. The algorithm has produced the optimum result or very close to the optimum result. The optimum result as presented in [9], empirically showed 10% of improvement of machining efficiency when compared to result from the simulator and as well better machining efficiency when compared to smallest-tool-possible (STP) and smallest residual possible (SRP).

Compared to enumeration procedures, the GA procedure reduces computational time significantly from about 120 minutes per data set to less than 30 seconds.

The limitation of this research is the fitness values are stuck in local optimum. Thus, Hybrid Evolutionary Approach can be considered. Moreover, combinatorial optimization algorithm using intelligent generation algorithm can be considered for better accuracy result and shorter computation time.

Acknowledgment

The support from Production System Laboratory of Bandung Institute of Technology is gratefully acknowledged

References

[1] I. Widaningsih, A. Ma'ruf, Suprayogi and D. Irianto, "Multi-Objectives Optimization on Cutting Layers and Cutting Tools Selection for Sculptured-Dies Cavity Rough Machining: A State-of-the-Art Review", *in Proceedings of the International Conference on Business and Engineering Management (ICONBEM)*, 2021. [Online]. Available: https://www.atlantis-press.com/proceedings/iconbem-21/125956941

[2] R. Mali, T. Gupta and J. Ramkumar, "A comprehensive review of free-form surface milling– Advances over a decade", *Journal of Manufacturing Processes*, vol. 62, pp. 132-167, 2021

[3] D. S. S. R. Kiran , S. P. Kumar, Multi objective optimization of tool life and total cost using 3-level full factorial method in CNC end milling process, *Int. J. Mech. Eng. & Rob. Res.* , vol. 2, no. 3, pp. 255-270, Jul. 2013.

[4] G. Kant and K. Sangwan, "Prediction and optimization of machining parameters for minimizing power consumption and surface roughness in machining", *Journal of Cleaner Production*, vol. 83, pp. 151-164, 2014.

[5] M. Gusenbauer and N. Haddaway, "Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of Google Scholar, PubMed, and 26 other resources", *Research Synthesis Methods*, vol. 11, no. 2, pp. 181-217, 2020.

[6] O. V. Arriaza, D-W. Kim, D. Y. Lee, M. A. Suhaimi, Trade-off analysis between machining time and energy consumption in impeller NC machining, *Robotics and Computer-Integrated Manufacturing*, vol. 43, pp. 164-170, 2017,

[7] A. A. Krimpenis,N. A. Fountas,T. Mantziouras, N. M. Vaxevanidis, "Optimizing CNC wood milling operations with the use of genetic algorithms on CAM software", *Wood Material Science and Engineering*, vol. 11, no. 2, pp. 102-115, 2014.

[8] Y-H. Chen, Y-S. Lee, S-C. Fang, "Optimal cutter selection and machining plane determination for process planning and NC machining of complex surfaces", *Journal of Manufacturing Systems*, 1998, Vol. 17, no. 5, 371-388.

[9] Y. Wang, H. Ma, C. Gao, H. Xu and X. Zhou, "A computer aided tool selection system for 3D die/mould-cavity NC machining using both a heuristic and analytical approach", *International Journal of Computer Integrated Manufacturing*, vol. 18, no. 8, pp. 686-701, 2005.

[10] I. Widaningsih, A. Ma'ruf, Suprayogi, D. Irianto, "Tool selection and cutting plane determination for maximizing efficiency in roughing process by using multi-objectives dynamic programming", *Journal of Advanced Manufacturing Technology,* 2021, in press.

[11] I. Widaningsih, A. Ma'ruf, Suprayogi, D. Irianto, "Binary Coding Enumeration for Multi-dimensional Problem in Sculptured Dies Cavity Roughing Optimization*",* in *13th International Seminar on Industrial Engineering and Management*, Jul. 2021, in press.

[12] T. Alam, S. Qamar, A. Dixit and M. Benaida, "Genetic algorithm: reviews, implementations, and applications", *International Journal of Engineering Pedagogy (iJEP)*, vol. 10, no. 6, p. 57, 2020. [Online]. Available: https://online-journals.org/index.php/i-jep/article/view/14567/8299

[13] S.Sigl and H. Mayer, "Hybrid Evolutionary Approaches to CNC Drill Route Optimization", in *Proc. of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* (CIMCA-IAWTIC'06), 2006, doi:10.1109/cimca.2005.1631379

[14] A.M. Kafaf, M. M. El-Sherbiny, W.F. A. EL-Wahed, N. A. Ismael, "A hybrid evolutionary approach to solve multi-objective optimization problems based on particle swarm optimizer and genetic algorithm", Mar. 2020.

[15] Y. Xiao, H. Zhang, Z. Jiang, Q. Gu and W. Yan, "Multiobjective optimization of machining center process route: Tradeoffs between energy and cost", *Journal of Cleaner Production*, vol. 280, p. 124171, 2021.