

GA-AHP method to support robotic polishing process planning

Wang, K¹., Ding, L., Dailami, F¹ and Matthews, J^{2*}

1. Bristol Robotics Laboratory (BRL), Stoke Gifford, Bristol, United Kingdom.
2. School of Engineering, University of the West of England, Cold harbour Lane Bristol, United Kingdom.

* Corresponding author - Email: Jason.Matthews@uwe.ac.uk

Abstract

The finishing stage of mould manufacturing is generally completed via mechanical polishing, manual conducted. Not only is this the most expensive phase of the process but also is currently struggling with a deficit of skilled workers. To address these issues, and support the wider needs of Industry 4.0, the manufacturing community has investigated robotic technologies to support the polishing process. The work reported here, investigating the polishing process planning and optimisation for mould manufacture is part of a larger project aiming to automate 80% of the current manual process. Presented in this article is an optimization strategy for robotic polishing process sequencing aiming at satisfying polishing sequence rules and shortest polishing time simultaneously. A hybrid approach combining both genetic algorithm (GA) and analytical hierarchical process (AHP) is proposed based on the specific characterises of polishing process sequencing. A multi-objective fitness function is defined using AHP including the calculation of polishing time and evaluation of polishing process rules. The proposed GA-based process sequencing has been successfully demonstrated on test piece examples.

Keywords: Computer-aided process planning; polishing rules; genetic algorithm; robotic; AHP

1. Introduction

In the manufacture of production tooling such as moulds and dies, surface finishing operations, buffing, and polishing are currently manual processes. Normally involving a skilled operative equipped with polishing tools. To remove a layer of material eradicating pre-machining marks and abrasions in the mould or die cavity and inserts, to achieve a required surface finish (Lee *et al.*, 2006 and Altan, *et al.* 2001;). An example is shown with the die insert in Figure 1, where 'A' shows the part post milling machined and part 'B' shows the component finished via polishing. The features of milled part, have different starting roughness.' Feature 'a,' starts at $1.6\mu\text{m}$; edge features 'b' and 'd' start at $0.8\mu\text{m}$, top faces 'c' and 'e' start at $1.92\mu\text{m}$. The required surface finish for the injection moulding process, is for features 'b' and 'd' to have $0.1\mu\text{m}$, for features 'a', 'c' and 'e' to has a final roughness of at least $0.8\mu\text{m}$. To achieve this the six tools shown for Figure 1C needed to be employed.

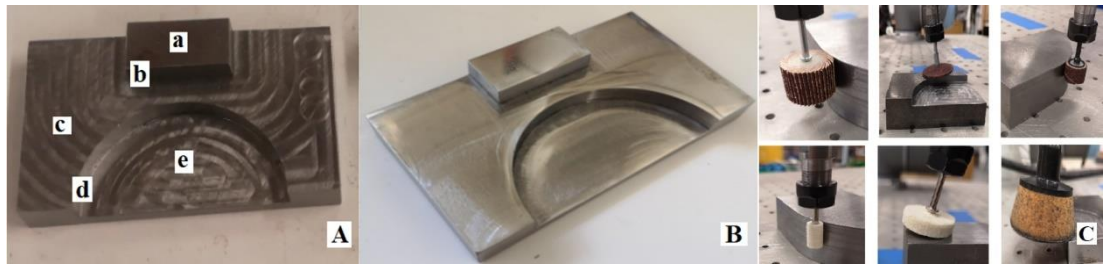


Figure 1 Die insert – 'A' initial state, 'B' post polishing

In the last two decades, the mould and die manufacturing industry currently faces two challenges. One of finances, in comparison to global low-cost production areas. It has been shown that up to twenty percent of the production costs and approximately fifty percent of the production time are assigned to finishing operations. Also, the industry is facing the pressure of the of skilled workers specializing in finishing operations (Ragaert *et al.*, 2014; Winther, 2003). With these factors in mind, robotics is seen as one potential solution. (Wang *et al.*, 2019; Kalt *et al* 2016). However, to extend its applications in the polishing process so as to finally substitute most of manual polishing work in industry, a fast and automatic generation of polishing process plan is necessary. Although Computer-aided process planning (CAPP) has already commonly been applied to manufacturing processes, few systems involve the detailed information of polishing process, such as types of polishing tools (cf. Figure 1C), abrasive papers, process variable settings, and even optimizing sequence of polishing operations, which robotic polishing requires. Meantime, as polishing process is generally carried out after finishing a series of cutting manufacturing operations, the

constraints for polishing process sequencing are not as same as cutting manufacturing process sequencing.

Presented in this article is a strategy of optimization for polishing process sequencing using Genetic Algorithm (GA) and analytical hierarchical process (AHP). Section 2 presents a review of recent research, Section 3 addresses the main characteristics of polishing process sequencing, which are different from cutting manufacturing process sequencing. Section 4 proposes a new GA-AHP optimization strategy for polishing process sequencing based on specific constraints and rules for polishing processes. To meet the requirements of multiple objective optimizations, AHP is applied to define the multi-objective fitness function on both the satisfaction of polishing process sequence requirements and minimum polishing time concurrently. The GA method for polishing process sequence, is then discussed, including a new proposed initialization algorithm and specific genetic operators particularly for the requirements on polishing process sequencing. The next Section provides the implementation of the proposed method and test example. The closing section draws conclusions and presents future work.

2. Background

Process sequencing is a complex issue and involves a large number of constraints and factors, such as various component details (e.g., geometrical tolerances, feature relationships, etc.), manufacturing rules, precedence requirements, time, and cost. A Genetic Algorithm is a classical evolutionary algorithm based on biological operators such as selection, crossover, and mutation. The genetic algorithm technique has advantages to generate good solutions to complex problems, and therefore attracts number of researchers on its application for process sequencing optimization. such as the work of Li *et al.*, (2005); Bo, Hua, and Yu (2006); and Fan and Wang (2012). Qi *et al* (2017) designed an enhanced hierarchical GA for typical manufacturing process optimization for metal structure production systems. A hierarchical structure aimed to solve simultaneously, three issues, process sequencing, layout selection, and machine selection. Knust *et al.*, (2017) presented a GA optimization method for hot forging processes with multiple objectives, based on shape geometry requirements. Su *et al.* (2018) developed a strategy-based GA for operational sequencing posed on edge selection, aiming to satisfy defined precedence constraints. Their edge selection strategy proports to improve the GA's converging efficiency. Dou *et al.*, (2018) presents an enhanced GA for sequencing aiming to minimize the overall cost. Their new elitist-based crossover strategy and an improved mutation method are adopted to keep the feasibility of the GA chromosomes.

For use in a CAM system, Čuboňová *et al.*, (2019) developed a GA-based optimization strategy for machining processes. The length of tool paths was considerably reduced

by using genetic algorithm to reduce machine times and optimize cutting parameters. A hybrid constrained 'permutation algorithm' and GA approach for process planning was presented by Falih and Shammari (2020). The constrained permutation algorithm is employed to produce an initial solution, to minimize the number of setup and tool changes. The authors employed a GA to search for best fit process plans. An optimisation algorithm amalgamating variable neighborhood search and a GA for process sequencing of large-size problem was presented by Luo *et al* (2020). Their version of variable neighborhood search was used to decompose a complex and sizable solution space into simple multi-neighborhood areas. An approach combining Tabu search, and GA was proposed by Shi and Xiong (2021). The goal of their hybrid approach was to minimize the tool changeover times in CNC milling. Tabu search was combined with the GA to improve the performance.

The polishing process is mainly performed by human workers more recently robotic polishing has gained more and more attention because of the reason mentioned earlier. Thus, the work to support polishing process sequencing is limited. Mohsin *et al.* (2019) proposed an approach based on tool path planning to improve the contact area per cut path and surface quality. The polishing parameters optimization were evaluated using design of experiments (DOE). Liu *et al.* (2022) presented a novel robotic polishing planning method that considered, the deformation problems when polishing the thin sheet metal parts. Their approach is based on Hertz theory, differential geometry and derived a polynomial equation. It combines a developed constant speed robot path planning method with the contact-area information to guarantee improved surface quality.

In summary, GAs have advantages for process planning, such as it is easy to represent and implement the operations' sequence in the coding environment using chromosomes; It is more likely to generate global optimization and avoid local optimal points by genetic operators like crossover and mutation; It is able to support multiple objectives optimization through definition of a multi-objective fitness function; etc. The previous research has shown GAs are good solutions for optimization and search problems, and therefore GAs is suitable technique for the applications on process optimization. However, the elements of a GA, including fitness function, crossover and mutation operators, initial selection strategies, determine the effectiveness of the GA's application. Meanwhile, GAs combined with other artificial intelligence techniques, such as different optimization algorithms, can improve performance of GA optimization and therefore provides a promising solution for more complex optimization problems. Additionally, there is few efforts on the polishing process sequencing. However, with the introduction of robots to polishing process, the requirements for automated polishing operation sequencing are increasing.

3. The characteristics of polishing process sequencing

Moulds and dies include a series of machined features and inserts that form the component during the production process. The surface finish of the final component directly relates to the machined finished of the mould or die or its inserts (Matthews and Ding, 2009). The higher the quality of surface finish required the more time and expense is exhausted in the manufacturing process. Therefore, the manufacturer will aim to obtain the minimum polishing time on each feature to satisfy customer requirements. E.g., hidden features may only need to be polished to a level that allows easy extraction from the mould. Most of previous CAPP research focus on cutting manufacturing process, including various machining (e.g., drilling, milling, turning, etc.) covering from blank, rough machining, semi-finishing to finishing machining. However, polishing process sequencing is different from general cutting manufacturing process sequencing:

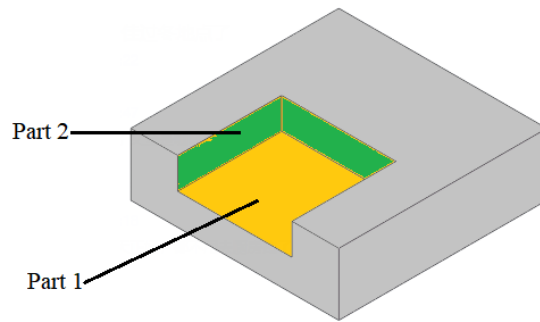


Figure 2. Illustration of polishing operation-parts

Table 1 Polishing operation-parts of the open pocket feature

operation part	Starting Roughness	Required Roughness	Polishing tool	Operation-parts
Part1	1.6 μ m	0.15 μ m	Polishing disc	Operation 1(#400)
				Operation 2(#800)
				Operation 3(#1000)
Part2	1.6 μ m	0.2 μ m	Cylinder polishing tool	Operation 1(#400)
				Operation 2(#800)

- For cutting manufacturing process, the manufacturing feature is treated as a whole (Ding and Yue, 2004). For example, all faces of the open pocket (shown in **Figure 2**) are generated by certain machining operation (e.g., milling) at same time. Differently, as shown **Table 1**, the open pocket must be broken down two geometrical parts for polishing, which the part1 will choose polishing disc with three operations using different abrasive papers, while the

part2 will use cylindrical polishing tool with two operations to finish. Thus, polishing features need to be further broken down to one or more geometrical groups, each of which could use different tools, abrasive papers, and setups. Such geometrical group in this research is called as polishing operation-part. Meanwhile, from the polishing view, in most situations, polishing operation-parts could be treated independently during process sequence. Also, different accuracy or surface finish could put polishing operation-parts, even within same feature, in different process precedence. For example, there are two features in the component shown in **Figure 3**. These two features can be further broken down to four polishing operation-parts. Although the polishing operation-parts A and B belong to a same feature (i.e., F1), and C and D belong feature F2, the operations for polishing operation-part A are arranged together with operations of operation-part C, not operation-part B, due to the same polishing tools and abrasive papers.

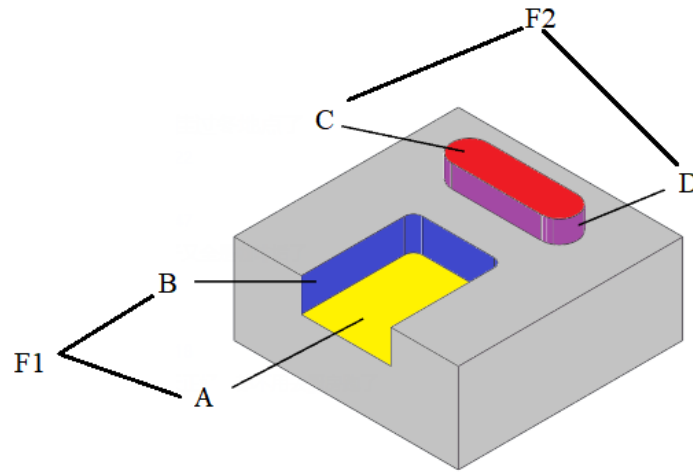


Figure 3. The arrangement of operation-parts in distinctive features

In addition, polishing operation-parts, which have same type, dimensions, and surface finish requirements, such as $p011$, $p021$, $p031$, and $p041$ shown in **Figure 3**, will use same polishing process and be polished together in industrial practice. Therefore, the concept of polishing operation-group, including the polishing operation-parts with same type, dimension, and surface finish requirement, is introduced here. For example, there are two polishing operation groups in **Figure 4**: $p011$, $p021$, $p031$, and $p041$; and $p012$, $p022$, $p032$, and $p042$. The polishing operation-group will be treated as a single polishing operation-part during the polishing process planning.

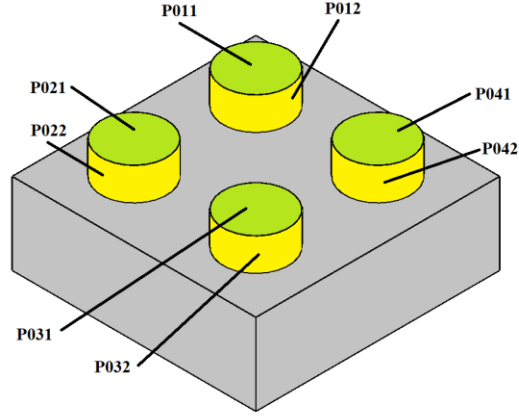


Figure 4. Examples of polishing operation-group

- Polishing process sequencing is different from general cutting manufacturing process sequencing. Polishing process is usually carried out after finishing a series of cutting manufacturing operations. Therefore, the rules for polishing process sequence cannot be the same as cutting manufacturing process sequences. For example, as shown in **Figure 5**, for cutting manufacturing process planning, which focuses on planning material removal processes (e.g., milling) from a blank, the parent feature (i.e., ft_2) should be processed before its child feature (i.e., ft_1) by considering their tool access directions and machining efficiency. However, at polishing stage, as these two features have already been rough/semi-finished machined, such precedence constraint are not need anymore. On contrary, if parent feature ft_2 has higher surface finished requirements than child feature ft_1 , then the feature ft_1 should have its fine-polishing operation before feature ft_2 . When one considers general subtractive machining manufacturing processes. They are typically conducted based on the order from rough machining, semi-finishing to finishing machining, and therefore, the features with highest indicative surface roughness are usually machined before the features with lowest indicative surface roughness. E.g., in producing a high precision hole. The part would be rough machined with a drill, giving high material removal rate, but leaving a surface finish range of $6.3\mu\text{m}$ - $1.6\mu\text{m}$. Then it would be reamed, leaving a surface finish range of $3.2\mu\text{m}$ - $0.8\mu\text{m}$. The final operation would be honing providing a surface finish range of $0.8\mu\text{m}$ - $0.1\mu\text{m}$. However, as the polishing process is one of the finishing machining, such precedence requirements are not needed except some special situations. Using the example in **Figure 5**, although in this instance, feature ft_3 has lower finished surface roughness requirement than the feature ft_2 , it does not matter which feature needs to be polished first as the two polishing features are not connected or adjacent. Differently, as feature ft_1 and feature ft_2 are

connected, the fine-polishing operation for the feature ft_1 (with lower finished surface roughness requirement) should be arranged prior to the fine-polishing operation for feature ft_2 (with higher finished surface roughness requirement).

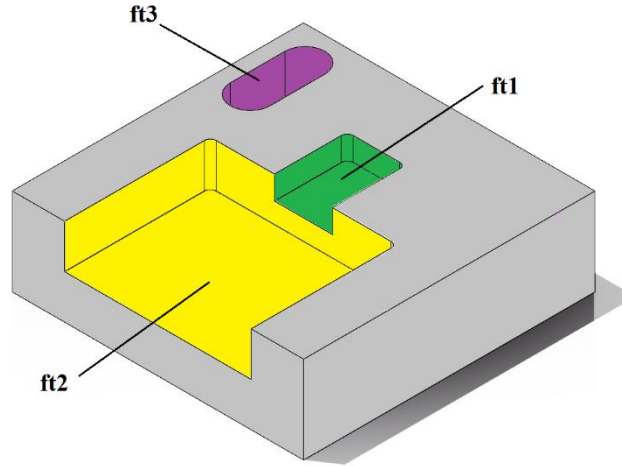


Figure 5. A component with three features

Thus, considering the specific requirements of polishing process, six polishing process sequencing rules are derived, which can be classified as two aspects: polishing precedence constraints and polishing efficiency constraints:

Polishing precedence constraints:

- **Rule 1:** To gain the target of surface finish, multiple polishing operations are usually required for a polishing operation-part. Rough polishing operations should be done before fine polishing operations.
- **Rule 2:** For a cutting feature, which geometry is obtained by removing a volume material from the initial blank, the polishing operation-parts with low required surface finish should be polished prior to the polishing operation-parts with high required surface finish, which guarantee the best surface finish awarded.
- **Rule 3:** If two polishing operation-parts are connected or adjacent, the polishing operation-parts with low required surface finish should be polished before the polishing operation-parts with high required surface finish, so that the best surface finish is achieved.

Polishing efficiency constraints:

- **Rule 4:** Polishing operation-parts to be polished using same machine setup should be polished successively.
- **Rule 5:** Operations, which use the same polishing tool should be arranged successively.
- **Rule 6:** Operations, which use same polishing tool and abrasive paper, should be conducted successively.

4. GA-AHP optimisation strategy

This paper proposes a GA-based polishing process sequencing, aiming to obtain an optimal point which satisfies both the polishing sequence rules and quickest polishing time. **Figure 6** depicts the procedure of the proposed genetic algorithm. The process starts by initializing populations with several valid candidates using an initial precedence constraints algorithm (details given in Section 4.2). Then performs a genetic operation to these candidates (i.e., selection, crossover and mutation discussed in Section 4.3). The multiple-objective fitness functions are calculated using AHP method (as described in Section 4.1). The output with the best fitness function is adopted. This iterative process stops when fitness function cannot be further reduced.

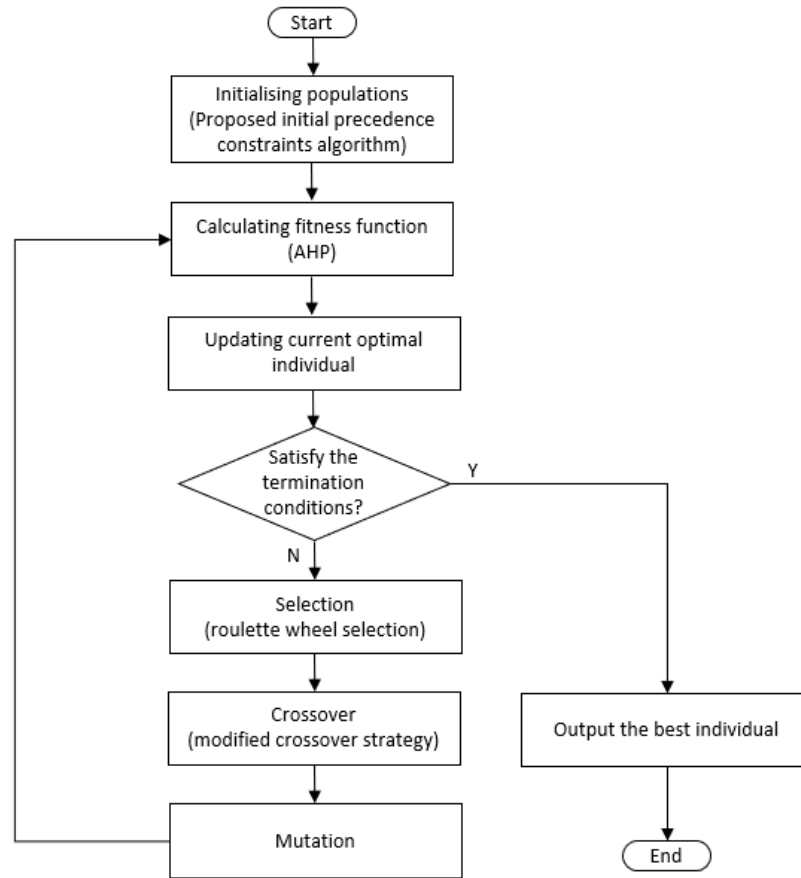


Figure 6 Procedure of the proposed genetic algorithm process

4.1. Fitness function calculation using AHP

With the application of GAs to operations like process planning or sequencing, it is important to have an appropriate fitness function, as this is considered as a performance benchmark. Therefore, it will indicate the degree of objective satisfaction of a solution searched (Ding and Matthews, 2009). Because of this, in any use of a GA, the fitness calculation is the most important mechanism. In this research, a fitness function, which aims at the shortest polishing processing time and satisfaction of the key polishing rules simultaneously, is proposed for the polishing process sequencing.

However, what has been demonstrated in practice, is that it is practically impossible to satisfy all sequence requirements within a polishing process sequence. E.g., a polishing process sequence may have shortest time but conflict with the polishing process rules. Oppose, a polishing process sequence could satisfy very well with sequencing rules but have longer polishing time. Another example, if two polishing operations have different machine setups but use same polishing tool and abrasive paper. According to the rule of polishing operations using same machine setup should be polished successively, the two polishing operations should not be arranged together, but it conflicts with the rule of operations, which use same polishing tool and abrasive paper, should be conducted successively.

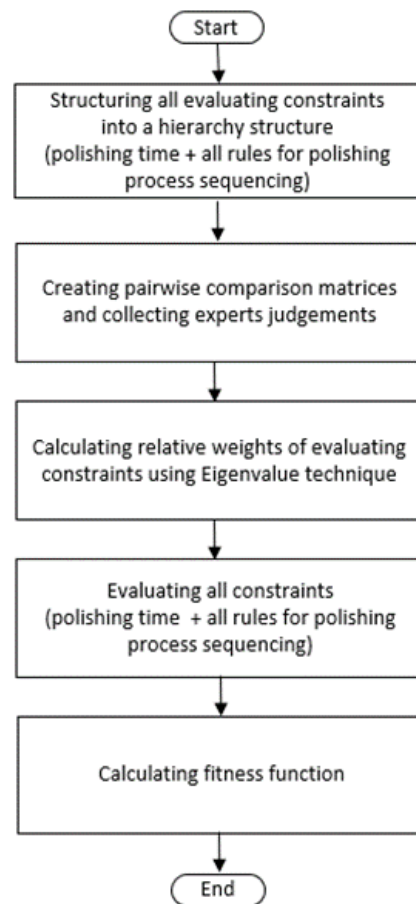


Figure 7. Procedure of AHP

Thus, it becomes importance introduce a grouping of weights which represent the relative importance of the requirements inside the polishing process sequencing. The weights with most importance are obtained via the application of Analytic Hierarchy Process (AHP). AHP has been shown to be an effective technique for multi-attribute decision-making (e.g., Ji and Jiang, 2003; Angelis and Lee, 1996). Figure 7 shows the AHP model proposed in the research.

Step 1: The evaluation criteria is hierarchal structured into four levels from an

overall objective to various sub-criteria (cf. Figure 8).

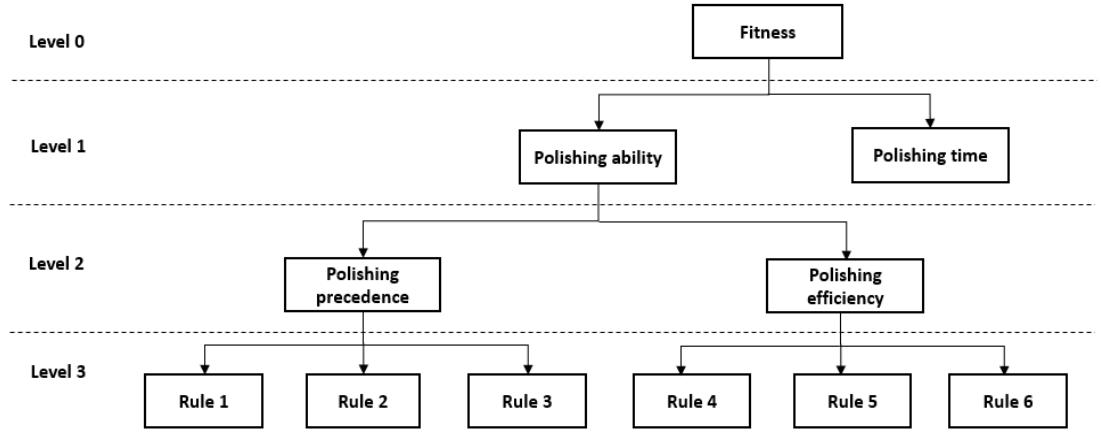


Figure 8. AHP evaluation criteria

Step 2: Based on expert judgement, to determine the relative weights of structured criteria, pairwise comparisons (Matrix R) are constructed. R is described by a number of r_{ij} which presents the relative importance between the i th and j th constraint. This is presented as follows:

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1m} \\ \vdots & \ddots & \vdots \\ r_{m1} & \cdots & r_{mm} \end{bmatrix}$$

where m is the selected constraints number

$$r_{ii} = 1$$

$$r_{ij} = 1/r_{ji}$$

In this research, an evaluation criteria table based on a one to nine scale is employed in the matrix R, as shown with **Table 2**. E.g., if Rule i (e.g., Rule 4) is considered to have very strong importance than Rule j (e.g., Rule 5) in the evaluation, a weight of ‘7’ is given to R_{ij} . On the contrary, the value of R_{ji} is set to ‘1/7’. Based on the proposed AHP evaluation criteria, four matrices have been defined, that is R_1 (2x2), R_{21} (2x2), R_{31} (3x3), and R_{32} (3x3).

Within these matrices, the defined value of each element is provided based on expert or suitably qualified persons knowledge and experience.

Table 2 Matrix evaluation criteria

Term	Importance intensity of (r_{ij})	Importance intensity of (r_{ji})
The i th and the j th constraint have	1	1

the same importance		
The i th constraint has moderate importance than the j th constraint	3	1-3
The i th constraint has strong importance than the j th constraint	5	1- 5
The i th constraint has very strong importance than the j th constraint	7	1 -7
The i th constraint has extreme importance than the j th constraint	9	1-9
The values in-between importance adjacent scale values	2, 4, 6, 8	1- 2, 1-4, 1-6, 1-8

Step 3: Use Eigenvalue techniques to calculate the weights.

a) For each row, all elements are added together to sum S_i :

$$S_i = \sum_{j=1}^m r_{ij}$$

where j represents the column;

i represents the row.

m is the number of the columns (= rows) in the matrix R.

b) Apply the normalized weight vector, \bar{W}_i

$$W_{ij} = \frac{r_{ij}}{S_i}$$

$$\bar{W}_i = \frac{1}{m} \sum_{j=1}^m W_{ij}$$

where j represents the column. $j = 1, \dots, m$;

m is the number of the columns (= rows) in the matrix R.

The proposed AHP evaluation criteria in this research only require small size of the matrix R and therefore the step of checking the consistency may not be necessary.

Step 4: Calculate polishing time: Basically, the calculation of the time for a polishing process is similar to other machining process, including the time for conducting polishing and other supporting activities, such as changing tools. The main difference for polishing process is the consideration of the abrasive paper change time, including changing the abrasive paper within an operation due to the wearing of the abrasive paper; changing abrasive paper with different levels according to the required Ra for different operations; and changing abrasive papers

when the polishing tools is changed. Thus, the duration for a polishing process is defined as followed:

$$T_{total_time} = \sum_{i=1}^n TP_i + \sum_{\substack{i=1 \\ j=2 \\ i \neq j}}^n TC_{ij}$$

$$TC_{ij} = TTC_{ij} + TTM_{ij} + TPC_{ij}$$

where T_{total_time} is the duration spent for the whole polishing process;

TP_i is the duration for the i th polishing operation (including tool path planning time, polishing time, and the time to change the abrasive paper within this operation);

TC_{ij} is the change time between operation i and j ;

TTC_{ij} is the tool changeover duration for the i th polishing operation to the j th polishing operation, including the abrasive papers changing time when the polishing tool is changed;

TTM_{ij} is the time for the polishing tool moving from the i th operation to the j th operation;

TPC_{ij} is the change time of different types of abrasive papers from the i th polishing operation to the j th polishing operation while polishing tool is same.

A relative evaluation value for the polishing time, f_t , can be gained as:

$$f_t = \frac{T_{total_time}}{T_{max}}$$

where T_{max} is the maximum polishing time permitted.

Step 5: The polishing process sequence rules are evaluated as the satisfaction degree. Apart from the weight system, the precedence value system is built on how good the precedence of two polishing operations meets the requirements of the rules for polishing process. For example, V_{hij} refers to the evaluation using rule h when operation j is performed behind operation i . If the polishing rule h support the two operations precedence, V_{hij} is assigned a small positive value less than one (i.e., $0 < V_{hij} \leq 1$). The less value, the fitter for the rule h . On the other hand, if the polishing rule h is against the two operations precedence, V_{hij} is set a value larger than one (i.e., $V_{hij} > 1$). The larger value, the more conflict for the rule h .

Step 6: Calculate fitness function as below:

$$W_k = W_q W_{kq}^S$$

$$Fitness = W_t^0 f_t + W_p^0 \sum_{k=1}^m \sum_{i=1}^n \sum_{\substack{j=1 \\ O_i > O_j}}^n W_k v_{kij}$$

where W_t^0 is the weight for the polishing time (Level 1);

W_p^0 is the weight for the polish ability (Level 1);

W_k is the weight of the k th constraint (Level 3) for the polishing ability (Level 1);

W_q is the weight of the q th constraint in the Level 2 for the polishing ability (Level 1);

W_{kq}^S is the weight of the k th constraint (Level 3) for the q th constraints in the Level 2;

f_t is the relative evaluating value for polishing time;

V_{kij} is the precedence value for the k th constraint (Level 3) if operation j is performed behind operation i .

4.2. Initialisation

Initialization produces a number of populations for polishing process planning of a component. Good initial populations should cover sufficiently valid search space. It means the initial populations need not only to include various viable solutions, but also to exclude those invalid solutions. For the polishing sequencing, on the one hand, the precedence constraints among features, even if the precedence/successive constraints among different polishing operation-parts in a feature, are not as important as in the cutting manufacturing process. However, on the other hand, for the multiple operations for a polishing operation-part, the rough polishing operation must be strictly conducted prior to fine polishing operation, if not, it will be infeasible. Therefore, a new algorithm to generate initial populations has been developed. At the first stage, a polishing operation-part will be selected randomly from pool of the unselected polishing operation-parts. Then, at the second stage, the polishing operations for the selected polishing operation-part will be allocated an unoccupied position randomly but strictly based on rough, semi-finishing and finishing polishing precedence constraints.

The algorithm can be illustrated by an example, as shown in **Figure 9** and **Table 3**, the component has two features with four polishing operation-parts (i.e., P011, P012, P021 and P022). Each of these polishing operation-parts further consists of several operations.

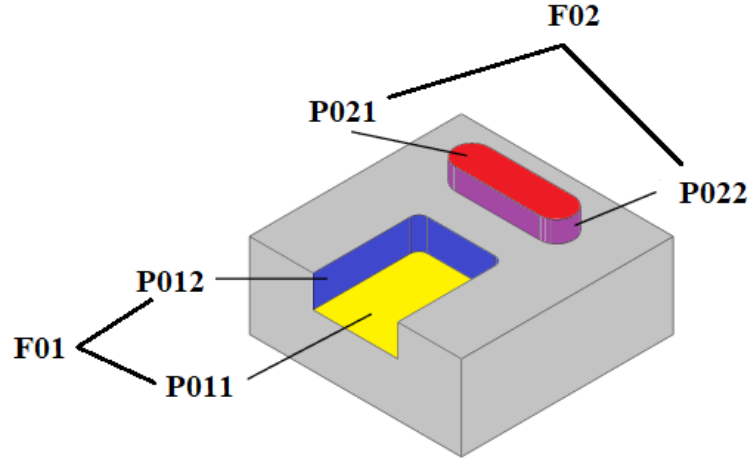


Figure 9 A component with four feature parts

Table 3 Operations for each feature part

P011	P0111(#400)	P012	P0121(#400)
	P0112(#800)		P0122(#800)
	P0113(#1000)		P0123(#1000)
	P0114(#1200)		P0124(#1200)
P021	P0211(#400)	P022	P0221(#400)
	P0212(#800)		P0222(#800)
	P0213(#1000)		P0223(#1000)
	P0214(#1200)		P0224(#1200)

First, there are four polishing operation-parts are unselected. A polishing operation-part is chosen randomly from these polishing operation-parts (e.g., $i = 2$). There are four operations for the selected polishing operation-part (i.e., P012): Strictly based on rough to finish polishing precedence, P0121, P0122, P0123 and P0124 allocated an unoccupied position in sequence.

- P0121: As P0121 must be carried out before P0122, P0123 and P0124, there must be three unused positions (for P0122, P0123 and P0124) behind P0121. That means 16 positions in the chain have not been used, but only 13 positions can be chosen for P0121, randomly choose $k \in [1,13]$, e.g., $k = 3$, the 3rd unused position, i.e., the 3rd position in the chain.
- P0122: behind P0121, only thirteen positions have not been used. However, P0122 must leave two unused positions for P0123 and P0124. Thus, randomly select $k \in [1, 11]$, e.g., $k = 2$, the 2nd unselected position after the position of P0121 (i.e., 3rd), i.e., the 5th position in the chain.
- P0123: different from P0122, P0123 is only required to keep one position for P0124. Therefore, post P0122, there are ten positions which are available, so

selected $k \in [1, 10]$, for example, if $k = 4$, the 4th unselected position after the position of P0122 (i.e., 5th), i.e., the 9th position in the chain.

- P0124: P0124 does not need to consider other operations and thus 7 positions are available behind P0123, $k \in [1, 7]$, such as $k = 5$, the 5th unselected position after P0123 position (i.e., 9th), i.e., the 14th position in the chain.

After all operations for P012 have been allocated, there are only three unselected polishing operation-parts (i.e., P011, P021 and P022) in the pool. Randomly selected one polishing operation-part (e.g., $i = 1$), which means P011 is chosen. P011 has four operations: P0111, P0112, P0113 and P0114, according to the rough to finish polishing precedence.

- P0111: P0111 must be arranged before P0112, P0113 and P0114. Although there are 12 unused positions in the chain, only 9 positions can be chosen for P0111, randomly choose $k \in [1, 9]$, e.g., $k = 2$, the 2nd unused position, i.e., the 2nd position in the chain.
- P0112: behind P0111, ten positions are available to select. Meanwhile two positions must leave for P0113 and P0114. Thus, randomly select $k \in [1, 8]$, e.g., $k = 3$, the 3rd unselected position after P0111 position (i.e., 2nd), i.e., the 7th position in the chain.
- P0113: after the position of P0112 and leaving one unused position for P0114, there are six positions available. Selected $k \in [1, 6]$, for example, $k = 6$, the 6th unselected position after P0112 position (i.e., 7th), i.e., the 15th position in the chain.
- P0114: the position of P0113 leaves only one position can be selected for P0114, $k \in [1, 1]$, e.g., $k = 1$, the 1st unselected position after P0113 position (i.e., 15th), i.e., the 16th position in the chain.

Similarly, the positions of P0211 to P0214, and P0221 to P0224 can be allocated, respectively. The details of the whole process are displayed in **Table 4**. As the new initial algorithm ensure the rough, semi-finishing and finishing polishing precedence constraints, the invalid solutions are excluded from the populations.

For **Table 4**,

- $[O_x - O_y]$ refers to the range of the number of unselected polishing operation-parts.
- P_{ijk} refers to the operation ID, i refers to the i th feature ID and j refers to the j th polishing operation-part.
- $[A_i - A_j]$ refers to the range of the number of the available positions and k is the random number generated from the range $[A_i - A_j]$.

- u refers to the number of unused positions.
- Green boxes represent vacant positions while gray boxes represent unavailable positions.

Table 4 Illustration of new initial algorithm

Large population size increases the accuracy of the optimization; however, it requires more search time. For same number of operations, compared to general optimisation, the valid search space for polishing process sequencing is much smaller. This is due to the strictly precedence constraint for the multiple operations for a polishing operation-part. However, beside the number of operations, the search space is highly related to the number of polishing operation-parts and the number of operations they include. Experimental evaluations showed that population size setting to around five to ten times of number of polishing operation-parts can achieve reasonable accuracy. Thus, considering the search time, in this research, the population size is setting as six times of the number of polishing operation parts the polishing process includes. For example, the population size sets to twenty-four for twenty operations polishing sequencing with four polishing operation-parts; while the population size sets to 72 for twenty operations polishing process sequencing with twelve polishing operation-parts.

4.3. Operators

There are three important generic operators needed to design in the genetic algorithm: selection operator, crossover operator, and mutation operator.

- 1) Selection operator: Selection refers to a process of selecting individuals (called parents) for the following operators (e.g., crossover and mutation) to produce the new generation. Once parents are chosen, they will have the chance to be used in genetic operations downstream such as crossover and mutation. In this research, the 'roulette wheel selection' strategy (Faris *et al.*, 2019) is used.

According to **Equation** shown below, the individuals' fitness is firstly converted to its proportional probability of selection (P_i), which represents the region of wheel.

$$P_i = \frac{1/F_i}{\sum_{i=1}^n 1/F_i}$$

Where F_i refers to the fitness of the i th individual of the populations, n is the size of the populations

Then, the individual, whose region that the fixed-point stops, is selected. Thus, it can be seen that the individuals having better fitness are more likely to be selected. Example of the roulette wheel selection method is presented in **Figure 10** and **Table 5**.

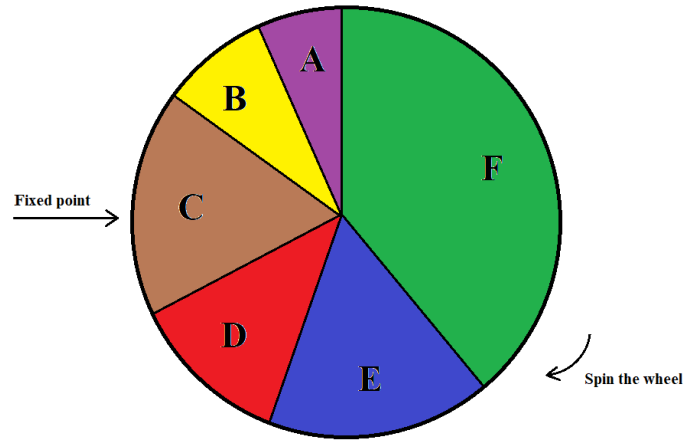


Figure 10 Example of the roulette wheel selection

Table 5 Fitness values and the according P_i values given for the example

	F_i	P_i
A	2.5	6.58%
B	2	8.23%
C	0.8	20.58%
D	1.6	10.29%
E	1.25	13.17%
F	0.4	41.15%

2) Crossover operator: Crossover refers to the genetic operator that splits and recombines two parents from the selected 'parents' population' to give birth to new solutions for the next generation. There are distinct types of crossovers based on the number of split positions and different recombining strategies between two parents. Obviously, the crossover strategies like single-point crossover, are not suitable for polishing process sequencing as it may result in an invalid sequence, i.e., some operations appear in a child more than once, or not appearing at all. In an attempt to overcome the drawback and to produce a valid process sequence, (Li, *et al.*, 2002) proposed a modified crossover strategy in which new solutions for the next generation (children) are produced following the steps described below.

- Step 1: Two selected parents are separated into a left section and a right section, breaking at a randomly selected splitting point. That is, Parents I and II are separated into four sections, namely Left-Parent I, Right-Parent I, Left-Parent II, and Right-Parent II.

- Step 2: Left-Parent I then forms the first section of Child I.
- Step 3: The second section of Child I consists of bits of Right-Parent I, in the order of appearance they have in Parent II.
- Step 4: Child II is produced in a similar way. That is, Left-Parent II forms the first section of Child II, and the second section of Child II consists of bits of Right-Parent II, in the order of appearance they have in Parent I.

The following example illustrates how a modified crossover works.

Parent I: OP1-OP2-OP3-OP4-OP5-OP6-OP7-OP8-OP9-OP10

Parent II: OP6-OP2-OP3-OP8-OP4-OP1-OP9-OP5-OP10-OP7

Assuming the splitting point is at position 5, Parent I is separated into two sections:

Left-Parent I: OP1-OP2-OP3-OP4-OP5

Right-Parent I: OP6-OP7-OP8-OP9-OP10

Bits of Right-Parent I in the order of appearance they have in Parent II:

OP6-OP8-OP9-OP10-OP7

Then, the new Child I created would be:

Child I: OP1-OP2-OP3-OP4-OP5|OP6-OP8-OP9-OP10-OP7

Similarly, parent II is also separated into two sections:

Left-Parent II: OP6-OP2-OP3-OP8-OP4

Right-Parent II: OP1-OP9-OP5-OP10-OP7

Bits of Right-Parent II in the order of appearance they have in Parent I:

OP1-OP5-OP7-OP9-OP10

Then, the new Child II created would be:

Child II: OP6-OP2-OP3-OP8-OP4|OP1-OP5-OP7-OP9-OP10

As shown in the example above, not only does the modified crossover guarantees that each operation appears in a child once and once only, but it also retains the precedence constraints used in the parents as much as possible. The crossover rate is chosen as 0.8.

- 3) Mutation: Mutation refers to the genetic operator that randomly changes one or more bits of the selected parent. To avoid an invalid sequence, the mutation strategy used in this research is: the value at two randomly selected positions is swapped. The following example illustrates how the strategy works:

Before mutation:

OP1-OP2-OP3-OP4-OP5-OP6-OP7-OP8

Assuming the two randomly selected positions are position 3 and position 6, then the result of mutation would be:

After mutation:

OP1-OP2-OP6-OP4-OP5-OP3-OP7-OP8

The mutation operator is used to prevent the populations from converging to a local minimum by providing the populations with new viable solutions that may have been lost during successive generations. However, a high mutation rate could destroy appropriate solutions and retrograde optimization into a random walk. Thus, the mutation rate is set low (i.e., 0.1).

4.4. Convergence

The iterative search may stop when the target is met. The target of this research is finding the most suitable polishing process sequencing, which means value of the fitness function reaches its lowest (in other word, no longer decreases).

5. Implementation

Based on the proposed method, a prototype system has been implemented to conduct the polishing process planning using Python 3.9.6 and corresponding packages like openpyxl and tkinter. Here is an example. **Figure 11** shows a mould, in which ten features need to be polished. The ten polishing features are further divided into twelve polishing operation-parts/groups. **Table 6** gives polishing operation selection of the example.

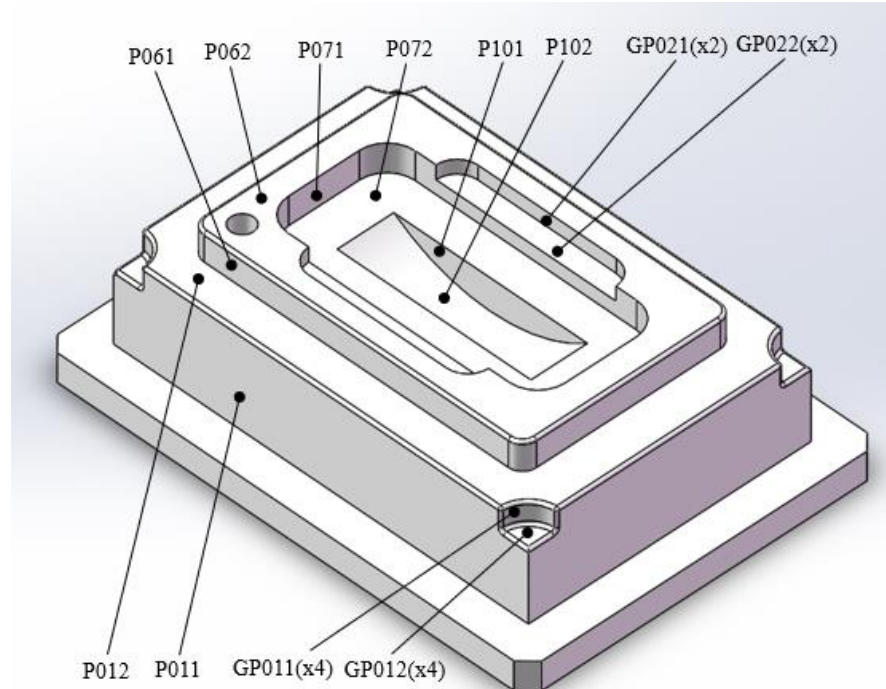


Figure 11 An example of a mould insert

Table 6 The results of polishing operation selection of the example

Operation_Part ID	Group element	Operation ID	index	Robot ID	Tool ID	Abrasive paper
P011		O0021	1	M001	T1021	400
P012		O3002	1	M001	T001	400
GP011	P021,P031,P041,P051	O0091	1	M001	T1015	400
GP011	P021,P031,P041,P051	O0092	2	M001	T1015	800
GP012	P022,P032,P042,P052	O0081	1	M001	T001	400
P061		O3110	1	M001	T1015	400
P061		O3122	2	M001	T1015	800
P062		O3002	1	M001	T001	400
P071		O3116	1	M001	T1015	400
P071		O3117	2	M001	T1015	800
P072		O1018	1	M001	T001	400
P072		O1019	2	M001	T001	800
P072		O1020	3	M001	T001	1000
GP021	P071,P081	O3121	1	M001	T1015	400
GP021	P071,P081	O3122	2	M001	T1015	800
GP022	P072,P082	O1518	1	M001	T001	400

The process sequencing is conducted based on the proposed GA-AHP optimization method. The population size, crossover and mutation rate are 72, 0.8 and 0.1, respectively. As shown in **Figure 12**, the process sequence is optimized and finally reach to its best fitness. The result is output and provides in the **Table 7**.

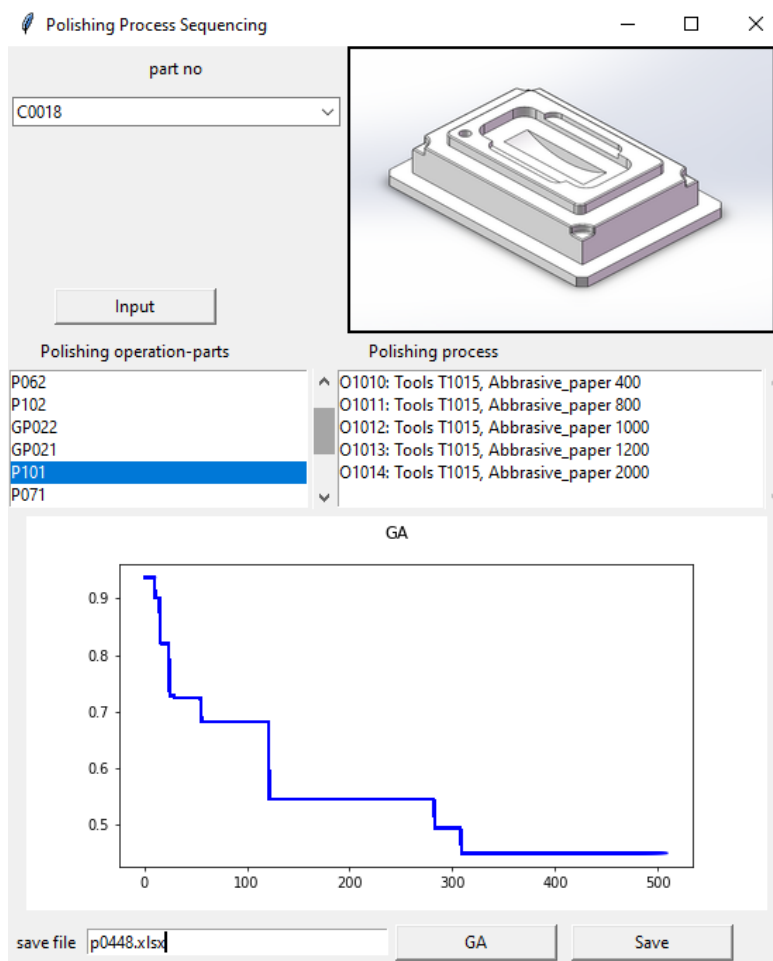


Figure 12 Process sequencing for the mould insert example

Table 7 The results of process sequence for the example

Sequence No	Operation ID	Operation Part ID	
1	O0021	P011	
2	O3116	P071	
3	O3121	P081	GP021
		P091	
4	O1010	P101	
5	O0091	P021	GP011
		P031	
		P041	
		P051	
6	O3110	P061	
7	O3002	P062	
8	O4118	P102	
9	O0081	P022	GP012
		P032	
		P042	
		P052	
10	O1518	P082	GP022
		P092	
11	O3002	P012	
12	O1018	P072	
13	O1019	P072	
14	O4119	P102	
15	O1020	P072	
16	O0092	P021	GP011
		P031	
		P041	
		P051	
17	O3117	P071	
18	O3122	P061	
19	O3122	P081	GP021
		P091	
20	O1011	P101	
21	O1012	P101	
22	O1013	P101	
23	O1014	P101	

6. Summary

For production tooling such as moulds and dies about twenty percent of the manufacture costs and around fifty percent of the manufacturing process time are apportioned to finishing operations, such as buffing and polishing. Robotics is seen as one way to reduce these costs and deal with issue of skilled workers shortage. A part of the movement towards robotic adoption is effective and automatic polishing process planning. This paper has presented a GA-AHP optimization strategy specially

for polishing process sequencing. The strategy is based on the specific constraints and rules for polishing sequencing, which previous research does not address. AHP has been employed to support multiple objectives optimization by defining the required multi-objective fitness function for the optimization, which is able to consider the best satisfaction of polishing process sequence rules and minimum polishing duration simultaneously. The proposed GA method with a new initialization algorithm and specific genetic operators optimizes the polishing process sequence until it reaches its best fitness. A prototype system based on the proposed method has been developed to demonstrate its effective.

7. Future works

It is apparent from the results that further work is necessary. Currently, the generated process planning is used to instruct robotic polishing operations, including their settings (e.g., polishing tools, abrasive papers, cutting speed, etc.) and their orders. However, the robot control code is generated manually. The research needs to be extended to interface with robot driver system to implement a completely unmanned operation environment. Integration of the polishing process plan with the robot control and drive system is required.

In addition, it is valued to explore the potential to further speed search process by combining the proposed GA-AHP optimization strategy with other algorithms, such as integrating particle swam optimization (PSO) for quick local search.

8. Acknowledgements

The authors express their gratitude to Mark Allonby (workshop technician UWE) and Nathan Churchill (technicians at BRL Sabre lab) for their assistance and facilitating the research during the COVID-19 pandemic. To the Bristol Robotics Laboratory for supporting the practical investigations and to the UWE Faculty of Environment and Technology bursary scheme for partially supporting the work.