International Journal for Quality Research 16(2) 417–428
ISSN 1800-6450

Predrag Mitic
Bogdan Nedic[1]

MULTI-HOLE DRILLING TOOL PATH OPTIMIZATION USING GENETIC ALGORITHM

Article info:
Received 21.06.2021.
Accepted 15.11.2021.

UDC – 66.011:622.243.2
DOI – 10.24874/IJQR16.02-06

Abstract: *The purpose of this paper is to present new approach in drilling path optimization based on genetic algorithm and to present a practically applicable software solution in everu day practice. It deals with multi-hole drilling problem with precedence constraints which is common case in metalworking industry. The starting point is the assumption that the geometry of the workpiece is already recognized, then the mathematical model is presented and genetic algorithm is used to generate and to optimize tool path. Proposed concept greatly reduced the costs of part production through improved machining efficiency and it is very suitable for small and medium enterprises (SMEs) which have limited resources. Proposed concept is realised through software solution implemented in object-oriented programming language Delphi, and can be used for automatic generation of NC code.*

Keywords: *Genetic algorithms, optimization, drilling tool path, multi-hole drilling*

1. Introduction

Cost optimization of production processes remains one of the major focus points of machine builders world-wide (Khatiwada et al., 2020). Machining in general and drilling in particular is one of the main production processes used to manufacture durable goods. Hole drilling is a process that uses a rotating drill bit to remove a circular cross-section of material from metallic or non-metallic materials. This process is a fundamental manufacturing process and thus is encountered in many industries and applications (Dewil et al., 2019). The same authors's state that non-cutting time can take up to 70% of the total time in the drilling process. This includes repositioning times betwen holes and tool switch times, and therefore this is an optimization problem that have significant impact on total production costs. In their research, they analysed 75 papers published between 1994 and 2017, and came to the following conclusions:

- 79% of papers dealing with basic single hole drilling problem (modelled as classic TSP-Traveller Salesman Problem)
- 13% of papers dealing with multi-hole drilling problem (modelled as PCTSP-Precedence Constraint Traveller Salesman Problem)
- 8% of papers dealing with multi-tool drilling with sequence dependent drilling time

In every day practice PCTSP optimization problem is the most common case, a since there are few studies related to this problem, there is space for new research that should be primarily related to new optimization models, and new algorithms.

Some researchers (Liu et al., 2013) indirectly deals with the PCTSP problem through process planning optimization of multi-

---

[1] Corresponding author: Bogdan Nedić
Email: nedic@kg.ac.rs

hole drilling operations, where the machining process of a hole consists of several individual operations with different machining tools, and apply ACO (Ant Colony Optimization) algorithm to solve the optimization problem. The goal of optimizaton is to minimize production costs. Khalkar et al. (2015) uses similar approach in the optimization of hole-making operations in conditions where a hole making operations drilling followed by reaming with sequence precedence and apply GA (Genetic Algorithm) to solve optimization problem.

Ghaebi and Solimanpur (2007) give the detailed mathematical model of PCTSP problem, which is modelled as a process planning of hole-making operations and the objective function is minimizing the tool airtime and tool switching time. For solving optimization problem ACO algorithm is applied. Hsieh et al. (2011) uses almost identical approach as Ghaebi and Solimanpur, the only difference is that they use PSO (Particle Swarm Optimization) algorithm for solving the optimization problem. Chen and Guo (2012) observe basic single hole drilling path optimization problem, with GA applied to solve the optimization problem.

The aim of this paper is to present the model for solving multi-hole drilling path optimization problem based on process planning of multi-hole drilling optimization, as a starting point for automatic NC code generation and a detailed GA for solving the optimization problem.

2. The problem definition and formulation

Multi-tool hole drilling deals with drilling a set of holes on a work piece where a sequence of drilling operations for each hole is determined beforehand. For example, in Figure 1 (Dewil et al., 2019) hole 1 needs to be drilled by only tool 1, hole 2 first needs to be drilled by tool 1 and then by tool 2, and hole 3 needs to be drilled by tools 1, 2, and 3 in that order.

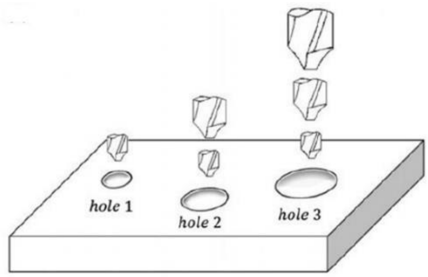


Figure 1. Multi-tool hole drilling

It is clear that hole 1 must be visited once, by the tool, hole 2 must be visited twice and hole 3 must be visited three times by the tool. The holes can have different diameter and in general tools 1, 2, 3 are different tools. In addition to the holes shown in figure 1, there is a tool magazine point in CNC machine. Figure 2 shows an operational precedence graph of the example, where 0 and 7 present the start and end of the hole drilling process. As a consequence of the example, it can be specified that the multi-tool hole drilling path optimization problem with precedence is identical to the well-known PCTSP since both of the problems shows the same characteristics (Tamjidy, 2015).

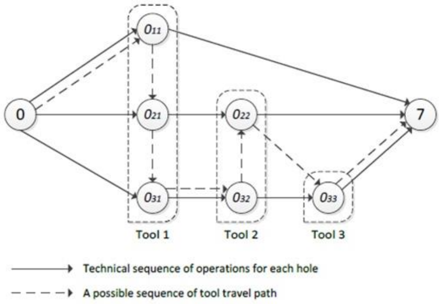


Figure 2. Technical sequence of operations

The PCTSP can be defined as follows: given a set of cities, the costs of moving from one city to another city, and a set of precedence constraints between the cities, find the shortest path that visits every city without violating a precedence constraint.

2.1. The problem modelling

It is clear that multi-hole drilling process planning optimization problem can be observed as a problem to determine the order in which a particular hole should be drilled with an appropriate tool in order to minimize the summation of tool airtime and tool switch time (Ghaebi & Solimanpur, 2007).

If the first and only operation od hole 1 (Figure 1) is denoted as O_{11}, the first operation of hole 2 O_{21}, the second operation of hole 2 O_{22} and in the same manner for hole 3, and if tool magazine point is denoted as point 0 the one accetable operation sequence for all 3 holes can be $O_{11} – O_{21} – O_{31} –O_{22} – O_{32} –O_{33}$, but the tool moving sequence will be $O_{01} – O_{11} – O_{21} – O_{31} – O_{30} – O_{22} – O_{32} – O_{30} – O_{33}$ or written with hole and tool magazine point numbers will be 0-1-2-3-0-2-3-0-3. In practice that means that spindle is moving from tool magazine with tool 1 in the spindle to hole 1, then to hole 2 and to hole 3, then goes to the tool magazine, change tool from 1 to tool 2, and goes to the hole 2, then hole 3, again goes to tool magazine for changing from tool 2 to tool 3 and with tool 3 goes to hole 3. The tool moving sequence is the tool path (Barclay et al., 2015) and it depends on the operation sequence, so it necessary first to determine the space of permissible or acceptable solutions of operation sequences.

The model, observed in this paper consist of:

Set of holes $C = \{C_0,\ C_1, C_2, \ldots, C_n\}$, where $n+1$ is the total number of holes+tool magazine point, C_h is hth hole, $h \in\{1\ldots\text{m}\}$ and C_0 is the tool magazine point in CNC machine. The center (C) of each hole(point) is given by its coordinates $C_h(x_h, y_h, z_h)$ and the position of the points is completely arbitrary.

Set of tools $T = \{T_1,\ T_2, T_3, \ldots, T_m\}$, where m is total number of tools in the magazine of CNC machine, T_t is tth tool, $t \in\{1\ldots\text{m}\}$

Set of operations $O = \{O_1,\ O_2, O_3, \ldots, O_k\}$, where k is the total number of operations in a the tool moving sequence

Ordered set of operations $\forall C_h\ h \in \{1\ldots n\}\ \exists$: $O_{hp} = \{O_{h1},\ O_{h2}, O_{h3}, \ldots, O_{hp_h}\}$, i.e. set of operations of the ith hole , where the hole has p_h operations.

Sequence of operations $l_{hp}\ \ \forall C_h\ h \in \{1\ldots n\}$ where the operations for every hole and the tool with which the operations are performed are predetermined which also means that l_{hp} operation of hth hole cannot be started until precedence operation l_{hp-1} of the same hool is completed.

The diameters of the holes can be different, as well as the technological procedure (ordered set of operations described above) for machining the holes.

One tool can be used for multiple operations.

In order to process all the holes, it is necessary to position the appropriate tool above each hth hole p_h times.

Whenever there is e need for tool change, the spindle with tool from its current position goes to the tool magazine point, change the tool, and then goes to the position above the hole which is next in sequence l_{hp} to be machined. The time needed to move from a hole to another hole is called airtime in the literature (Onwubolu & Clerc, 2004). In adition to this time, there is a time needed to move from current hole to tool magazine, and from tool magazine to another hole. In everyday practice this movement can be achieved or by moving the spindle or the machine workbench, depending of the machine structure. In the daily practice of CNC machine programming, these movements are almost always realized at constant high speed. The time needed for tool changing when spindle in positioned at the tool magazine point can be considered as a constant value for every tool.

The trajectory of the tool during the drilling process must be such that there is no collision, with parts of the clamping accessories and machine parts, but it must also satisfy the condition that there is no collision with the work piece geometry. If the coordinate z_h is excluded from the set points $C_h(x_h, y_h, z_h)$ the problem is considered as the problem of points in the plane.

Each tool path in which all holes are visited and each *ht*h hole of them is visited p_h times belongs to the space of permissible or acceptable solutions and between points the tool moves in a rectilinear constant high speed.

A 2,5 axes CNC machine moves in both *x* and *y* directions simultaneously and Euclidean distance function is to be used to calculate the distance between holes (points).

The holes are independent, and there is no predetermined order of drilling holes. One operation can be performed only with one predefined tool.

From the above, it is clear that the proposed model has *N* points that must be visited by the appropriate tool (operation=point).

2.2. Mathematical formulation of the proposed model

Let *i* and *j* be two arbitrary points (operations) from set O.

Input variables

- N number of points (operations) to be processed in CNC machine
- t_{ij} tool travelling time from operation i to operation j excluding processing time of operation j and tool change time if it is necessary $i, j = 1, \ldots,$ N
- the distance matrix D=[d_{ij}] between the points of the set O $i, j = 1$, ..., N
- constant high speed F_{bh}
- constant tool change time t_c
- constant high speed F_{bh}
- tool T_j for processing of operation O_j

Control variables of the mathematical model

In addition to the observed model's input variables, it is necessary to define a vector of control variables, that is, variables that describe the optimisation objectives stated.

$$X_{ij} = \begin{cases} 1 if\ tool\ T_i\ travels\ from\ i\ to\ j \\ 0 \qquad otherwise \end{cases} \tag{1}$$

$$Y_{ij} = \begin{cases} 1 if\ \text{j}\ is\ performed\ with\ tool\ T_i \\ 0 \qquad otherwise \end{cases} \tag{2}$$

$$P_{ij} = \begin{cases} 1 if\ i\ is\ precedence\ of\ j \\ 0 \qquad otherwise \end{cases} \tag{3}$$

u_i , Positive variable to avoid sub-tours; i = 0, ... , N (4)

Constraints of mathematical model

1) Constraint that ensure that every operation from set O is performed once on CNC machine

$$\forall j \in \{0 \ldots N\} \sum_{i=0}^{N} X_{ij} > 0 \tag{5}$$

2) Constraint that ensure the elimination of sub tours

$$u_i - u_j + N \cdot X_{ij} \leq N - 1 \tag{6}$$

Variables u_i and u_j are integer variables that define the order of points visited on a tour. This means that each point receives a number label, and these number should be sequential-MTZ formulation of sub-tour elimination constraints (Bazrafshan et al., 2021).

3) Constraint that ensures operation precedence for the tool T_j

$$u_j - u_i \geq 1 \quad i, j = 0, \ldots, N \quad P_{ij} = 1 \tag{7}$$

The objective function

The objective function of the mathematical model represents the criteria of optimisation (Singiresu, 2009).

As mentioned earlier in this paper, the objective of multi-hole drilling process planning optimization is to minimize airtime and tool switch time. In the proposed model under tool change time is considering the time of tool change in the magazine of CNC machine when the spindle is positioned in tool magazine point and this time is considering constant. But apart this time there is time needed for tool moving from point *i* to point 0 (tool magazine point) and from point 0 to point j if $T_i \neq T_j$ and this time must be considered because it depend from current tool position and the position of next point (operation) in operation sequence. The Euclidean distance between the points *i* and *j* and is given by

$$d_{ij} = \sqrt{\left(|x_j - x_i|\right)^2 + \left(|y_j - y_i|\right)^2} \qquad (8)$$

and the objective function can be written as

$$\begin{aligned} &\min \sum_{i=0}^{N}\sum_{j=0}^{N} t_{ij} \cdot X_{ij} + \\ &min \sum_{i=0}^{N}\sum_{j=0}^{N} (t_{i0} + t_{0j}) \cdot Y_{ij} \end{aligned} \qquad (9)$$

As the tool moves at a constant high speed F_{bh} time t_{ij} can be written as

$$t_{ij} = \frac{d_{ij}}{F_{bh}}$$

and as a consequence objective function can be written as

$$\begin{aligned} &\min \sum_{i=0}^{N}\sum_{j=0}^{N} d_{ij} \cdot X_{ij} + \\ &min \sum_{i=0}^{N}\sum_{j=0}^{N} (d_{i0} + d_{0j}) \cdot Y_{ij} \end{aligned} \qquad (10)$$

The objective function (10) minimizes the length of tool path, and as every point (operation) is determined with x and y coordinate also the drilling tool path is generated and optimized. If operation belong to the same hole then d_{ij}=0, and if the same tool is used for two adjacent operations then $d_{i0} + d_{0j} = 0$.

3. Proposedi GA for drilling tool path optimization

An acceptable drilling tool path represents permutations of drilling operations while respecting the constraints defined in the mathematical model which is basically an integer programming model. Genetic algorithms, as one of the modern, metaheuristic methods, are very suitable for solving this type of problem because in most cases they can find the global optimum with a very high probability (Singiresu, 2009). It should be noted that the efficiency of the genetic algorithm depends on the applied genetic crossover and mutation operators (Umbarkar & Sheth, 2015). For the PCTSP problems there are a number of these operators in the literature, which can be modified depending on the model being observed, thus making the genetic algorithm adaptable to the specifics of the model being observed.

There are many literature sources related to this method, i.e. the GA's mechanism is widely known (Vaupotic et al., 2013). Furthermore, only short definitions of GA-related terms and explaining the definitions in the observed model's context are given.

3.1. Definition of GA related terms in the context of observed model

A gene in the observed model is the primary carrier of informations i.e. points (drilling operations). The point is denoted as P_{hs} and every that should be visited by the tool contain the following informations:

- affiliation to hole and (index h in P_{hs})
- sequence of operation of *h*th hole (index *s* P_{hs})
- the tool T_s with which operation P_{hs} al.is performed
- x_h, y_h *x* and *y* coordinate of the *h*th hole

An individual or chromosome is represented by a combinations of genes and it is acceptable solution. In the observed model. this will be accetable tool path.

The population is a set of individuals, and in the observed model, it will be a set of all acceptable tool paths.

The parents are two accetable tool paths that combine to create new tool path.

Fitness is a function that tells us how good each tool path is. In the observed model, the path length is the fitness and it is defined by expression (10) in the mathematical model.

Crossover is the genetic operator defined as combining two individuals, i.e. tool paths to create new tool path.

The mutation is the genetic operator defined as the process in which one individual or one tool path participates, and the goal is to generate a new tool path. A mutation changes the value of one or more genes on a chromosome. The process of implementation of genetic operator's crossover and mutation in the observed model will be explained later.

3.2. Chromosome representation and decoding

In the construction of a genetic algorithm, the first step is to define an appropriate genetic representation or an appropriate coding method (Đorđević et al., 2013). An acceptable representation is the most critical factor influencing all other phases of the GA (Singiresu, 2009). It primarily depends on the model being observed so that the chromosome coding solution with a vector of real components will be given.

As mentioned earlier in this paper the geometry of working piece is already recognized, the sequence of operations and tools with which operations are performed for every hole are predetermined.

The coordinates of every hole is known so the distance matrix D=[d_{ij}] can be easily determined. Each point/operation is assigned ordinal number which uniquely defines the operation. In addition to this number, each point is assigned a hole number and the number of operations in the hole which represents the sequence of operation. A record of fully decoded chromosome is given in Table 1 for the example shown in Figure 1.

Table 1. Decoded chromosome

Hole	1	2	3	2	3	3
Point	1	2	4	3	5	6
Operation	1	2	3	2	2	3
Sequence	1	1	1	2	3	3
Tool	T_1	T_1	T_1	T_2	T_2	T_3
X coord.	10	25	98	25	98	98
Y coord.	15	11	25	11	25	25

As shown in Table 1, the individual is written in form of integer array 1-2-4-3-5-6, the precedence constraints are defined (for example operation 3 which is first operation of hole No.2 O_{21} must be performed before the operation 3 which is the second operation of hole No. 2 O_{22}) and this represent one accetable solution. All permutations without repetition of numbers from second row, respecting defined constraints represents all accetable solution of optimization problem.

3.3. Initial population

The initial population is formed by defining the hole operation's schedule as an integer array over the numbers of points, as shown in Table 1. The members of the integer array are the numbers of point from number 1 to total number of points N. The array members are selected randomly, taking into account the precedence constraints between operations in the holes. All members of the integer array have different numbers. The problem with generating the initial population is the implementation of constraints. In the observed model this problem is easily solved by comparing the affiliation of the operation to the hole based on the sequence contained in the gene and by checking if previous operation is chosen. The pseudo code is shown below:

```
for i=1, N
    genOK=false
while genOK=false do
begin
gen=random number(1, N)
pgen=previous gene in operation sequence
```

```
if gen.hole=pgen.hole then
 for j=1, i-1
   if path(J)=gen then genOK=true
 if gen.sequence=1 then genOK=true
end
path(I)=gen
next i
```

The first gene in every individual must be first operation of randomly chosen hole. If randomly chosen number does not satisfy the precedence constraints, the process is repeated until the appropriate number is not selected. In this way initial population is formed taking in account all constraints defined in the mathematical model.

3.4. Fitness calculation

Fitness is a function that defines how short the distance from the start to the end point is. This implements the goal function given by expression (10) in the mathematical model. The calculation of the fitness is done for each path in the population. The algorithm for calculating fitness is simple and in fact represents the calculation of the variable *path length*. The distance between every two adjacent points from a series of points is calculated, and these distances are added together. At the end, by reading decoded chromosome, whenever there is a need for tool change, the path between previous operation to tool magazine point, and from tool magazine point to current point is added and in that way, the calculation of the path length is completed. The fitness, i.e. the length of the trajectory is calculated for each path from the population. In the end, the best route is calculated from the current population and its ordinal number is placed in the variable *bestpath*. This determines the best individual from the current population as the individual with the shortest path length, ie the optimized tool path, which contains an array of hole numbers in the order in which the tool moves during drilling.

3.5. The parents

The parents represent two toolpaths that combine to create new toolpaths. The selection of parents or paths is done by comparing their fitness. It is clear that the parent should be the path that has the shortest length (the greatest fitness), and when this is determined by comparing with the fitness of other paths, then the logical variable *parent* is assigned the value of truth, ie that path is chosen in the set of paths for crossover. The comparison is repeated until the number of iterations defined by the size of the set of parents, which represents the input size of the genetic algorithm, is fulfilled.

3.6. Crossover operator

The crossover is a process in which two individuals are combined to obtain new individuals, ie the selection process selects parents and the newly created individuals are children. The genetic material of a child is a combination of the genetic material of both parents. with the problem that in the case of drilling path optimization problems, the holes can only be included once in the new path, so specific crossover operators are used to implement the crossing. The reviews of the most common crossover operators used in GA are given in (Ngyen et al., 2002) in (Wei, 2006) and in (Padmavathi & Yadlapalli 2017). It should be noted that for multi-hole drilling path optimization problem must be used crossover operator which preserve order. The Order-base Crossover (OX1) operator, was chosen for the realization of crossover operation in this paper. It constructs an offspring by choosing a substring of one parent and preserving the relative order of the elements of the other parent. For example, the following two parent strings: (1 2 3 4 5 6 7 8) and (2 4 6 8 7 5 3 l), and suppose that a first cut point between the second and the third bit and a second one between the fifth and the sixth bit is selected. Hence, (1 2 - 3 4 5 - 6 7 8) and (2 4 - 6 8 7 - 5 3 1). The offspring are created in the following way. Firstly, the string segments between the cut point are copied into the offspring, which give (* * 3 4 5 * * *) and (* * 6 8 7 * * *). Next, starting from the second cut point of one parent, the rest of the elements are copied in the order in which they appear in the other parent, also starting from the second cut point and omitting the elements that are already present. When the end of the parent string is reached, we continue from its first position. In those example, this gives the following children: (8 7 - 3 4 5 - 1 2 6) and (4 5 - 6 8 7 - 1 2 3).

In this paper, OX1 operator is slightly modified. Instead of passing on every newly born individual to the next generation, a new crossover is performed regardless of whether it fits better than its parents do. An individual is passed on to the next generation only under the condition that its fitness is greater than or equal to one of the parents' fitness, i.e. only if the newly formed individual is better than one of both parents in its characteristics. If the probability is less than 50%, individuals' transfer to the new generation is done without comparison, and there is no rejection. If the probability is greater than 50%, the individual worse than both parents is rejected. In this way, the next generation's diversity is maintained (because not only the best individuals are transferred to the next generation, and at the same time, the number of iterations and execution time of GA is significantly reduced.

3.7. Mutation operator

A swap mutation operator is chosen in this model, slightly modified due to the necessity to preserve the sequence of operations. The position of genes is randomly chosen. If these genes are the same i.e. if belong to the same hole then the mutation does not make sense (there will be an additional tool change) On the other hand, if the genes are not the same or do not belong to the same hole the mutation makes sense. Since the sequence of operations must be preserved within the hole, every chosen gene is check according to the same procedure shown in 3.3.

3.8. Pseudo code of proposed GA

Input parameters for GA shown in this paper are: N-number of points/operations, the population size, the number of parents, mutation rate and the number of generations.

```
Start
Enter the input parameters
Create Initial population
    Calculate the fitness of individuals of
    the first generation
    Generation = 1
        Repeat
            Selection of parents
            Children = 0
            Repeat
        Pick of two parents for a crossover
        (Parent1, Parent2)
    Crossover OX1 (Parent1, Parent2)
    Mutation
    Children = Children + 1
    Until Children = population size-total
    number of parents
    Generation = Generation + 1
    Calculate the fitness of individuals for
    the current generation
Until generation = total number of
generations
End.
```

4. Presentation of the independent software solution and experimental results

The application in which the previously presented solution is implemented is written in the object-oriented programming language Delphi. A database, i.e. a table with an example used to execute the application and display experimental results, was created in Microsoft Access, and it consists of one specific table. The user can change the table, entering new hole, operations and tools, so that it is possible to test the displayed solutions for different data. Also, it is possible to increase or decrease the number of tools in the database. The table is arranged to correspond to chromosome representation shown in Table 1.

Figure 3 shows the initial form of the application. The application consists of several parts:

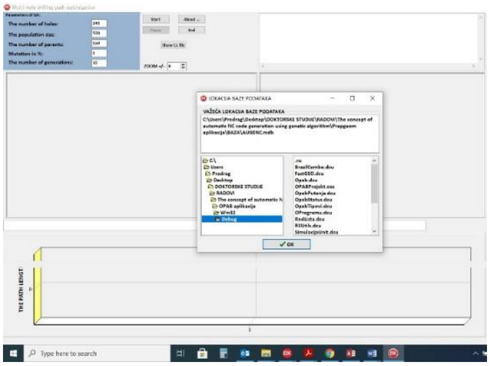


Figure 3. The initial form of the application

In the upper left corner is the part for entering the genetic algorithm's input parameters, in the upper-right part, there is a display of input data for the genetic algorithm's execution. Immediately after loading the application, a window appears to enter the path to the same database so that the input data set for executing the application can be read. As the first step in using the application, it is necessary to select the path to the database;

The left panel represents the space in which the graphic representation of the current drilling path is simultaneously displayed.

The right panel showing current data and graphical display during GA execution and the lower central part of the input screen represents the space in which, during and after the genetic algorithm's execution, the GA execution results are displayed. Window in the upper right corner displays, after the proces of generating and optimizing the tool path) the data necessary for NC code creation (drilling sequence, hole coordinates and required tools) as sort of CL (Cutter Load) file prepared for the processing in the appropriate post processors of CNC machines.

4.1. Experimental testing of proposed GA

For the experimental testing of the application the industrially inspired part was chosen (Figure 4).

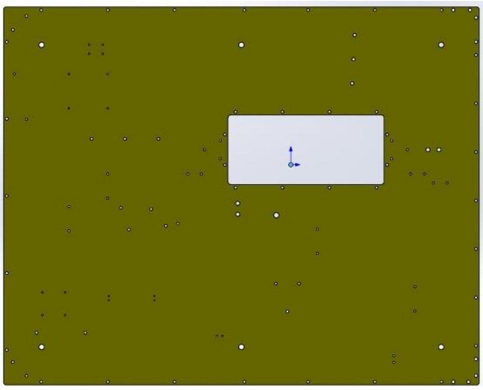


Figure 4. The part for experimental testing

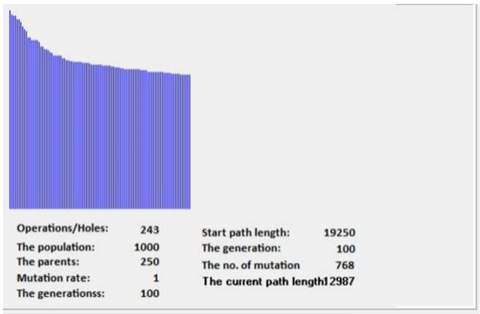
There are 104 holes in the part shown in Figure 4. The holes are:

- 4 holes M3
- 14 holes Ø5
- 22 holes M8
- 4 holes M10
- 42 holes Ø9
- 11 holes Ø5,5
- 2 holes Ø13,5
- 2 holes Ø17,5

Tools needed are centre drill, drills for every hole dimension and thread tappers for tapping the holes in total 13 tools. The part named Base plate is made form carbon steel, dimensions are 1567 x 1247 mm. In total there are 243 points (operations) considering technological aspect of presented part.

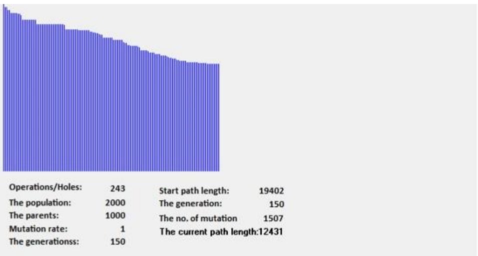
Parameters for GA execution and obtained result are shown in Table 2. GA is performed 4 times, with the parameters defined in columns I, II, III and IV. The results are shown in Figures 5, 6, 7 and 8 corresponding to the set of parameters I to IV, respectively.

The goal of testing was to find optimal input parameters for GA, and to measure the execution time, to see if it is suitable for application in real systems.



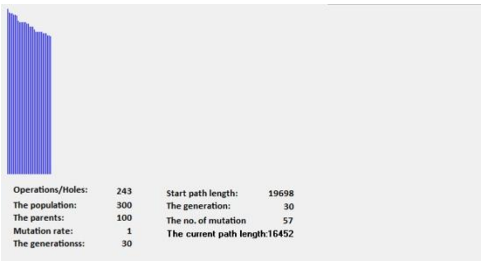
```
Operations/Holes:      243    Start path length:        19250
The population:       1000    The generation:             100
The parents:           250    The no. of mutation         768
Mutation rate:           1    The current path length12987
The generationss:      100
```

Figure 5. Execution results-group of parameters I



```
Operations/Holes:      243    Start path length:        19402
The population:       2000    The generation:             150
The parents:          1000    The no. of mutation        1507
Mutation rate:           1    The current path length:12431
The generationss:      150
```

Figure 6. Execution results-group of parameters II

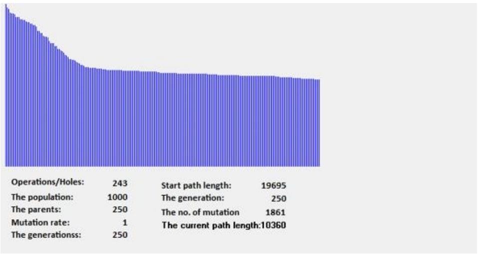


```
Operations/Holes:      243    Start path length:        19698
The population:        300    The generation:              30
The parents:           100    The no. of mutation          57
Mutation rate:           1    The current path length:16452
The generationss:       30
```

Figure 7. Execution results-group of parameters III

4.2. Results and discussion

The execution of GA with the first group of parameters (Figure 5) characterizes relatively fast execution time and path length of 12889 mm after the optimization process. The path length value is not stable because there is still a slope of the curve.



```
Operations/Holes:      243    Start path length:        19695
The population:       1000    The generation:             250
The parents:           250    The no. of mutation        1861
Mutation rate:           1    The current path length:10360
The generationss:      250
```

Figure 8. Execution results-group of parameters IV

There were significant number of mutations, but not as high as for the second and fourth execution of GA.

The second group of parameters characterizes very large number of paths in the initial population. The optimization result is slightly better than those for the first execution-difference is less than 5% in path length but the time of execution and the number of the mutations are significantly higher. The time of execution was 90 seconds respect 36 seconds for the first execution. The path length in the last seconds of the execution has changed very little, and the GA curve begins to flatten which means that global optimum is near.

The third execution of GA is characterized with significantly smaller population (only 300 paths) and significantly smaller number of generation. This graphic shown in Figure 7 shown the nature of GA-the path length is very far from global optimum because there is not enough manipulation with genetic material (the number of mutations are only 57). The conclusion is obvious. The proposed GA does not give satisfactory results for the shown input parameters and can't be used with small number of path in the initial population and the small number of generations.

The true nature and the strength of GA, in a positive sense is shown in Figure 8. For the same number of paths in the initial population, but significantly smaller number of parents, with high number of generations the best results are achieved with the time of

Table 2. The parameters for the execution of GA and experimental results

	I	**II**	**III**	**IV**
Number of points/operations	243	243	243	243
Population size	1000	2000	300	1000
Number of parents	250	1000	100	250
Mutation %	1	1	1	1
Number of generations	100	150	30	250
Path length in mm	12887	12431	16452	10360
Execution time of GA in seconds	36	90	10	92

execution of 92 seconds. The path length from its initial value of 19650 mm is reduced to 10360 mm which is almost 50% of reduction. The time of execution of 92 seconds, and considering that there was 243 points/operations make this algorithm very suitable for for practical application. Literature sources cited in this paper give an experimental result for max 42 points (Liu, et al, 2013) so comparing to the experimental results from the literature sources presented algorithm is very successful in solving PCTSP problem.

5. Conclusion

Based on the literature review it can be concluded that there is still space for new research in the field of drilling path optimization. In multi hole drilling processes, it is very important to find the best tool path which can reduce waste in non-cutting time.

In the other hand, there is a need there is a for as much automation as possible in the process of programming CNC machines which leads to a great cost reduction. The model presented in this paper, can be a starting point for completely autonomous CNC control unit. Presented algorithm is very effective in the terms of time needed for the execution, and as such, very practical practical for applicatin in small and medium enterprises. In addition to drilling, it can be used practically without any changes in the process of punching and cutting, to determine the position of tools, and its application as a subject of further research can be found in the process of laser cutting and water jet cutting because basically in these types of processing positioning machines and / or tools take place in a similar way as in the drilling process.

References:

Barclay, J., Dhokia, V., & Nassehi, A. (2015). Generating Milling Tool Paths for Prismatic Parts Using Genetic Programming. *Procedia CIRP*, *33*, 490-495. https://doi.org/10.1016/j.procir.2015.06.060

Bazrafshan, R., Zolfani, S., H., & S. Mirzapour Al-e-hashem, S. M. (2021). Comparison of the Sub-Tour Elimination Methods for the Asymmetric Traveling Salesman Problem Applying the SECA Method. *Axioms 2021, 10*(1), *19.* https://doi.org/10.3390/axioms10010019

Chen, J. M., & Guo, W. G. (2012). Path Optimization of the Drilling Hole Based on Genetic Algorithm. *Advanced Materials Research, 497,* 382–386.

Dewil, R., Küçükoğlu, I., Corrinne, L., & Cattrysse, D. (2019). A critical review of multi-hole drilling path optimization. *Archives of Computational Methods in Engineering*, *26*, 449-459. https://doi.org/10.1007/s11831-018-9251-x

Đorđević, A., Stefanović, M., Arsovski, S., Erić, M., Aleksić, A., & Nestić, S. (2013). Optimizacija obradnog procesa struganja korišćenjem genetskog algoritma. *40 Nacionalna konferencija o kvalitetu*, Fakultet inženjerskih nauka Univerziteta u Kragujevcu.

Ghaiebi, H., & Solimanpur, M. (2007). An ant algorithm for optimization of hole-making operations. *Computers & Industrial Engineering*, *52*(2), 308-319.

Hsieh, Y-C., Lee, Y-C., You, P-S. & Chen, T-C. (2011). Optimal Operation Sequence of Hole-Making with Multiple Tools in Manufacturing: a PSO Evolutionary Based Approach. *Key Engineering Materials, 460-461*, 398-403.

Khalkar, S., Yadav, D. & Singh A. (2015). Optimization of hole making operations for sequence precedence constraint. *International Journal of Innovative Emerging Research Engineering*, *2*(7), 26-31.

Khatiwada, D., Nepali, D., Raj, N., C., & Bhattarai, A. (2020). Tool path optimization for drilling holes using genetic algorithm. *International Journal of Machine Tools and Maintenance Engineering, 1*(1), 36-42.

Liu, X., Hong, Y., Zhonghua, N., Jianchang, Q., & Xiaoli, Q. (2013). Process planning optimization of hole-making operations using ant colony algorithm. *International Journal of Advanced Manufacturing Technology, 69*, 753-769.

Ngyen, H. D., Yoshikara, I., Yamamori, K., & Yasunaga, M. (2002). Greedy genetic algorithms for symmetric and assymetric TSP, *IPSJ Transactions on Mathematical Modeling and Its applications*, *43,* 165-175.

Onwubolu, G. C., & Clerc, M. (2004). Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization. *International Journal of Production Research*, *42*(3), 473-491.

Padmavathi, K. & Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. International Journal of Computer Applications, *162*(10), 34-36.

Singiresu, S. R. (2009). *Engineering optimisation-theory and practice*, Fourth edition. Hoboken, New Jersey: John Wiley & Sons, Inc.

Tamjidy M. (2015). Biogeography based optimization (BBO) algorithm to minimize non-productive time during hole-making process. *International Journal of Production Research*, *53*(6), 1880-1894. https://doi.org/10.1080/00207543.2014.965356

Umbarkar, A. J., & Sheth, P.D. (2015). Crossover operators in genetic algorithms: A review, Department of Information Technology. *ICTACT Journal on Soft Computing, 06*(01), 1083-1092. https:// doi: 10.21917/ijsc.2015.0150

Vaupotic, B., Kovacic, M., Ficko, M., & Balic J. (2006). Concept of automatic programming of NC machine for metal plate cutting by genetic algorithm method. J*ournal of Achivements in Materials and Manufacturing Engineering*, *14*(1), 131-139.

Wei, J-D. (2006). Approaches to the Travelling Salesman Problem using evolutionary computing algorithms, *Chang-Gung University Taiwan*. https://doi:10.5772/5584

Predrag Mitic
University of Kragujevac
Faculty of Engineering,
Kragujevac,Serbia
predrag2904@gmail.com

Bogdan Nedic
University of Kragujevac
Faculty of Engineering,
Kragujevac, Serbia
nedic@kg.ac.rs
ORCID 0000-0002-4236-3833