






RESEARCH ARTICLE

## Application of precedence constrained travelling salesman problem model for tool path optimization in CNC milling machines

Ilker Kucukoglu , Tulin Gunduz\* , Fatma Balkancioglu , Emine Chousein Topal   
Oznur Sayim 

Department of Industrial Engineering, Bursa Uludag University, Turkey  
[ikucukoglu@uludag.edu.tr](mailto:ikucukoglu@uludag.edu.tr), [tg@uludag.edu.tr](mailto:tg@uludag.edu.tr), [fatmabalkancioglu@gmail.com](mailto:fatmabalkancioglu@gmail.com), [emine.c.topal@gmail.com](mailto:emine.c.topal@gmail.com),  
[ssayim16@gmail.com](mailto:ssayim16@gmail.com)

### ARTICLE INFO

Article history:  
*Received: 3 August 2018*  
*Accepted: 28 March 2019*  
*Available Online: 28 May 2019*

Keywords:  
*Tool path optimization*  
*Mathematical modelling*  
*Travelling salesman problem*  
*Combinatorial optimization*  
*Satin Bowerbird Optimizer*

AMS Classification 2010:  
*90B23, 90B56, 90C27, 90C90*

### ABSTRACT

In this study, a tool path optimization problem in Computer Numerical Control (CNC) milling machines is considered to increase the operational efficiency rates of a company. In this context, tool path optimization problem of the company is formulated based on the precedence constrained travelling salesman problem (PCTSP), where the general form of the TSP model is extended by taking the precedence of the tool operations into account. The objective of the model is to minimize total idle and unnecessary times of the tools for internal operations. To solve the considered problem, a recent optimization algorithm, called Satin Bowerbird Optimizer (SBO), is used. Since the SBO is first introduced for the global optimization problems, the original version of the SBO is modified for the PCTSP with discretization and local search procedures. In computational studies, first, the performance of the proposed algorithm is tested on a well-known PCTSP benchmark problems by comparing the proposed algorithm against two recently proposed meta-heuristic approaches. Results of the comparisons show that the proposed algorithm outperforms the other two competitive algorithms by finding better results. Then, the proposed algorithm is carried out to optimize the hole drilling processes of three different products produced by the company. For this case, with up to 4.05% improvement on the operational times was provided for the real-life problem of the company. As a consequence, it should be noted that the proposed solution approach for the tool path optimization is capable of providing considerable time reductions on the CNC internal operations for the company.



### 1. Introduction

Nowadays, to achieve an effective and efficient production system for a company is very important due to the tough competition conditions. In this context, one of the most important factors that provide efficient production is increasing capacity utilization in operations, which directly affects factory production efficiency. With regards to the innovations in the field of mechanical engineering, Computer Numerical Control (CNC) machines are mostly employed in the various manufacturing process to increase production efficiency [1-3].

The total production time for a part in CNC machines basically consists of two components: The machining time when the tool is actually cutting material and the non-productive time when the tool is travelling in the air or the tool is switched in the magazine [4].

Regarding the machining time, it should be noted that there exist many researches have been studied in the literature to reduce total production time by optimizing the machining parameters [5, 6]. On the other hand, fewer studies are introduced in the field of optimization to reduce non-productive times in CNC machines [4]. One of the critical issues in process planning of CNC machines to reduce non-productive times is tool path planning for machining since the tool path generation in current CAM technology is still based on the only geometric computations and away from being an optimum manufacturing process, which may lead a considerable increase on total production time [7]. The tool path planning is simply named in the literature as tool path optimization problem (TPOP) and results in travelling salesman problem (TSP) as each tool-path contour can be considered as a city coordinate to be visited [8]. Since the TSP is known as an NP-Hard

combinatorial problem, the TPOP can also be considered as a complicated problem to solve.

The TPOP can be investigated as the extension of the hole drilling path optimization problem (HDPOP). The HDPOP simply considers the routing a single bit over a workpiece in such a way that all holes are visited in the fastest manner [9]. The HDPOP is classified into two versions concerning the number of tools required for the machining operations on the workpieces:

- Single tool hole drilling path optimization problem (STHDPOP)
- Multi-tool hole drilling path optimization problem (MTHDPOP)

The STHDPOP is the most basic version, where every hole has to be drilled by a single specific tool. Due to only one type of tool is used, a tool switch is not required for the operations. However, the processing time of each hole can be different with respect to hole depth [9].

In case the holes on a workpiece require different diameter or different type of finishing, the STHDPOP shade into the MTHDPOP. As a result of multiple tool usage, tool switch and tool travel times have to be taken into account [10, 11].

A detailed literature review on HDPOP is presented by Dewil et al. [9] and Abidin et al. [10], in which various heuristic approaches proposed to solve HDPOP are listed. Considering both single and multi-tool hole drilling path optimization problems, well-known simulated annealing algorithm [12], tabu search algorithm [13] ant colony optimization algorithm [11, 14, 15], particle swarm optimization algorithm [16-18], genetic algorithm [19-23] are proposed. In addition to these algorithms, various recent algorithms are also implemented to HDPOP, such as biogeography based optimization [24], charged system search algorithm [25], hybrid cuckoo search-genetic algorithm [26], shuffled frog leaping algorithm [18], optimal foraging algorithm [27], etc.

According to the review study of Abidin et al. [10], in which 41 papers are taken into account, the modelling approaches of HDPOP can be classified into three main groups: TSP based models, precedence constrained travelling salesman problem (PCTSP) based models, and travelling cutting tool problem (TCP) based models. From the reviewed papers, the TSP concept is the most widely used in modeling the HDPOP about 92%, while 5% of them consider the PCTSP and 3% of them consider the TCP. A similar result is reported by Dewil et al. [9] that only 7 of 53 reviewed papers consider the PCTSP based model.

In this study, TPOP in CNC milling machines of a company is taken into account to increase the internal operational efficiency of the machines. The mathematical model of the problem is formulated based on PCTSP. To solve the TPOP efficiently, a newly developed bio-inspired optimization algorithm called Satin Bowerbird Optimizer (SBO) is used. SBO is first

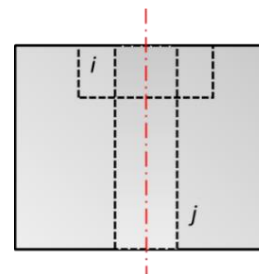
introduced by Moosavi and Bardsiri [28] to efficiently estimate software development effort. Since the original version of the SBO is introduced for the global optimization problems, the SBO is modified with two components: Discretization procedure for representing a solution for TPOP, and local search procedure. In this context, the main contributions of the proposed study are as follows:

- The TPOP is formulated based on the PCTSP.
- To the author' knowledge, this is the first application of the SBO in a combinatorial optimization problem.
- The original version of the SBO is modified with discretization and local search procedures.
- Detailed comparisons are presented for the proposed SBO with statistical significance tests.
- A real-life application of the SBO to the TPOP is carried out by considering hole drilling processes of three different products produced by the company.

The rest of the paper is organized as follows. In Section 2, the TPOP is described with the considered assumptions. Mathematical formulation of the TPOP is presented in Section 3 as a mixed integer mathematical model. Details of the proposed algorithm are given in Section 4. Section 5 presents the computational results of the proposed algorithm. Finally, conclusions are given in Section 6.

## 2. Problem definition

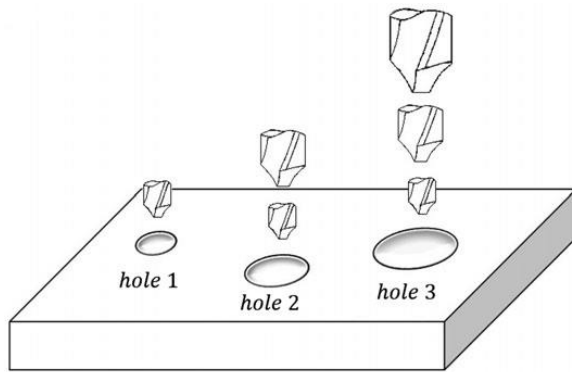
The TPOP addressed in this study is the extension of the MTHDPOP in which precedences of the operations are taken into account. In literature, this problem is called a multi-tool hole drilling path optimization problem with precedence constraints (MTHDPOP-PC). In MTHDPOP-PC, a specific sequence of drilling operations is defined for each hole [24]. Figure 1 shows an illustrative tool operation for the MTHDPOP-PC, where operation  $i$  has to be completed before operation  $j$ . In this case, the operation  $i$  can be defined as the precedence of operation  $j$ .



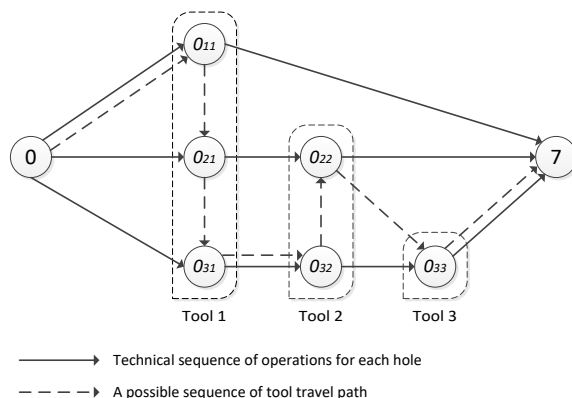
**Figure 1.** Illustrative representation of precedence drilling operations.

With regards to the precedence constraints, Figure 2a presents an example part of tool sequence for multi-tool hole drilling, where the sequence of operations for the holes are given in Table 1. According to the given

operational sequences, hole 1 ( $o_{11}$ ) needs to be drilled by only tool 1, hole 2 ( $o_{21}$  and  $o_{22}$ ) needs to be drilled by tools 1 and 2, respectively, and hole 3 ( $o_{31}$ ,  $o_{32}$  and  $o_{33}$ ) needs to be drilled by tools 1, 2, and 3 in that order. For better visualization, Figure 2b 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.



(a) Example part of multi-tool hole drilling.



(b) Technical sequence of operations.

Figure 2. Example for the MTHDPOP-PC [24].

Table 1. Sequence of operations for illustrative example.

Hole	Tool Sequence	Sequence of operation ( $o_{ij}$ )
Hole 1	1	$o_{11}$
Hole 2	1, 2	$o_{21} - o_{22}$
Hole 3	1, 2, 3	$o_{31} - o_{32} - o_{33}$

Based on the MTHDPOP-PC, the TPOP can be described as finding the best operational plan for the CNC milling machines that minimizes total idle and unnecessary times of the tools for internal operations regarding the following assumptions:

- In addition to the hole making, different type of milling operations in CNC milling machines (such as reaming, boring, counterboring, tapping, etc.) are considered on the workpiece.
- The production process of a workpiece starts and ends at the magazine.

- Each milling operation may need a pre-process before it starts. In other words, a milling operation cannot be processed until its precedence operations are completed.
- Each milling operation can be processed by only one type of tool and can be completed in one pass.
- Tools can be used for multiple milling operations.
- In case a tool switch operation is required between two milling operations, the spindle has to visit the magazine.

### 3. Model formulation

Since the MTHDPOP-PC is identical to the PCTSP, which is shown in the previous section, the mathematical model formulation of the TPOP is proposed based on the PCTSP model introduced by Kubo and Kasugai [29]. The proposed model is the extension of the well-known TSP model, where the general form of the TSP model is modified by taking the precedence of the tool operations into account. In this context, the proposed mathematical model of the TPOP is formulated as follows:

#### Notations

0 Tool magazine point in CNC machine

$N$  Number of points on the product to be processed in CNC machine. This can also be assumed as the number of milling operations operated in the CNC machine.

$t_{ij}$  Travelling time from operation  $i$  to operation  $j$  including processing time of operation  $i$  and tool switch time at the magazine if the tool is switched;  $i, j = 0, \dots, N$

$p_{ij}$  Binary data and 1 if milling operation  $i$  is precedence of milling operation  $j$ ;  $i, j = 1, \dots, N$

#### Decision Variables

$x_{ij}$  Binary variable and equals to 1 if the tool travels from operation  $i$  to operation  $j$ , 0 otherwise;  $i, j = 0, \dots, N$

$u_i$  Positive variable to avoid sub-tours;  $i = 0, \dots, N$ .

#### Model

$$\text{Min } Z = \sum_{i=0}^N \sum_{j=0}^N t_{ij} x_{ij} \quad (1)$$

s.t.

$$\sum_{\substack{j=0 \\ i \neq j}}^N x_{ij} = 1 \quad i = 0, \dots, N \quad (2)$$

$$\sum_{\substack{j=0 \\ i \neq j}}^N x_{ji} = 1 \quad i = 0, \dots, N \quad (3)$$

$$u_i - u_j + Nx_{ij} \leq N - 1 \quad i \neq j \quad i, j = 1, \dots, N \quad (4)$$

$$u_j - u_i \geq 1 \quad i, j = 1, \dots, N \quad p_{ij} = 1 \quad (5)$$

$$x_{ij} \in \{0,1\} \quad i, j = 0, \dots, N \quad (6)$$

$$u_i \geq 0 \text{ and integer} \quad i = 0, \dots, N \quad (7)$$

Eq. (1) defines the objective function of the model which aims to minimize total operational times in CNC milling machine. Eq. (2) and Eq. (3) ensure that each operation is operated once in the CNC milling machine. Eq. (4) eliminates the sub-tour for the tool path. These constraints assign an auxiliary non-repetitive variable to each node for a Hamilton tour [30, 31]. Operational precedence for the tool is guaranteed by Eq. (5). Finally, the decision variables of the proposed model are defined in Eq. (6) and Eq. (7).

#### 4. Proposed algorithm

The SBO is a recently proposed bio-inspired meta-heuristic algorithm introduced by Moosavi and Bardsiri [28] to efficiently estimate software development effort. The SBO is inspired by the attraction of the male bowerbirds to the female bowerbirds for mating by constructing a bower, a structure built from sticks and twigs, and decorating the surrounding area. Based on the behavior of the bowerbirds, the SBO is structured as follows.

Similar to other meta-heuristic algorithms, the SBO starts with a randomly generated population consists of  $NB$  bowers. Let  $\mathbf{X}_i = \{x_{i,1}, x_{i,2}, \dots, x_{i,D}\}$  represents the  $i^{th}$  bower, where  $D$  is the problem size. Each bower is generated by using the following equation

$$x_{i,k} = x_k^L + rand(x_k^U - x_k^L) \quad (8)$$

where  $i = 1, \dots, NB$  and  $k = 1, \dots, D$ . Here  $x_k^L$  and  $x_k^U$  are the lower and upper bounds of parameter  $k$ , respectively. Finally,  $rand$  is a uniformly distributed random number.

After the initialization step, the fitness of each bower is calculated by using equation (9), where  $f(\mathbf{X}_i)$  is the cost function of bower  $i$ . Then the bowers are sorted based on their fitness value in descending order. To prevent the experience of best bowerbird in population, elitism is applied for the population. To do this, the position of the best bower built by birds is identified as elite.

$$fit_i = \begin{cases} \frac{1}{1 + f(\mathbf{X}_i)}, & f(\mathbf{X}_i) \geq 0 \\ \frac{1}{1 + |f(\mathbf{X}_i)|}, & f(\mathbf{X}_i) < 0 \end{cases} \quad (9)$$

In each iteration, SBO starts to generate a new population by calculating the probability of the bowers to identify their attractiveness. The probabilities of the bowers are calculated as shown in Equation (10).

$$Prob_i = \frac{fit_i}{\sum_{n=1}^{NB} fit_n} \quad (10)$$

Following the probability calculation step, new changes at any bower are calculated by using the Equations (11) and (12). In Equation (11),  $\mathbf{X}_{elite}$  is the elite bower of the current population, and  $\mathbf{X}_j$  is the target bower, which is calculated by the roulette wheel selection procedure. Finally,  $\lambda_k$  is the attraction power in the goal bower, which is controlled by greatest step size parameter  $\alpha$ .

$$x_{i,k}^{new} = x_{i,k}^{old} + \lambda_k \left( \left( \frac{x_{j,k} + x_{elite,k}}{2} \right) - x_{i,k}^{old} \right) \quad (11)$$

$$\lambda_k = \frac{\alpha}{1 + Prob_j} \quad (12)$$

At the end of each cycle, a mutation procedure is carried out by applying random changes to the bowers with a certain probability, which is described as  $P\_mutation$  in this study. The random changes on the bower are determined by

$$x_{i,k}^{new} = x_{i,k}^{old} + (\sigma \times N(0,1)) \quad (13)$$

where  $N(0,1)$  is a standard normal distributed random number and  $\sigma$  is a proportion of space width which is calculated by using Equation 14. The  $\sigma$  in Equation 14 is controlled by parameter  $z$ , which is the percent of the difference between the  $x_k^L$  and  $x_k^U$ .

$$\sigma = z \times (x_k^U - x_k^L) \quad (14)$$

At the end of each iteration, the fitness values of newly generated bowers are evaluated. Then, the bowers from the old population and newly generated bowers are combined and re-sorted with respect to their fitness. The new population is formed by removing the last  $NB$  bowers from the sorted population. Finally, the elite is updated if the first bower in the new population is fitter than the existing elite.

According to the procedures of the SBO described above, Algorithm 1 presents the main steps of the algorithm. For more details for the SBO, readers can refer to [28] and access to the source code of the algorithm given by the authors.

Since the original version of the SBOs introduced to optimize global optimization problems, this paper introduces a modified version of the SBO for the TPOP, which is called modified satin bowerbird optimizer (MSBO). The MSBO integrates two main procedures to SBO: Discretization and local search. The discretization procedure converts the continuous solution vector of the algorithm to the discrete solution vector considering the precedence constraints. In the local search procedure, two simple movement

operators are carried out to improve the solution quality of the newly generated solution vectors. The details of both discretization and local search procedures are given in the following sub-sections.

**Algorithm 1.** Main steps of the SBO

1	: Initialization
2	: Fitness evaluation and sorting
3	: Identify the best bower as the elite
4	: <b>Repeat</b>
5	: Probability calculation
6	: <b>For Each</b> Bower
7	: Determination of new changes
8	: Mutation
9	: <b>End For</b>
10	: Evaluation of bowers
11	: Re-sorting and selection
12	: Update elite bower
13	: <b>Until</b> the termination criterion is satisfied

### 4.1. Discretization procedure

One of the critical issue for the continuous meta-heuristic algorithms while solving the combinatorial problems is discretization procedure to represent a solution for the considered problem. Since many problems require discrete search spaces, there exist several techniques to convert continuous solution to discrete solution, which can be classified into three main groups [32]: (i) rounding off generic technique, (ii) priority position techniques, (iii) specific techniques associated with meta-heuristic discretizations. In this study, the *smallest position value* rule, which is one of the priority position techniques introduced by [33], is used in the MSBO to convert a continuous solution vector to discrete solution vector. The *smallest position value* method converts the continuous values to a permutation order of the position values by sorting the positions with respect to ascending order of the position values.

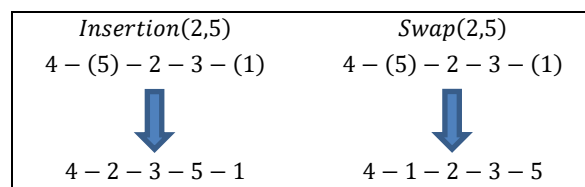
The *smallest position value* method provides a permutation order for a continuous solution vector. However, a feasible solution cannot always be produced for the TPOP by this method because of the precedences of some operations in the CNC machine. Therefore, the *smallest position value* is adapted to the TPOP with the following insertion rule. According to the position values in ascending order, a candidate position with the smallest position value can be inserted into the permutation order if and only if its precedence operations are inserted to the order previously. An example of the modified *smallest position value* rule for the TPOP is shown in Table 2, where the permutation order of the positions with respect to operational precedence is 4 – 5 – 2 – 3 – 1.

**Table 2.** Example of the modified *smallest position value* rule.

Position Index	1	2	3	4	5
Position Value	0.75	0.29	0.95	0.12	0.36
Precedence of Position	(2, 3)	(5)	(4, 2)	(-)	(4)

### 4.2. Local search

After a permutation ordered solution is obtained in the discretization step for each bower, a local search procedure is carried out by consecutively applying two simple operators: *Insertion(x, y)*, and *Swap(x, y)*. In *Insertion(x, y)* operator, a specific position (x) in permutation order is inserted into another specific location (y). In swap operation, locations of two specific positions (x, y) are replaced. Figure 3 shows an illustrative example of both insertion and swap operations. For both the operators, the best improvement strategy is taken into account and each operator is repeated if any improvement is provided at the end of the search.



**Figure 3.** Example of insertion and swap operators.

## 5. Computational results and discussion

Computational studies for the MSBO are formed into two parts. In the first part, the performance of the MSBO is tested by comparing the proposed algorithm with SBO and also two recent meta-heuristic algorithms. In the second part, the MSBO is carried out for the real-life problem of the company. Moreover, managerial insights of the results are discussed in the last sub-section.

All experiments are performed on a workstation equipped with a 3.4GHz Xeon E5-2643v3 and 64 GB RAM. However, a single thread is used for the algorithm runs. According to preliminary experiments, the parameter values of the MSBO are identified as follows:  $\alpha = 0.25$ ,  $z = 0.50$  and,  $P_{mutation} = 0.05$ .

### 5.1. Performance analyses of the MSBO

Since the PCTSP is also known as the sequential ordering problem (SOP), a well-known SOP dataset from the TSPLIB repository is used in order to test the performance of the proposed MSBO. The SOP dataset includes 41 different sized instances, where the number of nodes to be visited varies between 9-380. Each instance consists of a number of nodes and distances for each pair of nodes. The aim of the problem is to find a minimum Hamiltonian path from the first node to the last node with minimum length by considering the precedence constraints. Regarding the maximum TPOP size of the company, 27 of 41 instances from the dataset (up to 100 nodes) are used for the performance analyses.

In this subsection, the MBSO is first compared with the SBO and two variants of the MBSO. To point out the effect of local search procedure on the solution quality,

the first variant of the MBSO considers only insertion operation in the local search part, which is named MSBO<sup>I</sup>. Similarly, the second variant considers only swap operation, which is named MSBO<sup>S</sup>. The algorithms are repeated 30 times for each instance, and the results of each instance are identified via the average result of 30 runs (mean), and the standard deviation of the results (std). Table 3 presents the results of the SBO, MSBO<sup>I</sup>, MSBO<sup>S</sup>, and MSBO. Furthermore Table 3 shows a statistical comparison of the algorithms based on paired *t*-test with a significance

level of 0.05. For each pair of algorithms, the successful one is shown if there is a significant difference between the algorithm results. In this perspective, it should be clearly seen from Table 3 that the MSBO<sup>I</sup> and MSBO show better performance for all instances with respect to the SBO and MSBO<sup>S</sup>. On the other hand, the MSBO outperforms the MSBO<sup>I</sup> for three instances. However, the average results of the MBSO are mostly better than the average results of MSBO<sup>I</sup>. Thus, it should be concluded from Table 3 that the proposed MBSO outperforms the SBO, MSBO<sup>I</sup>, and MSBO<sup>S</sup>.

**Table 3.** Comparisons of SBO and MSBO variants

Instance	SBO (A)		MSBO <sup>I</sup> (B)		MSBO <sup>S</sup> (C)		MSBO (D)		Statistical Comparisons					
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	A-B	A-C	A-D	B-C	B-D	C-D
br17.10	58.1	1.6	55.0	0.0	55.0	0.0	55.0	0.0	B	C	D	-	-	-
br17.12	58.9	3.8	55.0	0.0	55.0	0.0	55.0	0.0	B	C	D	-	-	-
ESC07	2125.0	0.0	2125.0	0.0	2125.0	0.0	2125.0	0.0	-	-	-	-	-	-
ESC12	1722.6	49.2	1675.0	0.0	1675.0	0.0	1675.0	0.0	B	C	D	-	-	-
ESC25	3286.9	423.6	1723.9	38.9	1986.2	138.3	1711.6	28.2	B	C	D	B	-	D
ESC47	6626.3	815.3	2264.8	76.1	3104.9	143.8	2197.7	84.8	B	C	D	B	D	D
ESC63	123.8	16.5	62.0	0.0	64.2	0.7	62.0	0.0	B	C	D	B	-	D
ESC78	207887.0	497.9	18230.0	0.0	18347.8	79.9	18230.0	0.0	B	C	D	B	-	D
ft53.1	12527.4	1011.2	7621.1	66.3	8350.7	196.8	7619.1	71.5	B	C	D	B	-	D
ft53.2	12984.8	1129.9	8083.3	25.7	8843.5	219.5	8084.5	32.1	B	C	D	B	-	D
ft53.3	14796.0	420.6	10284.8	36.8	10960.9	145.8	10273.1	23.3	B	C	D	B	-	D
ft53.4	16700.8	526.4	14425.0	0.0	14643.3	112.7	14425.0	0.0	B	C	D	B	-	D
ft70.1	54076.0	1692.7	40344.1	190.90	42326.9	326.6	40386.9	229.3	B	C	D	B	-	D
ft70.2	54550.0	2196.0	41528.3	314.1	43778.0	317.1	41486.3	406.9	B	C	D	B	-	D
ft70.3	55101.6	1653.8	42889.6	194.7	46078.3	501.6	42822.4	201.7	B	C	D	B	-	D
ft70.4	58985.0	709.5	53606.1	64.7	54790.2	349.1	53616.8	66.3	B	C	D	B	-	D
p43.1	28794.5	238.7	28144.5	13.7	28163.7	21.5	28140.0	0.0	B	C	D	B	-	D
p43.2	40474.5	13476.9	28482.2	3.1	28487.5	6.5	28480.7	1.7	B	C	D	B	D	D
p43.3	47183.0	12728.0	28838.0	5.4	28851.3	18.3	28836.7	3.8	B	C	D	B	-	D
p43.4	87072.7	9200.6	83005.0	0.0	83040.7	42.5	83005.0	0.0	B	C	D	B	-	D
prob.42	623.5	55.3	302.8	17.4	357.1	15.0	259.7	5.4	B	C	D	B	D	D
rbg048a	431.0	21.3	351.0	0.0	353.2	1.7	351.0	0.0	B	C	D	B	-	D
rbg050c	533.2	20.1	467.1	0.3	472.5	3.2	467.1	0.3	B	C	D	B	-	D
ry48p.1	23246.1	2575.0	15861.4	100.3	16271.4	246.4	15829.7	75.4	B	C	D	B	-	D
ry48p.2	24065.5	1590.0	16686.4	44.5	17079.9	215.2	16705.4	51.3	B	C	D	B	-	D
ry48p.3	27496.1	1384.1	19894.0	0.0	20618.8	305.2	19894.0	0.0	B	C	D	B	-	D
ry48p.4	34109.2	899.2	31446.1	0.4	31605.0	136.3	31446.0	0.0	B	C	D	B	-	D

Another performance analysis for the MBSO is made by comparing the proposed algorithm with two recent meta-heuristic algorithms proposed for the PCTSP and SOP, which are Adaptive Evolutionary Algorithm (AEA) introduced by Sung and Jeong [34] and an improved Ant Colony System (ACS) introduced by Skinderowics [35]. Table 4 shows the available results of the AEA and ACS and comparisons between MBSO and other two algorithms, where “best” and “time” represent the best results and average computational time of the runs for a specified instance. To identify the better results for the comparisons, the smallest values in a row are written in bold. Table 4 additionally presents the best-known solutions of the instances, which are also known as optimal solution excepting “prob.100”, “ry48p.2”, and “ry48p.3”. Here, it should

be noted that the two results of AEA given by the authors are less than the optimum solutions. Therefore, it is not notable to compare the MBSO with AEA for these instances. For the other instances, it can be seen from Table 4 that proposed MBSO outperforms the AEA for each instance. With regards to the ACS solutions, better results are obtained for most of the instances by the MBSO.

For the computational times, it is reported by the Sung and Jeong [34] that the computational time of the AEA varies between 0.01-290.93 seconds for the considered problems. For the ACS, the authors report that the results are obtained with 60 seconds time limitation [35]. Regarding both the algorithm times, the CPU times of the MBSO, which are shown in Table 4, are acceptable for the real-life applications.



**Table 4.** Comparison of MSBO with AEA and ACS

Instance	Best-Known	AEA		ACS		MSBO			
		Best	Mean	Mean	Std	Best	Mean	Time (s)	Std
br17.10	55	55	55.9	NA	NA	55	<b>55.0</b>	1.82	0.0
br17.12	55	55	55.0	NA	NA	55	55.0	1.82	0.0
ESC07	2125	2125	2125.0	NA	NA	2125	2125.0	0.51	0.0
ESC12	1675	1675	1720.6	NA	NA	1675	<b>1675.0</b>	1.05	0.0
ESC25	1681	2354	3378.4	NA	NA	<b>1681</b>	<b>1711.6</b>	4.44	28.2
ESC47	1288	4160	5648.3	NA	NA	<b>2026</b>	<b>2197.7</b>	21.27	84.8
ESC63	62	73	80.9	NA	NA	<b>62</b>	<b>62.0</b>	46.38	0.0
ESC78	18230	NA	NA	NA	NA	18230	18230.0	89.23	0.0
ft53.1	7531	10619	11537.4	7702	<b>46.1</b>	<b>7568</b>	<b>7619.1</b>	27.98	71.5
ft53.2	8026	11000	12092.1	8348	156.6	<b>8026</b>	<b>8084.5</b>	29.02	<b>32.1</b>
ft53.3	10262	13231	13797.0	11271	605.5	<b>10262</b>	<b>10273.1</b>	29.45	<b>23.3</b>
ft53.4	14425	15579	16263.3	14639	101.1	<b>14425</b>	<b>14425.0</b>	35.96	<b>0.0</b>
ft70.1	39313	46390	48696.2	<b>40054</b>	<b>223.6</b>	<b>39930</b>	40386.9	79.03	229.3
ft70.2	40419	46485	49162.1	41629	409	<b>40587</b>	<b>41486.3</b>	73.40	<b>406.9</b>
ft70.3	42535	NA	NA	43946	436.6	42535	<b>42822.4</b>	65.90	<b>201.7</b>
ft70.4	53530	NA	NA	55305	362.9	53530	<b>53616.8</b>	68.07	<b>66.3</b>
p43.1	28140	28830	29107.0	NA	NA	<b>28140</b>	<b>28140.0</b>	17.30	0.0
p43.2	28480	28896	47979.5	NA	NA	<b>28480</b>	<b>28480.7</b>	17.69	1.7
p43.3	28835	28680*	29362.5	NA	NA	<b>28835</b>	<b>28836.7</b>	17.20	3.8
p43.4	83005	82960*	83393.5	NA	NA	<b>83005</b>	<b>83005.0</b>	18.07	0.0
prob.42	243	443	530.7	NA	NA	<b>248</b>	<b>259.7</b>	15.67	5.4
rbg048a	351	376	412.6	NA	NA	<b>351</b>	<b>351.0</b>	25.46	0.0
rbg050c	467	505	528.7	NA	NA	<b>467</b>	<b>467.1</b>	28.26	0.3
ry48p.1	15805	18650	21732.2	NA	NA	<b>15805</b>	<b>15829.7</b>	24.95	75.4
ry48p.2	16666	18499	22210.8	NA	NA	<b>16666</b>	<b>16705.4</b>	24.45	51.3
ry48p.3	19894	22480	25029.2	NA	NA	<b>19894</b>	<b>19894.0</b>	23.60	0.0
ry48p.4	31446	33961	34944.7	NA	NA	<b>31446</b>	<b>31446.0</b>	24.80	0.0

NA: Not available

\*Solution given in the related study is less than the optimal solution

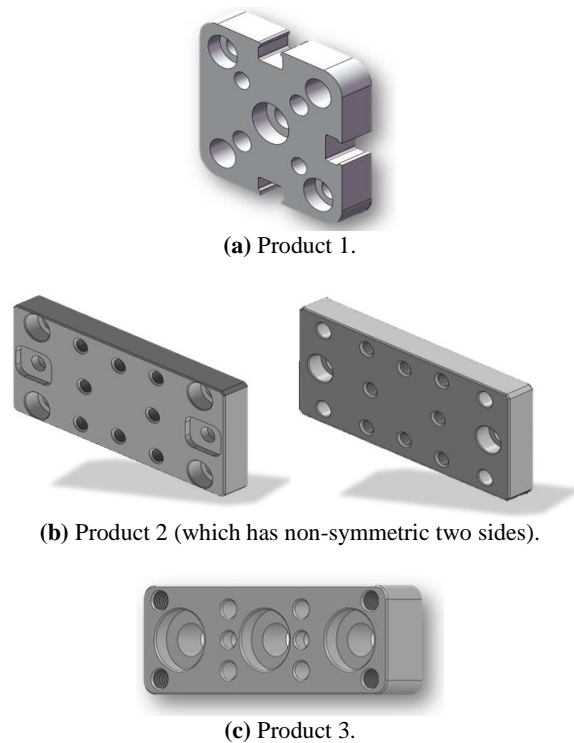
### 5.2. Application of the MSBO to real-life TPOP problem

In the second part of the computational studies, the proposed MSBO is carried out for the real-life TPOP problem of the company. In this context, three different products produced over than thousands in a year by the company are taken into account. Solid models of the products 1, 2, and 3 are presented in Figure 4a, 4b, and 4c respectively. Table 5 shows the number of milling operations and tools required to produce these products in CNC milling machines.

**Table 5.** Operational requirements of the products.

Product	# of milling operations	# of tools
Product 1	22	9
Product 2	58	6
Product 3	37	11

To compare the performance of the MSBO, first, the mathematical model of the problem is solved for each product by using Gurobi 7.5.1 solver on MPL 5.0 software with 10 hours time limitation. The input data of the models are determined by using SolidCAM software, which is also used by the company for their production process.



**Figure 4.** Solid models of the products considered for computational studies.

Following the Gurobi computations, the MSBO is performed for the TPOP problems of the company. Results of the Gurobi solver and MBSO are shown in Table 6. Additionally, operational times of the

productions carried out by the company are presented in Table 6, and percentage improvements provided by the Gurobi and MBSO are pointed out in the columns named "Imp%".

**Table 6.** Gurobi and MBSO results for three products.

Product	Operational time of company	Gurobi Solution			MBSO				
		Result (Upper Bound)	Imp%	CPU Time (s)	Best	Imp%	Mean	Time (s)	Std
Product 1	108.92	108.48	0.40	108.52	108.48	0.40	108.48	6.34	0.0
Product 2	78.78	77.03	2.22	36 000.00	75.59	4.05	75.61	56.07	~0.0
Product 3	110.71	109.69	0.92	36 000.00	109.69	0.92	109.7	35.03	~0.0

It should be concluded from Table 6 that an optimum solution is found by Gurobi for only product 1. For the other two problems, the computations are terminated at the end of the time limit and, upper bounds of the solution are taken into account for the comparisons. According to the Gurobi results, an improvement with up to 2.22% is provided for the drilling operations of the company. On the other hand, MBSO could find the optimum solution for product 1 and better result than the upper bound of the Gurobi solutions for product 2 (with 4.05% improvement). Regarding the average computational times of the MBSO, it should be noted that better results can be obtained by the proposed algorithm in shorter CPU times with respect to the Gurobi solver. As a result of the computational studies for these parts, it should be noted that considerable variable cost savings are provided for the company since these products are produced over than thousands in a year.

### 5.3. Managerial insights

The computational results of the MBSO show the efficiency and the robustness of the proposed solution methodology. Similar to most of the meta-heuristic algorithm, the tuned parameter set directly affects the performance of the algorithm. Therefore, a preliminary study is made in order to find the best parameter values for the MBSO. On the other hand, computational studies on the SOP dataset show that the local search procedure significantly affects the performance of the MBSO. Particularly, MBSO with the insertion move, and the combination of insertion-swap move show superior performance. Although, these procedures increase the computational times of the algorithm, with reasonable time limits better results can be obtained by using both local searches.

Another interesting observation of the results is the stability of the MBSO solutions. It can be clearly seen from the computations that the standard deviations of most of the results are 0. Comparing with the ACS results, a less standard deviation is observed by MBSO for 6 of 8 instances.

When the results of the case-stud are analyzed, an improvement is provided by the MBSO for each case

even though the number of problems is limited with three product types. For these experiments, the most significant saving is provided on the largest problem. Furthermore, the MBSO outperforms the Gurobi regarding both the results and CPU times.

As a consequence, the MBSO can be efficiently used in the optimization of CNC operations at the tactical and operational decision-making level. In particular, the proposed MBSO is capable of finding the optimal or near-optimal solution for the real-life TPOPs for the CNC machines. Since an improvement on a process in mass production system provides considerable cost savings for the companies, the proposed study has a potential to take forward the researches on this field. In practice, considering that such production parts consist of numerous CNC operations, operational times and machining costs can be significantly reduced by the MBSO.

### 6. Conclusion

This paper addresses the TPOP in CNC milling machines to improve the internal operational efficiency of a company. To find the best tool path in the CNC machines, the mathematical model of the problem is formulated based on the assumptions of PCTSP. With this assumption, tool movement between two operational points is allowed if the successor operation is not the precedence of the predecessor operations. To efficiently solve the considered problem, a newly developed SBO is taken into account and modified its original version with discretization and local search procedures. The computational studies for the proposed MSBO are formed into two parts. In the first part, the MSBO is carried out on a well-known benchmark problem set introduced for the PCTSP, and compared with two recent meta-heuristic algorithms. Results of these computations show that the proposed MSBO is capable of finding efficient solution for the TPOP by finding better results with respect to the other two algorithms. Then, the proposed MSBO is applied for real-life drilling operations of three products. For this case, with up to 4.05% improvement on the operational times is achieved by the MSBO.

As a future work, this study can be extended by



considering the following issues. In this study, MBSO is carried out for a set of PCTSP instances, in which the largest problem consists of 100 nodes. Thereupon, the proposed MBSO can be carried out for the larger PCTSPs, and its efficiency can be analyzed. On the other hand, the MBSO can be improved with additional search strategies since only insertion and swap operations are used in the MBSO. Finally, other bio-inspired algorithms can be modified for the PCTSP, and detailed comparative analysis can be reported.

### Acknowledgments

The authors thank to company consultants Mr. Mustafa Karataş who have shared their experiences and knowledge, and also to the employees of HID-TEK for their support.

### References

- [1] Bohez, E., Makhanov, S.S., Sonthipapoon, K. (2000). Adaptive nonlinear tool path optimization for five-axis machining. *International Journal of Production Research*, 38(17), 4329-4343.
- [2] Makhanov, S.S., Batanov, D., Bohez, E., Sonthipapoon, K., Anotai, W. (2002). On the tool-path optimization of a milling robot. *Computers and Industrial Engineering*, 43(3), 455-472.
- [3] Chandrasekaran, M., Muralidhar, M., Murali Krishna, C., Dixit, U.S. (2010). Application of soft computing techniques in machining performance prediction and optimization: A literature review. *The International Journal of Advanced Manufacturing Technology*, 46(5-8), 445-464.
- [4] D'Souza, K., Castelino, R., Wright, P.K. (2003). Tool-path optimization for minimizing airtime during machining. *Journal of Manufacturing Systems*, 22, 173-180.
- [5] Park, K.S., Kim, S.H. (1998). Artificial intelligence approaches to determination of CNC machining parameters in manufacturing: A review. *Artificial Intelligence in Engineering*, 12(1-2), 127-134.
- [6] Lasemi, A., Xue, D., Gu, P. (2010). Recent development in CNC machining of freeform surfaces: A state-of-the-art review. *Computer-Aided Design*, 42(7), 641-654.
- [7] Lazoglu, I., Manav, C., Murtezaoglu, Y. (2009). Tool path optimization for free form surface machining. *CIRP Annals*, 58(1), 101-104.
- [8] Oysu, C., Bingul, Z. (2009). Application of heuristic and hybrid-GASA algorithms to tool-path optimization problem for minimizing airtime during machining. *Engineering Applications of Artificial Intelligence*, 22(3), 389-396.
- [9] Dewil, R., Küçükoğlu, İ., Luteyn, C., Cattrysse, D. (2018). A critical review of multi-hole drilling path optimization. *Achieves of Computational Methods in Engineering*, In Press, 1-11.
- [10] Abidin, N.W.Z., Ab Rashid, M.F.F., Mohamed, N.M.Z.N. (2017). A review of multi-holes drilling path optimization using soft computing approaches. *Achieves of Computational Methods in Engineering*, In Press, 1-12.
- [11] Narooei, K.D., Ramli, R., Nizam, M., Rahman, A., Ibrahım, F., Qudeiri, J.A. (2014). Tool routing path optimization for multi-hole drilling based on ant colony optimization. *World Applied Sciences Journal*, 32(9), 1894-1898.
- [12] Linn, R.J., Liu, J., Kowe, P.S.H (1999). Efficient heuristics for drilling route optimization in printed circuit board manufacturing. *Journal of Electronics Manufacturing*, 8(2), 127-138.
- [13] Kolahan, F., Liang, M. (2000). Optimization of hole-making operations: A tabu search approach. *International Journal of Machine Tools and Manufacture*, 40(12), 1735-1753.
- [14] Abbas, A.T., Aly, M.F., Hamza, K. (2011). Optimum drilling path planning for a rectangular matrix of holes using ant colony optimization. *International Journal of Production Research*, 49(19), 5877-5891.
- [15] Montiel-Ross, O., Medina-Rodríguez, N., Sepúlveda, R., Melin, P. (2012). Methodology to optimize manufacturing time for a CNC using a high performance implementation of ACO. *International Journal of Advanced Robotic Systems*, 9, 1-10.
- [16] Zhu, G.-Y., Zhang, W.-B. (2008). Drilling path optimization by the particle swarm optimization algorithm with global convergence characteristics. *International Journal of Production Research*, 46(8), 2299-2311.
- [17] 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.
- [18] Dalavi, A.M., Pawar, P.J., Singh T.P. (2016). Optimal sequence of hole-making operations using particle swarm optimisation and shuffled frog leaping algorithm. *Engineering Review*, 36(2), 187-196.
- [19] Kumar, A., Pachauri, P.P. (2012). Optimization drilling sequence by genetic algorithm. *International Journal of Scientific Research Publications*, 2(9), 1-7.
- [20] Nabeel, P., Abid, K., Abdulrazzaq, H.F. (2014). Tool path optimization of drilling sequence in CNC machine using genetic algorithm. *Innovation System Design and Engineering*, 5(1), 15-26.
- [21] Khalkar, S., Yadav, D., Singh, A. (2015). Optimization of hole making operations for sequence precedence constraint. *International Journal of Innovative and Emerging Research in Engineering*, 2(7), 26-31.
- [22] Pezer, D. (2016). Efficiency of tool path optimization using genetic algorithm in relation to the optimization achieved with the CAM software.

- Procedia Engineering*, 149, 374-379.
- [23] Raja, C.K.B., Saravanan, M. (2017). Genetic algorithm for TSP in optimizing CNC tool path. *International Journal of Engineering Technology, Management and Applied Sciences*, 5(2), 139-146.
- [24] 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.
- [25] Kanagaraj, G., Ponnambalam, S.G., Loo, C.K. (2015). Charged system search algorithm for robotic drill path optimization. *Proceedings of the 2015 International Conference on Advanced Mechatronic Systems, Beijing, China*, 125-130.
- [26] Lim, W.C.E., Kanagaraj, G., Ponnambalam, S.G. (2016). A hybrid cuckoo search-genetic algorithm for hole-making sequence optimization, *Journal of Intelligent Manufacturing*, 27, 417-429.
- [27] Zhang, W.-B., Zhu, G.-Y. (2018). Drilling path optimization by optimal foraging algorithm. *IEEE Transactions on Industrial Informatics*, 14(7), 2847-2856.
- [28] Moosavi, S.H.S., Bardsiri, V.K. (2017). Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. *Engineering Applications of Artificial Intelligence*, 60, 1-15.
- [29] Kubo, M., Kasugai, H. (1991). The precedence constrained traveling salesman problem. *Journal of the Operations Research*, 34(2), 152-172.
- [30] Miller, C.E., Tucker, A.W., Zemlin, R.A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4), 326-329.
- [31] Bellmore, M., Nemhauser, G.L. (1968). The traveling salesman problem: A Survey. *Operations Research*, 16(3), 538-558.
- [32] Crawford, B., Soto, R., Astorga, G., Garcia, J., Castro, C., Paredes, F. (2017). Putting continuous metaheuristics to work in binary search spaces. *Hindawi-Complexity*, 2017, 1-17.
- [33] Tasgetiren, M.F., Liang, Y.-C., Sevkli, M., Gencyilmaz, G. (2007). A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, 177, 1930-1947.
- [34] Sung, J., Jeong, B. (2014). An adaptive evolutionary algorithm for travelling salesman problem with precedence constraints. *Hindawi-The Scientific World Journal*, 2014, 1-11.
- [35] Skinderowicz, E. (2017). An improved ant colony system for the sequential ordering problem. *Computers and Operations Research*, 86, 1-17.
- Ilker Kucukoglu** is a Research Assistant in the Industrial Engineering Department of Bursa Uludag University, Turkey. He received his BS from Gazi University (2008), his MS from Bursa Uludag University (2010) and his Ph.D. from Bursa Uludag University (2015) – all in Department of Industrial Engineering. His research interests include supply chain management, vehicle routing, green logistics, heuristic methods, optimization, and mathematical modelling.
- Tulin Gunduz** is currently an Associate Professor in the Department of Industrial Engineering at the Bursa Uludag University. She completed her undergraduate, master's and Ph.D. degrees in 1995, 1998 and 2005, respectively, at Bursa Uludag University, Department of Mechanical Engineering. Her Ph.D. was on Ergonomics. Her research interests are ergonomics, driver comfort, anthropometry, cognitive science and Industry 4.0
- Fatma Balkancioglu** is an Industrial Engineer and received her BS degree from Bursa Uludag University (2018).
- Emine Chousein Topal** is a BS student in Bursa Uludag University Industrial Engineering department.
- Oznur Sayim** is a BS student in Bursa Uludag University Industrial Engineering department.

