Journal of Optimization in Industrial Engineering
Vol.13, Issue 2, Summer & Autumn 2020, 1-15
DOI:10.22094/JOIE.2020.1869865.1660

Research Article

Machine Scheduling for Multitask Machining

Saleh Yavari [a], Ahmed Azab [a,*], Mohammed F. Baki [b], Mikel Alcelay [a], Justin Britt[a]

[a] *Production & Operations Management Research Lab, Faculty of Engineering, University of Windsor, Windsor, Canada*
[b] *Odette School of Business, University of Windsor, Windsor, Canada*
Received 28 May 2019; Revised 09 March 2020; Accepted12 March 2020

Abstract

Multitasking is an important part of today's manufacturing plants. Multitask machine tools are capable of processing multiple operations at the same time by applying a different set of part and tool holding devices. Mill-turns are multitask machines with the ability to perform a variety of operations with considerable accuracy and agility. One critical factor in simultaneous machining is to create a schedule for different operations to be completed in minimum make-span. A Mixed Integer Linear Programming (MILP) model is developed to address the machine scheduling problem. The adopted assumptions are more realistic when compared with the previous models. The model allows for processing multiple operations simultaneously on a single part; parts are being processed on the same setup and multiple turrets can process a single operation of a single job simultaneously performing multiple depths of cut. A Simulated Annealing (SA) algorithm with a novel initial solution and assignment approach is developed to solve large instances of the problem.

Keywords: Parallel machining; Multitasking; Mill-turn; Mixed integer linear programming; Scheduling; Simulated Annealing.

1. Introduction

Although the tendency to benefit from multitasking and parallel machining is increasing considerably in industry, there has not been enough attention in the literature to the topic. Mill-turns are machine tools that are similar to lathes with respect to structure, kinematics, and the usage of spindles as work-piece holding devices, but are different in that they also carry live milling tools in their multiple turrets (Lee and Chiou, 1999). These features make mill-turn machines agile and allow them to machine different work-pieces simultaneously. In addition, mill-turns are superior to traditional lathes and vertical mills in terms of speed and accuracy. Multitasking enables turning, milling, facing, boring, drilling, and tapping operations to be done on a single setup with minimum processing times (Crichigno Filho, 2012; She and Hung, 2008; Yip-Hoi and Dutta, 1993). A mill-turn machine can process a part which might need three to four different machine tools to be turned into a finished product. Moreover, multitasking usually allows machining the most complex parts in only one setup, which reduces part handling and costs related to extra fixtures and tooling.

Two concepts are key to the understanding of multitask machining: the Machining Unit (MU) and the Part Machining Location (PML) (Levin and Dutta, 1992). In mill-turns, machining tools are mounted in turrets (Battaïa *et al*., 2014). Turrets can mount different cutting tools for both milling and turning operations. Therefore, if a work-piece needs both turning and milling operations, both can be done on only one setup on a single machine tool (*i.e.*, there is no need to use separate milling and turning machine tools). A spindle or a chuck are common machining locations on a multitask machine (Levin and Dutta, 1996). Commonly used configurations of mill-turns usually have two spindles applied: a main spindle and a sub spindle (Chiu *et al.*, 1999). These spindles have access to each other so that if a specific part needs a certain type of machining process, it could be easily moved from one spindle to another (Battaïa *et al*., 2013). Figure 1 shows a work-piece held by the spindle and processed by a cutting tool mounted on a turret.

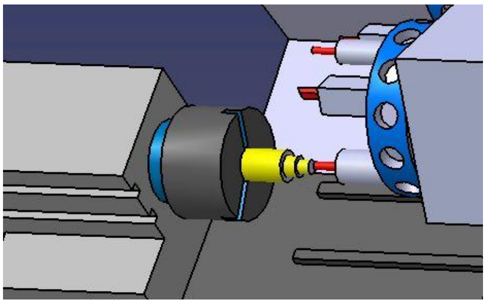

Fig 1. Work-piece held by a spindle and processed by a cutting tool mounted on a turret

There are various configurations of mill-turns. Most of the configurations have two spindles. However, in terms of turrets, they could be two or more, each of which could contain several cutting tools. The focus of this research is on the scheduling of operations on mill-turns with dual spindles and dual turrets. Figure 2 shows a common configuration of mill-turns with two spindles as work holding devices and two turrets as machining units.

*Corresponding author Email address: azab@uwindsor.ca

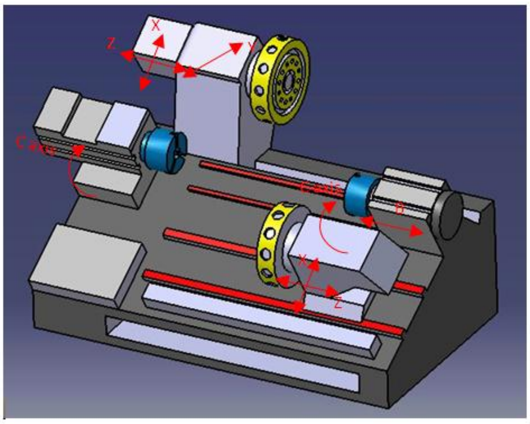

Fig. 2. A mill-turn with dual spindles and dual turrets

Despite the growing demand in the industry to apply multitasking and parallel machining, research done on the scheduling problem is quite limited. There has not been much literature on the problem of scheduling operations on mill-turns in comparison with traditional parallel machine scheduling problems (Fanjul-Peyro *et al.*, 2017; Low and Wu, 2016; Rajkanth *et al.*, 2017).

Miller (1989) and Miska (1990) introduce a new class of machines that can machine parts in less time and with fewer setups. Levin and Dutta (1992, 1996) propose process planning and sequencing of parallel machine tools. Yip-Hoi and Dutta (1993, 1995) discuss important components of multitasking, which are key to process planning. Azab and Naderi (2014) provide a variable neighborhood search metaheuristic for the problem of scheduling operations for multitask machining. They do not model the problem mathematically to define the capabilities and limitations of parallel machines. Furthermore, the capacity of turrets to simultaneously process a specific operation of a single job on a single machine, mode of operations, and concepts of single setup (done-in-one) were not considered in their study. Yip-Hoi and Dutta (1996) suggest a genetic algorithm to achieve an optimum completion time for all operations. However, no mathematical model is proposed to further extend the problem for larger and more complicated instances. Norman and Bean (2000) present a small size model for the problem of scheduling mill-turns and consider a shop where one work-piece is machined and scheduled. The capacity of machines in terms of applying multiple turrets for a single operation is not considered. Chiu *et al.* (1999) solve the problem of sequencing and scheduling operations on parallel machines for a shop with one part and one machine. In their study, operations are scheduled irrespective of the rotation of spindles and cutting tools and its effect on the final sequence. A Mixed Integer Linear Programming (MILP) model is developed by Naderi and Azab (2015). They formulate a mathematical model that does not consider the mode of operations and how it could affect the number of operations processed simultaneously on a single spindle. Also, the capacity of turrets to simultaneously process a specific operation of a single job on a single machine is not considered.

This work tackles the problem of production scheduling of mill-turn machines considering limitations and abilities that have not been addressed before.

The problem of scheduling different operations on mill-turns is addressed by developing a MILP model. The model optimizes the total make-span and creates a feasible solution for scheduling operations on spindles by applying different turrets. The model also assigns a feasible turret for processing each part. Solving the model generates an optimal solution to schedule different sets of operations on available machines. The model creates a sequence of operations for each job taken into consideration all constraints and capabilities regarding the used parallel machine tools. Due to the NP-complete nature of the problem (Naderi and Azab, 2015), a simulated annealing algorithm (SA) is developed to solve large instances of the problem at hand. An encoding method along with different search operators and mechanisms are proposed to elaborate the mechanism of the algorithm.

The remaining of this paper is organized as follows. Section two fully describes the problem. Section three contains the mathematical model along with a demonstrative example. Section four focuses on developing a metaheuristic algorithm. Test cases to evaluate the algorithm and model are provided in section five. Conclusions are presented in section six.

2. Problem Definition

There are many features that should be considered when modeling mill-turns. There are aspects of mill-turns that distinguish them from other machine tools. Features like pinch turning, operation mode and concepts of single setup will be explained in detail to further explore the physics of these machine tools and identify possible areas for improvements. Some of these features are basic concepts related to the mechanism of mill-turns and constitute the overlap in previously proposed models in the literature. These concepts include precedence relationships, capacities of spindles and turrets, and limitations of spindles and turrets for holding and processing parts.

The scheduling problem consists of n jobs and m machines. Each job has a different set of operations and precedence relations. Each operation could have one or two sets of different predecessors and successors. Mill-turn machines, which are considered for scheduling, have two work holding devices (spindles) and two machining units (turrets). Spindles and turrets have limitations associated with the loading and processing of parts. Each turret can process one job and operation at any time, and spindles can hold at most one job (part) at a time. One work-piece can be loaded on one spindle at a time. Turrets have some limitations in terms of cutting tools mounted on them and other restrictions regarding live tooling and cutting speed. Therefore, each turret is capable of machining a different set of operations. Each of these turrets can have access to both spindles (*i.e.*, they could work separately or together on the same spindle). Spindles also have some limitations in terms of holding parts. Hence, all spindles are not allowed to hold all jobs. Operations processed on a mill-turn machine are affected by the rotation of the work-piece and the cutting tools mounted on the turrets.

There are two major types of operations: ones which need the rotation of the work-piece, such as turning and facing, and those which require the rotation of cutting tools, such as milling and drilling. Two operations of the same job and the same mode (rotation of either work-piece or cutting tool) can be processed simultaneously on a single spindle. In other words, it is necessary for two operations having the same mode to be machined simultaneously by two different turrets on a single spindle. This feature is known as the operation mode in the literature. The mode of operations is in direct relation to the rotation of spindles. Turning operations need the rotation of the spindle and the workpiece held by it, whereas milling operations need the rotation of the cutting tool and the spindle could be stationary. Therefore, these two operations cannot be processed simultaneously on a single part. However, in the world of multitask machining, turn-milling is a different type of processing in which both the work-piece mounted in the spindle and the live milling tool rotates at the same time (Choudhury and Mangrulkar, 2000). The reason to apply turn-milling instead of turning operations is that turn-milling provides less contact time between the cutting tool and the part; therefore, there will be less heat produced during the machining process and the life of the cutting tool will increase (Savas and Ozay, 2008).

Turrets are not only capable of processing different operations of a single job, but they also have the ability to work simultaneously on the same operation and hence accelerate the process of machining the operation (Calleja *et al.*, 2015; Lu *et al.*, 2011). This feature is called pinch turning. The procedure is such that one cutting tool starts removing the material from the outside diameter of the part and the other cutting tool is idle at the beginning. Next, after a very short hesitation, the other cutting tool starts the machining procedure from the opposite side of the part. This halves multiple movements of turrets and cutting tools and thus reduces processing times considerably. Machining the product from raw material to a finished part on only one setup is another key feature of mill-turn machines. This feature removes the movements of parts from one machine to another and has a major effect on reducing costs related to part handling and incurred additional setup costs. It is assumed that machining procedures of different operations are not interrupted by any means and that there is no preemption. The parts considered for machining are assumed to be finished on a single spindle and the application of a sub-spindle is not required. Mill-turns can load bar stock and raw material automatically through a bar feeder. Therefore, it is assumed that no setup times are required.

3. Methodology

In this section, the MILP model and a demonstrative example are presented.

3.1 Mathematical model

The outcome of the model is a feasible sequence and schedule for each operation of each job. A specific turret and spindle are required to be assigned to each operation and an optimum start time of each operation based on precedence relations needs to be determined. Moreover, the model is flexible enough to generate plans for a different number of jobs and operations with multiple machines used for processing them. The model seeks the best assignments to optimize the make-span subject to several constraints. The assumptions, parameters, decision variables, objective function, and constraints used to develop the mathematical model are as below.

Assumptions:

- Each operation is assigned to one machine.
- On a machine tool, an operation is assigned to one spindle.
- Spindle engagement determines the number of active turrets (N) engaged in performing each operation, where $N \leq 2$.
- If a spindle is loading two operations simultaneously, then those operations should be of the same type.
- Each spindle can hold one job at a time.
- Each job is loaded on one spindle at a time.
- Each turret can process one operation of a job at a time.
- Precedence relations among operations of each job must be satisfied.
- No delay or pause is allowed between consecutive scheduled operations.
- If a job is assigned to a spindle as part of a setup on a machine tool, its operations will also be assigned to the same spindle of the same setup on the same machine tool. Parts are assumed to be processed on a single setup where only one spindle used and no movement of parts between machines is allowed.
- The application of a sub-spindle or a tailstock is not required.

Parameters:

m	total available machines	*i=1, 2…m*
n	total jobs to be scheduled	*j=1, 2…n*
e	spindle turret engagement	*e=1, 2*
ki	count of spindles of machine *i*	*k=1, 2*
hi	count of turrets of machine *i*	*h=1, 2*
lj	count of operations of job *j*	*l=1, 2… lj*
S ji	spindles, capable of loading job *j* on machine *i*	
$T_{j,i}$	turrets, capable of processing job *j* on machine *i*	
P_{jli}	machining time of operation *l* of job *j* on machine *i*	
N_{jl}	number of turrets used for processing each operation at a time	
$W_{jll'}$	binary parameter taking the value of 1 if operations *l* and *l'* of job *j* are different types of operation and 0 otherwise	

Decision variables:

The following continuous and binary decision variables are defined.

Continuous decision variable

St_{jl} Start time of operation *l* of job *j*

Binary decision variables

B_{ji} 1 if job *j* is processed on machine i, *0* otherwise

B'_{jli} 1 if operation *l* of job *j* is processed on machine i, 0 otherwise

B''_{jlike} 1 if operation *l* of job *j* is processed on spindle turret engagement *e* of spindle *k* of machine i, 0 otherwise

B'''_{jlih} 1 if operation *l* of job *j* is processed on turret *h* of machine$i$, 0 otherwise

B''''_{jik} 1 if job *j* is processed on spindle *k* of machine i, 0 otherwise

$Z_{jljl'}$ 1 if operation *l* of job *j* is processed after operation *l'* of job *j*, 0 otherwise

$Z_{jlj'l'}$ 1 if operation *l* of job *j* is processed after operation *l'* of job *j'*, 0 otherwise

The objective function for the MILP model is to minimize the total completion times of all operations on all machines:

Min C_{max}

where C_{max} is the maximum completion time. The model considers all available machines and machining times of different operations and seeks the best sequence of operations on each machine to minimize the make-span.

Constraints (1) and (2) are provided to ensure that each operation is processed by one spindle-turret engagement, one spindle, and one machine. The parameter *e* defines the capacity of the spindle in terms of loading different operations. Assuming both of the available turrets work on a single operation, spindles at any time have the capacity to load at most two operations.

$$\sum_{i=1}^{m}\sum_{e=1}^{2}\sum_{k\epsilon S_{j,i}} B''_{jlike} = 1 \quad \forall j,l \tag{1}$$

$$\sum_{i=1}^{m} B'_{jli} = 1 \qquad \forall j,l \tag{2}$$

Constraints (3) and (4) assign each job, each of which consists of a different set of operations, to one machine and spindle.

$$\sum_{i=1}^{m}\sum_{k\epsilon S_{j,i}} B''''_{jik} = 1 \qquad \forall j \tag{3}$$

$$\sum_{i=1}^{m} B_{ji} = 1 \qquad \forall j \tag{4}$$

Constraints (5) define the number of turrets used for processing operations of each job. N is the parameter that defines the number of turrets working on a specific operation. Some operations, like turning, have the capacity to be processed by two turrets to achieve a faster pace in the machining procedure.

$$\sum_{i=1}^{m}\sum_{h\epsilon T_{j,i}} B'''_{jlih} = N_{jl} \quad \forall j,l \tag{5}$$

Constraints (6) and (7) prevent simultaneous machining of two operations of the same job if they are not of the same type.

$$St_{jl} - St_{jl'} \geq B''_{jl'ike'}.P_{jl'i} - M(1 - Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 1 \ , e, e', i\ , k \in \{S_{j,i}\} \tag{6}$$

$$St_{jl'} - St_{jl} \geq B''_{jlike}.P_{jli} - M(Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 1 \ , e, e', i\ , k \in \{S_{j,i}\} \tag{7}$$

Constraints (8) to (11) are provided to allow for simultaneous machining of two different operations of the same job only if they are of the same type (either rotation of the spindle or the cutting tool). W is the parameter that defines if two operations are of the same type or not. If two operations of the same job are of the same type, W takes the value of zero and otherwise takes the value of one.

$$St_{jl} - St_{jl'} \geq -M(1 - Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 0 \ , e, e', i, k \in \{S_{j,i}\} \quad e \neq e' \tag{8}$$

$$St_{jl'} - St_{jl} \geq -M(Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 0 \ , e, e', i, k \in \{S_{j,i}\} \quad e \neq e' \tag{9}$$

$$St_{jl} - St_{jl'} \geq B''_{jl'ike'}.P_{jl'i} - M(1 - Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 0 \ , e, e', i, k \in \{S_{j,i}\} \quad e = e' \tag{10}$$

$$St_{jl'} - St_{jl} \geq B''_{jlike}.P_{jli} - M(Z_{jljl'}) - M(1 - B''_{jl'ike'}) - M(1 - B''_{jlike})$$
$$\forall j, l < l' \ \ni W_{jll'} = 0 \ , e, e', i, k \in \{S_{j,i}\} \quad e = e' \tag{11}$$

Constraints (12) and (13) are provided to prevent each spindle from holding more than one job at any time.

$$St_{jl} - St_{j'l'} \geq B''''_{j'ik}.P_{j'l'i} - M(1 - Z_{jlj'l'}) - M(1 - B''''_{j'ik}) - M(1 - B''''_{jik})$$
$$\forall j < j', l, j < n,, l', i\ , k \in \{S_{j,i} \cap S_{j',i}\} \tag{12}$$

$$St_{j'l'} - St_{jl} \geq B''''_{jik}.P_{jli} - M(Z_{jlj'l'}) - M(1 - B''''_{j'ik}) - M(1 - B''''_{jik})$$
$$\forall j < j', l, j < n,, l', i\ , k \in \{S_{j,i} \cap S_{j',i}\} \tag{13}$$

Constraints (14) and (15) state that each of the turrets used for machining purposes is capable of processing one operation at a time.

Journal of Optimization in Industrial Engineering
Vol.13, Issue 2, Summer & Autumn 2020, 1-15
DOI:10.22094/JOIE.2020.1869865.1660

$$St_{jl} - St_{jl'} \geq B'''_{jl'ih}.P_{jl'i} - M\left(1 - Z_{jljl'}\right) - M\left(1 - B'''_{jl'ih}\right) - M\left(1 - B'''_{jlih}\right)$$
$$\forall j, l < l', i, h \in \{T_{j,i}\} \quad (14)$$

$$St_{jl'} - St_{jl} \geq B'''_{jlih}.P_{jli} - M\left(Z_{jljl'}\right) - M\left(1 - B'''_{jl'ih}\right) - M\left(1 - B'''_{jlih}\right)$$
$$\forall j, l < l', i, h \in \{T_{j,i}\} \quad (15)$$

Constraints (16) and (17) indicate that each turret has the capability to process one work-piece at a time.

$$St_{jl} - St_{j\prime l'} \geq B'''_{j\prime l'ih}.P_{j\prime l'i} - M\left(1 - Z_{jlj\prime l'}\right) - M\left(1 - B'''_{j\prime l'ih}\right) - M\left(1 - B'''_{jlih}\right)$$
$$\forall j < j', l, j < n, , l', i, h \in \{T_{j,i} \cap T_{j',i}\} \quad (16)$$

$$St_{j\prime l\prime} - St_{jl} \geq B'''_{jlih}.P_{jli} - M\left(Z_{jlj\prime l'}\right) - M\left(1 - B'''_{j\prime l'ih}\right) - M\left(1 - B'''_{jlih}\right)$$
$$\forall j < j', l, j < n, , l', i, h \in \{T_{j,i} \cap T_{j',i}\} \quad (17)$$

Constraints (18) are provided to satisfy the precedence relations between operations of each job.

$$St_{jl} - St_{jl'} \geq P_{jl'i} \qquad \forall i, j, l < l' \in \{R_{j,l}\} \quad (18)$$

Constraints (19) and (20) are to assure that each job is assigned to a spindle and a turret that is capable of holding and/or processing that job.

$$B''_{jlike} = 0 \qquad \forall j, l, i, e, k \notin \{S_{j,i}\} \quad (19)$$

$$B'''_{jlih} = 0 \qquad \forall j, l, i, h \notin \{T_{j,i}\} \quad (20)$$

Constraints (21) to (24) state that if a job is assigned to a machine, then its operations are also assigned to the spindles, turrets, and capacities of the same machine.

$$B'_{jli} \leq B_{ji} \qquad \forall j, l, i \quad (21)$$

$$B''_{jlike} \leq B'_{jli} \qquad \forall j, l, i, e, k \in \{S_{j,i}\} \quad (22)$$

$$B'''_{jlih} \leq B'_{jli} \qquad \forall j, l, i, h \in \{T_{j,i}\} \quad (23)$$

$$B''_{jlike} \leq B''''_{jik} \qquad \forall j, l, i, e, k \in \{S_{j,i}\} \quad (24)$$

Constraints (25) determines the maximum completion time of each operation

$$C_{max} \geq St_{jl}+P_{jli} \qquad \forall j,l \tag{25}$$

Constraints (26) to (31) outline the decision variables.

$$St_{jl} \geq 0 \qquad \forall j,l \tag{26}$$

$$B_{ji} \in \{0,1\} \qquad \forall j,i \tag{27}$$

$$B'_{jli} \in \{0,1\} \qquad \forall j,l,i \tag{28}$$

$$B''_{jlike} \in \{0,1\} \qquad \forall j,l,i,k,e \tag{29}$$

$$B'''_{jlih} \in \{0,1\} \qquad \forall j,l,i,h \tag{30}$$

$$B''''_{jik} \in \{0,1\} \qquad \forall j,i,k \tag{31}$$

$$Z_{jljl'} \in \{0,1\} \qquad \forall j,l,l' \tag{32}$$

$$Z_{jlj'l'} \in \{0,1\} \qquad \forall j,l,j',l' \tag{33}$$

3.2 Demonstrative example

It is assumed that there are two machines (*m*), three different jobs (*n*), and that each job has a different number of *l* operations. One possible schedule is shown in Figure 3.

Each spindle is capable of holding a different series of jobs. For instance, $S_{3,2}$= {Ø}, which means there is no spindle on machine two which can hold job three. This is reflected in the final schedule where job three is scheduled on machine one. In this example, all turrets are assumed to be able to process all jobs for simplification purposes ($T_{j,i=}${1,2}). However, the set of turrets capable of processing a particular type of job can vary, and the model is capable of distinguishing them.

Precedence relations between operations (R_{jl}) must be satisfied when a schedule plan is created. For instance, operation three of job two has a precedence operation which is operation number two: $R_{2,3}$={2}. If two operations of the same job do not have a necessary precedence to be satisfied, they might start at the same time or their schedule might overlap only if they have the same operation mode. In order to check the operation mode, the model will refer to the data available for $W_{jll'}$. As shown in Figure 3, operations four and two of job one ($O_{1,2}$ and $O_{1,4}$) have an overlap because $O_{1,2}$ and $O_{1,4}$ do not have precedence relations with each other. Also, $W_{1,2,4}$={0}, which means operation two and four of job one are of the same type of operation (either milling or turning). This mechanism is the same for operations three and five of job one ($O_{1,3}$ and $O_{1,5}$) and operations three and four of job three ($O_{3,3}$ and $O_{3,4}$).

For job two, operations two and three are processed by two turrets on spindle number one of machine two. This is the concept related to processing a single operation of a single job by multiple turrets. The information regarding the number of turrets needed for processing each operation is provided. $N_{2,3}$={2} means operation number three of job two needs two turrets for machining. The parameter *N* for the rest of operations and jobs is one.

Each job has a different processing time on each machine. and the information regarding processing times is provided in **Error! Reference source not found.**. Another feature that is included in this illustrated example is the single setup concept, which involves completing all jobs without transferring them to other machines. For example, job one and three are completed on machine one and job two is processed on machine two. The total completion time of all operations on all machines ($C_{\max}$) is 20 seconds. The model will seek for the best allocations according to the information provided to complete the machining of all operations in the minimum amount of time.

The following data are assumed for a set of jobs and machines to be scheduled.

m=2 n=3 lj= {5,3,4}

$S_{1,1}$={1} $S_{1,2}$={1,2} $S_{2,1}$={2} $S_{2,2}$={1,2} $S_{3,1}$={1,2} $S_{3,2}$={Ø}

$T_{1,1}$={1,2} $T_{1,2}$={1,2} $T_{2,1}$={1,2} $T_{2,2}$={1,2} $T_{3,1}$={1,2} $T_{3,2}$={1,2}

$R_{1,1}$={Ø} $R_{1,2}$={1} $R_{1,3}$={Ø} $R_{1,4}$={1} $R_{1,5}$={4}

$R_{2,1}$={Ø} $R_{2,2}$= {1} $R_{2,3}$={2}

$R_{3,1}$={Ø} $R_{3,2}$={1} $R_{3,3}$={Ø} $R_{3,4}$={Ø}

Table 1
Processing times for the demonstrative example

job	machine	Operation				
		1	2	3	4	5
1	1	3	6	2	7	3
	2	3	9	4	3	6
2	1	4	8	6	-	-
	2	4	7	5	-	-
3	1	7	2	4	5	-
	2	8	5	8	6	-

$N_{1,1}$={1} $N_{1,2}$={1} $N_{1,3}$={1} $N_{1,4}$={1} $N_{1,5}$={1}

$N_{2,1}$={1} $N_{2,2}$={2} $N_{2,3}$={2}

$N_{3,1}$={1} $N_{3,2}$={1} $N_{3,3}$={1} $N_{3,4}$={1}

$W_{1,1,2}$={1} $W_{1,2,3}$={0} $W_{1,3,4}$={0} $W_{1,4,5}$={1} $W_{1,2,4}$={0} $W_{1,3,5}$={0}

$W_{2,1,2}$={1} $W_{2,2,3}$={0}

$W_{3,1,2}$={0} $W_{3,2,3}$={1} $W_{3,3,4}$={0}

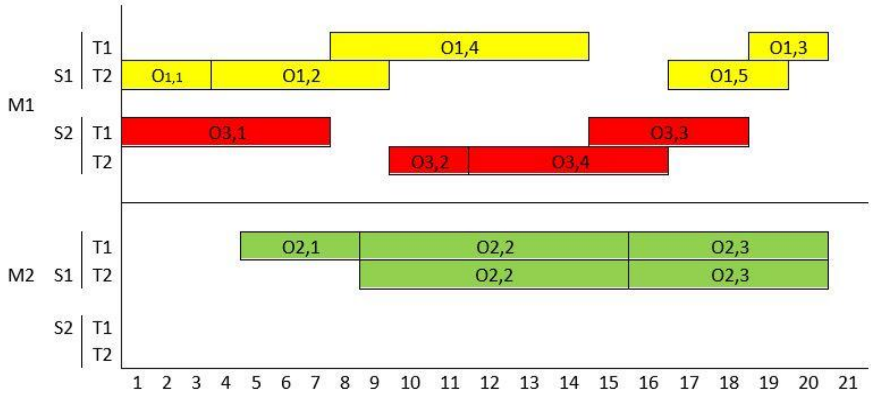

Fig. 3. Resultant schedule for the example

4. Simulated Annealing

Since the problem is computationally intractable, a metaheuristic algorithm is developed to solve larger sets of the problem. SA functions based on concepts related to annealing procedures and is used for optimization purposes (Černý, 1985; Kirkpatrick *et al.*, 1983). As shown in the flowchart in Figure 4, SA uses a framework that helps the algorithm to avoid getting surrounded by local optima. It provides a scheme to accept worse solutions for the problem with a negative exponential distribution $P = exp(-\Delta f/T)$ where $\Delta f = f(x_{new}) - f(x)$ (Metropolis *et al.*, 1953).

T is the parameter related to the acceptance criteria. At lower temperature values, worse solutions have lower chances of being accepted. The function $T = \alpha T$ is used to reduce the temperature, where α is considered as the rate at which temperature is decreased.

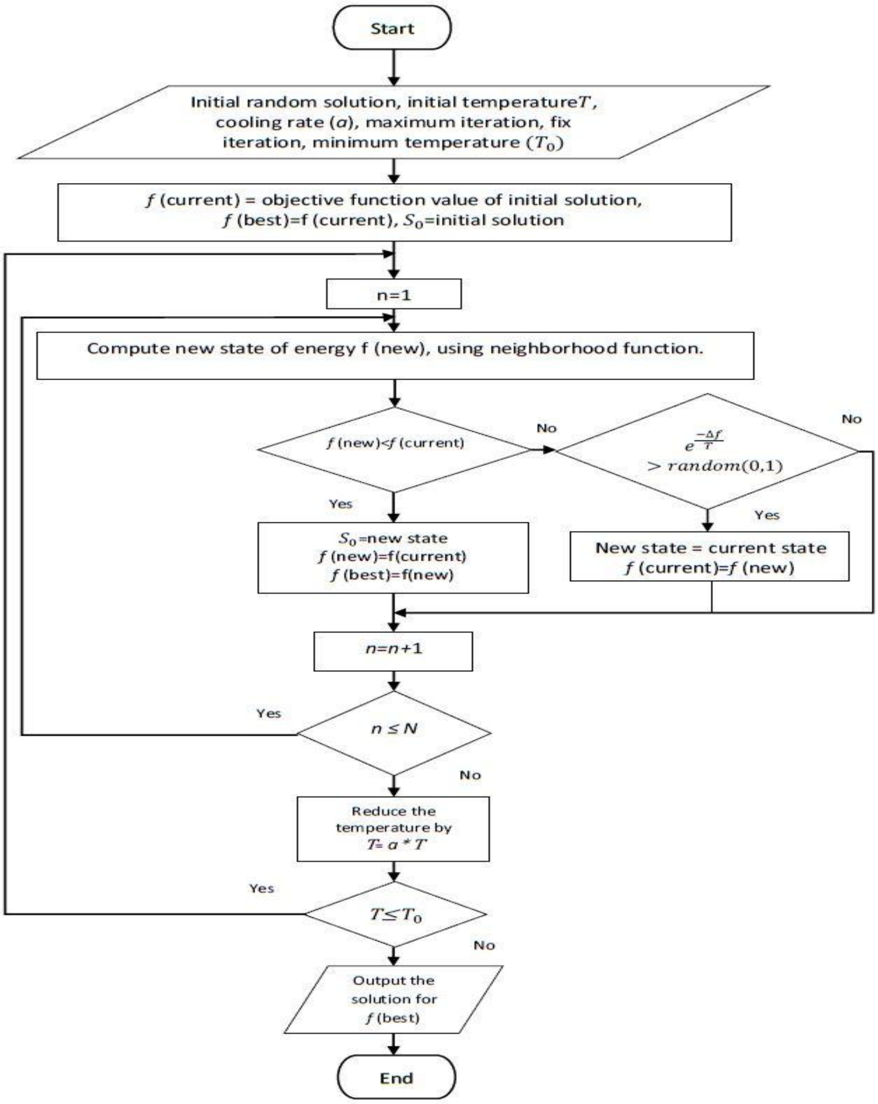

Fig. 4. Flowchart for Simulated Annealing

4.1. Initialization

An initial solution determines the first sequence at which operations are to be scheduled. The assignments of operations to machines, spindles, and turrets are done through a set of directions as shown in Figure 5. To tackle this problem, an initial random order of available jobs to be scheduled is created. Next, in order to apply the single setup concepts, each job is repeated to the number of its operations.

For instance, consider four jobs (*n*=4) where each job has the following number of operations: $l_j = \{3,5,3,4\}$. In this sequence $l_1 = \{3\}$ means the first job has three operations, $l_2 = \{5\}$ means the second job has five operations, $l_3 = \{3\}$ means the third job has three operations, and finally $l_4 = \{4\}$ means job number four has four operations.

One initial solution with these four jobs and their following operations could be I={4,4,4,4,2,2,2,2,2,3,3,3,1,1,1}. To create such a solution, an initial random order of jobs I={4,2,3,1} is created. Next, each job is repeated to the number of its operations. In this example, job number four has four operations. Therefore, it is repeated four times. Job number two has five operations and is repeated five times and so on. This mechanism is showed in Figure 5, where a random sequence of jobs (*J*) is created and called *r*. Each job will then be repeated as many times as its operations starting from the first job and put into an empty string called *q*.

Upon creation of the initial solution, it is put into a matrix *M*, which controls the assignment of operations on spindles, machine tools, and turrets. The variable number of columns in the matrix *M* is the total number of operations (*le*) of all jobs. *M* always has seven rows that are structured as follows:

- Row one: List of operations (*e.g.*, 4,4,4,4,2,2,2,2,2,3,3,3,1,1,1 for the for the initial solution *I*)
- Row two: Order in which operations are assigned, which may not be sequential due to precedence relations and is determined last based on the information in the other rows
- Row three: Machine number assigned to each operation
- Row four: Spindle dedicated to each operation
- Row five: Turret(s) used for processing each operation
- Row six: Start time of each operation
- Row seven: Finish time of each operation

For the initial solution *I*, the matrix *M* has seven rows and fifteen columns (total number of operations of all jobs). Once the initial solution is created and operations are listed in the matrix *M*, the rest of the assignment procedure will start through a set of rules.

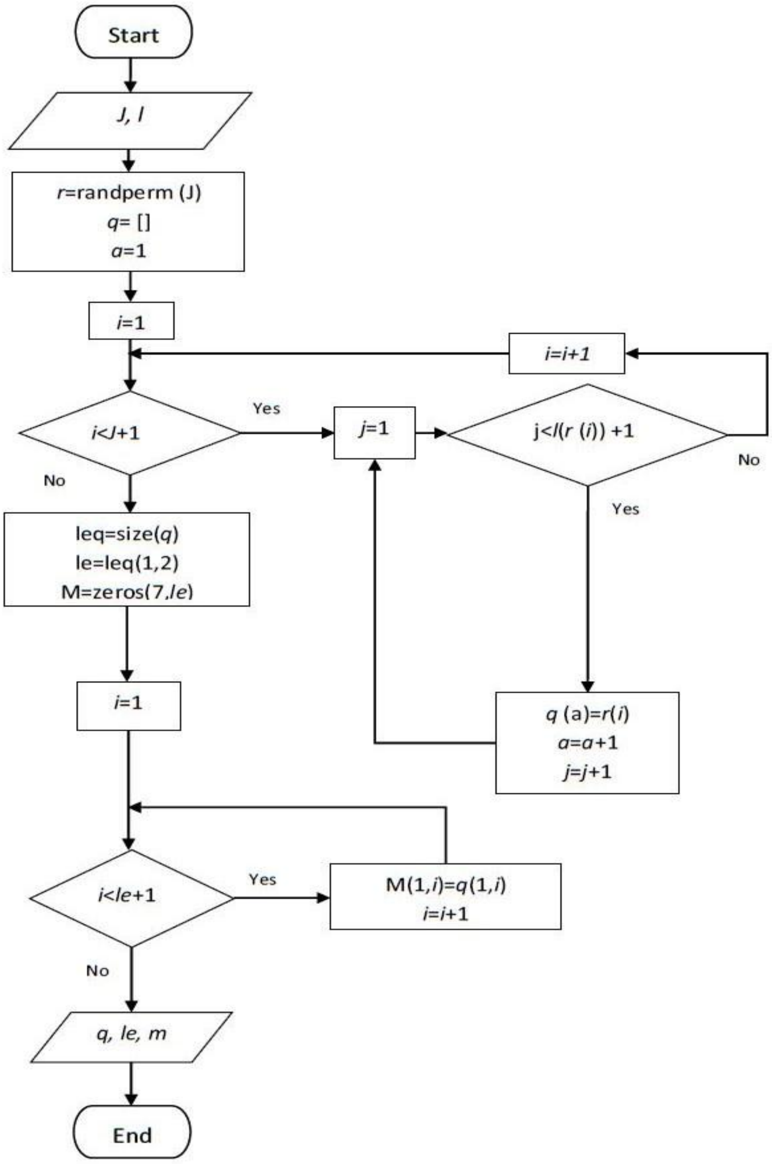

Figure 5. The flowchart for the initialization phase

Journal of Optimization in Industrial Engineering
Vol.13, Issue 2, Summer & Autumn 2020, 1-15
DOI:10.22094/JOIE.2020.1869865.1660

4.2. Neighborhood search

For the proposed metaheuristic algorithm, three different move operators are applied as neighborhood search structures. In each move operator, two different job numbers in the solution are chosen at random. For each move operator, the following action is then performed:

- Swap: The positions of the two job numbers are changed.
- Reversion: The order of job numbers, including the two jobs already selected, are reversed.
- Insertion: One of the job numbers is moved to the position right after the other.

To demonstrate how each move operator works, assume that there are six jobs and that the order of the job numbers to be scheduled is A={5,2,4,1,3,6}. Furthermore, define B as the order of the job numbers after applying a move operator. In addition, if a move operator requires two random job numbers to be selected, then assume that jobs two and three are chosen. For each move operator, the result B of applying the move operator under these assumptions to A is as follows:

- Swap: B=$\{5, \mathbf{3}, 4{,}1, \mathbf{2}, 6\}$.
- Reversion: B=$\{5, \mathbf{3}, \mathbf{1}, \mathbf{4}, \mathbf{2}, 6\}$.
- Insertion: B=$\{5{,}4{,}1{,}3, \mathbf{2}, 6\}$.

The method applied in this work is a mixed operator in which one random move operator is chosen for creating a new solution. This approach enables the algorithm to try different random solutions and discover the best sequence.

5. Test Cases

In this section, the MILP model is evaluated. In addition, the SA algorithm is tuned and then tested.

5.1 Input Data, Performance Measures, and Implementation

Eight test cases of different sizes are used to evaluate the MILP model and the SA algorithm. Full data for all instances are available in the online data repository associated with this paper.

Small instances are used to assess the mechanism and complexity of the mathematical model as well as the algorithm. By testing larger instances of the problem, the performance of the proposed algorithm is assessed.
The relative percentage deviation (RPD) method is used to analyze the SA algorithm. The calculation for RPD depends on the size of the instance.

For a small instance, RPD considers the minimum solution obtained from the MILP model and compares it with the SA algorithm solution:

$$RPD_{small\ instance} = \frac{SA\ Objective\ function\ value - Exact\ optimal\ objective\ function\ value}{Exact\ optimal\ objective\ function\ value} \times 100$$

For a large instance, RPD is calculated as:

$$RPD_{large\ instance} = \frac{Current\ objective\ function\ value - Best\ solution\ so\ far\ objective\ function\ value}{Best\ solution} \times 100$$

FICO Xpress Optimization software version 7.6 is used to algebraically model the problem at hand. Each run is set to terminate after 2000 seconds. SA is executed in MATLAB version R2015a on a PC with 3.2 GHz Intel Core 5 and 12 GB of RAM.

5.2 Size Complexity of the MILP

The mathematical model is evaluated through the size of the problem and number of decision variables and constraints applied in the model. In order to determine the size complexity, it is assumed that there are n jobs where all jobs have the same number of l operations. It is assumed each job has two sets of precedence relations among its operations and there are m machines in which all machines have dual spindles *(S)* and turrets *(T)*. Each operation is assumed to need one turret for machining purposes *(N=1)* and all operations are considered to be the same type *(W=0)*. For different problem instances tested using FICO Xpress, Table 2 shows the results for the size complexity of each problem instance.

Table 2
Size complexity for the proposed model

Job	Operation	Machine	Binary variables	Continuous variables	Constraints
5	5	3	821	25	9,965
10	5	3	2,266	50	34,930
15	5	5	5,176	75	124,455
15	10	5	19,726	150	991,380
20	10	10	31,301	200	1,721,840

5.3 Parameter tuning

SA has four parameters: maximum inner loop iteration or fix, annealing temperature (T), temperature reduction rate (α), and move operators. Naderi and Azab (2015) develop a similar SA algorithm and tune the parameter values using Taguchi design of experiments with the following levels:

- Inner Loop Iterations: 10, **50**, 100
- Annealing Temperature: 50, 200, **500**
- Temperature Reduction Rate: 0.95, **0.97**, 0.99
- Move operators: Shift, swap, **mixed**

The selected values have the lowest RPD and are shown in bold.

In addition to these parameters, a value for the number of iterations per outer loop must be chosen. Table 3 and Figure 6 show the computational results for a sensitivity analysis conducted for this parameter. In the analysis, the tuned parameter values shown in bold are used. Three possible values are tested: **250**, 125, and 65. The selected value has the lowest average RPD and is shown in bold.

The sensitivity analysis shows that RPD generally tends to rise with a decrease in the total outer loop iterations. However, this is not true for all test cases. For instance, the average RPD for test four, which has ten jobs and five machines, has decreased by the change in the outer loop iterations from 250 to 125. It could be discussed that the initial solution of the problem affects the inner and outer iterations required to find the near-optimum solutions. The initial solution created for this algorithm creates an initial random order of operations of each job to be scheduled. Based on the initial order, the first objective function value will be calculated and set as the best solution found so far. Therefore, the number of iterations required to find improved solutions in the neighborhood of the initial solution will affect the mechanism of the algorithm along with the four other parameters.

Table 3
Sensitivity of the outer loop iteration for the proposed SA

Size complexity			**Computational results**		
Test No.	**n**	**m**	**Average RPD /250 iteration**	**Average RPD /125 iteration**	**Average RPD /65 iteration**
1	4	2	0.52632	0.70175	0.70175
2	5	3	0.41237	0.61856	0.41237
3	10	3	3.07692	3.33333	3.93162
4	10	5	6.10390	5.97403	7.14286
5	15	3	3.42723	3.89671	3.66197
6	15	5	5.08772	5.26316	5.55556
7	20	3	2.12219	2.66881	3.18328
8	20	5	3.02789	3.18725	4.10359
Average			2.97307	3.20545	3.58662

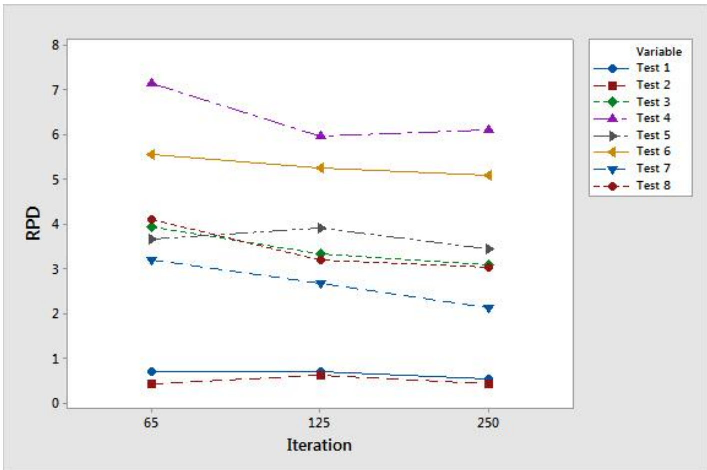

Fig. 6. Variation of RPD for different test cases with change in the outer loop iteration

5.4 Performance evaluation of the algorithm

SA is tested for small to larger instances. The SA results for small instances are compared with those obtained from the objective function of the mathematical model. For large instances, the results of the SA algorithm are compared with the previous works in the literature. Regardless of the size of the instance, the parameter values shown in section 5.3 are used in the proposed algorithm.

For small instances, four and five jobs are considered. For large instances, ten, fifteen and twenty jobs are selected. Each test case is tested for ten runs.

The results of the implemented algorithm are provided in Table 4 and Figure 7.

It is shown that the SA algorithm provides the lowest RPD for small instances. As the instance size increases, RPD tends to rise and peaks in the case with 10 jobs and 5 machines. Although the size increment of the problem has affected the average RPD, this is not the case for all test cases. Therefore, it can be concluded that the size complexity has not affected the performance of the proposed algorithm.

Another point that can be obtained from the results is that RPD tends to rise with increment in the number of machines applied.

The average of the RPDs obtained from the test cases is compared with the work of Naderi and Azab (2015). The average results of the RPDs are better than the only SA algorithm applied so far for the problem of scheduling multitask machines. The average standard deviation for test cases is less than 3.6, which is also a reasonable quantity for such an NP-complete problem.

Table 4
Computational results for SA

Size complexity			Computational results	
Test No.	**n**	**m**	**Standard Deviation**	**Average RPD (%)**
1	4	2	0.25000	0.52632
2	5	3	0.42164	0.41237
3	10	3	3.71782	3.07692
4	10	5	3.36815	6.10390
5	15	3	5.59861	3.42723
6	15	5	6.01941	5.08772
7	20	3	4.83506	2.12219
8	20	5	5.08156	3.02789
Average			**3.66153**	**2.97307**

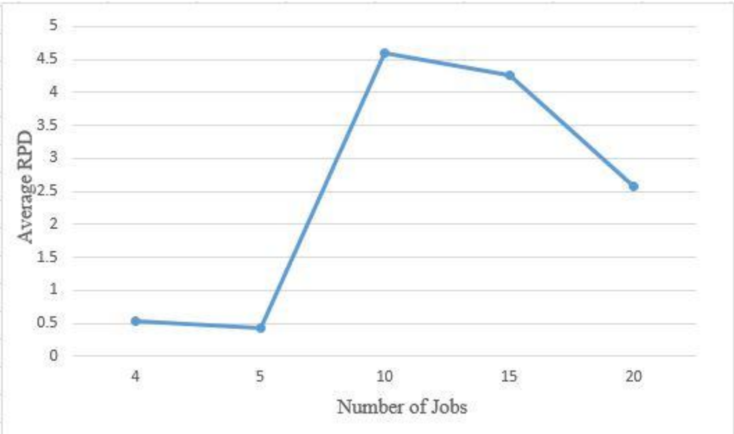

Fig. 7. The number of jobs versus the average RPD of SA

6. Conclusions

In this paper, the problem of scheduling operations on mill-turn machines is studied. The capacity of mill-turns in terms of simultaneous machining, high accuracy, and cycle time reduction makes them a suitable choice for today's manufacturing plants and shops. According to the literature, the focus of the limited research done so far on mill-turns has been on computer-aided process planning and issues related to it. The literature review also revealed that some critical features, such as the application of pinch turning, single setup, and definitions regarding the mode of operations, are absent in the previously proposed scheduling models. The problem is addressed through a MILP model. The model considers more realistic and practical features of these machines in comparison with previous works in the literature. Small test cases are developed in order to evaluate the proposed mathematical model. The complexity of the model in terms of decision variables and number of constraints applied is also evaluated. Due to the intractable combinatorial nature of the problem, a metaheuristic SA algorithm is developed to solve larger instances of the problem up to 20 jobs and 5 machines. Test cases show that SA is capable of solving all instances provided with the ability to solve some instances with less than one relative percentage deviation. The developed approach outperformed its counterparts in the literature; an average RPD of 2.97%, which is 0.23% improvement over the closest comparable approach.

In this research, a dual-spindle and dual-turret class of mill-turns is considered as a representative of a wide variety of these machine tools. Therefore, other configurations also need to be studied thoroughly as future work. Dual-spindle three-turret mill-turns or dual-spindle four-turret configurations are among those settings. These can be further explored to develop a model that represents their specific kinematics and mechanisms. Furthermore, attributes related to machining parameters, such as cutting speed and depth of cut and their effect on the scheduling problem, could also be further considered to cover other perspectives of mill-turns that have not been fully studied before.

Acknowledgements

A. Azab research is supported by Natural Sciences and Engineering Research Council (NSERC) grant number: 811008, 817019; M.F. Baki's research is funded by Research and Teaching Innovation Fund (RTIF) and Natural Sciences and Engineering Research Council (NSERC) Discovery Grant.

References

Azab, A., & Naderi, B. (2014). A variable neighborhood search metaheuristic for cellular manufacturing with multitask machine tools. *Procedia CIRP*, 20, 50-55.

Battaïa, O., Dolgui, A., Guschinsky, N., & Levin, G. (2014). Combinatorial techniques to optimally customize an automated production line with rotary transfer and turrets. *IIE Transactions*, 46(9), 867-879.

Battaïa, O., Gurevsky, E., Makssoud, F., & Dolgui, A. (2013). Equipment location in machining transfer lines with multi-spindle heads. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 12(2), 117-133.

Calleja, A., Fernández, A., Rodríguez, A., de Lacalle, L. N. L., & Lamikiz, A. (2015). Turn-milling of blades in turning centres and multitasking machines controlling tool tilt angle. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 229(8), 1324-1336.

Černý, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1), 41-51.

Chiu, N. C., Fang, S. C., & Lee, Y. S. (1999). Sequencing parallel machining operations by genetic algorithms. *Computers & Industrial Engineering*, 36(2), 259-280.

Choudhury, S. K., & Mangrulkar, K. S. (2000). Investigation of orthogonal turn-milling for the machining of rotationally symmetrical work pieces. *Journal of Materials Processing Technology*, 99(1-3), 120-128.

Crichigno Filho, J. M. (2012). Prediction of cutting forces in mill turning through process simulation using a five-axis machining center. *The International Journal of Advanced Manufacturing Technology*, 58(1-4), 71-80.

Fanjul-Peyro, L., Perea, F., & Ruiz, R. (2017). Models and matheuristics for the unrelated parallel machine scheduling problem with additional resources. *European Journal of Operational Research*, 260(2), 482-493.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598), 671-680.

Lee, Y. S., & Chiou, C. J. (1999). Unfolded projection approach to machining non-coaxial parts on mill-turn machines. *Computers in Industry*, 39(2), 147-173.

Levin, J. B., & Dutta, D. (1992). Computer-aided process planning for parallel machines. *Journal of Manufacturing Systems*, 11(2), 79-92.

Levin, J. B., & Dutta, D. (1996). PMPS: A prototype CAPP system for parallel machining. *Journal of Manufacturing Science and Engineering*, 118(3), 406-414.

Low, C., & Wu, G. H. (2016). Unrelated parallel-machine scheduling with controllable processing times and eligibility constraints to minimize the makespan. *Journal of Industrial and Production Engineering*, 33(4), 286-293.

Lu, K., Jing, M., Zhang, Y., & Liu, H. (2011). Industrial applications of chatter stability prediction and monitoring system for turning processes. *2011 IEEE International Conference on Mechatronics and Automation*, IEEE Press, 1923-1927.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087-1092.

Miller, P. C. (1989). Lathes turn to other tasks. *Tooling & Production*, 54(12), 54-60.

Miska, K. H. (1990). Driven tools turn on turning centers. *Manufacturing Engineering*, 104(5), 63-66.

Naderi, B., & Azab, A. (2015). Modeling and scheduling a flexible manufacturing cell with parallel processing capability. *CIRP Journal of Manufacturing Science and Technology*, *11*, *18-27*.

Norman, B. A., & Bean, J. C. (2000). Scheduling operations on parallel machine tools. *IIE Transactions*, 32(5), 449-460.

Rajkanth, R., Rajendran, C., & Ziegler, H. (2017). Heuristics to minimize the completion time variance of jobs on a single machine and on identical parallel machines. *The International Journal of Advanced Manufacturing Technology*, 88(5-8), 1923-1936.

Savas, V., & Ozay, C. (2008). The optimization of the surface roughness in the process of tangential turn-milling using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 37(3-4), 335-340.

She, C. H., & Hung, C. W. (2008). Development of multi-axis numerical control program for mill—turn machine. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(6), 741-745.

Yip-Hoi, D., & Dutta, D. (1993). Issues in computer-aided process planning for parallel machines. *Advances in Design Automation*, 65, 153-161.

Yip-Hoi, D., & Dutta, D. (1995). Data extraction from geometric models for process planning for parallel machines. *Journal of Manufacturing Systems*, 14(5), 307.

Yip-Hoi, D., & Dutta, D. (1996). A genetic algorithm application for sequencing operations in process planning for parallel machining. *IIE transactions*, 28(1), 55-68.

This article can be cited: Yavar, Saleh, et al. (2021). Machine Scheduling for Multitask Machining, ***Journal of Optimization in Industrial Engineering*. 13 (2), 1-15**

http://www.qjie.ir/article_673169.html
DOI:10.22094/JOIE.2020.1869865.1660