Simultaneous scheduling of machines and tools in a multimachine FMS with alternative routing using symbiotic organisms search algorithm

N. Sivarami Reddy*, **, M.Padma Lalitha*, S.P.Pandey** and G.S.Venkatesh*****

** Annamacharya Institute of Technology and Sciences, Rajampet, Andhra Pradesh, India.*
*** Director,REC, Azamgarh, Uttar Pradesh, India.*
**** Department. of Mechanical Engineering., VTU, Muddenahalli, Karnataka, India.*
***** Corresponding Author: siva.narapureddy@gmail.com*

Submitted ***:*** 25/05/2020
Revised ***:*** 05/06/2021
Accepted ***:*** 14/06/2021

ABSTRACT

This paper deals with simultaneous scheduling of machines and tools with alternate machines in a multimachine flexible manufacturing system (FMS) to minimize makespan (MS). Only one copy of each type of tools is made available due to economic restrictions, and the tools are stored in a central tool magazine (CTM) shared with and serves for several machines. The problem is to select machines from alternate machines for job-operations, allocation of tools to job-operations, and job-operations' sequencing on machines for MS minimization. This paper presents a nonlinear mixed integer programming (MIP) formulation to model the combined scheduling of machines and tools with alternate machines and a symbiotic organisms search algorithm (SOSA) built on the symbiotic interaction strategies that organisms employ to continue to exist in the ecosystem for solving the scheduling of machines and tools with alternate machines. The results have been tabulated and analyzed. There will be reduction in MS when the alternate machines are considered for job-operation.

Keywords: Alternate machines; Scheduling; FMS; Makespan; Optimization; Symbiotic organisms search algorithm.

1. INTRODUCTION

Manufacturing companies are expected to handle growing product complexities, shorter market time, new technologies, global competition threats, and quickly changing situation. FMS is setup to deal with manufacturing competition. FMS is an automated production system consisting of multipurpose machine tools which are computer numerical controlled (CNC), connected to an automated material handling system (MHS) (Agnetis et al., 1997). FMS aims to be flexible in manufacture without undermining the product quality. The flexibility of the FMS relies on the flexibilities of CNC machines, automated MHS, and control software. FMSs have been categorized into distinct kinds as per their workflow patterns, size, or manufacturing type. Four kinds of FMS are described from the planning and control point of perspective: single flexible machines (SFM), flexible manufacturing cells (FMC), multimachine FMS (MMFMS), and multicell FMS (MCFMS) (Saravanan and Noorul Haq, 2008). MMFMS is a class of FMS consisting of number of SFMs linked through an automated MHS that involves two or more pieces of equipment for material handling or is otherwise able to visit and serve two or more machines at a moment. Advantages such as cost reductions, enhanced utilizations, decreased work-in-process levels, and so on have already been proved by existing FMS implementations. FMS scheduling most significant goal is to boost resource usage and decrease idle time. Use of resources is improved by scheduling tasks so as to reduce the MS (Jyothi, 2012). One way to achieve high productivity in an FMS is to solve optimally or near optimally scheduling problems. A significant and active research area remains the effective and efficient scheduling strategies development for FMS. For the scheduling of N-machines and M-jobs problems, $(M!)^N$ sequences shall be examined for any performance to propose an optimal sequence. This means the search area is increasing exponentially due to a larger problem that makes scheduling an NP-hard problem. In fact, taking into account the job shop's scheduling problem (Lenstra and Kan, 1979) is NP-hard problem. Due to flexibility of the FMS, a given operation can usually be performed on a number of machines because of which jobs need not wait for a particular machine if it is busy, and it can be performed on an alternative machine. This causes reduction in waiting time and hence reduction in makespan. Because of alternative machines, the scheduling problem complexity increases further, and also it is expected that the availability of alternative machines improves the utilization of machines and tools. This paper considers machines and tools simultaneous scheduling with alternate machines and one copy of every type of tools without considering tool transfer times between machines for minimization of MS. Nonlinear MIP model and application of CSA which offers a technique of arriving at optimal operations sequence along with allocation of machines and tools to each operation are introduced.

1.1. Literature Review

The review of machines and tools scheduling is given below (Sivarami Reddy et al., 2019). Tang and Denardo (1988) addressed the problem of finding out the job sequence and tools which are placed before each job is processed on the machine for minimization of the tool switches. Chandra et al (1993) proposed a practical approach for deciding the jobs and tools sequence for minimization of the total setup and processing times, to make sure that jobs are performed before their delivery dates. Song et al. (1995) mentioned a heuristic algorithm for allocating parts and tools for minimization of tool switches between machines where every part needs to visit only one machine for its complete processing. Roh and Kim (1997) examined allotment of part and tool, and scheduling issues for the entire tardiness minimization under its tool movement plan where every part needs to visit only one machine for its complete processing in FMS. Agnetis et al. (1997) addressed joint part tool scheduling in an FMC and shown that tools are to be placed in a CTM and shared by machines using TT to minimize MS and maximum delay. When two machines are in need of the same tool, the Tabu search algorithm is applied to address the disagreement. Tsukada and Shin (1998) described a distributed artificial intelligence approach for tool scheduling and borrowing problem along with a strategy to share the tool in FMS to deal with the unexpected jobs' introduction in the dynamic environment. It is demonstrated that sharing of tool among distinct FMS cells helps to reduce tooling costs and improves the usage of tools. Jun, Kim, and Suh (1999) suggested a Greedy Search Algorithm for scheduling and supplying of tools, to decide the number of additional tools required when the FMS setup changes due to the change in product mix for MS minimization in FMS. Keung, Ip, and Lee (2001) investigated machine allocation and job shop scheduling on a parallel machining workstation in tandem with different parallel machines where tools are shared for minimization of tool switching cases and tool switches by means of GA approach. Ecker and Gupta (2005) proposed an algorithm to task precedence relationships with the intend of sequencing tasks so as to minimize the overall time required for tool changes on a SFM equipped with tool magazine, where every job requires one of the tools. Prabhaharan et al. (2006) addressed joint scheduling of operation and tool for MS minimization by using preference dispatching rule and simulated annealing algorithms in an FMC with "m" indistinguishable work cells and a CTM. Karzan and Azizoğlu (2008) suggested a Branch-Bound algorithm and a beam search technique with tool magazine's minimal capacity for jobs ordering and tool transporter (TT) movement such that TT movements are the minimum on a single versatile machine between machine and tool crib area. Suresh Kumar and Sridharan (2009) tackled the problems of tools sharing and scheduling in an FMS for mean tardiness, flow time, and conditional mean tardiness minimization as objectives employing "job scheduling preference rules and scheduling preference rules". Ant Colony Optimization Algorithm was suggested by Udhay kumar and Kumanan (2010) to produce optimal job and tool sequences to minimize makespan for FMS, and extended Giffer and Thompson algorithms (GTA) were used for generation of active schedule. Udhaya Kumar and Kumanan (2012) suggested untraditional optimization algorithms like PSO, SA, ACO, and GA to produce the optimum jobs and tools sequence for tardiness minimization in FMS, and modified GTA were used to generate efficient schedules. Aldrin Raj et al. (2014) suggested 4 heuristics, including preference shipment rules, revised nondelay schedule generation algorithms with 6 dissimilar rules, revised GTA, and Artificial Immune System algorithms for resolving machine and tools combined scheduling in FMS which consists of 4 machines and one CTM to minimize MS. Özpeynirci (2015) introduced a time-indexed mathematical model for machines and tools scheduling in FMS to minimize MS. Costa et al. (2016) have developed a hybrid GA that implements a local search enhancement scheme to resolve ' p ' jobs on ' q ' parallel machines scheduling undertaking tool change activity triggered by tool wear. Sivarami Reddy N et al. (2017a) suggested a CSA to solve machines and tools simultaneous scheduling with a copy of every type of tools to produce optimal sequences in an MMFMS for minimization of makespan, and it is shown that CSA outperforms current algorithms. Sivarami Reddy et al. (2017b) suggested 3 nature inspired algorithms viz FPA, SOSA, and CSA to solve machines and tools simultaneous scheduling through a copy of every tool's type to produce optimal sequences in an MMFMS for minimization of MSN and shown that FPA outperforms the other proposed algorithms and existing algorithms. Beezao et al. (2017) modeled and addressed the identical parallel machines problem with tooling constraints for minimizing MS using an adaptive large neighborhood search metaheuristic (ALNS). It is demonstrated that ALNS outperformed other existing algorithms. Baykasoğlu and Ozsoydan (2017) addressed automatic tool changer (ATC) indexing problem and tool changing problem concurrently in order to reduce nonmachining times in automatic machining centers using simulated annealing algorithm. Paiva and Carvalho (2017) addressed job ordering and tool changing problem (SSP) to find out the jobs order and the tool loading order so as to minimize the tool changes. A methodology which employs graph representation, heuristic methods, and local search methods is used for solving sequencing problem. Such techniques are integrated along with classical tooling approach to solve SSP in an algorithmic local search scheme. Gökgür et al. (2018) addressed constraint programming models in environments of parallel machines to solve tools scheduling and allotment problems with prearranged tools quantity in system due to financial limitations to minimize MS. Job transport times and tool switch times among machines are not taken into account in the references reviewed above.

Sivarami Reddy et al. (2018a) suggested CSA for solving machines and tools simultaneous scheduling problems without considering job transfer times and considering tool switch times among machines with minimum MSN as an objective and shown that tool transfer times have significant effect on MSN . Sivarami Reddy et al. (2018b) proposed three nature inspired algorithms, namely, SOSA, CSA, and FPA for solving joint jobs, machines, and tools scheduling problems with one copy of every tool type without taking job transport times into account and taking tool travel times into account between machines with minimum MSN as objective and showed that transfer times of tools have substantial impact on MSN.

The references examined above have provided useful insights into machines and tools simultaneous scheduling in FMS where operations are performed on specific machines, and they proposed different solution approaches, but there were no attempts to consider machines and tools concurrent scheduling with alternative machines and a restriction on the copies of every type of tools without considering tool switch times between machines (SMATWAM) to minimize MS in an FMS. This problem is to be addressed as it is expected that there will be reduction in MSN and the cost of duplicating the tools in each and every machine. Automated tool sharing system presents technical solution to the tools' high cost, where different machines are able to employ the same tool by dynamically moving it from one machine to another as tooling needs evolving. Because FMS is an interconnected manufacturing facility, it is important to concurrently schedule different FMS components.

Heuristic optimization algorithms (heuristics) seek excellent viable alternatives to optimization problems in cases where the problem's complex nature or the restricted time on hand for its solution does not allow the problem addressed to have an exact solution by employing an efficient metaheuristic algorithm. Cheng and Prayogo (2014) developed the SOSA and tested the SOSA on 26 distinct benchmark functions and in 4 different problems of engineering design. Compared to other approaches, such as GA, DE, bees algorithm, PSO, and particle bee algorithm, it has been stated that SOSA gave better results. Sivarami Reddy et al. (2016) proposed SOSA to deal with machines and tools simultaneous scheduling where operations are performed on specific machines (SMTWSM) to produce best optimal sequences with a copy of every type of tools in an MMFMS that minimize makespan, and it is shown that SOSA outperforms the existing algorithms. Sonmez et al. (2016) proposed SOSA to solve the dynamic economic dispatch with valve-point effects problem and shown that SOSA produces better results and has good convergence than other methods. Wu et al (2016) employed SOSA for training feed forward neural networks and shown that SOSA performs better than the other algorithms in terms of converging speed. S Reddy and Sivarami Reddy (2017) addressed the machines and automated guided vehicles (AGVs) concurrent scheduling in a MMFMS by using symbiotic organisms search algorithm (SOSA) for minimization of MSN, and it is made known that the results of SOSA are promising. Sivarami Reddy et al (2021) addressed concurrent scheduling of machines, AGVs, TT, and tools in a MMFMS to minimize MS considering tool and job transfer times with a copy of every tool type. Parameter setting, however, is one of the optimization algorithms' disadvantages as it is a time-consuming work. Algorithms with few parameters to adjust are easier for implementation. In SOSA, no specific algorithm parameters are required for algorithm operations, so SOS has been employed for solving this simultaneous scheduling problem. In this paper, a nonlinear MIP model for makespan minimization has been introduced. A metaheuristic called SOS algorithm is built as the problem is NP hard and engaged for makespan minimization. Section II in brief describes the environment of FMS, assumptions, and definition of problem. Section III includes the model formulation, the flow chart for makespan computation, and the problem input details. The implementation of SOS algorithm is described in Section IV. Different instances' types are tested, and results are made known and analyzed in Section V. Conclusion is offered in Section VI.

2. PROBLEM FORMULATION

Setup varies in FMS types and operations. Because having a common configuration is not feasible, the majority of research focuses on defined manufacturing systems. Specification of the system, assumptions, criteria for objective, and the problem addressed in this work are offered in the subsequent sections.

2.1. FMS Environment

Using common tool storage is typically followed in several manufacturing systems through one CTM serving multiple machines to cut down tool stocks. During the part's machining, demanded tool is shared from different machines or shipped from CTM by TT. CTM cuts down the tool's copies required in system and therefore cuts down tooling cost. Switching of tools and jobs between the machines will be carried out by TT and AGVs, respectively. The FMS in this work is assumed to consist of four CNC machines, a CTM with necessary tool types for all machines and a copy of every type, a TT, two indistinguishable AGVs, and each machine with ATC. There are L/U

stations where FMS releases parts for manufacturing, and the completed parts are deposited and transported to the final storage facility, respectively. To stock up the work-in-progress, an automatic storage and retrieval system (AS / RS) is available. Figure 1 shows its configuration.

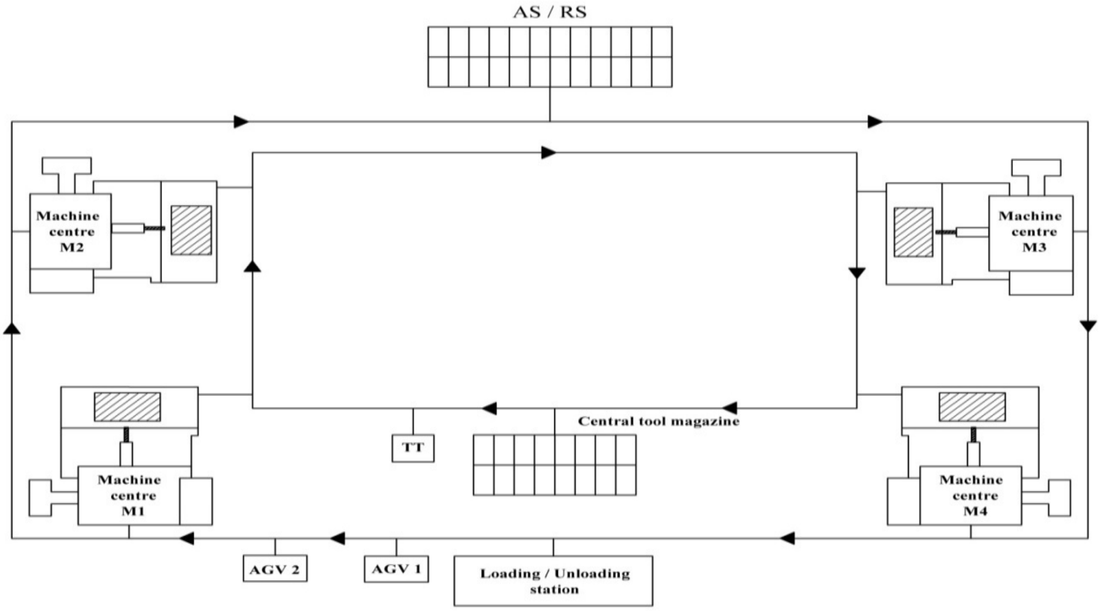

Figure 1. FMS environment.

2.2.1. Assumptions

The assumptions made are given below for the problem being studied.

- Initially all jobs, machines, and AGVs are available for use.
- Tool/ machine can perform only one job at any given time.
- Operation's preemption on machine is barred.
- The necessary machines and tools are identified before scheduling every operation.
- Every job has operations' set with its own order and known processing times.
- Processing times include setup times also.
- There is only one TT which moves the tools all through the FMS and tool switch times among machines are not taken into consideration.
- Tools are stored in a CTM.
- Transportation time of parts among machines is not taken into account.
- At the time of jobs' arrival, the service life given to the tools will be sufficient to perform the operations allocated to the respective tools.

2.2.2. Constraints

The process planning information for determining the sequence of operations for optimizing tolerance stack-up is provided in terms of precedence constraints for each job. Thus, we assume that each job has a prearranged operations order that cannot be altered.

2.2.3. Problem Definition

Consider an FMS with a job set of J of j job types {J_1, J_2, J_3... , J_j}, m machines {M_1, M_2,M_3 ... , M_m}, total operations of job set {1,2,3....n}, and k types of tools {t_1,t_2, t_3….t_k}with a copy of every type. Every machine can process different operations, and therefore every operation can be carried out on a number of alternative machines. A job's operations have a predetermined order of processing, and the sequence is known in advance. At most, one operation at a time can be processed by a machine. Until its predecessor operation is complete, an operation cannot begin.

The order of operations on a machine determines the setup requirements for a machine. The simultaneous scheduling problem is defined as determining the sequence of job-operations and assignment of machines and tools chosen from alternate machines and tool matrix, respectively, to those job-operations with a copy of every type that are shared for many operations among the machines for MS minimization in an MMFMS. The problem is stated in crisp, so unambiguous mathematical form is given in Section 3.

3. MODEL FORMULATION

In this section, a nonlinear MIP model is introduced to clearly specify the crucial parameters and their effect on the FMS scheduling problem.

3.1. Notations

Subscripts

j: index for a job
i, h,r,s: indices for operations
k: index for a tool
g: alternate machine index

Parameters and Sets

J = job sets on hand for processing

n_j = operations in job j

$n = \sum_{j \in J} n_j$, total operations in job set J

$I = \{1,2,3,----n\}$ index set for operations

$I_j = \left\{ N_j + 1, N_j + 2, ---- N_j + n_j \right\}$ is the indices set in I linked with job j , where N_j is the jobs' total operations indexed before j and $J_1 = 0$

$IF_i = I - \{h ; h \geq i , i, h \in I_j \}$ operations' index set without operation i and same job's following operations to operation i.

$IP_h = I - \{i; i \leq h, i, h \in I_j\}$, operations' index set without operation h and same job's preceding operations to operation i.

TL={ } the set of tool types to carry out the jobs' operations.

$R_k = \{\ \}$ indices set in I linked with tool type k in TL $k \in TL$

u= first operation that uses tool k, $u \in R_k, k \in TL$

v= preceding operation of i, $i, v \in R_k, k \in TL$

$RT_{kj} = I_j \bigcap R_k$ is the index set of operations in I common for tool k and job j for all k, j, $k \in TL, j \in J$

a_j = job j ready time

b_k = tool k ready time

$R_{(g)} = ready\ time\ of\ machine\ indicated\ by\ g\ \forall g = 1,2.....$

$R_i = ready\ time\ of\ machine\ for\ operation\ i$

$AM_i = \{(x, y) / x : \text{alternate machine, } y : \text{processing time for operation } i \text{ on that machine.}\}$ is the set of combination of alternate machines and processing times on those machines for operation i, for all i, $i \in I \cap I_j$, for all j, $j \in J$

$AM_i(g)$ = operation i processing time on the machine indicated by g $\forall g = 1,2,...., \forall i = 1,2,...n$

$S = (\max(R_{(g)}, b_k) + AM_i(g))$

$m_i = selected\ machine\ for\ operation\ \ i\ \ \forall i = 1,2,3,...n$

$MS = \{m3, m1, m4, m2, ------\}$ is a set of selected machines for operations associated in set I

$pt_i =$ operation i processing time on selected machine from available alternate machines.

$ct_i =$ operation i completion time on selected machine from available alternate machines.

Decision Variables

$$q_{rs} = \begin{cases} 1 & if\ c_r\ is\ less\ than\ c_s, where\ r\ and\ s\ are\ operations\ of\ different\ jobs \\ 0 & otherwise \end{cases}$$

$$\beta_{rs} = \begin{cases} 0 & if\ c_r\ is\ lessthan\ c_s\ , where\ r\ and\ s\ are\ operations\ of \\ & different\ jobs, r, s \in R_k \\ 1 & otherwise \end{cases}$$

3.2. Mathematical Model

The optimization problem is cast as minimization of objective function. The objective function, Z, may be conveniently chosen as the maximum completion time among all jobs' last operations i,e MS. The objective function for minimization of MS is

$$Z = \max(ct_i) \quad for\ all\ \ i,\ \ i \in I$$

Subject to the following constraints,

$$Z \geq ct_{N_j + n_j}\ \ \forall j, j \in J \tag{1}$$

The machine for an operation i from alternate machines is selected such that operation i is finished at the earliest as given in (2)

$$MS = \{m_i \,/\, S\ \ is\ \min.\ \ \forall g = 1,2..\} \qquad \forall i = 1,2,3,.....n \tag{2}$$

$$ct_i - ct_{i-1} \geq pt_i \qquad for\ all\ \ i-1, i \in I_j\,,\ j \in J\ \ ,i \in R_k,\ \ k \in TL \tag{3a}$$

$$ct_{N_j+1} \geq pt_{N_j+1} \qquad for\ all\ j\ \ and\ \ j \in J\ \ , N_j + 1 \in R_k, k \in TL \tag{3b}$$

$$ct_h \neq ct_i \quad h \neq i, for\ all\ \ h, i \in R_{k,}\ , k \in K \tag{3c}$$

$$\left.\begin{aligned} ct_r &\geq ct_s + pt_r - H\, q_{rs} \\ ct_s &\geq ct_r + pt_s - H(1 - q_{rs}) \end{aligned}\right\}$$

$$for\ all\ r \in I_j\ , s \in I_l\ \ where\ j, l \in J, j \langle\ l \tag{4}$$

$$\left.\begin{aligned} (1 + H\ \ \beta_{rs})\, ct_r &\geq ct_s + pt_r - H\, q_{rs} \\ (1 + H\, \beta_{rs})\, ct_s &\geq ct_r + pt_s - H(1 - q_{rs}) \end{aligned}\right\}$$

$$for\ all\ r \in I_j\ , s \in I_l\ \ where\ j, l \in J, j \langle\ l \tag{5}$$

$$\max(R_i, b_k) \leq ct_i - pt_{i,}\ \forall i,\ and\ i \in I \tag{6}$$

$$ct_i \geq 0\ for\ all\ i \in I$$

$$q_{rs} = 0,1\ for\ all\ \ r \in I_j, s \in I_l\ where\ j, l \in J, j\ \langle\ \ l$$

$$\beta_{rs} = 0,1\ for\ all\ \ r \in I_j, s \in I_l\ where\ j, l \in J, j\ \langle\ \ l, r, s \in R_k$$

In the formulation, machine and tool indices are not employed specifically because routing for each job is determined from the given multiple routes as per Equation (2), selected machines in the route are assigned to a set MS, and the machine and tool indices are known for each operation index in *I*. For the operations that belong to the same tool and different jobs, tool need not follow precedence constraints of jobs. For operations that belong to the same tool and same job, tool needs to follow precedence constraints of jobs. The objective of the nonlinear MIP formulation is MS minimization by reducing the cumulative finish times of the last operations of all jobs. The constraint set 3a is the operations' precedence constraints. The constraint set 3b is the constraint for the completion time of first operations of jobs. The constraint set 3c is the constraint for the operations that belong to the same tool. H is a large positive integer in the constraint sets 4 and 5 which ensures that no two operations allocated to the same machine or tool can be concurrently performed. The constraint 6 specifies that operation i can begin only after the maximum machine ready time and the tool ready time. The process times are nonnegative, and the decision variables take values of 0 and 1. However, this formulation is intractable due to its size and nonlinearity; thus, metaheuristic algorithm, namely, SOSA is used to obtain near optimal or optimal solutions.

Since the MS needs to be minimized, calculation of MS for a given schedule needs to be developed. Flowchart for such a computation is shown in Figure 2.

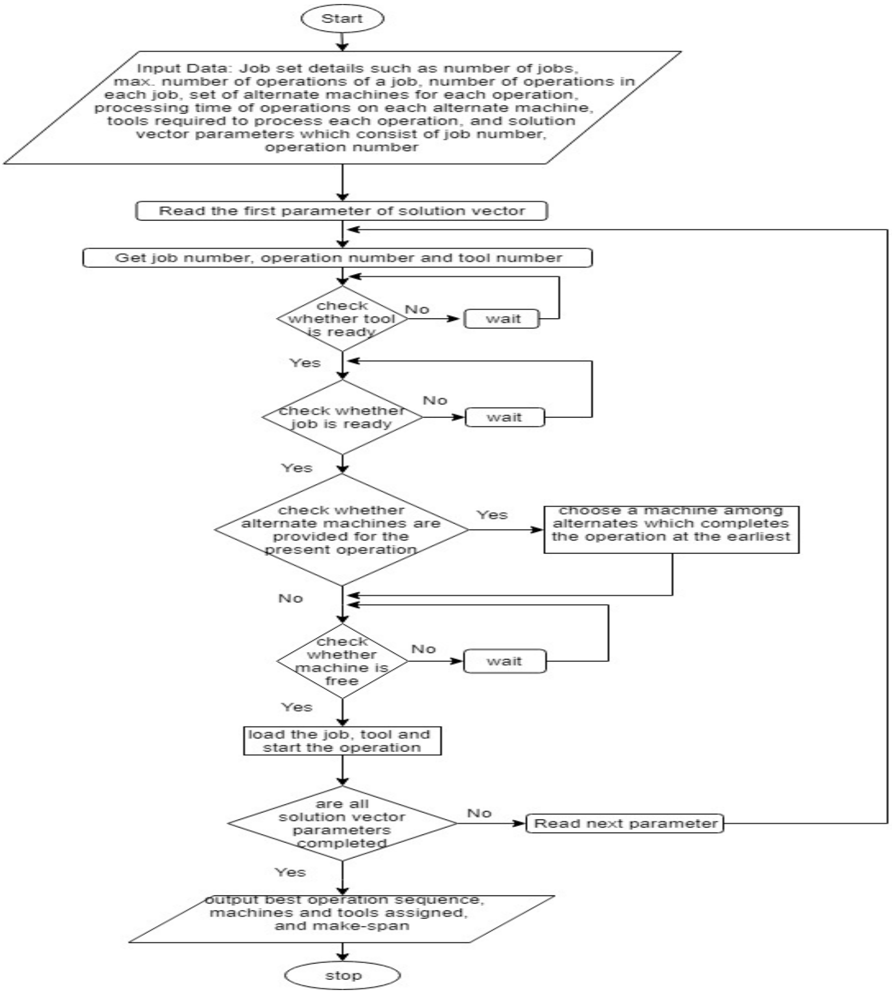
Start
Input Data: Job set details such as number of jobs, max. number of operations of a job, number of operations in each job, set of alternate machines for each operation, processing time of operations on each alternate machine, tools required to process each operation, and solution vector parameters which consist of job number, operation number
Read the first parameter of solution vector
Get job number, operation number and tool number
check whether tool is ready
No
wait
Yes
check whether job is ready
No
wait
Yes
check whether alternate machines are provided for the present operation
Yes
choose a machine among alternates which completes the operation at the earliest
No
check whether machine is free
No
wait
Yes
load the job, tool and start the operation
are all solution vector parameters completed
No
Read next parameter
Yes
output best operation sequence, machines and tools assigned, and make-span
stop

Figure 2. Flowchart for calculation of MS.

3.3. Input Data

Since the number of jobs, total operations, and machines in every job set differ from one to the other, TEN different job sets are considered for simulation. Same job sets employed by M.V. Satish Kumar et al. (2011) are employed for this problem, which are also the job sets employed by Bilge and Ulusoy (1995) but with the additional information such as the alternate machines and processing times on those machines to carry out the operations. The tools that are used to process the operations of jobs in above job sets are the same tools specified in the first ten job sets employed by Aldrin raj et al. (2014) which are also benchmark problems provided by Bilge and Ulusoy (1995) but with the additional information such as tools to carry out the operations. The job sets are given in Appendix I. Every job set has 5 to 8 different jobs, and each operation of a job is to be processed on alternate machines. Each job's entity in job set offers information about the alternate machines, operation processing time of a job on those machines, and tool needed for the operation. The data are as follows:

(i) number of jobs, each job's operations, and job's maximum operations in job set,
(ii) alternate machines needed for each job-operation (Alternate machine matrix),
(iii) time needed to carry out every job-operation on each alternate machine (process time matrix),
(iv) Tool to perform every job-operation (tool matrix) offered as an input.

4. IMPLEMENTATION OF SOS

The SOS algorithm operates on mutually beneficial behavior observed in nature between species. The relationship is known as symbiotic between two distinct species. Mutualism, commensalism, and parasitism are the prevalent symbiotic relationships observed in nature. Close relationship between exclusive species is called mutualism, leading in a mutual advantage. A connection between two separate species is called commensalism, which provides advantages to only one (without the other affecting). A connection between two separate species is called parasitism, which provides advantages to one species and harms the other.

The SOS algorithm begins with an initial population produced randomly, where there are n organisms in the system (i.e., eco size). In addition, in each phase, the updated solution is adopted if it has a superior functional value than the earlier one. The optimization course is repeated until the criterion of termination is met.

The fresh solutions are acquired through Equations (1) and (2) in the mutualism stage

$$X_i' = X_i + rand * (X_{best} - MV * BF1) \quad (7)$$

$$X_j' = X_j + rand * (X_{best} - MV * BF2) \quad (8)$$

where $MV = mean\ (X_i , X_k)$ (9)

BF1 = 1or 2 , BF2 = 1 or 2

The new solution is acquired through Equation(4) in the commensalism stage

$$X_i' = \mathbf{X}_i + rand * (X_{best} - X_k) \quad (10)$$

The parasite vector is produced in the parasite phase by changing the values of some of the ' Xi 'organism ' randomly selected design variables, modifying the randomly selected design variables using a randomly generated number within their bounds.

In this problem of simultaneous scheduling, SOS algorithm is employed to minimize makespan. Organism represents a feasible solution vector. Parameters in vector are same as job set’s total operations. In the solution vector, every parameter has to represent job-operation and the machine and tool that are allocated for job-operation. Therefore, job-operation, machine, and tool encoding is employed. The parameter’s first item represents the job number and the location in the order in which it happens in vector shows the operation number. The parameter's second item indicates the machine allocated to process that operation, selected from alternate machines, and the parameter's third item represents tool allocated to perform the operation selected from tool matrix. This coding is useful in checking precedent relationships between job’s operations in a vector.

Random Solution Generator (RSG)

RSG is devised that can offer solutions for the initial population. A solution vector is a constructed parameter after parameter by this generator. An operation must be eligible for assigning to a parameter. An operation is said to be qualified once all of its predecessors are allocated. Qualifying operations are placed in a set for scheduling next. In the beginning, this set is made up by the first operations of each job. At every iteration, one among the operations in the set is chosen arbitrarily and put next to the parameter in the vector. Then, machine and tool for that operation are picked up from alternate machines and tool matrix, respectively, for the operation and both are allotted to a parameter to end the vector. The set is kept up to date, and if the solution vector is not yet finished, the process will continue.

Limits function

It is used to ensure that the produced operations in a new solution are compatible with the precedence constraints requirement. If the precedence requirement constraints are not observed, the new solution will be corrected by the limits function so that the operations of the new solution vector will observe the precedence constraints requirement. SOS algorithm's step-by-step implementation procedure is outlined below.

Step 1: Eco system initialization.

- Initial population (organisms) is arbitrarily generated (positioned) through the use of RSG in a search space. Each member of population is a feasible solution and called as an organism.

$$\text{Initial population=}\begin{bmatrix} x_1{}^1 & x_2{}^1 & x_3{}^1 \\ x_1{}^2 & x_2{}^2 & x_3{}^2 \\ - & - & - \\ x_1{}^N & x_2{}^N & x_3{}^N \end{bmatrix}$$

where $x_1{}^i$ is operations sequence and $x_2{}^i$ and $x_3{}^i$ are machines allocation and tools allocation to the job-operations, respectively.

- Maximum iteration: reaching the maximum iteration is stopping criteria.

Step 2: Set i to 1 initially; solution x_1 is matched to x_i.

Step 3: Identify the best solution x_{best}.

Makespan for all vectors is determined by means of flowchart given in Figure 2, and identify the solution with the minimum makespan called as x_{best}.

Step 4: Mutualism phase

Use Equation 9 to determine the mutual vector. Benefit factors (BF1 and BF2) are obtained randomly by assigning either 1 or 2. New solutions of x_i and x_j are obtained using Equations 7 and 8. If the new solutions of x_i and x_j are infeasible, the use of limit function will make them feasible. Makespan values of new solutions are compared with old. Better ones are chosen as the next iteration solutions.

Step 5: Commensalism phase

Solution k is randomly selected such that $k \neq i$ and $k \neq j$.

New solution is obtained using Equation (10). If the fresh solution is infeasible, the use of limit function will make it feasible. Makespan value of fresh one is compared with the old one. If fresh one is superior, old one is replaced with new one.

Step 6: Parasitism phase

In random sizes, the parasite vector is developed employing a arbitrary number between the lower and upper bounds provided, and it is used to mutate x_i If the fresh solution of x_i is infeasible, the use of limit function will make it feasible. The new solution makespan is computed and compared to makespan of x_i.If it is superior, x_i is replaced with it.

Step 7: Repeat Steps from 3 to 6, if the current x_i is not the population's last member

Step 8: If termination requirements are not met, go to Step 2 and begin the next iteration; otherwise the best MS position in the memory is identified as the optimization problem’s solution.

The initial population is generated arbitrarily by the use of RSG in the proposed methodology. Every vector of the solution consists of parameters equal to the job set operations. The data mentioned in Section III.3 is offered as an input. The code is written in MATLAB and offers the schedule for job-operations and allocation of machines and tools to the corresponding job-operations.

At every generation, all candidates of population are selected for replacement. Therefore, NP competitions are provided to determine next generation members.

Example: Job set 5 is considered having five jobs, and operations in job set are 13. Therefore, the solution vector has 13 parameters and is given below with its job-operation, machine, and tool based coding:

3 3 3	1 1 4	1 4 1	3 2 4	5 3 2	3 2 2	2 3 2	4 4 2	4 3 3	5 1 1	2 2 1	1 1 4	2 3 3

Randomly generated solution vector for the initial population is shown below:

1 1 4	3 3 3	5 2 2	4 4 2	1 2 1	2 3 2	1 4 4	4 1 3	2 3 1	3 4 4	2 3 3	5 2 1	3 1 2

The generated solutions for initial population observe precedence constraints. Therefore, they are feasible solutions.

Assessment: Makespan value for each vector in the initial population is calculated, and the vector with the best makespan is chosen for the next generation.

For example, if the input vector is

(i) 4 4 2 1 1 4 3 3 3 2 2 2 5 3 2 4 1 1 3 4 4 2 2 1 2 3 3 5 2 1 1 4 1 1 4 4 3 1 2

after evaluating all the initial population vectors, the best (least makespan) is taken among them. Below is the best one in the initial population:

1 1 4 4 4 2 3 3 3 4 3 3 2 2 2 1 4 1 3 4 4 5 3 2 3 2 2 2 4 1 2 3 3 5 4 1 1 1 4

makespan=45.

Shown below is a vector j, and ‘ j ≠ i’ is picked up randomly:

(j) 1 1 4 3 3 3 5 2 2 4 4 2 1 2 1 2 3 2 1 4 4 4 1 3 2 3 1 3 4 4 2 3 3 5 2 1 3 1 2

New solutions are produced and provided below in the mutualism phase according to Equations 7 and 8.
New solution(i):

4 4 2 1 1 4 3 3 3 2 2 2 5 3 2 4 1 3 3 4 4 2 2 1 2 3 3 5 2 1 1 4 1 1 4 4 3 1 2

New solution(j):

1 1 4 3 3 3 5 2 2 4 4 2 1 2 1 2 3 2 1 4 4 4 1 3 2 3 1 3 4 4 2 3 3 5 2 1 3 1 2

Old solution will only be substituted by fresh solution, if there is better value for fresh solution. Otherwise, the old solution will continue.

New solution is produced as per Equation 10 and provided below in the commensalism stage.
A vector k ‘ k ≠ i’, and ‘k ≠ j’ is picked up randomly and shown below:

(k) 4 4 2 1 1 4 3 3 3 2 2 2 5 3 2 4 1 3 3 4 4 2 2 1 2 3 3 5 2 1 1 4 1 1 4 4 3 1 2

New solution(i):

4 4 2 1 1 4 2 2 2 3 3 3 4 1 3 5 3 2 3 4 4 2 2 1 2 3 3 5 2 1 3 1 2 1 4 1 1 4 4

Old solution will only be substituted by fresh solution, if there is better value for fresh solution. Otherwise, the old solution will continue.

In the parasitism phase, a vector l ‘l ≠ i ‘, ‘l ≠ j’, and ‘l ≠ k’ is picked up randomly to obtain a new solution and shown below:

(l) 1 1 4 3 3 3 5 2 2 4 4 2 1 2 1 2 3 2 1 4 4 4 1 3 2 3 1 3 4 4 2 3 3 5 2 1 3 1 2

New solution(i)

5 3 2 1 1 4 5 2 1 1 4 1 3 3 3 2 1 2 4 4 2 1 2 4 2 3 1 4 3 3 2 3 3 3 4 4 3 2 2

The new vector fitness value is assessed, and if it is superior to the solution i's fitness value, then vector i is substituted by new vector. This process is continued till candidates in population are over, and it is repeated until specified generations have been completed. The best sequence until now is given below:

1 1 4 4 4 2 3 3 3 4 3 3 2 2 2 1 4 1 3 4 4 5 3 2 3 2 2 2 4 1 2 3 3 5 4 1 1 1 4

5. RESULTS AND DISCUSSIONS

The proposed approach has been tested for SMATWAM on the job sets given in Appendix I. The proposed approach has been applied on different population sizes varying from 2 to 12 times of operations in the job set, and it is found that when population size is 10 times the operations, better results are noticed. Good results are obtained from 35 and 60 generations for most of the problems. So, 60 generations are taken into account as the stopping criteria. Simulation tests on the job sets given in Appendix I were performed 20 times for SMATWAM in a multimachine FMS for makespan minimization. The best MS from 20 simulation experiments is provided along with mean and standard deviation (SDV) in Table 1.

Table 1. Best MS of SMATWAM along with mean and SDV.

Job set Number	MS	mean	SDV
1	53	53	0.0000
2	54	54	0.0000
3	70	70	0.0000
4	54	54	0.0000
5	42	42	0.0000
6	83	84.4	1.1402
7	62	62	0.0000
8	90	90	0.0000
9	95	95	0.0000
10	106	106.4	0.5477

In Table 1, the nonzero SDV varies in the range [0.5477, 1.1402]. From this table, one interesting finding is that the SDV values are extremely small compared with the magnitude of the mean values. In fact, the coefficient of variation for non zero standard deviation varies in the range [0.005148, 0.0.013509]. Furthermore, the SDV is zero for 8 out of 10 problems. If one looks closer to the final solutions of 20 simulation experiments for these problems, one finds that distinct solutions with the same MS value exist. It means many optima alternatives are there, and the suggested SOSA is able to find them.

The best MS of SMATWAM along with MS of SMATWSM and percentage reduction in MS of former over later for various jobs are given in Table 2.

Table 2. MS of SMATWAM along with SMATWSM and % reduction in MS.

Job set	MS of SMATWAM	MS of SMATWSM	% Reduction in MS
1	53	69	23.19
2	54	80	32.50
3	70	80	12.50
4	54	61	11.48
5	42	48	12.50
6	83	88	5.68
7	62	70	11.43
8	90	131	31.30
9	95	113	15.93
10	106	136	22.06

5.1. Gantt Chart

Below the job set 5 optimal solution vector is given:

4 4 2 2 3 2 3 1 3 1 2 4 1 4 1 4 1 3 3 2 4 2 3 1 1 4 4 5 2 2 3 2 2 5 4 1 2 3 3

The solution above is given as Table 3 in the table form.

Table 3. Job set 5 optimal solution vector.

Job-operation	4-1	2-1	3-1	1-1	1-2	4-2	3-2	2-2	1-3	5-1	3-3	5-2	2-3
Machine number	4	3	1	2	4	1	2	3	4	2	2	4	3
Tool number	2	2	3	4	1	3	4	1	4	2	2	1	3

Figure 3 shows the above solution vector’s Gantt chart. A four-digit number represents an operation. For instance, in operation 4122, the 1st digit ' 4 ' indicates the job number, the 2nd digit ' 1 ' denotes the job-operation, the 3rd digit ' 2 ' indicates the required machine, and the 4th digit ' 1 ' represents tool allocated to the job-operation. The operations assigned for every machine and tool are indicated in the Gantt chart along with each operation's start and end times. In Figure 3, M4, M3, M2, and M1 denote machines, and T4, T3, T2, and T1 indicate tools.

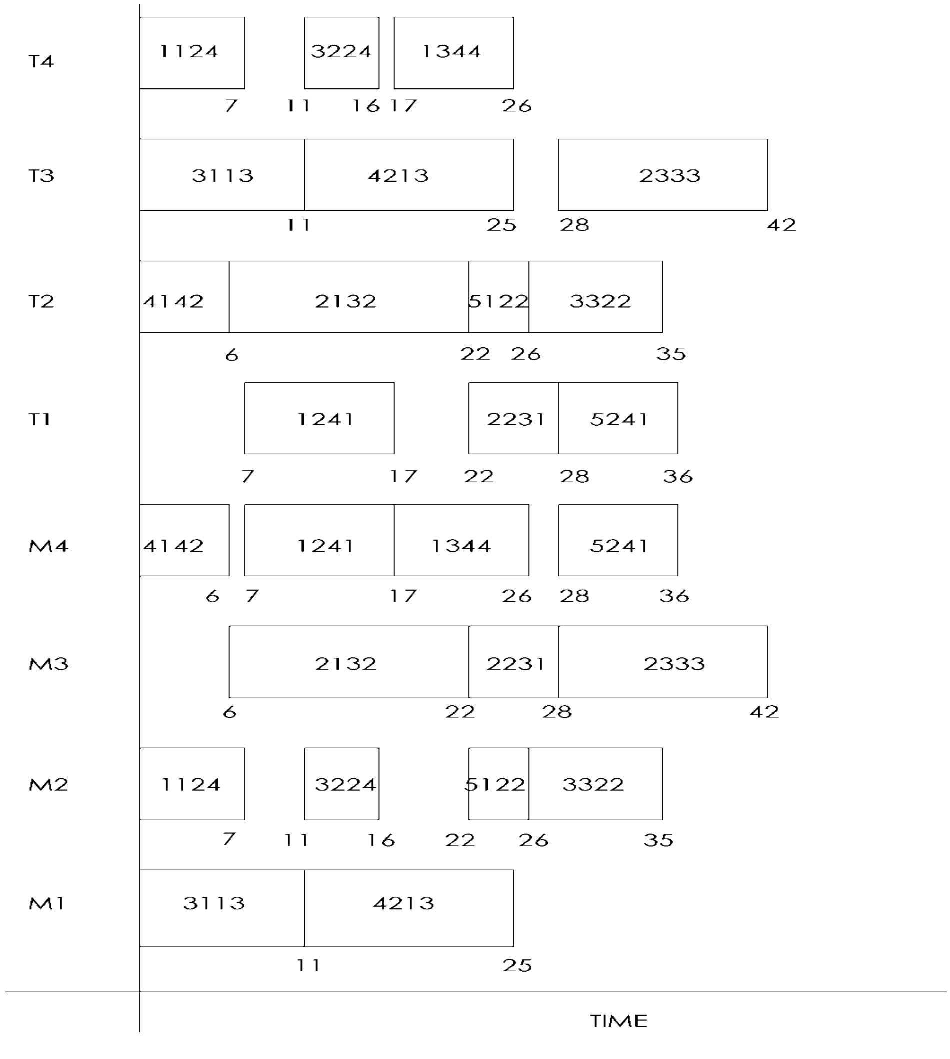

Figure 3. Gantt chart for job set 5.

5.2. Convergence Characterstics

Figure 5 shows SOS algorithm convergence characterstics for job set 4, case I, and lay out 1. The best value is 54, observed at iteration 37, and 0.82125 seconds is time per iteration.

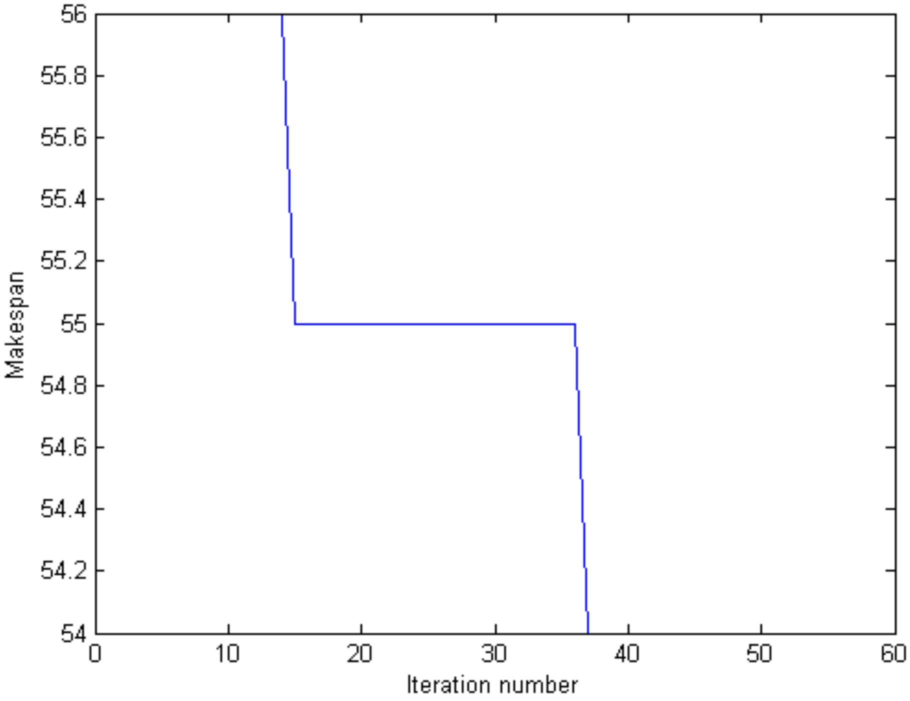

Figure 4. Convergence of SOSA for job set 4.

6. CONCLUSIONS

This paper introduces a nonlinear MIP model for the machines and tools joint scheduling with alternative machines and one copy of every type of tools in an MMFMS for minimum MS. Minimum MS that increases hardware utilization is taken into consideration as appropriate system's performance measure. This work is particularly important as there are no exact approaches for machine and tool simultaneous scheduling in FMS environment with alternative machines. FMS scheduling assumes an important and key role in effective resources utilization. To solve this optimization problem, a metaheuristic optimization algorithm, known as SOSA, is devised as the problem is NP hard. This scheduling problem involves allocation of suitable machine and tool to every job-operation, ordering, and synchronization of those job-operations on machines and restrictions such as limit on the copies of each tool variety for minimum MS. An algorithm for computation of MS is developed for a given schedule, and Figure 2 shows its flow chart. The proposed algorithm is checked on the job sets given in Appendix I. From Table 1, it is quite evident that SOSA is robust and able to find many optimal alternatives for the problems. From Table 2, it is quite evident that there is significant reduction in MS when the alternate machines are considered for job-operation and the percentage reduction in MS varies from 5.68 to 32.50. Reduction in MS reduces the operating cost of machines. In future work, subsystems such as AGVs, robots, and AS / RS and tool change activities induced by tool wear may also be included.

REFERENCES

Agnetis Alessandro , Arianna Alfieri, Paolo Brandimarte & PaoloPrinsecchi. 1997. Joint Job/Tool Scheduling in a Flexible Manufacturing Cell with No On-Board Tool Magazine. *Computer Integrated Manufacturing* System, *10(*1): 61–68. https://doi.org/10.1016/S0951-5240(96)00023-7

Aldrin Raj J, Ravindran D, Saravanan M & Prabaharan T. 2014. Simultaneous scheduling of machines and tools in multimachine flexible manufacturing system using artificial immune system algorithm. *International Journal of Computer Integrated Manufaturing*, 27(5):401–414. https://doi.org/10.1080/0951192X.2013.834461

Baykasoğlu Adil and Fehmi Burcin Ozsoydan (2017) Minimizing tool switching and indexing times with tool duplications in automatic machines, International Journal of Advanced Manufacturing Technology, 89,1775–1789. DOI 10.1007/s00170-016-9194-z

Beezao Andreza Cristina, Jean-Francüois Cordeau, Gilbert Laporte & Horacio Hideki Yanasse. 2017. Scheduling identical parallel machines with tooling constraints, *EuropeanJournal of Operational Research, 257(3) :* 834-844. doi: 10.1016/j.ejor.2016.08.008

Bilge Ümit & Gündüz Ulusoy . 1995. A time window approach to simultaneous scheduling of machines and material handling system in FMS. *Operations Research*, Vol. *43*:1058−1070. https://doi.org/10.1287/opre.43.6.1058

Chandra. P, S. Li & M. Stan . 1993. Jobs and tool sequencing in an automated manufacturing environment, International Journal of Production Research, 31(12): 2911-2925. https://doi.org/10.1080/00207549308956907

Cheng Min-Yuan & Doddy Prayogo. 2014. Symbiotic organisms search: a new metaheurstic optimization algorithm. *Computers and Structures*,139: 98–112.

Costa A. , F.A. Cappadonna & S. Fichera . 2016. Minimizing the total completion time on a parallel machine system with tool changes. *Computers & Industrial Engineering*, 9: 290–301. https://doi.org/10.1016/j.cie.2015.11.015

Ecker K.H. & J.N.D.Gupta. 2005. Scheduling tasks on a flexible manufacturing machine to minimize tool change delays. *European Journal of Operational Research*, *164*: 627–638. https://doi.org/10.1016/j.ejor.2003.12.025

Gökgür Burak ,Brahim Hnich &Selin Özpeynirci. 2018. Parallel machine scheduling with tool loading: a constraintprogramming approach. *International Journal of Production Research*., 56(16): 5541-5557.

Jyothi, S. D. 2012. Scheduling Flexible Manufacturing System Using Petri-nets and Genetic Algorithm. Project Report Indian Institute of Space Science and Technology, Thiruvananthapuram. https://www.semanticscholar.org › paper › SCHEDULING-FLEXIBLE-MAN...

Jun. Hong-Bae , Yeong-Dae Kim & Hyo-Won Suh. 1999. Heuristics for a Tool Provisioning Problem in a Flexible Manufacturing System with an Automatic Tool Transporter. *IEEE Transactions on Robotics and Automation*, 15(3): 488–496. **DOI:** 10.1109/70.768181

Karzan Fatma KilinÇ & Meral Azizoğlu. 2008. The tool transporter movements problem in flexible manufacturing systems. *International Journal of Poduction Research*, 46(11): 3059–3084. https://doi.org/10.1080/00207540601070135

Keung K. W. W. H. Ip &T. C. Lee. 2001. A genetic algorithm approach to the multiple machine tool selection problem. *Journal of Intelligent Manufacturing*, *12:* 331–342. https://doi.org/10.1023/A:1011215416734

Lenstra J.K. & A.H.G.Rinnooy Kan. 1979. Computational complexity of discrete optimization problems. *Ann. Discrete Math.*, *4*: 121–140. https://doi.org/10.1016/S0167-5060(08)70821-5

Özpeynirci Selin. 2015. A heuristic approach based on time-indexed modelling for scheduling and tool loading in flexibl manufacturing systems. *Int J Adv Manuf Technol*, *77*:1269–1274. https://doi.org/10.1007/s00170-014-6564-2

Paiva Gustavo Silva, Marco Antonio M. Carvalho, 2017. Improved Heuristic Algorithms for the Job Sequencing and Tool Switching Problem, *Computers and Operations Research*, 88 :208-219. doi: 10.1016/j.cor.2017.07.013

Prabaharan.T., P. R. Nakkeeran, & N. Jawahar. 2006. Sequencing and Scheduling of Job and Tool in Flexible Manufacturing Cell. *International Journal of Advanced Manufacturing Technology*, 29(3):729–745. https://doi.org/10.1007/s00170-005-2567-3

Roh. H.-K. & Y.-D. Kim. 1997. Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter, International Journal of Production Research, 35(11):2989-3004. https://doi.org/10.1080/002075497194255

S Reddy. Kommuri & N. Sivarami Reddy. 2017. Simultaneous Scheduling of Machines and AGVS in FMS by Using Symbiotic Organisms Search (SOS) Algorithm. *International Journal for Research in Applied Science & Engineering Technology*, 5(11):1780–1790. ISSN: 2321-9653 DOI: 10.22214/ijraset.2017.11258

Saravanan. M. & A. Noorul Haq. 2008. Evaluation of scatter-search approach for scheduling optimization of flexible manufacturing systems. *Int J Adv Manuf Technol*, *38*: 978–986. https://doi.org/10.1007/s00170-007-1134-5

Satish Kumar. M. V , Ranga Janardhana and C. S. P. Rao (2011). Simultaneous scheduling of machines and vehicles in an FMS environment with alternative routing. Int J Adv Manuf Technol 53, 339–351. https://doi.org/10.1007/s00170-010-2820-2

Sivarami Reddy Narapu Reddy, Dwivedula Venkata Ramamurthy, K. Prahlada Rao, M.Padma Lalitha. (2016). A Novel metaheuristic method for simultaneous scheduling of machines and tools in multi machine FMS. *IET Digital Library International Conference on Recent Trends in Engineering, Science & Technology 2016*, 1–6. ISBN-978-1-78561-785-0 **DOI:** 10.1049/cp.2016.1489

Sivarami Reddy. N, Dr.D.V.Ramamurthy, Dr. K. Prahlada Rao. and Dr. M. Padma Lalitha. (2017a). Simultaneous Scheduling of Machines and Tools in Multi Machine FMS Using Crow Search Algorithm. *International Journal of Engineering Science and Technology*, *9*(09S), 66–73.

Sivarami Reddy N., Dr. D.V. Ramamurthy and Dr. K .Prahlada Rao (2017b). Simultaneous Scheduling of Machines and Tools to minimise makespan in Multi Machine FMS using new nature inspired algorithms *Manufacturing Technology Today, 16(3),* 19-27

Sivarami Reddy N., D.V.Ramamurthy, K.Prahlada Rao. (2018a). "Simultaneous Scheduling of Machines and Tools Considering Tool Transfer Times in Multimachine FMS Using CSA." *International Conference on Intelligent Computing and Applications, Advances in Intelligent Systems and Computing 632*: 421–432.

Sivarami Redd, N, Ramamurthy, D.V. and Prahlada Rao, K, (2018b). Simultaneous scheduling of jobs, machines and tools considering tool transfer times in multi-machine FMS using new nature-inspired algorithms. *Int. J. Intelligent Systems Technologies and Applications*, *17*(1/2), 70–88.

Sivarami Reddy. N, D V. Ramamurthy, K Prahlada Rao. & M. Padma Lalitha. 2019. Integrated scheduling of machines, AGVs and tools in multi-machine FMS using crow search algorithm, International Journal of Computer Integrated Manufacturing, 32(11): 1117-1133. DOI:10.1080/0951192X.2019.1686171https://doi.org/10.1080/0951192X.2019.1686171

Sivarami Reddy N., D. V. Ramamurthy , K. Prahlada Rao & M. Padma Lalitha (2021). Practical simultaneous scheduling of machines, AGVs, tool transporter and tools in a multi machine FMS using symbiotic organisms search algorithm, International Journal of Computer Integrated Manufacturing, DOI: 10.1080/0951192X.2020.1858503 https://doi.org/10.1080/0951192X.2020.1858503

Song. Chang-Yong, Hark Hwang & Yeong-Daekim. 1995. Heuristic algorithm for the tool movment policy in Flexible Manufacturing systems. *Journal of Manufacturing Systems*, 14(3): 160-168. https://doi.org/10.1016/0278-6125(95)98884-9

Sonmez Yusuf, H. Tolga Kahraman, M. Kenan Dosoglu, Ugur Guvenc & Serhat Duman. 2017. Symbiotic organisms search algorithm for dynamic economic dispatch with valve-point effects, Journal of Experimental & Theoretical Artificial Intelligence,29(3): 495-515 . DOI:10.1080/0952813X.2016.1198935

Sureshkumar, N. & R. Sridharan. 2009. Simulation Modeling and Analysis of Tool Flow Control Decisions in a Flexible Manufacturing System. *Robotics and Computer Integrated Manufacturing*, *25*: 829–838. https://doi.org/10.1016/j.rcim.2008.12.004

Tang. Christopher S. & Eric V. Denardo. 1988. Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches. *Operations Research*, 36(5): 767–777. https://doi.org/10.1287/opre.36.5.767

Tsukada. T.K. & Kang G. Shin. 1998. Distributed Tool Sharing in Flexible Manufacturing Systems *IEEE Transactions on Robotics and Automation*, 14(3):.379–389. DOI**:** 10.1109/70.678448

Udhayakumar P. & S. Kumanan . 2010. Sequencing and scheduling of job and tool in a flexible manufacturing system using ant colony optimization algorithm. *Int J Adv Manuf Technol*, *50*:1075–1084. https://doi.org/10.1007/s00170-010-2583-9

Udhayakumar P. & S. Kumanan . 2012. Some metaheuristic approaches for optimising tardiness of job and tool in a flexible manufacturing system. *Int. J. Advanced Operations Management*, 4(3): 219-252. https://doi.org/10.1504/IJAOM.2012.047683

Wu Haizhou, Yongquan Zhou,Qifang Luo & Mohamed Abdel Basset. 2016. Training Feedforward Neural Networks Using Symbiotic Organisms Search Algorithm, Computational Intelligence and Neuroscience,2016: 1-14. http://dx.doi.org/10.1155/2016/9063065

APPENDIX I

Job set 1	Job set 2
Job 1: Mc1-Tl3 (8); Mc2-Tl4 (16); Mc4-Tl1 (12)	**Job 1: Mc1-Tl3 (10); Mc4-Tl1 (18)**
ALTl1: Mc2-Tl3 (9) Mc3-Tl4 (14) Mc1-Tl1 (13)	ALTl1: Mc3-Tl3 (12) Mc2-Tl1 (16)
ALTl2: Mc3-Tl3 (9) Mc4-Tl4 (17) Mc2-Tl1 (10)	ALTl2: Mc2-Tl3 (11) Mc3-Tl1 (17)
Job 2: Mc1-Tl2 (20); Mc3-Tl3 (10); Mc2-Tl1 (18)	**Job 2: Mc2-Tl2 (10); Mc4-Tl4 (18)**
ALTl1: Mc3-Tl2 (18) Mc1-Tl3 (13) Mc4-Tl1 (17)	ALTl1: Mc4-Tl2 (8); Mc1-Tl4 (20)

ALTl2: Mc2-Tl2 (21) Mc4-Tl3 (8) Mc3-Tl1 (19)	ALTl2: Mc3-Tl2(12); Mc2-Tl4 (16)
Job 3: Mc3-Tl1 (12); Mc4-Tl4 (8); Mc1-Tl2 (15)	**Job 3: Mc1-Tl4 (10); Mc3-Tl2 (20)**
ALTl1: Mc4-Tl1 (11) Mc2-Tl4 (10) Mc3-Tl2 (14)	ALTl1: Mc3-Tl4 (12); Mc2-Tl2 (18)
ALTl2: Mc1-Tl1 (11) Mc3-Tl4 (7) Mc2-Tl2 (17)	ALTl2: Mc2-Tl4 (8); Mc4-Tl2 (22)
Job 4: Mc4-Tl3 (14); Mc2-Tl4 (18)	**Job 4: Mc2-Tl3 (10); Mc3-Tl1 (15); Mc4-Tl2 (12)**
ALTl1: Mc1-Tl3 (16) Mc3-Tl4 (16)	ALTl1: Mc4-Tl3 (11) Mc1-Tl1 (17); Mc2-Tl2 (9)
ALTl2: Mc3-Tl3 (16) Mc1-Tl4 (16)	ALTl2: Mc1-Tl3 (13) Mc4-Tl1 (13); Mc3-Tl2 (11)
Job 5: Mc3-Tl2 (10); Mc1-Tl1 (15)	**Job 5: Mc1-Tl4 (10); Mc2-Tl1 (15); Mc4-Tl3 (12)**
ALTl1: Mc2-Tl2 (9) Mc4-Tl1 (16)	ALTl1: Mc4-Tl4 (12) Mc1-Tl1 (14); Mc3-Tl3 (11)
ALTl2: Mc4-Tl2 (11) Mc3-Tl1 (14)	ALTl2: Mc2-Tl4 (8) Mc3-Tl1 (14); Mc1-Tl3 (15)
Job set 3	**Job 6: Mc1-Tl4 (10); Mc2-Tl2 (15); Mc3-Tl3 (12)**
Job 1: Mc1-Tl4 (16); Mc3-Tl2 (15)	ALTl1: Mc3-Tl4 (11) Mc4-Tl2 (13) Mc2-Tl3 (13)
ALTl1: Mc3-Tl4 (14) Mc1-Tl2 (17)	ALTl2: Mc2-Tl4 (12) Mc3-Tl2 (14) Mc4-Tl3 (11)
ALTl2: Mc2-Tl4 (15) Mc4-Tl2 (16)	**Job set 4**
Job 2: Mc2-Tl1 (18); Mc4-Tl3 (15)	**Job 1: Mc4-Tl2 (11); Mc1-Tl3 (10); Mc2-Tl1 (7)**
ALTl1: Mc1-Tl1 (17) Mc3-Tl3 (16)	ALTl1: Mc3-Tl2(10) Mc2-Tl3(9) Mc4-Tl1(9)
ALTl2: Mc3-Tl1 (16) Mc2-Tl3 (17)	ALTl2: Mc2-Tl2(13) Mc4-Tl3(9) Mc3-Tl1(6)
Job 3: Mc1-Tl3 (10); Mc2-Tl1 (10)	**Job 2: Mc3-Tl1 (12); Mc2-Tl2 (10); Mc4-Tl4 (8)**
ALTl1: Mc3-Tl3 (19) Mc4-Tl1 (11)	ALTl1: Mc1-Tl1(10) Mc4-Tl2(11) Mc3-Tl4(9)
ALTl2: Mc4-Tl3 (19) Mc1-Tl1 (11)	ALTl2: Mc4-Tl1(11) Mc3-Tl2(9) Mc2-Tl4(10)
Job 4: Mc3-Tl3 (15); Mc4-Tl2 (10)	**Job 3: Mc2-Tl2 (7); Mc3-Tl4 (10); Mc1-Tl3 (9); Mc3-Tl2 (8)**
ALTl1: Mc4-Tl3 (17) Mc2-Tl2 (8)	ALTl1: Mc4-Tl2(10) Mc2-Tl4(9) Mc3-Tl3(8) Mc1-Tl2(7)

ALTl2: Mc1-Tl3 (17) Mc3-Tl2 (8)	ALTl2: Mc3-Tl2(9) Mc1-Tl4(11) Mc4-Tl3(7) Mc2-Tl2(7)
Job 5: Mc1-Tl4 (8); Mc2-Tl2 (10); Mc3-Tl3 (15); Mc4-Tl4 (17)	**Job 4: Mc2-Tl2 (7); Mc4-Tl3 (8); Mc1-Tl1 (12); Mc2-Tl2 (6)**
ALTl1: Mc3-Tl4(10) Mc4-Tl2(12) Mc2-Tl3(13)Mc1-Tl4(15)	ALTl1: Mc3-Tl2(9) Mc2-Tl3(6) Mc4-Tl1(10) Mc1-Tl2(8)
ALTl2: Mc2-Tl4(9) Mc1-Tl2(8) Mc4-Tl3(13)Mc3-Tl4(17)	ALTl2: Mc1-Tl2(8) Mc3-Tl3(9) Mc2-Tl1(9) Mc4-Tl2(6)
Job 6: Mc2-Tl3 (10); Mc3-Tl1 (15); Mc4-Tl2 (8); Mc1-Tl4 (15)	**Job 5: Mc1-Tl1 (9); Mc2-Tl3 (7); Mc4-Tl4 (8); Mc2-Tl2 (10);Mc3-Tl3 (8)**
ALTl1: Mc1-Tl3(8); Mc4-Tl1(14) ;Mc2-Tl2(10);Mc3-Tl4(16)	ALTl1: Mc3-Tl1(8) Mc4-Tl3(8) Mc3-Tl4(10) Mc1-Tl2(9)Mc2-Tl3(7)
ALTl2: Mc1-Tl3(9); Mc2-Tl1(16) ;Mc3-Tl2(9); Mc4-Tl4(14)	ALTl2: Mc2-Tl1(8) Mc3-Tl3(8) Mc1-Tl4(9) Mc3-Tl2(8)Mc4-Tl3(9)
Job set 5	**Job set 6**
Job 1: Mc1-Tl4 (6); Mc2-Tl1 (12); Mc4-Tl4 (9)	**Job 1: Mc1-Tl1 (9); Mc2-Tl2 (11); Mc4-Tl4 (7)**
ALTl1: Mc3-Tl4(7) Mc4-Tl1 (10) Mc2-Tl4 (10)	ALTl1: Mc3-Tl1 (10) Mc1-Tl2 (9) Mc2-Tl4 (8)
ALTl2: Mc2-Tl4(7) Mc3-Tl1 (10) Mc1-Tl4 (10)	ALTl2: Mc4-Tl1 (11) Mc3-Tl2 (10) Mc1-Tl4(6)
Job 2: Mc1-Tl2 (18); Mc3-Tl1 (6); Mc2-Tl3 (15)	**Job 2: Mc1-Tl1 (19); Mc2-Tl2 (20); Mc4-Tl4 (13)**
ALTl1: Mc3-Tl2 (16) Mc2-Tl1 (7) Mc1-Tl3 (16)	ALTl1: Mc4-Tl1 (20) Mc3-Tl2 (18) Mc2-Tl4 (14)
ALTl2: Mc2-Tl2 (17) Mc4-Tl1 (8) Mc3-Tl3 (14)	ALTl2: Mc2-Tl1 (18) Mc3-Tl2 (22) Mc1-Tl4 (12)
Job 3: Mc3-Tl3 (9); Mc4-Tl4 (3); Mc1-Tl2 (12)	**Job 3: Mc2-Tl2 (14); Mc3-Tl3 (20); Mc4-Tl4 (9)**
ALTl1: Mc1-Tl3 (11) Mc3-Tl4 (4) Mc2-Tl2 (9)	ALTl1: Mc3-Tl2 (12) Mc4-Tl3 (21) Mc1-Tl4 (10)
ALTl2: Mc4-Tl3 (10) Mc2-Tl4 (5) Mc3-Tl2 (11)	ALTl2: Mc4-Tl2 (15) Mc1-Tl3 (21) Mc2-Tl4 (7)
Job 4: Mc4-Tl2 (6); Mc2-Tl3 (15)	**Job 4: Mc2-Tl2 (14); Mc3-Tl3 (20); Mc4-Tl4 (9)**
ALTl1: Mc2-Tl2 (8) Mc3-Tl3 (13)	ALTl1: Mc4-Tl2 (13) Mc1-Tl3 (19) Mc3-Tl4 (11)
ALTl2: Mc3-Tl2 (7) Mc1-Tl3 (14)	ALTl2: Mc3-Tl2 (12) Mc4-Tl3 (19) Mc1-Tl4 (12)
Job 5: Mc3-Tl2 (3); Mc1-Tl1 (9)	**Job 5: Mc1-Tl1 (11); Mc3-Tl3 (16); Mc4-Tl4 (8)**

ALTl1: Mc4-Tl2 (5) Mc2-Tl1 (7)	ALTl1: Mc2-Tl1 (12) Mc1-Tl3 (14) Mc3-Tl4 (9)
ALTl2: Mc2-Tl2 (4) Mc4-Tl1 (8)	ALTl2: Mc3-Tl1 (12) Mc4-Tl3 (13) Mc2-Tl4 (10)
Job set 7	**Job 6: Mc1-Tl1 (10); Mc3-Tl3 (12); Mc4-Tl4 (10)**
Job 1: Mc1-Tl1 (6); Mc4-Tl4 (6)	ALTl1: Mc3-Tl1(9) Mc4-Tl3 (11) Mc2-Tl4 (12)
ALTl1: Mc3-Tl1 (7) Mc2-Tl4 (5)	ALTl2: Mc2-Tl1 (9) Mc1-Tl3 (11) Mc3-Tl4 (12)
ALTl2: Mc4-Tl1 (5) Mc3-Tl4 (7)	**Job set 8**
Job 2: Mc2-Tl2 (11); Mc4-Tl3 (9)	**Job 1: Mc2-Tl1 (12); Mc3-Tl2 (21); Mc4-Tl3 (11)**
ALTl1: Mc4-Tl2 (10) Mc3-Tl3 (10)	ALTl1: Mc1-Tl1 (13) Mc4-Tl2 (19) Mc3-Tl3 (12)
ALTl2: Mc3-Tl2 (12) Mc1-Tl3 (8)	ALTl2: Mc3-Tl1 (14) Mc2-Tl2 (18) Mc1-Tl3 (12)
Job 3: Mc2-Tl2 (9); Mc4-Tl4 (7)	**Job 2: Mc2-Tl4 (12); Mc3-Tl3 (21); Mc4-Tl2 (11)**
ALTl1: Mc3-Tl2 (10) Mc1-Tl4 (6)	ALTl2: Mc1-Tl4 (13) Mc4-Tl3 (19) Mc3-Tl2 (12)
ALTl2: Mc4-Tl2 (8) Mc3-Tl4 (8)	ALTl1: Mc3-Tl4 (14) Mc2-Tl3 (20) Mc1(10)
Job 4: Mc3-Tl3 (16); Mc4-Tl4 (7)	ALTl2: Mc1(13) Mc4(19) Mc3(12)
ALTl1: Mc2-Tl3 (14) Mc3-Tl4 (9)	**Job 3: Mc2-Tl2 (12); Mc3-Tl3 (21); Mc4-Tl4 (11)**
ALTl2: Mc1-Tl3 (17) Mc2-Tl4 (6)	ALTl1: Mc4-Tl2 (13) Mc1-Tl3 (22) Mc3-Tl4 (9)
Job 5: Mc1-Tl1 (9); Mc3-Tl3 (18)	ALTl2: Mc1-Tl2(10) Mc4-Tl3 (22) Mc2-Tl4 (12)
ALTl1: Mc4-Tl1 (11) Mc2-Tl3 (20)	**Job 4: Mc2-Tl1 (12); Mc3-Tl2 (21); Mc4-Tl3 (11)**
ALTl2: Mc2-Tl1 (10) Mc4-Tl3 (17)	ALTl1: Mc3-Tl1 (11) Mc1-Tl2 (23) Mc2-Tl3 (10)
Job 6: Mc2-Tl1 (13); Mc3-Tl2 (19); Mc4-Tl3 (6)	ALTl2: Mc4-Tl1 (13) Mc2-Tl2 (22) Mc3-Tl3 (9)
ALTl1: Mc3-Tl1 (12) Mc4-Tl2 (18) Mc1-Tl3 (8)	**Job 5: Mc1-Tl4 (10); Mc2-Tl1 (14); Mc3-Tl2 (18); Mc4-Tl3 (9)**
ALTl2: Mc1-Tl1 (14) Mc2-Tl2 (17) Mc3-Tl3 (7)	ALTl1: Mc4-Tl4 (13) Mc3-Tl1 (13) Mc1-Tl2 (17)Mc2-Tl3 (8)
Job 7: Mc1-Tl3 (10); Mc2-Tl4 (9); Mc3-Tl1 (13)	ALTl2: Mc3-Tl4 (11) Mc4-Tl1 (13) Mc2-Tl2 (16)Mc1-Tl3 (11)
ALTl1: Mc3-Tl3 (11) Mc1-Tl4 (10) Mc2-Tl1 (11)	**Job 6: Mc1-Tl2 (10); Mc2-Tl3 (14); Mc3-Tl (18); Mc4-Tl1 (9)**
ALTl2: Mc4-Tl3 (11) Mc3-Tl4 (10) Mc1-Tl1 (11)	ALTl1: Mc2-Tl2 (12) Mc4-Tl3 (13) Mc1-Tl (16)Mc3-Tl1 (10)

Job 8: Mc1-Tl4 (11); Mc2-Tl2 (9); Mc4-Tl3 (8)	ALTl2: Mc4-Tl2 (9) Mc3-Tl3 (15) Mc2-Tl (17)Mc1-Tl1 (10)
ALTl1: Mc2-Tl4 (9) Mc3-Tl2 (10) Mc1-Tl3 (9)	**Job set 10**
ALTl2: Mc4-Tl4 (10) Mc1-Tl2 (11) Mc3-Tl3 (7)	**Job 1: Mc1-Tl4 (11); Mc3-Tl3 (19); Mc2-Tl2 (16); Mc4-Tl1 (13)**
Job set 9	ALTl1: Mc2-Tl4 (13) Mc4-Tl3 (17) Mc3-Tl2 (14)Mc1-Tl1 (15)
Job 1: Mc3-Tl4 (9); Mc1-Tl2 (12); Mc2-Tl1 (9); Mc4-Tl3 (6)	ALTl2: Mc4-Tl4 (13) Mc2-Tl3 (18) Mc1-Tl2 (14)Mc3-Tl1 (14)
ALTl1: Mc4(10) Mc3(9) Mc1(10)Mc2(7)	**Job 2: Mc2-Tl3 (21); Mc3-Tl1 (16); Mc4-Tl2 (14)**
ALTl2: Mc2(8) Mc4(15) Mc3(8)Mc1(5)	ALTl1: Mc3-Tl3 (19) Mc4-Tl1 (17) Mc1-Tl2 (15)
Job 2: Mc3-Tl4 (16); Mc2-Tl2 (11); Mc4-Tl3 (9)	ALTl2: Mc1-Tl3 (20) Mc4-Tl1 (18) Mc3-Tl2 (13)
ALTl1: Mc1-Tl4 (14) Mc3-Tl2 (12) Mc2-Tl3 (10)	**Job 3: Mc3-Tl4 (8); Mc2-Tl2 (10); Mc1-Tl3 (14); Mc4-Tl2 (9)**
ALTl2: Mc2-Tl4 (18) Mc4-Tl2 (10) Mc3-Tl3 (8)	ALTl1: Mc2-Tl4(10); Mc4-Tl2 (8) ; Mc3-Tl3 (16); Mc1-Tl2 (7)
Job 3: Mc1-Tl4 (21); Mc2-Tl2 (18); Mc4-Tl1 (7)	ALTl2: Mc1-Tl4(10); Mc4-Tl2(8) ; Mc2-Tl3 (15); Mc3-Tl2(8)
ALTl1: Mc3-Tl4 (19) Mc4-Tl2 (19) Mc1-Tl1 (8)	**Job 4: Mc2-Tl3 (13); Mc3-Tl2 (20); Mc4-Tl4 (10)**
ALTl2: Mc4-Tl4 (22) Mc1-Tl2 (16) Mc2-Tl1 (8)	ALTl1: Mc1-Tl3 (15) Mc4-Tl2 (17) Mc3-Tl4 (11)
Job 4: Mc2-Tl4 (20); Mc3-Tl3 (22); Mc4-Tl2 (11)	ALTl2: Mc3-Tl3 (11) Mc1-Tl2 (20) Mc2-Tl4 (12)
ALTl1: Mc4-Tl4 (21) Mc1-Tl3 (20) Mc3-Tl2 (12)	**Job 5: Mc1-Tl4 (9); Mc3-Tl3 (16); Mc4-Tl1 (18)**
ALTl2: Mc3-Tl4 (22) Mc2-Tl3 (21) Mc1-Tl2 (10)	ALTl1: Mc3-Tl4 (12) Mc4-Tl3 (14) Mc1-Tl1 (17)
Job 5: Mc3-Tl4 (14); Mc1-Tl2 (16); Mc2-Tl1 (13); Mc4-Tl3 (9)	ALTl2: Mc4-Tl4 (10) Mc1-Tl3 (17) Mc2-Tl1 (16)
ALTl1: Mc2-Tl4 (15) Mc4-Tl2 (13) Mc1-Tl1 (14) Mc3-Tl3 (10)	**Job 6: Mc2-Tl4(19); Mc1-Tl3(21); Mc3-Tl1(11);Mc4-Tl2 (15)**
ALTl2: Mc1-Tl4 (16) Mc3-Tl2 (14) Mc4-Tl1 (11) Mc2-Tl3 (11)	ALTl1: Mc1-Tl4(16); Mc3-Tl3(22); Mc4-Tl1(14);Mc2-Tl2 (14)
	ALTl2: Mc4-Tl4(17); Mc2-Tl3(19); Mc1-Tl1(13);Mc3-Tl2 (17)