



Integrated scheduling and tool management in flexible manufacturing systems

M. Selim Akturk & Serkan Ozkan

To cite this article: M. Selim Akturk & Serkan Ozkan (2001) Integrated scheduling and tool management in flexible manufacturing systems, International Journal of Production Research, 39:12, 2697-2722, DOI: [10.1080/00207540110051941](https://doi.org/10.1080/00207540110051941)

To link to this article: <http://dx.doi.org/10.1080/00207540110051941>



Published online: 14 Nov 2010.



Submit your article to this journal [↗](#)



Article views: 99



View related articles [↗](#)



Citing articles: 21 View citing articles [↗](#)



Integrated scheduling and tool management in flexible manufacturing systems

M. SELIM AKTURK†* and SERKAN OZKAN†

A multistage algorithm is proposed that will solve the scheduling problem in a flexible manufacturing system by considering the interrelated subproblems of processing time control, tool allocation and machining conditions optimization. The main objective of the proposed algorithm is to minimize total production cost consisting of tooling, operational and tardiness costs. The proposed integrated approach recognizes an important trade-off in automated manufacturing systems that has been largely unrecognized, and which is believed can be effectively exploited to improve production efficiency and lead to substantial cost reductions.

1. Introduction

Manufacturing companies must rely on innovative developments in manufacturing technology to compete in today's world market. As a result of the progress in manufacturing technology and organization, the concept of flexible manufacturing systems (FMS) has emerged. The efficient operation of an FMS is a very difficult task, and in many implementations the available capacity is underutilized. In view of the high investment and operating costs of FMS, attention should be paid to their effective utilization. Their efficiency is, however, directly related to their design and operational strategies. Tool management is the most dynamic and critical facility in FMS and requires keen attention. Gray *et al.* (1993) and Veeramani *et al.* (1992) emphasize that lack of proper attention to cutting tool-related issues can prevent an FMS from reaching its fullest potential and can make it 'inflexible' in practice, since tool management systems affect product design options, machine loading, job batching, capacity scheduling and real-time part routing decisions. Hence, there is a growing need to integrate tool management more thoroughly into system design, planning and control, with increasing automation in manufacturing systems.

Proposed is a multistage algorithm that will solve the scheduling, tool allocation and machining conditions optimization problems by exploiting the interactions among these interrelated problems to minimize total production cost consisting of tooling, operational and tardiness costs in an FMS. Existing studies solve these problems independently at the different levels in the decision-making hierarchy. For example, in discrete parts manufacture, the way in which parts are processed by machines is calculated by finding the economically optimum process parameters for that part in isolation. Once calculated, processing and set-up time data are passed up to the system-planning level, in which decisions such as batch sizes and schedules are determined from the timing data along with system-level objective functions. In reality however,

Revision received July 2000.

†Department of Industrial Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey.

* To whom correspondence should be addressed. e-mail: akturk@bilkent.edu.tr

the time it takes to process each part is a controllable variable. It is certainly clear that the existing decomposition is suboptimal. Since it is well known that scheduling problems are extremely sensitive to processing time data, it seems that by selecting processing times appropriately, system resources can be utilized much more efficiently.

Gray *et al.* (1993) proposed an integrated conceptual framework for resource planning to examine how tool management issues, depending upon their scope, can be classified into system, machine and tool levels. For solving tool allocation problems at the system level, most of the existing studies use 0–1 binary variables to represent tool requirements. Sarin and Chen (1987) gave an MIP formulation under the assumption that the total-machining costs depend upon the tool–machine combination. Tool life is considered as a constraint in the model. The key tool management issues at the single machine level are loading and placing a set of tools in the machine's magazine, determining the part input sequence to meet certain magazine constraints and establishing tool replacement strategies. At the machine level, the existing studies, such as Kouvelis (1991) and Tang and Denardo (1988), minimize the tool switches due to a change in the part mix. Crama and Kluvert (1999) studied the complexity of tool management problems approximately by investigating the worst-case ratios of some of the polynomial–time approximation algorithms in the literature for solving single-machine tool-management problems. These studies assume constant processing times and tool lives, even though the tool-replacement frequency is directly related with the machining conditions selections. Further, in the multiple operation case, non-machining time components, such as the tool replacement due to tool wear, can have a significant impact on the total cost of production and the throughput of parts as shown by Tetzlaff (1996). Schweitzer and Seidmann (1991) and Schweitzer *et al.* (1991) present several non-linear queueing network optimization methodologies that determine the minimum cost processing rates given the throughput target, the work-in-process level, part routes, transport delays and tool cost functions. Lamond and Sodhi (1997) considered minimization of processing times on a flexible machine using tool life models without considering the tool-sharing opportunities between the parts. An overview of tool management approaches can be found in Crama (1997). Tool-management issues include the number and type of tools, and tool cutting speeds and feed rates at the tool level. These factors determine the quality of the parts produced and the effective capacity of the machines. These are critical choices in automated manufacturing because of the level of integration required between the various production functions. Machining conditions optimization for a single operation is a well-known problem, and several models and solution procedures have been developed as described in Hitomi (1989). However, these models consider only the contribution of machining time and tooling cost to the total cost of operation, usually ignoring the tool availability limitations and the contribution of non-machining time components to the operating cost, which could be very significant for the multiple operation case.

Scheduling problems are usually solved using fixed and predetermined processing time data passed from the machine level in the decision hierarchy. This approach ignores the interactions between scheduling and tool management decisions, hence a decision made at a higher level without considering its impact on the lower levels can lead to inferior or even infeasible results when we consider both constraints and parameters of the lower-level problems. In the literature of scheduling with controllable processing times, most of the studies assume that processing times have their own associated linearly varying costs, such as Vickson (1980) and van Wassenhove

and Baker (1982). Nowicki and Zdrzalka (1990) provide a summary of the existing results in this area.

In traditional tool-management approaches, the tool requirements for each operation are determined independently at the system level without considering the tool and machine level issues, such as tool sharing, loading of duplicate tools, alternative tooling possibilities, and the contention among the operations for a limited number of tools. Furthermore, the close relationship between the processing times and tool lives is ignored, although this relation might have a significant impact on system performance. All of the studies assume that processing times are known beforehand regardless of the machining conditions, although the processing times are controllable decision variables with their associated non-linear convex cost functions. The remainder of this paper is organized as follows. The next section defines the scope of the study with the underlying assumptions and the notation used throughout. Section 3 presents the proposed algorithm, while the computational results are discussed in Section 4. The proposed algorithm is applied on an example problem in Section 5. Finally, some concluding remarks are provided in Section 6.

2. Problem statement

In this study, it is assumed that there are multiple part types with different batch sizes, and each one has a distinct due date and a different weighting factor. For each individual part, there are multiple operations to be performed. Each operation corresponds to a removal of a predefined machinable volume, as discussed in Akturk and Avci (1996). For each operation, although there are alternative tool types with limited quantities on hand to perform the given metal-cutting operation, it is evident that only one cutting tool can be used at a time to accomplish this operation. Advances in cutting tool materials and designs will increase the cutting speeds at which machining is carried out, consequently reducing the machining time, but the initial tooling cost might be higher. Therefore, we consider a set of alternative cutting tool types for each machining operation such as HSS, carbides, coated tools, since no one cutting tool type is best for all purposes. As discussed above, tooling costs have a significant impact on both the fixed and variable cost of production. Therefore, in practice, there are only limited quantities on hand for each tool type to minimize tooling inventories.

For the operations, the cutting speed and feed rate will be taken as decision variables, and depth of cut, length and surface finish requirements are assumed to be given as input. Tool replacement is only allowed during the part changing and only a single tool can be changed at a time. This implies that tool-changing times are additive. There are multiple identical CNC machines with limited tool magazine capacities, and each machine can load/unload tools automatically. Each machine can work for a limited period. Besides the on-board tool magazines at each machine, there is also a central tool storage where the tools not assigned to any machine are kept. A robotic manipulator is used to transfer tools between the central storage and the machines. This configuration is similar to the FMS implementations discussed in Macchiaroli and Riemma (1996) and Mukhopadhyay and Sahu (1996).

Defining the scope of the present study, we wish to solve tool management and scheduling problems simultaneously. We will determine the tool management decisions consisting of tool allocation, i.e. how tools will be allocated to part types in terms of quantities and allocation scheme, and machining conditions selection, i.e. what the cutting speed and feed rate will be for each operation of each part, and scheduling decisions, i.e. which parts will be processed on which machine at what

time. The objective is to minimize the total production cost, which is comprised of tooling, operational and tardiness costs. After completion of a lot, remaining tool lives can be used for manufacturing of another lot. Thus, the actual usage of tools is included in the tooling cost and tool availability related constraints. The operational cost is the cost of operating the system. The tardiness cost is the weighted sum of tardiness of all parts, where tardiness of a part is either zero (in case it is completed before its due date) or otherwise it equal to the difference of its completion time and its due date. The final solution will satisfy both the tool management and scheduling constraints such that each operation is assigned to a single tool type from its candidate tools set, tool requirements do not exceed the amount of tools on hand, total time required to manufacture the parts on a machine does not exceed available machine hour capacity, and a machine can process at most one part at a time, i.e. there are non-interference constraints between the parts.

The notation used throughout is given below. In order to simplify the notation, we used a single subscript j for the cutting tool related parameters, such as C_j , α_j , etc., as if all parts have the same material composition, although a second index p can be easily added to each tool-related parameter to indicate the part material.

The parameters are:

- $\alpha_j, \beta_j, \gamma_j$ speed, feed, depth of cut exponents for tool j ,
- C_m, b, c, e specific coefficient and exponents of the machine power constraint,
- C_o operating cost of the CNC machine (\$/min),
- C_s, g, h, l specific coefficient and exponents of the surface roughness constraint,
- C_{t_j} cost of tool j (\$/tool),
- C_j Taylor's tool life expression parameter for tool j ,
- d_{pi} depth of cut for operation i of part p (inches.),
- D_{pi}, L_{pi} diameter and length of the generated surface for operation i of part p (inches),
- HP maximum available machine power (hp),
- SF_{pi} maximum allowable surface roughness for the operation i of part p (μ in),
- P, I_p, J set of all part types, all operations of part p and the tool types, respectively,
- Q_p batch size of part type p ,
- N_j number of available tools of type j ,
- w_p weight of part type p ,
- DD_p due date of part type p .

The decision variables are:

- n_{pij} number of tool type j required for completion of operation i of part type p ,
- v_{pij} cutting speed for operation i of part p using tool j (fpm),
- f_{pij} feed rate for operation i of part p using tool j (ipr),
- U_{pij} usage rate of tool j in the operation i of part type p ,
- r_{pij} number of parts that can be manufactured for operation i of part type p by tool j ,
- $t_{m_{pij}}$ machining time of operation i of part p using tool j ,
- R_j total tool requirement of tool type j ,
- t_{m_p} total machining time of part type p ,
- t_{s_p} total expected set-up time of part type p .

3. Proposed Algorithm

The constraints and the decision variables for machining conditions, tool allocation and scheduling problems interact with each other. In order to solve these inter-related problems simultaneously, a three-level resource-directed decomposition procedure is proposed by relaxing the scheduling-related constraints first, which can be called coupling constraints among the parts. For the reduced problem, we find the optimum machining conditions for all possible operation–tool pairs and select the tool that gives the minimum cost by solving the single-machine operation problem (SMOP) after relaxing the set of tool availability constraints in the first level. This will provide a lower bound for the tool allocation and machining conditions optimization problem. Later on, we impose the relaxed tool availability constraints and solve an integer programming (IP) formulation if any tool availability constraint is violated. In the second level, we find an initial schedule that minimizes the total production cost subject to the non-interference, precedence and state-dependent set-up time constraints for a given tool management decisions. Finally, we look for reduction possibilities in the processing times of the operations in order to make further improvements in the total production cost in the third level. These levels will be explained in detail below and will be presented in an example problem in Section 5.

3.1. Tool Allocation

In this level, a very efficient algorithm is proposed to find the optimum machining conditions and corresponding tool allocations for all operations that minimize the total manufacturing cost for a given set of constraints. These allocations most probably will not give the minimum processing time for each operation for the same feasible region, hence it may not correspond to the minimum production cost. Sometimes a smaller total production cost is obtained by increasing the production rate resulting from reducing unit processing times and sacrificing unit manufacturing costs. Hence, we also develop closed form expressions for the efficient frontier of the manufacturing cost and time interactions that will ultimately affect scheduling decisions. Before giving the steps of the algorithm, we will introduce the possible time components that should be included in the objective function of total cost for the manufacturing of a given batch size of a single part type. These components are classified into two distinct groups, namely machining time and non-machining time components. Machining time, $t_{m_{pij}}$, is the time required to complete a metal-cutting operation, as given in Gorczyca (1987). Taylor’s tool life expression is the relationship between machining time and tool life that can be expressed as a function of the machining conditions by using an extended form of Taylor’s tool life equation. The usage rate expression is obtained for the machining time to tool life ratio as:

$$U_{pij} = \frac{t_{m_{pij}}}{T_{pij}} = \frac{\pi D_{pi} L_{pi} / (12 v_{pij} f_{pij})}{C_j / (v_{pij}^{\alpha_j} f_{pij}^{\beta_j} d_{pi}^{\gamma_j})} = \frac{\pi D_{pi} L_{pi} d_{pi}^{\gamma_j}}{12 C_j v_{pij}^{1-\alpha_j} f_{pij}^{1-\beta_j}}$$

It is obvious that one cannot dedicate one tool to each operation, which would increase the number of tool types required by magnitudes and is infeasible in practice. As a result, we utilize the tool sharing concept and define a new tool usage rate term in order to implement tool sharing in practice. Consequently, we can find exactly how many operations can share the same cutting tool by calculating the

ratio of their machining time to the expected tool life given by the Taylor's tool life equation.

Non-machining time is the time required for all time-consuming events except the actual cutting operation. These should be minimized since they are directly affected by the tool management and scheduling decisions: t_l is the tool magazine loading time required to take the tool from the central storage and load on the magazine, t_r is the tool replacing time required for replacing a used tool with a new copy on the magazine, t_c is the tool changing time that accounts for the time necessary to move a tool from the tool holder to tool magazine and replace it back, t_t is the tool transfer time needed to relocate a tool from the ending point of an operation to the starting point of another operation when there is tool sharing, t_{rt} is the rapid travel motion time required to move the tool from a fixed point to the starting point of an operation or vice versa. Both machining and non-machining time components can be converted into their equivalent monetary units by multiplying them with the operating cost of the CNC machine, C_o . C_o is the labour and overhead rate applied to the metal-cutting operation in dollars per minute.

At this level both duplicate tool requirements and alternative tool usage are considered. After finding the best tool-operation assignments for each operation, we consider tool-sharing between the operations of each part in Step 1.5 to reduce the non-machining times by increasing the tool-sharing possibilities among the operations. The step by step illustration of this level is as follows.

Step 1.1. For every possible part, operation, tool triple, i.e. (p, i, j) , solve the following SMOP and initially set $r_{pij} = \lceil \frac{Q_p}{N_j} \rceil$ to ensure the feasibility in terms of the tool availability constraints, where $\lceil \cdot \rceil$ gives the smallest integer greater than or equal to the operand:

$$\begin{aligned} \text{minimize SMOP}_{pij} &= C_o \cdot t_{m_{pij}} + (C_t + C_o \cdot t_r) \cdot U_{pij} \\ &= C_1 \cdot v_{pij}^{-1} \cdot f_{pij}^{-1} + C_2 \cdot v_{pij}^{(\alpha_j-1)} \cdot f_{pij}^{(\beta_j-1)} \end{aligned}$$

$$\begin{aligned} \text{subject to } C_t' v_{pij}^{(\alpha_j-1)} f_{pij}^{(\beta_j-1)} &\leq 1 \quad (\text{tool life constraint}) \\ C_m' v_{pij}^b f_{pij}^c &\leq 1 \quad (\text{machine power constraint}) \\ C_s' v_{pij}^g f_{pij}^h &\leq 1 \quad (\text{surface roughness constraint}) \\ v_{pij}, f_{pij} &> 0, \end{aligned}$$

where

$$\begin{aligned} C_1 &= \frac{\pi D_{pi} L_{pi} C_o}{12}, & C_2 &= \frac{\pi D_{pi} L_{pi} d_{pi}^{\alpha_j} (C_t + C_o t_r)}{12 C_j} \\ C_t' &= \frac{\pi D_{pi} L_{pi} d_{pi}^{\alpha_j} r_{pij}}{12 C_j}, & C_m' &= \frac{C_m d_{pi}^e}{\text{HP}} \text{ and } C_s' = \frac{C_s d_{pi}^l}{\text{SF}_{pi}} \end{aligned}$$

In this formulation, we minimize the total manufacturing cost, which is the sum of machining, non-machining and tooling costs, to determine optimum v_{pij} , f_{pij} and U_{pij} . Consequently, $r_{pij} = \lfloor 1/U_{pij} \rfloor$ where, $\lfloor \cdot \rfloor$ gives the greatest integer smaller than the operand, and $n_{pij} = \lceil Q_p/r_{pij} \rceil$. The first constraint guarantees that machining time of an operation does not exceed available tool life. That means the machining conditions of an operation should be selected in a way that the remaining tool life is

enough to perform this operation. For example, if we are given that for the optimum solution only 10 parts can be manufactured for operation i of part type p by tool j , i.e. $r_{pij} = 10$, then the usage rate of each operation must be ≤ 0.1 . The machining resistance is in general given by the power function of cutting speed and feed rate, and it must not exceed the motor power of the machine tool as stated in the second constraint. In the last constraint, the surface roughness represents the quality requirement for the operation and should be less than a certain amount to ensure good product accuracy.

Step 1.2. Resolve SMOP for the requirement level, $k \in \{1, 2, \dots, n_{pij}\}$, of each triple (p, i, j) to find v_{pij}^k, J_{pij}^k and U_{pij}^k , and the corresponding manufacturing cost over the batch

$$TC_{pij}^k = Q_p \cdot (C_o t_{m_{pij}}^k + (C_v + C_o \cdot t_{r_j}) \cdot U_{pij}^k). \tag{1}$$

Step 1.3. For every (p, i) pair, find the (j, k) pair giving the minimum TC_{pij}^k and compute the tool type j requirement for every j as follows: $R_j = \sum_{(p,i)} Q_p * U_{pij}^k$, where $(p, i) = \operatorname{argmin}_{j,k} \{TC_{pij}^k\} \forall (p, i)$.

Step 1.4. If $R_j \leq N_j$ for every j , then the lower bound solution found in Step 1.3 gives the optimum tool allocations and machining conditions. Otherwise, solve the following integer programming (IP) formulation to find the best allocation for every operation that satisfies the tool availability constraints:

$$\begin{aligned} \min & \quad \sum_{p \in P} \sum_{i \in I_p} \sum_{j \in J} \sum_k TC_{pij}^k X_{pij}^k, \\ \text{subject to :} & \quad \sum_{j \in J} \sum_{k=1}^{n_{pij}+1} X_{pij}^k = 1 \quad \forall p \in P, i \in I_p \\ & \quad \sum_{p \in P} \sum_{i \in I_p} \sum_{k=1}^{n_{pij}+1} Q_p U_{pij}^k X_{pij}^k \leq N_j \quad \forall j \in J, \end{aligned}$$

where X_{pij}^k is a 0–1 binary decision variable which is equal to 1 if the machining of operation i of part p is assigned to tool j at the requirement level of k tools. We ensure that for every operation only a single alternative will be chosen, and the total tool usage will not exceed the available quantity for each tool in the first and second set of constraints, respectively. In our proposed model, an operation of a single part can be assigned to a cutting tool if its usage rate is ≤ 1 , i.e. $U_{pij} \leq 1$, due to surface finish requirements as discussed in Step 1.1. On the other hand, depending on the batch size and machining conditions, the number of tools required to produce a certain operation for a given batch size of part type p might be > 1 , i.e. $Q_p U_{pij} > 1$. In practice, we also know that the number of tools available for each tool type j is limited, denoted as N_j , for economical reasons. Therefore, the total tool usage for all operations of all part types for a certain tool type j must be $\leq N_j$, which is known as tool availability constraints.

Step 1.5. For each part, determine the operations of a part that use the same tool and check tool sharing possibilities if they satisfy the precedence relations and their total tool usage is < 1 . Calculate the machining and non-machining times of the composite operation.

An exact solution of the geometric programming formulation given in Step 1.1 can be found in a polynomial time as discussed in Akturk and Avci (1996). The proposed formulation can be very helpful in defining the influence of the machining conditions on the total manufacturing cost as depicted in figure 1. If we increase either v_{pij} or f_{pij} , or both, then we can reduce the machining time, but this will increase the tool usage, and equivalently non-machining and tooling costs. On the other hand, a heavy feed rate is conducive to formation of a built-up edge and a rough surface finish. Whereas high cutting speed improves the surface finish since it decreases the built-up edge formation on the face of a cutting tool. This paper makes a distinction between the machining time and the processing time. Machining time is defined as the time required to complete a metal cutting operation, which is the actual value-added operation, without considering the non-value adding components of non-machining times, such as tool replacing, tool changing, etc. Obviously, the actual processing time in practice will include both the machining and non-machining times. Therefore, the total processing time is the sum of machining and non-machining time components. However, it is not possible to calculate the exact non-machining time without knowing the current status of the tool magazines. Therefore, we initially approximate the expected non-machining time in terms of the usage rate and tool replacing time, and the processing time, $t = t_{mpij} + U_{pij}t_r$. The machining time is a strictly decreasing function of cutting speed, whereas the non-machining time is a strictly increasing function because of an increased usage rate a larger number of tool changes might be required (figure 2). Therefore, there exists a trade-off between the total manufacturing cost and the total processing time. In order to decrease the total processing time, we have to incur an additional manufacturing cost due to an increase in the non-machining and tooling costs. Furthermore, both the total manufacturing cost and total processing time are

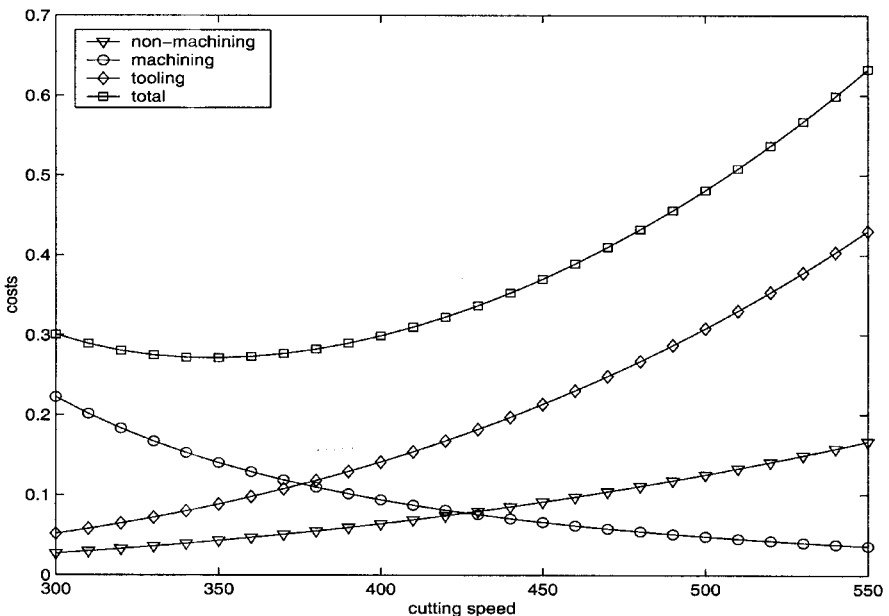


Figure 1. Costs versus cutting speed.

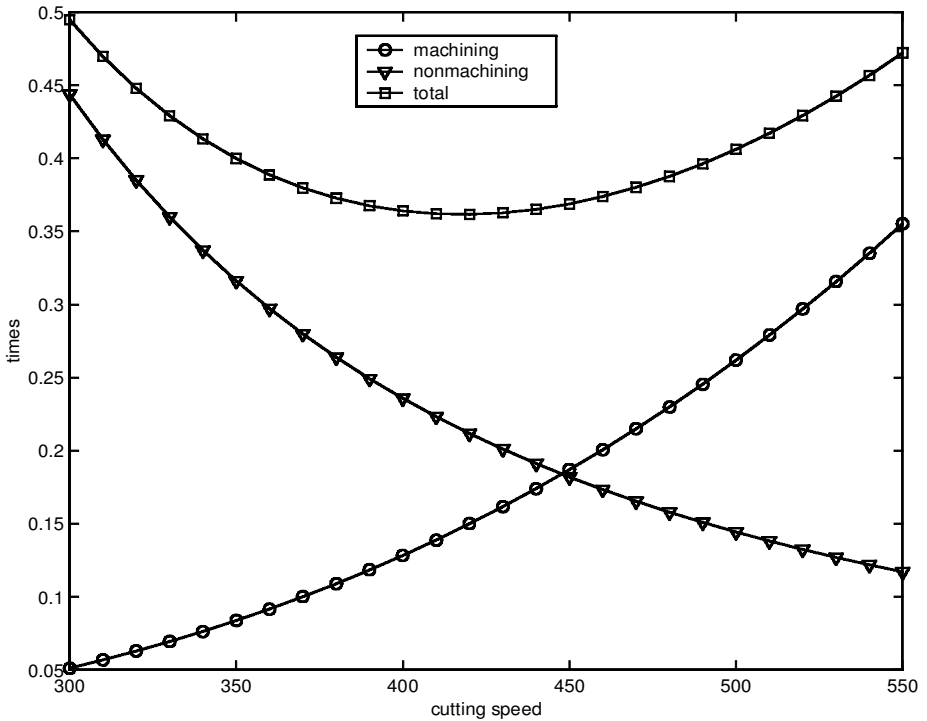


Figure 2. Times versus cutting speed.

convex in terms of the cutting speed. In order to prove convexity, it will be sufficient to show that $t_{m_{pij}}$ and U_{pij} are convex in terms of v_{pij} since any positive linear combination of these functions will be also convex. If we take the second derivatives of both functions:

$$\frac{\delta^2 t_{m_{pij}}}{\delta v_{pij}^2} = \frac{2\pi D_{pi} L_{pi}}{12v_{pij}^3 f_{pij}} > 0 \quad \text{and} \quad \frac{\delta^2 U_{pij}}{\delta v_{pij}^2} = \frac{\pi D_{pi} L_{pi} d_{pi}^{\alpha_j} (\alpha_j - 1) \alpha_j}{12C_j v_{pij}^{-1-\alpha_j} f_{pij}^{1-\beta_j}} > 0.$$

That means $t_{m_{pij}}$ is a strictly decreasing and U_{pij} is a strictly increasing convex functions of v_{pij} . The interval in which the processing times can vary is defined by the set of constraints. Akturk and Avci also prove that at least one of the surface roughness and machine power constraints is binding at optimality for SMOP. Thus, the machining conditions should always be set to a point on the boundary of the feasible region (figure 3). The portion of the boundary, where the processing times can be controlled, is called the efficient frontier and is determined according to the operational and tooling parameters. We will explain the derivation of the efficient frontier for a single operation on a numerical example in Section 5.3.

In order to find out the efficient frontier, we should first find four critical (v, f) pairs. The first pair (v_1, f_1) gives the machining conditions that minimize the manufacturing cost. In order to find the machining conditions that minimize the processing time, there are three possibilities, namely (v_2, f_2) , (v_3, f_3) and (v_4, f_4) . The second pair (v_2, f_2) is the intersection point at which both surface roughness and machine power constraints are tight.

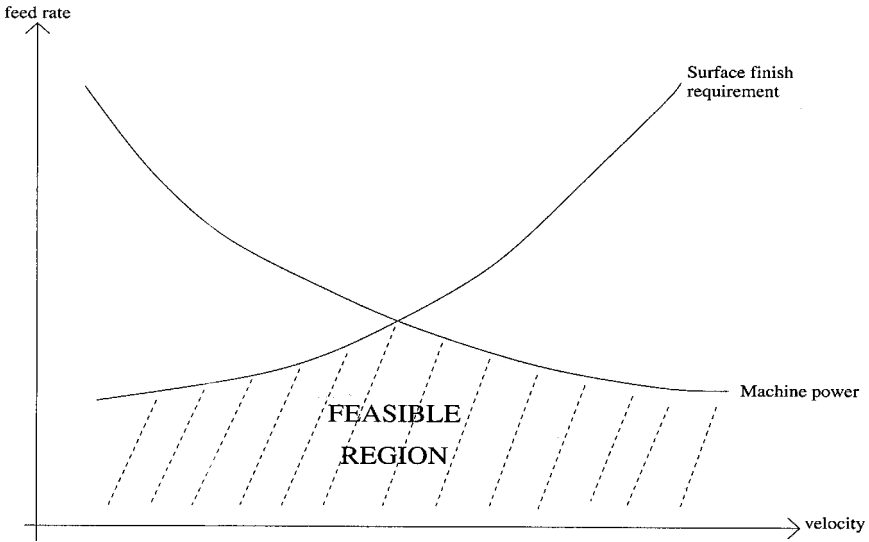


Figure 3. Feasible region.

$$v_2 = (C_s/SF)^{(c)/(hb-gc)} (C_m/HP)^{(-h)/(hb-gc)} d^{(lc-he)/(hb-gc)}$$

and

$$f_2 = (C_s/SF)^{(b)/(gc-hb)} (C_m/HP)^{(-g)/(gc-hb)} d^{(lb-ge)/(gc-hb)}.$$

The third pair (v_3, f_3) is the one that minimizes the total processing time on the surface roughness boundary. In order to find this pair, first we write the feed rate in terms of cutting speed using the surface roughness constraint as $f = (SF/C_s)^{1/h} d^{-1/h} v^{-g/h}$, then substitute this in the processing time expression

$$t = \frac{\pi DL}{12} \cdot (C_s/SF)^{1/h} d^{1/h} \left(v^{(g-h)/h} + (C_s/SF)^{-\beta/h} \cdot d^{(\gamma h - \beta)/h} \cdot \frac{t_r}{C_j} \cdot v^{(h(\alpha-1) - g(\beta-1))/h} \right).$$

We take the derivative of t with respect to v and solve the obtained expression for v to get v_3 . Finally, we substitute v_3 in the equation for f to get f_3 . However, the third pair of (v_3, f_3) will be on the efficient frontier if $v_3 < v_2$ since the surface roughness constraint is tight for velocities up to v_2 .

The last pair (v_4, f_4) is the one that minimizes the total processing time on the machine power boundary, hence we define the feed rate in terms of cutting speed using the machine power constraint, substitute this in the processing time expression, and take the following derivative to find v_4 :

$$\frac{dt}{dv} = \frac{\pi DL}{12} (C_m/HP_{\max})^{1/c} d^{e/c} v^{(b/c)-2} \times \left(\frac{b-c}{c} + (C_m/HP_{\max})^{-\beta/c} d^{(\gamma c - \beta e)/c} \frac{t_r}{C_j} \frac{-b(\beta-1) + c(\alpha-1)}{c} v^{(c\alpha - b\beta)/c} \right).$$

If both $(b/c - 1)$ and $(c(\alpha - 1) - b(\beta - 1))/c$ are non-negative, not simultaneously being zero, processing time will be a strictly increasing function of the velocity. Thus, the machine power constraint will not be active. If both $(b/c - 1)$ and $(c(\alpha - 1) - b(\beta - 1))/c$ are non-positive, not simultaneously being zero, processing

time will be a strictly decreasing function of velocity. However, this case is impossible, since $\alpha > \beta > 1$, hence $(\alpha - 1)/(\beta - 1) > 1$. If one of $(b/c - 1)$ and $(c(\alpha - 1) - b(\beta - 1))/c$ is positive and the other is negative, there is a pair (v_4, f_4) where it gives the minimum total processing time. v_4 can be solved by setting the derivative to zero and the corresponding f_4 can be obtained. However, the fourth pair of (v_4, f_4) will be on the efficient frontier if $v_4 > v_2$ since the machine power constraint is tight for velocities over v_2 .

Hence, we can explicitly define the efficient frontier as follows:

If $(v_2 \leq v_3) \wedge$ either $(b/c < 1 \wedge v_2 > v_4)$ or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) > b/c)$
 or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) < b/c \wedge v_2 > v_4)$
 then the efficient frontier is from (v_1, f_1) to (v_2, f_2) ,

else either $(b/c < 1 \wedge v_2 \leq v_4)$ or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) < b/c \wedge v_2 \leq v_4)$
 then the efficient frontier is from (v_1, f_1) to (v_4, f_4)

If $(v_2 > v_3) \wedge$ either $(b/c < 1 \wedge v_2 > v_4)$ or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) > b/c)$
 or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) < b/c \wedge v_2 > v_4)$
 then the efficient frontier is from (v_1, f_1) to (v_3, f_3) ,

else either $(b/c < 1 \wedge v_2 \leq v_4)$ or $(b/c > 1 \wedge (\alpha - 1)/(\beta - 1) < b/c \wedge v_2 \leq v_4)$
 then the efficient frontier is from (v_1, f_1) to (v_3, f_3) and from (v_2, f_2) to (v_4, f_4) .

For the last case, the efficient frontier is discontinuous, therefore some of the points in the second part might have a higher value in terms of the total processing time than the ones in the first part. Thus, in order to find the relevant range of the second part, the value of the processing time at (v_3, f_3) can be calculated and the expression for t for the second part can be solved to find a new v , called v_5 . If $v_5 < v_2$, then the second part starts from (v_2, f_2) , otherwise f_5 corresponding to v_5 is found and the second part starts from (v_5, f_5) .

3.2. Initial schedule

In order to find an initial schedule that will minimize total production cost, we propose two ranking indices. The first one is used for choosing the machine that each part type can be loaded on, and the other one is used for choosing the part type that will be processed. In the proposed scheduling algorithm, we are scheduling all parts of a particular part type in a given batch size, Q_p , i.e. lot splitting is not allowed. Since the non-machining times are state-dependent, we schedule one part type at a time and recalculate the non-machining times of the unscheduled part types after each assignment to consider the actual tool sharing possibilities. It is important to note that the shop orders of different part types will correspond to different customer requirements in terms of the required batch sizes, due dates, etc. Furthermore, firms have a variety of customers, some of which are more important than others. The importance of a customer order for a certain part type, w_p , can depend on a variety of factors, e.g. the firm's length of relationship with the customer, how frequently they provide business to the firm, and the potential of a customer to provide orders in the future. Therefore, it is important for manufacturing to reflect these priorities in

their scheduling decisions. In addition, in the presence of job tardiness penalties, it may not be enough to measure the shop floor performance by employing unweighted performance measures alone which treat each job in the shop as equally important.

The first index is the machine preference ranking index, MI_{pm} , for each part type, p , machine, m , pair given by the following equation:

$$MI_{pm} = \frac{w_p}{(t_{m_{pm}} + t_{s_{pm}})} (DD_p - t_m^c - (t_{m_{pm}} + t_{s_{pm}})). \quad (27)$$

This index is a combination of weighted shortest processing time and the minimum slack time rules. As indicated above, the total processing time of a part type consists of machining and non-machining time components. The proposed index gives a higher priority to the machine which is faster and requires less total non-machining time. The machine with the highest index becomes the preferred machine of that part type. The machining time of the part type, $t_{m_{pm}}$, is the same for all machines, which is determined according to the machining conditions selected at the first level. However, the non-machining time, $t_{s_{pm}}$, required for a part type will be different on each machine, since the tool replacing time depends on the number of tool replacements related to the current status of the tool magazines of the machines. The proposed ranking index is a dynamic rule because it is a function of the time t_m^c at which the machine m became free, as well as the w_p , DD_p , $t_{m_{pm}}$ and the current status of the tool magazine to calculate $t_{s_{pm}}$.

When we schedule a part type on a certain machine, the current status of the tool magazine of that machine usually changes, since the new part type either might require new additional cutting tools to be loaded or the existing tools on the tool magazine may not have enough remaining tool life. Therefore, this machine is called 'an altered machine', because both t_m^c (the completion time of the last scheduled part) and $t_{s_{pm}}$ of the unscheduled part types on this machine will be recalculated according to the new status of the tool magazine. For the remaining $m - 1$ machines, the current machine ranking indexes will remain the same.

In the calculation of the non-machining time for a part type p on machine m , $t_{s_{pm}}$, we first try to find how many parts of the batch can be processed by the tools currently present on the tool magazine. In order to find the actual tool sharing possibilities between the parts, we keep track of the exact remaining tool lives of each tool on the magazine. When a single copy of a tool is used for the whole batch, the remaining tool life, (r_{life_j}), can be found as $r_{life_j} = 1 - (Q_p U_{pij})$. However, if multiple copies of a tool are used, we have to find how many parts are processed by the last tool in order to find the remaining tool life. First, we find the number of parts that can be processed by a single tool given by r_{pij} . Then we sum up the number of parts that are processed up to the last tool and subtract this from the batch size to find the number of parts for the last tool, which is multiplied by the usage rate to find the remaining life of the tool currently loaded on the magazine as follows: $r_{life_j} = 1 - (Q_p - (n_{pij} - 1)r_{pij})U_{pij}$. If the whole batch cannot be processed by the tools present on the magazine and the magazine is full then an additional non-machining time will be incurred since one of the currently loaded tools must be unloaded to open up a new slot. The tool that will be unloaded is chosen as the one either that has zero remaining life or that has the shortest remaining life and is not required for the part in consideration.

The second index is the part ranking index, PI_{pm} , given by the following equation:

$$PI_{pm} = \frac{w_p}{(t_{m_{pm}} + t_{s_{pm}})} \exp \left\{ \frac{-\max \{DD_p - t_m^c - (t_{m_{pm}} + t_{s_{pm}}), 0\}}{k * \bar{p}_m} \right\}. \quad (3)$$

This index is calculated for each part type on its preferred machine which is determined according to the first index. The proposed index gives a higher priority to the part type which can be processed faster and has a less slack time. The main aim of this index is to reduce the amount of weighted tardiness.

In the second level, we first determine the preferred machine of each part type using the first index, and then select the part type which will be loaded using the second index. Once a part type is chosen to be loaded, the current status of the tool magazine is updated according to the necessary tool loadings and unloadings. The remaining tool lives are also recalculated by subtracting the usage amount of the loaded part from the initial tool lives. After we finish loading a part type, we recalculate the non-machining times required for the unscheduled part types only on the altered machine, and calculate the indexes MI_{pm} and PI_{pm} to choose the next part type to be loaded. We repeat this procedure until all part types are scheduled. We can illustrate this level step by step as follows:

- Step 2.1. Since initially all the tool magazines are empty, there is no difference between the machines. For each part type calculate the index PI_{pm} and select the part type with the highest PI_{pm} to load on the first machine.
- Step 2.2. After loading of a part type is completed, calculate the remaining lives of the tools currently loaded on the magazine and the average processing time, \bar{p}_m , of the unscheduled part types for each unaltered machine. On the other hand, calculate the non-machining time requirement of each unscheduled part type only on the altered machine.
- Step 2.3. Initially calculate the index MI_{pm} for all machines in order to choose the preferred machine for each unscheduled part type. However, after the first iteration, the index MI_{pm} should be calculated for only the altered machine, since it will remain the same for the unaltered machines.
- Step 2.4. After calculating the index PI_{pm} on the preferred machine for each part type, load the part type with the maximum index to its preferred machine.
- Step 2.5. If any tool with remaining life greater than zero but smaller than the minimum $\{U_{pij}\}$ is to be removed from the magazine, we check all the operations using this tool and increase the cutting speed of the most beneficial operation as much as possible to decrease the total production cost. Go to Step 2.2 until all part types are scheduled.

In sum, the non-machining time is a state-dependent variable, and it depends on the current arrangement of the tool magazine, i.e. which tools are loaded and their respective remaining tool lives. Therefore, we have to keep track of the state of the tool magazine after scheduling a certain part type that means we remove some cutting tools and add new ones, if necessary, and update the remaining tool lives for the tools that will be used to manufacture the given batch size. One of the primary objectives in the proposed algorithm is to minimize the total weighted tardiness. The reason we schedule one part type at a time is to calculate the slack values of each part type accurately. The slack value for each part type,

$DD_p - t_m^c - (t_{m,pm} + t_{s,pm})$, is not constant and changes over time, since both t_m^c and $t_{s,pm}$ are not constant, and depend on the current tool magazine arrangement of machine m and the set of part types that are already scheduled on machine m .

3.3. Final schedule

At the end of the first level, we obtain the tool allocations with their governing machining conditions. In this level, we not only determine the primary tool for any operation, but also find the best alternative tool for the same operation. At the third level, we allow that the processing times can be controlled via either the cutting speed or the feed rate. We choose the cutting speed as our controllable variable and determine the feed rate accordingly. Besides reducing the processing time by using the primary tool, we also consider alternative tool usage and batch splitting at this level if there is not sufficient slack amount of the primary tool. The third level can be considered as a left shift procedure, where we retain the same sequence found in the second level, but the starting times are shifted to the left in a Gantt chart representation by decreasing the processing times as much as possible to decrease the total production cost.

In order to find the efficient frontier, we classify each operation according to their tooling and operational parameters. We use piecewise linearization to approximate the non-linear efficient frontier into pieces of equal cutting speed range. If the last remaining piece has a shorter cutting speed range than the fixed step size, we will add it to the previous piece, otherwise we will consider it as a single piece. After doing the piecewise linearization, we propose a ranking index for each piece to choose the operation that will be crashed. This index shows us the opportunity cost-related to tardiness cost. The index of piece s of operation i of part p , TI_{pis} , is defined as:

$$TI_{pis} = \left(\frac{\Delta TC}{\Delta t} \right) / \sum_k w_k.$$

ΔTC shows us the increase in the total manufacturing cost consisting of the machining, non-machining and tooling costs, when we crash the processing times. Since (v_1, f_1) is the optimum solution for the total manufacturing cost, any (v, f) pair other than (v_1, f_1) will give a higher total manufacturing cost. Δt represents the total gain in the processing time as a result of the crashing. The numerator of the proposed index shows the increase in the total manufacturing cost for the time gain in the processing time, whereas the denominator shows the total gain in terms of tardiness cost, when a unit reduction in the total processing time of that operation is achieved. After calculating the index for each piece of every operation, we choose the most beneficial operation, that is the one with the smallest TI_{pis} , so that a unit gain in tardiness cost can be achieved with less cost. In sum, we start with the initial schedule and then look for the operation when it is crashed that results in the 'biggest bang for the buck' with respect to production cost improvement.

After choosing the most beneficial operation, if we have enough slack of the required tool to speed up the operation, we crash its processing time and find the new starting times of each part type on this machine. Otherwise, we consider the alternative tools to perform the same operation. If we have any gain in processing time when we use the alternative tool, i.e. $t_{ini} > t_{alt}$, then we calculate an alternative tooling index of operation i of part p , ATI_{pi} , as follows:

$$ATI_{pi} = \frac{(TC_{alt} - TC_{ini})}{(t_{ini} - t_{alt})} \bigg/ \sum_k w_k,$$

where TC_{alt} and TC_{ini} are the total manufacturing cost of operation i using the alternative and primary tools, respectively. Similarly, t_{alt} and t_{ini} show the total processing time of the alternative and primary tools, respectively. The numerator of the index again gives the additional cost incurred for unit time gain in the processing time when an alternative tool is used, whereas the denominator gives the total gain in tardiness cost for a unit reduction in the processing time. If this index < 1 , then we allow batch-splitting and calculate the amount of parts that can be processed by the primary tool and process the remaining parts with the alternative tool. The steps of the third level can be given as follows:

- Step 3.1. For every part-operation pair, determine the efficient frontier where the processing time can be controlled as discussed in Section 3.1
- Step 3.2. Using piecewise linearization, calculate the index TI_{pis} for each piece and sort the indices in increasing order. Select the smallest TI_{pis} value.
- Step 3.3. If there is enough slack of the required tool to meet the increased usage due to higher velocity, recalculate the total machining and setup time for the part type of this triple (p, i, s) and left shift the parts on the machine where the selected part type is scheduled.
- Step 3.4. If there is not enough amount of primary tool and alternative tool usage is beneficial, i.e. $ATI_{pi} < 1$, then repeat Step 3.3 for an alternative tool.
- Step 3.5. Recalculate the index values for the remaining operations and repeat the above procedure until there is no more beneficial operation to be crashed, i.e. both TI_{pis} and $ATI_{pi} > 1$ for every operation i of part type p .

4. Experimental design

In this section we test the efficiency of the proposed algorithm by comparing with some of the existing algorithms in the literature. All of the algorithms are coded in the C language and compiled with Gnu C compiler. The IP formulation in the first level of the proposed algorithm is solved by using callable library routines of CPLEX MIP solver on a Sparc station 10 under SunOS 5.4. There are seven experimental factors that can affect the efficiency of our algorithm, which are listed in table 1. The experimental design is a 2^7 full-factorial design as there are seven factors with two levels each. The number of replications of each combination is taken as five producing 640 different randomly generated runs.

Factors	Definition	Level 1	level 2
A	number of machines	2	5
B	number of part types	30	50
C	magazine capacity	10	20
D	tool availability (%)	80	120
E	number of tool types	10	20
F	due date tightness	tight	loose
G	tooling cost	UN ~ [0.8, 1.2]	UN ~ [1.2, 1.8]

Table 1. Experimental factors.

The number of machines and part types determine the size, product mix and load of the system. As the number of machines or part types increases, the scheduling decision becomes more important. The magazine capacity, which is identical for each machine, determines the number of tools that can be loaded simultaneously to the machine. It affects the actual setup time required for the parts. The fourth factor specifies the tightness of the tool availability constraint. The number of available tools on hand is taken as 80 and 120% of the required amount of tools for each tool type at low and high levels, respectively. The fifth factor is the number of tool types. As the number of tool types increases, the operation-tool assignment alternatives increase. The sixth factor is used to determine the due dates of the part types. In the tight case, due dates are randomly generated in the first half of the estimated makespan, whereas in the loose case, due dates are distributed in a wider range. The estimated makespan, MS , is calculated by dividing the sum of processing times to the number of machines. In the tight case, the due dates are chosen from the interval $UN \sim [0.1 \cdot MS, 0.5 \cdot MS]$, whereas, in the loose case, due dates are chosen from the interval $UN \sim [0.2 \cdot MS, 0.8 \cdot MS]$, where UN stands for the uniform distribution. Finally, the seventh factor is the tooling cost, which is likely to affect operation-tool assignments and the crashing decisions at the final level.

Other variables are treated as fixed parameters and generated as follows. The operating cost, C_o , is equal to 0.5/min, and $HP = 5$ h.p. The operation related parameters, D_{pi} and L_{pi} , are selected randomly from the interval $UN \sim [1.5, 2.5]$ and $UN \sim [5, 7]$, respectively. Batch sizes are selected from a discrete distribution with probability mass function of $f_Q(q) = \{0.3 \text{ when } Q = 10, 0.4 \text{ when } Q = 15, \text{ and } 0.3 \text{ when } Q = 20\}$. The number of operations per part is chosen from an integer interval $UN \sim [3, 5]$, and the last operation of each part is taken to be a finishing operation whereas the remaining operations to be roughing operations. SF_{pi} and d_{pi} are related with the type of operations. For roughing operations, $SF_{pi} = UN \sim [300, 500]$ and $d_{pi} = UN \sim [0.2, 0.3]$, and for the finishing operation, $SF_{pi} = UN \sim [30, 70]$ and $d_{pi} = UN \sim [0.025, 0.075]$. The weight of each part type is chosen from the integer interval $UN \sim [1, 3]$.

Five performance measures are used for comparison purposes, which are tooling, operational, tardiness and total production costs, and run time. The tooling cost is the total cost of tool usage in the system. The operational cost is the sum of machining and non-machining time costs, i.e. the total cost of operating the CNC machines. The tardiness cost is the total weighted cost of the part types that are tardy. The total production cost is the sum of these three cost terms. Finally, the run time is the computation time in seconds.

The experimental design is also applied to five existing algorithms in the literature, which are LPT-I, LPT-II, ARM, APS, and KTNS-CN. The first four are developed by Kim and Yano (1993) and the last one is proposed by Askin and Standridge (1993). In the LPT-I and LPT-II algorithms, the part type with the longest processing time is assigned to the machine which has the minimum load after the part type is assigned to it. The main difference between these two algorithms is that the first one ignores tool sharing possibility between the parts and therefore uses the constant setup time value calculated at the beginning. However, the second algorithm considers actual tool sharing possibilities and recalculates the setup time required for each unscheduled part type after a part type is loaded to a machine. ARM loads the unscheduled part type with the largest T/S to the machine with the largest T/S . The T/S ratio for a machine is the ratio of the remaining processing time

capacity of the machine to the remaining tool magazine capacity. The T/S ratio for a part type is the ratio of the processing time of the part type to the number of tool slots required for the part type. Each part type might have a different T/S ratio for each machine due to tool commonality. The basic idea of the ARM selection criterion is that larger items are packed in larger bins to achieve a better loading. APS loads the part type that requires the largest number of tool slots on the most preferred machine to that machine. The most preferred machine for a part type is the one on which minimum setup time is required. Tang and Denardo (1988) prove that the common sense rule keep tool needed soon (KTNS) is optimal for changing the tool magazine when there is a deterministic change time and all changes are due to part mix, ignoring tool changes due to tool wear. KTNS-CN algorithm by Askin and Standridge removes only as many tools as necessary to make way for the next part type. The tools removed are those that will not be needed again until the longest time in the future and loads the part type that requires the minimum setup time on its most preferred machine, in other words, the closest neighbour to the current status.

All of these algorithms use only 0–1 type variables when assigning tools to operations, although there may be cases where a single operation requires duplicate tools. They do not consider tool lives and assume that any tool can perform two to four part types. Their approach can be considered as a complete sharing. However, this might be an unrealistic assumption, since the tool life is dependent on the machining conditions and it may not be possible for each tool to be shared by the parts due to their usage amounts. Processing times are assumed to be fixed, and chosen from some probabilistic distribution. They do not consider the fact that processing times can be controllable via either the cutting speed or the feed rate. Finally, they do not consider alternative tool assignments for the operations. In order to make these algorithms comparable with our proposed algorithm, we modify them to consider duplicate tooling and actual tool lives.

The overall results of the algorithms are summarized in table 2. The table shows the minimum, average and the maximum values for the performance measures for all of the algorithms. INIT corresponds to the initial schedule of the proposed algorithm found after the second level, whereas the final schedule is denoted by FINAL. In sum, the average performance of the proposed algorithm is significantly better than the algorithms used in the literature in terms of the total production cost, although it requires a relatively higher computation time.

For the tooling cost, INIT gives the minimum average tooling cost, whereas FINAL gives the maximum average tooling cost. The reason why it gives the minimum at the initial schedule is that all tool sharing possibilities are evaluated. However, at the final schedule, in order to decrease the total cost, we increase tool usage which in turn results in higher tooling cost. LPT-I gives the highest tooling cost among the other algorithms, since it does not consider tool sharing possibility among parts. When we interpret the operational cost values, we see that our proposed algorithm results in the minimum average operational cost. The proposed algorithm has the minimum average tardiness cost as expected which is far below the tardiness costs of other algorithms, since they do not consider the scheduling problem while solving the tool management problem. For the total cost, the best average performance is achieved again by the proposed algorithm. Finally, LPT-I is the fastest algorithm in terms of run times as expected since it only calculates the setup time at the beginning and uses the same value throughout the algorithm.

Algorithm		Tooling	Operational	Tardiness	Total	Run time
LPT-I	min.	107.9	937.7	3788.0	4898.0	0.44
	ave	182.1	1423.1	18 448.6	19 953.8	0.97
	max.	303.5	2099.0	52 229.0	54 283.1	2.13
LPT-II	min.	77.3	941.4	3797.9	4899.1	0.50
	ave	155.0	1420.8	18413.8	19 889.6	1.10
	max.	281.1	2097.3	52 034.2	54 534.86	2.46
ARM	min.	60.5	937.4	3865.1	4939.9	0.55
	ave	145.7	1432.8	18 876.1	20 064.5	1.18
	max.	282.6	2102.0	52 200.3	53 927.7	2.12
APS	min.	74.7	933.5	1926.0	2969.9	0.53
	ave	152.6	1451.2	13 778.9	15 382.7	1.28
	max.	285.6	2096.3	51 698.9	53 718.9	3.09
KTNS-CN	min.	61.7	946.3	2134.5	3203.1	0.56
	ave	146.6	1440.4	14 504.7	16 041.8	1.32
	max.	293.2	2083.9	53 273.0	55 192.2	3.14
INIT	min.	58.6	898.0	1118.4	2181.4	0.46
	ave	143.3	1383.7	9735.9	11 262.9	1.19
	max.	276.1	2009.1	51 046.3	53 106.9	3.37
FINAL	min.	78.2	888.7	1023.0	2107.5	0.35
	ave	192.3	1379.7	9518.5	11 090.5	0.97
	max.	363.3	2017.3	49 925.9	52 022.5	2.98

Table 2. Comparison of the performance measures of algorithms.

However, other algorithms recalculate the setup times at each iteration. The run times of final schedule show the additional time required over the initial schedule.

We also applied a two-way analysis of variance (ANOVA) test on the performance measures. The significance levels (p) and F for each of them over seven factors are given in table 3. For the tooling cost, all the factors except the due date tightness are significant with $p \leq 0.000$. Among them, the factor B specifies the total tool requirement, the factor G determines unit tooling costs, whereas factors A and C restrict the tool sharing possibilities, hence affect the tooling cost incurred. Factors D and E limit the number of tools on hand, hence the allocation decisions. For the operational cost, only three factors are significant with $p \leq 0.000$. These are number of part types, tool magazine capacity and tool availability. The factor B determines the load on the system, therefore the cost of producing the parts. The factor C limits the tool sharing possibility, and hence affects the non-machining time. The factor D restricts the number of tools on hand. Each operation cannot always be assigned to its best tool alternative due to the tool availability constraints. Hence, this will result in increased machining times and consecutively increased total operational cost. For the tardiness cost, the factors A, B, D and F are significant with $p \leq 0.000$. The estimated makespan is a function of factors A, B and D, whereas the factor F directly determines the due date range of the part types.

We can summarize our findings as follows. The FMS design parameters of number of machines, part types and alternative cutting tool types, tool availabilities and tool magazine capacities have a significant impact on the operational decisions of part scheduling and tool management. As the load of the system increases, the

	Tooling cost		Operational cost		Tardiness cost		Total cost		Run time	
	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>	<i>F</i>	<i>p</i>
A	64.610	0.000	0.010	0.92	607.524	0.000	586.443	0.000	0.541	0.462
B	1445.849	0.000	2136.247	0.000	555.495	0.000	621.258	0.000	842.938	0.000
C	730.291	0.000	16.100	0.000	8.546	0.004	9.809	0.002	0.909	0.341
D	31.965	0.000	536.078	0.000	43.215	0.000	52.958	0.000	264.564	0.000
E	692.517	0.000	0.709	0.4	0.003	0.954	0.001	0.975	538.556	0.000
F	1.832	0.176	0.083	0.773	1856.99	0.000	1798.638	0.000	0.059	0.809
G	1656.553	0.000	4.761	0.03	3.034	0.082	3.872	0.05	6.26	0.013

Table 3. *F* and significance levels (*p*) for ANOVA results.

scheduling and tool management interaction becomes even more important. For example, if the tool magazine capacity decreases then the tool replacements are done more frequently, hence the non-machining times increase. As the number of tool types and tool availability are at their high values, the chance to find better operation-tool assignments increases and this will bring solution flexibility to the system, which in turn results in lower total production cost. Obviously, there is an opportunity cost since higher tool availability will increase the tool inventory cost, whereas larger tool magazine capacity will require a higher initial investment cost. Tool costs are not only the main determinants of the total tooling cost, but also affect the crashing decisions available for the final schedule. Finally, tool sharing is beneficial both in terms of tooling cost and operational cost via the non-machining cost. LPT-I algorithm, which ignores the tool sharing, always gives the maximum value for the cost measures.

5. Numerical example

This section illustrates the proposed algorithm on a numerical example to point out the important steps, by focusing on each level in the following subsections. Our example problem consists of 10 part types, two machines and 10 tool types. The part related data are summarized in table 4 and the tool related data are (tool number, t_{r_j} , t_{l_j} , t_{c_j} , t_{t_j} , t_{rt_j} , C_{t_j} , N_j) = (1, 0.87, 1.24, 0.40, 0.13, 0.06, 1.087, 4), (2, 0.75, 1.37, 0.48, 0.14, 0.07, 1.034, 1), (3, 0.82, 1.49, 0.38, 0.10, 0.09, 1.054, 1), (4, 0.72, 1.15, 0.31, 0.15, 0.14, 1.003, 22), (5, 0.93, 1.19, 0.32, 0.16, 0.12, 1.050, 4), (6, 0.92, 1.24, 0.32, 0.18, 0.10, 1.087, 1), (7, 0.75, 1.32, 0.33, 0.18, 0.10, 0.970, 5), (8, 0.94, 1.27, 0.42, 0.15, 0.11, 1.134, 2), (9, 0.84, 1.49, 0.38, 0.12, 0.09, 0.978, 2), and (10, 0.86, 1.31, 0.48, 0.20, 0.08, 0.829, 8). Each cutting tool type might have different non-machining time components of tool replacing, changing, loading, etc., depending on whether or not the tool

Part type number	1	2	3	4	5	6	7	8	9	10
No. of operations	5	5	3	5	3	4	5	3	3	4
Weight	2	2	3	1	3	3	3	1	2	1
Due date	41	170	65	111	177	181	67	92	40	49
Batch size	15	15	15	15	10	15	10	15	20	15

Table 4. Part type-related information.

Downloaded by [Bilkent University] at 03:04 13 November 2017

uses some special accessory. In our numerical example, the tool replacing time for tool type 1, $t_{r_1} = 0.87$, whereas it is 0.75 for tool type 2, and so forth.

5.1. Tool allocation

As the first step of this level, we solve SMOP to determine the optimum machining conditions for every possible part type, operation and tool triple. We calculate the cost measure given in equation 1 for each alternative, and find the (j, k) pair giving the minimum cost value for each (p, i) pair. If we consider the first operation of the first part type, i.e. (1,1), there are six alternative tool types that can perform this operation and tool 7 gives the minimum cost measure since (tool number, $t_{m_{pi}}$, TC_{pij}) = (1, 0.228, 2.033), (2, 0.315, 3.206), (3, 0.262, 3.008), (5, 0.134, 1.553), (7, 0.141, 1.4), and (8, 0.16, 1.53). We then compute the total requirement for each tool type j to check tool feasibility. If $R_j \leq N_j$ for every j , then the lower bound solution found above is optimum. Otherwise, an IP formulation given in Step 1.4 is solved to find the best allocation for each operation that satisfies tool availability constraints. In our example, we assume that there is an 80% tool availability and there is no precedence relation among the operations of a part. Therefore, we have to solve an IP formulation to find the optimum tool allocations. These best allocations for part type 1 can be summarized as (operation number, tool number, usage rate) triples as (1, 7, 0.134), (2, 4, 0.066), (3, 7, 0.066), (4, 4, 0.034), and (5, 4, 0.059).

Before the scheduling decision, we find out tool sharing possibilities among the operations of the same part, which will result in a direct reduction in the total non-machining time required for that part. If we consider part type 1, we can gather operations using the same tool as long as their total usage rate does not exceed 1, which means that the same tool can perform both of the operations. For example, the first and third operations use tool type 7 and their total usage ($0.134 + 0.066 = 0.2$) < 1 . Therefore, we gather these two operations into a single operation, whose processing time is the sum of the processing times of the individual operations ($0.141 + 0.24 = 0.381$). Consequently, we reduce the non-machining time required for each part by $t_{c_7} + t_{t_7} - t_{rt_7} = 0.33 + 0.18 - 0.10 = 0.41$ minute. Similarly, the second, fourth and fifth operations of this part can also be gathered since their total tool usage is 0.159. After gathering the possible operations, we calculate total machining and total expected set up time required for each part type to be used in the scheduling level as follows: (part no., t_m , t_s) = (1, 48.9, 4.76), (2, 104.6, 7.88), (3, 51.3, 6.31), (4, 58.2, 5.82), (5, 36.8, 3.57), (6, 73.2, 5.08), (7, 65.8, 7.48), (8, 64.1, 4.72), (9, 94.4, 5.65) and (10, 66.3, 5.19).

5.2. Initial schedule

As discussed above, we find an initial schedule by utilizing two ranking indices, MI_{pm} and PI_{pm} . Since all the tool magazines are empty at the initial state, there is no difference between the machines. So we just calculate the index PI_{pm} for each part type and load the one with the highest index to the first machine. The initial PI_{pm} are calculated as follows: (part, PI) = (1, 0.0373), (2, 0.0119), (3, 0.0495), (4, 0.0113), (5, 0.0259), (6, 0.0188), (7, 0.0409), (8, 0.0124), (9, 0.02) and (10, 0.014). Since part type 3 has the highest index value, it is loaded to the first machine. The operations of part type 3 with their allocated tools and associated usage rates are (1, 4, 0.065) and (2, 10, 0.267). Initially, part type 3 had three operations but after the tool sharing the last two operations were aggregated into a single operation. We calculate the remaining tool lives of the tools currently on the magazine, which are tool type 4 and 10. A

single copy of tool 4 is used for the whole batch, therefore the remaining life is equal to $r_{\text{life}_4} = 1 - (15 \cdot 0.065) = 0.025$. However, multiple copies of tool 10 are used because $Q_p \cdot U_{p_{ij}} > 1$. In order to find the remaining life of the tool currently loaded on the magazine, we have to find how many parts are processed by the last copy as discussed in Section 3.2 and $r_{\text{life}_{10}} = 1 - (15 - (4 \cdot 3)) \cdot 0.267 = 0.199$.

Then we update the current time of the first machine, which becomes the completion time of the last loaded part type to that machine, hence $t_1^c = 51.3 + 6.31 = 57.61$. Next, we calculate the average processing time of unscheduled part types on the second machine, i.e. $\bar{p}_2 = 73.51$. Afterwards, we calculate the actual setup time required on the first machine and MI_{pm} for both machines for each unscheduled part type in order to choose the preferred machine for each part type. However, in this example after the initial loading of part type 3 to the first machine, there is no difference in the setup times on the machines for each part type. Therefore, we will not calculate MI_{pm} . All the part types will prefer the machine at which they can start earlier to have more slack time. The second machine becomes the preferred one and part type 7 is assigned to it in the second iteration, The remaining tool lives of the tools on the second machine are calculated and the current time of the second machine is set equal to the completion time of part type 7. We repeat these steps until there is no more unscheduled part types.

Let's explain this procedure in detail after the partial schedule of M/C 1: {3 - 5 - 6}, and M/C 2: {7 - 1 - 9}. In the previous iteration, part type 9 is loaded to the second machine. Therefore, the setup time required on the second machine for the unscheduled part types, which are 2, 4, 8 and 10, should be recalculated due to the changes in the current status of the magazine. As an example, the calculations done for the part type 2 are shown below.

$$t_{m_{21}} = t_{m_{22}} = Q_p \cdot (t_{m_{216}} + t_{m_{224}} + t_{m_{236}} + t_{m_{244}} + t_{m_{253}} + 2(t_{c_6} + t_{t_6}) + 2(t_{c_4} + t_{t_4}) + 2(t_{c_3} + t_{t_3}) + t_{r_6} + t_{r_4}) = 15 \cdot (0.765 + 1.224 + 0.611 + 0.689 + 0.567 + 2(0.32 + 0.18) + 2(0.31 + 0.15) + 2(0.38 + 0.10) + 0.10 + 0.14) = 104.6$$

$$t_{s_{22}} = t_{i_3} + t_{i_4} + t_{i_6} + (3 \cdot t_{r_4}) + t_{r_6} = 1.49 + 1.15 + 1.24 + (3 \cdot 0.72) + 0.92 = 7.08.$$

Similarly, $t_{s_{21}}$ is calculated to be 8.20. After finding the total machining and non-machining times for part type 2, we calculate the machine preference index given by equation 2.

$$MI_{21} = (2/(104.6 + 8.2)) \cdot (170 - 176.26 - (104.6 + 8.2)) = -2.114.$$

MI_{22} is found to be -2.992 by the same way. Since MI_{21} is greater than MI_{22} , second machine is preferred by part type 2. Then we calculate the index PI_{21} given by equation (3).

$$PI_{21} = (2/(104.6 + 8.2)) \cdot \exp \{-\max \{170 - 176.26 - (104.6 + 8.2), 0\} / (2 * 80.04)\} = 0.0181.$$

The results of these calculations for all unscheduled part types are shown in table 5. Since part type 2 has the highest index value, it is loaded to the first machine. The initial schedule obtained at the end of this level is presented in tables 6 and 7 for the first and second machines, respectively. The cost components of the initial schedule

Part type no.	ts_{p2}	tm_{p2}	MI_{p1}	MI_{p2}	PI_{pm}^*
2	7.08	104.6	-2.114	-2.992	0.0181
4	5.82	58.2	-2.048	-2.809	0.0161
8	4.72	64.1	-2.264	-2.939	0.150
10	5.19	66.3	-2.842	-3.467	0.0154

Table 5. Example iteration of the proposed algorithm.

Part type no.	ts_{p1}	tm_{p1}	Completion time	Due date	Tardiness
3	6.31	51.3	57.61	65	0
5	3.57	36.8	97.98	177	0
6	5.08	73.2	176.26	181	0
2	8.20	104.52	288.98	170	118.98
8	2.63	64.05	355.66	92	263.66

Table 6. Schedule on the first machine.

Part type no.	ts_{p1}	tm_{p1}	Completion time	Due date	Tardiness
7	7.48	65.8	73.28	67	6.28
1	3.11	48.9	125.29	41	84.29
9	5.65	94.4	225.34	40	185.34
4	6.94	58.2	290.48	111	179.48
10	9.33	66.3	366.11	49	317.11

Table 7. Schedule on the second machine.

are calculated as follows. Total operational cost is the multiplication of the sum of the completion times of all part types on each machine by C_o , hence total operational cost = $0.5 \cdot (355.66 + 366.11) = 363.38$. Total tardiness cost is the weighted sum of the tardiness of the part types, which is equal to \$1556.31. In order to find the tooling cost, tools are closely monitored during the scheduling. Each time a tool is removed from the magazine, it is checked whether it is completely worn out or not. Thus, this cost component shows the exact tooling cost incurred in the system. The total tool usage amounts for each tool type are (1, 3.2), (4, 18.99), (5, 3.47), (7, 4.12), (8, 1.64), (9, 2), and (10, 7.48). Hence, the total tooling cost is \$38.55. If we sum up these three cost components then the total production cost is equal to \$1955.6.

Before going to the next section to find the final schedule, we will solve the same problem using the other algorithms. Their final schedules and corresponding cost values are shown in table 8. As it can be seen from these results, the initial schedule reduces total production cost nearly by 30% over the second best algorithm. Another interesting result is that all the cost terms of LPT-II are less than LPT-I, although they give the same sequence, since LPT-I does not consider tool sharing.

5.3. Final schedule

In the last level of our algorithm, we apply a left shift procedure to decrease the total production cost as much as possible. As an example, we will demonstrate how

Algorithm	M/C1	M/C2	Operational	Tardiness	Tooling	Total
LPT-I	2-7-8-3-1	9-6-10-4-5	371.63	3014.53	46.98	3433.14
LPT-II	2-7-8-3-1	9-6-10-4-5	360.25	2964.73	39.82	3364.80
ARM	9-4-2-7	3-1-10-8-6-5	359.40	2793.58	39.06	3192.04
APS	2-6-7-5-8	9-10-1-3-4	359.67	2788.32	38.89	3186.88
KTNS-CN	1-3-4-5-8-9	10-2-6-7	362.43	2348.09	38.44	2748.96

Table 8. Results of the other algorithms.

the efficient frontier is found for the second operation of the seventh part type, i.e. (7,2). Operational related parameters are $SF_{72} = 69$, $d_{72} = 0.179$, $D_{72} = 2.21$ and $L_{72} = 6.1$. The first extreme point $(v_1, f_1) = (340, 0.011)$ minimizes the total manufacturing cost, consisting of the machining, non-machining and tooling costs that is equal to $(0.157 + 0.040 + 0.093) = \0.290 . However, the total manufacturing time equals to 0.394 minute. The second extreme point will be the (v, f) pair that minimizes the total manufacturing time. (v_2, f_2) , being a candidate, is the intersection point of the two constraints that define the feasible region. The (v_2, f_2) pair for this operation is (480, 0.018), which results in the total manufacturing time of 0.368 min. Furthermore, we find another (v, f) pair that minimizes the total manufacturing time on the surface roughness constraint, which is $(v_3, f_3) = (420, 0.015)$. Since (v_3, f_3) gives a lower manufacturing cost of \$0.349 and time of 0.339 minute than (v_2, f_2) , the intersection point (v_2, f_2) is a dominated solution. Therefore, the efficient frontier for this operation is from (v_1, f_1) to (v_3, f_3) in the closed form of $SF_{pi} = C_s v_{pi}^g f_{pi}^h d_{pi}^l$, since $v_2 > v_3$, $b/c = 0.80/0.75 = 1.07 > 1$ and $(\alpha - 1) / (\beta - 1) = (3.7 - 1) / (1.28 - 1) = 9.64 > 1.07$, and any (v, f) pair on this curve is a non-dominated solution as seen in figure 4.

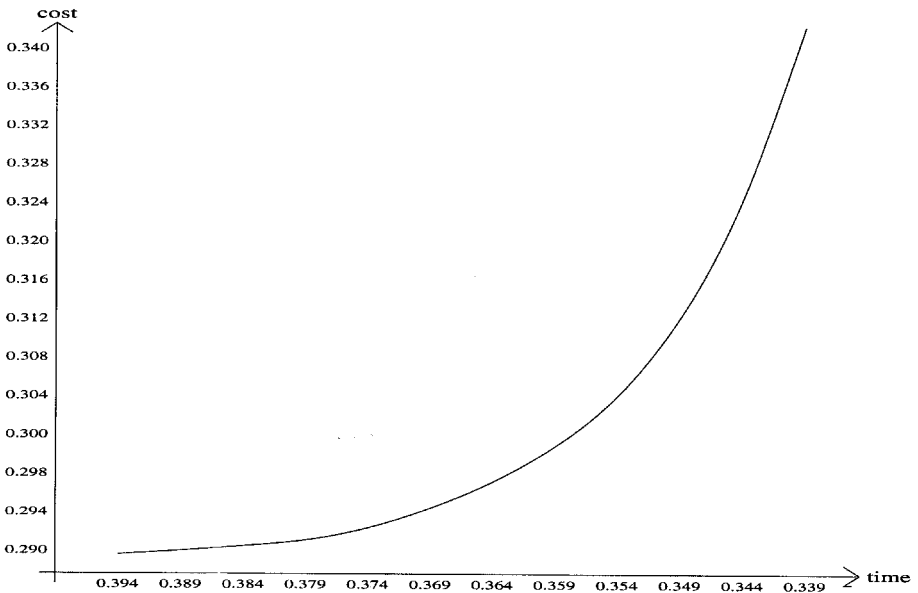


Figure 4. Efficient frontier of operation (7,2).

We perform a piecewise linearization of the efficient frontier after dividing it into two pieces for this operation, and calculate the TI_{pis} index for each piece. For the first piece; $\Delta TC = 0.307 - 0.290 = 0.017$, $\Delta t = 0.394 - 0.350 = 0.044$, $\sum_k w_k = 9$, and $TI_{721} = ((0.017/0.044)/9) = 0.043$. For the second piece; $\Delta TC = 0.339 - 0.307 = 0.032$, $\Delta t = 0.350 - 0.349 = 0.001$, $\sum_k w_k = 9$, and $TI_{722} = ((0.032/0.001)/9) = 3.56 > 1$. Therefore crashing the second piece will not be beneficial, since its associated cost is higher than its gain. As this operation has the smallest index, we crash its processing time after checking if there is enough slack of the required tool. At the end of the second level, the total usage of tool type 4 = 18.99. Therefore, the slack of tool type 4 = $22 - 18.99 = 3.01$ tools. As the cutting speed of the operation increases, the usage rate of the tool also increases. Therefore, the usage rate of tool type 4 is increased from 0.1 to 0.143. Since the batch size of part type 7 is 10, at least 0.43 units of additional tool is required. Hence, there is enough slack to decrease the manufacturing time. This operation was a part of the gathered operation and its tool usage = 0.299 under the new conditions, so these operations can still share the same tool. We then left shift all the parts on the second machine, and the total operational, tardiness and tooling costs become \$359.51, \$1550.03 and \$38.59, respectively.

We continue crashing the processing times until either there is no more beneficial operation, i.e. there is no operation with index < 1 , or there is not enough slack to fill the additional tool usage requirement due to the increased cutting speed. If there is not enough slack of the primary tool for an operation and an alternative tool can process the operation in less time than the primary tool, then we split the batch so that as many parts as possible are processed by the primary tool. We also check tool availability for the alternative tool. If alternative tool is also not available then our algorithm terminates since no more beneficial alternative remains. Let us demonstrate alternative tool usage on an example. As we continue to crash the processing times of the operations whose TI_{pis} values are smaller than 1, the operation (9,1) becomes the most beneficial operation with an TI_{911} index value of 0.76. This operation requires tool type 5 and the increase in the cutting speed increases the tool usage for a single part from 0.05 to 0.09. As the batch size is 20 units, we need additional 0.8 units of tool 5, which is $> 4 - 3.467 = 0.533$ units of remaining available tool life at the end of the second level. Therefore, it is not possible to process all of the parts with the new machining conditions since there is not enough tool to fill the additional requirement. The alternative tool for this operation is tool type 7 that performs the operation in 0.36 minutes, which is less than the primary tool manufacturing time, although it gives a larger total manufacturing cost. Hence, we can use this tool instead of the primary tool at least for some portion of the batch, so we calculate the index ATI_{pi} , given in Section 3.3. For operation (9,1), $TC_{alt} = 0.336$, $TC_{ini} = 0.274$, $t_{alt} = 0.36$, $t_{ini} = 0.38$, $\sum_k w_k = 4$, and $ATI_{91} = (\{(0.336 - 0.274)/(0.38 - 0.36)\}/4) = 0.825 < 1$, hence it is beneficial to use it. Since alternative tool usage is more expensive than using the primary tool, we find the number of parts that can be manufactured by the primary tool as: $\lfloor \frac{1}{0.09} \rfloor + \lfloor \frac{0.533}{0.09} \rfloor = 16$ parts. Thus, the remaining four parts will be manufactured by the alternative tool 7. We also check if the alternative tool has enough slack to fill the additional requirement before going to the left shift procedure. The usage rate for a single operation is 0.12 and a total of 0.48 units of tool 7 is required. The slack of tool 7 = 0.88, so that alternative tool can be used.

After performing all the crashing alternatives, the final values of total operational, tardiness and tooling costs are \$349.30, \$1380.76 and \$42.33, respectively, hence the total production cost is equal to \$1772.39. As a result of this final schedule, the total cost is reduced nearly by 10% over the initial schedule and by 36% over the second best algorithm of KTNS-CN.

6. Conclusions

We have studied the tool management and part scheduling with controllable processing times problems in flexible manufacturing systems. Most of the existing studies in the literature solve scheduling problems by using fixed and predetermined processing times passed from the CNC machine level, although the great sensitivity of scheduling models to timing data is well known. There is a strong interaction between scheduling and tool management decisions, and ignoring these interactions may lead to suboptimal, or even infeasible results, at the system level. We propose a new multistage approach that solves these interrelated problems simultaneously to minimize total production cost. We have also shown that there is a trade-off between the total manufacturing time and the total manufacturing cost, and derived a closed form expression for the efficient frontier of each manufacturing operation explicitly. Our computational experiments indicate that the use of proposed approach offers substantial cost savings over the traditional approach of solving these problems separately. The magnitude of savings is dependent on the system parameters.

Finally, there are several future research directions for this study. We assume that the FMS was composed of identical machines. The study can be enlarged to include non-identical machines with different machine powers or tool magazine capacities. This study can be incorporated into a larger system level study that includes the limitations of an integrated material handling system, such as automated guided vehicle systems for part delivery.

Acknowledgement

The authors thank an anonymous referee for constructive comments.

References

- AKTURK, M. S. and AVCI, S., 1996, Tool allocation and machining conditions optimization for CNC machines. *European Journal of Operational Research*, **94**, 335–348.
- ASKIN, R. G. and STANDRIDGE, C. R., 1993, *Modeling and Analysis of Manufacturing Systems* (Chichester: Wiley).
- CRAMA, Y., 1997, Combinatorial optimization problems for production scheduling in automated manufacturing systems. *European Journal of Operational Research*, **99**, 136–153.
- CRAMA, Y. and KLUVERT J., 1999, Worst case performance of approximation algorithms for tool management problems. *Naval Research Logistics*, **46**, 445–462.
- GORZYCA, F. E., 1987, *Application of Metal Cutting Theory* (Industrial Press).
- GRAY, A. E., SEIDMANN, A. and STECKE, K. E., 1993, A synthesis of decision models for tool management in automated manufacturing. *Management Science*, **39**, 549–567.
- HITOMI, K., 1989, Analysis of optimal machining speeds for automatic manufacturing. *International Journal of Production Research*, **27**, 1685–1691.
- KIM, Y. D. and YANO, C. A., 1993, Heuristic approaches for loading problems in flexible manufacturing systems. *IIE Transactions*, **25**, 26–39.
- KOUEVELIS, P., 1991, An optimal tool selection procedure for the initial design phase of a flexible manufacturing system. *European Journal of Operational Research*, **55**, 201–210.
- LAMOND, B. F. and SODHI, M. S., 1997, Using tool life models to minimize processing time on a flexible machine. *IIE Transactions*, **29**, 611–621.

- MACCHIAROLI, R. and RIEMMA, S., 1996, Design of a tool management system in a flexible cell. *International Journal of Production Research*, **34**, 767–784.
- MUKHOPADHYAY, S. K. and SAHU, S. K., 1996, Priority-based tool allocation in a flexible manufacturing system. *International Journal of Production Research*, **34**, 1995–2018.
- NOWICKI, E. and ZDRZALKA, S., 1990, A survey of results for sequencing problems with controllable processing times. *Discrete Applied Mathematics*, **26**, 271–287.
- SARIN, S. C. and CHEN, C. S., 1987, The machine loading and tool allocation problem in a flexible manufacturing system. *International Journal of Production Research*, **25**, 1081–1094.
- SCHWEITZER, P. J. and SEIDMANN, A., 1991, Optimizing processing rates for flexible manufacturing systems. *Management Science*, **37**, 454–466.
- SCHWEITZER, P. J., SEIDMANN, A. and GOES, P. B., 1991, Performance management in flexible manufacturing systems. *International Journal of Flexible Manufacturing Systems*, **4**, 17–50.
- TANG, C. S. and DENARDO, E. V., 1988, Models arising from a flexible manufacturing machine, Part I: minimization of the number of tool switches. *Operations Research*, **36**, 767–777.
- TETZLAFF, U. A. W., 1996, A queueing network model for flexible manufacturing systems with tool management. *IIE Transactions*, **28**, 309–317.
- VAN WASSENHOVE, L. N. and BAKER, K. R., 1982, A bicriterion approach to time/cost trade-offs in sequencing. *European Journal of Operational Research*, **11**, 48–54.
- VEERAMANI, D., UPTON, D. M. and BARASH, M. M., 1992, Cutting tool management in computer integrated manufacturing. *International Journal of Flexible Manufacturing Systems*, **4**, 237–265.
- VICKSON, R. G., 1980, Two single machine sequencing problems involving controllable job processing times. *AIIE Transactions*, **12**, 258–262.