



Due date and cost-based FMS loading, scheduling and tool management

Ayten Turkcan , M. Selim Akturk & Robert H. Storer

To cite this article: Ayten Turkcan , M. Selim Akturk & Robert H. Storer (2007) Due date and cost-based FMS loading, scheduling and tool management, International Journal of Production Research, 45:5, 1183-1213, DOI: [10.1080/00207540600559955](https://doi.org/10.1080/00207540600559955)

To link to this article: <http://dx.doi.org/10.1080/00207540600559955>



Published online: 22 Dec 2006.



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



Article views: 137



View related articles [↗](#)



Citing articles: 16 View citing articles [↗](#)

Due date and cost-based FMS loading, scheduling and tool management

AYTEN TURKCAN*†, M. SELIM AKTURK‡
and ROBERT H. STORER§

†Department of Industrial Engineering, Middle East Technical University,
06531 Ankara, Turkey

‡Department of Industrial Engineering, Bilkent University, 06800 Ankara, Turkey

§Department of Industrial and Systems Engineering, Lehigh University,
Bethlehem, PA 18015, USA

(Revision received October 2005)

In this study, we consider flexible manufacturing system loading, scheduling and tool management problems simultaneously. Our aim is to determine relevant tool management decisions, which are machining conditions selection and tool allocation, and to load and schedule parts on non-identical parallel CNC machines. The dual objectives are minimization of the manufacturing cost and total weighted tardiness. The manufacturing cost is comprised of machining and tooling costs (which are affected by machining conditions) and non-machining cost (which is affected by tool replacement decisions). We used both sequential and simultaneous approaches to solve our problem to show the superiority of the simultaneous approach. The proposed heuristics are used in a problem space genetic algorithm in order to generate a series of approximately efficient solutions.

Keywords: Tool management; Scheduling; Flexible manufacturing systems; Loading

1. Introduction

The nature of demand is changing in today's industrial world in that customers are looking for a large variety of products. In order to meet varying customer demands, firms should be flexible enough to produce parts in an efficient way. Flexible manufacturing systems have emerged with progress in manufacturing technology. A flexible manufacturing system (FMS) is a computer-controlled production system consisting of numerically controlled machines and an automated material handling system. Since the investment and operating costs of FMSs are very high, operation planning, scheduling and control activities should be performed efficiently.

In this study, we will consider FMS loading, scheduling and tool management problems, which form different levels of the production management

*Corresponding author. Email: ayten@ie.metu.edu.tr

hierarchy, simultaneously. The tool management problem at the tool level is the determination of the tool types to perform the operations and selection of machining conditions such as cutting speed and feed rate. In most existing studies, the tool management problem at the tool level is considered as an FMS design problem that should be solved before the planning and scheduling decisions are made. The next level in the hierarchy is the FMS loading problem. Stecke (1985) defined the FMS loading problem as the allocation of operations and the required tools for the parts among the machines subject to the technological and capacity constraints of the FMS. FMS scheduling determines the optimal input sequence of parts and the schedule of parts and tools on each machine. At the machine level, tool allocation and replacement decisions are the key tool management issues. Grieco *et al.* (2001) provide an extensive survey of the FMS loading problem and emphasize the importance of the interaction between different levels of the production management hierarchy in the applicability of the loading models to real-case situations.

Most of the existing studies in the literature solve the FMS loading, scheduling and tool management problems independently or sequentially due to the complexity of the overall problem (Rachamadugu and Stecke 1994). The FMS loading problem has received considerable attention from researchers. The survey paper by Grieco *et al.* (2001) provides insights into the areas that are not completely covered by the existing FMS loading methods at each component and gives directions for future research. An important factor that affects the loading problem is the characteristics of the FMS (machines, control system, tools and handling system, and parts, pallets and fixtures) (Grieco *et al.* 2001). In earlier FMSs, tool delivery and loading were not automatic, and the time spent on tool loading and replacement was significantly high. In order to reduce the time spent on tool loading and replacement, a batching approach was used as the tool management strategy. According to this approach, a batch of parts is determined, the required tools are loaded on the tool magazines, and the parts that are allocated to the machines are processed without replacing any tools. After all the parts in the batch have been processed, the tool magazine is loaded with the new tools required for the next batch. The scheduling problem is solved for each batch, after the parts have been grouped and loaded to the machines.

Recent advances in technology, such as automated material handling and tool loading, reduce the tool magazine size limitations and reduce the need for making loading decisions in advance of scheduling decisions (Rachamadugu and Stecke 1994). Because the material handling and tool loading operations are done automatically, the time spent on loading and replacing tools is reduced significantly. The machines with high capabilities decrease the machining times and hence increase the need for more frequent tool changes. The batching approach, which is the most commonly used tool management strategy, is inefficient for most systems with the current technology, especially when one considers due date and completion time related performance measures (Amoako-Gyampah 1994). The flexible approach, in which tools are replaced when necessary, is better for overall system performance. There are a few studies considering flexible tool management approaches (Amoako-Gyampah 1994, Roh and Kim 1997). These studies solve the FMS loading and scheduling problems simultaneously and use dispatching rules to solve these problems. However, the tool replacement decisions are made after the parts have been loaded and sequenced on each machine. The lack

of appropriate scheduling algorithms to manage the flow of the tools limits the exploitation of automatic tool transport systems.

The tool replacement decisions are very critical in determining non-machining times in flexible tool management approaches. The non-machining times, which are required to load, change and replace the tools, are ignored in most of the existing studies. Some studies consider the non-machining times as sequence-dependent setup times. However, one cannot realistically assume that the non-machining times are predetermined sequence-dependent setup times. The non-machining times depend on the current status of the tool magazine, which is in turn determined by the limited tool lives, the limited capacity of the tool magazine and all scheduling and tool allocation decisions made up to the current time. Some studies use average non-machining times while giving loading and scheduling decisions (Roh and Kim 1997, Akturk and Ozkan 2001, Fathi and Barnette 2002). The contribution of non-machining time could be significant for the multiple operation case. Therefore, the exact determination of the non-machining times is important. The over- or underestimation of the non-machining times will lead to suboptimal solutions for the overall problem. For a further discussion on tool magazine arrangement, we refer to Baykasoglu and Dereli (2004).

In most of the existing studies, tools are changed due to part mix. The limited lives of the tools, which depend on the workpiece material and cutting conditions, are not considered. However, as stated by Gray *et al.* (1993), tools are changed ten times more often due to wear than due to part mix. When the tool wears out, it needs replacement or reconditioning. The reconditioning operation is performed in the tool room, which does not normally work during an unpersonned shift. Since an FMS can work throughout all shifts, the availability of the tools during the unpersonned shifts is important. Therefore, the problem of tool life management is important to fully exploit the potential production capacity of the FMS (Grieco *et al.* 2001). In the existing literature, there are few studies that consider limited tool lives and tool change due to wear. Sarin and Chen (1987) solve machine loading and tool allocation problems in order to minimize machining cost comprised of tooling cost due to wear and machine usage cost. Akturk and Avci (1996) proposed a new method for determining the optimal machining conditions and tool allocation decisions with the objective of minimizing the sum of machining, non-machining, tooling and tool waste costs in a single machine environment. In this study, we consider the limited lives of the tools, which are affected by the machining conditions selected for each operation.

According to Grieco *et al.* (2001), most of the articles use an objective function that is not directly associated with the goals of the firm such as workload balancing among the machining centres, or minimization of the number of tool movements and changes. Although due date based objectives are very important for internal and external customer satisfaction, they are used only in a few studies. Roh and Kim (1997) consider the minimization of the total tardiness objective for solving loading and scheduling problems. Akturk and Ozkan (2001) proposed a multistage algorithm for solving the identical parallel machine scheduling problem with the objective of minimizing the sum of tooling, operational and tardiness costs. Bernardo and Lin (1994) consider the non-identical parallel machine scheduling problem with the objectives of minimizing total tardiness and setup costs.

In this study, we have two objectives: minimizing manufacturing cost (which is important for the manufacturer) and minimizing total weighted tardiness incorporating both part priorities and due dates (which is important for the customer).

This study considers most of the issues that are addressed as future research areas by Grieco *et al.* (2001). The proposed algorithm considers the interaction between the levels of the planning hierarchy by considering the loading problem simultaneously with the scheduling and tool allocation and replacement problems, which are lower level problems. The requests of the higher levels are considered by incorporating the due dates of the parts. Also, the machining conditions selection problem, which is solved at higher levels without considering the scheduling objectives, considers both the manufacturing cost and total weighted tardiness objectives in this study. The proposed algorithm considers alternative tools for the operations in order to increase the flexibility of the FMSs. The consideration of the limited tool lives is important to fully exploit the capacity of the FMSs, especially during the unpersonned shifts. The algorithm can easily incorporate tool sharing when a limited number of expensive tools exist. When the proposed simultaneous approach is incorporated into a control system, highly flexible FMSs can be achieved. In order to solve the problem, we propose a problem space genetic algorithm (PSGA) to find approximately efficient solutions, which provide alternative solutions to the decision maker (DM).

In section 2 the problem is defined with its underlying assumptions, and the mathematical formulation of the problem is given. The proposed PSGA is explained in section 3. The simultaneous algorithm which is used within PSGA is explained in section 4. In section 5, we propose a sequential algorithm in order to compare its performance with the proposed PSGA on a set of randomly generated problems as discussed in section 6. In the last section, some concluding remarks are provided.

2. Problem definition

A diagram of the production environment we consider in this study can be seen in figure 1. Since most of the new FMSs consist of parallel machines (Grieco *et al.* 2001), we consider a non-identical parallel CNC machine environment. The CNC turning machines are non-identical, because each CNC machine can have different tool magazine capacities, horsepower and tool change times and, hence, different operating costs. However, the machines are interchangeable. Thus, they can perform the given set of cutting operations when the required tools are loaded on their tool magazine. A machine can process one part at a time. Each part has a priority, which shows the importance of the part relative to the other parts. The individual parts have distinct due dates, because they are assumed to be produced for internal customers. The due dates do not include time windows. Since the parts that are processed on the CNC turning machines might go to other work centres for other operations such as milling, drilling and/or assembly, the production schedules of the succeeding work centres might impose distinct due dates for each part type. Also, when a master production plan, which is determined according to the customer demand, capacity of the work centres and due dates of the orders, is exploded through the bill of the materials, each part

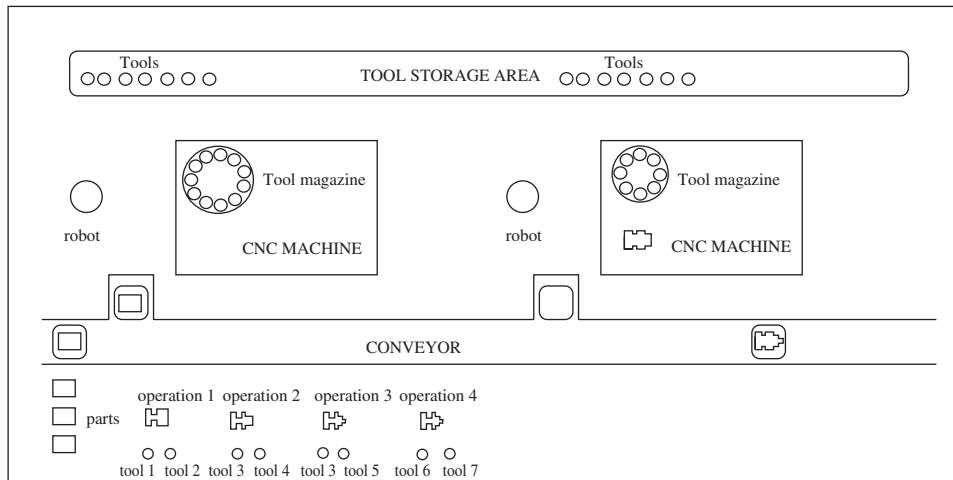


Figure 1. Production environment.

of the same type might require a distinct due date. The parts have multiple operations that should be performed on CNC turning machines. An ‘operation’ is defined as any cutting activity requiring a different type of tool or different processing requirements such as different diameter, depth or length of cut. There is a precedence relationship between the operations of each part. Each operation should be processed in a certain order. Changes in the operation sequence or integration of two or more operations are not possible. In this study, we assume that all the operations of a part should be performed on the same machine, since (in the existing CNC technology) the tool change times are significantly shorter than part loading and unloading times. This strategy, which is denoted as the tool movement policy, avoids the repositioning and re-setup of the parts, and, hence, decreases the total processing times and the processing costs (Mukhopadhyay and Sahu 1996).

In theory, there are many alternative ways to machine a part. The operations can be performed with different machining times or alternative tools. In most of the existing studies, only a single tool alternative giving the minimum manufacturing cost is selected to perform the corresponding operation. Consideration of alternative ways for performing the operations could allow a better exploitation of resources, as will be shown in this study. Since the processing of an operation cannot be interrupted for a tool change due to surface finish requirements, each operation should be processed with a single tool that has enough remaining life. Only one tool can be replaced at a time. This implies that tool changing times are additive. Since the tool magazines are integrated parts of the machines, the tools cannot be replaced while the machine is processing a part. The unassigned tools are kept in a central tool storage location. The tools are transferred between the tool storage area and the tool magazines of the machines by a robotic manipulator. Under these assumptions, we will determine the relevant tool management decisions (which are machining conditions selection and tool allocation), and loading and scheduling decisions, with the objectives of minimizing

the manufacturing cost and minimizing total weighted tardiness. The notation used is given in tables 1 and 2.

We first propose a mathematical programming (MP) model of the problem. The bi-criteria objectives of the minimization of the manufacturing cost, f^m , and the total weighted tardiness, f^t , can be written as follows:

$$\begin{aligned} \min f^m &= \sum_p \sum_i \sum_j \sum_m (C_{o_m} t_{pijm}^m + Ct_j U_{pijm} + C_{o_m} t_{pijm}^{nm}) Z_{pijm}, \\ \min f^t &= \sum_p \sum_s \sum_m w_p T_{psm} \\ &= \sum_p \sum_s \sum_m w_p \left(\max \left\{ 0, \sum_r \sum_{s'=1}^{s-1} (t_{rm}^p X_{rs'm} + t_{pm}^p - DD_p) X_{psm} \right\} \right), \end{aligned}$$

where

$$\begin{aligned} t_{pijm}^m &= \frac{\pi D_{pi} L_{pi}}{12 v_{pijm} f_{pijm}}, & U_{pijm} &= \frac{\pi D_{pi} L_{pi} (d_{pi})^{\gamma_j}}{12 C_j (v_{pijm})^{(1-\alpha_j)} (f_{pijm})^{(1-\beta_j)}}, \\ t_{pm}^p &= \sum_i \sum_j (t_{pijm}^m + t_{pijm}^{nm}) Z_{pijm}, \\ t_{pijm}^{nm} &= g(U_{pijm}, R_{jm}, TS_m, TH_m) \\ &= \begin{cases} 0, & \text{if } (j = TH_m) \wedge (U_{pijm} \leq R_{jm}), \\ tc_{jm}, & \text{if } (j \in TS_m) \wedge (U_{pijm} \leq R_{jm}), \\ tc_{jm} + tl_{jm}, & \text{if } ((j \in TS_m) \wedge (U_{pijm} > R_{jm}) \wedge (|TS_m| \neq TM_m)) \\ & \vee ((j \notin TS_m) \wedge (|TS_m| \neq TM_m)), \\ tc_{jm} + tr_{jm}, & \text{if } ((j \in TS_m) \wedge (U_{pijm} > R_{jm}) \wedge (|TS_m| = TM_m)) \\ & \vee ((j \notin TS_m) \wedge (|TS_m| = TM_m)). \end{cases} \end{aligned}$$

The first term in the manufacturing cost is the machining cost, which is incurred for the time spent to complete a metal cutting operation. The machining time is a function of the cutting speed, v_{pijm} , and the feed rate, f_{pijm} . As the cutting speed and the feed rate increase, the machining time decreases. The second term is the tooling cost. It is related to the tool usage rate, which is the ratio of the machining time to tool life. The tool usage rate decreases as the cutting speed and feed rate increase. The relationship between the machining conditions and the expected tool life is approximated by Taylor's tool life formula as discussed in Groover (2002). The third term is the non-machining cost that is incurred for replacing and loading tools. The non-machining cost depends on the current status of the tool magazine. The tool changing time, tc_{jm} , occurs when the tool currently loaded in the machine is not appropriate for the operation, and the required tool is already stored in the tool magazine. A tool loading time, tl_{jm} , is added to the non-machining time when the required tool is not in the tool magazine and a free slot exists on the tool magazine. A tool replacement time, tr_{jm} , occurs when there is no free slot for the required tool. In this case, a tool from the tool magazine should be removed in order to load the required tool. The second objective is the minimization of the total

Table 1. Parameters.

$\alpha_j, \beta_j, \gamma_j$	Speed, feed, depth of cut exponents for tool j
C_m, b, c, e	Specific coefficient and exponents of the machine power constraint
C_s, g, h, l	Specific coefficient and exponents of the surface roughness constraint
C_j	Taylor's tool life constant for tool j
d_{pi}	Depth of cut for operation i of part p (in)
D_{pi}	Diameter of the generated surface for operation i of part p (in)
L_{pi}	Length of the generated surface for operation i of part p (in)
HP_m	Maximum available machine power of machine m (hp)
SF_{pi}	Maximum allowable surface roughness for operation i of part p (μ in)
DD_p	Due date of part p
w_p	Weight of part p
Co_m	Operating cost of machine m (\$/min)
Ct_j	Cost of tool j (\$/tool)
tc_{jm}	Tool interchange time of tool j with the required tool for the next operation in machine m
tl_{jm}	Time required to take a single tool j from central tool storage and load on machine m when there is a free slot on the tool magazine
tr_{jm}	Tool replacing time of worn tool j with a new tool from central tool storage to machine m
TM_m	Tool magazine capacity of machine m
O_p	Operation set of part p
u	Weight of first objective (minimization of manufacturing cost) ($0 \leq u \leq 1$)

Table 2. Decision variables.

v_{pijm}	Cutting speed for operation i of part p using tool j on machine m (fpm)
f_{pijm}	Feed rate for operation i of part p using tool j on machine m (ipr)
Z_{pijm}	Binary variable which is equal to 1 if operation i of part p is assigned to machine m and uses tool j
U_{pijm}	Tool usage rate of operation i of part p using tool j on machine m
R_{jm}	The ratio of remaining tool life of tool j on machine m to the tool life of a new tool
TS_m	Set of tools on the tool magazine of machine m
TH_m	Type of tool on the tool holder of machine m
t_{pijm}^{nm}	Non-machining time of operation i of part p using tool j on machine m
t_{pijm}^m	Machining time of operation i of part p using tool j on machine m
f_{pm}^m, f_{pm}^t	Manufacturing cost and total weighted tardiness of part p on machine m
t_{pm}^p	Sum of machining and non-machining times of all operations of part p on machine m
T_{psm}	Tardiness value of part p when it is scheduled at sequence position s on machine m
X_{psm}	Binary variable which is equal to 1 when part p is scheduled at sequence position s on machine m

weighted tardiness, which depends on the starting time of a part, the machining and non-machining times of each operation, and the due date of the corresponding part. If a part is assigned to sequence position s on machine m , the tardiness is calculated by using the total processing times of the jobs scheduled up to s , and the processing time and due date of the corresponding part. The processing

time of a part is the sum of machining and non-machining times of all operations for the corresponding part. These two objectives usually conflict with each other. We can decrease the machining time, and hence machining cost, by increasing the cutting speed and feed rate. But, this will increase the tooling and non-machining costs. The total weighted tardiness increases or decreases according to changes in the sum of machining and non-machining times. In this study, we aim to find efficient solutions which provide alternatives to the decision maker (DM). A solution S is inefficient if there exists another solution S' such that $f^i(S') \leq f^i(S)$, $\forall i$ and $f^i(S') < f^i(S)$ for at least one i . If there is no solution like S' , then S is called an efficient solution.

In the proposed MP model, the first set of constraints represents the machining conditions selection constraints imposed on v_{pijm} and f_{pijm} . In order to find machining conditions for each operation–tool pair, we use the tool life, machine power and surface roughness constraints of the geometric programming model proposed by Akturk and Avci (1996). These constraints are as follows:

$$\left(\frac{\pi D_{pi} L_{pi} (d_{pi})^{\gamma_j}}{12 C_j}\right) (v_{pijm})^{(\alpha_j-1)} (f_{pijm})^{(\beta_j-1)} \leq 1, \quad \forall p, i, j, m, \quad (1)$$

$$\left(\frac{C_m (d_{pi})^e}{HP_m}\right) (v_{pijm})^b (f_{pijm})^c \leq 1, \quad \forall p, i, j, m, \quad (2)$$

$$\left(\frac{C_s (d_{pi})^f}{SF_{pi}}\right) (v_{pijm})^g (f_{pijm})^h \leq 1, \quad \forall p, i, j, m, \quad (3)$$

$$v_{pijm}, f_{pijm} > 0, \quad \forall p, i, j, m. \quad (4)$$

The first constraint is the tool life constraint. In order to perform each operation with a single tool, the tool usage rate should not exceed the available tool life. The second constraint (machine power constraint) guarantees the feasibility of the cut according to the machine's capacity. The third constraint is the surface roughness constraint which is necessary for quality requirements. The decision variables are the cutting speed and feed rate. The specific constants for the tools, which change according to the tool type and the parts' material, can be obtained from machining handbooks.

The second set of constraints is the tool allocation, part loading and scheduling constraints, which can be written as

$$\sum_s \sum_m X_{psm} = 1, \quad \forall p, \quad (5)$$

$$\sum_p X_{psm} \leq 1, \quad \forall m, s, \quad (6)$$

$$\sum_j \sum_m Z_{pijm} = 1, \quad \forall p, i, \quad (7)$$

$$\sum_j Z_{pijm} \leq \sum_j Z_{p,i+1,j,m}, \quad \forall p, i, m, \quad (8)$$

$$\sum_i \sum_j Z_{pijm} = |O_p| \sum_s X_{psm}, \quad \forall p, m, \tag{9}$$

$$X_{psm} \in \{0, 1\}, \quad \forall p, s, m \quad \text{and} \quad Z_{pijm} \in \{0, 1\}, \quad \forall p, i, j, m. \tag{10}$$

According to the fifth constraint, each part should be assigned to a single machine and a sequence position. Each sequence position on each machine should be occupied by at most one part, which is dictated by constraint (6). Constraint (7) guarantees that each operation of a part should be performed by a single tool alternative on a single machine. Constraints (8) and (9) state that all operations of a part should be performed on the same machine. The last set is integrality and non-negativity constraints.

The proposed mathematical model is nonlinear due to constraints (1)–(3), and the machining and tooling costs and total weighted tardiness. The non-machining cost, which depends on the current status of the tool magazine, is difficult to calculate at the beginning of the planning horizon since it depends on the previous tool allocation, and part loading and sequencing decisions. The overall problem we consider is quite complex and thus we develop heuristics to solve it as explained in the following sections.

3. Problem space genetic algorithm

Problem space search was proposed by Storer *et al.* (1992) and used successfully in solving different problems such as the scheduling of aircraft landings (Ernst *et al.* 1999), the melt scheduling in a steel manufacturer (Naphade *et al.* 2001) and the scheduling of a single machine with weighted tardiness objective (Avci *et al.* 2003). Problem space search uses a neighbourhood structure in the problem space, instead of a solution space. In problem space search, a constructive ‘problem-specific’ base heuristic, H , maps a problem instance data vector D to a solution sequence S , i.e. $H: D \rightarrow S$. Given any solution S , the objective function $V(S)$ can be calculated. Let δ be the set of perturbation vectors. A perturbation vector is decoded into a sequence by applying the base heuristic to the perturbed data ($H(D + \delta)$) and the objective value is obtained by applying $V(H(D + \delta))$. The optimization problem can be defined as finding the solution with minimum objective function value over all perturbation vectors ($\min_{\delta} V(H(D + \delta))$).

One advantage of using problem space search in this problem is that there is no need for a feasibility check, which will take significant computation time in loading and scheduling of flexible manufacturing systems with tooling constraints in classical local search methods using solution space as the neighbourhood structure. The basic steps of the proposed problem space genetic algorithm (PSGA) are as follows.

- (1) Set the generation number k equal to zero and form the initial perturbation matrix, A^k

$$A^k = \begin{matrix} 1 \\ \vdots \\ P \end{matrix} \begin{bmatrix} \delta_{1,(11)}^k & \cdots & \delta_{1,(pm)}^k & \cdots & \delta_{1,(jm)}^k & \cdots \\ \vdots & & \vdots & & \vdots & \\ \delta_{P,(11)}^k & \cdots & \delta_{P,(pm)}^k & \cdots & \delta_{P,(jm)}^k & \cdots \end{bmatrix},$$

where the values of $\delta_{y,(pm)}^k$ and $\delta_{y,(jm)}^k$ are randomly generated from a uniform distribution UN $[-\theta, \theta]$ and P is the population size. The perturbation values are generated for each part-machine pair, which is used to perturb the part selection index, PS_{pm} (as discussed in section 4), and for each tool-machine pair, which is used to perturb the tool index, TI_{jm} (as discussed in section 4.3).

- (2) For each perturbation vector, $i \in \{1, 2, \dots, P\}$, call the ‘base heuristic’ and solve the problem with perturbed part selection and tool removal indices. Find the total manufacturing cost and total weighted tardiness for each solution.
- (3) Update the non-dominated solution set according to the solutions found in generation k .
- (4) For each perturbation vector, $y \in \{1, 2, \dots, P\}$, calculate the fitness value, which depends on the normalized manufacturing cost ($f^m(y)$) and total weighted tardiness ($f^t(y)$) of the corresponding solution. The fitness value of each encoding is the weighted linear function of the two normalized objectives, which is calculated as follows:

$$\text{fitness}(y) = u \frac{f^m(y) - \min f^m}{\max f^m - \min f^m} + (1 - u) \frac{f^t(y) - \min f^t}{\max f^t - \min f^t}.$$

- (5) Assign selection probabilities to each member of the population according to the fitness values. The probability of selecting encoding y according to its fitness value is

$$\text{Prob}(y) = \frac{(\max_{y'} \text{fitness}(y') - \text{fitness}(y))^{sc}}{\sum_{y'=1}^P (\max_{y''} \text{fitness}(y'') - \text{fitness}(y'))^{sc}},$$

where sc is the selectivity constant.

- (6) Generate P encodings for the next generation by using sexual and asexual reproduction, the elitist approach and mutation operations (see Goldberg (1989) for more detailed information about GAs).
- (7) Go to Step 2 and repeat for a fixed number of generations and number of restarts.

The parameters of the proposed problem space genetic algorithm should be selected carefully in order to achieve a better performance. The elements of the perturbation vectors are generated from a uniform $[-\theta, \theta]$ distribution. The magnitude of the perturbation vectors is thus controlled by the parameter θ . If θ is too small, only a few solutions will be generated (repeatedly), thus the search space will not be rich. As θ approaches infinity, we are essentially generating random solutions. Thus θ must be chosen carefully in order to generate a diverse set of good solutions. As the selectivity constant, sc , increases, better solutions will have a greater chance of being selected. If sc is too large, the population will converge quickly, which is not desirable, since we are trying to find a diversified set of solutions. The algorithm could find better results as the number of generations increases. The number of restarts, NS , affects the diversity of the non-dominated solutions.

The performance of PSGA is very sensitive to the performance of the base heuristic as shown by Avci *et al.* (2003). The performance of PSGA increases with

a good, problem-specific base heuristic. In this study, we propose simultaneous and sequential algorithms that are used in PSGA as base heuristics. The proposed algorithms are explained in the following sections.

4. Simultaneous approach

In problem space search, one needs a problem-specific base heuristic, which maps a point in problem space to a solution. Our aim is to minimize the total manufacturing cost and total weighted tardiness. The machining and tooling costs are functions of the cutting speed and feed rate as explained in section 2. The non-machining cost and the total weighted tardiness are dynamic terms, i.e. they change as the state of the system changes. The status of the tool magazine determines the non-machining time, and hence the non-machining cost and the total weighted tardiness. According to the proposed base heuristic, tool allocation, part loading and scheduling decisions are made. The steps of the base heuristic are as follows.

- *Step 1.* Initialization: $UNS = \{1, 2, \dots, N\}$, $ALT = \{1, 2, \dots, M\}$ and $t_m^{now} = 0$ for all $m \in ALT$, where UNS is the set of unscheduled parts, ALT is the set of altered machines and t_m^{now} is the current time, on machine m .
- *Step 2.* For each part $p \in UNS$ and machine $m \in ALT$, calculate the increase in both manufacturing cost and total weighted tardiness as $f_{pm}^m = \sum_i \sum_j (Co_m t_{pijm}^m + Ct_j U_{pijm} + Co_m t_{pijm}^{nm})$ and $f_{pm}^t = \max\{0, t_m^{now} + \sum_i \sum_j (t_{pijm}^m + t_{pijm}^{nm}) - DD_p\}$, respectively.
- *Step 3.* Calculate the normalized values of the objectives as

$$f'_{pm} = \frac{f_{pm} - \min_{p,m} f_{pm}}{\max_{p,m} f_{pm} - \min_{p,m} f_{pm}}.$$

Find the machine, $m(p)$, giving the minimum weighted linear function of two objectives for each part $p \in UNS$, i.e. $m(p) = \arg \min_{m \in ALT} \times (u \cdot f_{pm}^m + (1 - u) \cdot f_{pm}^t)$.

- *Step 4.* Calculate the following part selection index for each $\{p, m(p)\}$ pair found in Step 3:

$$PS_{p,m(p)} = \frac{w_p}{t_{p,m(p)}^p} \exp \left[- \frac{\max\{0, DD_p - t_{m(p)}^{now} - t_{p,m(p)}^p\}}{k \cdot \bar{t}} \right],$$

where

$$\bar{t} = \frac{\sum_{p \in UNS} t_{p,m(p)}^p}{|UNS|}, \quad t_{p,m(p)}^p = \sum_i \sum_j (t_{p,i,j,m(p)}^m + t_{p,i,j,m(p)}^{nm}),$$

and k is a lookahead parameter. Normalize the part selection index as

$$PS'_{p,m(p)} = \frac{PS_{p,m(p)} - \min_{p,m(p)} \{PS_{p,m(p)}\}}{\max_{p,m(p)} \{PS_{p,m(p)}\} - \min_{p,m(p)} \{PS_{p,m(p)}\}}.$$

Select the $\{p, m(p)\}$ pair giving the maximum $PS'_{p,m(p)} + \delta_{p,m(p)}$, i.e. $\{p^*, m(p^*)\} = \arg \max_{p \in UNS} \{PS'_{p,m(p)} + \delta_{p,m(p)}\}$.

- *Step 5.* Assign part p^* to machine $m(p^*)$. Update the current time on machine $m(p^*)$ such that $t_{m(p^*)}^{\text{now}} = t_{m(p^*)}^{\text{now}} + t_{p^*,m(p^*)}^p$. The remaining life of the tools used by part p^* is updated as $R_{j,m(p^*)} = R_{j,m(p^*)} - U_{p^*,i,j,m(p^*)}$.
- *Step 6.* Update $\text{UNS} = \text{UNS} \setminus \{p^*\}$ and $\text{ALT} = \{m(p^*)\}$. If $\text{UNS} \neq \emptyset$, go to Step 2, else stop.

In the proposed algorithm, the parts are scheduled one at a time since the non-machining times depend on the tool magazine status and cannot be determined easily in advance. The part loading and scheduling, and tool allocation and replacement decisions are given at each decision point considering the current status of the tool magazine. Since the non-machining cost and weighted tardiness are dynamic terms in the objective function, they are recalculated after each assignment (Step 2).

In most of the existing studies considering parallel machine environments, the primary objective is balancing the workload. As the machines become available, a part is selected and loaded to the first available machine. However, balancing the workload might not be a good alternative when the manufacturing cost is considered. In the proposed algorithm, the parts are loaded to the machines that give the minimum increase in both manufacturing cost and total weighted tardiness at the current time. The machine giving the minimum increase is the most appropriate machine for that part at the current time. At some later time, since the tool magazine status changes, another machine may become better for that part. Since the two objectives have different ranges, the objective function values are normalized in order to prevent the dominance of one objective over the other. The weighted linear combination of the normalized objectives is used for determining the chosen machine (Step 3).

After the most suitable machine is determined for each part, a part selection index is used for selecting a part-machine pair (Step 4). The proposed index considers the weights of parts, the slack, and the sum of machining and non-machining times. The index gives higher priority to parts having less slack and shorter weighted processing time. The processing time is taken as the sum of machining and non-machining times. The part selection indices are normalized between zero and one and the perturbation values are added to the normalized indices. The part-machine pair giving the maximum perturbed part index is selected for scheduling. The algorithm continues until all parts are scheduled.

In the proposed base heuristic, there are three important issues: determination of machining conditions, selection of the tool that will be used to process each operation and calculation of non-machining times. These decisions are all implicit in Step 2 of the algorithm above. In the following subsections, we will explain each issue in detail.

4.1 Machining conditions selection

Machining conditions such as the cutting speed and feed rate affect the machining times and tool usage rates. The machining times are the primary inputs to the scheduling problem. As Sodhi *et al.* (2001) stated, the selection of cutting parameters is guided by the minimization of either the processing cost or the processing time. Tool usage rates affect the tool allocation and replacement decisions. In order to find

machining conditions, we use the geometric programming model proposed by Akturk and Avci (1996). The GP model is as follows:

$$\begin{aligned} \min \text{SMOP}_{pijm} &= C_o m^t_{pijm} + C_t j U_{pijm}, \\ \text{s.t. (1), (2), (3), (4)} & \text{ (from the MP model in section 2).} \end{aligned}$$

The model minimizes the sum of machining and tooling costs subject to tool life, machine power and surface roughness constraints. As discussed in a previous study by Turkcan *et al.* (2003), the surface roughness constraint is binding at optimality. When constraint (3) is tight, $C_s(d_{pi})^l(v_{pijm})^g(f_{pijm})^h/SF_{pi} = 1$. The objective function (the sum of machining and tooling costs) becomes a function of machining time such that $\text{SMOP}_{pijm} = C_o m^t_{pijm} + A_{pij}(t_{pijm}^m)^{B_j}$ where $t_{pijm}^m \geq t_{pijm}^{ml}$,

$$B_j = \frac{g(\beta_j - 1) - h(\alpha_j - 1)}{h - g},$$

and

$$A_{pij} = \left[\frac{\pi D_{pi} L_{pi} (d_{pi})^{\gamma_j} C_{tj}}{12 C_j} \right] \left[\frac{C_s(d_{pi})^l}{SF_{pi}} \right]^{(\alpha_j - \beta_j)/(h - g)} \left[\frac{\pi D_{pi} L_{pi}}{12} \right]^{(-B_j)}.$$

The sum of machining and tooling costs is a convex function of machining time. The minimum machining time, t_{pijm}^{ml} , is the point at which surface roughness and machine power or tool life constraints intersect. When non-machining times are ignored, the minimum machining times minimize the total weighted tardiness. Let t_{pijm}^{mu} be the machining time that minimizes SMOP_{pijm} . In this study, since we try to minimize both the manufacturing cost and total weighted tardiness, we select the machining time as the weighted linear combination of t_{pijm}^{mu} and t_{pijm}^{ml} , i.e. $t_{pijm}^m = u \cdot t_{pijm}^{mu} + (1 - u) \cdot t_{pijm}^{ml}$.

4.2 Tool selection

Advances in cutting tool materials and designs will increase the cutting speeds at which the machining can be carried out, consequently reducing the machining cost at the expense of higher initial tooling cost. Therefore, we consider a set of alternative cutting tool types for each machining operation, since no one cutting tool type is best for all purposes. The tool alternative giving the minimum sum of machining and tooling costs is selected as the primary tool in our study. In the first part of our computational study, we assume that all operations are performed with their primary tools. In section 6.1, we consider alternative tooling. As a result, we not only provide solution flexibility in choosing different tool alternatives, but also the capability to assign different processing times to each operation to improve the solution quality.

In the proposed base heuristic at Step 2, both objective function values change when alternative tools are considered. The main problem in calculating the change in objective functions is selecting the tool which will be used for performing the corresponding operation. If the primary tool is on the tool holder or on the tool magazine and the remaining life of that tool is enough to perform the operation, then the primary tool is used. If we cannot find the primary tool in the tool magazine, we then check whether one of the alternative tools exists in the tool magazine.

If at least one of the alternative tools exists in the tool magazine, we calculate the tool selection index, S_{pijm} , for the tools which have enough remaining life. The tool selection index for part p on machine m using tool j for operation i is calculated as follows:

$$S_{pijm} = (C_{o_m} t_{pijm}^m + Ct_j U_{pijm} + C_{o_m} t_{pijm}^{nm})(t_{pijm}^m + t_{pijm}^{nm}).$$

The first term in the index is the manufacturing cost, which is the sum of machining, tooling and non-machining costs. The second term is the total processing time, which is the sum of machining and non-machining times. According to the proposed index, the tool giving the least manufacturing cost and the least total processing time is preferable. The total processing time can be thought of as a lookahead mechanism which increases the priority of the tool giving less processing time, since the completion time of the corresponding operation is important for the weighted tardiness of the remaining unscheduled parts. We calculate the tool selection index for all alternative tools. The tool giving the minimum index is selected for processing the part.

4.3 Non-machining time calculation

According to the proposed base heuristic (PI), the non-machining times should be calculated at each decision point since the tool magazine status changes after each assignment. Therefore, the calculation of non-machining times is a very critical and time-consuming task. In order to decrease the computation time, we consider two alternatives for calculation of non-machining times. The first alternative, denoted as PI-avg, uses average non-machining times. We know that the non-machining time of an operation may take any one of four different values (0, tc_j , $tc_j + tl_j$, $tc_j + tr_j$), which change according to the tool magazine status as explained in section 2. For example, if a part has three operations, there are 3^4 non-machining time alternatives. We can find an average non-machining time by using all possible alternatives, assuming that they occur with equal probabilities, and use that single value throughout the PSGA. However, when we look at the actual non-machining times that are calculated in PSGA using PI, we see that the alternatives have different frequencies. Some non-machining time alternatives rarely occur. Therefore, PSGA using PI, which calculates non-machining times at each decision point, can be used, until we obtain a certain number of observations. After that point, we can use the average non-machining times calculated using the observed values by implicitly assuming that the non-machining times do not fluctuate. The best way to measure variability of the non-machining times is the coefficient of variation (CV) over a number of observations. In our study, we calculate the CV of a part p on machine m after 1000 non-machining time observations. If CV is smaller than 0.75, it is considered as steady and the average of non-machining times is used as the non-machining time of that part for the rest of the iterations. If it is greater than 0.75, the non-machining times are calculated at each decision point until the number of observations becomes 2000. If $CV \leq 0.75$ then we use the average non-machining time. Otherwise, we take 2000 more observations and recalculate the average non-machining times that will be used afterwards.

The second alternative, EXP, uses expected non-machining times. Since the remaining tool lives can be any value between 0 and 1, we assume that the

remaining life of a tool is uniformly distributed over [0.0, 1.0]. If the maximum remaining life of tools loaded on the tool magazine, $\max\{R_{jm}\}$, is less than the usage rate, U_{pjm} , then we have to load the required tool. If there is an empty slot on the tool magazine, the non-machining time is the sum of tool changing and loading times. If there is no empty slot, a tool is replaced. The non-machining time becomes the sum of tool changing and replacing times. If the remaining life of the tool is enough to perform the operation, the non-machining time is only the tool change time. The expected non-machining time for operation i of part p using tool j on machine m is calculated as follows:

$$E(t_{pjm}^{nm}) = \begin{cases} \int_0^{U_{pjm}} (tc_j + tr_j)dU' + \int_{U_{pjm}}^1 tc_j dU' = tr_j U_{pjm} + tc_j, & \text{if there is no empty slot,} \\ \int_0^{U_{pjm}} (tc_j + tl_j)dU' + \int_{U_{pjm}}^1 tc_j dU' = tl_j U_{pjm} + tc_j, & \text{otherwise.} \end{cases}$$

An important issue in non-machining time calculation is choosing which tool to remove when there is no free slot in the tool magazine for the required tool of the current operation. This critical decision affects the non-machining times of the succeeding parts that will be scheduled on that machine. Tang and Denardo (1988) proposed the keep tool needed soon (KTNS) rule for changing the tools on the tool magazine. The tools which are required most by the remaining unscheduled operations are kept on the tool magazine. If a tool that has not enough remaining life to perform any operation is required by most of the operations, it is kept on the tool magazine according to the KTNS rule. This tool will occupy a tool slot on the tool magazine, although it cannot be used for any other operation. In order to use the tool magazine capacity effectively, the remaining tool life and the number of operations that can be performed by the remaining tool life should be considered for determining the tool to be removed. We propose a new tool index, TI, which is calculated as follows for tool j on machine m :

$$TI_{jm} = R_{jm} \sum x_{pjm}.$$

If the remaining life of tool j on machine m , R_{jm} , is high and the number of operations that can be performed with the remaining life, $\sum x_{pjm}$, is also high, then it would be beneficial to keep that tool in the tool magazine. If we remove this tool, the tooling and the non-machining costs are likely to increase due to more frequent tool changes at later steps. The number of operations that can be performed with the remaining life is calculated by considering all unscheduled operations. In a previous study (Turkcan *et al.* 2003), we showed that the proposed tool index, TI, performs better than the KTNS rule.

In the proposed PSGA, the tool index is calculated for each tool loaded on the tool magazine. Then, the tool indices are normalized between zero and one and perturbation values generated to perturb the tool indices ($\delta_{y,(jm)}$) are added to the indices. The tool giving the time minimum perturbed index is removed from the tool magazine.

5. Sequential approach

In this study, we also propose a sequential algorithm (SLC) to solve the part grouping, loading, scheduling and tool management problems sequentially. The only reason that we propose a SLC algorithm is to show the relative merits of the proposed simultaneous algorithms discussed in section 4 over a sequential approach. The SLC algorithm uses the single linkage clustering method to solve the part-grouping problem, such that the parts are grouped into part families and then the part families are loaded on the machines. First, a 0–1 tool–part incidence matrix is formed. Then, the similarities between the parts are calculated according to the total number of tools and the number of common tools required to process the parts. The similarity coefficient is calculated as follows:

$$\delta_{pr} = \frac{|TR_p \cap TR_r|}{|TR_p \cup TR_r|},$$

where TR_p is the set of primary tools required to process all operations of part p . If parts p and r require the same tools for all operations, δ_{pr} will be equal to one. δ_{pr} will be smaller when the number of common tools decreases. The parts with maximum similarity coefficient are combined into a part family if the maximum coefficient is greater than a certain predetermined threshold value, δ^* . The similarity coefficient between the new part family s and all other part families p' is updated as $\delta_{sp'} = \min\{\delta_{p^*p'}, \delta_{r^*p'}\}$. If the grouping continues until the number of part families is equal to the number of machines, the parts that have totally different tooling requirements might be in the same group. The aim in grouping the parts is to reduce the number of tool replacements due to the part mix. After the part families are formed, they are assigned to machines with the objective of balancing the workload, which is the most commonly used objective in loading problems. Although the parts in the same group are assigned to the same machine, they are not considered as a batch that should be processed together since when we consider the part families as batches, the total weighted tardiness might increase significantly.

The second stage is the scheduling of the parts on each machine. Since we consider the total weighted tardiness objective in our study, we use the apparent tardiness cost (ATC) rule (Morton and Pentico 1993) to schedule the parts. The ATC rule, which considers the slack values and expected processing times, is the best performing dispatching algorithm for single machine weighted tardiness problems. At the last stage of multiple-stage procedures, the tool allocation decisions are made. In most of the existing studies, the keep-tool-needed-soon (KTNS) rule is used. In the proposed method, the tool allocation decisions are given while the parts are scheduled according to the ATC rule. A tool removal index, TI, which considers the remaining life of the tool and the number of operations that can be performed with the remaining life, is used for tool replacement.

In the SLC algorithm, we use well-known algorithms from the literature to solve the problems at each stage. Although these algorithms can find good solutions for each subproblem, they might not give a good solution for the overall problem, because they ignore the interaction between the problems and the decisions given at one stage become a limitation for the succeeding stage.

6. Computational study

We performed a computational study in order to test the performance of the proposed PSGA with different base heuristics. The performance of the proposed simultaneous approaches (PI, PI-avg, EXP) is compared with the performance of a sequential approach (SLC), which is the most commonly used approach in the literature.

In the following sections, first the experimental design factors and the parameters used in our computational study are explained. A small numerical example is provided in order to show which integration of subproblems has the main effect on the improvement. The numerical example also shows how a feasible solution is improved and becomes a better solution in the proposed PSGA. The results of the computational study are presented in the third section. The last section shows the effect of using alternative tooling on the performance of the proposed algorithm.

All algorithms are coded in the C language and compiled with Gnu C compiler. The problems are solved on a 400 MHz UltraSPARC station.

6.1 Experimental design

There are four experimental factors that can affect the efficiency of our algorithm. The factor levels can be seen in table 3. The experimental design is a 2^4 full-factorial design. We take five replications for each factor combination, resulting in 80 different randomly generated runs.

The number of operations per part, factor A, affects the size and load of the system. The tooling cost, factor B, affects the ratio of tooling cost to machining cost and hence the total machining and non-machining costs. When factor B is at the low level, the ratio is approximately 15–20%. The ratio is approximately 25–30% at the high level. Factor C, which determines the relative importance weight of parts with respect to each other, affects the total weighted tardiness objective and hence the machine allocation and scheduling decisions. Factor D is the due date tightness factor, which affects the weighted tardiness objective. When due dates are loose, the manufacturing cost objective dominates the weighted tardiness objective, and machines giving less manufacturing cost are selected. When due dates are tight it becomes more difficult to solve the problem, since the trade-off between the objectives increases. The average makespan in the due date tightness factor is calculated by using the expected processing times and is as follows:
 average makespan = $(\sum_p \sum_m (t_{pm}^m + E(t_{pm}^{nm}))) / (M \cdot M)$.

Table 3. Experimental design factors.

Factor	Definition	Low level	High level
A	Number of operations per part	UN ~ [2, 6]	UN ~ [6, 10]
B	Tooling cost (Ct_j)	UN ~ [20, 40]	UN ~ [50, 70]
C	Weight (w_p)	UN ~ [0.10, 0.70]	UN ~ [0.30, 0.50]
D	Due date tightness	Loose UN ~ [0.6, 1.0]	Tight UN ~ [0.1, 0.4]
		× average makespan	× average makespan

The other variables are assumed to be fixed parameters. We assume three non-identical parallel CNC machines. The first machine (Mazak Super Quick Turn 100M-Y) has a horse power of 10 and tool magazine capacity of 12. The horse power and tool magazine capacity of the second machine (Mazak Quick Turn 300) are 30 and 16, respectively. The third machine's (Mazak Integrex 400Y) horse power is 40 and tool magazine size is 40. The first two CNC machines have small tool magazines relative to the third one. As Grieco *et al.* (2001) stated, machines with large tool magazines not only require high investment costs but also result in high indexing times. The operating cost of a machine consists of depreciation, labour, electricity, and maintenance costs. The sum of labour, electricity and maintenance costs is 0.08 \$/min for all machines. The depreciation cost is determined by using the initial cost and the useful life of the machines. The first machine's initial cost is \$60 000 and useful life is 10 years. As the machine works 300 days/year, 480 min/day, the depreciation cost is \$60 000 (10 years \times 300 days/year \times 480 min/day) = 0.04 \$/min. The initial cost of the second machine is \$100 000 and the useful life is 10 years. The depreciation cost is 0.07 \$/min. The third machine has an initial cost of \$280 000 and a useful life of 20 years. The depreciation cost is 0.09 \$/min. The operating costs are 0.12, 0.15 and 0.17 for machines 1–3, respectively. There are 100 parts that will be scheduled.

There are 15 different tool types for roughing operations and five different tool types for finishing operations. Tool change times are selected randomly from $UN \sim [0.040, 0.044]$ for machine 1, $UN \sim [0.046, 0.050]$ for machine 2 and $UN \sim [0.20, 0.30]$ for machine 3. Tool change times are calculated by considering the tool indexing times, which are given in the catalogue of the CNC machines mentioned above, and tool magazine size. Tool loading times are selected randomly from $UN \sim [0.78, 0.80]$. The tool replacing times are two times greater than the tool loading times. The operation related parameters, D_{pi} and L_{pi} , are selected randomly from the interval $UN \sim [1.5, 3]$ and $UN \sim [4, 8]$, respectively. The last operation of each part is the finishing operation and the other operations are roughing operations. $SF_{pi} = UN \sim [300, 500]$ and $d_{pi} = UN \sim [0.2, 0.3]$ for roughing operations and $SF_{pi} = UN \sim [30, 70]$ and $d_{pi} = UN \sim [0.025, 0.075]$ for finishing operations. We assume that the objectives are equally important for the decision maker ($u = 0.5$).

6.2 Numerical example

The simultaneous approach to FMS loading, scheduling and tool management problems is intuitively better than the sequential approach, which is the most commonly used method in the existing systems. However, we do not know which integration of the subproblems has the main effect in the improvement. In order to see the effect of integration at different levels, we built two new algorithms. The first algorithm, SLC-SCHTOOL, solves the loading problem first. The group technology approach, used in the SLC algorithm, is used to group the parts and assign the parts to the machines in this algorithm. Then the scheduling and tool management problems are integrated and solved simultaneously in the second stage. The proposed part selection index, which is used in algorithm PI, is used to schedule parts on each machine. The actual non-machining times are calculated at each decision point for all unscheduled parts. The second algorithm, SLC-LOADSCH, integrates

the loading and scheduling problems. The loading and scheduling problems are solved as they are solved in algorithm PI. The only difference is that the expected processing times are used instead of calculating the actual non-machining times to find the proposed machine selection and part selection indices. The tool allocation decisions are given at the second stage after loading and scheduling decisions are made. The proposed algorithms are coded in C language and compared with the proposed sequential algorithm (SLC) and simultaneous algorithm (PI).

The experimental design settings discussed above are used to generate the data for the numerical example. In order to show the feasible solutions in solution space, we reduced the problem size. Ten parts are considered for scheduling on two CNC machines (Mazak Super Quick Turn 100M-Y, Mazak Quick Turn 300). The experimental factor levels for factors A, B, C and D are selected as 0, 1, 1 and 1, respectively. The due dates and weights of parts, the tools required to perform the operations, tool usage rates and the machining times of the operations for the example can be seen in table 4. The tooling parameters (tool costs and tool changing, loading and replacing times) are also generated according to the distributions given in the previous section. They are not included in the paper due to space limitations. In problem space search, the population size is taken as 30 at the first 40 iterations and as 20 at the last 40 iterations.

The numerical example is solved by using the algorithms SLC, SLC-SCHTOOL, SLC-LOADSCH, and PI. The sequence of parts on each machine and the objective function values are shown in table 5 for each algorithm. According to the results, SLC and SLC-SCHTOOL give the worst results in terms of both manufacturing cost and total weighted tardiness. In these two algorithms, the loading problem is solved in the upper level of the hierarchy. The solution of the loading problem imposes constraints on the lower level problems, which restricts the solution space unnecessarily and does not provide enough flexibility to the lower level problems. Both SLC-LOADSCH and PI give better results since they solve the loading and scheduling problems simultaneously. Therefore, we could claim that the integration of loading and scheduling problems has the main effect on the improvement, since they have a strong interaction with each other. The loading decisions might change at each decision point due to the current tools loaded on the machines, which affect the non-machining times.

The numerical example is also used to show how a feasible solution is improved in the solution space. The proposed PSGA using PI as the base heuristic is used to solve the example. First, the base heuristic PI is solved with the original problem data where all perturbation values are equal to zero. The solution found has a manufacturing cost of 39 and total weighted tardiness of 587. The sequence of the parts on each machine in the solution can be seen in table 6 (iteration 0).

At the first iteration of PSGA, 30 perturbation vectors are randomly generated. A perturbation vector consists of perturbation values for each part-machine pair (used to perturb the part selection index, PS_{pm}) and tool-machine pair (used to perturb the tool index, TI_{jm}). For example, when we perturb the part selection indices of the unscheduled parts, different parts might give the maximum value for different perturbation vectors. The part selected at a decision point might change according to the values of the perturbation values. Different solutions can be found by using the perturbation vectors. One of the perturbation vectors gave a solution dominating the initial solution. This solution has the same manufacturing cost (39)

Table 5. Numerical example—comparison of the algorithms.

Algorithm	Schedule		Objective	
	Machine 1	Machine 2	Manufacturing cost	Total weighted tardiness
SLC	6, 4, 2, 7, 10	3, 9, 5, 8, 1	39	632
SLC-SCHTOOL	6, 4, 2, 7, 10	3, 9, 5, 8, 1	39	632
SLC-LOADSCH	2, 9, 10, 1, 8	3, 6, 4, 7, 5	39	525
PI	6, 2, 1, 10, 5, 8	3, 4, 7, 9	38	606
	3, 6, 4, 5, 1	2, 7, 8, 9, 10	39	523

Table 6. Numerical example—non-dominated solutions.

Iteration	Schedule		Objective	
	Machine 1	Machine 2	Manufacturing cost	Total weighted tardiness
0	6, 3, 7, 5, 1, 8	4, 2, 9, 10	39	587
1	6, 3, 10, 2, 5, 1	4, 9, 7, 8	38	696
	6, 4, 3, 5, 1	2, 7, 8, 9, 10	39	531
2	4, 6, 9, 10, 5, 1	2, 3, 7, 8	38	639
	6, 4, 3, 5, 1	2, 7, 8, 9, 10	39	531
7	6, 4, 9, 10, 5, 1	2, 3, 7, 8	38	630
	6, 4, 3, 5, 1	2, 7, 8, 9, 10	39	531
26	6, 4, 9, 10, 5, 1	2, 3, 7, 8	38	630
	3, 6, 4, 5, 1	2, 7, 8, 9, 10	39	523
59	6, 2, 1, 10, 5, 8	3, 4, 7, 9	38	606
	3, 6, 4, 5, 1	2, 7, 8, 9, 10	39	523

with the initial solution, but has a better total weighted tardiness (531). Another non-dominated solution, which has a better manufacturing cost (38), but worse total weighted tardiness (696), is also found at the first iteration. At the end of the first iteration, fitness values are assigned to each perturbation vector by using the objective values of each solution. The perturbation vectors for the second generation are generated by using the standard crossover and mutation operations of genetic algorithms.

At the second iteration, a solution which has a manufacturing cost of 38 and a total weighted tardiness of 639 is found. This solution dominates the solution found at the first iteration with objectives of (38, 696). As given in table 6, the new solution is found by reassignment of parts 4 and 9 to machine 1 and parts 2 and 3 to machine 2. The sequence of parts 2 and 3 is also changed. The algorithm does not get stuck in a single assignment of parts to machines. It can find different solutions with different machine assignments and different part sequences on each machine. Moreover, at iterations 7, 26 and 59, the non-dominated solution set changes. In addition, table 7 shows the tool magazine status, non-machining times (NMT) and weighted tardiness (WT) of each part for different non-dominated solutions. It also highlights the effect of tool magazine status on the non-machining times. If the tools do not exist on the tool magazine, the non-machining time is high.

Table 7. Numerical example—improvement in the feasible solution.

Iteration	Machine 1				Machine 2				Objective		
	Parts	Tool magazine	NMT	WT	Parts	Tool magazine	NMT	WT			
0	6	16 – (1, 14, 10)	0.3	0	4	16 – (1, 2)	0.25	0	(39, 587)		
	3	16 – (1, 14, 10, 0)	0.11	17	2	15 – (16, 2, 1, 13, 3, 10)	0.51	80			
	7	19 – (1, 14, 10, 16, 0, 13, 7)	0.3	67	9	15 – (16, 2, 1, 8, 3, 10, 13, 11)	0.27	70			
	5	19 – (1, 2, 10, 16, 0, 13, 7, 9, 6, 14)	0.31	129	10	15 – (16, 2, 1, 8, 9, 10, 13, 11, 3, 0)	0.27	80			
	1	19 – (1, 2, 6, 16, 0, 13, 7, 9, 10, 14)	0.02	107							
	8	15 – (0, 2, 6, 16, 19, 13, 7, 9, 10, 14, 1)	0.12	37							
	1	6	16 – (1, 14, 10)	0.3	0	4	16 – (1, 2)	0.25		0	(38, 696)
		3	16 – (1, 14, 10, 0)	0.11	17	9	15 – (1, 2, 16, 11, 8, 13)	0.5		30	
10		15 – (1, 14, 10, 16, 0, 3, 9)	0.3	39	7	19 – (1, 2, 16, 11, 8, 0, 15, 13, 7)	0.39	94			
2		15 – (10, 14, 3, 16, 0, 9, 1, 13)	0.13	195	8	15 – (0, 2, 16, 11, 8, 19, 1, 13, 7)	0.02	16			
5		19 – (10, 2, 3, 16, 0, 14, 1, 13, 15, 6, 9)	0.31	173							
1		15 – (6, 2, 3, 16, 0, 14, 1, 13, 15, 10, 9)	0.02	132							
1		6	16 – (1, 14, 10)	0.3	0	2	15 – (1, 13, 3, 10)	0.51	15	(39, 531)	
		4	16 – (1, 14, 10, 2)	0.11	9	7	19 – (1, 0, 3, 10, 15, 13, 7)	0.39	61		
	3	16 – (1, 14, 10, 2, 0)	0.1	45	8	15 – (0, 19, 3, 10, 1, 13, 7)	0.02	11			
	5	19 – (1, 2, 10, 6, 0, 16, 14, 9)	0.31	108	9	15 – (0, 19, 3, 10, 1, 8, 7, 13, 11)	0.26	92			
	1	19 – (1, 2, 6, 10, 0, 16, 14, 9, 13)	0.12	96	10	15 – (3, 19, 9, 10, 1, 8, 7, 13, 11, 0)	0.15	94			
	3	16 – (1, 0)	0.2	0	2	15 – (1, 13, 3, 10)	0.51	15			
	6	16 – (1, 0, 10, 14)	0.21	0	7	19 – (1, 0, 3, 10, 15, 13, 7)	0.39	61			
	4	16 – (1, 0, 10, 14, 2)	0.1	46	8	15 – (0, 19, 3, 10, 1, 13, 7)	0.02	11			
26	5	19 – (1, 0, 10, 2, 6, 16, 14, 9)	0.31	108	9	15 – (0, 19, 3, 10, 1, 8, 7, 13, 11)	0.26	92	(39, 523)		
	1	19 – (1, 0, 6, 2, 10, 16, 14, 9, 13)	0.12	96	10	15 – (3, 19, 9, 10, 1, 8, 7, 13, 11, 0)	0.15	94			
	3	16 – (1, 0)	0.2	0	2	15 – (1, 13, 3, 10)	0.51	15			
	6	16 – (1, 0, 10, 14)	0.21	0	7	19 – (1, 0, 3, 10, 15, 13, 7)	0.39	61			
	4	16 – (1, 0, 10, 14, 2)	0.1	46	8	15 – (0, 19, 3, 10, 1, 13, 7)	0.02	11			
	5	19 – (1, 0, 10, 2, 6, 16, 14, 9)	0.31	108	9	15 – (0, 19, 3, 10, 1, 8, 7, 13, 11)	0.26	92			
	1	19 – (1, 0, 6, 2, 10, 16, 14, 9, 13)	0.12	96	10	15 – (3, 19, 9, 10, 1, 8, 7, 13, 11, 0)	0.15	94			
	6	16 – (1, 14, 10)	0.3	0	3	16 – (1, 0)	0.25	0			
59	2	15 – (16, 14, 3, 1, 13, 10)	0.32	81	4	16 – (1, 0, 2)	0.13	0	(38, 606)		
	1	19 – (16, 14, 3, 1, 15, 6, 10, 13)	0.21	54	7	19 – (1, 16, 2, 0, 13, 7)	0.38	59			
	10	15 – (16, 14, 9, 1, 3, 6, 10, 13, 19, 0)	0.21	89	9	15 – (1, 16, 2, 0, 8, 7, 19, 11, 13)	0.39	84			
	5	19 – (16, 2, 14, 1, 3, 15, 10, 13, 9, 0, 6)	0.12	198							
	8	15 – (16, 2, 14, 0, 3, 1, 10, 13, 9, 19, 6)	0.02	41							

When the tools already exist on the magazine, the non-machining times decrease since only tool changing times are incurred.

The numerical example shows how complicated our problem is. The non-machining times change as the tool magazine status changes. If we had used the solution space as the neighbourhood structure, we would have performed a feasibility check after each change in the sequence or machine assignment. The feasibility check is a time-consuming task in the loading and scheduling problem with tooling constraints. The non-dominated solutions found show how effectively the proposed algorithm can find different solutions and improve the non-dominated solution set. The algorithm searches a large problem space as can be understood from the resulting non-dominated solutions with different machine assignments and sequence changes (tables 6 and 7).

6.3 Results

We first compare the single pass implementations of the proposed base heuristics, PI, EXP and SLC. We could not use PI-avg as a single pass algorithm since we cannot accurately calculate the average non-machining times due to an insufficient number of observations. The minimum, average and maximum of the weighted linear combination of normalized objective functions for algorithms PI, EXP and SLC are (0.21, 0.36, 0.52), (0.30, 0.45, 0.69) and (0.48, 0.57, 0.89), respectively. The CPU times in seconds are 0.290, 0.035 and 0.037 for algorithms PI, EXP and SLC, respectively. According to the first performance measure, PI is the best algorithm, whereas SLC is the worst algorithm. However, when we look at the CPU times, we can see that PI is the most time-consuming algorithm. EXP and SLC have significantly smaller CPU times.

In order to test the performance of PSGA, the number of restarts (NS) is either one or four. For the single start case (NS1), the population size is 30 for the first 40 generations and 20 for the next 40 generations at level 1 (L1). At level 2 (L2), the population size is 30 for the first 80 generations and 20 for the next 80 generations. For the multistart case (NS4), the population sizes are 30 and 20 for the first 10 and last 10 generations, respectively, at level 1 (L1). The population sizes are 30 and 20 for the first 20 and last 20 generations, respectively, at level 2 (L2). The remaining factors of PSGA are set as fixed parameters after some trial runs. The perturbation values are uniformly generated from $UN \sim [-0.5, 0.5]$. The percentage of sexual reproduction is 80%. Uniform crossover, in which the perturbation values of the offspring come from either parent with probability 0.5, is used. The mutation probability is set as 0.05. Finally, sc , which is used for assigning probabilities to the members of the population, is taken as 2.

An important issue in PSGA is the selection of the problem data that will be perturbed. In our problem, the part selection index, PS_{pm} , which is used to assign parts to machines, and the tool removal index, TI_{jm} , which is used to remove tools from the tool magazine, are perturbed to find different solutions for algorithms PI, PI-avg and EXP. For algorithm SLC, ATC_{pm} and TI_{jm} are perturbed.

We solve 80 problems for each generation–population size level and number of restarts. The results are summarized in table 8. The first performance measure is the average of the weighted linear function of two normalized objectives (*normobj*). Since the two objectives have different ranges, we cannot take the average of the

Table 8. Computational results.

Algorithm	Repr.	NS	<i>normobj</i>		Deviation		CPU time	
			L1	L2	L1	L2	L1	L2
PI	GA	1	0.23	0.22	0.017	0.015	579	1167
	GA	4	0.24	0.23	0.021	0.019	589	1171
	RND	1	0.32	0.31	0.033	0.033	586	1170
	RND	4	0.32	0.31	0.032	0.032	592	1183
PI-avg	GA	1	0.26	0.25	0.022	0.019	122	185
	GA	4	0.28	0.26	0.023	0.020	126	188
	RND	1	0.33	0.33	0.034	0.034	129	196
	RND	4	0.33	0.33	0.033	0.033	130	197
EXP	GA	1	0.29	0.27	0.025	0.022	66	132
	GA	4	0.31	0.28	0.028	0.026	68	134
	RND	1	0.39	0.38	0.039	0.036	70	139
	RND	4	0.39	0.38	0.038	0.038	70	139
SLC	GA	1	0.53	0.53	0.040	0.034	33	66
	GA	4	0.54	0.53	0.042	0.037	34	66
	RND	1	0.53	0.53	0.042	0.034	33	65
	RND	4	0.54	0.53	0.042	0.041	34	66

actual values of the two objectives to compare different algorithms. The objective function values for a solution are normalized according to the minimum and maximum objective function values found by all algorithms and then the normalized objective function values are aggregated into a single value by using a weighted linear function such that

$$f' = u \frac{f^m - \min_y f_y^m}{\max_y f_y^m - \min_y f_y^m} + (1 - u) \frac{f^l - \min_y f_y^l}{\max_y f_y^l - \min_y f_y^l}.$$

The solution giving the minimum f' is taken as the best solution for the corresponding algorithm. The average is taken over 80 problems. First, we compare the performance of the PSGA using different base heuristics. As we can see from the results, PI gives the best results. When we consider the computation times (which is the third performance measure), we can see that PI is also the most time-consuming algorithm. The algorithms PI-avg and EXP give solutions that are close to the solutions found by PI and their computation times are significantly less than the computation time of PI. The relationship between the CPU times and the normalized objectives for single start runs at level 1 can clearly be seen in figure 2. According to these results we can say that the performance of PSGA improves significantly with a good problem-specific base heuristic. Although PI-avg and EXP have simplifying assumptions for the calculation of the non-machining times, they give significantly better results than the sequential approach, SLC. The second comparison shows the importance of using a GA to search the problem space. We compare the effectiveness of GA to a pure probabilistic search, denoted as RND. In probabilistic search, we create one very large first generation, and perform no genetic operations. The random search performed poorly relative to the GA. This indicates the value of an evolutionary strategy for this problem. Another comparison is between the

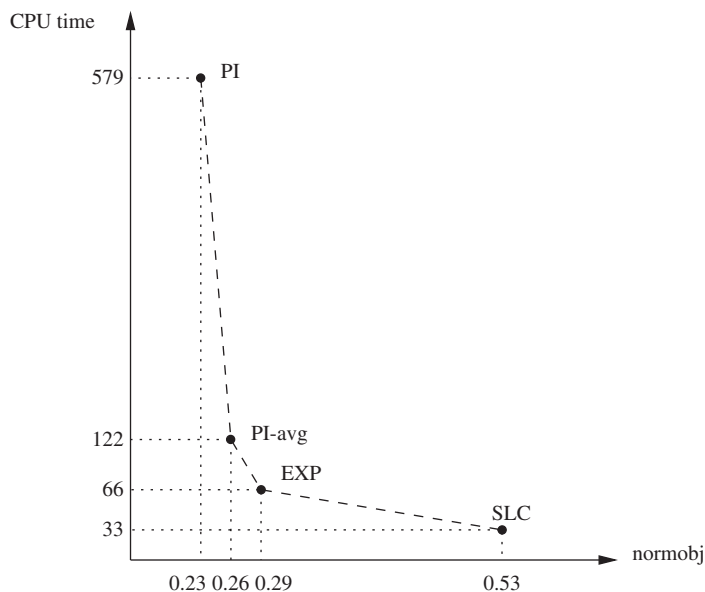


Figure 2. The relationship between *normobj* and CPU times for all algorithms at (L1, NS1).

population size–generation levels and single and multiple start runs. As the number of generations increases, PSGA should give better results. However, the *normobj* values are very close to each other for all algorithms. We also cannot see a significant difference between single start and multistart runs according to the *normobj* values.

For the first performance measure, we select a single solution giving the minimum weighted linear function of two normalized objectives, although we generate a number of approximately efficient solutions. The average number of non-dominated solutions for both generation–population size levels are very close to each other and it is around 12 for algorithms PI, PI-avg and EXP. The average number of non-dominated solutions is only one for SLC. In SLC, the parts families are allocated to machines at the beginning of the scheduling period and the allocation does not change throughout the algorithm. Sequence changes on each machine alone do not provide enough flexibility for the PSGA to generate more non-dominated solutions. In order to evaluate the quality of the non-dominated solution sets found by all algorithms, we use another performance measure, which is the average deviation from the approximate non-dominated solution set (*deviation*), as follows:

$$\frac{\sum_{i \in \text{NDS}} (\min_{i \in \text{NDS}^*} \{u[(f_i^m - f_i^m)/f_i^m] + (1-u)[(f_i^l - f_i^l)/f_i^l\})}{|\text{NDS}|}$$

The approximate non-dominated solution set, NDS^* , is found by using the non-dominated solutions of all proposed algorithms. In order to calculate the deviation of a solution in set NDS, first the distance of the solution to the solutions in set NDS^* is calculated in terms of the two objective function values. The minimum distance is taken as the deviation of that solution, which is calculated for all solutions in set NDS. The average of the deviations for all solutions is taken as the deviation of the set NDS from set NDS^* . A small numerical example will clarify

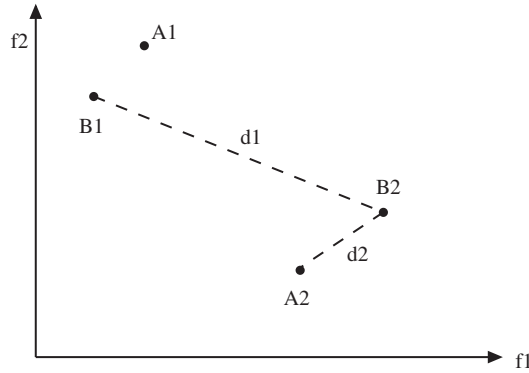


Figure 3. Approximate non-dominated solution set.

this performance measure. In figure 3, the solutions A1 and A2 are found by algorithm A and the solutions B1 and B2 are found by algorithm B. B1 and A2 form the approximate non-dominated solution set NDS^* . The deviation of point B1 from the non-dominated solution set is zero, because B1 is in set NDS^* . The deviation of point B2 is the minimum of $d1$ and $d2$. The deviation of set B is the average of the deviations of all the points in set B. In our computational experiments, the average of the minimum deviations of the set NDS from set NDS^* is taken over 80 runs. The relative ranking among the algorithms is consistent with the *normobj* measure. The GA is better than the RND case. When we compare the results between levels, as the number of generations increase, PSGA gives better results, as expected. Since the convergence is slow, the single start runs give better results than multiple start runs for all algorithms. For such a difficult problem, taking a long single start run is better than taking multiple shorter runs.

6.4 Alternative tooling

In order to see the effect of flexibility in choosing processing times, we solve the previously generated 80 problems with the proposed base heuristics considering the alternative tooling and compare it with the algorithms that assume all operations are performed with their primary tools. The first comparison is between the single pass runs. The minimum, average and maximum of the weighted linear combination of normalized objective functions for algorithms PI-ALT, EXP-ALT and SLC-ALT considering alternative tooling are (0.18, 0.33, 0.44), (0.30, 0.43, 0.63) and (0.48, 0.55, 0.88), respectively. The CPU times are 0.295, 0.035 and 0.034 for algorithms PI-ALT, EXP-ALT and SLC-ALT, respectively. According to the first performance measure, PI-ALT is the best algorithm. However, it has the largest computation time, but it is still relatively modest. EXP-ALT and SLC-ALT have significantly smaller CPU times and worse normalized objective function values.

The second comparison is among the PSGA runs. We solve PSGA with PI-ALT, PI-avg-ALT, EXP-ALT and SLC-ALT for level 1 (L1) and number of restarts 1 and 4. The average of the weighted linear function of normalized objectives, the average deviation from the approximate non-dominated solution set and

Table 9. Computational results—alternative tooling.

Algorithm	NS	<i>normobj</i>	Deviation	CPU time
PI-ALT	1	0.20	0.010	587
	4	0.22	0.015	595
PI-avg-ALT	1	0.24	0.016	123
	4	0.25	0.019	127
EXP-ALT	1	0.26	0.020	65
	4	0.28	0.023	67
SLC-ALT	1	0.51	0.014	31
	4	0.51	0.023	29

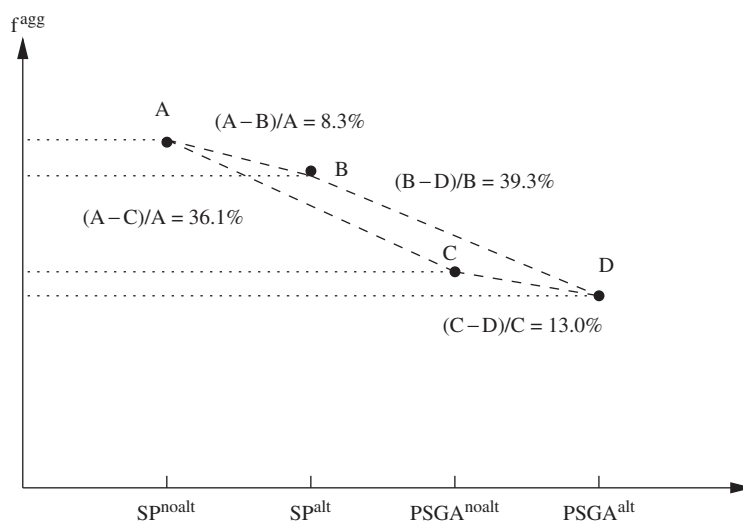


Figure 4. Comparison of percentage improvements.

the average CPU times over 80 runs can be seen in table 9. Although the CPU times are very close to the computational times of the algorithms PI, PI-avg, EXP and SLC, the average of the weighted linear function of normalized objectives and the deviation from the approximate solution set are improved when alternative tooling is considered.

The difference between the algorithms can be seen more clearly when we compare the percentage improvements in terms of *normobj* values. The percentage improvements between single pass implementations of PI and PI-ALT and PSGA using these two algorithms for single start runs can be seen in figure 4. The percentage improvement in weighted linear functions of two objectives between algorithms PI and PI-ALT, which is calculated as $(\text{normobj}(\text{SP}, \text{PI}) - \text{normobj}(\text{SP}, \text{PI} - \text{ALT})) / (\text{normobj}(\text{SP}, \text{PI})) \times 100 = ((0.36 - 0.33) / 0.36) \times 100$, is 8.3%. The performance of the single pass implementation of PI-ALT is significantly better than the single pass implementation of PI. When PI is used in PSGA the improvement for single start runs over the single pass implementation of PI is 36.1%. The improvement

of PSGA using PI-ALT over the single pass implementation of PI-ALT is 39.3%. The percentage improvements in the weighted linear function of normalized objectives for algorithms PI, PI-avg, EXP and SLC at level 1 can be seen in table 10. According to these results, we can say that the problem space search adds more to the objective of the single pass implementation of the algorithm with a better base heuristic. The percentage difference between the PSGA using PI and PSGA using PI-ALT is 13.0%. According to these results, we can say that the flexibility in choosing the processing times of operations provides better solutions. Also, the performance of PSGA increases more with a better base heuristic.

7. Concluding remarks

In this study, we consider FMS loading, scheduling and tool management problems. This study is among only a few which consider the interaction between different levels of the production management hierarchy by solving these problems simultaneously. We consider the objectives of minimizing the manufacturing cost comprised of machining, non-machining and tooling costs, and minimizing the total weighted tardiness. We propose different base heuristics that are employed in a PSGA to find approximately efficient solutions which provide alternative solutions to the decision maker. An important contribution of the proposed base heuristics is the calculation of the actual non-machining times at each decision point for loading and scheduling the parts (as done in the PI algorithm). The non-machining times depend on the current status of the tool magazines, and are determined by the limited tool lives, the limited capacity of the tool magazine, and all the scheduling and tool allocations decisions made up to the current time. However, this critical computation is very time consuming. In addition to the PI algorithm, we propose two different ways to estimate the non-machining times. The first alternative, PI-avg, uses average non-machining times, which are calculated by using the actual observations made in PSGA using PI. The second alternative, EXP, uses expected non-machining times. We also consider alternative tooling, which provides us flexibility in choosing the tool alternatives and hence the processing times. According to the computational results, the algorithms that consider alternative tooling perform better. As expected, different processing time alternatives for an operation increase the solution quality.

In order to show the solution quality of the proposed simultaneous approaches, we compare them with a sequential algorithm, SLC. The SLC solves the part loading and scheduling problem sequentially. The tool replacement decisions are made after the parts are scheduled on each machine. Although the sequential approach is simpler and requires less computation time compared to simultaneous approaches, the improvements of the proposed simultaneous approaches in terms of the normalized objective values over the SLC are large enough (40% and 68% on the average with and without alternative tooling) to show that there is a significant interaction between the tool management and scheduling decisions. Therefore, these two problems should not be viewed in isolation, which supports our claim and shows the advantage of a simultaneous approach over the SLC. As a result, the proposed approach not only improves the CNC machine efficiency but also becomes more responsive to customer due date requirements.

Table 10. Comparison of algorithms with and without alternative tooling (percentage improvement).

Algorithm	Single pass (A - B)/A	NS = 1			NS = 4		
		(A - C)/A	(B - D)/B	(C - D)/C	(A - C)/A	(B - D)/B	(C - D)/C
PI	8.3	36.1	39.3	13.0	33.3	33.3	8.3
PI-avg	4.4	35.6	39.5	7.7	31.1	34.9	10.7
EXP	3.5	7.0	7.3	10.3	5.3	7.3	9.7
SLC				3.8			5.6

Acknowledgement

This work was supported, in part, by NATO Collaborative Research Grant CRG-971489.

References

- Akturk, M.S. and Avci, S., Tool allocation and machining conditions optimization for CNC machines. *Eur. J. Oper. Res.*, 1996, **94**, 335–348.
- Akturk, M.S. and Ozkan, S., Integrated scheduling and tool management in flexible manufacturing systems. *Int. J. Prod. Res.*, 2001, **39**, 2697–2722.
- Amoako-Gyampah, K., A comparative study of FMS tool allocation and part type selection approaches for a varying part type mix. *Int. J. Flex. Manuf. Sys.*, 1994, **6**, 179–207.
- Avci, S., Akturk, M.S. and Storer, R.H., A problem space algorithm for single machine weighted tardiness problems. *IIE Transactions*, 2003, **35**, 479–486.
- Baykasoglu, A. and Dereli, T., Heuristic optimization system for the determination of index positions on CNC magazines with the consideration of cutting tool duplication. *Int. J. Prod. Res.*, 2004, **42**, 1281–1303.
- Bernardo, J.J. and Lin, K.-S., An interactive procedure for bi-criteria production scheduling. *Comput. Oper. Res.*, 1994, **21**, 677–688.
- Ernst, A.T., Krishnamoorthy, M. and Storer, R.H., Heuristic and exact algorithms for scheduling aircraft landings. *Networks*, 1999, **34**, 229–241.
- Fathi, Y. and Barnette, K.W., Heuristic procedures for the parallel machine problem with tool switches. *Int. J. Prod. Res.*, 2002, **40**, 151–164.
- Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1989 (Addison-Wesley: Boston, MA).
- Gray, A.E., Seidman, A. and Stecke, K.E., A synthesis of decision models for tool management in automated manufacturing. *Manag. Sci.*, 1993, **39**, 549–567.
- Grieco, A., Semeraro, Q. and Tolio, T., A review of different approaches to the FMS loading problem. *Int. J. Flex. Manuf. Sys.*, 2001, **13**, 361–384.
- Groover, M.P., *Fundamentals of Modern Manufacturing: Materials, Processes and Systems*, 2002 (Wiley: New York).
- Morton, T.E. and Pentico, D.W., *Heuristic Scheduling Systems: With Applications to Production Systems and Project Management*, 1993 (Wiley: New York).
- Mukhopadhyay, S.K. and Sahu, S.K., Priority-based tool allocation in a flexible manufacturing system. *Int. J. Prod. Res.*, 1996, **34**, 1995–2018.
- Naphade, K.S., Wu, S.D., Storer, R.H. and Doshi, B.J., Melt scheduling to trade off material waste and shipping performance. *Oper. Res.*, 2001, **49**, 629–645.
- Rachamadugu, R. and Stecke, K.E., Classification and review of FMS scheduling approaches. *Prod. Plan. Con.*, 1994, **5**, 2–20.
- Roh, H.-K. and Kim, Y.-D., Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter. *Int. J. Prod. Res.*, 1997, **35**, 2989–3003.
- Sarin, S.C. and Chen, C.S., The machine loading and tool allocation problem in a flexible manufacturing system. *Int. J. Prod. Res.*, 1987, **25**, 1081–1094.
- Sodhi, M.S., Lamond, B.F., Gautier, A. and Noël, M., Heuristics for determining economic processing rates in a flexible manufacturing system. *Eur. J. Oper. Res.*, 2001, **129**, 105–115.
- Stecke, K.E., Design, planning, scheduling and control problems of flexible manufacturing systems. *Ann. Oper. Res.*, 1985, **3**, 3–12.
- Storer, R.H., Wu, S.D. and Vaccari, R., New search spaces for sequencing problems with application to job shop scheduling. *Manag. Sci.*, 1992, **38**, 1495–1509.

- Tang, C.S. and Denardo, E.V., Models arising from a flexible manufacturing machine, Part I: minimization of the number of tool switches. *Oper. Res.*, 1988, **36**, 767–777.
- Turkcan, A., Akturk, M.S. and Storer, R.H., Nonidentical parallel CNC machine scheduling. *Int. J. Prod. Res.*, 2003, **41**, 2143–2168.