



Non-identical parallel CNC machine scheduling

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

To cite this article: Ayten Turkcan , M. Selim Akturk & Robert H. Storer (2003) Non-identical parallel CNC machine scheduling, International Journal of Production Research, 41:10, 2143-2168, DOI: [10.1080/0020754031000087184](https://doi.org/10.1080/0020754031000087184)

To link to this article: <http://dx.doi.org/10.1080/0020754031000087184>



Published online: 14 Nov 2010.



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



Article views: 71



View related articles [↗](#)



Citing articles: 12 View citing articles [↗](#)

Non-identical parallel CNC machine scheduling

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

In this study, we solve the non-identical parallel CNC machine scheduling problem. We have two objectives: minimizing the manufacturing cost (comprising machining, non-machining and tooling costs) and minimizing the total weighted tardiness. The tooling constraints affect the non-machining times as well as the machining conditions, such as cutting speed and feed rate, which in turn specify the machining times and tool lives. We propose a two-stage algorithm to find optimal machining conditions and to determine machine allocation, tool allocation and part scheduling decisions. The proposed algorithm generates different schedules according to the relative importance of the objectives.

1. Introduction

In this study, we solve a scheduling problem in a parallel CNC machine environment. Most of the problems in the parallel machine scheduling area are intractable. Only problems with certain restrictions such as unit processing times, or in-tree and out-tree precedence relations, are polynomially solvable. For NP-complete problems, exponential time algorithms (Azizoglu and Kirca 1999) and approximation algorithms (Alon *et al.* 1998) are proposed. Since the computation time of exact methods increases exponentially as the problem size increases, heuristics are also proposed (Koulamas 1997). Although the heuristics do not guarantee optimality, they provide a solution in reasonable computation times.

In CNC machine environments, parallel machine scheduling is more difficult due to tooling constraints. Furthermore CNC machines usually have different tool magazine capacities and horsepowers (i.e. they are non-identical), although they can perform the complete set of operations given that the required tools are loaded in their tool magazine. Since the parts have different tooling requirements and the tool magazines have limited capacities, the magazines cannot be loaded with all necessary tools. In most of the studies, tool changes are assumed to be due to part mix. However, as Gray *et al.* (1993) reported, the tools are changed 10 times more often due to wear than due to part mix. The machining conditions, such as cutting speed and feed rate, affect tool lives and processing times. When we increase the speed or feed rate, the tool usage rate increases. The tools wear more rapidly, and hence the number of tool changes increases. On the other hand, the processing times decrease as the cutting speed or feed rate increases.

Revision received October 2002.

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

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

* To whom correspondence should be addressed. email: akturk@bilkent.edu.tr

The primary inputs to the scheduling problem are the machining and non-machining times. The machining parameters affect the machining times of operations and tool lives. The tool magazine capacities together with the tool lives affect the non-machining times and tooling costs. In this study, we have two objectives: minimizing manufacturing cost and minimizing total weighted tardiness. We can combine the two objectives with weighted linear or Tchebycheff functions by assigning different weights to each according to their relative importance. There are few studies that consider tool management problem in FMS scheduling. Most of the existing studies that consider tooling constraints solve a loading problem instead of a scheduling problem. Tiwari and Vidyarthi (2000) propose a genetic algorithm to solve a machine loading problem with the availability of machining time and tool slot constraints. The objectives are minimizing the system unbalance and maximizing the throughput. A weighted linear combination of the two objectives is taken. Although balancing the workload is the most commonly used objective in the FMS scheduling literature with multiple machines, it does not consider the usual objectives found in practice such as minimizing machining costs or meeting due dates. Fathi and Barnette (2002) propose heuristics to solve a parallel machine scheduling problem. After the parts are allocated to the machines and the sequence is determined, the tools are changed according to the 'keep tool needed soon' (KTNS) rule of Tang and Denardo (1988). Roh and Kim (1997) propose three heuristics to solve the part loading, tool loading and part sequencing problems with the objective of minimizing total tardiness in a flexible manufacturing environment with identical parallel machines. They do not consider limited tool lives, but they assume that they have a fixed number of tools that travel between machines for the required operations. Randhawa and Kuo (1997) propose a multistage heuristic for minimizing either total flowtime, maximum tardiness or total tardiness in a non-identical parallel machine environment. Sequence-dependent setup times are incurred between parts of different part families. In most of the existing studies, the tool change times are considered as sequence-dependent setup times which are known in advance. However, in our study, the non-machining times depend on the current status of the tool magazine which is a determinant of all scheduling and tool allocation decisions made up to the current time. One cannot realistically assume that non-machining times are predetermined sequence-dependent setup times in practice. Akturk and Ozkan (2001) consider the approximated values of the non-machining times as dynamic terms which should be calculated at each decision point. They 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. In their study, the number of tools is limited. Therefore, when all the tools on hand are used, the alternative tools are considered for operations. They assume that all cost terms have equal weight and find a single optimum solution.

In this study, we propose a two-criteria, two-stage algorithm to generate alternative schedules in a non-identical parallel CNC machine environment. The remainder of this paper is organized as follows. In the following section, the problem is defined with its underlying assumptions. The proposed algorithm and a number of algorithms from the literature which will be used for comparison are explained in Sections 3 and 4, respectively. A numerical example is provided in order to clarify the basic steps of the proposed algorithm in Section 5. In Section 6, a computational

study is performed to test the performance of the proposed algorithm. In the last section, some concluding remarks are provided.

2. Problem definition

In this study, we make the following assumptions. There are multiple non-identical parallel CNC machines. Each CNC machine has a different tool magazine capacity, maximum available horsepower, operating cost and tool changing, loading and replacing times. A machine can process one part at a time. Each part has a distinct due date and a weight which shows the importance of the part relative to the other parts. The parts have multiple operations. There is a precedence relationship between the operations of each part. All the operations of a part should be processed on the same machine, since in the existing CNC technology the tool change times are significantly shorter than part loading and unloading times. Each operation can be performed by a set of alternative tool types. Processing of an operation cannot be interrupted for a tool change due to surface finish requirements. Therefore, it should be processed with a tool which 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 machines, the machine must be stopped for tool replacement. There is a central tool storage location where unassigned tools are kept. A robotic manipulator is used to transfer tools between central storage and the tool magazines of the machines. This production environment is similar to the one discussed by Macchiaroli and Riemma (1996). Under these assumptions, we will determine the relevant tool management decisions which are selection of machining conditions, tool allocation, and scheduling decisions.

We have two objectives: minimizing manufacturing cost (which is important for the manufacturer) and minimizing total weighted tardiness (which is important for the customer). These two objectives usually conflict with each other. We can decrease the machining time, and hence machining cost, by increasing the speed and feed rate. But, this will increase the non-machining and tooling costs. The total weighted tardiness increases or decreases according to the change in the sum of the machining and non-machining times. The proposed algorithm generates alternative solutions which minimize the two criteria of manufacturing costs and total weighted tardiness with different weights. In the end, the decision-maker selects a solution among the alternatives according to the relative importance of the objectives. In multiobjective optimization, the aim is to find a trade-off curve between the conflicting objectives. The objectives are combined into a single objective function by using different functions such as weighted linear, quadratic or Tchebycheff functions. The weighted linear function, which is calculated as $f(x) = \sum_{i=1}^k u_i f_i(x)$, is the most popular method in the operations research literature, but has some drawbacks. When the trade-off curve is non-convex, this method cannot find efficient solutions in non-convex regions. Also, the weighted linear function method with an evenly distributed set of weights may fail to produce an even distribution of points from all parts of the efficient solution set, as stated by Das and Dennis (1997). In this study, we also consider the weighted Tchebycheff function, which is a commonly used method in the multi-objective optimization literature. It is calculated as $f(x) = \max_{i \in \{1, \dots, k\}} \{u_i f_i(x)\}$. We solve our problem with different weight alternatives for finding the approximately efficient solutions. The notation used throughout this paper is given below:

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 (inches) |
| 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 (μin) |
| 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) |
| $tc_{jm}, tl_{jm}, tr_{jm}$ | tool changing, loading and replacing times of tool j on machine m |

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) |
| U_{pijm} | usage rate of tool j in operation i of part p on machine m |
| T_{pijm} | tool life of tool j for operation i of part p on machine m |
| C_{pijm}^m | machining cost of operation i of part p using tool j on machine m |
| C_{pijm}^{nm} | non-machining cost of operation i of part p using tool j on machine m |
| C_{pijm}^{tool} | tooling cost of operation i of part p using tool j on machine m |
| tard_{pm} | tardiness of part p on machine m |
| t_{pijm}^m | machining time of operation i of part p using tool j on machine m |
| t_{pijm}^{nm} | non-machining time of operation i of part p using tool j on machine m |
| t_{pm}^m | total machining time of part p on machine m |
| t_{pm}^{nm} | total non-machining time of part p on machine m |
| R_{jm} | remaining tool life of tool j on machine m |
| x_{pijm} | binary variable which is equal to 1 if $R_{jm} \leq U_{pijm}$ and 0 otherwise |

3. The algorithm

In this study, we propose a two-stage algorithm to solve the nonidentical parallel CNC machine scheduling problem. In the first stage, we find the optimal machining conditions that affect the machining times and tool usage rates. The machining times are the primary input to the scheduling problem and therefore should be known before the scheduling process starts. The tool usage rates affect the frequency of the tool changes due to wear. The optimal machining conditions, which minimize the sum of machining and tooling costs, is found by solving the single machining operation problem (SMOP) for each operation–tool pair. The non-machining cost

and the total weighted tardiness are not considered in the first stage because they depend on the current status of the tool magazine, which changes according to the previous part and tool allocation and tool replacement decisions. In the second stage of the algorithm, referred to as the scheduling stage, parts and tools are assigned to non-identical parallel machines and parts are scheduled on each machine. All the objective function terms are considered in the scheduling stage, while the parts are scheduled one at a time. The detailed information about the two stages can be found in the following sections.

3.1. Stage 1: Finding optimal machining conditions

In the first stage, our aim is to find optimal machining conditions such as the cutting speed and feed rate for each operation–tool pair. The machining conditions affect the machining times as well as tool usage rates, which, in turn affect the tool allocation and replacement decisions. Akturk and Avci (1996) propose a posynomial geometric programming (GP) model to solve the machining conditions optimization problem. A posynomial geometric program has the general form as follows:

$$\begin{aligned} &\text{minimize} && \sum_{k \in K_0} d_k \prod_{j=1}^n X_j^{a_{kj}} \\ &\text{subject to} && \sum_{k \in K_i} d_k \prod_{j=1}^n X_j^{a_{kj}} \leq 1 \quad i = 1, \dots, m \\ &&& X_j > 0 \quad j = 1, \dots, n \end{aligned}$$

where K_0, K_1, \dots, K_m are mutually disjoint sets that index the posynomial terms in the objective and constraints ($K_0 \cup K_1 \cup \dots \cup K_m = \{1, 2, \dots, M\}$), d_k are the corresponding weights ($d_k > 0$) and a_{kj} are exponents of arbitrary sign.

In order to solve a posynomial geometric program more efficiently, the dual of the model is taken as follows:

$$\begin{aligned} &\text{maximize} && \prod_{k=1}^M \left(\frac{d_k}{Y_k} \right)^{Y_k} \\ &\text{subject to} && \sum_{k \in K_0} Y_k = 1 \quad (\text{normality}) \\ &&& \sum_{k=1}^M Y_k a_{kj} = 0 \quad j = 1, 2, \dots, n \quad (\text{orthogonality}) \\ &&& Y_k \geq 0 \quad k = 1, 2, \dots, M. \end{aligned}$$

The degree of difficulty of the posynomial programs is calculated as [(number of posynomial terms k) – (number of variables j) – 1]. More detailed information about posynomial geometric programming can be found in Rardin (1998).

The GP model proposed by Akturk and Avci (1996) is as follows:

$$\begin{aligned}
 \text{(GP) minimize } \text{SMOP}_{pijm} &= C_{om} t_{pijm}^m + C_t U_{pijm} \\
 &= C'_1 v_{pijm}^{-1} f_{pijm}^{-1} + C'_2 v_{pijm}^{(\alpha_j-1)} f_{pijm}^{(\beta_j-1)} \\
 \text{subject to: } C'_t v_{pijm}^{\alpha_j-1} f_{pijm}^{\beta_j-1} &\leq 1 \quad (\text{tool life constraint}) \\
 C'_m v_{pijm}^b f_{pijm}^c &\leq 1 \quad (\text{machine power constraint}) \\
 C'_s v_{pijm}^g f_{pijm}^h &\leq 1 \quad (\text{surface roughness constraint}) \\
 v_{pijm}, f_{pijm} &> 0
 \end{aligned}$$

where

$$C'_1 = \frac{\pi D_{pi} L_{pi} C_{om}}{12}, \quad C'_2 = \frac{\pi D_{pi} L_{pi} d_{pi}^{\gamma_j} C_t}{12 C_j}, \quad C'_m = \frac{C_m d_{pi}^e}{\text{HP}_m}, \quad C'_s = \frac{C_s d_{pi}^l}{\text{SF}_{pi}}, \quad C'_t = \frac{\pi D_{pi} L_{pi} d_{pi}^{\gamma_j}}{12 C_j}.$$

The proposed GP model minimizes the sum of machining and tooling costs subject to tool life, machine power and surface roughness constraints. The first cost component in the objective function is the machining cost which is incurred for the time spent to complete a metal cutting operation. The second term is the tooling cost related to the tool usage rate which is the ratio of the machining time to tool life. Akturk and Ozkan (2001) also find optimal machining conditions in their study by using a GP model. But, they consider the approximated non-machining cost, which depends on the tool usage rate, machine cost and the expected tool change time, in their objective function. We do not consider non-machining times at this stage, because the non-machining times depend on the current status of the tool magazine which is determined by the previous scheduling and tooling decisions. In the GP model, according to the first constraint, the machining time cannot exceed the available tool life. The second constraint guarantees the feasibility of the cut by the available machine horsepower. The surface roughness constraint satisfies the quality requirement for the operation. The degree of difficulty of the proposed GP model is 2. There are eight cases (in which the constraints are tight or loose at optimality) that must be considered. The number of cases was reduced to six, by the following theorem of Akturk and Avci (1996).

Theorem 1: In the constrained SMOP, at least one of the surface roughness or machine power constraints must be tight at the optimal solution.

In the worst case, all six cases are evaluated to find an optimal solution. In this study, we solve this problem in a more efficient way. First, we take the dual of the GP model as follows:

$$\begin{aligned}
 \text{(Dual GP) maximize } Q^* &= \left(\frac{C'_1}{Y_1} \right)^{Y_1} \cdot \left(\frac{C'_2}{Y_2} \right)^{Y_2} \cdot (C'_t)^{Y_3} \cdot (C'_m)^{Y_4} \cdot (C'_s)^{Y_5} \\
 \text{subject to: } Y_1 + Y_2 &= 1 \quad (1) \\
 -Y_1 + (\alpha_j - 1) \cdot Y_2 + (\alpha_j - 1) \cdot Y_3 + b \cdot Y_4 + g \cdot Y_5 &= 0 \quad (2) \\
 -Y_1 + (\beta_j - 1) \cdot Y_2 + (\beta_j - 1) \cdot Y_3 + c \cdot Y_4 + h \cdot Y_5 &= 0 \quad (3) \\
 Y_1, Y_2, Y_3, Y_4, Y_5 &\geq 0.
 \end{aligned}$$

Then, we relax the tool life constraint, which reduces the degree of difficulty to 1. We use the following theorem of Akturk *et al.* (2002) in order to reduce the number of cases further.

Theorem 2: The surface roughness constraint must be tight at the optimal solution.

Proof: Since we relax the tool life constraint in the GP model, the dual variable Y_3 corresponding to that constraint disappears in dual GP. When the machine power constraint is tight and the surface roughness constraint is loose, the dual variable Y_5 corresponding to the surface roughness constraint is equal to zero due to complementary slackness conditions. When we solve the equations (1), (2) and (3), we find that $Y_4 = -((\alpha_j - \beta_j)/(b - c))Y_2$. Due to Gorczyca (1987), $b > c$ and $\alpha_j > \beta_j$. If $Y_2 > 0$, then $Y_4 < 0$, which makes this case infeasible. Therefore, the surface roughness constraint must be tight at optimality.

We solve the proposed GP model by using the primal and dual constraints and complementary slackness conditions. The optimal v and f pair can be found as follows:

- (1) Find the point that minimizes the objective function value in the surface roughness constraint (point C in figure 1). When the surface roughness constraint is tight, $C'_s(v_{p_{ijm}})^g(f_{p_{ijm}})^h = 1$, then $v_{p_{ijm}} = (C'_s)^{-1/g}(f_{p_{ijm}})^{-h/g}$. If we insert v into the objective function, then it becomes $C'_1(C'_s)^{1/g}(f_{p_{ijm}})^{(h-g)/g} + C'_2(C'_s)^{(1-\alpha_j)/g}(f_{p_{ijm}})^{(g(\beta_j-1)-h(\alpha_j-1))/g}$, which is a convex function of $f_{p_{ijm}}$. When we take the derivative of the objective function with respect to $f_{p_{ijm}}$, and solve it for f , we can find that

$$f_{p_{ijm}} = \left[\left(\frac{C'_1}{C'_2} \right)^g \left(\frac{g-h}{g(\beta_j-1)-h(\alpha_j-1)} \right)^g (C'_s)^{\alpha_j} \right]^{1/(g\beta_j-h\alpha_j)}$$

and

$$v_{p_{ijm}} = \left(\frac{1}{C'_s f_{p_{ijm}}^h} \right)^{1/g}$$

- (2) If the point found in the first step satisfies the machine power constraint, go to step 3. Otherwise, calculate the point at which the machine power and surface roughness constraints are both tight (point B in figure 1). The feed rate and cutting speed are calculated by solving the equations $C'_m(v_{p_{ijm}})^b(f_{p_{ijm}})^c = 1$ and $C'_s(v_{p_{ijm}})^g(f_{p_{ijm}})^h = 1$ simultaneously, which results in

$$f_{p_{ijm}} = [(C'_m)^g(C'_s)^{-b}]^{1/(hb-gc)} \quad \text{and} \quad v_{p_{ijm}} = \left(\frac{1}{C'_s f_{p_{ijm}}^h} \right)^{1/g}$$

- (3) If the optimal solution to the relaxed problem (point C or B) satisfies the tool life constraint, then it is optimal to the overall problem. Otherwise, calculate the point at which the tool life and surface roughness constraints are both tight (point A in figure 1). At point A , we solve the equations

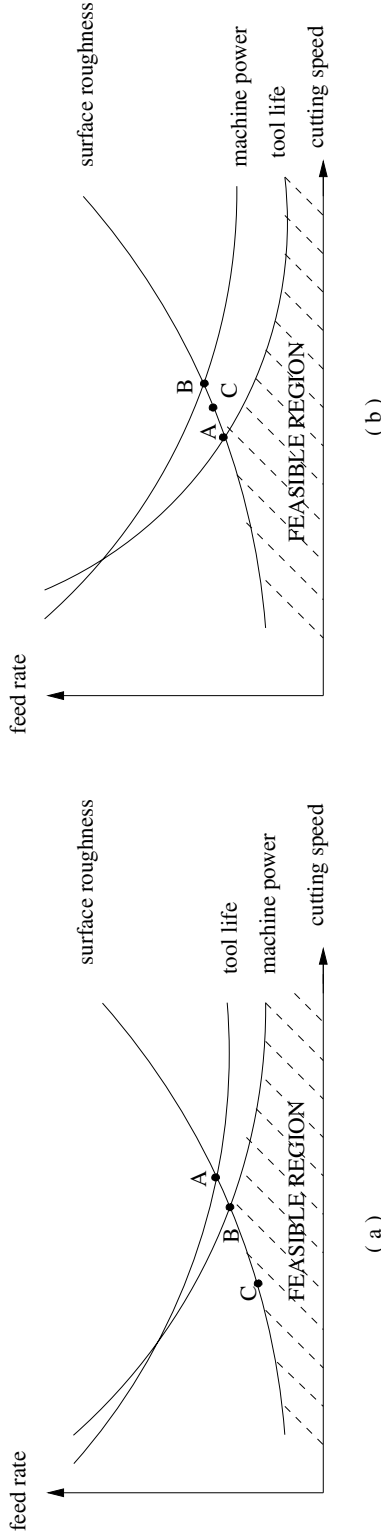


Figure 1. Feasible region.

$C_t'(v_{pijm})^{\alpha_j-1} (f_{pijm})^{\beta_j-1} = 1$ and $C_s'(v_{pijm})^g (f_{pijm})^h = 1$ simultaneously and the optimal solution is

$$f_{pijm} = [(C_t')^g (C_s')^{1-\alpha_j}]^{1/(h(\alpha_j-1)-g(\beta_j-1))} \quad \text{and} \quad v_{pijm} = \left(\frac{1}{C_s' f_{pijm}^h} \right)^{1/g}.$$

We can solve SMOP for each operation–tool pair very quickly by using the primal–dual relationships of geometric programming and problem parameters. After finding the optimal v and f values for each operation–tool pair, we proceed now to the second stage.

3.2. Stage 2: Scheduling

In the second stage, tool allocation, part allocation and scheduling decisions are made. Our aim is to minimize the total manufacturing cost and total weighted tardiness simultaneously. The machining and tooling costs are functions of the optimal cutting speed and feed rate that were determined in the first stage. The non-machining cost is incurred for the non-machining time spent to change, replace and load 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. 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 non-machining cost and the total weighted tardiness are dynamic terms. That means they change as the state of the system changes. Since the current status of the tool magazine depends on the previous part and tool allocation decisions, we schedule the parts one at a time and calculate the change in objective functions after each assignment. The calculation of the change in objective functions (in step 2) is important for selecting a machine for each part. The machine giving the minimum increase as a weighted linear or Tchebycheff function of two normalized objective functions is the most appropriate machine for the corresponding part at the current time. At some later time, another machine may become a better alternative for that part. The objectives are normalized (in step 3) in order to prevent the dominance of one objective over the other because of different ranges. After the most suitable machines are determined for unscheduled parts, a part–machine pair is selected according to the proposed part selection index (step 4). The part–machine pair with the highest priority index is selected and scheduled at step 5. The algorithm continues until all the parts are scheduled.

The steps of the second stage are as follows:

- Step 1. Initialization: $UNS = \{1, 2, \dots, N\}$ and $ALT = \{1, 2, \dots, M\}$
- 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

$$f_{pm}^m = \sum_i \sum_j (C_{pijm}^m + C_{pijm}^{tool} + C_{pijm}^{nm})$$

and

$$f_{pm}^t = \text{tard}_{pm} = w_p \max\{0, t_m^{\text{now}} + t_{pm}^{\text{tm}} + t_{pm}^{\text{tnm}} - \text{DD}_p\},$$

respectively, where $C_{pijm}^{\text{nm}} = \text{Co}_m t_{pijm}^{\text{nm}}$. All operations are performed with their primary tools which give the minimum sum of machining and tooling costs found at Stage 1.

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 Tchebycheff or linear function of two objectives for each part $p \in \text{UNS}$. When the weighted Tchebycheff function is used, then

$$m(p) = \arg \min_{\forall m} (\max\{u \cdot f_{pm}^m, (1-u) \cdot f_{pm}^t\}).$$

When the weighted linear distance function is used, then $m(p) = \arg \min_{\forall m} (u \cdot f_{pm}^m + (1-u) \cdot f_{pm}^t)$.

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

$$\text{PS}_{p,m(p)} = \frac{w_p}{t_{p,m(p)}^{\text{tm}} + t_{p,m(p)}^{\text{tnm}}} \cdot \exp \left[-\frac{\max\{0, \text{DD}_p - t_{m(p)}^{\text{now}} - t_{p,m(p)}^{\text{tm}} - t_{p,m(p)}^{\text{tnm}}\}}{k \cdot \bar{t}} \right]$$

where

$$\bar{t} = \frac{\sum_{p \in \text{UNS}} (t_{p,m(p)}^{\text{tm}} + t_{p,m(p)}^{\text{tnm}})}{|\text{UNS}|}$$

and k is a lookahead parameter. Select the $\{p, m(p)\}$ pair giving the maximum $\text{PS}_{p,m(p)}$, i.e. $\{p^*, m(p^*)\} = \arg \max_{p \in \text{UNS}} \{\text{PS}_{p,m(p)}\}$.

Step 5. Assign part p^* to machine $m(p^*)$. Update the current time on machine $m(p^*)$ such that $t_{m(p^*)}^{\text{now}} = t_{m(p^*)}^{\text{now}} + t_{p^*,m(p^*)}^{\text{tm}} + t_{p^*,m(p^*)}^{\text{tnm}}$. The remaining life of the tools used by part p^* are updated as $R_{jm(p^*)} = R_{jm(p^*)} - U_{p^*,ijm(p^*)}$.

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 second stage, we first calculate the increase in objective functions, f_{pm}^m and f_{pm}^t , for each unscheduled part p when it is assigned to machine m at time t_m^{now} . Since the non-machining cost and weighted tardiness are dynamic, they are recalculated after each assignment. In the third step, we normalize the objectives in order to prevent the dominance of one objective over the other, since they have different ranges. For each unscheduled part, we find the machine giving the minimum increase in the weighted linear or weighted Tchebycheff function of two objectives. The machine giving the minimum increase is the most appropriate machine for that part at the current time. At some later time, another machine may become better for that part since the tool magazine status changes. In the study of Akturk and Ozkan (2001), the most appropriate machine is selected according to a priority index which is a combination of the weighted shortest processing time and minimum slack rules. In their study, the operating costs of machines can be ignored since the machines are identical. In our study, we should consider the differences between the machines in terms of manufacturing cost since the machines are non-identical.

After finding these part–machine pairs, we use a part selection index to select a part–machine pair. The proposed part index considers the weights of parts, the slack, and the machining and non-machining times. The index gives higher priority to the parts having less slack and shorter weighted processing time. The processing time is taken as the sum of the machining and non-machining times. We select the part–machine pair giving the maximum part index and schedule that part. The algorithm continues until all parts are scheduled.

An important feature of the proposed index is the calculation of non-machining times. As opposed to the existing literature, we neither ignore nor approximate them. Based upon the current tool magazine arrangement and the remaining tool lives, we calculate the non-machining times exactly for each part on each machine, since machines differ from each other with respect to the existing tool magazine arrangements at time t^{now} and their tool magazine capacities. Although this increases the computation time requirements, it allows a more accurate portrayal of the operation of CNC machines. The existing CNC technology is quite capable of recording and updating the necessary parameters of the proposed algorithm. Unfortunately, most existing algorithms do not take advantage of this flexibility.

Another important issue in the proposed algorithm 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 consecutive parts that will be scheduled on that machine. Tang and Denardo (1988) proposed the KTNS rule for changing tools in the tool magazine. The tools which are required most by the remaining unscheduled operations are kept on the tool magazine. If a tool is required by most of the operations, it is kept on the tool magazine according to the KTNS rule even if it does not have enough remaining life to perform any operation. 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 when determining the tool to be removed. Akturk and Ozkan (2001) only consider the remaining life of the tools for tool removal. They do not consider the number of operations that could be performed with the tool. We propose a new tool index, TI, using these two indices, which can be calculated as follows for tool j on machine m :

$$\text{TI}_{jm} = R_{jm} \cdot \sum x_{pijm}.$$

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, $\sum x_{pijm}$, 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 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.

The proposed heuristic considers all problem-specific characteristics. The manufacturing cost and total weighted tardiness are considered for selecting the machine. The proposed part index considers the weights, due dates, machining and non-machining times, which affect both of the objective functions. The tool index is important for tool allocation decisions. By changing the relative importance of the objectives, the proposed algorithm can provide alternative solutions.

4. Algorithms from the literature

In order to test the performance of the proposed heuristics with a weighted linear function (PI-I) and with a weighted Tchebycheff function (PI-II), we tried to find algorithms from the literature for comparison. We could not find any algorithm which is specially designed for our problem. Therefore, we selected three algorithms that solve similar problems. Since the selected algorithms do not consider certain aspects of our problem, they are adapted somewhat.

The first algorithm LPT-ARM is adapted from the algorithms used in the study of Kim and Yano (1993). Although the proposed heuristics in that study are for loading problems in FMSs, we thought that we could use some of these rules for selecting the machine and selecting the part that should be assigned to that machine. The part selection index is calculated according to the ARM rule in Kim and Yano. According to the ARM rule, the part with the largest T/S value, which is the ratio of the processing time to the number of tool slots needed for the operation, is selected. Instead of using the number of tool slots needed for the operation, we use the non-machining time, because the tool slots needed for the operations do not reflect the real impact of the non-machining time. The tool magazine status and the tool loading, changing and replacing times should be considered for selection. The machine is selected according to the LPT rule. In the LPT-ARM algorithm, the part with the largest t_{pm}^{tm}/t_{pm}^{tnm} ratio is assigned to the machine that has the minimum load after the part is assigned to it. The basic steps of LPT-ARM are as follows:

- Step 1. $UNS = \{1, 2, \dots, N\}$ and $ALT = \{1, 2, \dots, M\}$.
- Step 2. For each $p \in UNS$ and $m \in ALT$, calculate the T/S value such that $T/S_{pm} = t_{pm}^{tm}/t_{pm}^{tnm}$.
- Step 3. Find $\{p^*, m\}$ giving the maximum T/S_{pm}, i.e. $\{p^*, m\} = \arg \max_{p \in UNS, \forall m} T/S_{pm}$.
- Step 4. Assign part p^* to machine m^* giving the minimum load after the part is assigned to it, where $m^* = \arg \min_{m \in \{1, 2, \dots, M\}} \{t_m^{now} + t_{p^*m}^{tm} + t_{p^*m}^{tnm}\}$.
- Step 5. Assign part p^* to machine m^* . Update the current time on machine m^* such that $t_{m^*}^{now} = t_{m^*}^{now} + t_{p^*m^*}^{tm} + t_{p^*m^*}^{tnm}$. The remaining life of the tools used by part p^* are updated as $R_{jm^*} = R_{jm^*} - U_{p^*ijm^*}$.
- Step 6. Update $UNS = UNS \setminus \{p^*\}$ and $ALT = \{m^*\}$. If $UNS \neq \emptyset$, go to step 2, else stop.

The second algorithm is the list scheduling algorithm of Roh and Kim (1997) (RK). They proposed a priority index that considers the machining time, number of additional tools needed, the average estimated waiting time for a tool and the due date. The part with the largest priority index is assigned to the first available machine. The index is adapted to solve our problem such that the actual non-machining time, which is calculated according to the current status of the tool magazine, is used instead of the estimated waiting time for the tools. The basic steps of the algorithm are as follows:

- Step 1. $UNS = \{1, 2, \dots, N\}$ and $ALT = \{1, 2, \dots, M\}$.
- Step 2. Find the first available machine m^* , i.e. $m^* = \arg \min_{m \in \{1, 2, \dots, M\}} \{t_m^{now}\}$.
- Step 3. Calculate PR_{pm^*} for all $p \in UNS$ where

$$PR_{pm^*} = \frac{w_p \cdot t_{pm^*}^{tnm}}{t_{pm^*}^{tm}} \cdot \max \left[1 - \frac{\max\{DD_p - t_{m^*}^{now} - t_{pm^*}^{tm}, 0\}}{t_{pm^*}^{tnm}}, 0 \right].$$

- Step 4.* Find p^* giving the maximum PR_{pm^*} ($p^* = \arg \max_{p \in \text{UNS}} \{PR_{pm^*}\}$).
- Step 5.* Assign part p^* to machine m^* . Update the current time on machine m^* such that $t_m^{\text{now}} = t_m^{\text{now}} + t_{p^*m^*}^{\text{tm}} + t_{p^*m^*}^{\text{tnm}}$. The remaining life of the tools used by part p^* are updated as $R_{jm^*} = R_{jm^*} - U_{p^*ijm^*}$.
- Step 6.* Update $\text{UNS} = \text{UNS} \setminus \{p^*\}$ and $\text{ALT} = \{m^*\}$. If $\text{UNS} \neq \emptyset$, go to step 2, else stop.

The third algorithm, denoted as K-AS, is from Koulamas (1997), and is modified to consider non-machining times by using the nearest neighbour (NN) rule as discussed by Askin and Standridge (1993). Koulamas' (1997) algorithm is a parallel machine scheduling algorithm which does not consider tooling constraints. In order to deal with the non-machining times, we use the NN rule. When we are sorting the unscheduled parts in the shortest processing time (SPT) order according to Koulamas' algorithm, we use the sum of the machining and non-machining times as the total processing time. At each iteration, an active job is identified for each machine by considering the due dates of the parts. Then a job is selected to be scheduled according to the slack or tardiness values. The modified algorithm considers the effect of tooling constraints on both the non-machining cost and total weighted tardiness. The basic steps of the algorithm are as follows:

- Step 1.* $\text{UNS} = \{1, 2, \dots, N\}$ and $\text{ALT} = \{1, 2, \dots, M\}$.
- Step 2.* For each $p \in \text{UNS}$ and $m \in \text{ALT}$, calculate the sum of the machining and non-machining times which determine the processing time.
- Step 3.* For each machine, sort the unscheduled parts according to the SPT rule. Ties are broken according to the earliest due date (EDD) rule. The sorted part set for machine m is S_m .
- Step 4.* For each machine m , determine an active part, p_m , that can be scheduled to that machine.
- Step 4.1.* If $|S_m| = 1$, then go to step 4.8, else label the first part in the sequence as the active part, part p_a .
- Step 4.2.* If $t_m^{\text{now}} + t_{p_a m}^{\text{tm}} + t_{p_a m}^{\text{tnm}} \geq DD_{p_a}$, then go to step 4.8.
- Step 4.3.* Select the next part, p_2 , from set S_m .
- Step 4.4.* If $t_m^{\text{now}} + t_{p_2 m}^{\text{tm}} + t_{p_2 m}^{\text{tnm}} \geq DD_{p_a}$, go to step 4.8.
- Step 4.5.* If $DD_{p_a} \leq DD_{p_2}$, go to step 4.7.
- Step 4.6.* Label part p_2 as the active part ($p_2 = p_a$). If p_a is the last part in set S_m , go to step 4.8, otherwise return to step 4.2.
- Step 4.7.* If p_2 is the last part in S_m , then go to step 4.8, else return to step 4.3.
- Step 4.8.* Part p_a is the active job ($p_m = p_a$).
- Step 5.* Select only one part-machine pair, $\{p_m, m\}$, among the alternatives. If there exists at least one job that can be scheduled without being late on its respective machine, then schedule the job which has minimum slack. Otherwise, select the job which gives minimum weighted tardiness.
- Step 6.* Assign part p_m to machine m . Update the current time on machine m such that $t_m^{\text{now}} = t_m^{\text{now}} + t_{p_m m}^{\text{tm}} + t_{p_m m}^{\text{tnm}}$. The remaining life of the tools used by part p_m are updated as $R_{jm} = R_{jm} - U_{p_m ijm}$.
- Step 7.* Update $\text{UNS} = \text{UNS} \setminus \{p_m\}$ and $\text{ALT} = \{m\}$. If $\text{UNS} \neq \emptyset$, go to step 2, else stop.

The five scheduling algorithms are combined with the proposed tool removal index, TI, and the KTNS rules, resulting in 10 different algorithms to be compared.

5. Numerical example

A numerical example will help to clarify the basic steps of the proposed algorithm. We have 10 parts, two parallel unrelated CNC machines and 10 different tool types. The number of operations for parts 1–10 are (5, 3, 5, 4, 3, 4, 5, 3, 4, 4), respectively. The tool magazine sizes are deliberately chosen to be small (4 and 5) to highlight the algorithm with fewer iterations. The available horsepowers are 3 and 5, and the operating costs are \$2.0 and 4.0 per unit time for machines 1 and 2, respectively. The tool changing, loading and replacing times of each tool type on each machine are selected randomly from distributions $UN \sim [0.1, 0.2]$, $UN \sim [0.2, 0.4]$ and $UN \sim [0.4, 0.8]$, respectively. There are two tool alternatives for each operation which are selected randomly from 10 different tool types. The tooling costs are (1.2, 1.7, 1.4, 2.0, 1.4, 1.7, 1.1, 1.5, 1.2, 1.8) for tools 0–9, respectively. The operation related parameters, D_{pi} and L_{pi} , are selected randomly from the interval $UN \sim [2.5, 5.0]$ and $UN \sim [4, 8]$, respectively. The last operation of each part is the finishing operation and the other operations are the 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. In the first stage, the GP model is solved for each operation–tool pair in order to find the tool alternative giving the minimum cost. The tools giving the minimum cost for each operation of each part, the usage rates of tools and the machining times on each machine can be seen in table 1. Different tool types can give minimum cost at different machines such as the last operation of part 7. Tool type 9 gives the minimum cost on machine 1, whereas tool type 2 gives the minimum cost on machine 2. The due dates and weights of parts are also given in the same table.

In the second stage, we will schedule the parts. The steps of this stage are as follows:

- (1) $UNS = \{1, 2, \dots, 10\}$ and $ALT = \{1, 2\}$.
- (2) The machining, tooling and non-machining costs, the manufacturing cost and the weighted tardiness for all $p \in UNS$ and $m \in ALT$ can be seen in table 2.

One example calculation for part 1 on machine 1 is as follows:

$$\begin{aligned}
 C_{11}^m &= Co_1 \cdot t_{pm}^{tm} = 2.0 \cdot 6.58 = 13.16 \\
 C_{11}^{tool} &= \sum_{i,j} Ct_j \cdot U_{1ij1} \\
 &= (1.1 \cdot 0.01 + 1.4 \cdot 0.10 + 1.8 \cdot 0.18 + 1.8 \cdot 0.11 + 1.1 \cdot 0.11) = 0.79 \\
 C_{11}^{nm} &= Co_1 \cdot t_{11}^{tnm} \\
 &= Co_1 \cdot [(tc_6 + tl_6) + (tc_4 + tl_4) + (tc_9 + tl_9) + (tc_6)] \\
 &= 2.0 \cdot [(0.38 + 0.13) + (0.36 + 0.19) + (0.34 + 0.17) + (0.13)] \\
 &= 3.40
 \end{aligned}$$

| Part | Machine 1 | | | | Machine 2 | | | | w_p |
|------|-----------------|--------------------------------|------------|-----------------|--------------------------------|------------|-----------------|---|-------|
| | Tools | Usage rates | t_{pm}^m | Tools | Usage rates | t_{pm}^m | DD _p | | |
| 1 | (6, 4, 9, 9, 6) | (0.01, 0.10, 0.18, 0.11, 0.11) | 6.58 | (6, 4, 9, 9, 6) | (0.04, 0.25, 0.40, 0.25, 0.25) | 3.41 | 18.2 | 2 | |
| 2 | (7, 3, 6) | (0.12, 0.03, 0.17) | 3.08 | (7, 3, 6) | (0.26, 0.07, 0.38) | 1.64 | 18.2 | 2 | |
| 3 | (9, 7, 4, 9, 3) | (0.16, 0.19, 0.13, 0.14, 0.24) | 9.34 | (9, 7, 4, 9, 3) | (0.36, 0.40, 0.31, 0.31, 0.45) | 4.94 | 16.7 | 4 | |
| 4 | (9, 7, 3, 6) | (0.14, 0.11, 0.05, 0.20) | 5.28 | (6, 7, 3, 6) | (0.09, 0.23, 0.12, 0.45) | 2.86 | 11.2 | 6 | |
| 5 | (3, 6, 4) | (0.03, 0.04, 0.46) | 3.54 | (3, 6, 4) | (0.07, 0.09, 0.69) | 1.93 | 19.5 | 4 | |
| 6 | (6, 9, 4, 6) | (0.03, 0.15, 0.05, 0.20) | 5.94 | (6, 9, 4, 6) | (0.07, 0.35, 0.12, 0.46) | 3.07 | 11.3 | 8 | |
| 7 | (5, 4, 3, 2, 9) | (0.14, 0.07, 0.05, 0.02, 0.57) | 7.81 | (5, 4, 3, 2, 2) | (0.36, 0.18, 0.12, 0.05, 0.45) | 4.40 | 6.5 | 2 | |
| 8 | (8, 6, 8) | (0.04, 0.07, 0.12) | 7.13 | (8, 6, 8) | (0.12, 0.16, 0.36) | 3.79 | 5.6 | 4 | |
| 9 | (6, 3, 6, 3) | (0.04, 0.08, 0.04, 0.31) | 6.52 | (6, 3, 6, 3) | (0.10, 0.18, 0.09, 0.63) | 3.41 | 20.0 | 7 | |
| 10 | (4, 7, 6, 4) | (0.10, 0.11, 0.03, 0.34) | 6.44 | (4, 6, 6, 4) | (0.24, 0.08, 0.07, 0.60) | 3.45 | 5.3 | 5 | |

Table 1. Numerical example parameters.

| Part | Machine 1 | | | | Machine 2 | | | |
|------|------------|-----------------|---------------|----------|------------|-----------------|---------------|----------|
| | C_{pm}^m | C_{pm}^{tool} | C_{pm}^{nm} | f_{pm} | C_{pm}^m | C_{pm}^{tool} | C_{pm}^{nm} | f_{pm} |
| 1 | 13.16 | 0.79 | 3.40 | 17.35 | 13.64 | 1.83 | 6.56 | 22.03 |
| 2 | 6.16 | 0.42 | 3.04 | 9.62 | 6.56 | 0.94 | 5.64 | 13.14 |
| 3 | 18.68 | 1.48 | 4.48 | 24.64 | 19.76 | 3.14 | 9.16 | 32.06 |
| 4 | 10.56 | 0.73 | 4.06 | 15.35 | 11.44 | 1.17 | 6.28 | 18.89 |
| 5 | 7.08 | 0.74 | 3.28 | 11.10 | 7.72 | 1.20 | 5.68 | 14.60 |
| 6 | 11.88 | 0.59 | 3.40 | 15.87 | 12.28 | 1.38 | 6.56 | 20.22 |
| 7 | 15.62 | 1.49 | 5.42 | 22.53 | 17.60 | 1.80 | 7.96 | 27.36 |
| 8 | 14.26 | 0.26 | 2.24 | 16.76 | 15.16 | 0.75 | 4.08 | 19.99 |
| 9 | 13.04 | 0.86 | 2.82 | 16.72 | 13.64 | 1.82 | 5.08 | 20.54 |
| 10 | 12.88 | 0.81 | 3.36 | 17.05 | 13.80 | 1.34 | 4.24 | 19.38 |

Table 2. Numerical example: Step 2.

$$f_{11}^m = C_{11}^m + C_{11}^{\text{tool}} + C_{11}^{\text{nm}} = 13.16 + 0.79 + 3.40 = 17.35$$

$$\begin{aligned} f_{11}^t &= w_1 \cdot \max\{0, t_1^{\text{now}} + t_{11}^{\text{tm}} + t_{11}^{\text{tmm}} - \text{DD}_1\} \\ &= 2.0 \cdot \max\{0, 0 + 6.58 + 1.70 - 18.2\} = 0. \end{aligned}$$

Step 3. First, we normalize the objectives by finding the minimum and maximum of f_{pm}^m and f_{pm}^t , then calculate the weighted linear function of the normalized objectives in order to determine the machine for each unscheduled part. The weighted linear function of the two objectives can be seen in table 2. The normalized weighted linear function for part 1 and machine 1 is calculated as follows:

$$\begin{aligned} f_{11} &= u \cdot \frac{f_{11}^m - \min f^m}{\max f^m - \min f^m} + (1 - u) \cdot \frac{f_{11}^t - \min f^t}{\max f^t - \min f^t} \\ &= 0.5 \cdot \frac{17.35 - 9.62}{32.06 - 9.62} + 0.5 \cdot \frac{0.0 - 0.0}{14.10 - 0.0} \\ &= 0.17. \end{aligned}$$

Since $f_{12} = 0.28$, machine 1 gives the minimum increase. That is, the most suitable machine for part 1 at this step is machine 1, $m_1 = 1$. The machines giving the minimum weighted linear function are (1, 1, 1, 1, 1, 2, 2, 1, 2) for parts 1–10, respectively.

Step 4. We calculate the part index for all $\{p, m(p)\}$ pairs. One example calculation for $\{1, 1\}$ is

$$\text{PS}_{11} = \frac{2}{6.58 + 1.70} \cdot \exp\left\{-\frac{\max\{0, 18.2 - 0 - 6.58 - 1.70\}}{2 \cdot 6.823}\right\} = 0.12.$$

In the first stage, the $\text{PS}_{p,m(p)}$ values for pairs $\{2, 1\}$, $\{3, 1\}$, $\{4, 1\}$, $\{5, 1\}$, $\{6, 1\}$, $\{7, 2\}$, $\{8, 2\}$, $\{9, 1\}$, $\{10, 2\}$ are 0.16, 0.24, 0.62, 0.27, 0.80, 0.31, 0.78, 0.36, 1.05, respectively, which can also be seen in table 3. The $\{10, 2\}$ pair gives the maximum $\text{PS}_{p,m(p)}$ value.

Step 5. We assign part 10 to machine 2. The current time on machine 2 is updated as $t_2^{\text{now}} = 0 + 3.45 + 1.06 = 4.51$. The current status of the tool magazine of machine 2 is (4) – (–, –, –, 6, –) where tool 4 is on the tool holder, tool 6 is in the fourth slot on the tool magazine and other tool slots are empty. The remaining lives are 0.16 and 0.85 for tools 4 and 6, respectively, i.e. (0.16) – (–, –, –, 0.85, –).

Step 6. $\text{UNS} = \text{UNS} \setminus \{10\}$. Machine 2 is the altered machine, $\text{ALT} = \{2\}$. We return to step 2.

Since only the current status of the tool magazine of machine 2 has changed, for the next step the objective function calculations are only performed for machine 2. Due to space limitations, the calculations in the remaining steps will not be shown in detail. In table 3, we can see the $\{p, m(p)\}$ pairs and $\text{PS}_{p,m(p)}$ values at each step. The current status of the altered machine at each step is shown in table 4. The total manufacturing cost is 174.69 and the total weighted tardiness is 129.68.

In order to see the effect of the proposed tool index, TI, we solve the same problem using the KTNS rule. When the KTNS rule is used, the tool loading and hence the scheduling decisions are affected. Up to stage 8, the part loading and tool

| Stage | $\{p, m(p)\}$ | $(PS_{p,m(p)})$ |
|-------|---|---|
| 1 | $\{(1, 1), \{2, 1\}, \{3, 1\}, \{4, 1\}, \{5, 1\}, \{6, 1\}, \{7, 2\}, \{8, 2\}, \{9, 1\}, \{10, 2\}\}$ | $(0.12, 0.16, 0.24, 0.62, 0.27, 0.80, 0.31, 0.78, 0.36, \mathbf{1.05})$ |
| 2 | $\{(1, 1), \{2, 1\}, \{3, 1\}, \{4, 1\}, \{5, 1\}, \{6, 1\}, \{7, 1\}, \{8, 1\}, \{9, 1\}\}$ | $(0.13, 0.18, 0.25, 0.64, 0.31, \mathbf{0.83}, 0.19, 0.48, 0.41)$ |
| 3 | $\{(1, 1), \{2, 1\}, \{3, 1\}, \{4, 2\}, \{5, 1\}, \{7, 2\}, \{8, 2\}, \{9, 1\}\}$ | $(0.21, 0.28, 0.37, \mathbf{1.17}, 0.49, 0.31, 0.87, 0.63)$ |
| 4 | $\{(1, 1), \{2, 1\}, \{3, 1\}, \{5, 1\}, \{7, 1\}, \{8, 2\}, \{9, 1\}\}$ | $(0.22, 0.30, 0.37, 0.53, 0.20, \mathbf{0.87}, 0.66)$ |
| 5 | $\{(1, 1), \{2, 1\}, \{3, 1\}, \{5, 1\}, \{7, 1\}, \{9, 1\}\}$ | $(0.22, 0.31, 0.37, 0.54, 0.20, \mathbf{0.67})$ |
| 6 | $\{(1, 1), \{2, 1\}, \{3, 2\}, \{5, 1\}, \{7, 1\}\}$ | $(0.28, 0.52, 0.59, \mathbf{0.99}, 0.21)$ |
| 7 | $\{(1, 1), \{2, 2\}, \{3, 2\}, \{7, 1\}\}$ | $(0.28, \mathbf{0.69}, 0.59, 0.21)$ |
| 8 | $\{(1, 1), \{3, 2\}, \{7, 1\}\}$ | $(0.28, \mathbf{0.58}, 0.21)$ |
| 9 | $\{(1, 1), \{7, 1\}\}$ | $(\mathbf{0.28}, 0.21)$ |
| 10 | $\{\{7, 1\}\}$ | $(\mathbf{0.20})$ |

Table 3. Part index values.

| Stage | Part | ALT | t_m^{now} | Status of tool magazine | Remaining tool lives |
|-------|------|-----|--------------------|-------------------------|---|
| 1 | 10 | 2 | 4.51 | (4) - (-, -, -, 6, -) | (0.16) - (-, -, -, 0.85, -) |
| 2 | 6 | 1 | 7.64 | (6) - (-, 9, 4, -) | (0.77) - (-, 0.85, 0.95, -) |
| 3 | 4 | 2 | 8.71 | (6) - (-, -, 7, 4, 3) | (0.31) - (-, -, 0.77, 0.16, 0.88) |
| 4 | 8 | 2 | 13.29 | (8) - (-, 6, 7, 4, 3) | (0.52) - (-, 0.15, 0.77, 0.16, 0.88) |
| 5 | 9 | 1 | 15.06 | (3) - (-, 9, 4, 6) | (0.61) - (-, 0.85, 0.95, 0.69) |
| 6 | 5 | 1 | 18.92 | (4) - (-, 9, 6, 3) | (0.49) - (-, 0.85, 0.65, 0.58) |
| 7 | 2 | 2 | 15.66 | (6) - (3, 6, 8, 4, 7) | (0.62) - (0.81, 0.15, 0.52, 0.16, 0.51) |
| 8 | 3 | 2 | 22.57 | (3) - (9, 6, 6, 7, 4) | (0.36) - (0.33, 0.15, 0.62, 0.11, 0.69) |
| 9 | 1 | 1 | 26.12 | (6) - (-, 4, 9, 3) | (0.53) - (-, 0.39, 0.56, 0.58) |
| 10 | 7 | 1 | 36.30 | (9) - (2, 5, 9, 4) | (0.43) - (0.98, 0.86, 0.56, 0.32) |

Table 4. Status of the tool magazine at each stage: TI.

loading decisions are the same since the tool magazine is not fully loaded until that stage. At the eighth stage, the tool loading, and hence the scheduling decisions change. When TI is used, the objective function values are (15.19, 15.84), (21.71, 4.86), (22.46, 50.84), (30.78, 23.48), (20.63, 43.98) and (29.44, 32.14) for part-machine pairs $\{1, 1\}$, $\{1, 2\}$, $\{3, 1\}$, $\{3, 2\}$, $\{7, 1\}$, $\{7, 2\}$, respectively. When the KTNS rule is used, the objective function values of $\{3, 2\}$ and $\{7, 2\}$ change due to the change in non-machining times. For example, the non-machining time for part 7 on machine 2 is calculated as

$$\begin{aligned} t_{72}^{\text{nm}} &= (\text{tr}_8 + \text{tc}_5) + (\text{tr}_4 + \text{tc}_4) + (\text{tc}_3) + (\text{tr}_5 + \text{tc}_2) \\ &= (0.54 + 0.15) + (0.57 + 0.16) + (0.20) + (0.73 + 0.16) = 2.51, \end{aligned}$$

when TI is used. It is

$$\begin{aligned} t_{72}^{\text{nm}} &= (\text{tr}_8 + \text{tc}_5) + (\text{tr}_7 + \text{tc}_4) + (\text{tc}_3) + (\text{tr}_5 + \text{tc}_2) \\ &= (0.54 + 0.15) + (0.79 + 0.16) + (0.20) + (0.73 + 0.16) = 2.73, \end{aligned}$$

when KTNS is used. When TI is used, the number of operations that can be performed with the remaining tool lives for tools (3, 6, 6, 4, 7) on the tool magazine are (2, 1, 2, 0, 1). When the KTNS rule is used, the remaining tool lives are not considered. The number of operations that require the corresponding tools are calculated as (2, 2, 2, 2, 1) for tools (3, 6, 6, 4, 7). When TI is used, tool 4 on the tool magazine is replaced with a new copy to perform the second operation. When the KTNS rule is used, tool 7, which is required by only one of the remaining operations, is removed from the tool magazine. These tool removal decisions affect the decisions at this step and the remaining steps. The objective functions' values increase to (34.18, 26.88) and (30.32, 32.58) for part-machine pairs $\{3, 2\}$ and $\{7, 2\}$, respectively. Obviously the change in tool magazine arrangement will affect the non-machining times and, consequently, the scheduling decisions. The most appropriate machine for part 3 is machine 2 when TI is used and machine 1 when KTNS is used. In the remaining steps, part 3 is assigned to machine 1, part 1 is assigned to machine 2 and part 7 is assigned to machine 1. When the KTNS rule is used instead of TI, the manufacturing cost is slightly decreased to 174.45, but the total weighted tardiness is increased to 153.66.

We compare the proposed heuristics with the weighted linear function (PI-I) and with the weighted Tchebycheff function (PI-II) with the scheduling algorithms

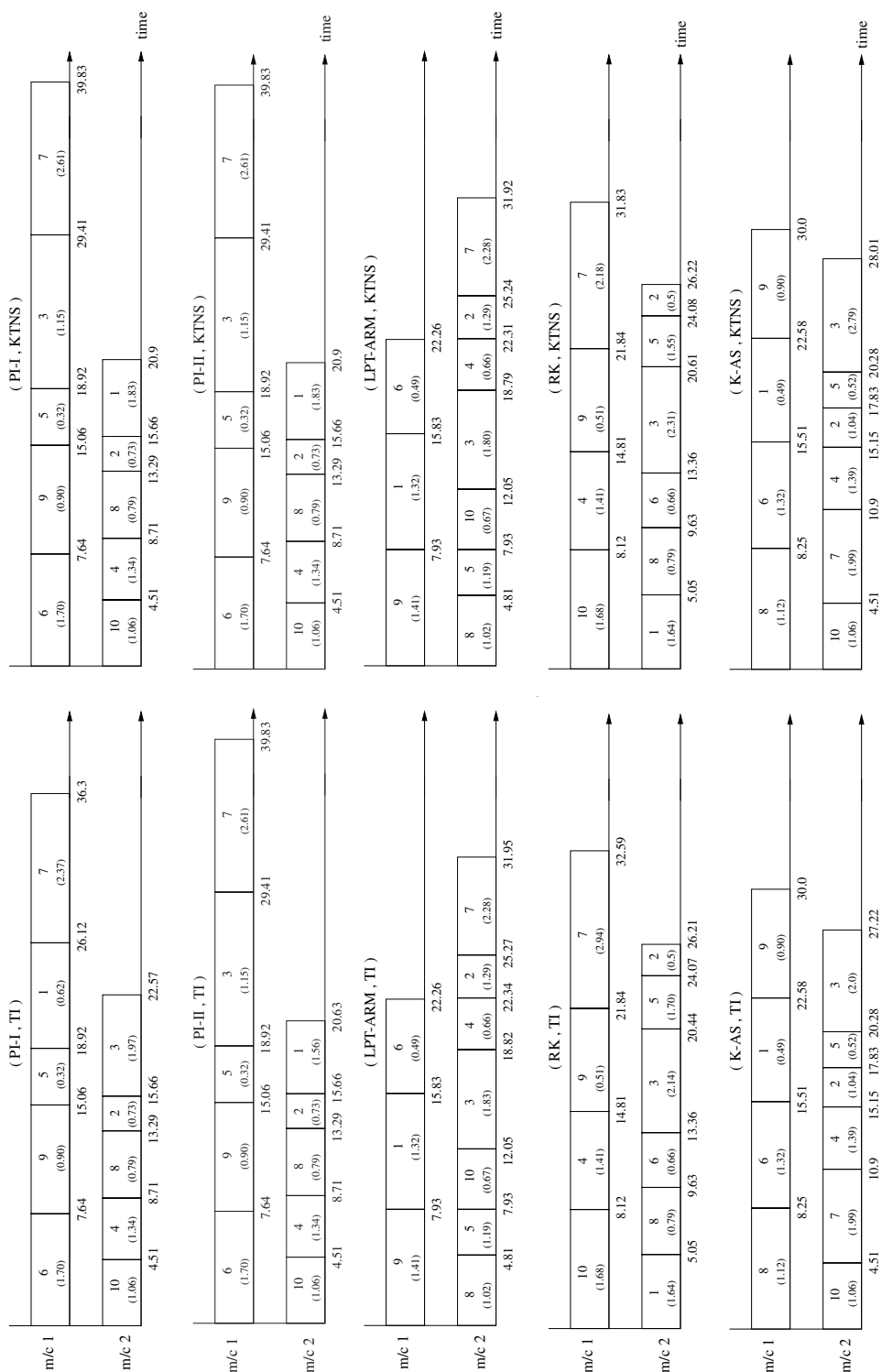


Figure 2. Numerical examples: Gantt charts for all algorithms.

explained in Section 4. The manufacturing cost and total weighted tardiness of the algorithms (PI-II, TI), (PI-II, KTNS), (LPT-ARM, TI), (LPT-ARM, KTNS), (RK, TI), (RK, KTNS), (K-AS, TI) and (K-AS, KTNS) for the numerical example are (173.37, 153.12), (174.45, 153.66), (184.90, 261.79), (184.78, 261.37), (183.15, 182.68), (181.71, 181.96), (180.97, 200.74), (184.13, 203.9), respectively. The Gantt charts for all algorithms can be seen in figure 2. The numbers in parentheses are the non-machining times for the corresponding parts. According to the results, we can see that LPT-ARM, RK and K-AS try to balance the workload of the machines. This increases the manufacturing cost because the number of parts loaded to the second machine, which has a higher operating cost, increases. The non-machining times and hence the total weighted tardiness increase due to more frequent tool changes in the second machine. This is because the tool usage rates for the operations on the second machine are higher. The algorithms PI-I and PI-II give smaller objective values than all other algorithms.

6. Computational results

We performed a computational study to test the performance of the proposed algorithm by comparing it with other algorithms from the literature. All algorithms were coded in the C language and compiled with the GNU C compiler. The problems were solved on a 400 Mhz UltraSPARC station. There are four primary experimental factors that affect the efficiency of our base heuristic which can be seen in table 5. The experimental design is a 2^4 full-factorial design with two levels each. We took five replications for each factor combination resulting in 80 different randomly generated runs.

The number of operations per part, factor A, affects the size of, and load on the system. The operating cost, factor B, which depends on the labour, electricity, maintenance and depreciation costs, affects the part loading decisions, hence the machining and non-machining costs. Factor C, which determines the tool changing, loading and replacement times, affects the non-machining times, hence the tool loading decisions and the non-machining cost. The tool loading and replacement times are calculated according to the robot travel time between the machine and the tool storage area. Factor D, the due date tightness factor, affects the weighted tardiness objective. 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 as

$$\text{average makespan} = \frac{(\sum_p \sum_m t_{pm}^{tm}/M) + (N \cdot \text{average non-machining time})}{M}.$$

The first and third factors affect the average non-machining times which are estimated as 1.25, 6, 2.5 and 12 for the factor combinations (0, 0), (0, 1), (1, 0) and (1, 1), respectively.

The other variables are assumed to be fixed parameters. We have three non-identical parallel CNC machines and 50 parts. The CNC machines in industry usually have different tool magazine capacities and horsepowers. Therefore, the horsepower of the machines is selected randomly from the interval $UN \sim [3, 5]$. The tool magazine capacities are selected from $UN \sim [10, 15]$. The weight for each part is selected from the integer interval $UN \sim [1, 3]$. There are two tool alternatives for each operation which are selected uniformly from 10 different tool types. The tooling costs, C_t , which depend on the tool material and the useful tool life, are

| Factor | Definition | Low level | High level |
|--------|---|--|--|
| A | Number of operations per part | $UN \sim [2, 6]$ | $UN \sim [6, 10]$ |
| B | Operating cost, Co_m | $UN \sim [1, 3]$ | $UN \sim [6, 10]$ |
| C | tool change times, tc_j tool loading times, t_{lj} tool replacing times, tr_j | $UN \sim [0.15, 0.25]$ $UN \sim [0.4, 0.6]$ $UN \sim [0.75, 0.95]$ | $UN \sim [0.75, 1.25]$ $UN \sim [2, 3]$ $UN \sim [3.75, 4.25]$ |
| D | Due date tightness | Loose $UN \sim [0.2, 0.8] \cdot \text{average makespan}$ | Tight $UN \sim [0.1, 0.4] \cdot \text{average makespan}$ |

Table 5. Experimental design factors.

selected from $UN \sim [0.30, 0.50]$. 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. The specific constants for the tools, which change according to the tool type and the parts' material, are obtained from machining handbooks.

We compare the proposed heuristics, PI-I and PI-II, with the scheduling algorithms, LPT-ARM, RK and K-AS. The proposed tool index, TI, is also compared with the KTNS rule from the literature.

The first performance measure we consider is the weighted linear function of the two objective function values for different weights, z . In table 6, we can see the average of the weighted linear functions of normalized objective values over 80 runs for all algorithms. Since the proposed algorithms, PI-I and PI-II, use different weights, u , ranging from 0.1 to 0.9 for the objectives, there are nine different solutions among which we select the best one according to the weights, z , used for comparisons. For the other algorithms, we have only one solution for each run. As we can see from the results, the proposed algorithm using the weighted linear distance function, PI-I, together with the proposed tool index, TI, gives the minimum average value for all weights, z . The second-best algorithm is PI-II using the proposed tool index. The proposed algorithms, PI-I and PI-II, using the KTNS rule also perform better than all other algorithms. This shows that the proposed scheduling algorithms are better than all other algorithms tested regardless of the tool index used. As the relative importance of the manufacturing cost increases, the gap between the proposed algorithms and the other algorithms increases. This means the performance of the other algorithms declines when the manufacturing cost becomes more important since they do not consider the machine operating costs. As we stated earlier, balancing the workload among the machines is more important for the other algorithms. K-AS, which considers the machining and non-machining times and the due dates of parts, is the third-best algorithm. For smaller z values, RK is better than LPT-ARM since it considers the slack in index calculations. As the importance of the second objective declines, LPT-ARM becomes better than RK.

In order to see the significance of the differences between the scheduling algorithms and the tool indices, we take the weighted linear combination of two func-

| Weight z | PI-I | | PI-II | | LPT-ARM | | RK | | K-AS | |
|---------------|--------|--------|--------|--------|---------|--------|--------|--------|--------|--------|
| | TI | KTNS | TI | KTNS | TI | KTNS | TI | KTNS | TI | KTNS |
| 0.1 | 0.0692 | 0.0802 | 0.0717 | 0.0832 | 0.2386 | 0.2583 | 0.1541 | 0.1869 | 0.1647 | 0.1827 |
| 0.2 | 0.1226 | 0.1402 | 0.1280 | 0.1473 | 0.2840 | 0.3132 | 0.2227 | 0.2744 | 0.2176 | 0.2441 |
| 0.3 | 0.1681 | 0.1953 | 0.1787 | 0.2078 | 0.3294 | 0.3682 | 0.2912 | 0.3619 | 0.2706 | 0.3054 |
| 0.4 | 0.2020 | 0.2424 | 0.2168 | 0.2622 | 0.3748 | 0.4231 | 0.3598 | 0.4494 | 0.3236 | 0.3668 |
| 0.5 | 0.2230 | 0.2835 | 0.2422 | 0.3101 | 0.4202 | 0.4781 | 0.4284 | 0.5368 | 0.3766 | 0.4281 |
| 0.6 | 0.2336 | 0.3152 | 0.2518 | 0.3460 | 0.4655 | 0.5330 | 0.4970 | 0.6243 | 0.4295 | 0.4895 |
| 0.7 | 0.2248 | 0.3351 | 0.2399 | 0.3685 | 0.5109 | 0.5879 | 0.5656 | 0.7118 | 0.4825 | 0.5508 |
| 0.8 | 0.1932 | 0.3451 | 0.2045 | 0.3805 | 0.5563 | 0.6429 | 0.6342 | 0.7992 | 0.5355 | 0.6122 |
| 0.9 | 0.1470 | 0.3466 | 0.1542 | 0.3817 | 0.6017 | 0.6978 | 0.7027 | 0.8867 | 0.5884 | 0.6735 |

Table 6. Weighted linear combination of two objectives for different weights.

| | Paired differences | | | | | t |
|------------------------------|--------------------|----------------|-----------------|--------|-------|-------|
| | Mean | Std. deviation | Std. error mean | 95% CI | | |
| | | | | Lower | Upper | |
| (PI-I,K,TNS)-(PI-I,TI) | 0.061 | 0.055 | 0.006 | 0.048 | 0.073 | 9.93 |
| (PI-II,K,TNS)-(PI-II,TI) | 0.068 | 0.058 | 0.006 | 0.055 | 0.081 | 10.54 |
| (LPT-ARM,K,TNS)-(LPT-ARM,TI) | 0.058 | 0.065 | 0.007 | 0.043 | 0.072 | 7.94 |
| (RK,K,TNS)-(RK,TI) | 0.108 | 0.069 | 0.008 | 0.093 | 0.124 | 14.15 |
| (K-AS,K,TNS)-(K-AS,TI) | 0.052 | 0.059 | 0.007 | 0.039 | 0.065 | 7.88 |
| (PI-II,TD)-(PI-I,TI) | 0.019 | 0.023 | 0.003 | 0.014 | 0.024 | 7.31 |
| (LPT-ARM,TD)-(PI-I,TI) | 0.197 | 0.065 | 0.007 | 0.183 | 0.212 | 26.98 |
| (RK,TD)-(PI-I,TI) | 0.205 | 0.069 | 0.008 | 0.190 | 0.221 | 26.72 |
| (K-AS,TD)-(PI-I,TI) | 0.154 | 0.065 | 0.007 | 0.139 | 0.168 | 21.15 |
| (LPT-ARM,TD)-(PI-II,TI) | 0.178 | 0.065 | 0.007 | 0.164 | 0.192 | 24.47 |
| (RK,TD)-(PI-II,TI) | 0.186 | 0.068 | 0.008 | 0.171 | 0.201 | 24.40 |
| (K-AS,TD)-(PI-II,TI) | 0.134 | 0.062 | 0.007 | 0.121 | 0.148 | 19.41 |

Table 7. Paired t -tested (linear, $w = 0.5$).

tions with $z = 0.5$ and perform a paired t -test between the algorithms. The results of the paired t -test can be seen in table 7. In the first five rows, we can see the significance of the differences between the algorithms using TI and KTNS. The algorithms give better solutions when they use the proposed tool index, TI, instead of KTNS. Next, we compare the scheduling algorithms using TI. The next four rows show the relative performance of PI-I using TI with respect to other scheduling algorithms. Our computational results indicate that PI-I performs significantly better than the other algorithms, but the difference between PI-I and PI-II is small. The largest difference is between PI-I and RK. The last three rows show that PI-II performs better than LPT-ARM, K-AS and RK.

The last performance measure is the running time of the algorithms. The average running times of algorithms (PI-I, TI), (PI-I, KTNS), (PI-II, TI), (PI-II, KTNS), (LPT-ARM, TI), (LPT-ARM, KTNS), (RK, TI), (RK, KTNS), (K-AS, TI), and (K-AS, KTNS) are 0.429, 0.951, 0.460, 1.101, 0.019, 0.037, 0.015, 0.034, 0.024, 0.042 seconds, respectively. Since the first four algorithms are solved for nine different weights, u , and are more sophisticated than the other algorithms, the running times are higher. The most time-consuming part of the proposed algorithms is calculating the increase in objective function values in order to find the most appropriate machine for each part. In LPT-ARM and RK, the parts are allocated to the first available machine. The results show the importance of machine selection in non-identical parallel machine environments. When the TI rule is used, the run times are about 50% of the run times when KTNS is used. When there is a tool that cannot perform any operation with its remaining life, the other tools in the tool magazine are not considered by the TI rule. But, since the KTNS rule does not consider the actual tool lives, it must often calculate the number of operations that require the corresponding tool for all tools in the tool magazine regardless of whether the remaining tool life is enough to perform the given operation. Therefore, the proposed tool index, TI, is not only more efficient than the KTNS rule, but also requires less computation time.

7. Concluding remarks

In this study, we consider a scheduling problem in a flexible manufacturing cell with non-identical parallel CNC machines, i.e. machines with different horsepower and tool magazine sizes. Flexible manufacturing cells are smaller in size in terms of the number of machines, and limited in scope and complexity compared to the flexible manufacturing systems that incorporate an automated material handling system, such as an AGV system. The tool management problem in CNC machine scheduling becomes more important with recent advances in technology. With improved automated material handling and tool loading, CNC machines with higher capabilities increase the need for a more effective tool management strategy. Since the material handling operations are done automatically, the tool loading and replacement times are reduced. In previous studies, the tools were loaded to the machines in order to process a batch of parts. The tools were not replaced until all the parts in the batch are processed. However, with today's technology, a more flexible approach, in which the tools are replaced when necessary, is more beneficial for the system performance. Therefore, we solve the tool management and scheduling problems simultaneously.

We have the objectives of minimizing the manufacturing cost comprised of machining, non-machining and tooling costs and minimizing the total weighted

tardiness. The first objective is very important for the manufacturer while the total weighted tardiness is important for the customer. In this study, our aim is to minimize these conflicting objectives simultaneously. We combine two objectives into a single objective by using either weighted linear or Tchebycheff functions and finding alternative solutions by assigning different weights to the objectives. The proposed two-stage algorithm is among only a few which consider the tool management problem in scheduling. In the first stage of the algorithm, optimal machining conditions are found by solving the GP model of Akturk and Avci (1996) with a more efficient solution method. In the second stage, the parts and tools are assigned to non-identical parallel CNC machines and parts are scheduled on each machine. We show the effectiveness of the proposed heuristic by comparing it with algorithms from the literature. The proposed heuristics with either a weighted linear function or weighted Tchebycheff function perform very well. We can use the proposed algorithm in the scheduling of flexible cells in a flexible manufacturing system. However, if one would like to extend our problem to consider the flexible manufacturing system as a whole, then the automated material handling activities between the cells should also be incorporated into the problem.

Acknowledgement

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

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.
- AKTURK, M. S. and OZKAN, S., 2001, Integrated scheduling and tool management in flexible manufacturing systems. *International Journal of Production Research*, **39**, 2697–2722.
- AKTURK, M. S., GHOSH, J. B. and KAYAN, R. K., 2002, Scheduling with tool changes to minimize total completion time under controllable machining conditions. Technical Report, Department of Industrial Engineering, Bilkent University, Turkey.
- ALON, N., AZAR, Y., WOEGINGER, G. J. and YADID, T., 1998, Approximation schemes for scheduling on parallel machines. *Journal of Scheduling*, **1**, 55–66.
- ASKIN, R. G. and STANDRIDGE, C. R., 1993, *Modeling Analysis of Manufacturing Systems* (New York: Wiley).
- AZIZOGLU, M. and KIRCA, O., 1999, On the minimization of total weighted flow time with identical and uniform parallel machines. *European Journal of Operational Research*, **113**, 91–100.
- DAS, I. and DENNIS, J. E., 1997, A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, **14**, 63–69.
- FATHI, Y. and BARNETTE, K. W., 2002, Heuristic procedures for the parallel machine problem with tool switches. *International Journal of Production Research*, **40**, 151–164.
- GORCZYCA, F. E., 1987, *Application of Metal Cutting Theory* (Industrial Press).
- GRAY, A. E., SEIDMAN, A. and STECKE, K. E., 1993, A synthesis of decision models for tool management in automated manufacturing. *Management Science*, **39**, 549–567.
- KIM, Y.-D. and YANO, C. A., 1993, Heuristic approaches for loading problems in flexible manufacturing systems. *IIE Transactions*, **25**, 26–38.
- KOULAMAS, C., 1997, Decomposition and hybrid simulated annealing heuristics for the parallel machine total tardiness problem. *Naval Research Logistics*, **44**, 109–125.
- MACCHIAROLI, R. and RIEMMA, S., 1996, Design of a tool management system in a flexible cell. *International Journal of Production Research*, **34**, 767–784.
- RARDIN, R. L., 1998, *Optimization in Operations Research* (Prentice Hall).

- RANDHAWA, S. U. and KUO, C.-H., 1997, Evaluating scheduling heuristics for non-identical parallel processors. *International Journal of Production Research*, **35**, 969–981.
- ROH, H.-K. and KIM, Y.-D., 1997, Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter. *International Journal of Production Research*, **35**, 2989–3003.
- 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.
- TIWARI, M. K. and VIDYARTHI, N. K., 2000, Solving machine loading problems in a flexible manufacturing system using a genetic algorithm based heuristic approach. *International Journal of Production Research*, **38**, 3357–3384.