

This article was downloaded by: [University of California Berkeley]

On: 27 September 2009

Access details: Access Details: [subscription number 912891657]

Publisher Taylor & Francis

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## IIE Transactions

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title-content=t713772245>

### Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties

Ayten Turkcan <sup>a</sup>; M. Selim Akturk <sup>b</sup>; Robert H. Storer <sup>c</sup>

<sup>a</sup> Regenstrief Center for Healthcare Engineering, Purdue University, West Lafayette, IN, USA <sup>b</sup> Department of Industrial Engineering, Bilkent University, Ankara, Turkey <sup>c</sup> Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA, USA

Online Publication Date: 01 December 2009

**To cite this Article** Turkcan, Ayten, Akturk, M. Selim and Storer, Robert H. (2009) 'Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties', IIE Transactions, 41:12, 1080 — 1095

**To link to this Article:** DOI: 10.1080/07408170902905995

**URL:** <http://dx.doi.org/10.1080/07408170902905995>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

# Predictive/reactive scheduling with controllable processing times and earliness-tardiness penalties

AYTEN TURKCAN<sup>1,\*</sup>, M. SELIM AKTURK<sup>2</sup> and ROBERT H. STORER<sup>3</sup>

<sup>1</sup>Regenstrief Center for Healthcare Engineering, Purdue University, West Lafayette, IN 47907, USA  
E-mail: aturkcan@purdue.edu

<sup>2</sup>Department of Industrial Engineering, Bilkent University, 06800 Ankara, Turkey

<sup>3</sup>Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015, USA

Received July 2006 and accepted February 2009

In this study, a machine scheduling problem with controllable processing times in a parallel-machine environment is considered. The objectives are the minimization of manufacturing cost, which is a convex function of processing time, and total weighted earliness and tardiness. It is assumed that parts have job-dependent earliness and tardiness penalties and distinct due dates, and idle time is allowed. The problem is formulated as a time-indexed integer programming model with discrete processing time alternatives for each part. A linear-relaxation-based algorithm is used to assign the parts to the machines and to find a sequence on each machine. A non-linear programming model is proposed to find the optimal starting and processing times of the parts for a given sequence. The proposed non-linear programming model is converted to a minimum-cost network flow model by piecewise linearization of the convex manufacturing cost in the objective function. The proposed method is used to find initial schedules in predictive scheduling. The proposed models are revised to incorporate a stability measure for reacting to unexpected disruptions such as machine breakdown, arrival of a new job, delay in the arrival or the shortage of materials in reactive scheduling.

**Keywords:** Scheduling, controllable processing times, earliness and tardiness, reactive scheduling

## 1. Introduction

In most scheduling studies the processing times of jobs are assumed to be known and fixed. However, in many manufacturing applications the processing times can be altered or controlled (albeit at higher cost) by using additional resources (e.g., manpower, fuel) or by changing machining conditions such as cutting speed and feed rate. Controllable processing times provide additional flexibility in finding solutions to the scheduling problem, which in turn can improve the overall performance of the production system. In this study our aim is to show the effectiveness of using controllable processing times in both predictive and reactive scheduling.

Most of the studies on scheduling with controllable processing times assume that the processing time is a linear function of the amount of resource allocated to the processing of the job as summarized in the recent survey of Shabtay and Steiner (2007). Thus, the manufacturing cost increases linearly with decreasing processing time. There are several

studies on scheduling with resource allocation in which the job processing time is a convex decreasing function of the amount allocated to the processing of the job. Shabtay and Kaspi (2004) and Gurel and Akturk (2007) study the problem of minimizing the total weighted flow time on a single machine with controllable processing times using different non-linear compression cost functions, whereas Shabtay and Kaspi (2006) study an identical parallel-machine scheduling problem with a convex resource consumption function to minimize the total completion time. Shabtay *et al.* (2007) consider the case of a convex resource consumption function to minimize the makespan in a two-machine flowshop with a no-wait restriction. In this study, we also assume that the manufacturing cost is a convex function of processing time. In existing studies, scheduling objectives such as makespan, total completion time, total earliness and tardiness are also considered besides minimizing the total compression costs. We consider the objective of minimizing the total weighted earliness and tardiness as the scheduling objective in addition to minimizing the manufacturing cost. When a job is finished earlier than its due date, an inventory holding cost is incurred. The total weighted tardiness objective is important for customer satisfaction. Late deliveries will incur additional

\*Corresponding author

transportation costs and may cause customer dissatisfaction, loss of goodwill and lost sales.

There are several papers that address scheduling problems with the earliness and tardiness criteria and fixed processing times. Baker and Scudder (1990) and Kanet and Sridharan (2000) provide extensive literature surveys on non-regular scheduling problems. Most of the cited studies consider single-machine problems with the assumption of a common due date. Authors considering single-machine problems with distinct due dates include Fry *et al.* (1984), Yano and Kim (1991), Hoogeveen and Van de Velde (1996), Chen and Lin (2002) and Sourd (2005). All of these papers propose a branch-and-bound (B&B) approach. Kedad-Sidhoum *et al.* (2008) consider parallel-machine problems with distinct due dates and propose lower bounds based on relaxation of time-indexed formulations of the problem.

In the earliness–tardiness scheduling literature, there are a few papers that consider controllable processing times in scheduling problems. Panwalkar and Rajagopalan (1992) and Liman *et al.* (1997) consider a single-machine problem with common due date and common due window assumptions, respectively. The common due date/window is a decision variable that is determined by the proposed assignment models in both studies. Alidaee and Ahmadian (1993) consider the unrelated parallel-machine scheduling problem with a common due date assumption and propose a transportation model to solve the problem. Cheng *et al.* (1996) study the unrelated parallel-machine scheduling problem with controllable processing times. The due date is assumed to be unrestrictedly large and the cost is a convex function of the amount that processing times are compressed. The problem is formulated as an assignment problem. All these studies, addressing both controllable processing times and earliness–tardiness penalties, assume some variety of a common due date thus limiting their applicability. This assumption is usually not justified in a manufacturing environment. Furthermore, this assumption simplifies the problem significantly by ignoring the issue of when and where to insert idle times. In this study, we solve an unrelated parallel-machine scheduling problem considering job-dependent earliness and tardiness penalties, inserted idle times, distinct due dates and controllable processing times simultaneously.

In many manufacturing environments, a schedule formed at the beginning of the planning horizon cannot be followed properly due to unexpected events such as machine breakdowns, the arrival of an important job, order cancellation, etc. The schedule often must be revised to restore feasibility, and/or to decrease the impact of the disturbance on system performance. Reactive scheduling is the term most often used to describe the problem of updating the schedule in response to an unexpected disruption. Several different methods such as B&B, dispatching rules and heuristics have been proposed to form full new or partial schedules. A review of the reactive scheduling literature can be found in Vieira *et al.* (2003). The controllable processing times

are not considered in reactive scheduling studies. However, they are commonly used in real manufacturing settings to deal with disturbances. In this study, the use of controllable processing times as an effective way to deal with shop floor disruptions is shown.

As an outline of the remainder of this paper, the problem is defined with its underlying assumptions in Section 2. We formulate the problem as a time-indexed Integer Programming (IP) model. The continuously controllable processing times are discretized in this formulation. Then, we propose a Linear Programming (LP)-relaxation-based algorithm to solve the problem in Section 3. The proposed algorithm uses the solution of the linear relaxation of the IP model to find an assignment of jobs to machines and find the sequence of jobs on each machine. A non-linear programming model is proposed to find optimal starting times and processing times of the parts for a given sequence. The proposed non-linear programming model is converted to a minimum-cost network flow model by piecewise linearization of the convex manufacturing cost in the objective function. In Section 4, the proposed time-indexed model and minimum cost network flow model are updated to consider stability measure in reactive scheduling problems. In Section 5, the effectiveness of controllable processing times in predictive and reactive scheduling problems is shown through computational results. In the last section, concluding remarks and future research directions are provided.

## 2. Problem definition

The notation used throughout the study is as follows.

*Parameters:*

$M$	=	number of machines;
$N$	=	number of jobs;
$N_m$	=	number of jobs assigned to machine $m$ ;
$T$	=	planning horizon, $t = 1, 2, \dots, T$ ;
$D$	=	set of disrupted jobs;
$f_{im}(p_{im})$	=	manufacturing cost, which is a convex function of processing time;
$\tilde{f}_{im}(\cdot)$	=	piecewise linearized manufacturing cost function;
$p_{im}^l, p_{im}^u$	=	lower and upper bounds for processing time of part $i$ on machine $m$ ;
$\epsilon_i$	=	earliness penalty for part $i$ ;
$\tau_i$	=	tardiness penalty for part $i$ ;
$d_i$	=	due date of part $i$ ;
$r_i$	=	release time of part $i$ ;
$t_m^{\text{start}}$	=	starting time period of machine $m$ ;
$ps_{imk}$	=	processing time setting $k$ of part $i$ on machine $m$ ;
$cost_{imtk}$	=	total cost of assigning part $i$ using processing time setting $k$ to time period $t$ on machine $m$ ( $t$ is the starting time period);

- $K_{im}$  = number of processing time settings for part  $i$  on machine  $m$ ;
- $f_{im}^{\min}$  = minimum manufacturing cost of part  $i$  on machine  $m$ ;
- $G_{im}$  = number of line segments in the piecewise linearized manufacturing cost function ( $\tilde{f}_{im}(\cdot)$ );
- $b_{im}^g$  = slope of the line segment  $g$  of the piecewise linearized manufacturing cost function ( $\tilde{f}_{im}(\cdot)$ );
- $\delta_{im}^g$  = breakpoints of the linearized manufacturing cost function ( $\tilde{f}_{im}(\cdot)$ );
- $C_i$  = completion time of part  $i$  in the initial schedule.

*Variables:*

- $p_{im}$  = processing time of part  $i$  on machine  $m$  ( $p_{im}^l \leq p_{im} \leq p_{im}^u$ );
- $S_{im}$  = starting time of part  $i$  on machine  $m$ ;
- $Z_{imtk}$  = binary variable that is equal to one if part  $i$  is assigned to machine  $m$ , starts to be processed at time period  $t$  and uses processing time setting  $k$ ;
- $I_{im}$  = inserted idle time before part  $i$  on machine  $m$ ;
- $E_{im}$  = earliness of part  $i$  on machine  $m$ ;
- $T_{im}$  = tardiness of part  $i$  on machine  $m$ ;
- $a_{im}^g$  = amount of compression in processing time in line segment  $g$  of  $\tilde{f}_{im}(\cdot)$ ;
- $C'_i$  = completion time of part  $i$  in the revised schedule;
- $D_{im}^-, D_{im}^+$  = negative and positive deviations of the completion time of part  $i$  in the revised schedule with respect to the completion time in the initial schedule.

In this study, the unrelated parallel-machine scheduling problem with controllable processing times problem is solved. The objective is to minimize the sum of weighted earliness and tardiness and the manufacturing cost. When part  $i$  is assigned to machine  $m$ , the weighted earliness and tardiness can be calculated as  $(\epsilon_i \max\{0, d_i - (S_{im} + p_{im})\})$  and  $(\tau_i \max\{0, (S_{im} + p_{im}) - d_i\})$ , respectively. The manufacturing cost is a function of processing time ( $p_{im}$ ), which can take any continuous value between the lower bound,  $p_{im}^l$ , and the upper bound,  $p_{im}^u$ . The manufacturing cost is assumed to be convex in this study. The main motivation in using a convex compression cost function is the convexity of the manufacturing cost, which is the sum of machining and tooling costs, in flexible manufacturing systems as discussed in Turkcan et al. (2003). The manufacturing cost, earliness and tardiness values as functions of processing time can be seen in Fig. 1.

In order to solve the unrelated parallel-machine scheduling problem with continuously controllable processing times and earliness–tardiness penalties the decisions that

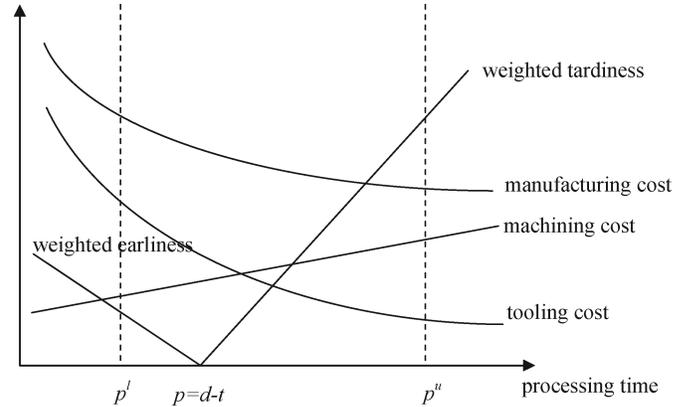


Fig. 1. Objective functions.

should be made are: assignment of parts to machines, determination of the processing times, and sequencing and scheduling of parts on each machine. Furthermore, it may be necessary to have inserted idle times between the processing of consecutive jobs due to consideration of a non-regular performance measure. It is important to note that even the single-machine version of this problem, where the jobs have different due dates and different earliness and tardiness weights with fixed processing times, is strongly NP-hard (Hoogeveen and Van de Velde, 1996). These optimization problems (assignment, sequencing, scheduling, determination of processing times and idle times) are not independent and it is very difficult to make all these decisions simultaneously. Therefore, we propose a two-stage algorithm below.

### 3. The proposed two-stage algorithm

A two-stage algorithm is proposed in order to solve the unrelated parallel-machine scheduling problem with continuously controllable processing times. In the first stage, a time-indexed IP model is used to assign parts to machines and to determine the sequence of parts on each machine. In the second stage, a non-linear programming model is used to determine the optimal start times and processing times for a given sequence of parts on each machine.

#### 3.1. Time-indexed IP model

In the literature, the existing time-indexed models consider fixed processing times for the parts (Van den Akker et al., 2000; Avci, 2001; Kedad-Sidhoum et al. 2008). The proposed time-indexed IP model incorporates alternative processing time settings, which take discrete values between the lower and upper bounds of the processing times for the corresponding parts. Shabtay and Steiner (2007) provide an extensive literature survey of continuously and discretely controllable processing times.

The proposed time-indexed IP model, which allocates parts to the machines, determines starting times and

processing times of the parts, is as follows:

$$(IP) \quad \min \sum_{i=1}^N \sum_{m=1}^M \sum_{k=1}^{K_{im}} \sum_{t=1}^{T-ps_{imk}+1} cost_{imtk} Z_{imtk}, \quad (1a)$$

subject to

$$\sum_{m=1}^M \sum_{k=1}^{K_{im}} \sum_{t=1}^{T-ps_{imk}+1} Z_{imtk} = 1, \quad i = 1, \dots, N, \quad (1b)$$

$$\sum_{i=1}^N \sum_{k=1}^{K_{im}} \sum_{u=\max\{t-ps_{imk}+1,1\}}^{\min\{T-ps_{imk}+1,t\}} Z_{imuk} \leq 1, \\ m = 1, \dots, M \quad \text{and} \quad t = 1, \dots, T, \quad (1c)$$

$$Z_{imtk} \in \{0, 1\}, \quad \forall i, m, t, k, \quad (1d)$$

where  $Z_{imtk}$  is a binary variable that is equal to one when part  $i$  using processing time setting  $k$  starts to be processed at time period  $t$  on machine  $m$ .  $ps_{imk}$  is the  $k$ th processing time setting for part  $i$  on machine  $m$  ( $ps_{i,m,1} < ps_{i,m,2} < \dots < ps_{i,m,K_{im}}$ ).  $cost_{imtk}$ , which is the sum of manufacturing cost and weighted earliness and tardiness, is the cost of assigning part  $i$  with processing time setting  $k$  to time period  $t$  on machine  $m$ . It is calculated as

$$cost_{imtk} = f_{im}(ps_{imk}) + \epsilon_i \max\{0, d_i - (t + ps_{imk} - 1)\} \\ + \tau_i \max\{0, (t + ps_{imk} - 1) - d_i\}.$$

In the proposed IP model, the first constraint (1b) guarantees that every part is scheduled exactly once. The second constraint (1c) is the machine capacity constraint stating that each time unit on each machine can be occupied by at most one part. The size of the IP model depends on the length of the unit time period used to discretize the processing times and the number of processing time settings. As the number of breakpoints increases, the accuracy of the time-indexed IP model increases at the expense of an increase in problem size and computation times.

The proposed time-indexed IP model is a variant of the generalized assignment problem for which an extensive number of studies exist. Cattrysse and Van Wassenhove (1992) provide a survey of algorithms for the generalized assignment problem. Since the LP relaxations of the time-indexed formulations provide strong bounds, time-indexed formulations have received considerable attention from researchers. The solutions found by solving the relaxed time-indexed model can be used to develop algorithms as is done for generalized assignment problems in the literature.

In this study, we propose a two-stage algorithm based on the LP relaxation of the time-indexed IP model. In the first stage, the LP relaxation of the IP model is solved. The LP relaxation might give a non-integral solution. This infeasible solution is used to find an assignment of parts to machines and part sequence on each machine. The maximum  $Z_{imtk}$  value is found for each part  $i$ , i.e.,  $\{m^*, t^*, k^*\} = \arg \max_{m,t,k} Z_{imtk}$ . The part is assigned to

machine  $m^*$ . The parts assigned to each machine are sequenced in non-decreasing order of their starting times,  $t^*$ . After the sequences of the parts on each machine are found, the optimal starting times, processing times of parts and the inserted idle times between parts should be determined in the second stage.

### 3.2. The non-linear programming model

After the LP relaxation of the IP model is solved, the parts are assigned to machines and the sequence on each machine is fixed as discussed above. The following non-linear programming model is solved for each machine  $m$  to determine the final schedule that includes finding both the processing times of the parts and the idle times that should be inserted between each part for a given sequence to minimize the sum of earliness, tardiness and manufacturing costs:

$$(SNLP_m) \quad \min \sum_{i=1}^{N_m} \epsilon_{[i]} E_{[i]m} + \sum_{i=1}^{N_m} \tau_{[i]} T_{[i]m} + \sum_{i=1}^{N_m} f_{[i]m}(p_{[i]m}), \quad (2a)$$

subject to

$$T_{[i]m} - E_{[i]m} = S_{[i]m} + p_{[i]m} - d_{[i]}, \quad \forall i, \quad (2b)$$

$$S_{[1]m} = I_{[1]m}, \quad (2c)$$

$$S_{[i]m} = S_{[i-1]m} + p_{[i-1]m} + I_{[i]m} \quad i = 2, \dots, N_m, \quad (2d)$$

$$p_{[i]m}^l \leq p_{[i]m} \leq p_{[i]m}^u, \quad \forall i, \quad (2e)$$

$$S_{[i]m}, T_{[i]m}, E_{[i]m}, p_{[i]m}, I_{[i]m} \geq 0, \quad \forall i. \quad (2f)$$

The first constraint (2b) calculates the earliness and tardiness value of each part according to the starting time, processing time and the due date of the corresponding part. The second and third constraints (2c) and (2d) are used for determining the starting times of the parts. The starting time of a part in sequence position  $[i]$ , is the sum of the starting time and processing time of the part in sequence position  $[i - 1]$  and the idle time between the parts in sequence positions  $[i - 1]$  and  $[i]$ . The fourth constraint (2e) gives the lower and upper bounds of the processing times. The processing times can take any continuous value between the lower and upper bounds.

**Lemma 1.** *The proposed non-linear programming model has: (i) a convex feasible set; and (ii) a convex objective function.*

**Proof.** We skip the detailed proof due to space limitations; it can, however, be obtained from the first author. We can briefly give the general outline of the proof such that: (i) each constraint is a linear equation or inequality, therefore the feasible region is convex; and (ii) the objective function is the sum of linear and convex functions and therefore it is convex. ■

#### 3.2.1. Piecewise linearization

In the proposed SNLP model, the manufacturing cost, which is a non-linear convex function of processing time,

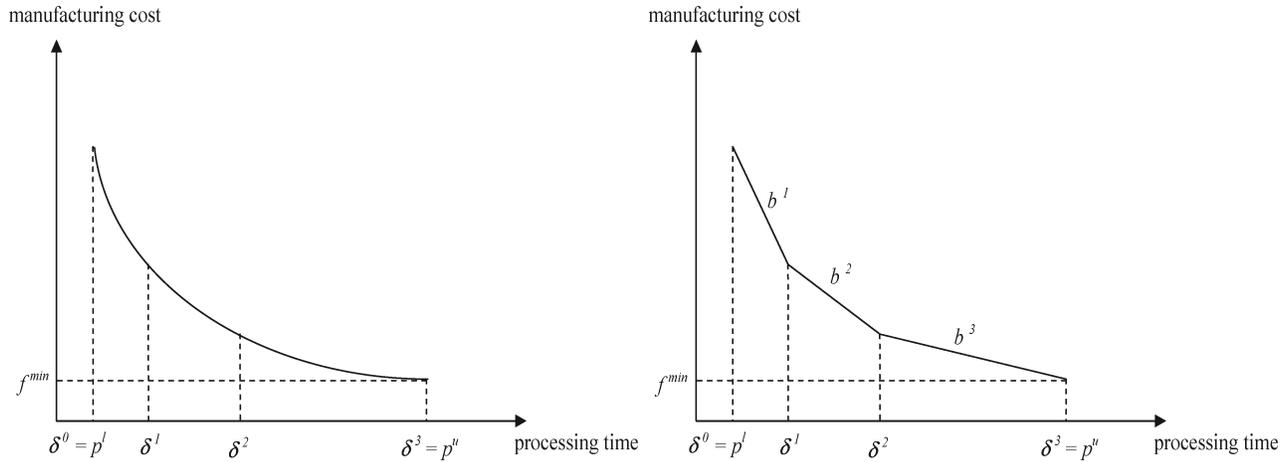


Fig. 2. Linearized manufacturing cost.

can be approximated with a piecewise linear function. The piecewise linearized function can be written as  $f_{[i]m}^{\min} - \sum_{g=1}^{G_{[i]m}} b_{[i]m}^g a_{[i]m}^g$ , where  $f_{[i]m}^{\min}$  is the minimum manufacturing cost,  $a_{[i]m}^g$  is the amount of compression in processing time in line segment  $g$  and  $b_{[i]m}^g$  is the slope of the line segment  $g$ ,  $g = 1, 2, \dots, G_{[i]m}$ . In Fig. 2, one can see an example of the piecewise linearized manufacturing cost function.

In order to find solutions that are close to the real optimal solution, the piecewise linearized function should be a good representation of the real function. The linearized function is found in such a way that the difference between the approximate and real function values at any point is less than a certain percentage ( $NWL$ ) of the real function value, i.e.,  $(\tilde{f}(x) - f(x))/f(x) \leq NWL$ , where  $\tilde{f}(x)$  is the approximate function value and  $f(x)$  is the real function value at point  $x$ . The algorithm used to find the approximate piecewise linear function is as follows.

**Algorithm 1.** Piecewise linearization algorithm

Step 1. Initialize  $g$ ,  $x_1$  and  $x_2$  ( $g = 0$ ,  $x_1 = p_{im}^l$  and  $x_2 = p_{im}^u$ ).

Step 2. Find point  $x^*$  giving the maximum difference between the approximate function,  $\tilde{f}(x)$ , and the real function,  $f(x)$ . Point  $x^*$  is calculated by taking the derivative of the difference, equating it to zero and solving the resulting function for  $x$ .

2.1. The approximate function is

$$\tilde{f}(x) = f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1).$$

2.2.  $x^*$  is the solution of the equation

$$\frac{\partial(\tilde{f}(x) - f(x))}{\partial x} = \frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{\partial f(x)}{\partial x} = 0.$$

Step 3. Calculate the maximum difference between the approximate function,  $\tilde{f}(x)$ , and the real function,  $f(x)$ .

Step 4. If the maximum difference is greater than a certain percentage of the real function value ( $\tilde{f}(x^*) - f(x^*) \geq f(x) \times NWL$ )

4.1. Divide the region between  $x_1$  and  $x_2$  by updating  $x_2$  as  $(x_1 + x_2)/2$ . Go to Step 2.

Step 5. Else if the maximum difference is less than a certain percentage of the real function value ( $\tilde{f}(x^*) - f(x^*) < f(x) \times NWL$ )

5.1. Assign the value of breakpoint  $g$  ( $\delta_{im}^g = x_1$ ) and increase  $g$  by one.

5.2. Update  $x_1$  and  $x_2$  ( $x_1 = x_2$  and  $x_2 = p_{im}^u$ ).

5.3. If  $x_1 \neq x_2$ , go to Step 2.

5.4. Otherwise, update the value of the last breakpoint ( $\delta_{im}^g = x_1$  and  $G_{im} = g$ ).

The time-indexed formulation, presented in Section 3.1, uses a set of discrete processing time alternatives ( $ps_{imk}$ ). Any method can be used to discretize the processing times between the lower and upper bounds. In the computational study section, we use Algorithm 1, the piecewise linearization algorithm, to determine the discrete processing time alternatives that will be used in the time-indexed IP model. That means, we first linearize the convex manufacturing cost function for each part on each machine and then solve the linear relaxation of time-indexed IP model. The processing time settings used in IP are found by dividing the breakpoints of the piecewise linear function to the unit time period and rounding them to the nearest integer values.

*Example (Piecewise linearization):* Suppose the manufacturing cost function of part  $i$  on machine  $m$  is  $f(x) = 2x + 1.5x^{-1.2}$ . The minimum and maximum processing times are 0.1 and 0.7, respectively.  $NWL = 0.2$ . The linearization algorithm works as follows.

Step 1. Initialize breakpoints  $x_1 = p_{im}^l = 0.1$ ,  $x_2 = p_{im}^u = 0.7$  and  $g = 0$ .

Step 2. The approximate linear function between  $x_1$  and  $x_2$  is

$$\begin{aligned} \tilde{f}(x) &= f(x_1) + \frac{f(x_2) - f(x_1)}{x_2 - x_1}(x - x_1) \\ &= 23.97 - 33.79(x - 0.1). \end{aligned}$$

The difference between the approximate function and the real function is  $\tilde{f}(x) - f(x) = 57.76 - 33.79x - 2x - 1.5x^{-1.2}$ . The point  $x^*$  giving the maximum difference is calculated by first taking the derivative of the difference  $(-35.79 + 1.8x^{-2.2})$ , equating it to zero and solving the resulting function for  $x$  ( $x = (35.79/1.8)^{-1/2.2} = 0.26$ ).

Step 3. The maximum difference between the approximate function and real function at point  $x^*$  is  $\tilde{f}(x^*) - f(x^*) = 57.76 - 33.79x^* - 2x^* - 1.5(x^*)^{-1.2} = 10.50$ .

Step 4. Since the maximum difference is greater than a certain percentage of the real function value (maximum difference =  $10.5 \geq f(x^*) \times NWL = 1.64$ ), the region between  $x_1$  and  $x_2$  is divided into two.  $x_1 = 0.1$  and  $x_2 = (x_1 + x_2)/2 = 0.4$ .

Steps 2 to 4 are repeated for  $(x_1 = 0.1, x_2 = 0.4)$ ,  $(0.1, 0.25)$  and  $(0.1, 0.175)$ . When  $x_1 = 0.1$  and  $x_2 = 0.175$ ,  $\tilde{f}(x) - f(x) = 1.78$  is less than  $f(x^*) \times NWL = 3.46$ . Therefore,  $x_1 = 0.1$  becomes the first breakpoint ( $\delta_{im}^0 = 0.1$ ).  $x_1$  and  $x_2$  are updated as 0.175 and 0.7, respectively. Steps 2 to 5 are repeated until all breakpoints are generated. The steps are summarized in Table 1. The breakpoints of the approximate piecewise linear function are found as 0.1, 0.175, 0.306 and 0.7. The discrete processing time settings are found by dividing each breakpoint by the unit time period and rounding it to the nearest integer. If the unit time period is set as 0.1, the discrete processing time settings, which will be used in the time-indexed IP model, will be 1, 2, 3 and 7.

For this example, the minimum manufacturing cost is 3.7 ( $f_{im}^{\min} = 2p_{im}^u + 1.5(p_{im}^u)^{-1.2} = 3.7$ ). The number of lines in the approximate function is three ( $G_{im} = 3$ ). The breakpoints are 0.1, 0.175, 0.306 and 0.7 ( $\delta_{im}^0 = p_{im}^1 = 0.1$ ,  $\delta_{im}^1 = 0.175$ ,  $\delta_{im}^2 = 0.306$  and  $\delta_{im}^3 = p_{im}^u = 0.7$ ). The slope

of line  $g$  is calculated as

$$b_{im}^g = \frac{f_{im}(\delta_{im}^g) - f_{im}(\delta_{im}^{g-1})}{\delta_{im}^g - \delta_{im}^{g-1}},$$

where  $b_{im}^1 = -13.23$ ,  $b_{im}^2 = -3.79$ ,  $b_{im}^3 = -0.82$ .

### 3.2.2. Minimum-cost network flow model

In the proposed SNLP model, the manufacturing cost,  $f_{[i]m}(p_{[i]m})$ , is replaced with  $f_{[i]m}^{\min} - \sum_{g=1}^{G_{[i]m}} b_{[i]m}^g a_{[i]m}^g$ , the processing time,  $p_{[i]m}$ , is replaced with  $p_{[i]m}^u - \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g$ , and the starting time,  $S_{[i]m}$ , is replaced with  $T_{[i]m} - E_{[i]m} - (p_{[i]m}^u - \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g) + d_{[i]}$ . The piecewise linearized cost function provides a LP model that has a special structure. The new model, MCNF $_m$ , is a minimum-cost network flow model for each machine  $m$ .

(MCNF $_m$ )

$$\begin{aligned} \min \sum_{i=1}^{N_m} \epsilon_{[i]} E_{[i]m} + \sum_{i=1}^{N_m} \tau_{[i]} T_{[i]m} + \sum_{i=1}^{N_m} \left( f_{[i]m}^{\min} \right. \\ \left. - \sum_{g=1}^{G_{[i]m}} b_{[i]m}^g a_{[i]m}^g \right), \end{aligned} \tag{3a}$$

subject to

$$T_{[1]m} - E_{[1]m} + \sum_{g=1}^{G_{[1]m}} a_{[1]m}^g - I_{[1]m} = p_{[1]m}^u - d_{[1]}, \tag{3b}$$

$$\begin{aligned} T_{[i]m} - E_{[i]m} - T_{[i-1]m} + E_{[i-1]m} + \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g - I_{[i]m} \\ = p_{[i]m}^u - d_{[i]} + d_{[i-1]}, \quad i = 2, \dots, N_m, \end{aligned} \tag{3c}$$

$$\begin{aligned} -T_{[N_m]m} + E_{[N_m]m} + \sum_{i=1}^{N_m} I_{[i]m} - \sum_{i=1}^{N_m} \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g \\ = - \sum_{i=1}^{N_m} p_{[i]m}^u + d_{[N_m]}, \end{aligned} \tag{3d}$$

$$a_{[i]m}^g \leq \delta_{[i]m}^g - \delta_{[i]m}^{g-1}, \quad \forall i, g, \tag{3e}$$

$$T_{[i]m}, E_{[i]m}, I_{[i]m}, a_{[i]m}^g \geq 0, \quad \forall i, g. \tag{3f}$$

Table 1. Piecewise linearization example

$x_1$	$x_2$	$f(x_1)$	$f(x_2)$	$\frac{f(x_2)-f(x_1)}{x_2-x_1}$	$x^*$	$f(x^*)$	$\tilde{f}(x^*)$	$\tilde{f}(x^*) - f(x^*)$	$f(x^*) \times NWL$
0.1	0.7	23.97	3.7	-33.79	0.26	8.18	18.67	10.50	1.64
0.1	0.4	23.97	5.30	-62.23	0.20	10.94	17.94	7.01	2.19
0.1	0.25	23.97	8.42	-103.71	0.16	14.15	18.06	3.91	2.83
<b>0.1</b>	0.175	23.97	12.50	-153.03	0.13	17.31	19.08	1.78	3.46
0.175	0.7	12.50	3.70	-16.75	0.34	6.08	9.65	3.58	1.22
0.175	0.438	12.50	4.92	-28.86	0.27	7.62	9.62	2.00	1.52
<b>0.175</b>	0.306	12.50	6.82	-43.26	0.23	9.17	10.08	0.91	1.83
<b>0.306</b>	0.7	6.82	3.70	-7.92	0.46	4.73	5.60	0.87	0.95
<b>0.7</b>	0.7								

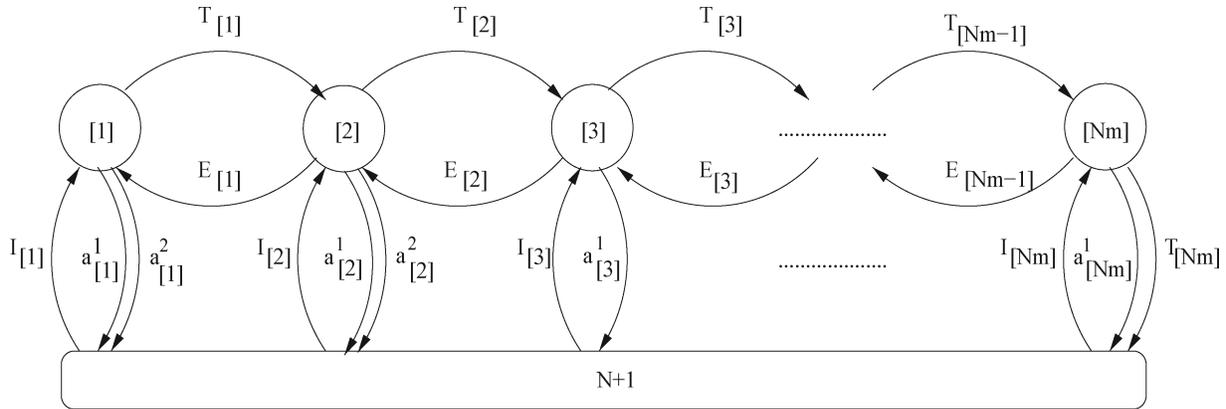


Fig. 3. Network graph.

Constraints (3b), (3c) and (3d) are used to calculate the earliness/tardiness, amount of compression in processing time of each part and idle time before each part. Constraint (3e) puts an upper bound on the amount of compression in the processing time. The upper bounds are based on the range of linear line segments in the approximate function. One can see a pictorial representation of the model in Fig. 3. Each constraint is represented by a node. For constraint (3c), the variables with coefficient +1 ( $T_{[i]m}$ ,  $E_{[i-1]m}$ ,  $a_{[i]m}^g$ ) are represented as outgoing arcs from the corresponding node  $i$ . The variables with coefficient -1 ( $E_{[i]m}$ ,  $T_{[i-1]m}$ ,  $I_{[i]m}$ ) are incoming arcs. Constraint (3d) is a dummy constraint, which is added to have a feasible solution for the minimum-cost network flow model. Constraint (3e) puts an upper bound on  $a_{[i]m}^g$ , which is an upper bound on the flow for the corresponding arc. The MCNF<sub>m</sub> model can be solved efficiently by minimum-cost network flow algorithms. More detailed information on minimum-cost network flow algorithms can be found in Ahuja *et al.* (1993). The proposed model finds the optimal starting times and the processing times of the parts for a fixed sequence of parts on a single machine. An important advantage of using the MCNF model is the fact that we are no longer restricted to considering only discrete processing time alternatives. The processing times can take any value between the lower and upper bounds.

4. Numerical example

A numerical example with two identical parallel-machines and ten parts will clarify the proposed algorithm. The due dates, earliness and tardiness penalties of the parts, the breakpoints of the processing times and the corresponding manufacturing costs are provided in Table 2.

The two-stage algorithm starts by solving the LP relaxation of the time-indexed IP model in the first stage. The LP relaxation gives a non-integral solution with the objective function value of 148.56. The LP relaxation solution

is  $Z_{1,1,3,4} = 0.6$ ,  $Z_{1,2,3,3} = 0.2$ ,  $Z_{1,2,25,3} = 0.2$ ,  $Z_{2,2,57,4} = 1$ ,  $Z_{3,2,41,5} = 1$ ,  $Z_{4,1,44,5} = 1$ ,  $Z_{5,1,36,4} = 0.8$ ,  $Z_{5,2,4,5} = 0.2$ ,  $Z_{6,1,79,5} = 1$ ,  $Z_{7,2,13,3} = 0.2$ ,  $Z_{7,2,17,3} = 0.2$ ,  $Z_{7,2,19,3} = 0.2$ ,  $Z_{7,2,21,3} = 0.2$ ,  $Z_{7,2,23,3} = 0.2$ ,  $Z_{8,2,1,4} = 0.6$ ,  $Z_{8,1,1,4} = 0.4$ ,  $Z_{9,1,1,2} = 0.6$ ,  $Z_{9,2,1,2} = 0.4$ ,  $Z_{10,1,31,3} = 0.6$ ,  $Z_{10,2,27,3} = 0.2$  and  $Z_{10,2,31,3} = 0.2$ . This infeasible solution is used to find an initial sequence on each machine. The maximum  $Z_{imtk}$  value is found for each part and the starting time  $t$  is used to find a sequence on each machine. The parts 1, 4, 5, 6, 9 and 10 are assigned to the first machine and parts 2, 3, 7 and 8 are assigned to the second machine. The parts are sequenced according to the time periods they are assigned. The sequence on the first machine is {9, 1, 10, 5, 4, 6} and the sequence on the second machine is {8, 7, 3, 2}. In the second stage, the MCNF<sub>m</sub> model is solved to find the optimal starting times and processing times for the given fixed sequences on each machine. The objective function value is found to be 174.6. A pictorial representation of the optimal solutions on machines 1 and 2 can be seen in Fig. 4.

Table 2. Data used in the numerical example

Part	$d$	$\epsilon$	$\tau$	Processing times	Machining costs
1	40	0.6	15	(4, 9, 16, 28, 40)	(255.9, 84.4, 35.8, 18.4, 13.7)
2	69	0.8	20	(1, 3, 6, 13)	(48.2, 10.6, 4.6, 2.9)
3	53	0.1	2.5	(1, 2, 4, 7, 13, 22, 30)	(351.2, 129.6, 48.2, 22.1, 10.3, 6.7, 6.2)
4	69	0.6	15	(2, 4, 8, 15, 26, 37)	(201.4, 76.3, 29.5, 13.6, 8.6, 7.7)
5	40	0.2	5	(1, 2, 3, 5, 9, 16, 22)	(197.8, 69.0, 37.4, 17.6, 8.0, 4.8, 4.4)
6	87	1	25	(1, 2, 4, 7, 9)	(23.3, 8.7, 3.6, 2.2, 2.0)
7	26	0.5	12.5	(1, 2, 4, 11)	(29.0, 11.1, 4.6, 2.3)
8	34	0.3	7.5	(9, 15, 25, 34)	(325.1, 150.6, 71.3, 46.9)
9	0	0.2	5	(1, 2)	(22.7, 7.9)
10	34	0.7	17.5	(1, 2, 4, 7, 10)	(26.5, 9.9, 4.1, 2.4, 2.2)

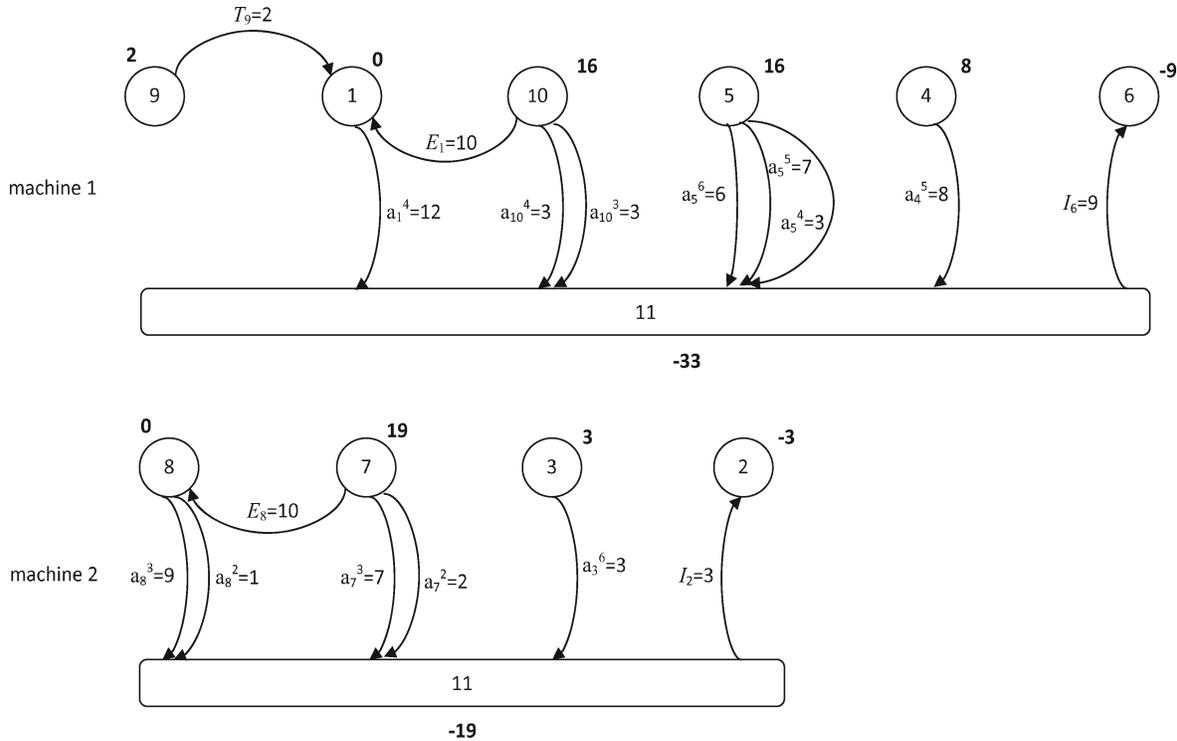


Fig. 4. Representations of the optimal solutions for MCNF<sub>1</sub> and MCNF<sub>2</sub>.

The arcs that have a positive flow in the optimal solution are shown in Fig. 4. All other arcs that have zero flow in the optimal solution are not shown. Part 9, which is the first part on machine 1, has a tardiness value of 2 ( $T_9 = 2$ ). Since  $a_{91}^1 = 0$ , the processing time of part 9 is two ( $p_{91} = p_{91}^u - a_{91}^1 = 2 - 0 = 2$ ). Since the idle time before part 9 is zero ( $I_9 = 0$ ), the start time of part 9 on machine 1 is also zero. Part 1, which is the second part on machine 1, has an earliness value of ten. The processing time of part 1 is  $p_{11} = p_{11}^u - a_{11}^1 - a_{11}^2 - a_{11}^3 - a_{11}^4 = 40 - 0 - 0 - 0 - 12 = 28$ . The start time of part 1 is  $S_{11} = T_{11} - E_{11} - (p_{11}^u - a_{11}^1 - a_{11}^2 - a_{11}^3 - a_{11}^4) + d_1 = 0 - 10 - (40 - 0 - 0 - 0 - 12) + 40 = 2$ . The start times are calculated by using  $S_{[i]m} = T_{[i]m} - E_{[i]m} - (p_{[i]m}^u - \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g) + d_{[i]}$  and the processing times are calculated by using  $p_{[i]m} = p_{[i]m}^u - \sum_{g=1}^{G_{[i]m}} a_{[i]m}^g$  for all parts on each machine. The Gantt chart of the solution found by the proposed algorithm can be seen in Fig. 5.

When we solve the time-indexed IP model of the numerical example optimally, we obtain a solution with an objective function value of 154. The Gantt chart of the solution found by solving the time-indexed IP model can be seen in Fig. 6.

### 5. Reactive scheduling

In reactive scheduling, different performance measures are often used to measure the quality of the new schedule formed after the disruption. The aim in reactive scheduling is to find a new schedule that is close to the initial schedule in terms of both stability and efficiency. The efficiency measure is often the same as that used in predictive scheduling and typically reflects schedule quality using objectives such as makespan, tardiness or total cost. The stability measures how much the new schedule is different from the initial schedule and reflects costs incurred due to changing

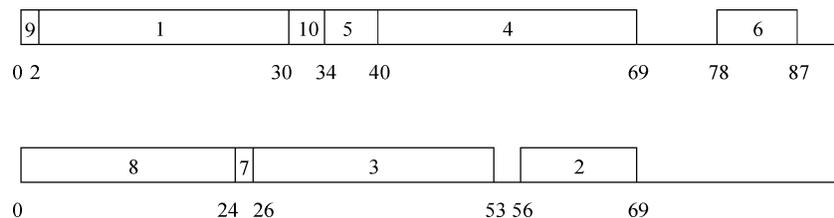


Fig. 5. Numerical example; two-stage algorithm.

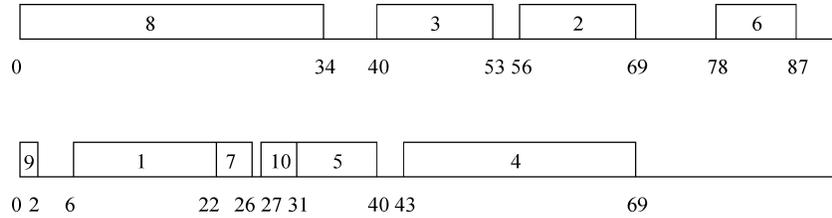


Fig. 6. Numerical example; IP.

plans based on the original schedule. Presumably the “correct” objective in reactive scheduling varies considerably from one manufacturing setting to another.

In this part of the study, the minimization of the sum of total weighted earliness, tardiness and manufacturing costs is used as the efficiency measure. It is the same objective used in determining the predictive schedule. In manufacturing systems, the initial schedule is used to determine the due dates of the parts in preceding workstations, the release times in succeeding workstations, delivery of raw material and transportation schedules. A disturbance changes the schedule in a workstation, and the change in the schedule affects the schedules of all preceding and succeeding stations. The aim is to find a new schedule which does not deviate too much from the initial schedule in terms of completion times. In this study, the absolute difference between the completion times of the parts in the initial and final schedules is used as the stability measure. As the stability measure improves, the new schedule gets closer to the initial schedule in terms of completion times. The stability is calculated as  $\sum_i |C_i - C'_i|$ , where  $C_i$  and  $C'_i$  are the completion times of part  $i$  in the initial and final schedules, respectively.

The time-indexed IP model can be revised to handle a variety of disruptions including machine breakdown, arrival of a new job, order cancellation, due date change, change in job priority and delay in the arrival or shortage of materials. The model is revised as follows in order to react to disturbances:

$$(IP') \quad \min \sum_{i \in D} \sum_{m=1}^M \sum_{k=1}^{K_{im}} \sum_{t=\max\{t_m^{\text{start}}, r_i\}}^{T-ps_{imk}+1} reactcost_{imtk} Z_{imtk}, \quad (4a)$$

subject to

$$\sum_{m=1}^M \sum_{k=1}^{K_{im}} \sum_{t=\max\{t_m^{\text{start}}, r_i\}}^{T-ps_{imk}+1} Z_{imtk} = 1, \quad \forall i \in D, \quad (4b)$$

$$\sum_{i \in D} \sum_{k=1}^{K_{im}} \sum_{u=\max\{t-ps_{imk}+1, t_m^{\text{start}}, r_i\}}^{\min\{T-ps_{imk}+1, t\}} Z_{imuk} \leq 1, \quad (4c)$$

$$\forall m \quad \text{and } t = t_m^{\text{start}}, \dots, T, \quad (4d)$$

$$Z_{imtk} \in \{0, 1\}, \quad \forall i, m, t, k.$$

The objective function is the sum of the efficiency measure used in the IP model (1a) (which is the sum of man-

ufacturing cost and total weighted earliness and tardiness) and stability measure (which is the deviation of completion times in the initial and final schedules). The new cost term is calculated as

$$reactcost_{imtk} = cost_{imtk} + |(t + ps_{imk} - 1) - C_i|.$$

The set of parts that are affected by the disruption is denoted by  $D$  and the revised IP' model is solved for these parts. If a machine breakdown occurs, the part that is being processed at the breakdown time and all other parts that are not processed yet are the parts affected by the disruption. When a new job arrives, an order is canceled, a change in the due dates or job priorities occurs, the parts that are not processed yet are also denoted as the affected jobs. The first constraint guarantees that every affected part is scheduled in the available time periods exactly once. The second constraint is the machine capacity constraint which states that each available time unit on a machine can be occupied by at most one part. The proposed model allows reassignment of parts to different machines.

The model handles different starting time periods for the machines ( $t_m^{\text{start}}$ ) and different release times for the parts ( $r_i$ ). When a machine breakdown occurs, the machine will be unavailable for a certain period of time. The starting time of the broken machine is found by adding the duration of breakdown to the breakdown time. The starting times of the other machines are the completion times of the last unaffected jobs in the sequences of the corresponding machines. The release times can be determined in different ways. For example, the starting times of the parts in the initial schedule could be considered as the release times in reactive scheduling to model the case where raw material is scheduled to arrive in just-in-time. Another method might assume the availability of all parts at the beginning of the scheduling period.

MCNF<sub>m</sub> must also be revised in order to cope with disruptions. When stability is incorporated into the model, the following model is achieved:

$$(MCNF'_m) \quad \min \sum_{i=1}^{N_m} \epsilon_{[i]} E_{[i]m} + \sum_{i=1}^{N_m} \tau_{[i]} T_{[i]m} + \sum_{i=1}^{N_m} \left( f_{[i]m}^{\min} - \sum_{g=1}^{G_{[i]m}} b_{[i]m}^g a_{[i]m}^g \right) + \sum_{i=1}^{N_m} D_{[i]}^- + \sum_{i=1}^{N_m} D_{[i]}^+, \quad (5a)$$



we can find good solutions for this complex problem and test the performance of the proposed two-stage algorithm as well as the time-indexed IP. Third, we want to show that the proposed model and algorithm can be used both in predictive and reactive scheduling environments. In the first part of the study we considered a predictive scheduling environment with the goal of finding both processing times and a schedule so as to minimize the sum of earliness, tardiness and manufacturing costs. In the second part of the study, we considered a reactive scheduling environment, and tested how the proposed methods perform in reaction to a disturbance. The proposed algorithm was coded in the C language and compiled with Gnu C compiler. The minimum-cost network flow model and the time-indexed IP model were solved using the callable library routines of CPLEX 7.1. The problems were solved on a 400 MHz UltraSPARC station.

For both computational studies, the same basic set of problems was used. We generated a variety of problems with 30, 50 and 100 parts and three, five and ten machines. For the three and five-machine problems, we considered different types of machines. For the problems with ten machines, we assumed that there were two identical machines each of the five different machine types. For a specific description and cost coefficients of these unrelated Computer Numerical Controlled (CNC) machines, we refer to Turkcan *et al.* (2007).

Due dates were randomly generated from the interval  $UN \sim [(1 - TF - RDD)/2, (1 - TF + RDD)/2] \times (\sum_i \bar{p}_i / M)$ , where  $TF$  is the tardiness factor,  $RDD$  is the range of due dates and  $\bar{p}_i = (p_i^l + p_i^u)/2$ . Both  $TF$  and  $RDD$  were set to 0.2, 0.5 and 0.8. The earliness weights were selected randomly from the interval  $UN \sim [0.1, 1.0]$ . The ratio of tardiness weight to earliness weight,  $\tau_i/\epsilon_i$ , was taken as 25 and 50 in the computational study reflecting the fact that tardiness is usually far more costly than earliness.

Since we have a well-defined convex cost function for turning operations using CNC machines, we used the same cost function in our computational study. We calculated the processing times of the parts using operation and tool-related parameters. With the specified values of these parameters, the average of the lower bounds of the processing times is 0.5 and the average of the upper bounds of the processing times is 5.1. Overall, the processing times vary between 0.01 and 42.2.

Two other parameters were employed in this study. The first one was the linearization constant ( $NWL$ ), which was used in the piecewise linearization of the non-linear convex manufacturing cost function. As the linearization constant decreases, the number of breakpoints in the approximate cost function increases, and the approximation becomes better. The second factor was the scale factor ( $SC$ ), which was used to find the unit time period. The unit time period, which was used to discretize the processing time settings in the IP formulation, was calculated as  $(\max_i p_i^u)/SC$ . As the scale factor increases, the unit time period decreases

and the processing times increase. Decreasing the unit time increases the accuracy of the proposed IP model, but at the expense of an increased computation time. Since there is a trade-off between the solution quality and computation time, we performed an initial computational study to determine the values of these two factors. We tried 0.05, 0.1 and 0.2 for the linearization constant, and 20, 30 and 40 for the scale factor. We solved a number of problems with these factor combinations. Based on the results of the initial runs, we found that the difference between different linearization constants was insignificant. As the scale factor increases, the solution quality increases. We therefore selected 0.2 for the linearization constant and 40 for the scale factor. With these values, the average number of breakpoints in the approximation to the manufacturing cost function was 3.8 for the time-indexed IP model and 5.4 for the proposed algorithm (the difference being accounted for by the discretization used in the IP model).

### 6.1. Predictive scheduling

For each combination of the levels of  $N$ ,  $M$ ,  $TF$ ,  $RDD$  and  $(\tau_i/\epsilon_i)$  ( $3 \times 3 \times 3 \times 3 \times 2$ ), we generated five replications resulting in 810 randomly generated problems. These problems were solved with both the time-indexed IP model and the proposed two-stage algorithm.

First, we report the percentage of total weighted earliness, tardiness and manufacturing cost in the total cost. The minimum percentage of total weighted earliness to the total cost is 0.13%. The maximum percentage is 33.41%. The percentage of total weighted tardiness changes between 0 and 96.41%. The percentage of manufacturing cost is between 3.44 and 94.38%. These results show that none of the objective function terms dominate in the total objective function.

In most studies in the literature, the processing times are assumed to be fixed at and equal to the most economical processing times in terms of machining and tooling costs. As pointed out previously, this might not be the best alternative for scheduling-related criteria. Controllable processing times provide flexibility in finding solutions with better overall objective function values. To demonstrate the value of controllable processing times, we first solved the 810 test problems with the proposed time-indexed IP model using both fixed processing times based on the upper ( $p^u$ ) and lower ( $p^l$ ) bound values and the controllable processing times denoted as ( $p$ ). We used the percentage improvement of the objective function found by using controllable processing times over fixed processing times for comparison. In Table 3, the minimum, average and maximum percentage improvements can be seen. The problems for which the optimal IP solution cannot be obtained by CPLEX are not included in these results. For (30, 3) problems, the average percentage improvement achieved by using controllable processing times instead of using  $p^u$

**Table 3.** Effect of the controllable processing times

$(N, M)$	$(p^u - p)/p^u$				$(p^l - p)/p^l$			
	Minimum	Average	Maximum	Number	Minimum	Average	Maximum	Number
(30, 3)	0.04	63.5	97.5	84	9.0	62.6	93.4	90
(30, 5)	0.0	56.2	96.9	88	9.8	64.1	93.9	90
(30, 10)	2.9	38.2	90.2	90	17.0	68.0	94.4	90
(50, 3)	4.2	50.0	97.2	43	4.5	57.8	94.4	89
(50, 5)	2.3	51.5	98.4	57	7.3	62.7	94.9	87
(50, 10)	2.9	45.3	97.0	76	11.8	69.1	95.9	88
(100, 3)				0	2.3	42.0	89.8	68
(100, 5)	43.3	43.3	43.3	1	3.9	43.2	91.4	63
(100, 10)	5.1	30.0	72.3	3	7.7	38.5	84.3	39

is 63.5%. The average is taken over the 84 runs out of 90 that were solved optimally by CPLEX. The improvement is 62.6% when controllable processing time is used instead of  $p^l$ .

In order to show that the linear relaxation of the time-indexed formulation gives a tight lower bound, we compare the LP relaxation solution and integer solution. The minimum, average and maximum percentage gap between the linear relaxation solution (LP) and IP solution can be seen in Table 4. The left half of the table shows the results only for those problems that the IP model could solve to optimality. For this half of the table, the percentage gap is between the optimal solution (IP) and the linear relaxation solution (LP). In the right half of the table, results are presented for problems the IP model cannot solve to optimality (within a 1800 second time limit), but for which a feasible solution was found. Thus, for these problems, the best feasible solution (*FEAS*) is used for comparisons. We also note that for some problems, CPLEX could not find a feasible solution. For example, 39 problems were solved optimally for the 100 job, 10-machine problems and the average percentage gap is 0.05%. A feasible solution was found within the given time limit for 22 problems and the percentage gap is 0.55%. CPLEX was unable to find a feasible solution to the IP model within the time limit for 29

problems. We can see from the results that the percentage gaps are very small. This shows that the solution obtained by solving the LP relaxation can be a good starting point to find feasible solutions.

In order to compare the performance of the proposed algorithm with IP, we look at the percentage gap of the solutions over the best solution. The best solution can be found by either the proposed algorithm or IP, since the proposed MCNF model is no longer restricted to consider only discrete processing time alternatives as opposed to the IP model. Table 5 shows the minimum, average and maximum percentage gap of solutions over the best solution and number of best solutions found in 90 runs for each job, machine  $(N, M)$  pair. For (30, 3) problems, the average gap for the algorithm is 6.27% and it is 1.64% for IP. The proposed algorithm finds the best solution in 42 problems among the 90 problems. IP finds the best solution in 48 problems. Even though the number of problems with the best solution are close to each other for both methods, the percentage gaps are smaller for IP. The variability in solution quality for the proposed algorithm is higher.

Table 6 shows the minimum, average and maximum computational times in seconds for the proposed algorithm and IP. The average computation times for the proposed algorithm are significantly better than the ones for IP. Most of

**Table 4.** Percentage gap between integer solution and LP relaxation solution of the time-indexed IP model

$(N, M)$	$\frac{IP-LP}{LP}$				$\frac{FEAS-LP}{LP}$			
	Minimum	Average	Maximum	Problems	Minimum	Average	Maximum	Problems
(30, 3)	0	0.22	1.57	90				
(30, 5)	0	0.14	0.75	90				
(30, 10)	0	0.06	0.24	90				
(50, 3)	0	0.22	1.89	89	0.25	0.25	0.25	1
(50, 5)	0	0.17	0.67	87	0.12	0.36	0.84	3
(50, 10)	0	0.12	0.69	88	0.06	0.15	0.23	2
(100, 3)	0.002	0.08	0.43	68	0.10	0.72	1.58	22
(100, 5)	0.003	0.07	0.31	63	0.05	2.08	9.62	27
(100, 10)*	0.003	0.05	0.21	39	0.02	0.55	4.93	22

\*29 problems unsolved by IP.

**Table 5.** Percentage gap of the solutions for the proposed algorithm and time-indexed IP model over the best solution and number of best solutions in 90 runs for each  $(N, M)$  pair

$(N, M)$	ALG				IP			
	Minimum	Average	Maximum	Number of best solutions	Minimum	Average	Maximum	Number of best solutions
(30, 3)	0	6.27	58.6	42	0	1.64	18.03	48
(30, 5)	0	13.82	125.88	32	0	0.49	4.96	58
(30, 10)	0	9.34	92.84	49	0	0.63	9.83	42
(50, 3)	0	4.64	55.53	52	0	2.79	37.78	38
(50, 5)	0	8.2	74.47	37	0	1.45	16	53
(50, 10)	0	11.48	65.04	32	0	1.31	25.1	58
(100, 3)	0	1.6	16.51	59	0	3.42	24.79	31
(100, 5)	0	3.94	27.4	47	0	1.86	18.6	43
(100, 10)*	0	2.23	25.46	58	0	1.18	9.02	32

\*Two problems cannot be solved by LP relaxation and 29 problems cannot be solved by IP in the 1800-second time limit.

the CPU time spent by the algorithm is for the solution of the LP relaxation.

## 6.2. Reactive scheduling

Our reactive scheduling computational study considered machine breakdown as the disruption on the shop floor. The breakdown could occur at any one of the machines at any time on the planning horizon,  $T$ . The machine breakdown time was selected randomly from the interval  $UN \sim [1, 0.3T]$ . We deliberately selected the breakdown time from the earlier periods so that a higher number of jobs needed to be rescheduled and hence the rescheduling problems were not trivial. Furthermore, the breakdown duration may be either short or long. The duration was selected randomly from the distribution  $UN \sim [0.1T, 0.3T]$  for the short-duration case and from the distribution  $UN \sim [0.5T, 0.7T]$  for the long duration case. Before the breakdown time, the initial schedule was followed, and the breakdown time was not known in advance. When the breakdown occurs, we assumed that we could determine the duration of the down time immediately. The 781 problems considered and solved in the previous section were solved for the short and long breakdown duration cases. The schedule found

by solving the IP model (in the predictive scheduling experiments) was used as the initial schedule. The 29 problems where the IP model could not find a feasible solution in 1800 seconds were not solved in this part of the study.

We considered three rescheduling methods for the purposes of comparison. The first one was a static pushback strategy, also called the Right-Shift (RS) algorithm, which is the most commonly cited method in the literature. In this method, when a machine breakdown occurs, the job sequences on each machine are kept the same, only the starting times on the failed machine are shifted to the right (in a Gantt chart representation) as far as necessary to accommodate the disruption. The second method (RS-MCNF) was a partial rescheduling of the jobs on the failed machine. The initial sequence of the jobs was not changed and the MCNF' model was solved to determine the starting and processing times of the parts. Thus the jobs on the failed machine were right-shifted, then the MCNF' model was used to recalculate the processing times to minimize cost. The third method entailed rescheduling all remaining jobs after the breakdown using the methods proposed in this paper. We tested both the proposed LP-relaxation-based two-stage algorithm (ALG) and the time-indexed IP model (IP') for this purpose.

**Table 6.** Average CPU times in seconds for the proposed algorithm and time-indexed IP model

$(N, M)$	ALG			IP		
	Minimum	Average	Maximum	Minimum	Average	Maximum
(30, 3)	2	11	42	5	43	399
(30, 5)	3	17	48	5	59	631
(30, 10)	4	23	64	10	46	305
(50, 3)	4	35	153	9	155	1811
(50, 5)	6	48	154	11	240	1832
(50, 10)	13	68	238	21	239	1852
(100, 3)	95	422	1048	95	795	1922
(100, 5)	113	595	2026	115	1006	1973
(100, 10)*	175	776	2021	278	1436	1991

\*Two problems cannot be solved by LP relaxation and 29 problems cannot be solved by IP in the 1800-second time limit.

**Table 7.** Percentage improvements over the RS method in terms of efficiency and stability

<i>(N, M)</i>	<i>RS-MCNF</i>				<i>ALG</i>				<i>IP'</i>			
	<i>Short</i>		<i>Long</i>		<i>Short</i>		<i>Long</i>		<i>Short</i>		<i>Long</i>	
	<i>Efficiency</i>	<i>Stability</i>										
(30, 3)	50	75	41	52	62	93	77	96	63	94	77	96
(30, 5)	41	73	32	50	51	92	67	96	53	93	68	97
(30, 10)	27	70	23	43	34	92	50	96	36	97	52	98
(50, 3)	54	78	46	56	65	84	81	94	65	85	81	93
(50, 5)	48	76	36	49	60	85	76	94	61	90	77	95
(50, 10)	31	64	20	35	42	81	61	93	45	91	62	97
(100, 3)	57	83	49	58	65	91	82	96	66	92	82	96
(100, 5)	52	83	46	60	58	83	75	95	59	84	75	95
(100, 10)	35	80	32	53	39	70	57	92	41	88	58	95

We considered the sum of earliness, tardiness and manufacturing costs as the efficiency measure, and the absolute difference between the completion times in the initial and final schedules as the stability measure. In order to show the performance of the proposed rescheduling methods, RS-MCNF, ALG and IP', over the RS method, the percentage improvements in terms of both the efficiency and stability are reported. The percentage improvement of a method over the RS method in terms of efficiency or stability is calculated as [efficiency (stability) of RS – efficiency (stability) of the other method]/efficiency (stability) of RS]. In Table 7, the average percentage improvements are shown. Each value in the table is the average of 90 values for all (N, M) values except the (100, 10) problems for the IP' method. The values for (100, 10) problems are the average of 61 values for the IP' method since 29 problems could not be solved with IP' in the given time limit.

According to the average percentage improvements in efficiency, “partial rescheduling” (RS-MCNF) gives significantly better objective function values than the RS method. When the MCNF model is solved for the fixed sequence of parts, the sum of earliness and tardiness decreases at the expense of increased manufacturing costs. Consequently, a better objective function value is found, which reflects what one might expect in practice. For example, when the breakdown duration is short, RS-MCNF yields a 50% improvement in the average cost for the (30, 3) problems. This clearly shows that we can obtain solutions better than the RS method by changing only processing times (and without changing the sequence of jobs or rerouting jobs to other machines). The average percentage improvement for the (30, 3) problems increases to 62% when the proposed algorithm is used and to 63% when the time-indexed IP model is used. The rescheduling of all jobs with the proposed algorithm and the time-indexed IP model gives the best efficiency values for all problem types. The percentage improvement in efficiency over all problems is 61.9% for the proposed algorithm and 63% for the time-indexed IP' model. The percentage of idle time in the initial schedule should affect the

performance of the algorithms. For a fixed tardiness factor, the idle time percentage increases as the range of due dates increases, and the percentage improvement of the algorithms over the RS method decreases. When there is enough idle time in the initial schedule, a simple algorithm such as the RS method does not increase the objective function value too much. When the tardiness factor increases, the idle time percentage decreases. However, as the number of tardy jobs increases due to tighter due dates, the processing times of the parts are already compressed in the initial schedule in order to decrease the total weighted tardiness value. In such a low idle time case, most of the processing times cannot be (inexpensively) decreased any further and the percentage improvement in efficiency decreases.

The stability measure shows the difference between the initial and final schedules in terms of completion times. As the stability measure decreases, the final schedule approaches the initial schedule. The experimental results indicate that all algorithms give better stability measures than the RS method. The proposed algorithm and the time-indexed IP' model give the best stability measures. There is no significant difference between the proposed algorithm and the IP' model.

The average computation times, which are very important in reactive scheduling, can be seen in Table 8 for all algorithms. The average CPU times of the RS and RS-MCNF algorithms are very small. The rescheduling of all affected jobs requires more computation time. The time-indexed formulation, which is solved by CPLEX, requires the most CPU time.

The computational results demonstrate that the proposed algorithm and the MCNF model can be used efficiently to solve reactive scheduling problems with controllable processing times. The controllable processing times provide flexibility in finding solutions with better efficiency and stability measures. The proposed algorithms can solve these problems in very reasonable computation times, which is an important issue in reactive scheduling.

**Table 8.** Average CPU times (in seconds)

$(N, M)$	RS		RS-MCNF		ALG		IP'	
	Short	Long	Short	Long	Short	Long	Short	Long
(30, 3)	0.014	0.015	0.012	0.013	3	4	27	15
(30, 5)	0.014	0.016	0.011	0.014	3	5	9	13
(30, 10)	0.016	0.016	0.012	0.014	4	6	8	12
(50, 3)	0.019	0.020	0.020	0.022	10	12	30	48
(50, 5)	0.019	0.021	0.018	0.021	12	16	84	59
(50, 10)	0.024	0.026	0.022	0.024	16	20	41	62
(100, 3)	0.048	0.058	0.049	0.063	147	172	439	557
(100, 5)	0.052	0.065	0.049	0.064	167	195	551	565
(100, 10)	0.059	0.073	0.050	0.070	140	184	469	661

## 7. Conclusions

In this study, we consider the problem of scheduling with controllable processing times and earliness–tardiness penalties. The aim is to determine the processing times and schedule for jobs on unrelated parallel CNC machines. We consider the objective of minimizing the manufacturing cost, and the total weighted earliness and tardiness. To the best of our knowledge, the parallel-machine scheduling problem that considers job-dependent earliness and tardiness penalties, idle times, distinct due dates and controllable processing times simultaneously, is not considered in the literature.

We propose a time-indexed IP model with discrete processing time alternatives for jobs. The model allocates the jobs to machines, and determines starting times and processing times. The time-indexed formulation requires discrete approximations, and has the disadvantage of size. A LP-relaxation-based algorithm is proposed to solve the problem in reasonable computation times. First, the linear relaxation of the time-indexed model is solved and the solution of the relaxation with partial machine, time period and processing time assignments is used to find a sequence for parts on each machine. Then, a non-linear programming model is proposed to solve the optimal idle time insertion problem (timetabling problem) for determining the start, finish and processing times of parts on each machine. The non-linear model is converted to a minimum-cost network flow model by piecewise linearization of the convex manufacturing cost. This allows us to solve the timetabling problem with controllable processing times quite efficiently.

The models and algorithms presented in this paper can have significant impact in reactive scheduling. First, our model considers both earliness and tardiness, an important consideration in reactive scheduling. The (predictive) master schedule serves as the basis for planning many other activities including the arrival of raw materials and parts from upstream processing on one end and the scheduling of downstream processing on the other. Deviations (earliness or tardiness) from the original schedule can be expected to increase cost. The second reason we believe our approach

is important is that it reflects the current practice of plant and line managers. A common approach to disruptions is in fact to reduce processing times through additional resources or changes in machining conditions. Thus, our approach formalizes existing practice. The computational results also show the “good” performance of the proposed rescheduling methods in terms of both the efficiency and stability.

## Acknowledgements

The first author is supported in part by the Scientific and Technical Research Council of Turkey under NATO-A2 research scholarship program. The authors would like to thank the Associate Editor and two anonymous referees whose constructive comments have been used to improve this paper.

## References

- Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993) *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, NJ.
- Alidaee, B. and Ahmadian, A. (1993) Two parallel-machine sequencing problems involving controllable processing times. *European Journal of Operational Research*, **70**, 335–341.
- Avci, S. (2001) Algorithms for generalized nonregular scheduling problems. Ph.D. thesis, Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA.
- Baker, K.R. and Scudder, G.D. (1990) Sequencing with earliness and tardiness penalties: a review. *Operations Research*, **38**, 22–36.
- Cattrysse, D.G. and van Wassenhove, L.N. (1992) A survey of algorithms for the generalized assignment problem. *European Journal of Operational Research*, **60**, 260–272.
- Chen, J.-Y. and Lin, S.-F. (2002) Minimizing weighted earliness and tardiness penalties in single-machine scheduling with idle time permitted. *Naval Research Logistics*, **49**, 760–780.
- Cheng, T.C.E., Chen, Z.-L. and Li, C.-L. (1996) Parallel-machine scheduling with controllable processing times. *IIE Transactions*, **28**, 177–180.
- Fry, T.D., Armstrong, R.D. and Blackstone, J.H. (1984) Minimizing weighted absolute deviation in single machine scheduling. *IIE Transactions*, **19**, 445–450.

- Gurel, S. and Akturk, M.S. (2007) Considering manufacturing cost and scheduling performance on a CNC turning machine. *European Journal of Operational Research*, **177**, 325–343.
- Hoogeveen, J.A. and Van de Velde, S.L. (1996) A branch-and-bound algorithm for single-machine earliness-tardiness scheduling with idle time. *INFORMS Journal on Computing*, **8**, 402–412.
- Kanet, J.J. and Sridharan, S.V. (2000) Scheduling with inserted idle time: problem taxonomy and literature review. *Operations Research*, **48**, 99–110.
- Kedad-Sidhoum, S., Solis, Y.R. and Sourd, F. (2008) Lower bounds for the earliness-tardiness scheduling problem on parallel-machines with distinct due dates. *European Journal of Operational Research*, **189**, 1305–1316.
- Liman, S.D., Panwalkar, S.S. and Thongmee, S. (1997) A single machine scheduling problem with common due window and controllable processing times. *Annals of Operations Research*, **70**, 145–154.
- Panwalkar, S.S. and Rajagopalan, R. (1992) Single machine sequencing with controllable processing times. *European Journal of Operational Research*, **59**, 298–301.
- Shabtay, D. and Kaspi, M. (2004) Minimizing the total weighted flow time in a single machine with controllable processing times. *Computers & Operations Research*, **31**, 2279–2289.
- Shabtay, D. and Kaspi, M. (2006) Parallel-machine scheduling with a convex resource consumption function. *European Journal of Operational Research*, **173**, 92–107.
- Shabtay, D., Kaspi, M. and Steiner, G. (2007) The no-wait two-machine flow-shop scheduling problem with convex resource-dependent processing times. *IIE Transactions*, **39**, 539–557.
- Shabtay, D. and Steiner, G. (2007) A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, **155**, 1643–1666.
- Sourd, F. (2005) Earliness-tardiness scheduling with setup considerations. *Computers & Operations Research*, **32**, 1849–1865.
- Turkcan, A., Akturk, M.S. and Storer, R.H. (2003) Non-identical parallel CNC machine scheduling. *International Journal of Production Research*, **41**(10), 2143–2168.
- Turkcan, A., Akturk, M.S. and Storer, R.H. (2007) Due date and cost-based FMS loading, scheduling and tool management. *International Journal of Production Research*, **45**(5), 1183–1213.
- Van den Akker, J.M., Hurkens, C.A.J. and Savelsbergh, M.W.P. (2000) Time-indexed formulations for single machine scheduling problems: column generation. *INFORMS Journal on Computing*, **12**(2), 111–124.
- Vieira, G.E., Herrmann, J.W. and Lin, E. (2003) Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of Scheduling*, **6**(1), 39–62.
- Yano, C.A. and Kim, Y. (1991) Algorithms for a class of single machine weighted tardiness and earliness problems. *European Journal of Operational Research*, **52**, 161–178.

## Biographies

Ayten Turkcan is an Associate Research Scientist at the Regenstrief Center for Healthcare Engineering at Purdue University. She received her B.S., M.S. and Ph.D. degrees in Industrial Engineering from Bilkent University, Turkey. Her research interests lie in scheduling in production, transportation and health care systems.

M. Selim Akturk is a Professor of Industrial Engineering at Bilkent University, Turkey. He holds a Ph.D. in Industrial Engineering from Lehigh University, USA and B.S.I.E. and M.S.I.E. degrees from the Middle East Technical University, Turkey. His current research interests include production scheduling, cellular manufacturing systems and advanced manufacturing technologies. He is a senior member of IIE and member of INFORMS.

Robert H. Storer is a Professor of Industrial and Systems Engineering and Co-Director of the Integrated Business and Engineering Honors Program at Lehigh University. He received his B.S. in Industrial and Operations Engineering from the University of Michigan in 1979, and M.S. and Ph.D. degrees in Industrial and Systems Engineering from the Georgia Institute of Technology in 1982 and 1987 respectively. His interests lie in operations research and applied statistics with particular interest in heuristic optimization, scheduling and logistics.