ELSEVIER
Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 167 (2005) 624–643
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

A new bounding mechanism for the CNC machine scheduling problems with controllable processing times

Rabia K. Kayan, M. Selim Akturk *

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

Available online 12 September 2004

Abstract

In this study, we determine the upper and lower bounds for the processing time of each job under controllable machining conditions. The proposed bounding scheme is used to find a set of discrete efficient points on the efficient frontier for a bi-criteria scheduling problem on a single CNC machine. We have two objectives; minimizing the manufacturing cost (comprised of machining and tooling costs) and minimizing makespan. The technological restrictions of the CNC machine along with the job specific parameters affect the machining conditions; such as cutting speed and feed rate, which in turn specify the processing times and tool lives. Since it is well known that scheduling problems are extremely sensitive to processing time data, system resources can be utilized much more efficiently by selecting processing times appropriately.

Keywords: Scheduling; Automated manufacturing; Bi-criteria; Controllable processing times; Makespan

1. Introduction

Most of the existing scheduling algorithms assume that the processing times are fixed and known. When we analyze the single machine total tardiness problem, $1\|\sum T_j$, as an example, there are two important parameters: the processing time vector, $\bar{p}$, and the due date vector, $\bar{d}$. In the literature, $\bar{p}$ is treated as a hard constraint, i.e. fixed and not allowed to change. On the other hand, $\bar{d}$ is considered as a soft constraint that means we are allowed to deviate from the desired due dates but a certain cost penalty is incurred for these

* Corresponding author. Tel.: +90 312 290 1360; fax: +90 312 266 4054.
E-mail address: akturk@bilkent.edu.tr (M.S. Akturk).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2004.07.012

deviations. In this study, one of the most important objectives is to show that the processing times can be treated as a soft constraint as well and their cost impact can be measured in terms of the corresponding machining and tooling costs. This problem becomes more evident for the CNC machines for which the machining conditions, i.e. cutting speed and feed rate, are controllable variables.

In the current literature, the process planning and scheduling levels are linked through timing data. After calculating locally optimal process parameters, i.e. machining conditions, the processing time is then passed to the scheduling level as data. In reality however, the time it takes to process each part is a controllable variable. Since it is well known that scheduling problems are extremely sensitive to processing time data, it seems that by selecting processing times appropriately, system resources can be utilized much more efficiently.

The cutting speed and feed rate are the machining parameters which constitute the machine settings and we can increase or decrease the processing time of a job by changing them. An increase in one of the machine settings will decrease the processing time but this will also decrease the life of the cutting tool because the job in process will use the tool more. Consequently, we incur an additional tooling cost, to which a manufacturer should always pay attention to use CNC machines effectively. In case of lower cutting speed, i.e. higher processing time, the completion times of jobs increase leading to increases in regular scheduling objectives such as minimizing tardiness, makespan or total completion times.

The optimization of the machining conditions for a single operation is a well known problem, where the decision variables are usually the cutting speed and the feed rate. These conditions are the key to economical machining operations. Knowledge of optimal cutting parameters for machining operations is required for process planning of metal cutting operations. Numerous models have been developed with the objective of determining optimal machining conditions. Malakooti and Deviprasad [7] formulate a metal cutting operation, specifically for a turning operation, as a discrete multiple objective problem. The objectives are to minimize cost per part, production time per part, and roughness of the work surface, simultaneously. Akturk and Avci [1] propose a solution procedure to make tool allocation and machining conditions selection decisions simultaneously. Akturk and Onen [2] develop a new algorithm to solve joint lot sizing, tool allocation and machining conditions optimizations problems to minimize total production cost. They show that it is possible to improve the overall solution by exploiting the interactions among these problems. Since machining conditions directly determine the processing time and tool usage rate of an operation, it is very essential to integrate process planning and scheduling decisions as well.

Processing time control and its impact on sequencing decisions and operational performance have received limited attention in the scheduling literature. A survey of the literature up to 1990 can be found in Nowicki and Zdrzalka [8]. Panwalkar and Rajagopalan [10] consider the static single machine sequencing problem with a common due date for all jobs in which job processing times are controllable with linear costs. They develop a method to find optimal processing times and an optimal sequence to minimize a cost function. Trick [12] focuses on assigning single-operation jobs to variable-speed machines while simultaneously controlling the processing speed of each machine. Zdrzalka [14] deals with the problem of scheduling jobs on a single machine in which each job has a release date, a delivery time and a controllable processing time, having its own associated linearly varying cost and propose an approximation algorithm for minimizing the overall schedule cost. Nowicki and Zdrzalka [9] present a bi-criterion approach of minimizing completion time and processing cost to preemptive scheduling of parallel machines with jobs having processing costs which are linear functions of variable processing times. Cheng et al. [3] consider a parallel machine scheduling problem with controllable processing times, where the job processing times can be compressed through incurring an additional cost, which is a convex function of the amount of compression. Daniels et al. [4] investigate the improvements in manufacturing performance that can be realized by broadening the scope of the production scheduling function to include both job sequencing and processing time control through the deployment of a flexible resource. Karabati and Kouvelis [6] solve the simultaneous scheduling and optimal processing times selection problem in a flow line operated under a cyclic scheduling policy. Vickson presents simple methods for solving two single machine sequencing problems when job processing times are themselves decision variables having their own linearly varying costs [13].

In the literature of scheduling with controllable processing times, most of the studies assume that the processing times can be crashed in a range with a linear compression cost.

It seems that there are two reasons for not solving the process planning and scheduling problems simultaneously. First of all, assigning arbitrary ranges to the processing times and assuming a linear compression cost may not be a realistic assumption. Furthermore, the scheduling problems are generally classified as NP-hard problems, and adding a nonlinear objective function and nonlinear constraints coming from the process planning problem will make these problems even more difficult to solve in practice. In this paper, we accomplished two things to alleviate some of these problems. We derived closed form expressions to determine exact upper and lower bounds for the processing times by considering CNC machine, cutting tool and machining operation specific parameters. Consequently, the nonlinear machining related constraints can be replaced with a simple linear bound. Moreover, we also developed an efficient frontier to establish a time/cost tradeoff for each manufacturing operation to link process planning and scheduling problems. By utilizing our results, someone could develop methods for building production schedules which include process planning level decisions as well as traditional scheduling decisions as will be demonstrated in Section 6.

2. Problem definition

There are N jobs, and each job corresponds to a metal cutting operation which can be performed by a different cutting tool. Our objective is to determine upper and lower bounds for the processing time of each job i under the bi-criteria objective of minimizing the manufacturing cost (comprised of machining and tooling costs) and minimizing any regular scheduling measure such as makespan, total completion time, etc. Let Y_i be the completion time of job i (the time at which the processing of job i is finished). A performance measure Z is regular if the scheduling objective is to minimize Z, and Z can increase only if at least one of the completion times in the schedule increases. Regular performance measures are functions that are nondecreasing in $Y_1, Y_2 \ldots, Y_n$. Suppose that $Z = f(Y_1, Y_2, \ldots, Y_n)$ is the value of the measure that characterizes schedule $\mathscr{S}$ and that $Z' = f(Y'_1, Y'_2, \ldots, Y'_n)$ represents the value of the same measure under some different schedule $\mathscr{S}'$. Then Z is regular as long as the following condition holds: $Z' > Z$ implies that $C'_i > C_i$ for some job i.

The notation used throughout the paper is as follows:

Parameters

α_i, β_i, γ_i speed, feed, depth of cut exponents for the tool
M, b, c, e specific coefficient and exponents of the machine power constraint
R, g, h, l specific coefficient and exponents of the surface roughness constraint
C_i Taylor's tool life expression parameter for the tool used for job i
O operating cost of the CNC machine (\$/minute)
T_i cost of the tool used for job i (\$)
d_i depth of cut for job i (in.)
D_i diameter of the generated surface for job i (in.)
L_i length of the generated surface for job i (in.)
H maximum available machine power (hp)
S_i maximum allowable surface roughness for job i (μin.)
N number of the jobs

Decision variables

v_i cutting speed of job i (fpm)
f_i feed rate of job i (ipr)
U_i tool usage rate of job i
P_i processing time of job i (min)

Let Z_1 be machining and tooling cost of all jobs, and equal to

$$Z_1 = \sum_{i=1}^{N}(O \cdot P_i + T_i \cdot U_i), \tag{1}$$

where machining and tooling costs are summed respectively. Tool usage rate of a job, U_i, is simply the ratio of processing time to the tool life. Each job has different usage rates depending on its depth of cut, diameter, length and surface finish requirements. The cutting tool becomes worn when the aggregation of usage rates of jobs operated by this tool exceeds one, in other words when the total processing time of the jobs exceeds tool life as discussed in Akturk and Avci [1]. The processing time and usage rate of job i for a turning operation are calculated as follows:

$$P_i = \frac{\pi \cdot D_i \cdot L_i}{12 \cdot v_i \cdot f_i}, \tag{2}$$

$$U_i = \frac{\pi \cdot D_i \cdot L_i \cdot d_i^{\gamma_i}}{12 \cdot C_i \cdot v_i^{(1-\alpha_i)} \cdot f_i^{(1-\beta_i)}}. \tag{3}$$

We can control the processing times and usage rates of jobs by changing the cutting speed and feed rate of the machine. While changing the machining conditions, we have to consider process planning constraints such as machine power, surface roughness and tool life as discussed in Akturk and Avci [1]. A mathematical model (NLP) of the overall scheduling problem is given below:

$$\text{Minimize} \quad Z_1 = \sum_{i=1}^{N}(O \cdot P_i + T_i \cdot U_i),$$

$$Z_1 = \sum_{i=1}^{N}\left(\frac{\pi D_i L_i O}{12} v_i^{-1} f_i^{-1} + \frac{\pi D_i L_i d_i^{\gamma_i} T_i}{12 C_i} v_i^{(\alpha_i-1)} f_i^{(\beta_i-1)}\right),$$

$$\text{Minimize} \quad Z_2 : \text{Any regular scheduling measure},$$

$$\text{Subject to} \quad T_i' v_i^{(\alpha_i-1)} f_i^{(\beta_i-1)} \leqslant 1, \quad i = 1, \ldots, N \quad (\text{TL}), \tag{4}$$

$$M_i' v_i^b f_i^c \leqslant 1, \quad i = 1 \ldots, N \quad (\text{MP}), \tag{5}$$

$$R_i' v_i^g f_i^h \leqslant 1, \quad i = 1, \ldots, N \quad (\text{SR}), \tag{6}$$

$$s_i + P_i \leqslant s_j \vee s_j + P_j \leqslant s_i, \quad i, j = 1, \ldots, N \wedge i \neq j \quad (\text{Non-interference}). \tag{7}$$

Other constraints of scheduling

$$v_i, f_i > 0, \quad s_i \geqslant 0, \qquad i = 1, \ldots, N, \tag{8}$$

where

$$T_i' = \frac{\pi D_i L_i d_i^{\gamma_i}}{12 C_i}, \quad M_i' = \frac{M d_i^e}{H}, \quad \text{and } R_i' = \frac{R d_i^f}{S_i}.$$

The first term in the objective function Z_1 is the machining cost and the second one is the tooling cost. The first three set of constraints are tool life (TL), machine power (MP) and surface roughness (SR) constraints, respectively. These nonlinear constraints ensure that while solving the machining conditions optimization problem we do not exceed the available machine power and cutting tool life, and satisfy the necessary quality requirements for each part. The noninterference constraints are included to prevent scheduling two different jobs at the same time on the CNC machine by using a set of disjunctive constraints where s_i stands for starting time of job i. This model can represent any scheduling problem with a regular measure and controllable processing times.

We have a nonlinear programming problem with nonlinear constraints. In a simplified form, the problem can be formulated as follows:

$$
\begin{array}{ll}
\text{Minimize} & Z_1(\bar{v},\bar{f}),\\
\text{Minimize} & Z_2(\bar{v},\bar{f},\bar{s}),\\
\text{Subject to} & \text{Constraints (4)–(6) imposed on } \bar{v} \text{ and } \bar{f},\\
 & \text{and constraint (7) imposed on } \bar{s},\bar{v} \text{ and } \bar{f},
\end{array}
$$

where $\bar{v} = (v_1,\ldots,v_N)$, $\bar{f} = (f_1,\ldots,f_N)$ and $\bar{s} = (s_1,\ldots,s_N)$.

In the next sections, we will represent our procedures to determine an efficient frontier and lower and upper bounds for the processing time of job i that will ease the solution procedures of such problems. For the next two sections, we will skip the indice i from the calculations for the sake of clearance and simplicity, because the lower and upper bound determination procedures will be the same for every job.

3. Determination of a lower bound

Tool life, machine power and surface roughness constraints on machining conditions can be seen in Fig. 1. At the intersection point with the surface roughness constraint, in case (a) machine power constraint is binding, while in (b) tool life constraint is binding. Minimizing the sum of machining and tooling costs of

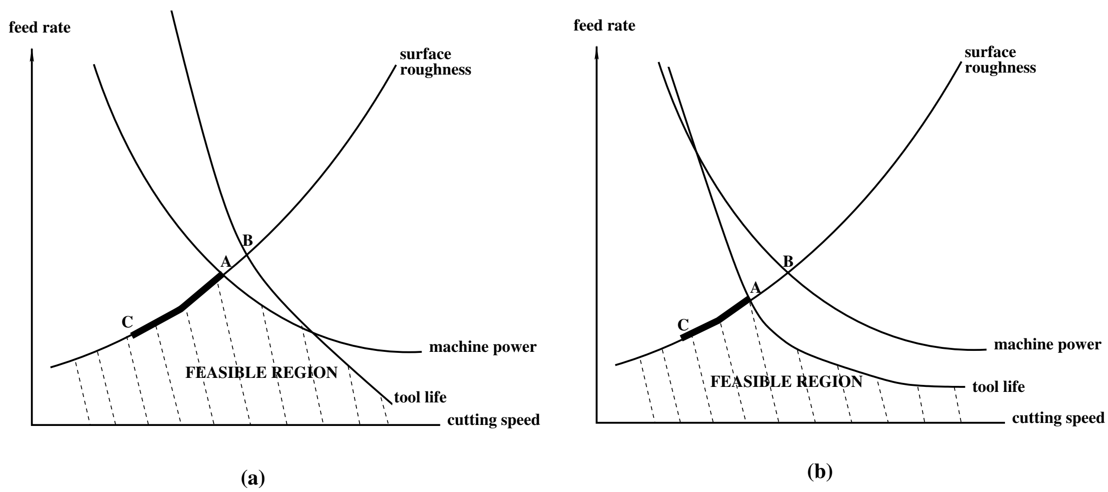

Fig. 1. Machine settings for upper and lower bounds and efficient frontier.

one job subject to these three constraints is called the single machining operation problem (SMOP). Two theorems on these constraints are given below.

Theorem 1 (Akturk and Avci [1]). *At least one of the surface roughness and machine power constraints is binding at optimality for SMOP.*

According to this theorem, any interior point of Fig. 1 for any case will give a higher processing time than the ones lying on the boundaries. Therefore, the machining conditions should always be set to a point on the boundary of the feasible region.

We proved the following theorem which will be used in determining both lower and upper bounds of the processing time.

Theorem 2. *The surface roughness constraint must be tight at optimality.*

Proof. The proof is given in Appendix A. □

According to Theorem 2, the upper edge of the feasible surface roughness line gives the minimum processing time and the lower edge gives the minimum usage rate for a given job. Therefore for any time and cost related objectives, the optimal (v,f) pair is on this line. This means that (v,f) pairs on the surface roughness constraint are dominant over all other (v,f) pairs with respect to regular scheduling measures. By the help of this theorem, the inequality (6) becomes an equation and we can write f in terms of v as follows:

$$f = \left(\frac{R \cdot d^l}{S} \cdot v^g\right)^{-1/h}. \tag{9}$$

By using Eqs. (2), (3) and (9), once we find the cutting speed value of a job, we can calculate feed rate, tool usage rate and processing time of it easily, or we can write U, v and f in terms P. This is the important practical part of the theorem which makes most of the calculations easier, since the number of independent variables is reduced to only one. U, v and f are written in terms of P below.

$$U = \left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha-g\beta)}{(h-g)}} \cdot d^{\frac{\gamma(h-g)+l(\alpha-\beta)}{(h-g)}} \cdot C^{-1} \cdot \left(\frac{R}{S}\right)^{\frac{(\alpha-\beta)}{(h-g)}} \cdot P^{\frac{h(\alpha-1)-g(\beta-1)}{(g-h)}}, \tag{10}$$

$$v = \left(\frac{\pi DL}{12}\right)^{\frac{h}{(h-g)}} \cdot \left(\frac{Rd^l}{S}\right)^{\frac{1}{(h-g)}} \cdot P^{\frac{h}{(g-h)}}, \tag{11}$$

$$f = \left(\frac{\pi DL}{12}\right)^{\frac{g}{(g-h)}} \cdot \left(\frac{Rd^l}{S}\right)^{\frac{1}{(g-h)}} \cdot P^{\frac{g}{(h-g)}}. \tag{12}$$

This theorem also shows that the minimum processing time is achieved by the (v,f) pair at the feasible intersection point on the (SR) constraint. This is the intersection of (SR) constraint with either (TL) or (MP) constraints. Point A in Fig. 1 shows this feasible intersection point, while point B corresponds to an infeasible one. In order to find the point A, we write (TL) and (MP) constraints in terms of P first.

Substituting v and f in terms of P, (TL) constraint reduces to:

$$\begin{aligned} 1 &\geqslant T' v^{(\alpha-1)} f^{(\beta-1)}, \\ 1 &\geqslant C^{-1}\left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha-g\beta)}{h-g}} d^{\frac{\gamma(h-g)+l(\alpha-\beta)}{h-g}} \left(\frac{R}{S}\right)^{\frac{(\alpha-\beta)}{h-g}} P^{\frac{h(\alpha-1)-g(\beta-1)}{g-h}}, \\ P &\geqslant C^{\frac{(g-h)}{h(\alpha-1)-g(\beta-1)}} \left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha-g\beta)}{h(\alpha-1)-g(\beta-1)}} d^{\frac{\gamma(h-g)+l(\alpha-\beta)}{h(\alpha-1)-g(\beta-1)}} \left(\frac{R}{S}\right)^{\frac{(\alpha-\beta)}{h(\alpha-1)-g(\beta-1)}}. \end{aligned} \tag{13}$$

Similarly, substituting v and f in terms of P, (MP) constraint reduces to:

$$\begin{aligned}
1 &\geqslant M'v^b f^c,\\
1 &\geqslant \frac{M}{H}\left(\frac{\pi DL}{12}\right)^{\frac{(hb-gc)}{h-g}} d^{\frac{e(h-g)+l(b-c)}{h-g}} \left(\frac{R}{S}\right)^{\frac{(b-c)}{h-g}} P^{\frac{(hb-gc)}{g-h}},\\
P &\geqslant \left(\frac{M}{H}\right)^{\frac{(h-g)}{(hb-gc)}} \left(\frac{\pi DL}{12}\right) d^{\frac{e(h-g)+l(b-c)}{(hb-bc)}} \left(\frac{R}{S}\right)^{\frac{(b-c)}{(hb-gc)}}.
\end{aligned} \tag{14}$$

From Eqs. (13) and (14), we can calculate the minimum value of the processing time which is the lower bound, P^{L}, as follows:

$$P^{\mathrm{L}} = \max \begin{bmatrix} C^{\frac{(g-h)}{h(\alpha-1)-g(\beta-1)}} \left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha-g\beta)}{h(\alpha-1)-g(\beta-1)}} d^{\frac{\gamma(h-g)+l(\alpha-\beta)}{h(\alpha-1)-g(\beta-1)}} \left(\frac{R}{S}\right)^{\frac{(\alpha-\beta)}{h(\alpha-1)-g(\beta-1)}} \\ \left(\frac{M}{H}\right)^{\frac{(h-g)}{(hb-gc)}} \left(\frac{\pi DL}{12}\right) d^{\frac{e(h-g)+l(b-c)}{(hb-bc)}} \left(\frac{R}{S}\right)^{\frac{(b-c)}{(hb-gc)}} \end{bmatrix}. \tag{15}$$

As a result, the minimum processing time for a job, i.e. the lower bound, can be found exactly using the job, CNC machine and cutting tool related parameters directly.

4. Determination of an upper bound

The optimal solution to SMOP (i.e. optimal (v,f) pair), which minimizes the sum of machining and tooling cost of one job under (TL), (MP) and (SR) constraints, provides an upper bound of the job processing time for scheduling problems with any regular measure. It also provides the lower bound of the objective function value that will be used to define $Z_1^{\min}$ in Section 6. The sum of machining and tooling costs are represented in Fig. 2. The point C represents the optimal solution of SMOP which is the same with point C in Fig. 1. Any point below this one on the (SR) constraint will result in a higher processing time and will

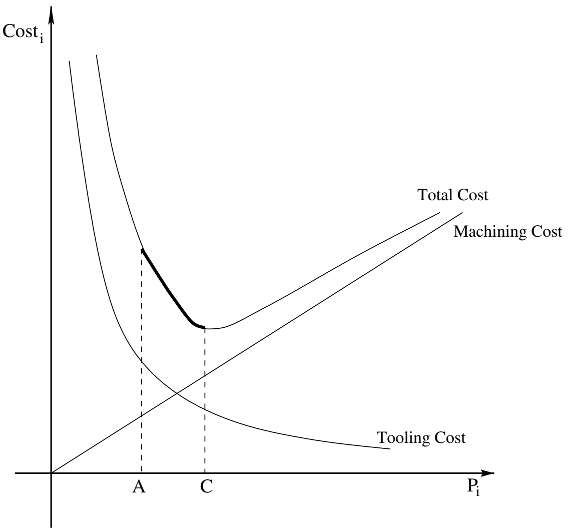

Fig. 2. Manufacturing cost components of a job.

not improve neither cost nor any regular scheduling measure. Although the points below will give higher machining cost but lower tooling cost, the summation of these two cost terms will be higher as shown in Fig. 2. As we know, completion time, tardiness or makespan related objectives never decrease with increasing processing time values for jobs. For example, Makespan $= \sum_{i=1}^{N} P_i$, which is directly proportional to the processing times of jobs. In other words, the points giving higher processing time values than point C (the ones below point C on (SR) in Fig. 1) will not improve the makespan criterion. Therefore, beyond this point, both Z_1 and Z_2 objectives become worse.

The relationship between processing time and terms of Z_1 and Z_2 can be seen more clearly by writing them in terms of processing time. By using Eq. (10), tooling cost reduces to:

$$\text{Tooling cost} = T \cdot U = \left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha - g\beta)}{(h-g)}} d^{\gamma} T(C)^{-1} (R')^{\frac{(\alpha-\beta)}{(h-g)}} \cdot P^{\frac{h(1-\alpha)-g(1-\beta)}{(h-g)}}.$$

Due to the possible values of the technical coefficients such that $\alpha > \beta > 1$, $h > 0$ and $g < 0$, the tooling cost is a convex function which is also obvious from Fig. 2 that it increases as processing time decreases and decreases as processing time increases.

Machining Cost $= O \cdot P$. Apparently machining cost is directly proportional to the processing time and the optimal machine settings giving the minimum cost which is the aggregation of tooling and machining costs will yield a processing time value and any deviation from it will not improve the solution since it has adverse effects on two types of costs.

As a result, any point below point C will give worse Z_1 and Z_2 values, therefore the corresponding processing time value of point C can be used as an upper bound for P in scheduling problems with regular measures. The next step is calculating this upper bound, P^{U}. The optimal P value when the SMOP is rewritten in terms of P is exactly P^{U}. The findings in this study simplified the SMOP even further and we can rewrite the SMOP in terms of P as follows:

$$\begin{aligned} &\text{Minimize} \quad O \cdot P + \left(\frac{\pi DL}{12}\right)^{\frac{(h\alpha - g\beta)}{(h-g)}} d^{\gamma} T(C)^{-1} (R')^{\frac{(\alpha-\beta)}{(h-g)}} \cdot P^{\frac{h(1-\alpha)-g(1-\beta)}{(h-g)}}, \\ &\text{Subject to} \quad P \geqslant P^{\mathrm{L}}. \end{aligned}$$

The objective function is the summation of machining and tooling costs respectively, and the only constraint can be calculated easily by using Eq. (15). After taking the derivative of the objective function with respect to P and solving it, feasibility check has to be made. If the resulting P value satisfies the constraint, then it is optimal. If not then, P^{L} is optimal yielding $P^{\mathrm{U}} = P^{\mathrm{L}}$. As a result, P^{U} is very easy to compute.

In Fig. 2, the curve between points A and C forms an efficient frontier of one job in terms of cost and processing time and this also provides a basis for efficient frontier of the whole schedule. Moreover, the nonlinear constraints of (4)–(6) can be replaced by a new linear bound of $P^{\mathrm{L}} \leqslant P \leqslant P^{\mathrm{U}}$ Since it is linear and P^{L} and P^{U} values are very easy to calculate, this replacement will decrease the computational requirements of the original nonlinear bi-criteria problem.

5. Numerical example

A job is given with the following attributes $S = 300$, $L = 5$, $d = 0.2$, and $D = 3.2$. The coefficients of the assigned tool are $(\alpha, \beta, \gamma, C, b, c, e, M, g, h, l, R, T) = (4, 1.4, 1.16, 40\,960\,000, 0.91, 0.78, 0.75, 2.394, -1.52, 1.004, 0.25, 204\,620\,000, 4)$. The operating cost and maximum power of the CNC machine are $O = 0.5$ \$/minute and $H = 10$ hp, respectively.

From Eq. (15), $P^{\mathrm{L}} = \max(0.18, 0.40) = 0.40$ min $= 24$ seconds. This means that the MP constraint is binding (see part (a) of Fig. 1).

To calculate the upper bound, we will solve the SMOP presented in the previous section.

$$\begin{aligned}&\text{Minimize} \quad 0.5P + 0.333P^{-1.43},\\ &\text{Subject to} \quad P \geqslant 0.40.\end{aligned}$$

By derivation, the value minimizing the objective function is $P = 0.98$. This also satisfies $P \geqslant 0.40$, therefore it is the optimal solution, and $P^{\mathrm{U}} = 0.98$ minute = 59 seconds.

The job in our problem can be processed in 24–59 seconds. Any duration which is not in this interval is either infeasible or worse in terms of the objective function value. The settings of machine are ($v = 445$ and $f = 0.024$) for the lower and ($v = 311$ and $f = 0.014$) for the upper bound of P. These two settings correspond to points A and C respectively in Figs. 1 and 2. These settings of the CNC machine are very easy to change by a single line in a G code or in a APT language. The feed rate is the speed of the cutting tool moving along the part profile or from one point to another. It is defined as the distance (in inches or millimeters) that the tool moves in 1 minute or in one revolution of the machine tool spindle. In this paper, we measured the feed rate in inches per minute (ipr). For example, the following G code in the NC program will be included to set the feed rate for the lower bound (we will use 0.014 for the upper bound).

```
G99 F0.024;
```

In the APT language, we have to add the following statement:

```
FEDRAT/0.024, IPR
```

In CNC programming, a feed rate statement should be specified before the motion statement. The FEDRAT statement is modal; it remains in effect until changed by another FEDRAT statement. In addition to the upper and lower bounds, we can also find a closed form equation of the efficient frontier of this job given in Fig. 3 to evaluate manufacturing cost and processing time tradeoffs as shown below:

$$v^{(h-g)} \cdot P^h = \left(\frac{R \cdot d^l}{S}\right) \cdot \left(\frac{\pi DL}{12}\right)^h \Rightarrow v^{2.524} \cdot P^{1.004} = 1\,921\,592.47.$$

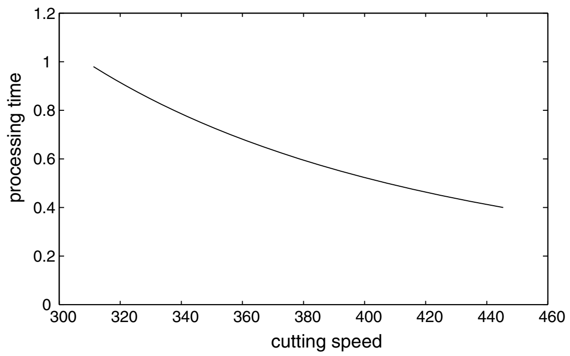

Fig. 3. Processing time and cutting speed tradeoff for a single manufacturing operation.

6. Efficient frontier of makespan and manufacturing cost

In this section, we propose an exact method and approximation approaches in order to determine a set of efficient points of makespan and manufacturing cost objectives. The NLP formulation presented in the problem definition section reduced to the formulation below as a consequence of the proposed bounding scheme.

$$\begin{aligned} &\text{Minimize} \quad Z_1 = \sum_{i=1}^{N}(O \cdot P_i + T_i \cdot U_i) \quad (\text{Machining Cost} + \text{Tooling Cost}), \\ &\text{Minimize} \quad Z_2 = \sum_{i=1}^{N} P_i \quad (\text{Makespan}), \\ &\text{Subject to} \quad P_i^{\mathrm{L}} \leqslant P_i \leqslant P_i^{\mathrm{U}} \quad \forall i. \end{aligned}$$

In Fig. 4, the point A is the point where Z_1 has its maximum value and Z_2 has its minimum. In fact this point is reached when machine settings of each job is assigned to the pair giving the minimum P, i.e. point A in Figs. 1 and 2. Also, at point C of Fig. 4, Z_1 is at the minimum while Z_2 is at the maximum. This point is also achieved when settings of each job is at point C (minimum cost point) in Figs. 1 and 2. Let $Z_1^{\max}$, $Z_1^{\min}$, $Z_2^{\max}$ and $Z_2^{\min}$ be these four points in Fig. 4. All points between A and C on the efficient frontier are efficient points. A point (Z_1^b, Z_2^b) is said to be efficient with respect to cost and makespan criteria if there does not exist another point (Z_1^d, Z_2^d) such that $Z_1^d \leqslant Z_1^b$ and $Z_2^d \leqslant Z_2^b$ with at least one holding as a strict inequality [11]. For a more detailed discussion on multicriteria scheduling, we refer to T'kindt and Billaut [11].

As discussed above, we can easily calculate the minimum and maximum makespan values for a given problem. The procedure of efficient point generation (EPG) can be outlined as follows:

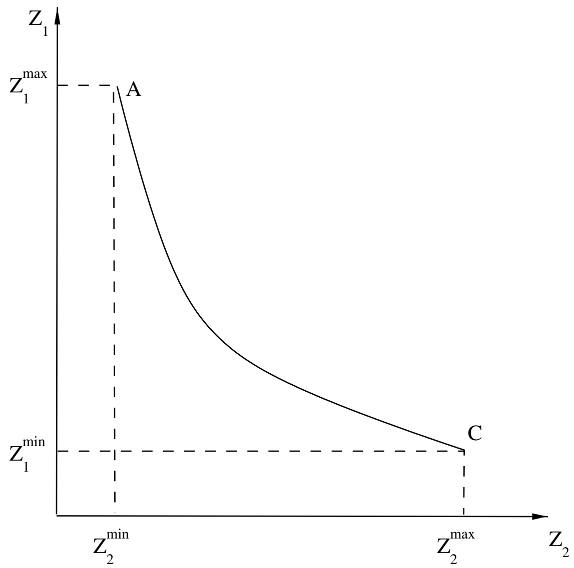

Fig. 4. Efficient frontier of makespan and total manufacturing cost objectives.

Procedure EPG:

Step 0. Calculate P_i^{U} and P_i^{L} for every job i, and then calculate $Z_1^{\max}$, $Z_2^{\max}$, $Z_1^{\min}$ and $Z_2^{\min}$.
Step 1. Initially let $k = \frac{Z_2^{\max} - Z_2^{\min}}{\Omega \cdot N}$ and $j = 1$.
Step 2. Let $Z_2^{[j]} = Z_2^{\max} - j \cdot k$. If $Z_1^{[j]} \leqslant Z_2^{\min}$, go to Step 5.
Step 3. Find the corresponding minimum $Z_1^{[j]}$ value for a given $Z_2^{[j]}$.
Step 4. Increase j by one, $j = j + 1$, and go to Step 2.
Step 5. List all $(Z_1^{[j]}, Z_2^{[j]})$ pairs for $j = 1, 2 \ldots, (\Omega \cdot N)$. These points are all efficient points.

Ω is the tuning parameter of the step-size k. For lower values of Ω (i.e. 0.25, 0.5) the step-size increase and we get fewer number of efficient points. On the other hand, the results get more accurate for higher values of $\Omega = 1, 2, \ldots$ since step size will be narrowed.

The main idea behind the EPG algorithm is to find a set of discrete efficient points on the efficient frontier such that we can provide an approximation of the continuous tradeoff curve. An important computational difficulty in this procedure is calculating the corresponding manufacturing cost value when the makespan value is given (Step 3). In other words, we can calculate the individual job processing times that minimize the total manufacturing cost $Z_1^{[j]}$ for a given makespan value, $Z_2^{[j]}$. This is the most challenging step of the algorithm and the other steps can easily be implemented if we find an appropriate procedure for Step 3. We first propose an exact algorithm in Section 6.1 in order to find the optimum $Z_1^{[j]}$ value for a given $Z_2^{[j]}$ in Step 3. Obviously, this exact algorithm could become computationally demanding for large problems. Therefore, we also propose four heuristic algorithms in Sections 6.2–6.5 to compute the value of $Z_1^{[j]}$ when $Z_2^{[j]}$ is fixed. These heuristics differ in terms of computational requirements and solution quality as discussed in Section 7.

6.1. Exact procedure

Given the makespan value, the corresponding manufacturing cost value can be found exactly via solving the formulation below.

$$\begin{aligned} \text{Minimize} \quad & Z_1 = Z_1^{[j]}, \\ \text{Subject to} \quad & Z_2 = Z_2^{[j]}, \\ & P_i^{\mathrm{L}} \leqslant P_i \leqslant p_i^{\mathrm{U}} \quad \forall i. \end{aligned}$$

If we rearrange the terms of U_i in terms of P_i in Eq. (10) then

$$U_i = A_i \cdot P_i^a,$$

where

$$A_i = \left(\frac{\pi D_i L_i}{12}\right)^{\frac{(h\alpha - g\beta)}{(h-g)}} \cdot d_i^{\frac{\gamma(h-g)+l(\alpha-\beta)}{(h-g)}} \cdot C^{-1} \cdot \left(\frac{R}{S_i}\right)^{\frac{(\alpha-\beta)}{(h-g)}}$$

and

$$a = \frac{h(\alpha - 1) - g(\beta - 1)}{(g - h)}.$$

After inserting this equation to the formulation, it reduces to:
(NLP):

$$\text{Minimize} \quad O \cdot \sum_{i=1}^{N} P_i + \sum_{i=1}^{N} T \cdot A_i \cdot (P_i)^a = Z_1^{[j]},$$

$$\text{Subject to} \quad \sum_{i=1}^{N} P_i = Z_2^{[j]},$$

$$P_i^{\mathrm{L}} \leqslant P_i \leqslant P_i^{\mathrm{U}} \quad \forall i.$$

Since summation of processing times are equal to the given makespan value, $Z_2^{[j]}$, the first part of the objective function is a constant value and the problem reduces to (NLP_R). From now on, let S(P) denote the value of optimal solution found for any problem P, then $\mathrm{S(NLP)} = \mathrm{S(NLP\_R)} + O \cdot Z_2^{[j]}$.
(NLP_R):

$$\text{Minimize} \quad \sum_{i=1}^{N} T \cdot A_i \cdot (P_i)^a, \tag{16}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} P_i = Z_2^{[j]}, \tag{17}$$

$$P_i^{\mathrm{L}} \leqslant P_i \leqslant P_i^{\mathrm{U}} \quad \forall i. \tag{18}$$

According to this algorithm, Step 3 of EPG is replaced by

Step 3. Solve NLP_R. Return $Z_1^{[j]} = \mathrm{S(NLP\_R)} + O \cdot Z_2^{[j]}$.

It gives the minimum manufacturing cost for a given makespan exactly. The benefit of the bounds become more apparent in this procedure because problems in which the nonlinear terms are restricted to the objective function are generally easier to solve than those in which nonlinearities appear both in objective function and constraints. In the following subsections, we propose four different approximation algorithms to solve the NLP_R problem.

6.2. Lagrangean relaxation

This algorithm starts with the NLP_R formulation. As an initial step, the first constraint, Eq. (17), of the NLP_R is dualized with a nonnegative Lagrangean multiplier Λ.
(LR_Λ):

$$\text{Minimize} \quad \sum_{i=1}^{N} T \cdot A_i \cdot (P_i)^a + \Lambda \left(\sum_{i=1}^{N} P_i - Z_2^{[j]} \right), \tag{19}$$

$$\text{Subject to} \quad P_i^{\mathrm{L}} \leqslant P_i \leqslant P_i^{\mathrm{U}} \quad \forall i.$$

We can find the P_i values for a given multiplier Λ by taking a derivative of the objective function (Eq. (19)) with respect to P_i and by equating it to zero as follows:

$$P_i = \left[\frac{T A_i a}{-\Lambda} \right]^{\frac{1}{1-a}}. \tag{20}$$

Since we did not consider the bounds, there might be some P_i values which have exceeded their upper bounds, or others which are lower than their lower bounds. Therefore, we set the P_i to the bound which is not satisfied. For instance, if the P_i value is greater than its upper bound, then it is set to the upper bound. P_i values are calculated for all i's and they are used to calculate the total cost, Z_1. It is important to note that at any iteration of the Lagrangean relaxation algorithm, P_i values may not be optimal for the LR_Λ, nor feasible for the (NLP), in terms of Eq. (17), but they still give an objective function value which is a lower bound for the S(NLP), i.e. $Z_1^{[j]}$.

The overall Lagrangean procedure can be outlined as follows:

Step 3.1. Initialize the Lagrangean multiplier and scalar δ (e.g., $\Lambda = 0$, $\delta = 2$). Set iteration number $r = 0$.

Step 3.2.1. Calculate P_i values via Eq. (20). If $P_i^r > P_i^{\mathrm{U}}$ for any job i then $P_i^r = P_i^{\mathrm{U}}$. Similarly, if $P_i^r < P_i^{\mathrm{L}}$ for any job i then $P_i^r = P_i^{\mathrm{L}}$. Calculate the amount of deviation, , as $\Upsilon = \sum_{i=1}^{N} P_i - Z_2^{[j]}$. If $-0.1k \leqslant \ \leqslant 0.05k$, calculate the total cost, report it as $Z_1^{[j]}$, and go to Step 4 of EPG.

Step 3.2.2. Set $r = r + 1$. If $r = 100$, report the total cost of the solution which has the smallest value found so far, as $Z_1^{[j]}$ and continue with Step 4 of EPG.

Step 3.2.3. Update the multiplier, $\Lambda = \Lambda + \delta$. δ is a scalar satisfying $0 < \delta \leqslant 2$. if Λ becomes negative, update δ until Λ is no longer negative. Return to Step 3.2.1.

6.3. Job response function—variation I

In the EPG procedure, we start from the maximum makespan value, and decrease it by a step-size k at each iteration. Initially, the processing time of each job is set to its upper bound. Therefore, the main problem at each iteration is to find P_i values in such a way that they minimize the manufacturing cost while their sum is equal to the desired makespan. This means that processing times of some jobs will be reduced. We propose a response function in order to find which jobs are more likely to give a minimum cost increase for a reduction in their processing times.

Let $R_i(\varepsilon)$ be the response of job i, i.e. the increase in the contribution of job i to the total cost, Z_1, as a response to ε amount decrease in its processing time.

$$R_i(\varepsilon) = A_i[(P_i - \varepsilon)^a - (P_i)^a].$$

Since it measures the response, distributing the desired reduction, k, to the processing times of the jobs proportional to their response values is the aim of this algorithm. In sum, we start with the P_i^{U} values and then look for the jobs when their processing times are reduced that result in the smallest increase with respect to the manufacturing cost, i.e. the "biggest bang for the buck" approach. The main steps of the algorithm are as follows:

Step 3.1. Let r be 1 and $\varepsilon = r\frac{k}{N}$ initially.

Step 3.2. Calculate $R_i(\varepsilon)$ for each job i and set

$$\lambda_i = \left(1 - \frac{R_i(\varepsilon)}{\sum_{l=1}^{N} R_l(\varepsilon)}\right) \Big/ (N - 1).$$

Step 3.3. Using multipliers λ_i, calculate the P_i values. $P_i = P_i^{\mathrm{U}} - \lambda_i k$. Check the bounds. If bounds are not satisfied, the tight bound is optimal, i.e. if the P_i value is less than its lower bound then it is set to its lower bound. Since some jobs may be set to their lower bounds, there will be a gap between the desired reduction and the total reduction in processing times. This difference is distributed among the processing times of jobs in the following steps until there is no gap.

Step 3.3.1. Let η be the difference between the desired reduction and the total reduction in processing times, and ρ be the set of jobs for which $P_i > P_i^{\mathrm{L}}$. Then,

$$\lambda_i = \left(1 - \frac{R_l(\varepsilon)}{\sum_{l\in\rho} R_l(\varepsilon)}\right) \Big/ (N-1), \quad \forall i \in \rho.$$

Step 3.3.2. Calculate the new P_i values; $P_i = P_i^{\mathrm{U}} - \lambda_i \cdot \eta$. Check the bounds. Repeat Step 3.3.1 until there exists no gap.

Step 3.3.3. Calculate the total manufacturing cost objective function value.

Step 3.4. Increase r by 1 and go to Step 3.2 until r is equal to N. Report the minimum cost found so far as $Z_1^{[j]}$.

For the smaller values of the response function, corresponding λ value gets higher which results in more reduction of the processing time from its upper bound. Since the λ_i values add up to 1 and bounds are always satisfied, summation of P_i's always add up to $Z_2^{[j]}$. Therefore, infeasibility is not faced in this procedure.

6.4. Job response function—variation II

$R_i(\varepsilon)$ corresponds to an increase on manufacturing cost when the processing time of job i is decreased by ε time units. Similarly, $R_i(1)$ is the sensitivity of job i to a unit decrease. Furthermore, at each iteration of the EPG procedure, we decrease the makespan by k time units. Therefore, $R_i(k)$ measures the corresponding change in Z_1 if the reduction is achieved by changing the processing time of job i only. Consequently, the amount of required decrease in the processing time of job i, ε, can take any value of $0, k/N, 2k/N, 3k/N, \ldots, k$.

In this variation, the k amount of processing time decrease is distributed to the jobs according to their responses as it is in the first variation but in the response calculations, ε takes values of $k/N, k/N-1, \ldots, k$. That means we first allocate the required amount k equally among N jobs, then $N-1$ jobs, etc. Obviously, different allocation schemes might lead to different solutions.

6.5. Knapsack-based algorithm

We can easily calculate the minimum (point C) and maximum (point A) manufacturing cost values for each job, as well as the exact nonlinear cost function between these two points as shown in Fig. 2. The corresponding processing times of these two points are simply the bounds on the processing time of job i. We first convert this nonlinear cost function into a set of discrete points in such a way that the difference between the actual cost value and the cost value at the selected point is less than the approximation error from the actual cost value. In order to generate alternative points for each job, we first draw a line between two adjacent points. If the approximation error, the maximum distance between the line and the cost curve, is greater than Ψ, then a middle point is inserted between these two points. This step is repeated until all points are added. For each point, we know the processing time and the corresponding manufacturing cost value. As a result, the problem reduces to choosing the optimal (Cost, Processing Time) pair for each job which gives the minimum cost and satisfies the desired makespan. The steps of the algorithm are:

Step 3.1. Find the alternative points for every job i. $(\mathrm{Cost}_{il}, P_{il})$ is the cost and processing time pair for job i when lth alternative is selected.

Step 3.2. Let F_i be the number of alternatives found above for job i. X_{il} is the binary decision variable which is equal to 1 if alternative l of the job i is selected. We solve the following knapsack problem and the optimal value of it is reported as $Z_1^{[j]}$.

$$\text{Minimize} \quad \sum_{i=1}^{N}\sum_{l=1}^{F_i} \text{Cost}_{il} X_{il}, \tag{21}$$

$$\text{Subject to} \quad \sum_{i=1}^{N}\sum_{l=1}^{F_i} P_{il} X_{il} = Z_2^{[j]}, \tag{22}$$

$$\sum_{l=1}^{F_i} X_{il} = 1 \quad \forall i = 1, 2, \ldots, N, \tag{23}$$

$$X_{il} \in \{0, 1\}.$$

This algorithm does not guarantee optimality for the original problem since we convert the nonlinear continuous function into a set of discrete points.

7. Computational results

In this section, we performed a computational study to test the performance of the four approximation algorithms by comparing them with the exact algorithm. All of the five algorithms are coded in the C language and compiled with the GNU C compiler. The MIP formulation used in the knapsack-based algorithm is solved using the callable library routines of CPLEX 7.1 MIP solver. The NLP formulation of the exact algorithm is formulated in GAMS 2.25 and solved by MINOS 5.3. All problems are solved on a 400 MHz UltraSPARC station. There are three experimental factors that can affect the efficiency of the algorithms as listed in Table 1. The experimental design is a 2^3 full factorial design with two different levels each.

The first factor is effective on the lower bound of the processing time, and the second one is used to control the upper bound. The last factor determines the size of the problem.

- *H:* Maximum available machine horse power for all jobs. The increase in the value of H in the machine power constraint shifts up the intersection point in Fig. 1 on the surface roughness constraint, which provides higher feasible cutting speed and feed rate values. The increase in (v,f) values in turn results in a wider range of possible processing time alternatives for each job.
- *T:* Tooling cost. The total manufacturing cost is composed of two parts (see Eq. (1)). The first part, machining cost, increases with increasing processing time values and the second part, tooling cost, decreases with increasing processing time values. When the cost of the tool increases, the impact of the second part on the total cost becomes higher. As a result, the optimal setting which minimizes the total cost will have lower v and f values (a point below the point C in Fig. 1). Since this point is used as an upper bound for the processing time (P^{U}), the upper bound increases when T increases.

Table 1
Experimental design factors

Factors	Definition	Level 1	Level 2
H	Machine power	5	10
T	Cost of the tool	UN[6, 10]	UN[13, 17]
N	Number of jobs	50	100

- N: The first two factors are used to control the bounds on the processing times. The number of jobs is a factor which determines the problem size. When the problem size is large, it takes more CPU time to solve the problem, and the resulting cost and makespan objective values are expected to be high.

There are 10 different types of tools, and each job is equally likely to be processed by one of them. The technological coefficients of the tools are given in Table 2. The other problem specific parameters are selected randomly from the intervals of $O = 1$, $S_i = \text{UN}[150, 250]$, $d_i = \text{UN}[0.05, 0.30]$, $L_i = \text{UN}[4,6]$, and $D_i = \text{UN}[1,4]$, where $\text{UN}[a, b]$ is uniform distribution in interval $[a, b]$. Furthermore, the maximum allowable approximation error, Ψ, in the knapsack-based algorithm is set to 0.05 after some trial runs. The parameter Ω, which determines the step-size and number of efficient points to be found in the EPG algorithm, is set to 0.4 when the number of jobs, N, is 50, and to 0.2 when N is equal to 100.

We proposed five algorithms, one being exact and others approximation approaches. We have a 2^3 full factorial design. We took five replications for each factor combination and 20 different EPG iterations for each replication resulting in $2^3 * 5 * 20 = 800$ individual runs for each algorithm. As discussed earlier, each algorithm finds the corresponding manufacturing cost value for a given desired makespan value. In order to give an idea how these results are collected for each algorithm, we report the results for each of the proposed algorithm for one sample replication with a factor combination of $(N, H, T) = (50, 1, 0)$ in Table 3. The first point (maximum makespan, minimum cost) and the last point (minimum makespan, maximum cost) in this table correspond to the points A and C in Fig. 4 respectively, and given as the same starting and ending points for each algorithm. They are easily calculated by substituting the upper and lower bounds of the processing times into the cost function. These results are also plotted in Fig. 5 to indicate the shape of the efficient frontier of our bi-criteria problem.

The performance measures used in evaluating the experimental results are the absolute percentage deviation of the approximation algorithms from the optimal solution and the run times in CPU seconds. The absolute percentage deviation of each run is calculated as $(app - opt)/opt$, where app is the manufacturing cost value delivered by an approximation algorithm and opt is the optimal solution for a given makespan value. For four approximation algorithms, the minimum, average and maximum values of the absolute percent deviations from the optimal results out of 800 runs are given in Table 4. The factor combinations at which the minimum and maximum values are achieved are also reported in the same table. When H and T are at their low levels, i.e. at level 0, we have the minimum range $(P^{\text{U}} - P^{\text{L}})$ for the processing time alternatives. On the other hand, the most difficult problem instance in terms of the computational requirements is the experimental setting of $(N, H, T) = (100, 1, 1)$ since we have the largest problem size and the maximum number of feasible processing time settings (or the maximum range of $P^{\text{U}} - P^{\text{L}}$). We report the average CPU times in CPU seconds for each parameter setting in Table 5.

Table 2
Technical coefficients of the tools

Tool	α	β	γ	C	b	c	e	M	g	h	l	R
1	4.00	1.40	1.16	40 960 000	0.91	0.78	0.75	2.394	−1.52	1.004	0.25	204 620 000
2	4.30	1.60	1.20	37 015 056	0.96	0.70	0.71	1.637	−1.60	1.005	0.30	259 500 000
3	3.70	1.30	1.10	13 767 340	0.90	0.75	0.72	2.315	−1.45	1.015	0.25	202 010 000
4	3.70	1.28	1.05	11 001 020	0.80	0.75	0.70	2.415	−1.63	1.052	0.30	205 740 000
5	4.10	1.26	1.05	48 724 925	0.80	0.77	0.69	2.545	−1.69	1.005	0.40	204 500 000
6	4.10	1.30	1.10	57 225 273	0.87	0.77	0.69	2.213	−1.55	1.005	0.25	202 220 000
7	3.70	1.30	1.05	13 767 340	0.83	0.75	0.73	2.321	−1.63	1.015	0.30	203 500 000
8	3.80	1.20	1.05	23 451 637	0.88	0.83	0.72	2.321	−1.55	1.016	0.18	213 570 000
9	4.20	1.65	1.20	56 158 018	0.90	0.78	0.65	1.706	−1.54	1.104	0.32	211 825 000
10	3.80	1.20	1.05	23 451 637	0.81	0.75	0.72	2.298	−1.55	1.016	0.18	203 500 000

Table 3
Results of the five algorithms for a single replication of factor combination (50, 1, 0)

Makespan	Manufacturing cost				
	Lagrangean	Response-I	Response-II	Knapsack	Optimal
35.4	60.2	60.2	60.2	60.2	60.2
33.9	60.3	60.3	60.8	64.2	60.2
32.5	60.9	60.6	61.4	68.3	60.5
31.0	68.1	61.0	65.8	72.5	61.0
29.5	61.9	61.8	66.1	76.8	61.7
28.0	82.0	62.9	70.1	81.3	62.7
26.5	65.5	64.3	86.0	85.7	64.0
25.0	71.3	66.3	106.3	91.5	65.8
23.5	68.3	76.1	115.2	97.6	68.1
22.1	78.8	86.4	121.3	104.9	70.9
20.6	93.1	104.2	138.9	111.0	74.5
19.1	79.7	122.4	144.8	118.5	79.2
17.6	102.9	135.3	149.9	126.1	85.1
16.1	93.6	145.6	169.1	135.1	92.4
14.6	103.4	170.7	194.8	143.1	102.2
13.1	116.8	201.0	216.5	153.8	115.6
11.7	134.4	245.4	275.3	166.1	134.5
10.2	160.4	292.1	328.9	181.5	158.5
8.7	198.5	348.6	375.6	206.1	195.3
7.2	259.1	374.4	388.1	257.7	251.2
5.7	405.7	405.7	405.7	405.7	405.7

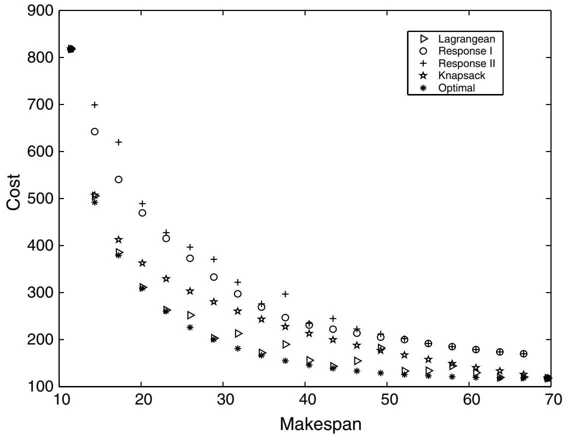

Fig. 5. Efficient points found by the proposed algorithms.

When we analyze the computational results, it is important to note that we can solve decent size problems optimally due to the proposed bounding scheme. This also supports our initial claim that the problems in which the nonlinear terms are restricted to the objective function are generally easier to solve than those in which nonlinearities appear both in objective function and constraints. For the larger size problems, the

Table 4
Absolute deviations of the four approximation algorithms

		min	ave	max
Lagrangean	% Deviation	0.00	9.90	213.03
	(N,H,T)	$(50,0,0)$	–	$(50,0,1)$
Response-I	% Deviation	0.00	35.52	131.98
	(N,H,T)	$(50,1,0)$	–	$(100,1,1)$
Response-II	% Deviation	0.06	41.75	198.21
	(N,H,T)	$(50,1,0)$	–	$(100,1,1)$
Knapsack	% Deviation	0.37	27.94	99.05
	(N,H,T)	$(100,1,0)$	–	$(100,1,1)$

Table 5
Average CPU times of the five algorithms in seconds

N	H	T	Lagrangean	Response-I	Response-II	Knapsack	Optimal
50	0	0	2.06	0.02	0.02	1.68	1.61
50	0	1	2.38	0.02	0.02	1.53	1.42
50	1	0	1.96	0.02	0.02	6.85	1.26
50	1	1	2.28	0.03	0.02	9.88	1.24
100	0	0	4.80	0.09	0.08	2.99	1.40
100	0	1	5.25	0.08	0.09	3.04	1.87
100	1	0	4.51	0.09	0.09	17.50	2.25
100	1	1	4.76	0.08	0.10	34.23	1.85

Lagrangean relaxation based algorithm seems promising in terms of the solution quality, whereas the job response function based algorithms in terms of the CPU time.

8. Conclusion

In this study, a bi-criteria scheduling problem is dealt in which the processing times are assumed to be controlled by adjusting the machine settings. During the machining conditions optimization study, we found a very practical theorem which can also be used in other scheduling studies. The bounds that we suggested ease the modeling of the processing times and also ease the efficient frontier determination in bi-criteria scheduling as demonstrated on the problem of minimizing makespan and manufacturing cost simultaneously. By utilizing the proposed bounding mechanism, we developed an exact algorithm and four heuristic approaches to determine a set of discrete efficient points to approximate the continuous tradeoff curve in a reasonable computation time. As a result, the proposed study provides a good starting point to demonstrate how the process planning and scheduling decisions can be integrated. For a further research, we will study total completion time problem on a single CNC machine instead of makespan criterion, so that job sequencing decisions and processing time selection problems will have an impact on each other.

Acknowledgements

This paper was a part of Rabia Köylü Kayan's Ph.D. dissertation, who passed away on December 21, 2003. She was a very bright researcher and a good friend to her colleagues. We all miss her.

Appendix A. Proof of Theorem 2

The geometric programming (GP) model proposed by Akturk and Avci [1] is as follows:

$$\begin{aligned}
\text{(GP)}\quad & \text{Minimize} \quad && \text{SMOP}_i = C'_1 v_i^{-1} f_i^{-1} + C'_2 v_i^{(\alpha_i-1)} f_i^{(\beta_i-1)}, \\
& \text{Subject to} && \text{Constraints (4)–(6)} \\
& && v_i, f_i > 0,
\end{aligned}$$

where

$$C'_1 = \frac{\pi D_i L_i O}{12} \quad \text{and} \quad C'_2 = \frac{\pi D_i L_i d_i^{\gamma_i} T_i}{12 C_i}.$$

Denoting the dual variables by $B_1, B_2, \ldots, B_5$ the dual formulation for the SMOP problem can be written as follows:

$$\text{(Dual GP) Maximize} \quad Q^* = \left(\frac{C'_1}{B_1}\right)^{B_1} \cdot \left(\frac{C'_2}{B_2}\right) B \cdot (T'_i)^{B_3} \cdot (M'_i)^{B_4} \cdot (R'_i)^{B_5},$$

$$\text{Subject to} \quad B_1 + B_2 = 1, \tag{24}$$

$$-B_1 + (\alpha_i - 1) \cdot B_2 + (\alpha_i - 1) \cdot B_3 + b \cdot B_4 + g \cdot B_5 = 0, \tag{25}$$

$$-B_1 + (\beta_i - 1) \cdot B_2 + (\beta_i - 1) \cdot B_3 + c \cdot B_4 + h \cdot B_5 = 0, \tag{26}$$

$$B_1, B_2, B_3, B_4, B_5 \geqslant 0,$$

where

$$T'_i = \frac{\pi D_i L_i d_i^{\gamma_i}}{12 C_i}, \quad M'_i = \frac{M d_i^e}{H} \quad \text{and} \quad R'_i = \frac{R d_i^l}{S_i}.$$

The first two dual variables B_1 and B_2 correspond to the each of the primal objective function terms, respectively. Therefore, their summation must be equal to 1, also known as normality constraint, as stated in the first dual constraint. The other dual variables B_3, B_4 and B_5 correspond to the primal problem constraints, respectively. Furthermore, there is a dual constraint for each primal variable, v_i and f_i, respectively, known as orthogonality constraints.

Each of the constraints of the primal problem can be either loose or tight at optimality. Due to Theorem 1, we know that at least one of the surface roughness and machine power constraints is binding at optimality for SMOP. Since the tool life constraint cannot be binding by itself, we can set the dual variable corresponding to the tool life constraint equal to zero ($B_3 = 0$) due to the complementary slackness conditions. When the machine power constraint is tight and 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 again. When we solve Eqs. (24)–(26), we find that $B_4 = -((\alpha_i - \beta_i)/(b - c))B_2$.

Due to Gorczyca [5], $b > c > 0$ and $\alpha_i > \beta_i > 1$ that means increasing cutting speed or feed rate always require more machine power and tool usage. Moreover machine power and tool life are more sensitive to the changes in cutting speed than feed rate yielding $b > c > 0$ and $\alpha_i > \beta_i > 1$. If $Y_2 > 0$, then $Y_4 < 0$, which makes this case infeasible. Therefore, the machine power constraint cannot be binding by itself.

In the other case where the surface roughness constraint is binding then B_5 should be nonnegative because of the dual feasibility constraints. Furthermore, the tool life and the machine power constraints are loose, so the corresponding dual variables B_3 and B_4 are both equal to zero due to the complementary slackness conditions. Therefore, the constraints of GP-dual problem are reduced to the following system:

$$B_1 + B_2 = 1,$$

$$-B_1 + (\alpha_i - 1) \cdot B_2 + g \cdot B_5 = 0,$$

$$-B_1 + (\beta_i - 1) \cdot B_2 + h \cdot B_5 = 0.$$

The solution for this system can be stated explicitly as follows:

$$B_1 = 1 - B_2, \quad B_2 = \frac{g - h}{g \cdot \beta_i - h \cdot \alpha_i} \quad \text{and} \quad B_5 = \frac{\alpha_i - \beta_i}{h \cdot \alpha_i - g \cdot \beta_i},$$

where $g \cdot \beta_i - h \cdot \alpha_i \neq 0$, since $g < 0$, $\alpha_i > \beta_i > 1$ and $h > 0$.

g is always negative since increasing the feed rate increases the surface roughness [5]. Consequently $B_5 > 0$ and $0 \leqslant B_1, B_2 \leqslant 1$, so we verify dual feasibility of the solution. Therefore, the surface roughness constraint must be tight at optimality.

References

[1] M.S. Akturk, S. Avci, Tool allocation and machining conditions optimization for CNC machines, European Journal of Operational Research 94 (1996) 335–348.
[2] M.S. Akturk, S. Onen, Joint lot sizing and tool management in a CNC environment, Computers in Industry 40 (1999) 61–75.
[3] T.C.E. Cheng, Z.L. Chen, L. Chung-Lun, Parallel machine scheduling with controllable processing times, IIE Transactions 28 (1996) 177–180.
[4] R.L. Daniels, B.J. Hoopes, J.B. Mazzola, Scheduling parallel manufacturing cells with resource flexibility, Management Science 42 (1996) 1260–1276.
[5] F.E. Gorczyca, Application of Metal Cutting Theory, Industrial Press, New York, 1987.
[6] S. Karabati, P. Kouvelis, Flow-line scheduling problem with controllable processing times, IIE Transactions 29 (1997) 1–14.
[7] B. Malakooti, J. Deviprasad, An interactive multiple criteria approach for parameter selection in metal cutting, Operations Research 37 (1989) 805–818.
[8] E. Nowicki, S. Zdrzalka, A survey of results for sequencing problems with controllable processing times, Discrete Applied Mathematics 26 (1990) 271–287.
[9] E. Nowicki, S. Zdrzalka, A bicriterion approach to preemptive scheduling of parallel machines with controllable job processing times, Discrete Applied Mathematics 63 (1995) 237–256.
[10] S.S. Panwalkar, R. Rajagopalan, Single-machine sequencing with controllable processing times, European Journal of Operational Research 59 (1992) 298–302.
[11] V. T'kindt, J.-C. Billaut, Multicriteria Scheduling: Theory, Models and Algorithms, Springer, Berlin, 2002.
[12] M.A. Trick, Scheduling multiple variable-speed machines, Operations Research 42 (1994) 234–248.
[13] R.G. Vickson, Choosing the job sequence and processing times to minimize processing plus flow cost on a single machine, Operations Research 28 (1980) 1155–1167.
[14] S. Zdrzalka, Scheduling jobs on a single machine with release dates, delivery times and controllable processing times: Worst case analysis, Operations Research Letters 10 (1991) 519–523.