2011 International Conference on Networking, Sensing and Control
Delft, the Netherlands, 11-13 April 2011

Scheduling with resource allocation and past-sequence-dependent setup times including maintenance

Zhanguo Zhu, Feng Chu, Linyan Sun, and Ming Liu

Abstract— **In this work, scheduling problems with resource allocation and past-sequence-dependent (p-s-d) setup times are considered. Under these settings, the actual job processing times are dependent on the amount of resource allocated and the setup times are proportionate to the length of the already processed jobs. The objective function is a combination of the total completion time and the total resource consumption. In addition, the machine may need maintenance to improve its production efficiency, so the maintenance is also integrated. The optimal job sequence, the optimal amount of resource allocated to each job, and the optimal maintenance position are determined jointly. It is shown that the problem under study is polynomial time solvable.**

I. INTRODUCTION

Scheduling problems with explicit considerations of setup times have been studied for more than fifty years in many industries of manufacturing and service [1]. Two classical types of setup times, sequence independent (Graves and Lee 1999 [8]) and sequence dependent (Yalaoui and Chu 2003 [22]), are usually discussed based on whether the setup time depends solely on the current job to be processed or depends on both of the current and the last preceding jobs. Different from the former cases, motivated by some phenomenons in high tech manufacturing, Koulamas and Kyparisis (2008) [12] proposed a new form of setup times which depend on all jobs scheduled. This new form called past-sequence-dependent (p-s-d) setup times has attracted much attention in very recent years. Kuo and Yang (2007) [13] studied single machine scheduling problem with p-s-d setup times and job-independent/dependent learning effect for some different objectives and provided polynomial time algorithms for each objective function. Later, Wang (2008) [21] extended their results by considering scheduling problem with the time-dependent learning effect and p-sd setup times. Biskup and Herrmann (2008) [2] analyzed scheduling problems with individual/common due date and p-s-d setup times. Cheng et al. (2010) [5] investigated a single machine scheduling problem where job deterioration and learning effect, and p-s-d setup times are considered concurrently.

Z. Zhu is with School of Management, Xi'an Jiaotong University, 710049, China; Université de Technologie de Troyes, Institut Charles Delaunay, FRE CNRS 2848, Laboratoire d'optimisation des systèmes industriels (LOSI), 12 rue Marie Curie - BP 2060, 10010 Troyes Cedex, France zhanguo.zhu@utt.fr

F. Chu is with Laboratoire d'informatique, biologie intégrative et systèmes complexes (IBISC), FRE CNRS 3190, Université d'Evry Val d'Essonne, 40 Rue du Pelvoux, CE1455 Courcouronnes, 91020 Evry Cedex, France feng.chu@ibisc.univ-evry.fr

L. Sun is with School of Management, State Key Laboratory for Mechanical Manufacturing Systems Engineering, The Key Lab of the Ministry of Education for process control and Efficiency Engineering, Xian Jiaotong University, 710049, China. lysun@mail.xjtu.edu.cn

M. Liu is with School of Economics & Management, Tongji University, Shanghai, 200092, P.R. China minyivg@gmail.com

Resource allocation plays an important role in many scheduling systems due to the control of actual job processing times through the allocation of a finite amount of related resource such as money, energy, overtime, manpower, and so on. The resource function used to describe the effects of resource allocation on job processing times usually consists of two different types in the literature (Daniels and Sarin 1989 [6], Monma (1990) et al. [17], Cheng et al. (1998) [4], Kaspi and Shabtay (2006) [10], and Koulamas et al. (2010) [11]). The linear and convex functions can be usually represented as follows: $p_j^A(u_j) = p_j - b_j u_j, 0 \leq u_j \leq \bar{u}_j < \frac{p_j}{b_j}$, where $j = 1, 2, ..., n$, $p_j^A(u_j)$ is the actual processing time of job j, p_j is the nominal processing time of job j, $b_j > 0$ is the compression rate of job j, and $p_j^A(u_j) = (\frac{p_j}{u_j})^k$, for $j = 1, 2, ..., n$, where k is a positive constant. For details on them, the readers may refer to the survey by Shabtay and Steiner (2007) [20]. In a very recent paper, Leyvand et al. (2010) [15] proposed a more general resource function which can vary even between or even within jobs than these represented above to describe more complex applications. They developed polynomial time algorithms for all studied scheduling problems. In this paper, this more general resource function will be adopted.

Scheduling with a rate-modifying activity that can be considered as a special case of maintenance is also a popular topic in recent years. A rate-modifying activity which can improve the production rate of a machine was first introduced by Lee and Leon (2001) [14]. Later, researchers extended scheduling problems with rate-modifying activity to a few different environments (He et al. (2005) [9], Mosheiov and Oron (2006) [18], Mosheiov and Sarig (2009) [19], Lodree and Geiger (2010) [16]).

It is necessary to note that no paper published deals with scheduling problems with p-s-d setup times in the context of resource allocation or maintenance (rate modifying activity). In this work, we consider scheduling problems with resource allocation, p-s-d setup times, maintenance simultaneously first. The objective function is a combination of the total completion time and the total resource cost.

II. Problem description

The problem studied can be described formally as follows. There is given a set of n jobs available for processing at time 0 on a single machine and job preemption is not allowed. $J_{[j]}$ denotes the job occupying the j job position in a sequence $\pi = J_{[1]}, J_{[2]}, ..., J_{[j]}, ..., J_{[n]}$. The actual job processing time is also affected by the amount of resource allocated. In this paper we adopt the resource function proposed by Leyvand et al. (2010) [15] to describe the relation between job processing time and the amount of resource allocated.

$$p_j^A = p_j(u_j), \quad (1)$$

where p_j^A denotes the actual processing time of job j, $u_j \geq 0$ is the amount of resource allocated to job j, and $p_j(u_j)$ is a bounded, differentiable convex function of u_j satisfying several properties described in the next section.

Before processing each job J_j, a p-s-d setup time is involved. Following the assumption in Koulamas and Kyparisis (2008)[12], a p-s-d setup time $s_{[j]}$ can be computed as

$$s_{[j]} = \alpha \sum_{l=1}^{j-1} p_{[l]}^A = \alpha \sum_{l=1}^{j-1} p_{[l]}(u_{[l]}), j = 2, ..., n, s_{[1]} = 0, \quad (2)$$

where $\alpha \geq 0$ is a constant.

When a maintenance, the position of which is i if it is scheduled immediately after the completion of job i, is introduced, the actual processing time and p-s-d setup time can be represented as Equations (3)-(6):

$$p_{[j]}^A = \begin{cases} p_{[j]}(u_{[j]}), & j = 1, 2, ..., i, \quad (3) \\ \beta_{[j]} p_{[j]}(u_{[j]}), & j = i+1, i+2, ..., n, \quad (4) \end{cases}$$

where $0 < \beta_j < 1$ is the improvement rate after the maintenance for job j. Equations (3) denote the actual processing times of jobs scheduled before the maintenance, while Equations (4) mean the actual processing times of jobs scheduled after the maintenance. The maintenance duration is φ. For any sequence π, $C_j = C_j(\pi)$ is use to denote the completion time of job $J_{[j]}$, $TC = \sum_{j=1}^n C_j$. The cost function of this problem is the combination of the total completion time and the total resource cost.

$$Z(\pi, u) = \mu_1 TC + \mu_2 \sum_{j=1}^n E_j u_j, \quad (7)$$

where μ_1, and μ_2 are given non-negative constants, and E_j is the unit resource cost.

Following the three-field notation of Graham et al. (1979)[7] for scheduling problem, the problems under study can be denoted as $1|RA, s_{psd}, M|\mu_1 TC + \mu_2 \sum_{j=1}^n E_j u_j$, where RA means "resource allocation", s_{psd} denotes "past-sequence-dependent setup times", and M denotes "maintenance".

III. Preliminary results

The following are the not very restrictive properties of function $p_j(u_j)$ proposed by Leyvand et al. (2010) [15].

Property 1. *$dp_j(u_j)/du_j \leq 0$ for $u_j \in [u_j^{min}, u_j^{max}]$, and u_j^{min} and u_j^{max} are the lower and upper bound of the amount of resource allocated to job j. The derivative $dp_j(u_j)/du_j$ can also be denoted as $p'_j(u_j)$ for convenience.*

Property 2. *$d^2 p_j(u_j)/d(u_j)^2 \geq 0$ for $u_j \in [u_j^{min}, u_j^{max}]$.*

Property 3. *There exists only one point u_j such that $p'_j(u_j) = m$, for $m \in [p'_j(u_j^{min}), p'_j(u_j^{max})]$, and it can be determined in constant time.*

The introduction of the maintenance brings some new characteristics to the actual job processing times, p-s-d setup times, and the total completion time.

The p-s-d setup time and actual processing time for job $J_{[j]}$, $j = 1, 2, ..., i, i+1, ...$, are given by

$P'_1 = s_{[1]} + p_{[1]}^A = p_{[1]}^A$,
$P'_2 = s_{[2]} + p_{[2]}^A = \alpha p_{[1]}^A + p_{[2]}^A$,
$P'_3 = s_{[3]} + p_{[3]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A$,
$P'_4 = s_{[4]} + p_{[4]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A$,
...
$P'_i = s_{[i]} + p_{[i]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + p_{[i]}^A$,
$P'_{i+1} = s_{[i+1]} + p_{[i+1]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \beta_{[i+1]} p_{[i+1]}^A$,
$P'_{i+2} = s_{[i+2]} + p_{[i+2]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \alpha \beta_{[i+1]} p_{[i+1]}^A + \beta_{[i+2]} p_{[i+2]}^A$,
$P'_{i+3} = s_{[i+3]} + p_{[i+3]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \alpha \beta_{[i+1]} p_{[i+1]}^A + \alpha \beta_{[i+2]} p_{[i+2]}^A + \beta_{[i+3]} p_{[i+3]}^A$,
...
$P'_{[n]} = s_{[n]} + p_{[n]}^A = \alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \alpha \beta_{[i+1]} p_{[i+1]}^A + \alpha \beta_{[i+2]} p_{[i+2]}^A + \alpha \beta_{[i+3]} p_{[i+3]}^A + ... + \alpha \beta_{[n-1]} p_{[n-1]}^A + \beta_{[n]} p_{[n]}^A$.

The completion time of each job ($j = 1, 2, ..., i, i+1, ..., n$) can be denoted as

$C_{[1]} = p_{[1]}^A$,
$C_{[2]} = p_{[1]}^A + (\alpha p_{[1]}^A + p_{[2]}^A)$,
$C_{[3]} = p_{[1]}^A + (\alpha p_{[1]}^A + p_{[2]}^A) + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A)$,
$C_{[4]} = p_{[1]}^A + (\alpha p_{[1]}^A + p_{[2]}^A) + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A) + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A)$,
...
$C_{[i]} = p_{[1]}^A + (\alpha p_{[1]}^A + p_{[2]}^A) + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A) + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A) + ... + (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + p_{[i]}^A)$,

$C_{[i+1]} = p_{[1]}^A$
$+ (\alpha p_{[1]}^A + p_{[2]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A) + ...$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + p_{[i]}^A) + \varphi$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \beta_{[i+1]} p_{[i+1]}^A)$,

$C_{[i+2]} = p_{[1]}^A$
$+ (\alpha p_{[1]}^A + p_{[2]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A) + ...$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + p_{[i]}^A) + \varphi$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \beta_{[i+1]} p_{[i+1]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + ... + \alpha p_{[i-1]}^A + \alpha p_{[i]}^A + \alpha \beta_{[i+1]} p_{[i+1]}^A + \beta_{[i+2]} p_{[i+2]}^A)$,

...
$C_{[n]} = p_{[1]}^A$
$+ (\alpha p_{[1]}^A + p_{[2]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + p_{[3]}^A)$
$+ (\alpha p_{[1]}^A + \alpha p_{[2]}^A + \alpha p_{[3]}^A + p_{[4]}^A) + ...$

$$s_{[j]} = \alpha \sum_{l=1}^{j-1} p^A_{[l]} = \begin{cases} \alpha \sum_{l=1}^{j-1} p_{[l]}(u_{[l]}), & j = 1, 2, ..., i, \quad (5) \\ \alpha \sum_{l=1}^{i} p_{[l]}(u_{[l]}) + \alpha \sum_{l=i+1}^{j-1} \beta_{[l]} p_{[l]}(u_{[l]}), & j = i+1, i+2, ..., n, \quad (6) \end{cases}$$

$+ (\alpha p^A_{[1]} + \alpha p^A_{[2]} + \alpha p^A_{[3]} + ... + \alpha p^A_{[i-1]} + p^A_{[i]}) + \varphi$
$+ (\alpha p^A_{[1]} + \alpha p^A_{[2]} + \alpha p^A_{[3]} + ... + \alpha p^A_{[i-1]} + \alpha p^A_{[i]} + \beta_{[i+1]} p^A_{[i+1]})$
$+ (\alpha p^A_{[1]} + \alpha p^A_{[2]} + \alpha p^A_{[3]} + ... + \alpha p^A_{[i-1]} + \alpha p^A_{[i]} + \alpha\beta_{[i+1]} p^A_{[i+1]} + \beta_{[i+2]} p^A_{[i+2]})$
$+ (\alpha p^A_{[1]} + \alpha p^A_{[2]} + \alpha p^A_{[3]} + ... + \alpha p^A_{[i-1]} + \alpha p^A_{[i]} + \alpha\beta_{[i+1]} p^A_{i+1} + \alpha\beta_{[i+2]} p^A_{[i+2]} + \beta_{[i+3]} p^A_{[i+3]}) + ...$
$+ (\alpha p^A_{[1]} + \alpha p^A_{[2]} + \alpha p^A_{[3]} + ... + \alpha p^A_{[i-1]} + \alpha p^A_{[i]} + \alpha\beta_{[i+1]} p^A_{[i+1]} + \alpha\beta_{[i+2]} p^A_{[i+2]} + \alpha\beta_{[i+3]} p^A_{[i+3]} + ... + \alpha\beta_{[n-1]} p^A_{[n-1]} + \beta_{[n]} p^A_{[n]})$.

The total completion time is obtained based on the induction

$TC = \sum_{j=1}^{i} \left((n-j+1) + \alpha \frac{(n-j)(n-j+1)}{2} \right) p_{[j]}(u_{[j]}) + \sum_{j=i+1}^{n} \left((n-j+1) + \alpha \frac{(n-j)(n-j+1)}{2} \right) \beta_{[j]} p_{[j]}(u_{[j]}) + (n-i)\varphi$.

IV. Optimal analysis

Substituting (3), (4), (5), and (6) into (7), the total cost function has the following new form as Equations (8):

Proof: $[of lemma 1]$ For any given sequence π, we take the derivative of the cost function given by equation (8) with respect to $u^*_{[j]}$ as the method in Leyvand et al. (2010) [15]. Let the derivative be equal to 0, and solve it. For $j = 1, 2, ..., n$,

When $j = 1, 2, ..., i$,
$\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} = \frac{d\left(w_j p_{[j]}(u_{[j]}) + \mu_2 E_{[j]} u_{[j]}\right)}{du_{[j]}} = w_j p'_{[j]}(u_{[j]}) + \mu_2 E_{[j]}$.

Let $w_j p'_{[j]}(u_{[j]}) + \mu_2 E_{[j]} = 0$, we obtain $p'_{[j]}(u_{[j]}) = \frac{-\mu_2 E_{[j]}}{w_j}$, and the solution of this equation is $\hat{u}_{[j]}$.

If $p'_{[j]}(u^{min}_{[j]}) \geq \frac{-\mu_2 E_{[j]}}{w_j}$, we obtain $\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} \geq 0$, and $Z_{[j]}(\pi, u)$ is an increasing function. For $u_{[j]} \in [u^{min}_{[j]}, u^{max}_{[j]}]$, the value $u^*_{[j]} = u^{min}_{[j]}$ leads to the minimal value of $Z_{[j]}(\pi, u)$.

If $p'_{[j]}(u^{max}_{[j]}) \leq \frac{-\mu_2 E_{[j]}}{w_j}$, we obtain $\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} \leq 0$, and $Z_{[j]}(\pi, u)$ is an decreasing function. For $u_{[j]} \in [u^{min}_{[j]}, u^{max}_{[j]}]$, the value $u^*_{[j]} = u^{max}_{[j]}$ leads to the minimal value of $Z_{[j]}(\pi, u)$.

If $p'_{[j]}(u^{min}_{[j]}) < \frac{-\mu_2 E_{[j]}}{w_j} < p'_{[j]}(u^{max}_{[j]})$, considering the properties 1-3, it is easy to conclude that $u^*_{[j]} = \hat{u}_{[j]}$ is the optimal value to minimize $Z_{[j]}(\pi, u)$.

When $j = i+1, i+2, ..., n$,
$\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} = \frac{d\left(w_j \beta_{[j]} p_{[j]}(u_{[j]}) + \mu_2 E_{[j]} u_{[j]}\right)}{du_{[j]}} = w_j \beta_{[j]} p'_{[j]}(u_{[j]}) + \mu_2 E_{[j]}$.

Let $w_j \beta_{[j]} p'_{[j]}(u_{[j]}) + \mu_2 E_{[j]} = 0$, we obtain $p'_{[j]}(u_{[j]}) = \frac{-\mu_2 E_{[j]}}{w_j \beta_{[j]}}$, and the solution of this equation is $\hat{u}_{[j]}$.

If $p'_{[j]}(u^{min}_{[j]}) \geq \frac{-\mu_2 E_{[j]}}{w_j \beta_{[j]}}$, we obtain $\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} \geq 0$, and $Z_{[j]}(\pi, u)$ is an increasing function. For $u_{[j]} \in [u^{min}_{[j]}, u^{max}_{[j]}]$, the value $u^*_{[j]} = u^{min}_{[j]}$ leads to the minimal value of $Z_{[j]}(\pi, u)$.

If $p'_{[j]}(u^{max}_{[j]}) \leq \frac{-\mu_2 E_{[j]}}{w_j \beta_{[j]}}$, we obtain $\frac{dZ_{[j]}(\pi, u)}{du_{[j]}} \leq 0$, and $Z_{[j]}(\pi, u)$ is an decreasing function. For $u_{[j]} \in [u^{min}_{[j]}, u^{max}_{[j]}]$, the value $u^*_{[j]} = u^{max}_{[j]}$ leads to the minimal value of $Z_{[j]}(\pi, u)$.

If $p'_{[j]}(u^{min}_{[j]}) < \frac{-\mu_2 E_{[j]}}{w_j \beta_{[j]}} < p'_{[j]}(u^{max}_{[j]})$, considering the properties 1-3, it is easy to conclude that $u^*_{[j]} = \hat{u}_{[j]}$ is the optimal value to minimize $Z_{[j]}(\pi, u)$.

■

Set G_{jr} is the cost incurred by job j scheduled in position r and $y_{jr} = 1$ if job j scheduled in position r, otherwise $y_{jr} = 0$, for $j = 1, 2, ..., n$ and $r = 1, 2, ..., n$.

The problem studied can be represented as the following linear programming problem.

(P1) $min \sum_{i=1}^{n} \sum_{r=1}^{n} G_{ir} y_{ir} + \mu_1 (n-i)\varphi$

subject to
$\sum_{r=1}^{n} y_{ir} = 1, i = 1, 2, ..., n,$

$\sum_{i=1}^{n} y_{ir} = 1, r = 1, 2, ..., n,$

$y_{ir} = 0$ or $1, i = 1, 2, ..., n; r = 1, 2, ..., n.$

The constraints ensure each job is scheduled in one position and each position is taken by only one job.

If the position of the maintenance is given, the last part of the objective function is a constant. The above programming problem is equivalent to solve the following linear assignment problem.

(P2) $min \sum_{i=1}^{n} \sum_{r=1}^{n} G_{ir} y_{ir}$

subject to
$\sum_{r=1}^{n} y_{ir} = 1, i = 1, 2, ..., n,$

$\sum_{i=1}^{n} y_{ir} = 1, r = 1, 2, ..., n,$

$$Z(\pi,u)=\mu_1[\sum_{j=1}^{i}\left((n-j+1)+\alpha\frac{(n-j)(n-j+1)}{2}\right)p_{[j]}(u_{[j]})+\sum_{j=i+1}^{n}\left((n-j+1)+\alpha\frac{(n-j)(n-j+1)}{2}\right)\beta_{[j]}p_{[j]}(u_{[j]})$$
$$+(n-i)\varphi]$$
$$+\sum_{j=1}^{n}\mu_2E_{[j]}u_{[j]} \quad (8)$$
$$=\sum_{j=1}^{i}w_jp_{[j]}(u_{[j]})+\sum_{j=i+1}^{n}w_j\beta_{[j]}p_{[j]}(u_{[j]})+\mu_1(n-i)\varphi+\sum_{j=1}^{n}\mu_2E_{[j]}u_{[j]}$$

let $w_j=\mu_1(n-j+1)\left(1+\alpha\frac{(n-j)}{2}\right)$

Lemma 1: As a function of any given job sequence, the optimal resource allocation $u_j^*(\pi)$ for job j is

$$u_{[j]}^*=\begin{cases} u_{[j]}^{min}, & \text{if } p'_{[j]}(u_{[j]}^{min})\geq\frac{-\mu_2E_{[j]}}{w_j}, for j=1,2,...,i, & (9)\\ u_{[j]}^{min}, & \text{if } p'_{[j]}(u_{[j]}^{min})\geq\frac{-\mu_2E_{[j]}}{w_j\beta_{[j]}}, for j=i+1,i+2,...,n, & (10)\\ \hat{u}_{[j]}, & \text{if } p'_{[j]}(u_{[j]}^{min})<\frac{-\mu_2E_{[j]}}{w_j}<p'_{[j]}(u_{[j]}^{max}), for j=1,2,...,i, & (11)\\ \hat{u}_{[j]}, & \text{if } p'_{[j]}(u_{[j]}^{min})<\frac{-\mu_2E_{[j]}}{w_j\beta_{[j]}}<p'_{[j]}(u_{[j]}^{max}), for j=i+1,i+2,...,n, & (12)\\ u_{[j]}^{max}, & \text{if } p'_{[j]}(u_{[j]}^{max})\leq\frac{-\mu_2E_{[j]}}{w_j}, for j=1,2,...,i. & (13)\\ u_{[j]}^{max}, & \text{if } p'_{[j]}(u_{[j]}^{max})\leq\frac{-\mu_2E_{[j]}}{w_j\beta_{[j]}}, for j=i+1,i+2,...,n. & (14)\end{cases}$$

$$G_{jr}=\begin{cases} w_rp_j(u_j^{min})+E_ju_j^{min}, & \text{if } p'_j(u_j^{min})\geq\frac{-\mu_2E_j}{w_r}, \text{ for } r=1,2,...,i, & (15)\\ w_rp_j(\hat{u}_j)+E_j\hat{u}_j, & \text{if } p'_j(u_j^{min})<\frac{-\mu_2E_j}{w_r}<p'_j(u_j^{max}), \text{ for } r=1,2,...,i & (16)\\ w_rp_j(u_j^{max})+E_ju_j^{max}, & \text{if } p'_j(u_j^{max})\leq\frac{-\mu_2E_j}{w_r}, \text{ for } r=1,2,...,i. & (17)\\ w_rp_j(u_j^{min})+E_ju_j^{min}, & \text{if } p'_j(u_j^{min})\geq\frac{-\mu_2E_j}{w_r\beta_{[j]}}, \text{ for } r=i+1,i+2,...,n, & (18)\\ w_rp_j(\hat{u}_j)+E_j\hat{u}_j, & \text{if } p'_j(u_j^{min})<\frac{-\mu_2E_j}{w_r\beta_{[j]}}<p'_j(u_j^{max}), \text{ for } r=i+1,i+2,...,n & (19)\\ w_rp_j(u_j^{max})+E_ju_j^{max}, & \text{if } p'_j(u_j^{max})\leq\frac{-\mu_2E_j}{w_r\beta_{[j]}}, \text{ for } r=i+1,i+2,...,n. & (20)\end{cases}$$

$y_{ir}=0$ or $1, i=1,2,...,n; r=1,2,...,n.$

The preceding analysis for the $1|RA,s_{psd},M|\mu_1TC+\mu_2\sum_{j=1}^{n}E_ju_j$ problem can be summarized with the following optimization algorithm.

Algorithm 1.

Step 1. Set $i=1$.
Step 2. For $j=1,2,...,n$ and $r=1,2,...,n$, calculate all values G_{jr} with equations (15), (16), (17), (18), (19), and (20).
Step 3. Solve the linear assignment problem (P2) to determine a local optimal job sequence π_l^* and record the total cost.
Step 4. Set $i=i+1$. If $i<n$, then go to Step 2. Otherwise go to Step 5.

Step 5. Order the total cost of all local optimal job sequence π_l^*, and the one with the minimal total cost is denoted as the global optimal job sequence (π^*).

Step 6. Calculate the optimal amount of resources allocated $u^*_{[j]}(\pi^*)$ by equations (9)-(14).

Step 7. Obtain the actual job processing times and p-s-d setup times with equations (3)- (4) and (5)- (6).

Theorem 1: For the $1|RA, s_{psd}, M|\mu_1 TC + \mu_2 \sum_{j=1}^{n} E_j u_j$ problem, the optimal job sequence π^* and resource allocation $u^*(\pi^*)$ can be obtained in $O(n^4)$ time.

Proof:

The position of the maintenance may be $1, 2, ..., n$, so the overall complexity of Algorithm 2 is $O(n^4)$ due to the complexity of Step 3 ($O(n^3)$)

■

V. Conclusions

This paper aims to study scheduling problem integrating general convex resource allocation function, p-s-d setup times, and maintenance for the objective of a combination of the total completion time. The problem is proved to be polynomially solvable and an algorithm is also proposed . The algorithm solves the problem in $O(n^4)$ time. Scheduling problems with other objectives, such as due-date based objectives, can be solved with similar methods. Multi-machine and multi-agent scheduling problems in the context of p-s-d setup times and resource allocation may be our future work.

VI. ACKNOWLEDGMENTS

This work is partially supported by NSF of China under Grants (70701029) and State-Sponsored Study Abroad Program of China.

References

[1] Allahverdi, A., Ng, C.T., Cheng, T.C.E., & Kovalyov, M.Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187, 985-1032.

[2] Biskup, D., & Herrmann, J. (2008). Single-machine scheduling against due dates with past-sequence-dependent setup times. *European Journal of Operational Research*, 191, 587-592.

[3] Brucker, P. (2001). Scheduling Algorithms, third ed., Springer, Berlin.

[4] Cheng, T.C.E., Janiak, A., & Kovalyov, M.Y. (1998). Bicriterion single machine scheduling with resource dependent processing times. *SIAM Journal on Optimization*, 8, 617-630.

[5] Cheng, T.C.E., Lee, W.C., & Wu, C.C., (2010). Scheduling problems with deteriorating jobs and learning effects including proportional setup times. *Computers & Industrial Engineering*, 58, 326-331.

[6] Daniels, R.L., & Sarin, R.K. (1989). Single machine scheduling with controllable processing times and number of jobs tardy. *Operations Research*, 37, 981-984.

[7] Graham, R.L., Lawler, E.L., Lenstra, J.K., & Kan, A.R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics*, 5, 287-326.

[8] Graves, G.H., & Lee, C.Y. (1999). Scheduling maintenance and semiresumable jobs on a single machine. *Naval Research Logistics*, 46, 845-863.

[9] He, Y., Ji, M., & Cheng, T.C.E. (2005). Single machine scheduling with a restricted rate-modifying activity. *Naval Research Logistics*, 52, 361-369.

[10] Kaspi, M., & Shabtay, D. (2006). A bicriterion approach to time/cost trade-offs in scheduling with convex resource-dependent job processing times and release dates. *Computers & Operations Research*, 33, 3015-3033.

[11] Koulamas, C., Gupta, S., & Kyparisis, G.J. (2010). A unified analysis for the single-machine scheduling problem with controllable and non-controllable variable job processing times. *European Journal of Operational Research*, 205, 479-482.

[12] Koulamas, C., & Kyparisis, G. J. (2008). Single-machine scheduling problems with past-sequence-dependent setup times. *European Journal of Operational Research*, 187, 68-72.

[13] Kuo, W.-H., & Yang, D.-L. (2007). Single-machine scheduling with past-sequence-dependent setup and learning effects. *Information Processing Letters*, 102, 22-26.

[14] Lee, C.-L., & Leon, V.-J. (2001). Machine scheduling with a rate modifying activity. *European Journal of Operational Research*, 129, 119-128.

[15] Leyvand, Y., Shabtay, D., & Steiner, G. (2010). A unified approach for scheduling with convex resource consumption functions using positional penalties. *European Journal of Operational Research*, 206, 301-312.

[16] Lodree Jr., E.J., & Geiger, C.D. (2010). A note on the optimal sequence position for a rate-modifying activity under simple linear deterioration. *European Journal of Operational Research*, 201, 644-648.

[17] Monma, C.L., Schrijver, A., Todd, M.J., & Wei, V.K. (1990). Convex resource allocation problems on directed acyclic graphs: Duality, complexity, special cases and extensions. *Mathematics of Operations Research*, 15, 736-748.

[18] Mosheiov, G., & Oron, D. (2006). Due-date assignment and maintenance activity scheduling problem. *Mathematical and Computer Modelling*, 44, 1053-1057.

[19] Mosheiov, G., & Sarig, A. (2009). Scheduling a maintenance activity and due-window assignment on a single machine. *Computers & Operations Research*, 36, 2541-2545.

[20] Shabtay, D., & Steiner, G. (2007). A survey of scheduling with controllable processing times. *Discrete Applied Mathematics*, 155, 1643-1666.

[21] Wang, J.-B. (2008). Single-machine scheduling with past-sequence dependent setup times and time-dependent learning effect. *Computers & Industrial Engineering*, 55, 584-591.

[22] Yalaoui, F., & Chu, C., (2003). An efficient heuristic approach for parallel machine scheduling with job splitting and sequence dependent setup times. *IIE Transactions*, 35, 183-190.