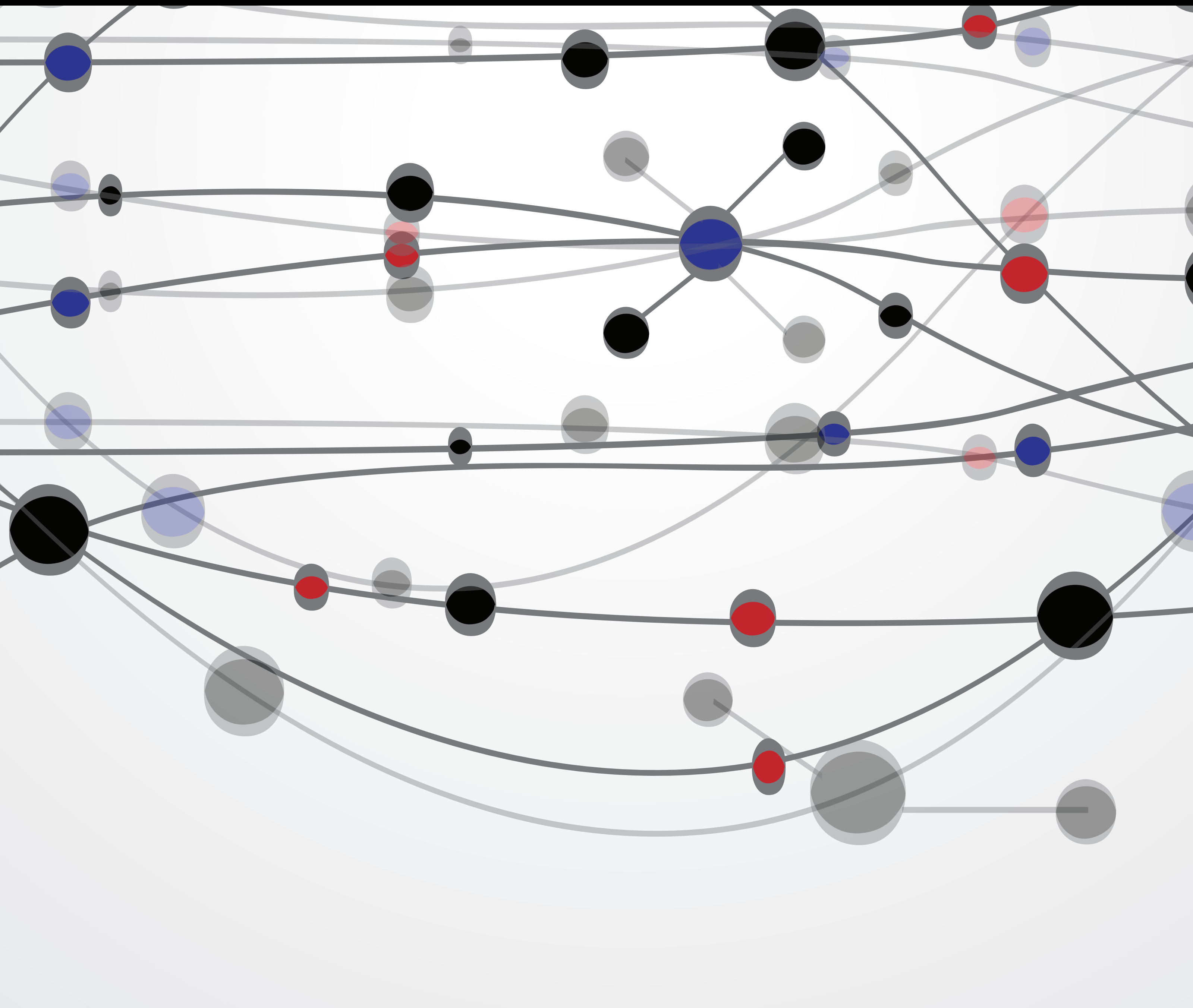
The Scientific World Journal

Recent Advances in Combinatorial Optimization

Guest Editors: Dehua Xu, Dar-Li Yang, Ming Liu, Feng Chu, and Imed Kacem



Recent Advances in Combinatorial Optimization

The Scientific World Journal

Recent Advances in Combinatorial Optimization

Guest Editors: Dehua Xu, Dar-Li Yang, Ming Liu, Feng Chu, and Imed Kacem

Contents

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 628265, 1 page
http://dx.doi.org/10.1155/2015/628265

Editorial

Recent Advances in Combinatorial Optimization

Dehua Xu,[1] Dar-Li Yang,[2] Ming Liu,[3] Feng Chu,[4] and Imed Kacem[5]

[1] *College of Science, East China Institute of Technology, Nanchang, Jiangxi 330013, China*
[2] *Department of Information Management, National Formosa University, Yun-Lin 632, Taiwan*
[3] *School of Economics & Management, Tongji University, Shanghai 200092, China*
[4] *Laboratoire IBISC, Université d'Evry-Val d'Essone, 40 rue Pelvoux, 91001 Evry, France*
[5] *LCOMS EA7306, Université de Lorraine, Ile du Saulcy, 57000 Metz, France*

Correspondence should be addressed to Dehua Xu; dhxu@ecit.cn

Received 8 March 2015; Accepted 8 March 2015

Combinatorial optimization is one of the most active branches of operations research. The essence of a combinatorial optimization problem is to find optimal solutions or near optimal solutions from a finite set of feasible solutions. In such problems, the size of feasible solution space usually increases exponentially with regard to the increase in the size of the input parameters. This issue, which has an acceptance rate of less than 30%, compiles six exciting papers.

In the paper "Single Machine Scheduling and Due Date Assignment with Past-Sequence-Dependent Setup Time and Position-Dependent Processing Time," by C.-L. Zhao et al., the authors study several objective functions including total earliness, the weighted number of tardy jobs, and the cost of due date assignment. They provide polynomial time algorithms for all the considered problems. In the paper "Scheduling Jobs and a Variable Maintenance on a Single Machine with Common Due-Date Assignment," by L. Wan, the author derives some properties on an optimal solution for the problem and proposes an optimal polynomial time algorithm for a special case with identical jobs. In the paper "Due-Window Assignment Scheduling with Variable Job Processing Times," by Y.-B. Wu and P. Ji, the authors prove that the problem can be solved in polynomial time. In the paper "Some Single-Machine Scheduling Problems with Learning Effects and Two Competing Agents," by H. Li et al., the authors investigate three problems arising from different combinations of the objectives of the two agents. They provide a polynomial time algorithm for one problem and two polynomial time algorithms for the other two problems under certain agreeable conditions. In the paper "An Order Insertion Scheduling Model of Logistics Service Supply Chain Considering Capacity and Time Factors," by W. Liu et al., the authors analyze order similarity coefficient and order insertion operation process and establish an order insertion scheduling model of LSSC with service capacity and time factors considerations. In the paper "Cooperative Fuzzy Games Approach to Setting Target Levels of ECs in Quality Function Deployment," by Z. Yang et al., the authors develop a cooperative game framework combined with fuzzy set theory to determine the target levels of the engineering characteristics in quality function deployment.

The papers published in this issue contain some interesting, creative, and valuable results and ideas. We do believe that all these papers will motivate further scientific research in combinatorial optimization and related areas.

Dehua Xu
Dar-Li Yang
Ming Liu
Feng Chu
Imed Kacem

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 740750, 7 pages
http://dx.doi.org/10.1155/2015/740750

Research Article

Due-Window Assignment Scheduling with Variable Job Processing Times

Yu-Bin Wu[1] and Ping Ji[2]

[1] *School of Science, Shenyang Aerospace University, Shenyang 110136, China*
[2] *Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*

Correspondence should be addressed to Yu-Bin Wu; wuyubin79@163.com

Received 18 August 2014; Accepted 11 September 2014

Academic Editor: Dehua Xu

We consider a common due-window assignment scheduling problem jobs with variable job processing times on a single machine, where the processing time of a job is a function of its position in a sequence (i.e., learning effect) or its starting time (i.e., deteriorating effect). The problem is to determine the optimal due-windows, and the processing sequence simultaneously to minimize a cost function includes earliness, tardiness, the window location, window size, and weighted number of tardy jobs. We prove that the problem can be solved in polynomial time.

1. Introduction

In most scheduling studies, job processing times are treated as constant numbers; however, in many practical situations, job processing times are affected by the learning effects and/or deteriorating (aging) effects. Learning effects and deteriorating (aging) effects are important for production and scheduling problems. For details on this line of the scheduling problems with learning effects (deteriorating effects), the reader is referred to a comprehensive survey by Biskup [1] (Gawiejnowicz [2]). Rudek [3] considered single machine scheduling problems with position-dependent job processing times (i.e., learning and aging effects). For the following objectives, the makespan with release dates, the maximum lateness, and the number of late jobs, they gave some results. J.-B. Wang and M.-Z. Wang [4] and Sun et al. [5] considered flow shop scheduling problems with general position-dependent learning effects. For some regular objective functions, they proposed heuristics. Sun et al. [6] considered flow shop scheduling problems with three special position-dependent learning effects. For the total weighted completion time minimization problem, they proposed heuristics. Lu et al. [7] considered single machine scheduling problems with learning effects and controllable processing times. For two due date assignment methods, they presented a polynomial-time optimization algorithm to minimize a multiobjective cost function.

J.-B. Wang and M.-Z. Wang [8] considered common due-window single machine scheduling with learning effects and controllable processing times. For a mule-objective cost function, they presented a polynomial-time optimization algorithm. J.-B. Wang and M.-Z. Wang [9] considered single machine scheduling problems with nonlinear deterioration. They showed that the makespan minimization problem can be solved in polynomial time. J.-B. Wang and M.-Z. Wang [10] considered three-machine flow shop scheduling with deteriorating jobs. For the makespan minimization problem, they proposed a branch-and-bound algorithm and two heuristic algorithms. X.-R Wang and J.-J. Wang [11] considered single machine scheduling problems with deteriorating jobs and convex resource dependent processing times. Xu et al. [12] considered single machine group scheduling with proportional linear deterioration and ready times. For the makespan minimization problem, they gave some results. Cheng et al. [13] considered a single machine common due-window assignment scheduling problem with deteriorating jobs. For a deteriorating maintenance activity, they provided polynomial-time solutions for a multiobjective cost. Yang et al. [14] considered a single machine multiple common due dates assignment resource allocation scheduling problems with general position-dependent deterioration effect. For a multiobjective cost, they proved that the problems can be solved in polynomial time, respectively. Liu et al. [15] considered single-machine common due-window assignment scheduling problem with deteriorating jobs. If the width of the common due-window is a given constant, they proved a mule-objective function cost problem can be solved in polynomial time. J.-B. Wang and C. Wang [16] and Wang et al. [17] considered due-window assignment scheduling problems with learning effects and deteriorating jobs at the same time.

The recent paper Li et al. [18] addresses single machine scheduling problem with deteriorating jobs. For common due date assignment (CON) and common flow allowance (i.e., all jobs have slack due date (SLK)) due date assignment methods, they showed that a multiobjective minimization problem can be solved in polynomial time, respectively. In this research, we continue the work of Li et al. [18] but focus on the common due-window assignment (CONW) scheduling problem (Yin et al. [19]). Under the learning effect and deteriorating jobs models, we prove that the CONW due-window assignment scheduling is solvable in polynomial time, respectively.

2. Problem Formulation

The following notations will be used throughout the paper:

J_j: Job j

J: Set of jobs (i.e., $J = \{J_1, J_2, \ldots, J_n\}$)

C_j: Completion time of job J_j

d_1: Earliest due date

D: Common due-window size

d_2: Latest due date $= d_1 + D$

E_j: Earliness of $J_j = \max\{0, d_1 - C_j\}$

T_j: Tardiness of $J_j = \max\{0, C_j - d_2\}$

E: Set of earliest jobs $= \{J_j \mid C_j < d_1\}$

T: Set of tardy jobs $= \{J_j \mid C_j > d_2\}$

$\overline{D}$: Set of on time jobs (i.e., $\overline{D} = J \setminus (E \cup T)$)

m: Number of set $\overline{D}$ jobs (i.e., $m = |\overline{D}|$)

γ_j: The penalty weight if J_j is tardy (i.e., $J_j \in T$)

$F(d_1, D, \pi) = \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$: The total cost function, where $\alpha > 0$, $\beta > 0$, and $\theta > 0$ are the unit due-window starting time, due-window size, and earliness penalties, respectively.

Consider a nonpreemptive single machine setting. There are n independent jobs $J = \{J_1, J_2, \ldots, J_n\}$ available at zero and preemption is not allowed. Let P_j denote the actual processing time for job J_j. In this research, we consider the following models.

Job Time-Dependent Deterioration Effect Model (See Li et al. [18]). Consider

$$P_j = a_j + bt, \tag{1}$$

where $a_j, b > 0, t$ are the basic (normal) processing time of J_j, the deteriorating rate, and the starting time of J_j, respectively.

Job-Position-Dependent Learning Effect Model (See Biskup [20]). Consider

$$P_j = a_j r^a, \tag{2}$$

where $a_j, a < 0, r$ are the basic (normal) processing time of J_j, the learning rate, and the position J_j in a processing sequence, respectively.

Our task of this paper is to determine the optimal earliest due date d_1, the common due-window size D, and a schedule π which minimizes the following objective function:

$$F(d_1, D, \pi) = \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j. \tag{3}$$

Then, using the common three-field notation introduced by Graham et al. [21], the corresponding scheduling problems are denoted by

$$\begin{gathered} 1 \mid P_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j, \\ 1 \mid P_j = a_j r^a \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j. \end{gathered} \tag{4}$$

3. Optimal Solutions

3.1. Job Time-Dependent Deterioration Effect Model

Lemma 1 (Li et al. [18]). *For a given schedule $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, if the starting time of the first job is 0, then $C_{[r]} = \sum_{j=1}^{r} a_{[j]}(1+b)^{r-j}$ and $\sum_{j=1}^{n} C_j = \sum_{j=1}^{n} a_{[j]} \sum_{i=0}^{n-j}(1+b)^i$.*

Lemma 2. *If $\alpha > \beta$, an optimal schedule exists in which the due-window starts at time zero.*

Proof. Suppose $\alpha > \beta$, and $d_1 > 0$; we shift X units of time to the left. The change in the total cost is given by $\Delta Z = -\alpha X + \beta X - \theta l X$, where l denotes the number of early jobs. Cleary, $\Delta Z < 0$. Therefore, a shift of d_1 (until $d_1 = 0$) can only decrease the total cost. □

Lemma 3. *An optimal schedule exists in which the due-window starting time (i.e., d_1), and the due-window completion time (i.e., d_2) coincide with job completion times, respectively.*

Proof. Suppose that there exists a schedule starting at time zero and containing jobs at the kth and the $(k+m)$th positions such that $C_k < d_1 < C_{k+1}, C_{k+m} < d_2 < C_{k+m+1}$.

When we shift d_2 to C_{k+m}, the change in the total cost is given by $-\beta(d_2 - C_{k+m})$.

When we shift d_1 to C_k, the change in the total cost is given by $(-\alpha + \beta + k\theta)(d_1 - C_k)$.

When we shift d_1 to C_{k+1}, the change in the total cost is given by $-(-\alpha + \beta + k\theta)(C_{k+1} - d_1)$.

Again, a shift of d_1 to C_k or to C_{k+1} does not increase the total cost.

Therefore, an optimal schedule exists such that both d_1 and d_2 coincide with job completion times. □

Lemma 4. *An optimal schedule exists in which the index of the job completed at the due-window starting time is* $k = \lceil(\beta - \alpha)/\theta\rceil$.

Proof. Using the classical small perturbation technique (see J.-B. Wang and C. Wang [16] and J.-B. Wang and M.-Z. Wang [8]), we measure the change in the total cost when moving d_1.

We shift d_1, X units of time to the left, and the effect of the total cost is

$$-\alpha X + \beta X - \theta (k - 1) X. \tag{5}$$

We shift d_1, X units of time to the right, and the effect of the total cost is

$$\alpha X - \beta X + \theta k X. \tag{6}$$

Both expressions (5) and (6) are clearly nonnegative due to the optimality of the original solution.

From $-\alpha X + \beta X - \theta(k-1)X \geq 0$ and $\alpha X - \beta X + \theta k X \geq 0$ we have $k \leq ((\beta - \alpha)/\theta) + 1$ and $k \geq (\beta - \alpha)/\theta$. And from the integrality of k, it follows that $k = \lceil(\beta - \alpha)/\theta\rceil$. □

Lemma 5. *For the problem* $1 \mid P_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$, *if the job sequence is* $\pi = (J_{[1]}, J_{[2]}, \ldots, J_{[n]})$ *and* $m = |\overline{D}|$, *then the objective function can be expressed as*

$$F(d_1, D, \pi, m) = \sum_{j=1}^{k+m} w_j a_{[j]} + \sum_{j=k+m+1}^{n} \gamma_{[j]}, \tag{7}$$

where

$$w_j = \begin{cases} \{(\alpha + k\theta) + \beta[(1+b)^m - 1]\}(1+b)^{k-j} & \\ \quad -\theta \sum_{i=0}^{k-j} (1+b)^i, & j = 1, 2, \ldots, k; \\ \beta (1+b)^{k+m-j}, & j = k+1, k+2, \ldots, k+m. \end{cases} \tag{8}$$

Proof. By Lemmas 1 and 3, we have

$$d_1 = C_{[k]} = \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j}, \tag{9}$$

$$\begin{aligned} D &= C_{[k+m]} - C_{[k]} \\ &= \sum_{j=1}^{k+m} a_{[j]} (1+b)^{k+m-j} - \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \\ &= \sum_{j=1}^{k} a_{[j]} (1+b)^{k+m-j} + \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} \\ &\quad - \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \\ &= \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} (1+b)^m + \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} \\ &\quad - \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \\ &= \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} [(1+b)^m - 1] \\ &\quad + \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j}, \end{aligned}$$

$$\begin{aligned} &F(d_1, D, \pi, m) \\ &\quad = \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j \\ &\quad = \alpha C_{[k]} + \beta \left\{ \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} [(1+b)^m - 1] \right. \\ &\qquad \left. + \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} \right\} \\ &\qquad + \theta \sum_{j=1}^{k} \left(C_{[k]} - C_{[j]}\right) + \sum_{J_j \in T} \gamma_{[j]} \\ &\quad = (\alpha + k\theta) C_{[k]} - \theta \sum_{j=1}^{k} C_{[j]} \\ &\qquad + \beta \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} [(1+b)^m - 1] \\ &\qquad + \beta \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} + \sum_{J_j \in T} \gamma_{[j]} \end{aligned}$$

$$
\begin{aligned}
&= (\alpha + k\theta) \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \\
&\quad + \beta \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \left[(1+b)^{m} - 1\right] \\
&\quad - \theta \sum_{j=1}^{k} a_{[j]} \sum_{i=0}^{k-j} (1+b)^{i} + \beta \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} \\
&\quad + \sum_{j=k+m+1}^{n} \gamma_{[j]} \\
&= \sum_{j=1}^{k} a_{[j]} (1+b)^{k-j} \left\{(\alpha + k\theta) + \beta\left[(1+b)^{m} - 1\right]\right\} \\
&\quad - \theta \sum_{j=1}^{k} a_{[j]} \sum_{i=0}^{k-j} (1+b)^{i} + \beta \sum_{j=k+1}^{k+m} a_{[j]} (1+b)^{k+m-j} \\
&\quad + \sum_{j=k+m+1}^{n} \gamma_{[j]} \\
&= \sum_{j=1}^{k+m} w_j a_{[j]} + \sum_{j=k+m+1}^{n} \gamma_{[j]}.
\end{aligned}
\tag{10}
$$

□

Corollary 6. *If $m = n - k$, then*

$$F\left(d_1, D, \pi, n-k\right) = \sum_{j=1}^{n} w_j a_{[j]}, \tag{11}$$

where

$$
w_j = \begin{cases}
\left\{(\alpha + k\theta) + \beta\left[(1+b)^{m} - 1\right]\right\}(1+b)^{k-j} & \\
\quad -\theta \sum_{i=0}^{k-j} (1+b)^{i}, & j = 1, 2, \ldots, k \\
\beta (1+b)^{k+m-j}, & j = k+1, k+2, \ldots, n.
\end{cases}
\tag{12}
$$

Equation (11) can be viewed as the scalar product of two vectors, w_j and $a_{[j]}$, respectively, ($j = 1, \ldots, n$). It is well known (from Hardy et al. [22]) that (11) is minimized by sorting the elements of the w_j and $a_{[j]}$ vectors in opposite orders. This procedure can be done in $O(n \log n)$ time. We refer to this rule as the HLP rule in the rest of the paper.

Theorem 7. *If the number of $\overline{D}$ jobs is given, then the problem $1 \mid P_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$ can be formulated as an assignment problem.*

Proof. We define z_{jr} as a 0/1 variable such that $z_{jr} = 1$ if job J_j is scheduled in position r, and $z_{jr} = 0$, otherwise. We can formulate the problem $1 \mid P_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$ as the following assignment problem:

$$
\begin{aligned}
\mathrm{AP}(m) \qquad \text{Min} \quad & \sum_{j=1}^{n} \sum_{r=1}^{n} C_{jr}^{m} z_{jr} \\
\text{Subject to} \quad & \\
& \sum_{r=1}^{n} z_{jr} = 1, \quad j = 1, 2, \ldots, n \\
& \sum_{j=1}^{n} z_{jr} = 1, \quad r = 1, 2, \ldots, n \\
& z_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n,
\end{aligned}
\tag{13}
$$

where

$$
C_{jr}^{m} = \begin{cases}
a_j \left[\left((\alpha + k\theta) + \beta\left((1+b)^{m} - 1\right)\right)(1+b)^{k-r} \right. & \\
\quad \left. -\theta \sum_{i=0}^{k-r} (1+b)^{i}\right], & r = 1, 2, \ldots, k \\
a_j \left[\beta (1+b)^{k+m-r}\right], & r = k+1, k+2, \ldots, k+m \\
\gamma_j, & r = k+m+1, \ldots, n.
\end{cases}
\tag{14}
$$

Therefore, based on the above analysis, we can obtain a polynomial algorithm for the problem $1 \mid P_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$. □

Algorithm 8.

Step 0. By Lemma 4, calculate $k = \lceil (\beta - \alpha)/\theta \rceil$.

Step 1. For m from 0 to $n - k - 1$, solve the above assignment problem AP(m) to obtain a local optimal schedule and the total cost $F(m)$.

Step 2. For $m = n - k$, first calculate the positional weights defined by (12) and assign the n jobs to the corresponding positions according to the HLP rule and then use (11) to evaluate the objective value $F(n - k)$.

Step 3. The global optimal schedule is the one with the minimum total cost given by $\min\{F(m) \mid 0 \le m \le n - k\}$.

Based on the above analysis, we have the following result.

Theorem 9. *The scheduling problem $1 \mid p_j = a_j + bt \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$ can be solved by Algorithm 8 in $O(n^4)$ time.*

Proof. For a given m, our problem becomes identical to the classical assignment problem and can be solved in $O(n^3)$ time. Since $0 \le m \le n - k \le n$, the overall time requirement of Algorithm 8 is $O(n^4)$. □

Example 10. Consider the instance with

$$\begin{gathered} n=5, \quad \alpha=2, \quad \beta=4, \quad \theta=0.5, \quad b=0.3, \\ a_1=4, \quad a_2=3, \quad a_3=6, \quad a_4=9, \quad a_5=11, \\ \gamma_1=6, \quad \gamma_2=4, \quad \gamma_3=5, \quad \gamma_4=3, \quad \gamma_5=30. \end{gathered} \tag{15}$$

Now we apply Algorithm 8 to solve Example 10.

Step 0. Calculate the index $k=\lceil(\beta-\alpha)/\theta\rceil=\lceil(4-2)/0.5\rceil=4$.

Step 1. When $m=0$, the C_{jr}^0 values can be calculated by (14) and given below:

$$C_{jr}^0=\begin{pmatrix} 22.7780 & 19.0600 & 16.2000 & 14 & 6 \\ 17.0835 & 14.2950 & 12.1500 & 10.5 & 4 \\ 34.1670 & 28.5900 & 24.3000 & 21 & 5 \\ 51.2505 & 42.8850 & 36.4500 & 31.5 & 3 \\ 62.6395 & 52.4150 & 44.5500 & 38.5 & 30 \end{pmatrix}. \tag{16}$$

The optimal job sequence is $(J_2, J_1, J_3, J_5, J_4)$.

The optimal objective value is $F(0)=101.9435$.

Step 2. When $m=1$, the w_j values can be calculated by (12):

$$\begin{gathered} w_1=8.3309, \quad w_2=6.7930, \quad w_3=5.6100, \\ w_4=4.7000, \quad w_5=4.0000. \end{gathered} \tag{17}$$

The optimal job sequence is $(J_2, J_1, J_3, J_4, J_5)$.

The optimal objective value is $F(1)=172.1247$.

Step 3. The global optimal objective is $\min\{F(0), F(1)\}=101.9435$. The global optimal schedule is $(J_2, J_1, J_3, J_5, J_4)$.

3.2. Job-Position-Dependent Learning Effect Model. By the same way as in the previous subsection, we consider the following scheduling problem: $1 \mid P_j=a_j r^a \mid \alpha d_1+\beta D+\theta\sum_{J_j\in E}E_j+\sum_{J_j\in T}\gamma_j$.

Lemma 11. *For a given schedule $\pi=(J_{[1]}, J_{[2]}, \ldots, J_{[n]})$, if the starting time of the first job is 0, then $C_{[r]}=\sum_{j=1}^r a_{[j]}j^a$ and $\sum_{j=1}^n C_j=\sum_{j=1}^n a_{[j]}(n+1-j)j^a$.*

Lemma 12. *For the problem $1 \mid p_j=a_j r^a \mid \alpha d_1+\beta D+\theta\sum_{J_j\in E}E_j+\sum_{J_j\in T}\gamma_j$, if the job sequence is $\pi=(J_{[1]}, J_{[2]}, \ldots, J_{[n]})$ and $m=|\overline{D}|$, then the objective function can be expressed as*

$$F(d_1, D, \pi, m)=\sum_{j=1}^{k+m}\overline{w}_j a_{[j]}+\sum_{j=k+m+1}^{n}\gamma_{[j]}, \tag{18}$$

where $\overline{w}_j=\begin{cases} (\alpha-\theta+\theta j)j^a, & j=1,2,\ldots,k; \\ \beta j^a & j=k+1,k+2,\ldots,k+m. \end{cases}$

Proof. By Lemmas 3 and 11, we have

$$\begin{aligned} F&(d, D, \pi, m) \\ &=\alpha d_1+\beta D+\theta\sum_{J_j\in E}E_j+\sum_{J_j\in T}\gamma_j \\ &=\alpha C_{[k]}+\beta\sum_{j=k+1}^{k+m}a_{[j]}(1+b)^{n-j} \\ &\quad+\theta\sum_{j=1}^{k}\left(C_{[k]}-C_{[j]}\right)+\sum_{J_j\in T}\gamma_{[j]} \\ &=(\alpha+k\theta)C_{[k]}-\theta\sum_{j=1}^{k}C_{[j]}+\beta\sum_{j=k+1}^{k+m}a_{[j]}(1+b)^{n-j} \\ &\quad+\sum_{J_j\in T}\gamma_{[j]} \\ &=(\alpha+k\theta)\sum_{j=1}^{k}a_{[j]}j^a-\theta\sum_{j=1}^{k}a_{[j]}(k+1-j)j^a \\ &\quad+\beta\sum_{j=k+1}^{k+m}a_{[j]}j^a+\sum_{j=k+m+1}^{n}\gamma_{[j]} \\ &=\sum_{j=1}^{k+m}w_j a_{[j]}+\sum_{j=k+m+1}^{n}\gamma_{[j]}. \end{aligned} \tag{19}$$

□

Corollary 13. *If $m=n-k$, then*

$$F(d_1, D, \pi, m)=\sum_{j=1}^{n}\overline{w}_j a_{[j]}, \tag{20}$$

where $\overline{w}_j=\begin{cases} (\alpha-\theta+\theta j)j^a, & j=1,2,\ldots,k; \\ \beta j^a & j=k+1,k+2,\ldots,k+m. \end{cases}$

Equation (20) can be viewed as the scalar product of two vectors, $\overline{w}_j$ and $a_{[j]}$ vectors, respectively. The procedure can be done in $O(n\log n)$ time by the HLP rule.

Theorem 14. *If we fix the number of $\overline{D}$ jobs, then the problem $1 \mid p_j=a_j r^a \mid \alpha d_1+\beta D+\theta\sum_{J_j\in E}E_j+\sum_{J_j\in T}\gamma_j$ can be formulated as an assignment problem.*

Proof. It is similar to the proof of Theorem 7.Again, we can define

$$\overline{C}_{jr}^m=\begin{cases} (\alpha-\theta-\theta r)r^a a_j, & r=1,2,\ldots,k \\ \beta r^a a_j, & r=k+1,k+2,\ldots,k+m \\ \gamma_j, & r=k+m+1,\ldots,n, \end{cases} \tag{21}$$

as the cost of assigning job J_j $(j=1,2,\ldots,n)$ to the rth $(r=1,2,\ldots,n)$ position in the schedule. Then the problem

$1 \mid p_j = a_j r^a \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$ can be formulated as the following assignment problem:

$$\begin{aligned} \text{AP}(m) \qquad & \text{Min} \quad \sum_{j=1}^{n} \sum_{r=1}^{n} \overline{C}_{jr}^{m} z_{jr} \\ & \text{Subject to} \\ & \sum_{r=1}^{n} z_{jr} = 1, \quad j = 1, 2, \ldots, n \\ & \sum_{j=1}^{n} z_{jr} = 1, \quad r = 1, 2, \ldots, n \\ & z_{jr} = 0 \text{ or } 1, \quad j, r = 1, 2, \ldots, n. \end{aligned} \tag{22}$$

□

Similar to Section 3.1, we have the following theorem.

Theorem 15. *The scheduling problem* $1 \mid p_j = a_j r^a \mid \alpha d_1 + \beta D + \theta \sum_{J_j \in E} E_j + \sum_{J_j \in T} \gamma_j$ *can be solved in* $O(n^4)$ *time.*

4. Conclusions

We have considered the single machine due-window assignment scheduling problem with variable job processing times. The objective is to minimize a linear combination of earliness, tardiness, the window location, window size, and weighted number of tardy jobs. We proposed a polynomial-time algorithm, respectively, for the learning effect and the deteriorating jobs. Obviously, if $a > 0$ (i.e., deterioration or aging effect) and $b < 0$ (i.e., shortening processing times), then the results of this paper still hold. In future research, we plan to explore more realistic settings, such as group technology scheduling problems, flexible flow shop scheduling problems, and unrelated parallel machines scheduling problems, or optimize other performance measures with variable job processing time.

Conflict of Interests

Yu-Bin Wu and Ping Ji declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments on earlier versions of this paper. This research was supported by the Science Research Foundation of Shenyang Aerospace University (Grant no. 201304Y) and the Research Grants Council of the Hong Kong Special Administrative Region, China (Project no. PolyU 517011).

References

[1] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[2] S. Gawiejnowicz, *Time-Dependent Scheduling*, Springer, Berlin , Germany, 2008.

[3] R. Rudek, "Scheduling problems with position dependent job processing times: computational complexity results," *Annals of Operations Research*, vol. 196, pp. 491–516, 2012.

[4] J.-B. Wang and M.-Z. Wang, "Worst-case behavior of simple sequencing rules in flow shop scheduling with general position-dependent learning effects," *Annals of Operations Research*, vol. 191, pp. 155–169, 2011.

[5] L.-H. Sun, K. Cui, J.-H. Chen, J. Wang, and X.-C. He, "Research on permutation flow shop scheduling problems with general position-dependent learning effects," *Annals of Operations Research*, vol. 211, pp. 473–480, 2013.

[6] L.-H. Sun, K. Cui, J.-H. Chen, J. Wang, and X.-C. He, "Some results of the worst-case analysis for flow shop scheduling with a learning effect," *Annals of Operations Research*, vol. 211, pp. 481–490, 2013.

[7] Y. Y. Lu, G. Li, Y. B. Wu, and P. Ji, "Optimal due-date assignment problem with learning effect and resource-dependent processing times," *Optimization Letters*, vol. 8, no. 1, pp. 113–127, 2014.

[8] J.-B. Wang and M.-Z. Wang, "Single-machine due-window assignment and scheduling with learning effect and resource-dependent processing times," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 5, Article ID 1450036, 2014.

[9] J.-B. Wang and M.-Z. Wang, "Single-machine scheduling with nonlinear deterioration," *Optimization Letters*, vol. 6, no. 1, pp. 87–98, 2012.

[10] J.-B. Wang and M.-Z. Wang, "Minimizing makespan in three-machine flow shops with deteriorating jobs," *Computers and Operations Research*, vol. 40, no. 2, pp. 547–557, 2013.

[11] X.-R. Wang and J.-J. Wang, "Single-machine scheduling with convex resource dependent processing times and deteriorating jobs," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 2388–2393, 2013.

[12] Y.-T. Xu, Y. Zhang, and X. Huang, "Single-machine ready times scheduling with group technology and proportional linear deterioration," *Applied Mathematical Modelling*, vol. 38, no. 1, pp. 384–391, 2014.

[13] T. C. E. Cheng, S.-J. Yang, and D.-L. Yang, "Common due-window assignment and scheduling of linear time-dependent deteriorating jobs and a deteriorating maintenance activity," *International Journal of Production Economics*, vol. 135, no. 1, pp. 154–161, 2012.

[14] S.-J. Yang, H.-T. Lee, and J.-Y. Guo, "Multiple common due dates assignment and scheduling problems with resource allocation and general position-dependent deterioration effect," *International Journal of Advanced Manufacturing Technology*, vol. 67, no. 1–4, pp. 181–188, 2013.

[15] J. Liu, Y. Wang, and X. Min, "Single-machine scheduling with common due-window assignment for deteriorating jobs," *Journal of the Operational Research Society*, vol. 65, no. 2, pp. 291–301, 2014.

[16] J.-B. Wang and C. Wang, "Single-machine due-window assignment problem with learning effect and deteriorating jobs," *Applied Mathematical Modelling*, vol. 35, no. 8, pp. 4017–4022, 2011.

[17] J.-B. Wang, L. Liu, and C. Wang, "Single machine SLK/DIF due window assignment problem with learning effect and deteriorating jobs," *Applied Mathematical Modelling*, vol. 37, no. 18-19, pp. 8394–8400, 2013.

[18] S. Li, C. T. Ng, and J. Yuan, "Scheduling deteriorating jobs with CON/SLK due date assignment on a single machine," *International Journal of Production Economics*, vol. 131, no. 2, pp. 747–751, 2011.

[19] Y. Yin, T. C. E. Cheng, C.-J. Hsu, and C.-C. Wu, "Single-machine batch delivery scheduling with an assignable common due window," *Omega*, vol. 41, no. 2, pp. 216–225, 2013.

[20] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[21] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[22] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, Cambridge, UK, 1967.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 530678, 16 pages
http://dx.doi.org/10.1155/2014/530678

Research Article

An Order Insertion Scheduling Model of Logistics Service Supply Chain Considering Capacity and Time Factors

Weihua Liu, Yi Yang, Shuqing Wang, and Yang Liu

College of Management and Economics, Tianjin University, Tianjin 300072, China

Correspondence should be addressed to Weihua Liu; lwhliu888@163.com

Received 26 June 2014; Revised 25 July 2014; Accepted 26 July 2014; Published 2 September 2014

Academic Editor: Dehua Xu

Order insertion often occurs in the scheduling process of logistics service supply chain (LSSC), which disturbs normal time scheduling especially in the environment of mass customization logistics service. This study analyses order similarity coefficient and order insertion operation process and then establishes an order insertion scheduling model of LSSC with service capacity and time factors considered. This model aims to minimize the average unit volume operation cost of logistics service integrator and maximize the average satisfaction degree of functional logistics service providers. In order to verify the viability and effectiveness of our model, a specific example is numerically analyzed. Some interesting conclusions are obtained. First, along with the increase of completion time delay coefficient permitted by customers, the possible inserting order volume first increases and then trends to be stable. Second, supply chain performance reaches the best when the volume of inserting order is equal to the surplus volume of the normal operation capacity in mass service process. Third, the larger the normal operation capacity in mass service process is, the bigger the possible inserting order's volume will be. Moreover, compared to increasing the completion time delay coefficient, improving the normal operation capacity of mass service process is more useful.

1. Introduction

Faced with presently growing demand for customized logistics services, many logistics enterprises expand their business beyond mass service and change logistics service mode to provide customized service. Specifically, these enterprises attempt to provide mass customization logistics services (MCLS) instead of mass logistics services [1]. In order to meet customized service requirements and achieve necessary mass service capabilities in the MCLS environment, logistics enterprises usually organize unions and integration [2]. And the competitiveness of the LSSC depends on the ability to offer mass customization service with the cost as low as possible through reasonable scheduling [3].

In LSSC scheduling, time scheduling is quite important, and it should balance customer demand and logistics service capacity. Compared with production supply chain, service cannot be reserved or buffered in the form of tangible products. Therefore, operation of service supply chain is much more easily influenced by outside environment, especially when there is an order insertion. Order insertion refers to the situation where new orders arrive and are required to be inserted into a scheduled order sequence when production capacity is fixed and resources are limited, which is common in the practice of service industry [4]. The insertion of new jobs into an existent schedule, as well as most of the other types of disruptions, may require the total or partial rescheduling of previously allocated and new jobs. For example, as one of the biggest express companies in China, Yuantong Express Company always faces the problem of insertion scheduling. Normally, at twelve o'clock at noon every day, Yuantong Express will collect all the packages in the morning and forward them to customers at transit centre together. However, some emergent orders happen occasionally, so Yuantong Express will consider the factors of time and operation cost, judge whether it could carry out insertion scheduling, and then make a new scheduling planning. Due to the abruptness and urgency of order insertion, it will be more difficult to make time schedule in LSSC considering logistics service capacity limitation, time requirements, and increased cost. Thus, how to arrange service capacity and operation time reasonably becomes a realistic problem every LSI faces.

While order insertion has been studied by many scholars in production supply chain scheduling so far, it is still a relatively new issue in service supply chain. Though some scholars have become interested in service supply chain scheduling problem recently, for example, [3, 5, 6], they consider more normal scheduling situation than order insertion. Obviously, time scheduling with order insertion is much more complex and thus worthwhile to research.

Based on the literature review and specific practical observation about logistics enterprises, it is found that under MCLS environment, LSI needs to focus on solving three problems during time scheduling within order insertion, which are the lack of existed research and thus the focus of paper.

First, due to the abruptness and urgency of order insertion, LSI will change the original order schedule for a rescheduled one. Therefore, it is necessary to discuss whether it is feasible to insert new orders with full consideration of original time schedule.

Second, if the new order to be inserted is similar to original orders, then how could the LSI make the best use of this similarity and reschedule the time under the MCLS environment. Factors such as logistics service capacity, time requirement from customers, and operation cost caused by order insertion have to be dealt with properly in the model.

Third, it is of great significance for the LSI managers to figure out what factors that do have influence on order-inserting decisions and what are the specific influence rules in practical scheduling process. With the help of these rules, LSI could deal with order insertion problem better.

These problems mentioned above would be answered in this paper. Based on the research of Liu et al. [3] and Liu et al. [6], this paper has further discussed order similarity coefficient and the order insertion process in LSSC, which contributes to two essential constraints. Furthermore, with full consideration of both capacity and time two factors, an order insertion scheduling model of LSSC has been established, aiming to minimize the average unit volume operation cost of the LSI and maximize the average satisfaction degree of FLSPs. All constraints in our model are different from those in previous researches. Under these conditions, some interesting findings are obtained. First, whether the new order could be inserted or not depends on its volume that will further affect supply chain comprehensive performance. In particular, supply chain will get the best performance when the inserted order's volume is equal to the surplus of the normal operation capacity of mass service process. Besides, time requirement from customers will also influence supply chain comprehensive performance, and some allowable delay in completion time appropriately will contribute to better performance. What is more, compared to increasing completion time delay coefficient, improving normal operation capacity of mass service process is more useful in increasing the upper limit of possible inserting order volume. Therefore, choosing to increase normal operation capacity of mass service process is a prior strategy when LSI needs to solve new order insertion problem.

The rest of the paper is organized as follows. Section 2 systematically reviews the existing researches of order insertion in supply chain scheduling. In Section 3, the problem and basic assumptions are described in detail and notations used in model building are listed specifically. Section 4 gives an order insertion scheduling model of LSSC considering capacity and time factors. In Section 5, the model solution is calculated within genetic algorithm. In Section 6, numerical examples are given to explore the influence of parameters related to new order on the time scheduling performance. Section 7 is a concluding section.

2. Literature Review

Our research is mainly concerned about the order insertion scheduling of LSSC under the environment of MCLC. Thus, the literature review is mainly related to MC and order insertion scheduling. Our research aims will be proposed after summarizing the literature development and its deficiencies.

2.1. Researches on MC and Scheduling in LSSC. Since Pine [7] proposed that mass customization mode would become the new frontier in business competition in 1993, MC mode has increasingly become the mainstream mode of operation after nearly 20 years of development and application. Due to its significant improvement on operational performance, mass customization has been extensively studied and applied in the field of production supply chain. So many scholars conducted monographic studies. Fogliatto et al. [8] reviewed the literature on MC production in detail since the 1980s. From the view of the current domestic and international research progress, researches on MC were mainly developed within the MC production mode in manufacturing industry, including MC mode and its product development; see, for example, [9], production planning and control technology in MC; see, for example, [10], cost study of MC; see, for example, [11], research on the factors and conditions that influence MC; see, for example, [12].

The studies on the supply chain scheduling with the mass customization production mode was a new upsurge in recent years. Operation scheduling under MC environment is more dynamic and of more complexity. Most of the researches on supply chain scheduling have been focusing on the manufacturing industry and have achieved further results. In 2003, Hall and Potts [13] published a paper named "Supply chain scheduling: batching and delivery," which is an earlier systematical research on the supply chain scheduling model. Many earlier studies on supply chain scheduling pay attention to the job shop scheduling within a single enterprise, for example, [14]. And they are mainly concerned about the arrangement of processing procedures and order operation sequence. Some scholars also are concerned about the coordination of assembly system in manufacturing enterprise; see, for example, [15]. However, the studies on the supply chain scheduling with the mass customization production mode was a new upsurge in recent years; see, for example, [16].

Many scholars have carried out targeted researches on the supply chain scheduling. Cost is the primary factor considered in many above researches; see, for example, [17]. And most of these researches assume that the order completion time required by customers or the delivery time required by suppliers was fixed. But as an important index reflecting supply chain agility, customers' time requirements might change in a lot of cases [18, 19] or the operation time requirements to LSPs are not with strict limitation but allow a certain amount of variation; see, for example, [20]. Thus, it is necessary to consider the influence of service completing time ahead of schedule or delay caused by customers or LSPs on the scheduling results [21]. Besides cost objective, punctual delivery of service order and FLSP's satisfaction also have a direct influence on customer satisfaction. Therefore, it is necessary to consider the influence of the different importance degree of different objective functions on the supply chain performance. However, the current literature has not addressed this issue.

Although now the research on supply chain scheduling under MC environment becomes more and more complete, the one on service supply chain field is still significantly deficient. Similar to the manufacturing supply chain, researches on service supply chain are mainly focusing on the service process scheduling; see, for example, [22] and the order assignment scheduling; see, for example, [2]. The most related researches to this paper are Liu et al. [3] and Liu et al. [6], in which time scheduling problem in LSSC is discussed. But they only focused on the scheduling of a decided set of orders without taking order insertion situation and the influence of capacity support on time scheduling result into consideration. Thus, in general, research on time scheduling is still far from sufficient. It is necessary to study the time scheduling problem in service supply chain field (especially in LSSC field).

2.2. Researches on Order Insertion Scheduling of Supply Chain. Order insertion is a special and important content in supply chain scheduling research. In production supply chain field, order insertion problem has gained much attention. Order insertion refers to inserting a new arrival order into a scheduled order sequence on the premise that the production capacity has been allocated. Sometimes the new inserted order will replace original ones and form a new order sequence. Therefore, previous order insertion scheduling researches mainly focused on two research emphasis. One is the order insertion method. Since order insertion process may break original production schedule, it may cause other original orders to be delayed. Thus, it is necessary and useful to explore reasonable order insertion methods. At present, common method of inserting a new order includes "right shift," "insertion in the end," and "total rescheduling." Some researches combined order insertion problem with other disturbance factors such as machine breakdown and boiled down. Another focus is the problem to decide the priority of inserting order.

Compared to that in make-to-stock production mode, order insertion problem in make-to-order mode has gained some attention as well. To find out the influencing factors of order insertion decision is very important to build order insertion models. Some scholars explored these influential factors under different situations, such as time, cost, scheduling efficiency, and scheduling stability. For example, in order insertion model, time constraint is often regarded as an important considering factor and decreasing time delay is always regarded as a crucial scheduling goal. Duron et al. [23] used operation time and lead time to characterize different original orders and assumed that new order insertion operation may cause delay in original orders' delivery. Duron et al. [24] tried to reduce original order delay caused by new order insertion operation through a real-time approach. Besides, many scholars regarded minimizing supply chain cost as a frequently used objective in order insertion model; see, for example, [25]. Gomes et al. [26] studied order insertion problem in make-to-order industries. They took scheduling efficiency and stability index as measures of the influence of rescheduling process on original schedule and introduced a reactive scheduling algorithm to update scheduling table.

As can be seen from the above review of the literatures, the existing researches have three deficiencies. First, in production supply chain, research on order insertion is mainly focused on the priority algorithm of inserting orders, which aim at finding excellent algorithm to improve optimizing efficiency. Moreover, many of the literatures assume that supply chain capacity can afford the new order insertion requirement and other original orders' satisfaction degree is not affected, but real situations are not the same. Second, new inserting order has its own features both in structure and required operations. The existing researches do not consider the factors that whether the new inserting order can be operated together with original orders considering these feathers. Meanwhile, it is not be discussed whether the FLSP's capacity can afford inserting operation. Third, in the existing researches on MC service supply chain, order insertion scheduling research considering time factors is rare. Thus, based on these three deficiencies, this paper will fully consider the similarity between original orders and the new inserting one as well as the influence of service capacity of supply chain on order insertion decision. In the MC service environment, this paper will deeply explore decision problem that whether a new arrival order can be inserted into original orders to be rescheduled. Furthermore, some useful references are offered for better study on order insertion issue.

3. Problem Description and Model Assumptions

In this section, the problem and basic assumptions are described in detail. Notations used in model building are listed as well. In Section 3.1, both the problems involved in the model and the decision process of order insertion are described. In Section 3.2, important assumptions in our model are listed specifically. In Section 3.3, related notions defined in this paper are provided in detail as well as the scheduling logic in our model.

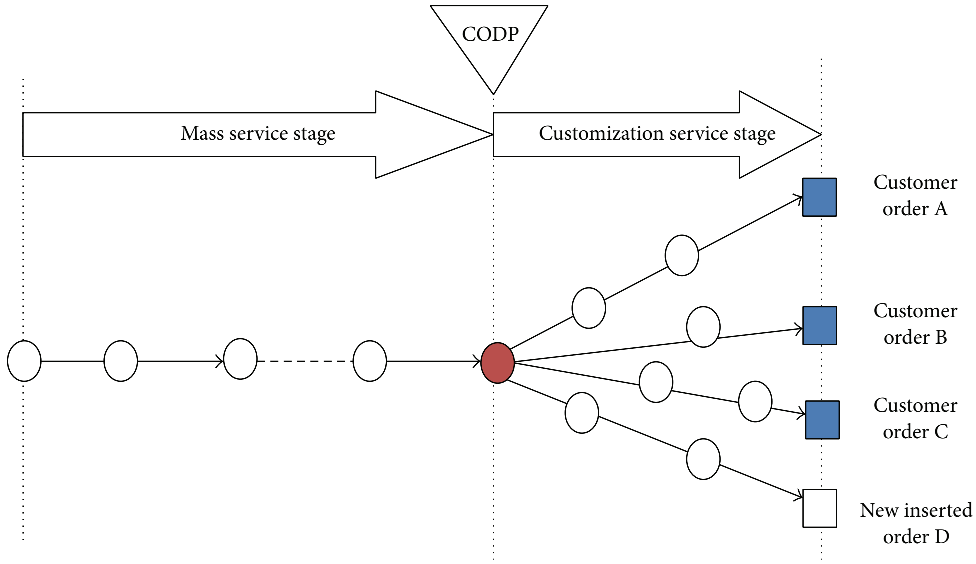


Figure 1: Customer orders' operation processes schematic diagram of general LSSC.

3.1. Problem Description. In a two-echelon LSSC with one LSI and many FLSPs, LSI accepts customers' service orders and hands them to multiple FLSPs to operate. And LSI faces multiple customer service orders at the same time and each logistics service order consists of multiple service processes, which could be divided into two types, that is, personalized service process and large-scale service process, where whether to integrate the large-scale service process of customer i and customer j $(j \neq i)$ to be operated together or not can be chosen. These two kinds of service processes are called "mass service process" and "customized service process," respectively, in this paper.

Since customer orders arrive in sequence, after the scheduling process of an original set of order has been finished, new order may occur to be inserted into schedule, including urgent order and order which asked to be operated first by customers. At this time, LSI needs to first decide whether this new arrival order could be inserted while synthetically considering the characteristic of new orders and FLSPs' capacity. Furthermore, scheduling decision model and method of order insertion problem should be thought over by LSI.

First, a specific example is used to illustrate this scheduling problem. See Figure 1; there are three original customer orders (order A, order B, and order C) whose partial service processes can be operated together in mass mode due to the similarity in their service content. Service processes after CODP will be operated in customized mode, respectively. Upon arrival of new customer order D, LSI needs to decide whether to insert this new order based on synthetically consideration of this new order's and original orders' characteristic as well as the FLSPs' operation capacity. For the convenience of study, it is necessary to simplify the process as shown in Figure 2. It is assumed that mass process is operated by FLSP 1 and the customized process of the jth customer order is operated by the jth FLSP of customized stage, respectively. The FLSP 1 completes the mass process and it has capacity limit. Namely, in a normal completion time T_1, FLSP 1 can finish an order whose volume is $\overline{N}$. If the FLSP is required to operate a task whose volume is more than $\overline{N}$, then order setup time increases or capacity reorganization is needed to be carried out. We assume that there is no capacity limit for the jth FLSP (in this example $j = 2, 3, 4, 5$) because of customization service.

One thing needs to be noted. Mass customization service could be normally divided into two stages: mass service stage and customization service stage. For the mass service stage, multiple orders are integrated and operated together, so it is necessary to consider the factors of time, operation cost, and service process for all the multiple orders. For the customization service stage, each order is finished by customization process; there is no relationship among multiple orders. Obviously, order insertion scheduling is an activity that new orders are required to be inserted into a scheduled order sequence. Thus insertion scheduling always be carried out in mass service stage but not happens in customization service stage.

Because the new inserted order is unpredictable, whether it can be inserted into original schedule should be considered. Therefore, the judging criteria are proposed in the Figure 3.

Judging criterion 1 is as follows: whether the mass service process of the new arrival order can be operated together with that of the existing orders; specifically, whether similarity between the mass process of new arrival order and the original ones exists. If the answer is positive, then turn to judging criteria two.

Judging criterion 2 is as follows: whether the order insertion operated is feasible in terms of time requirement and economic consideration, namely, use the time scheduling model proposed in this paper to carry through model judgment. If this model has solution, then the order insertion decision is feasible by the model judgment and FLSPs can

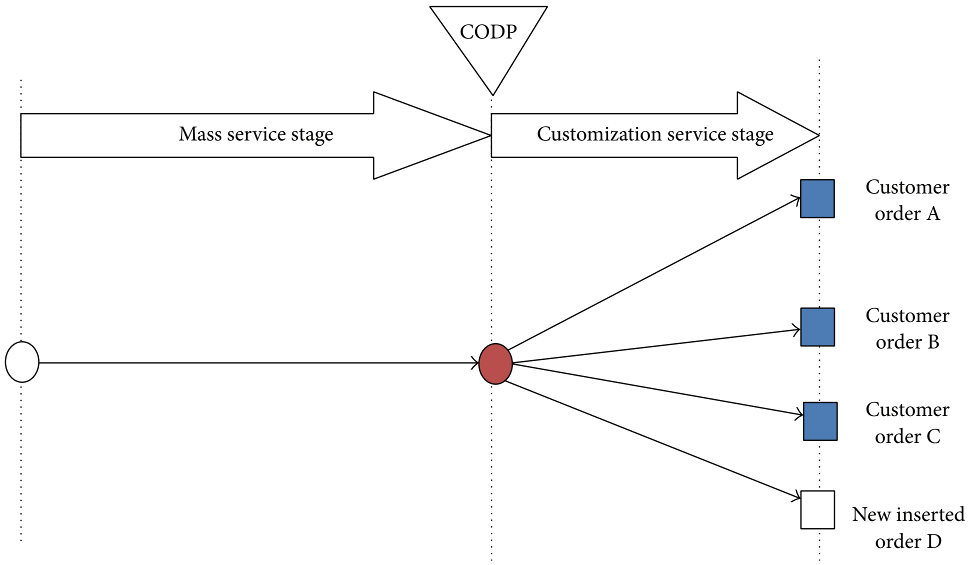


Figure 2: Customer orders' operation processes schematic diagram of LSSC which is simplified.

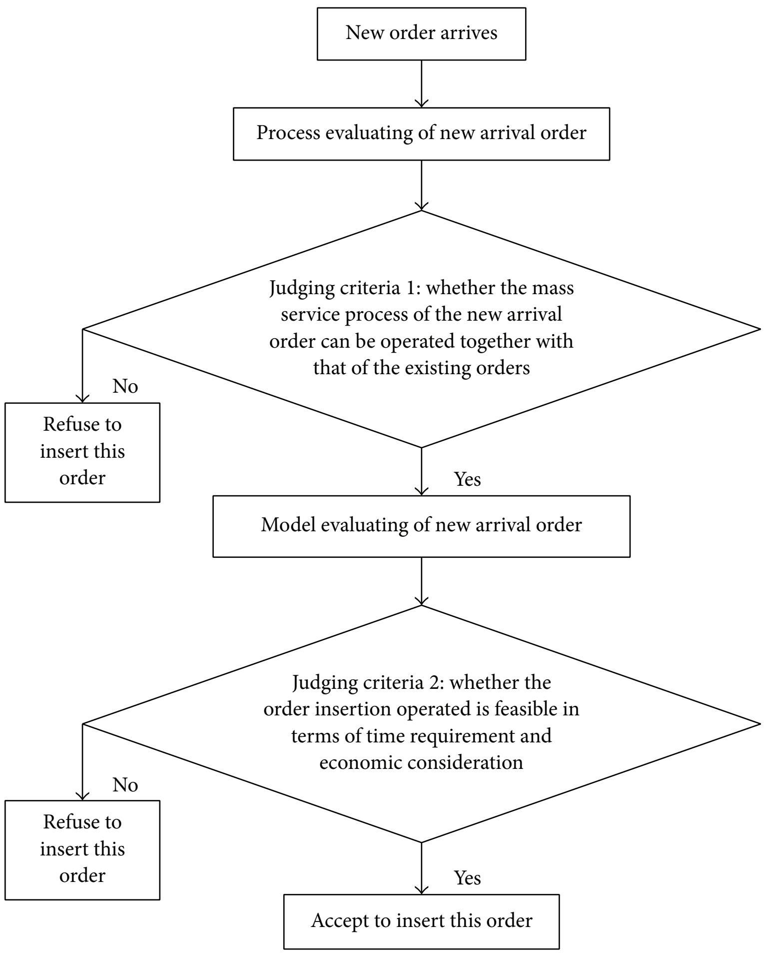


Figure 3: Determining process of new order's insertion decision.

carry on order insertion operation according to scheduling results. If this model has no solution, then this new order cannot be inserted into the original schedule. For example, it may not meet time requirement or profit requirement.

In this paper, two judging criteria are proposed in order to judge whether a new order could be inserted into the original schedule or not. If it is impossible to insert new order, a completely new scheduling plan should be put out.

The model parameters and variables are summarized in Table 1.

3.2. Model Assumptions. In order to build our model conveniently, some important assumptions are proposed as follows.

Assumption 1. Customer orders arrive at different time. Original orders arrive first and the new inserted order arrives later. Before arrival of new order, original orders have been scheduled. FLSPs have set their normal operation time and necessary capacity plan for each process according to schedule table. The new arrived order needs to go through two judging criteria mentioned above, but in this paper, we only focus on the second judging criterion and assume that the new order have passed the first judging criteria. That means it is assumed that the new inserted order could be scheduled with original orders together. If new arrival order is inserted into the original ones to be operated together, the normal operation time of original orders may be compressed or delayed.

Assumption 2. In our model, we assume that there is only one new arrived order that needs to be inserted and do not consider multiorder insertion problem. If the new order is able to be inserted, then in the rescheduling process, we view all the orders to have the same priority, since all the orders are operated together but not operated one by one in the mass process.

Assumption 3. If order is delayed, LSIs will be punished by customer; while if the order is finished in advance, they will not. Within the endurable time of customer, the unit time punish cost is C_j^{delay}. If the actual completion time is T_j, then the punish cost is $C_j^{\text{delay}}[T_j - T_j^{\exp}, 0]^+$. If the time delay is beyond customer's durable time, then customer order cannot continue being operated and supply chain collapses.

Assumption 4. Each provider can compress or delay their operation time through increasing input capacity, such as increasing vehicle or lengthening working time in order to meet customer's time requirement. Correspondingly, LSI needs to pay extra cost for capacity input increase. Extra cost for unit time compression or delay in mass process is C_1^{ext} and that in customized process is C_{2j}^{ext}.

Assumption 5. In mass service process, the case may occur that service capacity is insufficient due to FLSP's capacity limitation. But in customized service process, since each service order is operated by a specialized provider, service capacity is assumed to be always sufficient.

Assumption 6. Influence of new inserted order on CODP is not considered in this paper; that is, the CODP is assumed unchanged.

3.3. Preparation for Model Building

3.3.1. Order Similarity Coefficient. Similarity between original orders and new inserted order must be taken into consideration when dealing with order insertion problem. In this paper, λ is used todenote order similarity coefficient. Analysis on order similarity is a crucial step to consider order insertion decision. In production supply chain, clustering analysis on different orders is often carried out according to product's modular construction. But in service supply chain, different service orders have many differences and it is hard to choose a modular measure index like tangible products. Therefore, this paper will focus on the analysis of service order similarity coefficient.

Take the research findings of order similarity of tangible product for reference; see [27–30]; and taking service product features into consideration, a service order similarity coefficient is defined as a product of three indexes, which are customer demand similarity λ_1 (namely, time requirement similarity coefficient), service procedure similarity coefficient λ_2 (such as service standard similarity and service process similarity), and customer service product similarity coefficient λ_3 (such as function similarity and structure similarity of service product). The detailed calculation method for each kind of similarity coefficient will be introduced as follows.

(1) Customer Demand Similarity Coefficient λ_1. In the supply chain time scheduling, time requirement is the most important customer requirement. In this paper, time requirements similarities of different orders are used to denote customer demand similarity coefficient. The smaller the completion time requirement gap between original orders and new inserted order is, the more similar they are. And the average completion time of all the original orders is regarded as another benchmark. The closer the completion time requirement of new inserted order is, the bigger the similarity is. Detailed calculation method is shown in

$$\lambda_1 = \begin{cases} \dfrac{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}{T_{J_0+1}^{\exp}}, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} \le T_{J_0+1}^{\exp} \\ \dfrac{T_{J_0+1}^{\exp}}{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} > T_{J_0+1}^{\exp}. \end{cases} \quad (1)$$

(2) Service Procedure Similarity Coefficient λ_2. There are many differences for service procedure of different orders. Service procedure similarity between original orders and new inserted order has significant influence on the feasibility of order-inserting operation when facing order insertion decision. Generally speaking, similarity of service procedure consists of three parts. First is service standard similarity, such as service quality standard and standard for service staff. Second is service stage similarity, for example, whether there are some similar service stages between original orders and

Table 1: Notations for the model.

Notations	Description
C_1	The normal service cost per unit time per unit quantity of the FLSP in mass process in offering mass operation; this cost is the normal cost without time compression or delay in the operation when operated according to original schedule
C_1^{ext}	The extra service cost per unit time per unit quantity of the FLSP in mass process in offering mass operation; this cost is the extra cost due to inserting new order and rescheduling which caused extra time compression or delay in completion orders
C_{2j}	The normal service cost per unit time per unit quantity in offering customized operation for the jth customer order; this cost is the normal cost without time compression or delay in the operation when operated according to original schedule
C_{2j}^{ext}	The extra service cost per unit time per unit quantity of the FLSP in offering customized operation for the jth customer order; this cost is the extra cost due to inserting new order and rescheduling which caused extra time compression or delay in completion orders
C_j^{delay}	The penalty cost per unit time per unit quantity of the jth customer order as the order completion time is delayed; $j = 1, 2, \ldots, J_0, J_0 + 1$
F	The new order's price for per unit time per unit quantity offered by new order's customer
k	Since mass process has capacity limit, new order volume cannot increase infinitely; the new order volume is set to be no more than k times of the FLSP's normal upper limit capacity in mass service stage; $k > 0$
$\overline{N}$	The upper limit of FLSP's capacity in mass process, which is the upper limit of FLSP's capacity after scheduling according to original orders (due to real limitations, this upper limit may just be the sum volume of original orders or may be larger than the sum volume; if FLSP operates within this limit volume, operation time will not increase)
N_j	Volume of the jth order; $j = 1, 2, \ldots, J_0, J_0 + 1$
N_{j_0+1}	Volume of the new inserted order, where subscript $J_0 + 1$ characterizes the new inserted order, the same below
$N_{j_0+1}^{\max}$	The maximum of new inserted order's volume; in our numerical example, there are three original orders and one new inserted order; thus, $N_{j+1}^{\max}$ can be replaced by $N_4^{\max}$
$S_{\text{quantity},1}$	Service quantity satisfaction (capacity) degree of the mass service provider
$S_{\text{time},i}$	The ith FLSP's service time satisfaction degree, $i = 1, 2j$; $j = 1, 2, \ldots, J_0 + 1$
$S_{\text{quantity},1}^0$	Initial value of service quantity (capacity) satisfaction degree of the mass service provider
S_1	The satisfaction degree of the mass process provider
S_1^L	The lower limit of the satisfaction degree of the mass process provider
S_{2j}	The satisfaction degree of the customized process of the jth customer order; $j = 1, 2, \ldots, J_0 + 1$
S_{2j}^L	The lower limit of the satisfaction degree of the customized process of the jth customer order; $j = 1, 2, \ldots, J_0 + 1$
S_i	Satisfaction degree of the ith provider, $i = 1, 2$; $j = 1, 2, \ldots, J_0 + 1$
T_1	The normal operation time of original orders before new order's arrival in mass process; this normal operation time is generated by original order's scheduling result and is input parameter in numerical analysis
T_{2j}	The normal operation time of the jth original order before new order arrival in customized process. $j = 1, 2, 3, \ldots, J_0$, the same below
$T_j^{\exp}$	Completion time requirement of the jth customer order asked by customers; $j = 1, 2, \ldots, J_0, J_0 + 1$
T_j	Actual completion time of the jth customer order; $j = 1, 2, \ldots, J_0, J_0 + 1$
T_1^{ext}	Extra operation time of the provider in mass service process
T_{2j}^{ext}	Extra operation time of the jth customer order in customized process
w_1	The weight of objective function Z_1 in Z
w_2	The weight of objective function Z_2 in Z
Z_1	The total cost of LSI
Z_2	The average satisfaction of all processes in LSSC
$Z_1^{\min}$	The minimum of Z_1 when not considering the objective functions Z_2
Z	The objective function synthesized by Z_1 and Z_2, which is also called the comprehensive performance of LSSC
Z^*	The optimal value of Z
λ	Similarity coefficient of new inserted order and original orders

Table 1: Continued.

Notations	Description
λ_1	Time requirement similarity coefficient of new inserted order and original orders
λ_2	Service procedure similarity coefficient of new inserted order and original orders
λ_3	Customer service product similarity coefficient of new inserted order and original orders
β_j	The delay coefficient of the order completion time permitted by the jth customer for its order
Δpro	Profit increase resulted by new inserted order

Note: T_1^{ext}, T_{2j}^{ext} are decision variables.

new inserted order. The third is service process similarity. For example, if original orders have load or unload process but the new inserted order does not, then they are relatively different and the service step similarity coefficient is relatively small.

(3) Customer Service Product Similarity λ_3. It mainly refers to function and structure similarity of service product. For example, if operation for original orders and new inserted order are both transportation services for household chemicals, then they are of much similarity due to belonging to the same category. If original orders are transportation for steel and new inserted order is for cotton, obviously, they have less similarity.

Note that, since service procedure similarity coefficient λ_2 and customer service product similarity λ_3 are both difficult to be quantized. Therefore, values of λ_2 and λ_3 can be obtained by questionnaire or based on LSI's experience. Their value ranges from 0 to 1. Take the previous researches for [29, 30], the similarity coefficient is denoted as $\lambda = \lambda_1\lambda_2\lambda_3$. Thus, the order similarity coefficient can be shown as

$$\lambda = \lambda_1\lambda_2\lambda_3 = \begin{cases} \dfrac{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}{T_{J_0+1}^{\exp}}\lambda_2\lambda_3, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} \le T_{J_0+1}^{\exp} \\ \dfrac{T_{J_0+1}^{\exp}}{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}\lambda_2\lambda_3, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} > T_{J_0+1}^{\exp}. \end{cases} \quad (2)$$

3.3.2. Preparation Time for New Order. Different volume of new inserted order will make different influence on supply chain scheduling result. Obviously, the more the volume is, the more the operation stress of supply chain system will be. Along with the increase of new inserted order's volume, resource that needed to be prepared will increase. For example, FLSPs need to prepare more transportation vehicles or warehouses. Therefore, besides the increasing cost, new inserted order will cause increase of preparation time to redeploy resource. The influence of new inserted order's volume on time scheduling result should be reflected in this model. In general, increased order preparation time is positively correlated with three factors. The first one is the normal operation time of original orders in mass service process T_1. Second one is extra order volume $[\sum_{j=1}^{J_0+1} N_j - \overline{N}]^+$. Last one is the order similarity coefficient λ. t is used to denote the increased order preparation time caused by new inserted order as shown in

$$t = \frac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N}\right]^+}{\overline{N}}(1-\lambda)T_1 = \begin{cases} \dfrac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N}\right]^+}{\overline{N}}\left(1 - \dfrac{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}{T_{J_0+1}^{\exp}}\lambda_2\lambda_3\right)T_1, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} \le T_{J_0+1}^{\exp} \\ \dfrac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N}\right]^+}{\overline{N}}\left(1 - \dfrac{T_{J_0+1}^{\exp}}{(1/J_0)\sum_{j=1}^{J_0} T_j^{\exp}}\lambda_2\lambda_3\right)T_1, & \text{when } \dfrac{1}{J_0}\displaystyle\sum_{j=1}^{J_0} T_j^{\exp} > T_{J_0+1}^{\exp}, \end{cases} \quad (3)$$

where $[\sum_{j=1}^{J_0+1} N_j - \overline{N}]^+ = \max(\sum_{j=1}^{J_0+1} N_j - \overline{N}, 0)$. If $[\sum_{j=1}^{J_0+1} N_j - \overline{N}]^+ < 0$, then it is unnecessary to prepare for extra logistics service resource, such as vehicle. On the contrary, if $[\sum_{j=1}^{J_0+1} N_j - \overline{N}]^+ > 0$, it means that the new inserted order's volume is more than surplus of supply chain capacity; then extra preparation of logistics service resource is necessary and order preparation time will increase.

3.3.3. Order Rescheduling and Operation Time Logic. New inserted order will cause rescheduling of LSSC on the premise that the original order has been scheduled. Since the insertion of new order may cause completion time delay of original orders, it becomes a focusing goal for LSI to try to meet customer orders' time requirement through possible operation time compression. It is necessary to not only ensure original orders to be operated according to customer requirement but also guarantee profit increased after inserting a new order.

Based on the analysis above, the real completion time after order insertion could be decided by calculation. Namely, real completion time of the jth order is T_j = order preparation time (directly influenced by inserted order) + order operation time (which is able to be compressed or delayed); $j = 1, 2, \ldots, J_0 + 1$. Note that preparation time cannot be compressed, while operation time is compressible. As reflected in our model, this compressible (or deferrable) extra operation time is our scheduling content. Therefore, the decision variables are FLSP's extra operation time which aim at meeting customers' time requirement after inserting a new order, that is, T_i^{ext} $(i = 1, 2)$.

4. Model Building

This section will establish an order insertion scheduling model of LSSC considering capacity and time factors under order insertion situation. Section 4.1 will describe main model objectives, which are to minimize LSI's unit operation cost and to maximize the average satisfaction degree of all the providers after order insertion. Section 4.2 will present the main model constraints, which are time constraint, FLSP's satisfaction degree constraint, and capacity limit constraint.

4.1. Optimization Objectives of the Scheduling Model

4.1.1. Objective 1: To Minimize LSI's Unit Operation Cost after Order Insertion. The objective to minimize LSI's unit operation cost after order insertion could be expressed as

$$\text{Min } Z_1 = \frac{(f_1 + f_2 + f_3)}{\sum_{j=1}^{J_0+1} N_j}, \quad (4)$$

where f_1 is the total cost of normal operation in mass process and customization process. Consider

$$f_1 = C_1 T_1 \sum_{j=1}^{J_0+1} N_j + \sum_{j=1}^{J_0+1} \left(C_{2j} T_{2j} N_j \right). \quad (5)$$

f_2 is the extra operation cost in mass process and customized process. Consider

$$f_2 = C_1^{\text{ext}} \left| T_1^{\text{ext}} \right| \sum_{j=1}^{J_0+1} N_j + \sum_{j=1}^{J_0+1} \left(C_{2j}^{\text{ext}} \left| T_{2j}^{\text{ext}} \right| N_j \right). \quad (6)$$

f_3 is punishment cost for order completion time delay. Consider

$$f_3 = \sum_{j=1}^{J_0+1} \left[C_j^{\text{delay}} \left(T_j - T_j^{\text{exp}} \right)^+ \right],$$

$$T_j = T_1 + T_1^{\text{ext}} + T_{2j} + T_{2j}^{\text{ext}} + \frac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N} \right]^+}{\overline{N}} (1 - \lambda) T_1, \quad (7)$$

where $[f(x)]^+ = \max\{0, f(x)\}$, the same below. T_j is actual completion time of the jth customer order which consists of three parts, that is, completion time of mass process $T_1 + T_1^{\text{ext}}$, completion time of customized process $T_{2j} + T_{2j}^{\text{ext}}$, and increased order preparation time caused by new order insertion $([\sum_{j=1}^{J_0+1} N_j - \overline{N}]^+ / \overline{N})(1 - \lambda) T_1$.

4.1.2. Objective 2: To Maximize the Average Satisfaction Degree of All the Providers. FLSP's satisfaction degree is quite hard to be quantized in reality, but it is very important in scheduling. Here, two aspects are chosen to measure FLSP's satisfaction degree, which are the product of quantity satisfaction degree of service capacity and service time satisfaction degree [3].

(1) FLSP's Satisfaction Degree of Mass Process. (1) Quantity satisfaction degree of service capacity $S_{\text{quantity},1}$ reflects FLSP's utilization status in terms of service quantity in mass service process. When order volume is less than the upper service capacity limit, the bigger the utilization of service capacity is, the more satisfied the provider is. But when order volume exceeds the upper limit of service capacity, satisfaction degree will decrease because of the overload operation status. According to Assumption 4 and Liu et al. [2], FLSP's satisfaction degree of mass process $S_{\text{quantity},1}$ can be presented as follows:

$$S_{\text{quantity},1} = \begin{cases} S_{\text{quantity},1}^0 + \dfrac{1}{\overline{N}} \sum_{j=1}^{J_0+1} N_j \left(1 - S_{\text{quantity},1}^0 \right), & \text{when } 0 < \sum_{j=1}^{J_0+1} N_j \le \overline{N} \\ \dfrac{\overline{N}}{\sum_{j=1}^{J_0+1} N_j}, & \text{when } \sum_{j=1}^{J_0+1} N_j > \overline{N}, \end{cases} \quad (8)$$

where $S_{\text{quantity},1}^0$ means the initial satisfaction degree of provider in mass process when order volume is more than 0. It differs with different providers. $\overline{N}$ is the upper limit of FLSP's normal capacity in mass process.

(2) Service time satisfaction degree $S_{\text{time},1}$ reflects the satisfaction degree of provider for the service time schedule made by LSI. Generally speaking, when providers are operating as the schedule appointed in advance, their satisfaction degree is the highest. If LSI asks them to compress or delay their completion time suddenly, indeed, providers will become less satisfied. Therefore, the degree of closeness between actual completion time and normal operation time is used to denote FLSP's service time satisfaction degree. Consider

$$S_{\text{time},1} = \begin{cases} \dfrac{T_1}{T_1 + T_1^{\text{ext}}}, & T_1^{\text{ext}} \ge 0 \\ \dfrac{T_1 + T_1^{\text{ext}}}{T_1}, & T_1^{\text{ext}} < 0. \end{cases} \quad (9)$$

Thus, FLSP's satisfaction degree of mass process is shown as $S_1 = S_{\text{time},1} \times S_{\text{quantity},1}$.

(2) FLSP's Satisfaction Degree of Customized Process. According to Assumption 4, for customized process, operation volume of original orders is not affected by new order insertion. Thus, it is unnecessary to redeploy capacity. The jth FLSP's

satisfaction degree is only related to service time factor. Consider

$$S_{2j} = S_{\text{time},2j} \begin{cases} \dfrac{T_{2j}}{T_{2j} + T_{2j}^{\text{ext}}}, & \text{when } T_{2j}^{\text{ext}} \geq 0 \\ \dfrac{T_{2j} + T_{2j}^{\text{ext}}}{T_{2j}}, & \text{when } T_{2j}^{\text{ext}} < 0, \end{cases} \quad j = 1, 2, \ldots, J_0 + 1. \tag{10}$$

With the satisfaction time of mass and customized process integrated, the average satisfaction degree for all providers could be calculated as

$$\text{Max } Z_2 = \frac{S_1 + \sum_{j=1}^{J_0+1} S_{2j}}{1 + J_0 + 1} = \frac{S_{\text{time},1} S_{\text{quantity},1} + \sum_{j=1}^{J_0+1} S_{2j}}{1 + J_0 + 1}. \tag{11}$$

4.2. Constraints of the Scheduling Model

4.2.1. Constraint 1: To Meet Customers' Time Requirement. It is required that each customer order's completion time cannot be longer than the upper limit $T_j^{\text{exp}}(1 + \beta_j)$ set by the corresponding customer. Consider

$$\begin{aligned} T_j = T_1 + T_1^{\text{ext}} + T_{2j} + T_{2j}^{\text{ext}} + \frac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N}\right]^+}{\overline{N}} \lambda T_1 \\ \leq T_j^{\text{exp}} \left(1 + \beta_j\right), \end{aligned} \tag{12}$$

where β_j indicates the delay coefficient of the order completion time permitted by the jth customer for its order.

4.2.2. Constraint 2: LSI's Increased Profit Resulted by New Order Insertion Is Larger than 0. This constraint shows the necessary condition that LSI is willing to carry out order insertion decision. In other words, the price paid by customer for its inserted order must exceed order insertion cost of LSI. Then constraint 2 can be presented as follows:

$$\begin{aligned} \Delta\text{pro} = N_{j+1} F - \Bigg[& \left(f_1 + f_2 + f_3\right) \\ & - \left(C_1 T_1 \sum_{j=1}^{J_0} N_j + \sum_{j=1}^{J_0} \left(C_{2j} T_{2j} N_j \right) \right) \Bigg] > 0, \end{aligned} \tag{13}$$

where $(f_1 + f_2 + f_3)$ stands for the total cost of all the orders after new order inserted into original ones.

4.2.3. Constraint 3: Each FLSP's Satisfaction Degree Is Larger than Its Lower Limit. Consider

$$\begin{aligned} S_1 &= S_{\text{time},1} S_{\text{quantity},1} \geq S_1^L, \\ S_{2j} &= S_{\text{time},2j} \geq S_{2j}^L. \end{aligned} \tag{14}$$

4.2.4. Constraint 4: The Upper Limit of Capacity in Mass Process. According to **Assumption 4**, due to the existence of capacity constraint in mass process, it is impossible to increase new inserted order's volume infinitely. Here we set the new inserted order's volume as not more than k times of upper limit of FLSP's normal capacity in mass process. Please see the following formula:

$$N_{J_0+1} \leq k\overline{N}. \tag{15}$$

Besides, in actual scheduling process, a FLSP's compressed time cannot be longer than the normal operation time itself; namely, $T_1^{\text{ext}} + T_1 > 0$, $T_{2j}^{\text{ext}} + T_{2j} > 0$ should be fulfilled.

Based on the optimization objectives and constraints above, the whole model established in this paper is as follows:

$$\begin{aligned} \text{Min} \quad & Z_1 = \frac{1}{\sum_{j=1}^{J_0+1} N_j} \left(f_1 + f_2 + f_3\right) \\ \text{Max} \quad & Z_2 = \frac{S_1 + \sum_{j=1}^{J_0+1} S_{2j}}{1 + J_0 + 1} \\ & \quad = \frac{S_{\text{time},1} S_{\text{quantity},1} + \sum_{j=1}^{J_0+1} S_{2j}}{1 + J_0 + 1} \\ \text{subject to} \quad & T_1 + T_1^{\text{ext}} + T_{2j} + T_{2j}^{\text{ext}} \\ & \quad + \frac{\left[\sum_{j=1}^{J_0+1} N_j - \overline{N}\right]^+}{\overline{N}} \lambda T_1 \leq T_j^{\text{exp}} \left(1 + \beta_j\right) \\ & \Delta\text{pro} = N_{j+1} F - \Bigg[\left(f_1 + f_2 + f_3\right) \\ & \qquad - \left(C_1 T_1 \sum_{j=1}^{J_0} N_j \right. \\ & \qquad \left. + \sum_{j=1}^{J_0} \left(C_{2j} T_{2j} N_j \right) \right) \Bigg] > 0 \\ & S_1 \geq S_1^L, \qquad S_{2j} \geq S_{2j}^L, \\ & N_{J_0+1} \leq k \sum_{j=1}^{J_0} N_j, \qquad T_1^{\text{ext}} + T_1 > 0 \\ & T_{2j}^{\text{ext}} + T_{2j} > 0, \quad j = 1, 2, \ldots, J_0 + 1. \end{aligned} \tag{16}$$

5. Model Solution

5.1. Simplifying the Multiobjective Programming Model. The LSSC order insertion scheduling model has two objectives and seven constraints. It is a typical multiobjective programming problem. In this paper, the typical linear weighting method is chosen to solve our model. Objective Z_1 should dimensionally be transformed into a number in the range of $[0, 1]$. After the mathematical transformation, the synthesized objective function is shown as follows:

$$\max Z = w_1 \frac{Z_1^{\min}}{Z_1} + w_2 Z_2, \tag{17}$$

Table 2: Basic data (1).

Parameter	T_1	C_1	C_1^{ext}	$\overline{N}$	$S^0_{\text{quantity},1}$	$S^L_{\text{quantity},1}$	F	λ_2	λ_3	k
Value	29	10	18	110	0.3	0.5	3000	0.6	0.7	3

where w_1, w_2 represent the weights of Z_1 and Z_2, respectively. $w_1 \geq 0$, $w_2 \geq 0$, and $w_1 + w_2 = 1$. $Z_1^{\min}$ is the minimum of Z_1 when not considering other objective functions. Z is also called the comprehensive performance objective of LSSC.

5.2. Using the Genetic Algorithm to Solve the Model. The genetic algorithm is an effective method used to search for the optimal solution by simulating the natural selection process. As it uses multiple starting points to begin the search, it has a satisfactory global search capacity. For the combinatorial optimization problem, the genetic algorithm is quite effective to the solve NP problem, such as the production scheduling problem [31], travelling salesman problem [32], knapsack problem [33], and bin packing problem.

In this paper, instead of comparing or selecting a best method among different kinds of solution methods, we just choose an appropriate method. Given the superiority of the genetic algorithm in solving programming problems and the successful application to scheduling problems [31], this paper uses the genetic algorithm to solve the proposed model.

6. Numerical Analysis

By conducting a numerical analysis, this section illustrates the validity of model, explores the influence of relevant parameters on the scheduling results and further gives some effective recommendations for supply chain scheduling and optimization. Section 6.1 presents the basic data of the numerical example. Section 6.2 shows the scheduling results. Section 6.3 discusses the influence of the time delay coefficient β_j of order completion on the scheduling results of the LSSC. Section 6.4 presents the influence of the new inserted order's volume N_{j_0+1} on order insertion decision. Section 6.5 presents the influence of β_j on $N^{\max}_{j_0+1}$. Section 6.6 shows the influence of $\overline{N}$ on $N^{\max}_{j_0+1}$.

6.1. Numerical Example Description and Basic Data. The parameter values used in our model are shown in Tables 2 and 3.

6.2. Numerical Example Results. Genetic algorithm is adopted to solve the problem. It is assumed that the genetic population should be 800 and the hereditary algebra should be 800. And the program for our model is written within MATLAB 7.8 software and run on a PC with 1.6 GHz quad-core processor and 4 GB memories. Computer system is windows 7.0. Let $w_1 = w_2 = 0.5$ and based on the data in Tables 2 and 3, the calculation result is as follows.

The optimal solution is $Z = 0.9627$ and the corresponding scheduling results are as follows.

Mass service operation stage: $T_1^{\text{ext}} = -3.0044$.

Table 3: Basic data (2).

Parameter	$j = 1$	$j = 2$	$j = 3$	$j = 4$ (new inserted order)
T_{2j}	24	31	36.5	30
T_j^{exp}	53	60	65.5	55
C_j^{delay}	19	22	17	20
N_j	20	30	40	40
β_j	0	0	0	0
C_{2j}	28	22	37	32
C_{2j}^{ext}	35	41	44	35

Customized operation stage:

$$\begin{aligned}&\left[T_{21}^{\text{ext}} \;\; T_{22}^{\text{ext}} \;\; T_{23}^{\text{ext}} \;\; T_{24}^{\text{ext}}\right]\\&\quad = \left[-0.0043 \;\; 0.0012 \;\; 0.0018 \;\; -3.9113\right].\end{aligned} \tag{18}$$

According to the calculation results above, it is found that operation time in mass process needs to be compressed when order inserted, and the compressed time is 3.0044 units. Among customized processes of these four orders, the first customer order and the fourth customer order need to be operated in time compressed status, and the second and third customer order need to be delayed a little.

6.3. Effects of β_j on the Scheduling Performance of the LSSC. Generally speaking, customers' requirement for a service order's completion time may change, and time compression and delay requirement are both possible, which demands a certain degree of time flexibility in scheduling from the LSI. In model building, $\beta_j < 0$ means that the service order needs to be finished ahead of time; accordingly, $\beta_j > 0$ means that the service time needs to be delayed. In this section, the influence of the delay (or compression) coefficient of order completion time β_j on Z is discussed. With other model parameters unchanged, the results of Z are calculated corresponding to the changing β_j. For the convenience of calculation, let all β_j be the same value; namely, the time delay coefficient of order completion β of all the customer orders are the same. The results are shown in Table 4.

With the data in Table 4 plotted, Figure 4 is obtained.

Based on Table 4 and Figure 4, the following conclusions could be obtained.

(1) With the increase of β_j (from negative to positive), Z first increases and then tends to be stable, which means that a reasonable positive tolerance coefficient contributes to achieving the maximal value of comprehensive performance (i.e., in this numerical example, when $\beta_j = 0.2$, comprehensive performance reaches the maximum $Z = 0.9846$). Conversely, if β_j is negative, the maximal value of comprehensive performance cannot be reached. Moreover, a smaller time delay tolerance coefficient (i.e., the service

Table 4: The influence of β_j on comprehensive performance of LSSC Z.

β_j	Z
$[-0.4\ \ -0.4\ \ -0.4\ \ -0.4]$	No solution
$[-0.3\ \ -0.3\ \ -0.3\ \ -0.3]$	0.8243
$[-0.2\ \ -0.2\ \ -0.2\ \ -0.2]$	0.9022
$[-0.1\ \ -0.1\ \ -0.1\ \ -0.1]$	0.9463
$[\mathbf{0\ \ 0\ \ 0\ \ 0}]$	**0.9627**
$[0.1\ \ 0.1\ \ 0.1\ \ 0.1]$	0.9793
$[0.2\ \ 0.2\ \ 0.2\ \ 0.2]$	0.9846
$[0.3\ \ 0.3\ \ 0.3\ \ 0.3]$	0.9846

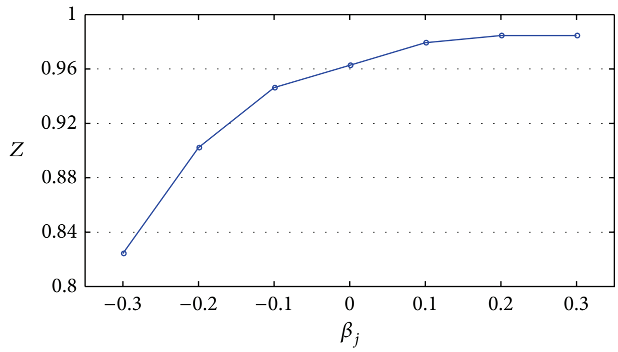


Figure 4: Curve of Z changed with β_j.

should be operated in time compression) results in poorer comprehensive performance. Therefore, it could be inferred that comprehensive performance may deteriorate when customers request shortening the order completion time of FLSP.

(2) If $\beta_j < -0.3$, the model has no solution, which means that the LSSC cannot operate in time compression without limit. Furthermore, the LSSC scheduling has certain restriction, and the order cannot be completed as early as the customer wants it.

(3) After β_j reaches a certain level (in this example, it is $\beta_j > 0.2$), Z tends to be stable. It has no contribution to improve the total performance of supply chain if LSSC continues to increase β_j. Therefore, in practice, it makes no sense to blindly negotiate with customer to reach the biggest value of β_j.

6.4. Effects of N_{j_0+1} on the Order Insertion Decision. It is easy to understand that the order insertion decision is affected by the volume of new inserted order, which is denoted by N_{j_0+1}. In Section 6.4, the effect of N_{j_0+1} on the order insertion decision is discussed in detail. Keep other parameters unchanged, just change new inserted order's volume N_{j_0+1} and try to find solution to our model. If solution exists, then calculate the corresponding value of Z and go on to increase N_{j_0+1} until model has no solution. The calculation result is shown in Table 5.

Table 5: Effect of N_{j_0+1} on comprehensive performance of LSSC Z.

N_{j_0+1}	Max Z	Z_1	Z_2
1	0.9743	$1.2690e+003$	0.9491
5	0.9769	$1.2737e+003$	0.9540
10	0.9802	$1.2795e+003$	0.9605
20	0.9865	$1.2896e+003$	0.9732
30	0.9734	$1.3247e+003$	0.9472
40	0.9627	$1.3580e+003$	0.9256
50	0.9521	$1.3940e+003$	0.9079
60	0.9446	$1.4253e+003$	0.8930
70	0.9384	$1.4516e+003$	0.8777
80	0.9089	$1.5218e+003$	0.8180
90	0.8720	$1.6059e+003$	0.7440
100	0.8348	$1.6890e+003$	0.6698
110	0.7977	$1.7709e+003$	0.5955
112	0.7902	$1.7874e+003$	0.5805
113	No solution		

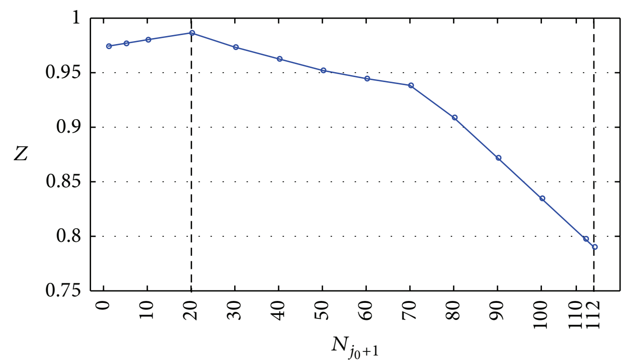


Figure 5: Curve of Z varied with N_{j_0+1}.

With the data in Table 5 plotted, Figure 5 is obtained.

Figure 5 and Table 5 indicate the following conclusions.

(1) Along with the increase of N_{j_0+1}, comprehensive performance of LSSC Z first increases and then tends to be stable. The inflection point of the curve occurs at the point whose value is the difference between normal operation capacity of mass process and the volume of original orders, which is called capacity surplus of mass process in this paper. Therefore, supply chain performance reaches the optimal when the new inserted order's volume is equal to the capacity surplus of mass process. It is easy to understand that in the situation above, new inserted order can be operated together with original orders without increase of extra order preparation time. Moreover, FLSP's normal operation capacity, such as the maximum loading capacity of truck or the maximum capacity of warehouse, is fully utilized in this situation. Hence, FLSP's satisfaction degree of service time and service quantity are both in high level. In this numerical example, it is reflected by the maximum value of $Z_2 = 0.9732$.

(2) Along with the continuous increase of N_{j_0+1} (here $N_{j_0+1} > 20$), comprehensive performance of LSSC Z decreases due to two reasons. On the one hand, with the

Table 6: Effect of β_j on $N_{j_0+1}^{\max}$.

β_j	$N_4^{\max}$	Compared with benchmark value, the growth proportion of $N_4^{\max}$
$[-0.3\ \ -0.3\ \ -0.3\ \ -0.3]$	52	−53.6%
$[-0.2\ \ -0.2\ \ -0.2\ \ -0.2]$	72	−35.7%
$[-0.1\ \ -0.1\ \ -0.1\ \ -0.1]$	92	−17.9%
Benchmark $[0\ \ 0\ \ 0\ \ 0]$	**112**	—
$[0.1\ \ 0.1\ \ 0.1\ \ 0.1]$	130	16.1%
$[0.2\ \ 0.2\ \ 0.2\ \ 0.2]$	130	16.1%
$[0.3\ \ 0.3\ \ 0.3\ \ 0.3]$	130	16.1%

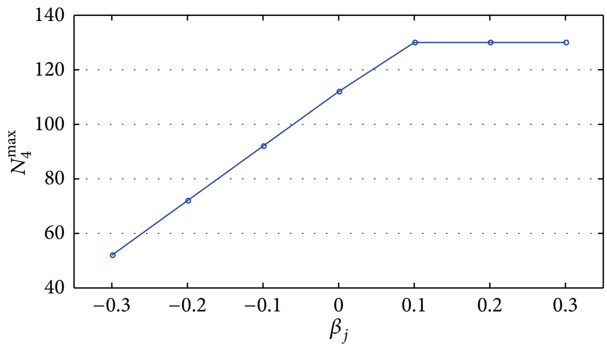


Figure 6: Curve of $N_4^{\max}$ varied with β_j.

increase of N_{j_0+1}, original orders' preparation time of mass process increases and resulted in time compression operation in each process. Apparently, extra cost will increase accordingly as well as Z_1. On the other hand, FLSP's satisfaction degree Z_2 will be decreased when their operating order volume exceeds the original schedule. Considering these two factors together, comprehensive performance of LSSC Z decreases gradually.

(3) Along with the continuous increase of N_{j_0+1} (which is more than 113 units in our example), supply chain cannot operate anymore. In other words, new order cannot be inserted.

6.5. Effects of β_j on the Upper Limit of New Order's Insertable Volume $N_{j_0+1}^{\max}$. In our numerical example, there are three original orders and one new inserted order. Thus, $N_{j_0+1}^{\max}$ can be replaced by $N_4^{\max}$. Keep other parameters unchanged and change the value of β_j. In each value of β_j, only change the value of N_{j+1} and calculate the upper limit of new order's insertable volume N_4 (which is denoted as $N_4^{\max}$). As described above, set the benchmark value $\beta_j = [0\ \ 0\ \ 0\ \ 0]$ and calculate the corresponding $N_4^{\max}$ when β_j is taken different values. Then the results are shown in Table 6.

With the data in Table 6 plotted, Figure 6 is obtained.

According to Figure 6, it is found that β_j has significant influence on new order's insertable volume $N_{j_0+1}^{\max}$. From the view of the overall trend, along with the increase of β_j, new

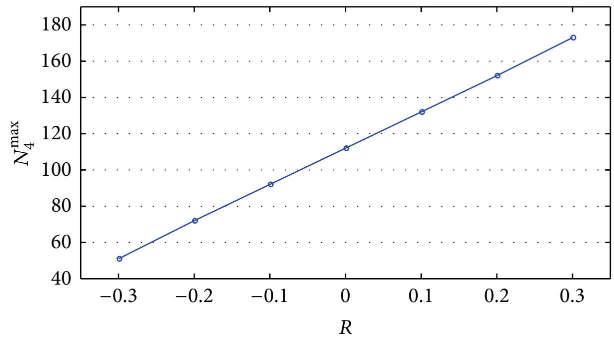


Figure 7: Curve of $N_4^{\max}$ varied with R.

order's insertable volume $N_4^{\max}$ increases and tends to be stable.

6.6. Effects of $\overline{N}$ on $N_{j_0+1}^{\max}$. In this section, the effect of upper limit of normal operation capacity of mass process $\overline{N}$ on $N_{j_0+1}^{\max}$ is explored. Keep other parameters unchanged and set $\overline{N}_0 = 110$ as benchmark. And R is used to be denoted as adjustment coefficient of normal operation capacity of mass process. In calculation, $\overline{N}$ can be presented as $\overline{N} = \overline{N}_0 \times (1 + R)$, $R \in (-1, +\infty)$. Then change the value of R and calculate corresponding upper limit of N_{j+1}, which is denoted by $N_{j+1}^{\max}$. This upper limit is the upper limit of insertable volume. Results are shown in Table 7. Basic data $\overline{N} = \overline{N}_0 = 110$ and corresponding $N_4^{\max} = 112$, which is benchmark value.

With the data in Table 7 plotted, Figure 7 could be obtained.

According to Figure 7, the following conclusions could be made.

(1) $\overline{N}$ has significant influence on new order's insertable volume $N_{j_0+1}^{\max}$. From the view of the overall trend, new order's insertable volume $N_{j_0+1}^{\max}$ increases along with the increase of $\overline{N}$.

(2) See from the view of quantitative relation, the increasing (or decreasing) proportion of new order's insertable volume $N_{j_0+1}^{\max}$ is larger than that of normal operation capacity of mass process $\overline{N}$. As shown in Table 7, in this example, if the adjustment coefficient increases (or decreases) 0.1 time based on the benchmark, namely, increasing (or decreasing) $112 \times 0.1 = 11.2$ units, absolute value of the increase (or decrease) in $N_4^{\max}$ is approximately 20 units, compared to benchmark value. The latter number can be calculated by the subtraction between the former item and the latter item in second column of Table 7. Therefore, intuitively, if $\overline{N}$ increases per unit, the $N_{j_0+1}^{\max}$ increases more than one unit. In this example, the number is approximately $20 \div 11.2 = 1.79$ units. In consequence, it can contribute to inserting relatively more extra orders for customers to choose a supply chain whose upper limit of normal operation capacity of mass process $\overline{N}$ is relatively large. Furthermore, for LSI, increasing $\overline{N}$ significantly significantly contributes to improving its order insertion capacity.

Table 7: Effect of $\overline{N}$ on $N_{j_0+1}^{\max}$.

Adjustment coefficient R	$N_4^{\max}$	Compared with benchmark value, the growth proportion of $N_4^{\max}$	Z	Z_1	Z_2
−0.3	51	−54.5%	0.7925	1.7713e + 003	0.5849
−0.2	72	−35.7%	0.7885	1.7849e + 003	0.5770
−0.1	92	−17.9%	0.7896	1.7861e + 003	0.5791
0 (benchmark)	**112**	—	**0.7902**	**1.7874e + 003**	**0.5805**
0.1	132	17.9%	0.7910	1.7881e + 003	0.5821
0.2	152	35.8%	0.7916	1.7889e + 003	0.5831
0.3	173	54.5%	0.7892	1.7956e + 003	0.5784

(3) Supply chain comprehensive performance Z is rarely affected by R and remains stable. It is found that when operating in capacity limiting conditions (namely, when the new inserted order's volume is the upper limit that supply chain can support), supply chain performance is almost the same. No matter if the upper limit of normal operation capacity of mass process is large or small, there is not big difference in overall performance. This conclusion is opposite to what we guessed and thus very interesting. Generally, it is usually guessed that a supply chain with larger operation capacity in mass process has greater performance when operating in capacity limit conditions. Obviously, any supply chain will show a relatively bad performance when operating in capacity limit conditions, since unit cost is high and FLSP's satisfaction degree is low.

(4) Combined with conclusions in Section 6.5, it is found that both $\overline{N}$ and β_j will significantly influence the maximum volume of insertable order. By comparison, improving the normal operation capacity of mass service process $\overline{N}$ is more useful in increasing maximum volume of insertable order. The reason is that after reaching a certain level (in our example it is 0.1 time of normal completion time), continuous increase in β_j makes no contribution in increasing insertable order volume. However, even if $\overline{N}$ increases to 0.3 time of benchmark, it still makes contribution to increasing maximum insertable order volume. This conclusion is relatively useful for LSI.

7. Main Conclusions and Management Insights

This section summarizes main conclusions and further explains related insights for researchers. And management insights for LSI are also discussed, which offers useful recommendations for scheduling decisions.

7.1. Main Conclusions Derived from the Scheduling Model. The following conclusions are based on the previous analysis.

(1) On the one hand, the smaller the time delay coefficient β_j of order completion is, the worse the supply chain performance will be. When β_j is less than a certain value, this scheduling model has no solution, which indicates operation time could not be compressed infinitely. On the other hand, if customers permit completion time delay, increase in β_j could improve supply chain comprehensive performance. However, supply chain performance will stop improving but remain stable after increasing to a certain level. Thus, it makes no sense to negotiate with customer blindly for the biggest value of β_j in practice.

(2) The delay coefficient of order completion time β_j permitted by customer obviously influences insertable order volume. Generally, along with the increase of β_j, order insertable volume gradually increases and tends to be stable after reaching a certain level.

(3) With inserted order's volume increasing, the comprehensive performance of LSSC Z first increases and then decreases. The curve of Z inflects at the point representing the difference between normal operation capacity of mass process and the volume of original orders, which is called capacity surplus of mass process in this paper. Therefore, supply chain performs best when the new inserted order's volume is equal to the capacity surplus of mass process. With N_{j_0+1} continuously increasing, the comprehensive performance of LSSC Z decreases. Furthermore, supply chain cannot operate anymore after new inserted order's volume reaches a certain level.

(4) The upper limit of normal operation capacity $\overline{N}$ has significant influence on new order's insertable volume $N_{j_0+1}^{\max}$. Generally, along with the increase of $\overline{N}$, new order's insertable volume $N_{j_0+1}^{\max}$ increases. Seen from the view of quantitative relation, the increasing (or decreasing) proportion of new order's insertable volume $N_{j_0+1}^{\max}$ is larger than that of normal operation capacity of mass process $\overline{N}$; that is $N_{j_0+1}^{\max}$ increases more than one unit when $\overline{N}$ increases one unit. Therefore, it is useful for customers to choose a supply chain whose has large normal operation capacity of mass process when it is expected to insert relatively more extra orders. Furthermore, it is quite effective to increase $\overline{N}$ when LSI plans to improve its capacity in order insertion.

(5) Both $\overline{N}$ and β_j will significantly influence the maximum volume of insertable order. With comparison, improving the normal operation capacity of mass service process $\overline{N}$ is more useful in increasing maximum volume of insertable order.

7.2. Implications for Researchers. This study establishes the LSSC order insertion model considering capacity and time factors and analyzes the order insertion problem in the MCLS environment, which could be referred to by other researchers. First, this study provides theoretical basis for further studies on the scheduling methods and performance optimization methods of LSSCs in the MCLS environment. For example, it is found that both the order completion time delay coefficient permitted by customer and the volume of new inserted order have influence on supply chain comprehensive performance and will further affect the order insertion decisions. Although both the normal operation capacity of mass process and the delay coefficient of order completion time permitted by customer will significantly influence the maximum volume of insertable order, improving the former one is more useful in increasing the maximum volume of insertable order. These conclusions could be useful for further studies on order insertion scheduling models. Second, the order similarity coefficient proposed by this paper provides reference for other researches on supply chain order insertion model. Third, researchers could develop integrated study on order insertion decision and CODP based on our model, and empirical research on that issue could also be conducted. In short, this study could offer a basic theoretical foundation for further studies on LSSC scheduling.

7.3. Implications for Managers. This research is developed on the background of MCLS, and the conclusions presented in this paper could serve as reference for the participants in LSSC, especially LSI. Specifically, three important points are shown as follows.

(1) For customers, it is useful to choose a supply chain whose normal operation capacity of mass process is relatively large for inserting relatively more extra orders. Thus, LSI should make efforts to improve their service capacity in mass process to face the challenge from newly increased order's demand.

(2) Supply chain performance reaches the optimal when the new inserted order's volume is equal to the capacity surplus of mass process. Besides, when a certain level is achieved, new order cannot be inserted and supply chain operation breaks down. Hence, it is sensible for LSI to choose the new order whose volume matches the capacity surplus of mass process to reach optimal supply chain performance.

(3) The insertable volume of new inserted order is affected by both order completion time requirements from customers and FLSP's normal operation capacity in mass process. And to increase service capacity in mass process is more useful for improving order insertion capacity. Therefore, LSI had better enhance the operation capacity instead of asking customer's permission for delaying completion time.

7.4. Research Limitations and Directions for Future Research. With full consideration of service capacity and time factor, an order insertion scheduling model of LSSC is established, aiming to minimize the average unit volume operation cost of the LSI and maximize the average satisfaction degree of FLSPs. And in order to verify the viability and effectiveness of our model, a specific example is numerically analyzed with MATLAB 7.8 software. Furthermore, effects that the order completion time delay coefficient permitted by customer and the new inserted order's volume have on supply chain comprehensive performance are discussed, as well as effects that the new inserted order's volume and the upper limit of normal operation capacity in mass process have on order insertion decisions. Many useful conclusions are obtained to improve LSI's time scheduling decision. However, this paper has several limitations. For example, the model solution and analysis are obtained with a numerical example, which may not represent all situations in reality. Besides, the influences of order insertion scheduling on CODP is not considered in our model. In practice, insertion of new order may cause CODP changing, which could be researched in future work. What is more, in our model, we assume that there is only one new arrived order that needs to be inserted and do not consider multiorder insertion problem. In the future, the multiorder insertion problem could be explored based on the order priority.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant no. 71372156), is supported by Humanity and Social Science Youth foundation of Ministry of Education of China (Grant no. 2013YJC630098), and is sponsored by China State Scholarship Fund (Grant no. 201308120087) and Independent Innovation Foundation of Tianjin University. The reviewers' comments are also highly appreciated.

References

[1] C. Chandra and J. Grabis, "Managing logistics for mass customization: the new production frontier," in *Proceedings of the 6th Biannual World Automation Congress—Image Processing, Biomedicine, Multimedia, Financial Engineering and Manufacturing—International Forum on Multimedia Image Processing (WAC '04)*, pp. 335–340, July 2004.

[2] W. H. Liu, X. C. Xu, Z. X. Ren, and Y. Peng, "An emergency order allocation model based on multi-provider in two-echelon logistics service supply chain," *Supply Chain Management*, vol. 16, no. 6, pp. 391–400, 2011.

[3] W. H. Liu, Y. Yang, X. Li, H. T. Xu, and D. Xie, "A time scheduling model of logistics service supply chain with mass customized logistics service," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 482978, 18 pages, 2012.

[4] J. V. Simons Jr., G. R. Russell, and M. E. Kraus, "Service case scheduling: an exploratory description and recommendations for solution," *International Journal of Service Industry Management*, vol. 15, no. 4, pp. 385–399, 2004.

[5] W. H. Liu, H. T. Xu, X. Y. Sun, Y. Yang, and Y. M. Mo, "Order allocation research of logistics service supply chain with mass customization logistics service," *Mathematical Problems in Engineering*, vol. 2013, Article ID 957260, 13 pages, 2013.

[6] W. H. Liu, Y. Yang, H. T. Xu, X. Y. Liu, Y. J. Wang, and Z. C. Liang, "A time scheduling model of logistics service supply chain based on the customer order decoupling point: a perspective from the constant service operation time," *ArticleThe Scientific World Journal*, vol. 2014, Article ID 756178, 22 pages, 2014.

[7] B. J. Pine, "Mass customizing products and services," *Strategy & Leadership*, vol. 21, no. 4, pp. 6–55, 1993.

[8] F. S. Fogliatto, G. J. C. da Silveira, and D. Borenstein, "The mass customization decade: an updated review of the literature," *International Journal of Production Economics*, vol. 138, no. 1, pp. 14–25, 2012.

[9] D. Yang, M. Dong, and X. Chang, "A dynamic constraint satisfaction approach for configuring structural products under mass customization," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1723–1737, 2012.

[10] X. Shao, "Integrated product and channel decision in mass customization," *IEEE Transactions on Engineering Management*, vol. 60, no. 1, pp. 30–45, 2013.

[11] H. Wang, "Defects tracking in mass customisation production using defects tracking matrix combined with principal component analysis," *International Journal of Production Research*, vol. 51, no. 6, pp. 1852–1868, 2013.

[12] D. Li, F. M. Chang, and S. Chang, "The relationship between affecting factors and mass-customisation level: the case of a pigment company in Taiwan," *International Journal of Production Research*, vol. 48, no. 18, pp. 5385–5395, 2010.

[13] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–674, 2003.

[14] C.-Y. Lee, S. Piramuthu, and Y.-K. Tsai, "Job shop scheduling with a genetic algorithm and machine learning," *International Journal of Production Research*, vol. 35, no. 4, pp. 1171–1191, 1997.

[15] Z. L. Chen and N. G. Hall, "Supply chain scheduling: conflict and cooperation in assembly systems," *Operations Research*, vol. 55, no. 6, pp. 1072–1089, 2007.

[16] J. M. Yao, "Scheduling optimisation of co-operator selection and task allocation in mass customisation supply chain based on collaborative benefits and risks," *International Journal of Production Research*, vol. 51, no. 8, pp. 2219–2239, 2013.

[17] E. Selvarajah and G. Steiner, "Approximation algorithms for the supplier's supply chain scheduling problem to minimize delivery and inventory holding costs," *Operations Research*, vol. 57, no. 2, pp. 426–438, 2009.

[18] N. Mikati, "Dependence of lead time on batch size studied by a system dynamics model," *International Journal of Production Research*, vol. 48, no. 18, pp. 5523–5532, 2010.

[19] A. A. Taleizadeh, S. T. A. Niaki, N. Shafii, R. G. Meibodi, and A. Jabbarzadeh, "A particle swarm optimization approach for constraint joint single buyer-single vendor inventory problem with changeable lead time and (r,Q) policy in supply chain," *International Journal of Advanced Manufacturing Technology*, vol. 51, no. 9–12, pp. 1209–1223, 2010.

[20] Y. Wang, "Evaluating flexibility on order quantity and delivery lead time for a supply chain system," *International Journal of Systems Science*, vol. 39, no. 12, pp. 1193–1202, 2008.

[21] J. Yao and L. Liu, "Optimization analysis of supply chain scheduling in mass customization," *International Journal of Production Economics*, vol. 117, no. 1, pp. 197–211, 2009.

[22] D. Tang and J. Chen, "Identification of postponement point in service delivery process: a description model," in *Proceedings of the 6th International Conference on Service Systems and Service Management (ICSSSM '09)*, pp. 335–339, June 2009.

[23] C. Duron, M. A. O. Louly, and J.-M. Proth, "The one machine scheduling problem: insertion of a job under the real-time constraint," *European Journal of Operational Research*, vol. 199, no. 3, pp. 695–701, 2009.

[24] C. Duron, J. Proth, and Y. Wardi, "Insertion of a random task in a schedule: a real-time approach," *European Journal of Operational Research*, vol. 164, no. 1, pp. 52–63, 2005.

[25] X. F. Huang, B. Wu, and J. Wang, "Modeling and optimization of inserting order schedule of agile supply chain," in *Proceedings of the International Conference on Frontiers of Manufacturing Science and Measuring Technology (ICFMM '11)*, pp. 910–915, 2011.

[26] M. C. Gomes, A. P. Barbosa-Póvoa, and A. Q. Novais, "A discrete time reactive scheduling model for new order insertion in job shop, make-to-order industries," *International Journal of Production Research*, vol. 48, no. 24, pp. 7395–7422, 2010.

[27] M. Wang and M. Rosenshine, "Scheduling for a combination of made-to-stock and made-to-order jobs in a job shop," *International Journal of Production Research*, vol. 21, no. 5, pp. 607–616, 1983.

[28] H. Tsubone, Y. Ishikawa, and H. Yamamoto, "Production planning system for a combination of make-to-stock and make-to-order products," *International Journal of Production Research*, vol. 40, no. 18, pp. 4835–4851, 2002.

[29] Y. C. Wang, X. F. Yu, L. Shi, and Z. Y. Huang, "Isolation and manufacturing orders based on the combined research," *Machinery Design & Manufacture*, vol. 11, pp. 245–247, 2010 (Chinese).

[30] A. D. Brack and M. Benkenstein, "Responses to other similar customers in a service setting—analyzing the moderating role of perceived performance risk," *Journal of Services Marketing*, vol. 28, no. 2, pp. 138–146, 2014.

[31] G. Meja, C. Montoya, J. Cardona, and A. L. Castro, "Petri nets and genetic algorithms for complex manufacturing systems scheduling," *International Journal of Production Research*, vol. 50, no. 3, pp. 791–803, 2012.

[32] L. D. Giovanni, G. Massi, and F. Pezzella, "An adaptive genetic algorithm for large-size open stack problems," *International Journal of Production Research*, vol. 51, no. 3, pp. 682–697, 2013.

[33] H. Yang and S. Wang, "Solving the 0/1 knapsack problem using rough sets and genetic algorithms," *Journal of the Chinese Institute of Industrial Engineers*, vol. 28, no. 5, pp. 360–369, 2011.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 620150, 9 pages
http://dx.doi.org/10.1155/2014/620150

Research Article

Single Machine Scheduling and Due Date Assignment with Past-Sequence-Dependent Setup Time and Position-Dependent Processing Time

Chuan-Li Zhao,[1] Chou-Jung Hsu,[2] and Hua-Feng Hsu[2]

[1] *College of Mathematics and Systems Sciences, Shenyang Normal University, Shenyang, Liaoning 110034, China*
[2] *Department of Industrial Management, Nan Kai University of Technology, Nantou 542, Taiwan*

Correspondence should be addressed to Chou-Jung Hsu; jrsheu@nkut.edu.tw

Received 23 July 2014; Accepted 14 August 2014; Published 27 August 2014

Academic Editor: Dehua Xu

This paper considers single machine scheduling and due date assignment with setup time. The setup time is proportional to the length of the already processed jobs; that is, the setup time is past-sequence-dependent (p-s-d). It is assumed that a job's processing time depends on its position in a sequence. The objective functions include total earliness, the weighted number of tardy jobs, and the cost of due date assignment. We analyze these problems with two different due date assignment methods. We first consider the model with job-dependent position effects. For each case, by converting the problem to a series of assignment problems, we proved that the problems can be solved in $O(n^4)$ time. For the model with job-independent position effects, we proved that the problems can be solved in $O(n^3)$ time by providing a dynamic programming algorithm.

1. Introduction

In many realistic scheduling environments, a job's processing time may be depending on its position in the sequence [1]. Two well-known special cases of this stream of research are (i) *positional deterioration (aging effect)*, where the processing time of a job increases as a function of its position in a processing sequence and (ii) *learning effect*, where the processing time of a job decreases as a function of its position in a processing sequence. Biskup [2] and Cheng and Wang [3] independently introduced the learning concept to scheduling research. Other studies include Mosheiov and Sidney [4], Mosheiov [5, 6], Wu et al. [7, 8], and Yin et al. [9, 10]. Biskup [11] presented an updated survey of the results on scheduling problems with the learning effect. Mosheiov [6] first mentioned the *aging effect* in scheduling research. Other studies include Mosheiov [12], Kuo and Yang [13], Janiak and Rudek [14], Zhao and Tang [15], and Rustogi and Strusevich [16], among others. Moreover, some studies consider scheduling problems with general position-dependent processing time. Mosheiov [17] considered a scheduling problem with general position-dependent processing time. The polynomial algorithm is derived for makespan minimization on an m-machine proportionate flow shop. Zhao et al. [18] studied scheduling and due date assignment problem. They provided a unified model for solving the single machine problems with rejection and position-dependent processing time. Rustogi and Strusevich [19] presented a critical review of the known results for scheduling models with various positional effects.

Koulamas and Kyparisis [20] first introduced a scheduling problem with past-sequence-dependent (p-s-d) setup time. They assumed that the job setup time is proportional to the sum of processing time of all already scheduled jobs. It is proved that the standard single machine scheduling with p-s-d setup time can be solvable in polynomial time when the objectives are the makespan, the total completion time, and the total absolute differences in completion time, respectively. Wang [21] studied the single machine scheduling problems with time-dependent learning effect and p-s-d setup time considerations. He showed that the makespan minimization problem, the total completion time minimization problem, and the sum of the quadratic job completion time minimization problem can be solved in polynomial time, respectively. Yin et al. [22] considered a single machine scheduling model with p-s-d setup time and a general learning effect. They showed that the single machine scheduling problems to minimize the makespan and the sum of the kth power of completion time are polynomially solvable under the proposed model. Hsu et al. [23] presented a polynomial-time algorithm for an unrelated parallel machine scheduling problem with setup time and learning effects to minimize the total completion time. Lee [24] proposed a model with the deteriorating jobs, the learning effect, and the p-s-d setup time. He provided the optimal schedules for some single machine problems. Huang et al. [25] considered some single machine scheduling problems with general time-dependent deterioration, position-dependent learning, and p-s-d setup time. They proved that the makespan minimization problem, the total completion time minimization problem, and the sum of the μth power of job completion time minimization problem can be solved by the SPT rule.

Meeting due dates is one of the most important objectives in scheduling (Gordon et al. [26]). In some situations, the tardiness penalties depend on whether the jobs are tardy, rather than how late they are. In these cases, the number of tardy jobs should be minimized (Yin et al. [27]). Kahlbacher and Cheng [28] considered scheduling problems to minimize costs for earliness, due date assignment, and weighted number of tardy jobs. They presented nearly a full classification for the single and multiple machine models. Shabtay and Steiner [29] studied two single machine scheduling problems. The objectives are to minimize the sum of weighted earliness, tardiness, and due date assignment penalties and minimize the weighted number of tardy jobs and due date assignment costs, respectively. They proved that both problems are strongly NP-hard and give polynomial solutions for some important special cases. Koulamas [30] considered the second problem of Shabtay and Steiner [29]. He presented a faster algorithm for a due date assignment problem with tardy jobs. Gordon and Strusevich [31] addressed the problems of single machine scheduling and due date assignment problems in which a job's processing time depends on its position in a processing sequence. The objective functions include the cost of the due dates, the total cost of discarded jobs that cannot be completed by their due dates, and the total earliness of the scheduled jobs. They presented polynomial-time dynamic programming algorithms for solving problems with two due date assignment methods, provided that the processing time of the jobs is positionally deteriorating. Hsu et al. [32] extended part of the objective functions proposed by Gordon and Strusevich [31] to the positional weighted earliness penalty and showed that the problems remain solvable in polynomial time.

2. Problem Formulation and Preliminaries

This paper studies the single machine scheduling problems with simultaneous consideration of due date assignment, p-s-d setup time, and position-dependent processing time.

The problem can be described as follows.

A set $N = \{J_1, J_2, ..., J_n\}$ of n jobs has to be scheduled on a single machine. All jobs are available for processing at time zero and preemption is not permitted. Each job J_j has a basic processing time p_j. The actual processing time of job J_j, if scheduled in position r of a sequence, is given by

$$p_j^A = g(j, r)\, p_j, \tag{1}$$

where $g(j,1), g(j,2), \ldots, g(j,n)$ represent an array of job-dependent positional factors.

Each job $J_j \in N$ has to be assigned a due date d_j, by which it is desirable to complete that job. Given a schedule, denote the completion time of job J_j by C_j. Job J_j is called tardy if $C_j > d_j$, and it is called nontardy if $C_j \le d_j$. Let $U_j = 1$ if job J_j is tardy and let $U_j = 0$ if job J_j is nontardy. The earliness of J_j is defined as $E_j = d_j - C_j$, provided that $C_j \le d_j$. In all problems considered in this paper, the jobs in set N have to be split into two subsets denoted by N_E and N_T. We refer to the jobs in set N_E as "nontardy," while the jobs in set N_T are termed "tardy." A penalty β_j is paid for the tardy job $J_j \in N_T$. Given a schedule $\pi = [J_{[i]}, J_{[2]}, \ldots, J_{[n]}]$, we assumed that the p-s-d setup time of $J_{[j]}$ is given as Koulamas and Kyparisis [20] did, as follows:

$$s_{[j]} = \delta \sum_{i=1}^{j-1} p_{[i]}^A, \quad j = 2, 3, \ldots, n, \; s_{[1]} = 0, \tag{2}$$

where $\delta \ge 0$ is a normalizing constant.

The purpose is to determine the optimal due dates and the processing sequence such that the following function is minimized:

$$F(\mathbf{d}, \pi) = \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \varphi(\mathbf{d}), \tag{3}$$

where π is the sequence of jobs, α is the positive unit earliness cost, $\mathbf{d}$ is the vector of the assigned due dates, and $\varphi(\mathbf{d})$ denotes the cost of assigning the due dates that depends on a specific rule chosen for due date assignment. We denote the problem as

$$1 \left| p_j^A = p_j g(j, r), S_{\text{psd}} \right| \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \varphi(\mathbf{d}). \tag{4}$$

Most of the presented results hold for a general positional effect, that is, for any function $g(j, r)$ that depends on both position r and job J_j. For each individual model, there is a particular rule that defines $g(j, r)$ and explains how exactly the value of p_j changes, for example.

(i) Job-Dependent Learning Effect (Mosheiov and Sidney [4]). The actual processing time of a job J_j, if scheduled in position r of a sequence, is given by

$$p_j^A = p_j r^{a_j}, \tag{5}$$

where $a_j \le 0$ is a job-dependent learning parameter (include $a_j = a$ as a special case, i.e., $p_j^A = p_j r^a$, Biskup [2]).

(ii) Job-Dependent Aging Effect (Zhao and Tang [15]). The actual processing time of a job j, if scheduled in position r of a sequence, is given by

$$p_j^A = p_j r^{a_j}, \tag{6}$$

where $a_j \geq 0$ is a job-dependent aging parameter (include $a_j = a$ as a special case, i.e., $p_j^A = p_j r^a$, Moshiev [17]).

(iii) Positional Exponential Deterioration (Wang [33]). The actual processing time of a job j, if scheduled in position r of a sequence, is given by

$$p_j^A = p_j a^{r-1}, \tag{7}$$

where $a \geq 1$ is a given positive constant representing a rate of deterioration, which is common for all jobs.

We study our problem with the two most frequently used due date assignment methods.

(i) The Common Due Date Assignment Method (usually referred to as *CON*). Where all jobs are assigned the same due date, such that is $d_j = d$ for $j = 1, 2, \ldots, n$ and $d \geq 0$ is a decision variable.

(ii) The Slack Due Date Assignment Method (usually referred to as *SLK*). Where all jobs are given an equal flow allowance that reflects equal waiting time (i.e., equal slacks), such that is $d_j = p_j^A + q$ for $j = 1, 2, \ldots, n$ and $q \geq 0$ is a decision variable.

We first provide some lemmas.

Lemma 1 (Hardy et al. [34]). *Let there be two sequences of numbers x_i and y_i ($i = 1, 2, \ldots, n$). The sum $\sum_{i=1}^n x_i y_i$ of products of the corresponding elements is the least if the sequences are monotonically ordered in the opposite sense.*

It is not difficult to see that the following property is valid for both the variants of our problem.

Lemma 2. *There exists an optimal schedule in which the following properties hold: (1) all the jobs are processed consecutively without idle time and the first job starts at time 0 for both the variants of the problem; (2) all the nontardy jobs are processed before all the tardy jobs for both the variants of the problem.*

3. The *CON* Due Date Assignment Method

In the *CON* model, $d_j = d$ ($j = 1, 2, \ldots, n$). We choose γd as the cost function $\varphi(\mathbf{d})$, where γ is a positive constant. Thus, it follows from function (3) that our problem is to minimize the objective function:

$$F(\mathbf{d}, \pi) = \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d. \tag{8}$$

The problem denotes

$$1 \left| p_j^A = p_j g(j, r), S_{\text{psd}}, CON \right| \alpha \sum_{J_j \in N_E} E_j + \alpha \sum_{J_j \in N_T} \beta_j U_j + \gamma d. \tag{9}$$

Kahlbacher and Cheng [28] provide an $O(n^4)$ time algorithm for the problem $1|CON|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d$. In this section, we consider a generalization of the basic model with p-s-d setup times and position-dependent processing times. As a result of Lemma 2, we can restrict our attention to those schedules without idle times and search for the optimal schedule only among the schedules in which one of the jobs is on time.

Let $\pi = [J_{[i]}, J_{[2]}, \ldots, J_{[n]}]$; then

$$\begin{aligned}
C_{[1]} &= g([1], 1) p_{[1]}, \\
C_{[2]} &= C_{[1]} + s_{[2]} + g([2], 2) p_{[2]} \\
&= g([1], 1) p_{[1]} + \delta g([1], 1) p_{[1]} + g([2], 2) p_{[2]} \\
&= \sum_{k=1}^{2} [1 + (2-k)\delta] g([k], k) p_{[k]}, \\
&\vdots \\
C_{[j]} &= \sum_{k=1}^{j} [1 + (j-k)\delta] g([k], k) p_{[k]}, \\
&\vdots \\
C_{[n]} &= \sum_{k=1}^{n} [1 + (n-k)\delta] g([k], k) p_{[k]}. \\
\sum_{j=1}^{n} C_{[j]} &= \sum_{j=1}^{n} (n-j+1) \left(1 + \frac{\delta(n-j)}{2}\right) g([j], j) p_{[j]}.
\end{aligned} \tag{10}$$

Note that we need only to consider the schedule in which all the nontardy jobs are processed before all the tardy jobs.

Lemma 3. *For the problem*

$$1 \left| p_j^A = p_j g(j, r), S_{psd}, CON \right| \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d, \tag{11}$$

if the number of jobs in N_E is h (denote $|N_E| = h$), there exists an optimal schedule $\pi = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$, such that $d = C_{[h]}$.

Proof. Suppose $\pi = [J_{[i]}, J_{[2]}, \ldots, J_{[n]}]$ is an optimal schedule. Since jobs $J_{[i]}, J_{[2]}, \ldots, J_{[h]}$ are nontardy, then $d \geq C_{[h]}$. Let $\Delta = d - C_{[h]}$. Moving the due date Δ units of time to the left such that $d = C_{[h]}$, the objective value will be decreasing $(\alpha j + \gamma)\Delta$, which is nonnegative. This means we can find a schedule with $d = C_{[h]}$ that is at least as good as π. □

As a consequence of Lemma 3, we consider the schedule $\pi = [J_{[i]}, J_{[2]}, \ldots, J_{[n]}]$ with $d = C_{[h]}$.

Thus,

$$\begin{aligned}
F(\mathbf{d},\pi) &= \alpha \sum_{J_j\in N_E} E_j + \sum_{J_j\in N_T} \beta_j U_j + \gamma d \\
&= \alpha \sum_{j=1}^{h} E_{[j]} + \sum_{j=h+1}^{n} \beta_{[j]} + \gamma d \\
&= \alpha \sum_{j=1}^{h} \left[C_{[h]} - C_{[j]}\right] + \sum_{j=h+1}^{n} \beta_{[j]} + \gamma C_{[h]} \\
&= (\alpha h + \gamma)\left(\sum_{j=1}^{h} \left[1 + (h-j)\delta\right] g([j], j)\, p_{[j]}\right) \\
&\quad - \alpha \sum_{j=1}^{h} (h-j+1)\left(1 + \frac{1}{2}\delta(h-j)\right) g([j], j)\, p_{[j]} \\
&\quad + \sum_{j=h+1}^{n} \beta_{[j]} \\
&= \sum_{j=1}^{h} \left\{\alpha(j-1) + \frac{1}{2}\alpha\delta(h-j)(h+j-1) \right. \\
&\qquad \left. + \left[1 + \delta(h-j)\right]\gamma\right\} g([j], j)\, p_{[j]} \\
&\quad + \sum_{j=h+1}^{n} \beta_{[j]} \\
&= \sum_{j=1}^{h} w_j g([j], j)\, p_{[j]} + \sum_{j=h+1}^{n} \beta_{[j]},
\end{aligned} \tag{12}$$

where

$$\begin{aligned}
w_j = &\left\{\alpha(j-1) + \frac{1}{2}\alpha\delta(h-j)(h+j-1)\right. \\
&\left. + \left[1 + \delta(h-j)\right]\gamma\right\}, \quad j = 1, 2, \ldots, h.
\end{aligned} \tag{13}$$

Let $x_{j,r}$ be a binary variable such that $x_{j,r} = 1$ if job J_j is scheduled in the rth position and $x_{j,r} = 0$; otherwise, $j, r = 1, 2, \ldots, n$. From (12), we can formulate the problem with objective (8) as the following assignment problem $A1(h)$, which can be solved in $O(n^3)$ time:

$$\begin{aligned}
\text{Min} \quad & \sum_{j=1}^{n}\sum_{r=1}^{n} c_{j,r} x_{j,r} \\
\text{s.t.} \quad & \sum_{r=1}^{n} x_{j,r} = 1, \quad j = 1, 2, \ldots, n, \\
& \sum_{j=1}^{n} x_{j,r} = 1, \quad r = 1, 2, \ldots, n, \\
& x_{j,r} \in \{0, 1\}, \quad j = 1, 2, \ldots, n, \; r = 1, 2, \ldots, n,
\end{aligned} \tag{14}$$

where

$$\begin{aligned}
c_{j,r} &= \begin{cases} w_r g(j, r)\, p_j & \text{for } r = 1, 2, \ldots, h \\ \beta_j & \text{for } r = h+1,\ h+2, \ldots, n, \end{cases} \\
w_r &= \left\{\alpha(r-1) + \frac{1}{2}\alpha\delta(h-r)(h+r-1)\right. \\
&\quad \left. + \left[1 + \delta(h-r)\right]\gamma\right\}, \quad r = 1, 2, \ldots, h.
\end{aligned} \tag{15}$$

Note that $c_{j,r}$ is the cost of assigning job J_j $(j = 1, 2, \ldots, n)$ in the rth $(r = 1, 2, \ldots, n)$ position in the schedule.

In order to derive the optimal solution, we have to solve the above assignment problem $A1(h)$ for any $h = 1, 2, \ldots, n$. We summarize the results of the above analysis and present the following solution algorithm.

Algorithm 4.

Step 1. For $h = 0$ ($d = 0$, all the jobs are tardy), calculate $F(0) = \sum_{j=1}^{n} \beta_j$.

Step 2. For h from 1 to n, solve the assignment problem $A1(h)$ and calculate the corresponding objective value $F(h)$.

Step 3. The optimal value of the function F is equal to $\min\{F(h) \mid h = 0, 1, 2, \ldots, n\}$.

As a result, we obtain the following theorem.

Theorem 5. *Problem* $1|p_j^A = p_j g(j,r), S_{psd}, CON|\alpha \sum_{J_j\in N_E} E_j + \sum_{J_j\in N_T} \beta_j U_j + \gamma d$ *can be solved in* $O(n^4)$ *time.*

We demonstrate our approach using the following example.

Example 6. Consider the problem $1|p_j^A = p_j g(j,r), S_{psd}, CON|\alpha \sum_{J_j\in N_E} E_j + \sum_{J_j\in N_T} \beta_j U_j + \gamma d$.

Let $n = 5$, $h = 3$. The weights are $\alpha = 1$, $\gamma = 2$, and $\delta = 0.1$.

The processing times are $p_1 = 7$, $p_2 = 6$, $p_3 = 5$, $p_4 = 2$, and $p_5 = 1$.

The tardy penalties are $\beta_1 = 10$, $\beta_2 = 8$, $\beta_3 = 4$, $\beta_4 = 5$, and $\beta_5 = 6$.

The positional effects are

$$g(j, r) = \begin{bmatrix} 2 & 1 & 3 & 2 & 4 \\ 1 & 3 & 2 & 2 & 3 \\ 2 & 3 & 1 & 4 & 3 \\ 1 & 2 & 3 & 1 & 3 \\ 2 & 1 & 2 & 3 & 4 \end{bmatrix}, \tag{16}$$

$$w_1 = 2.7, \quad w_2 = 3.4, \quad w_3 = 4.$$

The assignment problem $A1(3)$ is

$$\begin{aligned}
\text{Min} \quad & \sum_{j=1}^{n}\sum_{r=1}^{n} c_{j,r}x_{j,r} \\
\text{s.t.} \quad & \sum_{r=1}^{n} x_{j,r} = 1, \quad j = 1,2,\ldots,n, \\
& \sum_{j=1}^{n} x_{j,r} = 1, \quad r = 1,2,\ldots,n, \\
& x_{j,r} \in \{0,1\}, \quad j = 1,2,\ldots,n,\ r = 1,2,\ldots,n,
\end{aligned} \tag{17}$$

where

$$c_{j,r} = \begin{bmatrix} w_1 g(1,1)p_1 & w_2 g(1,2)p_1 & w_3 g(1,3)p_1 & \beta_1 & \beta_1 \\ w_1 g(2,1)p_2 & w_2 g(2,2)p_2 & w_3 g(2,3)p_2 & \beta_2 & \beta_2 \\ w_1 g(3,1)p_3 & w_2 g(3,2)p_3 & w_3 g(3,3)p_3 & \beta_3 & \beta_3 \\ w_1 g(4,1)p_4 & w_2 g(4,2)p_4 & w_3 g(4,3)p_4 & \beta_4 & \beta_4 \\ w_1 g(5,1)p_5 & w_2 g(5,2)p_5 & w_3 g(5,3)p_5 & \beta_5 & \beta_5 \end{bmatrix}
= \begin{bmatrix} 37.8 & 23.8 & 84 & 10 & 10 \\ 16.2 & 61.2 & 48 & 8 & 8 \\ 27 & 51 & 20 & 4 & 4 \\ 5.4 & 13.6 & 24 & 5 & 5 \\ 5.4 & 3.4 & 8 & 6 & 6 \end{bmatrix}. \tag{18}$$

The solution is

$$x_{j,r} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \tag{19}$$

The optimal sequence is $\pi^* = [J_4, J_5, J_3, J_1, J_2]$, $d^* = 8.5$. The total cost is 46.8.

4. The *SLK* Due Date Assignment Method

In the *SLK* model, $d_j = p_j^A + q$ $(j = 1,2,\ldots,n)$. We choose γq as the cost function $\varphi(\mathbf{d})$, where γ is a positive constant. Thus, it follows from function (3) that our problem is to minimize the objective function

$$F(\mathbf{d},\pi) = \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma q. \tag{20}$$

The problem denotes $1|p_j^A = p_j g(j,r), S_{\text{psd}}, SLK|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma q$.

Similar to the *CON* model, if the number of jobs in N_E is given, we have the following solution.

Lemma 7. *For the problem $1|p_j^A = p_j g(j,r), S_{psd}, SLK|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma q$, if $|N_E| = h$, there exists an optimal schedule $\pi = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$, such that the slack time $q = C_{[h-1]} + S_{[h]}$.*

Proof. The proof is similar to that of Lemma 3.

Let $\pi = [J_{[1]}, J_{[2]}, \ldots, J_{[n]}]$, $q = C_{[h-1]} + S_{[h]}$. Thus,

$$\begin{aligned}
q &= \sum_{j=1}^{h-1} [1 + (h-j-1)\delta]\, g([j],j)\, p_{[j]} \\
&\quad + \delta \sum_{j=1}^{h-1} g([j],j)\, p_{[j]} \\
&= \sum_{j=1}^{h-1} [1 + (h-j)\delta]\, g([j],j)\, p_{[j]}, \\
E_{[j]} &= d_{[j]} - C_{[j]} \\
&= g([j],j)\, p_{[j]} + q - C_{[j]} \\
&= g([j],j)\, p_{[j]} + \sum_{k=1}^{h-1} [1 + (h-k)\delta]\, g([k],k)\, p_{[k]} \\
&\quad - \sum_{k=1}^{j} [1 + (h-k)\delta]\, g([k],k)\, p_{[k]} \\
&= g([j],j)\, p_{[j]} + \sum_{k=j+1}^{h-1} [1 + (h-k)\delta]\, g([k],k)\, p_{[k]}, \\
&\qquad 1 \le j \le h-1.
\end{aligned} \tag{21}$$

Consequently,

$$\begin{aligned}
\sum_{j=1}^{h-1} E_{[j]} &= \sum_{j=1}^{h-1} \left\{ g([j],j)\, p_{[j]} + \sum_{k=j+1}^{h-1} [1 + (h-k)\delta]\, g([k],k)\, p_{[k]} \right\} \\
&= \sum_{j=1}^{h-1} g([j],j)\, p_{[j]} \\
&\quad + \sum_{j=2}^{h-1} (j-1)[1 + (h-j)\delta]\, g([j],j)\, p_{[j]}.
\end{aligned} \tag{22}$$

Therefore,

$$\begin{aligned}
F(\mathbf{d},\pi,h) &= \alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma q \\
&= \alpha \sum_{j=1}^{h-1} g([j],j)\, p_{[j]}
\end{aligned}$$

$$+\alpha\sum_{j=2}^{h-1}(j-1)[1+(h-j)\delta]g([j],j)p_{[j]}$$

$$+\sum_{j=h+1}^{n}\beta_{[j]}+\gamma q$$

$$=\alpha\sum_{j=1}^{h-1}g([j],j)p_{[j]}$$

$$+\alpha\sum_{j=2}^{h-1}(j-1)[1+(h-j)\delta]g([j],j)p_{[j]}$$

$$+\sum_{j=h+1}^{n}\beta_{[j]}+\gamma\sum_{j=1}^{h-1}[1+(h-j)\delta]g([j],j)p_{[j]}$$

$$=\sum_{j=1}^{h}w_j g([j],j)p_{[j]}+\sum_{j=h+1}^{n}\beta_{[j]}, \tag{23}$$

where

$$w_j = \begin{cases}\alpha+\gamma(1+(h-j)\delta) & \text{for } j=1,\\ \alpha+[\alpha(j-1)+\gamma][1+\delta(h-j)] & \text{for } j=2,\ldots,h-1,\\ 0 & \text{for } j=h.\end{cases} \tag{24}$$

□

Since the objective functions for *CON* and *SLK* due date assignment methods have the same structure, we have the following solution.

Let $x_{j,r}$ be a binary variable such that $x_{j,r}=1$ if job J_j is scheduled in the rth position and $x_{j,r}=0$; otherwise, $j,r=1,2,\ldots,n$. If $|N_E|=h$, then we can formulate the problem with objective (20) as the following assignment problem $A2(h)$, which can be solved in $O(n^3)$ time:

$$\begin{aligned}\text{Min}\quad & \sum_{j=1}^{n}\sum_{r=1}^{n}c_{j,r}x_{j,r}\\ \text{s.t.}\quad & \sum_{r=1}^{n}x_{j,r}=1,\quad j=1,2,\ldots,n,\\ & \sum_{j=1}^{n}x_{j,r}=1,\quad r=1,2,\ldots,n,\\ & x_{j,r}\in\{0,1\},\quad j=1,2,\ldots,n,\ r=1,2,\ldots,n,\end{aligned} \tag{25}$$

where

$$c_{j,r}=\begin{cases}w_r g(j,r)p_j & \text{for } r=1,2,\ldots,h,\\ \beta_j & \text{for } r=h+1,\ h+2,\ldots,n,\end{cases}$$

$$w_r = \begin{cases}\alpha+\gamma(1+(h-r)\delta) & \text{for } r=1,\\ \alpha+[\alpha(r-1)+\gamma][1+\delta(h-r)] & \text{for } r=2,\ldots,h-1,\\ 0 & \text{for } r=h.\end{cases} \tag{26}$$

In order to derive the optimal solution, we have to solve the above assignment problem $A2(h)$ for any $h=1,2,\ldots,n$.

As a result, we obtain the following theorem.

Theorem 8. *The problem* $1|p_j^A=p_jg(j,r),S_{psd},SLK|\alpha\sum_{J_j\in N_E}E_j+\sum_{J_j\in N_T}\beta_jU_j+\gamma d$ *can be solved in* $O(n^4)$ *time.*

5. Job-Independent Position Effects Case

In this section, we explore the model with job-independent position effects; that is, the actual processing time of job J_j, if scheduled in position r of a sequence, is given by $p_j^A=p_jg(r)$, where $g(1),g(2),\ldots,g(n)$ represent an array of job-independent positional factors. In Section 4, we have shown that the general version (job-dependent position effects) can be solved in $O(n^4)$ time. In the following, we present an $O(n^3)$ time dynamic programming algorithm for solving the special version with job-independent position effects. The main idea that will be used in the development of our algorithm is similar to that of Shabtay et al. [35].

Based on the properties proved in Section 4, we have the following solutions.

For the *CON* model, if $|N_E|=h$, $\pi=[J_{[1]},J_{[2]},\ldots,J_{[n]}]$, and $d=C_{[h]}$, then

$$F(\mathbf{d},\pi,h)=\sum_{j=1}^{h}\left\{\alpha(j-1)+\frac{1}{2}\alpha\delta(h-j)(h+j-1)+[1+\delta(h-j)]\gamma\right\}g(j)p_{[j]}+\sum_{j=h+1}^{n}\beta_{[j]}$$

$$=\sum_{j=1}^{h}\overline{w}_j g(j)p_{[j]}+\sum_{j=h+1}^{n}\beta_{[j]}, \tag{27}$$

where

$$\overline{w}_j=\left\{\alpha(j-1)+\frac{1}{2}\alpha\delta(h-j)(h+j-1)+[1+\delta(h-j)]\gamma\right\},\quad j=1,2,\ldots,h. \tag{28}$$

For the *SLK* model, if $|N_E| = h$, $\pi = [J_{[1]}, J_{[2]}, \dots, J_{[n]}]$, and $d = C_{[h-1]} + S_{[h]}$, then

$$\begin{aligned} F(\mathbf{d}, \pi, h) &= [\alpha + \gamma(1 + (h-j)\delta)]\, g(1)\, p_{[1]} \\ &\quad + \sum_{j=2}^{h-1} \{\alpha + [\alpha(j-1) + \gamma][1 + \delta(h-j)]\} \\ &\quad \times g(j)\, p_{[j]} + \sum_{j=h+1}^{n} \beta_{[j]} \\ &= \sum_{j=1}^{h} \overline{w}_j g(j)\, p_{[j]} + \sum_{j=h+1}^{n} \beta_{[j]}, \end{aligned} \tag{29}$$

where

$$\overline{w}_j = \begin{cases} \alpha + \gamma(1 + (h-j)\delta) & \text{for } j = 1, \\ \alpha + [\alpha(j-1) + \gamma][1 + \delta(h-j)] & \text{for } j = 2, \dots, h-1, \\ 0 & \text{for } j = h. \end{cases} \tag{30}$$

Using (27) and (29), and with any of the two previously mentioned due date assignments methods, let $w_j = \overline{w}_j g(j)$; the objective function can be formulated as $F(\mathbf{d}, \pi) = \sum_{j=1}^{n} w_j p_{\pi(j)}$ for the special case of $N_E = N$, where no jobs are tardy. From Lemma 1, the optimal job sequence is obtained by matching the largest w_j value to the job with the smallest p_j value, the second largest w_j value to the job with the second smallest p_j value, and so on. The index of the w_j matched with p_j specifies the position of job j in the optimal sequence. For example, first renumber the jobs in the LPT order such that $p_1 \geq p_2 \geq \cdots \geq p_n$, and then reorder the positional weights such that $w_{i_1} \leq w_{i_2} \leq \cdots \leq w_{i_n}$ ($i_1, i_2, \dots, i_n$ is a permutation of $1, 2, \dots, n$), schedules job j in the position i_j ($j = 1, 2, \dots, n$).

We now consider the due date assignment problem to minimize the objective function (3). Since the objective functions for all two due date assignment methods have the same structure, we provide a generic algorithm to solve these problems with two due date assignment methods. If set N_E is given, ($|N_E| = h$), then we can reorder the positional weights such that $w_{i_1} \leq w_{i_2} \leq \cdots \leq w_{i_h}$. Thus, an optimal job sequence of N_E is obtained in $O(h \log h)$ time. However, in order to find the optimal solution for the due date assignment problem, the contribution of the total cost of the tardy jobs must be taken into account. Below, we present a new dynamic programming algorithm. For a given h, the idea of a dynamic programming algorithm to minimize the function (3) is as follows. We define the states of the form (i, r), where i means that jobs $J_1, J_2, \dots, J_i$ have been considered and r, ($1 \leq r \leq \min\{i, h\}$), represents how many of these jobs have been sequenced as nontardy jobs. A state (i, r) is associated with $f(i, r)$, the smallest value of the objective function in the class of partial schedules for processing i jobs, provided that r of the these jobs has been sequenced nontardy. This method works by either each job tardy or nontardy. Next, all $f(i, r)$ values can be calculated by applying the recursion for $i = 1, 2, \dots, n$ and $r \geq \max\{1, h-(n-i)\}$. The condition is that $r \geq h-(n-i)$ is necessary to ensure that we do not consider states that might lead to a solution which has fewer than r jobs in set N_E: since $|N_E| = h$ and there are r jobs that have been sequenced nontardy among the first i jobs, the remaining $h-r$ nontardy jobs needed to be selected from the last $n-i$ jobs. The formal statement of the algorithm is below.

Algorithm 9.

Step 0. Renumber the jobs in the *LPT* order such that $p_1 \geq p_2 \geq \cdots \geq p_n$.

Step 1. Calculate positional weights $w_j = \overline{w}_j g(j)$ where

$$\overline{w}_j = \left\{\alpha(j-1) + \frac{1}{2}\alpha\delta(h-j)(h+j-1) + [1 + \delta(h-j)]\gamma\right\}, \quad (j = 1, 2, \dots, h) \tag{31}$$

for the *CON* model and $\overline{w}_1 = \alpha + \gamma[1 + (h-1)\delta]$, $\overline{w}_j = \alpha + [\alpha(j-1) + \gamma][1 + \delta(h-j)]$, ($j = 2, \dots, h-1$), and $\overline{w}_h = 0$ for the *SLK* model. Reorder the positional weights such that $w_{i_1} \leq w_{i_2} \leq \cdots \leq w_{i_h}$. Initialize $f(0, 0) = 0$, $f(i, r) = \infty$ for $r > i$.

Step 2. For i from 1 to n calculate

$$f(i, 0) = f(i-1, 0) + \beta_i, \tag{32}$$

$$f(i, r) = \min\left\{f(i-1, r) + \beta_i, f(i-1, r-1) + w_{i_r} p_i\right\}, \quad \max\{1, h-(n-i)\} \leq r \leq \min\{i, h\}. \tag{33}$$

Step 3. Compute the optimal value of the function $f^*(h) = f(n, h)$.

For a given h value, calculating all possible $f(n, h)$ values using the above recursion relation requires $O(nh)$ time. Since the value of positional weights (and the order of positional weights) can be altered by changing the h value, we must repeat the entire programming procedure for each $h = 0, 1, 2, \dots, n$. Thus the minimal objective value, F^*, is given by

$$F^* = \min_{h=0,1,\dots,n} \{f^*(h)\}. \tag{34}$$

Therefore, the following statement holds.

Theorem 10. *Both the problems* $1|p_j^A = p_j g(r), S_{psd}, CON|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d$ *and* $1|p_j^A = p_j g(r), S_{psd}, SLK|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d$ *can be solved in* $O(n^3)$ *time.*

Example 11. Consider the problem $1|p_j^A = p_j g(r), S_{\text{psd}}, CON|\alpha \sum_{J_j \in N_E} E_j + \sum_{J_j \in N_T} \beta_j U_j + \gamma d$.

Let $n = 5$ and $h = 3$. The weights are $\alpha = 1$, $\gamma = 2$, and $\delta = 0.1$.

The processing times are $p_1 = 7$, $p_2 = 6$, $p_3 = 5$, $p_4 = 2$, and $p_5 = 1$.

The tardy penalties are $\beta_1 = 10$, $\beta_2 = 8$, $\beta_3 = 4$, $\beta_4 = 5$, and $\beta_5 = 6$.

The positional effects are $g(1) = 8$, $g(2) = 7$, $g(3) = 3$, $g(4) = 4$, and $g(5) = 7$.

Positional weights are $w_1 = 21.6$, $w_2 = 23.8$, and $w_3 = 12$.

Positional weights are reordered such that $w_{i_1} \le w_{i_2} \le w_{i_3}$; that is, $w_{i_1} = w_3$, $w_{i_2} = w_1$, and $w_{i_3} = w_2$:

$$\begin{gathered}
f(0,0) = 0, \\
i = 1, \\
f(1,0) = f(0,0) + \beta_1 = 10, \\
f(1,1) = f(0,0) + w_{i_1} p_1 = 84, \\
i = 2, \\
f(2,0) = f(1,0) + \beta_2 = 18, \\
f(2,1) = \min\left\{f(1,1) + \beta_2, f(1,0) + w_{i_1} p_2\right\} \\
= \min\{92, 82\} = 82, \\
f(2,2) = f(1,1) + w_{i_2} p_2 = 203.6, \\
i = 3, \\
f(3,0) = f(2,0) + \beta_3 = 22, \\
f(3,1) = \min\left\{f(2,1) + \beta_3, f(2,0) + w_{i_1} p_3\right\} \\
= \min\{86, 78\} = 78, \\
f(3,2) = \min\left\{f(2,2) + \beta_3, f(2,1) + w_{i_2} p_3\right\} \\
= \min\{207.6, 190\} = 190, \\
f(3,3) = f(2,2) + w_{i_3} p_3 = 322.6, \\
i = 4, \\
f(4,0) = f(3,0) + \beta_4 = 27, \\
f(4,2) = \min\left\{f(3,2) + \beta_4, f(3,1) + w_{i_2} p_4\right\} \\
= \min\{195, 121.2\} = 121.2, \\
f(4,3) = \min\left\{f(3,3) + \beta_4, f(3,2) + w_{i_3} p_4\right\} \\
= \min\{327.6, 237.6\} = 237.6, \\
i = 5, \\
f(5,0) = f(4,0) + \beta_5 = 33, \\
f(5,3) = \min\left\{f(4,3) + \beta_5, f(4,2) + w_{i_3} p_5\right\} \\
= \min\{243.6, 145\} = 145.
\end{gathered} \tag{35}$$

Therefore, $f(3) = f(5,3) = 145$.

The optimal sequence is $\pi^* = [J_4, J_5, J_3, J_1, J_2]$, $d^* = 41.9$. The total cost is 145.

6. Conclusions

Scheduling problems involving position-dependent processing time have received increasing attention in recent years. In this paper, we considered single machine scheduling and due date assignment with setup time in which a job's processing time depends on its position in a sequence. The setup time is past-sequence-dependent (p-s-d). The objective functions include total earliness, the weighted number of tardy jobs, and the cost of due date assignment. The due date assignment methods used in this problem include common due date (*CON*) and equal slack (*SLK*). We have presented an $O(n^4)$ time algorithm for the general case and an $O(n^3)$ time dynamic programming algorithm for the special cases. In the paper, the model with position-dependent effects is considered. However, in some other situations, a job's processing time may be time-dependent or both position-dependent and time-dependent. Therefore, it is worthwhile for future research to investigate the model in which a job's processing time depends both on its position in a sequence and its start time. It is also interesting for future research to investigate the model in the context of other scheduling settings, including multimachine and job-shop scheduling.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the editor and three anonymous referees for their helpful comments on the earlier version of the paper. This paper was supported in part by the Ministry of Science and Technology of Taiwan under Grant no. MOST 103-2221-E-252-004.

References

[1] A. Bachman and A. Janiak, "Scheduling jobs with position-dependent processing times," *Journal of the Operational Research Society*, vol. 55, no. 3, pp. 257–264, 2004.

[2] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[3] T. C. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations," *Annals of Operations Research*, vol. 98, pp. 273–290 (2001), 2000.

[4] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.

[5] G. Mosheiov, "Scheduling problems with a learning effect," *European Journal of Operational Research*, vol. 132, no. 3, pp. 687–693, 2001.

[6] G. Mosheiov, "Parallel machine scheduling with a learning effect," *Journal of the Operational Research Society*, vol. 52, no. 10, pp. 1165–1169, 2001.

[7] C. Wu, W. Lee, and T. Chen, "Heuristic algorithms for solving the maximum lateness scheduling problem with learning considerations," *Computers and Industrial Engineering*, vol. 52, no. 1, pp. 124–132, 2007.

[8] C. Wu, P. Hsu, J. C. Chen, and N. S. Wang, "Genetic algorithm for minimizing the total weighted completion time scheduling problem with learning and release times," *Computers & Operations Research*, vol. 38, no. 7, pp. 1025–1034, 2011.

[9] Y. Q. Yin, W. H. Wu, T. C. E. Cheng, and C. C. Wu, "Single-machine scheduling with time-dependent and position-dependent deteriorating jobs," *International Journal of Computer Integrated Manufacturing*, 2014.

[10] Y. Q. Yin, T. C. E. Cheng, and C. C. Wu, "Scheduling with time-dependent processing times," *Mathematical Problems in Engineering*, vol. 2014, Article ID 201421, 2 pages, 2014.

[11] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[12] G. Mosheiov, "A note on scheduling deteriorating jobs," *Mathematical and Computer Modelling*, vol. 41, no. 8-9, pp. 883–886, 2005.

[13] W.-H. Kuo and D.-L. Yang, "Minimizing the makespan in a single-machine scheduling problem with the cyclic process of an aging effect," *Journal of the Operational Research Society*, vol. 59, no. 3, pp. 416–420, 2008.

[14] A. Janiak and R. Rudek, "Scheduling jobs under an aging effect," *Journal of the Operational Research Society*, vol. 61, no. 6, pp. 1041–1048, 2010.

[15] C. L. Zhao and H. Y. Tang, "Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan," *Applied Mathematical Modelling*, vol. 34, no. 3, pp. 837–841, 2010.

[16] K. Rustogi and V. A. Strusevich, "Single machine scheduling with general positional deterioration and rate-modifying maintenance," *Omega*, vol. 40, no. 6, pp. 791–804, 2012.

[17] G. Mosheiov, "Proportionate flowshops with general position-dependent processing times," *Information Processing Letters*, vol. 111, no. 4, pp. 174–177, 2011.

[18] C. Zhao, Y. Yin, T. C. E. Cheng, and C. Wu, "Single-machine scheduling and due date assignment with rejection and position-dependent processing times," *Journal of Industrial and Management Optimization*, vol. 10, no. 3, pp. 691–700, 2014.

[19] K. Rustogi and V. A. Strusevich, "Simple matching vs linear assignment in scheduling models with positional effects: a critical review," *European Journal of Operational Research*, vol. 222, no. 3, pp. 393–407, 2012.

[20] C. Koulamas and G. J. Kyparisis, "Single-machine scheduling problems with past-sequence-dependent setup times," *European Journal of Operational Research*, vol. 187, no. 3, pp. 1045–1049, 2008.

[21] J. Wang, "Single-machine scheduling with past-sequence-dependent setup times and time-dependent learning effect," *Computers and Industrial Engineering*, vol. 55, no. 3, pp. 584–591, 2008.

[22] Y. Yin, D. Xu, and J. Wang, "Some single-machine scheduling problems with past-sequence-dependent setup times and a general learning effect," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9–12, pp. 1123–1132, 2010.

[23] C.-J. Hsu, W.-H. Kuo, and D.-L. Yang, "Unrelated parallel machine scheduling with past-sequence-dependent setup time and learning effects," *Applied Mathematical Modelling*, vol. 35, no. 3, pp. 1492–1496, 2011.

[24] W. Lee, "Single-machine scheduling with past-sequence-dependent setup times and general effects of deterioration and learning," *Optimization Letters*, vol. 8, no. 1, pp. 135–144, 2014.

[25] X. Huang, G. Li, Y. Huo, and P. Ji, "Single machine scheduling with general time-dependent deterioration, position-dependent learning and past-sequence-dependent setup times," *Optimization Letters*, vol. 7, no. 8, pp. 1793–1804, 2013.

[26] V. Gordon, J. Proth, and C. Chu, "A survey of the state-of-the-art of common due date assignment and scheduling research," *European Journal of Operational Research*, vol. 139, no. 1, pp. 1–25, 2002.

[27] Y. Yin, M. Liu, T. C. E. Cheng, C. Wu, and S. Cheng, "Four single-machine scheduling problems involving due date determination decisions," *Information Sciences. An International Journal*, vol. 251, pp. 164–181, 2013.

[28] H. G. Kahlbacher and T. C. E. Cheng, "Parallel machine scheduling to minimize costs for earliness and number of tardy jobs," *Discrete Applied Mathematics*, vol. 47, no. 2, pp. 139–164, 1993.

[29] D. Shabtay and G. Steiner, "Two due date assignment problems in scheduling a single machine," *Operations Research Letters*, vol. 34, no. 6, pp. 683–691, 2006.

[30] C. Koulamas, "A faster algorithm for a due date assignment problem with tardy jobs," *Operations Research Letters*, vol. 38, no. 2, pp. 127–128, 2010.

[31] V. S. Gordon and V. A. Strusevich, "Machine scheduling and due date assignment with positionally dependent processing times," *European Journal of Operational Research*, vol. 198, pp. 57–62, 2009.

[32] C. Hsu, S. Yang, and D. Yang, "Two due date assignment problems with position-dependent processing time on a single-machine," *Computers and Industrial Engineering*, vol. 60, no. 4, pp. 796–800, 2011.

[33] J. Wang, "Flow shop scheduling jobs with position-dependent processing times," *Journal of Applied Mathematics & Computing*, vol. 18, no. 1-2, pp. 383–391, 2005.

[34] G. H. Hardy, J. E. Littlewood, and G. Polya, *Inequalities*, Cambridge University Press, London, UK, 1967.

[35] D. Shabtay, N. Gaspar, and L. Yedidsion, "A bicriteria approach to scheduling a single machine with job rejection and positional penalties," *Journal of Combinatorial Optimization*, vol. 23, no. 4, pp. 395–424, 2012.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 748905, 5 pages
http://dx.doi.org/10.1155/2014/748905

Research Article

Scheduling Jobs and a Variable Maintenance on a Single Machine with Common Due-Date Assignment

Long Wan

School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China

Correspondence should be addressed to Long Wan; cocu3328@163.com

Received 13 May 2014; Revised 7 July 2014; Accepted 7 July 2014; Published 22 July 2014

Academic Editor: Ming Liu

We investigate a common due-date assignment scheduling problem with a variable maintenance on a single machine. The goal is to minimize the total earliness, tardiness, and due-date cost. We derive some properties on an optimal solution for our problem. For a special case with identical jobs we propose an optimal polynomial time algorithm followed by a numerical example.

1. Introduction

Recently, as a competitive strategy to provide high quality service for customer demand, just-in-time (JIT) production has received considerable attention from the manufacturing enterprises [1]. In JIT production, jobs should be completed as close as possible to their due-dates. A job which is completed earlier or later than its due-date will incur penalty. Thus, both earliness and tardiness are discouraged in JIT production. Motivated by the JIT production, in the last two decades scheduling problem with due-date assignment has been extensively investigated. For the related surveys, we refer the readers to Cheng and Gupta [2], Baker and Scudder [3], and Gordon et al. [4]. For some recent related models on the due-date assignment scheduling, see Xu et al. [5], Gerstl and Mosheiov [6], Yin et al. [7], and Janiak et al. [8].

On the other hand, to prevent production disruption caused by machine breakdown, machine maintenance needs to be performed to perserve production efficiency. Since 1996, researchers begun to take maintenance into consideration in scheduling (see Lee [9], Lee and Chen [10], Kubzin and Strusevich [11], Mosheiov and Sidney [12], and Zhao and Tang [13]). For a maintenance (which is optional or mandatory), we usually use two parameters to define it. One is its starting time and the other is its duration. In some papers, the occupied period by maintenance was also called nonavailable interval. For the recent related survey, we refer the readers to Ma et al. [14].

In the papers by Kubzin and Strusevich [11] and Mosheiov and Sidney [12], they considered a more realistic case on the duration of maintenance. In their models, they assume the duration of maintenance is variable; that is, the duration of maintenance depends on its starting time in that the later maintenance is performed, the longer time is needed to perform the maintenance. Such maintenance can be called a variable maintenance or a deteriorating maintenance.

Another popular topic in recent years is that of scheduling with simultaneous considerations of due-date assignment and maintenance. Mosheiov and Oron [15] studied a single-machine scheduling problem jointly with rate-modifying activity and common due-date assignment considerations to minimize the total of earliness, tardiness, and due-date costs. X. Y. Wang and M. Z. Wang [16] addressed a single-machine slack due-date scheduling with a rate-modifying activity for minimizing the objective function which contains the total earliness, tardiness, and the common slack time costs. Yang et al. [17] investigated single-machine common due-date assignment and scheduling problems with an aging effect under a deteriorating maintenance consideration simultaneously. Yin et al. [18] considered a common due-date assignment and scheduling problem with a rate-modifying activity to minimize the due-date, earliness, tardiness, holding, and batch delivery cost.

In this paper we introduce a new scheduling model which combines the due-date assignment and the machine maintenance. We assume that the duration of maintenance is variable and the maintenance must be started prior to a given deadline.

As a practical example for the proposed model, we may consider the steel-making process in the steel plant [19]. In the steel-making process, a charge, that is, a concurrent smelting in the same converter, is regarded as a "job." A refining furnace is used to refine the charges. Naturally, the refining furnace is regarded as "the machine." In the refining process, there will be some garbage. Before a given deadline, we must clear the garbage to maintain the production efficiency and thus the clearing operation can be regarded a maintenance.

In the second section we provide the notation and formulation on our model. The third section derives some important properties on an optimal solution. In Section 4, we propose an optimal polynomial time algorithm for a special case with identical jobs, followed by a numerical example. Concluding remarks are discussed in the last section.

2. Problem Statement

Our problem can be described as follows. There are n independent jobs $J_1, J_2, \ldots, J_n$ to be nonpreemptively processed on a single machine, all of which are available at time zero. A mandatory maintenance must be started before a given deadline on the machine and the duration of maintenance depends on its starting time; that is, the duration is a nonnegative and nondecreasing function of the starting time. Let p_j and d_j denote the processing time and the due-date of job J_j, $j = 1, 2, \ldots, n$, respectively. For a given schedule, we use C_j to denote the completion time of job J_j, $j = 1, 2, \ldots, n$. We define the earliness and tardiness of job J_j as $E_j = \max\{d_j - C_j, 0\}$ and $T_j = \max\{C_j - d_j, 0\}$, $j = 1, 2, \ldots, n$, respectively. The unit earliness and tardiness penalties are denoted by α (>0) and β (>0), respectively. In the case of a common due-date (i.e., $d_j = d$, which is a decision variable), we denote the penalty per unit time of delaying the due-date by γ (>0). Furthermore, we denote the given deadline of maintenance with s_g and the duration of maintenance with l. Then according to our previous assumption, we can denote $l = f(s)$, where s ($\leq s_g$) is the starting time of maintenance and f is a nonnegative and nondecreasing function. The goal is to find an optimal sequence of all the jobs, the common due-date, and the staring time of maintenance such that the objective $\sum_{j=1}^{n}(\alpha E_j + \beta T_j) + \gamma d$ is minimized. Following the three-field notation proposed by Graham et al. [20], we denote our problem as $1, \mathrm{VM} \parallel \sum_{j=1}^{n}(\alpha E_j + \beta T_j) + \gamma d$, where VM in the first field stands for a variable maintenance.

3. The Properties on an Optimal Solution

The classical due-date assignment scheduling problem (without maintenance) was introduced by Panwalkar et al. [21]. In their model, in addition to the traditional job sequencing decisions, the common due-date is a decision variable. Both earliness and tardiness incur penalty cost. The goal is to find an optimal sequence of the jobs and the due-date that minimizes the total earliness, tardiness, and due-date cost. By using the small perturbations technique, Panwalkar et al. [21] proposed a polynomial time algorithm.

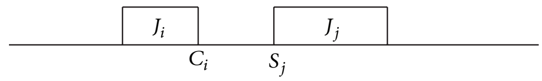


Figure 1: Idle time between jobs J_i and J_j.

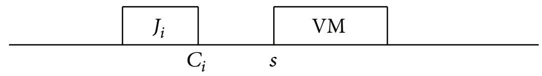


Figure 2: Idle time between job J_i and the maintenance VM.

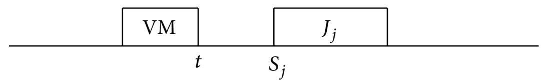


Figure 3: Idle time between the maintenance VM and job J_j.

In order to solve our problem, we first derive some properties on an optimal solution. We also use the small perturbations technique.

Lemma 1. *There exists an optimal solution in which the schedule starts at time zero and contains no idle time among the jobs, and the maintenance is scheduled between the two consecutive jobs without idle time.*

Proof. First we show that there is no idle time between the jobs.

Assume that there exists idle time between the jobs J_i and J_j, as shown in Figure 1, where C_i denotes the completion time of job J_i and S_j denotes the starting time of job J_j. Clearly we have $C_i < S_j$. If $d < C_i$, we move job J_j by $(S_j - C_i)$ units of time to the left without increasing the objective value. If $d > S_j$, we may move job J_i by $(S_j - C_i)$ units of time to the right without increasing the objective value. If $C_i \leq d \leq S_j$, we may move job J_i to the right and job J_j to the left such that job J_i just finishes at time d and job J_j starts at time d without increasing the objective value. In the end, by clearing the idle time between the jobs we always obtain a better solution.

Next, we show that the maintenance is scheduled between the two consecutive jobs without idle time.

Assume that there exists idle time between the job J_i and the maintenance, as shown in Figure 2, where J_i is scheduled before the maintenance and s denotes the starting time of the maintenance. Clearly, we have $C_i < s$. If $d < C_i$, we move the maintenance by $(s - C_i)$ units of time to the left without increasing the objective value. If $d > s$, we may move job J_i by $(s - C_i)$ units of time to the right without increasing the objective value. If $C_i \leq d \leq s$, we may move job J_i to the right and the maintenance to the left such that job J_i finishes at time d and the maintenance starts at time d without increasing the objective value.

Assume that there exists idle time between the maintenance and the job J_j, also shown in Figure 3, where J_j is scheduled after the maintenance and t denotes the finishing time of the maintenance.

Clearly, we know $t < S_j$. If $d < t$, we move job J_j by $(S_j - t)$ units of time to the left without increasing the objective value. If $d > S_j$, we may delay the starting time of maintenance such that the maintenance finishes at time S_j. If $t \le d \le S_j$, we may delay the starting time of maintenance and move job J_j to the left such that the maintenance finishes at time d and job J_j starts at time d.

By the above analysis, we can treat all the jobs and the maintenance as a consecutive whole without idle time.

Finally, we show that the schedule starts at time zero. Assume that there exists a solution which does not start at time zero. Then we move the whole to the left by some times to assure that the new schedule starts at time zero and reset a smaller common due-date than the original due-date to obtain a new solution, which does not increase the objective value.

With the above argument, we conclude Lemma 1 holds. □

Lemma 2. *The optimal common due-date is the completion time of the job in position m, where* $m = \lceil (n\beta - \gamma)/(\alpha + \beta) \rceil$.

Proof. First we show that in an optimal solution the common due-date d is the completion time of some job. We distinguish two cases.

Case 1. Consider a solution with $C_i < d < C_{i+1}$, where i denotes the job scheduled in the ith location. Let Z be the corresponding objective value. Define $x = d - C_i$ and $y = C_{i+1} - d$. Let Z_1 and Z_2 be the objective value for $d = C_i$ and $d = C_{i+1}$. Then

$$\begin{aligned} Z_1 &= Z + \beta(n-i)x - \alpha x i - \gamma x \\ &= Z + x(\beta(n-i) - \alpha i - \gamma), \\ Z_2 &= Z - \beta(n-i)y + \alpha i y + \gamma y \\ &= Z - y(\beta(n-i) - \alpha i - \gamma). \end{aligned} \tag{1}$$

Thus, we have $Z_1 \le Z$ if $\beta(n-i) - \alpha i - \gamma \le 0$ and $Z_2 \le Z$ otherwise. This implies that an optimal solution exists in which d is equal to the completion time of some job.

Case 2. Consider a solution with $s \le d \le s + f(s)$, where s denotes the starting time of maintenance and $f(s)$ denotes the duration of maintenance. Similar to Case 1, using the small perturbations technique we can show that the objective value can reduce by resetting $d = s$ or $d = s + f(s)$. Since the case $d = s$ is shown in Case 1, thus we only need to consider the case $d = s + f(s)$. Let Z be the corresponding objective value with $d = s + f(s)$ and Z_1 and Z_2 the corresponding objective values for $d = C_i$ and $d = C_{i+1}$, where the maintenance is scheduled between the job J_i and the job J_{i+1}; that is, $s = C_i$ and $s + f(s) = C_{i+1} - p_{i+1}$, as shown in Figure 4.

Then

$$\begin{aligned} Z_1 &= Z + f(s)(\beta(n-i) - \alpha i - \gamma), \\ Z_2 &= Z - p_{i+1}(\beta(n-i) - \alpha i - \gamma). \end{aligned} \tag{2}$$

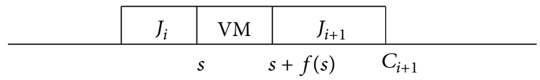


Figure 4: The case $s \le d \le s + f(s)$.

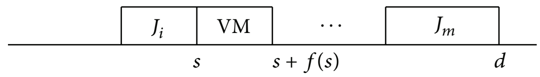


Figure 5: The maintenance starts at time s (>0).

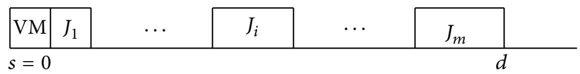


Figure 6: The maintenance starts at time 0.

Thus, we have $Z_1 \le Z$ if $\beta(n-i) - \alpha i - \gamma \le 0$ and $Z_2 \le Z$ otherwise.

With the above discussion, we conclude that in an optimal solution the optimal common due-date d is the completion time of some job.

Now, we assume that the common due-date d is the completion time of job in the mth location; that is, $d = C_m$. To prove that $m = \lceil (\beta n - \gamma)/(\alpha + \beta) \rceil$, let Z be the objective value of optimal solution. Applying Z_1 and Z_2 to the situation that $x = d - C_{m-1}$ and $y = C_{m+1} - d$, respectively, we conclude that

$$\begin{gathered} \beta(n-m+1) - \alpha(m-1) - \gamma \ge 0, \\ \beta(n-m) - \alpha m - \gamma \le 0. \end{gathered} \tag{3}$$

Thus, we have $m = \lceil (n\beta - \gamma)/(\alpha + \beta) \rceil$. □

Lemma 3. *In an optimal solution, the maintenance is scheduled either at time 0, or after the common due-date.*

Proof. Suppose that there exists a solution in which the maintenance starts at time s, where $s > 0$ and is scheduled before the common due-date d. Then the maintenance occupies the time interval $[s, s + f(s)]$ with $s \le s_g$ and $s + f(s) \le d$. Furthermore, we assume that the job J_i is just prior to the maintenance and the completion time of job J_m is equal to the due-date, as shown in Figure 5.

Now we construct a new solution as follows. Starting the maintenance at time zero and scheduling all the jobs according to their original order just after the maintenance. Setting the common due-date to the new completion time of job J_m. As shown in Figure 6, then we have the following.

(i) The duration of the maintenance decreases as it starts earlier.

(ii) The earliness of jobs J_i and its predecessors are reduced.

(iii) The common due-date d is reduced.

The above (i), (ii), and (iii) imply that the total earliness cost of job J_i and its predecessors and the due-date cost are reduced, and the earliness and tardiness cost of other jobs remain unchanged. Thus, we conclude that the maintenance should be scheduled either at time 0, or after the due-date. □

4. A Special Case 1, $\mathrm{VM}|p_j = p|\sum_j(\alpha E_j + \beta T_j) + \gamma d$

In this section we consider a special case for our problem. We assume all the jobs are identical; that is, $p_j = p$. Next, we propose a polynomial time algorithm for this special case based on the previous properties on an optimal solution.

Recall that the due-date is the completion time of job in the mth location, where $m = \lceil(\beta n - \gamma)/(\alpha + \beta)\rceil$. Because the jobs are identical jobs, by Lemma 3 we claim that the maintenance must be started at time 0 if $d = mp > s_g$ and the maintenance is started after the due-date otherwise. Then there are at most $n - m + 1$ choices for the starting time of the maintenance.

With the above analysis, we propose our algorithm as follows.

Algorithm H.

Step 1. If $d = mp > s_g$, where $m = \lceil(\beta n - \gamma)/(\alpha + \beta)\rceil$, construct schedule $\pi = (\mathrm{VM}, J_1, J_2, \ldots, J_n)$. Output it as our solution by setting the due-date as the completion time of job J_m and stop. Otherwise go to Step 2.

Step 2. Compute k such that $pk \le s_g \le p(k+1)$. Construct a series of schedules $\pi^0 = (\mathrm{VM}, J_1, J_2, \ldots, J_n)$, $\pi^i = (J_1, J_2, \ldots, J_i, \mathrm{VM}, J_{i+1}, \ldots, J_n)$, $i = m, m+1, \ldots, k$.

Step 3. Output the schedule with the minimal objective value from all the constructed schedules $\pi^0, \pi^i, i = m, m+1, \ldots, k$, and denote it as π, where $Z(\pi) = \mathrm{Min}_{i=0,m,m+1,\ldots,k} Z(\pi^i)$.

From the properties on an optimal solution as shown in Lemmas 1, 2, and 3, we conclude that Algorithm H is correct since all the possible cases are tried and we select the best one. For a given schedule, the computation of objective value requires $O(n)$ time. There are at most $n + 1$ schedules to be considered; thus the total running time is $O(n^2)$. Finally we obtain the following.

Theorem 4. *The $1, VM|p_j = p|\sum_j(\alpha E_j + \beta T_j) + \gamma d$ problem can be solved in $O(n^2)$ time.*

A Numerical Example. To illustrate Algorithm H, a solution of an instance of 10 jobs is demonstrated in the following.

The job processing times are identical with $p_j = 3$, $j = 1, 2, \ldots, 10$. The deadline of maintenance s_g is equal to 25 and the duration of maintenance l is equal to $2 + s/3$, where $s\ (\le s_g)$, as a decision variable, is the starting time of maintenance. The penalty parameters are as follows: $\alpha = 2$, $\beta = 3$, and $\gamma = 4$.

Applying Algorithm H, we first compute the parameters as follows: $m = \lceil(\beta n - \gamma)/(\alpha + \beta)\rceil = \lceil(3 \times 10 - 4)/(2 + 3)\rceil = 6$, $d = mp = 6 \times 3 = 18$, and $k = 8$ with $3 \times k \le s_g \le 3 \times (k+1)$. Because $16 < 25$, that is, $d < s_g$, we construct a series of schedules as follows:

$$\begin{aligned}\pi^0 &= (\mathrm{VM}, J_1, J_2, \ldots, J_{10}),\\ \pi^6 &= (J_1, J_2, \ldots, J_6, \mathrm{VM}, J_7, J_8, \ldots, J_{10}),\\ \pi^7 &= (J_1, J_2, \ldots, J_6, J_7, \mathrm{VM}, J_8, \ldots, J_{10}),\\ \pi^8 &= (J_1, J_2, \ldots, J_6, J_7, J_8, \mathrm{VM}, J_9, J_{10}).\end{aligned} \tag{4}$$

Their corresponding objective values are $Z(\pi^0) = 260$, $Z(\pi^6) = 348$, $Z(\pi^7) = 324$, and $Z(\pi^8) = 300$.

When comparing the costs in π^0, π^6, π^7, and π^8, we conclude that the global optimum is obtained in π^0, the maintenance starts at time zero, the common due-date d is equal to 20, six jobs are early, and four jobs are tardy. The total cost is $Z(\pi^0) = 260$.

5. Concluding Remarks

In this paper we consider the common due-date assignment scheduling problem with a variable maintenance on a single machine. The goal is to minimize the total earliness, tardiness, and due-date cost. We derive some properties on an optimal solution for our problem. For a special case with identical jobs we propose an optimal polynomial time algorithm running in $O(n^2)$ time.

For the general case with nonuniform processing times of jobs, whether problem is NP-hard or not is open and deserves the further research.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowlegments

The author is grateful to the editor and two anonymous referees for their helpful comments and suggestions.

References

[1] T. C. E. Cheng and S. Podolsky, *Just-in-Time Manufacturing: An Introduction*, Chapman & Hall, London, UK, 1996.

[2] T. C. E. Cheng and M. C. Gupta, "Survey of scheduling research involving due date determination decisions," *European Journal of Operational Research*, vol. 38, no. 2, pp. 156–166, 1989.

[3] K. R. Baker and G. D. Scudder, "Sequencing with earliness and tardiness penalties: a review," *Operations Research*, vol. 38, no. 1, pp. 22–35, 1990.

[4] V. S. Gordon, J. M. Proth, and C. Chu, "A survey of the state-of-the-art of common due date assignment and scheduling research," *European Journal of Operational Research*, vol. 139, no. 1, pp. 1–25, 2002.

[5] D. Xu, C.-L. Li, and Q. Han, "A note on parallel-machine due-window assignment," *Journal of the Operational Research Society*, vol. 63, no. 6, pp. 821–825, 2012.

[6] E. Gerstl and G. Mosheiov, "Minmax due-date assignment with a time window for acceptable lead-times," *Annals of Operations Research*, vol. 211, pp. 167–177, 2013.

[7] Y. Yin, T. C. E. Cheng, C. Hsu, and C. Wu, "Single-machine batch delivery scheduling with an assignable common due window," *Omega*, vol. 41, no. 2, pp. 216–225, 2013.

[8] A. Janiak, W. Janiak, M. Y. Kovalyov, E. Kozan, and E. Pesch, "Parallel machine scheduling and common due window assignment with job independent earliness and tardiness costs," *Information Sciences*, vol. 224, pp. 109–117, 2013.

[9] C. Lee, "Machine scheduling with an availability constraint," *Journal of Global Optimization*, vol. 9, no. 3-4, pp. 395–416, 1996.

[10] C.-Y. Lee and Z.-L. Chen, "Scheduling jobs and maintenance activities on parallel machines," *Naval Research Logistics*, vol. 47, no. 2, pp. 145–165, 2000.

[11] M. A. Kubzin and V. A. Strusevich, "Planning machine maintenance in two-machine shop scheduling," *Operations Research*, vol. 54, no. 4, pp. 789–800, 2006.

[12] G. Mosheiov and J. B. Sidney, "Scheduling a deteriorating maintenance activity on a single machine," *Journal of the Operational Research Society*, vol. 61, no. 5, pp. 882–887, 2010.

[13] C. Zhao and H. Tang, "Single machine scheduling with general job-dependent aging effect and maintenance activities to minimize makespan," *Applied Mathematical Modelling. Simulation and Computation for Engineering and Environmental Systems*, vol. 34, no. 3, pp. 837–841, 2010.

[14] Y. Ma, C. B. Chu, and C. R. Zuo, "A survey of scheduling with deterministic machine availability constraints," *Computers and Industrial Engineering*, vol. 58, no. 2, pp. 199–211, 2010.

[15] G. Mosheiov and D. Oron, "Due-date assignment and maintenance activity scheduling problem," *Mathematical and Computer Modelling*, vol. 44, no. 11-12, pp. 1053–1057, 2006.

[16] X. Y. Wang and M. Z. Wang, "Single machine common flow allowance scheduling with a rate-modifying activity," *Computers & Industrial Engineering*, vol. 59, no. 4, pp. 898–902, 2010.

[17] S. J. Yang, C. J. Hsu, and D. L. Yang, "Single-machine scheduling with due-date assignment and aging effect under a deteriorating maintenance activity consideration," *International Journal of Information and Management Sciences*, vol. 21, no. 2, pp. 177–195, 2010.

[18] Y. Q. Yin, T. C. E. Cheng, D. H. Xu, and C. C. Wu, "Common due date assignment and scheduling with a rate-modifying activity to minimize the due date, earliness, tardiness, holding, and batch delivery cost," *Computers and Industrial Engineering*, vol. 63, no. 1, pp. 223–234, 2012.

[19] L. Tang, P. B. Luh, J. Liu, and L. Fang, "Steel-making process scheduling using Lagrangian relaxation," *International Journal of Production Research*, vol. 40, no. 1, pp. 55–70, 2002.

[20] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.

[21] S. S. Panwalkar, M. L. Smith, and A. Seidmann, "Common due-date assignment to minimize total penalty for the one machine scheduling problem," *Operations Research*, vol. 30, pp. 391–399, 1982.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 673563, 9 pages
http://dx.doi.org/10.1155/2014/673563

Research Article

Cooperative Fuzzy Games Approach to Setting Target Levels of ECs in Quality Function Deployment

Zhihui Yang,[1,2] Yizeng Chen,[1] and Yunqiang Yin[2]

[1] *School of Management, Shanghai University, Shanghai 200444, China*
[2] *College of Sciences, East China Institute of Technology, Nanchang, Jiangxi 330013, China*

Correspondence should be addressed to Yizeng Chen; zhycyz@aliyun.com

Received 30 May 2014; Accepted 15 June 2014; Published 3 July 2014

Academic Editor: Dar-Li Yang

Quality function deployment (QFD) can provide a means of translating customer requirements (CRs) into engineering characteristics (ECs) for each stage of product development and production. The main objective of QFD-based product planning is to determine the target levels of ECs for a new product or service. QFD is a breakthrough tool which can effectively reduce the gap between CRs and a new product/service. Even though there are conflicts among some ECs, the objective of developing new product is to maximize the overall customer satisfaction. Therefore, there may be room for cooperation among ECs. A cooperative game framework combined with fuzzy set theory is developed to determine the target levels of the ECs in QFD. The key to develop the model is the formulation of the bargaining function. In the proposed methodology, the players are viewed as the membership functions of ECs to formulate the bargaining function. The solution for the proposed model is Pareto-optimal. An illustrated example is cited to demonstrate the application and performance of the proposed approach.

1. Introduction

Game theory is the discipline which studies multiple individuals implementing the corresponding strategy according to related strategies of other individuals under some situations. Sometimes we need to seek for the best strategies of each player taking into account that the others will also behave searching for their best. In this case, we call it as noncooperative games model. On the other hand, players just want to deal with the cooperation issues of the problem and consider how the agents allocate the benefits of their cooperation. This approach is called as the cooperative game [1, 2]. A cooperative game often assumes that each player is a part of a team and is willing to compromise his own payoff to improve the goal as a whole. A cooperative game proceeds with the intent that the team wants to allocate resources such that all players are as better off as possible, and an improvement in the payoff for one player does not result in a loss for other players. The bargaining scheme postulated by Nash yields a unique and optimal distribution of resources such that the arbitrated outcome is Pareto-optimal [1, 2].

Classical game theory is based on binary logic and the fully rational behavior assumption. Fuzzy logic is able to accommodate many of the binary-logic related dilemmas in classical game theory. In general, the players do not be having as fully rational decision makers in real games. Fuzzy logic is a tool for a formal representation of such behavior. Moreover, one of the outstanding limitations of the classical game theory is that it assumes that all the data are known exactly by all players. This assumption is often restrictive. In real world, it often happens that the players are not able to evaluate exactly the outcomes of different strategy profiles and their own preferences or the preferences of other players [3]. Therefore, to characterize the bounded rational behavior and games with imperfect or incomplete information, it is necessary to employ fuzzy logic into the game theory. Fuzzy logic was initiated by Zadeh for dealing with uncertainties [4]. From then on, fuzzy theory was extensively applied in many areas, such as decision sciences [5], control theory [6–8], and games theory [3, 9, 10].

Aubin [9] first studied the problem of fuzzy cooperative games. Dhingra and Rao [10] integrated the cooperative game theory and fuzzy set theory to yield a new optimization method. In this paper, the cooperative fuzzy game model, which was proposed by Dhingra and Rao [10], will be employed to QFD-based new product planning.

In the current economic globalization situation, more and more companies pay more attention to listen to the voice of customers. For many enterprises, the key to win competitive advantage is to develop the product with higher customer satisfaction, lower cost, and shorter product development cycle. The purpose of product innovation is that the designers can develop new products, which can attract customers and satisfy the demand of customers. Planning becomes essential in designing and manufacturing a new product efficiently at competitive cost within a short period of time [11]. As far product planning and development decisions are concerned extensively; the application of quality function deployment (QFD) has been applied in many areas. Originated in Japan in the late 1960s, QFD is a planning and problem-solving tool for translating customer requirements (CRs) into engineering characteristics (ECs) of a new product or service [12, 13]. QFD can help the designers systematically to determine ECs for developing a new product with maximum customer satisfaction. The QFD process includes four sets of matrices called houses of quality (HOQ) to relate CRs to product planning, parts deployment, process planning, and manufacturing operations [12, 13]. QFD is a breakthrough tool which can effectively reduce the gap between CRs and a new product or service.

The determination of the target levels of ECs is the core problem in QFD. The problem that setting target levels of ECs can be viewed as a game in which each player corresponds to the membership function of EC. Each player bargains with others to improve the payoff subjected to the limited resource. In the proposed methodology, the bargaining function is formulated as the geometric mean of the membership functions of ECs, and the development budget for the new product is fuzzified. Indeed, setting target levels of ECs is an optimization problem, in which the set of feasible solutions can be reduced to discrete, and the goal is to maximize the overall customer satisfaction. So setting target levels of ECs in QFD is also a combinatorial optimization problem.

The rest of this paper is organized as follows. Section 2 reviews some related work about the determination of target levels of ECs in QFD. Section 3 recalls the cooperative fuzzy game modeling approach proposed by Dhingra and Rao [10]. In Section 4, the fuzzy programming approach based on the fuzzy cooperative game model is put forward to determine the target levels of ECs. In Section 5, a motor car design is cited to illustrate the proposed methodology. Finally, the conclusions in this work are summarized in Section 6.

2. Related Work about QFD

In traditional QFD, the objective value of ECs is usually determined by the subjective experience of the QFD team. In order to determine the target levels of ECs objectively and accurately, the QFD team should develop the optimization model by taking the final importance of ECs and various constraints (cost, development time, technical feasibility, etc.) into account, where the goal is to help the QFD team to realize the overall customer satisfaction of new products catching up with or exceeding the competitors in the target market.

The determination of target levels of ECs under a fuzzy environment has gained extensive attention. Using a fuzzy ranking procedure, Zhou [14] investigated a mixed-integer linear programming model to optimize the target values of ECs. Fung et al. [15] developed a fuzzy inference model that features a fuzzy rule base to setting the target levels of ECs. Kim et al. [16] proposed a fuzzy multicriteria modeling approach to QFD planning in which fuzzy linear regression with symmetric triangular fuzzy numbers is used to estimate the functional relationships between CRs and ECs as well as among ECs. Taking into account the financial factors in the product design process, Tang et al. [17] developed a fuzzy formulation combined with a genetic-based interactive approach to QFD planning. To determine the target values of ECs, Bai and Kwong [18] proposed an inexact genetic algorithm approach to solve the model that takes the mutation along the weighted gradient direction as a genetic operator. Karsak [19] developed a fuzzy multiple objective programming approach that incorporates imprecise and subjective information inherent in the QFD planning process to determine the level of fulfillment of ECs.

There are two types of uncertainties in input in the QFD process: human perception and customer heterogeneity. To tackle the two types of uncertainties simultaneously, Chen et al. [20] developed two fuzzy expected value models to determine target values of ECs. By using dynamic programming proposed by Lai et al. [21], limited resources are allocated to the technical attributes. Y. Chen and L. Chen [22] developed a nonlinear-programming-based possibilistic regression approach. Fung et al. [23] developed a pair of hybrid linear programming models with asymmetric triangular fuzzy coefficients to estimate the functional relationships for product planning under uncertainties. Chen and Weng [24] proposed fuzzy goal programming models to determine the fulfillment levels of the ECs. Chen and Ngai [25] employed the method of imprecision (MoI) to perform multiple-attribute synthesis to generate a family of synthesis strategies by varying the value of s, which indicates the different compensation levels among ECs. A nonlinear-programming-based fuzzy regression approach was investigated in [26] to setting target levels of ECs. Chen and Ko [27] proposed fuzzy nonlinear-programming models based on Kano's concept to determine the fulfillment levels of parts characteristics with the aim of achieving the determined contribution levels of ECs for customer satisfaction. Delice and Güngör (2009) [28] investigated an approach to QFD processes to obtain the optimal solution from a limited number of design requirements alternatives with discrete value. Kwong et al. [29] investigated a generalized fuzzy least-squares regression approach to model customer satisfaction. Güngör et al. [30] used fuzzy analytic-network process (FANP) to determine the fulfillment levels of ECs. Liu [31] integrated fuzzy QFD and the prototype product selection model to develop a product design and selection approach that can substantially benefit developers in new product programming. Sener and Karsak [32] investigated an approach for determining target levels of ECs by integrating fuzzy linear regression and fuzzy multiple objective programming. Yang and Chen [33] employed fuzzy soft set theory to prioritize CRs and ECs in QFD. Jiang et al. [34] put forward a chaos-based fuzzy regression approach to model the relationships between customer satisfaction and ECs. Delice and Güngör (2013) developed a fuzzy mixed-integer goal programming model to setting the optimal discrete values of ECs [35]. Ko and Chen [36] established a new normalized relationship between CRs and ECs to improve the existing models' drawbacks and developed a fuzzy linear programming model to determine the optimal fulfillment levels of ECs. Considering several goals such as new product development time and cost, technological advancement, and manufacturability, Mungle et al. [37] proposed dynamical multiobjective evolutionary algorithm along with FANP and QFD to resolve product planning problem. Yuen [38] presented a hybrid framework of fuzzy cognitive network process, aggregative grading clustering, and QFD for the criteria evaluation and analysis in QFD.

The usefulness of these approaches is seriously limited because the performance of a complex product depends on some different, often conflicting, criteria that cannot be combined into a single measure of performance. Henceforth, a consideration of pursuing the maximization of the overall satisfaction of customers becomes a challenging problem to the design team. The process of setting the target levels of ECs is accomplished in a subjective adc manner or in a heuristic way. Due to many tradeoffs that may exist among implicit or plicit relationships between CRs and ECs and among ECs, these relationships cannot be identified using engineering knowledge. Due to cost and other resource constraints, tradeoffs are always needed. The purpose to setting target levels of ECs is to maximize the overall customer satisfaction. Therefore, there may be room for cooperation among ECs. In this study, the cooperative fuzzy game model, integrating the fuzzy set theory with the cooperative game theory, is employed to complex product planning.

3. Cooperative Games with Fuzzy Constraint

In this section, we recall the cooperative fuzzy game modeling approach proposed by Dhingra and Rao [10].

3.1. The Formulation of the Bargaining Function. Assume that there exists payoff functions $f_i(\vec{\mathbf{x}})$, $\vec{\mathbf{x}} \in S$ associated with each player i, where the set of alternatives S is convex and compact; the payoff of player i will be $f_i(\vec{\mathbf{x}})$. These players bargain with each other and hope a trade such that the payoff functions are maximized. The bargaining function $B(\cdot)$ should satisfy the following inequality:

$$\begin{aligned} \min\left(f_1, f_2, \ldots, f_m\right) &< B\left(f_1, f_2, \ldots, f_m\right) \\ &< \max\left(f_1, f_2, \ldots, f_m\right), \end{aligned} \tag{1}$$

where $B(\cdot)$ is a suitable operator that models a tradeoff among the goals $f_i = f_i(\vec{\mathbf{x}})$, $i = 1, 2, \ldots, m$. In this study, the operator $B(\cdot)$ is set as the geometric mean with weight. Therefore, to determine a solution accepted by all players, the bargaining function $B(\vec{\mathbf{x}})$ is formulated as follows:

$$B(\vec{\mathbf{x}}) = \prod_{i=1}^{m} \left(f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w)\right)^{1/m} \tag{2}$$

for $\vec{\mathbf{x}} \in S^* = \{X \in S \mid f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w) \ge 0\} \subset S$, where $f_i(\vec{\mathbf{x}}_w)$ is the worst value of the payoff function $f_i(\vec{\mathbf{x}})$ that player i is willing to accept.

The weights of all payoff functions in the bargaining function above are assumed to be equal. The generalized bargaining function is expressed as

$$B(\vec{\mathbf{x}}) = \prod_{i=1}^{m} \left(f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w)\right)^{w_i}, \tag{3}$$

where w_i denotes the weight of the payoff function $f_i(\vec{\mathbf{x}})$, such that $\sum_{i=1}^{m} w_i = 1$, $0 \le w_i \le 1$, $i = 1, 2, \ldots, m$.

3.2. The Fuzzification of the Constraint. The constraint of an optimal problem often includes some crisp inequality and crisp equality. However, in some practical problem, these inequality and equality are often expressed vaguely. For example, the upper bound of the budget for a project is often expressed as "about one million dollars." Thus the fuzzy logic is employed to characterize these inequality or equality. Assume that there are n_{fg} fuzzy inequalities and n_{fh} fuzzy equalities:

$$g_i(\vec{\mathbf{x}}) \lesssim a_i, \quad i = 1, 2, \ldots, n_{fg} \tag{4}$$

$$h_j(\vec{\mathbf{x}}) \cong b_j, \quad j = 1, 2, \ldots, n_{fh}. \tag{5}$$

The fuzzy inequality (4) can be characterized by the membership function as follows:

$$\mu_{\widetilde{G}_i}(\vec{\mathbf{x}}) = \begin{cases} 0, & g_i(\vec{\mathbf{x}}) \ge a_i + \delta_{a_i}, \\ \dfrac{a_i + \delta_{a_i} - g_i(\vec{\mathbf{x}})}{\delta_{a_i}}, & a_i < g_i(\vec{\mathbf{x}}) < a_i + \delta_{a_i}, \\ 1, & g_i(\vec{\mathbf{x}}) \le a_i, \end{cases} \tag{6}$$

where δ_{a_i} denotes the index that the upper bound of $g_i(\vec{\mathbf{x}})$ can be improved.

The fuzzy equality (5) can be characterized by the membership function as follows:

$$\mu_{\widetilde{H}_j}(\vec{\mathbf{x}}) = \begin{cases} 1 - \left|\dfrac{h_j(\vec{\mathbf{x}}) - b_j}{\tau_{b_j}}\right|, & b_j - \tau_{b_j} < h_j(\vec{\mathbf{x}}) < b_j + \tau_{b_j}, \\ 0, & \text{others}, \end{cases} \tag{7}$$

where τ_{b_j} denotes the index that the bound of $h_j(\vec{\mathbf{x}})$ can be improved. The values of δ_{a_i} and τ_{b_j} can all be determined by the decision maker according to the experience or in a trial and error manner.

According to Bellman and Zadeh [5], let $\lambda = \min_{i,j}\{\mu_{\widetilde{G}_i}(\vec{\mathbf{x}}), \mu_{\widetilde{H}_j}(\vec{\mathbf{x}})\}$; then the model to determine the value of λ is formulated as follows:

$$\max \lambda \tag{8a}$$

subject to

$$\lambda \le \mu_{\widetilde{G}_i}(\vec{\mathbf{x}}), \quad i = 1, 2, \ldots, n_{fg} \tag{8b}$$

$$\lambda \le \mu_{\widetilde{H}_j}(\vec{\mathbf{x}}), \quad j = 1, 2, \ldots, n_{fh}. \tag{8c}$$

3.3. The Formulation of the Cooperative Fuzzy Game Model. Combined the model (8a), (8b), and (8c) with the bargaining function expressed as (3), a cooperative fuzzy game model is formulated as follows:

$$\max B(\vec{\mathbf{x}}) + p\lambda \tag{9a}$$

subject to

$$\lambda \le \mu_{\widetilde{G}_i}(\vec{\mathbf{x}}), \quad i = 1, 2, \ldots, n_{fg} \tag{9b}$$

$$\lambda \le \mu_{\widetilde{H}_j}(\vec{\mathbf{x}}), \quad j = 1, 2, \ldots, n_{fh} \tag{9c}$$

$$\vec{\mathbf{x}} \in S^* = \{X \in S \mid f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w) \ge 0\} \subset S, \tag{9d}$$

where $B(\vec{\mathbf{x}}) = \prod_{i=1}^{m}(f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w))^{w_i}$ and the parameter "p" in formula (9a) is determined by the decision maker.

As pointed out by Dhingra and Rao [10], the objective function max $B(\vec{\mathbf{x}}) + p\lambda$ can reflect the tradeoff between the value of $B(\vec{\mathbf{x}})$ and the degree of constraint violation $1 - \lambda$.

3.4. Fuzzy Pareto-Optimality. The cooperative game is based on the concept of a Pareto-optimal solution. Considering a multiobjective problem as follows:

$$\max f(\vec{\mathbf{x}}) = (f_1(\vec{\mathbf{x}}), f_2(\vec{\mathbf{x}}), \ldots, f_m(\vec{\mathbf{x}}))^T, \tag{10a}$$

subject to

$$\vec{\mathbf{x}} \in S = \{\vec{\mathbf{x}} \in R^n \mid g_i(\vec{\mathbf{x}}) \le a_i, h_j(\vec{\mathbf{x}}) = b_j\}, \tag{10b}$$

where $f_1(\vec{\mathbf{x}}), f_2(\vec{\mathbf{x}}), \ldots, f_m(\vec{\mathbf{x}})$ are objective functions, $\vec{\mathbf{x}}$ is the vector of decision variables, and S is the set of feasible solutions.

For a multiple objective optimization problem with partly fuzzy constraints, the concept of Pareto-optimality used for optimization problems with crisp constraints needs to be revised to introduce the concept of a fuzzy Pareto-optimal solution. Thus Dhingra and Rao [10] extended the definition of Pareto-optimality as follows.

Let $f_i : R^n \to R$, $i = 1, 2, \ldots, m$, be the objective functions, $\mu_{\widetilde{G}_i} : R^n \to [0, 1]$, $i = 1, 2, \ldots, n_{fg}$, and $\mu_{\widetilde{H}_j} : R^n \to [0, 1]$; $j = 1, 2, \ldots, n_{fh}$ be the membership functions of fuzzy constraints. A solution $\vec{\mathbf{x}}^* \in S$ is said to be fuzzy Pareto-optimal if and only if, for any $\vec{\mathbf{x}}_0 \in S$, $f_i(\vec{\mathbf{x}}_0) \le f_i(\vec{\mathbf{x}}^*)$, $i = 1, \ldots, m$ with at least one stringent inequality, $\mu_{\widetilde{G}_i}(\vec{\mathbf{x}}_0) \ge \mu_{\widetilde{G}_i}(\vec{\mathbf{x}}^*)$, $i = 1, 2, \ldots, n_{fg}$ with at least one stringent inequality, and $\mu_{\widetilde{H}_j}(\vec{\mathbf{x}}_0) \ge \mu_{\widetilde{H}_j}(\vec{\mathbf{x}}^*)$, $j = 1, 2, \ldots, n_{fh}$ with at least one stringent inequality.

As pointed out by Dhingra and Rao [10], since the set of alternatives S is convex and compact, there exists an optimal solution of the problem (9a), (9b), (9c), and (9d) $\vec{\mathbf{x}}^* \in S$ and it is fuzzy Pareto-optimal for the parameter $p \ge 0$.

4. Programming Model Formulation

4.1. Notation. The notation used in this study can be summarized as follows:

CR_i is the ith customer requirement, $i = 1, 2, \ldots, m$;

EC_j is the jth engineering characteristic, $j = 1, 2, \ldots, n$;

r_{ij} is the strength of the correlation measure between CR_i and EC_j;

$R = (r_{ij})$ is the strength matrix between CRs and ECs;

w_i is the relative importance of CR_i, $i = 1, 2, \ldots, m$;

$\mathbf{w} = (w_1, w_2, \ldots, w_n)$ is the vector of the relative importance of CRs;

p_{jk} is the strength of the correlation measure between EC_j and EC_k;

$\mathbf{p}_j = (p_{j1}, p_{j2}, \ldots, p_{jn})$ is the jth row vector of the matrix $P = (p_{jk})_{n\times n}$, $j = 1, 2, \ldots, n$;

l_j is the value of EC_j, $j = 1, 2, \ldots, n$;

x_j is the level of attainment of EC_j, $0 \le x_j \le 1$, $j = 1, 2, \ldots, n$;

v_j is the relative importance of EC_j, $j = 1, 2, \ldots, n$;

$C(\vec{\mathbf{x}})$ is the total cost of product development, and it is a function varying with the vector $\vec{\mathbf{x}} = (x_1, x_2, \ldots x_n)$;

C_F is the fixed part of the development cost;

C_v is the variable part of the development cost;

c_j is the unit cost for x_j, $j = 1, 2, \ldots, n$;

T is the budget of the product development; and

t is the index that the upper bound of T can be improved.

4.2. Normalization of the Values of ECs. To cover all types of inputs, l_j should be normalized to a scale $[0, 1]$. The "smaller-the-better type (S-type)" and "larger-the-better type (L-type)" ECs can be normalized using the following formulas (11) and (12), respectively. Consider

$$x_j = \frac{l_j^{\max} - l_j}{l_j^{\max} - l_j^{\min}} \tag{11}$$

$$x_j = \frac{l_j - l_j^{\min}}{l_j^{\max} - l_j^{\min}}. \tag{12}$$

For L-type, $l_j^{\min}$ is the minimum value of EC_j that matches the performance of the main competitors and $l_j^{\max}$ is the maximized physical limit. Conversely, for S-type, $l_j^{\min}$ is the minimized physical limit minimum and $l_j^{\max}$ is the maximum value of EC_j that matches the performance of the main competitors.

4.3. Calculation of v_j. The relative importance of ECs, v_j, $j = 1, 2, \ldots, n$, can be calculated as

$$v_j = \frac{v'_j}{\sum_{j=1}^{n} v'_j}, \tag{13a}$$

$$v'_j = \mathbf{w} R \mathbf{p}_j^T. \tag{13b}$$

4.4. The Development Cost with Its Fuzzification. The development cost $C(\vec{\mathbf{x}})$ can be viewed as the sum of the fixed cost C_F and the variable cost C_v, where C_v is the sum of x_j with the unit cost c_j. Therefore the calculation formula of the development cost $C(\vec{\mathbf{x}})$ can be expressed as follows:

$$C(\vec{\mathbf{x}}) = C_F + C_V = C_F + \sum_{j=1}^{n} c_j x_j. \tag{14}$$

If the total cost of product development $C(\vec{\mathbf{x}})$ is constrained to a budget T, it can be expressed as $C(\vec{\mathbf{x}}) \leq T$.

In practical problem, the design team often needs to improve the upper limit of the budget to enhance the levels of ECs. Considering the budget T that can be expanded to $T + t$ $(t > 0)$ as it is needed, where t denotes the distance by which the upper bound of the budget can be moved, we can fuzzify the cost constraint as

$$\mu_{\tilde{C}}(\vec{\mathbf{x}}) = \begin{cases} 1, & C(\vec{\mathbf{x}}) < T, \\ \dfrac{T + t - C(\vec{\mathbf{x}})}{t}, & T \leq C(\vec{\mathbf{x}}) \leq T + t, \\ 0, & C(\vec{\mathbf{x}}) > T + t. \end{cases} \tag{15}$$

4.5. Development of the Programming Model. In this subsection, we will develop a model to determine the target values of ECs, in which the objective of the programming model is to maximize the overall customer satisfaction and to exceed the main competitors.

The overall customer satisfaction can be obtained by aggregating the membership functions of the x_j, $u_j(x_j)$, $j = 1, 2, \ldots, n$, and their relative weights v_j, $j = 1, 2, \ldots, n$. Existing research often utilizes the sum with weight to aggregate $\mu_j(x_j)$ and v_j. As introduced in Section 3, the bargaining function $B(\vec{\mathbf{x}}) = \prod_{i=1}^{m} (f_i(\vec{\mathbf{x}}) - f_i(\vec{\mathbf{x}}_w))^{w_i}$ is similar to the geometric mean with weight. So we formulate the bargaining function $B(\vec{\mathbf{x}})$ as

$$B(\vec{\mathbf{x}}) = \prod_{j=1}^{n} u_j(x_j)^{v_j}, \tag{16}$$

where the payoff function of the player j is $u_j(x_j)$ and its worst value is zero. Indeed $B(\vec{\mathbf{x}}) = \prod_{j=1}^{n} u_j(x_j)^{v_j}$ is the geometric mean with weight for the membership function $u_j(x_j)$, $j = 1, 2, \ldots, n$, and it also can represent the overall customer satisfaction. This function can realize the tradeoff amongst ECs. Therefore, the programming model is formulated as follows:

$$\max \; B(\vec{\mathbf{x}}) + p\lambda \tag{17a}$$

subject to

$$\lambda \leq \mu_{\tilde{C}}(\vec{\mathbf{x}}) \tag{17b}$$

$$\vec{\mathbf{x}} = (x_1, x_2, \ldots, x_n) \in [0, 1]^n, \tag{17c}$$

where $B(\vec{\mathbf{x}}) = \prod_{j=1}^{n} u_j(x_j)^{v_j}$ and the parameter "p" is determined by the decision maker.

Since the feasible set $[0, 1]^n$ is convex and compact, there exists a fuzzy Pareto-optimal solution of the problems (17a), (17b), and (17c) $\vec{\mathbf{x}}^* \in S$ for the parameter $p \geq 0$.

5. An Illustrated Example

5.1. Building a HOQ for the Motor Car. In QFD, target values of ECs identify the definitive and quantitative technical specifications to satisfy CRs. The main objective of QFD-based product planning is to determine the target values of ECs for a new product to maximize the overall customer satisfaction with the given limited resources. In this section we will illustrate the proposed methodology by using a design of motor car (Chen et al. 2005, 2008) [20, 25].

A corporation is developing a new type of motor car. As depicted in Table 1, five CRs are identified to represent the biggest concerns of the customers. They are "reducing the noise of the car" (CR_1), "enhancing the acceleration" (CR_2), "saving fuel" (CR_3), "improving security" (CR_4), and "seat comfort" (CR_5). Their relative weights are determined by analytic hierarchy process (AHP) and listed in the second column of the Table 1. Once CRs are identified, the ECs are tabulated in the house of quality in order to satisfy CRs. Based on the design team's experience and expert knowledge on this product, five ECs are determined, that is, "reducing the noise of the exhaust system" (EC_1), "increasing the horsepower of the engine" (EC_2), "reducing the amount of fuel per mile" (EC_3), "increasing the controlling force of the braking system" (EC_4), and "enlarging the space of the seat" (EC_5). These ECs are measured in units of dB, Horsepower, Gallon, Kg, and M^3, respectively. The negative and positive sign on ECs mean that the design team hopes to reduce and increase the target values of ECs, that is, EC_1, EC_3 belong to "S-type," and others belong to "L-type". The QFD team will identify the strength of the relationship between CRs and ECs. These relationships are indicated in the relationship matrix between the CRs and ECs. According to formulas (13a) and (13b), the relative importance of the five ECs is calculated as $(v_1, v_2, v_3, v_4, v_5) = (0.30, 0.19, 0.24, 0.19, 0.08)$, which are shown in the bottom of the HOQ. The level values of ECs of five main competitors, Comp1, Comp2, Comp3, Comp4, and Comp5, are shown in the HOQ. The objective of the design team is to determine the target values of ECs for our product, so that the overall customer satisfaction of our product can exceed the main competitors.

The HOQ for the motor car design is shown in Table 1.

Table 1: The house for the motor car, Chen and Ngai [25].

CRs	Weights of CRs	−	+	−	+	+
		Correlation matrix				
	ECs	EC_1 x_1	EC_2 x_2	EC_3 x_3	EC_4 x_4	EC_5 x_5
	EC_1	1	0	0	0	0
	EC_2	0	1	0.2	0	0
	EC_3	0	0.2	1	0	0
	EC_4	0	0	0	1	0
	EC_5	0	0	0	0	1
CRs	Weights of CRs	Relationship matrix				
CR_1	0.31	1	0	0.2	0	0
CR_2	0.25	0	0.6	0	0	0
CR_3	0.16	0	0	1	0	0
CR_4	0.20	0	0	0	1	0
CR_5	0.08	0	0	0	0	1
		Technical matrix				
	Units	dB	Horsepower	Gallon	Kg	M^3
	$Comp_1$	80	75	0.042	25	0.18
	$Comp_2$	65	65	0.034	24	0.20
	$Comp_3$	65	80	0.028	23	0.18
	$Comp_4$	75	60	0.032	15	0.14
	$Comp_5$	95	80	0.030	20	0.19
	min	60	55	0.027	13	0.12
	max	100	90	0.044	27	0.21
	Relative weights of ECs	0.30	0.19	0.24	0.19	0.08

5.2. Normalizing the Values of ECs. The values of EC_1 and EC_3 for the five competitors are normalized by using (11), and the values of EC_2, EC_4, and EC_5 for the five competitors are normalized are by using (12). The normalization results for the five ECs of the five competitors are listed in Table 2.

5.3. Representing Design Uncertainty and Fuzzy Cost. To represent the design uncertainty, Chen and Ngai [25] defined a kind of membership function for a trapezoidal fuzzy number. The membership functions of the five ECs formulated by Chen and Ngai [25] are as

$$\begin{aligned} u_1(x_1) &= x_1^{0.2}, \quad 0 \le x_1 \le 1 \\ u_2(x_2) &= x_2^{2}, \quad 0 \le x_2 \le 1 \\ u_3(x_3) &= x_3^{0.2}, \quad 0 \le x_3 \le 1 \\ u_4(x_4) &= x_4, \quad 0 \le x_4 \le 1 \\ u_5(x_5) &= x_5^{4}, \quad 0 \le x_5 \le 1. \end{aligned} \tag{18}$$

The above membership functions of the five ECs are depicted in Figure 1.

The fixed cost C_F for the basic design, the unit cost for the five ECs, the development budget T, and its telescopic indicator t are listed in Table 3.

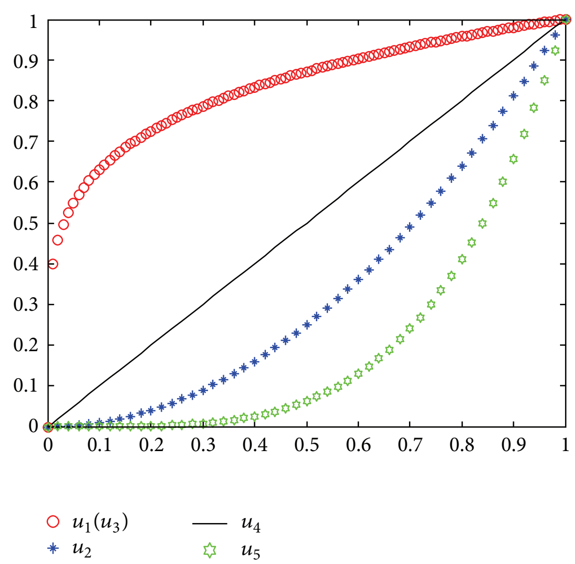


Figure 1: Membership functions of ECs.

Therefore, the development cost $C(\vec{\mathbf{x}})$ for the motor car design can be expressed as

$$C(\vec{\mathbf{x}}) = 50 + 25x_1 + 10x_2 + 15x_3 + 10x_4 + 8x_5. \tag{19}$$

Table 2: Normalization of the ECs of the five competitors.

	x_1	x_2	x_3	x_4	x_5
Comp$_1$	0.5000	0.5714	0.1176	0.8571	0.6667
Comp$_2$	0.8750	0.4286	0.5882	0.7857	0.8889
Comp$_3$	0.8750	0.7143	0.9412	0.7143	0.6667
Comp$_4$	0.6250	0.1429	0.7059	0.1429	0.2222
Comp$_5$	0.1250	0.7143	0.8235	0.5000	0.7778

Table 3: The fixed cost, unit cost, and budget (units).

C_F	c_1	c_2	c_3	c_4	c_5	T	t
50	25	10	15	10	8	75	5

Table 4: Membership degree for ECs and overall customer satisfaction of the five competitors.

	$\mu_1(x_1)$	$\mu_2(x_2)$	$\mu_3(x_3)$	$\mu_4(x_4)$	$\mu_5(x_5)$	$B(\vec{\mathbf{x}})$
Comp$_1$	0.8706	0.3265	0.6518	0.8571	0.1976	0.5969
Comp$_2$	0.7936	0.1837	0.8993	0.7857	0.6243	0.6064
Comp$_3$	0.9736	0.5495	0.9880	0.7143	0.1976	0.7274
Comp$_4$	0.9103	0.0204	0.9327	0.1429	0.0024	0.1946
Comp$_5$	0.6598	0.5102	0.9619	0.5000	0.3660	0.6225

Table 5: Target values of ECs with different value of the parameter "p."

p	x_1	x_2	x_3	x_4	x_5	$C(\vec{\mathbf{x}})$
0	0.0603	0.9490	0.0806	0.4794	1.0000	75
0.05	0.0599	0.9557	0.0804	0.4775	0.9954	75
0.50	0.0609	0.9497	0.0805	0.4832	0.9925	75
1	0.0602	0.9533	0.0796	0.4878	0.9862	75
2	0.0606	0.9621	0.0813	0.4844	0.9749	75
5	0.0600	0.9579	0.0799	0.4738	0.9981	75
20	0.0605	0.9522	0.0805	0.4782	0.9971	75

Considering the upper bound of the budget to be improved as it is needed, the membership function of the fuzzy cost can be formulated as

$$\mu_{\tilde{C}}(\vec{\mathbf{x}}) = \begin{cases} 1, & C(\vec{\mathbf{x}}) < 75, \\ \dfrac{80 - C(\vec{\mathbf{x}})}{80 - 75}, & 75 \le C(\vec{\mathbf{x}}) \le 80, \\ 0, & C(\vec{\mathbf{x}}) > 80. \end{cases} \quad (20)$$

5.4. Results and Discussion

5.4.1. Analysis of Results. According to the formulas (16) and (18), the results about the membership degree for ECs and the overall customer satisfaction of the five competitors are listed in Table 4, where the overall customer satisfaction of Comp$_3$ is 0.7274, which is largest amongst five competitors.

Combined the formulas (16), (18), and (20), the solution for the cooperative fuzzy game models (17a), (17b), and (17c) with different value of the parameter "p" is tabulated as Table 5. From Table 5 and Figure 2, it can be seen that the total cost is still 75, but the varying of the parameter "p" can facilitate the good performance in one EC to compensate for poor performance in other ECs slightly.

Table 6: The overall customer satisfaction with different value of the parameter "p."

p	$u_1(x_1)$	$u_2(x_2)$	$u_3(x_3)$	$u_4(x_4)$	$u_5(x_5)$	$B(\vec{\mathbf{x}})$
0	0.5702	0.9006	0.6043	0.4794	1.0000	0.6383
0.05	0.5696	0.9133	0.6041	0.4775	0.9817	0.6383
0.50	0.5715	0.9019	0.6041	0.4832	0.9705	0.6383
1	0.5701	0.9088	0.6028	0.4878	0.9458	0.6383
2	0.5709	0.9256	0.6054	0.4844	0.9033	0.6383
5	0.5696	0.9175	0.6033	0.4738	0.9924	0.6383
20	0.5706	0.9067	0.6042	0.4782	0.9883	0.6383

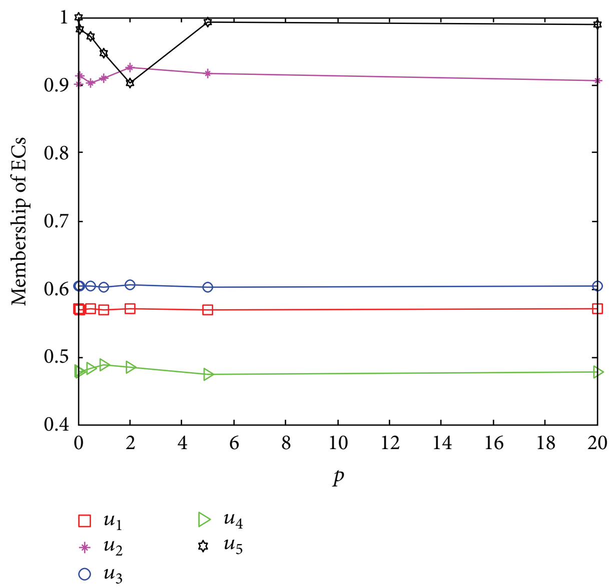


Figure 2: Membership of the five ECs with different value of "p."

The membership degree for ECs and their overall customer satisfaction with different value of the parameter "p" are shown in Table 6.

From Table 6 it can be seen that the overall customer satisfaction obtained from the proposed method is always 0.6383 though the value of the parameter "p" varies from 0 to 20. Indeed, as introduced in Section 3, because the feasible set $[0,1]^n$ is convex and compact, there exists a fuzzy Pareto-optimal solution of the problems (17a), (17b), and (17c) $\vec{\mathbf{x}}^* \in S$ for the parameter $p \ge 0$.

Moreover, the result in Table 6 shows that the overall customer satisfaction ($B(\vec{\mathbf{x}}) = 0.6383$) obtained from the proposed method exceeds four competitors only smaller than Comp$_3$ (0.7274).

5.4.2. Further Discussion. As discussed in Section 5.4.1, when the budget is limited as 75, the overall customer satisfaction $B(\vec{\mathbf{x}})$ is 0.6383, which is Pareto-optima. If we hope that the overall customer satisfaction of our new product exceeds all competitors, we must improve the budget. So we set the budget T as 70 and 80, respectively, and the telescopic

Table 7: Results with different budget when $p = 1$.

T	x_1	x_2	x_3	x_4	x_5	$B(\vec{\mathbf{x}})$
70	0.0480	0.7607	0.0638	0.3824	0.8015	0.5109
75	0.0602	0.9533	0.0796	0.4878	0.9862	0.6383
80	0.0967	1.0000	0.1291	0.7646	1.0000	0.7487

indicator t is still set as 5. For comparison, the results with different budget are listed in Table 7 when the parameter $p = 1$.

From Table 7, it can be seen that the overall customer satisfaction of our new product can exceed all competitors when we set the budget as 80.

6. Conclusion

In this study, to enhance the overall customer satisfaction, a cooperative game fuzzy framework is developed to determine the target values of the ECs in QFD, where each player corresponds to the membership function of ECs. The formulation of the bargaining function is the key in the proposed approach. A motor car product design is cited to illustrate the proposed approach. Results show that the overall customer satisfaction for the ECs obtained from the proposed methodology can exceed the main competitors. The advantage of the proposed methodology is that the solution for the model with limited resources is Pareto-optimal. Meanwhile, the varying of the parameter "p" can facilitate the good performance in one EC to compensate for poor performance in other ECs. It is important to note that there is no model that employs the cooperative fuzzy game modeling approach over QFD analysis.

Existing methods for determining the target levels of ECs in QFD often consider CRs and the relationships between CRs and ECs acquired previously. Therefore, it is very difficult that a new product or service fully meets customer expectations when it is ready to market. In order to tackle this problem, it is necessary to embed the dynamics customer requirements into QFD. For future research, we would like to develop fuzzy game framework to determine the target levels of ECs of the new product by considering future requirements that meet customer needs.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The work described in this paper was supported in part by the National Natural Science Foundation of China (no. NSFC71272177), the Science Foundation of Education Committee of Jiangxi, China (no. GJJ14469), and the funds of Innovation Program of Shanghai Municipal Education Committee, China (no. 12ZS101).

References

[1] J. F. Nash, "The bargaining problem," *Econometrica*, vol. 18, pp. 155–162, 1950.

[2] J. Nash, "Two-person cooperative games," *Econometrica*, vol. 21, pp. 128–140, 1953.

[3] M. Larbani, "Non cooperative fuzzy games in normal form: a survey," *Fuzzy Sets and Systems*, vol. 160, no. 22, pp. 3184–3210, 2009.

[4] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965.

[5] R. E. Bellman and L. A. Zadeh, "Decision-making in a fuzzy environment," *Management Science*, vol. 17, no. 4, pp. B141–B164, 1970.

[6] Y. Pan, M. J. Er, D. Huang, and Q. Wang, "Adaptive fuzzy control with guaranteed convergence of optimal approximation error," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 807–818, 2011.

[7] Y. Pan, M. J. Er, D. Huang, and T. Sun, "Practical adaptive fuzzy H_∞ tracking control of uncertain nonlinear systems," *International Journal of Fuzzy Systems*, vol. 14, no. 4, pp. 463–473, 2012.

[8] Y. P. Pan, M. J. Er, D. Huang, and Q. Wang, "Adaptive fuzzy control with guaranteed convergence of optimal approximation error," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 5, pp. 807–818, 2011.

[9] J. Aubin, *Mathematical Methods of Game and Economic Theory*, vol. 7, North-Holland, Amsterdam, The Netherlands, 1982.

[10] A. K. Dhingra and S. S. Rao, "A cooperative fuzzy game theoretic approach to multiple objective design optimization," *European Journal of Operational Research*, vol. 83, no. 3, pp. 547–567, 1995.

[11] Y. Chen, J. Tang, R. Y. K. Fung, and Z. Ren, "Fuzzy regression-based mathematical programming model for quality function deployment," *International Journal of Production Research*, vol. 42, no. 5, pp. 1009–1027, 2004.

[12] Y. Akao, *Quality Function Deployment: Integrating Customer Requirements into Product Design*, Productivity Press, Cambridge, Mass, USA, 1990.

[13] J. R. Hauser and D. Clausing, "The house of quality," *Harvard Business Review*, vol. 66, no. 3, pp. 63–73, 1988.

[14] M. Zhou, "Fuzzy logic and optimization models for implementing QFD," *Computers and Industrial Engineering*, vol. 35, no. 1–4, pp. 237–240, 1998.

[15] R. Y. K. Fung, K. Popplewell, and J. Xie, "An intelligent hybrid system for customer requirements analysis and product attribute targets determination," *International Journal of Production Research*, vol. 36, no. 1, pp. 13–34, 1998.

[16] K. Kim, H. Moskowitz, A. Dhingra, and G. Evans, "Fuzzy multicriteria models for quality function deployment," *European Journal of Operational Research*, vol. 121, no. 3, pp. 504–518, 2000.

[17] J. Tang, R. Y. K. Fung, B. Xu, and D. Wang, "A new approach to quality function deployment planning with financial consideration," *Computers and Operations Research*, vol. 29, no. 11, pp. 1447–1463, 2002.

[18] H. Bai and C. K. Kwong, "Inexact genetic algorithm approach to target values setting of engineering requirements in QFD," *International Journal of Production Research*, vol. 41, no. 16, pp. 3861–3881, 2003.

[19] E. E. Karsak, "Fuzzy multiple objective programming framework to prioritize design requirements in quality function

deployment," *Computers and Industrial Engineering*, vol. 47, no. 2-3, pp. 149–163, 2004.

[20] Y. Chen, R. Y. K. Fung, and J. Tang, "Fuzzy expected value modelling approach for determining target values of engineering characteristics in QFD," *International Journal of Production Research*, vol. 43, no. 17, pp. 3583–3604, 2005.

[21] X. Lai, M. Xie, and K. C. Tan, "Dynamic programming for QFD optimization," *Quality and Reliability Engineering International*, vol. 21, no. 8, pp. 769–780, 2005.

[22] Y. Chen and L. Chen, "A non-linear possibilistic regression approach to model functional relationships in product planning," *International Journal of Advanced Manufacturing Technology*, vol. 28, no. 11-12, pp. 1175–1181, 2006.

[23] R. Y. K. Fung, Y. Chen, and J. Tang, "Estimating the functional relationships for quality function deployment under uncertainties," *Fuzzy Sets and Systems*, vol. 157, no. 1, pp. 98–120, 2006.

[24] L. Chen and M. Weng, "An evaluation approach to engineering design in QFD processes using fuzzy goal programming models," *European Journal of Operational Research*, vol. 172, no. 1, pp. 230–248, 2006.

[25] Y. Z. Chen and E. W. T. Ngai, "A fuzzy QFD program modelling approach using the method of imprecision," *International Journal of Production Research*, vol. 46, no. 24, pp. 6823–6840, 2008.

[26] Z. Sener and E. E. Karsak, "A decision model for setting target levels in quality function deployment using nonlinear programming-based fuzzy regression and optimization," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 9-12, pp. 1173–1184, 2010.

[27] L. H. Chen and W. C. Ko, "Fuzzy approaches to quality function deployment for new product design," *Fuzzy Sets and Systems*, vol. 160, no. 18, pp. 2620–2639, 2009.

[28] E. K. Delice and Z. Güngör, "A new mixed integer linear programming model for product development using quality function deployment," *Computers and Industrial Engineering*, vol. 57, no. 3, pp. 906–912, 2009.

[29] C. K. Kwong, Y. Chen, K. Y. Chan, and X. Luo, "A generalised fuzzy least-squares regression approach to modelling relationships in QFD," *Journal of Engineering Design*, vol. 21, no. 5, pp. 601–613, 2010.

[30] Z. Güngör, E. K. Delice, and S. E. Kesen, "New product design using FDMS and FANP under fuzzy environment," *Applied Soft Computing Journal*, vol. 4, no. 11, pp. 3347–3356, 2011.

[31] H. Liu, "Product design and selection using fuzzy QFD and fuzzy MCDM approaches," *Applied Mathematical Modelling*, vol. 35, no. 1, pp. 482–496, 2011.

[32] Z. Sener and E. E. Karsak, "A combined fuzzy linear regression and fuzzy multiple objective programming approach for setting target levels in quality function deployment," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3015–3022, 2011.

[33] Z. Yang and Y. Chen, "Fuzzy soft set-based approach to prioritizing technical attributes in quality function deployment," *Neural Computing and Applications*, vol. 23, no. 7-8, pp. 2493–2500, 2013.

[34] H. Jiang, C. K. Kwong, W. H. Ip, and Z. Chen, "Chaos-based fuzzy regression approach to modeling customer satisfaction for product design," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 5, pp. 926–936, 2013.

[35] E. K. Delice and Z. Güngör, "Determining design requirements in QFD using fuzzy mixed-integer goal programming: Application of a decision support system," *International Journal of Production Research*, vol. 51, no. 21, pp. 6378–6396, 2013.

[36] W. C. Ko and L. H. Chen, "An approach of new product planning using quality function deployment and fuzzy linear programming model," *International Journal of Production Research*, vol. 52, no. 6, pp. 1728–1743, 2014.

[37] S. Mungle, S. Saurav, and M. K. Tiwari, "Multi-objective optimization approach to product- planning in quality function deployment incorporated with Fuzzy-ANP," in *Applications of Multi-Criteria and Game Theory Approaches*, Springer Series in Advanced Manufacturing, pp. 83–105, Springer, London, UK, 2014.

[38] K. K. F. Yuen, "A hybrid fuzzy quality function deployment framework using cognitive network process and aggregative grading clustering: an application to cloud software product development," *Neurocomputing*, 2014.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 471016, 7 pages
http://dx.doi.org/10.1155/2014/471016

Research Article

Some Single-Machine Scheduling Problems with Learning Effects and Two Competing Agents

Hongjie Li,[1] Zeyuan Li,[2] and Yunqiang Yin[3,4]

[1] *Mathematics Department, Zhoukou Normal University, Zhoukou 466001, China*
[2] *College of Mechanical and Electrical, Harbin Institute of Technology, Harbin 150001, China*
[3] *College of Sciences, East China Institute of Technology, Nanchang 330013, China*
[4] *Faculty of Science, Kunming University of Science and Technology, Kunming 650093, China*

Correspondence should be addressed to Yunqiang Yin; yunqiangyin@gmail.com

Received 29 April 2014; Accepted 10 May 2014; Published 25 May 2014

Academic Editor: Dar-Li Yang

This study considers a scheduling environment in which there are two agents and a set of jobs, each of which belongs to one of the two agents and its actual processing time is defined as a decreasing linear function of its starting time. Each of the two agents competes to process its respective jobs on a single machine and has its own scheduling objective to optimize. The objective is to assign the jobs so that the resulting schedule performs well with respect to the objectives of both agents. The objective functions addressed in this study include the maximum cost, the total weighted completion time, and the discounted total weighted completion time. We investigate three problems arising from different combinations of the objectives of the two agents. The computational complexity of the problems is discussed and solution algorithms where possible are presented.

1. Introduction

In traditional scheduling research, it is commonly assumed that the processing times of the jobs remain unchanged throughout the scheduling horizon. However, under certain circumstances, the job processing times may become short due to learning effects in the production environment. For example, Biskup [1] points out that the repeated processing of similar tasks will improve workers' efficiency; that is, it takes workers shorter times to process setups, operate machines or software, or handle raw materials and components. In such an environment, a job scheduled later will consume less time than the same job when scheduled earlier. Jobs in such a setting are said to be under the "learning effect" in the literature.

Biskup [1] and Cheng and Wang [2] first introduce the idea of learning into the field of scheduling independently. Since then, a large body of literature on scheduling with learning effects has emerged. Examples of such studies are Mosheiov [3], Mosheiov and Sidney [4], Bachman and Janiak [5], Janiak and Rudek [6], Wang [7], and Yin et al. [8]. Biskup [9] provides a comprehensive review of research on scheduling with learning effects. For more recent studies in this line of research, the reader is referred to Jiang et al. [10, 11], Yang [12], S.-J. Yang and D.-L. Yang [13], Wang et al. [14], Wu et al. [15], Xu et al. [16], and Yin et al. [8].

All the above papers consider the traditional case with a single agent. In recent years scheduling researchers have increasingly considered the setting of multiple competing agents, in which multiple agents need to process their own sets of jobs, competing for the use of a common resource and each agent has its own objective to optimize. However, there is little scheduling research in the multiagent setting in which the jobs are under learning effects. Liu et al. [17] study two models with two agents and position-dependent processing times. They assume that the actual processing time of job J_j is p_j+br in the aging-effect model, while the actual processing time of J_j is $p_j - br$ in the learning-effect model, where r represents the processed position of J_j and $b > 0$ denotes the aging or learning index. Ho et al. [18] define the actual processing time of job J_j as $p_j = a_j(1 - kt)$ if it is processed at time t, where a_j denotes the normal processing time of job J_j

and $k \geq 0$ represents a constant such that $k(t_0 + \sum_{j=1}^{n} a_j - a_{\min}) < 1$ with $a_{\min} = \min_{j=1,2,\ldots,n}\{a_j\}$. Inspired by Ho et al. [18], Yin et al. [19] consider some two-agent scheduling problems under the learning effect model proposed in Ho et al. [18], in which the objective functions for agent A include the maximum earliness cost, the total earliness cost, and the total weighted earliness cost, and the objective function for agent B is always the same, that is, maximum earliness cost, and the objective is to minimize the objective function of agent A while keeping the objective function of agent B not greater than a given level. Similar models have been further studied by Wang and Xia [20], Wang [21], and so on. For the other related two-agent works without time-dependent processing times, the reader can refer to Baker and Smith [22], Agnetis et al. [23, 24], Cheng et al. [25, 26], Ng et al. [27], Mor and Mosheiov [28], Lee et al. [29], Leung et al. [30], Wan et al. [31], Yin et al. [19, 32], Yu et al. [33], and Zhao and Lu [34].

This study introduces a new scheduling model in which both the two-agent concept and the learning effects exist, simultaneously. We consider the following objective functions: the maximum cost, total completion time, total weighted completion time, and discounted total weighted completion time. The structural properties of optimal schedules are derived and polynomial time algorithms are developed for the problems where possible.

The remaining part of the study is organized as follows: Section 2 introduces the notation and terminology used throughout the paper. Sections 3–6 analyze the computational complexity and derive the optimal properties of the problems under study. The last section concludes the paper and suggests topics for future research.

2. Model Formulation

The problem investigated in this paper can be formally described as follows. Suppose that there are two agents A and B, each of whom has a set of nonpreemptive jobs. The two agents compete to process their jobs on a common machine. Agent A has to process the job set $J^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$, while agent B has to process the job set $J^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$. All the jobs are available for processing at time t_0, where $t_0 \geq 0$. Let $X \in \{A, B\}$. The jobs belonging to agent X are called A-jobs. Associated with each job J_j^X ($j \in \{1, 2, \ldots, n_X\}$), there are normal processing time a_j^X and weight w_j^X. Due to the learning effect, the actual processing time p_j of job J_j^X is defined as

$$p_j^X = a_j^X (1 - kt), \quad j = 1, 2, \ldots, n_X, \tag{1}$$

where $t \geq t_0$ denotes job's starting time and $k \geq 0$ represents constant such that $k(t_0 + \sum_{J_j^X \in J^A \cup J^B} a_j^X - a_{\min}) < 1$, where $a_{\min} = \min_{J_j^X \in J^A \cup J^B}\{a_j^X\}$ (see Ho et al. [18] for details).

Given a feasible schedule S of the $n = n_A + n_B$ jobs, we use $C_j^X(S)$ to denote the completion time of job J_j^X and omit the argument S whenever this does not cause confusion. The makespan of agent X is $C_{\max}^X = \max_{j=1,2,\ldots,n_X}\{C_j^X\}$. For each job J_j^X, let $f_j^X(\cdot)$ be a nondecreasing function. In this case, the maximum cost is defined as $f_{\max}^X = \max_{j=1,2,\ldots,n_X}\{f_j^X(C_j^X)\}$. The objective function of agent X considered in this paper includes the following: $f_{\max}^X$ (maximum cost), $\sum C_j^X$ (total completion time), $\sum w_j^X C_j^X$ (total weighted completion time), and $\sum w_j^X(1 - e^{-rC_j^X})$ (discounted total weighted completion time).

Using the three-field notation scheme $\alpha|\beta|\gamma$ introduced by Graham et al. [35], the problems considered in this paper are denoted as follows: $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A(C^A) : f_{\max}^B(C^B) \leq U$, $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum C_j^A : f_{\max}^B(C^B) \leq U$, $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A C_j^A : f_{\max}^B(C^B) \leq U$, and $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B(C^B) \leq U$.

Note that all the objective functions involved in the considered problems are regular; that is, they are nondecreasing in the job completion times. Hence there is no benefit in keeping the machine idle.

3. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$

In this section we address the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \leq U$ and show that it can be solved optimally in polynomial time. We first develop some structural properties of optimal schedules for the problem which will be used in developing the algorithm.

Lemma 1 (see [19]). *For problem $1 \mid p_j^X = a_j^X(1 - kt) \mid C_{\max}$, the makespan is equal to*

$$\begin{aligned} &\left(t_0 - \frac{1}{k}\right) \prod_{j=1}^{n_X} \left(1 - ka_j^X\right) + \frac{1}{k} \\ &= \left(t_0 - \frac{1}{k}\right) \prod_{j=1}^{n_A} \left(1 - ka_j^A\right) \prod_{j=1}^{n_B} \left(1 - ka_j^B\right) + \frac{1}{k}. \end{aligned} \tag{2}$$

In the sequel, we set $u = (t_0 - (1/k)) \prod_{j=1}^{n_X}(1 - ka_j^X) + (1/k)$. Then the following results hold.

Proposition 2. *For the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \leq U$, if there is a B-job J_k^B such that $f_k^B(u) \leq U$, then there exists an optimal schedule such that J_k^B is scheduled last and there is no optimal schedule where an A-job is scheduled last.*

Proof. Assume that S is an optimal schedule where the B-job J_h^B is not scheduled in the last position. Let π denote the set of jobs scheduled prior to job J_h^B. We construct from S a new schedule S' by moving job J_h^B to the last position and leaving the other jobs unchanged in S. Then, the completion times of the jobs processed before job J_h^B in S' are the same as that in S since there is no change for any job preceding J_h^B in S. The jobs belonging to π are scheduled earlier, so their completion times are smaller in S' by Lemma 1. It follows that $f_k^X(C_k^X(S')) \leq f_k^X(C_k^X(S))$ for any job J_k^X in π, where $X \in \{A, B\}$. By the assumption that $f_h^B(u) \leq U$, job J_h^B

is feasible in S', so schedule S' is feasible and optimal, as required. □

For each B-job J_j^B, let us define a deadline D_j^B such that $f_j^B(C_j^B) \leq U$ for $C_j^B \leq D_j^B$ and $f_j^B(C_j^B) > U$ for $C_j^B > D_j^B$ (if the inverse function $f_j^B(\cdot)$ is available, the deadlines can be evaluated in constant time; otherwise, this requires logarithmic time).

Proposition 3. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$*, there exists an optimal schedule where the B-jobs are scheduled according to the nondecreasing order of* D_j^B.

Proof. Assume that S is an optimal schedule where the B-jobs are not scheduled according to the nondecreasing order of D_j^B. Let J_l^B and J_h^B be the first pair of jobs such that $D_l^B > D_h^B$. In this schedule, job J_l^B is processed earlier; then a set of A-jobs, denoted as π, are consecutively processed and then job J_h^B. In addition, denote by π' the set of jobs processed after job J_h^B. We construct from S a new schedule S' by extracting job J_l^B, reinserting it just after job J_h^B and leaving the other jobs unchanged in schedule S. Then the completion times of the jobs processed prior to job J_l^B in S' are the same as that in S. By Lemma 1, the completion time of job J_h^B in S equals that of job J_l^B in S'; that is, $C_l^B(S') = C_h^B(S)$, so the completion times of the jobs belonging to π' are identical in both S and S'. Since S is feasible, it follows that $C_l^B(S') = C_h^B(S) \leq D_h^B < D_l^B$, so job J_l^B is feasible in S'. The π-jobs and job J_h^B are scheduled earlier in S', implying that their actual processing times are smaller in S', so their completion times are earlier in S', and thus they remain feasible. Therefore, schedule S' is feasible and optimal.

Thus, repeating doing this procedure for all the B-jobs not sequenced according to nondecreasing order of D_j^B completes the proof. □

Proposition 4. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$*, if* $f_k^B(u) > U$ *for any B-job* J_k^B*, then there exists an optimal schedule where the A-job with the smallest cost* $f_k^A(u)$ *is processed in the last position.*

Proof. Assume that S is an optimal schedule where the A-job with the smallest cost J_h^A, that is, $f_h^A(u) = \min_{J_j^A \in J^A}\{f_j^A(u)\}$, is not processed in the last position. By the hypothesis, the last job in schedule S is an A-job, say J_l^A. This means $f_h^A(u) < f_l^A(u)$. In this schedule, job J_h^A is scheduled earlier. Let π denote the set of jobs scheduled after job J_h^A and prior to job J_l^A. We construct from S a new schedule S' by extracting job J_h^A, reinserting it just after job J_l^A, and leaving the other jobs unchanged in schedule S. There is no change for any job preceding J_h^A in S. We claim the following.

(1) Schedule S' is feasible. First, the completion times of the jobs processed prior to job J_h^A in S' are the same as that in S'. Since the jobs belonging to π are scheduled earlier in S', their actual processing times are smaller in S', so their completion times are earlier in S'. It follows that $f_k^X(C_k^X(S')) \leq f_k^X(C_k^X(S))$ for any job J_k^X in π, where $X \in \{A, B\}$, as required.

(2) Schedule S' is a better schedule than S. By Lemma 1, the completion time of the last job J_l^A in S equals that of the last job J_h^A in S'; that is, $C_l^A(S) = C_h^A(S') = u$. Thus, to prove that S' is better than S, it suffices to show that

$$\begin{aligned} &\max\left\{f_l^A\left(C_l^X\left(S'\right)\right), f_h^A\left(u\right)\right\} \\ &\quad\leq \max\left\{f_h^A\left(C_h^X\left(S'\right)\right), f_l^A\left(u\right)\right\}. \end{aligned} \tag{3}$$

Since $f_k^A(\cdot)$ is a nondecreasing function of the completion time of job J_k^A and $C_l^X(S') < u$, we have $f_l^A(C_l^X(S')) \leq f_l^A(u)$. Thus $\max\{f_l^A(C_l^X(S')), f_h^A(u)\} \leq f_l^A(u)$, as required.

The result follows. □

Summing up the above analysis, our algorithm for problem $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$ can be formally described as in Algorithm 1.

Theorem 5. *Algorithm 1 solves problem* $1 \mid p_j^X = a_j^X(1-kt) \mid f_{\max}^A : f_{\max}^B \leq U$ *in* $O(n_A^2 + n_B \log n_B)$ *time.*

Proof. Step 1 requires a sorting operation of the B-jobs, which takes $O(n_B \log n_B)$ time. Step 2 takes $O(n_B)$ time since the calculation of the $f_j^B(\cdot)$ functions in Step 2 can be evaluated in constant time by the assumption. In Step 3 we calculate the $f_j^A(\cdot)$ value for all the remaining unscheduled A-jobs, which takes $O(n_A)$ time. Thus, after n_A iterations, Step 3 can be executed in $O(n_A^2)$ time. Therefore, the overall time complexity of the algorithm is indeed $O(n_A^2 + n_B \log n_B)$. □

4. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \leq U$

Leung et al. [30] show that problem $1 \| \sum w_j^A C_j^A : f_{\max}^B \leq U$ is NP-hard in the strong sense. Since our problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \leq U$ is a generalization of the problem $1 \| \sum w_j^A C_j^A : f_{\max}^B \leq U$, then so is our problem. In what follows we show that the problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \leq U$ is polynomially solvable if the A-jobs have reversely agreeable weights; that is, $a_i^A \leq b_j^A$ implies $w_i^A \geq w_j^A$ for all jobs J_i^A and J_j^A. It is clear that Propositions 2 and 3 still hold for this problem. We modify Proposition 4 as follows.

Proposition 6. *For the problem* $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \leq U$*, if the A-jobs have reversely agreeable weights, then there exists an optimal schedule where the A-jobs are assigned according to the nondecreasing order of* a_j^A/w_j^A*, that is, in the weighted shortest processing time (WSPT) order.*

Input: $n_A, n_B, U, p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.
Step 1. Set $h = n_B$, $J = J^A$, $f_{\max}^A = 0$ and $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k$,; solve D_j^B from $f_j^B(D_j^B) = U$ for $j = 1, 2, \ldots, n_B$ and renumber them according to the non-decreasing order such that $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;
Step 2. If $h \ge 1$, then
    If $t \le D_{[h]}^B$, then
        set $h = h - 1$, $t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign job $J_{[h]}^B$ at time t, and go to Step 2;
    Else
        go to Step 3;
Else
    go to Step 3;
Step 3. If $J \ne \emptyset$, then
    select the job J_l^A from J with the the smallest cost, that is, $f_l^A(t) = \min_{J_j^A \in J^A}\{f_j^A(t)\}$,
    set $f_{\max}^A = \max\{f_{\max}^A, f_l^A(t)\}$, $t = (t - a_l^A)/(1 - ka_l^A)$, assign job J_l^A at time t, delete J_l^A from J, and go to Step 4;
Elseif $h \ge 1$
    report that the instance is not feasible;
Else
    go to Step 4;
Step 4. If J is not empty or $h \ge 1$, then
    go to Step 2.
Else
    stop.

Algorithm 1

Proof. Assume that S is an optimal schedule where A-jobs are not scheduled in the WSPT order. Let J_l^A and J_h^A be the first pair of jobs such that $a_l^A/w_l^A > a_h^A/w_h^A$. Then $a_l^A \ge a_h^A$ and $w_l^A \le w_h^A$ due to the fact that the A-jobs have reversely agreeable weights. Assume that, in schedule S, job J_l^A starts its processing at time T; then a set of B-jobs are consecutively processed and then job J_h^A. In addition, let π' denote the set of jobs processed after job J_h^A. We construct a new scheduling S' from S by swapping jobs J_l^A and J_h^A and leaving the other jobs unchanged. We conclude the following.

(1) Schedule S' is feasible. By Lemma 1, the completion time of job J_h^A in S equals that of job J_l^A in S'; that is, $C_l^A(S') = C_h^A(S)$, so the completion times of the jobs belonging to π' are identical in both S and S'. Since $a_l^A \ge a_h^A$, we have $C_h^A(S') = T + a_h^A(1-kT) \le T + a_l^A(1-kT) = C_l^A(S)$. Hence the π-jobs are scheduled earlier in S', implying that their actual processing times are smaller in S', so their completion times are earlier in S'. Hence $f_k^B(C_k^B(S')) \le f_k^B(C_k^B(S))$ for any job J_k^B in π, as required.

(2) Schedule S' is better than S. By the proof of (1), it is sufficient to show that

$$w_h^A C_h^A(S') + w_l^A C_l^A(S') \le w_l^A C_l^A(S) + w_h^A C_h^A(S). \quad (4)$$

Since $C_h^A(S') \le C_l^A(S)$ and $C_l^A(S') = C_h^A(S)$, we have

$$\begin{aligned} & w_l^A C_l^A(S) + w_h^A C_h^A(S) - \left(w_h^A C_h^A(S') + w_l^A C_l^A(S')\right) \\ & \ge w_l^A C_h^A(S') + w_h^A C_l^A(S') - \left(w_h^A C_h^A(S') + w_l^A C_l^A(S')\right) \\ & = \left(w_l^A - w_h^A\right)\left(C_h^A(S') - C_l^A(S')\right) \\ & \ge 0, \end{aligned} \quad (5)$$

as required.

Thus, repeating this swapping argument for all the A-jobs not sequenced in the WSPT order yields the theorem. □

Based on the results of Propositions 2, 3, and 6, our algorithm to solve the problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A C_j^A : f_{\max}^B \le U$ for the case where the A-jobs have reversely agreeable weights can be formally described as in Algorithm 2.

Theorem 7. *The problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j C_j^A : f_{\max}^B \le U$ can be solved in $O(n_A \log n_A + n_B \log n_B)$ time by applying Algorithm 2 if all A-jobs have reversely agreeable weights.*

Proof. The correctness comes from the above analysis. Now we turn to the time complexity of the algorithm. Step 1 requires two sorting operations of the A-jobs and B-jobs, respectively, which take $O(n_A \log n_A)$ time and $O(n_B \log n_B)$ time, respectively. Both Steps 2 and 3 take $O(2)$ time.

Input: $n_A, n_B, U, w^A = (w_1^A, w_2^A, \ldots, w_{n_A}^A)$, $p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.
Step 1. Set $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k, l = n_A, h = n_B$, and $\sum w_j^A C_j^A = 0$;
sort the A-jobs according to the non-decreasing order of a_j^A/w_j^A, that is,
$a_{[1]}^A/w_{[1]}^A \le a_{[2]}^A/w_{[2]}^A \le \cdots \le a_{[n_A]}^A/w_{[n_A]}^A$;
calculate the deadlines of the B-jobs from $f_j^B(D_j^B) = U$ and renumber them
according to the non-decreasing order such that $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;
Step 2. If $h \ge 1$, then
If $t \le D_{[h]}^B$, then
set $h = h - 1, t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign job $J_{[h]}^B$ at time t, and go to Step 2;
Else
go to Step 3;
Else
go to Step 3;
Step 3. If $l \ge 1$, then
set $l = l - 1, \sum w_j^A C_j^A = \sum w_j^A C_j^A + w_{[l]}^A t, t = (t - a_{[l]}^A)/(1 - ka_{[l]}^A)$, assign $J_{[l]}^A$ at time t, and go to Step 4;
Elseif $h \ge 1$
output that the instance is not feasible;
Else
go to Step 4;
Step 4. If $h \ge 1$ or $l \ge 1$, then
go to Step 2.
Else
stop.

Algorithm 2

Therefore, the overall time complexity of the algorithm is indeed $O(n_A \log n_A + n_B \log n_B)$. □

5. Problem $1 \mid p_j^X = a_j^X(1-kt) \mid \sum w_j^A(1-e^{-rC_j^A}) : f_{\max}^B \le U$

This section address the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$. We show that it is polynomially solvable if the A-jobs have reversely agreeable weights. It is clear that Propositions 2 and 3 still hold for this problem. We give Proposition 8 as follows.

Proposition 8. *For the problem* $1 \mid p_j^X = a_j^X(1 - kt) \mid w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$, *if the A-jobs have reversely agreeable weights, then there exists an optimal schedule where the A-jobs are assigned according to the nondecreasing order of* $(1 - e^{-ra_j^A})/w_j^A e^{-ra_j^A}$, *that is, in the weighted discount shortest processing time (WDSPT) order.*

Proof. We adopt the same notation as that used in the proof of Proposition 6. Assume that $(1 - e^{-ra_l^A})/w_l^A e^{-ra_l^A} > (1 - e^{-rp_h^A})/w_h^A e^{-rp_h^A}$. Since A-jobs have reversely agreeable weights, we have $a_l^A \ge a_h^A$ and $w_l^A \le w_h^A$. Then by the proof of Proposition 6, we know that $C_h^A(S') \le C_l^A(S)$, $C_l^A(S') = C_h^A(S)$, and $C_k^A(S') = C_j^A(S)$ for all the other jobs $J_k^A \in J_A/\{J_l^A, J_k^A\}$ and that schedule S' is feasible. To show that S' is better than S, it is sufficient to show that

$$\begin{aligned} &w_h^A\left(1 - e^{-rC_h^A(S')}\right) + w_l^A\left(1 - e^{-rC_l^A(S')}\right) \\ &\quad \le w_l^A\left(1 - e^{-rC_l^A(S)}\right) + w_h^A\left(1 - e^{-rC_h^A(S)}\right). \end{aligned} \tag{6}$$

In fact, since $r \in (0, 1)$, $C_h^A(S') \le C_l^A(S)$, and $C_l^A(S') = C_h^A(S)$, we have

$$\begin{aligned} &w_l^A\left(1 - e^{-rC_l^A(S)}\right) + w_h^A\left(1 - e^{-rC_h^A(S)}\right) \\ &\quad - \left(w_h^A\left(1 - e^{-rC_h^A(S')}\right) + w_l^A\left(1 - e^{-rC_l^A(S')}\right)\right) \\ &= w_h^A e^{-rC_h^A(S')} + w_l^A e^{-rC_l^A(S')} - w_l^A e^{-rC_l^A(S)} - w_h^A e^{-rC_h^A(S)} \\ &\ge w_h^A e^{-rC_h^A(S')} + w_l^A e^{-rC_h^A(S')} - w_l^A e^{-rC_h^A(S')} - w_h^A e^{-rC_l^A(S')} \\ &= \left(w_h^A - w_l^A\right)\left(e^{-rC_h^A(S')} - e^{-rC_l^A(S')}\right) \\ &\ge 0. \end{aligned} \tag{7}$$

Hence, $w_h^A(1 - e^{-rC_h^A(S')}) + w_l^A(1 - e^{-rC_l^A(S')}) \le w_l^A(1 - e^{-rC_l^A(S)}) + w_h^A(1 - e^{-rC_h^A(S)})$. Therefore, S' is not worse than S. Thus, repeating this swapping argument for all the A-jobs not sequenced in the WDSPT order yields the theorem. □

Based on the above analysis, our algorithm to solve the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ for the case where the A-jobs have reversely agreeable weights can be described as in Algorithm 3.

Input: $n_A, n_B, U, r, w^A = (w_1^A, w_2^A, \ldots, w_{n_A}^A)$, $p^A = (p_1^A, p_2^A, \ldots, p_{n_A}^A)$ and $p^B = (p_1^B, p_2^B, \ldots, p_{n_B}^B)$.

Step 1. Set $t = (t_0 - 1/k)\prod_{j=1}^{n_X}(1 - ka_j^X) + 1/k, l = n_A, h = n_B$, and $\sum w_j^A(1 - e^{-rC_j^A}) = 0$;
sort the A-jobs in non-decreasing order of $(1 - e^{-ra_j^A})/(w_j^A e^{-ra_j^A})$,
that is, $(1 - e^{-ra_{[1]}^A})/(w_{[1]}^A e^{-ra_{[1]}^A}) \le (1 - e^{-ra_{[2]}^A})/(w_{[2]}^A e^{-ra_{[2]}^A}) \le \cdots \le (1 - e^{-ra_{[n_A]}^A})/(w_{[n_A]}^A e^{-ra_{[n_A]}^A})$;
calculate the deadlines of the B-jobs from $f_j^B(D_j^B) = U$ and
renumber them according to the non-decreasing order, that is, $D_{[1]}^B \le D_{[2]}^B \le \cdots \le D_{[n_B]}^B$;

Step 2. If $h \ge 1$, then
  If $t \le D_{[h]}^B$, then
    set $h = h - 1$, $t = (t - a_{[h]}^B)/(1 - ka_{[h]}^B)$, assign $J_{[h]}^B$ at time t, and go to Step 2;
  Else
    go to Step 3;
Else
  go to Step 3;

Step 3. If $l \ge 1$, then
  set $l = l - 1$, $\sum w_j^A(1 - e^{-rC_j^A}) = \sum w_j^A(1 - e^{-rC_j^A}) + w_{[l]}^A(1 - e^{-rt})$, $t = (t - a_{[l]}^A)/(1 - ka_{[l]}^A)$,
  assign $J_{[l]}^A$ at time t, and go to Step 4;
Elseif $h \ge 1$
  output that the instance is not feasible;
Else
  go to Step 4;

Step 4. If $h \ge 1$ or $l \ge 1$, then
  go to Step 2.
Else
  stop.

Algorithm 3

Theorem 9. *The problem* $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ *can be solved in* $O(n_A \log n_A + n_B \log n_B)$ *time by applying Algorithm 3 for the case that the A-jobs have reversely agreeable weights.*

Proof. The proof is analogous to that of Theorem 7. □

6. Conclusions

This paper introduced a new scheduling model on a single machine that involves two agents and learning effects simultaneously. We studied the problem of finding an optimal schedule for agent A, subject to the constraint that the maximum cost of agent B does not exceed a given value. We derived the optimal structural properties of optimal schedules and provided polynomial time algorithms for the problem $1 \mid p_j^X = a_j^X(1 - kt) \mid f_{\max}^A : f_{\max}^B \le U$. We also showed that the problems $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j C_j^A : f_{\max}^B \le U$ and $1 \mid p_j^X = a_j^X(1 - kt) \mid \sum w_j^A(1 - e^{-rC_j^A}) : f_{\max}^B \le U$ can also be solved in polynomial time under certain agreeable conditions. Future research may consider the scheduling model with more than two agents.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors are grateful to the Editor and two anonymous referees for their helpful comments on an earlier version of their paper. This research was supported by the National Natural Science Foundation of China (no. 71301022).

References

[1] D. Biskup, "Single-machine scheduling with learning considerations," *European Journal of Operational Research*, vol. 115, no. 1, pp. 173–178, 1999.

[2] T. C. E. Cheng and G. Wang, "Single machine scheduling with learning effect considerations," *Annals of Operations Research*, vol. 98, no. 1–4, pp. 273–290, 2000.

[3] G. Mosheiov, "Scheduling problems with a learning effect," *European Journal of Operational Research*, vol. 132, no. 3, pp. 687–693, 2001.

[4] G. Mosheiov and J. B. Sidney, "Scheduling with general job-dependent learning curves," *European Journal of Operational Research*, vol. 147, no. 3, pp. 665–670, 2003.

[5] A. Bachman and A. Janiak, "Scheduling jobs with position-dependentprocessing times," *Journal of the Operational Research Society*, vol. 55, no. 3, pp. 257–264, 2004.

[6] A. Janiak and R. Rudek, "A new approach to the learning effect: beyond the learning curve restrictions," *Computers and Operations Research*, vol. 35, no. 11, pp. 3727–3736, 2008.

[7] J.-B. Wang, "Single-machine scheduling with general learning functions," *Computers and Mathematics with Applications*, vol. 56, no. 8, pp. 1941–1947, 2008.

[8] Y. Yin, D. Xu, K. Sun, and H. Li, "Some scheduling problems with general position-dependent and time-dependent learning effects," *Information Sciences*, vol. 179, no. 14, pp. 2416–2425, 2009.

[9] D. Biskup, "A state-of-the-art review on scheduling with learning effects," *European Journal of Operational Research*, vol. 188, no. 2, pp. 315–329, 2008.

[10] Z. Jiang, F. Chen, and H. Kang, "Single-machine scheduling problems with actual time-dependent and job-dependent learning effect," *European Journal of Operational Research*, vol. 227, no. 1, pp. 76–80, 2013.

[11] Z. Jiang, F. Chen, and C. Wu, "Minimizing the maximum lateness in a single-machine scheduling problem with the normal time-dependent and job-dependent learning effect," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9438–9441, 2012.

[12] S.-J. Yang, "Single-machine scheduling problems with both start-time dependent learning and position dependent aging effects under deteriorating maintenance consideration," *Applied Mathematics and Computation*, vol. 217, no. 7, pp. 3321–3329, 2010.

[13] S.-J. Yang and D.-L. Yang, "Minimizing the makespan on single-machine scheduling with aging effect and variable maintenance activities," *Omega*, vol. 38, no. 6, pp. 528–533, 2010.

[14] J.-B. Wang, L. Sun, and L. Sun, "Single machine scheduling with exponential sum-of-logarithm-processing-times based learning effect," *Applied Mathematical Modelling*, vol. 34, no. 10, pp. 2813–2819, 2010.

[15] C.-C. Wu, Y. Yin, and S.-R. Cheng, "Single-machine and two-machine flowshop scheduling problems with truncated position-based learning functions," *Journal of the Operational Research Society*, vol. 64, no. 1, pp. 147–156, 2013.

[16] D. Xu, Y. Yin, and H. Li, "Comments on "a note on minimizing maximum lateness in an m-machine scheduling problem with a learning effect"," *Applied Mathematics and Computation*, vol. 217, no. 2, pp. 939–943, 2010.

[17] P. Liu, X. Zhou, and L. Tang, "Two-agent single-machine scheduling with position-dependent processing times," *International Journal of Advanced Manufacturing Technology*, vol. 48, no. 1–4, pp. 325–331, 2010.

[18] K. I.-J. Ho, J. Y.-T. Leung, and W.-D. Wei, "Complexity of scheduling tasks with time-dependent execution times," *Information Processing Letters*, vol. 48, no. 6, pp. 315–320, 1993.

[19] Y. Yin, S.-R. Cheng, and C.-C. Wu, "Scheduling problems with two agents and a linear non-increasing deterioration to minimize earliness penalties," *Information Sciences*, vol. 189, pp. 282–292, 2012.

[20] J.-B. Wang and Z.-Q. Xia, "Scheduling jobs under decreasing linear deterioration," *Information Processing Letters*, vol. 94, no. 2, pp. 63–69, 2005.

[21] J.-B. Wang, "A note on single-machine scheduling with decreasing time-dependent job processing times," *Applied Mathematical Modelling*, vol. 34, no. 2, pp. 294–300, 2010.

[22] K. R. Baker and J. C. Smith, "A multiple-criterion model for machine scheduling," *Journal of Scheduling*, vol. 6, no. 1, pp. 7–16, 2003.

[23] A. Agnetis, P. B. Mirchandani, D. Pacciarelli, and A. Pacifici, "Scheduling problems with two competing agents," *Operations Research*, vol. 52, no. 2, pp. 229–242, 2004.

[24] A. Agnetis, D. Pacciarelli, and A. Pacifici, "Multi-agent single machine scheduling," *Annals of Operations Research*, vol. 150, no. 1, pp. 3–15, 2007.

[25] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine to minimize total weighted number of tardy jobs," *Theoretical Computer Science*, vol. 362, no. 1–3, pp. 273–281, 2006.

[26] T. C. E. Cheng, C. T. Ng, and J. J. Yuan, "Multi-agent scheduling on a single machine with max-form criteria," *European Journal of Operational Research*, vol. 188, no. 2, pp. 603–609, 2008.

[27] C. T. Ng, T. C. E. Cheng, and J. J. Yuan, "A note on the complexity of the problem of two-agent scheduling on a single machine," *Journal of Combinatorial Optimization*, vol. 12, no. 4, pp. 386–393, 2006.

[28] B. Mor and G. Mosheiov, "Scheduling problems with two competing agents to minimize minmax and minsum earliness measures," *European Journal of Operational Research*, vol. 206, no. 3, pp. 540–546, 2010.

[29] K. Lee, B.-C. Choi, J. Y.-T. Leung, and M. L. Pinedo, "Approximation algorithms for multi-agent scheduling to minimize total weighted completion time," *Information Processing Letters*, vol. 109, no. 16, pp. 913–917, 2009.

[30] J. Y.-T. Leung, M. Pinedo, and G. Wan, "Competitive two-agent scheduling and its applications," *Operations Research*, vol. 58, no. 2, pp. 458–469, 2010.

[31] G. Wan, S. R. Vakati, J. Y.-T. Leung, and M. Pinedo, "Scheduling two agents with controllable processing times," *European Journal of Operational Research*, vol. 205, no. 3, pp. 528–539, 2010.

[32] Y. Yin, S.-R. Cheng, T. C. E. Cheng, W.-H. Wu, and C.-C. Wu, "Two-agent single-machine scheduling with release times and deadlines," *International Journal of Shipping and Transport Logistics*, vol. 5, no. 1, pp. 75–94, 2013.

[33] X. Yu, Y. Zhang, D. Xu, and Y. Yin, "Single machine scheduling problem with two synergetic agents and piece-rate maintenance," *Applied Mathematical Modelling*, vol. 37, no. 3, pp. 1390–1399, 2013.

[34] K. Zhao and X. Lu, "Approximation schemes for two-agent scheduling on parallel machines," *Theoretical Computer Science*, vol. 468, pp. 114–121, 2013.

[35] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, "Optimization and heuristic in deterministic sequencing and scheduling: a survey," *Annals of Discrete Mathematics*, vol. 5, pp. 287–326, 1979.