Discrete Dynamics in Nature and Society

Scheduling Problems with Due Date Assignment

Guest Editors: Yunqiang Yin, Shuenn-Ren Cheng, John Y. Chiang, Jason C. H. Chen, Xuerong Mao, and Chin-Chia Wu

Scheduling Problems with Due Date Assignment

Discrete Dynamics in Nature and Society

Scheduling Problems with Due Date Assignment

Guest Editors: Yunqiang Yin, Shuenn-Ren Cheng, John Y. Chiang, Jason C. H. Chen, Xuerong Mao, and Chin-Chia Wu

Editorial Board

Contents

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 683269, 2 pages
http://dx.doi.org/10.1155/2015/683269

Editorial

Scheduling Problems with Due Date Assignment

Yunqiang Yin,[1] Shuenn-Ren Cheng,[2] John Y. Chiang,[3] Jason C. H. Chen,[4] Xuerong Mao,[5] and Chin-Chia Wu[6]

[1] *Faculty of Management and Economics, Kunming University of Science and Technology, Kunming 650093, China*
[2] *Graduate Institute of Business Administration, Cheng Shiu University, Kaohsiung, Taiwan*
[3] *Department of Computer Science & Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan*
[4] *School of Business, Gonzaga University, E 502 Boone Avenue, Spokane, WA 99258, USA*
[5] *Department of Mathematics and Statistics, University of Strathclyde, UK*
[6] *Department of Statistics, Feng Chia University, Taichung, Taiwan*

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

Received 18 March 2015; Accepted 18 March 2015

Scheduling problems with due date assignment have attracted much research attention in the past few decades due to the introduction of new concepts and practices of operations management. Meanwhile, there are a lot of works by building up a discrete mathematical approach restricted by the absence of general principles that could play the same role as the vibrational principles in operation research or application statistics. The issue on scheduling problems with due date assignment inspires many researchers to address many interesting manuscripts most of which are discussed widely over single-machine, flowshop, or parallel settings.

One manuscript addresses the topic on common due date assignment over a parallel-batching machine. H. Gong et al. consider the parallel-batching scheduling problem with delivery involving both batching scheduling and common due date assignment. They prove that the problem is NP-hard in the strong sense and build dynamic programming algorithms for two special cases with a given sequence of orders on the machine and a given batching in the production part, respectively. Three manuscripts concentrate on two-agent scheduling on single-machine setting. L. Wan investigates a two-agent scheduling problem on a single machine to minimize the maximum cost with position-dependent jobs. In the problem setting, the actual position-dependent processing time of one job is characterized by variable function dependent on the position of the job in the sequence. The author develops a feasible method to achieve all the Pareto optimal points in polynomial time. P. Liu and L. Duan consider a scheduling problem with resource dependent release times and two agents. Their objective is to find a schedule for the problem of minimizing A-agent's total amount of resource consumption with a constraint on B-agent's makespan. They propose a polynomial time algorithm to solve the problem. S. Liu et al. consider a two-agent scheduling on a single machine to minimize the A-agent's total completion time with the restriction that has an upper bound on B-agent's total completion time. They use a branch-and-bound and three simulated annealing algorithms to solve the proposed problem. Five manuscripts focus on new developments in scheduling approach in operation research. Q. Liu et al. consider the online scheduling problem on a single machine with the assumption that all jobs have their processing times in specified interval and all the jobs should be first processed on a single machine and then delivered by a vehicle to some customer. Public transit providers are often facing continuous pressure to improve service quality and reduce operating costs. Inspired by this observation, H. Li et al. address a bus driver scheduling problem. Sometime real-life large and complex problem instances often need new approaches to overcome the computational difficulties in solving them, and thus they propose a column generation based hyperheuristic for finding near-optimal solutions. Ship block construction space is an important bottleneck resource in the process of shipbuilding, so the production scheduling optimization is a key technology to improve the efficiency of shipbuilding. S. Hu et al. propose a hybrid heuristic algorithm for the ship block construction space scheduling problem. According to different driving behavioral characteristics of bus drivers, Z. Zhang et al. propose a cellular automata traffic model considering the bus lane-changing behavior with scheduling parameters. For a better service level of a train operating plan, W. Zhou et al. address an integrated optimization method of train planning and train scheduling.

Four potential topics in operation research or application statistics include to explore the key influencing factors of low-carbon development for logistics companies are investigated by using the data from the questionnaire by X. Y. Duan et al., using multiple-decision procedures to test the homogeneity of mean for k exponential distributions by H.-C. Chen et al., discussing a unified weight formula to find the sample variance from weighted successive difference by K.-H. Lo et al., and composite transcendental entire functions with certain gaps which have no unbounded Fatou component by C. Yang and S. Wang.

Finally, we hope these papers may enrich and provide a guide to the readers to treat scheduling with due date assignment or further developments in operation research or application statistics.

Yunqiang Yin
Shuenn-Ren Cheng
John Y. Chiang
Jason C. H. Chen
Xuerong Mao
Chin-Chia Wu

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 932680, 4 pages
http://dx.doi.org/10.1155/2015/932680

Research Article

Two-Agent Scheduling to Minimize the Maximum Cost with Position-Dependent Jobs

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 9 September 2014; Accepted 5 November 2014

Academic Editor: Chin-Chia Wu

This paper investigates a single-machine two-agent scheduling problem to minimize the maximum costs with position-dependent jobs. There are two agents, each with a set of independent jobs, competing to perform their jobs on a common machine. In our scheduling setting, the actual position-dependent processing time of one job is characterized by variable function dependent on the position of the job in the sequence. Each agent wants to fulfil the objective of minimizing the maximum cost of its own jobs. We develop a feasible method to achieve all the Pareto optimal points in polynomial time.

1. Introduction

Scheduling theory is very useful in industrial applications and provides much guidance in the real world. The traditional scheduling problems have only one agent to optimize its own objective. With the development of modern technology, more and more scheduling models occur. Once entering the 21st century, the existing one-agent scheduling models can not meet with the requirement of the time. Particularly, in the big data period, the internet must handle various instructions of online net citizens and meet with their different preferences. Under such environments, multiagent scheduling models appear. The ultimate theoretical results of multiagent scheduling problems are to get all the Pareto optimal points. Such a problem is usually called as *Pareto optimization problem* (PP for short). Since Pareto optimization problems are very hard to deal with large number of scheduling models, the researchers consider somewhat weaker scheduling problems like single-objective scheduling problems, that is, to optimize the objective of one agent with the restrictions of the objectives of the other agents. Such a problem is called *constrained optimization problem* (CP for short). And, in reality, the resolution of CP problems will be more helpful for the resolution of PP problems. There are many papers in which the ultimate resolution of PP problems depends on the research results of CP problems; [1] is one of the most outstanding and instructive papers.

Of multiagent problems, the two-agent problems are obviously most simple and most representative. Agnetis et al. [1] first defined the two problems for two-agent scheduling models and opened the door of studying two-agent scheduling problems. Baker and Smith [2] dealt with two-agent scheduling problems by combining the two objectives of both agents into one single objective. And the method can also obtain all the Pareto optimal points in some certain condition. Yuan et al. [3] pointed out the faults of some dynamic programming designed in [2] and developed correct methods. Biskup [4] first introduced position-dependent processing times to classic scheduling models and developed optimal algorithms for the considered problems. Further, Mosheiov [5] extended it to more scheduling problems, including single-machine problems, multicriteria problems, and parallel-machine problems. Bachman and Janiak considered some special position-dependent scheduling problem and proved that it is NP-hard in the strong sense. But with regret, the proof is not complete with lack of key statement of some places. Janiak and Kovalyov [6] supplemented the key statement and provided a new complete proof. Wang and Xia [7] and Wang [8] considered position-dependent processing times in some flow shop scheduling problems. Yin et al. [9] generalized the position-dependent functions of processing times and some theoretical results appearing in [4, 5]. Yang [10] simultaneously considered the jobs deterioration and the position-dependent learning effect under deteriorating maintenance environment and analyzed some results of single-machine scheduling models. Biskup [11] gave an excellent retrospect of scheduling theory with learning effects. Liu et al. [12] initially introduced two-agent models to position-dependent scheduling problems. They studied a single-machine two-agent CP scheduling problem in which one agent wants to minimize the total completion time subject to the maximum cost of the other agent bounded by a constant, where jobs' actual processing times are defined by linear functions dependent on their positions in the schedule. Furthermore, Wan [13] deepened the results and showed that the PP version of [12] is polynomially solvable. For more related papers, we refer to Yin et al. [14–16].

In the paper, we study the two-agent PP scheduling model introduced in [12, 13]. The objectives of the two agents are that both want to minimize the maximum cost of jobs. We develop a polynomial time algorithm to show that the PP problem can be efficiently solved. Section 2 provides some notations and problem description. A polynomial time algorithm is developed to show that the PP problem is solvable in Section 3.

2. Notations and Problem Statement

There are two agents, agent A and agent B, competing to schedule their respective jobs on a common single machine. The job sets of agent A and B are $\mathcal{J}^A = \{J_1^A, J_2^A, \dots, J_{n_A}^A\}$ and $\mathcal{J}^B = \{J_1^B, J_2^B, \dots, J_{n_B}^B\}$, respectively, where n_A and n_B are the numbers of the jobs of agents A and B, respectively. The normal processing time of J_j^X is denoted by p_j^X, $X = A, B$, $j = 1, 2, \dots, n_X$. Each job J_j^X has a nondecreasing and nonnegative cost function $f_j^X(\cdot)$ defined on the completion time. Let $\mathcal{J} = \mathcal{J}^A \cup \mathcal{J}^B$ and $n = n_A + n_B$. For a feasible schedule of $\mathcal{J}$, the following three conditions follow.

(i) The first job is processed at time zero.

(ii) The processing of the jobs of $\mathcal{J}$ does not overlap.

(iii) The machine processes the jobs of $\mathcal{J}$ without idle time.

From the conditions, any schedule is consistent with a sequence of $\mathcal{J}$. Generally, we let σ denote a sequence of $\mathcal{J}$, as well as the consistent schedule. Let p_{hr}^X denote the actual processing time of job J_h^X at position r, $p_{hr}^X = p_h^X + br$, where b represents the variable ratio. We assume that $|b| < p_{\min}/n$ if b is negative lest the actual processing time of some job is negative in some feasible schedules, where $p_{\min} = \min\{p_j^X : J_j^X \in \mathcal{J}\}$. Given a sequence σ, the completion time of job J_j^X is denoted by $C_j^X(\sigma)$. The cost of job J_j^X is equal to $f_j^X(C_j^X(\sigma))$. Therefore, the cost of agent X is defined as the maximum cost of jobs; that is, $f_{\max}^X(\sigma) = \max\{f_1^X(C_1^X(\sigma)), f_2^X(C_2^X(\sigma)), \dots, f_{n_X}^X(C_{n_X}^X(\sigma))\}$, $X = A, B$. For any job set $\mathcal{J}_0 \subseteq \mathcal{J}$, we define the completion time of $\mathcal{J}_0$ denoted by $C_{\mathcal{J}_0}(\sigma)$ to be the completion time of its last job. If there is no confusion, C_j^X, $C_{\mathcal{J}_0}$, and $f_{\max}^X$ are short for $C_j^X(\sigma)$, $C_{\mathcal{J}_0}(\sigma)$, and $f_{\max}^X(\sigma)$, respectively. For any two feasible schedules σ and ρ, we call σ *Pareto better* than ρ if and only if $(f_{\max}^A(\sigma), f_{\max}^B(\sigma)) \le (f_{\max}^A(\rho), f_{\max}^B(\rho))$ and at least one inequality strictly holds. σ is called *Pareto optimal* if there is no schedule Pareto better than σ. And $(f_{\max}^A(\sigma), f_{\max}^B(\sigma))$ is a Pareto optimal point. The objective is to find all the Pareto optimal points. Adopting the notation of [1], the PP problem can be denoted by $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$. For any job set $\mathcal{J}_0 \subseteq \mathcal{J}$, let $P(\mathcal{J}_0) = \sum_{J_j^X \in \mathcal{J}_0} p_j^X + |\mathcal{J}_0|(|\mathcal{J}_0| + 1)b/2$. Furthermore, if $\mathcal{J}_0 \subseteq \mathcal{J}_1$, then $P(\mathcal{J}_0) \le P(\mathcal{J}_1)$.

Lemma 1 (see [13]). *Let σ be a feasible schedule of job set $\mathcal{H} \subseteq \mathcal{J}$; then $C_{\mathcal{H}}(\sigma) = P(\mathcal{H})$.*

From Lemma 1, we know that the completion time of $\mathcal{H}$ does not depend on the processing order of jobs of $\mathcal{H}$.

Let $y = (y^A, y^B)$ be a number pair; we consider problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \le y^A : f_{\max}^B \le y^B$. For the pair y, we define the resulting due date of J_j^X to be $D_j^X(y)$ with $f_j^X(D_j^X(y)) \le y^X$ and $f_j^X(D_j^X(y) + 1) > y^X$, $X = A, B$; $j = 1, 2, \dots, n_X$. Let EDD(y) be the schedule in which we sequence the jobs of $\mathcal{J}$ in nondecreasing order of the resulting due dates and try to schedule A-jobs after B-jobs for breaking the tie if it exists. Similar to the proof of EDD rule [17], we can get the following lemma.

Lemma 2. *If problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \le y^A : f_{\max}^B \le y^B$ is feasible, then schedule EDD(y) is a feasible schedule.*

3. The Main Results

In this section, we will describe our algorithm to show that the PP problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$ can be solved in $O(n_A^3 n_B + n_A n_B^3)$. First, we design an algorithm to solve the problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$ and the problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \le U : f_{\max}^B$ for any given bound U.

Algorithm $\mathcal{A}$. For problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$.

Step 0. Initialize $\overline{\mathcal{J}} := \mathcal{J}$, $\overline{n} := n_A + n_B$, $\overline{n}_A := n_A$, $\overline{n}_B := n_B$, $f_{\max}^A := -\infty$.

Step 1. Calculate the current completion time $T := P(\overline{\mathcal{J}})$.

Step 2. If $\overline{n}_A = 0$ and $\overline{n}_B = 0$, then Stop and Return the value $f_{\max}^A$.

Step 3. If $\overline{n}_A = 0$ and $\overline{n}_B > 0$, let $\mathcal{K}$ be the set of B-jobs of $\overline{\mathcal{J}}$ with the cost at most U. If $\mathcal{K} = \emptyset$, then Stop and Return infeasible. Otherwise arbitrarily select one B-job J_k^B of $\mathcal{K}$ and schedule job J_k^B at position $\overline{n}$, $\overline{n} := \overline{n} - 1$, $\overline{n}_B := \overline{n}_B - 1$, $\overline{\mathcal{J}} := \overline{\mathcal{J}} \setminus \{J_k^B\}$, go to Step 1.

Step 4. If $\overline{n}_A > 0$ and $\overline{n}_B = 0$, we select an A-job J_l^A out of $\overline{\mathcal{J}}$ so that $f_l^A(T) = \min_{J_i^A \in \overline{\mathcal{J}}} f_i^A(T)$ and schedule J_l^A at position $\overline{n}$, $\overline{n} := \overline{n} - 1$, $\overline{n}_A := \overline{n}_A - 1$, $f_{\max}^A := \{f_{\max}^A, f_l^A(T)\}$, $\overline{\mathcal{J}} := \overline{\mathcal{J}} \setminus \{J_l^A\}$, go to Step 1.

Step 5. If $\overline{n}_A > 0$ and $\overline{n}_B > 0$, let $\mathcal{K}$ be the set of B-jobs of $\overline{\mathcal{J}}$ with the cost at most U. If $\mathcal{K} = \emptyset$, then we select an A-job J_l^A out of $\overline{\mathcal{J}}$ so that $f_l^A(T) = \min_{J_i^A \in \overline{\mathcal{J}}} f_i^A(T)$ and schedule J_l^A at position $\overline{n}$, $\overline{n} := \overline{n} - 1$, $\overline{n}_A := \overline{n}_A - 1$, $f_{\max}^A := \{f_{\max}^A, f_l^A(T)\}$, $\overline{\mathcal{J}} := \overline{\mathcal{J}} \setminus \{J_l^A\}$, go to Step 1. Otherwise arbitrarily select one B-job J_k^B of $\mathcal{K}$ and schedule job J_k^B at position $\overline{n}$, $\overline{n} := \overline{n} - 1$, $\overline{n}_B := \overline{n}_B - 1$, $\overline{\mathcal{J}} := \overline{\mathcal{J}} \setminus \{J_k^B\}$, go to Step 1.

We can apply Algorithm $\mathcal{A}$ to problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \le U : f_{\max}^B$ only by exchanging the identities of agent A and agent B.

Theorem 3. *Algorithm $\mathcal{A}$ solves the problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$ correctly.*

Proof. Suppose σ is the resulted sequence by applying Algorithm $\mathcal{A}$ to problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$. Let ρ be an optimal sequence of problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$; we only need to prove that $f_{\max}^A(\sigma) \le f_{\max}^A(\rho)$. Let $J_{i_0}^A$ be such an A-job that $f_{i_0}^A(\sigma) = f_{\max}^A(\sigma)$; we denote by $\mathcal{K}$ the set of jobs prior to job $J_{i_0}^A$ in σ including job $J_{i_0}^A$. Let $J_{\mathcal{K}}$ be the last job of $\mathcal{K}$ scheduled in ρ; we denote the set of jobs prior to job $J_{\mathcal{K}}$ in ρ including job $J_{\mathcal{K}}$ by $\mathcal{L}$. $\mathcal{K} \subseteq \mathcal{L}$ and $P(\mathcal{K}) \le P(\mathcal{L})$. According to Algorithm $\mathcal{A}$, for each B-job $J_j^B \in \mathcal{K}$, we have $f_j^B(P(\mathcal{K})) > U$. Since $P(\mathcal{K}) \le P(\mathcal{L})$, $f_j^B(P(\mathcal{L})) \ge f_j^B(P(\mathcal{K})) > U$. Furthermore, $J_{\mathcal{K}}$ must be an A-job $J_{j_0}^A$. By the algorithm, we know that $f_{i_0}^A(P(\mathcal{K})) = \min_{J_i^A \in \mathcal{K}}\{f_i^A(P(\mathcal{K}))\}$. So

$$\begin{aligned} f_{\max}^A(\rho) &\ge f_{j_0}^A\left(C_{j_0}^A(\rho)\right) = f_{j_0}^A(P(\mathcal{L})) \ge f_{j_0}^A(P(\mathcal{K})) \\ &\ge f_{i_0}^A(P(\mathcal{K})) = f_{\max}^A(\sigma). \end{aligned} \tag{1}$$

We complete the proof. □

In the following, we state a very popular method to get a Pareto optimal point of problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$. The method is first developed in [18] and then applied in [1, 19, 20].

Algorithm $\mathcal{B}(U)$. For problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$.

Step 0. Call Algorithm $\mathcal{A}$ to $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A : f_{\max}^B \le U$. If algorithm $\mathcal{A}$ returns infeasible, then let Algorithm $\mathcal{B}(U)$ stop and returns infeasible. Otherwise let x be the returned value $f_{\max}^A$. Call Algorithm $\mathcal{A}$ to $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \le x : f_{\max}^B$, let y be the returned value, then we stop the algorithm and return the pair (x, y).

Theorem 4. *Algorithm $\mathcal{B}(U)$ returns a Pareto optimal point (x, y) with $y \le U$. Furthermore, if $(\overline{x}, \overline{y})$ is also a Pareto optimal point with $\overline{y} \le U$, then $\overline{x} \ge x$ and $\overline{y} \le y$.*

For each schedule σ, we define the indicator function $\Delta^i{}_j(\sigma)$ as below:

$$\Delta^i{}_j(\sigma) = \begin{cases} 1, & \text{if } J_i^A \prec_\sigma J_j^B, \\ 0, & \text{if } J_j^B \prec_\sigma J_i^A. \end{cases} \tag{2}$$

Here $J_i^A \prec_\sigma J_j^B$ means that J_i^A is prior to J_j^B in σ and vice versa. Furthermore, we set $\Delta(\sigma) = \sum_{1 \le i \le n_A, 1 \le j \le n_B} \Delta^i{}_j(\sigma)$.

Lemma 5. *For any two Pareto optimal points $y = (y_1, y_2)$ and $z = (z_1, z_2)$, $\Delta(EDD(y)) < \Delta(EDD(z))$ only if $y_1 > z_1$ and $y_2 < z_2$.*

Proof. Since the cost function $f_j^X(\cdot)$ of job J_j^X is nondecreasing, $X = A, B$; $j = 1, 2, \ldots, n_X$, we have $D_i^A(y) \ge D_i^A(z)$ for any A-job J_i^A, $i = 1, 2, \ldots, n_A$ and $D_j^A(y) \le D_j^A(z)$ for any B-job J_j^B, $j = 1, 2, \ldots, n_B$. Therefore,

$$\begin{gathered} \Delta^i{}_j(\mathrm{EDD}(y)) \le \Delta^i{}_j(\mathrm{EDD}(z)), \\ i = 1, 2, \ldots, n_A, \quad j = 1, 2, \ldots, n_B. \end{gathered} \tag{3}$$

Note that $y = (y_1, y_2)$ and $z = (z_1, z_2)$ are two Pareto optimal points; we have $f_{\max}^A(\mathrm{EDD}(y)) = y_1, f_{\max}^B(\mathrm{EDD}(y)) = y_2$ and $f_{\max}^A(\mathrm{EDD}(z)) = z_1$, $f_{\max}^B(\mathrm{EDD}(z)) = z_2$. Let $J_{j_0}^B$ be such a B-job that $f_{j_0}^B(C_{j_0}^B(\mathrm{EDD}(z))) = z_2$ which means that $C_{j_0}^B(\mathrm{EDD}(z)) = D_{j_0}^B(z)$; we denote the job set of jobs prior to $J_{j_0}^B$ in $\mathrm{EDD}(z)$ including job $J_{j_0}^B$ by $\mathcal{K}$:

$$D_j^B(z) \le D_{j_0}^B(z), \quad J_j^B \in \mathcal{K}. \tag{4}$$

Let $J_{\mathcal{K}}$ be the last job of $\mathcal{K}$ scheduled in $\mathrm{EDD}(y)$, and we denote the set of jobs prior to job $J_{\mathcal{K}}$ in $\mathrm{EDD}(y)$ including job $J_{\mathcal{K}}$ by $\mathcal{L}$: $\mathcal{K} \subseteq \mathcal{L}$ and $P(\mathcal{K}) \le P(\mathcal{L})$. By (4),

$$\begin{gathered} f_j^B(P(\mathcal{L})) \ge f_j^B(P(\mathcal{K})) = f_j^B\left(D_{j_0}^B(z)\right) \ge z_2 > y_2, \\ J_j^B \in \mathcal{K}. \end{gathered} \tag{5}$$

It means that $J_{\mathcal{K}}$ is not a B-job but an A-job. And we denote the A-job by $J_{i_0}^A$; then

$$\Delta^{i_0}_{j_0}(\mathrm{EDD}(y)) < \Delta^{i_0}_{j_0}(\mathrm{EDD}(z)). \tag{6}$$

Therefore,

$$\begin{aligned} \Delta(\mathrm{EDD}(y)) &= \sum_{1 \le i \le n_A, 1 \le j \le n_B} \Delta^i{}_j(\mathrm{EDD}(y)) \\ &< \sum_{1 \le i \le n_A, 1 \le j \le n_B} \Delta^i{}_j(\mathrm{EDD}(z)) = \Delta(\mathrm{EDD}(z)). \end{aligned} \tag{7}$$

We complete the proof. □

Algorithm $\mathscr{C}$. For problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$.

Step 0. Initialize $U := +\infty$, $k := 1$.

Step 1. Call Algorithm $\mathscr{B}(U)$ for problem $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$.

Step 2. If Algorithm $\mathscr{A}(U)$ returns infeasible, then we stop and output the Pareto optimal points $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$. Otherwise, let (x, y) be the resulted schedule from Algorithm $\mathscr{A}(U)$; set $(x_k, y_k) := (x, y)$, $U := y_k - 1$, $k := k + 1$. And go to Step 1.

Theorem 6. *Algorithm* $\mathscr{C}$ *returns all the Pareto optimal points of problem* $1|p_{hr}^A = p_h^A + br, p_{kv}^B = p_k^B + bv|f_{\max}^A \circ f_{\max}^B$ *in* $O(n_A^3 n_B + n_A n_B^3)$.

Proof. It is obvious that Algorithm $\mathscr{C}$ returns all the Pareto optimal points. We then analyze the complexity of Algorithm $\mathscr{C}$. Note that Algorithm $\mathscr{A}$ can be implemented in $O(n_A^2 + n_B^2)$; Algorithm $\mathscr{B}(U)$ can also be implemented in $O(n_A^2 + n_B^2)$. For any feasible schedule τ, $0 \leq \Delta(\tau) \leq n_A n_B$. Hence, by Lemma 5, there are at most $n_A n_B + 1$ times of implementation of Algorithm $\mathscr{B}(U)$ during the running of Algorithm $\mathscr{C}$. Furthermore, we can get that Algorithm $\mathscr{C}$ can be completed in $O(n_A^3 n_B + n_A n_B^3)$. □

Conflict of Interests

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

Acknowledgments

This work was supported by the Natural Sciences Foundation (Grant no. 20142BAB211017) of Jiangxi Province and the School Subject (Grant no. 06162015) of Jiangxi University of Finance and Economics.

References

[1] 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.

[2] 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.

[3] J. J. Yuan, W. P. Shang, and Q. Feng, "A note on the scheduling with two families of jobs," *Journal of Scheduling*, vol. 8, no. 6, pp. 537–542, 2005.

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

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

[6] A. Janiak and M. Y. Kovalyov, "Scheduling jobs with position-dependent processing times," *Journal of the Operational Research Society*, vol. 63, no. 7, pp. 1018–1020, 2012.

[7] J.-B. Wang and Z.-Q. Xia, "Flow-shop scheduling with a learning effect," *Journal of the Operational Research Society*, vol. 56, no. 11, pp. 1325–1330, 2005.

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

[9] 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.

[10] 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.

[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] P. Liu, X. Zhou, and L. Tang, "Two-agent single-machine scheduling with position-dependent processing times," *The International Journal of Advanced Manufacturing Technology*, vol. 48, no. 1–4, pp. 325–331, 2010.

[13] L. Wan, "Two-agent pareto optimization problems with position-dependent jobs," In press.

[14] 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.

[15] 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.

[16] Y. 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.

[17] W. E. Smith, "Various optimizers for single-stage production," *Naval Research Logistics Quarterly*, vol. 3, pp. 59–66, 1956.

[18] J. A. Hoogeveen, "Single-machine scheduling to minimize a function of two or three maximum cost criteria," *Journal of Algorithms*, vol. 21, no. 2, pp. 415–433, 1996.

[19] F. Sourd, "Preemptive scheduling with two minimax criteria," *Annals of Operations Research*, vol. 107, pp. 303–319, 2001.

[20] L. Wan, L. J. Wei, J. L. Sheng, and J. J. Yuan, "Pareto optimization scheduling with two competing agents and linear non-increasing deterioration to minimize earliness penalties," submitted.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 851492, 9 pages
http://dx.doi.org/10.1155/2015/851492

Research Article

Recognition and Empirical Research on Key Influencing Factors of Low Carbon Development for Logistics Company

Xiang-Yun Duan,[1] Hsiu-Jung Chou,[2] and Shuenn-Ren Cheng[2]

[1] *Department of Modern Service, Zhejiang Shuren University, No. 8, Shuren Street, Gongshu District, Hangzhou, Zhejiang 310015, China*
[2] *Department of Business Administration, Cheng Shiu University, No. 840, Chengcing Road, Niaosong District, Kaohsiung City 83347, Taiwan*

Correspondence should be addressed to Hsiu-Jung Chou; hjc724@ms13.hinet.net

Received 1 July 2014; Accepted 15 September 2014

Academic Editor: Yunqiang Yin

The questionnaire paper used was work out by modifying the existing mature research scale and interviewing the experts and enterprises. The purpose of this study is to explore the key influencing factors of low carbon development for logistics companies by using the data from the questionnaire, which might contribute to further investigation of low carbon development pathway. Through correlation and stepwise regression analysis, differences between influencing factors and impact-degree were found since specific environmental behaviors of low carbon logistics might vary significantly. The driving factor of low carbon input comes from external points such as government and public external pressure, so the low carbon input is driven by an outside force. Yet, low carbon operation is significantly affected by internal factors such as low carbon behavior capacity, which is driven by an inside force.

1. Introduction

Although the low carbon economy has put forward ten years, the pollution of basic vital materials such as water, air, and food is still reported, indicating a big gap between the low carbon actual and target. The reduction targets and absolute cap of carbon were established by the international community, with low carbon logistics remaining as a major concern of large energy consumption industry. According to the report of World Economic Forum in 2009, so far the overall logistics industry still counted for 5.5 percent of global greenhouse gas emission. How to achieve low carbon development for logistics enterprise is not only a question of economics theory and method, but also an important policy decision, since the carbon pollution analysis and low carbon governing policies were made on the basis of making the low carbon logistics behavior, influencing factors of environmental behavior, and impact paths clear.

Scholars have already started the research of low carbon logistics from the industry and enterprise level. Industry research mainly focuses on transportation department of logistics and the low carbon growth pathways of transportation sector of the logistic industry that was studied from macroscopic aspect of modeling urban, infrastructure, and spatial determinants of mobility [1]. It was found that the transport sectors are incapable of action on standard pricing and have long relied on public facilities. In order to reduce emissions of transport sector, public facilities should be designed by controlling the mobility measures such as reducing the movement intensity, investing in public mode, and reengineering the production process to low mobility. The enterprise environmental management of transport department was also studied [2] and it was found that the environmental management performance is obviously different between large and small land transportation logistics enterprises, and small land transportation logistics enterprises need to improve environmental performance compared with large enterprises. These researches adopt similar research ideas, which mainly focus on the research sector and enterprise partial characteristics. According to the Carbon Disclosures Project Supply Chain Report in 2011, there are over 50 percent of average corporation's carbon emissions that typically result from its supply chain. It is well known that logistics is the important part of a supply chain, while the research on enterprises' environmental behavior and influencing factors on this part is yet on the exploration stage [3]. Based on the situations mentioned above, this paper began from the systematic literatures survey of related field and worked out the questionnaire by modified from well-established foreign scale and through the method of expert interview and enterprise interview and then studied characteristics of low carbon logistics behavior as well as the effect of various social and individual enterprises variables on low carbon logistics behavior by using the data from the questionnaire. The purpose of this study is to explore the key influencing factors of low carbon development for logistics companies, which might contribute to make further investigation of low carbon development pathway.

2. Literature Review and Hypotheses

From the viewpoint of utility, low carbon logistics is a kind of environmental behavior. Scholars have provided many classical research framework about environmental behavior and its influencing factors, which includes the theory of planned behavior (TPB), the value-belief-norm (VBN) theory, and ABC theory model. The theory of planned behavior (TPB) was built on the theory of reasoned action (TRA) by Ajzen, with perfect explanatory power to the decision-making process of the general behavior. TPB became the classic theory of social psychology and was applied to predict and study the behavior. According to the theory of planned behavior, behavioral intention is the direct factors to determine the behavior, which act successfully only under the circumstances of conditions that behavior ability, opportunity, and resources were controlled by an individual, while the behavioral attitude, subjective norms, and perceived behavior control are three major factors to determine the behavioral intention. The stronger behavior attitude, the more support from others, the bigger perceived behavior control, and the behavior intention become greater, and vice versa [4–6].

The value-belief-norm (VBN) theory established by Stern focused on the formation process of environmental action. On the basis of explaining the types and role of environment value, the functional relations between the subjective variables of values, beliefs, norms, and the positive environmental behavior were defined and the results indicate that the environment attitude was affected by environmental value system, environmental beliefs affecting environmental norms and then the environment behavior was formed. The continuous effects of environmental values, beliefs, and norms can explain the environmental behavior [7–9]. It was confirmed by many empirical researches since this theory has been issued and opened up a new perspective for the study of environmental behavior.

The scholars of Ajzent and Stern have focused on the impact of norms and beliefs on environmental behavior. Although the influence pathway is different, norms and beliefs are all the internal cognitive variables. Focusing on the role of external factors by Guagnano et al., the famous ABC theory model believes that the effect of attitude (A) and circumstances (C) on behavior (B) depends on the relative value rather than individual value. It was found that the environmental behaviors generally occur when the external factors are positive as a result of joint action of behavior attitudes and exterior condition; there is the strongest relationship between environmental behaviors and attitudes when the influence of external conditions are neutral, and exterior condition plays an adjustment role in environmental attitudes and behaviors [10].

Many scholars have studied the influential variables based on the above classical research framework. The direction of research includes the relationship among the environmental attitude, personality factors, situational factors, and the environmental behavior. Specifically, it was listed as follows.

2.1. The Relationship between Environmental Attitude and Environmental Behavior. The scholars of new institutionalism theory fully have recognized the factors of external specification, values, and tradition in the system environment and believed both of them bring convergence pressure to organization in the field. Kotzab et al. [11] have developed a scale to evaluate the environmental supply chain initiatives of the world's largest 100 retailing companies and found that environmental supply chain management can be characterized as green operations oriented; environmental sustainability initiatives of supply chain include eight categories and fundamental environmental attitude is the most important one. The environmental attitudes were defined as the general attention and beliefs on the ecological environment in the new environmental paradigm, and NEP scale as new environmental paradigm scale was put forward [12]. After that, the NEP scale was used widely as a tool to measure the environment. Many scholars have investigated the relationship between environmental attitudes and behaviors by using NEP scale and found that there is significant correlation between environmental attitudes and behavior, and the positive environmental attitude played a significant role in promoting the environmental behavior [13, 14]. As for environmental attitudes, scholars have consistently supported the positive effect of environmental responsibility and values on environment behavior. VBN theory clearly points out that the environmental value is an important antecedent of environmental behavior [8] and there are differences in the function routes of environmental responsibility. There are mainly two kinds of view of function route and the details are as follows: one's opinion tends to believe that environmental responsibility directly affects the environment behavior [9, 15]; another thinks that environmental responsibility indirectly affects the environment behavior through intermediary variable [16, 17].

Based on the above analysis, the first set of assumptions of this research was put forward:

H1: the environmental attitude of logistics enterprise remarkably influences the behavior of low carbon logistics;

H1a: the stronger the environmental value of logistics enterprise is, the more conducive it was to implement the low carbon logistics;

H1b: the stronger the environmental responsibility of logistics enterprise is, the more conducive it was to implement the low carbon logistics.

2.2. The Relationship between Personality Factors and Environmental Behavior. Scholars that studied the aspects of organization and management theory have focused on internal enterprise and the relationship between personality factors and environmental behavior. There are three research ideas: the first is the attitude-behavior route. Taking behavior as the guidance, researchers have detected that the environmental behavior willingness is the direct influential factor or high latent variables for environmental behavior through the empirical research on environmental behavior. The strong positive correlation between willingness of environmental behavior and ecological behavior was found through the investigation of 3000 sample of transport association in Switzerland, and its explanatory power reaches 75 percent [18]. Some studies have found that the higher the environmental awareness of enterprise members, especially of the leaders, the greater the possibility of using nice environmental behavior [19, 20]. The second is the resource-ability-behavior route. Based on the consideration of resources, it was found out that the complementary resources and capabilities of enterprises firstly affected the business ethical attitudes of manager and further impacted the enterprises' environmental strategy and related behavior [21]. The study was concerned with self-perception and found that perceived environmental uncertainty has a negative effect on corporate environmental behavior, by taking the small and medium enterprises in Taiwan as samples [22]. The third is psychology-behavior route. Some researcher paid attention to the psychological forming process of environmental behavior and discovered that company tends to make environmental behavior in the situation of deterrence and group psychology other than attitudes to better explain the environmental behavior [23].

According to the above analysis, the second set of hypotheses in this study was proposed:

H2: personality variables of logistics enterprise significantly affect the low carbon logistics behavior;

H2a: the stronger the environmental group psychology of logistics enterprise is, the more helpful it was to carry out low carbon logistics;

H2b: the richer the low carbon related knowledge of logistics enterprise is, the more helpful it was to carry out low carbon logistics;

H2c: the stronger the low carbon behavior willingness of logistics enterprise is, the more helpful it was to carry out low carbon logistics;

H2d: the stronger the self-efficacy of logistics enterprise is, the more helpful it was to carry out low carbon logistics;

H2e: the stronger the low carbon behavior capability of logistics enterprise, the more helpful it was to carry out low carbon logistics.

2.3. The Relationship between Situational Factors and Environmental Behavior. The responsible environmental behavior model was proposed by Hines et al., it has not only verified the significant impact of action technology, action strategy knowledge, and environmental knowledge on the environmental behavior, but also put forward that the situational factors such as economic condition, social pressure and whether or not have an opportunity to engage in environmental behavior was an important external cause of environment behavior [16]. Scholars have explored the influence of external environment pressure on the enterprise environment from different angles; it was found that financial and human resources play an important role in companies' implementation of reverse logistics, whereas tangible resources do not have much influence on the practice [24]; regulation system of environment and strong regulation by government can affect the environment behavior and force enterprises to make cost internalization of external conditions [25]; the pressure generated by social norms has significantly influenced on the environmental behavior [26]; logistics companies' willingness to adopt RFID technology is significantly influenced by the explicitness and accumulation of technology, organizational encouragement for innovation, quality of human resources, and governmental support [27]. And the technology progress and maturity play an effective role of resistance to the emission of pollutants [28].

Based on the above analysis, the third set of hypothesis in this study was raised:

H3: the situational factors have significantly impacted on low carbon logistics behavior;

H3a: the more the low carbon economic cost paid by logistics enterprises is, the more beneficial it was to promote low carbon logistics;

H3b: the greater the low carbon social constraints faced by logistics enterprises are, the more beneficial it was to promote low carbon logistics;

H3c: the more the guiding influence of low carbon policies and regulations is, the more beneficial it was to promote low carbon logistics;

H3d: the higher the low carbon technology maturity is, the more beneficial it was to promote low carbon logistics.

3. Design of Scale

The scale used in this study arises from the two aspects: one is modified from well-established scale from foreign countries and indigenized based on the actual situation of logistics enterprises in China, and the other is a developed measurement scale by defining and operating the research variables through the expert interviews and enterprise interview. Since there are mature scales to measure variables such as environmental values, environmental responsibility, self-efficacy, low carbon behavior intention, economic cost, social norms, policies and regulations, and technical maturity in foreign countries, this study uses them after local modification. While the variables with local characteristics such as group psychology and low carbon behavior capability have not been introduced into the study of enterprises' environment behavior in foreign countries, the scale was developed combined with the enterprise and expert interviews. Since there is no mature scale to study low carbon logistics behavior, this paper developed the measurement items from the operation angle and divided it into two aspects of low carbon input and low carbon operation by taking the suggestion of expert and enterprise as the reference.

To ensure that the questionnaire has good content validity, presurvey as to the initial questionnaire was conducted on a smaller scale. The validity and efficiency of initial questionnaire were tested by using the SPSS17.0 statistical software, and the variable scale in which alpha coefficient is lower than 0.5 was redesigned. The final questionnaire has been formed after deleting the inappropriate indexes scale that factor's alpha coefficient is less than 0.6 and correlation coefficient of the item total is less than 0.3.

The final questionnaire contains a total of three parts. The first part is the scale of low carbon logistics behavior with 10 items, which focuses on low carbon investment behavior and low carbon operation behavior of logistics enterprises. These two variables were confirmed by interviews with entrepreneur during the first period of investigation. The second part is the scale of influence factors of low carbon logistics with 41 items, including environmental attitude, personal characteristics of enterprises, and situation factor. There into, the environmental attitudes include two variables of environmental values and environmental responsibility, it was respectively consulted the scale of Dunlap et al., [29] and Gärling et al., [17] in the process of scaled designed, and modified by domestication. The personality factors include five variables of low carbon related knowledge, low carbon behavior willingness, environmental group psychology, self-efficacy, and low carbon behavior capability; it referred to the scaled of Frick et al. [30], Chan [31], Sherer et al. [32], and Tanner and Kast [33] when the scale was designed; the economic cost, social norms, policies and regulations, and technical maturity are four variables of situational factors, designed by using the scale of Stern [8, 9] for reference. The third part is the basic personal information with 8 items. The five-point Likert scale was used in both parts of one two in the questionnaire.

The formal questionnaires were handed out by telephone, email, and visits to enterprises to investigate 198 logistics enterprises in Zhejiang province. The type of enterprise covers the transport, storage, and integrated logistics enterprise and the research objects include executives and employees of logistics enterprise. In this study, 550 questionnaires were issued and 400 were retrieved during the period of investigation, 305 questionnaires were available in it after a screening process, and the rate of recovery of scale is 72.7 percent and valid by 55.45 percent.

4. Empirical Analysis

4.1. Scientific Test of Scale. The first thing is to test the construct validity. First of all, both KMO value and Bartlett's spherical of four variables needed testing; the four variables include low carbon logistics behavior, environment attitude of enterprise, enterprises' personality characteristics, and situational factors. The results of the test show that KMO values of four variables were 0.856, 0.821, 0.704, and 0.804, all above 0.7, and Bartlett's spherical values of the four variables have been verified by Bartlett's test. The total variance was 60.93%, 65.68%, 76.97%, and 72.20%; that is to say, the scale is suitable for factor analysis. In the second place, orthogonal rotation of the maximum variance was made on the four variables by using principal component analysis. According to the standard of characteristic roots greater than 1, the variables of both low carbon logistics behavior and enterprise environment attitude can be extracted 2 main factors, the variables of corporate personality characteristics can be extracted 5 main factors, and situational variables can be extracted 4 main factors. Last, but not least, it is necessary to compare factor loading. It is found that the loading of all measurements on their own respective factors is greater than 0.5; nevertheless, the loading on the other factors is less than 0.5. The results indicated that both convergent validity and discriminant validity of the scale are able to meet the requirements and construct validity is well.

In addition to construct validity testing, reliability testing was also needed. The main work of reliability test is to analyze Cronbach's alpha coefficient of the scale. Cronbach's alpha coefficients of four variable of low carbon logistics behavior, environment attitude of enterprise, enterprises' personality characteristics, and situational factors were 0.735, 0.749, 0.872 and 0.815, all of them were above 0.6, which show that the scale has good reliability.

4.2. The Descriptive Analysis of the Questionnaire. The descriptive analysis of the variables was shown in Table 1. From the description of the low carbon logistics behavior and impact factors, the mean of low carbon input and operation is 3.83 and 3.27, that is to say, the mean of low carbon logistics behavior below 4, which indicates that low carbon action level of logistics enterprises is still in the state of between half-done and mostly done. Since the score is low in general, the implementation of environment behavior of low carbon logistics is not satisfactory. Compared with the implementation of two kinds of environmental behavior, low carbon inputs are better than low carbon operations. Logistics enterprises pay more attention to low carbon input compared with low carbon operation, so the low carbon operation is relatively weak. The reason might be correlated with the degree of environmental responsibility, low carbon related knowledge, and low carbon behavior ability, while both of them get the lowest score in the influence factors.

The situational factors get the highest scores among the three influencing factors. The mean of variables of situational factors besides economic cost is above 4, which implies that the logistics enterprises recognize the effect of social norms, policies and regulations, and technology maturity on low

Table 1: The descriptive statistical analysis of variables.

Variable	Mean	Standard deviation	Variable	Mean	Standard deviation
Low carbon input	3.83	0.47	Self-efficacy	3.86	0.1
Low carbon operation	3.27	0.59	Low carbon capability	3.21	0.33
Environmental value	3.69	1.06	Economic cost	3.91	0.35
Environmental responsibility	3.41	0.15	Social norm	4.14	0.27
Group psychology	3.75	0.29	Policies and regulations	4.18	0.19
Low carbon knowledge	3.45	0.22	Technology maturity	4.23	0.16
Low carbon behavior intention	3.94	0.17			

Table 2: The correlation matrix between low carbon logistics behavior and influence factors.

	1	2	3	4	5	6	7	8	9	10	11	12	13
Low carbon input	1												
Low carbon operation	0.046	1											
Environmental value	0.351**	0.265**	1										
Environmental responsibility	0.288**	0.304**	0.323**	1									
Group psychology	0.300**	0.424**	0.06	0.041	1								
Low carbon knowledge	0.237**	0.167**	0.044	−0.044	0.009	1							
Low carbon behavior intention	0.170**	0.533**	0.213**	0.654**	0.386**	0.306**	1						
Self-efficacy	0.170**	0.563**	0.076	0.048	0.349**	0.108	0.235**	1					
Low carbon capability	0.173**	0.588**	0.079	0.249**	0.139*	0.427**	0.392**	0.460**	1				
Economic cost	0.103*	0.165**	0.125*	−0.057	0.387**	−0.087	0.205**	0.307**	0.134*	1			
Social norm	0.388**	0.448**	0.372**	0.254**	−0.002	−0.078	0.411**	0.335**	0.253**	0.179**	1		
Policies and regulations	0.465**	0.374**	0.279**	0.311**	0.405**	0.318**	0.593**	0.11	0.347**	0.353**	0.379**	1	1
Technology maturity	0.252**	0.235**	0.154**	0.009	0.592**	−0.195**	0.208**	0.351**	0.032	0.539**	0.112	0.389**	1

Double star means that the significant correlation exits at 0.01 significant levels and single star exists at 0.05 significant levels.

carbon environmental action, and the external situational factors have significant influence on the behavior of low carbon logistics. The lowest score of mean among the personality factors is low carbon behavior ability with the score of 3.21, showing that the ability of logistics can deal with the low carbon business half-done. Low carbon knowledge is related with the variable of low carbon capability, and the mean of low carbon knowledge is 3.45, which shows that enterprises generally know something about low carbon and should strengthen knowledge learning to enhance the ability of low carbon behavior. The low carbon behavior intention gets the highest mean score of 3.94, indicating that the logistics enterprises are willing to implement low carbon development. Although logistics enterprises have strong willingness to implement the low carbon behavior under the background of low carbon economy, the absence of low carbon behavior ability and low carbon related knowledge might influence the low carbon development in varying degree. The score of mean was followed by self-efficacy at 3.86 and group psychology 3.75, reflecting that the self-cognitive ability to engage in low carbon behavior of logistics enterprises is still at a moderate level that the low carbon logistics was more or less influenced by related parties and was characteristic of conformity tendency. In the environmental factors, the mean of environmental values is 3.69 and environmental responsibility is 3.41, showing that logistics enterprises have basic environmental value but have not been updated to positive environmental value of altruism and ecological type; that is to say, although there is basic environmental awareness, it still makes further improvement on the environmental responsibility.

4.3. The Correlation Analysis of Low Carbon Logistics Behavior and Influence Factors. From the correlation relationship between low carbon logistics behavior and influencing factors (shown in Table 2), 11 influencing factors of three types of variables are correlated with low carbon logistics behavior. Besides the economic costs, other variables were positively correlated with the behavior of low carbon investment and low carbon operation at 0.01 significant levels. Furthermore, the variables of economic costs have correlation with low carbon input at 0.05 significant levels.

From the correlation degree, the variables of policies and regulations have the highest correlation with low carbon input behavior, followed by social norms, environmental values, and group psychology. The first two variables are situational factors and yet there is relatively low correlation relationship between personality factors and low carbon input. It demonstrates that the low carbon input behavior of logistics enterprises is the behavior propelled by outside force,

Table 3: The regression analysis within low-carbon investment and operation as the dependent variable.

Variable	The dependent variable					
	Low carbon input behavior			Low carbon operation behavior		
The independent variable	Standardized regression coefficient of β	t value	Sig.	Standardized regression coefficient of β	t value	Sig.
(Constant)		−0.752	0.006		−4.745	0
Environmental value	0.16	4.204	0.000	0.088	2.393	0.017
Environmental responsibility	0.274	4.522	0.000			
Group psychology	0.344	6.352	0.000	0.331	7.287	0.000
Low carbon knowledge	0.278	4.733	0.000			
Low carbon behavior intention	0.59	8.85	0.000	0.331	7.863	0.000
Self-efficacy	0.185	4.008	0.000	0.158	3.684	0.000
Low carbon capability	0.096	1.993	0.047	0.333	8.09	0.000
Economic cost	0.197	4.602	0.000	0.121	3.055	0.002
Social norm				0.263	6.141	0.000
Policies and regulations	0.71	13.52	0.000	0.166	3.659	0.000
Technology maturity	0.119	2.063	0.040			

and external pressure from the government and the public is the driving factors to promote low carbon input behavior. External guidance and pressure are the main power to promote low carbon input of logistics enterprises, which further verified the conclusion that the variable mean reflected. There is the highest correlation relationship between low carbon operation behavior and low carbon behavior capability, followed by self-efficacy, low carbon behavior intention, social norms, and group psychology. Not only four variables of them belong to personality factors, but also the first three variables are personality factors, indicating that the low carbon operation of logistics enterprises is mainly driven by inside force and enterprise itself becomes the driving factor to promote the low carbon operation behavior. Different from the one-off behavior of low carbon input, low carbon operation is a continuous process, whose operation effect depends on the function of each enterprise, and it is obvious that the variables of enterprises individual characteristic significantly influences the low carbon operation. Otherwise, the variables with striking correlation with two categories of environmental behavior are social norms and group psychology, which obviously reflects the cultural characteristics of enterprises in china.

4.4. The Regression Analysis. Through the methods of multiple stepwise regressions, the most influential and explanatory variables of two kinds of low carbon logistics behavior were selected according to the standard coefficient values of β. In the significant test of the regression analysis, the greater the t value is, the more meaningful the corresponding variables indicate, and the corresponding variable should be deleted when the t value is close to zero. At less than 0.05 significant level, the main influential variables of low carbon investment behavior are policies and regulations, low carbon behavior intention, group psychology, low carbon related knowledge, environmental responsibility, economic cost, self-efficacy, environmental values, technical maturity, and low carbon behavior ability. The sorting order shows that the primary influential factors of low carbon investment behavior are the policies and regulations of external factors, whose standard coefficient value of β is 0.71 which is far more than other variables, followed by individual features factors of enterprise. Combined with the results of correlation analysis, it was found that the low carbon investment behavior of logistics company is mainly affected by the external environment factors, and logistics enterprises will implement low carbon investment behavior when the external factors were pushed, low carbon investment intention was internalized, and the resource of software and hardware was matched, while the main influential variables of low carbon operation behavior are low carbon behavior ability, low carbon behavior intention, group psychology, social norms, policies and regulations, self-efficacy, and economic cost and environmental values. The first five variables are personality and situation factors of enterprise, and among them the first three variables are corporate personality factors which further confirm the results of correlation analysis. The implementation of low carbon operation behavior was ensured due to the internal drive of enterprise. Meanwhile, the greatest influential variables are enterprise individual characteristics, followed by situational factors. The internal resources, low carbon behavior willingness, and external force were needed to promote the low carbon operation behavior. The data was shown in Table 3.

Summing up the above analysis, the hypotheses of H1, H2, and H3 are partly established. The empirical results supported the hypothesis H1a of H1, which shows that the assumption that the stronger the environmental value of logistics enterprise is, the more conducive it was to implement the low carbon logistics is valid. The hypothesis H1b was only supported for low carbon input behavior but not low carbon operation behavior; that is to say, the assumption that the stronger the environmental responsibility of logistics enterprise, the more conducive it was to implement the low carbon logistics is invalid. The empirical results also supported the hypotheses H2a, H2c, H2d, and H2e of H2; that is to say, the assumption of H2a that the stronger the environmental group psychology of logistics enterprise is, the more helpful it was to carry out low carbon logistics were established; the assumption of H2c that the stronger the low carbon behavior willingness of logistics enterprise is, the more helpful it was to carry out low carbon logistics was verified; the assumption of H2d that the stronger the self-efficacy of logistics enterprise is, the more helpful it was to carry out low carbon logistics was founded; and the assumption of H2e that the stronger the low carbon behavior capability of logistics enterprise is, the more helpful it was to carry out low carbon logistics was also testified. However, hypothesis H2b was only valid for low carbon input behavior but not low carbon operation behavior, indicating that the assumption that the richer the low carbon related knowledge of logistics enterprise is, the more helpful it was to carry out low carbon logistics is invalid. The hypothesis H3a and H3c of H3 were tenable, which means not only the assumption that the more low carbon economic cost paid by logistics enterprises the more beneficial it was to promote low carbon Logistics, but also the supposition that the more guiding influence of low carbon policies and regulations the more beneficial it was to promote low carbon Logistics are confirmed. While the hypothesis H3b that the greater the low carbon social constraints faced by logistics enterprises are, the more beneficial it was to promote low carbon logistics was merely valid for low carbon operation behavior, the hypothesis H3d that the higher the low carbon technology maturity, the more beneficial it was to promote low carbon logistics was only established for low carbon input behavior.

5. Conclusions and Suggestions

The empirical results partly supported the hypothesis that low carbon logistics behavior was remarkably correlated with influence factors. According to the regression analysis, the main influential factors of low carbon logistics behavior were different and theinfluence degree might vary significantly since there is a great difference between the specific environmental behaviors of low carbon logistics.

The environmental values were significantly associated with low carbon logistics behavior among environmental attitude, and the stronger the environment value of logistics enterprise is, the more helpful it was to carry out low carbon logistics. It shows that the environmental values should be penetrated into all aspects of corporate environmental behavior and become the dominant ideology of low carbon investment behavior and low carbon operations behavior of logistics enterprises. Due to the lower popularity rate of low carbon technology knowledge, a lot of people do not have a clear cognition and concept about low carbon. Otherwise, since the ideology of conquering nature and extorting from nature is popular in the promotion of modern industrial revolution, value deviation among the group lies and there is considerable resistance in the formation and practice of environmental values. Therefore, the corporate environmental propaganda should be strengthened by applying the full power of the social, school, civil, or official and other agencies and by increasing publicity work of low carbon knowledge and skills to increase the popularization rate of environmental education from the quantity and to enrich the content of environmental education from the quality. Changing the traditional indoctrination education mode of environmental education, exerting the role of the nongovernmental environmental protection organization, expanding the environmental cultural industry chain, and promoting the logistics enterprise to change the extensive mode of development were needed to construct the environmental value that harmonizes between enterprises and the nature.

Among the personality characteristics of enterprise, low carbon logistics behavior has high correlation with low carbon behavior intention, low carbon behavior capability, group psychology, and self-efficacy. The stronger the low carbon intention, the low carbon capability, the group psychology, and the self-efficacy are, the more beneficial it was to promote low carbon Logistics. Through the analysis of correlation and regression, personality factors are the driving factors of low carbon operation behavior for logistics enterprises; that is to say, the low carbon operation behavior was driven by inner force. The low carbon consciousness and low carbon behavior intention of logistics enterprises are influenced by many factors such as the government, the market management standard, and enterprise managers, while the scale and technology ability of enterprise can affect the low carbon behavior ability. In order to upgrade the low carbon behavior ability, enterprises are ought to strengthen the cultivation of logistics enterprises' environment ability, to organize learning and communication, to spread the low carbon experience, raise the low carbon consciousness of enterprise managers, encourage low carbon technology innovation, and cultivate enterprise environment ability of development and maintenance in the future. To boost up the willingness of low carbon behavior of logistics enterprises, efforts should be made to perfect the liability regulations of environmental pollution, quantify the compensation standards of environmental pollution, promote environmental responsibility, and safeguard the environmental interests of enterprises, while, among situational factors, low carbon logistics behavior was significantly related to policies and regulations and economic cost. The more the guiding influence of low carbon policies and regulations is and the higher the low carbon cost are, the more beneficial it was to promote low carbon logistics. In particular, policies and regulations have significant effect on low carbon investment behavior and become the first consideration, reflecting that low carbon inputs behavior of logistics enterprises is a behavior driven by external forces and should continually play the role of the policies and regulations in the low carbon environment in the future. In order to provide a guarantee for low carbon logistics, increasing the investment of cleaner production and environmental pollution control through the restraint system, improving the process, refining and perfecting the regulations, and formulating corresponding preferential policies to encourage enterprises to carry out environmental protection training and publicity are in great need. It was needed to clear the reward means and procedures such as tax preferences and subsidies of low carbon and to encourage and guide the enterprises to make positive environmental management action by using the legal means and economic means such as tax, subsidy, and environmental legislation. Besides speeding up the construction of monitoring system, the irregular check of environmental protection was also needed to guide enterprises to low carbon invest and operate.

The object of the study is logistics enterprises. Compared with industrial enterprises, logistics enterprises scale is relatively small, levels of profitability is low, investment of capital, technology, and research is week, and the ability to withstand risks is lack, so the low carbon operation was affected by personality factors such as low carbon behavior ability, and external factors such as policies and regulations are the driving factors of low carbon investment. While the enterprise scale, period of investment return, and scientific research strength are all different, the general industrial enterprises will be more concerned about the personality factors in the process of low carbon input. The research found that environmental values were significantly associated with low carbon logistics, logistics behavior of low carbon invest is significantly affected by external factors such as social norms, and logistics behavior of low carbon operate is the internal driving behavior strongly affected by corporate personality factors such as low carbon behavior ability. The management implications of it is to give advice for government and logistics managers that in order to achieve low carbon logistics goal, environmental propaganda to establish value orientation, play actively the function of policies and regulations in the low carbon logistics, and continue to improve enterprise's ability and willingness of low carbon environment should be strengthened. One of the limitations of this study is that it only studies the common influence factors of low carbon environmental behavior and do not focus on the specific influence factors such as the scale of enterprise and enterprise culture, which might affect the explanatory ability of the regression model more or less. Otherwise, this study is focused on developing low carbon logistics behavior scale and measuring the effects of the three types of influence factors, but it has yet to explore the relationship among variables. The relationship among variables and their impact should be studied and the influence factors such as enterprise culture, type, and scale should be concerned in the future research.

Conflict of Interests

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

Acknowledgments

This paper is the stage results of two subjects funded by Zhejiang Modern Service Research Center, which is a philosophy and social sciences supporting research base in Zhejiang province. The two subjects are low carbon logistics development pathway below e-commerce economy and the mechanism of the effects of e-commerce on low carbon logistics on the background of symbiotic relationship.

References

[1] H.-D. Waisman, C. Guivarch, and F. Lecocq, "The transportation sector and low-carbon growth pathways: modelling urban, infrastructure, and spatial determinants of mobility," *Climate Policy*, vol. 13, no. 1, pp. 106–129, 2013.

[2] P. Oberhofer and E. Furst, "Environmental management in the transport sector: findings of a quantitative survey," *EuroMed Journal of Business*, vol. 7, no. 3, pp. 268–279, 2012.

[3] Y. L. Wang and H. Jiang, "The theoretical frames and evolutionary logic analysis of the enterprise environment behavior," *Foreign Economic and Management*, vol. 8, pp. 26–34, 2012.

[4] I. Ajzen, "From intentions to actions: a theory of planned behavior," in *Action Control*, J. Kuhi and J. Beckmann, Eds., SSSP Springer Series in Social Psychology, pp. 11–39, Springer, Heidelberg, Germany, 1985.

[5] I. Ajzen and T. J. Madden, "Prediction of goal-directed behavior: attitudes, intentions, and perceived behavioral control," *Journal of Experimental Social Psychology*, vol. 22, no. 5, pp. 453–474, 1986.

[6] I. Ajzen, "The theory of planned behavior," *Organizational Behavior and Human Decision Processes*, vol. 50, no. 2, pp. 179–211, 1991.

[7] P. C. Stern and T. Dietz, "The value basis of environmental concern," *Journal of Social Issues*, vol. 50, pp. 65–84, 1994.

[8] P. C. Stern, T. Dietz, T. Abel, G. A. Guagnano, and L. Kalof, "A value-belief-norm theory of support for social movements: the case of environmentalism," *Human Ecology Review*, vol. 6, no. 2, pp. 81–97, 1999.

[9] P. C. Stern, "Toward a coherent theory of environmentally significant behavior," *Journal of Social Issues*, vol. 56, no. 3, pp. 407–424, 2000.

[10] G. A. Guagnano, P. C. Stern, and T. Dietz, "Influences on attitude-behavior relationships: a natural experiment with curbside recycling," *Environment and Behavior*, vol. 27, no. 5, pp. 699–718, 1995.

[11] H. Kotzab, H. M. Munch, B. de Faultrier, and C. Teller, "Environmental retail supply chains: when global Goliaths become environmental Davids," *International Journal of Retail and Distribution Management*, vol. 39, no. 9, pp. 658–681, 2011.

[12] R. E. Dunlap, "Environmental s ociology: a personal perspective on its first quarter century," *Organization and Environment*, vol. 15, no. 1, pp. 10–29, 2002.

[13] P. Wesley Schultz and L. Zelezny, "Values as predictors of environmental attitudes: evidence for consistency across 14 countries," *Journal of Environmental Psychology*, vol. 19, no. 3, pp. 255–265, 1999.

[14] J. Thøgersen and F. Ölander, "To what degree are environmentally beneficial choices reflective of a general conservation stance?" *Environment & Behavior*, vol. 38, no. 4, pp. 550–569, 2006.

[15] A. M. Nordlund and J. Garvill, "Value structures behind proenvironmental behavior," *Environment & Behavior*, vol. 34, no. 6, pp. 740–756, 2002.

[16] J. M. Hines, H. R. Hungerford, and A. N. Tomera, "Analysis and synthesis is of research on responsible environmental behavior: a meta-analysis," *The Journal of Environmental Education*, vol. 18, no. 2, pp. 1–8, 1986.

[17] T. Gärling, S. Fujii, A. Gärling, and C. Jakobsson, “Moderating effects of social value orientation on determinants of proenvironmental behavior intention,” *Journal of Environmental Psychology*, vol. 23, no. 1, pp. 1–9, 2003.

[18] F. G. Kaiser, S. Wölfing, and U. Fuhrer, “Environmental attitude and ecological behaviour,” *Journal of Environmental Psychology*, vol. 19, no. 1, pp. 1–19, 1999.

[19] L. M. Andersson and T. S. Bateman, “Individual environmental initiative: championing natural environmental issues in U.S. business organizations,” *Academy of Management Journal*, vol. 43, no. 4, pp. 548–570, 2000.

[20] D. A. Waldman and D. Siegel, “Defining the socially responsible leader,” *Leadership Quarterly*, vol. 19, no. 1, pp. 117–131, 2008.

[21] M. D. López-Gamero, E. Claver-Cortés, and J. F. Molina-Azorín, “Complementary resources and capabilities for an ethical and environmental management: a qual/quan study,” *Journal of Business Ethics*, vol. 82, no. 3, pp. 701–732, 2008.

[22] C.-Y. Lin and Y.-H. Ho, “The influences of environmental uncertainty on corporate green behavior: an empirical study with small and medium-size enterprises,” *Social Behavior and Personality*, vol. 38, no. 5, pp. 691–696, 2010.

[23] C. Lee and R. T. Green, “Cross-cultural examination of the Fishbein behavior intention model,” *Journal of International Business Studies*, vol. 22, no. 2, pp. 289–305, 1991.

[24] G. T. S. Ho, K. L. Choy, C. H. Y. Lam, and D. W. C. Wong, “Factors influencing implementation of reverse logistics: a survey among Hong Kong businesses,” *Measuring Business Excellence*, vol. 16, no. 3, pp. 29–46, 2012.

[25] A. Biglan, “The role of advocacy organizations in reducing negative externalities,” *Journal of Organizational Behavior Management*, vol. 29, no. 3, pp. 215–230, 2009.

[26] M. Fu, “The environmental Kuznets curve in China: morphology, inflection point and influence factors,” *Quantitative & Technical Economics*, vol. 11, pp. 40–50, 2008.

[27] C.-Y. Lin and Y.-H. Ho, “RFID technology adoption and supply chain performance: an empirical study in China’s logistics industry,” *Supply Chain Management*, vol. 14, no. 5, pp. 369–378, 2009.

[28] S. X. Li, “Preference, technology and the environment Kuznets curve,” *Journal of Zhongnan University of Economics and Law*, vol. 1, pp. 31–38, 2012.

[29] R. E. Dunlap, K. D. V. Liere, A. G. Mertig, and R. E. Jones, “Measuring endorsement of the new ecological paradigm: a revised NEP scale,” *Journal of Social Issues*, vol. 56, no. 3, pp. 425–442, 2000.

[30] J. Frick, F. G. Kaiser, and M. Wilson, “Environmental knowledge and conservation behavior: exploring prevalence and structure in a representative sample,” *Personality and Individual Differences*, vol. 37, no. 8, pp. 1597–1613, 2004.

[31] R. Y. K. Chan, “Determinants of Chinese consumers'green purchase behavior,” *Psychology and Marketing*, vol. 18, no. 4, pp. 389–413, 2001.

[32] M. Sherer, J. E. Maddux, B. Mercandante, B. Jacobs, and R. W. Rogers, “The self-efficacy scale: construction and validation,” *Psychological Reports*, vol. 51, pp. 663–671, 1982.

[33] C. Tanner and S. W. Kast, “Promoting sustainable consumption: determinants of green purchases by Swiss consumers,” *Psychology and Marketing*, vol. 20, no. 10, pp. 883–902, 2003.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 681854, 8 pages
http://dx.doi.org/10.1155/2015/681854

Research Article

A Single-Machine Two-Agent Scheduling Problem by a Branch-and-Bound and Three Simulated Annealing Algorithms

Shangchia Liu,[1] Wen-Hsiang Wu,[2] Chao-Chung Kang,[3] Win-Chin Lin,[4] and Zhenmin Cheng[5]

[1] *Business Administration Department, Fu Jen Catholic University, New Taipei City 24205, Taiwan*
[2] *Department of Healthcare Management, Yuanpei University, Hsinchu 30015, Taiwan*
[3] *Department of Business Administration and Graduate Institute of Management, Providence University, Shalu, Taichung 43301, Taiwan*
[4] *Department of Statistics, Feng Chia University, Taichung 40724, Taiwan*
[5] *Business College, Beijing Union University, Beijing 100101, China*

Correspondence should be addressed to Shangchia Liu; 056298@mail.fju.edu.tw

Received 30 June 2014; Accepted 11 August 2014

Academic Editor: Chin-Chia Wu

In the field of distributed decision making, different agents share a common processing resource, and each agent wants to minimize a cost function depending on its jobs only. These issues arise in different application contexts, including real-time systems, integrated service networks, industrial districts, and telecommunication systems. Motivated by its importance on practical applications, we consider two-agent scheduling on a single machine where the objective is to minimize the total completion time of the jobs of the first agent with the restriction that an upper bound is allowed the total completion time of the jobs for the second agent. For solving the proposed problem, a branch-and-bound and three simulated annealing algorithms are developed for the optimal solution, respectively. In addition, the extensive computational experiments are also conducted to test the performance of the algorithms.

1. Introduction

Multiagent simulation is a branch of artificial intelligence that offers a promising approach to dealing with multistakeholder management systems, such as common pool resources. It provides a framework which allows analysis of stakeholders' (or agents') interactions and decision making. For example, in forest plantation comanagement, Purnomo and Guizol [1] further pointed out "Comanagement of forest resources is a process of governance that enables all relevant stakeholders to participate in the decision-making processes. Illegal logging and forest degradation are currently increasing, and logging bans are ineffective in reducing forest degradation. At the same time interest in forest plantations and concern about poverty problems of neighboring people whose livelihoods depend on forest services and products continue to increase rapidly. Governments have identified the development of small forest plantations as an opportunity to provide wood supplies to forest industries and to reduce poverty. However, the development of small plantations is very slow due to an imbalance of power and suspicion between communities and large companies." In blood cell population dynamics, Bessonov et al. [2] gave some possible explanations of the mechanism for recovery of the system under important blood loss or blood diseases such as anemia. Agnetis et al. [3] also indicated that multiple agents compete on the usage of a common processing resource in different application environments and different methodological fields, such as artificial intelligence, decision theory, operations research, and so forth.

Scheduling with multiple agents has received growing attention in recently years. Agnetis et al. [4] and Baker and Smith [5] were among the pioneers to introduce the concept of multiagent into scheduling problems. Yuan et al. [6] discussed two dynamic programming recursions in Baker and Smith [5] and proposed a polynomial-time algorithm for the same problem. Cheng et al. [7] addressed the feasibility model of multiagent scheduling on a single machine where each agent's objective function is to minimize the total weighted number of tardy jobs. Ng et al. [8] proposed a two-agent scheduling problem on a single machine, where the objective is to minimize the total completion time of the first agent with the restriction that the number of tardy jobs of the second agent cannot exceed a given number. Agnetis et al. [3] discussed the complexity some single-machine scheduling problems with several agents. Cheng et al. [9] studied multiagent scheduling on a single machine where the objective functions of the agents are of the max-form. Agnetis et al. [10] used a branch-and-bound method to solve several two-agent scheduling problems on a single machine. Lee et al. [11] considered a multiagent scheduling problem on a single machine in which each agent is responsible for his own set of jobs and wishes to minimize the total weighted completion time of his own set of jobs. Recently, Leung et al. [12] generalized the single machine problems proposed by Agnetis et al. [4] to the environment with multiple identical machines in parallel. Yin et al. [13] considered several two-agent scheduling problems with assignable due dates on a single machine, where the goal was to assign a due date from a given set of due dates and a position in the sequence to each job so that the weighted sum of the objectives of both agents is minimized. For different combinations of the objectives, which include the maximum lateness, total (weighted) tardiness, and total (weighted) number of tardy jobs, they provided the complexity results and solve the corresponding problems, if possible. Yin et al. [14] investigated a scheduling environment with two agents and a linear nonincreasing deterioration with the objective of scheduling the jobs such that the combined schedule performs well with respect to the measures of both agents. Three different objective functions were considered for one agent, including the maximum earliness cost, total earliness cost, and total weighted earliness cost, while keeping the maximum earliness cost of the other agent below or at a fixed level U. They proposed the optimal (nondominated) properties and present the complexity results for the problems. Yin et al. [15] addressed a two-agent scheduling problem on a single machine where the objective is to minimize the total weighted earliness cost of all jobs, while keeping the earliness cost of one agent below or at a fixed level Q. A mixed-integer programming (MIP) model was first formulated to find the optimal solution which is useful for small-size problem instances and then a branch-and-bound algorithm incorporating with several dominance properties, a lower bound and a simulated annealing heuristic algorithm were provided to derive the optimal solution for medium- to large-size problem instances. For more recent works with two-agent issues, readers can refer to Nong et al. [16], Wan et al. [17], Luo et al. [18], Yin et al. [19], and so forth. In addition, for more multiple-agent works with time-dependent, readers can refer to Liu and Tang [20], Liu et al. [21], Cheng et al. [22, 23], Wu et al. [24], Li and Hsu [25], Yin et al. [26–28], Wu [29], and Li et al. [30], and so forth.

For the importance of multiple agents competing on the usage of a common processing resource in different application environments and different methodological fields, we considered two-agent scheduling on a single machine. The objective is to minimize the total completion time of the jobs of the first agent with the restriction that an upper bound is allowed the total completion time of the jobs for the second agent which has been shown binary NP-hard by Agnetis et al. [4]. The problem is described as follows. There are n jobs which belong to one of the agents AG_0 or AG_1. For each job j, there is a normal processing time p_j and an agent code I_j, where $I_j = 0$ if $J_j \in AG_0$ or $I_j = 1$ if $J_j \in AG_1$. All the jobs are available at time zero. Under a schedule S, let $C_j(S)$ be the completion time of job j. The objective of this paper is to find an optimal schedule to minimize $\sum_{j=1}^{n} C_j(S)(1-I_j)$ subject to $\sum_{j=1}^{n} C_j(S)I_j < U$, where U is a control upper bound.

The remainder of this paper is organized as follows. In Section 2 we present some dominance properties and develop a lower bound to speed up the search for the optimal solution, followed by discussions of a branch-and-bound and three simulated annealing (SA) algorithms. We present the results of extensive computational experiments to assess the performance of all of the proposed algorithms under different experimental conditions in Section 3. We conclude the paper and suggest topics for further research in the last section.

2. Dominance Properties

We will develop some adjacent dominance rules by using the pairwise interchange method. Assume that S_1 and S_2 denote two given job schedules in which the difference between S_1 and S_2 is a pairwise interchange of two adjacent jobs i and j. That is, $S_1 = (\sigma, i, j, \sigma')$ and $S_2 = (\sigma, j, i, \sigma')$, where σ and σ' each denote a partial sequence. In addition, let t be the starting time of the last job in σ.

Property 1. If jobs $i, j \in AG_0$, and $p_i < p_j$, then S_1 dominates S_2.

Property 2. If jobs $i, j \in AG_1$, $\sum_{i\in\sigma\cap AG_1} C_i + 2t + 2p_i + p_j < U$, and $p_i < p_j$, then S_1 dominates S_2.

Next, we give a proposition to determine the feasibility of the partial schedule. Let (π, π^c) be a sequence of jobs where π is the scheduled part with k jobs and σ^c is the unscheduled part with $(n-k)$ jobs. Among the unscheduled jobs, let $p_{(1)} = \min_{J_j\in\pi^c\cap AG_1}\{p_j\}$. Moreover, let $C_{[k]}$ be the completion times of the last job in π. Also, let π' and π'' denote the unscheduled jobs in AG_0 and AG_1 arranged in the smallest processing times (SPT) order, respectively.

Property 3. If there is a job $i \in AG_1$ such that $\sum_{i\in\pi\cap AG_1} C_i + C_{[k]} + p_{(1)} > U$, then (π, π^c) is not a feasible sequence.

Property 4. If all unscheduled jobs are belonging to AG_0, then (π, π') dominates (π, π^c).

Property 5. If all unscheduled jobs are belonging to AG_1, then (π, π^c) can be determined by (π, π''). That is, (π, π'') either is feasible solution or can be deleted.

2.1. Lower Bound. Assume that PS is a partial schedule in which the order of the first k jobs is determined and let US be the unscheduled part with $(n - k)$ jobs. Among the unscheduled jobs, there are n_0 jobs from agent AG_0 and n_1 jobs from agent AG_1. Moreover, let $C_{[k]}$ denote the completion times of the kth job in PS. The completion time for the $(k + j)$th job is

$$C_{[k+j]} \geq C_{[k]} + \sum_{i=1}^{j} p_{(k+i)}, \quad \text{for } 1 \leq j \leq n_0. \tag{1}$$

Then a lower bound can be obtained as follows:

$$\sum_{j=1}^{n} C_j(S)\left(1 - I_j\right) \geq \sum_{j=1}^{k} C_j(S)\left(1 - I_j\right) + \sum_{j=1}^{n_0} \widehat{C}_{(j)}(S), \tag{2}$$

where $\widehat{C}_{(j)} = C_{[k]} + \sum_{i=1}^{j} p_{(k+i)}$.

2.2. Simulated Annealing Algorithms. Simulated annealing has become one of the most popular metaheuristic methods to solve combinatorial optimization problems since it was proposed by Kirkpatrick et al. [31]. For example, Kim et al. [32] applied the method in scheduling of raw-material unloading from ships at a steelworks. Sun et al. [33] used the technique for allocating product lots to customer orders in semiconductor manufacturing supply chains. Moreover, the method has the advantage of avoiding getting trapped in a local optimum because of its hill climbing moves, which are governed by a control parameter. Thus, we will use SA to derive near-optimal solutions for our problem. The steps of SA algorithms are summarized as follows.

2.2.1. Initial Sequence. Three initial sequences for the three SA algorithms are adopted for our problem. For the first initial sequence in SA_1, the sequence of jobs of AG_1 is arranged according to the smallest processing times (SPT) order, followed by arranging the sequence of jobs of AG_0 in the shortest processing time (SPT) order. In order to get a good initial one, two more initial sequences are considered. For the second initial sequence in SA_2, the pairwise interchange is used to the initial sequence produce by SA_1. For the third initial sequence in SA_3, the NEH method [34] is applied to the initial sequence produce by SA_1.

2.2.2. Neighborhood Generation. The pairwise interchange (PI) neighborhood generation method is adopted in the algorithms.

2.2.3. Acceptance Probability. When a new feasible sequence is generated, it is accepted if its objective value is smaller than that of the original sequence; otherwise, it is accepted with a probability that decreases as the process evolves. The probability of acceptance is generated from an exponential distribution:

$$P(\text{accept}) = \exp(-\alpha \times \Delta\text{TC}), \tag{3}$$

where α is a control parameter and ΔTC is the change in the objective value. In addition, we use the method suggested by Ben-Arieh and Maimon [35] to change α in the kth iteration as follows:

$$\alpha = \frac{k}{\delta}, \tag{4}$$

where δ is an experimental constant. After preliminary trials, $\delta = 6$ is used in our experiments.

If the total completion time of jobs from agent AG_0 increases as a result of a random pairwise interchange, the new sequence is accepted when $P(\text{accept}) > r$, where r is randomly sampled from the uniform distribution $U(0, 1)$.

2.2.4. Stopping Rule. All three proposed SAs are stopped after $100n$ iterations, where n is the number of jobs.

3. Computational Experiments

The extensive computational experiments were conducted to test the performances of the branch-and-bound algorithm and the three simulated annealing algorithms. All the algorithms were coded in Fortran using Compaq Visual Fortran version 6.6 and performed the experiments on a personal computer powered by an Intel(R) Core(TM)2 Quad CPU 2.66 GHz with 4 GB RAM operating under Windows XP. The job processing times were generated from a uniform distribution over the integers 1–100. For the control upper bound U, we arrange the jobs of AG_1 by the smallest processing times (SPT) order and compute out the total completion times of the jobs of AG_1, (recorded as TC_1). Then following the sequence of the jobs of AG_1, we arrange the sequence of jobs of AG_0 in the shortest processing time (SPT) order and compute out the total completion times of jobs of AG_1 and AG_0, (recoded as TC_{1+2}). We let $U = (1 - \alpha)TC_1 + \alpha TC_{1+2}$, where $0 < \alpha < 1$. Moreover, α was taken as the values of 0.25, 0.5, and 0.75, while the proportion of the jobs of agent AG_1 at pro = 0.25, 0.5, and 0.75 in the tests.

For the branch-and-bound algorithm, the average and standard deviation of the numbers of nodes and the execution times (in seconds) were recorded. For the SA heuristics, the mean and standard deviation percentage errors were recorded. The percentage error of a solution obtained from a heuristic algorithm was given by

$$\frac{V - V^*}{V^*}, \tag{5}$$

where V and V^* are denoted as the total completion time of the heuristic and the optimal solution, respectively. The computational times of the heuristic algorithms were not recorded because they all were fast in generating solutions.

The computational experiments were divided into two parts. In the first part of the experiments, three number of jobs were tested at $n = 8$, 12, and 16. As a result, 27 experimental situations were examined. 50 instances were randomly generated for each case. The results were summarized in Table 1 that includes the CPU time (mean and standard deviation) and the number of nodes for the branch-and-bound algorithm.

Table 1: Performance of the branch-and-bound and heuristic algorithms (n = 8~16).

n	Pro	α	Valid sample size	Branch-and-bound algorithm CPU time Mean	Std.	Number of nodes Mean	Std.	SA1 Error percentages Mean	Std.	SA2 Mean	Std.	SA3 Mean	Std.	SA4 Mean	Std.
8	0.25	0.25	50	0.00	0.01	526	318	0.28	1.46	0.28	1.46	0.28	1.46	0.28	1.46
		0.50	50	0.00	0.01	748	588	1.29	4.12	0.93	3.11	0.86	3.09	0.24	1.69
		0.75	50	0.01	0.01	1383	923	3.42	7.29	2.83	5.73	2.27	4.80	0.62	2.46
	0.50	0.25	50	0.01	0.01	2229	1371	0.01	0.06	0.01	0.06	0.00	0.00	0.00	0.00
		0.50	50	0.01	0.01	2500	1412	0.50	1.78	0.14	0.56	0.20	0.86	0.04	0.20
		0.75	50	0.01	0.01	3183	1139	0.92	2.03	1.27	2.26	1.48	2.76	0.26	1.02
	0.75	0.25	50	0.01	0.01	1091	217	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.50	50	0.01	0.01	1091	217	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.75	50	0.01	0.01	1089	196	0.36	1.18	0.11	0.48	0.49	1.33	0.03	0.24
	Average			0.01	0.01	1538	709	0.75	1.99	0.62	1.52	0.62	1.59	0.16	0.79
12	0.25	0.25	50	0.78	0.86	97087	113419	0.29	0.93	0.61	2.30	0.30	1.38	0.05	0.20
		0.50	50	1.36	1.96	175858	261636	1.86	4.41	3.99	7.81	1.45	3.86	0.77	1.95
		0.75	50	3.92	5.24	535992	751273	4.78	7.25	4.40	6.82	5.30	8.43	2.12	4.26
	0.50	0.25	50	19.83	22.75	2278911	2794764	0.02	0.07	0.04	0.20	0.01	0.05	0.00	0.00
		0.50	50	27.91	33.86	3405293	4448908	0.64	1.24	0.53	1.16	0.32	0.73	0.17	0.45
		0.75	50	35.38	21.51	4692766	3096131	2.07	2.16	2.37	2.43	2.37	2.74	0.69	0.92
	0.75	0.25	50	6.30	2.18	794003	304461	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.50	50	6.30	2.18	794439	304555	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.75	50	6.60	1.82	862132	262277	0.44	0.66	0.74	1.22	0.55	1.19	0.15	0.33
	Average			12.04	10.26	1515165	1370825	1.12	1.86	1.41	2.44	1.14	2.04	0.44	0.90
16	0.25	0.25	48	846.82	1691.36	62738963	129480137	0.36	0.92	0.39	1.06	0.35	1.16	0.18	0.77
		0.50	46	1432.63	2411.56	105554536	181805362	1.99	3.46	2.49	4.28	1.88	4.38	0.95	2.29
		0.75	33	1446.76	2629.13	106507557	197947458	2.72	3.14	7.79	12.00	4.30	6.64	1.83	2.28
	0.50	0.25	22	8044.31	6058.59	429572906	333869433	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.50	20	7267.99	5697.96	387550386	314387045	0.10	0.29	0.00	0.00	0.11	0.51	0.00	0.00
		0.75	3	7493.01	5776.30	411927486	331822988	0.93	1.07	6.91	6.17	6.65	4.11	0.93	1.07
	0.75	0.25	13	9502.54	3053.15	625205281	214788228	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.50	13	9478.74	3044.52	625205281	214788228	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.75	7	9965.98	3653.10	652692683	258677186	0.41	0.73	0.24	0.63	1.03	1.06	0.00	0.00
	Average			6164.31	3779.52	378550564	241951785	0.72	1.07	1.98	2.68	1.59	1.98	0.43	0.71

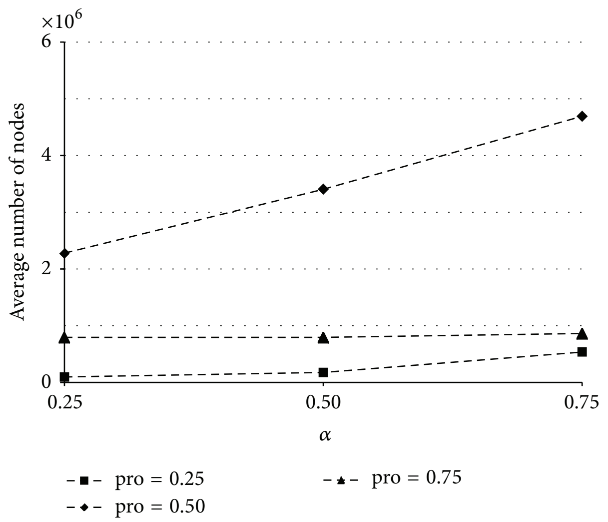


Figure 1: Performance of the branch-and-bound algorithms (n = 12).

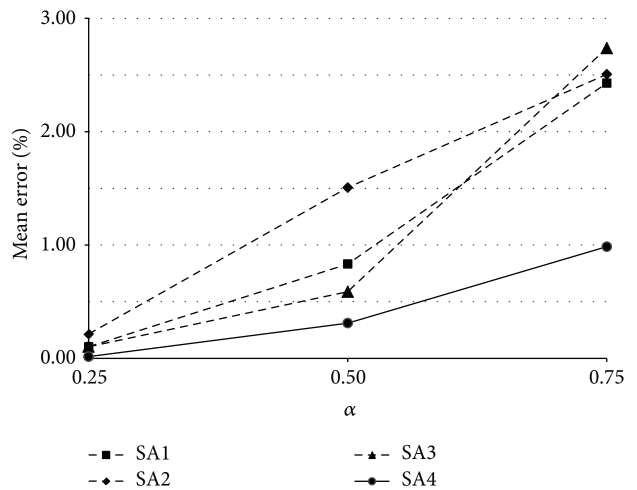


Figure 2: Performance of the SA algorithms (n = 12).

For the performance of the branch-and-bound algorithm, it can be observed from Table 1 that the number of nodes and the mean of the CPU time increase when n becomes bigger. The difficult situations occur at $n = 16$. Especially, the instances with a bigger value of α ($\alpha = 0.75$) are difficult to solve than those with a smaller one ($\alpha = 0.25, 0.50$). Moreover, the instances with a bigger value of pro (pro = 0.5, 0.75) are difficult to solve than those with a smaller one (pro = 0.25). The most difficult case is located at pro = 0.5, 0.75, and $\alpha = 0.75$ as the number of instances, which can be solved out within less 10^{-9} nodes, were declined to 10 or below. In addition, as shown in Figure 1 and Table 1, the instances with pro = 0.5 took more nodes or CPU time than others, and the trend became clear as the value of α got bigger.

For the performance of the proposed SAs, it can be seen in Table 1 that the means of the percentage error of SA_1 (between 0.0% and 4.78%) are lower than those of SA_2 (between 0.0% and 7.79%) and those of SA_3 (between 0.0% and 6.65%.) Moreover, the trend of the standard deviations of the percentage error keeps the similar pattern. As shown in Table 1, the standard deviations of the percentage error of SA_1, SA_2, and SA_3 were between 0.0% and 7.29%, 0.0% and 12.0%, and 0.0% and 8.43%, respectively. It released that the performance of SA_1 was slightly better than the other two. Furthermore, Figure 2 and Table 1 indicated that the means of the percentage error of SA_1, SA_2, and SA_3 are slightly affected by the parameter α. For example, the means of the percentage errors of SA_1, SA_2, and SA_3 were lager than 2% at $\alpha = 0.25$. Specifically, the behavior became clear at pro = 0.25 and 0.5. It also can be observed that there is no absolutely dominance relationship among three proposed SAs. Due to get a good quality solution, we further combined three proposed SAs into one (recorded as SA_4 and $SA_4 = \min\{SA_i, i = 1, 2, 3\}$). The means of the percentage error of SA_4 were located between 0.0% and 2.12%. Meanwhile its standard deviations of the percentage error were between 0.0% and 4.26%. Moreover, the impacts of pro and α were not present for proposed three SAs.

In the second part of the experiments, the performances of the proposed SA heuristics were further tested for large numbers of jobs. Three different numbers of jobs were considered at $n = 20$, 40, and 60. The proportion of the jobs of agent AG_1 at pro = 0.25, 0.5, and 0.75 in the tests. Moreover, α was taken as the values of 0.25, 0.5, and 0.75. As a result, 27 experimental situations were tested. 50 instances were randomly generated for each situation. The relative deviance percentage with respect to the best known solution was reported for each instance. The mean execution time and mean relative deviance percentage were also recorded for each SA heuristic. The relative deviation percentage RDP was given by

$$\frac{SA_i - SA^*}{SA^*}, \tag{6}$$

where SA_i is the value of the objective function generated by SA_i and $SA^* = \min\{SA_i, i = 1, 2, 3\}$ is the smallest value of the objective function obtained from the three SA heuristics. The results were summarized in Table 2.

As shown in Figure 3 and Table 2, it can be seen that the RPD means of SA_1, SA_2, and SA_3 are getting slightly bigger as the value of α increases. In general, the RPD means of SA_3 are lower than those of SA_1 and SA_2. Furthermore, all of the RDP means of SA_1, SA_2, and SA_3 were less than 2%. Figure 2 also indicated that there is no absolutely dominance relationship among three proposed SAs.

4. Conclusions

This paper explored the single-machine two-agent scheduling problem where the objective is to minimize the total completion time of the jobs belonging to the first agent with the restriction that total completion time of jobs from the second agent has an upper bound. Due to the fact that the problem under study is binary NP hard, a branch-and-bound

Table 2: RPD of heuristic algorithms (n = 20~60).

n	Pro	α	SA1 CPU time Mean	SA1 CPU time Std.	SA1 RPD Mean	SA1 RPD Std.	SA2 CPU time Mean	SA2 CPU time Std.	SA2 RPD Mean	SA2 RPD Std.	SA3 CPU time Mean	SA3 CPU time Std.	SA3 RPD Mean	SA3 RPD Std.
20	0.25	25	0.0019	0.0051	0.45	1.44	0.0019	0.0051	0.41	1.19	0.0016	0.0047	0.51	1.75
		50	0.0019	0.0051	1.39	2.67	0.0019	0.0051	1.87	5.51	0.0022	0.0055	0.89	1.88
		75	0.0019	0.0051	2.86	4.81	0.0019	0.0051	4.96	6.80	0.0019	0.0051	2.01	3.66
	0.5	25	0.0019	0.0051	0.02	0.07	0.0019	0.0051	0.00	0.01	0.0016	0.0047	0.01	0.06
		50	0.0019	0.0051	0.51	0.81	0.0019	0.0051	0.39	0.67	0.0022	0.0055	0.51	0.78
		75	0.0019	0.0051	1.05	1.52	0.0022	0.0055	1.19	2.08	0.0022	0.0055	1.30	2.03
	0.75	25	0.0019	0.0051	0.00	0.00	0.0013	0.0043	0.00	0.00	0.0019	0.0051	0.00	0.00
		50	0.0019	0.0051	0.00	0.00	0.0016	0.0047	0.00	0.00	0.0019	0.0051	0.00	0.00
		75	0.0019	0.0051	0.50	0.58	0.0019	0.0051	0.58	0.75	0.0019	0.0051	0.34	0.55
Average			0.0019	0.0051	0.75	1.32	0.0018	0.0050	1.04	1.89	0.0019	0.0052	0.62	1.19
40	0.25	25	0.0066	0.0078	0.10	0.30	0.0072	0.0079	0.16	0.52	0.0066	0.0078	0.00	0.00
		50	0.0072	0.0079	0.92	1.51	0.0078	0.0079	1.22	2.09	0.0072	0.0079	0.39	1.47
		75	0.0078	0.0079	3.66	4.43	0.0084	0.0079	3.91	5.84	0.0081	0.0079	2.82	4.04
	0.5	25	0.0066	0.0078	0.02	0.06	0.0066	0.0078	0.00	0.00	0.0066	0.0078	0.00	0.00
		50	0.0075	0.0079	0.39	0.43	0.0078	0.0079	0.31	0.57	0.0078	0.0079	0.46	0.79
		75	0.0084	0.0079	0.81	0.81	0.0091	0.0078	1.03	1.69	0.0091	0.0078	0.71	1.12
	0.75	25	0.0066	0.0078	0.00	0.00	0.0063	0.0077	0.00	0.00	0.0063	0.0077	0.00	0.00
		50	0.0063	0.0077	0.00	0.00	0.0063	0.0077	0.00	0.00	0.0063	0.0077	0.00	0.00
		75	0.0072	0.0079	0.32	0.45	0.0078	0.0079	0.25	0.36	0.0081	0.0079	0.25	0.31
Average			0.0071	0.0078	0.69	0.89	0.0075	0.0078	0.76	1.23	0.0073	0.0078	0.51	0.86
60	0.25	25	0.0159	0.0022	0.07	0.20	0.0178	0.0055	0.09	0.23	0.0147	0.0037	0.02	0.11
		50	0.0175	0.0051	1.48	2.72	0.0197	0.0069	1.20	1.88	0.0163	0.0044	0.28	0.88
		75	0.0197	0.0069	2.38	2.51	0.0219	0.0077	3.74	5.10	0.0206	0.0074	2.58	4.10
	0.5	25	0.0147	0.0037	0.01	0.03	0.0147	0.0037	0.00	0.00	0.0144	0.0043	0.00	0.00
		50	0.0175	0.0060	0.39	0.57	0.0184	0.0061	0.27	0.35	0.0181	0.0066	0.23	0.42
		75	0.0209	0.0075	0.66	1.03	0.0231	0.0079	0.62	0.73	0.0222	0.0078	0.66	0.90
	0.75	25	0.0141	0.0047	0.00	0.00	0.0141	0.0047	0.00	0.00	0.0141	0.0047	0.00	0.00
		50	0.0141	0.0047	0.00	0.00	0.0141	0.0047	0.00	0.00	0.0141	0.0047	0.00	0.00
		75	0.0175	0.0051	0.21	0.27	0.0197	0.0069	0.22	0.28	0.0194	0.0067	0.21	0.25
Average			0.0169	0.0051	0.58	0.82	0.0182	0.0060	0.68	0.95	0.0171	0.0056	0.44	0.74

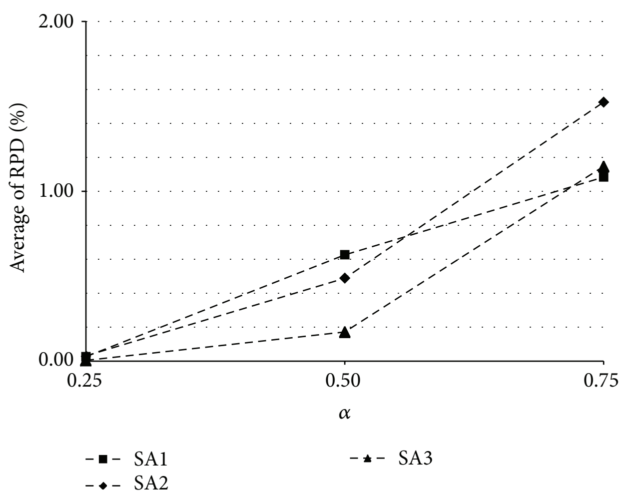


Figure 3: Performance of the SA algorithms (n = 60).

algorithm incorporating several dominance properties and a lower bound was proposed for the optimal solution. Three simulated annealing algorithms were also proposed for near-optimal solutions. The computational results show that, with the help of the proposed heuristic initial solutions, the branch-and-bound algorithm performs well in terms of number of nodes and execution time when the number of jobs is fewer than or equal to 16. Moreover, the computational experiments also show that the proposed combined SA_4 performs well since its mean error percentage was less than 2.12% for all the tested cases. Further research lies in devising efficient and effective methods to solve the problem with significantly larger numbers of jobs.

Conflict of Interests

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

Acknowledgment

This research was supported in part by the foundation of Beijing Municipal Education on Science and Technology Plan Project (KM 20121417015).

References

[1] H. Purnomo and P. Guizol, “Simulating forest plantation co-management with a multi-agent system,” *Mathematical and Computer Modelling*, vol. 44, no. 5-6, pp. 535–552, 2006.

[2] N. Bessonov, I. Demin, L. Pujo-Menjouet, and V. Volpert, “A multi-agent model describing self-renewal of differentiation effects on the blood cell population,” *Mathematical and Computer Modelling*, vol. 49, no. 11-12, pp. 2116–2127, 2009.

[3] A. Agnetis, D. Pacciarelli, and A. Pacifici, “Multi-agent single machine scheduling,” *Annals of Operations Research*, vol. 150, no. 1, pp. 3–15, 2007.

[4] 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.

[5] 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.

[6] J. J. Yuan, W. P. Shang, and Q. Feng, “A note on the scheduling with two families of jobs,” *Journal of Scheduling*, vol. 8, no. 6, pp. 537–542, 2005.

[7] T. C. 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.

[8] C. T. Ng, T. C. 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. 387–394, 2006.

[9] T. C. 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.

[10] A. Agnetis, G. de Pascale, and D. Pacciarelli, “A Lagrangian approach to single-machine scheduling problems with two competing agents,” *Journal of Scheduling*, vol. 12, no. 4, pp. 401–415, 2009.

[11] K. Lee, B. C. Choi, J. Y. 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.

[12] J. Y. Leung, M. Pinedo, and G. Wan, “Competitive two-agent scheduling and its applications,” *Operations Research*, vol. 58, no. 2, pp. 458–469, 2010.

[13] Y. Yin, S.-R. Cheng, T. C. Cheng, C.-C. Wu, and W.-H. Wu, “Two-agent single-machine scheduling with assignable due dates,” *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1674–1685, 2012.

[14] 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.

[15] Y. Yin, C.-C. Wu, W.-H. Wu, C.-J. Hsu, and W.-H. Wu, “A branch-and-bound procedure for a single-machine earliness scheduling problem with two agents,” *Applied Soft Computing Journal*, vol. 13, no. 2, pp. 1042–1054, 2013.

[16] Q. Q. Nong, T. C. Cheng, and C. T. Ng, “Two-agent scheduling to minimize the total cost,” *European Journal of Operational Research*, vol. 215, no. 1, pp. 39–44, 2011.

[17] G. Wan, S. R. Vakati, J. Y. Leung, and M. Pinedo, “Scheduling two agents with controllable processing times,” *European Journal of Operational Research*, vol. 205, no. 3, pp. 528–539, 2010.

[18] W. Luo, L. Chen, and G. Zhang, “Approximation schemes for two-machine flow shop scheduling with two agents,” *Journal of Combinatorial Optimization*, vol. 24, no. 3, pp. 229–239, 2012.

[19] 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.

[20] P. Liu and L. Tang, “Two-agent scheduling with linear deteriorating jobs on a single machine,” in *Computing and Combinatorics: Proceedings of the 14th Annual International Conference, COCOON 2008 Dalian, China, June 27–29, 2008*, vol. 5092 of *Lecture Notes in Computer Science*, pp. 642–650, Springer, Berlin, Germany, 2008.

[21] P. Liu, N. Yi, and X. Zhou, “Two-agent single-machine scheduling problems under increasing linear deterioration,” *Applied Mathematical Modelling*, vol. 35, no. 5, pp. 2290–2296, 2011.

[22] T. C. E. Cheng, W.-H. Wu, S.-R. Cheng, and C.-C. Wu, “Two-agent scheduling with position-based deteriorating jobs and learning effects,” *Applied Mathematics and Computation*, vol. 217, no. 21, pp. 8804–8824, 2011.

[23] T. C. Cheng, Y. H. Chung, S. C. Liao, and W. C. Lee, “Two-agent singe-machine scheduling with release times to minimize the total weighted completion time,” *Computers & Operations Research*, vol. 40, no. 1, pp. 353–361, 2013.

[24] C.-C. Wu, S.-K. Huang, and W.-C. Lee, “Two-agent scheduling with learning consideration,” *Computers and Industrial Engineering*, vol. 61, no. 4, pp. 1324–1335, 2011.

[25] D.-C. Li and P.-H. Hsu, “Solving a two-agent single-machine scheduling problem considering learning effect,” *Computers and Operations Research*, vol. 39, no. 7, pp. 1644–1651, 2012.

[26] Y. Yin, M. Liu, J. Hao, and M. Zhou, “Single-machine scheduling with job-position-dependent learning and time-dependent deterioration,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 42, no. 1, pp. 192–200, 2012.

[27] Y. 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.

[28] Y. 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.

[29] W.-H. Wu, “Solving a two-agent single-machine learning scheduling problem,” *International Journal of Computer Integrated Manufacturing*, vol. 27, no. 1, pp. 20–35, 2014.

[30] D.-C. Li, P.-H. Hsu, and C.-C. Chang, “A genetic algorithm-based approach for single-machine scheduling with learning effect and release time,” *Mathematical Problems in Engineering*, vol. 2014, Article ID 249493, 12 pages, 2014.

[31] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[32] B.-I. Kim, S. Y. Chang, J. Chang et al., "Scheduling of raw-material unloading from ships at a steelworks," *Production Planning and Control*, vol. 22, no. 4, pp. 389–402, 2011.

[33] Y. Sun, J. W. Fowler, and D. L. Shunk, "Policies for allocating product lots to customer orders in semiconductor manufacturing supply chains," *Production Planning and Control*, vol. 22, no. 1, pp. 69–80, 2011.

[34] M. Nawaz, E. E. Enscore Jr., and I. Ham, "A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem," *Omega*, vol. 11, no. 1, pp. 91–95, 1983.

[35] D. Ben-Arieh and O. Maimon, "Annealing method for PCB assembly scheduling on two sequential machines," *International Journal of Computer Integrated Manufacturing*, vol. 5, no. 6, pp. 361–367, 1992.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 419042, 9 pages
http://dx.doi.org/10.1155/2015/419042

Research Article

Integrated Optimization of Service-Oriented Train Plan and Schedule on Intercity Rail Network with Varying Demand

Wenliang Zhou, Junli Tian, Lianbo Deng, and Jin Qin

School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China

Correspondence should be addressed to Wenliang Zhou; zwl_0631@csu.edu.cn

Received 8 December 2014; Accepted 21 January 2015

Academic Editor: Govindan Kannan

For a better service level of a train operating plan, we propose an integrated optimization method of train planning and train scheduling, which generally are optimized, respectively. Based on the cost analysis of both passengers travelling and enterprises operation, and the constraint analysis of trains operation, we construct a multiobjective function and build an integrated optimization model with the aim of reducing both passenger travel costs and enterprise operating costs. Then, a solving algorithm is established based on the simulated annealing algorithm. Finally, using as an example the Changzhutan intercity rail network, as an example we analyze the optimized results and the influence of the model parameters on the results.

1. Introduction

A passenger train operating plan (TOP) is not only the basis of train organization and station operation for enterprise, but also the foundation of train choice for passengers travelling by rail transit (e.g., an urban railway system, intercity railway system, and high-speed railway system). A high-quality TOP directly contributes to improving the level of passenger service and boosts enterprise operation efficiency. The TOP generally should arrange origin and destination stations, run routes, intermediate stations, vehicle numbers, and schedule for trains. More broadly, it also determines the crew scheduling and usage plan of the locomotive or electric multiple units (EMUs). However, the latter part of the TOP is not considered in this paper. Due to the complexity and difficulty of solving the TOP of a large-scale rail network, the TOP problem is usually divided into two subproblems: the train planning problem and train scheduling problem, which are solved one by one. Firstly, the train plan is optimized to arrange the origin and destination station, run route, intermediate stations, vehicle number, and frequency of trains with the aim of improving both the passenger travel benefit and enterprise operation profit. And then the train scheduling problem is solved in order to schedule each train's departure and arrival time at each station based on the former optimized train plan.

Most research on the train planning problem so far concentrates on designing an optimization model and its algorithm, aiming at getting a better service-level and high-benefit train plan with constraints of line and station capacity and rail resources (e.g., maximum departure number per day and available vehicles, etc.). Anthony [1] gave a basic frame for solving the passenger train planning problem as early as 1965. Chang et al. [2] proposed a multiobjective model and its algorithm of a train plan with the aim of reducing both enterprise operating costs and passenger travel cost. Yaghini et al. [3] took into account the passenger direct ratio besides travel costs in optimizing train plan. Wang et al. [4] provided an optimization method for a periodic train plan. Recently, some studies [5–7] combined the passenger train choice problem into the train plan problem and accordingly proposed the bilevel programming method of a train plan based on the leader-following relation between formulating a train plan and passenger train choice. For more examples of train plan optimization, see Schmidt and Schöbel [8], Goossens et al. [9], and Schöbel and Scholl [10].

The train scheduling problem is to generally find an optimal or satisfying train timetable with a given optimization objective, subject to a lot of operational and safety constraints (e.g., arrival and departure headway requirements). A branch-and-bound algorithm, Lagrangian relaxation algorithm, and simulation method are widespread used to solve this problem. Higgins et al. [11] developed a branch-and-bound solution framework and some heuristic techniques to find feasible train timetables, and Zhou and Zhong [12] further incorporated some effective rules into the branch-and-bound algorithm for improving its solving efficiency. Brännlund et al. [13] proposed a Lagrangian relaxation approach to find a profit-maximizing train timetable. Dorfman and Medanic [14] proposed an effective simulation approach called TAS to solve the large-scale and real-world train scheduling problem, and Li et al. [15] and Xu et al. [16] further improved TAS by introducing some modified rules and efficient strategies inserted into it. For more studies of train scheduling, refer to Jong et al. [17], Sahana et al. [18], Yalçinkaya and Mirac Bayhan [19], and Zhou et al. [20].

Obviously, optimizing a train plan and train schedule successively has some drawbacks in enhancing the passenger service level and satisfying varying travel demands of intercity rail. First, with the lack of time information, when optimizing a train plan, it is impossible to describe in detail passenger transfer time, wait time, and in-vehicle time determined exactly by a train timetable. Thus, improving passenger travel time is beyond the train plan problem to some extent. And the optimization of a train timetable generally aims to minimize the total travel time of trains, but not of passengers, because it has no passenger volume information about the train. Moreover, this two-stage method cannot make trains' time distribution fit passenger demand distribution better in one day. To overcome the drawbacks thoroughly, combining the train plan and train schedule as a whole, that is, TOP, an integrated optimization of them is an effective alternative. Compared with the two-stage approach, the integrated optimization method has the following differences.

(1) It is to optimize train plan and train schedule simultaneously based on a rail network and its passengers demand distributions, while the two-stage method is firstly to determine a train plan which is taken as one input when scheduling trains latter. Thus, the integrated method has the decision variables and constraints of both train planning and train scheduling.

(2) Although reducing passengers travel costs and enterprise operating costs is taken as the objective in both two methods, their calculation is based on a train schedule in the integrated method while that is only based on a train plan in the two-stage method.

It should be noted that the efficiency of this integrated optimization is not a knotty obstacle for an intercity rail network with a relatively small scale owing to the improvement of computer speed and the development of modern optimization algorithm.

The main contributions of this paper are as follows.

(1) An integrated optimization model of train planning and scheduling is built to minimize both passenger travel costs and enterprise operating costs. It can more exactly and fully describe passenger travel costs.

(2) A solving algorithm based on simulated annealing algorithms (SA) is designed to solve the proposed optimization model.

The remainder of this paper is organized as follows. In Section 2, we describe the problem of TOP optimization and analyze passenger travel costs and enterprise operating costs. In Section 3, we discuss the constraints and multiobjective function and present the integrated optimization model of TOP. In Section 4, we design a solving algorithm based on SA. Moreover, the case of the Changzhutan intercity rail network is used to illustrate the application of the proposed model and algorithm and also to analyze the impact of their parameters on passenger travel costs and enterprise operating costs in Section 5. Finally, the conclusion and further study are given in Section 6.

2. Problem Description

An intercity rail network (S, E) is represented by a set of stations $S = \{1, 2, \ldots, K\}$ and a set of double-track sections $E = \{e(k, k') \mid k, k' \in S\}$ in which $e(k, k')$ and $e(k', k)$ show, respectively, the down and up direction sections connecting station k and k'. The mileage of $e(k, k')$ is denoted by $l(k, k')$ or by $l(e)$.

Intercity rail passenger flow has the obvious characteristic of fluctuating with the time of day, and it has peak hours and low hours of travel. So, it is called varying demand in this paper. The varying demand from origin r to destination s in one day is denoted by a function of time t denoted by $q_{rs}(t)$.

For simplification, the following assumptions are made based on the actual condition of intercity railway in this paper.

(A1) The research range is limited to an independent intercity rail network and passenger total demand of one day among stations is not affected by travel costs determined by the TOP.

(A2) The intercity rail network provides only one speed type (e.g., 200 km/h) of train servicing passengers, and all vehicles have the same capacity for passengers.

(A3) The network capacity is enough to satisfy passengers travelling by the mileage-shortest route; thus, all passengers can travel with those routes.

(A4) Passengers get on the train according to their arriving order.

The TOP can be expressed as a set of trains Ω, and each train is made up by route, vehicle number, and schedule. The route of train i is denoted by S_i, which is composed of a set of stations or a set of sections, the vehicle number of train i is expressed by b_i, and the sequences of departure time and arrival time arranged by ascending order are denoted by X_i, Y_i, respectively. Meanwhile, the arrival time and departure time of train i at station k are denoted by y_{ik}, x_{ik}, respectively.

2.1. Analysis of Passenger Travel Costs. Passenger travel costs mainly consist of wait time at the origin station, transfer time including necessary walking time and wait time during the process from getting off the train to getting on board of another train at the transfer station, in-vehicle time, and fare spending. Considering the additional inconvenience produced by transfer, an additional cost is imposed on transfer passengers besides transfer time. This additional cost contributes to avoiding transfer for passengers when they have other nontransfer paths for travelling. Under the assumption (A3), passenger fare spending calculated by travel mileage multiplying price rate per mileage is a constant and is not considered in this paper.

Wait time at the origin station depends on passengers' arriving time and boarding time. When passengers arrive at station r at time t and wait there until boarding train i at time x_{ir}, their wait time $c_1(r, i)$ can be calculated by

$$c_1(r, i) = x_{ir} - t. \tag{1}$$

When passengers transfer in station k with train i and transfer out with train j, their transfer time $c_2(i, j, k)$ can be determined as follows according to departure time x_{jk} of train j and arrival time y_{ik} of train i:

$$c_2(i, j, k) = x_{jk} - y_{ik}. \tag{2}$$

Moreover, their additional cost of transfer $c_3(i, j, k)$ can be given as ρ multiple of their transfer time; namely,

$$c_3(i, j, k) = \rho c_2(i, j, k). \tag{3}$$

In-vehicle time comprises train operation time and dwell time of each intermediate station. When passengers travel with train i from station k to station k', their in-vehicle time $c_4(i, k, k')$ spent on this train is

$$c_4(i, k, k') = y_{ik'} - x_{ik}. \tag{4}$$

Passenger travel cost is the total of wait time, transfer time and transfer additional cost, and in-vehicle time. For passengers travelling by path p from station r to station s, their travel cost $C(r, s, p)$ is

$$\begin{aligned} C(r, s, p) = c_1(r, i_p^0) &+ \sum_{(i,j,k)\in p} (c_2(i, j, k) + c_3(i, j, k)) \\ &+ \sum_{(i,k,k')\in p} c_4(i, k, k'), \end{aligned} \tag{5}$$

where p is the travel path of passengers, i_p^0 is the first train for passengers of path p travelling, and $(i, j, k) \in p$ shows passengers of path p needing to transfer from train i to train j at station k. And $(i, k, k') \in p$ means that passengers of path p have to travel by train i when going from station k to station k'.

2.2. Analysis of Enterprise Operating Costs. With the action of assumptions (A1) and (A3), an intercity rail enterprise has a fixed ticket income, the product of passenger flow, and its corresponding fare. Thus, the operating costs are considered only in this paper. Operating cost is the sum of the following three components: that is, train organization cost F_O, rail line cost F_L, and rail vehicle cost F_V. It is represented as

$$F = F_O + F_L + F_V. \tag{6}$$

Train organization cost is the fee spent mainly on the train crew and the organizing operation at the train's origin station. It is the product of train number N and the organization cost ω per train; namely,

$$F_O = N\omega. \tag{7}$$

Rail line cost is generated for line maintenance and is directly related to the total travel mileage of a train. It can be expressed as

$$F_L = \psi \sum_{i\in\Omega} l_i, \tag{8}$$

where ψ is the maintenance cost per kilometer line and l_i is the travel mileage of train i.

Vehicle cost is used for vehicle maintenance. It can be calculated as

$$F_V = \sum_{i\in\Omega} (\chi_0 + \chi l_i) b_i, \tag{9}$$

where χ_0 is the fixed cost for each vehicle maintenance and χ is the average maintenance cost of vehicle per mileage.

3. Optimization Model

3.1. Analysis of Constraints. Train origin and destination must be a technical station that has the areas and facilities for a train's technical operation and servicing work. The set of technical stations on an intercity rail network is denoted by S_t, and then the origin station o_i and destination station d_i of train i must be included in set S_t; namely,

$$o_i, d_i \in S_t \quad \forall i. \tag{10}$$

The vehicle number of a train should be set for an upper bound limited by the length of station track. The train vehicle number of upper bound for all travel routes is expressed as $\bar{b}$. That is,

$$b_i \le \bar{b} \quad \forall i. \tag{11}$$

Meanwhile, the vehicle number of a train should not be less than the number that makes this train operate without profit when it reaches its passenger capacity. When train i reaches its passenger capacity, its operating cost F_i and ticket income R_i can be given, respectively, by

$$\begin{aligned} F_i &= \omega + \psi l_i + (\chi_0 + \chi l_i) b_i, \\ R_i &= \phi l_i b_i \varphi, \end{aligned} \tag{12}$$

where ϕ is the fare rate per passenger per kilometer and φ is the passenger capacity of vehicle.

To make train i profitable, its ticket income R_i should be more than the operating cost F_i; that is,

$$R_i > F_i. \tag{13}$$

Based on that, the vehicle number of train i should satisfy another constraint as

$$b_i \geq \left\lceil \frac{\omega + \psi l_i}{\phi l_i \varphi - \chi_0 - \chi l_i} \right\rceil \quad \forall i, \tag{14}$$

where $\lceil\rceil$ is the symbol of rounding up.

The train schedule should meet the constraint of operating time period from time t_s to time t_e. That is,

$$\begin{aligned} t_s &\leq x_{ik} \leq t_e \quad \forall i,\ k \in G_i, \\ t_s &\leq y_{ik} \leq t_e \quad \forall i,\ k \in G_i. \end{aligned} \tag{15}$$

Two same-direction trains departing from or arriving at the same station should satisfy the minimum safety time interval; namely,

$$\begin{aligned} x_{jk} - x_{ik} &\geq \tau_f \quad \forall i \neq j;\ k \in G_i \cap G_j, \\ y_{jk} - y_{ik} &\geq \tau_d \quad \forall i \neq j;\ k \in G_i \cap G_j, \end{aligned} \tag{16}$$

where τ_f, τ_d are separately the minimum safety time interval between departure operations and between arrival operations.

In addition, a train's departure and arrival time in section should meet the constraint of minimum total run time. The technical speed of the train is denoted by v, and train additional times for starting and stopping in section e are expressed by τ'_e, τ''_e, respectively. That is,

$$y_{jk'} - x_{ik} \geq \delta_i^k \tau'_e + \frac{l(e)}{v} + \delta_i^{k'} \tau''_e \quad \forall i,\ e\left(k, k'\right) \in G_i, \tag{17}$$

where δ_i^k is the symbol of describing whether train i should stop at station k or not. If train i stops at station k, then $\delta_i^k = 1$; otherwise, $\delta_i^k = 0$.

Meanwhile, a train's arrival and departure time at the station should satisfy the constraint of minimum dwell time related to the volume of passengers getting on and getting off train. That is,

$$x_{jk} - y_{ik} \geq \Gamma_{ik} \quad \forall i,\ k \in G_i, \tag{18}$$

where Γ_{ik} is the minimum dwell time of train i at station k for ensuring that passengers get on and off safely. It can be given by

$$\Gamma_{ik}\left(q_{ik}\right) = \begin{cases} 0, & q_{ik} = 0, \\ 1, & 0 < q_{ik} \leq \overline{q}, \\ 1 + \left\lceil \dfrac{q_{ik} - \overline{q}}{\xi} \right\rceil, & q_{ik} > \overline{q}, \end{cases} \tag{19}$$

where q_{ik} is the volume of passengers getting on and getting off train i at station k, $\overline{q}$ is the maximum number of passengers for getting on and off the train in one minute, and ξ is the parameter affecting the increase of train dwell time.

3.2. Objective Function and Optimization Model. The cost minimization of intercity rail transit system, that is, minimizing both enterprise operating costs and passenger travel costs, is mostly used as the optimization objective of the train plan in many studies [2, 5–7]. In this paper, it is also adopted as the optimization objective of the TOP, but passenger travel costs including not only in-vehicle time, but also wait time and transfer time, are more full-scale and are calculated more exactly.

The objective function is expressed as the weighted sum of an enterprise's operating costs and passengers' travel costs. That is,

$$\begin{aligned} \min Z &= \alpha F + (1 - \alpha) C \\ &= \alpha F + (1 - \alpha) \lambda \sum_r \sum_s \sum_{p \in P_{rs}} q(r, s, p) C(r, s, p) \\ &= \alpha \left(N\omega + \psi \sum_{i \in \Omega} l_i + \sum_{i \in \Omega} \left(\chi_0 + \chi l_i\right) b_i \right) \\ &\quad + (1 - \alpha) \lambda \\ &\quad \cdot \sum_r \sum_s \sum_{p \in P_{rs}} q(r, s, p) \\ &\quad \cdot \left(c_1\left(r, i_p^0\right) + \sum_{(i,j,k) \in p} \left(c_2(i, j, k) + c_3(i, j, k)\right) + \sum_{(i,k,k') \in p} c_4\left(i, k, k'\right) \right), \end{aligned} \tag{20}$$

where α is the weight parameter balancing the enterprise's operating costs and passengers' travel costs, λ is the average time value of passengers, and $q(r, s, p)$ is the volume of passengers of path p from origin r to destination s.

Based on the above analysis, with the decision variables of train set Ω, the optimization model (M1) of the TOP consists of the objective function (20) and all constraints (10), (11), and (14) through (18). It should be noted that model (M1) has to determine not only each train's route, vehicle number, and schedule, but also train number.

4. Optimization Algorithm Based on SA

4.1. Algorithm for Passenger Train Choice and Calculation of Passenger Travel Costs. All passengers have to obey the rule of time-space priority when choosing a train. In other words, passengers arriving at a station earlier have the priority of boarding the train, but they also have to yield to those on the train as to the limit of train capacity. For that, passengers are distributed to trains according to the ascending order of train departure and arrival time treated as the decision-making time.

At the decision-making time of a train departing, which passengers waiting in the station will choose this train and how many of them can get on it should be determined. Passengers waiting in the station can be divided into two parts, original departing and transferring passengers. The sets of original departing passengers and transferring passengers at station k are denoted by G_k and H_k, respectively. For original departing passengers $g \in G_k$, their earliest arriving time is t_g and destination is s_g. And for transferring passengers $h \in H_k$, their transferring in time is t_h, destination is s_h, and their number is q_h.

When train i departs from station k, passengers whose cost-shortest path from station k to their destination contains train i need to get on it, but the number of those who can get on board successfully depends on the empty seat number u_i of train i. Based first-arriving-first-boarding principle, the number of passengers getting on the train is given as

$$\overline{q}_g = \begin{cases} \int_{t_g}^{t^*} q_{ks_g}(t)\,dt & t_g \le t^* \\ 0 & t_g > t^* \end{cases} \quad g \in \overline{G}_k \in G_k,$$
$$\overline{q}_h = \begin{cases} q_h & t_h \le t^* \\ 0 & t_h > t^* \end{cases} \quad h \in \overline{H}_k \in H_k, \tag{21}$$

where $\overline{G}_k, \overline{H}_k$ are the sets of original departing passengers and transferring passengers needing to get on train i at station k and t^* is the time boundary deciding what time passengers arriving can get on board of the train. It means that passengers arriving before t^* can get on the train, but those arriving after this time cannot get on board, because there are no empty seats left. When the passenger total number of $\overline{G}_k, \overline{H}_k$ is less than u_i, then $t^* = x_{ik}$. Otherwise, the value of t^* can be calculated by solving the following equality:

$$\sum_{h\in\overline{H}_k | t_h \le t^*} q_h + \sum_{g\in\overline{G}_k | t_g \le t^*} \int_{t_g}^{t^*} q_{ks_g}(t)\,dt = u_i. \tag{22}$$

The wait time, transfer time, and additional cost for transfer at station k of passengers getting on train i can be calculated by

$$\begin{aligned} c_1(k,i) &= \sum_{g\in\overline{G}_k} c_1^g(k,i) \\ &= \sum_{g\in\overline{G}_k} \max\left\{\int_{t_g}^{t^*} q_{ks_g}(t)(x_{ik}-t)\,dt, 0\right\}, \\ c_2(k,i) &= \sum_{h\in\overline{H}_k} c_2^h = \sum_{h\in\overline{H}_k} (x_{ik}-t_h)\overline{q}_h, \\ c_3(k,i) &= \rho c_2(k,i). \end{aligned} \tag{23}$$

At the decision-making time of train i arriving station k, passengers having arrived at their destination or whose cost-shortest path from station k to their destination does not include train i again have to get off. The set of passengers arriving at station k with train i is denoted by B_i, with the subset of those getting off the train being denoted by $\overline{B}_i$. For passengers $b \in B_i$, their destination is s_b, and their number is q_b. The in-vehicle time of passengers $b \in B_i$ from the rear station k' to station k can be calculated by

$$c_4^b(i,k',k) = \begin{cases} y_{ik} - x_{ik'}, & b \in \overline{B}_i, \\ x_{ik} - x_{ik'}, & b \in B_i - \overline{B}_i. \end{cases} \tag{24}$$

And the total of their in-vehicle time is given as

$$c_4(i,k',k) = \sum_{b\in B_i} q_b c_4^b(i,k',k). \tag{25}$$

Based on the above analysis, Algorithm 1 for passenger train choice and calculation of passenger travel costs is described as follows.

Algorithm 1. Consider the following.

Step 1 (initialization). Set $B_i = \emptyset$ and $u_i = b_i\varphi$ of each train and $H_k = \emptyset$ of each station. Find all original departing passengers G_k of each station and let $C = 0$ as the total travel costs of passengers.

Step 2 (find the earliest decision-making time t). If t corresponds to departure time x_{ik}, then go to Step 2.1; otherwise, if t corresponds to arrival time y_{ik}, then go to Step 2.2.

Step 2.1. Determine boarding passengers $\overline{G}_k$, $\overline{H}_k$ and their number $\overline{q}_g$ ($g \in \overline{G}_k$), $\overline{q}_h$ ($h \in \overline{H}_k$). Then, calculate their wait time $c_1(k,i)$, transfer time $c_2(k,i)$, and transfer additional cost $c_3(k,i)$. Set $C = C + c_1(k,i) + c_2(k,i) + c_3(k,i)$ and update empty seats number u_i, transferring passengers H_k, original departing passengers G_k, and train passengers B_i. Go to Step 3.

Step 2.2. Determine getting-off passengers $\overline{B}_i$ and calculate their in-vehicle time $c_4(i,k',k)$. Then, set $C = C + c_4(i,k',k)$ and update empty seats number u_i, transferring passengers H_k and train passengers B_i.

Step 3 (judge whether there are other decision-making times or not). If yes, then return to Step 2. Otherwise, C is the passengers' total travel costs, and terminate this algorithm.

4.2. The General Algorithm for Optimizing TOP

4.2.1. Generation of an Initial Solution of TOP. Trains of the initial solution are created one by one based on the varying demand on the network. A new train is organized with departing time t when the product of its boarding passengers' number $\overline{Q}$ and their average wait time $\mathscr{R}$ including wait time and transfer wait time at a technical station satisfies

$$Q\mathscr{R} = M \tag{26}$$

and its vehicle number is determined by

$$b = \frac{Q}{\varphi\ell}, \tag{27}$$

where M is the control parameter for organizing one new train and ℓ is the average usage rate of train capacity.

From formula (26), we know that a train should be organized either when there are enough passengers waiting for boarding, or when some passengers have waited for too long.

The newly created train is assumed to stop at all the passed stations during the process of generating the initial solution. When it arrives at the next technical station, if the sum of passengers waiting to get on and those on the train is more than ς percent of its capacity, it moves forward along the direction with the largest value of ς. Otherwise, it stops here as its destination.

4.2.2. Generation of a Neighbor Solution of TOP. A new solution is generated by changing the train's route, stop stations, vehicle number, and starting time of the current solution with the probability method. As for the train route, it is adjusted by adding some new sections to its front and end or removing partial sections depending on train's operating costs and passenger volume of them. Two Boolean variables ϱ and ϖ both created by Bernoulli distribution are, respectively, used to indicate whether partial sections should be added to the train route and removed from it. If $\varrho = 1$, the corresponding sections are added to the train route, and when $\varpi = 1$, the corresponding sections are removed from it. As for the train sections μ between two technical stations, the probability of $\varrho = 1$ is given by

$$\varkappa = 1 - e^{-W_\mu C_u / F_\mu T}, \tag{28}$$

where W_μ, C_u, and F_μ are the passenger-kilometer, passengers' average travel costs, and operating cost in train section μ respectively, and T is the current temperature.

For new sections μ, the probability of $\varpi = 1$ is given by

$$\varkappa = 1 - e^{-Q_u / \theta_u T}, \tag{29}$$

where Q_u is the number of passengers transferring from or transferring to the current train and θ_u is their average transfer cost.

The alteration of train stop stations is also based on a Bernoulli distribution. For station k of train i, the probability of a train's stop is given by

$$\varkappa_{ik} = e^{-Q_{ik} C_{ik} T}, \tag{30}$$

where Q_{ik} is the number of passengers getting on and off the train and C_{ik} is the average travel cost of passengers getting on the train at station k.

The modification of both the train vehicle number and starting time is given as follows:

$$d^* = \begin{cases} d + \eta (B - d) & \zeta \geq 0 \\ d + \eta (d - A) & \zeta < 0, \end{cases} \tag{31}$$

$$\zeta = T \left[\left(1 + \frac{1}{T} \right)^{|2\xi - 1|} - 1 \right] \operatorname{sgn} (\xi - 0.5), \tag{32}$$

where $d^*, d \in [A, B]$ are the value of current solution and neighboring solution, respectively, and ξ is generated by the next probability density function, which makes the vehicle number and starting time of one train with low benefit or efficiency have a high adjustment chance. Consider

$$y = (\gamma - 1)(x - 0.5) + 1, \quad x \in [0, 1], \tag{33}$$

where γ is determined by the indexes of train i. For train vehicle number and starting time, it is given, respectively, by

$$\begin{gathered} \gamma = 1 - \frac{W_i C_i / F_i}{\operatorname{avg}(WC/F)}, \\ \gamma = \frac{C_i / Q_i}{\operatorname{avg}(C/Q)}, \end{gathered} \tag{34}$$

where W_i is the passenger kilometer, F_i is the average operating cost per vehicle, C_i is the total wait time and transfer cost, and Q_i is the number of passengers on the train.

In formulas (28), (29), (30), and (32), the calculation of their probabilities is mainly based on train's service level, passenger volume, operating costs, and the current temperature as a parameter of SA, and the higher the current temperature is, the larger their probabilities are.

With the above generation method of an initial solution and a neighborhood solution, the general Algorithm 2 based on SA for optimizing TOP is described as follows.

Algorithm 2. Consider the following.

Step 1 (initialization). Generate the initial feasible solution Ω under the initial temperature T_0 and then calculate the objective value $Z(\Omega)$ based on simulating passenger train choice and calculating passenger travel costs with Algorithm 1. Set $\beta = 0$ as the current running times of the outer cycle. Let $n = 0$ be the current running times of the inner cycle and let $T = T_0$ be the current temperature. Set $T_{\min}$ as the minimum temperature of the outer cycle and Υ as the number of iterations at each temperature.

Step 2 (construction of neighborhood). Generate a new solution Ω' and calculate its objective value corresponding to $Z(\Omega')$ based on simulating passenger train choice and calculating passenger travel costs with Algorithm 1.

Step 3 (metropolis sampling). When $Z(\Omega) > Z(\Omega')$, then set $\Omega = \Omega'$; otherwise, if $\exp(\Delta Z/T) >$ rand (rand is a random number in $(0, 1)$ and ΔZ is the difference between them, the current and optimal solution), then let $\Omega = \Omega'$. Then, set $n = n + 1$.

Step 4 (test of the termination criterion of the inner cycle). If $n = \Upsilon$, terminate the inner cycle and let $\beta = \beta + 1$; otherwise, return to Step 2.

Step 5 (cooling schedule). Calculate the temperature $T(\beta)$.

Step 6 (test of the termination criterion of the outer cycle). When $T(k) \leq T_{\min}$, terminate this algorithm and output the optimal solution; otherwise, return to Step 2.

Table 1: Parameter values of model (M1).

Symbol	Value	Unit	Symbol	Value	Unit
ρ	2.0	—	χ_0	4200	¥/vehicle
ω	25000	¥/train	χ	6.5	¥/vehicle·km
ψ	31.5	¥/train·km	$\overline{b}$	10	Vehicle
ϕ	2.5	¥/passenger·km	φ	80	Passenger/vehicle
t_s	6:30	—	t_e	22:00	—
τ_f	6	Min	τ_d	8	Min
ν	200	Km/h	τ_e'	1	Min
τ_e''	1	Min	$\overline{q}$	200	Passenger
ξ	150	Passenger/min	λ	80	¥/h

Table 2: Parameter values of algorithm.

Symbol	Value	Unit	Symbol	Value	Unit
M	15500	Passenger·min	ℓ	0.5	—
ς	0.4	—	T_0	1000	—
$T_{\min}$	1	—	Υ	50	—

5. Numerical Studies in Changzhutan Intercity Rail Network

The Changzhutan intercity rail network in the cluster including the cities of Changsha, Zhuzhou, and Xiangtan of China is planned to be completed in 2016. It consists of 21 stations and has the total length of 96 km. The above algorithm is developed with computer language C# on the platform of Microsoft Visual Studio.net and runs on the computer with the system of Microsoft Windows XP (Home Edition), RAM configuration of Pentium(R) Dual-Core CPU E5800, 3.19 GHz, 2.96 GB. The values of parameters in model (M1) and its solving algorithm are given in Tables 1 and 2, respectively.

Firstly, some observations on the convergence process of the algorithm with the value of α being 0.2, 0.5, and 0.8, respectively, are made. The change relations between the best objective values with the total computing times of algorithm running are shown in Figure 1. As seen from it, the objective values decline sharply with the computing time in the first 10 minutes or so for both three instances and then drop slowly until about 17 minutes. After that, they became stable, which indicates that the algorithm has converged to a better solution.

Table 3 shows the optimization results with the value of α being 0.4, 0.6, and 0.8, respectively. From these results, passenger average wait time and each operating cost vary sharply with a different value of α, but the differences of average transfer cost, proportion of transfer passengers, and passenger in-vehicle speed are smaller. This is because the number of operating trains rising with the increase of α mainly determines the enterprise operating cost, and the higher the trains' departure frequency is, the shorter the wait time varying passengers have. But trains can have a high travel speed, and their arrival and departure time can connect well, no matter how many trains there are.

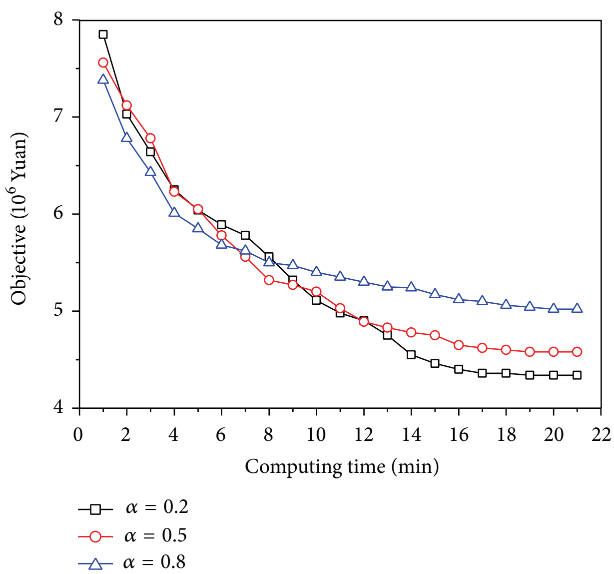


Figure 1: Convergence of the solution with different values of weight para α.

For different values of α, the percentage distributions of passenger wait time are shown in Figure 2. As we can see, regardless of $\alpha = 0.4$, $\alpha = 0.6$, or $\alpha = 0.8$, their passenger percentage distributions are similar to a normal distribution. But their wait time with the maximum percentage increases from 10.2 min to 11.8 min and then to 18.3 min with the increase of α. The wait time of 75% of the passengers is mainly concentrated in 0 to 16 min both when $\alpha = 0.4$ and $\alpha = 0.6$, and that of 80% of the passengers is located in 0 to 20 min while $\alpha = 0.8$. The maximum wait time of these three cases is 30 min, which is the ultimate value passengers can bear.

Table 3: Optimization results with different value of α.

The value of weight para α	0.4	0.6	0.8
Average wait time per passenger (min)	13.6	14.1	22.2
Average transfer time per passenger (min)	19.6	20.2	21.5
Average transfer additional cost per passenger (min)	39.2	40.4	43.0
The proportion of transfer passenger (%)	11.3	12.5	12.2
Passenger in-vehicle speed (km/h)	172.4	171.6	172.8
Train organization cost (¥)	3550000	3175000	2725000
Rail line cost (¥)	237069	231030	206010
Rail vehicle cost (¥)	458272	435136	334520

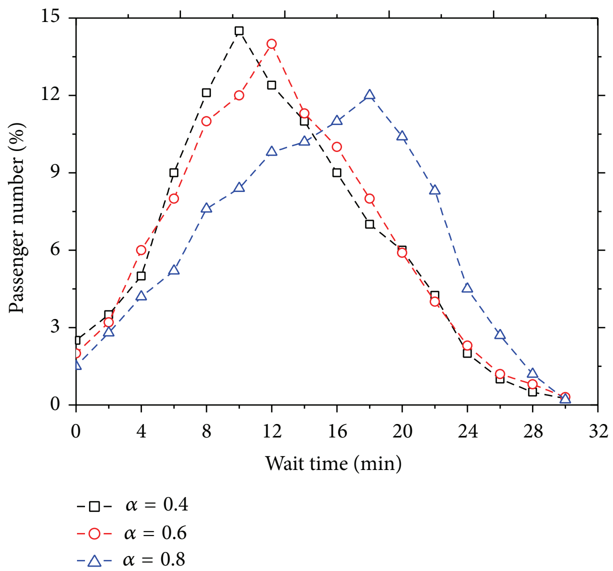


Figure 2: Distribution of passenger wait time.

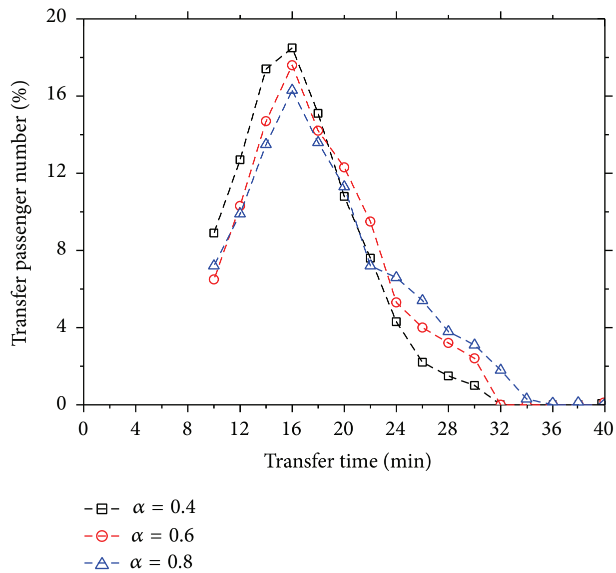


Figure 3: Distribution of passenger transfer time.

The percentage distributions of passenger transfer time with a different value of α are shown in Figure 3. As passenger average walking time for each transfer is assumed to be 10 min, passenger minimum transfer times when $\alpha = 0.4$, $\alpha = 0.6$, and $\alpha = 0.8$ are all 10 min. As seen in Figure 3, the transfer time with the maximum percentage of about 18% does not vary with the different value of α, and it is 16 min or so, corresponding to a passenger transfer wait time of 6 min, in all three cases. Moreover, the transfer time of 90% of the transfer passengers is mainly concentrated in 10 to 24 min. Through a comprehensive comparison of the transfer time and the transfer passenger number of three cases, it can be found that the average transfer time and total number of transfer passenger with $\alpha = 0.4$ are slightly less than these with $\alpha = 0.8$, but their differences are very small, which indicates that the factor α has a little effect on passenger service level of transfer.

For determining the influence of weight parameter α, the objective values composed of enterprise operating cost and passenger travel cost are calculated with different values of α, and the change in these two partial costs for various α is shown in Figure 4. As we can see, operating cost decreases rapidly when α increases from 0.1 to 0.3, and later it has a relative slow-down speed as α continues to increase. However, travel time increases smoothly with α increasing from 0.1 to 0.9. A balance with the minimum of their total can be made between these two parts when $\alpha = 0.7$ is taken as a reasonable value.

6. Conclusion and Further Study

In this paper, for the integrated optimization of train planning and train scheduling, based on analyzing passenger travel costs and enterprise operating costs, we present their integrated optimization model aiming to minimize both passenger and enterprise costs with the constraints of trains operating and build a solution algorithm based on SA algorithm. From the analysis of the optimization results for the Changzhutan intercity rail network, the proposed model and algorithm can effectively obtain a satisfactory TOP, and a solution with the total minimum of operating costs and travel costs can be reached when the value of weight parameter α is about 0.7.

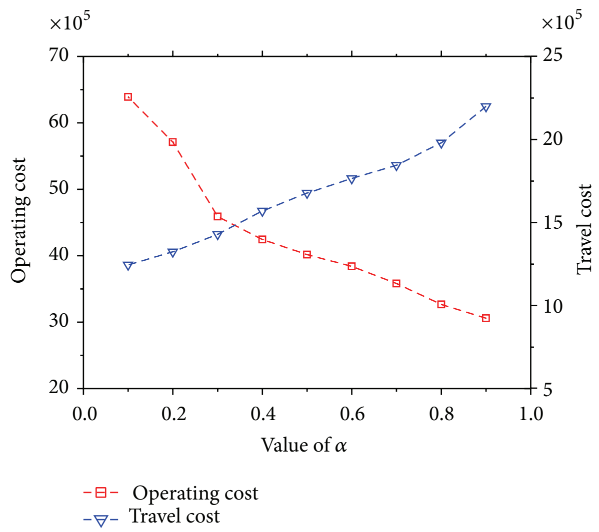


Figure 4: Relationship between objective function values and weight parameter α.

As passenger demand of intercity rail largely depends on their service level under the competitive environment between railway and highway, one further research area is to optimize TOP considering this effect. Another one is to study it involving the allocation of vehicles to train, which can determine more exactly the train operating costs.

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 Natural Science Foundation of China (Grant nos. 71401182 and 71471179), Doctoral Scientific Foundation of the Ministry of Education of China (Grant no. 20120162120042), and Natural Science Foundation of Hunan Province (Grant no. 14JJ3030).

References

[1] R. N. Anthony, *Planning and Control Systems: A Framework for Analysis*, Harvard University, Boston, Mass, USA, 1965.

[2] Y. H. Chang, C. H. Yeh, and C. C. Shen, "A multiobjective model for passenger train services planning: application to Taiwan's high-speed rail line," *Transportation Research Part B: Methodological*, vol. 34, no. 2, pp. 91–106, 2000.

[3] M. Yaghini, M. Momeni, and M. Sarmadi, "An improved local branching approach for train formation planning," *Applied Mathematical Modelling*, vol. 37, no. 4, pp. 2300–2307, 2013.

[4] B. Wang, H. Yang, and Z. H. Zhang, "Research on train plan of Jingjin intercity railway based on cycle timetable," *Journal of the China Railway Society*, vol. 29, no. 2, pp. 8–13, 2007.

[5] F. Shi, L. B. Deng, and L. Huo, "Bi-level programming model and algorithm of passenger train operation plan," *China Railway Science*, vol. 28, no. 3, pp. 110–116, 2007.

[6] R. Borndörfer, M. Grötschel, and M. E. Pfetsch, "A column-generation approach to line planning in public transport," *Transportation Science*, vol. 41, no. 1, pp. 123–132, 2007.

[7] B. H. Park, Y.-I. Seo, S.-P. Hong, and H.-L. Rho, "Column generation approach to line planning with various halting patterns—application to the Korean high-speed railway," *Asia-Pacific Journal of Operational Research*, vol. 30, no. 4, Article ID 1350006, pp. 134–151, 2013.

[8] M. Schmidt and A. Schöbel, "The complexity of integrating routing decisions in public transportation models," in *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS '10)*, vol. 14, pp. 156–169, September 2010.

[9] J.-W. Goossens, S. van Hoesel, and L. Kroon, "A branch-and-cut approach for solving railway line-planning problems," *Transportation Science*, vol. 38, no. 3, pp. 379–393, 2004.

[10] A. Schöbel and S. Scholl, "Line planning with minimal traveling time," in *Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS '05)*, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, 2005.

[11] A. Higgins, E. Kozan, and L. Ferreira, "Optimal scheduling of trains on a single line track," *Transportation Research Part B: Methodological*, vol. 30, no. 2, pp. 147–161, 1996.

[12] X. Zhou and M. Zhong, "Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds," *Transportation Research Part B: Methodological*, vol. 41, no. 3, pp. 320–341, 2007.

[13] U. Brännlund, P. O. Lindberg, A. Nõu, and J.-E. Nilsson, "Railway timetabling using Lagrangian relaxation," *Transportation Science*, vol. 32, no. 4, pp. 358–369, 1998.

[14] M. J. Dorfman and J. Medanic, "Scheduling trains on a railway network using a discrete event model of railway traffic," *Transportation Research Part B: Methodological*, vol. 38, no. 1, pp. 81–98, 2004.

[15] F. Li, Z. Y. Gao, K. P. Li, and L. X. Yang, "Efficient scheduling of railway traffic based on global information of train," *Transportation Research Part B: Methodological*, vol. 42, no. 10, pp. 1008–1030, 2008.

[16] X. Xu, K. Li, L. Yang, and J. Ye, "Balanced train timetabling on a single-line railway with optimized velocity," *Applied Mathematical Modelling*, vol. 38, no. 3, pp. 894–909, 2014.

[17] J. C. Jong, S. Chang, and Y. C. R. Lai, "Development of a two-stage hybrid method for solving high speed rail train scheduling problem," *Annals of Operations Research*, vol. 42, no. 8, pp. 212–226, 2012.

[18] S. K. Sahana, A. Jain, and P. K. Mahanti, "Ant colony optimization for train scheduling: an analysis," *International Journal of Intelligent Systems and Applications*, vol. 6, no. 2, pp. 321–342, 2014.

[19] Ö. Yalçinkaya and G. Mirac Bayhan, "A feasible timetable generator simulation modelling framework for train scheduling problem," *Simulation Modelling Practice and Theory*, vol. 20, no. 1, pp. 124–141, 2012.

[20] W. Zhou, L. Deng, M. Xie, and X. Yang, "Coordination optimization of the first and last trains' departure time on urban rail transit network," *Advances in Mechanical Engineering*, vol. 2013, Article ID 848292, 12 pages, 2013.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 638104, 10 pages
http://dx.doi.org/10.1155/2015/638104

Research Article

A Column Generation Based Hyper-Heuristic to the Bus Driver Scheduling Problem

Hong Li,[1] Ying Wang,[2,3] Shi Li,[4] and Sujian Li[1]

[1] *School of Mechanical Engineering, University of Science and Technology, Beijing 100083, China*
[2] *State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*
[3] *School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China*
[4] *Laboratoire Systémes et Transports, Université de Technologie de Belfort-Montbéliard, 90000 Belfort, France*

Correspondence should be addressed to Ying Wang; wangy@bjtu.edu.cn

Received 2 January 2015; Accepted 9 February 2015

Academic Editor: Chin-Chia Wu

Public transit providers are facing continuous pressure to improve service quality and reduce operating costs. Bus driver scheduling is among the most studied problems in this area. Based on this, flexible and powerful optimization algorithms have thus been developed and used for many years to help them with this challenge. Particularly, real-life large and complex problem instances often need new approaches to overcome the computational difficulties in solving them. Thus, we propose a column generation based hyper-heuristic for finding near-optimal solutions. Our approach takes advantages of the benefits offered by heuristic method since the column selection mode is driven by a hyper-heuristic using various strategies for the column generation subproblem. The performance of the proposed algorithm is compared with the approaches in the literature. Computational results on real-life instances are presented and discussed.

1. Introduction

The bus driver scheduling problem (BDSP) is an extremely complex and important part of the operational planning process of transport companies. Efficient schedules can make significant monetary savings for transportation operators since driver wages are a very large cost element. Furthermore, good crew planning ensures a high job satisfaction.

The process of bus driver scheduling involves partitioning the vehicle work into a set of legal driver shifts that should reflect the operator's definition of efficiency [1]. While the constraints according to labor rules and requirement differ from country to country, even company to company, the evaluation criteria and objectives may differ as well.

A great deal of research efforts on this problem has been made in this area since the 1960s. Many scientific approaches and systems to bus diver scheduling have been reported in a series of workshops [2–6]. Useful review and relevant review to these approaches have been given by [7]. Among these planning and scheduling processes, driver scheduling remains to be a challenging problem, and new approaches have been constantly sought to solve this problem [8].

Column generation is introduced by [9] and has been widely used in the past several years for problems in transportation, scheduling, and combinatorial optimization. The approach requires one to decompose the original problem into two problems, which are termed restricted master problem and subproblem, respectively. In the literature, column generation has been effectively applied to solve crew scheduling and other related problems (see [9–11]), particularly for real-world problems of bus driving scheduling (see [12]). However, it is well-known that one of the drawbacks of column generation is the so-called "tailing-off" effect: a lot of iterations that do not significantly modify the optimal value of the restricted master problem (RMP). Although successfully used, this method suffers from slow convergence that somewhat limits its efficiency and usability [13]. Meanwhile, hyper-heuristic is an emerging methodology in search and optimization [14]. Instead of using exact methods in most column generation, we propose a column generation based

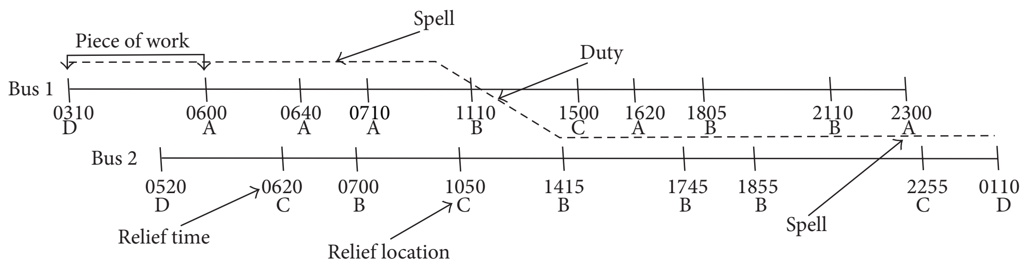


Figure 1: Relationship in vehicle block.

hyper-heuristic method for achieving good quality solutions in short running time.

This paper is organized in the following way. In Section 2, we give a problem definition of bus driver scheduling problem. A discussion of the proposed approach is given in Section 3. In Section 4, comparative results using real-life instance, some of which are large instances, are given. Finally, we conclude in Section 5.

2. The Bus Driver Scheduling Problem

2.1. Terminology. Before scheduling the drivers tasks, the vehicle routes have to be constructed. A *trip* is a movement of a vehicle in a given path. It is the basic unit of service in the sense that each trip must be operated by a single vehicle. A *vehicle block* is a sequence of trips to be done by one vehicle from the time that it leaves the depot until it returns to the depot. From the viewpoint of driver scheduling, drivers can only be relieved at some designated places called *relief points*, which are represented by letter codes (A, B, and C) in Figure 1. The times when the vehicles are at the relief points, marked on the horizontal timescale, are known as *relief opportunities*. The work between two consecutive relief opportunities on the same vehicle is called *piece of work (or task)* for the driver. The work of a driver in a day is known as a *duty (or shift)*. Note that not all relief opportunities will be used to relieve drivers and therefore a driver may be covering a number of consecutive pieces of work, called a *spell*.

An example of a duty built between two vehicle blocks is given in Figure 1. The duty is composed of two spells from the two vehicle blocks.

According to which time period the duties cover, the legal duties can be classified to the following five types.

Early Duty starts early in the morning taking the buses out of garage before the morning peak. The working time of a day that this duty covers is between 05:00 and 12:00.

Late Duty starts in the afternoon and ends in the night. The working time of a day that this duty covers is between 12:00 and 20:00.

Night Duty works in the late evening buses returning the buses to the garage. The working time of a day that this duty covers is between 22:00 and 05:00.

Day Duty starts in the morning and ends in the afternoon. The working time of a day that this duty covers is between 08:00 and 16:00.

Middle Duty works during the period of the morning and the evening peaks. The working time of a day that this duty covers has two parts. One is between 05:00 and 08:00 and the other is between 16:00 and 22:00.

The user could impose certain constraints on any of these duties to ensure that the final schedule does not contain too many or too few duties of particular type. The constrains can be imposed either initially or when reoptimizing an existing solution.

2.2. Mathematical Formulation. The set covering model is the fundamental model of the bus driver scheduling. In the public transport such models have been popular for solving crew scheduling problems for many years [9, 15]. In the formulation adopted here, we will consider our approach based on the set covering formulation of the problem. Although the schedule must ultimately have only one duty assigned to each piece of work, it is better to allow the formulation to include the possibility of overcover and to use set covering rather than set partitioning. On one hand, integer solutions may have some constraints covered by two or more nonzero variables, which in scheduling terms would mean two or more duties assigned to the same piece of work. The reason for this is that there can be a danger that there may be no feasible solution that "fit-together" without causing any over-cover. On the other hand, the produced solutions do not usually have overcovered pieces of work; the set covering model most commonly results in two duties overlapping slightly. If overcover occurs, it can be easily eliminated by editing the schedule afterwards.

Let $N = \{D_1, D_2, D_3, \ldots, D_n\}$ be the set of legal duties and $M = \{1, 2, \ldots, m\}$ be the set of pieces of work to be covered. We can define the problem as the construction of a matrix, where drivers appear in columns and pieces of works in lines. For each element $a_{ij} \in 0, 1$, $i \in M$, $j \in N$, $a_{ij} = 1$ indicates the duty j covers piece of work i, and it is 0 otherwise. It can be formulated as follows:

$$\min W_1 \sum_{j=1}^{n} c_j x_j + W_2 \sum_{j=1}^{n} x_j \tag{1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1, \quad i \in M, \tag{2}$$

$$x_j \in 0, 1, \quad j \in N, \tag{3}$$

where n is the number of duties; m is number of work pieces; c_j is the cost of duty j; x_j is equal to 1 if duty j is used in the solution and 0 otherwise. W_1 and W_2 are weight constants. Generally, the main objective of the transport company is to obtain a solution with a minimum number of duties. For this objective, we consider here $W_2 \gg W_1$.

The constraints (2) indicate that each piece of work must be covered by at least one duty. We consider the costs c_j in our work based on [15]:

$$c_j = \mathrm{OV}_j + \mathrm{IT}_j, \tag{4}$$

where

$$\mathrm{OV}_j = \max\left(0, \left[\mathrm{FT}\left(t_{\mathrm{lst}}\right) - \mathrm{ST}\left(t_{\mathrm{fst}}\right)\right] - \mathrm{NWT}\right), \tag{5}$$

$$\mathrm{IT}_j = \max\left(0, \mathrm{NWT} - \left[\mathrm{FT}\left(t_{\mathrm{lst}}\right) - \mathrm{ST}\left(t_{\mathrm{fst}}\right)\right]\right) + \cdots \sum_{i=1}^{m-1} \max\left(0, \mathrm{ST}\left(t_{i+1}\right) - \mathrm{FT}\left(t_i\right)\right), \tag{6}$$

where c_j is the cost of the driver j; OV_j is the extra-time over the normal time of work established by law for driver j (8 hours); IT_j is the idle time of driver j (time the driver is not working during his shift); NWT is the normal working time (8 hours); $\mathrm{FT}(t_i)$ is the finish time of the task i; $\mathrm{ST}(t_i)$ is the start time of the task i; t_{fst} is the first task of the j driver's shift; t_{lst} is the last task of the j driver's shift.

3. Solution Methodology

The column generation algorithm for the BDSP is initialed by giving a small set of duties (columns). These duties constitute the initial columns in a restricted master problem (RMP). Feedback from the optimal dual values of the RMP is then used to solve a pricing subproblem, which aims to identify duties with a negative reduced cost that improves the objective function value of the RMP. The pricing subproblem for the BDSP is typically modeled as a constrained shortest path problem solved over a directed acyclic graph and then solved using dynamic programming techniques [9]. Nevertheless, because of the drawback that the subproblems for generating the columns would be computationally expensive in real problem instances [16], column generation did not make much progress toward solving large instances. Thus, the main work relies on the subproblem to speed up the whole process of column generation. Here, we propose the column generation based hyper-heuristic methodology to solve the BDSP. This approach provides two significant advantages. First, due to the fact that column generation is known for its poor convergence, it is not necessary to select only one column with lowest reduced cost; in fact, any column with a negative reduced cost will do. Using this observation can improve the overall efficiency. In our research the idea of "rapid adding columns" is to add a set of columns quickly by iteratively solving subproblem using a set of low heuristics. Thus, more than one duty with negative reduced costs may even be possible to be brought into the restricted subset per iteration. Second, the diverse columns with negative reduced cost that are selected (or generated) can be performed efficiently, see [13], to avoid the rapid and useless increase in RMP size. In fact, our approach may select a good "diversity" of duties by using these low heuristics to improve the solution without a large increase in computation time. The pricing step plays an important role in terms of the efficiency of the column generation method.

An outline of the algorithm is given as follows.

Step 0 (preprocessing). Generate all valid duties. Then, these duties form a column pool P.

Step 1 (construct an initial solution). Take a small subset of the duties $p \subset P$ as initial set of columns.

Step 2 (solving the RMP). Solve the LP over the current duty subset and compute the shadow prices of the set of columns p.

Step 3 (duty management). Control the column pool P and remove the columns in the RMP if necessary.

Step 4 (selection of new duties). Select duties with negative reduced cost which will improve the solution, and add them to the RMP.

Step 5 (stopping criterion). If the stopping criterion is met, then go to Step 6; else go to Step 2.

Step 6 (finding integer solutions). Solve ILP and obtain an integer solution.

The overall execution is iterated between Steps 2 and 6. In the first iteration it uses the columns created in Step 0.

In the remainder of this section, we provide details of each of the above steps.

3.1. Step 0: Preprocessing. In our approach we generate all feasible duties explicitly in a preprocessing step. That is, all the possible combinations of pieces of work are valid according to a set of labor contracts and regulations. One of the advantages of generating duties beforehand is that different duty sets can be identified so that experienced schedulers can "share" their own knowledge rather than relying on a "black box" to produce the schedules. Error correction or experimental changes can thus be carried out relatively quickly before searching for resulting schedules [17]. Moreover, preprocessing is an important procedure to express "preferences" and "undesirableness," which are defined by a user-defined strategy. As discussed in [16], soft parameters are used to limit the generation of duties, for example, limiting the maximum number of spells in a duty to four. Meanwhile,

(1) Given a bus schedule, a maximum working time of duty T, and a required number of pieces N_r; Set $N_p = 1$, $j = 1$
(2) **for** all bus vehicles V **do**
(3) Select the relief opportunities to generate a set of piece of work $M = \{M_1, M_2, \ldots, M_m\}$
(4) Find the piece $M_i \in M$ such that the length is shortest (denoted as $t_{\min}$) in M
(5) Compute a maximum number of piece $N_{\max} = T/t_{\min}$
(6) **end for**
(7) **while** $N_p \le \mathrm{Min}(N_r, N_{\max})$ **do**
(8) **for** piece of work $M_{[j][i]} \in M$ **do**
(9) **for** piece of work $M_{[j][i+1]} \in M$ **do**
(10) ...
(11) **for** piece of work $M_{[j][N_p]} \in M$ **do**
(12) **if** Duty D_i which covers $(M_{[j][i]}, M_{[j][i+1]}, \ldots, M_{[j][N_p]})$ satisfies the constrains (9)–(13) **then**
(13) Add generated duty i to set of duties D
(14) N_p++; j++
(15) **end if**
(16) **end for**
(17) **end for**
(18) **end for**
(19) **end while**1111eq:eps

ALGORITHM 1: Generation of all valid potential duties.

a set of preferred duties can be identified relatively quickly; for example, duties with idle time less than average are preferred. Here, this kind of preprocessing could help us to find a subset of "good" duties when we need to use them during the search process.

Formally, we make the following definitions.

Definition 1. Given a duty D_j, let R_i^j denote a relief opportunity which is start point in a piece of work i covered by the duty D_j. For a given relief opportunity R_i^j, we define a relief time and relief point in R_i^j as $\mathrm{ST}(t_i)$ and $\mathrm{AS}(t_i)$, respectively. Then, $R_i^j = (\mathrm{ST}(t_i), \mathrm{AS}(t_i))$.

Definition 2. Given a duty D_j, let L_i^j denote a relief opportunity which is end point in a piece of work i covered by the duty D_j. For a given relief opportunity R_i^j, we define a relief time and relief point in L_i^j as $\mathrm{FT}(t_i)$ and $\mathrm{DS}(t_i)$, respectively. Then, $L_i^j = (\mathrm{FT}(t_i), \mathrm{DS}(t_i))$.

Based on the above definitions, a piece of work i covered by the duty D_j is denoted by

$$M_{ji} = \left(R_i^j, L_i^j\right). \tag{7}$$

Hence, n pieces of work composed of a duty D_j can be defined as

$$D_j = \left(M_{j1}, M_{j2}, \ldots, M_{jn}\right). \tag{8}$$

According to Definitions 1 and 2, each duty is forced to satisfy all the following basic constraints:

$$\left[\mathrm{FT}\left(t_{\mathrm{lst}}\right) - \mathrm{ST}\left(t_{\mathrm{fst}}\right)\right] \le \mathrm{MWT}, \tag{9}$$

$$\sum_{t_i \in D_j} \left(\mathrm{FT}\left(t_i\right) - \mathrm{ST}\left(t_i\right)\right) \le \mathrm{MVT}, \tag{10}$$

$$\mathrm{FT}\left(t_i\right) \le \mathrm{ST}\left(t_{i+1}\right), \tag{11}$$

$$\mathrm{AS}\left(t_i\right) = \mathrm{DS}\left(t_{i+1}\right), \tag{12}$$

$$\begin{gathered} \mathrm{AS}\left(t_i\right) \ne \mathrm{DS}\left(t_{i+1}\right), \\ \mathrm{ST}\left(t_{i+1}\right) - \mathrm{FT}\left(t_i\right) \ge T_{i,i+1}, \end{gathered} \tag{13}$$

where MWT is the maximum working time (8 hours + 2 hours); MVT is the maximum valid working time; $\mathrm{AS}(t_i)$ is an arrival station of task i; $\mathrm{DS}(t_{i+1})$ is a depart station of task $(i + 1)$; $T_{i,i+1}$ is the necessary time to reach $\mathrm{DS}(t_{i+1})$ from $\mathrm{AS}(t_i)$.

Here we propose a method to generate the valid potential duties as follows in Algorithm 1. The algorithm is initialized by setting a number of pieces N_p and i equal to 0. The relief opportunities are selected to generate a set of all the possible pieces of work M according to a given bus schedule. Notice that a maximum number of pieces, which is limited to compose a duty, are different in some practical cases, for example, no more than four spells in [18]. Another example is from [19] and in their case the maximum is equal to 2. Hence, we will refer to a required number of pieces given by the bus company as N_r. Let us denote $t_{\min}$ corresponding to the shortest length of piece in M (line (4)). Then, we can compute a maximum number of pieces $N_{\max}$ (maximum working time of duty divided by the shortest time of piece) on line (5). N_p is thus determined by N_r and $N_{\max}$. Finally, the process adds one piece of work after one piece of work into a duty until the maximum number of pieces is met. Nested loops on lines (7)–(19) are iterated gradually through all pieces of work of each vehicle. Line (12) checks if the duties satisfy the constrains (9)–(13) before adding the duties to the pool P.

3.2. Step 1: Construct an Initial Solution. For computing initial shadow prices, we propose a greedy constructive heuristic.

(1) Initialize $Q \leftarrow \emptyset$
(2) **while** $Q \neq M$ **do**
(3) Choose $D_j \in S$ with minimize cost c'_j
(4) $Q_{k+1} = Q_k \cup D_j$
(5) **end while**
(6) Find an initial solution to BDSP

Algorithm 2: Greedy constructive heuristic.

An initial solution is made by a sequential mechanism, whose main objective is to obtain initial values of dual variables to select more duties and improve the linear solution in Step 2. The procedure is presented in Algorithm 2. Given a set of legal duties $S = \{D_1, D_2, D_3, \ldots, D_n\}$ and a set of pieces of work to be covered $M = \{1, 2, \ldots, m\}$, let Q be a set of the pieces of work covered so far. As it is a constructive heuristic, a new duty not yet assigned is added in a greedy way to the current schedule at each iteration. With this objective, we calculate a new cost of duty c' as a greedy function for our choice at each iteration. The new cost of duty c'_j can be computed as the following:

$$c'_j = \frac{c_j}{\sum_{i=1}^{m} \beta_{ij}}, \quad j = 1, 2, \ldots, n, \tag{14}$$

where c_j is the cost of duty D_j in the rth iteration; $\beta_{ij} = 1$ if not yet covered piece of work i is covered by D_j after the rth iteration, and 0 otherwise. Particularly, if $\sum_{i=1}^{m} \beta_{ij} = 0$, $c'_j = c_j$.

Finally, the process is repeated until all pieces of work are covered.

3.3. Step 2: The Restricted Master Problem. Following a typical column generation, the LP relaxation of the IP is necessary to be applied; that is, we relax the integer constraints (4). Thus, constraints (4) can be rewritten as

$$x_j \in R, \quad j \in N. \tag{15}$$

The purpose of solving RMP is to find the dual prices, or shadow prices, corresponding to the pieces of work in the problem. These values are used to calculate the reduced cost of new duties described in Step 4.

3.4. Step 3: Duty Management. Duty management (or column management) is an integral part of any successful column generation algorithm as discussed in [18–20].

In this research, there are two reasons why this process is applied.

Firstly, at each column generation iteration we insert all columns with negative reduced costs that have been found by the pricing algorithms into the RMP (see Step 4). Thus, it is clear that the procedure to RMP is not efficient if there are more columns than a certain maximum amount. In our test, all the duties with reduced cost greater than a given threshold will be removed whenever the number of duties exceeds 50,000. Similar strategies are used in [21]. This process is particularly important when the RMP keeps growing enormously during the column generation process.

Secondly, the selection of duties (see Step 4) can sometimes be sped up without loss of quality if the number of duties generated can be reduced intelligently [18]. Since the number of duties for some real-life problems may be quite large, we consider to take measures to prevent "bad" quality of duty into our procedure. The following procedure is iterated as long as the number of duties stays above 100,000. Each potential duty will be ranked by combining two attributes: an index which reflects its apparent efficiency (driving time divided by working time) and a ratio which reflects the identical duties; for example, a ratio of duty which is wholly contained by another duty is 0. Then, the lowest ranked duties are discarded until some definitive conditions are met. The removal of these duties reduces the overall size of a subproblem, which reduces run-time and memory usage requirement.

3.5. Step 4: Selection of New Duties. The major component of column generation is the procedure for selecting (generating) candidate duties to bring into the restricted subset in each iteration. The goal is to improve the LP relaxation of the restricted master problem. Thus, the objective of this subproblem is to select the negative reduced cost of duties not already in the restricted subset. We now provide the details of this procedure to identify duties with negative reduced costs at each iteration. The reduced cost is obtained from the cost and dual values j of each corresponding set covering constraint as given in the previous section. As only negative reduced-cost routes can enter the restricted master problem, the reduced cost should be minimized or constrained to be negative. Using the dual variables from the LP solution, the reduced cost of each duty is calculated by

$$\tilde{c} = c_j - \sum_{i=1}^{m} \pi_i a_{ij}, \quad j = 1, 2, \ldots, n, \tag{16}$$

where $\tilde{c}$ is the reduced cost of duty (column) j; c_j is cost of duty j; π_i is the dual price of task i.

In this research, we develop a heuristic method based on the idea of hyper-heuristic that, while efficient, is not guaranteed to identify all the columns. To speed up the search for columns with negative reduced cost, we perform heuristic pricing. Besides saving computing time, this also allows generating multiple columns in each run. For one heuristic method is greedy, by applying different heuristic methods, we can keep the diversity of columns and avoid getting into a local optimal direction. Based on the framework of hyper-heuristic [14, 22], the competition rules are applied to guide the selection of the low-level heuristic during the search process. When a low-level heuristic is applied, the performance of this low-level heuristic will be selected by a function Δ. To prevent holding a large computation time by a poor performance of low heuristic, the simple ranking [23] is used to exclude the worst heuristic according to its scores in the last application at each iteration. There are four different cases (shown in Table 1) to reflect the important performance of a low-level heuristic.

(1) Determine properties of each duty: number of spells, duty type and cost.
(2) Construct different neighborhood structures.
(3) Begin with one neighborhood structure S_i' which is randomly selected to search.
(4) **while** countmove < Maxumummove **do**
(5) Candidate duty D_i is chosen randomly in S_i'
(6) $D_{\text{incumbent}} \leftarrow D_i$
(7) $D_{\text{current}} \leftarrow$ Apply 2-exchange
(8) **if** current duty is better than incumbent **then**
(9) $D_{\text{incumbent}} \leftarrow D_{\text{current}}$
(10) **end if**
(11) countmove++
(12) **end while**
(13) Add $D_{\text{incumbent}}$ to RMP if it has a negative reduced cost

Algorithm 3: Local search.

Table 1: Reward in the particular cases.

Case	Description	Score
Case 1	A negative reduced cost of duty found is the minimum among others.	3
Case 2	The execution time is less than other heuristics.	2
Case 3	The CPU seconds elapsed is much more than other heuristics since it was last called.	1
Case 4	No duty with negative reduced cost found.	−1

At each iteration, Δ of each low-level heuristic is calculated by

$$\Delta = \sum_{k=1}^{4} r_k, \tag{17}$$

where r_k is the score of case k.

Note that if each low-heuristic has the same total score at given iteration, no low-heuristic will be excluded. Five heuristics select new duties in the same time at the beginning.

3.5.1. Swap Heuristic. The authors in [24] consider that if a column in the optimal solution obtained by Step 2 is modified appropriately, it is likely to get a new column with negative reduced cost. For this reason, we try to select a number of columns in this heuristic.

Swap heuristic is formally written as follows. Given a duty D_j from optimal solution $S = \{D_1, D_2, \ldots, D_n\}$ of the restricted master problem, we decompose the duty D_j that is randomly selected in partial consecutive duty into h, k pieces of work (POW). Moreover, assuming that $h \le k$ and $h + k = n$, then D_j is decomposed as $D_{j1} = \cup_{i=1}^{h}\text{POW}$, $D_{j2} = \cup_{i=h+1}^{n}\text{POW}$. Therefore, all the duties which contain either $D_{j1} = \cup_{i=1}^{h}\text{POW}$ or $D_{j2} = \cup_{i=h+1}^{n}\text{POW}$ can be found in P (see in Step 0). Thus, we can select the duty with negative reduced cost to RMP. This process is executed until either at least one new duty with negative reduced cost can be found or a predefined number of attempts without success is reached.

3.5.2. Local Search Version 1. As we know, the element of search space consists of various duties. Thus, we could consider several neighborhood relations according to the properties of a duty, that is, duty type, number of spells, and cost. Then, one of neighbors is randomly selected in order to obtain an enough "good" duty after applying a local search. In the following, we will describe a neighborhood relation that we consider in our case.

It was mentioned earlier that five types of a duty are specified: Early Duty, Late Duty, Night Duty, Day Duty, and Middle Duty. Let $S_i' = (D_1', D_2', \ldots, D_n')$ be a subset of duties that each duty $D_i' \in S_i'$ is performed in the same duty type. As illustrated in the pseudo code of Algorithm 3, the neighborhoods S_i' allow the search to move (Step (3) of Algorithm 3). Then, the local search starts from a duty D_i by using 2-exchange, which follows a random sequence of duties in S_i' (Steps (5)–(7) of Algorithm 3). If D_{current} with less reduced cost is found, $D_{\text{incumbent}}$ will be replaced (Step (8) of Algorithm 3). All the above steps will be repeated until the predefined number of iterations is reached.

3.5.3. Local Search Version 2. This heuristic is similar to *local search version* 1, except the neighborhood selection mode that a greedy strategy is used. We define a set of neighborhoods used according to the number of spells. Thus, these different neighborhoods can be ranked in a descendant order. The duty with the maximum number of spells will be checked at each iteration. However, another neighbor will be selected according to the rank if a duty with negative reduced cost cannot be found in this neighbor after 10 attempts without success. And so on like this way to search, the entire search stops when a suitable stopping condition is met (here we use a maximum allowed number of iterations based on the CPU time).

3.5.4. A Greedy-Based Heuristic Version 1. Instead of searching the whole space, the first heuristic attempts to drive the search towards definitive zone of space. We maintain a reasonable size of the duties which are deemed to search efficiently (i.e., a limited number of duties $N_{\text{limit}} \subset N$). Then we try to find a "desired" duty in this subset.

More exactly, let P_{cost} be a pool which only stores the duties in which the costs are less than an adaptive threshold. This threshold is dynamically controlled by a parameter η. Then, the duties are ordered in a descendant order according to their costs. Duties, therefore, will be checked one by one according to this ranking order until one of some stopping criteria is met, which is either a time limit or finding one duty with negative reduced cost. In fact, the heuristic cannot find a "desired" duty after a certain time when entire search is stopped; it perhaps means that the threshold should be increased. Here we define that the threshold will be updated by $\min(c_j)$.

3.5.5. A Greedy-Based Heuristic Version 2. This heuristic is similar to a greedy-based heuristic version 1, except that a duty which is selected randomly in the pool P will be checked at each iteration.

3.6. Step 5: Stopping Criterion. In the context of this work, two stopping criteria are defined in our approach: (1) no duty with negative reduced cost is found and (2) the time limit is reached.

3.7. Step 6: Finding Integer Solutions. As mentioned earlier, column generation is used to solve the relaxed master problem. Hence, the resulting LP solution produced a solution with fractional numbers of duties and an integer solution must be found. In this case, a branch and bound procedure is used in order to deliver an integer solution. Several branching strategies compatible with column generation techniques are discussed [21, 25]. Constraint branching, which was originally proposed in [26], is widely used for branching strategies. In theory, constraint branching is used to set partitioning models and could lead to suboptimality if applied to inequality constraints, but since the optimal integer solution to the current problem will not necessarily equate to the optimal schedule and users can seldom specify what they mean by optimality, the emphasis is on finding a good integer solution quickly. Formally, let $J(s,t)$ be a set of variables from the optimal fractional solution in the RMP, where each variable covers tasks s and t simultaneously and $\sum_{j\in J(s,t)} x_j$ as the sum of solution values of the variables in the set $J(s,t)$. In [26], any optimal fractional solution contains at least one constraint pair s, t, for which $\sum_{j\in J(s,t)} x_j$ lies strictly between zero and one (see a proof in [27]):

$$0 < \sum_{j\in J(s,t)} x_j < 1. \tag{18}$$

The equivalent of the zero branch and the one branch are, respectively,

$$\sum_{j\in J(s,t)} x_j \leq 0, \tag{19}$$

$$1 \leq \sum_{j\in J(s,t)} x_j. \tag{20}$$

Applying the above branching strategy on our problem, branching is done by dividing the solution space into two sets. The branching can be imposed on the 0-branch:

$$\sum_{j\in J(s,t)} x_j = 0. \tag{21}$$

The 0-branch forces all duties covering both tasks are banned. Then on the 1-branch,

$$1 \leq \sum_{j\in J(s,t)} x_j. \tag{22}$$

The 1-branch implies that any duty which covers r_1 (or r_2) but not r_2 (or r_1) must be excluded.

Instead of solving the linear program to optimality, we can choose to prematurely end the process if the relative gap between upper bound and the lower bound is smaller than ε. Upper bound on the objective value is given by the integer solutions found.

A lower bound can be obtained by Step 3.

LB is the lower bound on the optimal solution value computed by solving the LP relaxation of (1) and rounding up the corresponding value. Column generation iterations can be terminated when this gap is less than a user-defined tolerance, giving a near-optimal solution of defined quality.

Definition 3. T_b is the total running time of bus, which can be computed from a bus schedule.

Assuming that the users predefine MWT as the maximum working time of a driver per day, N_E given to estimate the number of drivers expected is calculated by

$$N_E = \left\lceil \frac{T_b}{\text{MWT}} \right\rceil. \tag{23}$$

Proposition 4. *N_E is a tight low bound.*

Proof. Given a bus schedule, T_b is constant. Let $D_S = (d_1, d_2, \ldots, d_n)$ be the set of duties in an optimum solution S. Note N_S is a number of drivers in S.

Case a. the effective working hours of duty $d \in D_S$ are the maximum working time MWT. Therefore,

$$N_E = N_S = \left\lceil \frac{T_b}{\text{MWT}} \right\rceil. \tag{24}$$

Case b. It has occurred when there is at least one duty $d_i \in D_S$ such that the effective working hours are not equal to MWT:

$$N_E < N_S = \left\lceil \frac{T_b}{\text{MWT}} \right\rceil + n, \tag{25}$$

where $1 \leq n$ and integer.

Hence N_E is a tight low bound. □

The process continues until the gap is less than a user-defined tolerance, giving a near-optimal solution of defined quality. However, this could be time consuming for large problems. Therefore, the branching process will terminate the search once one integral solution has been found if a given execution time is exceeded.

Table 2: Comparative results against the best known schedules.

Data	Number of pieces of work	Number of legal duties	Best known solutions	
			Number of duties in schedule	Time (seconds)
A1	25	1,006	12	0.10
A2	50	10,552	20	4.63
A3	100	42,473	40	1.88
A4	250	990,065	81	70.90
A5	500	8,388,608	145	6567.80
B1	200	17,131	25	—
B2	265	19,835	73	—

4. Computational Results

In this section, we present the results obtained for the proposed approach on several real-world instances. These instances vary in size and complexity. The instances, denoted by A1–A5in Table 2, are from Mauri & Lorena [15]. For the last two instances, the test data comes from two lines of bus in Beijing. Table 2 shows the size (pieces of work) and the best known results of real-world problems.

In order to evaluate the quality of the solution obtained by our approach presented here, the measure of the quality of a heuristic schedule is shown as follows:

$$\text{Devation} = \left(\frac{\text{result} - \text{bestknown}}{\text{bestknown}}\right) \times 100\%. \tag{26}$$

All computational experiments were performed in a personal computer with Intel Core 2 Duo CPU T5870 that is processor of 2.00 GHz, 1.99 GHz with 2.00 G of RAM memory on Microsoft Windows 2002. The whole implementation was developed in the C # language and has been complied using the NET framework version 2.2.30729. The ILP solver used in experiments is LINGO 11.0. This software has been used to solve RMP after each pricing iteration. The threshold has been set to 5 min $\tilde{c}$ in Step 3. The computation time to explore search space in the subproblem is limited to 2 hours (3,600 seconds).

The results of Table 3 show that in most of cases of instances A1–A5 our method could find solutions which are the same as the best known solutions. We can also find that our method does not perform well in some small size instances like A1–A3 compared to PTA/LP and TA/LP, even to SA. However, as the size of instance increases, our method performs better. This is because that, for small instance, our method is too complex and its computing time is long. But for large instance, the classic methods meet the problem of combinational explosion which is the typical feature of NP problem, while our method begins to show its advantage of saving computing time.

In Table 4 we present the "manual" solutions of last two instances currently used in the company and the solutions obtained with our approach. Our solutions represent a reduction of −8% and −8.22%, respectively, on the manual solutions. This improved schedule was very acceptable to the user.

Table 3: Comparative results against the best known schedules.

Problem	Used method	Number of duties	Deviation%	Times
A1	PTA/LP[a]	12	0	0.10
	TA/LP[b]	12		0.30
	SA[c]	12		1.90
	SA_20[d]	12		35.48
	CGBH[e]	12		10.58
A2	PTA/LP	20	0	4.63
	TA/LP	20		41.64
	SA	20		42.87
	SA_20	20		949.04
	CGBH	20		39.53
A3	PTA/LP	40	2.5	1.88
	TA/LP	40		4.98
	SA	40		7.60
	SA_20	40		173.26
	CGBH	41		150.25
A4	PTA/LP	81	1.23	70.90
	TA/LP	82		229.68
	SA	85		199.86
	SA_20	83		3749.02
	CGBH	82		1309.07
A5	PTA/LP	145	6.89	6567.80
	TA/LP	151		6717.30
	SA	153		7061.52
	SA_20	153		143565.20
	CGBH	155		4038.13
Average			2.12	

[a]Population training algorithm/linear programming with more iterative.
[b]Population training algorithm/linear programming with few iterative.
[c]Simulated annealing.
[d]Simulated annealing with 20 executions.
[e]Colunm generation based hyper-heuristic approach.

5. Conclusion

We describe a methodology for finding near-optimal solutions to bus driver scheduling problem. Suitable combination of column generation and hyper-heuristic can benefit much from synergy and exhibit higher performance with respect to solution quality and time for some cases, whereas the combination of column generation and hyper-heuristic also

Table 4: Comparison with manual solutions.

Type of Duty	B1			B2		
	Manual Solution	Our Solution	Deviation%	Manual Solution	Our Solution	Deviation%
Early Duty	8	6		27	25	
Late Duty	8	2		27	19	
Day Duty	0	4		0	6	
Middle Duty	9	11		19	17	
Total Duty	25	23	−8	73	67	−8.22

needs substantial further research in order to make them fully developed. Our future work on such hybrid approach will be still continued.

Conflict of Interests

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

Acknowledgment

This work is supported by the Fundamental Research Funds for the Central Universities (2014JBM072).

References

[1] H. R. Lourenço, J. P. Paixão, and R. Portugal, "Multiobjective metaheuristics for the bus-driver scheduling problem," *Transportation Science*, vol. 35, no. 3, pp. 331–343, 2001.

[2] A. Wren, *Computer Scheduling of Public Transport*, North-Holland Publishing, Amsterdam, The Netherlands, 1981.

[3] M. Desrochers and J. M. E. Rousseau, *Computer Scheduling of Public Transport*, Elsevier, Berlin, Germany, 2009.

[4] J. R. Daduna, I. Branco, and J. M. P. Paixão, *Computer-Aided Transit Scheduling: Proceedings of the Sixth International Workshop on Computer-Aided Scheduling of Public Transport*, vol. 430, Lecture Notes in Economics and Mathematical Systems, 1995.

[5] M. D. Hickman, P. B. Mirchandani, and S. Voss, *Computer-Aided Systems in Public Transport*, Springer, Berlin, Germany, 2008.

[6] S. Voss and J. R. Daduna, *Computer-Aided Scheduling of Public Transport*, Springer, 2001.

[7] A. Wren and J. Rousseau, *Bus Driver Scheduling—An Overview*, Springer, Berlin, Germany, 1995.

[8] L. Zhao, "A heuristic method for analyzing driver scheduling problem," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 36, no. 3, pp. 521–531, 2006.

[9] M. Desrochers and F. Soumis, "A column generation approach to the urban transit crew scheduling problem," *Transportation Science*, vol. 23, pp. 1–13, 1989.

[10] S. Lavoie, M. Minoux, and E. Odier, *A New Approach of Crew Pairing Problems by Column Generation and Application to Air Transports*, Université de Montréal, Centre de Recherche sur les Transports, 1985.

[11] T. Nishi, Y. Muroi, and M. Inuiguchi, "Column generation with dual inequalities for railway crew scheduling problems," *Public Transport*, vol. 3, no. 1, pp. 25–42, 2011.

[12] S. Fores, L. Proll, and A. Wren, *A Column Generation Approach to Bus Driver Scheduling*, Elsevier, 1996.

[13] N. Touati Moungla, L. Létocart, and A. Nagih, "Solutions diversification in a column generation algorithm," *Algorithmic Operations Research*, vol. 5, no. 2, pp. 86–95, 2010.

[14] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, "Hyper-heuristics: an emerging direction in modern search technology," in *Handbook of Metaheuristics*, vol. 57 of *International Series in Operations Research & Management Science*, pp. 457–474, Springer, Berlin, Germany, 2003.

[15] G. R. Mauri and L. A. N. Lorena, "A new hybrid heuristic for driver scheduling," *International Journal of Hybrid Intelligent Systems*, vol. 4, pp. 39–47, 2007.

[16] R. S. K. Kwan and A. Kwan, "Effective search space control for large and/or complex driver scheduling problems," *Annals of Operations Research*, vol. 155, no. 1, pp. 417–435, 2007.

[17] S. Fores, L. Proll, and A. Wren, "Experiences with a flexible driver scheduler," in *Computer-Aided Scheduling of Public Transport*, S. Vo and J. R. Daduna, Eds., vol. 505 of *Lecture Notes in Economics and Mathematical Systems*, pp. 137–152, Springer, Berlin, Germany, 2001.

[18] A. Wren, S. Fores, A. Kwan, R. Kwan, M. Parker, and L. Proll, "A flexible system for scheduling drivers," *Journal of Scheduling*, vol. 6, no. 5, pp. 437–455, 2003.

[19] D. Huisman, R. Freling, and A. P. M. Wagelmans, "Multiple-depot integrated vehicle and crew scheduling," *Transportation Science*, vol. 39, no. 4, pp. 491–502, 2005.

[20] İ. Mutera, Ş. İ. Birbila, K. Bülbüla et al., "Solving a robust airline crew pairing problem with column generation," *Computers & Operations Research*, vol. 40, no. 3, pp. 815–830, 2013.

[21] F. Liberatore, G. Righini, and M. Salani, "A column generation algorithm for the vehicle routing problem with soft time windows," *4OR*, vol. 9, no. 1, pp. 49–82, 2011.

[22] E. K. Burker, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches," in *Handbook of Metaheuristics*, M. Gendreau and J. Potvin, Eds., vol. 146, pp. 449–468, Springer, Boston, MA, USA, 2010.

[23] E. Burke, T. Curtois, M. Hyde et al., "Iterated local search vs. hyper-heuristics: towards general-purpose search algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '10)*, pp. 1–8, Barcelona, Spain, July 2010.

[24] Z.-L. Chen and H. Xu, "Dynamic column generation for dynamic vehicle routing with time windows," *Transportation Science*, vol. 40, no. 1, pp. 74–88, 2006.

[25] M. Mesquita and A. Paias, "Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem," *Computers and Operations Research*, vol. 35, no. 5, pp. 1562–1575, 2008.

[26] D. M. Ryan and B. A. Foster, "An integer programming approach to scheduling," *Computer Scheduling of Public Transport*, vol. 1, pp. 269–280, 1981.

[27] C. Barnhart, N. L. Boland, L. W. Clarke, E. L. Johnson, G. L. Nemhauser, and R. G. Shenoi, "Flight string models for aircraft fleeting and routing," *Transportation Science*, vol. 32, no. 3, pp. 208–220, 1998.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 348582, 7 pages
http://dx.doi.org/10.1155/2015/348582

Research Article

A Cellular Automata Traffic Flow Model considering Bus Lane Changing Behavior with Scheduling Parameters

Zun-dong Zhang,[1] Yan-fang Yang,[2] Wenjiao Qi,[1] Abderrahim Chariete,[3] and Xing-xiang Lin[4]

[1] *Beijing Key Lab of Urban Intelligent Traffic Control Technology, North China University of Technology, Beijing 100041, China*
[2] *State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China*
[3] *OPERA Laboratory, University of Technology of Belfort-Montbéliard, 90000 Belfort, France*
[4] *Xiamen Traffic Management Bureau, Xiamen 361003, China*

Correspondence should be addressed to Zun-dong Zhang; zdzhang@ncut.edu.cn

Received 19 November 2014; Revised 22 January 2015; Accepted 23 January 2015

Academic Editor: Shuenn-Ren Cheng

According to different driving behavioral characteristics of bus drivers, a cellular automata traffic model considering the bus lane changing behavior with scheduling parameters is proposed in this paper. Traffic bottleneck problems caused by bus stops are simulated in multiple lanes roads with no-bay bus stations. With the mixed traffic flow composed of different bus arrival rate, flow-density graph, density distribution graph, and temporal-spatial graph are presented. Furthermore, the mixed traffic flow characteristics are analyzed. Numerical experiment results show that the proposed model can generate a variety of complicated realistic phenomena in the traffic system with bus stops and provide theoretical basis for better using of traffic flow model.

1. Introduction

Because cellular automata have characteristics that its model is simple and can effectively reproduce the characteristics of the complex transport phenomena when computer-implemented; after it was introduced into the transport field, it has been rapidly developed [1–5]. As one of the major transportation systems, the operating status of public transportation system can affect the entire transportation system status, so the study of impact of public transport on road traffic flow has become a hot issue in recent years. Jia et al. [6] and Li et al. [7] compared dynamic characteristics of road traffic flow in harbor style and nonharbor style two different settings of bus station and analyzed the impact of bus proportion and the length of the bus station. Song and Zhao [8] and so forth used two-lane cellular automaton model simulated traffic bottleneck problems caused by interactions of entrance ramp and bus stops and discussed bus stops upstream and downstream two cases at the entrance ramp, respectively. Zhu [9] compared and analyzed urban traffic flow characteristics in the two-lane traffic system under three conditions: having bus lanes, intermittent bus specific road, and no bus lanes. Luo et al. [10] simulated and analyzed traffic flow of bus lines based on different departure rate and arrival rate.

However, the current bus behavior modeling lacks of further refinement on into and out station behaviors, that buses using a fixed stop time, first in first out (FIFO) principle, and so forth. But in reality, there are often cases that multiple buses simultaneously inbound, the actual stop time of bus is infected by passengers getting on and off, and buses last in first out (LIFO), and so forth. So the current bus behavior modeling is inconsistent with the actual situation.

In response to the problems existing in the model above, this paper takes multilane cellular automaton model as background, by defining bus parking traffic behavior rules in bus lanes and nonbay bus stations; the paper examines its impact on road traffic flows.

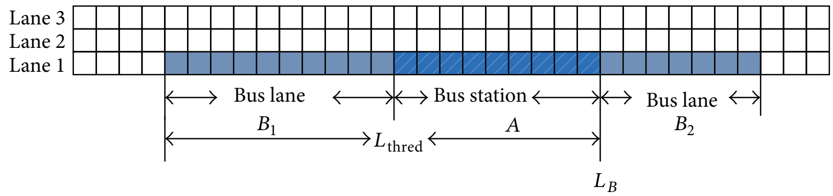


Figure 1: Schematic illustration of bus station.

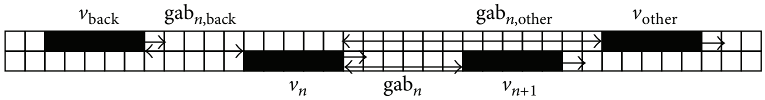


Figure 2: Schematic illustration of lane changing rules.

2. The CA Model

This paper discusses the nonharbor bus stops impact on traffic flow characteristics. By applying cellular automata model, sections that have bus stations are described as the system shown in Figure 1. The sections have three lanes (lane 1, lane 2, and lane 3); section length has L cells. On the road, the vehicles are free to change. At the bus lanes, social vehicles are forced to change lanes into nonbus lanes to travel; buses on nondedicated lane need to change lanes into lane 1 to achieve getting into station and parking. The front part of bus station is set in L_B. The system considers social vehicles and buses two types; among them, social vehicles are divided into cars and coaches, and a car occupies 5 cells; a bus and a coach both occupy 10 cells. The maximum speed of cars is $v_{\max 1} = 25$ cells; the maximum speed of buses and coaches is $v_{\max 2} = 15$ cells.

2.1. Lane Change Rules. In addition to areas A and $B_{1,2}$, vehicles can change lanes freely on the other sections; vehicle liberal lane changing rules are as follows:

$$\begin{gathered}\text{gap}_n < \min\{v_n + 1, v_{\max}\},\\ \text{gap}_{n,\text{other}} > \text{gap}_n,\\ \text{gap}_{n,\text{back}} > v_{\max}.\end{gathered} \tag{1}$$

Among them, gap_n, $\text{gap}_{n,\text{other}}$, and $\text{gap}_{n,\text{back}}$, respectively, represent empty cellular number between vehicle n and original lane vehicle $n+1$, the front and behind vehicle in target lane (as shown in Figure 2). Particularly, when there are overlapping sections between vehicle n and the front and behind vehicle in target lane, $\text{gap}_{n,\text{other}}$ and $\text{gap}_{n,\text{back}}$ can take negative number. $\text{gap}_n < \min\{v_{n+1}, v_{\max}\}$, $\text{gap}_{n,\text{other}} > \text{gap}_n$ indicates lane changing motivations and shows that the vehicle cannot continue acceleration driving on the current lane, whereas the target lane driving conditions are better than the current lane. $\text{gap}_{n,\text{back}} > v_{\max}$ is safety conditions that guarantee the car will not collide with the behind car on the target lane after it changes the lane.

In areas A and $B_{1,2}$, buses have special lane changing rules; specific rules are as follows.

2.1.1. In-Station Rules. Buses in the system have three states; use k to denote, namely, (1) buses not parking in the stop, $k = 0$; (2) buses parking in the stop, $k = 1$; (3) buses have left the stop, $k = 2$.

In the areas A and $B_{1,2}$, the state of the bus is $k = 0$; at this time the bus lane changing motivation is different to liberal lane changing motivation, whether driving conditions on lane 1 is good or bad; buses that have not stopped on lane 2 must change lanes to lane 1 (bus lanes) to realize getting into station and parking. At this time the mandatory lane changing rules are

$$\begin{gathered}\text{gap}_{n,\text{other}} \geq v_n - v_{\text{other}},\\ \text{gap}_{n,\text{back}} \geq v_{\text{back}} - v_n.\end{gathered} \tag{2}$$

Equation (2) shows that as long as the current buses do not collide with the front and behind vehicle in target lane, lane changes can be achieved. Meanwhile, if buses on the right lane did not park, then they are prohibited from changing to left lanes before getting into station and parking.

If the Formula (2) is not meet, then the bus n will send a signal to the behind vehicle on the target lane, behind vehicle on the target lane avoids the bus with a certain probability, then it meets the condition that

$$\begin{gathered}\text{gap}_{n,\text{back}} \geq 0,\\ \text{gap}_{n,\text{back}} < v_{\text{back}} - v,\\ \text{rand}\,() < p, \qquad p = 1 - \frac{L_B - x_n}{L_{\text{thred}}},\\ v_{\text{back}} = \max\left\{0, \text{gap}_{n,\text{back}}\right\}.\end{gathered} \tag{3}$$

Among them, rand() represents the random number in $[0,1]$, x_n is the location of the bus n, L_{thred} is the total length of the regional A and B_1, and p is the probability of behind car avoidance; as the bus n is closer to L_B, the greater the probability of p, the higher the probability of collision avoidance.

If the bus n on the left lane does not change lanes successfully until driving to L_B, then the bus will stop at this location, in order to wait for the opportunity to change lanes.

Buses begin to stop next to the buses in front after entering the bus station, in the bus station; when the bus speed is 0, we consider the bus stops and parking; passengers begin to get on and off, at this time, $k = 1$.

2.1.2. Out-Station Rules. After the bus parking, $k = 1$, stop time of bus n is T_s^n; due to the different load factor of each bus, passengers getting on and off time is also different; assume the car parking time T_s^n is uniformly distributed. When the parking time meets $t_{\text{stop}}^n = T_s^n$, bus n completes parking, at this time the status of bus n changes to $k = 2$; bus n can continue moving forward.

If the vehicle $n + 1$ in front of bus n is still parking, then bus n looks for changing lane chances in probability p_{change}; otherwise it will still remain in the place; parking cumulative time is $t_{\text{stop}}^n = T_s^n + 1$; then the changing lane conditions are

$$\text{gap}_{n,\text{back}} \geq 0, \text{gap}_{n,\text{back}} > v_{\text{back}},$$
$$\text{rand}\,() < p_{\text{change}}, \quad p_{\text{change}} = 1 - \frac{T_{\max} - t_{\text{stop}}^n}{T_{\max}}. \tag{4}$$

Among them, rand() represents the random number in $[0, 1]$ and $T_{\max}$ is the maximum time bus drivers can wait for; the longer the wait time, the greater the probability of the bus changing to the left lanes. After pulling out of the bus station, buses have finished parking; in area B_2 they change lanes to liberal lane, at this time, and social vehicles cannot change lanes to bus lanes.

2.2. The Forward Movement Model. Forward movement rules of vehicles are improved on the basis of NaSch model [11]; evolution rules are as follows:

(1) acceleration: $v_n \rightarrow \min(v_n + 1, v_{\max})$;
(2) deceleration: $v_n \rightarrow \min(v_n, d_n)$;
(3) for nonstop buses, decelerations for the purpose of changing lanes, pit stop, and parking ($k = 0$): $v_n \rightarrow \min\{v_n, d_{Bn}\}$;
(4) random slow (with probability q): $v_n \rightarrow \max(v_n - 1, 0)$;
(5) location update: $x_n \rightarrow x_n + v_n$.

Among them, v_n represents the number of empty cells between the n vehicle and the $n+1$ vehicle, $d_n = x_{n+1} - x_n - l_{n+1}$ denotes the number of empty cells between the n vehicle and the front vehicle (namely, the $n + 1$ vehicle), and l_{n+1} is the length of the $n + 1$ vehicle, that is, the number of cells it occupies.

2.3. Boundary Conditions. The system uses the open boundary conditions, assuming that the left-most cell of each lane corresponds to $x = 1$, rightmost cell corresponds to $x = L$, x_{last} denotes head position of leftmost vehicle, and x_{lead} denotes head position of rightmost vehicle; in time step $t \rightarrow t + 1$, when the update of lane location on the road is completed, update the value of x_{last} and x_{lead}. If $x_{\text{last}} - l - 1 > 0$, then the vehicles in speed $\min\{x_{\text{last}} - l - 1, v_{\max}\}$ enter cells in probability α; inflow vehicles set buses according to a certain proportion R.

In the exit of lane, the system uses fully open boundary conditions; that is, when the head position of vehicle meets $x_{\text{lead}} > L$, the leading vehicle will exit the system, and the next vehicle will become the leading vehicle and will travel without hinder.

3. Simulation and Analysis

The system uses three lane cellular automaton models; buses, coaches, and cars are randomly placed at the initial state. Assuming road length is $L = 2000$ cells, departure probability of the system is $\alpha = 0.9$, system random slow probability is $q = 0.1$, head position of bus is $L_B = 1000$, the length of bus is 50 cells, namely, it can accommodate five buses at the same time, special lane changing interval length is $L_{\text{thred}} = 120$ cells, and bus stop time is uniformly distributed to $[30, 60]$. Simulation length is 20000 times; select the simulation data in the later 15000 times and make statistics of them.

Select the established bus into and out station model in [8]; for comparison, compare their simulation results, in which the proportion of buses is $R = 0.2$. First, make statistics of the traffic density of the system; its statistical formulas are as follows:

$$\rho = \frac{5(N_1 + 2N_2)}{4L},$$
$$v = \frac{N_1\bar{v}_1 + 2N_2\bar{v}_2}{N_1 + 2N_2}, \tag{5}$$
$$q = \rho v = \frac{5(N_1 + 2N_2)}{4L} \cdot \frac{N_1\bar{v}_1 + 2N_2\bar{v}_2}{N_1 + 2N_2}.$$

Among them, N_1 is the number of small cars in the systems; N_2 is the number of big cars and buses in the systems, $\bar{v}_1$ is the average speed of small cars in the systems, and $\bar{v}_2$ is the average speed of big cars and buses in the systems.

Flow and density comparative results of the two models are shown in Figure 3. As can be seen from Figure 3, at the same density value, the flow in the improved simulation model is bigger than that in the model of [8]. It mainly regulates the avoidance behavior of into and out station, plus the behavior of outbound advance; it also reduces the waiting time of buses getting into (or out) station; thereby it reduces the congestion probability of bus stop in the upstream.

Figure 4 is density-distribution diagram. As can be seen from the chart, the peak of solid line appears in the location of the bus station (950–1000 cells); it reflects the principles that haven't get into station buses have priority to get into station. When there is vacancy in bus station, buses which are not into station on adjacent lanes can choose to get into station and stop; however, the peak of dotted line appears in the pit mouth of bus station (950–900 cells), that buses must wait in line at the bus stop line. The downstream density of solid line is higher than the density of the dotted line; due to buses getting into station and parking buses timely, it increases downstream flow of buses. From the figure,

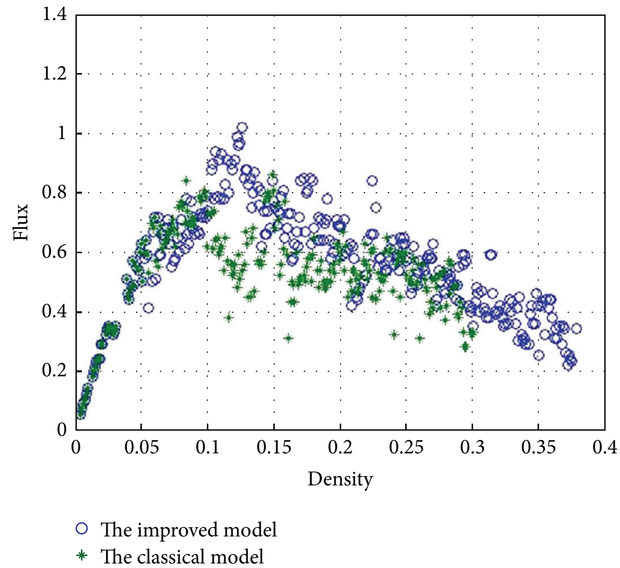


FIGURE 3: Flux-density diagram. "∘" is the value of proposed model; "∗" is the value of the classical model cited in [8].

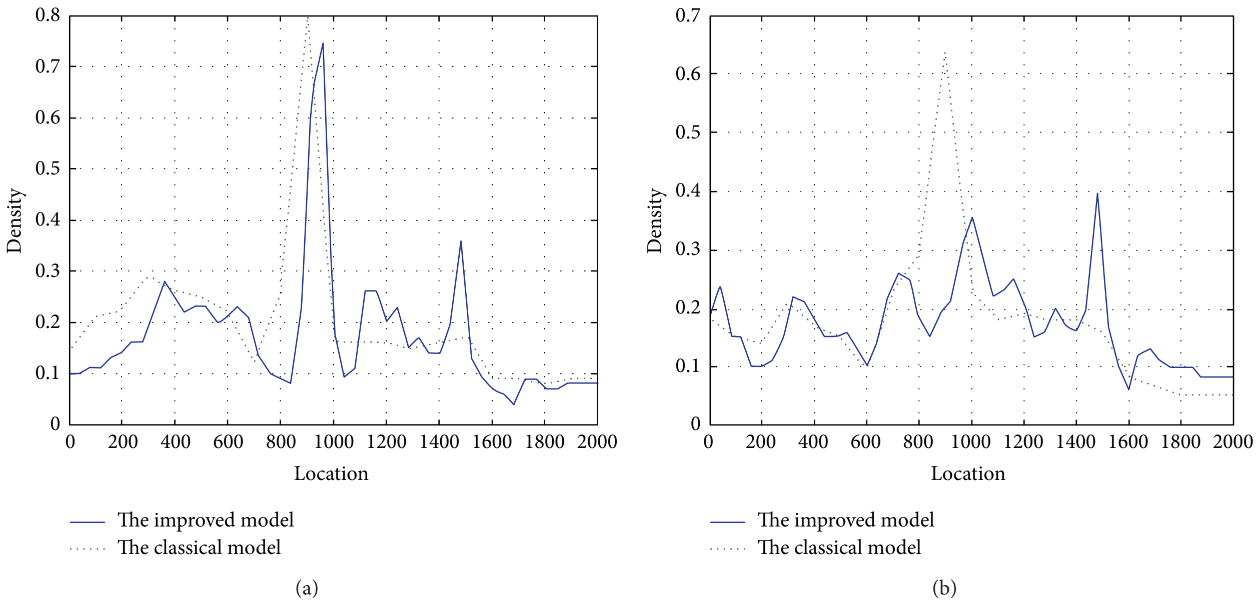


FIGURE 4: Density-distribution diagrams. (a) shows the density distribution in lane 1 and (b) shows the density distribution in lane 2.

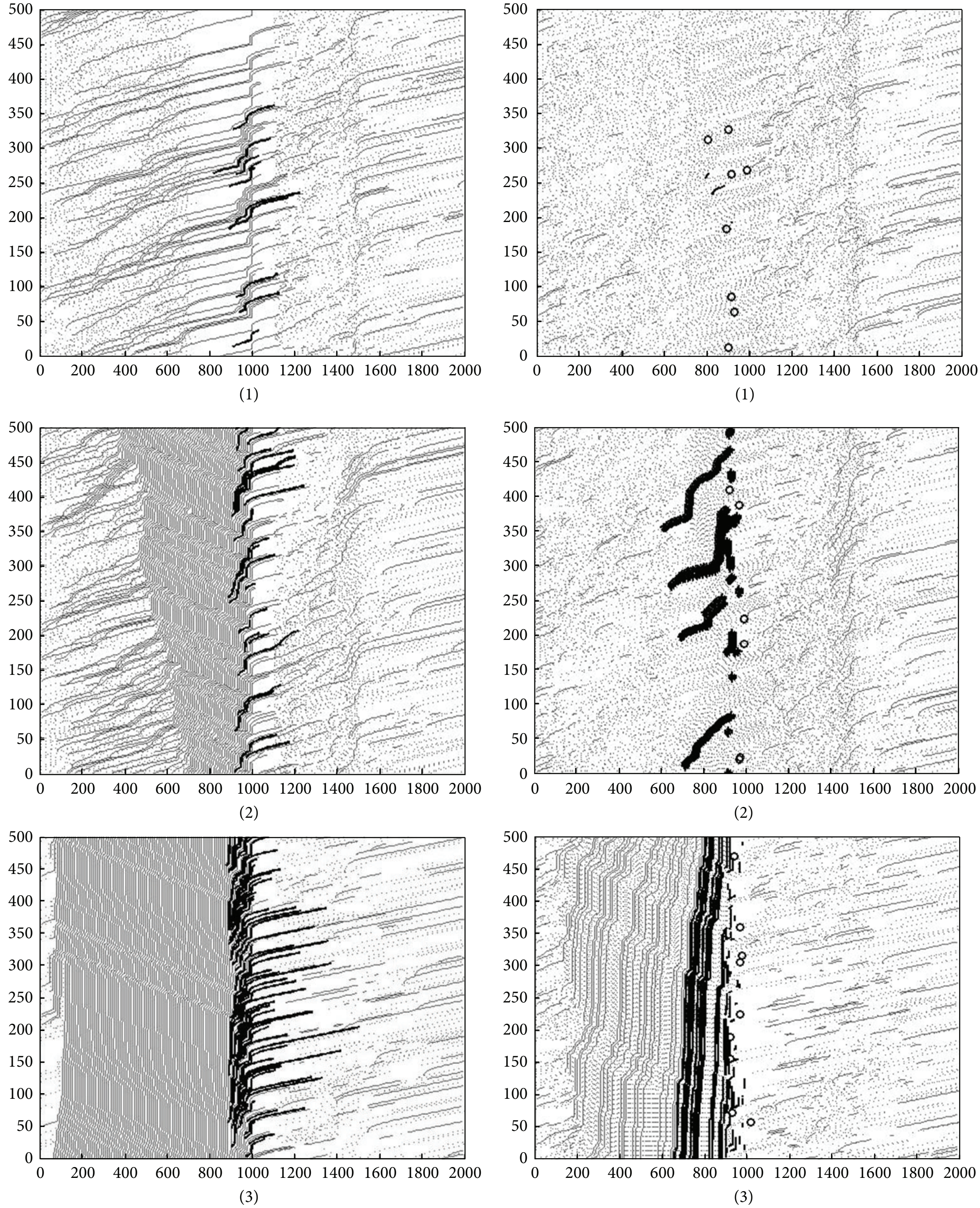
the density distribution of solid line is much more evenly than the density distribution of dotted line. Density peak of dotted line appears at the upstream of bus station; this is mainly because unsuccessful changing lane buses are queuing stint here.

In summary, the bus station processing ability in improved model should be higher than that in the original model.

Select bus proportion $R = 0.15$, 0.2, and 0.3, respectively, and analyze its traffic flow characteristics. Firstly, analyze the influences of bus proportion on bus getting into station behavior, as shown in Figures 5 and 6.

The black lines in the first column indicate vehicles at the location of bus station, which successfully change lanes from other lanes and get into station meanwhile to vehicles at the bus lane; the black line in the second column indicates vehicles that change into bus lanes from lane 2 at the location of bus station, while the black dots indicate second successfully changing lane vehicles, namely, from lane 3 swap to lane 2, and buses that finally swap to the bus lane (lane 1).

As can be seen from the comparison of the experimental results, when the bus proportion is low, cases that buses get into station simultaneously are very few; namely, buses complete lane change in front of bus stations; vehicles that



(a) (b)

Figure 5: Space-time diagram over 500 time steps. Horizontal axis is for position and vertical axis is for time. (a) is for lane 1 and (b) are for lane 2. The first row: $R = 0.15$, the second row: $R = 0.2$; the third row: $R = 0.3$.

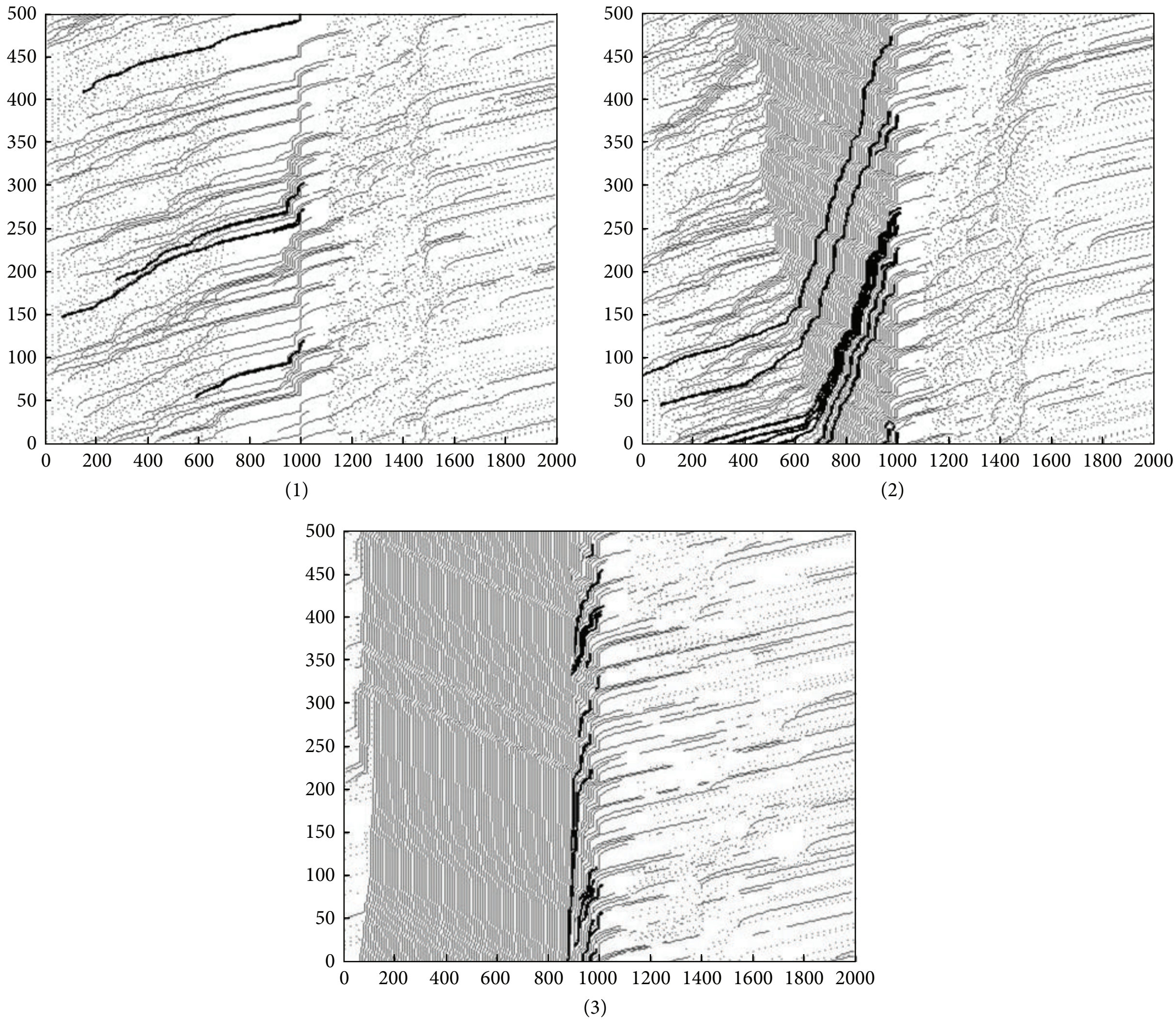


Figure 6: Space-time diagram for pulling out of bus stops. (1) $R = 0.15$, (2) $R = 0.2$, and (3) $R = 0.3$.

simultaneous to buses get into station are mainly vehicles that outside bus lanes (lane 1). When lane change to the bus lane is not successful, buses can only queue and wait to get into station on lane 2. With the increase of the bus proportion, unsuccessful probability of buses changing lanes into bus lane becomes higher; buses on lane 2 can only move forward while waiting on the original lane, so cases that get into station at the same time increase.

Secondly, analyze the influence of bus proportion on bus getting out station behavior, as shown in Figure 6. With the increase of bus proportion, the proportion of buses getting into station increases, and the proportion of outbound advance also increases.

4. Conclusions

This paper simulates and studies the nonbay bus station system on sections under the open boundary conditions and defines the buses getting into/out station rules and deeply studies the impact of buses getting into/out station behavior on traffic flow. It analyzes traffic flow characteristics by using flow and density diagram, density distribution, and space-time diagram.

As can be seen from the simulation results, the avoidance behavior rule, buses meanwhile stint and advance outbound rules; it reduces the chances of congestion in the system caused by buses queue and wait to get into station, and it is more suitable to the reality. Meanwhile, the probability of occurrence of the above acts increases as buses proportion increases.

Conflict of Interests

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

Acknowledgments

This paper is supported by the Scientists Training Project on Urban Road Traffic Intelligent Control (no.

PXM2014_014212_000053) and the Key Project of the Ministry of Public Security of China (no. 201302ZDYJ012) and the National Science & Technology Pillar Program of China (no. 2014BAG01B02).

References

[1] B. S. Kerner and S. L. Klenov, "Spatial-temporal patterns in heterogeneous traffic flow with a variety of driver behavioural characteristics and vehicle parameters," *Journal of Physics A: Mathematical and General*, vol. 37, no. 37, pp. 8753–8788, 2004.

[2] J.-X. Ding, H.-J. Huang, and T.-Q. Tang, "Random noise in driving behavior based on cellular automata," *Journal of Jilin University (Engineering and Technology Edition)*, vol. 39, supplement 2, pp. 66–70, 2009.

[3] L.-J. Peng and R. Kang, "One-dimensional cellular automaton model of traffic flow considering drivers' features," *Acta Physica Sinica*, vol. 58, no. 2, pp. 830–835, 2009.

[4] S. Kukida, J. Tanimoto, and A. Hagishima, "Analysis of the influence of lane changing on traffic-flow dynamics based on the cellular automaton model," *International Journal of Modern Physics C*, vol. 22, no. 3, pp. 271–281, 2011.

[5] X.-M. Liu and L. Wang, "Cellular automaton traffic flow model considering on-street parking," *Journal of Jilin University*, vol. 42, no. 2, pp. 327–333, 2012.

[6] B. Jia, X.-G. Li, R. Jiang, and Z.-Y. Gao, "The influence of bus stop on the dynamics of traffic flow," *Acta Physica Sinica*, vol. 58, no. 10, pp. 6845–6851, 2009.

[7] Q.-D. Li, L.-Y. Dong, and S.-Q. Dai, "Investigation on traffic bottleneck induce by bus stopping with a two-lane cellular automaton model," *Acta Physica Sinica*, vol. 58, no. 11, pp. 7584–7590, 2009.

[8] Y.-K. Song and X.-M. Zhao, "Combined bottleneck effect of on-ramp and bus stop in a cellular automaton model," *Chinese Physics B*, vol. 18, no. 12, pp. 5242–5248, 2009.

[9] H. B. Zhu, "Numerical study of urban traffic flow with dedicated bus lane and intermittent bus lane," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 16, pp. 3134–3139, 2010.

[10] Y.-J. Luo, B. Jia, X.-G. Li, C. Wang, and Z.-Y. Gao, "A realistic cellular automata model of bus route system based on open boundary," *Transportation Research Part C: Emerging Technologies*, vol. 25, pp. 202–213, 2012.

[11] K. Nagel and M. Schreckenberg, "A cellular automata model for traffic flow," *Journal de Physique*, vol. 2, pp. 2221–2229, 1992.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 149182, 6 pages
http://dx.doi.org/10.1155/2015/149182

Research Article

Bounded Fatou Components of Composite Transcendental Entire Functions with Gaps

Cunji Yang and Shaoming Wang

Department of Mathematics and Computer Sciences, Dali University, Dali, Yunnan 671003, China

Correspondence should be addressed to Cunji Yang; kmycj@126.com

Received 28 August 2014; Accepted 10 October 2014

Academic Editor: Chin-Chia Wu

We prove that composite transcendental entire functions with certain gaps have no unbounded Fatou component.

1. Introduction

Let f be a transcendental entire function. We write $f^1 = f$, and $f^n = f \circ f^{n-1}$ for $n \geq 2$ for the nth iterate of f. The Fatou set or set of normality $F(f)$ of f consists of all z in the complex plane $\mathbb{C}$ which has a neighborhood U such that the family $\{f^n \mid U : n \geq 1\}$ is a normal family. The Julia set $J(f)$ of f is $J(f) = \mathbb{C} \setminus F(f)$. For the fundamental results in the iteration theory of rational and entire functions, we refer to the original papers of Fatou [1–3], and Julia [4] and the books of Beardon [5], Carleson and Gamelin [6], Milnor [7], and Ren [8].

Let U be a connected component of $F(f)$. Then $f^n(U) \subseteq U_n$, where U_n is a component of $F(f)$. If there is a smallest positive integer p such that $U_p = U$, then U is periodic of period p. In particular, if $p = 1$, then U is called invariant. If for some integer $n \geq 1$, U_n is periodic, while U is not periodic then U is called preperiodic. If U is periodic and $f^n \mid U \to \infty$ then U is called a Baker domain. If all the U_n are disjoint, that is, for $n \neq m$, $U_n \neq U_m$ then U is called a wandering domain. Let f be a transcendental entire function. In 1981, Baker [9] proposed whether every component of $F(f)$ is bounded if the growth of f is sufficiently small. The appropriate growth condition would appear to be of order 1/2, minimal type at most. In [9], Baker observed that this condition is best possible in the following sense: for any sufficiently large positive a, the function

$$f(z) = z^{-1/2} \sin \sqrt{z} + z + a \tag{1}$$

is of order 1/2, mean type, and has an unbounded component U of $F(f)$ containing a segment $[x_0, \infty)$ of the positive real axis, such that $f^n(z) \to \infty$ as $n \to \infty$, locally uniformly in U.

It is conjectured that if the order of f is less than 1/2, minimal type, then every component of $F(f)$ is bounded. It is still open for wandering domains, although there are several remarkable results for wandering domains under the assumptions that the growth satisfies, in addition, some regular conditions; see [10–17].

Suppose that $f(z) = \sum_{n=0}^{\infty} a_n z^n$ is an entire function with gaps; that is, some of the a_n are zero, in a certain sense. Then the function has the form $f(z) = \sum_{k=0}^{\infty} a_k z^{n_k}$. We say that $f(z)$ has Fabry gaps if $n_k/k \to \infty$ as $k \to \infty$, and $f(z)$ has Fejér gaps if $\sum_{k=1}^{\infty}(1/n_k) < \infty$.

Wang [18] proved that every component of the Fatou set of an entire function with certain gaps is bounded, by using the properties of the entire functions with such gaps. Wang [18] obtained the following result.

Theorem 1. *Let $f(z) = \sum_{k=0}^{\infty} a_k z^{n_k}$ be an entire function with $0 < \mu \leq \rho < \infty$. If $f(z)$ has Fabry gaps, then every component of $F(f)$ is bounded.*

For Fejér gap, Wang [18] proposed the following problem. Let $f(z)$ be an entire function with Fejér gaps, that is,

$$\sum_{k=1}^{\infty} \frac{1}{n_k} < \infty. \tag{2}$$

Is every component of $F(f)$ bounded?

For composite of entire function, Qiao [19] proved the following result.

Theorem 2. *Let $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1$ be a transcendental entire function, where f_j $(j = 1, 2, \ldots, N)$ are entire functions with order $\rho_j < 1/2$. Then every nonwandering component is bounded.*

Cao and Wang [20] proved the following result.

Theorem 3. *Let $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1$, where f_j $(j = 1, 2, \ldots, N)$ are nonconstant holomorphic maps, each having order less than 1/2. If there is a number $i \in \{1, 2, \ldots, N\}$ such that the lower order of f_i is greater than 0, then every component of $F(h)$ is bounded.*

Singh [21] proved the following result.

Theorem 4. *Let $\mathscr{L}$ be the set of all entire functions f such that, for given $\varepsilon > 0$,*

$$\log m(r, f) \geq (1 - \varepsilon) \log M(r, f) \tag{3}$$

holds for all r outside a set of logarithmic density 0. Let $\mathscr{F} = \bigcup_{K \geq 1} \mathscr{F}_K$ where $\mathscr{F}_K$ is the set of all transcendental entire functions f such that

$$\log \log M(r, f) \geq (\log r)^{1/K}. \tag{4}$$

If $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1(z)$, where $f_j \in \mathscr{F} \bigcap \mathscr{L}$ $(j = 1, 2, \ldots, N)$, then every component of $F(h)$ is bounded.

2. Preliminaries

We use the standard notations for the maximum modulus $M(r, f)$, minimum modulus $m(r, f)$, order of growth ρ, and lower order of growth μ of a function f; namely,

$$\begin{aligned} M(r, f) &= \max\{|f(z)| : |z| = r\}, \\ m(r, f) &= \min\{|f(z)| : |z| = r\}, \\ \rho &= \limsup_{r \to \infty} \frac{\log \log M(r, f)}{\log r}, \\ \mu &= \liminf_{r \to \infty} \frac{\log \log M(r, f)}{\log r}. \end{aligned} \tag{5}$$

Briefly, we also denote maximum modulus $M(r, f)$ and minimum modulus $m(r, f)$ by $M(r)$ and $m(r)$.

Let E be a set in $\mathbb{C}$. The logarithmic measure of a set E is defined by $\int_E (dt/t)$. If $E \subset [1, \infty)$, $E(a, b)$ denote the part of E in the interval (a, b), that is, $E(a, b) = E \bigcap (a, b)$, then the upper logarithmic density of the set E is defined by

$$\overline{\text{logdens}} E = \limsup_{r \to \infty} \frac{1}{\log r} \int_{E(1,r)} \frac{dt}{t}; \tag{6}$$

the lower logarithmic density of the set E is defined by

$$\underline{\text{logdens}} E = \liminf_{r \to \infty} \frac{1}{\log r} \int_{E(1,r)} \frac{dt}{t}. \tag{7}$$

If the upper and lower logarithmic density are equal, their common value is called the logarithmic density of E.

Lemma 5 (see [22]). *Let f be a transcendental entire function. Then there exists $R > 0$ such that, for all $r \geq R$ and all $c > 1$,*

$$\log M(r^c, f) \geq c \log M(r, f). \tag{8}$$

Lemma 6 (see [23]). *Let f be an entire function of finite order with Fabry gaps. Then for given $\varepsilon > 0$,*

$$\log m(r, f) \geq (1 - \varepsilon) \log M(r, f) \tag{9}$$

holds for all r outside a set of logarithmic density 0.

Lemma 7 (see [24]). *Let $f = \sum_{k=0}^{\infty} a_k z^{n_k}$ be an entire function with*

$$n_k > k \log k (\log \log k)^{\alpha} \quad \text{as } k \longrightarrow \infty \tag{10}$$

for some $\alpha > 2$. Then for given $\varepsilon > 0$,

$$\log m(r, f) \geq (1 - \varepsilon) \log M(r, f) \tag{11}$$

holds for all r outside a set of logarithmic density 0.

Lemma 8 (see [25]). *If $f = \sum_{k=0}^{\infty} a_k z^{n_k}$ satisfies the gap-condition $n_k > k(\log k)^{2+\eta}$, then for given $\varepsilon > 0$,*

$$\log m(r, f) \geq (1 - \varepsilon) \log M(r, f) \tag{12}$$

holds for all r outside a set of finite logarithmic measure.

Lemma 9 (see [26]). *For an entire function $f(z)$ with Fejér gaps and $\varepsilon > 0$,*

$$\log m(r, f) \geq (1 - \varepsilon) \log M(r, f). \tag{13}$$

Lemma 10 (see [9]). *Let D be a domain and K a compact subset of D. Let G be the family of all holomorphic functions g on D which omit the values 0, 1 and satisfy the condition that $|g| \geq 1$ on K. Then there exist constants A and B such that $|g(z')| < A|g(z)|^B$, for every $g \in G$ and every $z, z' \in K$.*

Lemma 11. *Let f be a transcendental entire function of finite order with Fabry gaps. Then there exist $L > 1$ and $R > 0$ such that, for all $r \geq R$, there exists σ satisfying $r \leq \sigma \leq r^L$ and $m(\sigma, f) = M(r, f)$.*

Proof. By Lemma 6, given any $\varepsilon > 0$, we have $m(r, f) > M(r, f)^{1-\varepsilon}$, for all r outside a set E of logarithmic density 0. Let $c > 1$. Then there exists R_0 such that, for all $R > R_0$, there exists $s \in [R, R^c]$ such that $m(s, f) > M(s, f)^{1-\varepsilon}$. If not, then there exists a sequence $R_j \to \infty$ such that $m(s,f) \le M(s,f)^{1-\varepsilon}$, for every $s \in [R_j, R_j^c]$. Thus $[R_j, R_j^c] \subset E$. So

$$\begin{aligned}\underline{\text{logdens}}E &= \liminf_{r\to\infty} \frac{1}{\log r}\int_{E(1,r)} \frac{dt}{t} \\ &\ge \liminf_{j\to\infty} \frac{1}{\log R_j^c}\int_{E(1,R_j^c)} \frac{dt}{t} \\ &\ge \liminf_{j\to\infty} \frac{1}{c\log R_j}\int_{R_j}^{R_j^c} \frac{dt}{t} \\ &= 1-\frac{1}{c},\end{aligned} \tag{14}$$

contradicting logdense$(E) = 0$.

Set $R^{1-2\varepsilon} = r$. There exists $s \in [r^{1/(1-2\varepsilon)}, r^{c/(1-2\varepsilon)}]$ such that $m(s,f) > M(s,f)^{1-\varepsilon}$. By Lemma 6,

$$\begin{aligned} m(s,f) &> M(s,f)^{1-\varepsilon} \ge M\left(r^{1/(1-2\varepsilon)}, f\right)^{1-\varepsilon} \\ &\ge M(r,f)^{(1-\varepsilon)/(1-2\varepsilon)} \ge M(r,f). \end{aligned} \tag{15}$$

Since $m(r,f) \le M(r,f)$, there exists $\sigma \in [r,s] \subset [r, r^L]$, where $L = c/(1-2\varepsilon)$ such that $m(\sigma,f) = M(r,f)$. This completes the proof of Lemma 11. □

By Lemma 9 and the same method in the proof of Lemma 11, we can prove the following result.

Lemma 12. *Let f be a transcendental entire function of finite order with Fejér gaps. Then there exist $L > 1$ and $R > 0$ such that, for all $r \ge R$, there exists σ satisfying $r \le \sigma \le r^L$ and $m(\sigma,f) = M(r,f)$.*

By Lemma 7 and the same method in the proof of Lemma 11, we can prove the following result.

Lemma 13. *Let f be a transcendental entire function of finite order with*

$$n_k > k\log k(\log\log k)^{\alpha} \quad \text{as } k \longrightarrow \infty \tag{16}$$

for some $\alpha > 2$. Then there exist $L > 1$ and $R > 0$ such that, for all $r \ge R$, there exists σ satisfying $r \le \sigma \le r^L$ and $m(\sigma,f) = M(r,f)$.

3. Main Results

In 2012, the authors proved some results on the bounded Fatou components of transcendental entire functions with gaps; see [27]. In this paper, we investigate the iteration of the composite entire functions with gaps and obtain the following results.

Theorem 14. *Let $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1(z)$ be a transcendental entire function, where $f_j = \sum_{k=0}^{\infty} a_{jk} z^{n_{jk}}$ $(j = 1,2,\ldots,N)$ have Fabry gaps of finite order. If*

$$M(r,f_j) \ge \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{17}$$

then $F(h)$ has no unbounded component.

Proof. The proof follows the idea of Theorems 3 and 4. Since f_j is a transcendental entire function of finite order with Fabry gaps, by Lemma 11, there exist $L_j > 1$ and r_j sufficiently large such that, for $r \ge r_j$, there exists $\sigma^{(j)}$ such that $r \le \sigma^{(j)} \le r^{L_j}$ and

$$m\left(\sigma^{(j)}, f_j\right) = M\left(r, f_j\right), \quad j = 1,2,\ldots,N. \tag{18}$$

Since f_j is a transcendental entire function of finite order, h must be a transcendental entire function. It follows that there exists a number $r_0 > 1$ such that $M(r,h) > r^2$ for all $r > r_0$, and there exists a number $t_j > 1$ such that

$$\exp\left(r^{t_j}\right) \ge M\left(r,f_j\right)^{(\log M(r,f_j))^{1/p}} \tag{19}$$

for all r sufficiently large. In fact, if there is a sequence $\{r_n\}$ which tend to ∞ such that

$$\exp\left(r_n^t\right) < M\left(r_n, f_j\right)^{(\log M(r_n,f_j))^{1/p}}, \tag{20}$$

then

$$t\log r_n < \frac{1+p}{p}\cdot \log\log M\left(r_n, f_j\right). \tag{21}$$

So

$$t \le \frac{1+p}{p}\lim_{n\to\infty}\frac{\log\log M\left(r_n,f_j\right)}{\log r_n} \le \frac{1+p}{p}\rho. \tag{22}$$

Which gives a contradiction if we take any $t > ((1+p)/p)\rho$.

Given $R_1^{(1)} > r_0$, define inductively

$$\begin{aligned} R_n^{(j+1)} &= M\left(R_n^{(j)}, f_j\right), \quad j = 1,2,\ldots,N-1; \\ R_{n+1}^{(1)} &= M\left(R_n^{(N)}, f_N\right). \end{aligned} \tag{23}$$

It is easy to see that, for all $n \in \mathbb{N}$,

$$\left(R_n^{(1)}\right)^2 < M\left(R_n^{(1)}, h\right) \le R_{n+1}^{(1)}. \tag{24}$$

Thus

$$R_{n+1}^{(1)} > \left(R_1^{(1)}\right)^{2^n} \longrightarrow \infty \quad \text{as } n \longrightarrow \infty \tag{25}$$

and $R_n^{(j)} \to \infty$ as $n \to \infty$ for all $j = 1,2,\ldots,N$.

Take number n_0 sufficiently large, for $n \ge n_0$, by Lemma 11, there exists $\sigma_n^{(j)}$ such that

$$\begin{aligned} \exp\left(\left(R_n^{(j)}\right)^{t_j}\right) &\le \sigma_n^{(j)} \le \exp\left(L_j\left(R_n^{(j)}\right)^{t_j}\right), \\ m\left(\sigma_n^{(j)}, f_j\right) &= M\left(\exp\left(\left(R_n^{(j)}\right)^{t_j}\right), f_j\right) \end{aligned} \tag{26}$$

for $n \geq n_0$. By the hypotheses of Theorem 14, (19), and Lemma 11, we have

$$\begin{aligned}
& m\left(\sigma_n^{(j)}, f_j\right) \\
& = M\left(\exp\left(\left(R_n^{(j)}\right)^{t_j}\right), f_j\right) \\
& \geq M\left(M\left(R_n^{(j)}, f_j\right)^{(\log M(R_n^{(j)}, f_j))^{1/p}}, f_j\right) \\
& \geq \exp\left(\exp\left(\log M\left(R_n^{(j)}, f_j\right)^{(\log M(R_n^{(j)}, f_j))^{1/p}}\right)^p\right) \\
& = \exp\left(\exp\left(\left(\log M\left(R_n^{(j)}, f_j\right)\right)\left(\log M\left(R_n^{(j)}, f_j\right)\right)^p\right)\right) \\
& = \exp\left(M\left(R_n^{(j)}, f_j\right)^{(\log M(R_n^{(j)}, f_j))^p}\right) \\
& \geq \exp\left(L_{j+1} M\left(R_n^{(j)}, f_j\right)^{t_{j+1}}\right) \\
& = \exp\left(L_{j+1}\left(R_n^{(j+1)}\right)^{t_{j+1}}\right)
\end{aligned} \tag{27}$$

for $j = 1, 2, \ldots, N-1$; and

$$\begin{aligned}
& m\left(\sigma_n^{(N)}, f_N\right) \\
& = M\left(\exp\left(\left(R_n^{(N)}\right)^{t_N}\right), f_N\right) \\
& \geq M\left(M\left(R_n^{(N)}, f_N\right)^{(\log M(R_n^{(N)}, f_N))^{1/p}}, f_N\right) \\
& \geq \exp\left(\exp\left(\log M\left(R_n^{(N)}, f_N\right)^{(\log M(R_n^{(N)}, f_N))^{1/p}}\right)^p\right) \\
& = \exp\left(\exp\left(\left(\log M\left(R_n^{(N)}, f_N\right)\right)\left(\log M\left(R_n^{(N)}, f_N\right)\right)^p\right)\right) \\
& = \exp\left(M\left(R_n^{(N)}, f_N\right)^{(\log M(R_n^{(N)}, f_N))^p}\right) \\
& \geq \exp\left(L_1 M\left(R_n^{(N)}, f_N\right)^{t_1}\right) \\
& = \exp\left(L_1\left(R_{n+1}^{(1)}\right)^{t_1}\right).
\end{aligned} \tag{28}$$

Suppose on contrary that $F(h)$ has an unbounded component D. Without loss of generality we may assume 0, 1 belong to $J(h)$. Hence each map h^n omits the values 0, 1 in D. It follows from the unboundedness and connectivity of D that there exists $n_1 \geq n_0 \in \mathbb{N}$ such that D meets the circles

$$\begin{aligned}
\alpha_n^{(j)} &= \left\{z : |z| = R_n^{(j)}\right\}, \\
\beta_n^{(j)} &= \left\{z : |z| = \exp\left(L_j\left(R_n^{(j)}\right)^{t_j}\right)\right\}, \\
\gamma_n^{(j)} &= \left\{z : |z| = \sigma_n^{(j)}\right\}
\end{aligned} \tag{29}$$

for all $n \geq n_1$, $j = 1, 2, \ldots, N$.

We choose a value $k \in \mathbb{N}$ such that $k \geq n_1$ and note that D must contain a path Γ joining a point $\omega_k^{(1)} \in \alpha_k^{(1)}$ to a point $\tau_{k+1}^{(1)} \in \beta_{k+1}^{(1)}$. It is clear that Γ contains two subsets Γ', Γ'' such that Γ' joins $\omega_k^{(1)} \in \alpha_k^{(1)}$ to $\eta_k^{(1)} \in \beta_k^{(1)}$ and contains $\xi_k^{(1)} \in \gamma_k^{(1)}$ and Γ'' joins $\delta_{k+1}^{(1)} \in \alpha_{k+1}^{(1)}$ to $\tau_{k+1}^{(1)} \in \beta_{k+1}^{(1)}$. We know that $R_k^{(2)} = M(R_k^{(1)}, f_1)$ and so $R_k^{(2)} \geq |f_1(\omega_k^{(1)})|$. Also $m(\sigma_k^{(1)}, f_1) \geq \exp(L_2(R_k^{(2)})^{t_2})$ and so $|f_1(\xi_k^{(1)})| \geq \exp(L_2(R_k^{(2)})^{t_2})$. Hence $f_1(\Gamma')$ contains an arc joining a point $\omega_k^{(2)} \in \alpha_k^{(2)}$ to a point $\eta_k^{(2)} \in \beta_k^{(2)}$. Similarly $f_1(\Gamma'')$ contains an arc joining a point $\delta_{k+1}^{(2)} \in \alpha_{k+1}^{(2)}$ to a point $\tau_{k+2}^{(2)} \in \beta_{k+2}^{(2)}$.

Repeating the process inductively we obtain that $h(\Gamma') = f_N \circ f_{N-1} \circ \cdots \circ f_1(\Gamma')$ contains an arc joining $\omega_{k+1}^{(1)} \in \alpha_{k+1}^{(1)}$ to a point $\eta_{k+1}^{(1)} \in \beta_{k+1}^{(1)}$ and $h(\Gamma'')$ contains an arc joining a point $\delta_{k+2}^{(1)} \in \alpha_{k+2}^{(1)}$ to a point $\tau_{k+2}^{(1)} \in \beta_{k+2}^{(1)}$.

Since Γ' and Γ'' are two subsets of Γ, it follows that $h(\Gamma)$ must contain an arc joining $\omega_{k+1}^{(1)} \in \alpha_{k+1}^{(1)}$ to the point $\tau_{k+2}^{(1)} \in \beta_{k+2}^{(1)}$. By induction it now follows that $h^n(\Gamma)$ contains an arc joining a point $\omega_{k+n}^{(1)} \in \alpha_{k+n}^{(1)}$ to the point $\tau_{k+n+1}^{(1)} \in \beta_{k+n+1}^{(1)}$.

Thus $h^n(D)$ is a component of $F(h)$ containing $h^n(\Gamma)$, and, on Γ, h^n takes a value of modulus at least $R_{k+n}^{(1)}$ and $R_{k+n}^{(1)} \to \infty$ as $n \to \infty$. Thus we conclude that $R_{k+n}^{(1)} \to \infty$ locally uniformly in D. Hence there exists $N \in \mathbb{N}$ such that, for all $z \in \Gamma$, $|h^n(z)| > 1$ for all $n > N$. Thus the family $\{h^n\}_{n>N}$ satisfies the conditions of Lemma 10 on Γ, and so there exist constants A, B such that $|h^n(z')| < A|h^n(z)|^B$ for all $n > N$ and for all $z, z' \in \Gamma$. Choose $z_n, z_n' \in \Gamma$ with $n > N$ such that $h^n(z_n) = \omega_{k+n}^{(1)} \in \alpha_{k+n}^{(1)}$ and $h^n(z_n') = \tau_{k+n+1}^{(1)} \in \beta_{k+n+1}^{(1)}$; we have

$$\begin{aligned}
M\left(R_{k+n}^{(1)}, h\right) &= M\left(R_{k+n}^{(1)}, f_N \circ f_{N-1} \circ \cdots \circ f_1\right) \\
&\leq M\left(R_{k+n}^{(N)}, f_N\right) = R_{k+n+1}^{(1)} \\
&< \exp\left(L_1\left(R_{k+n+1}^{(1)}\right)^{t_1}\right) = \left|h^n\left(z_n'\right)\right| \\
&< A\left|h^n\left(z_n\right)\right|^B = A\left|R_{k+n}^{(1)}\right|^B
\end{aligned} \tag{30}$$

for all $n > N$ which contradicts the fact that h is a transcendental entire function and $R_{k+n}^{(1)} \to \infty$ as $n \to \infty$. This completes the proof of Theorem 14. □

Corollary 15 (see [27]). *Let $f(z) = \sum_{k=0}^{\infty} a_k z^{n_k}$ be a transcendental entire function of finite order with Fabry gaps. If*

$$M(r, f) \geq \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{31}$$

then $F(f)$ has no unbounded component.

Remark 16. If f satisfies $M(r, f) \geq \exp(\exp(\log r)^p)$, $0 < p < 1$, then $\rho \geq 0$, so Corollary 15 is an extension of Theorem 1.

By Lemma 12 and the same method used in the proof of Theorem 14, we can show the following result.

Theorem 17. *Let* $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1(z)$ *be a transcendental entire function, where* $f_j = \sum_{k=0}^{\infty} a_{jk} z^{n_{jk}}$ $(j = 1, 2, \ldots, N)$ *have Fejér gaps. If*

$$M(r, f_j) \geq \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{32}$$

then $F(h)$ *has no unbounded component.*

By Theorem 17, we have the following.

Corollary 18 (see [27]). *Let* $f(z) = \sum_{k=0}^{\infty} a_k z^{n_k}$ *be a transcendental entire function with Fejér gaps. If*

$$M(r, f) \geq \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{33}$$

then $F(f)$ *has no unbounded component.*

Remark 19. Corollary 18 partly answers the problem of Wang on the Fejér gaps.

By Lemma 13 and the same method of the proof of Theorem 14, we can show Theorem 20.

Theorem 20. *Let* $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1(z)$ *be a transcendental entire function, where* $f_j = \sum_{k=0}^{\infty} a_{jk} z^{n_{jk}}$ $(j = 1, 2, \ldots, N)$ *are entire functions with the gap-conditions*

$$n_{jk} > k \log k (\log\log k)^{\alpha} \quad \text{as } k \longrightarrow \infty \text{ for some } \alpha > 2. \tag{34}$$

If

$$M(r, f_j) \geq \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{35}$$

then $F(h)$ *has no unbounded component.*

By Theorem 20, we have the following.

Corollary 21. *Let* $h(z) = f_N \circ f_{N-1} \circ \cdots \circ f_1(z)$ *be a transcendental entire function, where* $f_j = \sum_{k=0}^{\infty} a_{jk} z^{n_{jk}}$ $(j = 1, 2, \ldots, N)$ *are entire functions with the gap-conditions* $n_{jk} > k(\log k)^{2+\eta}$ *for some* $\eta > 0$*. If*

$$M(r, f_j) \geq \exp\left(\exp(\log r)^p\right), \quad 0 < p < 1, \tag{36}$$

then $F(h)$ *has no unbounded component.*

Conflict of Interests

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

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 11261002), the Natural Science Foundation of Yunnan Province of China (Grant no. 2012FZ167), and the Educational Commission of Yunnan Province of China (Grant no. 2012z121).

References

[1] P. Fatou, "Sur les equations fonctionnelles," *Bulletin de la Société Mathématique de France*, vol. 47, pp. 161–271, 1919.

[2] P. Fatou, "Sur les equations fonctionnelles," *Bulletin de la Société Mathématique de France*, vol. 48, pp. 33–94, 1920.

[3] P. Fatou, "Sur l'iteration des fonctions transcendents entiers," *Acta Mathematica*, vol. 47, no. 4, pp. 337–370, 1926.

[4] G. Julia, "Memoire sur l'iteration des fonctions rationnelles," *Journal de Mathématiques Pures et Appliquées*, vol. 8, pp. 47–245, 1918.

[5] A. F. Beardon, *Iteration of Rational Functions*, vol. 132 of *Graduate Texts in Mathematics*, Springer, New York, NY, USA, 1991.

[6] L. Carleson and T. W. Gamelin, *Complex Dynamics*, Springer, New York, NY, USA, 1993.

[7] J. Milnor, *Dynamics in One Complex Variable: Introductory Lectures*, Vieweg, Berlin, Germany, 2nd edition, 2000.

[8] F. Y. Ren, *Complex Analysis Dynamics*, Fudan University Press, Shanghai, China, 1996, (Chinese).

[9] I. N. Baker, "The iteration of polynomials and transcendental entire functions," *The Australian Mathematical Society (Series A)*, vol. 30, no. 4, pp. 483–495, 1981.

[10] J. M. Anderson and A. Hinkkanen, "Unbounded domains of normality," *Proceedings of the American Mathematical Society*, vol. 126, no. 11, pp. 3243–3252, 1998.

[11] A. Hinkkanen, "Entire functions with bounded Fatou components," in *Transcendental Dynamics and Complex Analysis*, pp. 187–216, Cambridge University Press, 2008.

[12] A. Hinkkanen, "Entire functions with no unbounded Fatou component," in *Complex Analysis and Dynamical Systems II*, vol. 382 of *Contemporary Mathematics*, pp. 217–226, American Mathematical Society, Providence, RI, USA, 2005.

[13] Y. Wang, "Bounded domains of the Fatou set of an entire function," *Israel Journal of Mathematics*, vol. 121, pp. 55–60, 2001.

[14] J. H. Zheng, "Unbounded domains of normality of entire functions of small growth," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 128, no. 2, pp. 355–361, 2000.

[15] 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.

[16] Y. Yin, M. Liu, J. Hao, and M. Zhou, "Single-machine scheduling with job-position-dependent learning and time-dependent deterioration," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 42, no. 1, pp. 192–200, 2012.

[17] 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.

[18] Y. F. Wang, "On the Fatou set of an entire function with gaps," *The Tohoku Mathematical Journal*, vol. 53, no. 1, pp. 163–170, 2001.

[19] J. Y. Qiao, "Stable set for iteration of entire functions," *Acta Mathematica Sinica*, vol. 37, no. 5, pp. 702–708, 1994 (Chinese).

[20] C. L. Cao and Y. F. Wang, "Boundedness of Fatou components of holomorphic maps," *Journal of Dynamics and Differential Equations*, vol. 16, no. 2, pp. 377–384, 2004.

[21] A. P. Singh, "Composite entire functions with no unbounded Fatou components," *Journal of Mathematical Analysis and Applications*, vol. 335, no. 2, pp. 907–914, 2007.

[22] P. J. Rippon and G. M. Stallard, "Functions of small growth with no unbounded Fatou components," *Journal d'Analyse Mathématique*, vol. 108, pp. 61–86, 2009.

[23] W. H. Fuchs, "Proof of a conjecture of G. Póya concerning gap series," *Illinois Journal of Mathematics*, vol. 7, pp. 661–667, 1963.

[24] W. K. Hayman, "Angular value distribution of power series with gaps," *Proceedings of the London Mathematical Society*, vol. 24, pp. 590–624, 1972.

[25] T. Kovari, "A gap-theorem for entire functions of infinite order," *The Michigan Mathematical Journal*, vol. 12, no. 2, pp. 133–140, 1965.

[26] T. Murai, "The deficiency of entire functions with Fejér gaps," *Annales de l'Institut Fourier*, vol. 33, no. 3, pp. 39–58, 1983.

[27] C. J. Yang and Y. H. Li, "Bounded Fatou components of transcendental entire functions with gaps," *Acta Mathematica Sinica*, vol. 55, no. 1, pp. 179–186, 2012 (Chinese).

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 805294, 7 pages
http://dx.doi.org/10.1155/2015/805294

Research Article

Online Scheduling on a Single Machine with Grouped Processing Times

Qijia Liu,[1] Long Wan,[2] and Lijun Wei[2]

[1] *School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China*
[2] *School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China*

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

Received 17 October 2014; Accepted 10 November 2014

Academic Editor: Shuenn-Ren Cheng

We consider the online scheduling problem on a single machine with the assumption that all jobs have their processing times in $[p, (1+\alpha)p]$, where $p > 0$ and $\alpha = (\sqrt{5}-1)/2$. All jobs arrive over time, and each job and its processing time become known at its arrival time. The jobs should be first processed on a single machine and then delivered by a vehicle to some customer. When the capacity of the vehicle is infinite, we provide an online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$. When the capacity of the vehicle is finite, that is, the vehicle can deliver at most c jobs at a time, we provide another best possible online algorithm with the competitive ratio of $(\sqrt{5}+1)/2$.

1. Introduction and Problem Formulation

Production and distribution operations are two important segments in a supply chain; it is critical to integrate these two segments in a scheduling problem. In the last few years, scheduling problem with job delivery coordination is a very important topic. For the offline version, the earliest research of this topic is the one by Potts [1]. They considered the single machine scheduling problem to minimize the maximum delivery completion time and provided a heuristic algorithm with the worst-case performance ratio of 3/2. For more papers, the reader may refer to [2–6].

For the online scheduling, there are three models which are commonly considered [7]. The first one assumes that there exists no release date and the jobs arrive in a list. The second one assumes that the processing time of each job is unknown until the job finishes. The third one assumes that each job arrives over time. When the machine is idle, an online algorithm decides which one of the available jobs is scheduled, if any. In this paper, we study the third model in which jobs arrive over time. For an online algorithm H, we use the competitive ratio

$$\rho_H = \sup\left\{\frac{H(I)}{\mathrm{OPT}(I)} : I \text{ is an instance with } \mathrm{OPT}(I) > 0\right\} \quad (1)$$

to measure the performance of H, where, for an instance I, $H(I)$ is used to denote the objective value of the schedule generated by the online algorithm H and $\mathrm{OPT}(I)$ is the objective value of an offline optimal schedule.

For the online version of scheduling problem with job delivery, Hoogeveen and Vestjens [8] considered the online single machine scheduling problem. They provided a best possible online algorithm with the competitive ratio of $(\sqrt{5}+1)/2$. When restarts are allowed, van den Akker et al. [9] proposed a best possible online algorithm with the competitive ratio of 3/2. The online integrated production-distribution scheduling problems to minimize the sum of the total weighted flow time and the total delivery cost were considered by Averbakh [10]. For several cases of the problem, the author provided efficient online algorithms and used competitive analysis to study their worst-case performance ratio. Yuan et al. [11] studied a single batch machine online scheduling problem with restricted delivery times; that is, the delivery time of each job is less than or equal to its processing time. They provided a best possible online algorithm with competitive ratio of $(\sqrt{5}+1)/2$. For the case in which the delivery time of each job is greater than or equal to its processing time, Tian et al. [12] proposed a best possible online algorithm with competitive ratio of

$(\sqrt{5}+1)/2$. Liu et al. [13] studied online scheduling problem on a single machine with bounded distribution. They proved a lower bound of competitive ratios for all online algorithms and proposed a best possible online algorithm with a competitive ratio of $(1/2)(\sqrt{5+\beta^2+2\beta}+1-\beta)$. Liu et al. [14] studied online scheduling problems on a single machine with deteriorating jobs. For each model, they presented an optimal online algorithm.

Meanwhile, the scheduling problems in which the jobs have the equal length processing times are widely considered; the reader may refer to [15–17]. However, it is more reasonable to suppose that the jobs have their processing times being in a given interval. In semiconductor manufacturing integrated circuits are produced through the same technical processes. Therefore they have their processing times in a given interval. Fang et al. [18] studied online scheduling problems on a batch machine with the assumption that all jobs have their processing times in $[p,(1+\alpha)p]$, where $p>0$ and $\alpha=(\sqrt{5}-1)/2$. For the two scheduling problems, they provided best possible online algorithms with the competitive ratio of $(\sqrt{5}+1)/2$.

In this paper, we study the single machine online scheduling problem with job delivery in which the jobs have their processing times in $[p,(1+\alpha)p]$, where $p>0$ and $\alpha=(\sqrt{5}-1)/2$. The jobs should be first processed on a single machine and then delivered by a vehicle to some customer. When the capacity of the vehicle is infinite, we provide an online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$. When the capacity of the vehicle is finite and fixed, that is, the vehicle can deliver at most c jobs to the customer at a time and $c\geq 2$, we provide another online algorithm with the best competitive ratio of $(\sqrt{5}+1)/2$.

2. Problem Formulation and Preliminaries

In the online scheduling problem on a single machine to minimize the maximum delivery completion time, there are n jobs $J_1,\dots,J_n$ which arrive over time. We do not know any information of these jobs in advance, including the number of the jobs and processing time p_j and arrival time r_j of each job J_j. The jobs are first processed on the single machine. Once the vehicle is available, the completed jobs can be delivered immediately in batches by the vehicle to some customer. Let T be the round-trip transportation between the machine and the customer. Since the customer is unknown in advance, we assume that T is known immediately once the first job arrives. The problem can be denoted by $1 \to D \mid \text{online}, r_j, v=1, c, p_j \in [p,(1+\alpha)p] \mid D_{\max}$. Here "$1\to D$" means that jobs are first processed on a single machine and then the completed jobs must be delivered in batches to the customer; "$v=1$" means that there is one vehicle to deliver the jobs. "c" means the capacity of the vehicle; "$p_j\in[p,(1+\alpha)p]$" means that the jobs have their processing times in $[p,(1+\alpha)p]$. We use D_j to denote the time at which the vehicle transports J_j to the customer and returns to the machine. $D_{\max}=\max_{j=1}^{n}\{D_j\}$ denotes the time when the vehicle finishes delivering the last delivery batch to the destination and returns to the machine.

Indeed, when $c=1$ and T is known in advance, Ng and Lu [19] provided an optimal online algorithm for the general case in which the processing time of each job is a nonnegative number. When $c\geq 2$ and T is known in advance, they provided an online algorithm which is not best possible. However, in this paper, we assume that $c\geq 2$ and T is unknown in advance. We deal with two variants of the online scheduling problem: $1\to D\mid \text{online}, r_j, v=1, c=\infty, p_j\in[p,(1+\alpha)p]\mid D_{\max}$ when the capacity of the vehicle is infinite and $1\to D\mid \text{online}, r_j, v=1, c<\infty, p_j\in[p,(1+\alpha)p]\mid D_{\max}$ when the capacity of the vehicle is finite and fixed.

Let σ be a schedule generated by an online algorithm H. Let B_k be a delivery batch in σ. Then we define the following notation:

(i) $s_j(\sigma)$, the starting time of J_j on the machine in σ;

(ii) $c_j(\sigma)$, the completion time of J_j on the machine in σ;

(iii) $s(B_k)=\min\{s_j(\sigma): J_j\in B_k\}$;

(iv) $c(B_k)=\max\{c_j(\sigma): J_j\in B_k\}$;

(v) $p(B_k)=\sum_{J_j\in B_k} p_j$;

(vi) $U(t)$, the set of the jobs which are available and unprocessed at time t;

(vii) $A(t)$, the set of jobs which are completed on the machine and are waiting for the vehicle to be delivered at time t;

(viii) $\rho(B_k)$, the ready time of delivery batch B_k, which is the maximum completion time of the jobs assigned to batch B_k; in fact $\rho(B_k)=c(B_k)$;

(ix) $\delta(B_k)$, the departure time for the vehicle to delivery batch B_k from the machine to the customer; note that $\delta(B_k)\geq\rho(B_k)$ in any feasible solution;

(x) $D(B_k)=\delta(B_k)+T$, the delivery completion time of the delivery batch B_k;

(xi) we say that a set of delivery batches $\mathscr{B}=\{B_1,\dots,B_k\}$ in this order is continuously delivered in a schedule if the time interval $[\delta(B_1),D(B_k)]$ is totally occupied by the vehicle for delivering the batches in $\mathscr{B}$; equivalently, we have $\delta(B_i)=D(B_{i-1})=\delta(B_{i-1})+T$ for $2\leq i\leq k$;

(xii) suppose that $\mathscr{B}'=\{B'_1,\dots,B'_k\}$ in this order is the set of delivery batches in a schedule; we say that there exists an idle time of the vehicle directly before $\delta(B'_i)$ in a schedule if $D(B'_{i-1})<\delta(B'_i)$, for $2\leq i\leq k$; similarly, we say that there exists no idle time of the vehicle directly before $\delta(B'_i)$ in a schedule if $D(B'_{i-1})=\delta(B'_i)$, for $2\leq i\leq k$.

3. A Lower Bound

We assume that $c\geq 2$. Let ε be a sufficiently small positive number that tends to zero. Let N be a sufficiently large integer. For any online algorithm H, we consider the following job instance generated by the adversary. Let $D^H_{\max}$ and $D^*_{\max}$ be the objective values obtained by algorithm H and optimal offline algorithm, respectively. All jobs in the instance have a processing time p. Let $T = Np$. At time 0, the adversary releases a job J_1. Assume that algorithm H starts processing J_1 at time t. If $t \geq \alpha(p+T)$, then no other jobs arrive. Clearly, we have $D^H_{\max} \geq \alpha(p+T) + p + T$. However, the optimal schedule is to process J_1 at time 0 and deliver it at time p. Then we have $D^*_{\max} = p + T$. Thus,

$$\frac{D^H_{\max}}{D^*_{\max}} \geq \frac{\alpha(p+T) + p + T}{p+T} = 1 + \alpha. \tag{2}$$

Suppose in the following that $t < \alpha(p+T)$; we assume that the vehicle delivers J_1 at time s. If $s \geq \alpha(p+T)$, then no other jobs arrive. Then we have

$$\frac{D^H_{\max}}{D^*_{\max}} \geq \frac{\alpha(p+T) + T}{p+T} = \frac{\alpha(N+1)p + Np}{(N+1)p} \longrightarrow 1 + \alpha, \tag{3}$$

when $N \to +\infty$. If $s < \alpha(p+T)$, then J_2 arrives at time $s+\varepsilon$. Clearly, we have $D^H_{\max} \geq s+T+T$. Note that $s \geq p$. The optimal schedule is to process J_2 at time $s+\varepsilon$ and then to deliver J_1 and J_2 together; that is, $D^*_{\max} = s + \varepsilon + p + T$. Thus, we have

$$\begin{aligned} \frac{D^H_{\max}}{D^*_{\max}} &\geq \frac{s+T+T}{s+\varepsilon+p+T} \longrightarrow \frac{s+T+T}{s+p+T} \\ &= 1 + \frac{T-p}{s+p+T} > 1 + \frac{(N-1)p}{(\alpha+1)(N+1)p} \\ &\longrightarrow 1 + \frac{1}{1+\alpha} = 1 + \alpha, \end{aligned} \tag{4}$$

when $\varepsilon \to 0$ and $N \to +\infty$.

The above discussion implies the following two lemmas.

Lemma 1. *For the problem $1 \to D \mid online, r_j, v = 1, c = \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$, there exists no online algorithm with a competitive ratio of less than $1+\alpha$.*

Lemma 2. *For the problem $1 \to D \mid online, r_j, v = 1, c < \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$, where $c \geq 2$, there exists no online algorithm with a competitive ratio of less than $1+\alpha$.*

4. The Case of $c = \infty$

In this section, we consider the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$, where the capacity of the vehicle is infinite; that is, all jobs can be delivered simultaneously in the same batch.

Algorithm D^∞

Processing Stage. At time t, if the machine is available and $U(t) \neq \emptyset$, we choose the job of $U(t)$ with the shortest processing time to process at time t. Otherwise, do nothing but wait.

Delivery Stage. At time t with $A(t) \neq \emptyset$, we consider $A(t)$ as a delivery batch. If the machine is available, $t \geq \alpha T$, and $U(t) = \emptyset$, we deliver the delivery batch $A(t)$ at time t. Otherwise, do nothing but wait.

Indeed, the algorithm of the processing stage of D^∞ is optimal for the problem $1 \mid \text{online}, r_j \mid C_{\max}$, because the algorithm cannot produce unnecessary idle time on the machine. For the delivery stage, the machine is available which means that there is not a job which is processing at this time and $U(t) = \emptyset$ means that there is not a job which is available and unprocessed at time t. That is, only if there is not an available job which is unprocessed or processing, the vehicle determines to deliver.

Let σ and π be the schedules that are generated by algorithm D^∞ and optimal offline algorithm, respectively. Let $D_{\max}(\sigma)$ and $D_{\max}(\pi)$ be the objective values obtained from the schedule σ and the optimal offline schedule π, respectively. Let $C_{\max}(\sigma)$ and $C_{\max}(\pi)$ be the makespans of the schedule σ and the schedule π, respectively. For the problem $1 \mid \text{online}, r_j \mid C_{\max}$, let $C^*_{\max}$ be the optimal offline makespan. Next, we get the following lemma.

Lemma 3. *Consider the following*

$$C^*_{\max} = C_{\max}(\sigma). \tag{5}$$

Proof. Since the algorithm of processing stage of D^∞ cannot produce an unnecessary idle time on the machine, that is, the machine is always busy as long as there are jobs which are available and unprocessed, we have $C^*_{\max} = C_{\max}(\sigma)$. □

As the capacity of the vehicle is infinite, we have the following lemma.

Lemma 4. *Consider the following*

$$D_{\max}(\pi) \geq C_{\max}(\pi) + T \geq C^*_{\max} + T \geq r_l + T. \tag{6}$$

Theorem 5. *The competitive ratio of algorithm D^∞ is at most $1+\alpha$; that is, $D_{\max}(\sigma)/D_{\max}(\pi) \leq 1+\alpha$.*

Proof. Let $B_1, \ldots, B_k$ be the delivery batches in σ. Set $n_B = |\{B_1, \ldots, B_k\}|$. Assume that the last arrival time of the jobs is r_l. If $\delta(B_k) = \alpha T$, then we have $n_B = 1$. Thus $D_{\max}(\sigma) = \delta(B_k) + T \leq (1+\alpha)D_{\max}(\pi)$, as required. Suppose in the following that $\delta(B_k) > \alpha T$. If $\delta(B_k) = C_{\max}(\sigma)$, from Lemmas 3 and 4, we have

$$D_{\max}(\sigma) = C_{\max}(\sigma) + T = C^*_{\max} + T \leq D_{\max}(\pi). \tag{7}$$

If $\delta(B_k) \neq C_{\max}(\sigma)$, that is, the delivery batch B_k follows immediately behind the delivery batch B_{k-1}, then we have $\delta(B_k) = \delta(B_{k-1}) + T$. Thus,

$$D_{\max}(\sigma) = \delta(B_{k-1}) + 2T. \tag{8}$$

As the capacity of the vehicle is infinite, by algorithm D^∞, we have $r_l > \delta(B_{k-1})$. According to the algorithm D^∞, we have $\delta(B_j) \geq \alpha T$, for all $1 \leq j \leq k$. Then

$$D_{\max}(\pi) \geq r_l + T > \delta(B_{k-1}) + T \geq \alpha T + T. \tag{9}$$

Thus, we have $T \leq \alpha D_{\max}(\pi)$. From Lemma 4, (8) can be rewritten as

$$\begin{aligned} D_{\max}(\sigma) &= \delta(B_{k-1}) + 2T < r_l + 2T \\ &\leq D_{\max}(\pi) + \alpha D_{\max}(\pi). \end{aligned} \tag{10}$$

The result follows. □

Theorem 5 and Lemma 1 imply that D^{∞} is a best possible online algorithm for the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$. Indeed, according to the proof of Theorem 5, D^{∞} is also a best possible online algorithm for the problem $1 \to D \mid \text{online}, r_j, v = 1, c = \infty \mid D_{\max}$.

5. The Case of $c < \infty$

In this section, we consider the problem $1 \to D \mid \text{online}, r_j, v = 1, c < \infty, p_j \in [p, (1+\alpha)p] \mid D_{\max}$, where the capacity of the vehicle is finite and fixed; that is, the vehicle can deliver at most c jobs in the same batch to some customer at a time.

Algorithm D^c

Processing Stage. At time t, if the machine is available and $U(t) \neq \emptyset$, we choose the job of $U(t)$ with the shortest processing time to process at time t. Otherwise, do nothing but wait.

Delivery Stage

Step 0. If the vehicle is idle, $t \geq \alpha T$, and $A(t) \neq \emptyset$, determine $|A(t)|$ to be the number of the jobs of $A(t)$.

Step 1. If $|A(t)| \geq c$, select the earliest completed c jobs in $A(t)$ as a delivery batch and deliver this batch at time t.

Step 2. If $0 < |A(t)| < c$, then go to the following steps.

Step 2.A. If the machine is available and $U(t) = \emptyset$, that is, there is not a job which is unprocessed or processing at time t, we consider jobs in $A(t)$ as a delivery batch and deliver this batch at time t.

Step 2.B. If the machine is busy or $U(t) \neq \emptyset$, that is, there exists a job which is unprocessed or processing at time t, wait until the machine is available and $U(t) = \emptyset$ or the next arrival.

Step 3. Go to Step 0.

Similarly, the algorithm of the processing stage of D^c is optimal for the problem $1 \mid \text{online}, r_j \mid C_{\max}$, because the algorithm cannot produce unnecessary idle time. For the delivery stage, the machine is available which means that there is not a job which is processing at this time and $U(t) = \emptyset$ means that there is not a job which is available and unprocessed at time t. That is, only if there is not a job which is unprocessed or processing the vehicle determines to deliver. Meanwhile, the algorithm implies that the job with a smaller completion time is delivered no later than that with a larger completion time.

Let μ and π be the schedules that are generated by algorithm D^c and optimal offline algorithm, respectively. Let $D_{\max}(\mu)$ and $D_{\max}(\pi)$ be the objective values obtained from the schedule μ and the optimal offline schedule π, respectively. Let $C_{\max}(\mu)$ and $C_{\max}(\pi)$ be the makespans of the schedule μ and the schedule π, respectively. For the problem $1 \mid \text{online}, r_j \mid C_{\max}$, let $C^*_{\max}$ be the optimal offline makespan.

As the algorithm of processing stage of D^{∞} is same as the algorithm of processing stage of D^c, we also have the following lemma.

Lemma 6. *Consider the following*

$$C^*_{\max} = C_{\max}(\mu). \tag{11}$$

As the capacity of the vehicle is finite and fixed, there is at least one delivery batch. Let n be the number of the jobs in the schedule. Let $k^* = \lceil n/c \rceil$. Then there are at least k^* delivery batches in any feasible schedule. Thus the following lemma can be observed.

Lemma 7. *Consider the following*

$$D_{\max}(\pi) \geq \max\{C_{\max}(\pi) + T, p + k^* T\}. \tag{12}$$

A delivery batch is called *full* if it contains exactly c jobs. Otherwise, it is *nonfull*. Let $B_1, \ldots, B_k$ be the delivery batches in μ. Assume that the last arrival time of the jobs is r_l.

Lemma 8. *If $T \leq p$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. As $T \leq p$, we have $D_{\max}(\mu) = C_{\max}(\mu) + T$. From Lemmas 6 and 7, we have

$$\begin{aligned} D_{\max}(\mu) &= C_{\max}(\mu) + T \\ &= C^*_{\max} + T \\ &\leq C_{\max}(\pi) + T \leq D_{\max}(\pi). \end{aligned} \tag{13}$$

The lemma follows. □

Lemma 9. *If $T > p$ and $C_{\max}(\mu) \leq \alpha T$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. Note that $C_{\max}(\mu) \leq \alpha T$. According to the algorithm of delivery stage of D^c, we have $\delta(B_1) = \alpha T$ and $\delta(B_j) > \alpha T$ for all $2 \leq j \leq k$. Then we get $k = k^*$. Thus,

$$D_{\max}(\mu) = \delta(B_1) + kT = \alpha T + k^* T \leq (1+\alpha) D_{\max}(\pi). \tag{14}$$

The lemma follows. □

Lemma 10. *If $T > p$ and $C_{\max}(\mu) > \alpha T$, then $D_{\max}(\mu)/D_{\max}(\pi) \leq 1 + \alpha$.*

Proof. Let B_t be the earliest delivery batch such that $B_t, \ldots, B_k$ are continuous delivery batch in μ. If $B_t = B_k$, as $C_{\max}(\mu) > \alpha T$, we have $\delta(B_k) = C_{\max}(\mu)$; then $D_{\max}(\mu) = C_{\max}(\mu) + T \leq D_{\max}(\pi)$, as required.

If $B_t \neq B_k$ and $\delta(B_{k-1}) \leq C_{\max}(\mu)$, as $C_{\max}(\mu) > \alpha T$, we have $C_{\max}(\pi) \geq C^*_{\max} = C_{\max}(\mu) > \alpha T$. Then

$$D_{\max}(\pi) \geq C_{\max}(\pi) + T \geq (\alpha + 1) T. \tag{15}$$

Thus, we have $D_{\max}(\mu) = \delta(B_{k-1}) + 2T \leq C_{\max}(\mu) + 2T \leq D_{\max}(\pi) + T \leq (1+\alpha) D_{\max}(\pi)$, as required. Suppose in the following that $B_t \neq B_k$ and $\delta(B_{k-1}) > C_{\max}(\mu)$. Then B_{k-1} follows immediately behind the delivery batch B_{k-2}, and so $|\{B_t, \ldots, B_k\}| \geq 3$. Now, there are two possibilities to be discussed.

Case 1. There are some nonfull delivery batches in $B_t, \ldots, B_{k-1}$. Denote the last nonfull delivery batch before B_k by B_s, where $t \leq s \leq k-1$. Obviously, we have $\delta(B_s) \geq \alpha T$ and $D_{\max}(\mu) = \delta(B_s) + (k-s+1)T$. As B_s is nonfull, by algorithm D^c, we have $U(\delta(B_s)) = \emptyset$ and there is not a job which is processing at time $\delta(B_s)$. Then, for each $s+1 \leq i \leq k$, the jobs of B_i arrive after $\delta(B_s)$. Note that $B_{s+1}, \ldots, B_{k-1}$ are full delivery batches. Thus, we have

$$D_{\max}(\pi) \geq \delta(B_s) + (k-s)T. \tag{16}$$

From (15) and (16), we have

$$\begin{aligned} D_{\max}(\mu) &= \delta(B_s) + (k-s+1)T \\ &\leq D_{\max}(\pi) + T \leq (1+\alpha) D_{\max}(\pi), \end{aligned} \tag{17}$$

as required.

Case 2. In μ, all the delivery batches $B_t, \ldots, B_{k-1}$ are full. As $|\{B_t, \ldots, B_k\}| \geq 3$, we have $k^* \geq 3$. From Lemma 7, we have $T \leq (1/3)D_{\max}(\pi)$.

Case 2.1 ($\delta(B_t) = \alpha T$). Then $B_t = B_1$ and $D_{\max}(\mu) = \delta(B_1) + kT$. As all the delivery batches $B_t, \ldots, B_{k-1}$ are full, we have $k = k^*$. Then $D_{\max}(\mu) = \delta(B_1) + kT = \alpha T + k^* T \leq (1+\alpha)D_{\max}(\pi)$, as required.

Case 2.2 ($\delta(B_t) > \alpha T$). The jobs in B_t are processed on the machine in this interval $[s(B_t), c(B_t)]$. Note that B_t is the earliest delivery batch such that $B_t, \ldots, B_k$ are continuous delivery batch in μ. Then there exists an idle time of the vehicle directly before $\delta(B_t)$. Then $\delta(B_t) = \rho(B_t) = c(B_t)$. By algorithm D^c, the job with a smaller completion time is delivered no later than that with a larger completion time; then $s(B_{t+1}) \geq c(B_t)$. Noting that $B_t, \ldots, B_k$ are continuous delivery batch, we have $\delta(B_{t+1}) = \delta(B_t) + T$. However, $p_j \in [p, (1+\alpha)p]$, for all $1 \leq j \leq n$, and B_{t+1} is full; we have

$$\begin{aligned} \delta(B_t) + T = \delta(B_{t+1}) &\geq c(B_{t+1}) \\ &\geq s(B_{t+1}) + cp \geq c(B_t) + cp. \end{aligned} \tag{18}$$

Thus, we have

$$p \leq \frac{T}{c}. \tag{19}$$

If there exists an idle time in $[s(B_t), c(B_t)]$ on the machine in μ, assume that s_i is the earliest time such that there exists no idle time in $[s_i, c(B_t)]$, where $s_i > s(B_t)$. Then s_i is the arrival time of some job. By algorithm D^c, all jobs of $B_{t+1}, \ldots, B_k$ arrive at or after s_i. Assume that there are m jobs which are processed in $[s_i, c(B_t)]$ on the machine, where $m < c$. Noting that $B_{t+1}, \ldots, B_{k-1}$ are full, then there are at least $m + (k-t-1)c + 1$ jobs which arrive at or after s_i. In fact, these jobs need at least $k-t$ delivery batches to be delivered. Thus, we have

$$D_{\max}(\pi) \geq s_i + mp + (k-t)T. \tag{20}$$

As $p_j \in [p, (1+\alpha)p]$, for all $1 \leq j \leq n$, we have

$$\begin{aligned} D_{\max}(\mu) &= \delta(B_t) + (k-t+1)T \\ &\leq s_i + m(1+\alpha)p + (k-t+1)T. \end{aligned} \tag{21}$$

From (19), (20), and (21), we have

$$\begin{aligned} D_{\max}(\mu) - D_{\max}(\pi) &\leq m\alpha p + T \leq \alpha T + T \\ &\leq \frac{1+\alpha}{3} D_{\max}(\pi) < \alpha D_{\max}(\pi), \end{aligned} \tag{22}$$

as required.

If there exists no idle time in $[s(B_t), c(B_t)]$ on the machine in μ, assume that s_0 is the earliest time such that there exists no idle time in $[s_0, c(B_t)]$. Obviously, $s_0 \leq s(B_t)$ and s_0 is the arrival time of some job. By algorithm D^c, all jobs of $B_{t+1}, \ldots, B_k$ arrive at or after s_0. Assume that there are n' jobs which arrive at or after s_0 and there are z jobs which are processed on the machine in $[s_0, c(B_t)]$, where $z \geq c$. Then there are $n' - z$ jobs which are processed on the machine in $[c(B_t), C_{\max}(\mu)]$. As $p_j \in [p, (1+\alpha)p]$, we have

$$D_{\max}(\mu) \leq s_0 + z(1+\alpha)p + (k-t+1)T. \tag{23}$$

Let $n' = k'c + \nu$, where $n' \leq n$ and $k' \leq k^* - 1$. Then

$$D_{\max}(\pi) \geq s_0 + \nu p + \left(k' + 1\right)T. \tag{24}$$

In addition, $n' - z$ jobs which are processed in $[c(B_t), C_{\max}(\mu)]$ are delivered by $k-t$ delivery batches, because these jobs are delivered in $B_{t+1}, \ldots, B_k$. Then $\lceil (n'-z)/c \rceil = k-t$; that is, $k-t-1 < (n'-z)/c \leq k-t$. Hence, $k'c + \nu - z = n' - z > c(k-t-1)$; that is,

$$c(k-t-1) - \left(k'c + \nu\right) < -z. \tag{25}$$

On the other hand, as $|\{B_t, \ldots, B_k\}| \geq 3$, we have $|\{B_{t+1}, \ldots, B_k\}| \geq 2$; then $c + 1 \leq n' - z = k'c + \nu - z$; that is,

$$\frac{z}{c} < k' + \frac{\nu}{c} - 1. \tag{26}$$

Now, from (23), (24), and (25), we have

$$\begin{aligned} &D_{\max}(\mu) - D_{\max}(\pi) \\ &\quad = z(1+\alpha)p - \nu p + (k-t+1)T - \left(k'+1\right)T \\ &\quad = (z-\nu)p + \alpha z p + (k-t)T - k'T \\ &\quad \leq \frac{z-\nu}{c}T + \alpha\frac{z}{c}T + \frac{c(k-t-1) - \left(ck' + \nu\right)}{c}T \\ &\qquad + T + \frac{\nu}{c}T \\ &\quad \leq \frac{z-\nu}{c}T + \alpha\frac{z}{c}T - \frac{z}{c}T + \frac{\nu}{c}T + T \\ &\quad = \alpha\frac{z}{c}T + T = \alpha\left(\frac{z}{c} + \frac{1}{\alpha}\right)T. \end{aligned} \tag{27}$$

If $z = c$, since $T \le (1/3)D_{\max}(\pi)$, we have $D_{\max}(\mu) - D_{\max}(\pi) \le (1+\alpha)T \le \alpha D_{\max}(\pi)$, as required. If $z > c$, then there exists no idle time immediately before $s(B_t)$ on the machine. As the job with a smaller completion time is delivered no later than that with a larger completion time, we have $\rho(B_{t-1}) = c(B_{t-1}) = s(B_t)$. Furthermore, there exists an idle time of the vehicle before $\delta(B_t)$; then $s(B_t) + T = \rho(B_{t-1}) + T \le \delta(B_{t-1}) + T < \delta(B_t) = c(B_t) \le s(B_t) + c(1+\alpha)p$, implying that

$$p > \frac{\alpha}{c}T. \tag{28}$$

However,

$$\left(\alpha\frac{\nu}{c} + 1\right) - \left(\frac{\nu}{c} + \alpha\right) = \frac{(c-\nu)(1-\alpha)}{c} > 0. \tag{29}$$

From (24), (26), (28), and (29), (27) can be rewritten as

$$\begin{aligned} D_{\max}(\mu) - D_{\max}(\pi) &\le \alpha\left(\frac{z}{c} + \frac{1}{\alpha}\right)T \\ &\le \alpha\left(k' + \frac{\nu}{c} - 1 + \frac{1}{\alpha}\right)T \\ &= \alpha\left(k' + \frac{\nu}{c} + \alpha\right)T \\ &\le \alpha\left(k' + \alpha\frac{\nu}{c} + 1\right)T \\ &\le \alpha\left[\nu p + \left(k' + 1\right)T\right] \\ &\le \alpha D_{\max}(\pi). \end{aligned} \tag{30}$$

This completes the proof. □

Theorem 11. *For the problem*

$$\begin{gathered} 1 \longrightarrow D \mid online, \quad r_j, \nu = 1, \quad c < \infty, \\ p_j \in [p, (1+\alpha)p] \mid D_{\max}, \end{gathered} \tag{31}$$

where the capacity of the vehicle is finite and fixed, D^c is the best possible online algorithm with a competitive ratio of $1+\alpha$.

Proof. From Lemmas 8, 9, and 10, the competitive ratio of online algorithm D^c is $1+\alpha$. The result follows from Lemma 2. □

Conflict of Interests

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

Acknowledgments

This work was supported by the Natural Sciences Foundation (Grant no. 20142BAB211017) of Jiangxi Province and the School Subject (Grant no. 06162015) of Jiangxi University of Finance and Economics.

References

[1] C. N. Potts, "Analysis of a heuristic for one machine sequencing with release dates and delivery times," *Operations Research*, vol. 28, no. 6, pp. 1436–1441, 1980.

[2] Y. C. Chang and C. Y. Lee, "Machine scheduling with job delivery coordination," *European Journal of Operational Research*, vol. 158, no. 2, pp. 470–487, 2004.

[3] Z.-L. Chen and G. L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.

[4] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.

[5] C.-Y. Lee and Z.-L. Chen, "Machine scheduling with transportation considerations," *Journal of Scheduling*, vol. 4, no. 1, pp. 3–24, 2001.

[6] W. Zhong, G. Dósa, and Z. Tan, "On the machine scheduling problem with job delivery coordination," *European Journal of Operational Research*, vol. 182, no. 3, pp. 1057–1072, 2007.

[7] K. Pruhs, J. Sgall, and E. Torng, "Online scheduling," in *Handbook of Scheduling: Algorithm, Models, and Performance Analysis*, J. Y.-T. Leung, Ed., Chapman and Hall/CRC, Boca Raton, Fla, USA, 2004.

[8] J. A. Hoogeveen and A. P. Vestjens, "A best possible deterministic on-line algorithm for minimizing maximum delivery time on a single machine," *SIAM Journal on Discrete Mathematics*, vol. 13, no. 1, pp. 56–63, 2000.

[9] M. van den Akker, H. Hoogeveen, and N. Vakhania, "Restarts can help in the on-line minimization of the maximum delivery time on a single machine," *Journal of Scheduling*, vol. 3, no. 6, pp. 333–341, 2000.

[10] I. Averbakh, "On-line integrated production-distribution scheduling problems with capacitated deliveries," *European Journal of Operational Research*, vol. 200, no. 2, pp. 377–384, 2010.

[11] J. Yuan, S. Li, J. Tian, and R. Fu, "A best on-line algorithm for the single machine parallel-batch scheduling with restricted delivery times," *Journal of Combinatorial Optimization*, vol. 17, no. 2, pp. 206–213, 2009.

[12] J. Tian, R. Fu, and J. Yuan, "An on-line algorithm for the single machine unbounded parallel-batching scheduling with large delivery times," *Information Processing Letters*, vol. 111, no. 21-22, pp. 1048–1053, 2011.

[13] M. Liu, C. Chu, Y. Xu, and F. Zheng, "An optimal online algorithm for single machine scheduling with bounded delivery times," *European Journal of Operational Research*, vol. 201, no. 3, pp. 693–700, 2010.

[14] M. Liu, F. F. Zheng, S. J. Wang, and J. Z. Huo, "Optimal algorithms for online single machine scheduling with deteriorating jobs," *Theoretical Computer Science*, vol. 445, pp. 75–81, 2012.

[15] Y. Lin and W. Li, "Parallel machine scheduling of machine-dependent jobs with unit-length," *European Journal of Operational Research*, vol. 156, no. 1, pp. 261–266, 2004.

[16] W. Li and S. Li, "Online batch scheduling of equal-length jobs on two identical batch machines to maximise the number of early jobs," *International Journal of Systems Science*, vol. 46, no. 4, pp. 652–661, 2015.

[17] W. Li, Z. Zhang, and S. Yang, "Online algorithms for scheduling unit length jobs on parallel-batch machines with lookahead," *Information Processing Letters*, vol. 112, no. 7, pp. 292–297, 2012.

[18] Y. Fang, P. Liu, and X. Lu, "Optimal on-line algorithms for one batch machine with grouped processing times," *Journal of Combinatorial Optimization*, vol. 22, no. 4, pp. 509–516, 2011.

[19] C. T. Ng and L. Lu, "On-line integrated production and outbound distribution scheduling to minimize the maximum delivery completion time," *Journal of Scheduling*, vol. 15, no. 3, pp. 391–398, 2012.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 503297, 4 pages
http://dx.doi.org/10.1155/2015/503297

Research Article

A Note on Two-Agent Scheduling with Resource Dependent Release Times on a Single Machine

Peng Liu[1] and Lini Duan[2]

[1] *School of Management, Shenyang University of Technology, Shenyang 110870, China*
[2] *School of Economics & Management, Shenyang University of Chemical Technology, Shenyang 110142, China*

Correspondence should be addressed to Peng Liu; liup7802@163.com

Received 6 October 2014; Accepted 30 November 2014

Academic Editor: Shuenn-Ren Cheng

We consider a scheduling problem in which both resource dependent release times and two agents exist simultaneously. Two agents share a common single machine, and each agent wants to minimize a cost function dependent on its own jobs. The release time of each A-agent's job is related to the amount of resource consumed. The objective is to find a schedule for the problem of minimizing A-agent's total amount of resource consumption with a constraint on B-agent's makespan. The optimal properties and the optimal polynomial time algorithm are proposed to solve the scheduling problem.

1. Introduction

Machine scheduling problems with multiagent have received increasing attention in recent years. Different agents share a common processing machine, and each agent wants to minimize a cost function depending on its jobs only. Scheduling with multiple agents is firstly introduced by Baker and Smith [1] and Agnetis et al. [2]. Baker and Smith [1] consider the problem of minimizing a convex combination of the agents' objective functions. They provide some dominance properties and demonstrate where the problem becomes computationally difficult. Agnetis et al. [2] address the scheduling problem in which two agents compete for the usage of shared processing resource and each agent has his own criterion to optimize. They study the constrained optimization problem and the Pareto-optimization problem on a single machine and the shop environment. Agnetis et al. [3] analyze the complexity of some multiagent scheduling problems on a single machine and propose solution algorithms. Lee et al. [4] provide approximation algorithms for multiagent scheduling to minimize total weighted completion time. Gawiejnowicz et al. [5] consider a single-machine two-agent scheduling problem with proportionally deteriorating job processing times. Yin et al. [6] consider several two-agent scheduling problems with assignable due dates on a single machine. Cheng et al. [7] study a two-agent single-machine scheduling problem with release times to minimize the total weighted completion time. Yu et al. [8] investigate several single-machine two-synergetic-agent scheduling problems. Yin et al. [9] consider two-agent single-machine scheduling problems with unrestricted due date assignment.

The scheduling problems with resource consumption have been studied for many years. Cheng and Janiak [10] study the resource-optimal control of job completion on a single machine with a constraint on maximum job completion time. Vasilev and Foote [11] investigate a single machine scheduling problem where the resource consumed depends on the release times of jobs. Kaspi and Shabtay [12] consider the problem of scheduling jobs on a single machine where job-processing times are controllable through the allocation of a common limited resource. Wang and Cheng [13] consider the single machine scheduling problem with resource dependent release times and processing times. Wei et al. [14] consider the single-machine scheduling with time-and-resource-dependent processing times. J.-B. Wang and M.-Z. Wang [15] consider the single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time. Yin et al. [16] consider single-machine due window assignment and scheduling problems with a common flow allowance and controllable job processing time. Lu et al. [17] consider a single-machine earliness-tardiness scheduling problem with due-date assignment, in which the processing time of a job is a function of its position in a sequence and its resource allocation.

However, to the best of our knowledge, no work has been done on models with both aspects of resource dependent release times and multiagent in the literature. These two categories of scheduling problems have been extensively and separately researched over the last two decades. In this paper, we study the two-agent scheduling problems on a single machine with resource dependent release times, where the goal is to find a schedule that minimizes the objective function of one agent with the restriction that the objective function of the other agent cannot exceed a given bound. The problems under consideration fall into the category of scheduling problems with resource consumption and multiple agents. Such a scheduling problem commonly arises in the steel industry. Janiak [18] describes a practical scheduling problem with resource dependent release times in steel mills, where batches of ingots have to be preheated before they can be hot rolled in a blooming mill, and the ingot preheating time is inversely proportional to the total amount of resources consumed.

The remainder of this paper is organized as follows. In Section 2, we describe the proposed problem. In Section 3, we develop the optimal polynomial time algorithm for the two-agent single-machine scheduling problem. Section 4 gives some concluding remarks.

2. Problem Description

We now describe our problem formally. There are two families of independent and nonpreemptive jobs $J^A = \{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and $J^B = \{J_1^B, J_2^B, \ldots, J_{n_B}^B\}$ to be processed on a common single machine. The jobs in J^A and J^B are called A-agent's jobs and B-agent's jobs, respectively. Associated with each job J_h^A, let p_h^A denote the processing time and $h = 1, 2, \ldots, n_A$. The release time r_h^A is related to the amount of the resource f_h^A, consumed on job J_h^A. A strictly decreasing continuous function is given $f : R^+ \rightarrow R^+$. We refer to f as the resource consumption function. We assume that $f_h^A = f(r_h^A)$. Each A-agent's job can start at any time after the release time of the job, and idle time between jobs is allowed. Since the consumption function f is strictly decreasing continuously, we may assume that each job starts as soon as it becomes available. That is, we can take $s_h^A = r_h^A$, where s_h^A is the starting time of job J_h^A. Associated with each job J_k^B, let p_k^B and r_k^B denote the processing time and the release times, respectively, and $k = 1, 2, \ldots, n_B$. Let π indicate a feasible schedule of the $n = n_A + n_B$ jobs. Let $C_k^B(\pi)$ denote the completion time of B-agent's job J_k^B under schedule π. The objective function of agent A is to minimize the total amount of resource consumption $\sum_{h=1}^{n_A} f(r_h^A)$. The objective function of agent B is to minimize the makespan $C_{\max}^B = \max_{k=1,2,\ldots,n_B}\{C_k^B(\pi)\}$.

The goal is to minimize the total amount of resource consumption $\sum_{h=1}^{n_A} f(r_h^A)$ of agent A with the restriction that the makespan $C_{\max}^B$ of agent B cannot exceed a given bound U. If the value U is too small, an instance of the scheduling problem may not have feasible solutions. If there is at least one feasible solution, we say that the instance is *feasible*. According to the three-field notation $\psi_1 \mid \psi_2 \mid \psi_3$ of Graham et al. [19], the scheduling problem is denoted as $1\| \sum_{h=1}^{n_A} f(r_h^A) : C_{\max}^B \leq U$.

3. Main Results

In this section, we develop an optimal polynomial time algorithm to solve the problem $1\| \sum_{h=1}^{n_A} f(r_h^A) : C_{\max}^B \leq U$.

Given a sequence $\pi = \{J_1^B, J_2^B, \ldots, J_{n_B}^B, J_1^A, J_2^A, \ldots, J_{n_A}^A\}$, for each B-agent' job, the completion time C_k^B may be completed recursively as $C_1^B = r_1^B + p_1^B$, $C_k^B = \max\{r_k^B, C_{k-1}^B\} + p_k^B$, $k = 2, \ldots, n_B$.

Thus the completion time of job J_k^B may also be taken as $C_k^B = \max_{1\leq j\leq k}\{r_j^B + \sum_{l=j}^{k} p_l^B\}$.

Moreover, let the maximum job completion time be denoted by $C_{\max}^B = \max_{1\leq k\leq n_B}\{C_k^B\}$; then $C_{\max}^B = \max_{1\leq k\leq n_B}\{r_k^B + \sum_{i=k}^{n_B} p_i^B\}$.

Lemma 1. *Given a sequence $\pi = \{J_1^B, J_2^B, \ldots, J_{n_B}^B, J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ and a constant U, define $C_\pi^B = \max_{1\leq k\leq n_B}\{r_k^B + \sum_{i=k}^{n_B} p_i^B\}$. Then, if $U < C_\pi^B$, the sequence π corresponds to an infeasible schedule.*

Now we can define bounds for the constraint U. Define $\underline{U} = \max_{1\leq k\leq n_B}\{r_k^B + \sum_{i=k}^{n_B} p_i^B\}$. The analysis in the following section will be confined to the case in which $\underline{U} \leq U$.

Lemma 2. *An optimal schedule exists in which the A-agent's jobs are processed in the nonincreasing order of processing times p_h^A.*

Proof. The resource consumption function f is a strictly decreasing continuous function to A-agent's jobs. Since releasing A-agent's jobs sooner consumes more resource, A-agent's jobs should be released as late as possible. Hence A-agent's jobs should be released in nonincreasing order of p_h^A. □

Lemma 3. *An optimal schedule exists in which the B-agent's jobs are processed in the nondecreasing order of release times r_k^B.*

Proof. The makespan of agent B is the maximum completion time of B-agent's jobs on the single machine; that is, the makespan of agent B is the completion time of the last B-agent's job. Using a pairwise job interchange argument, we can process B-agent's jobs in the nondecreasing order of release times r_k^B. □

Next, an algorithm to determine an optimal schedule of the problem $1\| \sum_{h=1}^{n_A} f(r_h^A) : C_{\max}^B \leq U$ is developed as follows.

Algorithm 4.

Step 1. Arrange the A-agent's jobs as $\{J_1^A, J_2^A, \ldots, J_{n_A}^A\}$ according to the nonincreasing order of p_h^A and denote all B-agent's jobs sequenced by the nondecreasing order of r_k^B as a dummy job $B1$.

Step 2. Define sequence $S = \{B1, J_1^A, J_2^A, \ldots, J_{n_A-1}^A, J_{n_A}^A\}$ and calculate $C_{\max}^B$ for agent B. The sequence S is an optimal schedule and the starting times of A-agent's job are given by $s_1^A = U$, $s_h^A = s_{h-1}^A + p_{h-1}^A = s_1^A + \sum_{i=1}^{h-1} p_i^A$, $h = 2, 3, \ldots, n_A$.

Theorem 5. *Algorithm 4 generates an optimal schedule for the problem* $1\|\sum_{h=1}^{n_A} f(r_h^A) : C_{\max}^B \le U$ *in* $O(n_A \log n_A + n_B \log n_B)$ *time.*

Proof. The proof of optimality is straightforward from the results of Lemmas 1–3. We now turn to time complexity. The time to sequence the jobs of set J^A according to the nonincreasing order of p_h^A is $O(n_A \log n_A)$. The time to sequence the jobs of set J^B according to the nondecreasing order of r_k^B is $O(n_B \log n_B)$. Creating dummy job $B1$ incurs $O(n_B)$ operations. So the overall computational complexity of Algorithm 4 is bounded by $O(n_A \log n_A + n_B \log n_B)$. This completes the proof. □

4. Conclusions

In this paper, we combine two important issues in scheduling that recently have received increasing attention from researchers: resource dependent release times and multiple agents. Our goal is to find a schedule for the problem of minimizing A-agent's total amount of resource consumption with a constraint on B-agent's makespan. We propose the optimal properties and the optimal polynomial time algorithm for the considered scheduling problem.

The future research may be directed to analyze the problems with other objective functions such as minimizing the number of late jobs, the total weighted completion time and tardiness. An interesting research topic is also to analyze the scheduling problem with more than two agents or in other machine environments.

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 reviewers for their helpful comments and suggestions. This research is supported by the National Natural Science Foundation of China (Grant no. 71001074) and the Program for Liaoning Excellent Talents in University (Grant no. WJQ2013003).

References

[1] 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.

[2] 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.

[3] A. Agnetis, D. Pacciarelli, and A. Pacifici, "Multi-agent single machine scheduling," *Annals of Operations Research*, vol. 150, no. 1, pp. 3–15, 2007.

[4] K. Lee, B.-C. Choi, J. Y. 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.

[5] S. Gawiejnowicz, W.-C. Lee, C.-L. Lin, and C.-C. Wu, "Single-machine scheduling of proportionally deteriorating jobs by two agents," *Journal of the Operational Research Society*, vol. 62, no. 11, pp. 1983–1991, 2011.

[6] Y. Yin, S.-R. Cheng, T. C. Cheng, C.-C. Wu, and W.-H. Wu, "Two-agent single-machine scheduling with assignable due dates," *Applied Mathematics and Computation*, vol. 219, no. 4, pp. 1674–1685, 2012.

[7] T. C. Cheng, Y. H. Chung, S. C. Liao, and W. C. Lee, "Two-agent singe-machine scheduling with release times to minimize the total weighted completion time," *Computers and Operations Research*, vol. 40, no. 1, pp. 353–361, 2013.

[8] X. Yu, Y. Zhang, D. Xu, and Y. Yin, "Single machine scheduling problem with two synergetic agents and piece-rate maintenance," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 37, no. 3, pp. 1390–1399, 2013.

[9] Y. Q. Yin, T. C. E. Cheng, X. Q. Yang, and C. C. Wu, "Two-agent single-machine scheduling with unrestricted due date assignment," *Computers & Industrial Engineering*, vol. 79, pp. 148–155, 2014.

[10] T. C. Cheng and A. Janiak, "Resource optimal control in some single-machine scheduling problems," *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1243–1246, 1994.

[11] S. H. Vasilev and B. L. Foote, "On minimizing resource consumption with constraints on the makespan and the total completion time," *European Journal of Operational Research*, vol. 96, no. 3, pp. 612–621, 1997.

[12] M. Kaspi and D. Shabtay, "Convex resource allocation for minimizing the makespan in a single machine with job release dates," *Computers and Operations Research*, vol. 31, no. 9, pp. 1481–1489, 2004.

[13] X. Wang and T. C. Cheng, "Single machine scheduling with resource dependent release times and processing times," *European Journal of Operational Research*, vol. 162, no. 3, pp. 727–739, 2005.

[14] C.-M. Wei, J.-B. Wang, and P. Ji, "Single-machine scheduling with time-and-resource-dependent processing times," *Applied Mathematical Modelling: Simulation and Computation for Engineering and Environmental Systems*, vol. 36, no. 2, pp. 792–798, 2012.

[15] J.-B. Wang and M.-Z. Wang, "Single-machine scheduling to minimize total convex resource consumption with a constraint on total weighted flow time," *Computers & Operations Research*, vol. 39, no. 3, pp. 492–497, 2012.

[16] Y. Yin, T. C. E. Cheng, C.-C. Wu, and S.-R. Cheng, "Single-machine due window assignment and scheduling with a common flow allowance and controllable job processing time," *Journal of the Operational Research Society*, vol. 65, no. 1, pp. 1–13, 2014.

[17] 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.

[18] A. Janiak, "Minimization of the blooming mill standstills-mathematical model, suboptimal algorithms," *Mechanika*, vol. 8, no. 2, pp. 37–49, 1989.

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

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 841637, 6 pages
http://dx.doi.org/10.1155/2015/841637

Research Article

A Hybrid Heuristic Algorithm for Ship Block Construction Space Scheduling Problem

Shicheng Hu,[1] Tengjiao Liu,[1] Song Wang,[1] Yonggui Kao,[2] and Xuedong Sun[3]

[1] *School of Economics and Management, Harbin Institute of Technology, Weihai 264209, China*
[2] *Department of Mathematics, Harbin Institute of Technology, Weihai 264209, China*
[3] *School of Software, Sun Yat-sen University, Guangzhou 510275, China*

Correspondence should be addressed to Xuedong Sun; sunxdys@163.com

Received 27 December 2014; Accepted 17 January 2015

Academic Editor: Shuenn-Ren Cheng

Ship block construction space is an important bottleneck resource in the process of shipbuilding, so the production scheduling optimization is a key technology to improve the efficiency of shipbuilding. With respect to ship block construction space scheduling problem, a hybrid heuristic algorithm is proposed in this paper. Firstly, Bottom-Left-Fill (BLF) process is introduced. Next, an initial solution is obtained by guiding the sorting process with corners. Then on the basis of the initial solution, the simulated annealing arithmetic (SA) is used to improve the solution by offering a possibility to accept worse neighbor solutions in order to escape from local optimum. Finally, the simulation experiments are conducted to verify the effectiveness of the algorithm.

1. Introduction

Space is the key resource in ship block construction process. How to minimize the makespan of the project under space resource and precedence constraints is a complicated scheduling problem. As for this problem, two related problems are involved: resource constrained project scheduling problem (RCPSP) and bin packing problem.

RCPSP can be described as a problem which should be scheduled under the limits of technology and other constraints to meet the objective of a project [1]. Generally, the goal is to get the shortest project makespan under the available resources and precedence constraints. The methods can be classified into two categories: exact methods and heuristic methods. Hartmann [2] puts forward that RCPSP belongs to NP-hard problem because it is always used to extend machine scheduling problem [3, 4]. So with the augment of problem scale, the computational complexity will increase rapidly. Many researchers have used exact algorithm to solve RCPSP [5, 6]; however, most of them proposed that exact algorithm is not feasible in reality. Priority rules proposed by Kelley [7] indicated that RCPSP can be solved by heuristic algorithms. Liu and Wang [8] tried to reduce the project makespan by heuristic algorithms and achieved good results. Bhaskar et al. [9] utilized parallel methods and priority rules to solve RCPSP with fuzzy activity times. Lee et al. [10] proposed a ship block construction space scheduling problem and described this problem theoretically. Koh et al. [11] solved the scheduling problem in shipbuilding company by heuristic algorithm.

The bin packing problem is putting more boxes into a limited bin in order to minimum the height. This problem can be classified into two categories, two-dimensional and three-dimensional problems. For the former one, researchers tend to solve bin packing problems by heuristic methods. They are Bottom-Left (BL) algorithm [12], Bottom-Left-Fill (BLF) algorithm [13]. In addition, Belov et al. [14] considered adapting one-dimensional problem for solving two-dimensional problems. Chan et al. [15] tried to solve two-dimensional problems by heuristics with stochastic neighborhood structures. For three-dimensional problems, the most popular 3BF [16], proposed by Silvano Martello in 2000, figure out the problem of how to choose the most suitable cubes. Alvarez-Valdes et al. [17] used a GRASP/Path relinking algorithm to solve multiple bin-size bin packing problems. Liao and Hsu [18] found new lower bounds to improve the efficiency of three-dimensional problems.

This paper is organized as follows. After introduction, Section 2 presents a mathematical model. Section 3 gives the hybrid heuristic algorithm for this problem. In Section 4, we conduct simulation experiments to verify the effectiveness of the algorithm. Section 5 proposes general conclusions.

2. Ship Block Construction Space Scheduling Problem

Ship block construction space scheduling problem can be described as a project which includes n activities $N = \{1, 2, \ldots, n\}$; place is required to process activities. A activity can be defined as $a_{ik} = e_{ik} \times c_{ik}$, where e_{ik} and c_{ik} represent the length and width of place k which activity i needs. The duration of activity i is t_i. I_i and H_i are the start and finish time of activity i. During the project, there are m places $M = \{1, 2, \ldots, m\}$ which can be defined as $B_k = E_k \times C_k$.

During the project, every activity is under precedence constraints, and we propose that P_j is the predecessors of activity j. So j can not be started if any one of its predecessors in P_j has not been finished. We assume that the start time of the whole project is 0. For the convenience of modeling, we also introduce two dummy nodes: activity 0 and $n+1$. They do not need time and space. 0 is the predecessor of all the activities in the project; meanwhile, $n+1$ is the successor of all the activities. So I_{n+1} is the makespan of the project. In addition, we regard an activity as a cube which is represented as $z_{ik} = e_{ik} \times c_{ik} \times t_i$. All the activities can be rotated in horizontal with 90 degree, and $r_i = 1$ and 0, indicating that activity i is rotated and not, respectively. The model of this problem is shown in what follows:

$$\text{Minimize: } I_{n+1}, \quad (1)$$

$$I_i + t_i \le I_j, \quad \forall i \in P_j, \quad (2)$$

$$z_{ik} \cap z_{jk} = \varphi, \quad \forall i, j \in N, \ i < j, \ \forall k \in M, \quad (3)$$

$$\left[(1 - r_i)\, w_{ik} + r_i e_{ik}\right] \le \left[(1 - r_i)\, c_{ik} + r_i e_{ik}\right] + x_i, \quad (4)$$

$$\left[(1 - r_i)\, c_{ik} + r_i e_{ik}\right] + x_i \le E_k, \quad (5)$$

$$\left[(1 - r_i)\, e_{ik} + r_i c_{ik}\right] \le \left[(1 - r_i)\, e_{ik} + r_i c_{ik}\right] + y_i, \quad (6)$$

$$\left[(1 - r_i)\, e_{ik} + r_i c_{ik}\right] + y_i \le C_k, \quad (7)$$

$$r_i \in \{0, 1\}. \quad (8)$$

In the model, formula (1) is the objective of the problem, (2) proposes the precedence constraints, and (3) means that two cubes cannot overlap. Formulae (4)–(7) denote that each cube should be completed within available place. The cube can be rotated horizontally and 0 and 1 represent if the cube is rotated, as formula (8) has shown.

3. A Hybrid Heuristic Algorithm

3.1. Initial Solution Method. In this paper, we apply BLF to get the initial solution. BLF, presented by Chazelle [13] in 1983, belongs to heuristic algorithm. In this algorithm, the method of placing cubes is determined by corners [19].

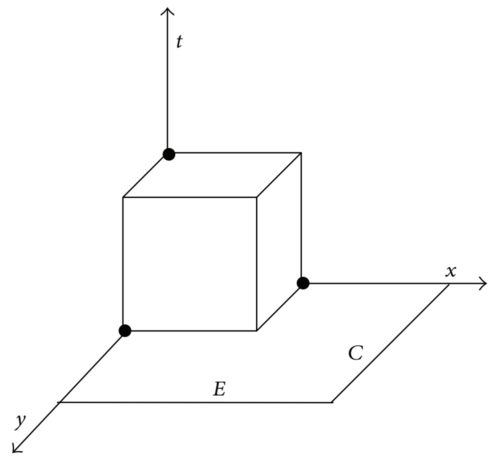


Figure 1: Corners.

3.1.1. Corner. In BLF, it is important to find corners to place the activity. Firstly, we will try the lowest and leftmost point (lowest point first); if this placement can match the activity, then place it in the position and update corners; otherwise, try next point until a corner is found. The corners can be represented as $\{(x, y, t), (e, c, h)\}$, where (x, y, t) indicates the location of the corners and (e, c, h) represents the available space of the point in x-y-z dimensions. See Figure 1; E and C denote the length and width of the place and t represents the duration of the project.

3.1.2. Bottom-Left-Fill Algorithm. In this paper, we represent the activities as cubes whose length, width, and height are limited. For these cubes, the length and width denote the length and width of the place they need, and the height represents the time they need. Meanwhile, we regard the unrestricted height as timeline t. When all activities have been placed, the total height equals the duration of the project. In the process, we find activities that can be placed at each time point first and then consider the sequence of these activities. We place the activity with larger bottom area first; if equal, choose the higher height; if still equal, select the one with the longer length. We describe the algorithm as follows.

(1) We embed the place (the platform in ship block construction space) in a three-dimensional coordinate system, put the bottom-left-rear point on the origin (0, 0, 0), and put length, width, and height on x, y, and t axis, respectively. E and C represent length and width of the place.

(2) Set 5 sets: A_n, C_n, D_n, I, and B. A_n is a set of activities which have been scheduled but not yet completed. C_n is a set of activities which have been scheduled and already completed. D_n is a set of activities which can be scheduled but have not been scheduled yet. I is a set of corners, while B is a set of all activities in input order.

(3) Move the activities which can be scheduled from B to D_n and then sort these activities: firstly, interchange the length and width of the activities so that $e_i \geq c_i$ can be met for any activity b_i; then sort the activities in D_n by placement policy mentioned above, and the first activity should be scheduled in D_n after being sorted. Detect the corners in I until the first eligible corner is found by t-y-x rule, which means that the smaller t among the corners is better. Then, consider the smaller y. Finally, select the smaller x. Put the bottom-left-rear point of the activity on the selected corner, and the new eligible corners will be new elements of I; meanwhile delete the corner which is used this time from I. Record the activities and their corners sequentially.

(4) If an activity in A_n has been completed, move it from A_n to C_n and then move its nearest successor from B to D_n. Sort activities in D_n by rules mentioned in (3). Repeat (3) and (4) until all activities are completed.

(5) Select the higher t of all corners as the duration of the project.

3.2. Optimization Solution Method. Proposed by Steinbrunn et al. [20], simulated annealing (SA) is a metaheuristic algorithm which offers a possibility to accept worse neighbor solutions to escape from local optimum. Until 1983, Kirkpatrick et al. [21] transformed this idea to SA arithmetic and applied it to traveling salesman problem successfully.

Chan et al. [22] develop a hybrid algorithm which gleans the ideas both from tabu search and sample sort simulated annealing to solve distributed scheduling problem. Bouleimen and Lecocq [23] implemented SA to RCPSP and MRCPSP. Yannibelli and Amandi [24] combined SA and other algorithms to solve a multiobjective project scheduling problem. In SA arithmetic, we describe the neighborhood as the modification of the order of two activities which can be scheduled at the same time. d_t represents temperature decreasing rate, L denotes the initial value of neighborhood, d_L represents the neighbors that are generated once the temperature drops, f denotes current duration, f_{best} denotes the shortest duration till now. For our ship block construction space scheduling problem, the SA can be described as follows.

(1) Initialization: the initial and final temperature is S_t and E_t, respectively. The algorithm iterates from initial solution B. The value of initial evaluation function is $f = 3D-PS(B)$, while the evaluation function value of initial optimal solution $B_{\text{best}} = B$ is $f_{\text{best}} = f$.

(2) t denotes the current temperature and L_t denotes the length of current neighborhood. Initially $t = S_t$, $L_t = L$.

(3) $j = 1$.

(4) Select a new solution B' from the neighborhood $N(B)$ of B.

(5) Calculate the incremental $d_f = f - f'$.

(6) If $d_f > 0$, accept B' as a new solution, $f = f'$, $B = B'$; if $f' < f_{\text{best}}$, then $f_{\text{best}} = f'$, $B_{\text{best}} = B'$. Else accept B' as a new solution with a probability $P = \exp(10 * d_f/T)$.

(7) $j = j + 1$, if $j \leq L_t$, go to step (4); else go to step (8).

(8) $L_t = d_L$, $t^* = d_t$, and compare t and E_t. If $t \geq E_t$, go to step (3); else the termination condition is met and output current f_{best} and B_{best}; end the program.

Table 1: The parameters of activities.

Activity	Duration	Length, width	Place
1	0	0, 0	1
2	8	2, 1	1
3	4	2, 2	1
4	6	2, 1	2
5	3	2, 2	2
6	8	2, 1	2
7	5	2, 1	2
8	9	3, 1	1
9	2	3, 2	1
10	3	1, 1	1
11	7	2, 2	1
12	2	2, 2	2
13	7	3, 1	1
14	9	2, 1	2
15	4	3, 2	1
16	6	2, 1	2
17	3	2, 1	1
18	0	0, 0	2

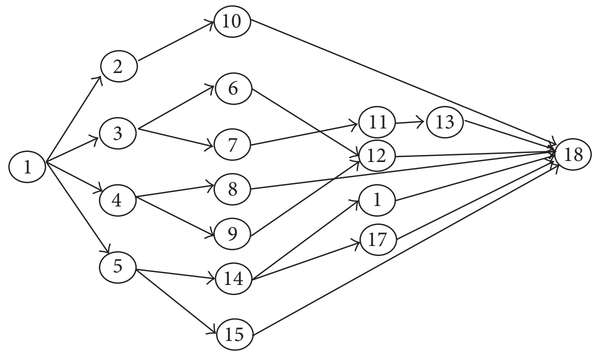


Figure 2: The network diagram of activities.

4. Simulation Experiment

We use a set of activities to verify our algorithm. The parameters of these activities are shown in Table 1, and the network diagram of activities is presented in Figure 2. There are 2 places in the experiment, x, y, and t represent length, width, and time, respectively. The length and width of place 1 are 3 and 3, while 3 and 2 are for place 2.

4.1. Generation of Initial Solution. We get the initial solution as follows.

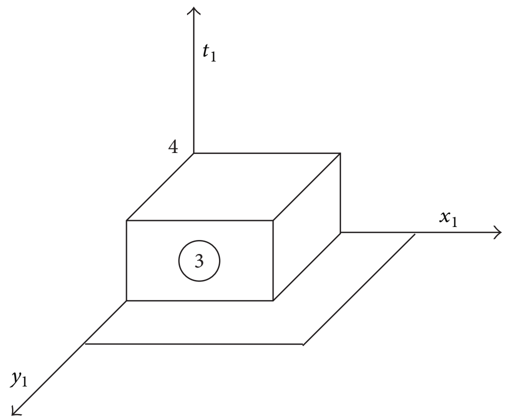


Figure 3: The status of place 1 while activity 3 is put.

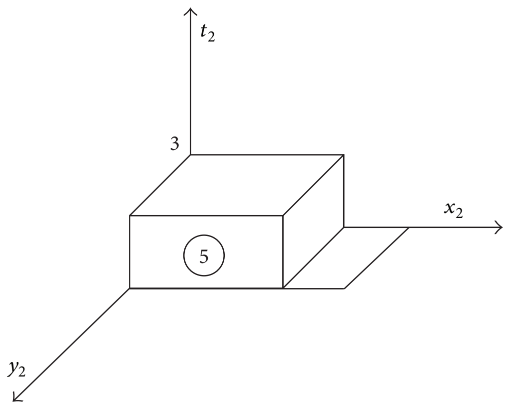


Figure 4: The status of place 2 while activity 5 is put.

(1) At first, $t = 0$. $\{2, 3, 4, 5\}$ can be arranged at this time. According to the rule above, we adjust the order to $\{3, 5, 2, 4\}$. Activity 3 will be completed in place 1, so we put the bottom-left-rear point of it at the origin of place 1. So three new corners $\{(0, 0, 4), (3, 3, \infty)\}$, $\{(0, 2, 0), (3, 1, \infty)\}$, $\{(2, 0, 0), (1, 3, \infty)\}$ are produced. Update A_n, D_n, I, $C_n = \{1\}$, $A_n = \{3\}$, $D_n = \{5, 2, 4\}$, $I_1 = \{\{(2, 0, 0), (1, 3, \infty)\}, \{(0, 2, 0), (3, 1, \infty)\}, \{(0, 0, 4), (3, 3, \infty)\}\}$, $I_2 = \{\{(0, 0, 0), (3, 2, \infty)\}\}$. See Figure 3.

(2) Next, activity 5 will be scheduled on place 2. We put the bottom-left-rear point of activity 5 at the origin of place 2. Then new corners $\{(0, 0, 3), (3, 2, \infty)\}$, $\{(0, 2, 0), (3, 0, \infty)\}$, $\{(2, 0, 0), (1, 2, \infty)\}$ are produced. According to the rule, we delete $\{(0, 2, 0), (3, 0, \infty)\}$. Update A_n, D_n, I. $C_n = \{1\}$, $A_n = \{3, 5\}$, $D_n = \{2, 4\}$, $I_1 = \{\{(2, 0, 0), (1, 3, \infty)\}, \{(0, 2, 0), (3, 1, \infty)\}, \{(0, 0, 4), (3, 3, \infty)\}\}$, $I_2 = \{\{(2, 0, 0), (1, 2, \infty)\}, \{(0, 0, 3), (3, 2, \infty)\}\}$. See Figure 4.

(3) Detect the corners of I_1 and $\{(2, 0, 0), (1, 3, \infty)\}$ is selected for activity 2. After activity 2 is scheduled, three new corners are produced and only $\{(2, 0, 8), (1, 3, \infty)\}$ is eligible to be a corner. Considering that part of the space has been occupied by activity 2, we

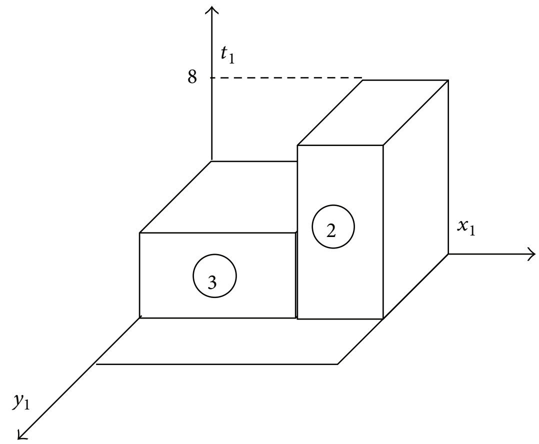


Figure 5: The status of place 1 while activity 2 is put.

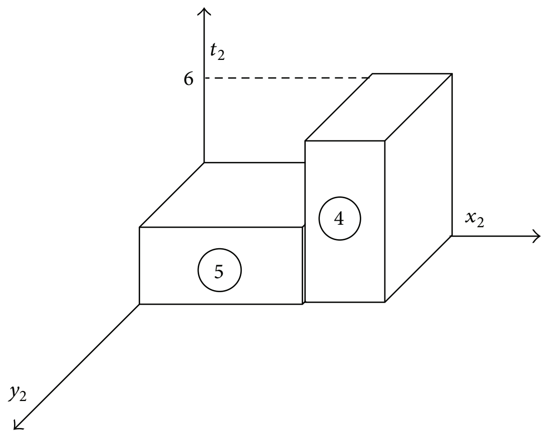


Figure 6: The status of place 2 while activity 4 is put.

should retest the available space of previous corners when we update I_1. $\{(0, 0, 4), (3, 3, \infty)\}$ should be modified into $\{(0, 0, 4), (2, 3, \infty)\}$. Then $C_n = \{1\}$, $A_n = \{3, 5, 2\}$, $D_n = \{4\}$, $I_1 = \{\{(0, 2, 0), (3, 1, \infty)\}, \{(0, 0, 4), (2, 3, \infty)\}, \{(2, 0, 8), (1, 3, \infty)\}\}$, $I_2 = \{\{(2, 0, 0), (1, 2, \infty)\}, \{(0, 0, 3), (3, 2, \infty)\}\}$. See Figure 5.

(4) At time 0, activity 4 will be put in order. Detect I_2 and put it on $\{(2, 0, 0), (1, 2, \infty)\}$. After testing, only $\{(2, 0, 6), (1, 2, \infty)\}$ can be a new corner. Similarly, the available space of previous corners in I_2 should also be retested and $\{(0, 0, 3), (3, 2, \infty)\}$ should be changed into $\{(0, 0, 3), (2, 2, \infty)\}$. $C_n = \{1\}$, $A_n = \{3, 5, 2, 4\}$, $D = \emptyset$, $I_1 = \{\{(0, 2, 0), (3, 1, \infty)\}, \{(0, 0, 4), (2, 3, \infty)\}, \{(2, 0, 8), (1, 3, \infty)\}\}$, $I_2 = \{\{(0, 0, 3), (2, 2, \infty)\}, \{(2, 0, 6), (1, 2, \infty)\}\}$. See Figure 6.

(5) At time 0, space is still available, but it is not enough for any remaining activity. So next time point is $t = 3$, and activity 5 is completed at this time. $C_n = \{1, 5\}$, $A_n = \{3, 2, 4\}$, $D_n = \{14, 15\}$. After ordering, D_n is changed into $\{15, 14\}$. In order to arrange activity 15, we detect I_1. If there is any corner whose value of t is less than 3, we should change it into 3 and modify available space of this

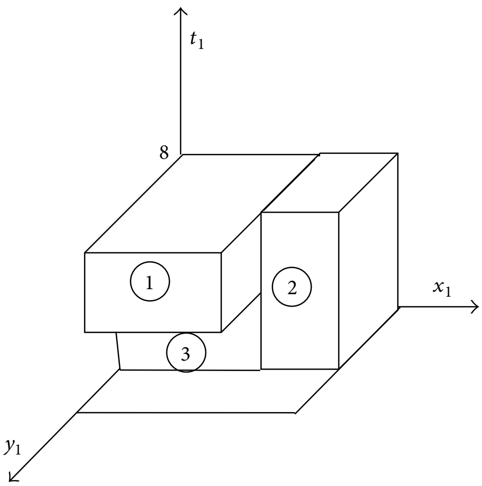


Figure 7: The status of place 1 while activity 15 is put.

corner. For example, $\{(0,2,0),(3,1,\infty)\}$ should be changed into $\{(0,2,3),(3,1,\infty)\}$ before activity 15 is put. Meanwhile, the space of this corner should be changed into $\{(0,2,3),(3,1,1)\}$ after activity 15 is put. We lay activity 15 on $\{(0,0,4),(2,3,\infty)\}$. $C_n = \{1,5\}$, $A_n = \{3,2,4,15\}$, $D_n = \{14\}$, $I_1 = \{\{(0,2,3),(3,1,1)\},\{(0,0,8),(3,3,\infty)\}\}$, $I_2 = \{\{(0,0,3),(2,2,\infty)\},\{(2,0,6),(1,2,\infty)\}\}$. See Figure 7.

(6) According to this train of thought, we get an initial solution $\{3(0,0,0)\}$, $\{5(0,0,0)\}$, $\{2(2,0,0)\}$, $\{4(2,0,0)\}$, $\{15(0,0,4)\}$, $\{14(0,0,3)\}$, $\{6(0,1,4)\}$, $\{7(2,0,6)\}$, $\{9(0,0,8)\}$, $\{8(0,2,8)\}$, $\{10(0,0,10)\}$, $\{11(1,0,11)\}$, $\{12(0,0,12)\}$, $\{16(2,0,12)\}$, $\{17(0,0,13)\}$, $\{13(0,0,18)\}$, in which $\{B(x,y,t)\}$ means activity B should be started at time t, and the space it needs can be represented as (x,y). According to the initial solution, the whole duration of the project is 25. The status of place 1 and place 2 is shown in Figure 8.

4.2. Optimization of Initial Solution. In SA, the order of activities has great influence on the duration of the project. So by searching the neighborhood and increasing the size of neighborhood dynamically, we manage to use SA and improve the initial solution.

We show a step of improvement as follows.

(1) $f = f_{\text{best}} = 25$, $B = B_{\text{best}} = \{3,5,2,4,15,14,6,7,9,8,10,11,12,16,17,13\}$.

(2) $t = S_t$, $L_t = L$.

(3) If $t \geq E_t$, go to step (4); else output current f_{best} and B_{best}; finish the program.

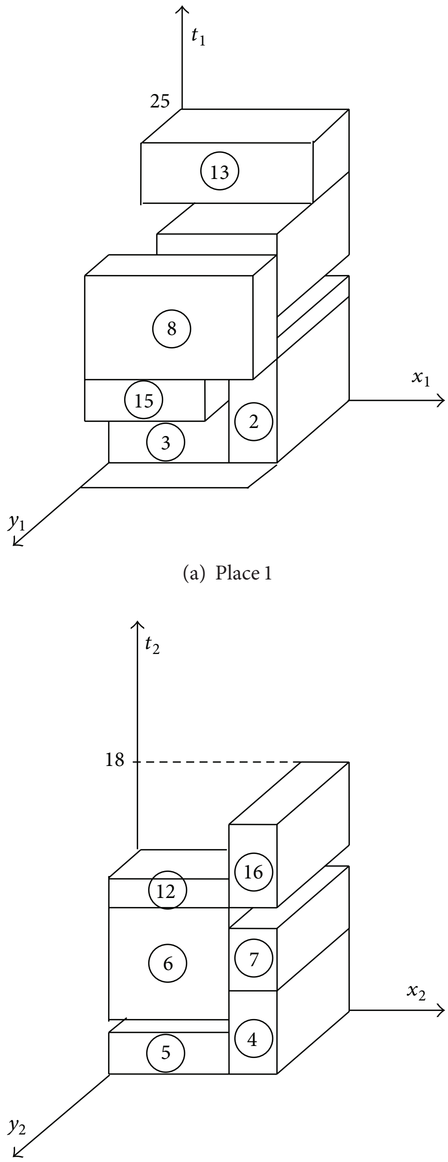
(4) $j = 1$.

(5) If $j > L_t$, $L_t += d_L$, and go to step (4); else go to next step.

(6) Select B' from the neighborhood $N(B)$ of B. In our example, we swap the order of activity 6 and activity 7. Then, we get another solution $\{3(0,0,0)\}$, $\{5(0,0,0)\}$, $\{2(2,0,0)\}$, $\{4(2,0,0)\}$, $\{15(0,0,4)\}$, $\{14(0,0,3)\}$, $\{7(0,1,4)\}$, $\{6(2,0,6)\}$, $\{9(0,0,8)\}$, $\{8(0,2,8)\}$, $\{10(0,0,10)\}$, $\{11(1,0,10)\}$, $\{16(0,0,12)\}$, $\{17(0,0,13)\}$, $\{12(0,0,18)\}$, $\{13(0,2,17)\}$. $f' = 24$, $B' = \{3,5,2,4,15,14,7,6,9,8,10,11,16,17,12,13\}$. $d_f = f - f' = 25 - 24 = 1$. If $d_f > 0$, go to step (7); else, go to step (8). The current d_f is 1, so go to step (7).

(7) Compare f' and f_{best}, if $f' < f_{\text{best}}$, $f_{\text{best}} = f'$, $B_{\text{best}} = B'$; else $j = j + 1$ and go to step (5). In this example, $f' < f_{\text{best}}$, so $f_{\text{best}} = 24$, $B_{\text{best}} = \{3,5,2,4,15,14,7,6,9,8,10,11,16,17,12,13\}$.

(8) Generate a number x randomly, if $x < \exp(10 * d_f/t)$, $f = f'$, $B = B'$, $j = j + 1$, and go to step (5).



(a) Place 1

(b) Place 2

Figure 8: The status while all activities are put.

After this improvement, the optimal solution reduces from 25 to 24. So it is proved that the method is effective.

5. Conclusion

In this paper, we combine BLF and SA to solve ship block construction space scheduling problem. During the procedure of scheduling activities, we guide the sorting process with corners. Then, the sorting of initial solution can be changed by SA. However, how to improve the searching efficiency will be the future research, especially when the number of blocks is very large.

Conflict of Interests

The authors declare that they have no financial and personal relationships with other people or organizations that can inappropriately influence their work, and they also declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by NSFC under Grant no. 61202345, NSFS under Grant no. ZR2012FM006, and SDPW under Grant no. IMZQWH010016. The authors thank the 2 reviewers and handling editor for their meticulous reading of the paper and constructive comments which greatly improved the paper.

References

[1] R. Kolisch, "Efficient priority rules for the resource-constrained project scheduling problem," *Journal of Operations Management*, vol. 14, no. 3, pp. 179–192, 1996.

[2] S. Hartmann, "A competitive genetic algorithm for resource-constrained project scheduling," *Naval Research Logistics*, vol. 45, no. 7, pp. 733–750, 1998.

[3] Y. Q. Yin, T. C. E. Cheng, X. Q. Yang, and C.-C. Wu, "Two-agent single-machine scheduling with unrestricted due date assignment," *Computers & Industrial Engineering*, vol. 79, pp. 148–155, 2015.

[4] Y. Yin, S. R. Cheng, and C. C. Wu, "Parallel-machine scheduling to minimize flowtime, holding, and batch delivery costs," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 6, Article ID 1450044, 17 pages, 2014.

[5] E. L. Demeulemeester and W. S. Herroelen, "New benchmark results for the resource-constrained project scheduling problem," *Management Science*, vol. 43, no. 11, pp. 1485–1492, 1997.

[6] P. Raj, "Solving spatial scheduling problem: an analytical approach," in *Proceedings of the 37th International Conference on Computers and Industrial Engineering*, pp. 2002–2011, 2007.

[7] J. E. Kelley, "The critical-path method: resources planning and scheduling," *Industrial Scheduling*, vol. 13, pp. 347–365, 1963.

[8] Z. Liu and H. Wang, "Heuristic algorithm for RCPSP with the objective of minimizing activities' cost," *Journal of Systems Engineering and Electronics*, vol. 17, no. 1, pp. 96–102, 2006.

[9] T. Bhaskar, M. N. Pal, and A. K. Pal, "A heuristic method for RCPSP with fuzzy activity times," *European Journal of Operational Research*, vol. 208, no. 1, pp. 57–66, 2011.

[10] K. J. Lee, J. K. Lee, and S. Y. Choi, "A spatial scheduling system and its application to shipbuilding: DAS-CURVE," *Expert Systems with Applications*, vol. 10, no. 3-4, pp. 311–324, 1996.

[11] S. Koh, R. Logendran, D. Choi, and S. Woo, "Spatial scheduling for shape-changing mega-blocks in a shipbuilding company," *International Journal of Production Research*, vol. 49, no. 23, pp. 7135–7149, 2011.

[12] B. S. Baker, E. G. Coffman Jr., and R. L. Rivest, "Orthogonal packings in two dimensions," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 846–855, 1980.

[13] B. Chazelle, "bottom-left bin packing heuristic: an efficient implementation," *IEEE Transactions on Computers*, vol. 32, no. 8, pp. 697–707, 1983.

[14] G. Belov, G. Scheithauer, and E. A. Mukhacheva, "One-dimensional heuristics adapted for two-dimensional rectangular strip packing," *Journal of the Operational Research Society*, vol. 59, no. 6, pp. 823–832, 2008.

[15] T. M. Chan, F. Alvelos, E. Silva, and J. M. V. de Carvalho, "Heuristics with stochastic neighborhood structures for two-dimensional bin packing and cutting stock problems," *Asia-Pacific Journal of Operational Research*, vol. 28, no. 2, pp. 255–278, 2011.

[16] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256–267, 2000.

[17] R. Alvarez-Valdes, F. Parreño, and J. M. Tamarit, "A GRASP/Path relinking algorithm for two- and three-dimensional multiple bin-size bin packing problems," *Computers and Operations Research*, vol. 40, no. 12, pp. 3081–3090, 2013.

[18] C.-S. Liao and C.-H. Hsu, "New lower bounds for the three-dimensional orthogonal bin packing problem," *European Journal of Operational Research*, vol. 225, no. 2, pp. 244–252, 2013.

[19] M. Kucuk and M. Ermis, "A new hybrid evolutionary algorithm for three-dimensional packing problems," in *Proceedings of the IEEE International Conference on Systems Man and Cybernetics (SMC '10)*, pp. 4029–4034, IEEE, Istanbul, Turkey, October 2010.

[20] M. Steinbrunn, G. Moerkotte, and A. Kemper, "Heuristic and randomized optimization for the join ordering problem," *VLDB Journal*, vol. 6, no. 3, pp. 191–208, 1997.

[21] S. Kirkpatrick, J. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[22] F. T. S. Chan, A. Prakash, H. L. Ma, and C. S. Wong, "A hybrid Tabu sample-sort simulated annealing approach for solving distributed scheduling problem," *International Journal of Production Research*, vol. 51, no. 9, pp. 2602–2619, 2013.

[23] K. Bouleimen and H. Lecocq, "A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version," *European Journal of Operational Research*, vol. 149, no. 2, pp. 268–281, 2003.

[24] V. Yannibelli and A. Amandi, "Hybridizing a multi-objective simulated annealing algorithm with a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem," *Expert Systems with Applications*, vol. 40, no. 7, pp. 2421–2434, 2013.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 464390, 7 pages
http://dx.doi.org/10.1155/2015/464390

Research Article

Scheduling and Common Due Date Assignment on a Single Parallel-Batching Machine with Batch Delivery

Hua Gong, Biao Zhang, and Wuliang Peng

School of Science, The Key Laboratory of Liaoning Province for Advanced Manufacturing Technology and Equipment, Shenyang Ligong University, Shenyang 100159, China

Correspondence should be addressed to Hua Gong; gonghua1018@sina.com

Received 28 December 2014; Accepted 26 January 2015

Academic Editor: Chin-Chia Wu

We consider a single parallel-batching machine scheduling problem with delivery involving both batching scheduling and common due date assignment. The orders are first processed on the single parallel-batching machine and then delivered in batches to the customers. The batching machine can process several orders at the same time. The processing time of a production batch on the machine is equal to the longest processing time of the orders assigned into this batch. A common due date for all the orders in the same delivery batch and a delivery date for each order need to be determined in order to minimize total weighted flow time. We first prove that this problem is NP hard in the strong sense. Two optimal algorithms by using dynamic programming are derived for the two special cases with a given sequence of orders on the machine and a given batching in the production part, respectively.

1. Introduction

A wide variety of practical problems are closely related to the batching production and delivery in batches considered in this paper. In a supply chain of many industries, production operation and delivery operation are two key operational functions. Many industries first produce products and transport their finished products to the customers directly without holding intermediate inventories. For example, there exist the operations with production of steel ingots and delivery of the finished ingots in the iron and steel industry. The steel ingots need to heat to a high temperature in a soaking pit in order to keep the steady performance of the steel ingots. The soaking pit heats several ingots simultaneously where the soaking pit can be viewed as a parallel-batching machine. Then the steady ingots finished in the soaking pit are directly transported to the corresponding customers in batches. This is the motivation for the considered problem. All ingots in the same shipment for delivery to the customer have a common due date. The common due date is a decision variable. Production operation and delivery operation are linked together without any intermediate step since no inventory is involved. The customer does not want to accept his order earlier than the due date. Thus, the orders are finished before the due date, and they have to be storage in inventory for the industries. It is important to coordinate these two operations and schedule them jointly in order to achieve optimal inventory level in the iron and steel industry. Hence, the coordination between production and delivery has become more practical and become one of the most important topics. In such supply chain application, it is well understood that coordinated production and delivery scheduling can significantly reduce the inventory level and the customer service level for the decision maker.

Motivated by the above-described applications in the supply chain, we address a coordinated scheduling problem of production on a single batching machine and delivery operation in batches. The orders are first processed on the single batching machine and then delivered to the customers directly without intermediate. In the production part, the production schedule specifies the processing sequencing and batching of the orders on the machine. Completed orders are delivered in batches to the customers by homogeneous vehicles. We assume that each delivery shipment can only carry up to a number of orders. The delivery schedule specifies how many batches to use, how to batch, and the delivery date of each batch from the machine. The production batches may be partitioned into delivery batches to determine the delivery date. A common due date for the same delivery batch has to be determined. A delivery batch has the same delivery date and the same due date. The problem is to find a coordinated schedule of production batching and delivery batching such that an objective function that takes into account the inventory level is optimized. The inventory level is measured by a function of the times when the finished orders are delivered to the customers. A common due date for the orders in the same delivery batch is the flow time of each order in this batch. Therefore, the objective is to minimize total weighted flow time of orders.

The described problem in this paper falls into two categories: batch delivery and batching scheduling. Many researchers have made contributions in the coordinated scheduling area with production and delivery. We briefly discuss some work related to the problems where the objective is to minimize the sum of the total flow time and delivery cost. Cheng and Kahlbacher [1] and Cheng et al. [2] study a single machine scheduling problem with batch delivery cost, respectively. Wang and Cheng [3] consider a scheduling problem on the parallel machines with batch deliveries where the batch delivery date is equal to the completion time of the last job in a batch. Hall and Potts [4, 5] consider the single machine and identical-parallel machine scheduling problems with batch deliveries in batches involving the due dates, but without a transporter availability constraint. Yin et al. [6] consider a batch delivery scheduling problem with unrelated or uniform parallel machines. Wan and Zhang [7] consider a coordinated scheduling on parallel identical machines with batch delivery with objective of minimizing the sum of job arrival times. Yin et al. [8] address a single machine scheduling with batch delivery in which jobs have an assignable common due window. Chen [9] gives a detail survey for the production-delivery coordination. Ji et al. [10] study a single machine scheduling problem with release times and job delivery to minimize total weighted completion time of jobs. For the objective with minimizing the flow time and delivery cost, Mazdeh et al. [11] consider a single machine scheduling problem with batch deliveries and job release times. Mazdeh et al. [12] and Selvarajah et al. [13] study the single machine scheduling problem with delivery in batches, respectively. Chen and Vairaktarakis [14] consider single machine and parallel machine scheduling problems with delivery routing for the completed jobs to the customers. Yin et al. [15] consider a scheduling problem of nonresumable and simultaneously available jobs on a single machine and delivery in batches where the machine has a fixed unavailability interval.

Batching is an important feature of many practical industries in the supply chain. The parallel-batching scheduling problems considered in this paper are motivated by burn-in operations in the very large-scale integrated circuit manufacturing. The detail surveys for the batching scheduling problems can be seen in [16–18]. For the coordinated scheduling problem with the parallel-batching machine and transportation, Tang and Liu [19] study a flow shop scheduling problem with a parallel-batching machine where the intermediate transportation is considered between two machines. Tang and Gong [20] consider a coordinated scheduling problem with transportation before and after production scheduling on a single parallel-batching machine. In [19, 20], they assume that the processing time of a batch on the batching machine is a constant.

The problem formulated above combines three types of decisions: scheduling, batching, and due date assignment. The independent and simultaneously available orders have to be scheduled and partitioned into batches during processing and delivery. After the orders in a batch finished processing on the parallel-batching machine, they must be delivered to the customers in batches within limited delivery capacity. In a real supply chain, the managers may not stipulate the delivery for each specific order. In such a coordinated system, the linkage between order production scheduling and delivery dispatching of finished orders is extremely important in order to determine the common due date for each delivery batch. Our work differs from the above where we study not only the batching and sequencing schedule of orders on the parallel-batching machine, but also the batching for the finished orders in the delivery part where the delivery date and the common due date for each delivery batch need to be determined.

The rest of this paper is organized as follows. In Section 2, we describe the problem and introduce the notation to be used. In Section 3, we give a strongly NP-hardness proof by a polynomial-time reduction. In Section 4, we develop the efficient algorithms for two special cases. Finally, our conclusion is presented in the last section.

2. Model and Preliminary Results

In this paper, a coordinated scheduling problem on a single parallel-batching machine with batch delivery is addressed in order to determine the common due date for each delivery. The problem can be described as follows.

(1) All the orders $N = \{J_1, J_2, \ldots, J_n\}$ have the same production-delivery routing: they are first processed on a single parallel-batching machine in the production part, and then they are transported to the customers in batches in the delivery part.

(2) In the production part, all orders and the machine are available at time 0. The parallel-batching machine has a capacity limit B and can process several orders simultaneously. A production batch is defined as a set of orders processed on the parallel-batching machine at the same time. The processing time of a production batch on the machine is equal to the longest processing time of the orders assigned into this batch.

(3) In the delivery part, each delivery shipment will transported by a dedicated vehicle. Due to limited vehicle capacity C, each delivery shipment can carry up to a number of orders. A delivery batch is defined as a set of orders transported together in the same shipment.

(4) The vehicle is stationed at the processing facility at time 0 and must go back to the facility once it finishes a delivery batch. A delivery vehicle can depart from the processing facility only when all the orders to be delivered have finished processing.

(5) All orders are not interruption. When the processing of a batch is executed, it cannot be interrupted, and other orders cannot be introduced into the batch. Orders processed in a batch have the same starting time and completion time on the machine.

(6) The production batches may be partitioned into delivery batches to determine their delivery dates. A common due date for the same delivery batch has to be determined. A delivery batch has the same delivery date and the same due date.

Next, we present the notation that will be used throughout the paper:

p_i: the processing time of order J_i on the parallel-batching machine, $i = 1, 2, \ldots, n$.

t_1: the delivery time of the vehicle from the machine to the customer;

t_2: the return time of the vehicle from the customer back to the machine;

w_i: the weight of order J_i, $i = 1, 2, \ldots, n$;

Decision variables are as follows:

B_k: the production batch k;

C_i: the completion time of order J_i on the machine, $i = 1, 2, \ldots, n$;

D_l: the delivery batch l;

d_l: the common due date of delivery batch D_l;

F_i: the flow time of order J_i which is defined as the time when it arrives at the customer, $i = 1, 2, \ldots, n$, $F_i = d_l$, if $J_i \in D_l$;

$\sum w_i F_i$: total weighted flow time.

Here, the flow times of the orders assigned into the same delivery batch are equal to the common due date of the delivery batch. For convenience, we denote our problem by (P). Next, we present three general properties that will be useful throughout our study.

Lemma 1. *There exists an optimal schedule that production batches of the orders are processed continuously from time 0 without idle time on the single parallel-batching machine.*

Proof. If there is any idle time between two continuous batches in the production part, the latter batch can be moved earlier to be processed on the machine. The objective value is not increased. □

Lemma 2. *There exists an optimal schedule where the delivery date of a delivery batch is either the completion time of the last order on the machine which is included in the delivery batch or the available time of the vehicle immediately.*

Proof. Assume that there exists a delivery batch such that its delivery date does not fit any above condition. That is neither at the completion time of the last order on the machine which is included in the delivery batch nor the available time of the vehicle immediately. Then we can move the delivery batch such that this delivery date can fit either of those conditions. It is easy to see that this delivery batch can be transported at that earlier time to the customer without increasing the total weighted flow time. □

Lemma 3. *The delivery batch must contain all orders which are processed on the machine but not delivered within the scope of the delivery capacity to allow.*

Proof. The finished orders on the parallel-batching machine should be delivered to the customer as early as possible. The delivery batch should contain orders as many as possible which are processed on the machine but not delivered. □

3. Strongly NP Hardness

In this section, we show that problem (P) is strongly NP hard by a reduction from 3-partition problem, which is known to be NP hard in the strong sense (see Garey and Johnson [21]).

The following theorem shows the computational complexity of problem (P).

Theorem 4. *Problem (P) is strongly NP hard.*

Proof. We next perform a polynomial time reduction from 3-partition problem.

3-Partition Problem (3-PP). Given $3h + 1$ positive integers $\{a_1, a_2, \ldots, a_{3h}\}$ and a satisfying $a/4 < a_i < a/2$ for $i = 1, 2, \ldots, 3h$ and $\sum_{i=1}^{3h} a_i = ha$, can $\{1, 2, \ldots, 3h\}$ be partitioned into h sets $S_1, S_2, \ldots, S_h$ such that $\sum_{i \in S_l} a_i = b$ for $1 \le l \le h$? Without loss of generality, if there exists a solution to 3-partition problem, then the elements can be numbered such that $a_{3j-2} + a_{3j-1} + a_{3j} = a$, for $j = 1, 2, \ldots, h$.

Given a 3-PP instance, we construct an instance for (P) as follows.

There are $6h + 6$ orders with machine capacity of 2 and delivery capacity of 6.

Assume that $t_1 = t_2 = a/2$.

Processing times and weights of orders as follows:

$p_i = p_{i+3h} = w_i = w_{i+3h} = a_i$, $i = 1, 2, \ldots, 3h$;

$p_i = 0$, $w_i = a$, $i = 6h + 1, \ldots, 6h + 6$;

threshold: $y = a^2(h^2 + 2h + 3)$.

We prove there exists a schedule for this instance with total weighted flow time less than or equal to y if and only if there exists a feasible solution to 3-partition problem. Note that the processing times of orders $J_{6h+1}, \ldots, J_{6h+6}$ are equal to 0. We first prove the following two properties: in a solution to this instance of (P) with total weighted flow time less than or equal to y, (1) orders $J_{6h+1}, \ldots, J_{6h+6}$ form the first delivery batch to the customer; (2) there is no idle time once the vehicle starts to deliver from time point 0 in the delivery part. Then the related total weighted flow time of these orders in the first delivery batch is $3a^2$.

If Part. Consider that the sets H_l, $l = 1, 2, \ldots, h$, form a solution to 3-partition. The batching for the production and delivery is shown as follows. In the production part, the orders $J_1, \ldots, J_6$ are partitioned into $3h$ production batches on the machine where $B_1 = \{J_1, J_{3h+1}\}$, $B_2 = \{J_2, J_{3h+2}\}, \ldots$, $B_{3h} = \{J_{3h}, J_{6h}\}$. In the delivery part, all the orders are partitioned into $h+1$ delivery batches where each delivery batch contains 6 orders; that is, $D_1 = \{J_{6h+1}, \ldots, J_{6h+6}\}$, $D_2 = \{J_1, J_2, J_3, J_{3h+1} J_{3h+2}, J_{3h+3}\}, \ldots, D_{h+1} = \{J_{3h-1}, J_{3h-2}, J_{3h}, J_{6h-2}, J_{6h-1}, J_{6h}\}$. The delivery date and the common due date of D_l are $(l-1)a$ and $(l-1/2)a$, respectively, for $l = 1, 2, \ldots, h+1$. It is easy to see that total weighted flow time of orders in the above schedule is y.

Only If Part. Suppose that there exists a schedule for the instance of (P) with total weighted flow time less than or equal to y. Since the machine capacity is 2, we can see that each production batch on the machine contains exactly two orders with identical processing times: $B_1 = \{J_1, J_{3h+1}\}$, $B_2 = \{J_2, J_{3h+2}\}, \ldots, B_{3h} = \{J_{3h}, J_{6h}\}$. Due to the delivery capacity of 6, the delivery batches of the orders are at least $h+1$ in the delivery part. Assume that total weighted flow time to minimize a solution with k ($k \geq h$) delivery batches for these $6h$ orders except the first delivery batch. From property (1), we can derive total weighted flow time of this problem as follows:

$$\begin{aligned} \text{Minimize} \quad & 2\left(x_1\left(x_1 + \frac{a}{2}\right) + x_2\left(x_1 + x_2 + \frac{a}{2}\right) \right. \\ & \left. + \cdots + x_k\left(x_1 + \cdots + x_k + \frac{a}{2}\right)\right) + 3a^2 \quad (1) \\ \text{Subject to} \quad & x_1 + \cdots + x_k = ha, \end{aligned}$$

where x_j is total processing time of orders assigned into the same batch D_j, for $j = 1, \ldots, k$. The total weighted flow time can be formulated as

$$\begin{aligned} & \left(\sum_{j=1}^{k} x_j\right)^2 + \sum_{j=1}^{k} x_j^2 + (h+3)\,a^2 \\ & \quad = \sum_{j=1}^{k} x_j^2 + \left(h^2 + h + 3\right) a^2. \end{aligned} \quad (2)$$

Note that the second term is minimized by setting $x_1 = \cdots = x_k = ha/k$. We obtain that total weighted flow time for the k delivery batches is greater than or equal to $\sum_{j=1}^{k}(ha/k)^2 + (h^2 + h + 3)a^2$. Then this expression means a minimum objective value with respect to k at $k = h$, and this minimum total weighted flow time is y. From (1), it means that these $6h$ orders are partitioned into h delivery batches. Furthermore, the minimum total weighted flow time is contributed by $h+1$ delivery batches. Each delivery batch contains exactly 6 orders which are assigned into three production batches, that is, $D_1 = \{J_{6h+1}, \ldots, J_{6h+6}\}$, $D_2 = \{J_1, J_2, J_3, J_{3h+1} J_{3h+2}, J_{3h+3}\}, \ldots, D_{h+1} = \{J_{3h-1}, J_{3h-2}, J_{3h}, J_{6h-2}, J_{6h-1}, J_{6h}\}$. According to (2), the delivery date of the delivery batch D_j is $(j-1)a$, for $j = 1, 2, \ldots, h+1$. The common due date of D_j is $(j - 1/2)a$, for $j = 1, 2, \ldots, h+1$. From the above discussion, we see that $a_{3j-2} + a_{3j-1} + a_{3j} = a$, for $j = 1, 2, \ldots, h$. This means that the 3-Partition problem has a solution. Combining the "if" part and "only if" part, we have proved the theorem. □

4. Solving Two Special Cases

In this section, we consider two special cases of problem (P): (1) the problem where the order sequence on the machine in the production part is predetermined and (2) the problem where the batching on the machine in the production part is predetermined. Case (1) occurs when direct sequence is the production strategy used for the customer, while case (2) occurs when direct batching is the production strategy used for the producer. In practice, a direct sequence strategy will be used when the customer's demands are relatively high, while it is more likely that a direct batching will be used when the production cost is relatively high.

4.1. The Predetermined Sequence of Orders. In this case of the problem, the sequence of orders on the batching machine is predetermined $J_1 \prec J_2 \prec \cdots \prec J_n$. If $J_i \prec J_j$, then order J_i is in preference to order J_j or they are assigned into the same production batch to be processed on the machine. Then the case of (P) can be expressed as (P-case1). In this case, we need to make decisions on the production batching on the machine and delivery batching in the delivery part according to the predetermined sequence of orders. The following dynamic programming algorithm solves the case of the problem.

Algorithm DP1. (a) Number the orders in the predetermined sequence and (b) function values.

Let $g(l,k)$ be the minimum completion time of order J_k on the parallel-batching machine if orders $J_1, \ldots, J_k$ have been scheduled such that the current production batch contains orders $J_{l+1}, \ldots, J_k$ and $0 < k - l \leq B$.

Let $f(i,j)$ be the minimum common due date of the current batch containing $J_{i+1}, \ldots, J_j$ when they are delivered to the customer and $0 < j - i \leq C$.

Define $z(j)$ as the minimum total weighted flow time for the first scheduled orders $J_1, \ldots, J_j$.

(c) Dynamic programming is as follows.

Initial conditions are as follows:

$$f(0,j) = \min_{0<j-k\leq B} \{g(k,j)\} + t_1, \quad j = 1, \ldots, C, \quad (3)$$

where $g(k,j) = \min_{0<k-h\leq B}\{g(h,k) + \max\{p_{k+1}, \ldots, p_j\}\}$, and $g(0,0) = 0$, $z(0) = 0$.

Recurrence relations are as follows:

$$z(j) = \min_{0<j-i\leq C} \left\{ z(i) + \sum_{s=i+1}^{j} w_s f(i,j) \right\}, \quad (4)$$

where $f(i,j) = \min_{0<i-l\leq C}\{\max_{0<j-k\leq B}\{f(l,i) + t_2, g(k,j)\} + t_1\}$ and $g(k,j) = \min_{h\leq l}\{g(h,k) + \max\{p_{k+1}, \ldots, p_j\} \mid i \leq k < \min\{l + B + 1, j\}, 0 < k - h \leq B\}$.

Optimal solution value is $z(n)$.

Theorem 5. *Algorithm DP1 solves problem (P-case 1) in $O(n^4)$ time.*

Proof. Due to the predetermined order sequence, the batching on the machine, the batching, and the delivery date of orders for the delivery need to be decided. From Lemma 2, the delivery date of a new delivery batch has two possibilities: the vehicle transports a current delivery batch immediately or the vehicle waits the last order of the current delivery batch that finished processing on the machine. Therefore, the delivery date of the current delivery batch can be derived as $\max_{0<j-k\leq B}\{f(l,i)+t_2, g(k,j)\}$. Hence, we can determine the common due date of the new delivery batch according to its delivery date. Since the delivery capacity and the production capacity are limited, the relations satisfy that $0 < j-k \leq B$ and $0 < i-l \leq C$. The increase of the total weighted flow time for the current delivery batch is computed by its common due date $f(i,j)$. Then the total weighted flow time of orders $J_1,\ldots,J_j$ is $z(j) = \min_{0<j-i\leq C}\{z(i)+\sum_{s=i+1}^{j} w_s f(i,j)\}$. This proves the correctness of the recursive relations and the optimality of the algorithm.

The time complexity of Algorithm DP1 can be established as follows. Due to $k,l,i,j \leq n$, we can obtain that the overall time complexity of Algorithm DP1 is bounded by $O(n^4)$. □

4.2. The Given Batching of Orders in the Production Part. In this case of the problem, the production batching of orders on the machine is predetermined $B_1, B_2, \ldots, B_k$. Then the case can be expressed as (P-case 2). In this case, we need to make decisions on the sequence of the given production batches on the machine and the delivery batching in the delivery part. Let $P(B_i)$ and $W(B_i)$ be the processing time of production batch B_i and the total weight of production batch B_i, respectively.

Lemma 6. *For problem (P-case 2), there exists an optimal schedule in which the production batches are processed in the nondecreasing $P(B_i)/W(B_i)$ rule on the machine.*

Proof (by contradiction). Consider an optimal schedule π^* formed from a sequence that is not processed in the nondecreasing $P(B_i)/W(B_i)$ rule. In this schedule, there must be at least two adjacent batches, say B_i followed by B_j, such that

$$\frac{P(B_i)}{W(B_i)} > \frac{P(B_j)}{W(B_j)}. \tag{5}$$

Now consider the schedule π' formed by interchanging the positions of these two batches. Clearly, the flow times of all the batches preceding the pair under consideration will remain unaffected and so will the flow times of all the succeeding batches. Let $z(\pi^*)$ and $z(\pi')$ denote the total weighted flow time of $\pi^*$and π', respectively. Now if the completion time of the last batch preceding the pair B_i and B_j is s, then the flow times of two batches in π^* are $F(B_i) = s+P(B_i)+t_1$ and $F(B_j) = s+P(B_i)+P(B_j)+t_1$. We can obtain the flow times of two batches in π' are $F'(B_j) = s+P(B_j)+t_1$ and $F'(B_i) = s+P(B_i)+P(B_j)+t_1 = F(B_i)$. Then

$$\begin{aligned} z(\pi') - z(\pi^*) &= W(B_i)F'(B_i) + W(B_j)F'(B_j) \\ &\quad - W(B_i)F(B_i) - W(B_j)F(B_j) \\ &= W(B_i)F(B_j) - W(B_j)F(B_i) < 0. \end{aligned} \tag{6}$$

Comparing terms, it is clear that total weighted flow time of π' is smaller than that of π^*. Thus interchanging the positions of these two batches B_i and B_j can decrease overall weighted flow time. A finite number of repetitions of this argument establish that there exists an optimal schedule where the production batches are processed in the nondecreasing $P(B_i)/W(B_i)$ rule on the machine. □

Based on Lemma 6, the orders processed in the same production batch have the same starting time and completion time on the machine. Hence, the processing on the machine is not influenced by the sequence of orders in a batch. If the orders of the same production batch can be transported to the customer in the same delivery batch, the sequence of order in one batch is irrelevant to the delivery. Otherwise, the sequence of orders in the same production batch affects the batching decision in the delivery part. We have the following lemma to partition the orders into the delivery batches.

Lemma 7. *For problem (P-case 2), there exists an optimal schedule in which the orders in the same production batch are sequenced in the longest weight rule of orders in the delivery part.*

Proof. Assume that there exists an optimal schedule π^*. If π^* contains orders that are not delivered in the longest weight rule of orders of the same production batch for two adjacent orders J_i and J_j in the same production batch, we assume that J_i is the last order to be delivered in some batch D_l and J_j is the first order to be processed in the next batch D_{l+1} such that $w_i < w_j$ in π^*. Let π' be a schedule obtained by swapping J_i and J_j. After swapping orders J_i and J_j, π' contains two new delivery bathes which contain jobs $D_l \cup \{J_j\} \setminus \{J_i\}$ and $D_{l+1} \cup \{J_i\} \setminus \{J_j\}$. All other orders remain the same schedule in π' as in π^*. Then the flow times of orders J_i and J_j are $F'_i = F_j$ and $F'_j = F_i$. We have

$$\begin{aligned} z(\pi') - z(\pi^*) &= w_j F_i + w_i F_j - w_j F_j - w_i F_i \\ &= (w_i - w_j)(F_j - F_i) < 0. \end{aligned} \tag{7}$$

Therefore, regardless of whether J_i and J_j are transported in the same delivery batch or not, the total weighted flow time associated with π' is less than that associated with π^*. We can establish the property using a finite number of repetitions of this argument such that there exists an optimal schedule where the orders in the same production batch are sequenced in the longest weight rule of orders in the delivery part. □

The following dynamic solves the case of (P-case 2).

Algorithm DP2. (a) Sequence the given production batches $B_1, B_2, \ldots, B_k$ in the nondecreasing $P(B_i)/W(B_i)$ rule on the machine. (b) Calculate the completion time of order J_i on the parallel-batching machine C_i, $i = 1, 2, \ldots, n$. (c) Number the orders in the shortest weight rule in each production batch. (d) Function values are as follows.

Let $f(i, j)$ be the minimum common due date of the current batch containing orders $J_{i+1}, \ldots, J_j$ when they are delivered to the customer, $0 < j - i \le C$ and $l \le j \le n$.

Let $z(j)$ be the minimum total weighted flow time for the first scheduled orders $J_1, \ldots, J_j$.

(e) Dynamic programming is as follows.

Initial conditions are as follows.

$$f(0, j, 1) = C_j + t_1, \quad j = 1, \ldots, C. \tag{8}$$

Recurrence relations are as follows:

$$z(j) = \min \left\{ z(i) + \sum_{s=i+1}^{j} w_s f(i, j, l) \mid 0 < j - i \le C, \left\lceil \frac{j}{C} \right\rceil \le l \le j \right\}, \tag{9}$$

where $f(i, j, l) = \min_{\substack{0<s-i\le C \\ \lceil i/C \rceil \le l-1 \le i}} \{\max_{0<s-i\le C}\{f(s, i, l-1) + t_2, C_j\} + t_1\}$.

Optimal solution value is $z(n)$.

Theorem 8. *Algorithm DP2 can solve problem (P-case 2) in $O(n^3)$ time.*

Proof. By Lemmas 6 and 7, in the schedule for the given production batches $B_1, B_2, \ldots, B_k$, where the production schedule follows the nondecreasing $P(B_i)/W(B_i)$ rule on the parallel-batching machine, the delivery schedule follows the longest weight rule for the orders in the same batch. In the recursive relations, the function value is computed by trying every possible size $\{i + 1, \ldots, j\}$ of the last delivery batch and the delivery date of this batch. The common due date contributed by the last batch is $\max_{0<s-i\le C}\{f(s, i, l-1) + t_2, G_j\}$. This proves the correctness of the recursive relations and the optimality of the algorithm.

The time complexity of Algorithm DP2 can be established as follows. Sorting the production batches in the nondecreasing $P(B_i)/W(B_i)$ rule requires $O(n \log n)$ time. Sequencing the orders of the same batch in the longest weight rule takes $O(n \log n)$ time. In addition, by definition, we have $l, i, j \le n$. Therefore, the overall time complexity of Algorithm DP 2 is bounded by $O(n^3)$. □

5. Conclusions

In this paper, we propose a coordinated scheduling problem with production operation and delivery operation in the supply chain with batching scheduling and common due date assignment. The orders are processed on a single parallel-batching machine and then the finished orders are transported in batches to the customer. The parallel-batching machine has a limited capacity and the number of orders in each delivery batch cannot exceed the delivery capacity. All orders in the same delivery batch to the customer have a common due date. The common due date of each delivery batch needs to be determined. Inventory level in the supply chain is measured by a function of the common due date for each delivery batch when the completed orders are delivered to the customers. The objective is to minimize total weighted flow time. We first prove that this problem is NP hard in the strong sense. For the two special cases with a given sequence of orders on the machine and a given batching in the production part, we derive two optimal algorithms based on the dynamic programming, respectively. A natural extension of this work would consider approximation algorithms for the problem with the objective of the total weighted flow time. Another extension of this work would consider other objective functions.

Conflict of Interests

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

Acknowledgments

This research is partly supported by the National Natural Science Foundation of China (Grant no. 71101097 and Grant no. 71071100), the National High Technology Research and Development Program of China (863, Grant no. 2014AA041401), and the Liaoning BaiQianWan Talents Program (Grant no. 2014921043).

References

[1] T. C. E. Cheng and H. G. Kahlbacher, "Scheduling with delivery and earliness penalties," *Asia-Pacific Journal of Operational Research*, vol. 10, pp. 145–152, 1993.

[2] T. C. E. Cheng, V. S. Gordon, and M. Y. Kovalyov, "Single machine scheduling with batch deliveries," *European Journal of Operational Research*, vol. 94, no. 2, pp. 277–283, 1996.

[3] G. Wang and T. C. E. Cheng, "Parallel machine scheduling with batch delivery costs," *International Journal of Production Economics*, vol. 68, no. 2, pp. 177–183, 2000.

[4] N. G. Hall and C. N. Potts, "Supply chain scheduling: batching and delivery," *Operations Research*, vol. 51, no. 4, pp. 566–584, 2003.

[5] N. G. Hall and C. N. Potts, "The coordination of scheduling and batch deliveries," *Annals of Operations Research*, vol. 135, pp. 41–64, 2005.

[6] Y. Yin, S.-R. Cheng, and C.-C. Wu, "Parallel-machine scheduling to minimize flowtime, holding, and batch delivery costs," *Asia-Pacific Journal of Operational Research*, vol. 31, no. 6, pp. 1–17, 2014.

[7] L. Wan and A. Zhang, "Coordinated scheduling on parallel machines with batch delivery," *International Journal of Production Economics*, vol. 150, pp. 199–203, 2014.

[8] 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.

[9] Z.-L. Chen, "Integrated production and outbound distribution scheduling: review and extensions," *Operations Research*, vol. 58, no. 1, pp. 130–148, 2010.

[10] M. Ji, Y. He, and T. C. Cheng, "Batch delivery scheduling with batch delivery cost on a single machine," *European Journal of Operational Research*, vol. 176, no. 2, pp. 745–755, 2007.

[11] M. M. Mazdeh, M. Sarhadi, and K. S. Hindi, "A branch-and-bound algorithm for single-machine scheduling with batch delivery and job release times," *Computers & Operations Research*, vol. 35, no. 4, pp. 1099–1111, 2008.

[12] M. M. Mazdeh, S. Shashaani, A. Ashouri, and K. S. Hindi, "Single-machine batch scheduling minimizing weighted flow times and delivery costs," *Applied Mathematical Modelling*, vol. 35, no. 1, pp. 563–570, 2011.

[13] E. Selvarajah, G. Steiner, and R. Zhang, "Single machine batch scheduling with release times and delivery costs," *Journal of Scheduling*, vol. 16, no. 1, pp. 69–79, 2013.

[14] Z.-L. Chen and G. L. Vairaktarakis, "Integrated scheduling of production and distribution operations," *Management Science*, vol. 51, no. 4, pp. 614–628, 2005.

[15] Y. Yin, D. Ye, and G. Zhang, "Single machine batch scheduling to minimize the sum of total flow time and batch delivery cost with an unavailability interval," *Information Sciences*, vol. 274, pp. 310–322, 2014.

[16] P. Brucker, A. Gladky, H. Hoogeveen et al., "Scheduling a batching machine," *Journal of Scheduling*, vol. 1, no. 1, pp. 31–54, 1998.

[17] C. N. Potts and M. Y. Kovalyov, "Scheduling with batching: a review," *European Journal of Operational Research*, vol. 120, no. 2, pp. 228–249, 2000.

[18] C. Y. Lee, R. Uzsoy, and L. A. Martin-Vega, "Efficient algorithms for scheduling semiconductor burn-in operations," *Operations Research*, vol. 40, no. 4, pp. 764–775, 1992.

[19] L. Tang and P. Liu, "Flowshop scheduling problems with transportation or deterioration between the batching and single machines," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1289–1295, 2009.

[20] L. Tang and H. Gong, "The coordination of transportation and batching scheduling," *Applied Mathematical Modelling*, vol. 33, no. 10, pp. 3854–3862, 2009.

[21] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, NY, USA, 1979.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2014, Article ID 210761, 4 pages
http://dx.doi.org/10.1155/2014/210761

Research Article

A Unified Weight Formula for Calculating the Sample Variance from Weighted Successive Differences

Kuo-Hung Lo,[1,2] Tien-Lung Sun,[1] and Juei-Chao Chen[3]

[1] *Department of Industrial Engineering and Management, Yuan Ze University, Zhongli 32003, Taiwan*
[2] *Department of Marketing & Logistics Management, Yu Da University of Science and Technology, Miaoli City 36100, Taiwan*
[3] *Department of Statistics and Information Science, Fu Jen Catholic University, New Taipei City 24205, Taiwan*

Correspondence should be addressed to Juei-Chao Chen; 006884@mail.fju.edu.tw

Received 1 July 2014; Accepted 6 August 2014; Published 27 August 2014

Academic Editor: Chin-Chia Wu

The basic formula to calculate sample variance is based on the sum of squared differences from mean. From computational perspective, mean calculation is nondesired as it can introduce computing errors. Previous researches have proposed to use weighted formula of the successive differences to calculate sample variance to avoid mean calculation. But their weighted formula is not in a unified format in the sense that it has to be represented as two formulas. This paper proposes a unified weight formula for sample variance calculation from weighted successive differences. A proof is provided to show that sample variance calculated using the proposed unified weighted formula is mathematically equivalent to the basic definition.

1. Introduction

Sample variance calculation is a fundamental task in many data analysis applications. The basic formula for calculating a sample variance is based on the sum of squared differences from mean. Given that a set of data is $x_1, x_2, \ldots, x_n$, the sample variance, denoted as S_n^2, is calculated as follows:

$$S_n^2 = \frac{\mathrm{SS}_n}{(n-1)}, \quad n \geq 2, \tag{1}$$

where $\mathrm{SS}_n = \sum_{i=1}^{n} (x_i - \overline{x}_n)^2$ is the sum of squared differences from the mean and $\overline{x}_n = (1/n) \sum_{i=1}^{n} x_i$ is the sample mean. Von Neumann et al. [1] pointed out that (1) does not take into account the order of the observations. They proposed to instead use successive differences of data so that the order can be considered. Specifically, they used

$$D_n^2 = \frac{1}{2(n-1)} \sum_{i=2}^{n} d_i^2, \tag{2}$$

where the subscript i refers to temporal order of the data and $d_i = x_i - x_{i-1}$. Define $\{x_0, d_1, \ldots, d_n\}$ as the successive differences of the input data. From computational perspective, von Neumann's formula is also advantageous as it avoids a mean calculation that may introduce rounding errors.

The problem with D_n^2 is that it is not mathematically equivalent to the basic definition. This problem was independently solved by Eilon and Chowdhury [2] and Joarder [3] where weighted successive differences were used to derive a formula that is mathematically equivalent to the basic definition.

Eilon and Chowdhury [2] considered a job scheduling problem where they wanted to minimize the variance of the job's waiting time. Let y_i be the waiting time of the ith job. By definition, $y_1 = 0$, as the first job does not have waiting time, and $y_i = \sum_{j=1}^{i-1} p_j$, for $i = 1, 2, \ldots, n$, where n is the number of jobs and p_j is the processing time of job j. The objective is to minimize the variance of the waiting time, or equivalently $\mathrm{SS}_n = \sum_{i=1}^{n} (y_i - \overline{y}_n)^2$. For this purpose, there is a need to quickly update SS_n when job i and j are swapped. Notice that when job i and j are swapped, most of the jobs' waiting time will change accordingly, and $\overline{y}_n$ and SS_n have to be recalculated. To avoid recalculating $\overline{y}_n$ when updating SS_n, Eilon and Chowdhury derived a formula to calculate SS_n from successive differences. By definition, the successive differences of the waiting time are the processing time; that is, $p_i = y_{i+1} - y_i$, $i = 1, 2, \ldots, n-1$. So,

$$\sum_{i=1}^{n}\left(y_i - \overline{y}_n\right)^2 = \frac{1}{n}\left(\sum_{i=1}^{n-1}(n-i)\,ip_i^2 + \sum_{i,j=1,i\neq j}^{n-1}(n-i)\,jp_ip_j\right) = \frac{1}{n}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}c_{ij}^{*}p_ip_j, \quad (3)$$

where

$$c_{ij}^{*} = \begin{cases}(n-i)\,j, & i \geq j,\\ (n-j)\,i, & i < j,\end{cases} \quad i, j = 1, 2, \ldots, n-1. \quad (4)$$

Equation (3) is not a general formula for calculating SS_n as y_1 is zero. Vani and Raghavachari [4] gave a more general formula by considering the job's completion time rather than waiting time. Let $z_i = \sum_{j=1}^{i} p_j$ be the completion time of job i. They rewrote (3) as follows:

$$\begin{aligned}\sum_{i=1}^{n}\left(z_i - \overline{z}_n\right)^2 &= \frac{1}{n}\sum_{i=1}^{n}(n-i+1)(i-1)\,p_i^2 \\ &\quad + \frac{2}{n}\sum_{i=1}^{n}(i-1)\,p_i\sum_{j=i+1}^{n}(n-j+1)\,p_j \\ &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n}c_{ij}p_ip_j,\end{aligned} \quad (5)$$

where

$$c_{ij} = \begin{cases}(n-i+1)(j-1), & i \geq j,\\ (n-j+1)(i-1), & i < j,\end{cases} \quad (6)$$

for $i, j = 1, 2, \ldots, n$. In an independent work, Joarder [3] also derived a formula similar to (5). He then converted its double sum structure into a quadratic form wherein

$$\sum_{i=1}^{n}\left(x_i - \overline{x}_n\right)^2 = \frac{1}{n}\mathbf{d}_{1\times n}^{T}\mathbf{c}_{n\times n}\mathbf{d}_{n\times 1}, \quad (7)$$

where $\mathbf{d}_{1\times n}^{T} = \left[d_1 \;\; d_2 \;\; \cdots \;\; d_n\right]_{1\times n}$ is a vector of the successive differences and $\mathbf{c}_{n\times n} = \left[c_{ij}\right]_{n\times n}$ is a weight matrix with c_{ij} as defined in (6).

One problem with the weight formula in (6) is that it is not in a unified format but has to be represented as two formulas. This deficiency prohibits a compact representation that would facilitate further derivations. To solve this problem, we derive a unified weight formula for sample variance calculations from weighted successive differences. Joarder [3] derived an updating formula to calculate a variance from weighted successive differences. But, his formula contains a dynamically increased number of updating items. Using the unified weight formula, we show [5] that we can improve Joarder's formula by reducing the updating items to a fixed number of only two items.

2. Main Results

Theorem 1. *Given that a temporally order of the observations $x_1, x_2, \ldots, x_n$ the sum of squared differences about the mean can be represented as*

$$SS_n = \sum_{i=1}^{n}\left(x_i - \overline{x}_n\right)^2 = \frac{1}{n}\mathbf{d}_{1\times n}^{T}\mathbf{w}_{n\times n}\mathbf{d}_{n\times 1}, \quad (8)$$

where $\mathbf{d}_{1\times n}^{T} = [d_1 \;\; d_2 \;\; \cdots \;\; d_n]_{1\times n}$, $\{d_i = x_i - x_{i-1}\}$, for $x_0 = 0$, $i = 1, 2, \ldots, n$, and $\mathbf{w}_{n\times n} = \left[w_{ij}\right]_{n\times n}$ are the $n \times n$ symmetric matrix with

$$w_{ij} = (n+1)(i+j-1) - ij - \frac{n}{2}\left(i + j + \left|i - j\right|\right), \quad i, j = 1, 2, \ldots, n. \quad (9)$$

Proof. First write

$$\sum_{i=1}^{n}\left(x_i - \overline{x}_n\right)^2 = \left(\mathbf{x}_{n\times 1} - \overline{\mathbf{x}}_{n\times 1}\right)^{T}\left(\mathbf{x}_{n\times 1} - \overline{\mathbf{x}}_{n\times 1}\right), \quad (10)$$

for $\mathbf{x}^T = \left[x_1 \;\; x_2 \;\; \cdots \;\; x_n\right]_{1\times n}$. Now, $\mathbf{x}$ can be presented as

$$\mathbf{x}^T = \left[d_1 \;\; \sum_{i=1}^{2}d_i \;\; \cdots \;\; \sum_{i=1}^{n}d_i\right]_{1\times n} = \mathbf{P}_{n\times n}\mathbf{d}_{n\times 1}, \quad (11)$$

for

$$\mathbf{P}_{n\times n} = \begin{bmatrix}1 & 0 & \cdots & 0\\ 1 & 1 & \ddots & \vdots\\ \vdots & \vdots & \ddots & 0\\ 1 & 1 & \cdots & 1\end{bmatrix}_{n\times n}, \quad (12)$$

that is, the row i column j element of $\mathbf{P}_{n\times n}$ is

$$p_{ij} = \begin{cases}1, & j \leq i\\ 0, & j > i,\end{cases} \quad (13)$$

for $i, j = 1, 2, \ldots, n$.

Next, the mean of x can be written as

$$\overline{x}_n = \frac{1}{n}\sum_{i=1}^{n}x_i = \frac{1}{n}\sum_{i=1}^{n}(n-i+1)\,d_i. \quad (14)$$

In vector form this is

$$\overline{\mathbf{x}}_{n\times 1} = \mathbf{Q}_{n\times n}\mathbf{d}_{n\times 1}, \quad (15)$$

where

$$\mathbf{Q}_{n\times n} = \frac{1}{n}\begin{bmatrix}n & n-1 & \cdots & 1\\ n & n-1 & \cdots & 1\\ \vdots & \vdots & \cdots & \vdots\\ n & n-1 & \cdots & 1\end{bmatrix}_{n\times n}; \quad (16)$$

the row i column j element of $\mathbf{Q}_{n\times n}$ is

$$q_{ij} = \left[\frac{1}{n}(n-j+1)\right]_{n\times n}, \quad i, j = 1, 2, \ldots, n. \tag{17}$$

Now observe that

$$\begin{aligned}
&(\mathbf{x}_{n\times 1} - \overline{\mathbf{x}}_{n\times 1})^T (\mathbf{x}_{n\times 1} - \overline{\mathbf{x}}_{n\times 1}) \\
&= [(\mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n})\mathbf{d}_{n\times 1}]^T [(\mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n})\mathbf{d}_{n\times 1}] \\
&= \mathbf{d}_{1\times n}^T (\mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n})^T (\mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n})\mathbf{d}_{n\times 1} \\
&= \mathbf{d}_{1\times n}^T \left(\mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n} - \mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n} \right. \\
&\qquad \left. - \mathbf{Q}_{n\times n}^T \mathbf{P}_{n\times n} + \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n}\right)\mathbf{d}_{n\times 1}.
\end{aligned} \tag{18}$$

Thus we need to obtain expressions for calculating $\mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n}$, $\mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n}$, $\mathbf{Q}_{n\times n}^T \mathbf{P}_{n\times n}$, and $\mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n}$.

First

$$\begin{aligned}
\mathbf{L}_{n\times n} &= \mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n} \\
&= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{n\times n} \cdot \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{n\times n} \\
&= \left[\sum_{m=1}^{n} p_{im} p_{mj}\right]_{n\times n} = \left[l_{ij}\right]_{n\times n},
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
l_{ij} &= \begin{cases} n - j + 1, & i \le j \\ n - i + 1, & i > j \end{cases} \\
&= n - \frac{1}{2}(i + j + |i - j| - 2), \quad i, j = 1, 2, \ldots, n.
\end{aligned} \tag{20}$$

Then

$$\begin{aligned}
\mathbf{B}_{n\times n} &= \mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n} \\
&= \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}_{n\times n} \cdot \frac{1}{n} \begin{bmatrix} n & n-1 & \cdots & 1 \\ n & n-1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n & n-1 & \cdots & 1 \end{bmatrix}_{n\times n} \\
&= \left[\sum_{m=1}^{n} p_{im} q_{mj}\right]_{n\times n} = \left[b_{ij}\right]_{n\times n},
\end{aligned} \tag{21}$$

where

$$\begin{aligned}
b_{ij} &= \frac{1}{n}(n - j + 1)(n - i + 1), \quad i, j = 1, 2, \ldots, n, \\
b_{ij} &= (b_{ij})^T = \frac{1}{n}(n - i + 1)(n - j + 1), \quad i, j = 1, 2, \ldots, n.
\end{aligned} \tag{22}$$

That is,

$$\mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n} = \left(\mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n}\right)^T = \mathbf{Q}_{n\times n}^T \mathbf{P}_{n\times n}. \tag{23}$$

Finally,

$$\begin{aligned}
\mathbf{R}_{n\times n} &= \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n} \\
&= \frac{1}{n} \begin{bmatrix} n & n & \cdots & n \\ n-1 & n-1 & \cdots & n-1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}_{n\times n} \\
&\quad \cdot \frac{1}{n} \begin{bmatrix} n & n-1 & \cdots & 1 \\ n & n-1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ n & n-1 & \cdots & 1 \end{bmatrix}_{n\times n} \\
&= \left[\sum_{m=1}^{n} q_{im} q_{mj}\right]_{n\times n} = \left[r_{ij}\right]_{n\times n},
\end{aligned} \tag{24}$$

where

$$r_{ij} = \frac{1}{n}(n - i + 1)(n - j + 1), \quad i, j = 1, 2, \ldots, n. \tag{25}$$

We now can see that $\mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n} = \mathbf{Q}_{n\times n}^T \mathbf{P}_{n\times n} = \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n}$ and, hence,

$$\begin{aligned}
&\mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n} - \mathbf{P}_{n\times n}^T \mathbf{Q}_{n\times n} - \mathbf{Q}_{n\times n}^T \mathbf{P}_{n\times n} + \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n} \\
&\quad = \mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n}.
\end{aligned} \tag{26}$$

A direct calculation produces as follows:

$$\begin{aligned}
&l_{ij} - r_{ij} \\
&= n - \frac{1}{2}(i + j + |i - j| - 2) - \frac{1}{n}(n - j + 1)(n - i + 1) \\
&= \frac{1}{n}\left(n^2 - \frac{n}{2}(i + j + |i - j|) + n - (n - i + 1)(n - j + 1)\right) \\
&= \frac{1}{n}\left((n + 1)(i + j - 1) - ij - \frac{n}{2}(i + j + |i - j|)\right) = \frac{1}{n} w_{ij}.
\end{aligned} \tag{27}$$

Thus, $\mathbf{P}_{n\times n}^T \mathbf{P}_{n\times n} - \mathbf{Q}_{n\times n}^T \mathbf{Q}_{n\times n} = \mathbf{w}_{n\times n}$ and the proof is complete. □

3. Numerical Example

This section gives a numerical example to illustrate sample variance calculation using the nonunified weight formula c_{ij} given in (6) and the unified formula w_{ij} given in (9). We take a sample data set $x_1 = 5$, $x_2 = 14$, $x_3 = 9$, and $x_4 = 6$ from Ross [6, Page 145] where the data are used to illustrate the variance updating process using the one-pass algorithm proposed in van Reeken [7]. The successive differences for this data set are

$$\begin{aligned}
d_1 &= 5, \qquad d_2 = 14 - 5 = 9, \\
d_3 &= 9 - 14 = -5, \qquad d_4 = 6 - 9 = -3,
\end{aligned} \tag{28}$$

and the successive differences vector is $\mathbf{d}^T_{1\times 4} = [\,5\ \ 9\ \ -5\ \ -3\,]_{1\times 4}$. Using the nonunified weight formula, the weight matrix $\mathbf{c}_{4\times 4} = [c_{ij}]_{4\times 4}$ is constructed using two formulas: one for the lower triangular matrix and one for the strictly upper triangular matrix. For the lower triangular matrix with $i \geq j$, $c_{ij} = (n - i + 1)(j - 1)$. For example, $c_{11} = (4 - 1 + 1)(1 - 1) = 0$, $c_{21} = (4 - 2 + 1)(1 - 1) = 0$, $c_{31} = (4 - 3 + 1)(1 - 1) = 0$, and $c_{32} = (4 - 3 + 1)(2 - 1) = 2$. For the strictly upper triangular matrix with $i < j$, the weight formula is $c_{ij} = (n - j + 1)(i - 1)$. For example, $c_{12} = (4 - 2 + 1)(1 - 1) = 0$, $c_{13} = (4 - 3 + 1)(1 - 1) = 0$, and $c_{23} = (4 - 3 + 1)(2 - 1) = 2$. Combining the lower triangular matrix and the strictly upper triangular matrix we can get

$$\mathbf{c}_{4\times 4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}. \tag{29}$$

The variance is then calculated as

$$\begin{aligned} S_4^2 &= \frac{1}{4-1} \cdot \frac{1}{4} \mathbf{d}^T_{1\times 4} \mathbf{c}_{4\times 4} \mathbf{d}_{4\times 1} \\ &= \frac{1}{12} \begin{bmatrix} 5 & 9 & -5 & -3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ -5 \\ -3 \end{bmatrix} = \frac{49}{3}. \end{aligned} \tag{30}$$

Now with our approach, the weight matrix $\mathbf{W}_{4\times 4} = [w_{ij}]_{4\times 4}$ is constructed using the unified weight formula given in (9). For example, $w_{12} = (4+1)(1+2-1) - 1\times 2 - (4/2)(1 + 2 + |1 - 2|) = 0$ and $w_{21} = (4 + 1)(2 + 1 - 1) - 2 \times 1 - (4/2)(2+1+|2-1|) = 0$. Similarly, w_{23} and w_{32} are calculated as $w_{23} = (4+1)(2+3-1) - 2\times 3 - (4/2)(2+3+|2-3|) = 2$ and $w_{32} = (4+1)(3+2-1) - 3\times 2 - (4/2)(3+2+|3-2|) = 2$. The other weights are calculated in a similar manner to produce

$$\mathbf{W}_{4\times 4} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix}. \tag{31}$$

The variance is then calculated as

$$\begin{aligned} S_4^2 &= \frac{1}{4-1} \cdot \frac{1}{4} \mathbf{d}^T_{1\times 4} \mathbf{w}_{4\times 4} \mathbf{d}_{4\times 1} \\ &= \frac{1}{12} \begin{bmatrix} 5 & 9 & -5 & -3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 2 & 1 \\ 0 & 2 & 4 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 5 \\ 9 \\ -5 \\ -3 \end{bmatrix} = \frac{49}{3}. \end{aligned} \tag{32}$$

4. Conclusions

Sample variance calculation using weighted successive differences is advantageous from a computational perspective as it avoids a mean calculation which may introduce rounding errors. However, the weight formula that has been proposed in previous research is not in a unified format. Instead, it has to be represented as two formulas. This deficiency prohibits compact representation of further derivations. This paper derives a unified weight formula for calculating a sample variance from weighted successive differences. We have employed this compute formula to improve variance updating formula in Vani and Raghavachari [4] or Joarder [3].

Conflict of Interests

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

References

[1] J. von Neumann, R. H. Kent, H. R. Bellinson, and B. I. Hart, "The mean square successive difference," *The Annals of Mathematical Statistics*, vol. 12, pp. 153–162, 1941.

[2] S. Eilon and I. G. Chowdhury, "Minimising waiting time variance in the single machine problem," *Management Science*, vol. 23, no. 6, pp. 567–575, 1977.

[3] A. H. Joarder, "Sample variance and first-order differences of observations," *The Mathematical Scientist*, vol. 28, no. 2, pp. 129–133, 2003.

[4] V. Vani and M. Raghavachari, "Deterministic and random single machine sequencing with variance minimization," *Operations Research*, vol. 35, no. 1, pp. 111–120, 1987.

[5] T. L. Sun, K. H. Lo, and J. C. Chen, "An updating formula to calculate sample variance from weighted successive differences," manuscript.

[6] S. M. Ross, *Introduction to Probability and Statistics for Engineers and Scientists*, Wiley, New York, NY, USA, 1987.

[7] A. J. van Reeken, "Dealing with neelys algorithms," *Communications of the ACM*, vol. 11, no. 3, pp. 149–150, 1968.

Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2014, Article ID 701074, 5 pages
http://dx.doi.org/10.1155/2014/701074

Research Article

Multiple-Decision Procedures for Testing the Homogeneity of Mean for k Exponential Distributions

Han-Ching Chen, Her Pei Shan, and Nae-Sheng Wang

Department of Statistics, Feng Chia University, No. 100, Wenhua Road, Xitun District, Taichung City 407, Taiwan

Correspondence should be addressed to Han-Ching Chen; hang.ching11@msa.hinet.net

Received 25 June 2014; Accepted 4 August 2014; Published 19 August 2014

Academic Editor: Yunqiang Yin

In multiple-decision procedures, a crucial objective is to determine the association between the probability of a correct decision (CD) and the sample size. A review of some methods is provided, including a subset selection formulation proposed by Huang and Panchapakesan, a multidecision procedure for testing the homogeneity of means by Huang and Lin, and a similar procedure for testing the homogeneity of variances by Lin and Huang. In this paper, we focus on the use of the Lin and Huang method for testing the null hypothesis H_0 of homogeneity of means for k exponential distributions. We discuss the decision rule R, evaluation of the critical value C, and the infimum of $P(\mathrm{CD} \mid R)$ for k independent random samples from k exponential distributions. In addition, we also observed that a lower bound for the probability of CD relative to the number of the common sample size is determined based on the desired probability of CD when the largest mean is sufficiently larger than the other means. We explain the results by using two examples.

1. Introduction

A multiple-decision problem can be defined as a situation where a person or a group of people must select the number of possible actions from a given finite set. Gupta and Huang [1] and Lin and Gupta [2] presented the selection procedures relevant to multiple-decision theory, including indifference zone selection and subset selection. They suggested that preferences among alternatives can be determined by maximizing the expected value of a numerical utility function or equivalently minimizing the expected value of a loss function. They indicated that the subset selection procedures have been studied and applied widely in determining the required sample size, which is the number of replications or batches used for selecting the optimal population among k populations and for selecting a subset.

Huang and Panchapakesan [3] suggested a modification of the subset selection formulation on the largest mean and the smallest variance. Huang and Lin [4] presented a multidecision procedure for testing the homogeneity of means when the sample sizes and unknown variance are unequal. Lin and Huang [5] used a similar procedure for testing the hypothesis H_0 regarding the homogeneity of the variances. The purpose of this paper was to use the Lin and Huang method for testing the hypothesis H_0 regarding the homogeneity of the means for k exponential distributions. When H_0, the hypothesis, is rejected, the main objective was to obtain a nonempty subset E of the k populations that will include the population related to the largest means (called the best population). In this case, a correct decision (CD) is said to occur if the selected subset E contains the best populations.

The paper is organized as follows. In Section 2, we introduce the definitions and notations of decision rule R for k exponential distributions. In Section 3, we discuss the evaluation of the critical value of our test and the infimum of the probability of a correct decision CD. In Section 4, the performance of the method is illustrated with two examples and the behavior of our procedure is analyzed. Finally, concluding remarks are provided in Section 5.

2. Related Concepts of the Decision Rule

In this section, we use the Lin and Huang [5] method to identify the decision rule R for k exponential distributions.

Let $X_{i1}, X_{i2}, \ldots, X_{in_i}$, $i = 1, 2, \ldots, k$, be k independent random samples from k exponential distribution $\Gamma(1, \theta_i)$, $\theta_i > 0$, $i = 1, 2, \ldots, k$. We define the $\tau_i = \theta_i / (\prod_{j=1}^{k} \theta_j)^{1/k}$ as the distance between θ_i and all other θ_j's.

Then the MLE of τ_i is

$$\widehat{\tau}_i = \frac{\widehat{\theta}_i}{\left(\prod_{j=1}^{k} \widehat{\theta}_j\right)^{1/k}}, \tag{1}$$

where

$$\widehat{\theta}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} = \overline{X}_{i.} \quad i = 1, 2, \ldots, k. \tag{2}$$

For testing $H_0 : \theta_1 = \theta_2 = \ldots, \theta_k$, the test statistic that arises naturally is $\max_{1 \le i \le k} \widehat{\tau}_i$.

We now present the steps of decision rule R for k exponential distribution as follows.

First, given α, where $0 < \alpha < 1$, we want to find a C such that the condition

$$P \left\{ \max_{1 \le i \le k} \widehat{\tau}_i \ge C \mid H_0 \right\} \le \alpha, \tag{3}$$

where C is the critical value for the decision rule R and α is a given probability of Type I error at level α.

Second, given $\Delta > 0$ and P^*, where $1/k < P^* < 1$, we want to find a nonempty subset $E = \{1 \le j \le k \mid \widehat{\tau}_j \ge C\}$ of the k populations that contains the best populations and it is necessary that $\inf_{\underline{\theta} \in \Omega_\Delta} P\{\mathrm{CD} \mid R\} \ge P^*$, where $P(\mathrm{CD} \mid R) = P_{\Omega_\Delta}(\max_{1 \le i \le k} \widehat{\tau}_i \ge C$ and $\widehat{\tau}_{(k)} \ge C)$ and $\Omega_\Delta = \{\underline{\theta} = (\theta_1, \ldots, \theta_k) \mid \tau_{[k]} \ge \Delta\}$, where $\tau_{[1]} \le \tau_{[2]} \le \cdots \le \tau_{[k]}$ denote the ordered τ_i and $\widehat{\tau}_i = \widehat{\theta}_i / (\prod_{j=1}^{k} \widehat{\theta}_j)^{1/k}$ and $\widehat{\theta}_{(k)}$ is associated with the population having the largest $\theta_{[k]}$.

3. Assessment of the Critical Value C and the Infimum of $P(\mathrm{CD} \mid R)$

In this section, we want to estimate the critical value C and the infimum of $P(\mathrm{CD} \mid R)$ for k exponential distribution.

Lemma 1. *Let $X_{i2}, \ldots, X_{in_i}$, $i = 1, 2, \ldots, k$, be k independent random samples from k exponential distribution $\Gamma(1, \theta_i)$, $i = 1, 2, \ldots, k$. The MLE of θ_i is*

$$\begin{gathered} \widehat{\theta}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij} = \overline{X}_{i.} \quad i = 1, 2, \ldots, k, \\ n_i \widehat{\theta}_i \sim \Gamma(n_i, \theta i), \qquad \frac{\widehat{\theta}_i}{\theta_i} \sim \Gamma\left(n_i, \frac{1}{n_i}\right). \end{gathered} \tag{4}$$

Lemma 2. *The MLE of τ_i is*

$$\begin{gathered} \widehat{\tau}_i = \frac{\widehat{\theta}_i}{\left(\prod_{j=1}^{k} \widehat{\theta}_j\right)^{1/k}}, \quad \text{for } \widehat{\theta}_i = \overline{X}_{i.}\ i = 1, 2, \ldots, k, \\ \ln \widehat{\tau}_i - \ln \tau_i = \sum_{j=1}^{k} c_j \frac{\ln \widehat{\theta}_j}{\theta_j}, \end{gathered} \tag{5}$$

where

$$c_j = \begin{cases} 1 - \dfrac{1}{k} & j = i, \\ -\dfrac{1}{k} & j \ne i. \end{cases} \tag{6}$$

Thus, $\ln \widehat{\tau}_i - \ln \tau_i$ is a linear combination of independent log-gamma random variables with coefficients $(1 - (1/k))$ for $\ln(\widehat{\theta}_i/\theta_i)$ and $-1/k$ for $\ln(\widehat{\theta}_j/\theta_j)$, $j = 1, 2, \ldots, k$.

Lemma 3. *According to the Lin and Huang [5] appendix, we can get*

$$\Pr(\ln \widehat{\tau}_i - \ln \tau_i < C) \cong \frac{1}{1 + \exp\{-\pi (C - b_i) / \sqrt{3a_i}\}}, \tag{7}$$

where

$$a_i = \frac{(k-1)^2}{k^2 n_i} + \frac{1}{k^2} \sum_{\substack{j=1 \\ j \ne i}}^{k} \frac{1}{n_j}, \qquad b_i = -\frac{1}{2n_i} + \frac{1}{k} \sum_{j=1}^{k} \frac{1}{2n_j}. \tag{8}$$

Theorem 4. *Under the same assumption of Lemma 1, for testing $H_0 : \theta_1 = \theta_2 = \ldots, \theta_k$, given the samples sizes $n_1, \ldots, n_k$ and $0 < \alpha < 1$, the critical value C for the decision rule R satisfies the $\Pr\{\max_{1 \le i \le k} \widehat{\tau}_i \ge C \mid H_0\} \le \alpha$ which is approximately $C = \exp((\sqrt{3a_{[k]}}/\pi) \ln((k/\alpha) - 1) + b_{[k]})$, where $a_{[k]} = \max_i a_i$, $b_{[k]} = \max_i b_i$ and (a_i, b_i) are given by (8). Further, given $\Delta > 0$, one then has $\inf_{\underline{\theta} \in \Omega_\Delta} P\{\mathrm{CD} \mid R\} \ge 1/(1 + \exp\{\pi(\ln C - \Delta - b_{[1]})/\sqrt{3a_{[1]}}\}) = P^*$, where $b_{[1]} = \min_i b_i$ and $a_{[1]} = \min_i a_i$.*

Proof. Under H_0, we have $\tau_i = \theta_i / (\prod_{j=1}^{k} \theta_j)^{1/k} = 1$ for each $i = 1, 2, \ldots, k$.

Therefore, the $\ln \tau_i = 0$ for each $i = 1, 2, \ldots, k$. And

$$\begin{aligned} &\Pr\left\{ \max_{1 \le i \le k} \widehat{\tau}_i \ge C \mid H_0 \right\} \\ &\quad = \Pr\left\{ \max_{1 \le i \le k} \ln \widehat{\tau}_i \ge \ln C, \text{for some } i = 1, \ldots, k \mid H_0 \right\} \\ &\quad = \Pr\{\ln \widehat{\tau}_i - \ln \tau_i \ge \ln C - \ln \tau_i, \text{ for some } i, \\ &\qquad\qquad i = 1, 2, \ldots, k \mid H_0\} \\ &\quad \le \sum_{i=1}^{k} \Pr\{\ln \widehat{\tau}_i - \ln \tau_i \ge \ln C - \ln \tau_i\} \\ &\quad = \sum_{i=1}^{k} \Pr\{\ln \widehat{\tau}_i - \ln \tau_i \ge \ln C\} \quad (\text{since the} \ln \tau_i = 0) \\ &\quad \cong \sum_{i=1}^{k} \left(1 - \frac{1}{1 + \exp\{-\pi(\ln C - b_i)/\sqrt{3a_i}\}}\right) \\ &\qquad\qquad (\text{by Lemma 3}) \\ &\quad = \sum_{i=1}^{k} \left[\frac{1}{1 + \exp\{(\pi \ln C - b_i)/\sqrt{3a_i}\}}\right] \\ &\quad \le \frac{k}{1 + \exp\{\pi(\ln C - b_{[k]})/\sqrt{3a_{[k]}}\}} \equiv \alpha \quad (\text{say}). \end{aligned} \tag{9}$$

Table 1: Times to breakdown (in minutes) at each of the five voltage levels.

Voltage level (kV)	n_1	Breakdown times
30	11	17.05, 22.66, 21.02, 175.88, 139.07, 144.12, 20.46, 43.40, 194.90, 47.30, 7.74
32	15	0.40, 82.85, 9.88, 89.29, 215.10, 2.75, 0.79, 15.93, 3.91, 0.27, 0.69, 100.58, 27.80, 13.95, 53.24
34	19	0.96, 4.15, 0.19, 0.78, 8.01, 31.75, 7.35, 6.50, 8.27, 33.91, 32.52, 3.16, 4.85, 2.78, 4.67, 1.31, 12.06, 36.71, 72.89
36	15	1.97, 0.59, 2.58, 1.69, 2.71, 25.50, 0.35, 0.99, 3.99, 3.67, 2.07, 0.96, 5.35, 2.90, 13.77
38	8	0.47, 0.73, 1.40, 0.74, 0.39, 1.13, 0.09, 2.38

Table 2: Computed values.

	1	2	3	4	5
n_i	11	15	19	15	8
a_i	0.0706	0.0561	0.0477	0.0561	0.0911
b_i	−0.0053	0.0069	0.0139	0.0069	−0.0223
$\widehat{\theta}_i$	75.981	41.174	14.35895	4.606	0.9162
$\widehat{\tau}_i$	6.6858	3.623	1.2635	0.4053	0.0806

We have

$$1+\exp\left\{\frac{\pi\left(\ln C-b_{[k]}\right)}{\sqrt{3a_{[k]}}}\right\}=\frac{k}{\alpha},$$
$$\ln C-b_{[k]}=\frac{\sqrt{3}\sqrt{a_{[k]}}}{\pi}\ln\left(\frac{k}{\alpha}-1\right). \tag{10}$$

Therefore, the critical value C is

$$C=\exp\left(\frac{\sqrt{3a_{[k]}}}{\pi}\ln\left(\frac{k}{\alpha}-1\right)+b_{[k]}\right). \tag{11}$$

However

$$\begin{aligned}
&\inf_{\underline{\theta}\in\Omega_\Delta} P\{\mathrm{CD}\mid R\}\\
&\quad=\inf_{\underline{\theta}\in\Omega_\Delta}\{\Pr(\widehat{\tau}_{(k)}\geq C)\}\\
&\quad=\inf_{\underline{\theta}\in\Omega_\Delta}\{\Pr(\ln\widehat{\tau}_{(k)}\geq \ln c)\}\\
&\quad\geq\inf_{\underline{\theta}\in\Omega_d} p(\ln\widehat{\tau}_{(k)}-\ln\tau_{[k]}\geq \ln C-\ln\Delta)\\
&\quad\cong 1-\frac{1}{1+\exp\{-\pi(\ln C-\ln\Delta-b_{(k)})/\sqrt{3a_{(k)}}\}}\\
&\quad=\frac{1}{1+\exp\{\pi(\ln C-\ln\Delta-b_{(k)})/\sqrt{3a_{(k)}}\}}\\
&\qquad\qquad\text{(by Lemma 3)}\\
&\quad\geq\frac{1}{1+\exp\{\pi(\ln C-\ln\Delta-b_{[1]})/\sqrt{3a_{[1]}}\}}\equiv P^*,
\end{aligned} \tag{12}$$

which is the desired result. □

Theorem 5. *Under the same assumption of Lemma 1 and assuming $n_1=n_2=\cdots=n_k=n$, given level α, where $0<\alpha<1$, and n, the critical value C is*

$$C=\exp\left(\frac{\sqrt{3}}{\pi}\sqrt{\frac{(k-1)}{kn}}\ln\left(\frac{k}{\alpha}-1\right)\right). \tag{13}$$

Furthermore, given P^*, where $1/k<P^*<1$ and $\Delta>0$, under the decision rule that R satisfies $\Pr\{\max_{1\leq i\leq k}\ln\widehat{\tau}_i\geq \ln C\mid H_0\}\leq\alpha$ and $\inf_{\underline{\theta}\in\Omega_\Delta}P\{\mathrm{CD}\mid R\}\geq P^*$, we have the common sample size n as follows:

$$n=\left[\frac{(k-1)}{k}\left(\frac{\ln\{(k-\alpha)P^*\}-\ln\{\alpha(1-P^*)\}}{\left(\sqrt{3}/\pi\right)\ln\Delta}\right)^2\right]+1, \tag{14}$$

where $[x]$ denotes the lowest integer greater than or equal to x.

Proof. By Theorem 4, $n_i=n$, $i=1,2,\ldots,k$, we have

$$a_i=\frac{k-1}{nk},\qquad b_i=0\quad\text{for each } i=1,2,\ldots,k, \tag{15}$$

and the critical value C is

$$\begin{aligned}
C&=\exp\left(\frac{\sqrt{3a_{(k)}}}{\pi}\ln\left(\frac{k}{\alpha}-1\right)+b_{(k)}\right)\\
&=\exp\left(\frac{\sqrt{3}}{\pi}\sqrt{\frac{k-1}{nk}}\ln\left(\frac{k}{\alpha}-1\right)+b_{(k)}\right)\\
&=\exp\left(\frac{\sqrt{3}}{\pi}\sqrt{\frac{k-1}{nk}}\ln\left(\frac{k}{\alpha}-1\right)\right),
\end{aligned} \tag{16}$$

which is the desired result.

Given P^*, where $1/k<P^*<1$ and $\Delta>0$, using the propriety of Theorem 4, we have

$$\begin{aligned}
&\inf_{\underline{\theta}\in\Omega_\Delta}P\{\mathrm{CD}\mid R\}\\
&\qquad\geq\frac{1}{1+\exp\left\{\pi(\ln C-\ln\Delta)/\sqrt{3a}\right\}}\overset{\text{Let}}{\geq}P^*,
\end{aligned} \tag{17}$$

and we have $\pi(\ln C-\ln\Delta)/\sqrt{3}a\leq\ln(1/P^*-1)$.

Table 3: Infimum of $P\{\mathrm{CD} \mid R\}$.

Δ	2	2.2	2.4	2.6	2.8	3.0	3.2
P^*	0.3201	0.5095	0.6815	0.8062	0.8850	0.9318	0.9589

Table 4: Sample size n for $k = 3, 4, 5, 6$, $\alpha = 0.01$ (0.05), and $P^* = 0.60$.

k	Δ										
	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
3	69 (38)	52 (28)	41 (23)	33 (19)	28 (16)	24 (14)	22 (12)	19 (11)	17 (10)	16 (9)	15 (9)
4	84 (48)	63 (36)	50 (28)	41 (23)	35 (20)	30 (17)	26 (15)	23 (14)	21 (12)	19 (11)	18 (11)
5	96 (56)	72 (42)	57 (33)	47 (27)	39 (23)	34 (20)	30 (18)	27 (16)	24 (14)	22 (13)	20 (12)
6	106 (62)	79 (47)	62 (37)	51 (30)	43 (26)	37 (22)	33 (20)	29 (18)	26 (16)	24 (14)	22 (13)

Note: the numbers in parentheses represent the fitted values at the level; $\alpha = 0.05$.

Table 5: Sample size n for $k = 3, 4, 5, 6$, $\alpha = 0.01$ (0.05), and $P^* = 0.80$.

k	Δ										
	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
3	92 (55)	69 (42)	54 (33)	45 (27)	38 (23)	32 (20)	29 (18)	25 (16)	23 (14)	21 (13)	19 (12)
4	112 (69)	84 (51)	66 (41)	54 (33)	46 (28)	39 (24)	34 (22)	31 (19)	28 (17)	25 (16)	23 (15)
5	126 (79)	94 (59)	74 (47)	61 (38)	51 (32)	44 (28)	39 (25)	35 (22)	31 (20)	28 (18)	26 (17)
6	138 (87)	103 (65)	81 (52)	66 (42)	56 (36)	48 (31)	42 (26)	38 (24)	34 (22)	31 (20)	28 (18)

Note: the numbers in parentheses represent the fitted values at the level; $\alpha = 0.05$.

Using (15) and $C = \exp((\sqrt{3}a/\pi)\sqrt{(k-1)/nk}\ln((k/\alpha)-1))$, we then have the minimal sample size n as follows:

$$n \geq \left[\frac{(k-1)}{k} \left(\frac{\ln\{(k-\alpha)P^*\} - \ln\{\alpha(1-P^*)\}}{\left(\sqrt{3}/\pi\right)\ln\Delta} \right)^2 \right] + 1. \quad (18)$$

□

Remark 6. The $\widehat{\theta}_i$ defined in this study fulfills Lawless Corollary 4.1.1. (Type II censored test property) [6]. When the observations are Type II censored data, we can take $\widehat{\theta}_i = 2T_i/\theta_i$, where $T = \sum_{i=1}^{r} t_{(i)} + (n-r)t_{(r)}$ and $t_{(r)}$ are the first r ordered observation of a random sample of size n from the exponential distribution. In this case, the $T = \sum_{i=1}^{r} w_i$, where $w_i = nt_{(i)}$, $i = 1, 2 \ldots, r$, remain unchanged.

4. Examples

In this section, we provide two examples to explain the results of performing Theorems 4 and 5.

Example 1. This example is from Nelson [7]. In this example, the results of a life test experiment are described in which specimens of electrical insulating fluid were subjected to a constant voltage stress. The length of time until each specimen failed, or "broke down," was observed. Table 1 gives results for five groups of specimens, tested at voltages ranging from 30 to 38 kilovolts (kV). We use the data on times to breakdown (in minutes) at each of the five voltage levels for our example.

The computed values are given in Table 2 based on the assumption that $\alpha = 0.01$.

We obtained $C = 2.8511$. Because $\widehat{\tau}_1 \geq C$ and $\widehat{\tau}_2 \geq C$, using the decision rule R, we reject $H_0 : \widehat{\tau}_1 = \widehat{\tau}_2 = \widehat{\tau}_3 = \widehat{\tau}_4 = \widehat{\tau}_5$ and select the subset containing populations 1 and 2. We identified these two populations as contributing substantially. We claim that the select subset contains the population with the largest mean.

For selected values of $\Delta, \Delta = 2(0.2)3.2, \inf_{\underline{\theta}\in\Omega_\Delta} P\{\mathrm{CD} \mid R\} = P^*$ is tabulated in Table 3. The probability of a correct decision is at least 0.3201 when Δ is 2. This probability increases to 0.9589 when Δ is 3.2.

Example 2. Based on the same assumption as Theorem 5, given the number of populations k, $k = 3, 4, 5$, and 6, as well as $\alpha = 0.05$ and 0.01 and $\Delta = 1.5(0.1)2.5$ and $P^* = 0.6, 0.8, 0.9$, and 0.95, we can determine n by using (14), so that $\inf_{\underline{\theta}\in\Omega_\Delta} P\{\mathrm{CD} \mid R\} \geq P^*$. Several selected combinations of n in each case are tabulated in Tables 4, 5, 6, and 7 which show the populations k that have the minimal sample size n required to satisfy the P^*.

5. Concluding Remarks

In this study, we considered the methods of the Lin and Huang theorems to propose a framework for analyzing and synthesizing multiple-decision procedures used for testing the homogeneity of means for k exponential distributions [5].

Table 6: Sample size n for $k = 3, 4, 5, 6$, $\alpha = 0.01$ (0.05), and $P^* = 0.90$.

k	Δ										
	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
3	114 (72)	85 (54)	67 (43)	55 (35)	46 (30)	40 (26)	35 (23)	31 (20)	28 (18)	26 (17)	24 (15)
4	137 (89)	103 (67)	81 (53)	66 (43)	56 (36)	48 (31)	42 (28)	37 (25)	34 (22)	31 (20)	28 (19)
5	154 (101)	115 (76)	91 (60)	74 (49)	63 (41)	54 (36)	47 (31)	42 (28)	38 (25)	34 (23)	31 (21)
6	168 (111)	125 (83)	99 (66)	81 (54)	68 (45)	58 (39)	51 (34)	45 (30)	41 (27)	37 (25)	34 (23)

Note: the numbers in parentheses represent the fitted values at the level; $\alpha = 0.05$.

Table 7: Sample size n for $k = 3, 4, 5, 6$, $\alpha = 0.01$ (0.05), and $P^* = 0.95$.

k	Δ										
	1.5	1.6	1.7	1.8	1.9	2.0	2.1	2.2	2.3	2.4	2.5
3	136 (90)	102 (68)	80 (53)	66 (44)	55 (37)	48 (32)	42 (28)	37 (25)	33 (23)	30 (21)	28 (19)
4	163 (110)	122 (82)	96 (65)	78 (53)	66 (45)	57 (39)	50 (34)	44 (30)	40 (27)	36 (25)	33 (23)
5	183 (124)	136 (93)	107 (73)	88 (60)	74 (51)	64 (44)	56 (38)	49 (34)	44 (31)	40 (28)	37 (26)
6	198 (136)	148 (102)	116 (80)	95 (65)	80 (55)	69 (47)	60 (42)	53 (37)	48 (33)	44 (30)	40 (28)

Note: the numbers in parentheses represent the fitted values at the level; $\alpha = 0.05$.

We provided two examples and present the main results to explain Theorems 4 and 5 which can select the subset containing the population with the largest mean and effectively determine common sample size n to satisfy the requirement of P^*. This paper presents the use of one technique to both select the optimal system among k systems and construct an optimal rule for selecting a subset of k independent random samples. We suggest employing the methods to facilitate the development of traditional statistical analyses used in the methodologies, techniques, and software applied in performing multiple-decision procedures for testing the homogeneity of means for k exponential distributions problems, such as life testing and reliability engineering.

Conflict of Interests

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

References

[1] S. S. Gupta and D. Y. Huang, "A note on optimal subset selection procedures," *The Annals of Statistics*, vol. 8, no. 5, pp. 1164–1167, 1980.

[2] X. Lin and S. S. Gupta, *Multiple decision theory: Ranking and selection problems [Ph.D. thesis]*, Purdue University, 1999.

[3] D. Y. Huang and S. Panchapakesan, "A modified subset selection formulation with special reference to one-way and two-way layout experiments," *Communications in Statistics A: Theory and Methods*, vol. 5, no. 7, pp. 621–633, 1976.

[4] D. Huang and C. Lin, "Multiple decision procedures for testing homogeneity of normal means with unequal unknown variances," in *Advances in Statistical Decision Theory and Applications*, S. Panchapakesan and N. Balakrishnan, Eds., Statistics for Industry and Technology, pp. 253–262, Birkhäuser, Boston, Mass, USA, 1997.

[5] C. Lin and D. Huang, "On some multiple decision procedures for normal variances," *Communications in Statistics: Simulation and Computation*, vol. 36, no. 1–3, pp. 265–275, 2007.

[6] J. F. Lawless, *Statistical Models and Methods for Lifetime Data*, New York, Wiley-Interscience, 2nd edition, 2003.

[7] W. B. Nelson, "Graphical analysis of accelerated life test data with the inverse power law model," *IEEE Transactions on Reliability*, vol. 21, pp. 2–11, 1972.