Batch delivery scheduling of trucks integrated with parallel machine schedule of job orders from multi-customers

Cheol Min JOO* and Byung Soo KIM**
*Division of Mechatronics Engineering, Dongseo University
47, Jurye-ro, Sasang-gu, Busan, 47011, Republic of Korea
**Department of Industrial and Management Engineering, Incheon National University
119, Academy-ro, Songdo-dong, Yeonsu-gu, Incheon, 22012, Republic of Korea
E-mail: bskim@inu.ac.kr

Received: 31 March 2016; Revised: 9 January 2017; Accepted: 2 April 2018

Abstract
This article considers an integration of two-echelon supply chain management (SCM) problem between a manufacturing site and customers. In the first echelon, jobs ordered by a number of customers are arranged and manufactured by one of a number of identical parallel machines. In the second, jobs are grouped by customer in batches and then delivered via trucks with a limited capacity. The problem is to determine batch delivery schedule of identical trucks. The batch delivery schedule is integrated with a parallel machine schedule of job orders from multi-customers. So, the objective of the problem is to simultaneously determine machine scheduling, batching and truck delivery scheduling to the corresponding customer to minimize the delivery completion time of whole the batched jobs. To solve the problem, two approaches are addressed in this article. The first approach uses a mathematical model (mixed integer programming model) to obtain the optimal solution. Since the problem is NP-hard, three kinds of genetic algorithm-based heuristics are proposed to increase solution efficiency for the second approach. The performances of the algorithms are compared using computational experiments with randomly generated examples. The computational experiments illustrate that the one of the proposed algorithms is capable of near-optimal solutions within a reasonable computing time.

Keywords: GA-based heuristics, Supply chain optimization, Scheduling, Batching, Mixed integer programming

1. Introduction

The typical supply chain (SC) includes suppliers, manufacturing sites, distribution centers, and customers. SCs are generally complex and are characterized by numerous activities defined as procurement, production, storage, distribution, and control of goods. The SC in practice spreads over multiple functions and activities creating challenges in effective SC coordination. One interesting challenge in the competitive global market environment requires maintaining lower inventories across the SC to improve responsiveness to customers. Meeting this challenge requires a closer interaction between production and distribution activities, and SC members must coordinate and work towards a unified system (Arshinder et al., 2008).

This article discusses a two-echelon supply chain management (SCM) problem that involves an integrated problem for simultaneously determining machine scheduling, batching, and truck delivery scheduling between a manufacturing site and customers. In the first echelon, jobs are arranged and manufactured by one of a number of identical parallel machines. In the second, jobs are grouped by customer and then delivered via trucks with a limited capacity. Traditionally, these problems are considered separately. The first is referred to as the machine scheduling problem and the second are batching and truck scheduling. The problems have been widely studied independently in operations research, though their integration has received little attention. One of the main reasons that few studies attempt to address them...
Simultaneously is that the problems are already extremely hard to solve independently. Recently, the coordination of SC activities has received a lot of attention in production or operations management. Rather than isolating the machine scheduling problem in production, several researchers have designed models that integrate several functions to minimize the total cost to improve customer service (Potts and Strusevich, 2009). The integrated production and distribution problems are widely seen in manufacturing industries. For example, many manufacturing companies with a make-to-order business policy schedule production and distribution on the order information and notify customers of the expected delivery time. During the manufacturing and distribution scheduling processes, any inefficient activities could result in higher operating cost, poor resource utilization (i.e., machines, human resources, and delivery trucks, etc.), and late delivery, thus potentially decreasing future demand due to poorer customer service.

In this study, jobs ordered by a number of customers are first manufactured by a set of identical parallel machines in a manufacturing plant. Then, these jobs ordered by the same customer are grouped and loaded into one of multiple available trucks with limited capacity, and delivered to the associated customer, with the truck returning directly to the production plant for the next delivery after the current delivery. The objective is to simultaneously determine machine scheduling, batching and truck delivery scheduling to the corresponding customer to minimize the delivery completion time of the batched jobs. Figure 1 illustrates an example of the integrated parallel machine scheduling, batching, and truck delivery scheduling with a limited capacity and the corresponding machine and truck delivery schedules in Gantt chart.

Fig. 1. Illustration of machine and truck delivery scheduling problem with limited capacity and the corresponding machine and truck delivery schedules in Gantt chart
processed by machine 2, respectively in the manufacturing plant. For the scheduling of trucks, three batches with job 1, jobs 4 and 5, and job 6 are grouped and sequentially delivered to customers 2, 3, and 3 by truck 1 and three batches with job 2, jobs 3, 8, and 9, and job 7 are grouped and sequentially delivered to customers 1, 2, and 1 by truck 2.

The rest of this article is organized as follows. The next section is devoted to a review of the related literature. Section 3 presents a new mixed integer programming formulation of the problem, and three kinds of genetic algorithm-based (GA-based) heuristics proposed to find near-optimal solutions in Section 4. In Section 5, the performances of the proposed algorithms are evaluated through computational experiments. Finally, Section 6 concludes the paper and suggests future research.

2. Literature review

Research on an integrated scheduling problem between the two-echelon SCs including production and distribution is growing very rapidly, even though the many researchers are interested in the problem recently. In many production applications including production facilities of make-to-order or time sensitive products such as perishable or seasonal products, the finished orders is necessary to immediately or shortly deliver to after production. Therefore, the need for integration between activities is avoidable for improving the customer service.

To our best knowledge, Chen (2010) is the only comprehensive review on integrated production and outbound distribution scheduling problems within a single time period. There have been considerable studies conducted in production-transportation integration with emphasis on the road transportation and vehicle routing problem (Chang and Lee, 2004; Chang et al., 2014; Chen et al., 2009; Hunter and Van Buer, 1996; Lee and Chen, 2001; Li et al., 2005a; Tang and Liu, 2009; Van Buer et al., 1999; Xuan, 2011; Zhong et al., 2007). Hunter and Van Buer (1996) are among the pioneers who studied integrated production and distribution problem (IPDP) on serving multiple products to multiple customers. Their models incorporated vehicle routing options and were studied using real data from a newspaper company. Van Buer et al. (1999) extended their models to consider the reuse of trucks for the next deliveries (truck recycling) and concluded its cost efficiency by numerical studies. Lee and Chen (2001) studied machine scheduling problems that considered either intermediate delivery between machines or the finished product deliveries. Several related studies and extensions can be found in Chang and Lee (2004). Li et al. (2005a), and Zhong et al. (2007) considered an IPDP with pickup and delivery arrangements of materials and finished jobs.

There are a few related studies and extensions on integrated production and distribution scheduling under different production types and transportation modes. Li et al. (2005a) developed a single-machine scheduling model that incorporates routing decisions of a delivery vehicle which serves customers at different locations. The objective of the paper is to minimize the sum of job arrival times. They develop a polynomial time algorithm for the case when the number of customers is fixed. Mazdeh et al. (2007) are interested in the single machine scheduling problem for delivery in batches to customers. The objective on the problem is to minimize the sum of flow times by considering the possibility of delivering jobs in batches and introducing batch delivery costs. For solving the problem, structural properties of the problem are investigated and used to devise a branch-and-bound solution scheme. Ji et al. (2007) considered an integrating scheduling problem in which independent and simultaneously available jobs are to be processed on a single machine and the jobs are delivered in batches and the delivery of a batch equals the completion time of the last job in the batch. The objective on the problem is to minimize the sum of the total weighted flow time and delivery cost. They provide optimal algorithms for two special cases of the problem. Xuan (2011) studied a class of hybrid flowshop scheduling problem characterized by release time, transportation time and transportation capacity of one unit for each transporter. This problem is formulated as an integer programming model and a Lagrangian relaxation algorithm is proposed to solve the problem. Li et al. (2008) addressed a coordinated scheduling problem of parallel machine assembly manufacturing and multi-destination transportation in the make-to-order consumer electronics supply chain. Chang et al. (2014) considered an integrated production and distribution scheduling problem in which jobs are first processed by one of the unrelated parallel machines and then distributed to corresponding customers by capacitated vehicles without intermediate inventory. The objective is to find a joint production and distribution schedule so that the weighted sum of total weighted job delivery time and the total distribution cost is minimized. They derive a mathematical model for describing the problem and propose an algorithm using ant colony optimization. Cakici et al. (2014) investigated the integrated production and distribution scheduling problem in a supply chain. The manufacturer’s production environment
is modeled as a parallel machine system. A single capacitated vehicle is employed to deliver products in batches to multiple customers. The scheduling problem can also be viewed as either parallel machines with delivery considerations or a flexible flowshop. They presented a mathematical model and propose near-optimal heuristics to minimize total weighted completion time. A few studies on production scheduling considering air transportation with fixed departure dates are reported. Li et al. (2004) studied the synchronization of single machine scheduling and air transportation with single destination. The overall problem is decomposed into air transportation problem and single machine scheduling problem. They formulated two problems and then presented a backward heuristic algorithm for single machine scheduling. Li et al. (2005a) extended their previous work to consider multiple destinations in air transportation problem. Li et al. (2006) presented the air transportation allocation has the structure of regular transportation problem, while the single machine scheduling problem is NP-hard. Li et al. (2005b) also proposed a forward heuristic and a backward heuristic for single machine. Zandieh and Molla-Alizadeh-Zavardehi (2008) developed some mathematical models for two problems with different delivery assumptions (with delivery tardiness and without delivery tardiness) regarding due window. Zandieh and Molla-Alizadeh-Zavardehi (2009) extended their work considering various capacities with different transportation cost and also charter flights (commercial flights). Delavar et al. (2010) investigated the problem of determining both production schedule and air transportation allocation of orders to optimize customer service at minimum total cost. In order to solve the given problem, two genetic algorithm (GA) approaches are developed. Recently, Hajiaghaei-Keshteli et al. (2014) proposed an integrated production and transportation model, which considers rail transportation, which is firstly developed to deliver the orders from a facility to the customers (warehouses). The problem is to determine both production schedule and rail transportation allocation of orders to optimize customer service at minimum total cost. Hall and Potts (2003) considered a three echelon supply chain by setting up classical scheduling objectives for the supplier, the manufacturer, and their cooperation. They also identified incentives and mechanisms for the cooperation and demonstrated implications for improving supply chain operations efficiency. Hall and Potts (2005) extended their work to minimize four different scheduling costs, based on the completion time, maximum lateness, total tardiness, and number of late jobs, in both single and parallel machine environments. Chen and Pandoor (2006) considered the supply chain consisting of multiple overseas plants and a domestic distribution center (DC). A manufacture of the plants produces time-sensitive products that have a large variety, a short life cycle, and are sold in a very short selling season. In the study, given a set of orders, they determine which orders are to be assigned to each plant, find a schedule for processing the assigned orders at each plant, and find a schedule for shipping the completed orders from each plant to the DC, such that a certain performance measure is optimized. Chen and Lee (2008) investigated a general two echelon scheduling problem, in which jobs of different importance are processed by one first-stage processor and then, in the second stage, the completed jobs need to be batch delivered to various pre-specified destinations in one of a number of available transportation modes. Our objective is to minimize the sum of weighted job delivery time and total transportation cost. Pandoor and Chen (2009) studied an integrated production and distribution scheduling model in a two-stage supply chain consisting of one or more suppliers, a warehouse, and a customer. The problem is to find jointly a cyclic production schedule for each supplier, a cyclic delivery schedule from each supplier to the warehouse, and a cyclic delivery schedule from the warehouse to the customer so that the customer demand for each product is satisfied without backlog to minimize total production, inventory and distribution cost. Zegordi et al. (2010) considered the scheduling of products and vehicles in a two-stage supply chain environment. The first stage contains multiple suppliers with different production speeds, while the second stage is composed of multiple vehicles with different speed and different transportation capacity and those delivers the batches of the products collecting from the suppliers to a single manufacturing facility. They propose a gendered genetic algorithm (GGA) to solve the problem. Cakici et al. (2013) addressed the problem of loading and scheduling of batching machines in an environment with job release times and incompatible job families. Their study gives a motivation that batch scheduling models are applicable to the distribution stage of supply-chain scheduling problems. In this environment, transportation units can be viewed as batching machines in which a group of jobs (orders) are processed (transported) together. Jobs become available for pickup at different times and are delivered to customers by capacitated vehicles. Gao et al. (2015) addressed a variation of the integrated batch production and distribution problem in which batch of orders are manufactured and delivered in a set of batches by single vehicle with limited capacity. They emphasize the no-wait condition between the production and distribution of each batch, and prove that the general version of this integrated operational scheduling problem is strongly NP-hard. They also investigate to explore the optimal solution structures of two special cases, which are identical order processing time and identical delivery time, respectively. Then, they propose polynomial algorithms for the cases using the optimality
structures.
From the literature, it can be concluded that the topic of this research is an important in both academia and industry. To the best of our knowledge, there is no research focusing on batch delivery scheduling of trucks integrated with parallel machine schedule of job orders from multi-customers to minimize the delivery completion times of whole the batched jobs.

3. Mathematical model

In this section, we derive a mathematical model for batch delivery scheduling of trucks integrated with parallel machine schedule of job orders from multi-customers to minimize the delivery completion times of whole the batched jobs. The parameters and decision variables in the mathematical model are defined as follows:

<Parameters>

\( J \) : the set of jobs
\( M \) : the set of machines
\( B \) : the set of maximum available batches (\( \equiv J \))
\( T \) : the set of trucks
\( C \) : the set of customers
\( p_j \) : processing time of job \( j \in J \) at the manufacturing plant
\( h_n \) : transportation time (includes return time) to customer \( n \in C \)
\( R_{jn} \) : 1, if job \( j \in J \) is ordered by customer \( n \in C \); 0 otherwise
\( v_j \) : volume of job \( j \in J \)
\( V \) : truck capacity
\( Q \) : Big number

<Continuous decision variables>

\( x_i \) : starting time of job \( i \) at the manufacturing plant.
\( r_k \) : shipping time for batch \( k \)
\( C_{\text{max}} \) : delivery completion time of whole the batched jobs

<Binary decision variables>

\( y_{im}^M \) : 1 if job \( i \) is assigned to machine \( m \) at the manufacturing plant; 0 otherwise.
\( y_{ik}^B \) : 1 if job \( i \) is assigned to batch \( k \); 0 otherwise.
\( y_{kt}^T \) : 1 if batch \( k \) is assigned to truck \( t \); 0 otherwise.
\( y_{kn}^C \) : 1 if batch \( k \) is assigned to customer \( n \); 0 otherwise.
\( z_{im}^M \) : 1 if job \( i \) precedes job \( j \ (j \neq i) \) at machine \( m \) at the manufacturing plant; 0 otherwise.
\( p_{im}^M \) : 1 if job \( i \) is the first job at machine \( m \) at the manufacturing plant; 0 otherwise.
\( z_{kl}^T \) : 1 if batch \( k \) precedes batch \( l \ (l \neq k) \) in truck \( t \); 0 otherwise.
\( e_{kt} \) : 1 if batch \( k \) is the first batch in truck \( t \); 0 otherwise.

The initial set \( B \) is defined as the set of maximum available batches, because the batching is a decision target in the model. Some of batches in the set \( B \) have no assigned jobs in a solution, and the batches are ignored in the truck delivery scheduling (binary decision variables of the unassigned batches \( y_{kn}^C = 0 \) for all \( n \in C \)).
The Mixed integer programming (MIP) model for the integrated scheduling problem is as follows:

Minimize

\[
   z = C_{\text{max}}
\]

Subject to

\[
   x_i + p_i \leq x_j + Q \cdot \left(1 - \sum_{m \in M} z_{ijm} \right), \quad \text{for } \forall i, j \in J; j \neq i \tag{2}
\]

\[
   \sum_{m \in M} y_{im} = 1, \quad \text{for } \forall i \in J \tag{3}
\]

\[
   \sum_{i} F_{im} \leq 1, \quad \text{for } \forall m \in M \tag{4}
\]

\[
   F_{im} + \sum_{j \neq i} z_{ijm} = y_{im}, \quad \text{for } \forall i \in J; \forall m \in M \tag{5}
\]

\[
   \sum_{j \neq i} z_{ijm} \leq y_{im}, \quad \text{for } \forall i \in J; \forall m \in M \tag{6}
\]

\[
   \sum_{k \in B} y_{ik} = 1, \quad \text{for } \forall i \in J \tag{7}
\]

\[
   \sum_{i} v_i y_{ik} \leq V, \quad \text{for } \forall k \in B \tag{8}
\]

\[
   y_{ik}^R + y_{ik}^B \leq 1 + \sum_{m \in C} R_{in} \cdot R_{jm} \quad \text{for } \forall k \in B; \forall i, j \in J \text{ and } i < j \tag{9}
\]

\[
   r_k \geq (x_i + p_i) - Q \cdot (1 - y_{ik}^R) \quad \text{for } \forall i \in J; \forall k \in B \tag{10}
\]

\[
   \sum_{n \in C} y_{kn} \leq 1 \quad \text{for } \forall k \in B \tag{11}
\]

\[
   y_{kn}^C \geq R_{in} \cdot y_{ik}^B \quad \text{for } \forall i \in J; \forall k \in B; \forall n \in C \tag{12}
\]

\[
   r_k + \sum_{n \in C} h_n \cdot y_{kn} \leq r_l + Q \cdot \left(1 - \sum_{t \in T} z_{lkt} \right), \quad \text{for } \forall k, l \in B; k \neq l \tag{13}
\]

\[
   \sum_{t \in T} y_{kt}^T = 1, \quad \text{for } \forall k \in B \tag{14}
\]

\[
   \sum_{k \in B} F_{kt} \leq 1, \quad \text{for } \forall t \in T \tag{15}
\]

\[
   F_{kt} + \sum_{l \neq k} z_{klkt} = y_{kt}^T, \quad \text{for } \forall k \in B; \forall t \in T \tag{16}
\]

\[
   \sum_{t \in T} y_{kt}^T \leq y_{kt}^T, \quad \text{for } \forall k \in B; \forall t \in T \tag{17}
\]

\[
   r_k + \sum_{n \in C} h_n \cdot y_{kn}^C \leq C_{\text{max}} \quad \text{for } \forall k \in B \tag{18}
\]

\[
   x_i, r_k, C_{\text{max}} \geq 0, \quad \text{for } \forall i \in J; \forall k \in B \tag{19}
\]

\[
   y_{im}^M y_{in}^M y_{jt}^T y_{kn}^C = 0 \text{ or } 1, \quad \text{for } \forall i, j \in J; \forall k \in B; \forall t \in T; \forall n \in C \tag{20}
\]

\[
   z_{ijm}^M z_{klkt}^T y_{it}^T F_{kt}^T F_{kt}^T = 0 \text{ or } 1, \quad \text{for } \forall i, j \in J; \forall m \in M; \forall k, l \in B; \forall t \in T \tag{21}
\]

Constraint (2) ensures the precedence relationship of jobs assigned to the same machine at the manufacturing plant.
and calculates the starting time for each job. Constraint (3) confirms that each job is processed at exactly one of the machines. Constraints (4)-(6) ensure that jobs assigned to the same machine must appear once in their sequence. Constraint (4) guarantees that at most one job is positioned at the beginning of the sequence before all jobs on each machine. Constraint (5) ensures that if a job is assigned to a machine, then it will be immediately succeeded by one job. Similarly, Constraint (6) indicates that if a job is assigned to a machine, it can be succeeded by at most one job. The job at the last position of the sequence on a machine will not have a succeeding job. Constraint (7) forces each job into exactly one batch. Constraint (8) confirms that the volume of each batch must be less than or equal to the capacity of one truck. Constraint (9) ensures that all jobs in the same batch belong to the same customer. Constraint (10) calculates the delivery ready time of each batch, which is the longest job manufacturing completion time in the batch. Constraints (11)–(12) confirm the each batch delivery to the customer by the relationship between the ordered customer and the jobs in the batch. Constraint (13) calculates the precedence relationship of batches assigned to the same truck and calculates the shipping time of each batch. Constraint (14) confirms that each batch is delivered by exactly one truck. Constraints (15)-(17) ensure that batches assigned to the same truck must appear once in their sequence. Constraint (15) guarantees that at most one batch is positioned at the beginning of the sequence before all batches on each truck. Constraint (16) stipulates that if a batch is assigned to a truck, it will be immediately succeeded by one batch. Similarly, Constraint (17) ensures that if a batch is assigned to a truck, it is succeeded by at most one batch. Constraint (18) calculates the delivery completion time of whole the batched jobs.

4. Genetic algorithm-based heuristics

The proposed problem reduces the parallel machine problem $P_m||C_{\text{max}}$, assuming transportation time for each truck to be $h_n = 0$, where $n \in C$ in the scheduling notation scheme of Graham et al. (1979), which is NP hard. Thus, we focus on proposing effective and efficient heuristic algorithms for the integrated problem to determine machine scheduling, batching, and truck delivery scheduling, simultaneously. GA is one of the most powerful and broadly applicable meta-heuristics based on principles from the theory of evolution Gen and Cheng (2000). Several studies proposed GA to solve the scheduling problem in various supply chain structures and obtained near-optimal solutions in a reasonable amount of computation time (Delavar et al., 2010; Hajiaghaei-Keshteli, 2011; Hajiaghaei-Keshteli et al., 2014, 2010; Zegordi et al., 2010). Thus, we also propose GA-based heuristics to simultaneously solve machine scheduling, batching, and truck delivery scheduling.

In GA based heuristics, the representation of a solution (chromosome) impacts on the solution’s performance. Basically, we use single string arrays as the chromosome for machine scheduling, batching, and truck delivery scheduling, respectively. A dispatching rule is required to represent job assignments to resources and the sequence of jobs at each resource with a single string array. Based on previous studies, a single string array with a dispatching rule performs well for sequencing and resource assignment (Joo and Kim, 2012; Lee et al., 2012).

Figure 2 introduces examples of dispatching rules for machine schedule, batching, and truck schedule from each single string array. There are 9 jobs ordered by 3 customers, 2 parallel machines, and 2 delivery trucks. Customer 1 orders jobs 2 and 7; customer 2 orders jobs 1, 3, 8, and 9; and customer 3 orders jobs 4, 5, and 6. The processing times and volumes of each job are 3, 4, 4, 6, 5, 4, 6, 2, and 15, 20, 5, 10, 5, 10, 5, respectively. The truck capacity is fixed at 20. The transportation times to each customer are 8, 6, and 7, for customers 1, 2, and 3, respectively. From the string array in Figure 2(a), the corresponding machine schedule is derived from the machine assigning rule, which is based on the job completion time. To assign a job to one of the machines, the completion of machines for the assigned job to the end of the sequence in the corresponding machine are calculated, and then the job is assigned to the machine with the shortest completion time. The general pseudo code of the machine assigning rule is as follows:

**Procedure**: machine assigning rule  
**Inputs**: a string array with jobs $J_1, J_2, \ldots, J_n$, and processing time of job $p_j$  
**Output**: a corresponding machine schedule

**Begin**

Let current completion time of machine $k, c_k^0 = 0$.  
Let job index $\theta = 1$.  


While ($\theta \leq n$)
  Let machine index $k = 1$.
  While ($k \leq$ number of machines)
    Calculates temporary completion time $C_k^{\theta-1} + p_{(j_\theta)}$.
    \[ k = k + 1. \]
  End While
  Select the machine $\delta$ with the smallest temporary completion time.
  \[ C_{\delta}^{\theta} = C_{\delta}^{\theta-1} + p_{(j_\theta)} \]
  \[ \theta = \theta + 1. \]
End While

From the string array in Figure 2(b), the corresponding delivery job batches are derived by the batching rule, which is based on the customer and job volume. To assign a job to a batch, the customer and the current available capacity of the existing batches are checked, and then the job is assigned to the first available batch with the volume being less than the current available capacity of the batch and the appropriate customer. If there is no batch satisfied by these conditions, the job is assigned to a new batch. The general pseudo code of the batching rule is as follows:

Procedure : batching rule
Inputs : a string array with jobs ($j_1, j_2, \ldots, j_n$), ordering customer of job $O_j$, volume of job $v_j$, and truck capacity $V$
Output : a corresponding job batches

Begin
  Let current number of batches $N_B = 1$.
  Let the set of jobs in the first batch $SB_1 = \{ j_1 \}$
  Let the delivery destination(customer) of the first batch $CB_1 = O_{(j_1)}$.
  Let current volume of the first batch $VB_1 = v_{(j_1)}$.
  Let job index $\theta = 2$.
  While ($\theta \leq n$)
    Let batch index $k = 1$.
    While ($k \leq N_B$)
      If ($O_{(j_\theta)} = CB_k$ and ($VB_k + v_{(j_\theta)} \leq V$))
        \[ SB_k = SB_k + \{ j_\theta \}. \]
        Break While;
      End If
    End While
    If (There is no available batch to assign $j_\theta$)
      \[ N_B = N_B + 1. \]
      \[ SB_{N_B} = \{ j_\theta \}. \]
      \[ CB_{N_B} = O_{(j_\theta)}. \]
      \[ VB_{N_B} = v_{(j_\theta)}. \]
    End If
    \[ \theta = \theta + 1. \]
  End While
End

From the string array in Figure 2(c), the corresponding truck schedule is derived by the truck assigning rule, which is based on the batch delivery completion time. To assign a batch to a truck, the assigned trucks’ delivery completion time to the end of the sequence in the corresponding truck are calculated, and then the batch is assigned to the truck with the shortest delivery completion time. The general pseudo code of the truck assigning rule is as follows:
**Procedure**: truck assigning rule

**Inputs**: a string array with batches \((b_1, b_2, \cdots, b_m)\), delivery ready time of batch \(RB_i\), and transportation time of batch \(TB_i\)

**Output**: a corresponding truck schedule

**Begin**

Let current completion time of truck \(k\), \(C_k^0 = 0\).

Let batch index \(\theta = 1\).

**While** \((\theta \leq m)\)

Let truck index \(k = 1\).

**While** \((k \leq \text{number of trucks})\)

Calculates temporary delivery completion time \(\max(C_k^{\theta-1}, RB_{(b_{\theta})}) + TB_{(b_{\theta})}\).

\(k = k + 1\).

**End While**

Select the truck \(\delta\) with the smallest temporary delivery completion time.

\(C_\delta^\theta = \max(C_\delta^{\theta-1}, RB_{(b_{\theta})}) + TB_{(b_{\theta})}\)

\(\theta = \theta + 1\).

**End While**

**End**

---

**Fig. 2. Examples of rule-based corresponding schedules from a string array**

Based on the chromosomes consist of single string array with the dispatching rule, we design three kinds of GA-based heuristics for machine scheduling, batching, and truck delivery scheduling simultaneously, called the three-stage GA (GA_ST) in which the three GAs independently process each stage. The single-stage GA with independent dispatching rules uses chromosomes with triple string arrays (GA_TS) and the single-stage GA with integrated dispatching rules uses chromosomes with a single string array (GA_SS).
GA_ST is a three-stage algorithm. For the first stage, the GA for the machine schedule processes a chromosome with a single string array. The single string array represents a job order to apply a machine assignment rule that determines the job assignments to machines and the sequences of assigned jobs in each machine. For the second stage, GA for batching using chromosome processes with a single string array. The completion times of jobs for the best machine schedule from the first stage become the input data. The single string array represents a job order to apply a delivery batching rule that groups the jobs to be delivered to customers. For the third stage, GA for truck scheduling processes a chromosome with a single string array. The ready and transportation times for batches of the best batching from the second stage become the input data. The single string array represents a batch order for applying a truck assignment rule that determines the batch assignments to trucks and the sequences of assigned batches for each truck. Figure 3 describes the entire GA_ST structure.

GA_TS is a single-stage algorithm using a chromosome with three string arrays representing machine scheduling, job batching, and truck scheduling, respectively. The first string array represents a job order to apply a machine assignment rule that determines the jobs assigned to machines and the sequences of assigned jobs for each machine. The job completion times of jobs from the machine scheduling become the input data for the batching. The second string array represents a job order to apply a delivery batching rule that groups the jobs for delivery to customers via truck. The ready and transportation times of batches from the batching become the input data for the truck scheduling. The last string array represents a batch order to apply a truck assignment rule that determines the batches assigned to trucks and the sequences of assigned batches in each truck. Figure 4 illustrates how the corresponding machine scheduling, delivery job batching, and truck scheduling are constructed and the calculation of fitness from the chromosome with the three GA_TS string arrays.

GA_SS is a single-stage algorithm using a chromosome with a single string array for machine scheduling that represents a job order to apply a machine assignment rule that determines the job assignments to machines and the
sequences of assigned jobs for each. From the machine schedule, the corresponding string array is described according to the order of the job completion times. The corresponding string array from the machine schedule represents a job order to apply a delivery batching rule that groups the jobs for truck delivery to the customers. From the delivery job batches, the corresponding string array is described according to the order of batch ready times. The corresponding string array from the delivery job batches represents a batch order to apply a truck assignment rule that determines the batch assignments to trucks and the sequences of assigned batches for each truck. The completion times of jobs from the machine scheduling become the input data for the batching, and the ready and transportation times of batches from the batching become the input data for the truck scheduling. Figure 5 shows the construction of the corresponding machine scheduling, delivery job batching, and truck scheduling and the calculation of fitness from the chromosome with a single string array in GA_SS.

In GA-based heuristics, a set of chromosomes forms a population. The initial population is randomly generated for the first generation. The chromosomes in the population are evaluated using a fitness value by the delivery completion time of whole the batched jobs to the corresponding customers. This value is the objective function value proposed in Section 3. The chromosomes that have a higher fitness value (lower objective function value) than the average fitness of the current population make up the potential parent pool. In this study, the population of the next generation is primarily composed of the best chromosomes migrating from the current generation and new chromosomes (i.e., children) reproduced by genetic operators (with randomly selected parents from the potential parent pool). The next generation is evaluated, and this process is repeated until a stopping criterion (maximum number of generations) is met. GA has two evolutionary operators—crossover and mutation. These operators enhance the performance of solutions by propagating similarities and unexpected genetic characteristics to offspring. Crossover combines two parents to reproduce a child preserving their characteristics. The order crossover is utilized for GA_ST, GA_TS, and GA_SS. A crossing point is randomly selected for the crossover from the genes of one parent. The sub-section of genes, before the crossing point, is then inherited from the parent to the child. The sub-section of genes in the other parent, after the crossing point, must consider the order of each legitimate gene when filling the remaining places of the child so as to avoid violating the feasibility. Mutation maintains the diversity of a population in the successive generations but also exploits the solution space. The swap mutation is utilized for GA_ST, GA_TS, and GA_SS. For the swap mutation, two genes are randomly selected from the parent and exchange each gene to avoid an infeasible solution. In GA_TS, the crossover and mutation are independently and randomly applied to any one, two, or three sub-chromosomes for the manufacturing sequence, batching, and distribution sequence. The general pseudo code of the basis GA for GA_ST, GA_TS, and GA_SS is as follows:

**Procedure**: the basis GA

**Inputs**: crossover rate $P_c$, mutation rate $P_m$, population size $S_{pop}$, and maximum generation $G_{max}$.

**Output**: the near optimal solution

**Begin**
Let generation index $\theta = 1$.
Randomly generate an initial population for the first generation.

While ($\theta \leq G_{max}$)
  Let population index $\delta = 1$.
  While ($\delta \leq S_{pop}$)
    Randomly select two chromosomes from current population.
    If (random number $\leq P_C$)
      Do order crossover operations.
    End If
    If (random number $\leq P_M$)
      Do swap mutation operations.
    End If
    Calculate the fitness value of each schedule.
    $\delta = \delta + 1$.
  End While
  Construct the next generation by a roulette-wheel from the children.
  $\theta = \theta + 1$.
End While

5. Computational results

To evaluate the performances of the GA-based heuristics, computational experiments are conducted using randomly generated test problems. Since the complexity of a problem highly depends on the number of jobs ($J$), machines ($M$), trucks ($T$), and customers ($C$), several instances of two problem groups of small and large problems are randomly generated according to the four complexity factors. The instances are given information such as job processing times at the manufacturing plant, transportation time to customers, volume of jobs, and batch capacity. The processing time at the manufacturing plant is uniformly generated from [60, 120], the transportation time to customer is uniformly generated from [60, 240], the volume of jobs is uniformly generated from [5, 10], and the batch capacity is fixed at 20. For the GA parameter values used in the experiments, GA_TS, GA_SS, and each stage of GA_ST are running with a population size of $J$ and a generation size of 1000, and fixed crossover and mutation rates of 0.8 and 0.2, respectively, which are predetermined by extensive preliminary experiments. GA_TS, GA_SS, and each stage of GA_ST are running with $J \times 1000$ iterations for equal comparison with the GA-based heuristics. All experiments solving each test problem use ILOG CPLEX and the GA-based heuristics were executed on a PC with a 3.50 GHz processor and 4 GB RAM. The relative percentage deviation ($RPD$), expressed using equation (22), mean absolute deviation ($MAD$), and the computing time of 10 replications by each of the genetic algorithms are calculated.

$$ RPD(\%) = \frac{GA_{sol} - Best}{Best} \times 100, $$

(22)

Where $GA_{sol}$ is the solution obtained by GA_ST, GA_TS, or GA_SS and $Best$ is the best solution of all experiments for each test problem. $Best$ can be the optimal solution if CPLEX obtains the optimal solution.

To demonstrate the solvability of the mathematical model, we evaluate the best solution of GA-based heuristics with the optimal solution with CPLEX using a total of 16 instances for small sized problems. The instances are constructed by randomly generating 4 instances in each of 5 to 8 jobs. In each instance, the number of machines, trucks, and customers are randomly chosen by a uniform distribution from [2, 3], [2, 4], and [2, 4], respectively. IBM ILOG CPLEX 12.5 is used to find optimal solutions with the mathematical programming presented in Section 3. Since the computing time for CPLEX significantly increases as the number of jobs increases (See Figure 6), we imposed a 7200(sec.) time limit and a particular run is simply terminated if the optimal solution had not been found and verified in that amount of time.
Fig. 6. Computing time comparison: CPLEX and GA_SS

Table 1 presents the results obtained from the developed CPLEX and GA-based heuristics for small problems. The first column indicates the instance number and columns 1-5 show the experimental condition by indicating the number of jobs (J), machines (M), trucks (T), and customers (C). The sixth and seventh columns show the optimal values obtained and the computing times. Columns 8-10, 11-13, and 14-16 report the average MADs, average RPDs, and average computing times for GA_ST, GA_TS, and GA_SS, respectively. The low values of RPDs and MADs in Table 1 indicate that all proposed algorithms perform well for small problems. Especially, the average RPD and MAD for GA_TS and GA_SS are smaller than for GA_ST, and the values are extremely small. The computing times of all three GA-based heuristics are still less than 1 second. These results mean that GA_TS and GA_SS are very effective and efficient algorithms with low variation for small problems.

| No. | J  | M  | T  | C  | Opt. value | CPLEX Time (Sec.) | MAD (%) | RPD (%) | CPLEX Time (Sec.) | GA_ST Time (Sec.) | GA_TS Time (Sec.) | GA_SS Time (Sec.) | MAD (%) | RPD (%) | Time (Sec.) |
|-----|----|----|----|----|------------|-------------------|---------|---------|-------------------|-------------------|-------------------|-------------------|---------|---------|----------|
| 1   | 5  | 2  | 2  | 2  | 404        | 28                | 6.16    | 8.54    | 0.09              | 0.00              | 0.00              | 0.05              | 0.00    | 0.00    | 0.04     |
| 2   | 5  | 2  | 3  | 2  | 313        | 2                 | 0.00    | 0.00    | 0.00              | 0.00              | 0.00              | 0.00              | 0.00    | 0.00    | 0.04     |
| 3   | 5  | 3  | 2  | 3  | 299        | 1                 | 0.00    | 0.00    | 0.00              | 0.00              | 0.00              | 0.00              | 0.00    | 0.00    | 0.04     |
| 4   | 5  | 3  | 3  | 2  | 287        | 3                 | 0.00    | 0.00    | 0.00              | 0.00              | 0.00              | 0.00              | 0.00    | 0.00    | 0.04     |
| 5   | 6  | 2  | 2  | 3  | 391        | 547               | 3.72    | 7.29    | 0.11              | 1.06              | 1.07              | 0.06              | 1.00    | 0.84    | 0.05     |
| 6   | 6  | 2  | 3  | 2  | 429        | 42                | 0.96    | 0.61    | 0.11              | 0.00              | 0.00              | 0.06              | 0.00    | 0.00    | 0.05     |
| 7   | 6  | 3  | 2  | 4  | 410        | 18                | 3.03    | 4.46    | 0.13              | 0.62              | 0.39              | 0.07              | 0.00    | 3.17    | 0.06     |
| 8   | 6  | 3  | 3  | 2  | 320        | 4                 | 2.03    | 3.63    | 0.11              | 0.00              | 0.00              | 0.06              | 0.00    | 0.00    | 0.05     |
| 9   | 7  | 2  | 3  | 2  | 408        | 247               | 2.15    | 5.69    | 0.15              | 0.00              | 0.00              | 0.08              | 0.00    | 0.00    | 0.07     |
| 10  | 7  | 2  | 4  | 3  | 512        | 88                | 2.84    | 9.10    | 0.15              | 0.14              | 0.66              | 0.09              | 0.14    | 1.29    | 0.07     |
| 11  | 7  | 3  | 2  | 3  | 440        | 84                | 2.30    | 5.77    | 0.15              | 0.33              | 0.18              | 0.09              | 0.00    | 4.55    | 0.07     |
| 12  | 7  | 3  | 3  | 2  | 378        | 273               | 4.42    | 23.89   | 0.15              | 0.00              | 0.00              | 0.09              | 0.00    | 0.00    | 0.07     |
| 13  | 8  | 2  | 2  | 4  | 450 < 7200 | 2.07              | 9.56    | 0.18    | 0.28              | 0.20              | 0.10              | 0.28              | 0.20    | 0.09    | 0.09     |
| 14  | 8  | 2  | 3  | 2  | 504 < 7200 | 2.74              | 18.57   | 0.18    | 0.04              | 0.02              | 0.10              | 0.00              | 0.00    | 0.00    | 0.09     |
| 15  | 8  | 3  | 3  | 2  | 320 < 7200 | 1.10              | 22.75   | 0.19    | 2.24              | 1.50              | 0.11              | 0.00              | 0.00    | 0.10    | 0.09     |
| 16  | 8  | 3  | 2  | 3  | 447 < 7200 | 4.04              | 16.49   | 0.19    | 0.84              | 0.60              | 0.11              | 0.98              | 2.64    | 0.09    | 0.06     |

Avg. 2.35 9.42 0.13 0.39 0.32 0.08 0.18 0.81 0.06

To evaluate the performance of GA-based heuristics and gain the insight into the algorithms by altering the complexity parameters, we relatively compare the best solution of GA_ST, GA_TS and GA_SS using a total of 32 instances for large problems. The instances are constructed by randomly generating 8 instances in each 60, 80, 100, and 120 jobs, respectively. In each instance, the number of machines, trucks, and customers are randomly chosen by a uniform distribution from [10, 20], [5, 10], and [10, 20], respectively.
Table 2 shows the results for large problems. Similar to the columns in Table 1, the average RPDs, MADs, and computing times for GA_ST, GA_TS, and GA_SS are presented for combinations of the number of jobs (J), machines (M), trucks (T), and customers (C). The RPDs and MADs of GA_TS and GA_SS are much better than those of GA_ST for most large problems and consume less computing time. The results indicate that the proposed single-stage GA-based heuristics of a simple chromosome with a dispatching rule significantly improves the GA's performance compared to the GA_ST chromosome. Furthermore, the RPDs and MADs for GA_SS are better than GA_TS with slightly less computing time. This is because the dispatching rule used in GA_SS enhances the exploitation of a good solution by constructing batching and truck scheduling depending on good machine scheduling and batching determination during the single-stage GA-based heuristic process. In Figure 7, the computation time for GA_SS shows a polynomial growth behavior by the increase of the job size in large problems in Table 2.

| No. | J   | M | T | C | Best | GA_ST (Sec) | GA_TS (Sec) | GA_SS (Sec) |
|-----|-----|---|---|---|------|-------------|-------------|-------------|
|     |     |   |   |   |      | MAD (%)   | RPD (%)    | Time    | MAD (%)   | RPD (%)    | Time    |
| 1   | 60  | 19| 5 | 16| 1106 | 1.57      | 2.31       | 12.35    | 0.35      | 0.56       | 9.15    |
| 2   | 14  | 5 | 18|   | 992  | 1.58      | 5.87       | 11.53    | 0.71      | 1.54       | 8.32    |
| 3   | 15  | 10| 16|   | 517  | 1.40      | 17.96      | 11.54    | 1.31      | 7.66       | 8.36    |
| 4   | 10  | 9 | 20|   | 647  | 1.89      | 16.02      | 11.91    | 1.75      | 9.73       | 8.43    |
| 5   | 15  | 7 | 12|   | 748  | 1.98      | 11.28      | 11.59    | 1.00      | 3.94       | 8.40    |
| 6   | 20  | 6 | 14|   | 806  | 1.67      | 4.19       | 11.73    | 0.80      | 2.53       | 8.32    |
| 7   | 10  | 10| 13|   | 631  | 2.09      | 20.52      | 10.94    | 1.01      | 8.54       | 8.07    |
| 8   | 13  | 9 | 10|   | 551  | 1.15      | 16.67      | 11.08    | 1.67      | 9.57       | 8.17    |
| 9   | 80  | 14| 5 | 15| 1158 | 1.40      | 9.68       | 23.13    | 1.66      | 4.00       | 17.01  |
| 10  | 14  | 7 | 14|   | 873  | 1.56      | 14.02      | 24.36    | 1.58      | 7.01       | 17.77  |
| 11  | 15  | 6 | 18|   | 953  | 1.03      | 9.02       | 23.72    | 1.10      | 3.86       | 17.65  |
| 12  | 17  | 9 | 18|   | 802  | 1.14      | 9.78       | 24.62    | 1.75      | 4.25       | 18.21  |
| 13  | 10  | 6 | 11|   | 994  | 2.07      | 17.92      | 23.06    | 1.70      | 11.57      | 17.31  |
| 14  | 14  | 9 | 12|   | 673  | 1.67      | 21.90      | 23.11    | 1.63      | 13.57      | 17.14  |
| 15  | 18  | 9 | 15|   | 744  | 1.59      | 12.08      | 23.74    | 1.53      | 7.89       | 17.80  |
| 16  | 17  | 10| 12|   | 676  | 1.17      | 14.28      | 23.82    | 1.87      | 8.26       | 17.70  |
| 17  | 100 | 15| 7 | 12| 1140 | 1.31      | 10.10      | 42.33    | 1.11      | 10.73      | 31.89  |
| 18  | 15  | 6 | 14|   | 1211 | 1.31      | 9.49       | 41.20    | 1.41      | 6.00       | 31.55  |
| 19  | 19  | 5 | 17|   | 1465 | 0.99      | 4.98       | 42.50    | 1.00      | 1.97       | 32.02  |
| 20  | 10  | 5 | 10|   | 1197 | 2.80      | 20.98      | 40.83    | 1.75      | 14.01      | 30.87  |
| 21  | 17  | 8 | 10|   | 878  | 1.91      | 14.39      | 42.92    | 1.46      | 11.20      | 32.83  |
| 22  | 20  | 7 | 10|   | 998  | 1.80      | 11.12      | 42.61    | 0.84      | 6.91       | 32.60  |
| 23  | 15  | 8 | 19|   | 996  | 3.11      | 15.26      | 42.69    | 1.80      | 10.48      | 32.59  |
| 24  | 11  | 8 | 11|   | 1044 | 2.30      | 19.32      | 41.15    | 1.60      | 16.23      | 31.39  |
| 25  | 120 | 10| 6 | 12| 1398 | 1.93      | 22.82      | 68.54    | 1.46      | 16.18      | 52.10  |
| 26  | 17  | 7 | 20|   | 1182 | 1.25      | 11.72      | 68.42    | 1.08      | 9.25       | 52.54  |
| 27  | 13  | 6 | 19|   | 1474 | 2.51      | 13.30      | 65.25    | 1.61      | 8.98       | 49.98  |
| 28  | 15  | 9 | 19|   | 887  | 1.12      | 19.87      | 63.33    | 0.90      | 16.22      | 48.29  |
| 29  | 15  | 7 | 16|   | 1229 | 1.64      | 11.96      | 62.62    | 1.24      | 9.16       | 47.81  |
| 30  | 16  | 9 | 17|   | 1085 | 1.32      | 15.74      | 62.98    | 1.53      | 10.95      | 48.75  |
| 31  | 18  | 6 | 10|   | 1152 | 0.80      | 12.63      | 63.44    | 1.75      | 10.12      | 49.37  |
| 32  | 15  | 10| 12|   | 852  | 1.46      | 21.04      | 62.97    | 1.70      | 19.30      | 48.45  |

Avg. |    |    |    |    |      | 1.64   | 13.69   | 35.50   | 1.36      | 8.82       | 26.90  |

[DOI: 10.1299/jamds.2018jamdsn0041] © 2018 The Japan Society of Mechanical Engineers
Fig. 7. Polynomial growth trend in computing time by GA_SS for large-sized problems

Fig. 8. Mean plots and Tukey HSD intervals at the 95% confidence level of GA_ST, GA_TS and GA_SS

(a) Group by job size
(b) Group by truck size
(c) Group by machine size
(d) Group by customer size

Fig. 9. Mean plots and Tukey HSD intervals at the 95% confidence level for job size, truck size, machine size and customer size.
To validate the results, it is interesting to check if the observed differences in the RPD\textsuperscript{a} values of each implemented algorithm are statistically significant. Figure 8 shows the mean plots and Tukey HSD intervals at the 95% confidence level for all problems in Table 2, and clearly shows that there are statistically significant differences between the RPD\textsuperscript{a} values among GA\textsubscript{ST}, GA\textsubscript{TS}, and GA\textsubscript{SS} because there is no overlap between the algorithms. The observed differences between GA\textsubscript{SS} and the other algorithms (GA\textsubscript{ST} and GA\textsubscript{TS}) are more statistically significant as the number of jobs and the number of trucks increase, as shown in Graphs (a) and (b) in Figure 9. Meanwhile, the observed differences are less statistically significant as the number of machines and the number of customers increase, as shown in Graph (c) and (d) in Figure 9. This is mainly due to the complexity of the problems, as the complexity increases as the number of jobs and the number of trucks increase. However it decreases as the average number of jobs per machine and the average number of trucks per customer decrease. The results in Figure 8 indicate that GA\textsubscript{SS} gives the best performance for the integrated machine scheduling, batching, and truck delivery scheduling problem in any job, truck, machine, and customer size.

We evaluated the performances of the GA-based heuristics by randomly generated test problems with the processing time uniformly generated from [60, 120] and the transportation time uniformly generated from [60, 240]. To verify that the range of processing/transportation times do not affect the performance of the proposed algorithms, we additionally test the algorithms with various ranges of processing/transportation times. The test groups of processing/transportation time range are listed in Table 3, and the test results are summarized in Table 4.

### Table 3. Groups of processing/transportation time range

| Group | Processing Time | Transportation Time |
|-------|-----------------|---------------------|
| A     | U[60, 120]      | U[60, 120]          |
| B     | U[120, 240]     | U[120, 240]         |
| C     | U[60, 120]      | U[120, 240]         |
| D     | U[120, 240]     | U[60, 120]          |
| E     | U[60, 120]      | U[60, 240]          |
| F     | U[60, 240]      | U[60, 120]          |
| G     | U[60, 240]      | U[60, 240]          |

### Table 4. Test results for groups of processing/transportation time range

| No. | J/M/T/C | Group | Best | GA\textsubscript{ST} | GA\textsubscript{TS} | GA\textsubscript{SS} |
|-----|---------|-------|------|-----------------------|-----------------------|-----------------------|
|     |         |       | MAD (%) | RPD (%) | Time (Sec) | MAD (%) | RPD (%) | Time (Sec) | MAD (%) | RPD (%) | Time (Sec) |
| 1   | 6/19/5/16 | A     | 591   | 1.74 | 7.77 | 12.18 | 1.20 | 2.61 | 9.15 | 0.35 | 0.42 | 8.95 |
| 2   |          | B     | 1368  | 2.05 | 4.63 | 12.27 | 0.38 | 2.08 | 9.06 | 0.24 | 0.31 | 8.85 |
| 3   |          | C     | 1264  | 0.99 | 2.12 | 12.36 | 0.57 | 0.80 | 9.15 | 0.30 | 0.65 | 9.01 |
| 4   |          | D     | 748   | 2.05 | 16.36 | 12.38 | 1.78 | 7.26 | 9.18 | 0.56 | 0.63 | 8.96 |
| 5   |          | E     | 1063  | 1.13 | 2.81 | 12.18 | 0.37 | 1.64 | 9.09 | 0.15 | 0.32 | 8.88 |
| 6   |          | F     | 596   | 1.46 | 18.07 | 12.05 | 2.00 | 8.89 | 8.93 | 0.43 | 0.97 | 8.84 |
| 7   |          | G     | 910   | 3.40 | 10.40 | 11.78 | 0.95 | 5.84 | 8.75 | 0.47 | 0.67 | 8.66 |
| 8   | 8/14/5/15 | A     | 767   | 1.52 | 14.63 | 23.26 | 1.55 | 9.84 | 17.39 | 0.60 | 0.77 | 16.74 |
| 9   |          | B     | 1613  | 1.71 | 12.53 | 23.44 | 1.55 | 8.48 | 17.50 | 0.41 | 0.85 | 16.74 |
| 10  |          | C     | 1513  | 1.24 | 4.68 | 23.31 | 0.44 | 1.36 | 17.51 | 0.24 | 0.31 | 16.72 |
| 11  |          | D     | 1166  | 1.14 | 10.10 | 23.51 | 1.03 | 5.86 | 17.65 | 0.64 | 1.02 | 16.96 |
| 12  |          | E     | 1236  | 2.19 | 8.23 | 23.28 | 0.71 | 3.01 | 17.53 | 0.30 | 0.57 | 16.82 |
| 13  |          | F     | 1012  | 1.37 | 11.17 | 23.13 | 1.24 | 6.99 | 17.39 | 0.31 | 0.84 | 16.80 |
| 14  |          | G     | 1401  | 2.52 | 15.07 | 23.60 | 0.96 | 8.82 | 17.91 | 0.47 | 0.89 | 17.21 |
| 15  | 10/15/7/12 | A     | 710   | 1.34 | 21.06 | 42.96 | 1.57 | 15.76 | 32.51 | 0.47 | 0.93 | 31.33 |
| 16  |          | B     | 1405  | 2.05 | 18.33 | 42.42 | 1.17 | 16.15 | 32.31 | 0.54 | 1.40 | 30.68 |
| 17  |          | C     | 1132  | 1.80 | 12.72 | 41.26 | 1.22 | 8.95 | 31.22 | 0.36 | 0.41 | 29.63 |
| 18  |          | D     | 1317  | 0.73 | 7.38 | 42.39 | 0.84 | 4.60 | 32.15 | 0.60 | 0.78 | 30.76 |
Figure 10 shows the mean plots and Tukey HSD intervals at the 95% confidence level for all problems in Table 4, and clearly shows that the range of processing/transportation times do not affect the performance of the proposed algorithms.

6. Conclusions

This study considers batch delivery scheduling of trucks integrated with parallel machine schedule of job orders from multi-customers to minimize the delivery completion times of whole the batched jobs. The jobs are first manufactured by one of identical parallel machines and then batched jobs are delivered to corresponding customers by identical trucks. If one of available trucks is ready, a group of jobs should be loaded and delivered to the corresponding customer. The main decision here is to simultaneously determine machine scheduling, batching, and truck delivery scheduling to minimize the delivery completion times of whole the batched jobs.

We then propose two approaches to address the problem. The first approach uses a mixed integer programming model. Since the problem is NP-hard, we propose three GA-based heuristics (GA_ST, GA_TS, and GA_SS) to increase solution efficiency. The test results indicate that GA_SS is solvable within a reasonable computing time with the best performance for operational problems in practice.

There are several possible interesting extensions to our study. First, this study assumes that each truck delivers products to only one customer. Integrating the routing problem with the current study, where each truck visits more than one customer, would be more challenging. Second, this study deals with identical parallel machine scheduling at a manufacturing plant. To generalize this study across manufacturing plants, non-identical parallel machine scheduling would be more interesting and practical. Finally, there are also potentially unlimited opportunities based on other meta-
heuristics, such as populated variable neighborhood search (p-VNS) and harmonic search (HS), among others.

Acknowledgment

This work was supported by Dongseo University, “Dongseo Frontier Project” Research Fund of 2015.

References

Arshinder, Kanda, A., Deshmukh, S.G., 2008. Supply chain coordination: Perspectives, empirical studies and research directions. Int. J. Prod. Econ. doi:10.1016/j.ijpe.2008.05.011

Cakici, E., Mason, S.J., Fowler, J.W., Geismar, H.N., 2013. Batch scheduling on parallel machines with dynamic job arrivals and incompatible job families. Int. J. Prod. Res. 51, 2462–2477. doi:10.1080/00207543.2012.748227

Cakici, E., Mason, S.J., Geismar, H.N., Fowler, J.W., 2014. Scheduling parallel machines with single vehicle delivery. J. Heuristics 20, 511–537. doi:10.1007/s10732-014-9249-y

Chang, Y.C., Lee, C.Y., 2004. Machine scheduling with job delivery coordination. Eur. J. Oper. Res. 158, 470–487. doi:10.1016/S0377-2217(03)00364-3

Chang, Y.C., Li, V.C., Chiang, C.J., 2014. An ant colony optimization heuristic for an integrated production and distribution scheduling problem. Eng. Optim. 46, 503–520. doi:10.1080/0305215x.2013.786062

Chen, B., Lee, C.Y., 2008. Logistics scheduling with batching and transportation. Eur. J. Oper. Res. 189, 871–876. doi:10.1016/j.ejor.2006.11.047

Chen, H.K., Hsueh, C.F., Chang, M.S., 2009. Production scheduling and vehicle routing with time windows for perishable food products. Comput. Oper. Res. 36, 2311–2319. doi:10.1016/j.cor.2008.09.010

Chen, Z.-L., 2010. Integrated Production and Outbound Distribution Scheduling: Review and Extensions. Oper. Res. 58, 130–148. doi:10.1287/opre.1080.0688

Delavar, M.R., Hajiaghaei-Kesheteli, M., Molla-Alizadeh-Zavardehi, S., 2010. Genetic algorithms for coordinated scheduling of production and air transportation. Expert Syst. Appl. 37, 8255–8266. doi:10.1016/j.eswa.2010.05.060

Gao, S., Qi, L., Lei, L., 2015. Integrated batch production and distribution scheduling with limited vehicle capacity. Int. J. Prod. Econ. 160, 13–25. doi:10.1016/j.ijpe.2014.08.017

Gen, M., Cheng, R., 2000. Genetic algorithms and engineering optimization. John Wiley & Sons.

Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R., 1979. Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey. Ann. Discret. Math. 5, 287–326. doi:10.1016/S0167-5060(08)70356-X

Hajiaghaei-Kesheteli, M., 2011. The allocation of customers to potential distribution centers in supply chain networks: GA and AIA approaches, in: Applied Soft Computing Journal. pp. 2069–2078. doi:10.1016/j.asoc.2010.07.004

Hajiaghaei-Kesheteli, M., Aminnayeri, M., Ghomi, S.M.T.F., 2014. Integrated scheduling of production and rail transportation. Comput. Ind. Eng. 74, 240–256.

Hajiaghaei-Kesheteli, M., Molla-Alizadeh-Zavardehi, S., Tavakkoli-Moghaddam, R., 2010. Addressing a nonlinear fixed-charge transportation problem using a spanning tree-based genetic algorithm. Comput. Ind. Eng. 59, 259–271. doi:10.1016/j.cie.2010.04.007

Hall, N.G., Potts, C.N., 2005. The coordination of scheduling and batch deliveries. Ann. Oper. Res. 135, 41–64. doi:10.1007/s10479-005-6234-8

Hall, N.G., Potts, C.N., 2003. Supply chain scheduling: batching and delivery. Oper. Res. 51, 566–584. doi:10.1287/opre.51.4.566.16106

Hunter, A.P., Van Buer, M.G., 1996. The newspaper production/distribution problem. J. Bus. Logist. 17, 85–107.

Ji, M., He, Y., Cheng, T.C.E., 2007. Batch delivery scheduling with batch delivery cost on a single machine. Eur. J. Oper. Res. 176, 745–755. doi:10.1016/j.ejor.2005.09.006

Joo, C.M., Kim, B.S., 2012. Genetic algorithm with an effective dispatching method for unrelated parallel machine scheduling with sequence dependent and machine dependent setup times. IE interfaces 25, 357–364.

Lee, C.Y., Chen, Z.L., 2001. Machine scheduling with transportation considerations. J. Sched. 4, 3–24. doi:10.1002/1099-1425(20010102)4:1<3::AID-JOS57>3.0.CO;2-D

Lee, K., Kim, B.S., Joo, C.M., 2012. Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance. Expert Syst. Appl. 39, 12975–12983.

Li, K., Ganesan, V.K., Sivakumar, A.I., 2006. Scheduling of single stage assembly with air transportation in a consumer
Li, K., Ganesan, V.K., Sivakumar, A.I., Mathirajan, M., 2005. Methodologies for synchronised scheduling of assembly and air transportation in a consumer electronics supply chain. Int. J. Logist. Syst. Manag. 2, 52–67.

Li, K., Sivakumar, A.I., Ganesan, V.K., 2008. Analysis and algorithms for coordinated scheduling of parallel machine manufacturing and 3PL transportation. Int. J. Prod. Econ. 115, 482–491. doi:10.1016/j.ijpe.2008.07.007

Li, K., Sivakumar, A.I., Mathirajan, M., Ganesan, V.K., 2004. Solution methodology for synchronizing assembly manufacturing and air transportation of consumer electronics supply chain. Int. J. Bus. 9.

Li, K.P., Ganesan, V.K., Sivakumar, A.I., 2005. Synchronized scheduling of assembly and multi-destination air-transportation in a consumer electronics supply chain. Int. J. Prod. Res. 43, 2671–2685. doi:10.1080/00207540500066895

Mazdeh, M.M., Sarhadi, M., Hindi, K.S., 2007. A branch-and-bound algorithm for single-machine scheduling with batch delivery minimizing flow times and delivery costs. Eur. J. Oper. Res. 183, 74–86. doi:10.1016/j.ejor.2006.09.087

Potts, C.N., Strusevich, V.a., 2009. Fifty years of scheduling: a survey of milestones. J. Oper. Res. Soc. 60, S41–S68. doi:10.1057/jors.2009.2

Pundoor, G., Chen, Z.L., 2009. Joint cyclic production and delivery scheduling in a two-stage supply chain. Int. J. Prod. Econ. 119, 55–74. doi:10.1016/j.ijpe.2009.01.007

Tang, L., Liu, P., 2009. Two-machine flowshop scheduling problems involving a batching machine with transportation or deterioration consideration. Appl. Math. Model. 33, 1187–1199. doi:10.1016/j.apm.2008.01.013

Van Buer, M.G., Woodruff, D.L., Olson, R.T., 1999. Solving the medium newspaper production/distribution problem. Eur. J. Oper. Res. 115, 237–253. doi:10.1016/S0377-2217(98)00300-2

Xuan, H., 2011. Hybrid Flowshop Scheduling withFinite Transportation Capacity. Appl. Mech. Mater. 65, 574–578. doi:10.4028/www.scientific.net/AMM.65.574

Zandieh, M., Molla-Alizadeh-Zavardehi, S., 2009. Synchronizing production and air transportation scheduling using mathematical programming models. J. Comput. Appl. Math. 230, 546–558. doi:10.1016/j.cam.2008.12.022

Zandieh, M., Molla-Alizadeh-Zavardehi, S., 2008. Synchronized production and distribution scheduling with due window. J. Appl. Sci. 8, 2752–2757. doi:10.3923/jas.2008.2752.2757

Zegordi, S.H., Abadi, I.N.K., Nia, M.A.B., 2010. A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. Comput. Ind. Eng. 58, 373–381. doi:10.1016/j.cie.2009.06.012

Zhong, W., Zhong, W., Dósa, G., Tan, Z., 2007. On the machine scheduling problem with job delivery coordination. Eur. J. Oper. Res. 182, 1057–1072. doi:10.1016/j.ejor.2006.09.059