International Journal of Innovative
Computing, Information and Control
Volume 8, Number 1(B), January 2012
 ISSN 1349-4198
pp. 553–566

SCHEDULING OF PARALLEL MACHINES WITH JOB DELIVERY COORDINATION

Cheng-Hsiang Liu[1], Bor-Yuh Leu[2] and Sheng-Yuan Hsu[3]

[1]Department of Industrial Management
National Pingtung University of Science and Technology
No. 1, Shuefu Rd., Neipu, Pingtung 912, Taiwan
chliu.uw@gmail.com

[2]Department of Industrial Engineering and Management
Minghsin University of Science and Technology
No. 1, Xinxing Rd., Xinfeng, Hsinchu 30401, Taiwan

[3]Department of Industrial Engineering and Management
Chienkuo Technology University
No. 1, Chieh-Sou N. Rd., Changhua 500, Taiwan

Received September 2010; revised January 2011

Abstract. *This paper examines an integrated scheduling model for production and distribution operations. In this model, a set of jobs involving different amounts of storage space in delivery trucks was processed by either one of two parallel machines and delivered by a single truck to one customer area. The objective was to minimize the time required for all jobs to be completed and delivered to the customer area, and the truck returned to the factory; this variable was denoted as $C_{\max}$. This problem was shown to be NP-hard in the strong sense. Previous related studies have focused on developing optimization-based solutions; however, such optimization-based approaches require a significant amount of computing time to produce an optimal solution. The time required to compute a production schedule is very important in business practice, and thus, we need to develop an effective heuristic to solve these problems. Two heuristics (H1 and H3) and a GA-based algorithm were developed by this study, and simulation experiments were conducted to evaluate the performance of the proposed approaches. The experimental results showed that if transportation time is short or truck capacity is large, then the proposed heuristic H3 should be the scheduling approach of choice; otherwise, the GA-based algorithm is the optimal scheduling approach.*

Keywords: Parallel, Machines scheduling, Heuristic, Genetic algorithms, Production-distribution coordination

1. Introduction.

Supply chain management has become one of the most important topics in manufacturing research over the past decade. A supply chain includes all interaction between suppliers, manufacturers, distributors and customers. The popularity of just-in-time (JIT) concepts has created a trend in which interactions between the various stages in a supply chain are considered increasingly important, especially, interaction occurring between the job scheduling (production stage) and delivery of final products (distribution stage). These concepts have also enhanced the practical value of coordinated models. However, because traditional scheduling assumes an infinite number of available vehicles to deliver products to customers and does not take into account the time required for deliveries, such processes assume that goods can be delivered to customers without delay [1-3]. Thus, in recent years, how to achieve optimal coordination of production and distribution stages in the supply chain to produce ideal overall system performance has become an increasing focus of attention for both industry practitioners and academic researchers.

Wang and Cheng [4] studied parallel machine scheduling in terms of batch delivery cost. They showed that the problem of minimizing the sum of total flow time and delivery cost is NP-complete in the strong sense, and provided a dynamic programming algorithm to solve the problem. Lee and Chen [5] analyzed the complexities of a category of scheduling problems related to the coordination of machine scheduling and job transportation. They demonstrated the computational difficulty of many of these problems and proposed polynomial and pseudo-polynomial algorithms to solve some of the problems. Chang and Lee [6] extended Lee and Chen's work to situations in which each order occupies a different amount of physical space in a delivery truck. Three different problems were discussed in their work with the aim of minimizing the time required for all jobs to be completed and delivered to the customer area and the truck returned: (i) single machine scheduling with delivery of completed jobs to one customer area; (ii) parallel machine scheduling with delivery of completed jobs to one customer area; and (iii) single machine scheduling with delivery of completed jobs to two customer areas. They proved that these problems are NP-hard in the strong sense, and presented three polynomial time heuristics to solve the three problems: H1 with a worst-case ratio of 5/3 (problem (i)); H2 with a worst case ratio of 2 (problem (ii)); and H3 with a worst case ratio of 2 (problem (iii)). He et al. [7] presented a modification of H1 (MH1) by applying a fully polynomial time approximation scheme (FPTAS) to the knapsack problem, with a worst-case ratio of 53/35. Additionally, a modification of MH1 that leads to the optimal algorithm with a worst-case ratio of $3/2 + \varepsilon$ (where ε is a positive number and can be arbitrarily close to 0) was proposed by Zhong et al. [8]. Zhong et al. [8] also presented a modification of H2 (MH2) with a worst-case ratio of 5/3; MH1 and MH2 both incorporated an FPTAS for the knapsack problem as a sub-procedure. Lu and Yuan [9] provided a heuristic with an optimal worst-case performance ratio of 3/2 to solve the problem of single machine scheduling with delivery of completed jobs to one customer area. Chen and Vairaktarakis [10] considered a two-stage scheduling problem in which the first stage consists of manufacturing and the second stage is delivery to customers. Two machine configurations were included in the processing facility – single machine and parallel machines. Their objective was to combine customer service level and total distribution cost. Customer service level is measured by a function of the times when jobs are delivered to customers. For each of the problems studied, they provided an algorithm or a proof of intractability accompanied by a heuristic algorithm with worst-case and asymptotic performance analysis. Li et al. [11] considered a single-machine scheduling model that incorporated the route decisions of a delivery truck driver serving customers at different locations, to minimize the sum of time required for each order to reach the respective customer. Li et al. first demonstrated that the problem under examination was NP-hard in the strong sense and then developed polynomial time algorithms for a number of special problem cases. Another set of scheduling problems somewhat related to those discussed in this study related to two-stage scheduling models that took issues of job priority into account [12,13].

This study began by examining a simplified version of the two-stage scheduling problem in which the first stage is job production and the second stage is job delivery. Regarding the case in which all jobs are processed by either one of two parallel machines and delivered by a single truck to one customer area, the objective was to minimize the time required for all jobs to be completed and delivered to the customer area and the truck returned; this variable was denoted as $C_{\max}$. Each job occupies a different amount of physical space in the delivery truck. In such a case, if several different jobs are combined into one batch for delivery purposes, this reduces the number of individual deliveries; however, reduction in number of individual deliveries may result in greater $C_{\max}$. Chang and Lee [6] and Zhong et al. [8] proved that this problem is NP-hard in the strong sense. This implies that determining the optimal schedule using mathematical models may require much time and computational resources. The time required to compute a coordinated schedule is very important in business practice. Although the H2 proposed by Chang and Lee [6] produces "good" solutions more quickly than the optimization model, its performance is very sensitive to problem instance variability. Therefore, this paper first presents two modified heuristics, based on H2, which lead to a better possible algorithm. Additionally, this study developed a genetic algorithm (GA) based procedure in an attempt to provide effective solutions within a reasonable amount of time. The meta-heuristic approach and the two heuristics were compared with H2 in terms of solution quality. Experimental results showed that the proposed heuristics performed remarkably well, which led to the conclusion that these heuristics are significantly superior to H2 in terms of $C_{\max}$.

The remainder of this paper is organized as follows: problem formulation is described in Section 2; in Sections 3 and 4, the mechanisms of the proposed algorithm are discussed; some general observations are presented and the significance of the experimental results is discussed in Section 5; in Section 6, we provide a summary of the results and suggestions for future research.

2. Problem Statement and Notations.

The problem of parallel machine scheduling and job delivery can be described as follows: there are n jobs $(J_1, J_2, ..., J_n)$ with each job (J_i) having a processing time (p_i) and a size of (s_i), which represents the physical space J_i occupies when this job is loaded in the vehicle. These jobs are first processed by one of the two parallel machines. The completed jobs are then delivered by a truck with fixed capacity to one customer area. The truck is initially located at the factory, is available to deliver finished jobs in batches, and has a capacity Q. That is, the total physical space occupied by the jobs loaded into the truck at any given time cannot exceed Q. Each delivery involves the same transportation time T. The purpose of this study was to schedule jobs in such a manner so as to minimize $C_{\max}$.

2.1. Notations.

B_k: the set of all jobs in the kth delivery batch, $k = 1, 2, ..., B$.

B: the number of delivery batches, $\left\lceil \sum_{i=1}^{n} s_i/Q \right\rceil \leq B \leq n$.

Ct_k: the time of delivery of B_k by the truck to the customer and return to the factory.

$Mt_k^{(1)}$: the completion time of processing of B_k by machine 1.

$Mt_k^{(2)}$: the completion time of processing of B_k by machine 2.

St_k: the time of departure of the truck to deliver B_k.

$$X_{ijk} = \begin{cases} 1, & \text{if job } i \text{ is assigned to be processed by machine } j \text{ and belongs to delivery batch } k; \\ 0, & \text{if otherwise.} \end{cases}$$

2.2. Problem formation.

The following mixed integer programming (MIP) model represents the problem investigated in this paper. This model can be used to determine a coordinated schedule of production and distribution with the goal of minimizing $C_{\max}$.

$$Min \quad Z = C_{\max} \tag{1}$$

$$s.t. \quad \sum_{j=1}^{2} \sum_{k=1}^{B} X_{ijk} = 1, \quad \forall i \tag{2}$$

$$\sum_{i=1}^{N} \sum_{j=1}^{2} s_i X_{ijk} \leq Q, \quad \forall k \tag{3}$$

$$Mt_k^{(1)} \geq Mt_{k-1}^{(1)} + \sum_{i=1}^{n} p_i X_{i1k}, \quad \forall k,\ k > 1 \tag{4}$$

$$Mt_k^{(2)} \geq Mt_{k-1}^{(2)} + \sum_{i=1}^{n} p_i X_{i2k}, \quad \forall k,\ k > 1 \tag{5}$$

$$St_k \geq Mt_k^{(1)}, \quad \forall k \tag{6}$$

$$St_k \geq Mt_k^{(2)}, \quad \forall k \tag{7}$$

$$Ct_1 \geq St_1 + T \tag{8}$$

$$Ct_k \geq Ct_{k-1} + T, \quad \forall k \tag{9}$$

$$Ct_k \geq St_k + T, \quad \forall k \tag{10}$$

$$C_{\max} \geq Ct_k, \quad \forall k \tag{11}$$

$$X_{ijk} \in \{0, 1\}, \quad \forall i, j, k \tag{12}$$

$$Mt_k^{(1)} \geq 0, \quad \forall k \tag{13}$$

$$Mt_k^{(2)} \geq 0, \quad \forall k \tag{14}$$

$$St_k \geq 0, \quad \forall k \tag{15}$$

$$Ct_k \geq 0, \quad \forall k \tag{16}$$

$$1 \leq i \leq n \tag{17}$$

$$1 \leq j \leq n \tag{18}$$

The objective function (1) minimizes $C_{\max}$. Each order must be assigned to exactly one machine and that order must be assigned to exactly one batch, as demonstrated in Equation (2). Equation (3) ensures that the collective size of all orders placed in the same batch does not exceed the truck capacity. Equations (4)-(7) require that the time of the departure of the truck to deliver B_k exceeds the maximum time required for completion of any order belonging to a batch on the machines. Equations (8)-(10) define the property of the completion time of delivering of B_k by the truck. These equations indicate that the truck may start to deliver one batch after the jobs of this batch have been processed by the parallel machine and the truck has completed delivery of the previous batch and returned to the factory. In terms of constraints, Equation (11) defines the properties of decision variables $C_{\max}$ and Ct_k. Finally, Equations (12)-(18) are domain constraints for the variables used in the formation.

The model assumptions are as follows.

- The loading and unloading times are included in the transportation times of the jobs, and all transportation times are assumed to be job-independent.
- Preemption is disallowed, i.e., once the processing of a job has begun, it cannot be stopped.
- The machines cannot process more than one job at any given time.
- Unlimited buffer for WIP.
- Resource storage and machine failure are not considered.
- All jobs are ready for processing at the beginning of each planning period.
- All jobs in Batch k precede each of those in Batch $k + 1$.

3. **Heuristics.** Although the MIP model provides the optimal solution, variables and constraints increase drastically when the number of jobs increases. Therefore, Chang and Lee [6] presented a heuristic (H2) with a worst case ratio of 2 to solve the coordinated scheduling problem. The design of H2 can be described as follows.

Step 1: Assign jobs to delivery batches using the First Fit Decreasing (FFD) algorithm. Let the total number of the resulting batches be b^{H2}.
Step 2: Calculate the sum of the processing times of the jobs in Bk and denote it as P_k, for $k = 1, 2, ..., b^{H2}$. Re-index these batches so that $P_1 \leq P_2 \leq ... \leq P_b^{H2}$.
Step 3: Beginning with B_1, assign batches one by one to the machine that has the smaller load before the batch is assigned. Within each batch, jobs are sequenced arbitrarily.
Step 4: Dispatch each completed but undelivered batch whenever the delivery truck becomes available. If multiple batches have been completed when the delivery truck is available, then dispatch the batch with the smallest index.

The steps of the FFD algorithm can be described as follows [6].

Step 1: First sort the jobs in order of descending size.
Step 2: Assign the largest job to B_1.
Step 3: If the ith largest job is considered then assign it to the lowest indexed batch, such that the total job size of the corresponding batch does not exceed Q.

Although H2 produces "good" solutions more quickly than the optimization model, its performance is very sensitive to problem instance variability. This study presents two modified heuristics, H1 and H3, to improve the performance of H2. According to the lemma of Chang and Lee [6], the makespan of the optimal schedule (C^*) is greater than or equal to $Max(u + K \times T, C(M)^* + T)$, where u is the departure time of the jobs in the first delivery batch, K is the number of delivery batches, and $C(M)^*$ is the point in the optimal schedule when the machines have finished processing the last job. The idea behind the proposed H1 is to schedule jobs to minimize $C_{\max}$ as much as possible to $C(M)^* + T$. Hence, H1 allocates jobs to the machines based on the procedure proposed by Sule [14], to achieve a minimum length of time on the optimal schedule required for the machines to complete processing of the last job, $C(M)$.

Heuristic 1 (H1)

Step 1: Arrange the jobs in descending order of processing time.
Step 2: The lower bound of the minimum achievable makespan is given by the sum of the processing times divided by 2.
Step 3: Begin allocating the jobs to one machine until one of the following occurs:
(a) The sum of the processing times of the jobs assigned to the machine under consideration becomes equal to the lower bound. If this happens, begin assigning jobs to the next available machine.
(b) The sum of the processing times of the jobs allocated to the machine exceeds the lower bound. If this happens, then the job that has caused the sum to exceed the lower bound is allocated in the following manner: If the sum of the processing times on the other machine is less than the lower bound, and the allocation of the job will not cause the cumulative processing time of the machine to exceed the lower bound, the job is assigned to this machine. If the assignment of the job will increase the sum beyond the lower bound in both machines, the job is assigned to the machine in which such increase will be minimal.
Step 4: In each machine, jobs are sequenced in SPT order.
Step 5: Calculate the first delivery time (ρ_1). The ρ_1 is defined as $Max(C_1, C(M) \bmod T)$, where C_1 is the point in time when the machines complete processing of the first job. The set of all possible delivery times is $\{\rho_1, \rho_1 + T, \rho_1 + 2T, ..., \rho_1 + KT\}$.
Step 6: Assign the jobs to batches using the FFD algorithm with the constraint that each job must be finished at or before its delivery time.

The Heuristic 3 (H3) is presented below. One point prevents further enhancement of the performance of H2. In H2, machine loads may sometimes become unbalanced, which can cause problems in minimizing $C_{\max}$, because H2 assigns delivery batches one by one to the machine that has a smaller load before the batch is assigned (all jobs in the same batch are assigned to the same machine). By contrast, after grouping jobs into delivery batches, H3 allocates jobs one by one to the machines while considering the existing machine loads prior to job assignment, in order to achieve a minimum $C_{\max}$.

Heuristic 3 (H3)

Step 1: Assign jobs to batches using the FFD algorithm. Let the total number of resulting batches be b^{H3}.

Step 2: Calculate the sum of the processing times of the jobs in B_k and denote it as P_k, for $k = 1, 2, ..., b^{H3}$. Re-index these batches so that $P_1 \leq P_2 \leq ... \leq P_b^{H3}$.

Step 3: Within each batch, jobs are sequenced according to the LPT (Longest Processing Time) rule. Beginning with B_1, assign jobs one by one to the machine that has a smaller load before the job is assigned (all jobs in the same batch may not be assigned to the same machine).

Step 4: Dispatch each completed but undelivered batch whenever the truck becomes available. If multiple batches have been completed when the truck becomes available, deliver the one with the smallest batch index k.

4. GA-based Algorithm.

The genetic algorithm (GA), which was first introduced by Holland [15], has proven to be particularly useful for solving complex combinational problems. This paper considers the case in which a GA is used to determine a coordinated schedule. First, the GA chromosome structure must be defined. In this study, a chromosome was divided into three segments: job sequence, job-to-machine assignment, and job-to-batch assignment. If we consider an example problem of six jobs, then the coordinated schedule can be represented as illustrated in Figure 1.

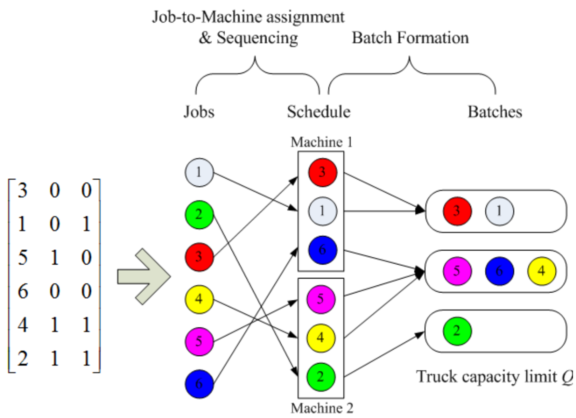

FIGURE 1. The chromosome structure of the proposed GA

where Y_{i1} denotes the index of the job in position J of the job sequence ($J \in [1, ..., 6]$). If a job is assigned to machine 1, the gene Y_{i2} is set to zero; if assigned to machine 2, it is set to one. If a job is the last job of a batch, Y_{i3} is set to one; otherwise it is set to zero.

The following operations describe one generation of a GA. A fixed number of chromosomes are generated to create the initial population of the GA. Let the size of the initial population be PS chromosomes. The fitness of each chromosome is obtained from

Equation (19).

$$FV_c = \frac{MAX - Z_c + MIN}{AVE}, \quad \forall c = 1, \ldots, PS \tag{19}$$

where FV_c is the fitness value of the cth chromosome, MAX is the maximum objective value of the same generation, Z_c is the objective value of the cth chromosome, MIN is the minimum objective value of the same generation and AVE is the average objective value of the same generation. Once the fitness value of each chromosome has been assessed, the best $\theta\%$ of the population is transferred from the previous generation to the current generation. A total of $PS \times (100 - \theta)\%$ new chromosomes must now be generated. The Roulette Wheel Selection, which selects members from the population of chromosomes in a manner proportional to their fitness, is implemented to select chromosomes from the previous generation for the crossover operation. Under this scheme, the fitter chromosomes have a higher probability of being chosen. Two chromosomes are selected, which are commonly referred to as *parents*. A feasible subset of genes is swapped between two parents, producing two new chromosomes, referred to as *offspring*. After a few iterations of crossover operations, the objective of each chromosome in the population often tends to reach some common value. To mitigate this, mutation is used to propagate offspring with more diverse characteristics. Hence, after the crossover operation, mutation is applied subject to the probability of introducing new genes within the selected chromosome. After the offspring are created, their fitness values are assessed. The performance of the crossover and mutation operations depends mainly on the representation of results used. In the representation of results in this study, a chromosome was divided into three segments: job sequence, job-to-machine assignment and job-to-batch assignment. Under this representation, row-based crossover (RX) and row-based mutation (RM) could be applied to the coordinated scheduling problem. In order to implement RX, a random row index was generated. The genes in the rows were interchanged with two chromosomes to form two new offspring. Figure 2 shows that the genes in the third row were exchanged between Parent 1 and Parent 2. In this study, the RM operation was implemented and defined as the re-generation of the Y_{ij} values in randomly selected row i. Figure 3 shows a mutation operation in the example described above. Figure 3 shows that the genes in the fourth row were mutated. This procedure was repeated until the terminating criteria were met.

$$\begin{array}{ccc} \textit{Parent } 1 & & \textit{Offspring } 1 \\ \begin{bmatrix} 3 & 0 & 0 \\ 1 & 0 & 1 \\ 5 & 1 & 0 \\ 6 & 1 & 0 \\ 4 & 0 & 1 \\ 2 & 1 & 1 \end{bmatrix} & \Rightarrow & \begin{bmatrix} 3 & 0 & 0 \\ 1 & 0 & 1 \\ 4 & 0 & 1 \\ 6 & 1 & 0 \\ 5 & 0 & 1 \\ 2 & 1 & 1 \end{bmatrix} \\ \textit{Parent } 2 & & \textit{Offspring } 2 \\ \begin{bmatrix} 6 & 0 & 1 \\ 3 & 1 & 0 \\ 4 & 0 & 1 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \\ 5 & 1 & 1 \end{bmatrix} & \Rightarrow & \begin{bmatrix} 6 & 0 & 1 \\ 3 & 1 & 0 \\ 5 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \\ 4 & 1 & 1 \end{bmatrix} \end{array}$$

FIGURE 2. Crossover operation of the proposed GA

$$\longrightarrow \begin{bmatrix} 3 & 0 & 0 \\ 1 & 0 & 1 \\ 5 & 1 & 0 \\ 6 & 1 & 0 \\ 4 & 0 & 1 \\ 2 & 1 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 6 & 0 & 0 \\ 1 & 0 & 1 \\ 5 & 1 & 0 \\ 3 & 0 & 1 \\ 4 & 0 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

Figure 3. Mutation operation of the proposed GA

As the genes used in both crossover and mutation operations were randomly generated, the feasibility of the resulting offspring was not known in advance. It is evident that both crossover and mutation operations do not always produce a feasible solution. Hence, an additional feasibility checking routine was performed after an offspring was generated. This correction mechanism was designed to move jobs from batches in which capacity had been exceeded to other batches with surplus capacity. This correction mechanism can be algorithmically stated as follows:

```
For each Batch B_k
{
   If Size(B_k) > Q Then
   {
      If Size(B_k) − Q + Size (B_{k+1}) ⩽ Q Then
         Select the job J* that occupies a minimum amount of storage space in the
         truck;
         B_{k+1} = B_{k+1} ∪ J*;
      Else
         Insert a new batch in position (k + 1);
         B_{k+1} = B_{k+1} ∪ J*;
      B_k = B_k − J*;
   }
}
```

5. **Experimental Design and Results.** This study conducted computational experiments to verify the effectiveness of the proposed heuristics H1, H3 and the GA-based algorithm. These methods were coded in eM-Plant 4.6, a simulation package developed by Tecnomatix Technologies Ltd., and implemented in a PC with a Pentium III 1300 MHz CPU and 384 MB RAM. In the experiments, job size and processing time were both uniformly distributed over the integer set $[1, 9]$. The factors to be evaluated were the number of jobs (n), transportation time (T) and truck capacity (Q). The first factor was the number of jobs with four levels $(10, 20, 30, 50)$. The experiments were also conducted to evaluate different scheduling approaches in consideration of the three different levels of T $(5, 10, 15)$ when Q is set as a constant $(Q = 20)$. This study also investigated the effects of the different levels of Q $(15, 20, 25)$ on the performance of the scheduling approaches when $T = 10$. Thus, 24 different treatments were produced for every scheduling approach studied. For each combination of number of jobs, transportation time and truck capacity, researchers randomly generated 10 problem scenarios. In each scenario, the $C_{\max}$ yielded by H1, H3 and the GA-based algorithm were compared with the solution value of algorithm H2 as proposed by Chang and Lee [6].

After some preliminary tests, the GA parameters were set as follows: maximum number of iterations (GEN) = 500, population size (PS) = 100, crossover rate = 0.8 and mutation

TABLE 1. Makespan average for experiments when $Q = 20$

Number of Jobs (n)	Transportation Time (T)	Method	$C_{\max}$	Improved %
50	15	H1	204.8	−1.14
		H2	202.5	–
		H3	199.3	1.58
		GA	***197.0***	***2.72***
	10	H1	143.8	3.55
		H2	149.1	–
		H3	138.8	6.91
		GA	***137.9***	***7.51***
	5	H1	130.8	9.10
		H2	143.9	–
		H3	130.5	9.31
		GA	***130.4***	***9.38***
30	15	H1	128.0	−0.87
		H2	126.9	–
		H3	123.3	2.84
		GA	***120.2***	***5.28***
	10	H1	92.9	4.72
		H2	97.5	–
		H3	88.9	8.82
		GA	***88.5***	***9.23***
	5	H1	81.9	11.46
		H2	92.5	–
		H3	***81.6***	***11.78***
		GA	***81.6***	***11.78***
20	15	H1	94.2	1.46
		H2	95.6	–
		H3	90.9	4.92
		GA	***87.4***	***8.58***
	10	H1	67.7	3.97
		H2	70.5	–
		H3	62.4	11.49
		GA	***60.9***	***13.62***
	5	H1	55.3	14.40
		H2	64.6	–
		H3	***55.2***	***14.55***
		GA	***55.2***	***14.55***
10	15	H1	58.4	−1.04
		H2	57.8	–
		H3	53.5	7.44
		GA	***50.2***	***13.15***
	10	H1	41.5	3.94
		H2	43.2	–
		H3	37.3	13.66
		GA	***36.7***	***15.05***
	5	H1	31.1	18.37
		H2	38.1	–
		H3	30.8	19.16
		GA	***30.5***	***19.95***

TABLE 2. Makespan average for experiments when $T = 10$

Number of Jobs (n)	Truck Capacity (Q)	Method	$C_{\max}$	Improved %
50	15	H1	181.3	−1.85
		H2	178.0	–
		H3	175.6	1.35
		GA	***175.2***	***1.57***
	20	H1	143.8	3.55
		H2	149.1	–
		H3	138.8	6.91
		GA	***137.9***	***7.51***
	25	H1	137.6	7.34
		H2	148.5	–
		H3	133.7	9.97
		GA	***133.6***	***10.03***
30	15	H1	112.0	−3.13
		H2	108.6	–
		H3	107.0	1.47
		GA	***106.3***	***2.12***
	20	H1	92.9	4.72
		H2	97.5	–
		H3	88.9	8.82
		GA	***88.5***	***9.23***
	25	H1	87.0	9.94
		H2	96.6	–
		H3	***84.1***	***12.94***
		GA	***84.1***	***12.94***
20	15	H1	80.0	-3.23
		H2	77.5	–
		H3	74.7	3.61
		GA	***72.9***	***5.94***
	20	H1	67.7	3.97
		H2	70.5	–
		H3	62.4	11.49
		GA	***60.9***	***13.62***
	25	H1	60.7	10.34
		H2	67.7	–
		H3	58.8	13.15
		GA	***58.6***	***13.44***
10	15	H1	52.5	−3.14
		H2	50.9	–
		H3	46.7	8.25
		GA	***45.1***	***11.39***
	20	H1	41.5	3.94
		H2	43.2	–
		H3	37.3	13.66
		GA	***36.7***	***15.05***
	25	H1	38.6	9.81
		H2	42.8	–
		H3	36.6	14.49
		GA	***36.4***	***14.95***

rate = 0.05. The results of the factorial experiment are summarized in Tables 1 and 2. Each item in these tables is an average of the 10 problem scenarios. Boldface and italic are used to indicate the best result for each factor combination. The "Improved %" column in the table indicates the percentage difference between the average objective value as obtained by the current heuristic and that obtained by H2.

Table 1 displays the $C_{\max}$ for four heuristics under $Q = 20$ for different levels of T. Table 1 shows that in the special problem scenario in which transportation time (T) decreases, the performance of the H2 proposed by Chang and Lee [6] grows poorer. The performance of the H3 and GA-based algorithm proposed in this study was surprisingly strong, and surpassed the H2 in all design factor combinations. When T decreased, the performance difference between H1 and H2 became clear. When $n = 50$ and $T = 15$, H2 was superior to H1 by 1.14%. When T decreased, H1 tended to outperform H2 by 9.10%. The average relative gaps between H1, H3 and the GA-based algorithm closely approached 0 when $T = 5$. Similar trends were observed from Table 2, in which H3 and the GA-based algorithm remain the more effective approaches overall. The advantages of H1, H3 and the GA-based algorithm over H2 increased as the given capacity of the truck (Q) increased.

The main objective of this study was to investigate the relative effects of H1, H2, H3 and the GA-based algorithm on different design factor combinations. Because common random-number streams were used to generate the 10 observations in each factor combination, the sample observations were not independent. As a result, it was essential to use the paired t-test to detect any significant statistical differences in the performance of every pair of approaches. To achieve a confidence level of 95%, this study used the *Bonferroni* approach to control the confidence level for each comparison. Tables 3 and 4 show the results of the paired t-tests. The approaches are listed in descending order of performance and are grouped into homogeneous subsets, which are labeled with a different letter if the difference between the means of measuring the performance of the two approaches in the subset did not significantly exceed the prescribed level. Based on the compared measures, the approach with "A" was significantly superior to the approach with "B" and the approach with "B" was significantly superior to the approach with "C". Based on Tables 3 and 4, the tests suggest that the GA-based algorithm and H3 are significantly superior to H2. As Q increased to 25 or T decreased to 5, the test results indicated that H1 significantly outperforms H2.

In summary, H2 is not applicable to problem instances in which the truck is a non-bottleneck, such as when transportation time (T) is reduced or the given capacity of the truck (Q) increases. The reason for these trends may be the fact that when $T = 5$ or $Q = 25$, minimizing $C_{\max}$ is equivalent to minimizing the time required for the machines to complete processing of the last job, $C(M)$. The heuristics proposed in this study, H1 and H3, both clearly consider balance of the machine loads when assigning jobs to the machines in order to minimize $C(M)$, so as to minimize $C_{\max}$. In addition, both H3 and the GA-based algorithm demonstrated significant performance improvement over H2, regardless of which levels T and Q were set to. Both approaches performed comparably if the truck was a non-bottleneck ($T \leq 10$ or $Q \geq 20$). The average computing time was 55s for the GA-based algorithm when the problem size was increased to 50 jobs; H3 required on average less than 1s to find a heuristic solution to a 50-order problem example. This means that if the truck is a non-bottleneck, then the proposed heuristic H3 should be the optimal scheduling approach; otherwise, the GA-based algorithm is the scheduling approach of choice.

Table 3. Results of paired t-test when $Q = 20$

Number of Jobs (n)	Transportation Time (T)	Method	Significance
50	15	GA	A
		H3	B
		H2	C
		H1	C
	10	GA	A
		H3	A
		H1	B
		H2	B
	5	GA	A
		H3	A
		H1	A
		H2	B
30	15	GA	A
		H3	B
		H2	C
		H1	C
	10	GA	A
		H3	A
		H1	B
		H2	B
	5	GA	A
		H3	A
		H1	A
		H2	B
20	15	GA	A
		H3	B
		H1	C
		H2	C
	10	GA	A
		H3	A
		H1	B
		H2	B
	5	GA	A
		H3	A
		H1	A
		H2	B
10	15	GA	A
		H3	B
		H2	C
		H1	C
	10	GA	A
		H3	A
		H1	B
		H2	B
	5	GA	A
		H3	A
		H1	A
		H2	B

Table 4. Results of paired t-test when $T = 10$

Number of Jobs (n)	Truck Capacity (Q)	Method	Significance
50	15	GA	A
		H3	A
		H2	B
		H1	B
	20	GA	A
		H3	A
		H1	B
		H2	B
	25	GA	A
		H3	A
		H1	A
		H2	B
30	15	GA	A
		H3	B
		H2	C
		H1	C
	20	GA	A
		H3	A
		H1	B
		H2	B
	25	GA	A
		H3	A
		H1	A
		H2	B
20	15	GA	A
		H3	B
		H2	C
		H1	C
	20	GA	A
		H3	A
		H1	B
		H2	B
	25	GA	A
		H3	A
		H1	A
		H2	B
10	15	GA	A
		H3	A
		H2	B
		H1	B
	20	GA	A
		H3	A
		H1	B
		H2	B
	25	GA	A
		H3	A
		H1	A
		H2	B

6. **Conclusions and Future Research.** This study examined a production-distribution system with one supplier and one or more customers located in close proximity to each other (defined as a customer area). The goal was to optimize an objective function that considered the length of time required for the truck to complete delivery of the last batch to the customer(s) and return to the factory. We assumed a problem instance in which jobs were processed by either one of two parallel machines and delivered by a single truck to one customer area. In particular, this study addressed a situation in which every job occupied a different amount of storage space in the truck. We presented two simple heuristics (H1 and H3) as well as a GA-based algorithm for this coordinated scheduling problem. Computational tests showed that the GA-based algorithm was the optimal approach overall. However, there was no significant difference between the GA-based algorithm and H3 when the truck was a non-bottleneck. Thus, H3 is the optimal scheduling approach when transportation time is short or the truck capacity is large; otherwise, the GA-based algorithm is the scheduling approach of choice.

This study did not consider shipments that can serve more than one customer. Such a problem would include route decisions for each shipment, and new algorithms/heuristics would be required to solve such a problem. In future research, problems involving different performance measures, such as deadline-related criteria, could be considered. We are currently conducting research on these topics.

REFERENCES

[1] K. Sun and H. Li, Single-machine scheduling with an actual time-dependent learning consideration, *International Journal of Innovative Computing, Information and Control*, vol.5, no.10(A), pp.3189-3199, 2009.

[2] P. C. Chang, S. H. Chen and V. Mani, Parametric analysis of bi-criterion single machine scheduling with a learning effect, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.2033-2043, 2008.

[3] J.-J. Wang, F. Liu and P. He, Rescheduling under predictive disruption of WSPT schedule for single machine scheduling, *ICIC Express Letters*, vol.4, no.2, pp.467-472, 2010.

[4] G. Wang and T. C. E. Cheng, Parallel machine scheduling with batch delivery costs, *International Journal of Production Economics*, vol.68, pp.177-183, 2000.

[5] C.-Y. Lee and Z.-L. Chen, Machine scheduling with transportation considerations, *Journal of Scheduling*, vol.4, pp.3-24, 2001.

[6] Y.-C. Chang and C.-Y. Lee, Machine scheduling with job delivery coordination, *European Journal of Operational Research*, vol.158, pp.470-487, 2004.

[7] Y. He, W. Zhong and H. Gu, Improved algorithms for two single machine scheduling problems, *Theoretical Computer Science*, vol.363, pp.257-265, 2006.

[8] W. Zhong, G. Dósa and Z. Tan, On the machine scheduling problem with job delivery coordination, *European Journal of Operational Research*, vol.182, pp.1057-1072, 2007.

[9] L. Lu and J. Yuan, Single machine scheduling with job delivery to minimize makespan, *Asia-Pacific Journal of Operational Research*, vol.25, no.1, pp.1-10, 2008.

[10] Z.-L. Chen and G. L. Vairaktarakis, Integrated scheduling of production and distribution operations, *Management Science*, vol.51, no.4, pp.614-628, 2005.

[11] C.-L. Li, G. Vairaktarakis and C.-Y. Lee, Machine scheduling with deliveries to multiple customer locations, *European Journal of Operational Research*, vol.164, pp.39-51, 2005.

[12] M. Ji, Y. He and T. C. E. Cheng, Batch delivery scheduling with batch delivery cost on a single machine, *European Journal of Operational Research*, vol.176, pp.745-755, 2007.

[13] B. Chen and C.-Y. Lee, Logistics scheduling with batching and transportation, *European Journal of Operational Research*, vol.189, pp.871-876, 2008.

[14] D. R. Sule, *Industrial Scheduling*, PWS Publishing Company, 1996.

[15] J. Holland, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, MI, USA, 1975.