Engineering Optimization
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/geno20

An ant colony optimization heuristic for an integrated production and distribution scheduling problem

Yung-Chia Chang[a], Vincent C. Li[b] & Chia-Ju Chiang[a]

[a] Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan

[b] Department of Business Administration, National Chiayi University, Chiayi, Taiwan

Published online: 10 May 2013.

To cite this article: Yung-Chia Chang, Vincent C. Li & Chia-Ju Chiang (2014) An ant colony optimization heuristic for an integrated production and distribution scheduling problem, Engineering Optimization, 46:4, 503-520, DOI: 10.1080/0305215X.2013.786062

To link to this article: http://dx.doi.org/10.1080/0305215X.2013.786062

Engineering Optimization, 2014
Vol. 46, No. 4, 503–520, http://dx.doi.org/10.1080/0305215X.2013.786062

An ant colony optimization heuristic for an integrated production and distribution scheduling problem

Yung-Chia Chang[a], Vincent C. Li[b]* and Chia-Ju Chiang[a]

[a]*Department of Industrial Engineering and Management, National Chiao Tung University, Hsinchu, Taiwan;* [b]*Department of Business Administration, National Chiayi University, Chiayi, Taiwan*

(*Received 19 May 2012; final version received 1 March 2013*)

Make-to-order or direct-order business models that require close interaction between production and distribution activities have been adopted by many enterprises in order to be competitive in demanding markets. This article considers 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. This article presents a mathematical model for describing the problem and designs an algorithm using ant colony optimization. Computational experiments illustrate that the algorithm developed is capable of generating near-optimal solutions. The computational results also demonstrate the value of integrating production and distribution in the model for the studied problem.

Keywords: integrated scheduling; production and distribution operations; unrelated parallel machines; ant colony optimization

1. Introduction

In the current competitive global market, companies are forced to lower the amount of inventory needed across their supply chain but still have to be responsive to customers' requirements. Reduced inventory creates a closer interaction between production and distribution activities and thus increases the practical usefulness of integrated models (Sarmiento and Nagi 1999).

Consider a make-to-order business model where products are custom-made and are delivered to customers within a very short lead-time directly from the factory without the intermediate step of finished product inventory. For example, in a typical computer direct-order system, there are hundreds of configurations available for a customer to choose from when ordering a computer. It would be impractical and not cost-effective to store assembled and packaged computers with a particular configuration before knowing what customers will order. Many manufacturers are now operating under the concept of postponement to delay product differentiation until closer to the time the product is sold. As a result, the factory only keeps inventory at the component level and has to start computer assembly and packaging after receiving customer orders. However, owing to keen market competition, a business operating under this direct-sale model must deliver completed orders to customers within a very short time-frame. Therefore, production and

*Corresponding author. Email: vincentcli@mail.ncyu.edu.tw

distribution operations are linked together directly without any intermediate steps (Chen 2004). Similar characteristics can also be found in the service industry, especially for time-sensitive services.

Traditional approaches deal with production and distribution one by one with little or no coordination between these two stages. When there is a sufficient amount of inventory between production and distribution, the operations of these two stages can be decoupled; hence, traditional approaches are able to deliver reasonable and effective solutions. However, in make-to-order or direct-order business models, there is not much inventory between the production and distribution stages, so close interaction between these two stages is essential.

This article discusses a two-stage supply-chain problem that involves the integration of *production and distribution scheduling* at the individual job level. For such problems, jobs are, at the first stage, arranged to be processed by some machines and then delivered, at the second stage, to some customers who may reside at different locations by some means of transportation (*e.g.* delivery vehicles). Traditionally, the problem in each of these two stages is considered separately. The problems at the first and second stages are called the job scheduling problem (*e.g.* Pinedo 2002) and vehicle routing problem (Golden, Raghavan, and Wasil 2008), respectively. Both problems have been widely studied in the past. However, not until recently has attention been given to models that address the integration of production and distribution operations. One of the reasons that few studies attempt to address these two stages simultaneously is that the problem of a single stage by itself is already tremendously hard to solve.

Among the research that explicitly addresses the integration of production and distribution scheduling, Lee and Chen (2001) studied machine scheduling problems that considered either intermediate delivery between machines or the finished product deliveries. Li, Vairaktarakis, and Lee (2005) extended Lee and Chen's work by considering the situations where product may demand different amounts of storage space for delivery and where deliveries are made to multiple locations via a direct shipping method, respectively. These studies aimed at optimizing the customer service level measured by job delivery times subject to a delivery vehicle availability constraint. Averbakh (2010) studied the on-line integrated production–distribution problems with capacitated batches and aimed to minimize the sum of weighted flow time and delivery cost. There are models that consider the trade-off between the total transportation cost and customer service performance measures related to due date or job delivery time (Chen and Vairaktarakis 2005; Hall and Potts 2005; Pundoor and Chen 2005; Wang and Lee 2005; Chen and Pundoor 2006; Li and Vairaktarakis 2007). In terms of delivery methods, most of the existing literature considered the problems with batch delivery to customers by the direct shipping method (Lee and Chen 2001; Hall and Potts 2003; Hall and Potts 2005; Li and Ou 2005; Li and Ou 2007; Zhong, Dosa, and Tan 2007; Chen and Lee 2008), but only a few studies discussed routing decisions (Chen and Vairaktarakis 2005; Li, Vairaktarakis, and Lee 2005; Armstrong, Gao, and Lei 2008; Geismar *et al.* 2008).

Most of the special cases of this problem class have already been proved as NP hard, especially when there are multiple machines for production and/or when there are multiple customer locations (Lee and Chen 2001; Chang and Lee 2004; Li *et al.* 2005; Pundoor and Chen 2005). As pointed out by Chen and Vairaktarakis (2005), more research is needed to model production–distribution interactions and develop problem-solving techniques that can be used in practice. Chen (2010) has recently provided an extensive review on this problem class.

In this study, jobs are first arranged to be processed by a manufacturing facility modelled as a shop with a set of unrelated parallel machines (Lenstra and Shmoys 1990) (*i.e.* there is no relationship between the processing times of a job on different machines). After production, jobs are delivered to customers residing at different locations by homogeneous capacitated vehicles. The objective is to find a joint production and distribution schedule so that the overall cost is minimized. This article presents a mathematical model for this integrated production and distribution scheduling problem. Since this problem is NP hard, a heuristic is designed using ant colony optimization (ACO) to find near-optimal solutions of the studied problem.

The remainder of this article is organized as follows. Section 2 defines the studied problem and gives a mixed integer programming formulation. Section 3 presents the design of the ACO heuristic. Computational experiments are discussed in Section 4 along with the analysis of results. Section 5 concludes this article with future research directions.

2. Problem definition and mathematical formulation

The integrated production and distribution scheduling problem is described as follows. There are n jobs $N = \{1, 2, \ldots, n\}$ ordered by q customers $Q = \{1, 2, \ldots, q\}$ residing at different locations within an underlying transportation network. Let $N_h \subset N$ be the subset of jobs ordered by customer h, and $n_h = |N_h|$ for $h \in Q$, where $N = N_1 \cup \ldots \cup N_q$ and $n = n_1 + \cdots + n_q$. The weight of job $j \in N$ is denoted as w_j, which represents the relative importance of job j among others. Jobs are first processed in a manufacturing system and then delivered to their respective customers.

In the production part, m unrelated parallel machines $M = \{1, 2, \ldots, m\}$ are available in the manufacturing system. Each job needs to be processed by one of the machines without interruption. The processing time of job j on machine k is p_{kj}. All jobs and machines are available at time 0. Given a production schedule, completion time C_j represents the time when job j is completed by one of the machines. Clearly, job j cannot be picked up for delivery until time C_j.

In the distribution part, a set of v vehicles ($v \in V = \{1, 2, \ldots, v\}, v \geq n$) of the same capacity are available to deliver jobs. Each vehicle is limited to carry up to c ($c < n$) jobs in one delivery and is initially located at the manufacturing facility. The transportation cost incurred by each delivery consists of a fixed cost f and a heterogeneous variable cost depending on the particular route taken by the vehicle. Parameters b_{0i}, b_{ij} and b_{j0} denote the variable cost for travelling from the manufacturing facility to customer i, from customer i to customer j, and from customer j to the manufacturing facility, respectively. The corresponding travel times are denoted as t_{0i}, t_{ij} and t_{j0}, respectively. A delivery vehicle can depart from the manufacturing facility only when each of the jobs allocated to the corresponding batch has been processed by some machine. The delivery time of job $j \in N$, D_j, is defined as *the time when job* j *is delivered to its customer*.

Similar to Chen and Vairaktarakis (2005), the objective function considers both customer service and total distribution cost, where the former is captured by the weighted sum of the job completion delivery times of job $j \in N$ (*i.e.* $\Sigma_{j \in N} w_j D_j$, denoted as $\Sigma\, wD$) and the latter is the sum of distribution costs (denoted as T). Therefore, the goal is to minimize $\theta \Sigma wD + (1 - \theta)T$, where θ is a user-defined parameter between 0 and 1, which reflects the decision-maker's relative preference on the customer service and the total distribution cost.

To introduce the formulation, more notations are defined as follows:

d_0^u: the time when vehicle $u \in V$ departs from the manufacturing facility
d_h^u: the time when vehicle $u \in V$ arrives at customer $h \in Q$.

The decision variables are:

$x_{ij}^k = 1$ if job $i \in N$ is processed *immediately* before job $j \in N \backslash \{i\}$ on machine $k \in M$
$y_{hl}^u = 1$ if vehicle $u \in V$ travels from customer $h \in Q$ to customer $l \in Q \backslash \{h\}$ (*i.e.* they are visited consecutively by the same vehicle)
$z_i^u = 1$ if job $i \in N$ is delivered by vehicle $u \in V$.

For convenience, two dummy jobs are introduced, job 0 and job $n + 1$, which do not take any time to be processed or consume any vehicle capacity, and C_0, the completion time of job 0, is set to 0. Moreover, let customer 0 represent the location of the manufacturing facility where each vehicle is initially located, and customer $q+1$ represents the location in the manufacturing facility where each vehicle will return to. There is no travel time or cost between customer 0 and customer $q+1$. The problem formulation is given as follows.

$$\min \quad \theta \sum_{j\in N} w_j D_j + (1-\theta)\left(f \sum_{u\in V}\sum_{h\in Q} y_{0h}^u + \sum_{u\in V} \sum_{\substack{h,l\in Q\cup\{0,q+1\}\\ h\neq l}} b_{hl} y_{hl}^u \right) \tag{1}$$

subject to

$$\sum_{\substack{i\in N\cup\{0\}\\ i\neq j}} \sum_{k\in M} x_{ij}^k = 1, \quad \forall j \in N\cup\{n+1\} \tag{2}$$

$$\sum_{j\in N} x_{0j}^k \leq 1, \quad \forall k\in M \tag{3}$$

$$C_j = \sum_{k\in M}\left(\sum_{\substack{i\in N\cup\{0\}\\ i\neq j}} p_{ki} x_{ij}^k + p_{kj} \right), \quad \forall j\in N\cup\{n+1\} \tag{4}$$

$$\sum_{l\in Q\cup\{q+1\}\backslash\{h\}} y_{hl}^u - \sum_{l\in Q\cup\{0\}\backslash\{h\}} y_{lh}^u = 0, \quad \forall u\in V\ h\in Q \tag{5}$$

$$\sum_{h\in Q} y_{h,q+1}^u - \sum_{h\in Q} y_{0,h}^u = 0, \quad \forall u\in V \tag{6}$$

$$\sum_{h\in Q} y_{0h}^u \leq 1, \quad \forall u\in V \tag{7}$$

$$n_h \sum_{l\in Q\backslash\{h\}} y_{lh}^u - \sum_{j\in N_h} z_j^u \geq 0, \quad \forall u\in V,\ h\in Q \tag{8}$$

$$\sum_{u\in V} z_j^u = 1, \quad \forall j\in N \tag{9}$$

$$\sum_{j\in Q} z_j^u \leq c, \quad \forall u\in V \tag{10}$$

$$d_h^u = \sum_{l\in Q\cup\{0\}\backslash\{h\}} (d_l^u + t_{lh}) y_{lh}^u, \quad \forall u\in V,\ h\in Q \tag{11}$$

$$C_j - \sum_{u\in V} d_0^u z_j^u \leq 0, \quad \forall j\in N \tag{12}$$

$$D_j - \sum_{u\in V} d_h^u z_j^u = 0, \quad \forall j\in N_h,\ h\in Q \tag{13}$$

$$x_{ij}^k \in \{0,1\}, \quad \forall i\in N\cup\{0\},\ j\in N\cup\{n+1\},\ k\in M \tag{14}$$

$$e_{ij}^k \in \{0,1\}, \quad \forall i\in N\cup\{0\},\ j\in N\cup\{n+1\},\ i\neq j,\ k\in M \tag{15}$$

$$y_{hl}^u \in \{0,1\}, \quad \forall h\in Q\cup\{0\},\ l\in Q\cup\{q+1\},\ u\in V \tag{16}$$

$$z_j^u \in \{0,1\}, \quad \forall j\in N,\ u\in V \tag{17}$$

Equation (1) minimizes the weighted sum of the total weighted job delivery time and total distribution cost. Equation (2) ensures that each job is processed exactly once. Equation (3) ensures that each machine can have at most one job as the first job to process. Equation (4) defines the time when each job is completely processed by a machine and is ready for delivery. Equations (5) and (6) represent the conservation-flow requirements. Equation (7) ensures that each vehicle is used at most once. Equation (8) makes sure that if a job is delivered to its customer by a vehicle, then the vehicle has to visit that customer. Equation (9) guarantees that each job will be delivered to its customer exactly once. Equation (10) associates with the vehicle capacity. Equations (11) and (13) define the job delivery time. Equation (12) makes sure that a job will not be delivered until it has been processed by a machine. The binary restriction for some of the decision variables is shown in Equations (14) to (17).

Clearly, this formulation is very complicated. Equation (11) even incorporates nonlinear constraints that make this problem intractable. It is easy to see that this problem is NP hard in the strong sense since a reduction from the travelling salesman problem (TSP), whose complexity has been clarified as NP hard in the strong sense (Garey and Johnson 1979), can be taken as a special case of this problem with $m = 1$ and $p_j = 0$, $\forall j \in N$. It is unlikely that a polynomial algorithm can be found for this problem; hence, there is a need to explore methodologies to ease the computational effort while still obtaining good-quality solutions.

3. Research methodology

Owing to its complexity, the problem class of this study appears to be a good candidate for the ACO methods. ACO is a population-based metaheuristic approach proposed by Colorni, Dorigo, and Maniezzo (1991) and refined by Dorigo (1992), and Dorigo and Di Caro (1999). Dorigo and Stützle (2004) revised the original algorithm and added some new features for consideration when applying ACO. ACO has been successfully applied to many combinatorial optimization problems, including but not limited to the TSP (Dorigo and Gambardella 1997; Liu 2005), vehicle routing problems (VRP) (*e.g.* Bullnheimer, Hartl, and Strauss 1999), scheduling problems (Stützle *et al.* 1998; Gagné, Price, and Gravel 2002; T'kindt *et al.* 2002; Shyu, Lin, and Yin 2004; Huang and Liao 2008) and multi-objective optimization (Afshar, Sharifi, and Jalali 2009). Dorigo, Birattari, and Stützle (2006) also showed the most successful applications of ACO. Gagné, Price, and Gravel (2002) reported that ACO has a certain advantage for larger problems. Therefore, in this article, an ACO heuristic is designed to obtain near-optimal solutions for the studied problem. Recent research trends in ACO can be found in Dorigo and Stützle (2010).

Per Chen and Vairaktarakis (2005), this problem has the following straightforward optimality properties.

LEMMA 1 *There exists an optimal schedule to the studied problem such that*

(1) *there is no idle time between the jobs processed on each machine in the manufacturing facility in the production part of the schedule*; *and*
(2) *the departure time of each batch is the production finishing time of the last job included in the batch.*

3.1. *Framework of the proposed ACO algorithm*

ACO is a probabilistic search procedure whose central component is the pheromone model, which is employed to probabilistically sample the search space. In this study, a number of ants probabilistically construct solutions to the problem according to the designed pheromone model

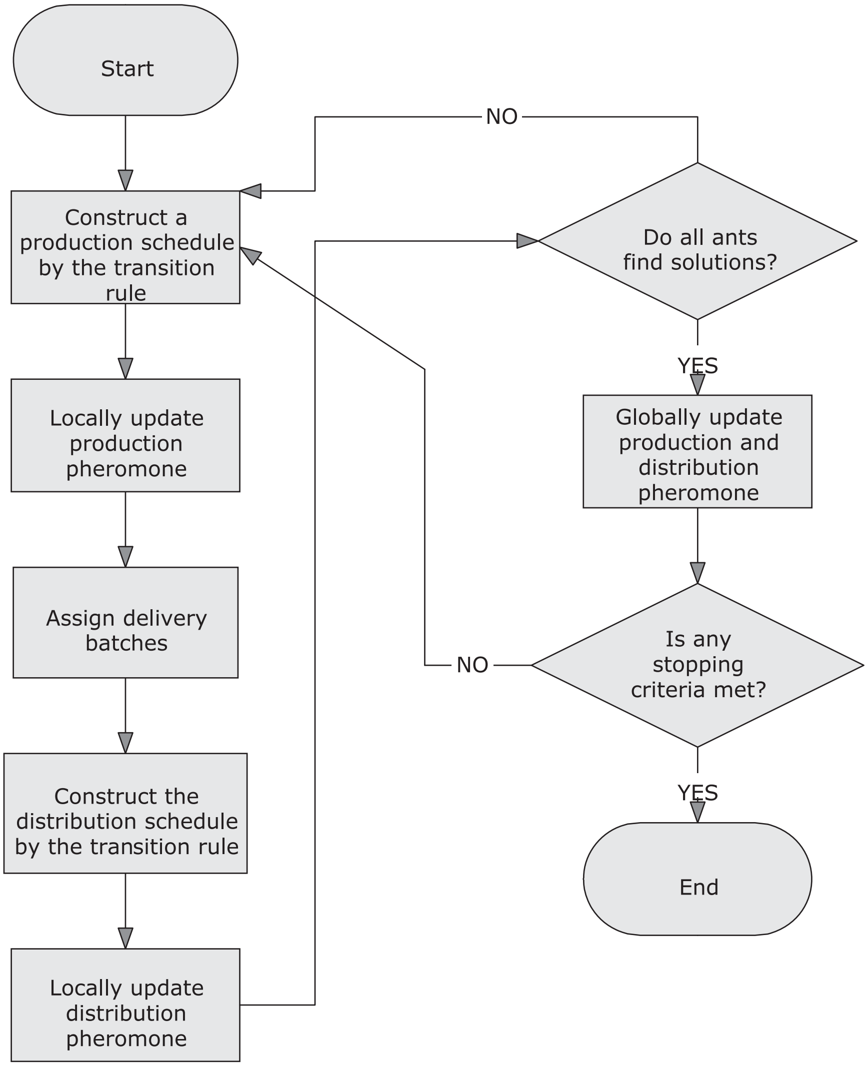

Figure 1. Framework of the proposed ant colony optimization algorithm.

at each iteration. Before the next iteration starts, some of the solutions are used for performing a pheromone update. A pheromone matrix for production and a pheromone matrix for distribution are designed. The framework of the proposed ACO algorithm is depicted in Figure 1. In this framework, a production schedule is constructed using the transition rules that also take into account the required delivery times incurred at the distribution stage. After updating locally the production pheromone matrix, a delivery schedule is constructed according to the previously constructed production schedule. The distribution pheromone matrix is locally updated after the distribution schedule is constructed. The same procedure repeats until all ants find their feasible solutions. Both production and distribution pheromone matrices are updated afterwards to reward the best-found trail (solution). The ACO algorithm is terminated when the predetermined maximum number of iterations is reached. Section 3.2 describes the details of the proposed ACO algorithm.

Table 1. Job–machine mapping for a three-job–two-machine problem.

Machine	1	1	1	2	2	2
Job	1	2	3	1	2	3
Operation number	1	2	3	4	5	6

3.2. *Construction of schedules*

There are three steps for constructing the path of ants. The production schedule is formed first, then the delivery batches and sequence are arranged accordingly, and finally the route of each vehicle is determined.

3.2.1. *Transformation of job and machine relationship for production*

For an unrelated parallel machine problem, each job can only be processed by one of the machines while the job processing time is machine dependent. To construct a production schedule, the job–machine assignment is transferred as a set of alternatives. An alternative represents a mapped relationship between a pair comprising a job and a machine. Taking a three-job–two-machine problem, for example, there are six alternatives resulting from this transformation (Table 1).

Through this transformation, both machines and jobs are taken into account to be used for constructing the production pheromone matrix. Although there may be multiple alternatives associated with a job, only one of the alternatives can be selected for each job. A feasible production schedule is then represented by the n selected alternatives out of nm choices of alternatives. If each alternative is referred as a node, then constructing a feasible production schedule is equivalent to finding a path consisting of n nodes from the underlying network consisting of nm nodes. In what follows, the terms 'node' and 'alternative' are used interchangeably. For the same three-job–two-machine example described earlier, the corresponding production pheromone matrix has six rows and six columns. Let τ_{ij}^{p} be the (i,j) entry of the production pheromone matrix, where τ_{ij}^{p} represents the intensity of the pheromone trail associated with the path from node i to node j for the production schedule. Let $\Omega(i)$ be the job corresponding to alternative i. Then τ_{ij}^{p} equals 0 if $\Omega(i) = \Omega(j)$. For instance, in the three-job–two-machine example, both τ_{14}^{p} and τ_{41}^{p} equal 0 since both alternatives 1 and 4 are related to job 1. Suppose an ant k has just chosen node i, then $P^k(i)$ is the set of nodes already visited by ant k and $E^k(i)$ is the set of nodes eligible to be visited for ant k from node i. In this three-job–two-machine example, suppose node 4 is chosen first, then $P^k(4) = \{4\}$. As alternative 4 has been selected, alternative 1, the other alternative associated with job 1, is no longer eligible to be selected. Consequently, $E^k(4) = \{2, 3, 5, 6\}$. If node 6 is selected next, then $P^k(6) = \{4, 6\}$ and $E^k(6) = \{2, 5\}$. At this moment, alternatives 4 and 6 have been included in the solution, and only alternatives 2 and 5 designated for job 2 can be visited next.

3.2.2. *State transition rule for production*

To search for better solutions, ants may rely on their past experience that leads to a better trail, and they also need to find other trials that have not been explored. These two strategies are referred to as 'exploitation' and 'exploration', respectively. The main purpose of exploitation is to improve the currently found solutions. However, this strategy may cause the search to be trapped into a local optimal solution. Therefore, exploration provides ants with opportunities to escape from local optima and to search better solutions in a new area. The state transition rule proposed by Dorigo and Gambardella (1997) is employed to determine the next-visit node.

Let η^p_{iu} be the heuristic information of node u from node i for ant k for the production schedule. Moreover, let $s(j)$ be the customer who ordered job j. The heuristic information is determined by Equation (18):

$$\eta^p_{iu} = \left(\sum_{x \in P^k(i)} w_{\Omega(x)} C_{\Omega(x)} + w_{\Omega(u)} C_{\Omega(u)} + t_{s(\Omega(i))s(\Omega(u))} \right)^{-1}, \quad u \in E^k(i) \tag{18}$$

In Equation (18), the current position of ant k is node i, and node u is a node eligible to be visited from node i. This equation encourages the ant to select an alternative such that its corresponding job has smaller weighted completion time and shorter delivery time. The decision on which node to visit next depends on the preference between exploration and exploitation. In the interval [0, 1], let π be a random number generated from the uniform distribution and π_p be a constant which indicates the desired probability of exploitation in the production scheduling. If $\pi \le \pi_p$, then the next node the ant visits is determined by Equation (19), where α_p and β_p are two parameters chosen to represent the relative importance of the pheromone value and the heuristic information, respectively. If Equation (19) is applied, ant k uses exploitation to determine the next node to visit.

$$j = \arg\max_{u \in E^k(i)} \{(\tau^p_{iu})^{\alpha_p} (\eta^p_{iu})^{\beta_p}\}, \quad \text{if } \pi \le \pi_p \tag{19}$$

On the other hand, if $\pi > \pi_p$, then ant k is allowed to explore a new trail, as a strategy of exploration. The probability of selecting alternative u among all eligible alternatives is determined by Equation (20).

$$P_{iu} = \frac{(\tau^p_{iu})^{\alpha_p} (\eta^p_{iu})^{\beta_p}}{\sum_{x \in E^k(i)} (\tau^p_{ix})^{\alpha_p} (\eta^p_{ix})^{\beta_p}}, \quad \text{if } \pi > \pi_p \tag{20}$$

3.2.3. *Allocation of distribution batches*

The next stage of the heuristic is to determine which jobs are to be included in a batch and the time at which when the vehicle carrying the batch of jobs leaves the production facility. Routing for a batch is required if the jobs in the batch belong to different customers. This routing subproblem is in fact a TSP, a well-known NP-hard problem.

To simplify the work required in this stage, a dynamic programming algorithm, called DPA, is designed to assign jobs into batches according to their job completion times as well as the estimated transportation cost. The jobs are first arranged in non-decreasing order of the job completion times. This sequence is called the first-in–first-out (FIFO) sequence. Let $j_{(i)}$ be the job that is finished on the ith position ($i = 1, 2, \ldots, n$) in the FIFO sequence. The jobs are then allocated to vehicles accordingly. To speed up path construction, the routing decision is avoided in DPA by assuming that jobs are also delivered per the FIFO sequence. The actual routing is determined when constructing the distribution schedule (described in Section 3.2.4). Let $V(j_{(i)})$ be the total weighted sum of the production and distribution cost (*i.e.* the objective function value) for delivering the first i ($i = 1, 2, \ldots, n$) jobs per the FIFO sequence. DPA is described as follows:

The initial condition:

$$V(j_{(i)}) = \begin{cases} 0 & \text{if } i = 0 \\ \infty & \text{otherwise} \end{cases} \tag{21}$$

Recursive relationship:

$$V(j_{(i)}) = \min \left\{ \begin{array}{l} V(j_{(i-h)}) + \theta \left(hC_{j_{(i)}} + t_{0s(\Omega(j_{(i-h+1)}))} + \sum_{x=i-h+1}^{i-1} t_{s(\Omega(j_{(x)}))s(\Omega(j_{(x+1)}))} \right) \\ \quad + (1-\theta)\left(f + b_{0s(\Omega(j_{(i-h+1)}))} + \sum_{x=i-h+1}^{i-1} b_{s(\Omega(j_{(x)}))s(\Omega(j_{(x+1)}))} \right) \\ \quad : h = 1, \ldots, \min(c, j_{(i)}) \end{array} \right\}$$
$$\text{for } i = 1, \ldots, n. \tag{22}$$

The optimal objective value $= V(j_{(n)})$.

The initial condition is stated in Equation (21). In Equation (22), h represents the number of jobs delivered with $j_{(i)}$ in one batch. The goal is to select a number of jobs to carry along with $j_{(i)}$ to minimize $V(j_{(i)})$. The cost terms shown in Equation (22) represent the additional cost of carrying these h jobs along with $j_{(i)}$ while assuming that jobs are also delivered according to the FIFO sequence. The departure time of the corresponding vehicle is the completion time of $j_{(i)}$ by Lemma 1(2).

3.2.4. *State transition rule for distribution*

The state transition rule of the distribution schedule is similar to the rule of the production schedule. After allocating all the jobs to some vehicles, the route of each vehicle is determined. Let a be the current node that ant k is visiting and $S_{kv}(a)$ be the set of nodes that still need to be visited by the vth vehicle of ant k. If there is another undelivered job ordered by the same customer of job a, then ant k will visit the job first. Otherwise, the next job will be selected according to the preference between exploitation and exploration. Let τ^d_{ab} be the value corresponding to the (a, b) entry in the distribution pheromone matrix and η^d_{ab} be heuristic information of node b from node a. The heuristic information is determined by Equation (23).

$$\eta^d_{ab} = \begin{cases} \dfrac{w_b}{d_{ab}}, & \text{for } d_{ab} > 0 \\ \infty, & \text{otherwise} \end{cases} \tag{23}$$

In Equation (23), w_b is the weight of job b, and d_{ab} is the transportation time between the customers of jobs a and b. If both jobs are for the same customer, the transportation time is assumed to be 0. Similar to the notation used in Section 3.2.2, in the interval $[0, 1]$, let π be a random number generated from the uniform distribution and π_d be a constant which indicates the desired probability of exploitation in the distribution part. If $\pi \leq \pi_d$, then the next job ant k delivers is determined by Equation (24).

$$b = \arg\max_{u \in S_{kv}(a)} \{(\tau^d_{au})^{\alpha_d} (\eta^d_{au})^{\beta_d}\}, \quad \text{if } \pi \leq \pi_d \tag{24}$$

In Equation (24), α_d and β_d are two parameters chosen to represent the relative importance of the pheromone value and the heuristic information, respectively. On the other hand, if $\pi > \pi_d$, ant k is allowed to explore new trails. The probability of node x being selected at node a is determined by Equation (25).

$$P_{ax} = \frac{(\tau_{ax})^{\alpha} (\eta_{ax})^{\beta}}{\sum_{u \in S_k(a)} (\tau_{au})^{\alpha} (\eta_{au})^{\beta}} \quad \text{for } x \in S_k(a), \text{ if } \pi > \pi_d \tag{25}$$

3.2.5. *Pheromone update*

Pheromone, the medium that ants use to exchange information and communicate with each other, needs to be updated according to the moves of ants and solution quality. The pheromone is updated locally and globally. During the construction of a path, the pheromone can be updated locally using Equation (26) right after a node (an alternative in the production matrix or a job in the distribution matrix) j has been chosen by the ant k from node i.

$$\tau_{ij}^{\text{new}} = (1-\varphi)\tau_{ij}^{\text{old}} + \varphi\tau_0, \quad \forall i,j \tag{26}$$

In Equation (26), τ_{ij}^{old} and τ_{ij}^{new} are the current and updated pheromone values corresponding to the (i,j) entry of the production pheromone matrix, respectively, where φ is the predetermined evaporation rate taking value between 0 and 1, and τ_0 is the predetermined initial pheromone value. The effect of adjusting the pheromone level from node i to node j in Equation (26) is to increase the chance for other ants to visit different paths. Moreover, pheromone density is confined within a range between τ_{min} and τ_{max} to avoid a very high or low pheromone density so as to increase diversification.

In this article, two pheromone matrices are designed, one for production and the other for distribution. The size of the production pheromone matrix is nm by nm while that the size of the distribution pheromone matrix is n by n. These two matrices seem independent, but a feedback mechanism of pheromone intensity is designed to integrate these two matrices. That is, when the distribution pheromone matrix is updated, the production pheromone matrix is revised for the corresponding alternatives of each job. The goal of interaction between these two matrices is to generate some influence on the production stage from the distribution stage so as to enhance the integration of these two stages.

After all ants have constructed their feasible schedules in an iteration, both production and distribution pheromone matrices are globally updated using Equation (27) in order to intensify the better-found solutions and to lead ants to explore or exploit trails in later iterations.

$$\tau_{ij}^{t+1} = (1-\rho)\tau_{ij}^{t} + \rho\Delta\tau_{ij}^{*} \tag{27}$$

In Equation (27), t refers to the current completed iteration, and ρ represents the predetermined global pheromone evaporation rate. The symbols τ_{ij}^{t} and τ_{ij}^{t+1} represent the (i,j) entry of the pheromone matrix in iterations t and $t+1$, respectively. The increase in pheromone value of path (i,j) as a result of the best-found solution is denoted by $\Delta\tau_{ij}^{*}$ and is determined by Equation (28).

$$\Delta\tau_{ij}^{*} = \begin{cases} \dfrac{L_{gb}}{N_{gb}}\tau_0, & \text{if path}(i,j) \in \text{the best solution found in iteration } t \\ 0, & \text{otherwise} \end{cases} \tag{28}$$

In Equation (28), L_{gb} is the objective value of the best solution found up to the tth iteration, and N_{gb} stands for the best objective value found in the current iteration. The reward on pheromone decreases with increasing value of N_{gb}.

3.3. *Summary of ACO heuristic*

The heuristic of this article is developed from ACO and consists of path construction and pheromone update. The construction process involves three steps. First, the production schedule is constructed by assigning each job to some machine and determining the job order of production. The initial solution is randomly generated per the method described in Section 3.2.1. Then, the distribution batches are allocated by following the order that jobs are finished in the first step.

Dynamic programming is employed to determine the delivery batches. Finally, the distribution sequence within each vehicle is determined.

With this framework, the procedure of the heuristic (referred to as ACO_DPA) is summarized as follows.

Step 1: Initializing parameters

The following parameters are determined in this step: the number of ants (*n_ants*), the number of iterations (*iter*), the initial value of pheromone intensity of each path (τ_0), the weight of pheromone intensity (α_p, α_d), the weight of heuristic information (β_p, β_d), the probabilities of exploitation degree (π_p, π_d), and local and global pheromone evaporation rates (φ, ρ).

Step 2: Construction of production schedule

After transforming the mapping between jobs and machines into a series of alternatives, construct the path for each ant according to the preference of exploitation and exploration. The construction continues until the completion of a path.

Step 3: Local pheromone update for production

Once a new production operation is scheduled following the current operation, update the pheromone intensity from the current operation to the new one.

Step 4: Allocation of distribution batches

After the production schedule has been completely constructed, determine the total number of batches required and associate jobs with each batch.

Step 5: Planning of distribution for each batch of jobs

Within each batch of jobs set in step 4, determine the sequence of delivery. The delivery starts from the manufacturing facility, and visits customers following the preference of exploitation and exploration. Jobs in a batch that belongs to the same customer will be delivered at the same time.

Step 6: Local pheromone update for distribution

Once a new job has been selected for delivery, update the pheromone intensity from the current job to the new one. This step also triggers the pheromone feedback mechanism, which updates the production pheromone matrix as well.

Step 7: Global pheromone update

Once all the ants have finished the construction of the entire path, the pheromone of the path with the minimal objective function value is increased while the pheromone of the other paths is reduced from pheromone evaporation.

Step 8: Stopping criterion

If the number of iterations has reached the allowed maximum, stop; otherwise go back to step 2.

4. Computational experiments

This section demonstrates the computational results for a set of problem instances tested on an AMD Dual-Core Opteron 2218 CPU with 2 GB of RAM. The ACO_DPA algorithm is coded in Matlab.

4.1. *Design of test problem instances*

To test the effectiveness and efficiency of ACO_DPA, 162 problem instances were generated. These instances are distinguished by the number of machines ($m = 2, 4$ or 8), number of customers ($q = 5$, 10 or 20), number of jobs ($n = 25$, 50 or 100), vehicle capacity ($c = 5$, 10 or 20) and objective relative preference ($\theta = 0.2$, 0.5 or 0.8). Note that not all the above combinations are used. When $q = 10$, n takes only the value of 50 or 100; whereas when $q = 20$, n takes only the value of 100. In the production stage, jobs are distinguished by their processing times. The length of processing times can be long, average and short for large, medium and small jobs, respectively. Denote $U(x_1, x_2)$ as the uniform distribution between x_1 and x_2. The processing times of large, medium and small jobs were randomly created from $U(30, 35)$, $U(15, 20)$ and $U(1, 5)$, respectively. A fixed cost of 100 for each vehicle being used was assumed for the distribution stage. The x-coordinate and y-coordinate of each customer's location were both generated from $U(-30, 30)$ and the manufacturing facility was located at the origin. The distribution time and cost were calculated according to the Euclidean distance between the locations of each pair of customers.

4.2. *Tuning of parameters*

Preliminary experiments were conducted to find the appropriate value for each of the parameters. The following values have either been superior or achieved the best compromise between solution quality and computational time, and were thus used for all further experiments in this study: $n\_ants = 40$, $iter = 80$, $\tau_0 = 0.0005$, $(\alpha_p, \beta_p) = (2, 0.1)$, $(\alpha_d, \beta_d) = (0.05, 0.1)$, $\pi_p = 0.95$, $\pi_d = 0.6$, $\varphi = 0.7$ and $\rho = 0.005$.

4.3. *Effectiveness of ACO_DPA*

Chang, Chang, and Chang (2013) used column generation to solve a special case of the studied problem where all the parallel machines are identical. To test the solution quality, the best solutions obtained by ACO_DPA were compared with the results obtained using the exact method. Owing to memory restrictions on the personal computer platform, the exact method designed by Chang *et al.* was capable of finding optimal solutions to small-scale problem instances only. Therefore, the problem instances tested here are all on a small scale: limited to three or four customers, vehicle capacity of four jobs, and the number of machines being two, four or six. The job processing times, distribution times and costs were all randomly generated from $U(1, 10)$. The same objective function as Equation (1) is used in both methods where the total production cost and total distribution cost are equally preferred (*i.e.* $\theta = 0.5$). Let Z^* and Z_{DPA} be the best objective function value obtained by the exact method and ACO_DPA, respectively. The gap between these two methods is calculated using $(Z_{\text{DPA}} - Z^*)/Z_{\text{DPA}}$. Table 2 lists the solutions as well as the computer run-time required by ACO_DPA and the exact method, respectively. As can

Table 2. Gap between Z_{DPA} and Z^* (obtained from column generation).

No.	q	m	n	Obj. by ACO_DPA	Optimal obj. value	Gap on obj.	Run time of ACO_DPA	Run time of exact method
1	3	2	7	84.0	84.	0	1.1	92.4
2	3	2	9	122.5	122.5	0	1.2	1273.8
3	3	4	6	59.5	59.5	0	2.8	10.2
4	3	4	8	77.5	77.5	0	3.0	13.1
5	3	4	10	90.5	90.5	0	3.5	133.4
6	3	6	8	74.5	70.5	5.7%	5.8	9.7
7	3	6	10	92.5	91.0	1.6%	6.4	13.5
8	3	6	12	80.0	78.0	2.6%	6.9	50.3
9	4	2	7	76.0	76.0	0	1.1	50.3
10	4	4	8	63.0	63.0	0	3.2	18.7
11	4	4	9	94.5	91.0	3.8%	3.2	302.2
12	4	6	10	77.0	72.0	6.9%	6.3	49.1
13	4	6	10	95.0	92.5	2.7%	6.6	345.8
14	4	6	10	95.0	93.0	2.2%	7.0	558.2

Note: CPU time measured in seconds.

be seen, ACO_DPA obtains the optimal solution in seven out of 14 problem instances, while the average gap on the objective function values is 2.32%. These results show that ACO_DPA has the potential to obtain very good results in just a fraction of time in comparison to the exact approach. The price of a complete enumeration of this problem is pretty expensive. There are already $m^n n!$ possible schedules for the production part. Incorporating the distribution part makes the problem even more difficult to solve.

4.4. *Results of experiments*

The effectiveness of ACO_DPA is evaluated from several different perspectives as discussed in the following subsections.

4.4.1. *Effectiveness of using DPA*

The purpose of designing the DPA between the production and distribution in the ACO heuristic is to allocate finished jobs to batches before distribution. To evaluate the effectiveness of DPA, it was compared with another alternative called the fully loaded strategy. In the fully loaded strategy, jobs are scheduled to be distributed per their completion times. Except for the last one, all vehicles have to wait until they are fully loaded. Let Z_F denote the best objective function value obtained by the fully loaded strategy. The gap between Z_F and Z_{DPA} is calculated using $(Z_F - Z_{\text{DPA}})/Z_F$. A multiple regression analysis for the foregoing value was conducted by considering m, q, n, c and θ as the independent variables. The results of the multiple regression analysis are summarized in Table 3.

Table 3 shows an insignificant impact of m or q on the relevant differences. According to this observation, the results were consolidated by taking the average value across different m and q values, respectively. The average gaps between Z_F and Z_{DPA} are summarized in Table 4, from which the following two observations are made:

(1) The average gap increases as θ increases because larger value of θ makes the job delivery time contribute more to the objective function value. If a vehicle has to wait until it is fully loaded, the delivery times of some of the jobs will be increased owing to extra waiting time.
(2) The larger the vehicle capacity, the more effective the DPA, in general. When the capacity is doubled from 5 to 10 and from 10 to 20, the average gap increases to 1.89 (= 20.47%/10.84%) and 1.82 (= 37.28%/20.47%) times the original value, respectively. As the capacity increases, there are more choices to combine jobs for distribution. Consequently, the batching decision is more important when the vehicle capacity is large.

Table 3. Multiple regression analysis for the gap between Z_F and Z_{DPA}.

	Regression summary for dependent variable: $R = 0.94319306$, $R^2 = 0.88961315$, Adjusted $R^2 = 0.88607511$ $F(5,156) = 251.44$, $p < 0.0000$, Std. Error of estimate: 0.04728					
$N = 162$	Beta	Std. Err. of Beta	B	Std. Err. of B	$T(156)$	p
Intercept			−0.052884	0.015780	−3.35134	0.0010009
M	−0.026795	0.026601	−0.001500	0.001489	−1.00731	0.315347
Q	0.002466	0.030583	0.000065	0.000800	0.08065	0.935826
N	−0.207404	0.030583	−0.000955	0.000141	−6.78174	0.000000
C	0.775728	0.026601	0.017372	0.000596	29.16173	0.000000
θ	0.494594	0.026601	0.281991	0.015166	18.59313	0.000000

Table 4. Average gaps between Z_F and Z_{DPA}.

	$\theta = 0.2$			$\theta = 0.5$			$\theta = 0.8$			
c	$n = 25$ (%)	$n = 50$ (%)	$n = 100$ (%)	$n = 25$ (%)	$n = 50$ (%)	$n = 100$ (%)	$n = 25$ (%)	$n = 50$ (%)	$n = 100$ (%)	Average
5	9.70	6.00	4.30	11.80	10.00	7.20	21.90	16.30	10.40	10.84
10	9.70	12.80	10.80	23.80	22.10	15.40	34.40	31.90	23.30	20.47
20	23.60	23.80	21.10	42.70	39.30	34.40	56.80	50.50	43.30	37.28

Table 5. Comparison of run time between dynamic programming algorithm (DPA) and fully loaded strategy.

n	c	T_{DPA}	T_F	Gap (%)
25	5	27.84	17.38	38
25	10	35.23	22.47	36
25	20	48.21	30.17	37
50	5	110.44	71.01	36
50	10	126.79	81.72	36
50	20	172.78	109.87	36
100	5	514.93	363.29	29
100	10	551.48	387.57	30
100	20	660.48	445.84	32

Note: CPU time measured in seconds.

In summary, the impact of including DPA is significant as the average gap between Z_{DPA} and Z_F is 22.9% for these 162 problem instances. This result indicates that the fully loaded strategy cannot yield better solutions than loading vehicles partially, even though the former strategy minimizes the number of vehicles used, thus justifying the value of using DPA in the ACO heuristic.

To illustrate the run-time performance of ACO_DPA as well as the difference in run-time between using DPA and using the fully loaded strategy in ACO, the run-time information is organized in Table 5. Let T_{DPA} and T_F be the run-time of using DPA and the fully loaded strategy in ACO, respectively. The gap between T_{DPA} and T_F is calculated as $(T_{\text{DPA}} - T_F)/T_{\text{DPA}}$. The results in Table 5 were consolidated by taking the average value across different θ, m and q values since these three parameters do not affect the run-time gap significantly. As seen in Table 5, DPA spends 30% more run-time than the fully loaded strategy does, but the improvement of solution quality is significant. Take the set of problems with $n = 100$ and $c = 20$ as an example. Applying DPA in ACO requires 32% more CPU time (per Table 5) but improves the solution quality by 37.28% (per Table 4).

4.4.2. *Effectiveness of incorporating pheromone feedback mechanism*

Recall that a pheromone feedback mechanism is designed in ACO_DPA. To evaluate the effectiveness of this feedback mechanism, the results obtained with and without it were compared. Let Z_{off} be the objective function value when the feedback mechanism is switched off. The gap between Z_{off} and Z_{DPA} is calculated as $(Z_{\text{off}} - Z_{\text{DPA}})/Z_{\text{off}}$. A multiple regression analysis for the gap between these two variants was conducted by considering m, q, n, c and θ as the independent variables. The results show that q and θ do not have a significant impact on the gap. Thus, the results shown in Table 6 were consolidated by taking the average value across different q and θ values, respectively. The average gaps between Z_{off} and Z_{DPA} with $n = 25$, 50 and 100 are 17.04%, 11.04% and 3.99%, respectively, per Table 6. The effect of the feedback mechanism

Table 6. Average gaps between Z_{off} and Z_{DPA}.

	$n = 25$			$n = 50$			$n = 100$		
M	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)
2	20.49	16.80	22.55	9.96	13.64	12.38	7.20	5.96	7.12
4	11.74	16.10	17.06	8.99	8.74	13.64	1.25	1.69	3.80
8	14.97	18.72	14.97	8.29	9.12	14.57	0.01	3.34	5.57

decreases as the number of jobs increases. It is observed that the effect from the pheromone feedback is dampened by pheromone evaporation when the number of jobs is relatively large.

4.4.3. *Value of integrating production and distribution stages*

In practice, the scheduling decisions of production and distribution are usually made separately and often sequentially by different decision-makers. The effect of integrating the production and distribution scheduling is evaluated by comparing the results obtained using the integrated approach presented in this article (ACO_DPA) with the method that addresses production and distribution sequentially.

A two-step algorithm involving an ACO (named ACO_SA) was designed. In the first stage (production) of ACO_SA, only customer service (measured by the completion time) is considered. The goal in this stage is to minimize the total weighted completion times, $\sum_{j=1}^{n} W_j C_j$. To construct one path in the production stage, ACO_SA uses Equation (29) to determine the heuristic information of node u from node i for ant k. In contrast to Equation (18) in ACO_DPA, ACO_SA does not consider the distribution time from the job associated with node i to the job associated with node u.

$$\eta_{iu} = \left(\sum_{x \in P^k(i)} w_{\omega(x)} C_{\omega(x)} + w_{\omega(u)} C_{\omega(u)} \right)^{-1}, \quad u \in E^k(i) \tag{29}$$

After the production schedule has been determined, jobs are waiting to be allocated to batches before distribution. Here, jobs are scheduled to be distributed according to their completion times. Dynamic programming is employed to assign jobs into delivery batches. The recursive relationship in algorithm DPA, *i.e.* Equation (22), is modified as shown in Equation (30). The difference between Equations (22) and (30) is the consideration of job delivery times.

$$V(j_{(i)}) = \min \left\{ V(j_{(i-h)} + \left(f + c_{0j_{(u)}} + \sum_{x=u}^{i-1} c_{j_{(x)}j_{(x+1)}} \right) \middle| h = 1, \ldots, \min(c, j_{(i)}) \right\} \tag{30}$$

Once the allocation of batches has been performed, ACO_SA utilizes the construction method to determine the delivery routes as in ACO_DPA. The difference is that the pheromone feedback mechanism is not incorporated into ACO_SA.

Let Z_{SA} be the objective function value obtained using ACO_SA. The improvement percentage of the overall cost from Z_{SA} to Z_{DPA} is calculated as $(Z_{SA} - Z_{\text{DPA}})/Z_{SA}$. A multiple regression analysis for the gap between ACO_SA and ACO_DPA was conducted by considering m, q, n, c and θ as the independent variables. The results show an insignificant impact of q on the relevant differences. Therefore, the results were consolidated by taking the average value of q. The average improvement percentages between Z_{SA} and Z_{DPA} are summarized in Table 7.

As seen in Table 7, the average improvement increases as θ increases when n is fixed at a constant value, indicating the merit of using ACO_DPA when the focus is on the weighted sum of

Table 7. Value of integration: average improvement percentage from Z_{SA} to Z_{DPA}.

		$\theta = 0.2$			$\theta = 0.5$			$\theta = 0.8$		
n	m	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)	$c = 5$ (%)	$c = 10$ (%)	$c = 20$ (%)
25	2	14.88	6.21	28.42	13.19	23.00	48.20	12.61	38.42	53.15
	4	12.56	4.09	30.01	19.09	14.68	44.24	24.34	36.52	59.55
	8	9.72	10.32	32.78	14.87	26.22	49.38	24.84	29.68	49.70
50	2	6.91	5.84	19.82	6.15	13.09	23.56	14.62	17.41	29.87
	4	2.83	4.08	14.47	5.66	9.50	23.18	9.43	16.77	29.63
	8	1.82	5.70	12.61	5.97	7.60	21.96	7.52	14.74	26.03
100	2	7.14	7.68	17.96	5.47	13.91	28.57	11.80	22.85	34.28
	4	5.21	3.99	9.87	8.49	8.01	24.56	4.39	14.92	35.68
	8	2.37	6.25	11.31	3.32	7.31	22.09	6.65	13.76	36.16

job delivery times. It is also found that the average improvement is larger when $n = 25$ compared with that when $n = 50$ or 100. Note that this result does not imply that the value of integration is less significant for larger scale problem instances. For example, consider two cases where the overall costs for ACO_SA and ACO_DPA are 4 and 3 at a smaller n value, and 100 and 80 at a bigger n value, respectively. The improvement rate of the former and the latter case is 25% and 20%, respectively. Even though the improvement rate is larger when n is smaller, the absolute improvement is very small.

Moreover, it is observed that the average improvement percentage increases when c increases for a fixed n value. As the vehicle capacity gets larger, there are more opportunities to integrate different combinations of job batches for delivery. Consequently, the integration effect becomes more significant when the vehicle capacity is larger.

Last but not least, the average gap between ACO_SA and ACO_DPA for 162 instances is 18.04%, demonstrating the value and importance of the integration of both production and distribution stages.

5. Conclusions and future research directions

Traditional approaches consider production–distribution scheduling separately and sequentially with little or no coordination between these two stages. The current trend for inventory reduction across stages within supply chains has caused some businesses to consider seriously integrating the production and distribution activities in order to stay profitable. In this article, an integrated production and distribution scheduling problem is studied. This type of problem can be commonly found in businesses that keep very little finished good inventory, such as e-business direct sales, express delivery services and food catering.

An ACO heuristic was designed to obtain near-optimal solutions for the studied problem. In this ACO algorithm, a polynomial algorithm involving dynamic programming, DPA, is designed to assign jobs into delivery batches. A pheromone feedback mechanism is embedded to increase the integration of both production and distribution stages during schedule construction. Computational experiments are conducted to evaluate the effectiveness of the ACO algorithm. The computational results show that the present ACO algorithm is capable of generating near-optimal solutions, and that the DPA and the designed pheromone feedback mechanism can improve the quality of solutions obtained. The value of integration was also evaluated in this study by comparing the results of using ACO to solve the production and distribution scheduling problem sequentially with those obtained using an integrated approach. It was found that the value of integration is significant.

The integrated production and distribution scheduling problem still requires more discussion in the literature and in-depth investigation in practice. Further research can be conducted to explore more practical approaches to solving this class of problems. Moreover, it is also important to discuss such problems under different cost structures, such as including the performance measures related to due date.

Acknowledgements

Funding for the first author was provided by National Science Council of Taiwan under grant numbers NSC 98-2410-H-009-005 and NSC 101-2410-H-009-005-MY2. The authors are also grateful to Dr Chi-Leung Chu for providing suggestions on an earlier version of this article.

References

Afshar, A., F. Sharifi, and M. R. Jalali. 2009. "Non-dominated Archiving Multi-colony Ant Algorithm for Multi-objective Optimization: Application to Multi-purpose Reservoir Operation." *Engineering Optimization* 41 (4): 313–325.

Armstrong, R., S. Gao, and L. Lei. 2008. "A Zero-Inventory Production and Distribution Problem with a Fixed Customer Sequence." *Annals of Operations Research* 159 (1): 395–414.

Averbakh, I. 2010. "On-line Integrated Production–Distribution Scheduling Problems with Capacitated Deliveries." *European Journal of Operational Research* 200 (2): 377–384.

Bullnheimer, B., R. F. Hartl, and C. Strauss. 1999. "An Improved Ant System Algorithm for the Vehicle Routing Problem." *Annals of Operations Research* 89: 319–328.

Chang, Y. C., K. H. Chang, and T. K. Chang. 2013. "Applied Column Generation-based Approach to Solve Supply Chain Scheduling Problems." *International Journal of Production Research*, in press.

Chang, Y.-C., and C.-Y. Lee. 2004. "Machine Scheduling with Job Delivery Coordination." *European Journal of Operational Research* 158 (2): 470–487.

Chen, Z.-L. 2004. "Integrated Production and Distribution Operations: Taxonomy, Models and Review." In *Handbook of Quantitative Supply Chain Analysis: Modeling in the e-Business Era*, edited by D. Simchi-Levi, S. D. Wu and Z.-J. Shen, 711–746. New York: Kluwer Academic Publishers.

Chen, Z. L. 2010. "Integrated Production and Outbound Distribution Scheduling: Review and Extensions." *Operations Research* 58 (1): 130–148.

Chen, B., and C.-Y. Lee. 2008. "Logistics Scheduling with Batching and Transportation." *European Journal of Operational Research* 189 (3): 871–876.

Chen, Z.-L., and G. Pundoor. 2006. "Order Assignment and Scheduling in a Supply Chain." *Operations Research* 54 (3): 555–572.

Chen, Z.-L., and G. L. Vairaktarakis. 2005. "Integrated Scheduling of Production and Distribution Operations." *Management Science* 51 (4): 614–628.

Colorni, V., M. Dorigo, and V. Maniezzo. 1991. "Distributed Optimization by Ant Colonies." In *First European Conference on Artificial Life*, edited by F. Varela and P. Bourgine, 134–142. Amsterdam: Elsevier.

Dorigo, M. 1992. *Optimization, learning and natural algorithms*. Ph.D. thesis, Politecnico di Milano.

Dorigo, M., M. Birattari, and T. Stützle. 2006. "Artificial Ants as a Computational Intelligence Technique." *IEEE Computational Intelligence Magazine* 1: 28–39.

Dorigo, M., and G. Di Caro. 1999. "The Ant Colony Optimization Meta-heuristic." In *New Ideas in Optimization*, 11–32. London: McGraw-Hill.

Dorigo, M., and L. Gambardella. 1997. "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem." *IEEE Transactions on Evolutionary Computation* 1 (1): 53–66.

Dorigo, M., and T. Stützle. 2004. *Ant Colony Optimization*. Boston, MA: MIT Press.

Dorigo, M., and T. Stützle. 2010. "Ant Colony Optimization: Overview and Recent Advances." In *Handbook of Metaheuristics*, 227–263. New York: Springer.

Gagné, C., W. L. Price, and M. Gravel. 2002. "Comparing an ACO Algorithm with other Heuristics for the Single Machine Scheduling Problem with Sequence-dependent Setup Times." *Journal of the Operational Research Society* 53 (8): 895–506.

Garey, M. R., and D. S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: W. H. Freeman.

Geismar, H. N., G. Laporte, L. Lei, and C. Sriskandarajah. 2008. "The Integrated Production and Transportation Scheduling Problem for a Product with a Short Life Span and Non-instantaneous Transportation time." *Informs Journal on Computing* 20 (1): 21–33.

Golden, B. L., S. Raghavan, and E. A. Wasil, eds. 2008. *The Vehicle Routing Problem: Latest Advances and New Challenges*. New York: Springer.

Hall, N. G., and C. N. Potts. 2003. "Supply Chain Scheduling: Batching and Delivery." *Operations Research* 51 (4): 566–584.

Hall, N. G., and C. N. Potts. 2005. "The Coordination of Scheduling and Batch Deliveries." *Annals of Operations Research* 135: 41–64.

Huang, K.-L., and C.-J. Liao. 2008. "Ant Colony Optimization Combined with Taboo Search for the Job Shop Scheduling Problem." *Computers and Operations Research* 35 (4): 1030–1046.

Lee, C. Y., and Z. L. Chen. 2001. "Machine Scheduling with Transportation Considerations." *Journal of Scheduling* 4 (1): 3–24.

Lenstra, J. K., and D. B. Shmoys. 1990. "Approximation Algorithms for Scheduling Unrelated Parallel Machines." *Mathematical Programming* 46 (3): 259–271.

Li, C.-L., and J. Ou. 2005. "Machine Scheduling with Pickup and Delivery." *Naval Research Logistics* 52 (7): 617–630.

Li, C.-L., and J. Ou. 2007. "Coordinated Scheduling of Customer Orders with Decentralized Machine Locations." *IIE Transactions* 39 (9): 899–909.

Li, C.-L., and G. Vairaktarakis. 2007. "Coordinating Production and Distribution of Jobs with Bundling Operations." *IIE Transactions* 39 (2): 203–215.

Li, C.-L., G. Vairaktarakis, and C.-Y. Lee. 2005. "Machine Scheduling with Deliveries to Multiple Customer Locations." *European Journal of Operational Research* 164 (1): 39–51.

Liu, J-L. 2005. "Rank-based Ant Colony Optimization Applied to Dynamic Traveling Salesman Problems." *Engineering Optimization* 37 (8): 831–847.

Pinedo, M. 2002. *Scheduling: Theory, Algorithms, and Systems*. 2nd ed. Upper Saddle River, NJ: Prentice Hall.

Pundoor, G., and Z.-L. Chen. 2005. "Scheduling a Production–distribution System to Optimize the Tradeoff between Delivery Tardiness and Distribution Cost." *Naval Research Logistics* 52 (6): 571–589.

Sarmiento, A. M., and R. Nagi. 1999. "A Review of Integrated Analysis of Production–Distribution Systems." *IIE Transactions* 31 (11): 1061–1074.

Shyu, S. J., B. M. T. Lin, and P. Y. Yin. 2004. "Application of Ant Colony Optimization for No-wait Flowshop Scheduling Problem to Minimize the Total Completion Time." *Computers & Industrial Engineering* 47 (2–3): 181–193.

Stützle, T., F. Intellektik, F. Informatik, and T. H. Darmstadt. 1998. "An Ant Approach to the Flow Shop Problem." *6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*, September 7–10. Aachen, Germany: ELITE Foundation.

T'kindt, V., N. Monmarche, F. Tercinet, and D. Laugt. 2002. "An Ant Colony Optimization Algorithm to Solve a 2-machine Bicriteria Flowshop Scheduling Problem." *European Journal of Operational Research* 142 (2): 250–257.

Wang, H., and C.-Y. Lee. 2005. "Production and Transport Logistics Scheduling with Two Transport Mode Choices." *Naval Research Logistics* 52 (8): 796–809.

Zhong, W., G. Dosa, and Z. Tan. 2007. "On the Machine Scheduling Problem with Job Delivery Coordination." *European Journal of Operational Research* 182 (3): 1057–1072.