European Journal of Operational Research 259 (2017) 906–916

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Production, Manufacturing and Logistics

Integrated production and distribution scheduling with a perishable product

Priyantha Devapriya[a], William Ferrell[b,*], Neil Geismar[c]

[a] Partnerships & Strategic Affiliations, Augusta University Medical Center, Augusta, GA 30912, USA
[b] Department of Industrial Engineering, Clemson University, Clemson, SC 29634, USA
[c] Mays Business School, Texas A&M University, College Station, TX 77843, USA

A R T I C L E I N F O

Article history:
Received 16 June 2015
Accepted 7 September 2016
Available online 14 September 2016

Keywords:
Production and distribution
Integrated scheduling
Genetic algorithms
Memetic algorithms
Logistics

A B S T R A C T

This research focuses on the practical problem of a perishable product that must be produced and distributed before it becomes unusable but at minimum cost. The problem has some features of the integrated production and distribution scheduling problem in that we seek to determine the fleet size and the trucks' routes subject to a planning horizon constraint. In particular, this research differs because the product has a limited lifetime, the total demand must be satisfied within a planning horizon, multiple trucks can be used, and the production schedule and the distribution sequence are considered. A mixed integer programming model is formulated to solve the problem and, then, heuristics based on evolutionary algorithms are provided to resolve the models.

1. Introduction

This research is motivated by a practical problem in which a production facility with limited capacity is producing a perishable product that requires delivery to a set of geographically dispersed customer sites on or before its effective lifetime expires. The specific version of this problem that is considered here involves a single plant producing a single product that must be delivered to a known set of customers with known demand before the end of the planning horizon. The objective is to determine the number of trucks, their routes, and the production sequence that minimizes the total cost of distribution while meeting the demands at the disparate locations before the product lifetime expires. It is assumed that the cost of distribution includes two components: a fixed cost associated with each truck hired that is paid once during the planning horizon and a variable cost per unit distance traveled. As is true in practice, the fixed cost is significantly higher than the variable cost. The problem is complicated by adding the realistic assumptions that the product lifetime begins immediately after the batch is produced and that the product is useless after its lifetime has expired and cannot be delivered; deliveries after the lifetime is expired are not partially useful or partially degraded. For example, these assumptions will limit the potential of building to stock (e.g., production in advance of the delivery truck arrival) for later delivery. They will also make it difficult to combine deliveries to distance customers to increase truck utilization. Problems with the general characteristics similar to ours are referred to as single plant, integrated production and distribution scheduling problem (IPDSP).

With the general problem now explained, a mixed integer linear formulation for the IPDSP is proposed. It is then shown that IPDSP cannot be solved to optimality with commercial MIP software except for very small problems so heuristics are developed to obtain approximate solutions. Two lower bounds to IPDSP are presented along with two meta-heuristics for finding good, approximate solutions to more realistically sized problems. A numerical study is used to illustrate the types of information that can be ascertained by using the proposed approach to this problem. The paper concludes with a brief discussion of a few research extensions of IPDSP.

2. Literature review

The complete literature history leading to the IPDSP is rich, varied and large. As such, it is impossible to provide a comprehensive treatment here so we will only note some of the work that is most relevant to this research. To begin, one of the common themes in supply chain research is making two or more decisions simultaneously that traditionally were made independently like inventory and distribution. Not only do these provide interesting research problems but their solution can provide real saving in practice as well. The research addresses such an integrated problem. Another

* Corresponding author.
E-mail addresses: ddevapriya@augusta.edu (P. Devapriya), fwillia@clemson.edu, ngeismar@mays.tamu.edu (W. Ferrell).

http://dx.doi.org/10.1016/j.ejor.2016.09.019
0377-2217/

segment of the literature for decades addresses the vehicle routing problem but the more specialized fleet size and mix vehicle routing problem (FSMVRP) is closely related to this research. In FSMVRP, the problem of finding the composition of a fleet and the routing of the fleet is solved to minimize the cost under a given set of problem constraints. A form of FSMVRP was addressed by Clarke and Wright (1964) and work continues on this problem today; however, to our knowledge none of the literature on FSMVRP addresses the fleet sizing problem with multiple use of vehicles where fleet is a decision variable.

Research on the production–distribution integration problem that characterizes part of this research dates back to at least King and Lowe (1980). Chang and Lee (2003) developed heuristics for three cases involving machine scheduling on single and parallel machines, distribution with a single vehicle, and one or two customer zones which they proved were NP hard. In a similar theme, Li, Vairaktarakis, and Lee (2005) developed a polynomial time algorithm for the NP hard problem of single-machine scheduling and distribution to single and multiple customers with certain restrictions.

When perishable or time sensitive products are considered, most of the literature is quite recent (Chen, 2010). For example, a very interesting thread of research can be found in the food engineering literature. For example, Hsu, Hung, and Li (2007) extended a vehicle routing problem with time-windows to address food distribution from a distribution center and included randomness for food degradation and travel time. Osvald and Stirn (2008) used the same general approach as Hsu but incorporated a tabu search procedure to determine significant reductions in losses of food quality using data from distribution to Slovenian food markets. Chen, Hsueh, and Chang (2009) also used this same approach with an operations research focus to model a production and distribution situation with probabilistic customer demands. They developed a nonlinear model that maximizes the expected profit of the supplier.

There are other similar situations that are parallel with our problem. Hurter and Van Buer, (1996) treat newspaper distribution as perishable goods and the objective is to find a feasible production and distribution schedule with a minimum number of identical delivery vans so that all newspapers are delivered to drop-off points on or before a given time. Garcia, Lozano, and Canca (2004) consider a multi-plant problem with parallel machines at each plant and find a polynomial time algorithm when production had no bottlenecks. Garcia and Lozano (2004) analyze a ready-mix concrete distribution operation with a finite capacity plant. Their objective is to maximize the total customer orders satisfied before the due date with unlimited truck availability. The scenario in these latter two studies are limited to direct deliveries and only one order is delivered on each trip by each truck. Chen and Pundoor (2006) analyze the integration of order assignment, production scheduling, and distribution scheduling of a manufacturer that produces a large variety of products with short life cycles and a short selling season. For a given set of orders, they solve the problem of determining: (1) which order to be assigned to which plant, (2) a production schedule, and (3) a shipping schedule for completed orders so that a given set of performance measures are maximized. The model considers a direct shipping strategy from plants to the distribution center. They consider different variations of the problem and develop fast heuristics to resolve more complex versions. In a related study, Geismar, Laporte, Lei, and Sriskandarajah (2008) analyze an extension with one finite capacity production plant and truck. The objective is to minimize the total time required to satisfy all demand. Finally, the production and distribution of Radioisotope F-18 was investigated by Lee, Kim, Johnson, and Lee (2014). Since this cancer treating isotope has a half-life of just 110 minutes, production and distribution to medical end-users is critical. They addressed this with a mixed integer model and search algorithms, and illustrated how the number of vehicles could be reduced using real data.

This research is an extension of Geismar et al. (2008) and shares the features such as short lifetime of the product, capacitated delivery, and fixed production rate at the plant. The most significant differences are that the fleet size is a decision variable and the existence of a planning horizon constraint.

3. A mixed-integer linear formulation for IPDSP

In this section a mixed-integer formulation is given for the IPDSP. The key assumptions behind this model are:

- The lifetime of the product begins when the production for batch to be placed in a truck is completed.
- All customer locations are assumed to be in the 2-D plane with the distance matrix symmetric and known.
- The production plant has a fixed capacity that is sufficient to satisfy the total demand of customers within the planning horizon.
- Production at the plant is scheduled in such a way that a truck is able to deliver goods to customer sites before expiration. That is, a feasible solution is to have a unique truck deliver to each customer but the cost would be quite high.
- Distribution is subject to the following assumptions: (1) each customer's demand must be satisfied in one delivery, (2) orders for more than one customer are allowed in one truck, (3) a truck is allowed to carry less than a full truckload, and (4) trucks should return to the plant after completing the delivery of one route and start the next route until the planning horizon has expired.

As mentioned earlier, the goal is to minimize the total cost for a given planning horizon with total cost consisting of a fixed cost F that is incurred once in the planning period for each truck and a variable cost per unit distance traveled which is set at 1 without loss of generality. Let the given planning horizon be H time units, the production rate at the plant be r units per unit time, and the lifetime of the product be B time units after production of the batch is completed. Let $N=\{0, 1, 2, 3, \ldots, n\}$ denote the customers where 0 is the plant and $N'=N\backslash\{0\}$ are the n customers geographically scattered in the 2-D plane. It is assumed that fleet of identical trucks is used to deliver goods starting from the plant and each customer i has a demand q_i, $i \in N'$ that must be satisfied within H. Each truck has a finite capacity C, where $\max_{i\in N'} q_i \le C \le \sum_{i=1}^{n} q_i$ and each truck starts its route from the plant, visits a customer or sequence of customers and returns to the plant. A truck can be used multiple times within H. $T=\{1, \ldots, H\}$ is a set of discrete time periods dividing H into equal intervals and τ_{ij} is the travel time from customer i to customer j.

The notation for this IPDSP model is, then:

<u>Parameters</u> (inputs to the model)

$\tau_{ij}=$	travel time from customer i to customer j(time units)
$B=$	lifetime of the product (time units)
$C=$	capacity of each truck (units)
$C_{ij}=$	travel cost between customers i to customer j (monetary units)
$F=$	fixed cost of each truck hired (monetary units)
$q_i=$	demand of customer i (units)
$r=$	production rate of plant (units/unit time)

<u>Decision variables</u>

$$X_{ijkm} = \begin{cases} 1 \text{ if truck } m \text{ visits customer } j \text{ after customer } i \text{ in trip} \\ k \quad i, j \in N \text{ and } k, m \in N' \\ 0 \quad \text{otherwise} \end{cases}$$

$$P_{kmt} = \begin{cases} 1 \text{ if the Plant is producing for trip } k \text{ of truck } m \text{ at} \\ \quad \text{time epoch } t \quad t \in T \text{ and } k, m \in N' \\ 0 \quad \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if truck } i \text{ is used for delivery} \quad i \in N' \\ 0 \quad \text{otherwise} \end{cases}$$

$$Z_{km} = \begin{cases} 1 \text{ if trip } k \text{ of truck } m \text{ is part of the optimal schedule} \\ \quad k,\ m \in N' \\ 0 \quad \text{otherwise} \end{cases}$$

$e_i =$ set of real numbers used in the sub $-$ tour elimination constraints (Miller et al. 1960)

Internally computed variables

$t^d_{km}=$ Delivery start time for trip k of truck m, $k,\ m \in N'$

$t^p_{km}=$ Production start time for trip k of truck m, $k,\ m \in N'$

The cost based model is:

$$\min \sum_{\substack{i,j \in N \\ k,m \in N'}} C_{ij} X_{ijkm} + F \sum_{m \in N'} y_m \tag{1}$$

Subject to:

$$\sum_{i,j \in N} X_{ijkm}\, q_j \le C \qquad \forall k,\ m \in N',\ q_0 = 0 \tag{2}$$

$$\sum_{\substack{i \in N \\ k,m \in N'}} X_{ijkm} = 1 \qquad \forall j \in N' \tag{3}$$

$$\sum_{\substack{j \in N \\ k,m \in N'}} X_{ijkm} = 1 \qquad \forall i \in N' \tag{4}$$

$$\sum_{j,k \in N'} X_{0jkm} = \sum_{k \in N'} Z_{km} \qquad \forall m \in N' \tag{5}$$

$$\sum_{j,k \in N'} X_{j0km} = \sum_{k \in N'} Z_{km} \qquad \forall m \in N' \tag{6}$$

$$\sum_{i \in N} X_{ijkm} = \sum_{i \in N} X_{jikm} \qquad \forall m,\ k, j \in N' \tag{7}$$

$$e_i - e_j + 1 \le n\left(1 - X_{ijkm}\right) \quad \forall i,\ j,\ k;\ m \in N' \tag{8}$$

$$X_{ijkm} \le y_m \qquad \forall i,\ j \in N;\ k,\ m \in N' \tag{9}$$

$$X_{ijkm} \le Z_{km} \qquad \forall i,\ j \in N;\ k,\ m \in N' \tag{10}$$

$$Z_{km} \le y_m \qquad \forall k,\ m \in N' \tag{11}$$

$$t^d_{km} - \left(t^p_{km} + \frac{1}{r} \sum_{i,j \in N} X_{ijkm}\, q_j \right) + \sum_{\substack{i \in N \\ j \in N'}} X_{ijkm}\, \tau_{ij} \le B \quad \forall k,\ m \in N' \tag{12}$$

$$t^d_{km} + \sum_{i,j \in N} X_{ijkm}\, \tau_{ij} \le H \qquad \forall k,\ m \in N' \tag{13}$$

$$t^p_{km} + \frac{1}{r} \sum_{i,j \in N} X_{ijkm}\, q_i \le t^d_{km} \quad \forall k,\ m \in N' \tag{14}$$

$$t^d_{km} + \sum_{\substack{i \in N \\ j \in N}} X_{ijkm}\, \tau_{ij} \le t^d_{(k+1)m} \quad \forall k,\ m \in N' \tag{15}$$

$$t^p_{km} + \frac{1}{r} \sum_{\substack{i \in N \\ j \in N}} X_{ijkm}\, q_j \le t^p_{(k+1)m} \forall k,\ m \in N' \tag{16}$$

$$\sum_{k,m \in N'} P_{kmt} \le 1 \qquad \forall t \in T \tag{17}$$

$$\sum_{t \in T} P_{kmt} \le \frac{1}{r} \sum_{i,j \in N} X_{ijkm}\, q_i \qquad \forall k,\ m \in N' \tag{18}$$

$$t^p_{km} \le t\, P_{kmt} + M\left(1 - P_{kmt}\right) \quad \forall k,\ m \in N';\ t \in T \tag{19}$$

$$\begin{aligned} & t^p_{km} + \tfrac{1}{r} \sum_{\substack{i \in N \\ j \in N'}} X_{ijkm}\, q_j \ge t\, P_{kmt} && \forall k,\ m \in N';\ t \in T \\ & X_{ijkm},\ P_{kmt},\ Z_{km},\ y_m \in \{0, 1\} && \forall k,\ m,\ j \in N';\ i \in N \\ & t^d_{km},\ t^p_{km},\ e_i \text{ integer} && \forall k,\ m,\ j \in N';\ i \in N \end{aligned} \tag{20}$$

Constraints (2) through (8) are typical vehicle routing constraints. Constraint (2) is the truck capacity constraint and constraints (3) and (4) ensure that a customer is visited exactly once. Constraints (5) and (6) ensure the number of times a truck leaves the plant and comes back to the plant is the same as the number of trips for that particular truck. Constraint (7) is the continuity constraint for each trip. The constraints in (8) are the subtour elimination constraints from Miller, Tucker, and Zemlin (1960). Constraints (9) and (10) ensure that a customer cannot be visited by given trip of a given truck without hiring that particular truck. Constraint (11) reduces the number of constraints in the model. Constraint (12) forces delivery before the lifetime is exhausted while constraint (13) ensures all deliveries are completed before the end of the planning horizon. Constraint (14) requires production be completed before the delivery can begin and constraint (15) ensures that a truck cannot start a new trip before completing the previous trip and returning to the plant. Constraint (16) ensures that the plant cannot start producing for trip $(k+1)$ before completing the production for trip k. Constraints (17)–(20) together ensure the availability of the single plant at any given time epoch t. Constraint (17) ensures that the plant can produce only for one trip at any given time period. Constraint (18) ensures that the production time of the plant for a batch is the minimum amount of the time required to meet demand of the corresponding route. Constraints (19) and (20) together ensure that when the plant starts producing for a trip it continuously produces until the total demand is produced for that trip.

IPDSP belongs to the class of NPC (Nondeterministic Polynomial-time Complete) problems because it is an extension of the traveling salesman problem (TSP) and the TSP belongs to the class of NPC problems. This means that all known algorithms that define an optimal solution require exponentially increasing computational time as the number of customers(n) increases; therefore, heuristic methods which provide approximate solutions are justified and are required for realistic sized problems. To illustrate this, Table 1 reports the results of solving the model using CPLEX 10 with $B = 50$, $C = 60$ and $F = 200$.

CPLEX solved the problem with 4 customers to optimality in less than 0.01 hours; however, CPLEX could not solve any problem bigger than 4 customers to optimality within 20 hours. The % Gap corresponds to the gap between the current best integer solution and the current best node given by CPLEX. For any problem with more than 9 customers, CPLEX even could not find a feasible solution within 10 days. As shown in Table 1, when the number

Table 1
Problem size growth with number of customers and comparison of optimal solution and heuristic solutions.

Problem size	Planning horizon	Number of variables	Number of constraints	% Gap	CPLEX solution	CPLEX computational time (hours)	IPDSP-MA1 solution	IPDSP-MA1 computational time (minute)	IPDSP-MA2 solution	IPDSP-MA2 computational time (minute)
4	150	2921	6295	0	293	0.01	293	<1	N/A	
5	150	4836	10,551	21.71	910	>20	710	<1	N/A	
6	200	7429	16,495	22.53	624	>20	624	<1	N/A	
7	200	10,844	24,553	26.19	766	>20	766	1	N/A	
8	200	15,249	35,223	30.9	728	>20	703	1	N/A	
9	200	24,886	57,225	42.01	1034	>20	878	2	N/A	
10	200	32,821	76,801	n/a	None	10 days	1144	2	N/A	
12	200	54,169	130,953	n/a	None	10 days	1399	3	1410	2
15	250	104,206	262,551	n/a	None	10 days	1182	3	1189	3
20	350	319,241	806,751	n/a	None	10 days	1842	5	1855	4

of customers increases, problem size grows exponentially. As mentioned previously, since IPDSP is NPC, approximation approaches are justified so attention is now turned to a heuristic approach to approximately solve the IPDSP using evolutionary algorithms.

4. Heuristic solution approach for IPDSP

The heuristic solution approach used in this paper to solve IPDSP is based on the "route first cluster second" method from the vehicle routing literature (Beasley, 1983). First a route containing all customers that starts and ends at the plant is found. Then the route is partitioned into sub-tours based on truck capacity and lifetime constraints. The production quantities at the plant exactly match the demands of respective sub-tours. For example, if a given route is partitioned into five sub-tours then the plant will produce five batches. It is assumed that plant capacity is sufficient for production to initially be scheduled so each batch is completed exactly when a truck is available to deliver. This is called a no-wait schedule. If the plant has idle time between two consecutive batches, the no-wait schedule is compressed subject to the product lifetime constraint so that overall makespan is maximally compressed. Section 4.2 provides additional details of the procedure. This heuristic approach has been shown to produce a solution in other situations which motivates adaption here.

The following notation is used in the development of the heuristics. A sequence containing all customers is denoted by σ, a sub-tour of σ is σ_j and $\sigma(i_j)$ denotes the ith customer of the jth sub-tour. The sequence σ can be divided into k sub-tours with the jth sub-tour $\sigma_j=(P,\ \sigma(1_j),\ \sigma(2_j),\ \dots,\ \sigma(i-1)_j,\ \sigma(i_j),\ P)$ where $j=\{1,\ 2,\ 3,\ \dots,\ k\}$. In σ_j, first customer is denoted by $\sigma(1_j)$ while the last customer is denoted by $\sigma(i_j)$. Each sub-tour must consist of at least one customer and $P$is the plant. Since all distances are positive and the goal is to minimize cost, it is assumed that each customer belongs to exactly one sub-tour and that each sub-tour is served once by one truck. For a given σ, if the total demand cannot be satisfied by one truck within H, the problem is expanded to assigning $s \geq 2$ trucks to the k sub-tours. The Euclidian distance $l_{i,j}$ between customers i and j and the speed of the truck are used to compute the travel time, $\tau_{i,j}$. To ensure feasibility, $\tau_{0,i} \leq B$ is imposed $\forall i \in N\backslash\{0\}$. Finally, it is assumed that there is no inventory held at any stage of the process and the distances $l_{i,j}$ are symmetric and satisfy the triangle inequality.

4.1. Optimal tour partitioning

Partitioning a sequence σ into sub-tours is constrained by both the truck's capacity and the product's lifetime. An optimal tour partitioning procedure proposed in Beasley (1983) and later called algorithm Split by Prins (2004) is used in this research to partition σ. Let σ be a sequence with n customers. Algorithm Split uses a directed graph G with $n+1$ nodes and a set of arcs where each arc corresponds to a feasible sub-tour in σ and arc weights correspond to the distance of each sub-tour. Node 0 represents the plant and the other n nodes represent customers in the sequence while the last node represents the last customer in the sequence. The graph G contains an arc between nodes i and j if the sub-tour visiting customers represented by nodes $i+1$ through j along the sequence are feasible, both in terms of the truck capacity and the product lifetime. The arcs in the shortest path of G from node 0 to the last node optimally partition the sequence into sub-tours minimizing the distance required to visit all customers.

4.2. The makespan compression of IPDSP

This section describes how a no-wait schedule is calculated and compressed using a polynomial time algorithm for a given sequence and a given set of sub-tours when more than one truck is required to satisfy the demand within H. During some or all of the time that a truck is delivering sub-tour σ_j, the plant is producing products for sub-tour σ_{j+1}. Let $q_{\sigma(k)}$ be the demand of a customer k in a given sub-tour. Let m_j, m_{j+1} be the number of customers in sub-tour σ_j and σ_{j+1}, respectively, p_{j+1} be the time required to produce the required amount of product for sub-tour σ_{j+1}, and T_{σ_j} be the total travel time of sub-tour j. Then $p_{j+1} = \frac{1}{r}\sum_{k=1_{j+1}}^{m_{j+1}} q_{\sigma(k)}$ and $T_{\sigma_j} = \tau_{0,\sigma(1_j)} + \sum_{k=1_j}^{(m-1)_j} \tau_{\sigma(k),\sigma(k+1)} + \tau_{\sigma(s_j),0}$ where the delivery time from the plant to the last customer in sub-tour j is given by $T_{\sigma_j} - \tau_{\sigma(m_j),0}$ and $\tau_{\sigma(m_j),0}$ is the travel time between the last customer in sub-tour j and the plant.

Unlike the one truck case, the delivery finish times of all sub-tours in progress must be monitored in order to obtain the best potential delivery start time for the next sub-tour in a no-wait schedule. For example, consider an IPDSP that has two trucks and eight sub-tours; the no-wait delivery start time of the fifth sub-tour is to be determined. Since there are only two trucks, delivery start time of the fifth sub-tour depends on the delivery of sub-tours being delivered by the two trucks. These two sub-tours can be any combination of two sub-tour subsets from the set $\{1,\ 2,\ 3,\ 4\}$ and, depending on the distance, one truck could complete more than one sub-tours before the other truck completes one. The theorem given below establishes this concept and provides the way to calculate a no-wait schedule for a given sequence of IPDSP.

Theorem 1. Let s trucks be in use and let the delivery finish time of the s sub-tours in progress at a particular time epoch be F_g, where $g \in \{1,\ 2,\ \dots,\ s\}$. Further, let v_j be the delivery start time of sub-tour σ_j in a no-wait schedule. The delivery of sub-tour σ_{j+1} with m_{j+1} customers is started at time v_{j+1} where

$$v_{j+1} = \max\left[\min_{1\leq g\leq s} F_g,\quad v_j + \frac{1}{r}\sum_{k=1_{j+1}}^{m_{j+1}} q_{\sigma(k)}\right] \tag{21}$$

Table 2
Algorithm compress example data.

Sub-tour	Production time (p_j)	Delivery time ($T_{\sigma_j} - \tau_{\sigma(i_j),0}$)	Return time ($\tau_{\sigma(i_j),0}$)
σ_1	2	17	2
σ_2	3	4	1
σ_3	3	2	1
σ_4	6	4	2
σ_5	3	5	1
σ_6	1	5	3

Proof. Since no inventory is held at the plant, production for sub-tour $j+1$ cannot begin before time v_j. Given v_j, delivery of sub-tour σ_{j+1} cannot be started earlier than V_{j+1}, where $V_{j+1} = (v_j + \frac{1}{r}\sum_{k=1_{j+1}}^{m_{j+1}} q_{\sigma(k)})$. However, in order to start the routing on sub-tour $j+1$, at least one truck should be available at the plant when the production for sub-tour $j+1$ is completed. The earliest delivery finish time of the last s sub-tours is $\min_{1\le g\le s} F_g$. So in order to obtain a no-wait schedule, v_{j+1} should be scheduled at $\max(\min_{1\le g\le s} F_g, \quad V_{j+1})$ ■

The next issue to be addressed is how to compress a no-wait schedule to reduce the makespan. Assume a sequence σ generates k sub-tours. Let u_j and v_j be the no-wait production and delivery start times of sub-tour σ_j serving customers $\sigma(1_j)$, $\sigma(2_j)$, ..., $\sigma(i_j)$. If plant idle time exists between the production start times corresponding to sub-tours $j-1$ and j, then it may be possible to start the production for sub-tour j earlier resulting in an earlier delivery start time. Let u'_j, v'_j be the respective compressed production and delivery start times for the sub-tour j where $j=1, 2, 3, ..., k$. For example, u'_1, v'_1 are the compressed production and delivery start times of sub-tour 1 whereas u_1, v_1 are the no-wait production and delivery start time of sub-tour 1. Let f_j be the delivery finish time of sub-tour j. At time f_j, the truck assigned to sub-tour j returns to the plant after delivery.

4.2.1. Algorithm-Compress

Algorithm-Compress compresses the makespan obtained in a no-wait schedule. Given below is the Algorithm-Compress procedure when $s \ge 1$ trucks are used in a no-wait schedule with k sub-tours.

Step 1: Obtain a no-wait schedule for the selected route using Eq. (21). Set $v'_1=v_1$, $v'_2=v_2$, ..., $v_s=v_s$, and $u'_1=u_1$, $u'_2=u_2$, ..., $u'_s=u_s$.

Step 2: Start compression at $j=s+1$. If $u_{s+1}-v_s > 0$, compress u_{s+1} by x_{s+1}, where $x_{s+1} = \min\,[B-(T_{\sigma_j}-\tau_{\sigma(i_{s+1}),0}),\ (u_{s+1}-v_s)]$. Set $u'_{s+1}=u_{s+1}-x_{s+1}$ and $v'_{s+1}=v_{s+1}$. Calculate the no-wait schedule for sub-tours $j=s+2$, $s+3$, ..., k, using the new values of u'_{s+1} and v'_{s+1}.

Step 3: Increase the value of j sequentially and apply *Step 2.* Stop when $j=k+1$. Set the makespan f, for the selected route to $\max_{1\le j\le k}(v'_j+T_{\sigma_j})$.

If s trucks are used in IPDSP, calculating the no-wait production start time of the jth sub-tour requires that all s sub-tours in progress be monitored. If there are $k \le n$ sub-tours, then calculation of the no-wait schedule has a time complexity of $O(s\,k)$. In the worst case, there can be n sub-tours and n trucks for a given sequence; thus, calculation of a no-wait schedule has a complexity of $O(n^2)$. In Algorithm-Compress, $(n-s)$ no-wait schedules have to be calculated. Therefore, Algorithm-Compress has a worst case complexity of $O(n^3)$.

Compressing the no-wait schedule is an intermediate step in the algorithm that is designed to reduce the makespan but can have the practical effect of making an infeasible no-wait schedule feasible. Consider the following example to illustrate this process based on capacity and lifetime constraints with $n = 10$, $s = 21$, and $B = 17$. A sequence has been grouped into six sub-tours as illustrated in Table 2 and the no-wait schedule developed and illustrated in Fig. 1.

Notice that the no-wait schedule has a makespan of 30 time units but before starting the production for the third sub-tour, the plant has an idle time of 2 time units. Since the delivery time of sub-tour 3 is 2 time units, the production start time of sub-tour 3 can be advanced by a maximum of $B-2=15$ time units. Since the plant has available idle time of 2 time units, u_3 is compressed by 2 time units and $u_3'=5$. With u_3' now defined, a new no-wait schedule can be calculated for u_j, $j=4, 5, 6$ as shown in Fig. 2.

This can have significant impacts. First, if the planning horizon had been 29 time units, the compression created a feasible no-wait schedule from an infeasible one. Second, note that the trucks delivering sub-tours σ_5 and σ_6 switched. For completeness, the algorithm would perform a second compression on u_5 that would result in the schedule illustrate in Fig. 3. This compression does not reduce the makespan although it does advances production.

5. Lower bounds on the total cost of the IPDSP

The evaluation of heuristic results for large problems is done using a lower bound on the minimum cost. In this section, two lower bounds are established for the total cost TC of the IPDSP. A lower bound to IPDSP can be written as the sum of two terms; (1) the lower bound on the fixed hiring cost of trucks (2) the lower bound on the transportation cost. In Section 7, the maximum of

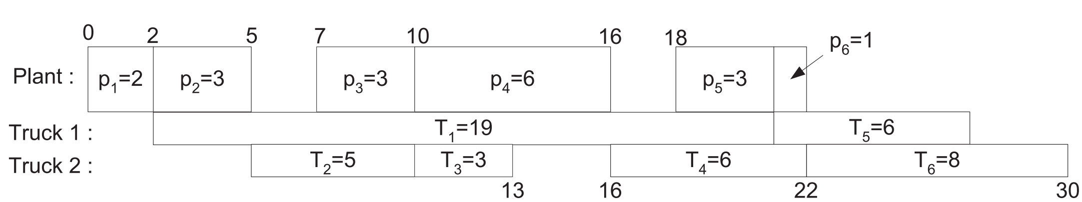

Fig. 1. No-wait schedule for compression example.

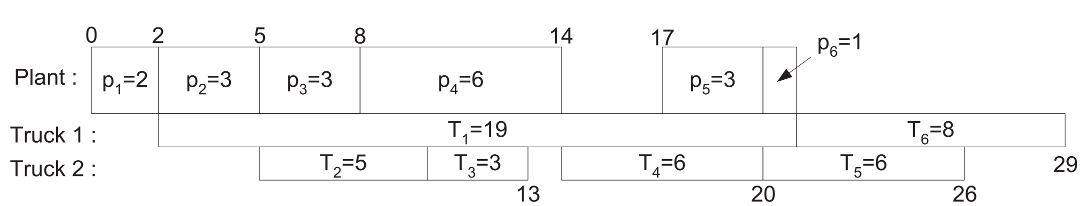

Fig. 2. No-wait schedule after the first compression.

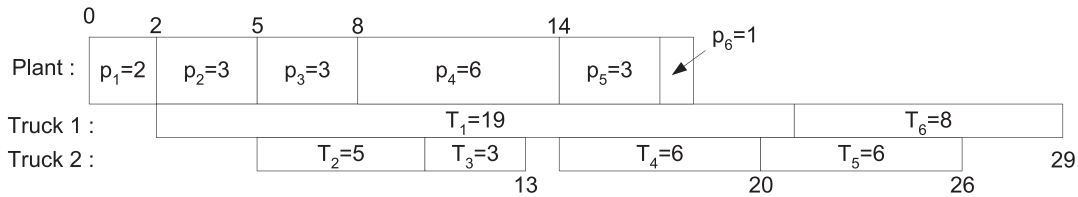

Fig. 3. No-wait schedule after the second compression.

the two lower bounds is used to evaluate the results obtained from the proposed heuristics for IPDSP.

Theorem 2. Assume the following are known: (1) an IPDSP with n customers, each with demand q_i (i=1, 2, 3, ..., n), (2) a matrix of travel times between customers i and j of $\tau_{i,j}$, $0 \le i, j \le n$, where 0 denotes the plant, (3) a fixed production rate r, (4) a fixed cost of hiring a truck F, and (5) the truck's capacity C. Let

$$g = \left\lfloor \frac{\sum_{i=1}^{n} q_i}{C} \right\rfloor + 1,$$

and G be the set of $(g-1)$ closest customers to the plant excluding the furthest customer from the plant. H is the given planning horizon. If each truck travels at a constant speed 1 then the lower bound LB_1 on TC of IPDSP is

$$TC \ge 2\max_{1\le i\le n}\{\tau_{0,i}\} + 2\sum_{i\in G}\tau_{0,i} + F\left\lceil \frac{\frac{1}{r}\min\limits_{1\le i\le n} q_i + \left\{2\max\limits_{1\le i\le n}\{\tau_{0,i}\} + 2\sum_{i\in G}\tau_{0,i}\right\}}{H} \right\rceil$$

Proof. A lower bound on the number of sub-tours required to satisfy the total demand is$g$. One sub-tour delivers to the farthest customer, possibly including some others, so it requires at least $2\max_{1\le i\le n}\{\tau_{0,i}\}$ time units. The other $g-1$ sub-tours satisfy the demand of all other customers at least in $2\sum_{i\in G}\tau_{0,i}$. The term $\frac{1}{r}\min_{1\le i\le n} q_i$determines minimum truck waiting time at the plant to start the first sub-tour. The minimum truck wait time is equivalent to the production time required to satisfy the smallest customer demand. Thus, the term within the ceilings gives the lower bound on the number of trucks required to hire in order to satisfy the total demand within the planning horizon H. The lower bound on the hiring cost is obtained by multiplying the number of trucks by the fixed cost F. The sum of two terms $2\max_{1\le i\le n}\{\tau_{0,i}\} + 2\sum_{i\in G}\tau_{0,i}$ gives a lower bound on the travel cost. ■

The second lower bound to IPDSP is based on a lower bound on the classical bin packing problem given in Bourjolly and Rebettez (2005). They observed that no two customers whose demand is greater than half of the truck's capacity Ccan be served in one sub-tour and, at most, two customers whose demand is between $C/2$ and $C/3$ can be served in one sub-tour. More formally, when q_i is the demand of customer i, let $A = \{i: \ q_i > \frac{C}{2}\}$, $G = \{j: \ \frac{C}{2} \ge q_j > \frac{C}{3}\}$ and $K = \{k: \ q_k \le \frac{C}{3}\}$: then, only one customer in A can be served in one sub-tour, no more than two customers in G can be served in one sub-tour and at least three customers in K can be served in one sub-tour. The perishable constraint of IPDSP is incorporated into the lower bound as explained below.

To begin, the number of sub-tours must be greater than or equal to $|A|$. A customer $i \in G$ can be served on the same sub-tour with a customer $j \in A$, if $q_i + q_j \le C$ and $\tau_{0,i} + \tau_{i,j} \le B$ or $\tau_{0,j} + \tau_{j,i} \le B$. Let $\bar{h}$ be the number of customers in G that are not included in a sub-tour with a customer in A. Then $\lceil \frac{\bar{h}}{2} \rceil$ is the smallest number of additional sub-tours that must be made assuming no perishable constraint. The number of sub-tours obtained ignoring the perishable constraint is always less than or equal to that obtained considering the perishable constraint. Let $G^{\bar{A}}$ be the set of customers that cannot be included in a sub-tour with a customer in A and $g(q_i)$ be the contribution to the lower bound from customers in K. Using the notation in Bourjolly and Rebettez (2005), let $W(a, b)=\{i \in N\colon a < q_i \le b\}$ be the set of customers having demand larger than a but smaller than or equal to b, and $\bar{W}(a, b) = \{i \in N: a \le q_i \le b\}$be the set of customers having demand larger than or equal to a but smaller than or equal to b. Then, the minimum number of sub-tours, $T_{\min}$, required to satisfy the total demand is given by $|A| + \lceil \frac{\bar{h}}{2} \rceil + g(q_i)$, where

$$g(q_i) = \max_{0\le q_i\le C/3}\left\{ \max\left(0, \left\lceil \frac{\sum_{i\in\bar{W}(q_i, C-q_i)} q_i}{C} - |A| - \left\lceil \frac{\bar{h}}{2} \right\rceil \right\rceil \right) \right\}$$

Theorem 3. Given an IPDSP with n customers having demand q_i, i=1, 2, 3, ..., n and a matrix of travel times $\tau_{i,j}$, $0 \le i, j \le n$, where 0 denotes the plant, and a fixed production rate r, a fixed cost of hiring F, and the truck capacity C. Let Hbe the given planning horizon, each truck travels at a constant speed of 1 unit distance per unit time, and there is a unit travel cost per unit distance. Then, the lower bound, LB_2 on TC of IPDSP is

$$TC \ge 2\left(\sum_{f\in A}\tau_{0,f} + \sum_{i=1}^{\lceil \bar{h}/2\rceil}\tau_{0,g_{2i}} + \sum_{i=1}^{g(q_i)}\tau_{0,k_{3i}}\right) + F\left\lceil \frac{\frac{1}{r}\min\limits_{1\le i\le n} q_i + 2\left(\sum_{f\in A}\tau_{0,f} + \sum_{i=1}^{\lceil \bar{h}/2\rceil}\tau_{0,g_{2i}} + \sum_{i=1}^{g(q_i)}\tau_{0,k_{3i}}\right)}{H} \right\rceil$$

Proof. Using the same notation as in the proof of LB1, the first term is the lower bound on the transportation cost of the IPDSP. The second term is the lower bound on the fixed cost of the minimum number of trucks required to meet the total demand within the planning horizon. In order to form LB_2, distances are first calculated and since no two customers in A can be served in the same sub-tour, a distance of at least $2\sum_{f\in A}\tau_{0,f}$ has to be traveled. Next, the elements of $G^{\bar{A}}$ are sorted in ascending order of the distance from the plant and the ith element of the sorted list $G^{\bar{A}}$ be g_i. Because at most two customers in $G^{\bar{A}}$ can be served in one sub-tour, a distance of at least $2\sum_{i=1}^{\lceil \bar{h}/2\rceil}\tau_{0,g_{2i}}$, $i \in G^{\bar{A}}$, should be traveled in $\lceil \frac{\bar{h}}{2} \rceil$ sub-tours. Then, the elements of K are sorted in ascending order with the ith element in K denoted by k_i. Since at least three customers in K can be served in one sub-tour, a distance of at least $2\sum_{i=1}^{g(q_i)}\tau_{0,k_{3i}}$, $i \in K$, should be traveled in $g(q_i)$ sub-tours. Hence, $2(\sum_{f\in A}\tau_{0,f} + \sum_{i=1}^{\lceil \bar{h}/2\rceil}\tau_{0,g_{2i}} + \sum_{i=1}^{g(q_i)}\tau_{0,k_{3i}})$ gives a lower bound on the travel cost in IPDSP. A lower bound on the travel time is obtained by adding the minimum truck wait time at the plant to start the first sub-tour, $\frac{1}{r}\min_{1\le i\le n} q_i$, and the lower bound on the total travel time obtained above. ■

6. Evolutionary approaches to IPDSP

Two heuristics are now provided to find approximate solutions to IPDSP that use genetic algorithms (GAs) (Holland, 1975). This approach seeks to mimic biological evolutionary processes and has been successfully used in a variety of complex optimization problems. This research only uses well-known ideas in GAs; hence, only a very brief description is provided here and readers interested in details are encouraged to consult (Goldberg, 1989).

6.1. IPDSP-GA

In this heuristic, a chromosome (σ) is a sequence of customers – each customer is a gene. The fitness value is the total cost and is calculated using a process that takes advantage of theories previously developed in this paper. First, the chromosome is partitioned into feasible sub-tours using the optimal tour partitioning algorithm in Section 4.1. Then, a no-wait schedule is calculated using Theorem 1 to find the makespan with one truck and the makespan is improved by Algorithm-compress. If the compressed makespan is smaller than the planning horizon then the total cost is calculated with one truck. Otherwise, one more truck is added to the fleet and above procedure is repeated until a feasible solution is obtained and the total cost is the fitness.

The first generation chromosomes are randomly generated. The genetic operators used here are standard two-point crossover with a simple roulette wheel selection process where the probabilities are based on the relative fitness and swap mutation. Elite reproduction is used to control the generation to generation progression. The stopping criterion used is a fixed number of generations. The step-by-step procedure is now presented.

Step 1: Establish the initial population of chromosomes by randomly assigning sequences that connect the plant to all n customers. Go to *Step 2*.

Step 2: Divide each sequence into sub-tours using algorithm Split. Set the number of trucks to one, $s=1$.

Step 3: For each chromosome: (i) obtain the no-wait schedule, (ii) compress the no-wait schedule using Algorithm-Compress and (iii) find the compressed makespan. If the compressed makespan is less than or equal to horizon (*H*), calculate the fitness with$s$trucks and go to *Step 4.* If not, and $s<n$set $s=s+1$ and repeat *Step 3.*

Step 4: If all sequences in the current generation have not been considered, go to *Step 3*.

Otherwise go to *Step 5*.

Step 5: Perform crossover and mutation and evaluate the fitness using *Steps 2* and *3*. If all generations have been considered go to *Step 6.* If not, create the next generation and go to *Step 2.*

Step 6: Select the chromosome corresponding to the minimum cost and decode the sequence as well as the number of trucks.

The parameters used in IPDSP-GA were determined by starting with values commonly found in the literature for TSP problems and then using a simple factorial design with 150 chromosomes per generation to explore values around them. The results were that a mutation probability of 30% and a crossover probability of 70% were best and are used in the computational examples. All the problem instances in the computational study were run with a population size of 150 chromosomes and a stopping criterion of 250 generations.

6.2. Memetic algorithm approaches

To enhance the performance of GAs, methods that combine domain-specific local search and evolutionary algorithms have received special attention and are called memetic (MA) or hybrid algorithms (Moscato, 1999; Prins, 2004). In this research, two MA's are proposed to improve the performance of the GA.

6.2.1. IPDSP-MA1

Local search performed in this research consists of pairwise swapping. Let a and b be any pair of genes in the chromosome that was found to produce minimum cost after executing IPDSP-GA and let x and y be the preceding genes a and b, respectively. (Recall, genes are customers.) The pairwise swap is implemented using the following patterns with the fitness function evaluated after each step.

(1) Remove gene a and insert it after gene b
(2) Remove gene a and gene x and insert them after gene $b$in the same order
(3) Remove genes x and gene a and insert them after gene b in the same order
(4) Swap gene a with gene b
(5) Swap gene x with gene b
(6) Swap genes a with gene b, and gene x with gene y

Local search for a given gene pair is continued until an improvement in the solution is found or all six moves are implemented. If an improvement is found, then the solution chromosome is updated and the search for that pair is discontinued. If not the move is omitted and the next move is implemented. Local search is continued until all possible genes pairs are subjected to the search.

6.2.2. IPDSP-MA2

MA2 combines local search with careful management of the chromosomes in the population. GA portion of IPDSP-MA2 begins as in IPDSP-GA with the initial population being randomly generated; however, the population is searched and “clones” are replaced. A clone is defined as a chromosome that has the same fitness value (e.g. total cost) as another. As before, two-point crossover is used with a roulette wheel selection to select the parents but before the children are added to the next generation's population, they are checked to ensure they are not clones of chromosomes already in the population. In each generation the child chromosome with the highest fitness is selected for a local search using the six moves outlined applied in IPDSP-MA1. Similar to IPDSP-GA, IPDSP-MA2 uses a predetermined number of generations as the stopping criteria and the chromosome with smallest fitness value in the last generation is selected as the best solution.

In IPDSP-MA2, population management is important so that no clones are created. The local search allows a much smaller population to be used which saves computational time; however, exactly how small is a critical parameter so that IPDSP-MA2 converges to a good solution within reasonable computational time. A Kruskal–Wallis test for median total cost with multiple comparisons is conducted to determine population size. A random problem was used for the test and the population size was varied from 2 to 18 sequences. The results were that 8 performed well and this size was used in the examples below.

7. Computational study

This section consists of two computational studies; (1) a comparison of optimal solution from CPLEX with proposed heuristics using a set of test problems, (2) a detailed computational

Table 3
Problem set 1, $n=20$, $H=550$, $B=50$, $C=60$.

Problem	Lower bound	IPDSP-GA		IPDSP-MA1		IPDSP-MA2	
		Solution	% Gap above the lower bound	Solution	% Gap above the lower bound	Solution	% Gap above the lower bound
1	742	1191	60.47	1190	60.44	1200	61.71
2	706	1110	57.28	1108	56.99	1119	58.55
3	1020	1267	24.18	1266	24.14	1281	25.61
4	1024	1169	14.19	1169	14.17	1191	16.33
5	446	876	96.49	821	84.04	868	94.51
6	640	1064	66.33	1064	66.23	1071	67.39
7	952	1130	18.70	1130	18.70	1145	20.25
8	630	1026	62.89	1023	62.31	1033	63.95
9	1162	1293	11.27	1293	11.27	1294	11.39
10	1142	1187	3.96	1187	3.95	1199	4.97
11	1166	1282	9.97	1282	9.96	1288	10.49
12	1160	1214	4.69	1214	4.69	1223	5.43
13	1068	1451	35.89	1450	35.75	1566	46.63
14	1146	1254	9.42	1254	9.42	1272	11.03
15	1248	1420	13.78	1420	13.76	1419	13.71
16	720	1156	60.56	1153	60.08	1173	62.97
17	980	1140	16.35	1139	16.20	1144	16.76
18	714	1234	72.81	1233	72.72	1248	74.83
19	660	1150	74.21	1146	73.68	1166	76.61
20	746	1093	46.47	1089	45.95	1098	47.20
21	1090	1254	15.05	1254	15.05	1256	15.26
22	1072	1211	12.94	1211	12.93	1215	13.31
23	722	1016	40.66	1015	40.61	1018	40.96
24	1130	1849	63.63	1849	63.63	1849	63.65
25	1228	1344	9.45	1344	9.45	1351	10.01
26	1136	1312	15.54	1307	15.02	1326	16.72
27	1150	1347	17.15	1347	17.14	1361	18.36
28	1058	1392	31.58	1384	30.79	1422	34.41
29	734	1118	52.29	1117	52.17	1133	54.32
30	1028	1138	10.71	1138	10.70	1151	11.92

study on larger problems to compare the performance of proposed heuristics.

To explore the aforementioned heuristics and illustrate the types of information that can be gleaned from the model, a number of tests have been conducted on randomly generated problem sets. Test problems have been generated with the number of customers ranging from 4 to 40 customers. The number of customers is selected and then the demand is randomly assigned between 1 and the truck capacity. Random locations are assigned to each customer in the x–y plane but within a distance from the plant that the direct travel time is less than the effective lifetime of the product. For each problem, the results obtained from the heuristics are compared with each other, with the lower bound and with the optimal solution for those problems that could be solved using CPLEX.

Table 1 compares the optimal solution obtained by CPLEX, if found, and the heuristic solutions. It should be noted that smaller problems are not tested with IPDSP-MA2 due to the difficulty arising from population management. Recall, IPDSP-MA2 maintains a well-spaced population that does not contain clones and when the problem size becomes small, there aren't enough different solutions. The IPDSP-MA1 is run for 30 random starts for each test problem and average values are given in Table 1. For the 4 customer problem, IPDSP-MA1 finds the optimal solution. For problems with 6 and 7 customers, both IPDSP-MA1 and CPLEX finds the same solution; however, it is possible this solution is not optimal because CPLEX did not terminate the branch-and-bound method based on the % gap between the current best node (best lower bound) and the current best integer solution. For problems with 10 or more customers, CPLEX never found a feasible solution within hours of computation time and its use was abandoned for this more realistic situations. For comparison, the average time it takes IPDSP-MA1 and IPDSP-MA2 to find the recorded solution is provided for comparison.

Using the method described above, 30 problem instances are created for the detailed computational study with 20, 30 and 40 customers and each was solved using IPDSP-GA, IPDSP-MA1, and IPDSP-MA2. For each problem instance, each heuristic is run for 300 replications and the results displayed in Tables 3–5. The replications for each problem were performed on the Clemson University high throughput computing grid consisting of about 1400 personal computers (http://citi.clemson.edu/htc) with each instance requiring between 6 and 7 minutes. The Tables allow the heuristic to be compared between one another as well as against the highest lower bound on the total cost of the IPDSP for each problem. The metric used to report the deviation of heuristic results from the lower bound is:

$$\% \text{ Gap above lower bound} = \frac{\text{Heuristic result} - \text{Lower bound}}{\text{Lower bound}}$$

The number of vehicles used in the best solutions varied for the different replications within a given number of customers and, occasionally, between different heuristics. For the 20 customer cases, 2 or 3 vehicles were required and, looking across all replications of all heuristics, 2 vehicles were required well over three fourths of the time. There was no obvious pattern to the routes each vehicle took but, again looking at all replications across all heuristics, a common "type" of solution was for each vehicle to have between 8 and 12 different routes with one vehicle handling 3 or 4 more customers than the other. The 30 customer cases also required 2 or 3 vehicles but, here, 3 vehicles were needed over two thirds of the time. Finally, for 40 customers, 2 vehicles were required in a few cases but 3 and 4 vehicles are required in the vast majority with about an equal number of replications for each. Clearly, the parameters of the numerical example dictate the number of vehicles to a large degree; however, it is important to note that in nearly all cases, the different heuristics used the same number of vehicles.

Table 4
Problem set 2, $n = 30$, $H = 550$, $B = 50$, $C = 60$.

Problem	Lower bound	IPDSP-GA		IPDSP-MA1		IPDSP-MA2	
		Solution	% Gap above the lower bound	Solution	% Gap above the lower bound	Solution	% Gap above the lower bound
1	1052	1800	71.12	1790	70.12	1784	69.61
2	1428	1760	23.22	1758	23.09	1754	22.85
3	1086	1301	19.82	1294	19.14	1288	18.57
4	1866	2048	9.74	2046	9.65	2052	9.97
5	1398	1664	19.02	1651	18.07	1678	20.04
6	1768	1964	11.08	1961	10.91	1964	11.07
7	1262	1773	40.46	1769	40.19	1771	40.32
8	1260	1777	41.03	1772	40.64	1772	40.61
9	1220	1350	10.69	1349	10.56	1350	10.67
10	1168	1385	18.55	1381	18.20	1378	17.96
11	1262	1794	42.18	1792	42.03	1796	42.33
12	1338	1766	32.01	1765	31.89	1766	32.01
13	1270	1705	34.22	1702	34.03	1701	33.91
14	1150	1714	49.05	1710	48.67	1704	48.20
15	1790	1957	9.32	1956	9.25	1956	9.30
16	1126	1300	15.43	1294	14.88	1291	14.66
17	1334	1829	37.11	1826	36.89	1826	36.88
18	976	1413	44.82	1403	43.77	1399	43.34
19	1446	1782	23.26	1779	22.99	1776	22.79
20	1296	1734	33.81	1730	33.45	1731	33.54
21	1248	1717	37.60	1712	37.21	1716	37.50
22	970	1341	38.24	1334	37.54	1334	37.56
23	1172	1698	44.90	1691	44.28	1688	44.02
24	1462	1856	26.97	1855	26.89	1857	27.03
25	1154	1376	19.28	1374	19.08	1370	18.73
26	1090	1638	50.30	1573	44.31	1579	44.84
27	966	1283	32.84	1277	32.17	1268	31.23
28	1280	1786	39.55	1778	38.92	1773	38.48
29	1200	1678	39.87	1672	39.37	1683	40.25
30	1314	1451	10.40	1445	9.95	1444	9.92

Table 5
Problem set 3, $n = 40$, $H = 550$, $B = 50$, $C = 60$.

Problem	Lower bound	IPDSP-GA		IPDSP-MA1		IPDSP-MA2	
		Solution	% Gap above the lower bound	Solution	% Gap above the lower bound	Solution	% Gap above the lower bound
1	2032	2497	22.87	2495	22.80	2500	23.03
2	1738	2368	36.25	2327	33.92	2353	35.39
3	1424	1983	39.27	1979	38.95	1975	38.67
4	1480	1963	32.64	1955	32.06	1947	31.55
5	1188	1801	51.61	1782	50.01	1770	49.03
6	1468	1919	30.71	1911	30.18	1903	29.63
7	1834	2024	10.34	2020	10.15	2016	9.93
8	2034	2199	8.13	2175	6.91	2165	6.43
9	1712	1996	16.58	1994	16.48	1989	16.19
10	1718	2040	18.77	2033	18.36	2032	18.30
11	1924	2078	8.02	2072	7.68	2071	7.64
12	1754	2095	19.44	2088	19.05	2087	18.98
13	1916	2566	33.94	2565	33.87	2564	33.80
14	1262	1968	55.91	1943	53.97	1935	53.35
15	1456	2180	49.72	2144	47.27	2128	46.18
16	1842	2117	14.93	2106	14.35	2105	14.27
17	1368	1981	44.84	1967	43.82	1952	42.67
18	1108	1859	67.76	1842	66.20	1833	65.40
19	1328	1832	37.98	1828	37.63	1825	37.42
20	1346	1830	35.99	1827	35.74	1819	35.12
21	1390	2039	46.71	2019	45.22	2010	44.58
22	1910	2490	30.36	2484	30.03	2475	29.56
23	2158	2576	19.39	2572	19.18	2570	19.08
24	1756	2029	15.56	2027	15.43	2018	14.93
25	1390	2074	49.19	2060	48.22	2042	46.93
26	1244	1908	53.34	1889	51.88	1873	50.59
27	2056	2485	20.87	2473	20.30	2473	20.30
28	1742	2000	14.81	1987	14.05	1978	13.57
29	1880	2355	25.25	2301	22.41	2251	19.76
30	1484	2061	38.85	2050	38.15	2030	36.83

Table 6
Ranked means for multiple comparisons.

Problem size	Ranked means		
	$\bar{R}_{IPDSP-GA}$	$\bar{R}_{IPDSP-MA1}$	$\bar{R}_{IPDSP-MA2}$
20	13,304	13,199	13,998
30	13,771	13,330	13,401
40	13,920	13,433	13,148

Table 7
Difference between ranked means for multiple comparisons.

Problem size	Difference in ranked means		
	$\bar{R}_{IPDSP-GA} - \bar{R}_{IPDSP-MA1}$	$\bar{R}_{IPDSP-GA} - \bar{R}_{IPDSP-MA2}$	$\bar{R}_{IPDSP-MA1} - \bar{R}_{IPDSP-MA2}$
20	105	694	799
30	441	370	71
40	486	771	285

Three heuristics are statistically compared for the median of the percentage gap for each problem. Recall that each problem size has 30 randomly generated problems and each problem is tested with 300 random starts creating 9000 solutions for each heuristic. Even though each of the 30 problems of any given size are randomly generated with the same parameter values, including product lifetime, truck capacity, plant production rate and truck speed, the demands and customer locations are different from one problem to another. Therefore, statistical analysis is conducted on the percentage gap above lower bound of all problems of a given size. Initially, an Anderson–Darling test was conducted to check the normality of each population with 27,000 data points with mean of 34.56%, standard deviation 25.79% and concluded that the data does not follow a normal distribution with a p-value <0.005. This suggests that normality assumption is not satisfied and non-parametric tests are required to compare the heuristics used in this research. The Kruskal–Wallis test for median is conducted for all problem sizes and with a lower p-value of 0.000 we concluded that there is enough statistical evidence to reject the null hypothesis that all medians are statistically similar. In order to find out which medians are different multiple comparison tests are conducted on ranked means for each problem size. The test statistic for Fisher's least significant difference between samples i and j ($LSD_{i\,j}$) is compared to the difference of ranked means between samples i and j denoted by $\bar{R}_i$ and $\bar{R}_j$, respectively. Given the total sample size N, individual sample sizes n_i, n_j and a level of significance α, the least significant difference between samples i and j is calculated by $LSD_{i\,j} = Z_{\alpha/2}\sqrt{(\frac{N(N+1)}{12})(\frac{1}{n_i}+\frac{1}{n_j})}$. Two treatments i and j are declared different if $|\bar{R}_i - \bar{R}_j| > LSD_{i\,j}$. In this computational study, each treatment has a sample size of 9000 resulting $LSD_{i\,j}=228$. Tables 6 and 7 show the ranked means and pairwise differences of ranked means, respectively.

As shown in Table 7, for problems with 20 customers $\bar{R}_{IPDSP-GA} - \bar{R}_{IPDSP-MA1} = 105$ which is lower than 228, hence IPDSP-GA and IPDSP-MA1 are not statistically different. However, IPDSP-GA and IPDSP-MA2 are significantly different from each other since $\bar{R}_{IPDSP-GA} - \bar{R}_{IPDSP-MA2} > 228$. Similarly, IPDSP-MA1 and IPDSP-MA2 are also significantly different. The data shows that the highest median is reported under IPDSP-MA2. Even though IPDSP-GA and IPDSP-MA1 are not statistically different, IPDSP-MA1 always has a similar or smaller median than IPDSP-GA. Thus IPDSP-MA1 is selected as the best heuristic for 20 customer problems. A similar analysis for 30 customer problems shows that IPDSP-GA is significantly different from both IPDSP-MA1 and IPDSP-MA2 while IPDSP-MA1 and IPDSP-MA2 are not significantly different. IPDSP-MA1 has a higher median than IPDSP-MA2. Thus IPDSP-MA2 is selected as the best heuristic for 30 customer problems. As shown in Table 7, for 40 customer problems, all i and j pairs of multiple comparisons of ranked means are greater than $LSD_{i\,j}=228$. We conclude that each treatment is significantly different from other and the treatment with minimum median IPDSP-MA2 is selected as the best for 40 customer problems.

8. Conclusions and future research

This paper addresses a practical problem in production and distribution in which the product has a limited lifetime. A mathematical programming model was developed and then resolved. Almost trivially small problems could be solved optimally but all others required heuristic approaches since the IPDSP is NP-hard. A lower bound was calculated to which heuristic solutions for example problems could be compared and the heuristics were also compared to each other. As the number of customers grows, different heuristics appears to perform better based on the % gap above the lower bound. Based on the examples tested, it appears that the metaheuristic IPDSP-MA2 which includes careful control of the chromosomes in the population along with local search performs best for larger problems like 40 customers.

For future research there are a number of variations of this problem that could be interesting but would certainly lead to more complexity and, hence, a premium on solution techniques. One is to extend the model to include trucks with variable capacity and costs that reflect this. This could likely be accommodated within the general framework of the current model but would yield results that are more closely aligned with practice. Another more realistic modification would be to include more than one product and have each with a different lifetime. This introduces a new level of complication including fundamentally different delivery characteristics because each customer could have multiple products delivered by the same or different trucks to different decisions regarding production. For example, if two products are considered a customer could require both products that are delivered by two trucks or one truck in one or two deliveries. Production scheduling at the plant is also changed because production can be done at the same plant with or without setup, or two different plants located at the same place.

Finally, we have observed that a feature of the real problem is that multiple plants are frequently located in an area and as the company adds customers a nontrivial number can be services by more than one plant. Adding this fundamentally different idea to the model presented here is important to consider and when addressed, the extensions mentioned will further enhance the realism.

References

Beasley, J. (1983). Route first-cluster second methods for vehicle routing. *Omega, 11*, 403–408.
Bourjolly, J., & Rebettez, V. (2005). An analysis of lower bound procedures for the bin packing problem. *Computers & Operations Research, 32*, 395–405.
Chang, Y.-C., & Lee, C.-Y. (2003). Machine scheduling with job delivery coordination. *European Journal of Operational Research, 158*, 470–487.
Chen, H.-K., Hsueh, C.-F., & Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers and Operations Research, 36*, 2311–2319.
Chen, Z. L. (2010). Integrated production and outbound scheduling: review and extensions. *Operations Research, 58*, 130–148.
Chen, Z. L., & Pundoor, G. (2006). Order assignment and scheduling in a supply chain. *Operations Research, 54*, 555–572.
Clarke, G., & Wright, W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*, 568–581.
Garcia, J. M., Lozano, S., & Canca, D. (2004). Coordinated scheduling of production and delivery from multiple plants. *Robotics and Computer Integrated Manufacturing, 20*, 191–198.

Garcia, J. M., & Lozano, S. (2004). Production and vehicle scheduling for ready-mix operations. *Computers and Industrial Engineering, 46*, 803–816.
Geismar, H. N., Laporte, G., Lei, L., & Sriskandarajah, C. (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*, 21–31.
Goldberg, D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional.
Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor: University of Michigan Press.
Hsu, C.-I., Hung, S.-F., & Li, H.-C. (2007). Vehicle routing problem with time-windows for perishable food delivery. *Journal of Food Engineering, 80*, 465–475.
Hurter, A. P., & Van Buer, M. G. (1996). The newspaper production/distribution problem. *Journal of Business Logistics, 17*, 85–107.
King, R., & Lowe, R. (1980). Coordinating decisions for increased profits. *Interfaces, 10*, 4–19.
Lee, J., Kim, B., Johnson, A., & Lee, K. (2014). The nuclear medicine production and delivery problem. *European Journal of Operational Research, 236*, 461–472.
Li, C.-L., Vairaktarakis, G., & Lee, C.-Y. (2005). Machine scheduling with deliveries to multiple customer locations. *European Journal of Operational Research, 164*, 39–51.
Miller, C. E., Tucker, A. W., & Zemlin, R. A. (1960). Integer programming formulations and traveling salesman problems. *Journal of the ACM, 7*, 326–329.
Moscato, P. (1999). Memetic algorithms: A short introduction. In D Corne, M Dorigo, & F Glover (Eds.), *New ideas in optimization* (pp. 219–234). New York: McGraw-Hill.
Osvald, A., & Stirn, L. Z. (2008). A vehicle routing algorithm for the distribution of fresh vegetables and similar perishable food. *Journal of Food Engineering, 85*, 285–295.
Prins, C. A. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research, 31*, 1985–2002.