We consider a capacitated multi-stage operations scheduling problem encountered in a real life supply chain for disaster relief, consisting of component suppliers, a packaging contractor, local distribution centers, and many customer demand points. Each customer order has a due date and can be fulfilled by either existing inventory at distribution centers or a new assembly operation at the packaging contractor. The problem is to find a production and distribution schedule to minimize the total tardiness in fulfilling the customer orders, which is computationally a difficult problem, especially when supply chain lead time is involved. In this paper, we present a polynomial time algorithm that solves this problem optimally when the order sizes are identical. The proposed solution process is based upon decomposition, and is demonstrated step-by-step through a numerical example. Future extensions of this study are also discussed.

* Corresponding Author. E-mail address: klee5@york.cuny.edu

I. INTRODUCTION

It is known that the demand side of a supply chain often changes at a pace faster than that of the supply side (CSCMP, 2012), which in turn increases the importance of a highly coordinated production, inventory, and distribution across a multi-echelon supply chain network as it has an immediate impact on customer services. Such a coordinated and integrated operations planning and scheduling is particularly critical during an emergency situation for timely provisions of life-saving supplies to the people in affected areas. The process of coordinating the production, inventory allocation, and on time delivery of life-saving supplies during and after a disaster is also called emergency logistics (http://www.americares.org/).

There has been a significant effort devoted by researchers to emergency logistics during the past two decades. Haghani and Oh (1996) studied a logistics problem encountered in disaster relief management. A logistics network with multi-commodity, multi-modal and time constraints was transformed into a time-space network, upon which a mixed integer programming model was formulated. Two heuristics were then proposed: one resorted to Lagrangian decomposition by exploring the network structure property and the other used linear programming relaxation. Barbarosoglu et al. (2002) developed a mathematical model for helicopter mission planning during disaster relief operations. The planning included both tactical decisions and operations decisions. The authors proposed a two-level framework including two mixed integer programming models: the top level covered helicopter fleet determination, helicopter crew assignment and tour numbers determination to minimize the total cost, while the bottom level covered helicopter routing, transportation and refueling to minimize the makespan. An iterative
coordination process was used to generate non-dominant solutions for the multi-objective problem and then an interactive procedure was proposed for the decision-making to choose the best feasible solution. Özdamar et al. (2004) studied the dynamic time-dependent transportation problem that was solved repetitively at given time intervals during an emergency logistic planning. Their model integrated the multi-commodity network flow problem and the vehicle routing problem (VRP). The authors also discussed the differences between VRP in a regular scenario and that in an emergency situation. Vehicles were treated as commodities in their study. Therefore the problem was modeled as a multi-period multi-commodity network flow problem with arc capacities as variables instead of parameters (imposed by the capacity of vehicles). A Lagrangian decomposition based iterative methodology was proposed to solve the problem. Altay and Green (2006) discussed the potential applications of operations research (OR) in the field of disaster operations management. They reviewed those related literature and categorized the results upon four programmatic phases in emergency management: mitigation, preparedness, response and recovery. According to their report, a majority of papers were on mitigation while fewer were on the remaining categories. Sheu (2007) worked on relief distribution in the crucial rescue periods after a disaster, and proposed a three-tier network with relief suppliers, urgent relief distribution centers and relief-demanding areas. Affected areas were grouped according to the urgency extent, where the urgency attribute was measured by a fuzzy method. A two-stage demand-driven multi-objective (demand fill rate and time-varying distribution cost) dynamic programming optimization model was proposed with one stage for the distribution between relief suppliers and distribution centers and the other for the distribution between distribution centers and affected areas. Yi and Özdamar (2007) considered the evacuation planning of wounded people and location selection for temporary medical centers. By further extending the model of Özdamar et al. (2004), the authors treated vehicles as commodities to avoid individual tracking of each vehicle. After the initial solution was obtained, they then solved a system of linear equations to extract from the optimal solution an exact schedule for each vehicle in pseudo-polynomial time. The location problem was also coped implicitly by allocating optimal service rate to medical centers. The two-stage procedure was shown to be computationally more efficient comparing to a VRP based single-stage formulation. Caunhye et al. (2012) conducted a comprehensive review on the applications of operations research to emergency logistics. The scope of their review included pre-disaster planning about facility location, stock pre-position, and evacuation, and short-term post-disaster planning about resource allocation, commodity flow, and the combination of both. They also categorized the literature into those related to facility location and those related to relief distribution and casualty transportation. They pointed out the lack of models for explicit response time minimization because of the potential complexity for tracking response time, and that computation efficiency is a main reason for the absence of comprehensive operations model for emergency logistics. More recently, Lee et al. (2013a) presented a structural analysis for an emergency logistics optimization problem involving both renewable resources (medical teams that perform treatments to the patients in shelters after a disaster) and non-renewable resources (medical and emergency supplies).

Note that, on the other hand, there has been an increasing amount of work on supply chain operations scheduling in recent years. Representative work in this growing area can be found in Li et al. (2005), Dawande et al. (2006), Lei et al. (2006), Chen and Hall (2007), Li and Ou (2007), Stecke and Zhao (2007), Geismar et al. (2008), Hall et al. (2008), Lei et al. (2009), Wang and Cheng (2009), Chen (2010), Hall and Liu (2010), Mula et al. (2010), Zhong et al. (2010), and Wang and Lei (2012). An excellent review on the model structures, optimality properties, computational tractability and solution methodologies can be found in the work by Chen (2010). Several studies reviewed by Chen (2010) are closely related to our problem. For example, Li et al. (2005) considered the
coordination between a single machine scheduling and routing to minimize the sum of job arrival times, showed that such a two-stage network problem is NP-hard, and then proposed a polynomial algorithm for the case with a fixed number of customers. Li and Ou (2007) studied the scheduling problem encountered in an application where two tasks are finished on two machines at different locations and then shipped to a distribution center for bundling and final shipping to customers. The objective was to minimize the sum of the total shipping cost and the customer waiting cost. Polynomial time heuristics are developed together with a worst case performance analysis.

A major difference in the literature between emergency logistics operations management and supply chain scheduling is that the former focuses more on customer services and timely provision of deliveries without involving the manufacturers, while the later focuses more on total cost reduction through coordination among production, inventory and distribution operations. This difference alters the structure of problem objective functions and consequently the solution methodologies.

The real life operations scheduling problem that motivated our study was encountered from the practice of assembling, packaging, and delivery of rescue packs to support disaster relief effort during and after the hit of natural disasters (e.g., earthquakes, flooding, and hurricanes). In particular, we focus in this study on a four-stage supply chain process (see Figure 1) for a single product (e.g., the standard rescue pack for a particular affected area). The bill-of-material (BOM) for the product is given. For example, each unit of product requires 100 units of Certi-Strip Adhesive Bandages, 50 units of Ammonia Inhalants, 25 units of Certi-Burn Cream, 10 units of Trauma Dressing, and 10 units of medical gloves, etc. Depending on the type of natural disaster and the specific needs for local disaster relief, the BOM composition for the product may vary from case to case (Ferris, 2010).

This supply chain consists of component suppliers, a single packaging contractor (the PC), distribution centers (DCs), and many customer demand points (e.g., hospitals, rescue shelters, or local offices of non-profit organizations). Each customer order is defined by its quantity, due date, and a lateness penalty. During and after a disaster, different locations in an affected area suffered from different levels of damage may request rescue supplies under different levels of urgency, which leads to a heterogeneity in order due dates and late penalties. In this study we assume that all customer orders are of identical size (i.e., the unit size) and each order can be fulfilled by either the existing inventory of DCs or a new assembly operation at the PC. Each DC carries a limited inventory for the product (supplied by various external sources that participate in the disaster relief) and takes a DC-dependent time for handling the orders (e.g., for information processing, order validation, packaging/labeling, waiting for the local vehicles, and loading the shipments, etc.). However, DCs do not perform assembly operations and thus are only responsible for allocating existing inventories to fulfill the customer orders. On the other hand, the PC is capable of assembling-packaging additional supplies of the product when needed. As a contracted manufacturer, the PC only carries an inbound inventory of components to be used in various final products (i.e., various medical packs), instead of an outbound inventory for any particular disaster-dependent finished product. Depending on the amount of orders to be fulfilled, the PC may require its suppliers to send additional component supplies. Whenever this is the case, the PC’s operation will halt until all the new shipments from its suppliers arrive, which further delays the assembly and shipping operations. Note that DCs do not perform assembly operations and are only responsible for allocating their existing inventories to the demand points. Therefore, there are no replenishing flows to DCs from upstream suppliers.

The problem is to assign customer orders to DCs and the PC, which in turn determines whether and how long the PC has to wait for the additional supplies of components from its suppliers, so that the total weighted tardiness in delivering the shipments to the customers is minimized. Furthermore, the shipping times between DCs and customers, and that between
each component supplier and the PC, must be explicitly considered. Orders to be fulfilled by the new assembly/packaging operation must be sequenced at the PC due to the packaging requirement.

**FIGURE 1. A HYPOTHETICAL MULTI-STAGE NETWORK FOR EMERGENCY OPERATIONS**

In general, the inventory allocation plus production sequencing to minimize the total weighted tardiness is a very difficult optimization problem, especially when multi-stage lead/shipping times are involved. In this paper, we propose a search algorithm that finds the optimal operations schedule to this problem in strongly polynomial time when the customer order sizes are identical (e.g., one truck load). Such an algorithm can be used as a subroutine embedded in a heuristic for solving general versions of the problem such as those with multiple PCs in the network or order-dependent bill-of-materials. To our knowledge, the result presented in this paper is among the very few, if not the first one, that efficiently solves a subproblem of emergency logistics optimally.

The remaining part of this paper is organized as follows. In Section 2, a mixed integer programming model for the integrated operations scheduling problem with unit order sizes is introduced. In Section 3, a strongly polynomial-time algorithm and a formal proof of its time complexity are presented. In Section 4, we demonstrate the use of this proposed algorithm in steps through a numerical example. Finally, we conclude the study and discuss its future extensions in Section 5.

**II. A FORMAL PROBLEM DEFINITION**

For a formal definition of the problem described in Section 1, we introduce following notations.

**Components:** Let $N$ be the set of components needed for the product according to the given BOM. Each unit of the final product requires $C_n$ units of component $n$, $n \in N$. Note that if each customer order requires a customer-dependent BOM, then the proposed solution approach must be revised, which shall lead to a different level of time complexity of the solution process (see Lee et al. 2013b).

**Customer demand points:** The set of demand points is denoted by $H$. Each demand point $h$, $h \in H$, places an order together with a due date $d_h$ and a penalty $w_h$ for each time unit of delay, which measure the urgency of the order. Since all the customer order sizes are identical, without
loss of generality, we assume unit size orders.

**Distribution centers (DCs):** The set of DCs in the network is denoted by \( K \), where DC \( k, k \in K \), has a given inventory of the final product, \( I_k \), which can fulfill at most \( I_k \) orders (since order sizes are of unit-size). Let \( a_{kh} \) stand for the time for DC \( k \) to process the order of customer \( h \), and \( \tau_{kh} \) stand for the shipping time from \( k \) to \( h \). Note that, the maximum supply quantity of a DC’s inventory and the lead time from a DC to a customer are constants, while the maximum supply quantity of the PC and the lead time from the PC to a customer depend on whether the suppliers are involved. Thus, when the PC is involved, which in turn introduces the order sequencing at PC together with the lead times for shipping from component suppliers, this is no longer a simple assignment problem.

**The packaging contractor (PC):** The PC takes \( R \) time units to produce (i.e., assemble and package) an order. That is, the production time needed for each order is a constant. The PC carries an inbound inventory \( I_n \) for component \( n, n \in N \), but does not carry any outbound inventory of the final product. Let \( \bar{\tau}_n \) be the shipping time from the PC to customer \( h \). Depending on the orders assigned to the PC, additional supplies of components may need to be shipped in from respective suppliers. When this is the case, the lead time between suppliers and the PC must be considered. Note that the PC will not start its production until sufficient supplies of the required components are ready.

**Suppliers:** There are multiple suppliers for each component \( n \). Let \( J_n \) be the set of suppliers for component \( n \) and \( \hat{\tau}_{nj} \) be the shipping time from the \( j \)-th supplier in \( J_n \) to the PC. Without loss of generality, we assume that \( \hat{\tau}_{nj} \leq \hat{\tau}_{nj+1} \) for all \( j = 1, \ldots, |J_n| - 1 \). Let \( \hat{I}_n \) be the inventory level for component \( n \) at the \( j \)-th supplier in \( J_n \). Let \( J \) be the set of all suppliers, i.e. \( J := \bigcup_{n \in N} J_n \).

Note that, if we know the number of orders to be produced by the PC, then the total required production time (for assembly and packaging operations) is known since the production time for each order is a given constant \( R \). Furthermore, if the starting time of the production (i.e., the time at which all the required component supplies are ready) is given, say \( S \), then the first order in the production sequence will be completed, and depart for the respective customer, at time \( S + R \). Similarly, the second order will be completed and depart at time \( S + 2R \), etc. Let a *time slot* denote a time period of duration \( R \) that is needed for processing an order at the PC. We now introduce the following decision variables:

\[
\begin{align*}
& x_{kh} \quad \text{Binary variable, and } x_{kh} = 1 \text{ if demand point } h \text{ is served by DC } k, \text{ and 0 otherwise, } \forall k \in K, \forall h \in H; \\
& \bar{x}_{kh} \quad \text{Binary variable, and } \bar{x}_{kh} = 1 \text{ if the order of demand point } h \text{ is produced during the } i\text{-th time slot at the PC, and 0 otherwise, } \forall i, 1 \leq i \leq |H|, \forall h \in H; \\
& q_{nj} \quad \text{The flow quantity from the } j\text{-th supplier of component } n \text{ to the PC, } \forall n \in N, \forall j \in J_n; \\
& y_{nj} \quad \text{Binary variable, and } y_{nj} = 1 \text{ if there is a flow quantity from the } j\text{-th supplier of component } n \text{ to the PC, and 0 otherwise, } \forall n \in N, \forall j \in J_n; \\
& Q \quad \text{Total quantity produced (or the total number of orders fulfilled) by the PC;} \\
& t_h \quad \text{Arrival time of the shipment at the demand point } h, \forall h \in H; \\
& T_h \quad \text{The tardiness in delivery to demand point } h, \forall h \in H; \\
& S \quad \text{Production starting time at the PC;}
\end{align*}
\]

The mathematical model that formally defines our problem is as follows, where \( M \) is a sufficiently large number.

**Model \( P^s \)**

**Minimize** \( Z = \sum_{h \in H} w_h T_h \) \hspace{1cm} (1)

**Subject to**

1) **Capacity and flow balance constraints**

All the customer orders must be fulfilled

\[
\sum_{h \in H} x_{kh} + \sum_{i=1}^{\left|I_{kh}\right|} \bar{x}_{ih} = 1 \quad \forall h \in H \hspace{1cm} (2)
\]

The total outflow quantity (i.e., the number of
unit-size orders) from DC \( k \) to customers cannot exceed the inventory at DC \( k \)
\[
\sum_{h \in H} x_{kh} \leq I_k \quad \forall k \in K \tag{3}
\]
Each time slot in the PC’s operations can be assigned to at most one order
\[
\sum_{h \in H} \bar{x}_{ih} \leq 1 \quad \forall i, l \leq i \leq |H| \tag{4}
\]
Total number of orders produced by the PC equals to the total number of assigned time slots (because of the unit order sizes)
\[
Q = \sum_{h \in H} \sum_{i=1}^{[H]} \bar{x}_{ih} \tag{5}
\]
The shortage in component inventory must be provided by its suppliers
\[
\sum_{j \in J} q_{nj} \geq C_n \times Q - \bar{T}_n \quad \forall n \in N \tag{6}
\]
Constraints that establish the relationships between linear and binary variables
\[
q_{nj} \leq M \cdot y_{nj} \quad \forall n \in N, \forall j \in J_n \tag{7}
\]

2) Lead time constraints
The PC’s production must wait until the latest arrival of components from suppliers (if any)
\[
S \geq \bar{x}_{nj} \cdot y_{nj} \quad \forall n \in N, \forall j \in J_n \tag{8}
\]
Constraints that define the earliest possible arrival time at each demand point
\[
t_h \geq x_{kh} \cdot (a_{kh} + \tau_{kh}) \quad \forall k \in K, \forall h \in H \tag{9}
\]
\[
t_h \geq S + i \cdot R + \bar{x}_{ih} - M \cdot (1 - \bar{x}_{ih}) \quad \forall h \in H, \forall i, 1 \leq i \leq |H| \tag{10}
\]
Constraints that define the delivery tardiness at each demand point
\[
T_h \geq t_h - d_h \quad \forall h \in H \tag{11}
\]

3) Domain constraints
\[
x_{kh}, \bar{x}_{ih}, y_{nj} \in \{0,1\} \tag{12}
\]
All other variables are non-negative.

In the following discussion, we shall use \( \text{P}^s \) to denote this optimization problem, and show that problem \( \text{P}^s \) is solvable in strongly polynomial time.

Our study contributes to the literature of emergency logistics in three aspects. First, we explicitly model the total weighted tardiness in the objective function for the optimization. As pointed out by Caunhye et al. (2012), there were very few models dealing with the objective concerning delivery times in the current literature because of potential complexity of tracking multi-stage lead time. The need to track the network lead time in this study also makes the existing literature results not directly applicable to our case. Second, our analysis on the structural properties of the emergency logistics optimization problem identifies a special case that can be solved in strongly polynomial time, and facilitates the design of heuristics for the general cases of supply chain operations scheduling problems. Third, to our knowledge, this is the first result on supply chain operations scheduling that involves sequencing customer orders together with multi-stage lead times.

III. A DECOMPOSITION-BASED ALGORITHM FOR SOLVING \( \text{P}^s \)

In this section, we propose a strongly polynomial time algorithm for solving \( \text{P}^s \). To start, we decompose \( \text{P}^s \) into two sub-problems with the single PC as the disjunction point. Note that the customer orders can be fulfilled by either the inventory of local DCs or those newly produced by the PC, and that decision variable \( Q \) denotes the total quantity (or the total number of orders) produced by the PC. Since \( |H| \) is the total demand quantity and \( \sum_{k \in K} I_k \) is the total inventory from all DCs, \( \max \left\{ 0, |H| - \sum_{k \in K} I_k \right\} \) is the minimum production quantity (or the minimum number of orders fulfilled) by the PC. Thus, while the optimal value of \( Q \) will be determined by the search process of the algorithm to be proposed, its lower and upper bounds are known as a priori, and are given as

\[
\left[ \max \left\{ 0, |H| - \sum_{k \in K} I_k \right\}, \ |H| \right].
\]

Let \( \text{P}(Q) \) denote \( \text{P}^s \) under a given production quantity \( Q \), and the basic idea of our proposed algorithm is outlined as follows. For any given \( Q \) value, we decompose problem \( \text{P}(Q) \) into a downstream problem, denoted as \( \text{P}^d(Q) \), and a downstream problem, denoted as \( \text{P}^u(Q) \).

Problem \( \text{P}^d(Q) \) is concerned with the selection of suppliers that supplies enough components
(i.e., supplies) for producing $Q$ units of the product (i.e., $Q$ orders) so that the production starting time at the PC can be as early as possible. After solving $P^A(Q)$ optimally, we obtain the earliest feasible starting time at the PC, which ensues the lead times, denoted as $S(Q)$. Upon the given $Q$ and $S(Q)$, $P^B(Q)$ is about how to assign customer orders to inventory of DCs and to the time slots of the PC to minimize the total weighted tardiness in delivery to the customers. In following analysis, we shall show that for any given $Q$, the two sub-problems, $P^A(Q)$ and $P^B(Q)$, can both be solved in strongly polynomial time. Since the number of possible values of $Q$ is bounded by constant $|H|$, the original problem $P^b$ is strongly polynomial time solvable. A flow chart describing the solution procedure is presented in Figure 2. In the rest part of this section, we will elaborate on how to solve problems $P^A(Q)$ and $P^B(Q)$, respectively.

3.1. Solving the Upstream Problem under a Given Q Value, $P^A(Q)$

First, we consider the upstream network that consists of $|J|$ component suppliers and the PC. Given that the production quantity (or the total number of orders of unit size) is $Q$, our objective for optimization here is to find the schedule to minimize the latest arrival time of the components at the PC.

In order to calculate the lead time from component suppliers, we need to check whether the existing inbound inventory for components at the PC is enough for producing $Q$ orders. If yes, then the production at PC may start at time zero. Otherwise, the production starting time equals to the longest lead time from the suppliers who need to send out additional supplies of components. Therefore, the production starting time is a function of production quantity $Q$, denoted as $S(Q)$, and is defined as

$$S(Q) := \max_{n \in \mathbb{N}} \{S_n(Q)\}$$

where $S_n(Q)$ is the earliest possible ready time of component $n$ for production quantity $Q$. Note that, for any given $Q$, additional quantity of component $n$ to be shipped in from suppliers in $J_n$ is defined by $\max\{0, C_nQ - \hat{I}_n\}$. If $\max\{0, C_nQ - \hat{I}_n\} > 0$, then $S_n(Q) > 0$ implying a positive lead time for component $n$. Since suppliers for component $n$ is ordered in the increasing order of their shipping time to the PC (i.e., $\hat{t}_{nj} \leq \hat{t}_{n,j+1}$ for all $j = 1, ..., |J_n| - 1$), $S_n(Q)$ is defined as follows:

$$S_n(Q) = \begin{cases} 0 & \text{if } C_nQ \leq \hat{I}_n \\ \hat{r}_{nj} & \text{for } j = \min \{ \hat{r}_{nj} \neq \hat{t}_{nj} \} \left( \sum_{n=1}^{\infty} \hat{I}_n \geq C_nQ - \hat{I}_n \right) & \text{if } C_nQ > \hat{I}_n \end{cases}$$

(13)

Now, we consider the time complexity to solve $P^A(Q)$ for $Q \geq 0$. Equation (13) shows that the value of $S_n(Q)$ can be determined in $O(|J_n|)$ time. Thus, it takes $O(|J|)$ time to compute $S(Q)$.

We can also consider an extension to the situation where each component $n$ has, instead of a set of suppliers with finite inventory, a set of candidate suppliers with heterogeneous production rates and inventory levels. Each supplier for component $n$ may have a finite inventory or a production facility with a non-zero production rate or both. Let $\hat{I}_{nj}$ and $\hat{r}_{nj}$ be the inventory level and the production rate for component $n$ of the $j$-th supplier in $J_n$, respectively.

Now, let $Q_n(t)$ be the maximum quantity of component $n$ to be available at the PC in time $t$. Then, $Q_n(t)$ can be defined as follows:

$$Q_n(t) = \begin{cases} \hat{I}_n & \text{if } 0 \leq t < \hat{t}_{nl} \\ \hat{I}_n + \sum_{j=1}^{\infty} (\hat{I}_{nj} + \hat{r}_{nj}(t - \hat{t}_{nj})) & \text{if } \hat{t}_{nj} \leq t < \hat{t}_{n,j+1} \end{cases}$$

(14)

Figure 3 shows an example where there are three suppliers for component $n$, where supplier 1 carries inventory but does not have production facility, supplier 2 has a production facility but no inventory, and supplier 3 has both. $S_n(Q)$, which denotes the minimum lead time needed to replenish the right quantity of component $n$ for producing quantity $Q$ at the PC, can be defined as follows:
A Solvable Case of Emergency Supply Chain Scheduling Problem with Multi-stage Lead Times

Lee, Kangbok, Lei, Lei, and Dong, Hui

S_n(Q) = \begin{cases} 
0 & \text{if } C_nQ \leq \tilde{I}_n \\
\arg\min_{t \geq 0} \{Q_n(t) \geq C_nQ\} & \text{if } C_nQ > \tilde{I}_n 
\end{cases}

Since Q_n(t) is a non-decreasing piece-wise linear function of time t with O(|J_n|) non-differential points, S_n(Q) can be calculated in O(|J_n|) time. Therefore, in this extended model, since we have S(Q) := \max_{n \in N} \{S_n(Q)\}, it still takes a linear time of the number of suppliers for all components, O(|J|), to compute S(Q).

FIGURE 2. DECOMPOSITION-BASED OPTIMAL SOLUTION PROCEDURE

Start

Q = \max \left\{ 0, |H| - \sum_{k=1}^{|H|} I_k \right\}

Solve upstream problem P^A(Q)

S(Q)

Solve downstream problem P^B(Q)

Z(Q)

Q = Q + 1

Yes

Q \leq |H|?

No

Q^* = \arg\min \{ Z(Q) \ where \ \max \left\{ 0, |H| - \sum_{k=1}^{|H|} I_k \right\} \leq Q \leq |H| \}

End
3.2. Solving the Downstream Problem under Given $Q$ and $S(Q)$, $P^D(Q)$

For any given pair of $Q$ and $S(Q)$, we now show that the respective downstream problem, $P^D(Q)$, can be formulated as a minimum cost flow problem. To do so, let us construct a directed network $G = (V, A)$ with a source node $SRC \in V$ and a sink node $SNK \in V$. With any given $Q$, let the set of time slots at the PC be $L := \{1, \ldots, Q\}$, where each time slot will be assigned to one of the $Q$ orders. The node set $V$ and the arc set $A$ can then be defined as follows (see Figure 4).

The node set is defined as $V := \bigcup_{\alpha=0}^5 V_\alpha$ where

$V_0 = \{SRC\}$: The set of the source node, and $|V_0|=1$;

$V_1 = \{TSlt \mid i \in L\}$: The set of nodes representing the $Q$ time slots at the PC;

$V_2 = \{PInv\}$: The set of a dummy node representing the total amount of DC inventory used to fulfill the customer orders;

$V_3 = \{KInv_k \mid k \in K\}$: The set of nodes each representing a DC’s inventory used to fulfill the customer orders;

$V_4 = \{HDmd_h \mid h \in H\}$: The set of nodes representing demand points;

$V_5 = \{SNK\}$: The set of the sink node, and $|V_5|=1$.

The arc set is defined as $A := A_0 \cup A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5$, where each subset of arcs is defined below along with each arc flow’s lower bound ($LB$) and upper bound ($UB$).

The costs of arcs in this network are defined as follows:

- The cost of an arc in $A_{34}$ is $a(TSlt_i, HDmd_h) = w_i \max\{0, S(Q) + iR + \bar{r}_h - d_h\}$

- The cost of an arc in $A_{34}$ is $a(KInv_k, HDmd_h) = w_h \max\{0, a_{kh} + \tau_{kh} - d_h\}$

- The costs of the other arcs are equal to zeros.
For each arc \((u, v) \in A\), we have flow 
\(f(u, v)\) as a decision variable subject to upper bound \(UB(u, v)\) and lower bound \(LB(u, v)\), and cost \(a(u, v)\). The cost of sending this flow over arc \((u, v)\) is \(a(u, v) \cdot f(u, v)\). Therefore, \(P^B(Q)\) becomes the one to find flows minimizing the total cost:

\[Z(Q) = \sum_{(u,v) \in A} a(u,v) \cdot f(u,v) \]
subject to the following constraints:

1) **Flow capacity constraint**
\(LB(u, v) \leq f(u, v) \leq UB(u, v)\) for \((u, v) \in A\),

2) **Flow conservation constraint**
\[\sum_{(w,u) \in A} f(w,u) = \sum_{(u,w) \in A} f(w,u)\] for \(u \in V \setminus \{SRC, SNK\}\),

3) **Flow requirement constraint**
\[\sum_{(SRC,u) \in A} f(SRC,u) = |H| \text{ and } \sum_{(u,SNK) \in A} f(u,SNK) = |H|\].

Note that this network model can also be extended to handle additional constraints. For example, if there is a flow capacity limit for each arc between a particular DC and a demand point, we can revise the upper bound of the arc accordingly. Note that this network formulation gives a precise description of \(P^B(Q)\). First of all, the number of time slots equals exactly to the production quantity at the PC. From the discussion above, we see that a customer order can be served by either i) a time slot at the PC, or ii) the inventory of a DC. If the \(i\)-th time slot at the PC serves demand point \(h\), then the arrival time at demand point \(h\) equals to the summation of the starting time of the initial production at the PC, \(S(Q)\), the total waiting time in sequence of the \(i\)-th time slot including its production time, \(i \cdot R\), and the shipping time from the PC to demand point \(h\), \(\tau_h\), or \(S(Q) + iR + \tau_h\). If DC \(k\) serves demand point \(h\), then the arrival time at demand point \(h\) is the summation of the loading time at DC \(k\) for demand point \(h\), \(a_{kh}\), and the shipping time from DC \(k\) to demand point \(h\), \(\tau_{kh}\), or \(a_{kh} + \tau_{kh}\). In the optimal solution, if the flow over arc \((TSlt_i, HDmd_h)\) equals 1 then it implies that the \(i\)-th time slot serves demand point \(h\), and if the flow over arc \((KInv_k, HDmd_h)\) equals 1 then it implies the inventory of DC \(k\) serves demand point \(h\).

Also note that the number of time slots is bounded from above by the total number of customer orders, or \(|E| = Q \leq |H|\), and thus the total number of nodes in the network is bounded by \(O(|K|+|H|)\) and the total number of arcs is bounded by \(O(|H|+|K|))\). The minimum cost flow problem defined on \(G=(V', E')\) can be solved in \(O(|E'| \log |V'|) \times (|E'|+|V'| \log |V'|)\) time according to Orlin (1988).
FIGURE 4. THE NETWORK FLOW MODEL FOR $P^B(Q)$

Thus, $P^B(Q)$ can be solved in $O\left(\left|\mathcal{H}\right|\left|\mathcal{K}\right| + \left|\mathcal{H}\right|^2 \log \left(\left|\mathcal{K}\right| + \left|\mathcal{H}\right|\right)\right)$ time. Recall that $P^A(Q)$ can be solved in $O(1)$ time. Therefore, for each given production quantity $Q$, $P(Q)$ can be solved in $O(1)$ time. Furthermore, since the production quantity is bounded by $0$, the total number of the possible values of $Q$ is bounded from above by $|H|$. By enumerating all possible values for production quantity $Q$ and solving all the respective sub-problems $P^A(Q)$ and $P^B(Q)$, we can determine the optimal production quantity to minimize the objective function value (1) of $P^s$. Therefore, the time complexity of the proposed algorithm for solving $P^s$ is bounded from above by $O\left(\left|\mathcal{H}\right|\left|\mathcal{K}\right| + \left|\mathcal{H}\right|^2 \log \left(\left|\mathcal{K}\right| + \left|\mathcal{H}\right|\right)\right)$.

IV. A NUMERICAL EXAMPLE OF THE PROPOSED ALGORITHM

Since our proposed algorithm solves $P^s$ optimally in polynomial time, as discussed in Section 3, we will not perform any empirical study in this paper. Instead, we shall demonstrate the step-by-step solution process by this proposed algorithm in deriving the optimal solution to an example of $P^s$. In this numerical example, the network consists of six component suppliers for three components such that $|J_1|=1$, $|J_2|=2$ and $|J_3|=3$, a single PC, three local DC, and ten demand points. Assuming the time needed for the PC to produce an order is $R=2$, and other parameter values are given in Tables 1-3. As shown in Tables 1-3, $|H| - \sum_{k} I_k = 1 > 0$, and the range for the possible production quantity at the PC, $Q$, is $\left[\max\left\{0, |H| - \sum_{k} I_k\right\}, |H|\right] = [1, 10]$. 

Lee, Kangbok, Lei, Lei, and Dong, Hui
A Solvable Case of Emergency Supply Chain Scheduling Problem with Multi-stage Lead Times

Journal of Supply Chain and Operations Management, Volume 11, Number 2, October, 2013

40
Lee, Kangbok, Lei, Lei, and Dong, Hui
A Solvable Case of Emergency Supply Chain Scheduling Problem with Multi-stage Lead Times

**TABLE 1. PARAMETERS OF DEMAND POINTS**

<table>
<thead>
<tr>
<th>$h$</th>
<th>$d_h$</th>
<th>$w_h$</th>
<th>$\bar{\tau}_h$</th>
<th>$\tau_{bh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$k = 1$</td>
<td>$k = 2$</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>7</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>6</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>6</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>10</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>4</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

**TABLE 2. PARAMETERS OF COMPONENT SUPPLIERS AND THE PC**

<table>
<thead>
<tr>
<th>$n$</th>
<th>$C_n$</th>
<th>$I_n$</th>
<th>$I_{nj}$</th>
<th>$\bar{\tau}_{nj}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$j = 1$</td>
<td>$j = 2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$j = 1$</td>
<td>$j = 2$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>6</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Journal of Supply Chain and Operations Management, Volume 11, Number 2, October, 2013
TABLE 3. PARAMETERS OF DCS

<table>
<thead>
<tr>
<th>$k$</th>
<th>$I_k$</th>
<th>$a_{kh}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>h = 1</td>
<td>h = 2</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.5</td>
</tr>
</tbody>
</table>

For each possible value of $Q$, we apply the decomposition algorithm to solve the respective sub-problems. For example, when $Q = 3$, the sub-problems $P^A(Q)$ and $P^B(Q)$ are defined and solved, step-by-step, as follows.

For $P^A(3)$, we have $S_1(3) = 0$, $S_2(3) = 0$ and $S_3(3) = \tau_{3i} = 2$. Therefore, the earliest possible production starting time at the PC is $S(3) = \max \{0, 0, 2\} = 2$.

For $P^B(3)$, flow cost $a(TSl_t, HDmd_h)$ = $w_h \max \{0, 2 + i2 + \tau_h - d_h\}$ is calculated and its values for $i = 1, 2, 3$, and $h = 1, \ldots, 10$ are summarized as above.

By solving $P^B(3)$, we have the optimal value of 153 when $Q = 3$. Similarly, we solve all the sub-problems with respect to $Q = 1, 2, \ldots, 10$. Table 4 summarizes the results.

As we can see from the results listed in Table 4, the optimal production quantity (or the number of orders) produced by the PC is $Q^* = 4$. Note that the total tardiness is not a unimodal function of the production quantity and thus we need to enumerate all sub-problems with possible production quantities. The optimal assignment plan is summarized in Table 5, where DC $k$ stands for the $k$-th DC and TS $i$ stands for the $i$-th time slot at the PC.

V. CONCLUDING REMARKS

We studied the integrated production and transportation scheduling problem of a capacitated supply chain network consisting of component suppliers, a PC, DCs, and many customer demand points, with multi-stage lead times. The objective is to minimize the total weighted tardiness in the delivery to customers, which is a widely adapted industry performance measure for supply chains involved in disaster reliefs and emergency logistics. Assuming the customer orders are unit-sized, a decomposition-based algorithm for assigning customer orders to DCs and the PC (and then the selection of suppliers) is proposed, which finds the optimal solution to the respective integrated operations scheduling problem in strongly polynomial time. A numerical example that shows the step-by-step solution process of this proposed search algorithm is also presented.

There are several extensions from this study. First, a heuristic that uses the proposed polynomial time algorithm as a subroutine to
solve a more general integrated operations scheduling problem is of great interest, such as the one with customer-dependent order quantities where an order of size $D_h > 1$ can be treated as $D_h$ orders of unit size. Each shelter or hospital has a different capacity for accommodating affected people or patients, which lead to a different order size. The other one is to extend the proposed solution approach to allow each customer to place multiple orders, allowing an inventory at each customer site, over a given interval of multiple time periods. Each customer demand point places a sequence of orders over the time, which is also a common practice of disaster relief. Hospitals in an affected area usually place an order before the hit of an anticipated natural disaster and then few subsequent orders after the disaster has arrived, depending on their local needs. When multiple shipments to a customer are allowed, we can extend the proposed algorithm directly to this case as well.

### TABLE 4. NUMERICAL EXAMPLE RESULTS

<table>
<thead>
<tr>
<th>$Q$</th>
<th>$P^A(Q)$</th>
<th>$P^B(Q)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$S_1(Q)$</td>
<td>$S_2(Q)$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

### TABLE 5. OPTIMAL ASSIGNMENT PLAN ($Q^* = 4$)

<table>
<thead>
<tr>
<th>$h$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>DC 3</td>
<td>DC 1</td>
<td>TS 1</td>
<td>TS 2</td>
<td>DC 2</td>
<td>TS 3</td>
<td>DC 2</td>
<td>TS 4</td>
<td>DC 3</td>
<td>DC 3</td>
</tr>
</tbody>
</table>
VI. REFERENCES


Stecke, K.E., X. Zhao, 2007. Production and transportation integration for a make-to-


**Web References**
