Heuristics For A Joint Capacitated Production, Inventory, And Distribution Model with Production Setup Times

Leyla Ozsen*
San Francisco State University, San Francisco, California, USA

Paul Intrevado
University of San Francisco, San Francisco, California, USA

Stewart Liu
San Francisco State University, San Francisco, California, USA

In recent years, supply chain networks have been increasing in complexity due to the globalization of manufacturing operations. In a capacitated planning problem with intermediate distribution centers (DCs), decisions must be made for production lot sizes, production schedules, shipment amounts between locations and more. These models generally consist of a non-linear objective and are difficult to solve. In this paper, we formulate the Capacitated Production, Inventory, and Distribution Problem (CPIDP) for multiple products on a single production line in the supply chain network and develop an efficient simulated annealing (SA) algorithm and various improvement heuristics. We then present the computational results that demonstrate the reliable and robust performance of our approach.

* Corresponding Author. E-mail address: lozsen@sfsu.edu

I. INTRODUCTION

The integrated model considered in this paper is motivated by the current operations of a major global manufacturing company that produces paper-based consumer products. It has several plants in Latin America in each of which various products are manufactured on a single production line. The products are then shipped from its plants to distribution centers (DCs). Distribution operations such as storage in the DCs and the shipments from the DCs to the retailer stores are executed largely by other companies. In order to reduce the total costs of transportation, inventory management, and production setup, accrued by such a firm, the complex interactions between production and distribution operations must be analyzed simultaneously.

The Capacitated Production, Inventory, and Distribution Problem (CPIDP) introduced in this paper addresses a supply chain network consisting of multiple plants and DCs. The plants have limited production capacity and only a single production line. The objective is to minimize the sum of transportation, inventory holding, and production setup costs. The CPIDP simultaneously determines production lot sizes, production scheduling, and shipments of multiple products between plants and DCs.

There are two streams of research in the logistics literature that are related to the CPIDP. One of which is the Economic Lot Scheduling Problem (ELSP) (Hanssmann, 1962; Maxwell, 1964; Bomberger, 1966; Hsu, 1983). However, these authors only consider determining a cyclic schedule for making multiple products on a single production line within one plant without handling a complex supply chain network consisting of multiple plants and multiple DCs. The other stream of research related to the CPIDP proposes integrated production, inventory, and distribution models (Blumenfeld, Burns, Diltz, and Daganzo 1985; Benjamin, 1989; Tang, Yung, and Ip, 2004; Yung, Tang, Ip, and Wang, 2006); however these models all assume the Economic Order Quantity (EOQ) policy for inventory management. The EOQ assumes that each product can be treated independently and does not consider scheduling of multiple products. Therefore, most previous integrated models do not fit the environment we wish to model, which has limited capacity at the plants and hence products cannot be considered independently (e.g. inventory management for multiple products on a single production line).

In this paper, we introduce the CPIDP that combines a variant of the Economic Lot Scheduling Problem (ELSP) with the transportation problem and thus brings together the two above-mentioned disjoint research efforts. The CPIDP is also able to capture the capacity limitations at the plants not just through fixed exogenous capacity constraints but through setup times. To the best of our knowledge, there is no model in the literature that integrates transportation, inventory management, and production with setup times for multiple products in the supply chain network.

We prove that the CPIDP is NP-hard by showing that a special case of the problem is equivalent to the Generalized Assignment Problem (GAP). A formal proof is given in Appendix A. Therefore, we develop various heuristic approaches including Simulated Annealing (SA) and assess their performance under a variety of input parameters and control parameter settings.

Our model also provides managerial insights into the behavior of production and distribution operations that are adopted by firms. Schmenner (1979) discusses various multiple plants manufacturing strategies qualitatively based on survey data. Two major plant strategies Schmenner identifies in his seminal work are “product plant strategy” and “market area plant strategy”. In a product plant strategy, plants make distinct products, each plant serving the company’s entire market demand for that specific product. A market area plant strategy leads the firm to operate at the other end of the spectrum, in that each plant produces all or most of the products, serving only a limited geographical market area. However, there seems to exist no quantitative model that produces solutions leading to the two extreme strategies identified by Schmenner (1979).

We claim that the CPIDP provides a quantitative basis for the strategic analysis by Schmenner (1979). In our extensive computational experiments, the pattern of production and distribution solutions generated by our proposed model spans the range between the two extreme multiple-plant manufacturing strategies practiced by firms, depending on the relative magnitudes of setup and transportation costs.

This paper is organized as the following: Section 2 reviews previous work related to our novel model and simulated annealing (SA) solution approaches in the domain of production scheduling and distribution planning. Section 3 presents the proposed problem description and the mathematical model formulation. Section 4 describes various solution approaches including simulated annealing (SA) for solving the problem. In Section 5, we systematically assess the performance of various heuristics and investigate the impact of input parameters on the resulting network structure. Section 6 discusses the conclusions of the research and possible future studies.

II. LITERATURE REVIEW

2.1. Integrated Models and Solution Approaches

Traditional approaches had analyzed production, inventory, and transportation operations separately rather than considering the entire operation as a whole (Bertazzi and Speranza, 1999; Glock, 2012). For example, when deciding inventory levels and shipments of products on a given logistics network, typically inventory levels were determined first and then the shipments of products were determined. However, as manufacturing industries have gone global and more competitive, simultaneous analysis of the complex interactions between production and distribution operations had become essential to achieve efficiency (Sarmiento and Nagi, 1999). Even though this integrated analysis received much attention, as of 1999 there were still a limited number of contributions in this area (Bertazzi and Speranza, 1999; Sarmiento and Nagi 1999), especially in the integrated analysis of production and distribution with the goal of minimizing the sum of transportation, inventory, and production setup costs for the multiple products in the multiple supplier/destination settings.

The earlier integrated network models focus on the single plant case; Chandra and Fisher (1994) integrate production scheduling and vehicle routing problems and Blumenfeld, Burns, Diltz, and Daganzo (1985) develop an economic order quantity (EOQ) based model to analyze the trade-offs between inventory, transportation and production setup costs. For a multiple supply and demand point network, Benjamin (1989) presents a formulation that combines the production lot size and the transportation problem and devises a heuristic based on Bender's decomposition. However, the author assumes that the annual production level at each supply point is pre-determined and only one product case is considered. Following Benjamin's work, after a gap of 15 plus years, Tang, Yung, and Ip (2004) and Yung, Tang, Ip, and Wang (2006) relax the key assumption that the annual production amount at each plant is pre-determined. The problem is more difficult to solve since the annual production level affects production lot size, shipment amounts, and periodic order quantities. To handle the increased complexity, Tang, Yung, and Ip (2004) break down the entire problem into two hierarchical subproblems and solve them sequentially. Yung, Tang, Ip, and Wang (2006) point out that the sequential solution approach in Tang, Yung, and Ip (2004) does not capture the trade-offs between production and transportation costs; to simultaneously consider transportation costs when deciding production lot size and annual production, Yung, Tang, Ip, and Wang (2006) develop a more sophisticated method, using Lagrange multipliers.

Other integrated network models include Park (2005), who studies an integrated production planning and transportation model with multiple retailers, products, and periods. A two-phase heuristic model is used to solve the resulting mixed integer program. The heuristic first obtains an initial solution for distribution and production decisions. Transportation parameters are then modified to improve the solution. Zegordi, Abadi, and Nia (2010) study a two-stage supply chain scheduling problem where multiple suppliers must deliver parts to one single manufacturing plant. They prove that the model is NP-Hard, and a gendered genetic algorithm (GGA) is shown to perform well for this setting. Darvish (2016) presents an alternative approach to the scheduling and lot-sizing problem in a multi-plant, multi-period setting, under the idea of the Physical Internet (PI). While the PI allows transportation models to be simplified and improves transportation efficiency overall, at the writing of this paper, the real-world logistics infrastructure still differs from the PI significantly.

The two distinguishing aspects of our work from the aforementioned integrated models are one, the way the capacity constraints are modeled and second, the assumed inventory model for handling multiple products in each plant. The previous integrated models capture the capacity limitations through hard capacity constraints, which model the average production flow through plants and Distribution Centers (DCs); the CPIDP we propose models the capacity limitation dynamically through setup times, as production lot sizes at each plant are being solved for simultaneously. Furthermore, these previous integrated models all assume a class of Economic Order Quantity (EOQ) policy for inventory management. The EOQ implies that each product can be handled independently of all others, so it does not consider scheduling of multiple products. In certain circumstances like inventory management for multiple products on a single production line, the EOQ cannot be a reasonable assumption.

To this second point, the other stream of research closely related to the CPIDP is the inventory management of multiple products on a single production line, referred to as the Economic Lot Scheduling Problem (ELSP), which has been extensively studied in the literature. Hanssmann (1962) solves the ELSP with the assumption that cycle time for each product is all the same and no set up is considered. Maxwell (1964) considers the setup time required during changeover from one product to another product. Elmaghraby (1978) reviews various early contributions to the ELSP and divides these into two broad categories: analytical approaches that achieve the optimum of a restricted version of the original problem; and heuristic approaches that achieve "good" solutions of the original problem. Hsu (1983) shows that a restricted version of the ELSP is NP-hard. The author takes into consideration different cycle times, but restricts the frequency to powers of two. Drexl and Kimms (1997) review work in the well-established area of lot sizing and scheduling.

The previous works in the ELSP literature go beyond the EOQ when considering lot sizing within a single plant but does not capture the trade-offs between production, inventory, and transportation costs. On the other hand, the integrated models although have been extended to consider the multi-plant case, they all assume the EOQ model which is not appropriate for the multiple product case on a single production line. To the best of our knowledge, there seem to exist no model that brings the approaches in these two distinct research streams. Therefore, it is worthwhile studying a new integrated model to determine production lot size, production scheduling, and shipment amounts of different products on a given supply chain network. In the next section, we briefly review the Simulated Annealing (SA) approach and some applications in the domain of production and distribution problems.

2.2. Simulated Annealing

Simulated Annealing (SA) is a well-known and widely applied meta-heuristic that potentially produces high quality solutions to various NP-hard problems (Silver, 2004). It has been shown that SA algorithm converges asymptotically to the global optimum, although the worst-case time complexity can be exponential (Eglese, 1990). Even though a wide variety of heuristic approaches such as constructive and local improvement methods have been proposed to solve NP-hard problems in reasonable computation time, the fundamental weakness of these approaches is that they get trapped in a local optimum (Silver, 2004; Eglese, 1990). SA is a powerful solution technique for NP-hard problems in that SA can overcome this drawback by allowing non-improving moves with a certain probability during the neighborhood search (Eglese, 1990).

We briefly review some terminology used in the simulated annealing process. First an initial feasible solution is obtained using some construction heuristic. Simulated annealing algorithm improves an incumbent solution through neighborhood search. A neighborhood is the set of feasible solutions that can be generated in a single move from the current solution. We generate a move from the current solution to its neighborhood, and then evaluate the change in the objective function value (e.g., the total cost). If the cost is reduced, then we accept this move (downhill move) and update the current solution. Otherwise, we accept this move (up-hill move) only with some probability. At the beginning of a SA algorithm, the temperature, which is a control parameter to adjust the acceptance probability for inferior solutions, is set to high values allowing almost any non-improving movement during the neighborhood search. By carefully and slowly cooling the current temperature, we reduce the acceptance probability that a non-improving move can be accepted. At each temperature, SA tries to move from the current solution to its neighborhood for some pre-specified number of iterations; we keep track of the overall best solution at a given temperature. Simulated Annealing algorithm terminates when the minimum temperature, the maximum available iteration, or maximum available run time is reached (Ovacik, Rajagopalan, and Uzsoy, 2000).

Koulamas, Antony, and Jaen (1994) show that SA method has been widely used for solving various operations research problems and can be very useful, particularly when the alternative solution approaches involve enumeration. Vidal (1993) applies SA method to the traveling salesman, telecommunication network design, and facility location problems. We can also find such applications of SA in the domain of production and distribution. Palmer (1996) develops a SA algorithm to coordinate process planning and production scheduling and evaluates the appropriateness of various other solution approaches. Kim and Kim (1996) tackle the short-term production scheduling problem using SA since it is hard to solve the problem optimally within reasonable computational times if processing capacities of a manufacturing system and priority among products are considered. Jayaraman and Ross (2003) propose a SA algorithm to solve a distribution network design model using cross-docking. To deal with practical large-scale instances of this NP-hard problem, a SA heuristic is devised and extensively tested with various control parameter settings for various problem sizes. The solution gap between the optimal solution using LINGO and the heuristic solution from SA on average is less than 4% for all data sets. Running time of SA is much faster than that of LINGO.

Simulated annealing has been widely applied in a broad area of production scheduling and distribution problems and has been reported to provide high quality solution in this area. It is also easy to implement. Therefore, we explore the use of SA to the integrated and capacitated production, inventory, and distribution model that we propose.

III. PROBLEM DESCRIPTION AND MODEL FORMULATION

3.1. Introduction

The Capacitated Production, Inventory, Distribution Problem (CPIDP) mimics the business operations of a major global manufacturing company that produces various paper-based consumer products in plants located across Latin America; each plant has a single production line capable of manufacturing multiple products. The products are then shipped from the plants to the Distribution Centers (DCs), operated by third parties. Accordingly, for the development of the CPIDP, we assume that the logistics network such as the location and capacities of plants and the location of DCs are given; and we focus on modeling the operational and tactical decisions, such as the production lot sizes at the plants during each production cycle, when each product should be produced (the length of a production cycle), and shipment quantities from the plants to the DCs to fulfill the annual demand for each product occurring at the DCs.

The objective of the CPIDP is to minimize the sum of four costs ensuing these operational and tactical decisions. The first cost is the production cost that may differ across plants due to exchange rates, tariffs, labor costs, etc. The second is the transportation cost from the plants to the DCs. The third cost is the setup cost that occurs during changeover from one product to another on the single production line at each plant. The fourth cost is the inventory holding cost that occurs during the production runs. These costs may differ depending on the location of the plant and also across various products.

We take into account production capacity limitations at the plants, in each of which multiple products can be manufactured one at a time on a single production line. Changeover (setup) time for switching among different products is considered. We assume that setup cost is independent of production sequence and is proportional to the setup time (Bomberger, 1966). In determining when each product should be produced, we use the common cycle approach, that is there is exactly one setup for each product in a production cycle (Hanssmann, 1962; Maxwell, 1964; Nahmias, 2004). This approach to modeling the capacity limitations differ from the classical approach of simply limiting the average throughput of a product at a plant and is a distinguishing feature of the model in comparison to other models integrating various operational decisions.

We assume that the demands are known and constant which is a reasonable assumption for the type of products the CPIDP targets. It is then reasonable to assume that the total available production capacity across plants is large enough to satisfy the total demand from the distribution centers, so that shortage is not allowed during production runs. Furthermore, we assume single-sourcing at the DCs, that is the demand for a given product at a DC is served by a single plant, thereby, we need to decide on the assignment of DCs to plants for each product.

3.2. Mathematical Formulation

We define the following notation to formulate the CPIDP described above. Note that all parameters and decision variables are respectively denoted by the use of lower-case and upper-case letters.

Indices, Sets, and Input parameters:
$i \in I = \{1, 2, \cdots, I\}$: *A set of plants*
$j \in J = \{1, 2, \cdots, J\}$: *A set of products*
$k \in K = \{1, 2, \cdots, K\}$: *A set of distribution centers*

s_{ijk}: Unit transportation cost for product *j* from plant *i* to distribution center *k*
c_{ij}: Unit production cost for product *j* at plant *i*
d_{jk}: The demand for product *j* at distribution center *k*
p_{ij}: The annual production rate for product *j* at plant *i*
f_{ij}: The setup cost of product *j* at plant *i*
h_{ij}: Unit inventory holding cost of product *j* at plant *i*
a_{ij}: The setup (changeover) time of product *j* at plant *i*

Variables:
$X_{ijk} =$ 1, if the demand for product *j* at DC *k* is served by plant *i*, 0 otherwise
λ_{ij}: Annual production quantity of product *j* at plant *i*
Q_{ij}: Lot-size of product *j* at plant *i*
T_i: Production cycle time at plant *i* (year)
$Y_{ij} = 1$, if plant *i* produces product *j*, 0 otherwise

As explained previously, during the common production cycle, T_i, we assume that exactly one lot of product *j* is manufactured. Because we also assume that no shortage is allowed in a production cycle, the lot for product *j* should be large enough to meet the demand during a production cycle. Thus, the lot size for product *j* at plant *i* must be:

$$Q_{ij} = \lambda_{ij} T_i \tag{1}$$

Extending the works of Hanssmann (1962), Maxwell (1964), and Nahmias (2004) to multi-plant case, the average annual setup and inventory holding cost for all products at plant *i* is given by:

$$\sum_j \left(f_{ij} \frac{\lambda_{ij}}{Q_{ij}} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{Q_{ij}}{2} \right) \tag{2}$$

The first term in (2) is the annual production setup costs at plant *i* and the second term is the annual inventory holding costs for all products at plant *i*. Substituting equation (1) into (2), we derive the total cost of production setup and inventory holding costs across all plants in terms of T_i as follows:

$$\sum_i \sum_j \left(\frac{f_{ij}}{T_i} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{\lambda_{ij} T_i}{2} \right) \qquad (3)$$

The reason for expressing these two costs in terms of T_i rather than Q_{ij} is to reduce the number of variables in the mathematical formulation. Also, it should be noted that the constraints will also be expressed in terms of T_i . The sum of production costs at the plants and transportation costs for shipping products between the plants and the distribution centers are given by:

$$\sum_i \sum_j \sum_k \left(c_{ij} + s_{ijk} \right) d_{jk} X_{ijk} \qquad (4)$$

Given the above notation and cost components, the formulation for the CPIDP is as follows:

$$Minimize \sum_i \sum_j \sum_k \left(c_{ij} + s_{ijk} \right) d_{jk} X_{ijk} + \sum_i \sum_j \left(\frac{f_{ij}}{T_i} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{\lambda_{ij} T_i}{2} \right) \qquad (5)$$

Subject To

$$\sum_i X_{ijk} = 1 \quad \forall j,k \qquad (6)$$

$$\sum_k d_{jk} X_{ijk} = \lambda_{ij} \quad \forall i,j \qquad (7)$$

$$\sum_j \frac{\lambda_{ij}}{p_{ij}} \le 1 \quad \forall i \qquad (8)$$

$$\sum_j \left(a_{ij} Y_{ij} + \frac{\lambda_{ij} T_i}{p_{ij}} \right) \le T_i \quad \forall i \qquad (9)$$

$$X_{ijk} \le Y_{ij} \quad \forall i,j,k \qquad (10)$$

$$\lambda_{ij} \ge 0 \quad \forall i,j \qquad (11)$$

$$T_i \ge 0 \quad \forall i \qquad (12)$$

$$X_{ijk} \in \{0,1\} \quad \forall i,j,k \qquad (13)$$

$$Y_{ij} \in \{0,1\} \quad \forall i,j \qquad (14)$$

The objective function (5) minimizes the sum of production costs at the plants, shipment costs of all products from the plants to the distribution centers, and the costs of production setup and inventory holding at each plant. Constraint (6) along with (13) indicates that the entire demand for a product at a distribution center should be satisfied by exactly one plant. Constraints (7) are flow balance constraints ensuring that for each product, total annual shipment out of a plant is equal to the total quantity produced at that plant. Constraint (8) ensures that the assigned production quantities across all products at a given plant does not exceed the capacity of that plant (see also Nahmias, (2004)). The left-hand side of this constraint can also be interpreted as the utilization level of a plant. Constraint (9) ensures that within each production cycle, there is sufficient time to account for both the set-up and production run time of all assigned products to that plant. The second term in equation (9) is the actual production uptime, Q_{ij}/p_{ij}, re-written in terms of T_i using equation (1). Therefore, constraint (9) indicates that the common cycle time at a plant should be larger than the sum of setup times and actual production uptimes at that plant. Constraint (10) is the linkage constraint, ensuring that if product j is not being produced at plant i then no DC can be receiving shipments of product j from plant i. Equations (11) and (12) are non-negativity constraints. Equations (13) and (14) are binary constraints.

Since the CPIDP combines the transportation problem with the Economic Lot Scheduling Problem (ELSP), the parts of production setup and inventory costs in the objective function and constraints (8) and (9) follow the ELSP formulation provided by Maxwell (1964). We prove that the CPIDP is NP-hard by showing that a special case of the problem is equivalent to the Generalized Assignment Problem (GAP) (Refer to Appendix A).

IV. SOLUTION METHODOLOGIES

4.1. Introduction

As stated above, the CPIDP is NP-hard. Therefore, in this section we develop various heuristics including a construction heuristic providing an initial feasible solution in Section 4.2. In Section 4.3, we describe various improvement heuristics for finding a better solution based on the initial solution provided by the greedy algorithm. Lastly, the simulated annealing approach that yields the best solution for most instances of the CPIDP is explained in Section 4.4.

4.2. Greedy Heuristic

The basic idea of the greedy heuristic we propose is to first determine the optimal production cycle time, T_i at plant i in a closed form by temporarily assuming that the annual production level λ_{ij} is known at plant i. The optimal cycle times at plant i, T_i^* is then substituted back into the mathematical formulation to actually determine the optimal annual production levels and the shipments minimizing the total cost.

Suppose momentarily the annual production levels at a plant for all products are λ_{ij}. Then we can utilize the following subproblem and solve for the optimal production cycle time:

$$\textit{Minimize} \sum_i \sum_j \left(\frac{f_{ij}}{T_i} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{\lambda_{ij} T_i}{2} \right) \quad (15)$$

Subject To

$$\sum_j \left(a_{ij} Y_{ij} + \frac{\lambda_{ij} T_i}{p_{ij}} \right) \leq T_i \quad \forall i \quad (16)$$

$$T_i \geq 0 \quad \forall i \quad (17)$$

If we define $G(T_i) = \sum_j \left(\frac{f_{ij}}{T_i} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{\lambda_{ij} T_i}{2} \right)$, then the objective function (15) is equivalent to $\min \sum_i G(T_i)$. In order to find the optimal cycle time T_i at plant i minimizing the objective (15), we differentiate $G(T_i)$ with respect to T_i, yielding

$$\frac{\partial G(T_i)}{\partial T_i} = 0 \quad \Rightarrow \quad -\frac{\sum_j f_{ij} Y_{ij}}{T_i^2} + \frac{\sum_j h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \lambda_{ij}}{2} = 0$$

$$T_i^{CC} = \sqrt{\frac{2 \sum_j f_{ij} Y_{ij}}{\sum_j h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \lambda_{ij}}}$$

Since the sign of second derivative of $G(T_i)$ is positive (refer to the Appendix B), T_i^{CC} is the cycle time that minimizes the objective (15) without considering constraint (16). Constraint (16) indicates that cycle time at plant i should be large enough to accommodate both production setup and production up time, and can be rewritten as

$$T_i \geq \frac{\sum_j a_{ij} Y_{ij}}{1 - \sum_j \frac{\lambda_{ij}}{p_{ij}}} \quad \forall i \tag{18}$$

The right-hand side of (18) is equivalent to the minimum time required for accommodating both production setup and production up time. Therefore, the optimal cycle time at plant *i* is as follows:

$$T_i^* = Max\,(T_i^{CC}, T_i^{MIN}) \quad where\ T_i^{MIN} = \frac{\sum_j a_{ij} Y_{ij}}{1 - \sum_j \frac{\lambda_{ij}}{p_{ij}}} \tag{19}$$

Substituting T_i^* into (15) gives the optimal objective value for a given λ_{ij}. The greedy heuristic described next utilizes this result.

Because we assume that a product at a DC is served by a single plant, we can assign a product at a specific DC to a plant until the capacity of that plant is filled up, thereby fixing the specific plant-product assignment, X_{ijk}, for that DC and increasing the corresponding assigned production level, λ_{ij}, at that plant. Given the new production levels at a DC, we can compute the optimal cycle time using equation (19) and the ensuing costs. At every step of the greedy heuristics, we search for the best plant-product assignment such that the total cost given in equation (5) is increased the least. We repeat this process until every product at a DC is assigned to a plant.

The greedy heuristic algorithm is summarized as follows:

Step1. Sort the demands d_{jk} for product j at DC k in descending order so that larger demands are assigned earlier in the process.
Step2. Find best plant i^*_{jk} that serves each demand (d_{jk})

Step2-a. Repeat the following? procedure for every plant *i*,

$Assign\ d_{jk}\ to\ plant\ i\ (X_{ijk} \leftarrow 1)$

$\lambda_{ij} \leftarrow \lambda_{ij} + d_{jk} X_{ijk}$

$If \sum_j \frac{\lambda_{ij}}{p_{ij}} \leq 1\ at\ plant\ i, \quad then\ keep\ the\ updated\ \lambda_{ij}, X_{ijk}$

otherwise, go to next plant

Calculate T_i from eqn (19)

Calculate total cost from eqn (5)

Step2-b. Find best plant $i^*_{jk} \leftarrow \arg\min_i \{\text{total cost}\}$

Step3. Assign demand d_{jk} to the found best plant i^*_{jk}

$X_{i^* jk} \leftarrow 1$

$\lambda_{i^* j} \leftarrow \lambda_{i^* j} + d_{jk} X_{i^* jk}$

Step4. Repeat Steps 2-3 until every demand d_{jk} is assigned

4.3. Improvement Heuristics

We discuss four different improvement heuristics applied to the initial feasible solution constructed by the greedy heuristic. The greedy heuristic typically fails to find the global optimum in that at each iteration we make the best decision locally for production lot size, production cycle time, and shipments. One possible way to improve the initial solution is to reassign or swap production quantities determined by the greedy heuristic if it reduces the total cost. For example, suppose that the production amounts for product 1 at plant 1 is λ_{11}=100 unit and that for product 1 at plant 2 is λ_{21} =70 unit. We swap the production amounts for product 1 at these two plants if the capacity constraints at these plants are satisfied. If the total cost at plant 1 decreases more than the total cost at plant 2 increases after the product exchange, then we improve the solution by swapping production levels. Since the total production amount for product 1 does not change, total demand requirement is satisfied.

Improvement heuristics are categorized into four groups by how many products can be reassigned or swapped:

i. RWP: It reassigns the entire production amount for a specific product at a plant to other plants within the capacities of plants.

ii. RPD: It reassigns only production for serving the demand of a product at a specific DC to other plants within the capacities of plants.

iii. SWP: It swaps the entire production for a specific product at a plant with that of another plant within the capacities of plants.

iv. SPD: It swaps only production for serving the demand of a product at a specific DC with that of another plant within the capacities of plants.

Each improvement heuristic is run repeatedly until no further improvement is obtained. Thus, it guarantees that a local minimum is found at the termination of each local improvement heuristic.

4.4. Simulated Annealing

We implement a Simulated Annealing (SA) algorithm to obtain higher quality solutions than those provided by the improvement heuristics described in Section 4.3. In most cases, the improvement heuristics can only obtain a locally optimal solution. To avoid getting trapped in the local optima, simulated annealing has been widely used for a broad range of applications (Koulamas, Antony, and Jaen, 1994; Vidal, 1993), partly because it is easy to implement and understand. Moreover, SA can often find high quality solutions close to the global optimum and this optimum does not depend on the initial solution (Eglese, 1990).

Let *S* be the solution vector consisting of $(X_{ijk}, \lambda_{ij}, Y_{ij}, T_i)$ and *L* the number of iterations to be performed at each given temperature, *T*. Let ΔE be the change in the objective function value. The pseudo code for the simulated annealing algorithm is as follows:

```
Initial solution S = S0
Repeat until stop criterion temperature T reaches stopping rule Tf
  At each given temperature
        Repeat L times
                Generate a new set of solution S′ from current solution S
                If ΔE < 0 , then accept this move and the update solution ( S ← S′ )
Else if P = exp(−ΔE / T) > uniform[0..1], then accept this move and update the solution
  Decrease Temperature ( T_new = c T_old , where c is cooling factor)
In the entire process, keep the best-known total cost and solution, and at the end return
best-known cost and the associated solution.
```

In order to develop an efficient simulated annealing algorithm, we need to find sufficiently large neighborhoods. As a perturbation method, swapping can be utilized. Alternatively, genetic algorithms such as mutation or inversion can also be used. Initially, we tried using the SWP described in Section 4.3 to generate a new solution. However, the SWP does not provide a sufficiently large neighborhood to yield high quality solutions. Therefore, we changed the perturbation method to the SPD, providing a much larger neighborhood than the SWP.

The following pseudocode describes how to obtain a neighboring solution using the SPD procedure.

Generate_New_Solution (S)

Step 1. Randomly select a plant-product assignment X_{ijk} from current solution – i.e. generate a random number for index i, j, k using uniform distribution.

Step 2. Select randomly plant-product assignment $X_{i'j'k'}$ to be swapped – i.e. generate a random number for index i' ($\neq i$), j', and k'.

Step 3. Swap X_{ijk} with $X_{i'j'k'}$ based on the SPD explained in Section 4.3

$$X_{i'jk} \leftarrow X_{ijk}, \quad X_{ijk} \leftarrow 0$$

$$X_{ij'k'} \leftarrow X_{i'j'k'}, \quad X_{i'j'k'} \leftarrow 0$$

$$\lambda_{i'j} \leftarrow \lambda_{i'j} + D_{jk}, \quad \lambda_{ij} \leftarrow \lambda_{ij} - D_{jk}$$

$$\lambda_{ij'} \leftarrow \lambda_{ij'} + D_{j'k'}, \quad \lambda_{i'j'} \leftarrow \lambda_{i'j'} - D_{j'k'}$$

$$If \sum_j \frac{\lambda_{ij}}{p_{ij}} \leq 1 \ at \ plant \ i' \ or \ i,$$

Calculate T_i from equation (19)

Calculate total cost from equation (5)

Else, go to Step 1.

Step 4. Return a new generated solution vector S'.

It is also important to choose an appropriate initial temperature and a cooling rate that should then be decreased at a sufficiently slow rate. First, we set the initial temperature very high so that almost any non-improving move can be accepted at the beginning of the simulated annealing process. We use the geometric-cooling rule $T_{new} = cT_{old}$, where $c < 1$ and close to 1 (Jayaraman and Ross, 2003). Cooling schedules such as initial temperature, stopping rule T_f and the number of iterations at a given temperature are mostly decided based on empirical experience and are useful to improve quality of solutions and decrease CPU time.

V. COMPUTATIONAL RESULTS

In this section, we present the computational results for various combinations of the heuristics discussed in Section 4. In Section 5.1, we describe our experimental design. In Section 5.2, we compare the performance of our heuristics for various problem sizes. In Section 5.3, we discuss the impact of some key input parameters on the solution structure and draw managerial insights into the behavior of production and distribution operations. Section 5.4 presents our observations on implementing the simulated annealing heuristic.

5.1. Experimental Design

We start with listing the possible heuristics to be tested in the main experiments. In Section 4, we introduced a Greedy Heuristic (GH) for constructing an initial feasible solution, four different Improvement Heuristics (IHs), and the Simulated Annealing (SA) algorithm to find a high-quality solution. Starting from the initial solution provided by the GH, each improvement heuristic is run until no further improvement is obtained; this guarantees that a local minimum is found at the termination of each local improvement heuristic. The SA algorithm also starts from the GH solution and is run until the stopping rule is reached. In pilot experiments, DICOPT (Discrete and Continuous Optimizer) and BARON, which are mixed-integer nonlinear programming (MINLP) commercial solvers, are also tested for comparison purposes with our heuristics. However, both of them produce infeasible solutions for the majority of our problem instances. The six heuristics to be further tested are summarized in Table 1.

TABLE 1. ALGORITHMS TO BE TESTED IN MAIN EXPERIMENTS

Identifier	Description
GH	Greedy Heuristic
GH+RWP	RWP starting from the initial solution provided by GH
GH+RPD	RPD starting from the initial solution provided by GH
GH+SWP	SWP starting from the initial solution provided by GH
GH+SPD	SPD starting from the initial solution provided by GH
GH+SA	Simulated annealing starting from the initial solution provided by GH

Because there are no references in the literature for instance generation to our problem, we develop our own random problem instances to compare the performance of our proposed heuristics and discover some useful characteristics of the

CPIDP structure. First, we assume that the unit production cost is the same across the plants. This does not impact the analysis of the solution structure because in practice the unit production costs can be combined with the transportation costs and then generalized as the unit variable cost. Thus, unit production cost is not considered in the experiment. Unit transportation cost is proportional to the distance between plant i and DC k. The locations of plants i and DCs k are uniformly chosen from the square bounded by points (0,0) to (100,100). Transportation cost is expressed as follows:

$s_{ijk} = \delta \cdot dist(i,k)$ where δ: scaling factor (0,1), $dist(i,k)$: euclidean distance

Demands for a product at a DC, (d_{jk}), are uniformly selected over the interval (1000, 20000). The annual production rate is uniformly chosen over the interval $\left(|J|\cdot|K|\cdot d_l,\ \ |J|\cdot|K|\cdot d_h/2.0\right)$ where $d_l = 1000$ and $d_h = 20000$, because plant capacities are related to the number of products, the number of DCs, and the volume of demand. The data generated for the annual production rate at plants should be checked to ensure that the sum of plant capacities can satisfy the total demand from all DCs. The setup cost is generated uniformly in the range (500, 10000). We assume that setup time, f_{ij}, which affects cycle time decisions, is proportional to the setup cost. The setup time is then expressed as $a_{ij} = f_{ij}/\alpha$ $\ where\ \alpha = 10E6$. The scaling factor α is set to be so large because the unit for set up time a_{ij} is a year, which is consistent with the time units for all other parameters in the formulation. Unit holding cost h_{ij} is selected uniformly over the interval (10, 50). Some randomly generated data are based on the paper by Yung, Tang, Ip, and Wang (2006).

We choose nine problem configurations in our experiment. Each problem size is defined by the number of plants, products, and DCs. We generate ten random instances for each problem configuration. The six heuristics are tested on these 90 instances. Furthermore, in order to show whether the SA algorithm consistently converges to a good solution, we run ten independent replications of SA for each instance. We use different random number seeds for the replications, since SA is a randomized process.

To observe the impact of transportation and setup costs on the solution structure, we conduct sensitivity analysis explained further in Section 5.3. Five levels of the weight factor, W, are applied to the ratio of the weight on setup cost to the weight on transport cost: W1=0.01, W2=0.1, W3=1, W4=10, W5=100. The values for the ratios are carefully chosen through pilot experiments so that meaningful insights can be drawn. The sensitivity analysis is repeated for each problem instance so that we can generalize some of our observations on the behavior of production and distribution operations.

In measuring the performance of our heuristics for solving the CPIDP, we use the average percent error versus the best-known solution acquired from any tested heuristic, instead of measuring the optimality gap (Rardin, 2001), as explained next.

5.2. The Performance of Various Heuristics

As a performance measure, we define the average percent error as follows:

$$\frac{1}{r}\sum_{i=1}^{r}\left(\frac{TC(\text{each algorithm}) - TC(\text{best-known algorithm})}{TC(\text{best-known algorithm})}\right)$$

where $TC(\cdot)$ = total cost; r = number of replications.

For a given problem instance, we calculate the percent error of each heuristic's solution to the best-known heuristic for that instance. Once we calculate the percent error for each instance, we take the average across ten instances that was created for each problem configuration. For the SA, because we do ten independent replications using different random number seeds, we obtain ten different solutions. Thus, we compare the best solution provided by the SA to the other heuristics. To report how long the SA algorithm takes, the CPU time is added up for all ten replications of the SA. We also report the worst percent error for each problem configuration so that we can observe how reliable each heuristic is for various instances.

Tables 2 and 3 show that SA clearly outperforms other heuristics for all problem sizes in terms of the solution quality. The SA heuristic improves the solution quality by 0.88-13.63% on average. The worst percent error values show that greedy heuristic and local improvement heuristics are not reliable for various problem sizes and random input data. The worst-case percent errors for greedy heuristic and local improvement heuristics range from 1.8% to 23.04%.

On the other hand, the local improvement heuristics except the SPD have significantly shorter CPU times. Table 3 compares the execution time of each heuristic for various problem sizes. All heuristics are coded in Java and run on a Sun Fire V440 workstation with 16GB memory.

TABLE 2. PERCENT ERRORS FOR VARIOUS PROBLEM SIZES VERSUS ALGORITHMS

			GH	GH+RPW	GH+RPD	GH+SWP	GH+SPD	GH+SA
P1	5*10*5	Average	13.63%	8.07%	7.67%	6.14%	7.36%	0.02%
		Worst	21.48%	13.55%	13.10%	13.55%	12.46%	0.12%
P2	5*10*10	Average	9.14%	5.78%	3.00%	4.48%	2.67%	0.01%
		Worst	17.85%	12.78%	6.17%	10.58%	6.17%	0.11%
P3	5*20*10	Average	10.89%	3.35%	2.73%	2.73%	2.68%	0.00%
		Worst	23.04%	5.71%	4.64%	4.84%	4.50%	0.00%
P4	5*50*10	Average	10.68%	1.38%	2.56%	1.16%	2.75%	0.20%
		Worst	14.33%	3.21%	5.40%	3.45%	5.38%	1.95%
P5	10*10*10	Average	6.80%	5.31%	3.86%	4.99%	3.77%	0.00%
		Worst	11.80%	10.27%	8.12%	10.27%	8.86%	0.00%
P6	10*20*10	Average	6.63%	3.01%	2.81%	2.52%	2.89%	0.07%
		Worst	17.62%	4.43%	5.13%	4.21%	5.86%	0.69%
P7	10*20*20	Average	4.79%	2.47%	2.59%	2.43%	2.51%	0.00%
		Worst	12.64%	3.89%	4.31%	4.81%	4.72%	0.00%
P8	10*50*10	Average	4.48%	1.69%	2.32%	1.65%	2.42%	0.00%
		Worst	7.70%	3.51%	4.88%	3.39%	4.86%	0.04%
P9	10*50*20	Average	4.70%	0.88%	2.11%	0.89%	2.16%	0.06%
		Worst	8.48%	1.80%	3.35%	2.07%	3.44%	0.53%
	Overall average		7.97%	3.55%	3.29%	3.00%	3.25%	0.04%
	Overall worst		23.04%	13.55%	13.10%	13.55%	12.46%	1.95%

TABLE 3. AVERAGE CPU RUN TIME FOR VARIOUS PROBLEM SIZES (SEC)

	|I|*|J|*|K|	GH	GH+RPW	GH+RPD	GH+SWP	GH+SPD	GH+SA
P1	5*10*5	0.01	0.07	0.09	0.10	1.25	47.57
P2	5*10*10	0.04	0.09	0.14	0.21	10.06	133.66
P3	5*20*10	0.11	0.15	0.38	0.81	39.20	378.75
P4	5*50*10	0.43	0.58	0.93	3.60	219.09	1316.43
P5	10*10*10	0.16	0.24	0.65	1.13	43.28	494.20
P6	10*20*10	0.39	0.57	1.31	3.31	148.45	942.76
P7	10*20*20	1.09	1.34	2.87	6.04	904.99	2575.53
P8	10*50*10	1.63	2.01	2.89	18.19	826.69	2561.08
P9	10*50*20	4.35	4.95	8.26	28.90	4646.25	7353.84
Overall average		0.91	1.11	1.95	6.92	759.92	1755.98

As expected, the run time of the simulated annealing is very slow compared to the other heuristics. CPU run time of SA is added up for all 10 replications, because we choose the best solution across all 10 replications.

The reason that GH+SPD takes so much time compared to other local improvement heuristics is that it requires the largest $|I|*(|I|-1)*(|J|*|K|)^2$ iterations for a single run. Because GH+SPD is run repeatedly if there is any improvement during any iteration, execution time of GH+SPD increases steeply with increased problem sizes.

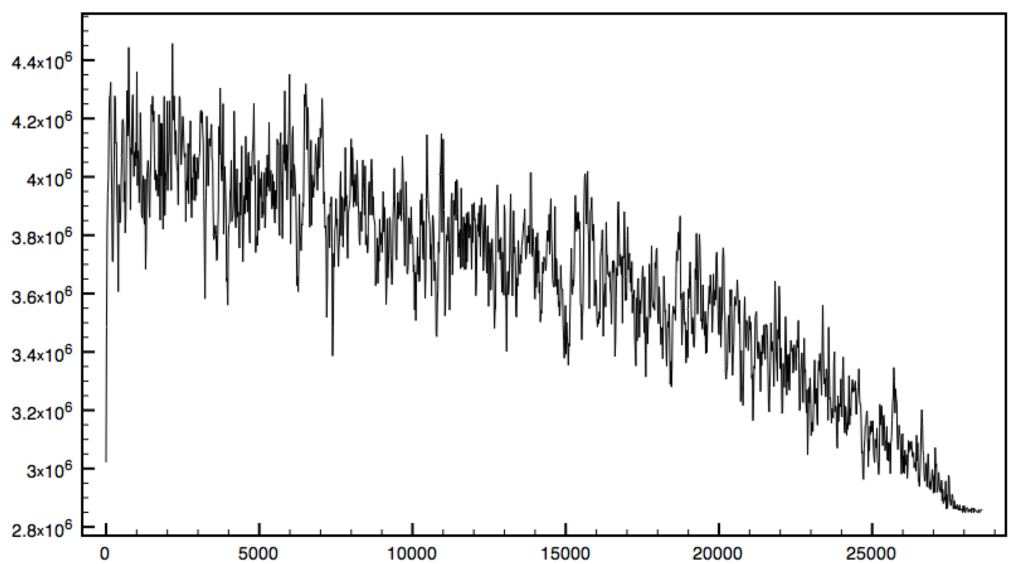

FIGURE 1. TOTAL COST VS. NUMBER OF ITERATIONS

TABLE 4. IMPACT OF TRANSPORT AND SETUP COSTS ON SOLUTION STRUCTURE

		Number of products		Number of plants	
		Average	Max	Average	Max
5-10-10	W1=0.01	8.60	10	4.30	5
	W2=0.1	7.98	10	3.99	5
	W3=1	4.64	9	2.32	4
	W4=10	2.88	5	1.44	3
	W5=100	2.26	3	1.13	2
5-50-10	W1=0.01	44.64	50	4.46	5
	W2=0.1	34.30	48	3.43	5
	W3=1	18.22	24	1.82	4
	W4=10	10.60	13	1.06	3
	W5=100	10.24	12	1.02	2
10-50-20	W1=0.01	38.33	50	7.67	9
	W2=0.1	30.67	49	5.99	9
	W3=1	16.20	28	3.24	6
	W4=10	9.05	13	1.81	3
	W5=100	5.52	8	1.10	2

Figure 1 shows the change of total cost vs. number of iteration in GH+SA heuristic for problem size 5*10*5. Each data point plotted corresponds to the total cost of the solution at that iteration that is either accepted due to having improved the total cost or with a probability distribution for up-hill climbing. From Figure 1, we can see that the SA heuristic experiences several local minima before it reaches the best-found solution.

5.3. Impact of Input Parameters

In Section 5.3.1, we investigate the impact of some important parameters on the solution and capture the specific characteristics of the CPIDP. In Section 5.3.2, we also examine how the solution quality would change with a slower or faster annealing process by adjusting the cooling rate.

5.3.1. Effects of transportation cost, production setup cost, and production rate

For each problem instance and weight factor, outlined in Section 5.1, we determine the average number of products assigned to each plant and the average number of plants that serve each product. Then we take the average of these averages across all 10 instances for that problem size and weight factor. The maximum number of products and plants are obtained in a similar way. Table 4 shows the results for three different problem sizes and five weight factors.

Since the weight factor is defined as the ratio of the weight on setup cost to the weight on transport cost, *W1*=0.01 indicates that transportation cost is significant. As the value of the weight factor increases, the setup cost becomes more significant.

As we see in Table 4, when transportation cost is very significant *(W1=0.01)*, the average number of products manufactured at a plant is close to the total number of products for each problem size. For example, 8.60 is close to 10 and 44.64 close to 50. This indicates that each plant serves as many products as possible to meet the demands at the nearest DC, within the capacity of that plant. That is because setup cost contributes to a relatively small portion of the total cost and thus we are not concerned about how frequently changeover happens. For three different problem sizes, each plant serves 77%~90% of entire products on average when transportation cost is very significant *(W1=0.01)*.

As the weight factor, W, increases, the setup cost becomes significant and the average number of products manufactured at a plant decreases, in that we cannot ignore the increased setup cost from frequent changeovers. Moreover, the average number of plants that manufacture each product also decreases. This means that each product is served by as small a number of plants as possible, so that the duplication of setup costs at various plants can be avoided.

When the setup cost is very significant ($W5$=100), each product is manufactured at a single plant and similarly, the average number of plants manufacturing each product is close to 1 for all problem sizes. For example, 1.13 for 5-10-10 and 1.02 for 5-50-10. Even the maximum number of plants that serve each product does not exceed 2. This means that each product is produced by a single plant minimizing the setup cost associated with switching from one product to another product.

As a result, transportation and setup costs have opposite effects on the solution structure for all problem sizes. Our various heuristics identify the equilibrium capturing the trade-offs between these conflicting costs. As a matter of fact, the results of Table 4 provide a quantitative basis for the strategic analysis that Schmenner (1979) has suggested. The pattern of production and distribution operation spans the range between the two extreme multiple-plant manufacturing strategies proposed by Schmenner (1979) based on the relative magnitudes of setup and transportation costs.

In the ELSP literature, the production plants are assumed to have excess capacity, e.g., $P_{ij} \gg \lambda_{ij}$. To understand the effects of production rate on the total cost components, we conduct additional sensitivity analysis. The results of this analysis are summarized for the first problem size. Similar observations are made for the other problem sizes.

TABLE 5. EFFECT OF PRODUCTION RATES ON COST COMPONENTS

Production Rate multiplier (PR)	Total Cost ($)	Transport Cost ($)	Percentage	Setup & Inventory Cost	Percentage
0.5	2906072	1954029	67%	952043	33%
1	2849022	1946866	68%	902156	32%
2	2907723	1973825	68%	933898	32%
3	2916572	2004699	69%	911873	31%
5	2928590	2004699	68%	923891	32%
10	2937488	2004699	68%	932789	32%

By comparing the solutions to PR=0.5 and PR=1 in Table 5, we can observe that in a tight capacity environment production assignments made by the heuristics might be suboptimal due to the tight capacity limits. As the production rate increases, production assignment quality improves resulting in better solutions. However, once the production rate increases to a certain level (*PR*=2), additional capacity does not contribute to an improvement in the quality of solutions. For larger capacities (*PR* ≥ 3), the production assignment remains the same, but results in a greater total cost due to increased production setup cost. That is because the increase in production rate decreases the optimal cycle time (see equation 19), thereby increasing the setup cost (see the first term in equation 3), assuming that all input parameters except production rate (e.g., *PR*=3,5, or 10) are the same. Moreover, the inventory holding costs may still vary, but is likely to increase (see the second term in equation 3). Through this sensitivity analysis, we provide insights into the appropriate range of production rates that will minimize the total cost.

5.3.2. Impact of cooling rate on the solution quality

To see how the solution quality would change with a slower or faster annealing process, the SA heuristic is tested with different cooling rates. Figure 2 illustrates that as the cooling rate increases the solution quality improves. It demonstrates that a slower annealing process in the SA heuristic produces higher quality solutions than a faster annealing process by doing more extensive search. For example, a slower annealing (Cooling rate > 0.85) improves 2~7% of solution for various problem sizes.

However, as we can see in Figures 2 and 3, there exists a trade-off between the solution quality and computational efficiency. As the annealing process becomes slower, the execution time increases steeply especially for large problem sizes. For example, an increase in the cooling rate from 0.95 to 0.99 improves the solution quality only slightly (0.54~0.91%), but the running time increase by about 300%. Therefore, we will need to find a balance between solution quality and computational efforts for SA heuristics.

In Section 5.4, we describe some of our observations on the Simulated Annealing process and provide some insights into how the SA works. As discussed in Section 4.4, clearly using the SPD in the SA process performs better than using the SWP, because the SPD provides a much larger neighborhood than the SWP. Perhaps more interestingly, another factor that may contribute to a better solution is related to the change in the total cost. Assuming that smaller product exchanges lead to smaller changes in the total cost, we can see that the SPD does not cause as large a change in the total cost as the SWP does. The change in cost affects the acceptance probability directly by $p = \exp\left(\frac{-\Delta E}{T}\right)$. If other SA parameter settings are the same, small changes in total cost are expected to lead to more opportunities for improving the solution in the middle to low ranges of temperature by allowing up-hill climbing in that the acceptance probability decreases more slowly.

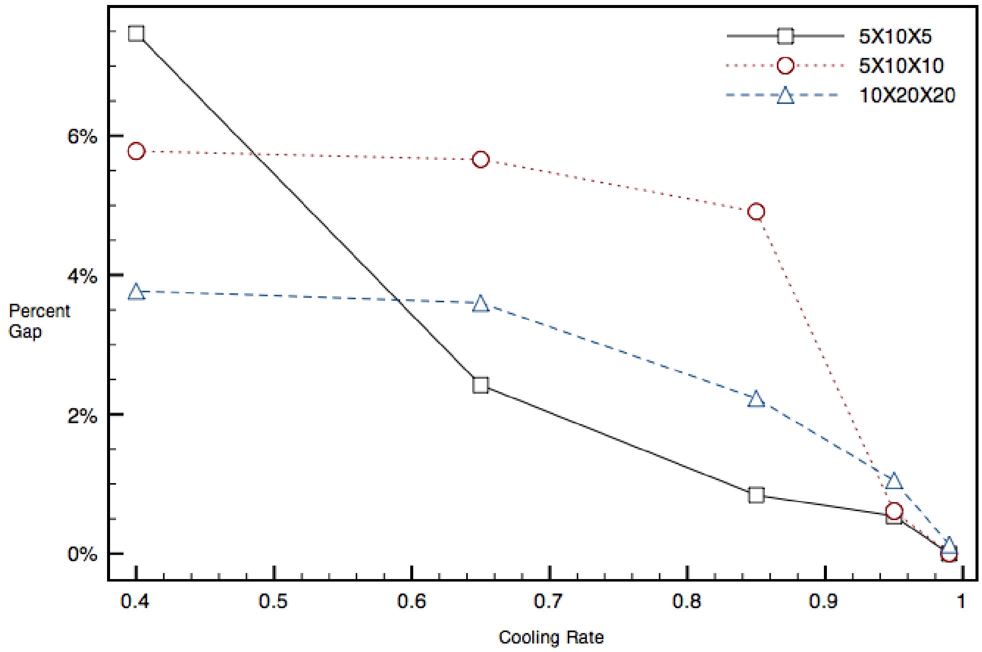

FIGURE 2. SOLUTION GAP VS. COOLING RATE

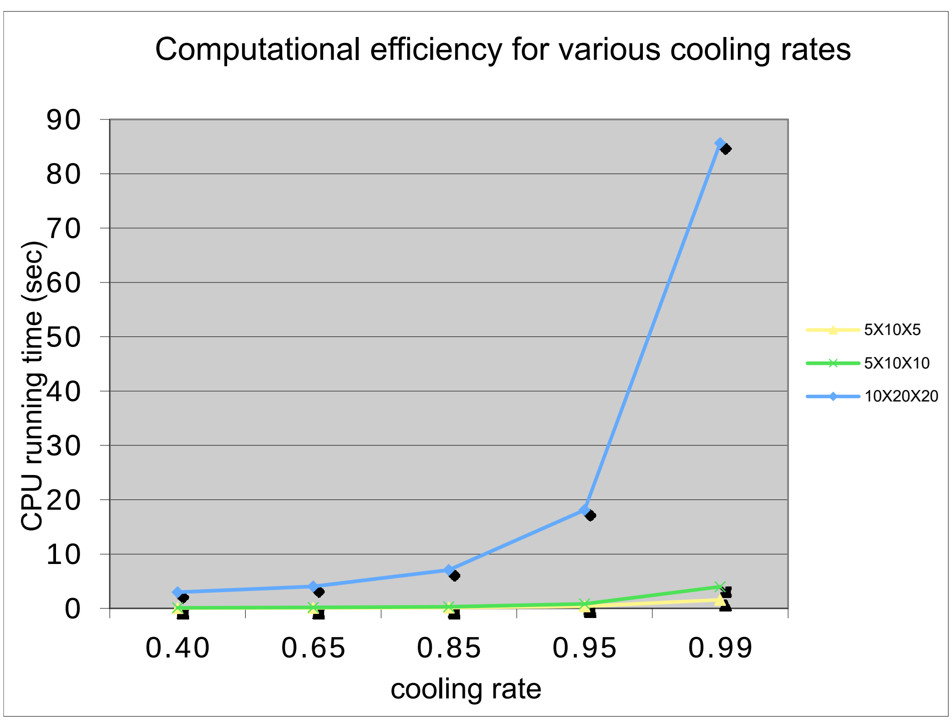

FIGURE 3. COMPUTATIONAL EFFICIENCY FOR VARIOUS COOLING RATES

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, we introduce a new integrated model that combines production, inventory, and transportation decisions with production setup time considerations in a complex network environment such as multiple products, multiple suppliers with limited capacities, and destinations. The CPIDP takes into account production capacity limitations at the plants; multiple products are manufactured, one at a time, on a single production line, in accordance with the production environment of a multi-national manufacturing firm. Due to this consideration, the CPIDP is able to capture the capacity limitations at the plants not just through fixed capacity constraints limiting average flow, but through setup time considerations.

The modeling assumptions result in non-linear terms in both the objective function and the constraints, rendering the CPIDP a difficult problem to solve. We develop various heuristics for solving the CPIDP and assess the performance of our heuristics for different problem sizes and the SA control parameter settings. An extensive computational experiment shows that the SA algorithm yields significantly better solutions than the other local improvement heuristics, being evidence of the effectiveness and potentiality of the SA as a solution approach to solve complex integrated production and distribution problems. On the other hand, nonlinear MILP solvers are unable to find even feasible solutions for most of the instances.

Finally, our investigation into the impact of input parameters on the solution structure provides decision makers with insights into the trade-offs between transportation costs and setup costs at the plants. Decision makers can use this model to assess the differences in the cost structure resulting from the implementation of various multi-plant strategies, which are often practiced by firms as summarized in Schmenner (1979). Moreover, the appropriate range of production rate helps managers avoid wasting the valuable resources of the company as well as make better decisions for minimizing the total cost.

As future research, we can develop some interesting extensions of the CPIDP. Currently the CPIDP assumes that all plants are already built in a specific location and so we are given the logistics network. However, by relaxing the fixed structure of a given logistic network and adding fixed costs for possible candidate locations, we can extend the CPIDP to a coordinated network design model that integrates the strategic decisions such as plant locations into operational and tactical decisions such as production lot sizes and shipments of multiple products. Moreover, we can also take into consideration the explicit costs or savings from closing plants that are already part of the existing logistic network if that plant does not manufacture any products in the process of optimizing the production and distribution operations.

As discussed in Section 5.3.1, the production rate can also be consideration and modeled as a decision variable so that we can make optimal decisions for production and distribution without wasting the valuable resources of a company such as production capacities. In addition, the demands at the DCs are currently given which may be appropriate for certain products. However, the CPIDP can be extended to the stochastic version, handling uncertainty in demands for various products.

ACKNOWLEDGMENT

The authors would like to thank Tae Hoon Kim for his contributions to this research.

REFERENCES

Benjamin, J. (1989). An Analysis of Inventory and Transportation Costs in a Constrained Network. *Transportation Science*, *23*(3), 177-183.

Bertazzi, L. & Speranza, M. (1999). Models and Algorithms for the Minimization of Inventory and Transportation Costs: a Survey. *New Trends in Distribution Logistics*, *480*, 137-157.

Blumenfeld, D., Burns, L., Diltz, J., & Daganzo, C. (1985). Analyzing Trade-offs Between Transportation, Inventory and Production Costs on Freight Networks. *Transportation Research Part B-methodological*, *19*(5), 361-380.

Bomberger, E. E. (1966). A Dynamic Programming Approach to a Lot Size Scheduling Problem. *Management Science*, *12*(11), 778-784.

Chandra, P. & Fisher, M. (1994). Coordination of Production and Distribution Planning. *European Journal of Operational Research Eur. J. Oper. Res*, *72*(3), 503-517.

Darvish, M., Larrain, H., & Coelho, L. C. (2016). A dynamic multi-plant lot-sizing and distribution problem. *International Journal of Production Research*, *54*(22), 6707-6717.

Drexl, A. & Kimms, A. (1997). Lot sizing and scheduling - Survey and extensions. *European Journal of Operational Research Eur. J. Oper. Res*, *99* (3), 221-235.

Eglese, R. (1990). Simulated Annealing - a Tool for Operational-research. *European Journal of Operational Research Eur. J. Oper. Res*, *46*(3), 271-281.

Elmaghraby, S. E. (1978) The economic lot scheduling problem (ELSP): Review and extensions. *Management Science*, 24 (6), 587-598

Glock, C. H. (2012). The joint economic lot size problem: A review. *International Journal of Production Economics*, *135*(2), 671-686.

Hanssmann, F. (1962). *Operations research in production and inventory control.* Wiley.

Hsu, W. (1983). On the General Feasibility Test of Scheduling Lot Sizes for Several Products on One Machine. *Management Science*, *29*(1), 93-105.

Jayaraman, V. & Ross, a. (2003). A Simulated Annealing Methodology to Distribution Network Design and Management. *European Journal of Operational Research*, *144*(3), 629-645.

Kim, J. & Kim, Y. (1996). Simulated Annealing and Genetic Algorithms for Scheduling Products With Multi-level Product Structure. *Computers & Operations Research*, *23*(9), 857-868.

Koulamas, C., Antony, S., & Jaen, R. (1994). A Survey of Simulated Annealing Applications to Operations-research Problems. *Omega-international Journal of Management Science*, *22*(1), 41-56.

Maxwell, W. L. (1964). Scheduling of Economic Lot Sizes. *Naval Research Logistics Quarterly*, *11*(2-3), 89-&.

Nahmias, S. (2004). *Production and Operations Analysis with Student CD.* McGraw-Hill/Irwin.

Ovacik, I.M. & Rajagopalan, S. & Uzsoy, R. (2000). Integrating Interval Estimates of Global Optima and Local Search Methods for Combinatorial Optimization Problems

Palmer, G. (1996). A Simulated Annealing Approach to Integrated Production Scheduling. *Journal of Intelligent Manufacturing*, *7*(3), 163-176.

Park, Y. B. (2005). An integrated approach for production and distribution planning in supply chain management. *International Journal of Production Research*, *43*(6), 1205-1224.

Rogers, J. (1958). A Computational Approach to the Economic Lot Scheduling Problem. *Management Science Manage. Sci*, *4*(3), 264-291.

Uzsoy, R. L. R., Reha. (2001). Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. *Journal of Heuristics*, *7*(3), 261-304.

Sarmiento, A. & Nagi, R. (1999). A Review of Integrated Analysis of Production-distribution Systems. *Iie Transactions*, *31*(11), 1061-1074.

Schmenner, R. (1979). Look Beyond the Obvious in Plant Location. *Harvard Business Review*, *57*(1), 126-132.

Silver, E., A. (2004). An Overview of Heuristic Solution Methods. *Journal of the Operational Research Society J. Oper. Res. Soc*, *55*(9), 936-956.

Tang, J., Yung, K., & Ip, A. (2004). Heuristics-based Integrated Decisions for Logistics Network Systems. *Journal of Manufacturing Systems*, *23*(1), 1-13.

Vidal, R. (1993). *Applied Simulated Annealing (lecture Notes in Economics & Mathematical Systems).* Springer-Verlag Berlin and Heidelberg GmbH & Co. K.

Wolsey, Laurence A. (1998). Integer Programming. Wiley-Interscience.

Yung, K., Tang, J., Ip, A., & Wang, D. (2006). Heuristics for Joint Decisions in Production, Transportation, and Order Quantity. *Transportation Science*, *40*(1), 99-116.

Zegordi, S. H., Abadi, I. K., & Nia, M. B. (2010). A novel genetic algorithm for solving production and transportation scheduling in a two-stage supply chain. *Computers & Industrial Engineering*, *58*(3), 373-381.

APPENDIX A.

Proof that the CPIDP is NP-hard

In order to prove that the CPIDP is NP-hard, we show that the CPIDP is reducible in polynomial time to the GAP, which is known to be NP-hard. (Laurence A. Wolsey, 1998)

To show that $GAP \underset{p}{\leq} CPIDP$, we consider the following special case of the CPIDP:

$$(P1)\ Minimize \sum_i \sum_j \sum_k \left(c_{ij} + s_{ijk}\right) d_{jk} X_{ijk}$$

$$subject\ to$$

$$\sum_i X_{ijk} = 1 \quad \forall j,k$$

$$\sum_k d_{jk} X_{ijk} = \lambda_{ij} \quad \forall i,j$$

$$\sum_j \frac{\lambda_{ij}}{p_{ij}} \leq 1 \quad \forall i$$

$$\lambda_{ij} \geq 0 \quad \forall i,j$$

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

Next, we consider only single product of the P1, the mathematical formulation is as follows:

$$(P2)\ Minimize \sum_i \sum_k \left(c_i + s_{ik}\right) d_k X_{ik}$$

$$subject\ to$$

$$\sum_i X_{ik} = 1 \quad \forall k$$

$$\sum_k d_k X_{ik} \leq p_i \quad \forall i$$

$$X_{ik} \in \{0,1\} \quad \forall i,k$$

Consider $\left(c_i + s_{ik}\right)d_k$ as the profit depending on agent *i*-task *k* assignment and consider d_k as the cost dependent on task *k*. Suppose each agent *i* has its own budget p_i. Then the P2 would be equivalent to the GAP.

Furthermore, because the special case of the CPIDP (P2) is a subset of the original problem (CPIDP), the P2 is polynomially reducible to the CPIDP ($SSP \underset{p}{\leq} CPIDP$). Since the P2 is equivalent to the GAP, the GAP is also polynomially reducible to the CPIDP ($GAP \underset{p}{\leq} CPIDP$). Therefore, the CPIDP is NP-hard.

APPENDIX B.

The second derivative test of $G(T_i)$ is as follows:

We define $G(T_i) = \sum_j \left(\frac{f_{ij}}{T_i} Y_{ij} + h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \frac{\lambda_{ij} T_i}{2} \right)$ where $T_i \geq 0 \quad \forall i = 1, \ldots, n$

We set the first derivative of $G(T_i)$ equal to zero and then use the sign of the second derivative to determine whether the resulting cycle time T_i is a minimum or maximum.

$$\frac{\partial G(T_i)}{\partial T_i} = 0 \quad \Rightarrow \quad -\frac{\sum_j f_{ij} Y_{ij}}{T_i^2} + \frac{\sum_j h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \lambda_{ij}}{2} = 0 \quad \forall i$$

$$T_i^{CC} = \sqrt{\frac{2 \sum_j f_{ij} Y_{ij}}{\sum_j h_{ij} \left(1 - \frac{\lambda_{ij}}{p_{ij}} \right) \lambda_{ij}}} \quad \forall i$$

$$\frac{\partial^2 G(T_i)}{\partial T_i^2} = 0 \quad \Rightarrow \quad \frac{2 \sum_j f_{ij} Y_{ij}}{T_i^3} > 0 \quad \forall i \quad \textit{when } T_i = T_i^{cc}$$

Therefore, T_i^{cc} is the optimal cycle time that minimizes $G(T_i)$.