Journal of Industrial and Systems Engineering
Vol. 11, No. 2, pp. 1-20
Spring (April) 2018

Optimal design of cross docking supply chain networks with time-varying uncertain demands

J. Behnamian[1], S.M.T.FatemiGhomi[2*], F.Jolai[3], M.Telgerdi[2]

[1] *Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran*
[2] *Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran*
[3] *Department of Industrial Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran*

behnamian@basu.ac.ir, fatemi@aut.ac.ir, fjolai@ut.ac.ir, m_telgerdi@aut.ac.ir

Abstract

This paper proposes an integrated network design model for a post-distribution cross-docking strategy, comprising multi product production facilities with shared production resources, capacitated cross docks with setup cost and customer zones with time windows constraints. The model is dynamic in terms of time-varying uncertain demands, whereas uncertainty is expressed with scenario approach and contains both ‘‘wait-and-see’’ and ‘‘here-and-now’’ decisions. Inventory is just permitted in plants and over several time periods. The objective of the model is to minimize the sum of the fixed location costs for establishing cross docking centers and inventory related costs across the supply chain while ensuring that the limited service rate of cross docking centers and production facilities, and also the lead time requirements of customers are not violated. The problem is formulated as a mixed-integer linear programming problem and solved to global optimality using CPLEX. Due to the difficulty of obtaining the optimum solution in medium and large-scale problems, two heuristics that generate globally feasible, near optimal solution, Imperialistic competitive algorithm (ICA) and simulated annealing (SA), are also proposed as heuristics. We find that CPLEX is not able to solve some of the sets to optimality and turned out to run out of memory, but it performs quite well for small test sets, as compared with the two heuristics. While SA is a faster heuristic method in terms of runtime, ICA generates better results on average, but in more time.

Keywords: Facilities planning and design, cross-docking, mixed integer model, heuristics.

1- Introduction

Driven by the increasingly demanding customer requirement, the logistics service practitioners have devoted a great deal of effort to enhance the quick response ability in the distribution system. Due to its widely perceived performance excellence, cross-docking has been recognized and accepted as a powerful

*Corresponding author

tool for speeding up the flow of goods and eliminating the warehouse in the supply chain. It is a way that accelerates the product flow to shorten the lead time from suppliers to customers, as well as an approach that reduces and even eliminates the inventory at the warehouse for goods are not stored in the warehouse but directly conveyed from receiving dock to shipping dock. Industrial applications of cross-docking can be found in Home Depot where transportation costs are reduced, in Wal-Mart which essentially eliminates its inventory holding cost, in COSCO which saves labor costs conventionally paid for order picking, packing and shipping, and in FedEx Freight which achieves cost-effective transportation (Shi, Liu, Shang and Cui, 2013).

The ‘‘problem’’ of cross docking network design is very broad and means different things to different enterprises. It generally refers to a strategic activity that will take one or more of the following decisions:

- Where to locate new facilities (be they production, storage, logistics, etc., when one of them is a cross dock).
- Significant changes to existing facilities, e.g. expansion, contraction or closure.
- Sourcing decisions—what suppliers and supply base to use for each facility.
- Allocation decisions—e.g., what products should be produced at each production facility; which markets should be served by which cross dock (Shapiro, 1999).

Making strategic decisions, such as where to locate cross docks, and more tactical/operational decisions, such as creating schedules, should preferably be a single decision process. Clearly, much more benefit could be achieved by simultaneously considering the production aspects and other issues related to integration of inventory, transportation, supplier selection, and investment budgeting decisions (Melo, Nickel and Saldanhada Gama, 2006)

One aspect of major importance in many decision making problems regards the need to deal with uncertain data. Cross docking location problems are no exception. In fact, a cross docking location problem is a network design problem which is often solved as part of a strategic decision making process and thus, the solution may have a long lasting effect and its implementation may take considerable time. Moreover, often, such implementation must be finished before the network starts being operated. Accordingly, several parameters involved in cross docking location problems may not be known precisely when the network design decisions are made. This is typically the case with the set-up costs for the facilities and with the demands to be transported between the nodes. Particularly, cross-docks have to operate today in an uncertain and dynamic environment, among others due to an intense competition in the transport and logistics sector and ever-increasing traffic. Dealing with uncertainty is important and flexibility becomes a major topic. Unrealistic assumptions and too rigid approaches prevent an efficient cross-dock operation. As the operational control of a cross-dock is a going concern, ‘one-shot optimization’ is not sufficient. Because of these complicated problems, it is worthwhile to consider approaches that proved to be useful in other domains. Most of the presented papers are not necessarily appropriate for a dynamic environment, as the considered (operational) problems are assumed to be static. Of course, this is a simplification of reality. The control of a cross-dock is a going concern and so these problems are inherently dynamic (Belle, Valckenaers and Cattrysse, 2012).

The value of cross-docking can be affected by the parameters of the system. Firstly, the structure of the supply chain can influence the benefits of cross-docking. For example, in a system where one cross-dock serves multiple stores, the number of stores may influence the system performance. Secondly, the probability distribution of marketing demand also influences the performance of cross-docking. Other factors, such as the variance of demand, order size, unit inventory holding and shortage cost, also affect the value of cross-docking (Tang, 2007).

Compared with a traditional distribution center where goods are first stored, cross-docking significantly changes the modes of members in the distribution system to handle the demand uncertainty related to inventory overstock and shortage cost, and is expected to achieve the economies of scale and to satisfy the customers' requirement better (Tang, 2007).

In this work we propose an integrated network design model for a post-distribution cross-docking strategy, comprising multiproduct production facilities with shared production resources, capacitated cross docks with setup cost and customer zones with time windows constraints, in which demands are both uncertain and time varying. Inventory is just permitted in plants and over several time periods. We are given a set of scenarios, each of which species a realization of the demands and has a fixed probability of occurrence. The objective is to find the standard min-expected-cost over all scenarios, subject to the constraints. Customers may be assigned to different facilities in different scenarios. (If customers must be assigned to the same facility in every scenario, the problem reduces to a deterministic problem in which the uncertain parameters are replaced by their means).

The objective of the model is to minimize the sum of the fixed location costs for establishing cross docking centers and inventory related costs across the supply chain while ensuring that the limited service rate of each cross docking center, and also the lead time requirements of customers are not violated. The consequent formulation is a linear mixed integer programming

By the knowledge of the authors, the problems present in the literature of cross docking location and cross docking network design are different from the proposed problem of this paper in the aspects:

1- All of the previous studies of cross docking location and cross docking network design are deterministic and this is the first work which considers uncertainty in cross docking location and network design problems. The model contains uncertain time varying demands which uncertainty is applied by considering different scenarios for different products in various periods.

2- All of the previous studies of cross docking location and cross docking network design are single period and this is the first work which considers multi periods with different lengths.

3- Network design models for cross docking should be based on not only the cost factors of traditional facility location models but also the operational considerations which may critically affect the performance of a cross docking system such as manufacturing decisions. None of the previous studies of cross docking location and cross docking network design problems have considered the manufacturing considerations in their works and have not applied the production constraints in their model.

4- This is the first work which considers single/multiple allocation at the same time in modeling cross docking network problem.

5- This is the first time in cross docking network design problems which different costs and constraints have considered for different products.

6- This is the first time which we have considered different time windows for the demand of different customers for different products for various plants.

This paper aims to fulfill the gap in cross docking network design models by proposing a multi-product, multi echelon cross docking network design model containing uncertain, time varying demands for products which operational considerations such as production and lead time constraints have applied. Due to the difficulty of this problem and its NP-hardness, two metaheuristics, Imperialistic competitive algorithm (ICA) and simulated annealing (SA), are proposed for solving the medium and large-scale problems.

2- Literature review

Cross-docking practitioners have to deal with many decisions during the design and operational phase of cross-docks. These decisions can have a serious impact on the efficiency, so they have to be carefully taken. In the literature, several decision problems are studied. Some of these problems are more concerned about decisions with effects on a longer term (strategic or tactical), while others deal with short-term decisions (operational). This section gives a review of the literature about cross- docking location and cross-docking networks.

The location of one or more cross-docks is part of the design of a distribution network or supply chain. An important strategic decision that has to be made concerns the position of these cross-docks. This problem cannot be handled isolated from the decisions that determine how the goods flow through this network.

2-1- Location of cross docks

The problem where to locate facilities (e.g. cross docks or plants) has attracted a considerable amount of attention. A first study about the location of cross-docks is performed by Sung and Song (2003). In the considered problem, goods have to be transported from supply to demand nodes via a cross-dock (direct shipments are not allowed). The cross-dock can be chosen from a set of possible cross-dock locations, each with an associated fixed cost. The demands are assumed to be deterministic and there are two types of vehicles with different capacity and cost. The aim is to find which cross-docks should be used and how many vehicles are needed on each link in order to minimize the total cost. This total cost consists of the fixed costs of the used cross-docks and the transportation costs. The authors present an integer programming model of the problem. This model is very similar to the model presented by Reeves et al.(1995) and Musa et al. (2010). Compared with these two papers, however, the approach of Sung and Song (2003) does not consider direct shipments but does include the location decision. Because the problem is NP-hard, a Tabu search-based algorithm is proposed to solve the model.

Sung and Yang (2008) extend this work and propose a small improvement to the Tabu search algorithm. The authors also present a set-partitioning-based formulation of the problem and propose a branch-and-price algorithm based on this formulation to obtain exact solutions. The computational results show that this algorithm gives better results in terms of the number of problem instances solved and the required computation time compared with the results obtained by solving the integer programming model using the optimization software ingredients CPLEX.

Gumus and Bookbinder (2004) study a similar problem, but now direct shipments are allowed and multiple products are considered. The facility cost for each cross-dock consists of a fixed cost and an operational cost charged per unit load. The transportation cost also has two components: a fixed cost for each truck and a variable cost per unit load per unit distance. Another cost which is taken into account is the cost for in-transit inventory. The authors provide a mixed integer programming model of the problem. The Influence of several cost parameters is studied by solving several smaller problem instances optimally (with the optimization software packages LINGO and CPLEX).A different approach is taken by Jayaraman and Ross (2003). They study a multi-echelon problem in which goods (from multiple product families) have to be transported from a central manufacturing plant to one or more distribution centers. From there, the goods are moved via cross-docks to the customers. The problem is tackled in two stages. In the first stage, a strategic model is used to select the best set of locations for the distribution centers and cross-docks. The authors provide an integer programming model that aims to minimize the fixed costs associated with operating open distribution centers and cross-docks and the various transportation costs. Demand splitting is not allowed, customers have to be assigned to single cross-docks while cross-docks have to be assigned to single distribution centers only. In the second stage, an operational model decides upon the quantities of each product type that need to be transported via distribution centers and cross-docks. The model aims to minimize the transportation costs while satisfying customer demand.

This model is less restrictive than the first model. It relaxes for instance the demand splitting assumption, individual vehicles are not considered and the transportation cost is proportional to the quantity to ship. The authors propose a simulated annealing approach to solve larger problem instances. The computational experiments on generated problem instances indicate that the heuristic gives results with a deviation of about 4% of the optimal solution (obtained with LINGO), but 300–400 times faster.

The same authors present two other heuristics to tackle the problem in (Ross and Jayaraman, 2008). Both heuristics are based on simulated annealing but use an extra mechanism to avoid local optimal solutions. The first heuristic makes use of a Tabu list; the second heuristic allows a sudden re-scaling of the 'system temperature'. For both heuristics, the solution quality and computational performance are tested for different 'cooling schemes'. The experimental results indicate that the simulated annealing heuristic combined with Tabu search gives better solutions in a more time. Bachlaus et al. (2008) consider a multi-echelon supply chain network, including suppliers, plants, distribution centers, cross-docks and customers. The goal is to optimize the material flow throughout the supply chain and to identify the optimal number and location of suppliers, plants, distribution centers and cross-docks. The problem is formulated as a multi-objective optimization model that aims to minimize the total cost and to maximize the plant and volume flexibility. Because of the computational complexity of the problem, the authors propose a variant of particle swarm optimization (PSO) to solve the problem. Computational experiments are done and the results show that the proposed solution approach gives better results than a genetic algorithm and two other PSO variants.

Mousavi et al. (2014) proposed a two-phase mixed-integer programming for the location of cross-docking facilities and vehicle routing scheduling problems in distribution networks. For this purpose, a two-phase mixed-integer programming (MIP) is formulated. They also developed a simulated annealing for solving the problem in the large-size instances. Yu et al. (2016) proposed a multi-period cross-docking distribution problem included manufacturers, cross-docks and customers with multiple products, consolidation of customer orders and time windows. To minimize the total cost, which includes transportation cost, inventory cost and penalty cost, they developed a particle swarm optimization algorithm with multiple social learning terms.

2-2- Cross-docking networks

Some authors do not study problems concerning a single cross-dock, but consider a network that contains one or more cross-docks. The aim is to determine the flow of goods through a network in order to reduce costs, while making supply meet demand.

The work of Lim et al. (2005) extends the traditional transshipment problem. The transshipment problem consists of a number of supply, transshipment and demand nodes. The links between these nodes have different capacity limits and costs. The objective is to find a minimum cost flow that meets all demands and the capacity constraints. In the extended transshipment problem, storage is allowed at the transshipment centers. These centers can be considered as cross-docks because the aim of the model is to minimize or eliminate holdover inventory. Moreover, this problem takes supplier and customer time windows into account and considers the capacity and holding costs of the cross-docks (Cho,2009). All shipments have to pass via a cross-dock, so no direct shipments are considered. Similar to the original problem, the objective is to minimize the total cost (transportation costs and holding costs) while satisfying demand, time windows and capacity constraints.

In the special case when only one delivery or departure is allowed within a time window and the departure and arrival times are fixed (single shipping–single delivery with fixed schedules), a genetic algorithm is developed by Miao et al. (2008). This heuristic gives better results (in terms of solution quality and computation time) than solving the integer programming formulation of the problem with CPLEX (with a time limit).

Chen et al. (2006) study a similar problem which they call the multiple cross-dock problems. The major differences are that supplies and demands are not-splittable and different products can be considered. Also, transportation time is not taken into account. An integer programming formulation of the problem is provided, together with a proof of its NP-completeness. The authors propose three heuristics (simulated annealing, Tabu search and a combination of both) to solve the problem. These heuristics provide better solutions than those obtained by solving the integer programming formulation with CPLEX, within only less than 10% the time used by CPLEX. Among the three heuristics, Tabu search seems to give the best results (Cho, 2009).

The previous studies represent the shipment of goods as flows. Individual transportation units are not considered and the transportation cost is proportional to the quantity to ship. However, to take advantage of consolidation, the vehicle transportation cost should be taken into account. A first approach that considers the transportation vehicles explicitly (and this is why the authors regard it as cross-docking) is taken by Donaldson et al. (1999). In the considered problem, the objective is to determine whether to route freight directly from suppliers to customers or via a cross-dock and how many vehicles should be scheduled on each transportation link in order to minimize the transportation costs.

Compared with the previous approaches, however, this problem is more simplified (e.g. storage at the cross-docks is not considered and the synchronization of inbound and outbound trucks is left out of the problem). The authors eliminate links with a large transportation time in an attempt to consider time windows. The authors present an integer programming model. Because the problem is difficult to solve with branch-and-bound algorithms, an alternative approach is proposed. In this approach, an iterative procedure is used in which either the integrality restrictions on the links from supply nodes to the cross-docks or on the links from the cross-docks to the destination nodes are relaxed. This relaxation heuristic provides near optimal solutions in an acceptable time. The authors used this approach to compare several scenarios (with a different number of cross-docks in different places) for the network design of a postal service company.

The same problem is also studied by Musa et al. (2010). They propose an ant colony optimization (ACO) heuristic to solve the problem and show that this heuristic gives in a short time slightly better results than a branch-and-bound approach with the optimization software package LINDO, which requires a much longer time.

Ma et al. (2011) takes most of the above-mentioned concerns into account. The so-called shipment consolidation problem (SCP) considers supplier and customer time windows and also the transportation times between the network nodes. Moreover, storage at the transshipment centers (cross-docks) is taken into account; shipments can be transported directly to their destination or via a cross-dock and the transportation cost accounts for the number of trucks. However, only one type of products is considered. The objective is to minimize the total cost (transportation and inventory cost) while satisfying the time windows constraints. The authors present an integer programming model of the problem and show that it is NP-complete in the strong sense. Therefore, the authors propose a (two-stage) heuristic algorithm to solve the problem. The computational experiments indicate that the proposed heuristic gives competitive results compared to CPLEX (with a time limit) within a much shorter time.

Yan and Tang (2009) compare the costs of a traditional distribution center with the cost of pre-distribution and post-distribution cross-docking. In pre-distribution cross-docking, it is assumed that the goods are directly loaded into outbound trucks. The suppliers are responsible for the necessary preparation and sorting to facilitate immediate loading at the cross-dock. This requires that the suppliers know the order quantities for each destination. In post-distribution cross-docking, the preparation and sorting happens at the cross-dock itself. This incurs higher costs at the cross-dock, but allows assigning the goods to destinations upon arrival at the cross-dock. In this way, the influence of the fluctuating demand can be reduced by pooling the risk during the transportation period from the supplier to the cross-dock. The results indicate that pre-distribution cross-docking is preferred when the demand is stable and the lead time between supplier and cross-dock is short. However, when the demand is uncertain and the lead time is long, the benefits of reallocating goods among stores outweigh the higher operational costs and so post-distribution cross-docking is preferred. Post-distribution also seems to be preferable if the number of destinations or the unit holding cost increases. The results of numerical experiments suggest that a higher uncertainty of demand, a higher unit transshipment cost and a longer lead time from the supplier to the cross-dock make post-distribution cross- docking more preferred. Pre-distribution cross-docking is more preferable when the unit holding cost or unit backorder cost is very high or very low.

Belle et al. (2012) have recently done a comprehensive review of the works which have been done in different aspects relating to cross docks. Their work is one of the last works done in the field of cross docking and has classified the entire cross docking papers' in seven groups: location of cross-docks, layout design, cross-docking networks, vehicle routing, dock door assignment, truck scheduling and temporary storage. In the last part of their paper those have investigated necessary research opportunities in this research area which on the subject of our work these are the most important fields which have not worked yet or needs more attention in future:

- Assuming Interchangeable products in a cross docking supply chain.
- Robustness against different kinds of deviations in supply chain.
- Integrating several problems in one approach (strategic and operational).
- Operating in an uncertain, dynamic environment.

Seyedhoseini et al. (2015) presented a mixed-integer model to optimize the location and design of cross docks at the same time to minimize the total transportation and operating costs. Their proposed model combines queuing theory for design aspects in the cross-docking network design. In this paper, a real case also had been examined to prepare an illustration for performance of the model. Recently, Buijs et al. (2014) presented a general classification scheme for cross-docking research based on the inputs and outputs for each problem aspect. After classifying the existing cross-docking research, they proposed several future research opportunities in developing decision models with practical and scientific relevance. They also described two real-life illustrative problems in cross-docking networks to highlight the importance of synchronization.

Ladier and Alpan (2016) proposed robust models for the truck scheduling model with time windows. In this study, the reformulations are based on minimax and minimization of the expected regret and resource redundancy and time redundancy. Their numerical study on different models showed that the methods based on resource redundancy give good results in the cross-docking case. Cóccola et al. (2015) considered a realistic problem studying the convenience of direct delivery, avoiding some cross-docking transfers in consolidated supply chains. They proposed a methodology for finding solutions based on the use of column generation embedded into an incomplete branch-and-price tree.

Enderer et al. (2017) considered an integrated cross-dock door assignment and vehicle routing problem arising in the operation of cross-dock terminals with the objective of minimizing the total material handling and transportation costs. This problem consists of assigning origins to inbound doors, transferring commodities between doors, and routing vehicles from outbound doors to destinations. They proposed two formulations of the problem as well as a column generation algorithm. Ahmadizar et al. (2015) developed a hybrid genetic algorithm and a model for two-level vehicle routing with cross-docking in a three-echelon supply chain. In this study, by considering the transportation costs and the fact that a given product type may be supplied by different suppliers at different prices, the routing of inbound vehicles between cross-docks and suppliers in the pickup process and the routing of outbound vehicles between cross-docks and retailers in the delivery process are determined. The goal is to assign products to suppliers and cross-docks, to optimize the routes and schedules of inbound and outbound vehicles, and to consolidate products so that the sum of the purchasing, transportation and holding costs is minimized.

Nikolopoulou et al. (2017) considers the problem of satisfying transportation requests from a set of suppliers to a set of customers for moving products between a pick-up and a delivery location pair. A commonly adopted approach is the direct-shipping of products, without using intermediate transshipment points, or in-transit merge of shipments. Another alternative strategy that often appears in practice is to use an intermediate cross-dock facility, acting as a consolidation point for transported products. The goal of this paper was to evaluate these inherently different distribution options and to conduct a comprehensive comparative analysis regarding their cost-effectiveness. Furthermore, the authors developed a local-search algorithm. Cota et al. (2016) undertake a study of truck scheduling in a cross-docking facility. They formulated the problem as a two-stage hybrid flow-shop problem, subject to cross-docking constraints with the objective of minimizing the makespan. For this study, the authors proposed a time-indexed mixed integer linear programming formulation and a polynomial time heuristic.

3- Problem statement

The problem considers the design of a multiproduct, three-echelon cross docking supply chain. Plants cannot produce all products included in the company’s portfolio. For each plant the production capacity and the availability of production resources is subject to certain constraints. Cross docks (if established) have specified maximum and minimum capacities. Some of the cross docks can be supplied from more than one production plant and some of them are single source. In the same manner, some of the customers can be supplied from more than one cross dock and the others are single source.

The volume of each good arriving at the cross docking center is specified in terms of packages for different products and the goods are interchangeable. It is assumed that two types of vehicles are available in the network. As in the typical trucking transportation network, the two types of vehicles are classified in terms of transportation cost and transportation time, as those of trucks and vans. Moreover; it is also assumed that the number of available vehicles is not limited under the situations where these vehicles are allowed to be rented easily in the market. So, the transportation costs can be calculated generally based on the associated transportation distances, types of vehicles and the type of the products.

A potential direct service is represented between the plants and customers, which refers to the possibility of routing freight demands from plants to customers without any intermediate stops allowed at cross docks. In order to deal with direct shipments, we have imagined cross docks in every plant location which the cost for transporting products between that plant and cross dock is zero but the cost of transporting products from that cross docks to customers is the direct shipment cost.

To ensure required customer service levels, it is often necessary to set time constraints for each customer demand. The lead time considered in this paper contains: time needed to transport products from plants to cross docks, operation time in cross docks and time needed to transport products from cross docks to customers. This implies that the service time feasibility for each plant destination path directing from an origin node through an intermediate node to the associated destination node can be decided based on fulfillment of transporting the associated freight demand within any required time limit.

4- Mathematical formulation

With making the following changes in Longinidis and Georgiadis (2009) our problem can be formulated as a MILP optimization model.

1. Elimination of warehouses as the supply chain network.
2. Substitution of distribution centers with cross docks.
3. Addition of a new set namely "vehicles"
4. Addition of transshipment time, and lead time
5. Addition customer service constraint

Before the model is presented, we introduce the parameters and indices employed. They are defined as follows.

Notations

Indices

e	Production resources
i	Products
k	Cross docks
L	Customer zones
S	Product demand scenarios
T	Time periods
V	Vehicles

sets

k^{ss}	Set of cross docks that should be supplied by a single plant
l^{ss}	Set of customer zones that should be supplied by a single cross dock

Parameters

C_{ik}^{DH}	Unit handling cost for product i at cross dock k
C_k^D	Annualized fixed cost of establishing cross dock at location k

C_{ij}^{p}	Unit production cost for product *i* at plant *j*
C_{ijkv}^{TR}	Unit transportation cost of product *i* transferred from plant *j* to cross dock *k* by vehicle *v*
C_{iklv}^{TR}	Unit transportation cost of product *i* transferred from cross dock *k* to customer *l* by vehicle *v*
C_{ijt}^{I}	Unit inventory cost of product *i* at plant *j* during time period *t*
$D_{k}^{\max}$	Maximum capacity of cross dock *k*
$D_{k}^{\min}$	Minimum capacity of cross dock *k*
$D_{ilt}^{[s]}$	Demand for product *i* from customer zone *l* during time period *t* under scenario *s*
$I_{ijt}^{[s],\min}$	Minimum inventory of product *i* held in plant j at the end of time period *t* under scenario *s*
$I_{ijt}^{[s],\max}$	Maximum inventory of product *i* held in plant j at the end of time period *t* under scenario *s*
NS	Number of product demand scenarios
$p_{ijt}^{[s],\max}$	Maximum production capacity of plant *j* for product *i* during time period *t* under scenario *s*
$p_{ijt}^{[s],\min}$	Minimum production capacity of plant *j* for product *i* during time period *t* under scenario *s*
$Q_{jk}^{[s],\min}$	Minimum rate of flow of products that can be transferred from plant *j* to cross dock *k* under scenario *s*
$Q_{kl}^{[s],\min}$	Minimum rate of flow of products that can be transferred from cross dock *k* to customer *l* under scenario *s*
$Q_{ijk}^{[s],\max}$	Maximum rate of flow of product *i* that can be transferred from plant *j* to cross dock *k* under scenario *s*
$Q_{ikl}^{[s],\max}$	Maximum rate of flow of product *i* that can be transferred from cross dock *k* to customer *l* under scenario *s*
R_{je}	total rate of availability of resource *e*, at plant *j*
Δt_{t}	Duration of time period *t*
T_{jkv}	Transshipment time between plant *j* and cross dock *k* by vehicle *v* (lead time from the plant j to cross dock *k* by vehicle *v*)
T_{klv}	Transshipment time between cross dock *k* and customer *l* by vehicle *v*
T_{ik}	Time spending on product *i* at cross dock *k*
ST_{lji}	Service time of customer *l* for product *i* in plant *j*
D_{k}	Capacity of cross dock *k*

Continuous variables

$I_{ijt}^{[s]}$	Inventory level of product i being held at plant j at the end of time period t under scenario s
$p_{ijt}^{[s]}$	Production rate of product i in plant j during time period t under scenario s
$Q_{ijkvt}^{[s]}$	Rate of flow of product i transferred from plant j to cross dock k by vehicle v during time period t under scenario s
$Q_{iklvt}^{[s]}$	Rate of flow of product i transferred from cross dock k to customer l by vehicle v during time period t under scenario s

Binary variables

Y_k	1 if cross dock k is to be established, 0 otherwise
$x_{ijkvt}^{[s]}$	1 if product i is to be transported from plant j to cross dock k by vehicle v during time period t under scenario s ,0 otherwise
$x_{iklvt}^{[s]}$	1 if product i is to be transported from cross dock k to customer l by vehicle v during time period t under scenario s ,0 otherwise

Greek symbols

γ_{ik}	Coefficient relating capacity of cross dock k for holding product i
ρ_{ije}	Coefficient of rate of utilization resource e in plant j to produce product i
ψ_s	Probability of product demand scenario s occurring during the life time of the network
ω_{ik}	Coefficient relating capacity of distribution center k to inventory of product i held

The model has time-varying uncertain demands in which uncertainty is modeled through a scenario approach. In this model, the expected value of the cost of the network taken over all the scenarios and during the operation of the network must be minimized. Now the problem can now be reformulated as:

$$\min \sum_t \Delta t_t (\sum_k C_k^D Y_k) + \sum_{s=1}^{NS} \psi_s (\sum_t \Delta t_t (\sum_{i,j} C_{ij}^p p_{ijt}^s + \sum_{i,j,k,v} C_{ijkv}^{TR} Q_{ijkvt}^{[s]} + \sum_{i,k,l,v} C_{iklv}^{TR} Q_{iklvt}^{[s]} + \sum_{i,k} C_{ik}^{DH} (\sum_{j,v} Q_{ijkvt}^{[s]}))) \tag{1}$$

$$+ \sum_{s=1}^{NS} \psi_s (\sum_t \Delta t_t (C_{ijt}^I \frac{I_{ijt}^{[s]} + I_{ij,t-1}^{[s]}}{2}))$$

$$S.t.: \quad x_{ijkvt}^{[s]} \le Y_k \quad \forall k \notin k^{ss} i, j, v, t, s = 1, \ldots, NS \tag{2}$$

$$\sum_j x_{ijkvt}^{[s]} = Y_k \quad \forall k \in k^{ss} i, v, t, s = 1, \ldots, NS \tag{3}$$

$$x_{iklvt}^{[s]} \le Y_k \quad \forall l \notin l^{ss} i, l, v, t, s = 1, \ldots, NS \tag{4}$$

$$\sum_k x_{iklvt}^{[s]} = 1 \quad \forall l \in l^{ss} i, v, t, s = 1, \ldots, NS \tag{5}$$

$$Q_{ijkvt}^{[s]} \le Q_{ijk}^{[s],\max} x_{ijkvt}^{[s]} \quad \forall i, j, v, k, t, s = 1, \ldots, NS \tag{6}$$

$$Q_{iklvt}^{[s]} \le Q_{ikl}^{[s],\max} x_{iklvt}^{[s]} \quad \forall i, l, v, k, t, s = 1, \ldots, NS \tag{7}$$

$$\sum_{i,v} Q_{ijkvt}^{[s]} \geq Q_{jk}^{[s],\min} x_{ijkvt}^{[s]} \quad \forall j,k,t,s=1,\ldots,NS \tag{8}$$

$$\sum_{i,v} Q_{iklvt}^{[s]} \geq Q_{kl}^{[s],\min} x_{iklvt}^{[s]} \quad \forall k,l,t,s=1,\ldots,NS \tag{9}$$

$$\sum_{j,v} Q_{ijkvt}^{[s]} = \sum_{l,v} Q_{iklvt}^{[s]} \quad \forall i,k,t,s=1,\ldots,NS \tag{10}$$

$$\sum_{k,v} Q_{ijkvt}^{[s]} \leq p_{ijt}^{s} \Delta t_t + I_{i,j,t-1}^{[s]} \quad \forall i,j,t,s=1,\ldots,NS \tag{11}$$

$$I_{ijt}^{[s]} = I_{i,j,t-1}^{[s]} + (p_{ijt}^{[s]} - \sum_{k,v} Q_{ijkvt}^{[s]})\Delta t_t \quad \forall i,j,k,v,t,s=1,\ldots,NS \tag{12}$$

$$\sum_{k,v} Q_{iklvt}^{[s]} = D_{ilt}^{[s]} \quad \forall i,l,t,s=1,\ldots,NS \tag{13}$$

$$p_{ijt}^{[s],\min} \leq p_{ijt}^{[s]} \leq p_{ijt}^{[s],\max} \quad \forall i,j,t,s=1,\ldots,NS \tag{14}$$

$$\sum_{i} \rho_{ije} p_{ijt}^{s} \leq R_{je} \quad \forall j,e,t,s=1,\ldots,N \tag{15}$$

$$D_k^{\min} Y_k \leq D_k \leq D_k^{\max} Y_k \quad \forall k \tag{16}$$

$$D_k \gamma_{ik} \geq \sum_{j,v} \omega_{ik} Q_{ijkvt}^{[s]} \quad \forall i,k,t,s=1,\ldots,NS \tag{17}$$

$$I_{ijt}^{[s]} \geq I_{ijt}^{[s],\min} \quad \forall i,j,t,s=1,\ldots,NS \tag{18}$$

$$I_{ijt}^{[s]} \leq I_{ijt}^{[s],\max} \quad \forall i,j,t,s=1,\ldots,NS \tag{19}$$

$$T_{jkv} x_{ijkvt}^{[s]} + T_{klv} x_{iklvt}^{[s]} + T_{ik} Y_k \leq ST_{l,j,i} \tag{20}$$

$$p_{ijt}^{[s]} \geq 0 \quad \forall i,j,t,s=1,\ldots,NS \tag{21}$$

$$I_{ijt}^{[s]} \geq 0 \quad \forall i,j,t,s=1,\ldots,NS \tag{22}$$

$$Q_{ijkvt}^{[s]} \geq 0 \quad \forall i,j,k,v,t,s=1,\ldots,NS \tag{23}$$

$$Q_{iklvt}^{[s]} \geq 0 \quad \forall i,k,l,v,t,s=1,\ldots,NS \tag{24}$$

The objective (1) is to minimize the overall expected cost of the cross-docking network and it includes the following five terms, i.e.,(*i*) fixed infrastructure cost $\sum_k C_k^D Y_k$, (*ii*) production cost $\sum_{i,j} C_{ij}^p p_{ijt}^{[s]}$, (*iii*)material handling cost at cross docks $\sum_{i,k} C_{ik}^{DH} (\sum_{j,v} Q_{ijkvt}^{[s]})$, (*iv*) inventory holding cost in plants $\sum_{i,j} C_{ijt}^{I} (I_{ijt}^{[s]} + I_{ij,t-1}^{[s]})/2$, as inventories vary linearly over each time period, and (*v*) transportation cost $\sum_{i,j,k,v,t} C_{ijkv}^{TR} Q_{ijkvt}^{[s]} + \sum_{i,k,l,v,t} C_{iklv}^{TR} Q_{iklvt}^{[s]}$. It is important to notice that to have a meaningful objective function, it is assume the probability of scenarios occurring in practice is known and is denoted by ψ_s and so we have $\sum_{s=1}^{NS} \psi_s = 1$.

Constraint (2) state that the link between a plant *j* and cross dock *k* can exist only if cross dock *k* is also established. Constraint (3) shows the certain cross docks can be served by a single plant, if needed. Constraint (2) ensures that a cross dock *k* and a customer *L* will link, if the cross dock also exists. Requirement (5) guarantees that some customer zones may be subject to a single sourcing constraint requiring that they be served by exactly one cross dock. Constraint (6) shows that if plant *j* to cross dock *k* connection exists, corresponding flow of material *i* can take place. Constraint (7) shows that if cross dock *k* to customer zone l connection exists, corresponding flow of material *i* can take place. Note that, the upper bound appearing on the right hand side of (7) can be calculated similar to the upper bound introduced in Tsiakis et al [22].Constraint sets (8) and (9) determine a minimum total flow rate of material in a transportation link between two locations in the network. Constraint (10) ensures that the amount of product *i* which is sent to cross dock *k* from different plants, must be equal with the amount of that product which is sent from that cross dock to the customers. Constraint (11) ensures that, for that product at the end of the previous period, the amount of each product that can be sent from each plant to different cross docks at each period must be less than the production rate of that product in that plant plus inventory. We suppose that inventory can be kept at plants. If no inventories were held at the plants, the actual rate of production of product *i* by plant *j* would equal the total flow of this product from plant *j* to all cross docks *k*. In this regard, Constraint (12) shows that the available inventory of product *i* held in plant *j* at the end of period *t* is equal to the inventory held at the end of period *t*-1 plus any product accumulated in the plant due to the production during the period, minus any product transported from the plant to cross docks during the same period. Constraint (13) states the total flow of each product *i* received by each customer zone l from the distribution centers is equal to the corresponding market demand. Constraint (14) ensures that the orders received from the cross docks show the ability of the manufacturing plants to cover the customers' demands. The production's rate of each product at any plant cannot exceed certain bounds. This constraint also shows always there is a minimum production capacity for any of the products. Moreover, this equality ensures that there is often a minimum production rate that must be maintained while the plant is operating. Since in our model, it is assumed some resources can be used by several production lines and at different stages of the production of each product, Constraint (15) states such share usage limits the availability of the resource. In this constraint, coefficient ρ_{ije} and R_{je} are the amount of resource that used by plant *j* to produce a unit amount of product *i* and the total rate of availability of resource at plant *j*, respectively. Regarding the capacities of cross docks and the quantity of products which can be stored temporarily before their conveyance at the market, Constraint (16) established the lower and upper bounds for the capacity of a cross dock *k* between given $D_k^{\min}$ and $D_k^{\max}$, respectively. Constraint (17) states the capacity of a cross dock for each kind of products cannot be less than the sum of the currents from different plants to that cross dock for that kind of product at any time period under each scenario. Constraints (18-19) establish the needing bounds for safety stock. Each customer has a special service time for different products manufactured in different plants that must be satisfied. The constraint below states this limitation for different customers. The sum of the time to transship the products from plant j to customer l must be less or equal the customer service time (20). All continuous variables must be non-negative (21-24).

5- Heuristic algorithms

Solving combinatorial optimization problems amounts to finding the 'best' or 'optimal' solution among a finite or denumerable number of alternative solutions (Davis,1993). Over the past 40 years, a wide variety of combinatorial problems has appeared from computer science, engineering, and business logistics (such as the cross-docking problem studied here), and procurement. It has been shown that these combinatorial problems belong to the class of NP-complete problems (Papadimitriou and Steiglitz, 1998). Therefore, solving these problems to optimality cannot be obtained in reasonable amounts of computational time, and the solution approaches include enumeration or heuristic approaches. In both approaches, it is desirable to have algorithms that yield high quality solutions in little computation time that are also applicable to a wide variety of combinatorial optimization problems (Ross and Jayaraman, 2008).

The simulated annealing algorithm, which is one of the selected approaches employed in the current study, is one such high quality general algorithm that is both a randomized (search) algorithm and asymptotically an optimization algorithm (Garey and Johnson, 1979). As a meta-heuristic, simulated annealing has become one of the most popular because of its ease of use and the asymptotic results of convergence to optimal solutions (Ross and Jayaraman, 2008).

Imperialist competitive algorithm is proposed by Atashpaz and Lucas. (2007). They showed the algorithm's capability in dealing with different types of optimization problems (2008). This algorithm is also used by Shokrollahpour et al. (2011) they have calibrated the parameters of this algorithm using the Taguchi method. In comparison with the best algorithm proposed previously, the ICA indicates an improvement and their results have been confirmed statistically. Our aim is to compare their ICA algorithm, as an effective population based algorithm with SA, one of the most popular algorithms because of its ease of use and the asymptotic results of convergence to optimal solutions.

5-1- Simulated annealing

We have used the CROSS-SA algorithm of Jayaraman and Ross (2003) and have made a change in the algorithm which helps the algorithm to avoid local optimal solutions and helps to find new best solutions not in the neighboring of the just best solution. For this point, after each perturbation of the configured system if for example, 20 consecutive evaluations yield no improvement, we start to generate a population of random solutions over all the solution area, and then we compute the cost of the generated population and get the best solution. Here we must compare the new solution with the last best solution of the algorithm and continue the algorithm.

The search for least cost solutions is guided by a control parameter known as temperature, T, and the rate at which we systematically lower it, called alpha. Both alpha and the temperature determine the acceptance of inferior solutions.

5-1-1- CROSS-SA heuristic step

CROSS_SA heuristic algorithm of Jayaraman and Ross (2003) has 7 steps which we have added a new step (step 6) to the algorithm which helps the algorithm to avoid local optimal solutions by finding new best solutions not in the neighboring of the just best solution.

Step 1: Initialization. Initial and final values of the control parameter temperature, known as T_0 and T_f respectively, are specified. Alpha, the cooling rate is specified along with the maximum number of iterations at each temperature value. An initial cross-dock solution is randomly generated by assigning customers-demand flows between cross-docks, and finally to the manufacturers. Status indicators on the facilities and product family assignments are set. This results in an initial feasible solution providing product flows. The objective function value of this solution becomes the objective function value for the best configuration found, incumbent configuration, C_b and the newest configuration $C_{b'}$. All counters are set to 1.

Step 2: Check feasibilities. The algorithm now evaluates product flow assignments for cross docks and manufacturers to ensure no capacity violations exist. We also check the other constraints to be satisfied. If the configuration is not feasible, we return to step 1.

Step 3: Generate a feasible neighboring solution. Once the problem has been initialized, an objective function value is computed, and feasibility ensured, the current feasible cross-dock system configuration is then perturbed (modified) by selecting several customer zones and reassigning their demands between cross-docks. This is accomplished by randomly selecting some customer zones to perturb. Their flow is randomly assigned to another cross-dock combination. All feasibilities are checked once again.

Step 4: Evaluate incumbent solution with neighboring solution. If the objective function value of the neighboring solution is greater than that of the incumbent ($C_{b'} > C_b$), proceed to step 5. Otherwise, if the objective function's value of the newest configuration improves over the incumbent ($C_{b'} < C_b$), the neighboring solution becomes the incumbent. We then compare this solution to the best solution found thus far. If the objective function value of the newest configuration is less than that of the best one found so far, then replace the best solution with that of the neighboring solution. Proceed to step 7.

Step 5: Examine Metropolis condition. Determine the difference between the neighboring solution and the incumbent solution as: $\Delta C = C_{b'} - C_b$. The Metropolis criterion is then used to determine the probability at which the relatively inferior neighboring solution should be accepted. This probability is computed as:

$$PROB(A) = e^{-\frac{\Delta C}{T_c}}$$

Where T_c is the current temperature. A random number is then generated from the interval (0,1), If this random number is less than $PROB(A)$. Then the neighboring solution replaces the incumbent. Proceed to step 7.

Step 6: jumping in the solution area: after each perturbation of the configured system if for example, 20 consecutive evaluations yield no improvement, we start to generate a population of random solutions over all the solution area, and then we compute the cost of the generated population and get the best solution. Here we go to step 4 and check this new solution in the algorithm.

Step 7: Increment counters. Update memory and status variables. Increase the counters by one. If the iteration counter value is less than or equal to the maximum iterations for the temperature level, then return to step 3. Otherwise, go to step 7.

Step 8: Adjust temperature. Adjust temperature by the cooling rate. Mathematically this is $T_{c+1} = T_c \times \alpha \quad \alpha \in [0.2, 0.99]$; where T_c is the temperature used to compute acceptance probability at iteration i and α is the cooling rate in (0, 1). If the new value of T_c is greater than or equal to the stopping value T_f , then reset iteration counters to one and return to step 3. Otherwise, stop.

5-2- Imperialist competitive algorithm

Imperialist competitive algorithm is proposed by Atashpaz and Lucas (2007). They showed the algorithm's capability in dealing with different types of optimization problems (2008).

Similar to other evolutionary algorithms, this algorithm starts with an initial population of solution which is named country. Some of the best countries in the population are chosen to be the 'imperialists' and the rest are the 'colonies' of these imperialists. All the colonies of initial population are distributed among the imperialists based on their power. A set of one imperialist and its colonies is called an 'empire' (Shokrollahpour, Zandieh and Dorri, 2011).

The power of an empire which is equivalent to the fitness value in a genetic algorithm (GA) is inversely proportional to its cost. After distribution of all colonies among imperialists, these colonies start moving towards their relevant imperialist country. The total power of an empire relates to both the power of the imperialist country and the power of its colonies. This fact will be modeled by defining the total power of an empire by adding the percentage of the mean power of colonies to their imperialists. Then the imperialistic competition begins among all the empires. Any empire which is not strong enough to compete and cannot increase its power (or at least prevent decreasing it) will be eliminated. The imperialistic competition will lead slightly to an increase in the power of powerful empires and a decrease in the power of weaker ones. Weak empires will lose their power and finally they will collapse. The movement of colonies towards their relevant imperialists through the competition among empires and also the collapse mechanism will hopefully cause all the countries to converge to a state in which there is just one empire in the world and all the other countries are colonies of that empire. In this ideal new world, colonies have the same position and power as the imperialist (Shokrollahpour, Zandieh and Dorri, 2011).The implementation of this algorithm in assembly flow shop is as follows:

Begin ICA

1. Initialize the empires.
2. Move the colonies toward their relevant imperialist (assimilating).
3. If there is a colony in an empire which has a lower cost than that of imperialist, exchange the positions of that colony and the imperialist.
4. Compute the total cost of all empires (related to the power of both imperialist and its colonies).
5. Pick the weakest colony from the weakest empire and give it to the empire that has the most likelihood to possess it (imperialistic competition).
6. Change some weakest colonies with new ones randomly (revolution).
7. Eliminate the powerless empires.
8. If stopping criteria met, stop, if not, go to step 2.

End ICA (Shokrollahpour et al., 2011).

6- Numerical experiment

To evaluate simulated annealing and imperialistic competitive algorithm performance on the cross-docking problem, several problems were developed. The setup for each problem is described by the number of cross-docks, number of customer zones, number of plants and number of product families.

Table 1. Computational design of the problem

Problem	Product families (I)	Plants (J)	Cross docks (K)	Customer zones (L)
1	14	3	15	18
2	21	4	20	27
3	21	4	30	36
4	28	6	30	36
5	42	9	45	54

The heuristic algorithms are coded in Matlab and executed on a of Intel(R) core(TM) i5 CPU 2.4 GHz PCs with 4GB memory. For comparison, we also test the MIP model given in CPLEX, a popular off-the-shelf optimization software package. The purpose of the experiments is to check the validity and effectiveness of the developed model and the effectiveness of the proposed heuristic methods by comparing the results with what CPLEX is able to provide with no time limit.

For our Cross-SA, max-rejection for the new solutions is set on 250, max run for the model is set on 500, max acceptance of new solutions is set on 500, the initial search for the heuristic is set on 500 and we have considered five amounts for alpha. The initial temperature for the heuristic is set on 1and the minimum temperature to end the heuristic is set on a very close amount to zero.

For our Cross-ICA, the initial population contains 100 countries, number of initial imperialist is set on 20, number of decades is set on 50, the revolution rate for the heuristic is set on 0.3 and Zeta is set on 0.02.

The computational results are summarized in table 2. For each cell, one problem was solved ten times by the SA , ICA and CPLEX and the best performance of each problem is reported. Performance analysis of approximation algorithms such as simulated annealing concentrates on solution quality and running time. Therefore, algorithm performance was measured by solution quality and computation time.

Table 2. Computational results

Test set	**CPLEX**			**SA**				**ICA**		
I-J-K-L	Cost	Time (s)	Gap (%)	α	Min cost	Time (s)	Gap (%)	Min cost	Time (s)	Gap (%)
14.3.15.18	5498211	27	0.0037	0.2	5722587	283	0.0408	5637149	572	0.0253
				0.45	5718229	311	0.0400			
				0.65	5707326	330	0.0380			
				0.85	5700204	356	0.0367			
				0.95	5665128	413	0.0304			
21.4.20.27	20532488	124	0.0057	0.2	21142202	1553	0.0297	20964535	3823	0.0210
				0.45	21041445	2056	0.0248			
				0.65	21040406	2256	0.0247			
				0.85	20992447	2690	0.0224			
				0.95	20982446	2922	0.0219			
21.4.30.36	23418827	633	0.0063	0.2	24130092	2738	0.0304	23810115	4859	0.0167
				0.45	24090165	3214	0.0287			
				0.65	23938160	3552	0.0222			
				0.85	23945081	3968	0.0225			
				0.95	23932395	4205	0.0219			
28.6.30.36	---------	------	--------	0.2	29499917	3410	-----	29197685	5302	-----
				0.45	29494202	3742	-----			
				0.65	29296712	4028	-----			
				0.85	29334494	4501	-----			
				0.95	29204339	4885	-----			
42.9.45.54	---------	------	--------	0.2	66945453	4950	-----	66820331	6655	-----
				0.45	66967829	5211	-----			
				0.65	66952094	5752	-----			
				0.85	66950094	6175	-----			
				0.95	66909158	6476	-----			

As can be seen for the two big sizes of problem CPLEX turned out to run out of memory, which made it quite difficult to summarize or to take the average of the computational results. From Table 2, we find that CPLEX is not able to solve some of the sets to optimality, but it performs quite well for small test sets, as compared with the two heuristics. As the size of test sets increases, performance of ICA is close to

CPLEX and performance of SA is worse than ICA. While SA is a faster heuristic method in terms of runtime, ICA generates better results on average.

7- Conclusion and future works

In this work we propose an integrated network design model for a post-distribution cross-docking strategy, comprising multi product production facilities with shared production resources, capacitated cross docks with setup cost and customer zones with time windows constraints. The model is dynamic in terms of time-varying uncertain demands, whereas uncertainty is expressed with scenario approach and contains both ‘‘wait-and-see’’ and ‘‘here-and-now’’ decisions. Inventory is just permitted in plants and over several time periods.

The objective of the model is to minimize the sum of the fixed location costs for establishing cross docking centers and inventory related costs across the supply chain while ensuring that the limited service rate of cross docking centers and production facilities, and also the lead time requirements of customers are not violated.

The problem is formulated as a mixed-integer linear programming problem and solved to global optimality using CPLEX/GAMS. Two heuristics that generate globally feasible, near optimal solution, Imperialistic competitive algorithm and simulated annealing, are also proposed as heuristics for solving the problem.

The proposed MILP model aims to assist senior operations management to take decisions about production allocation, production capacity per site, purchase of raw materials and network configuration taking into account transient demand conditions. The purpose of the model is to be used not as frequent as an advanced planning scheduling (APS) system (daily, weekly or monthly) but for longer periods (such as quarterly, six months or yearly) to address strategic and tactical supply design aspects. Its allocation decisions are set as production targets for the APS systems to optimize production sequences.

From table 2, we find that CPLEX is not able to solve some of the sets to optimality, but it performs quite well for small test sets, as compared with the two heuristics. As the size of test sets increases, performance of ICA is close to CPLEX and performance of SA is worse than ICA. While SA is a faster heuristic method in terms of runtime, ICA generates better results on average.

For improving the applicability, the approaches should be more robust and dynamic. Discussions with cross-docking practitioners reveal that there are (serious) deviations between the predicted and actual information. Particularly, cross-docks have to operate today in an uncertain and dynamic environment. Dealing with uncertainty is important and flexibility becomes a major topic. Unrealistic assumptions and too rigid approaches prevent an efficient cross-dock operation. Most papers are concerned with just one problem. Furthermore, as these problems are interdependent, improvements are expected when they can be solved together. So, future research is required that integrates several problems in one approach. Another subject which should be considered is lateral transshipment between plants or customers which will reduce the effect of uncertainty in cross docking supply chains.

References

Ahmadizar, F., Zeynivand, M., & Arkat, J. (2015). Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. *Applied Mathematical Modelling*, *39*(22), 7065-7081.

Atashpaz-Gargari, E., Hashemzadeh, F., & Lucas, C. (2008, June). Designing MIMO PIID controller using colonial competitive algorithm: applied to distillation column process. In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*(pp. 1929-1934). IEEE.

Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *Evolutionary computation, 2007. CEC 2007. IEEE Congress on* (pp. 4661-4667). IEEE.

Bachlaus, M., Pandey, M. K., Mahajan, C., Shankar, R., & Tiwari, M. K. (2008). Designing an integrated multi-echelon agile supply chain network: a hybrid taguchi-particle swarm optimization approach. *Journal of Intelligent Manufacturing*, *19*(6), 747.

Buijs, P., Vis, I. F., & Carlo, H. J. (2014). Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research*, *239*(3), 593-608.

Chen, P., Guo, Y., Lim, A., & Rodrigues, B. (2006). Multiple crossdocks with inventory and time windows. *Computers & operations research*, *33*(1), 43-63.

Cho, N. (2009). *Integrated network design models for crossdocking and warehousing strategies with tactical considerations* (Doctoral dissertation, Purdue University).

Cóccola, M., Méndez, C. A., & Dondo, R. G. (2015). A branch-and-price approach to evaluate the role of cross-docking operations in consolidated supply chains. *Computers & Chemical Engineering*, *80*, 15-29

Cota, P. M., Gimenez, B. M., Araújo, D. P., Nogueira, T. H., de Souza, M. C., & Ravetti, M. G. (2016). Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. *Computers & Industrial Engineering*, *95*, 135-143.

Davis, T. (1993). Effective supply chain management. *Sloan management review*, *34*(4), 35.

Donaldson, H., Johnson, E., Ratliff, H., & Zhang, M. (1999). Schedule-driven cross-docking networks: Technical report. *Georgia Institute of Technology*.

Enderer, F., Contardo, C., & Contreras, I. (2017). Integrating dock-door assignment and vehicle routing with cross-docking. *Computers & Operations Research*, *88*, 30-43.

Garey, M. R. (1979). DS Johnson Computers and intractability. *A Guide to the Theory of NP-Completeness*.

Gümüş, M., & Bookbinder, J. H. (2004). Cross-docking and its implications in location-distribution systems. *Journal of Business Logistics*, *25*(2), 199-228.

Jayaraman, V., & Ross, A. (2003). A simulated annealing methodology to distribution network design and management. *European Journal of Operational Research*, *144*(3), 629-645.

Ladier, A. L., & Alpan, G. (2016). Robust cross-dock scheduling with time windows. *Computers & Industrial Engineering*, *99*, 16-28.

Lim, A., Miao, Z., Rodrigues, B., & Xu, Z. (2005). Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics (NRL)*, *52*(8), 724-733.

Longinidis, P., & Georgiadis, M. C. (2011). Integration of financial statement analysis in the optimal design of supply chain networks under demand uncertainty. *International journal of production economics*, *129*(2), 262-276.

Ma, H., Miao, Z., Lim, A., & Rodrigues, B. (2011). Crossdocking distribution networks with setup cost and time window constraint. *Omega*, *39*(1), 64-72.

Melo, M. T., Nickel, S., & Da Gama, F. S. (2006). Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research*, *33*(1), 181-208.

Miao, Z., Fu, K., Fei, Q., & Wang, F. (2008, June). Meta-heuristic algorithm for the transshipment problem with fixed transportation schedules. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 601-610). Springer, Berlin, Heidelberg.

Mousavi, S. M., Tavakkoli-Moghaddam, R., & Siadat, A. (2013). Optimal design of the cross-docking in distribution networks: Heuristic solution approach. *International Journal of Engineering-Transactions A: Basics*, *27*(4), 533.

Musa, R., Arnaout, J. P., & Jung, H. (2010). Ant colony optimization algorithm to solve for the transportation problem of cross-docking network. *Computers & Industrial Engineering*, *59*(1), 85-92.

Nikolopoulou, A. I., Repoussis, P. P., Tarantilis, C. D., & Zachariadis, E. E. (2017). Moving products between location pairs: Cross-docking versus direct-shipping. *European Journal of Operational Research*, *256*(3), 803-819.

Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial optimization: algorithms and complexity*. Courier Corporation

Reeves, C. R. (1995). Modern heuristic techniques for combinatorial problems. Advanced topics in computer science. *Modern Heuristic Techniques for Combinatorial Problems: Advanced Topics in Computer Science*.

Ross, A., & Jayaraman, V. (2008). An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. *Computers & Industrial Engineering*, *55*(1), 64-79

Seyedhoseini, S. M., Rashid, R., & Teimoury, E. (2015). Developing a cross-docking network design model under uncertain environment. *Journal of Industrial Engineering International*, *11*(2), 225-236.

Shapiro, J. F. (1999). On the connections among activity-based costing, mathematical programming models for analyzing strategic decisions, and the resource-based view of the firm. *European Journal of Operational Research*, *118*(2), 295-314.

Shi, W., Liu, Z., Shang, J., & Cui, Y. (2013). Multi-criteria robust design of a JIT-based cross-docking distribution center for an auto parts supply chain. *European Journal of Operational Research*, *229*(3), 695-706.

Shokrollahpour, E., Zandieh, M., & Dorri, B. (2011). A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem. *International Journal of Production Research*, *49*(11), 3087-3103.

Sung, C. S., & Yang, W. (2008). An exact algorithm for a cross-docking supply chain network design problem. *Journal of the Operational Research Society*, *59*(1), 119-136

Sung, C. S., & Song, S. H. (2003). Integrated service network design for a cross-docking supply chain network. *Journal of the Operational Research Society*, *54*(12), 1283-1295.

Tang, S. (2007). *The impacts of cross-docking operation on supply chain management* (Doctoral dissertation, The Hong Kong Polytechnic University).

Tsiakis, P., Shah, N., & Pantelides, C. C. (2001). Design of multi-echelon supply chain networks under demand uncertainty. *Industrial & Engineering Chemistry Research*, *40*(16), 3585-3604.

Van Belle, J., Valckenaers, P., & Cattrysse, D. (2012). Cross-docking: State of the art. *Omega*, *40*(6), 827-846.

Yan, H., & Tang, S. L. (2009). Pre-distribution and post-distribution cross-docking operations. *Transportation Research Part E: Logistics and Transportation Review*, *45*(6), 843-859.

Yu, V. F., Jewpanya, P., & Kachitvichyanukul, V. (2016). Particle swarm optimization for the multi-period cross-docking distribution problem with time windows. *International Journal of Production Research*, *54*(2), 509-525.