Journal of Industrial and Systems Engineering
Vol. 9, No. 3, pp 1-16
Summer (July) 2016

Truck scheduling problem in a cross-docking system with release time constraint

Jamal Arkat[1*], Parak Qods[1], Fardin Ahmadizar[1]
Department of Industrial Engineering, University of Kurdistan
j.arkat@uok.ac.ir, parak.qods@gmail.com, f.ahmadizar@uok.ac.ir

Abstract

In a supply chain, cross-docking is one of the most innovative systems for ameliorating the operational performance at distribution centers. Cross-docking is a logistical strategy in which freight is unloaded from inbound trucks and (almost) directly loaded into outbound trucks, with little or no storage in between, thus no inventory remains at the distribution center. In this study, we consider the scheduling problem of inbound and outbound trucks with multiple dock doors, aiming at the minimization of the makespan. The considered scheduling problem determines where and when the trucks must be processed; also due to the interchangeability specification of products, product assignment is done simultaneously as well. Inbound trucks enter the system according to their release times', however, there is no mandatory time constraint for outbound truck presence at a designated stack door; they should just observe their relative docking sequences. Moreover, a loading sequence is determined for each of the outbound trucks. In this research, a mathematical model is derived to find the optimal solution. Since the problem under study is NP-hard, a simulated annealing algorithm is adapted to find the (near-) optimal solution, as the mathematical model will not be applicable to solve large-scale real-world cases. Numerical examples have been done in order to specify the efficiency of the metaheuristic algorithm in comparison with the results obtained from solving the mathematical model.

Keywords: Cross-docking, Truck scheduling, Release time, Simulated annealing algorithm.

1- Introduction

Many logistics companies are trying to develop new distribution strategies in order to operate their supply chains in an efficient manner. The endeavor to find new strategies is a consequence of customers ordering small quantities of various products and at the same time demanding a more accurate and timely delivery. Cross-docking is one of the innovative strategies to minimize unnecessary inventory and enhance the customer service level (Apte and Viswanathan, 2000). This is a logistics strategy nowadays used by many companies in different industries (e.g. retail firms and less-than-truckload (LTL) logistics providers). The basic idea behind cross-docking is to transfer incoming shipments directly to outbound vehicles without storing them in between. This practice can serve different goals: the consolidation of shipments, a shorter delivery lead time, the reduction of costs, etc.

*Corresponding author.
ISSN: 1735-8272,

The four major functions of warehousing in a traditional distribution center are: receiving, storage, order picking and shipping, and cross-docking serves to eliminate the two most expensive handling operations: storage and order picking. Hence cross-docking can be described as *"the process of consolidating freight with the same destination (but coming from several origins), with minimal handling and with little or no storage between unloading and loading of the goods"*(Van Belle, Valckenaers and Cattrysse, 2012). In this strategy, storage is not allowed unless it is for a short period of a time. An explicit limit is hard to define, but many authors assume that 24h is a maximum storage time (eg.Bartholdi and Gue, 2004). In practice, a cross-dock has multiple loading doors (or dock doors) where trucks can be loaded or unloaded. Inbound trailers are assigned to strip doors for unloading their freight, and then the unloaded products are transferred to their appropriate stack doors in order to be loaded on the outbound trailers.

Cross-docking corresponds with the goals of lean supply chain management, which includes smaller volumes of more visible inventories that are delivered more frequently and faster. Furthermore we can specify several advantages of employing cross-docking in comparison with traditional distribution centers: cost reduction (warehousing costs, inventory-holding costs, handling costs and labor costs), shorter delivery lead time from supplier to customer, improved customer service, reduction of storage space, faster inventory turnover, fewer overstocks, reduced risk for loss and damage. Despite the fact that these advantages make cross-docking an interesting logistic strategy that can give companies substantial competitive advantages, it is not always the best strategy to be employed in every case and every condition. Apte and Viswanathan (2000) discussed some crucial factors that influence the suitability of cross-docking compared with traditional distribution. The two most important factors are product demand rate and unit stock-out costs. Therefore, it is better to use cross-docking when we have products with stable demand rates and low unit stock-out costs and traditional warehousing is preferred for the opposite situation with an unstable demand and high unit stock-out costs. If we have an unstable product demand rate and low unit stock-out costs or vice versa, still cross-docking can be used when proper systems and planning tools are in place to keep the number of stock-outs to a reasonable level. In order to distinguish between different types of cross-docking, several traits are considered that can be divided into three categories: physical characteristics, which includes shape, number of dock doors, and internal transportation, operational characteristics, which lead to service mode and preemption, and flow characteristics, which include arrival pattern, departure time, product interchangeability, and temporary storage.

There are many decisions that cross-docking practitioners have to tackle during the design and operational phases of cross-docks. The efficiency of cross-docking highly depends on these decisions; therefore, they should be taken carefully. An extensive review of the existing literature about cross-docking problems is done by Van Belle, Valckenaers and Cattrysse (2012), which range from strategic and tactical to operational problems. The strategic decisions for implementing a cross-docking system are those made about the cross-dock's location and layout design. Once the cross-dock is available, it will be a part of a supply network (with one or more cross-docks). A tactical decision that has to be made then is how the goods will flow through the network, and then the operational decisions include vehicle routing, dock door assignment, truck scheduling and temporary storage. Since we are concerned with truck scheduling problems in this research, we just present a brief review of the existing literature about truck scheduling problems in a cross-docking system. The truck scheduling problem is concerned with the assignment of inbound and outbound trucks to different dock doors of a cross-dock (Boysen and Fliedner, 2010,Van Belle, Valckenaers and Cattrysse, 2012). We can regard the dock doors as resources that should be scheduled over time. *Where* (at which door) and *when* a truck should be docked to start its processing is determined by finding a solution to the problem. To distinguish between different types of truck scheduling problems, Van Belle, Valckenaers and Cattrysse (2012) divide the papers in three categories. The first category considers cross-docks with only one strip and one stack door. Therefore in this case, truck scheduling is reduced to sequencing the inbound and outbound trucks. The second category includes studies which consider cross-docks with multiple inbound and outbound doors but only tackle the scheduling of inbound or outbound trucks. And the last category is dedicated to the studies which consider scheduling of both inbound and outbound trucks with multiple dock doors.

In the first category, which is concerned with scheduling trucks in a cross-dock with only one strip and stack door, Yu and Egbelu (2008) developed a mixed integer mathematical model with the objective of minimizing the makespan of a cross-docking operation, when there is just a single strip and a single stack door. No arrival and departure times are considered, and the products are assumed to be interchangeable. Hence, the assignments of products from inbound to outbound trucks must be determined as well. In addition, a truck changeover time is taken into account, and the transferring time between the strip and the stack door is fixed. They assumed that unloading the products can be done in any sequence. Due to the limitation of the loading and unloading dock doors, if a product is unloaded from an inbound truck and the destined outbound truck has not arrived for loading, then the product will be put into temporary storage. Therefore, they proposed nine sequencing strategies for both inbound trucks and outbound trucks to minimize the number of products passing through temporary storage and thereby reduce the total operation time. In order to increase the efficiency of the operation of the cross docking, Maknoon and Baptiste (2009) proposed a dynamic programming algorithm, an evolutionary algorithm and a heuristic approach. They maximized the ratio between the total number of directly transiting products to the total number of transiting products by optimizing the sequence of both inbound and outbound trucks. They assumed a constant transferring time inside the cross-dock as there is only one strip and one stack door. For scheduling the trucks in the cross-docking system based on the proposing model by Yu and Egbelu (2008), Vahdani and Zandieh (2010)applied five metaheuristic algorithms, GA, TS, SA, electromagnetism-like algorithm (EMA) and VNS. According to the obtained results, VNS is recommended for scheduling trucks in cross docking systems. For the same cross-docking systems, five metaheuristic algorithms were applied by Boloori Arabani, Fatemi Ghomi and Zandieh (2011): GA, TS, particle swarm optimization (PSO), ACO and differential evolution (DE). Based on the obtained analysis, the GA, PSO, ACO and DE algorithms have relatively similar behavior in acquiring the best objective function, makespan, while the TS shows different results. A dynamic programming method for optimizing the sequence of inbound and outbound trucks at cross-docking terminals was studied by Boysen, Fliedner, and Scholl (2010). They merged individual handling times for products with service slots to which inbound and outbound trucks are assigned. This approach is different from what Yu and Egbelu (2008)employed, as they dealt with more detailed handling times of products that are in principle hard to obtain. A slot comprises the time required for completely unloading an inbound truck and completely loading an outbound truck. Boloori Arabani, Fatemi Ghomi, and Zandieh (2010) dealt with a multi-criteria cross-docking scheduling for customers whose manufacturing systems are just-in-time. The simultaneous minimization of two criteria (earliness and tardiness) was the aim of this study. Three metaheuristics, GA, PSO and DE, were proposed for the scheduling problem. In order to solve the cross-docking scheduling problem, Boloori Arabani, Zandieh and Fatemi Ghomi (2011)addressed three famous multi-objective algorithms including non-dominated sorting genetic algorithm-II (NSGA-II), strong Pareto evolutionary algorithm-II (SPEA-II), and sub-population genetic algorithm-II (SPGAII). The objective was to minimize the total operational time and the total lateness of all outbound trailers. Boloori Arabani, Zandieh and Fatemi Ghomi (2012)dealt with a scheduling problem of inbound and outbound trucks in a cross-docking system. Two objectives, minimization of the total operation time (makespan) and minimization of the total lateness of outbound trucks, were taken into account simultaneously. For solving the above-mentioned problem, three multi-objective algorithms based on the sub-population concept of evolutionary algorithms were developed. Storage in cross-docking systems is a subject that Sadykov (2012) studied, and he aimed to reduce it. When products are unloaded at the inbound door but the corresponding outbound truck is not immediately available at the outbound door, they are temporarily stocked in a storage area and cause increasing storage costs. The problem under study is concerned with scheduling both inbound and outbound trucks when there is just a single strip and single stack door, and for optimizing it, a dynamic programming algorithm is proposed.

The second category concerns with the problems that consider the scheduling of inbound or outbound trucks, one of the problems was studied by McWilliams, Stanfield and Geiger (2005, 2008). They considered scheduling inbound trucks at a cross-dock used in the parcel delivery industry, in which unloaded parcels are transported to outbound trucks by a fixed network of conveyors. The assignment of doors as either strip or stack doors is fixed, as this is a stationary network. The transferring time of the parcels is dependent on the assignment of trucks to dock doors, and also on congestion of the conveyor network. They presented a simulation-based scheduling algorithm to minimize the makespan. Since simulation optimization is computationally expensive, a decomposition approach was also proposed to tackle a similar problem (McWilliams 2009, McWilliams 2010). The objective was then to balance the workload. A minimax model was derived for the problem and was solved with several (meta-) heuristic methods. A dynamic version of this problem was also studied by McWilliams (2009). A multi-door cross-docking problem was considered by Alpan, Larbi and Penz (2011) in which temporary storage was allowed. When given an inbound truck sequence, a bounded dynamic programming was proposed for determining the optimal outbound truck sequence, in a way that the total cost was minimized. Also Boysen, Briskorn and Tschöke (2013) considered an operational truck scheduling problem in a cross-docking system with multiple doors. They assumed that the scheduling of all outbound trucks was done beforehand, and the corresponding departure times were fixed. Therefore, the aim of this study was the optimization of the inbound truck sequence. Provided a good was not loaded onto the designated outbound truck before its departure, profit was lost. In order to maximize the total profit, heuristics called decomposition procedures and simulated annealing were developed. Six metaheuristic algorithms–SA, TS, ACO, DE and two hybrid DE algorithms–were proposed by Liao, Egbelu and Chang (2013) to solve a multi-door cross-docking problem in which dock door assignment and the sequence of inbound trucks were both taken into consideration. The aim of this paper was to minimize the total weighted tardiness. Konur and Golias (2013)assumed that inbound truck arrival times were to be unknown to the cross dock operator. However, the operator knows the incoming trucks arrival time windows, i.e. the lower and upper bounds on the truck arrival times. A GA based heuristic was proposed for finding Pareto-efficient schedules for inbound trucks. In the research of Alpan et al. (2011), Boysen et al. (2013), Liao et al. (2013) and Konur and Golias (2013), only the sequences of inbound trucks or outbound trucks were considered to be optimized.

The literature related to the third category, which consists of scheduling both inbound and outbound trucks, is rare. A dynamic programming and simulated annealing method for truck scheduling in the cross-docking operation was developed by Boysen (2010). This work dealt with a special truck scheduling problem arising in the (zero-inventory) cross-docks of the food industry, where strict cooling requirements forbid an intermediate storage inside the terminal, so that after products are unloaded from the inbound trucks, the products must be loaded on the outbound trucks, directly. The problem is formalized such that different operational objectives, i.e. the flow time, processing time and tardiness of outbound trucks, are minimized. Lee, Kim and Joo (2012) proposed a mixed integer programming (MIP) model for door-assigning and sequencing of trucks in a multi-door cross-docking problem. For maximizing the number of products that can be shipped within a given working horizon several GAs were proposed. Joo and Kim (2013) considered a truck scheduling problem for three types of truck–inbound trucks, outbound trucks and compound trucks. The compound trucks play the roles of inbound trucks and outbound trucks. Two metaheuristic algorithms, GA and self-evolution algorithm were proposed for minimizing makespan. In the research of Boysen (2010), Lee et al. (2012) and Joo and Kim (2013), the moving times of products between different inbound doors and outbound doors were assumed to be the same. However, the time requirements for moving goods inside the cross-docking generally depend on the corresponding distance between the dock doors to which the respective inbound and outbound trucks are assigned. When taking multiple inbound doors and multiple outbound doors into consideration, the assignment of trucks to inbound doors and outbound doors should be taken into account. Yiyo Kuo (2013)studied a problem which aimed to improve the efficiency of multi-door cross-docking by optimizing both inbound and outbound truck sequencing and both inbound and outbound truck dock assignment. The objective was to minimize the makespan. In order to optimize the problem, a model for calculating the makespan was proposed. When given a sequence of all inbound and outbound trucks, the calculation model could assign all inbound and outbound trucks to all inbound and outbound doors based on first come first served (FCFS) strategy and then calculate the makespan. The proposed makespan calculation model was then integrated with a variable neighborhood search (VNS) which could optimize the sequence of all inbound and outbound trucks. Moreover, four Simulated Annealing (SA) algorithms were adopted for comparison.

This research considers a scheduling problem concerning the assignment of trucks to dock doors, and likewise determining the docking sequences of inbound and outbound trucks at each door. The objective of this study is to find the best truck scheduling, so that the makespan is minimized. This research is an extension of the model presented by Lee, Kim and Joo (2012), in which no mandatory constrains exist for the presence of outbound trucks at their designated stack doors; moreover, there is a releasing time for each of the inbound trucks, and they might not be available in the beginning of the working horizon. Also, for each of the outbound trucks a loading sequence is determined, such that assigned products from each inbound truck can be transferred independently, and the ones which have arrived sooner at the designated stack door must be loaded first. The outline of this paper is as follows: in the next section, the mathematical model is presented. Section 3 presents the metaheuristic algorithm, and Section 4 shows a numerical example. At the end, a conclusion and future research direction are provided in Section 6.

2- Problem description

This research aims to find the best scheduling of inbound and outbound trucks in a cross-docking system with multiple dock doors in order that the makespan is minimized. Hence a mixed integer programming model is derived to find the optimal solution–determining where and when inbound and outbound trucks should be processed. In addition to specifying the assignment of inbound and outbound trucks to doors and the docking sequences, product assignments must be determined simultaneously. A loading sequence for each of the outbound trucks will be determined as well, such that assigned products from each inbound truck can be transferred independently, and the ones which have arrived sooner at the designated stack door must be loaded first. To clarify the problem under study, we first state the basic assumptions that are taken into account.

- There is a releasing time for each of the inbound trucks, so they might not be available in the beginning of the working horizon.
- All outbound trucks are available in the beginning and waiting for loading in the outbound truck yards.
- An exclusive mode of service is considered, i.e. each dock door is assigned either to inbound or to outbound trucks.
- Arriving goods are unloaded from the inbound trucks and transferred to the appropriate outbound docks where they are loaded into outbound trucks. Other internal operations–like sorting and labeling–are not considered. Sufficient personnel and equipment are assumed to be available for performing all loading, unloading and transferring operations.
- Preemption of loading or unloading a truck is not allowed. So, a docked truck has to be completely processed before it leaves the dock.
- For an inbound truck which has arrived at an inbound door, only one product unit can be unloaded at a time. The total unloading time of an inbound truck is dependent on the number of product units to be unloaded.
- The time needed to transfer goods from inbound to outbound trucks is directly proportional to the rectilinear distance between the dock doors to which the trucks are assigned, as the transfer velocities are all the same.
- Intermediate storage inside the cross-dock is allowed. This means that the products assigned to each outbound truck can be transferred to the appropriate stack door before its arrival. The capacity of the storage area is infinite.
- The truck changeover time is fixed.
- Products are interchangeable.
- There is a loading precedence for each outbound truck, such that the goods which arrive earlier must be loaded first.

Based on the aforementioned assumptions, a mixed integer programming model is derived for calculating the makespan. The following notation is used in the mathematical model:

Indices

$i, i' = 1, 2 \dots I$ indices of inbound truck

$j, j' = 1, 2 \dots J$ indices of outbound truck

$l = 1, 2 \dots L$ index of inbound door

$m = 1, 2 \dots M$ index of outbound door

Parameters

T_{lm} the distance between inbound door l and outbound door m

Tw_i releasing time of inbound truck i

r_i the number of product units that are initially loaded in inbound truck i

S_j the number of product units that are initially needed for outbound truck j

UD unit time for loading or unloading products

Tc truck changeover time

N a large positive number

Variables

X_{ij} the number of units of product that are transferred from inbound truck i to outbound truck j

EI_i the time that inbound truck i enters the inbound door

EJ_j the time that outbound truck j enters the outbound door

DI_i the time at which inbound truck i leaves the inbound door

DOJ_{ij} the time at which loading of products transferred from inbound truck i to outbound truck j is completed

DJ_j the time that outbound truck j leaves the outbound door

LT_{ij} the time at which productstransferred from inbound truck i to outbound truck j can be loaded

C_{max} makespan

v_{ij} a binary variable that is one if at least one product unit transferred from inbound truck i to outbound truck j; otherwise it is zero.

ID_{il} a binary variable that is one if inbound truck i is assigned to inbound door l; otherwise it is zero.

OD_{jm} a binary variable that is one if outbound truck j is assigned to outbound door m; otherwise it is zero.

$p_{ii'l}$ a binary variable that is one if truck i preceded truck i' in the inbound truck sequence at strip door l when $i \neq i'$ or truck i is the first truck at strip door l when $i = i'$; otherwise it is zero.

$q_{jj'm}$ a binary variable that is one if truck j preceded truck j' in the outbound truck sequence at stack door m when $j \neq j'$ or truck j is the first truck at stack door m when $j = j'$; otherwise it is zero.

Y_{ij} a binary variable that is one if the assigned products from inbound truck i arrived at the designated stack door before the presence of the respective outbound truck j;otherwise it is zero.

$G_{i'ij}$ a binary variable that is one if $v_{ij} = v_{i'j} = 1$ and truck i' precedes truck i in the loading sequence for outbound truck$j$ when $i \neq i'$ or $v_{ij} = 1$ and products from truck i have the first priority for loading when $i = i'$; otherwise it is zero.

$H_{ii'j}$ a binary variable that is one if the assigned products from inbound truck i for outbound truck j arrived at the relative stack door before the loading of assigned products from inbound truck i' on outbound truck j is finished; otherwise it is zero.

The mixed integer programming model is given below:

$$Min\ C_{max} \tag{1}$$

Subject to:

$$\sum_{j=1}^{J} X_{ij} = r_i\ ;\ \forall i \tag{2}$$

$$\sum_{i=1}^{I} X_{ij} = S_j\ ;\ \forall j \tag{3}$$

$$X_{ij} \leq Nv_{ij}\ ;\ \forall i, j \tag{4}$$

$$\sum_{l=1}^{L} ID_{il} = 1\ ;\ \forall i \tag{5}$$

$$\sum_{m=1}^{M} OD_{jm} = 1\ ;\ \forall j \tag{6}$$

$$\sum_{i=1}^{I} P_{iil} = 1\ ;\ \forall l \tag{7}$$

$$\sum_{i'=1}^{I} P_{i'il} = ID_{il}\ ;\ \forall l, i \tag{8}$$

$$\sum_{\substack{i'=1 \\ i'\neq i}}^{I} P_{ii'l} \leq ID_{il}\ ;\ \forall l, i \tag{9}$$

$$\sum_{j=1}^{J} q_{jjm} = 1\ ;\ \forall m \tag{10}$$

$$\sum_{j'=1}^{J} q_{j'jm} = OD_{jm}\ ;\ \forall m, j \tag{11}$$

$$\sum_{\substack{j'=1 \\ j'\neq j}}^{J} q_{jj'm} \leq OD_{jm}\ ;\ \forall m, j \tag{12}$$

$$EI_i \geq Tw_i\ ;\ \forall i \tag{13}$$

$$DI_i = EI_i + UDr_i\ ;\ \forall i \tag{14}$$

$$DI_i + T_c \leq EI_{i'} + N\left(1 - \sum_{l=1}^{L} P_{ii'l}\right) \; ; \; \forall i, i', i \neq i' \tag{15}$$

$$LT_{ij} = DI_i \upsilon_{ij} + \left(\sum_{l=1}^{L}\sum_{m=1}^{M} T_{lm} ID_{il} OD_{jm} \upsilon_{ij}\right) \; ; \; \forall i, j \tag{16}$$

$$DJ_j + T_c \leq EJ_{j'} + N\left(1 - \sum_{m=1}^{M} q_{jj'm}\right) \; ; \; \forall j, j', j \neq j' \tag{17}$$

$$\sum_{i \in I} G_{iij} = 1 \; ; \; \forall j \tag{18}$$

$$\sum_{i' \in I} G_{i'ij} = \upsilon_{ij} \; ; \; \forall i, j \tag{19}$$

$$\sum_{\substack{i' \in I \\ i' \neq i}} G_{ii'j} \leq \upsilon_{ij} \; ; \; \forall i, j \tag{20}$$

$$DOJ_{ij} - \left(EJ_j + x_{ij}UD\right) \geq -N\left(2 - Y_{ij} - G_{iij}\right) ; \; \forall i, j \tag{21}$$

$$DOJ_{ij} - \left(DOJ_{i'j} + x_{ij}UD\right) \geq -N\left(2 - Y_{ij} - G_{iij} - G_{i'ij}\right) ; \; \forall i, i', j \tag{22}$$

$$DOJ_{ij} - \left(LT_{ij} + x_{ij}UD\right) \geq -N\left(2 - Y_{ij} - G_{iij}\right) ; \; \forall i, j \tag{23}$$

$$DOJ_{ij} - \left(DOJ_{i'j} + x_{ij}UD\right) \geq -N\left(Y_{ij} + G_{iij} + 1 - H_{ii'j}\right) ; \; \forall i, i', j \tag{24}$$

$$DOJ_{ij} - \left(LT_{ij} + x_{ij}UD\right) \geq -N\left(Y_{ij} + G_{iij} + H_{ii'j}\right) ; \; \forall i, j \tag{25}$$

$$DJ_j \geq DOJ_{ij} \; ; \; \forall i, j \tag{26}$$

$$C_{max} \geq DJ_j \; ; \; \forall j \tag{27}$$

$$x_{ij} \geq 0 \; \& \; Integer \; ; \quad \forall \, i, j \tag{28}$$

$$EI_i, DI_i, EJ_j, DJ_j, DOJ_{ij}, LT_{ij} \geq 0 \; ; \; \forall \, i, j \tag{29}$$

$$ID_{il}, OD_{jm}, \upsilon_{ij}, G_{i'ij}, Y_{ij}, H_{ii'j}, p_{ii'l}, q_{jj'm} \in \{0, 1\} \; ; \; \forall \, i, i', j, j', l, m \tag{30}$$

The objective is to minimize the makespan. Constraint (2) ensures that the total number of units of product that transfer from each of the inbound trucks to all outbound trucks must be exactly the same as the number of units of product which were initially loaded in each of the inbound trucks. Similarly constraint (3) ensures that the total number of units of product that transfer from all inbound trucks to each outbound truck is exactly the same as the number of units of product needed for each outbound truck. Constraint (4) enforces the relationship between variable x_{ij} and variableυ_{ij}. Constraints (5) and (6) define that each of the inbound trucks and outbound trucks must be assigned to an inbound door and outbound door respectively. Constraints (7)-(9) ensure that inbound trucks assigned to the same door must appear once in the sequence. Constraint (7) guarantees that only one inbound truck at each door is positioned at the beginning of the sequence. Constraint (8) describes that if an inbound truck is assigned to a door, it is preceded by an inbound truck. Constraint (9) expresses that if an inbound truck is assigned to a door, it can be succeeded, at most, by one inbound truck. The inbound truck that is in the last sequential position at a door will not have a succeeding truck. Similarly constraints (10)-(12) define that outbound trucks assigned to the same door must appear only once in the sequence. Constraint (13) ensures that the entering time of each inbound truck must be greater-equal than its releasing time. Constraint (14) expresses that the departure time of each inbound truck is equal to the starting time of its unloading plus the time needed for unloading the goods in it. Constraint (15) defines that the starting time of a succeeding inbound truck at a door is equal to or greater than the sum of the departure time of its preceding truck and the changeover time. Constraint (16) depicts the time which products transferred from an inbound truck to an outbound truck can be loaded. Constraint (17) defines that the starting time of a succeeding outbound truck at a door is equal to or greater than the sum of the departure time of its preceding truck and the changeover time. Constraints (18)-(20) ensure that the goods transferred from an inbound truck to its designated outbound truck must appear only once in the sequence of its loading. Constraints (21)-(25) compute the time that goods transferred from an inbound truck to an outbound truck are loaded, such that only one of these constraints will be enabled for each of the transferred products. Constraint (26) defines the departure time of each outbound truck, and constraint (27) expresses that C*max* must be equal to or greater than the departure times of outbound trucks. Constraints (28)-(30) specify the type of variables in the model.

In the abovementioned model, constraint (16) is nonlinear, and also, variables $y_{ij}, H_{ii'j}$ are dependent on other variables in the model. Therefore, to linearize constraint (16) and give a precise mathematical form of variables$y_{ij}, H_{ii'j}$, we need to add several constraints and change the model into a mixed integer programming model.

The complexity of this problem highly depends on the number of inbound and outbound trucks and the number of strip and stack doors. Moreover, in the studies of Lee, Kim and Joo (2012),it has been proved that this problem is NP-hard, because as the number of inbound and outbound trucks increases, the computational time grows exponentially. Therefore, a simulated annealing algorithm is proposed to solve the abovementioned problem.

3- Simulated Annealing

In this section, we present a Simulated Annealing algorithm in order to find the optimal solution for the aforementioned objective function-makespan, as the mathematical model will not be applicable for finding the optimal solution for the large-scale real-world cases. Simulated annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, to locate a good approximation of the global optimum of a given function in a large search space. It was independently invented by Kirkpatrick,Gelatt and Vecchi (1983).Since then, simulated annealing has been used in many other applications. Papers that have applied simulated annealing to cross-docking related problems include Jayaraman and Ross (2003), Ross and Jayaraman (2008), Chen et al. (2006), Vahdani and Zandieh (2010), Soltani and Sadjadi (2010), and Liao, Egbelu and Chang (2013).

The name and inspiration for this generic probabilistic meta-algorithm came from the technique of annealing in metallurgy, which involves heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy. Slow cooling gives the atoms more chances of finding configurations with lower internal energy than their initial states.SA is a neighborhood search technique that has produced good results for combinatorial problems. The major advantage of SA over other methods is its ability to avoid becoming trapped at local minima. Also SA is one of the metaheuristic algorithms which can definitely converge to the optimal solution while its required conditions are met. The algorithm employs a random search, which accepts not only changes that improve the objective function but also some changes that do not improve it. SA is a variation of hill climbing in which some non-improving moves may be made during the search process. The basic structure of SA algorithm is presented in Table 1 where the following notations are used:

S Current solution

S^* Best solution

$f(S)$ Value of objective function in solution S

n Repetition counter

T_0 Initial temperature

$SubIt$ Number of repetitions allowed at each temperature level

P Probability of accepting S_n when it is not better than S

Note: While performing a maximization problem, the objective function is multiplied by -1 to obtain a capable form.

Table 1: Basic structure of SA algorithm

```
Initialize the SA control parameter (T0,SubIt)
Select an initial solution, S0
Set T = T0, S = S0, S* = S0; calculate f(S0)
While the stop criterion is not reached do
Set n = 1 ;
While n < SubIt do
        Generate solution Sn in the neighborhood of S0;
        Calculate Δ = f(Sn) − f(S)
        If Δ≤ 0
                S = Sn; n = n + 1 ;
        else
                generate a random number, r ∈ (0,1)
                if (r ≤ p = exp (−Δ/T))
                        S = Sn ; n = n + 1 ;
                end
        end
        If ( f(S) < f(S*) )
                S* = Sn ;
        end
end
Reduce the temperature T;
end
```

The algorithm starts with an initial solution to the problem. In the inner cycle, the SA is repeated while $n < SubIt$, and a neighboring solution S_n of the current solution S is generated. If $\Delta \leq 0$, S_n is better than S, so the generated solution replaces the current solution; otherwise, the solution is accepted with a criterion probability ($\exp(-\frac{\Delta}{T})$). The value of temperature T decreases in each iteration of the outer cycle of the algorithm. Obviously, the probability of accepting the worst solution decreases as the temperature decreases in each outer cycle. The performance of SA depends on the definition of several control parameters:

a) The initial temperature, T_0, should be high enough so that, in the first iteration of the algorithm, the probability of accepting the worst solution is at least 80%.
b) The most commonly used temperature reducing function is geometric: $T_i = C.T_{i-1}$ where $C < 1$ and is constant. Typically, $0.7 \leq C \leq 0.95$.
c) The length of each temperature level, $SubIt$, determines the number of solutions generated at a certain temperature, T.
d) The stopping criterion defines when the system has reached a desired energy level, and is based on:

- The total number of solutions generated.
- The temperature at which the desired energy level is reached (the freezing temperature).
- The ratio between the number of solutions accepted and the number of solutions generated.

Obviously, each of these control parameters is chosen according to the specific problem on hand. In the proposed SA algorithm, the stopping criterion is the total number of solutions generated. The two most important factors in designing SA algorithm are the solution representation and neighborhood generating procedure, which highly influence the convergence speed of the SA.

A combination has been used for representing the solution, such that integer-valued lists are considered for assignment of inbound and outbound trucks to inbound and outbound doors and also for product assignments, where the length of each part is based on the size of the respective part. Permutation is proposed for showing the docking sequences of trucks and docking sequences of assigned shipments for each outbound truck. Figure 1 shows a feasible solution representation, in which there are6 inbound trucks and 4 outbound trucks, with 3 strips and stack doors.

1	2	2	3	1	1
5	2	6	3	4	1

1	2	3	3
2	4	1	3

1	1	2	2	2	3	4	4	5	6	6
2	3	2	2	1	2	3	2	1	1	1

Figure 1: An example of the solution representation

The first part of the solution representation, which is a matrix with two rows, demonstrates the docking assignment and docking sequence of inbound trucks. The first row of the matrix shows the assignment of inbound trucks to strip doors, the length of which is equal to the number of inbound trucks, and the values vary among the number of inbound doors. The second row of this matrix represents the docking sequence of inbound trucks, the length of which is equal to the number of inbound trucks, and the values are a permutation of the number of inbound trucks. In the same manner, the second part of the solution representation demonstrates the docking assignment and docking sequence of outbound trucks. Similarly, the first row of this matrix shows the assignment of outbound trucks to stack doors, but its length is equal tothe number of outbound trucks, and its values rang among the number of stack doors. Also, the second row of this matrix represents the docking sequence of outbound trucks, the length of which is equal to the number of outbound trucks, and the values are a permutation of the number of outbound trucks. The last part of the solution representation indicates the product assignment, which is comprised of two rows, the lengths of which are equal to the total number of products that are initially loaded into inbound trucks. In the first row of this part, the number of each inbound truck is repeated in an ascending manner based on the number of products it has, and the second row consists of the number of outbound trucks, which are set based on their demands but in a random manner. To clarify the third part of the solution representation, consider the third part of the solution shown in Figure 1. Suppose that the total number of products in inbound trucks is11 units, and the number of products that are initially loaded into inbound truck 1, 2, 3, 4, 5 and 6 are respectively 2, 3, 1, 2, 1 and 2. Also, the demands of outbound trucks 1, 2 and 3 are respectively 4, 5 and 2 units. The second row shows the product assignment, which is as follows: for outbound truck 1 with 4-unit demand, 1 unit of its demand is provided by inbound truck 2, another unit is provided by inbound truck 5, and 2 units of this demand is supplied by inbound truck 6. The demand of other outbound trucks can be explained in a similar way.

The most important point in this solution representation is: all the randomly generated solutions are feasible; thus, there is no need for considering penalty cost. As mentioned above, the solution representation is comprised of three different parts; hence, different neighborhood functions have been employed for each part of the solution. For making new sequences, a roulette wheel selection is applied to choose one of the three neighboring operators. Various operators have been considered for generating sequences: paired swap, inversion and insertion. In order to create a new docking assignment, one of the arrays is chosen randomly, and its value will then be substituted with a different number in the list. Also, for generating new product assignment, two positions are chosen randomly in the product list; then, the values of these two chosen positions will be changed only if these values are provided from different inbound trucks in the given list.

4- Numerical examples

In this section, two sets of problems are tested by the proposed SA algorithm and GAMS software CPLEX solver, in order to evaluate the efficiency of SA algorithm. Due to the complexity of this problem, which highly depends on the number of inbound and outbound trucks and also inbound and outbound doors, two problem groups are generated randomly based on these parameters. The first group of problems includes having less than 4 inbound and outbound trucks and less than 3 strip and stack doors. This group of problems is created to compare the solutions found by SA to the optimal solution that is obtained by GAMS software. For the first set of test problems, CPLEX solver was able to find the global optimum in less than two hours. For each of the SA parameters, different values are proposed by random, and one of the parameters will then be changed in its given domain while other parameters are fixed. This process is repeated 20 times for each of the changes. The best combination of SA parameters is then chosen.

The second group of problems includes the ones in which CPLEX cannot obtain the optimal solution in the given time (two hours); however, several instances are run with CPLEX to show its disability as compared to SA algorithm in regard to computational time and found solution. For this group of instances, adjusting SA parameters is done as well as for the first group. All the experiments are shown in Table5&6. All the needed data are chosen randomly. For example, the time needed for loading and unloading one product is assumed 5 time units, and the truck changeover time is 20 time units. The transferring time among different strip and stack doors is calculated through rectilinear distance. Also needed values for inbound trucks entering times are chosen by random and are shown in Table 2.The needed time windows are the same for all the tested problems. All the experiments utilize CPLEX and SA, and are executed on a PC with a 2.50 GHz Intel Core i7 CPU and 16GB RAM.

Table 2. proposed inbound trucks time windows

Inbound truck	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Time windows	40	10	0	70	35	89	120	200	90	250	110	20	0	75	205	220	300

Table 3. Final adjusted values for parameters of the first group of problems

Parameters	**Adjusted value**
Initial Temperature	100
Maximum number of main iterations	100
Maximum number of sub iterations	120
Cooling rate	0.93

Table 4. Final adjusted values for parameters of the second group of problems

Parameters	**Adjusted value**
Initial Temperature	2000
Maximum number of main iterations	400
Maximum number of sub iterations	300
Cooling rate	0.96

Table 5. Results of the problems in the first group

Number of I/O Truck	Number of I/O Dock	Number of shipment	CPLEX		SA algorithm				
			Time (sec.)	Cmax	Cmax				Average time (sec.)
					Run 1	Run 2	Run 3	Run 4	
2/2	2/2	45	1	290	290	290	290	290	5.5
3/3	3/2	55	611	285	285	285	285	285	6.5
3/3	3/1	50	776	380	380	380	380	380	9.6
3/3	1/2	47	9	319	319	319	319	319	8.5
4/4	1/2	70	1260	455	455	455	455	455	11.8
4/2	1/2	70	1	455	455	455	455	455	9

Table 6. Results of the problems in the second group

Number of I/O Truck	Number of I/O Dock	Number of shipment	SA algorithm					CPLEX
			Cmax				Average time (sec.)	
			Run 1	Run 2	Run 3	Run 4		
4/4	3/3	57	215	215	215	215	159	215
5/3	2/2	60	260	260	260	260	129	264
7/5	3/3	80	243	239	235	243	240	269
7/7	4/3	90	250	259	254	250	281	305
8/4	4/2	98	329	329	329	329	226	365
10/5	5/4	100	310	310	310	310	317	345
10/6	7/5	115	328	328	328	328	410	350
12/5	3/3	110	339	330	339	333	329	355
12/7	5/3	140	380	380	380	380	507	*
12/7	8/4	140	329	329	329	329	593	*
12/10	7/6	140	328	328	328	328	711	373
15/10	8/7	153	322	322	322	322	865	441
15/10	9/6	153	322	322	322	322	903	*
17/13	10/8	171	372	372	372	372	1242	*
17/13	5/3	171	439	439	439	439	955	*

The test results of the problems in Group 1 are reported in Table5. The structure of the table is as follows: the first three columns of the table show the problem information, the next column contains the results obtained by CPLEX, and the last column indicates the results acquired by SA algorithm. By comparing the obtained results of SA and the optimal solution which is found by CPLEX, we can claim that the proposed SA algorithm has good performance for small sized truck scheduling problems. Each of the reported instances was run four times with the proposed SA, and every time, the obtained result was the same as the optimal relative solution, which shows the efficiency of the proposed SA for small instances. Nevertheless, as the number of trucks increases, the computational time of CPLEX enhances significantly, while the proposed SA's computational time goes along a polynomial function. In Table 6, the results of the problems in Group 2 are summarized, which contains the instance parameters, the obtained makespan by SA algorithm, the average computational time of the proposed SA, and the objective function of the best solution obtained by CPLEX solver in

2 hours' time limit for each of the given problems. A star in the CPLEX results column means that the solver was not able to find a feasible solution in two hours. However, the computational times of the proposed SA for this group of problems are small enough to obtain solutions in a reasonable time frame.

5- Conclusion and future works

In this paper, a truck scheduling problem was studied which deals with scheduling of both inbound and outbound trucks at a cross-docking system with multiple dock doors. The objective of this study was to find the best docking assignment and docking sequence for inbound and outbound trucks in a way that the makespan is minimized. Therefore, for this aim, determining the door assignment and docking sequences for all inbound and outbound trucks had to be done simultaneously. Due to the interchangeability characteristic of products, product assignment had to be determined as well. In regard to the proposed objective function, a loading sequence was determined for each of the outbound trucks, which led to minimization of the makespan, and also helped to reduce the occupied storage space in the cross-dock, as a limited storage space is available in reality. In addition, considering the transferring time between different dock doors was another option that could help the objective function to be minimized. The problem was formulated as a mixed integer programming model in order to find the optimal solution. Although CPLEX could be used for solving the small sized problems, it got inefficient and impractical for solving large sized problems because of the increased computational time requirement. Therefore, a simulated annealing algorithm was proposed for solving the large sized problems to (near-) optimal. The obtained results indicated the effectiveness of the proposed SA, as the computational time is significantly small as compared to the running time of CPLEX solver.

Although various real-world details were taken into account, several others have not been considered, like limited storage capacity and internal congestion. Also, this problem was assumed to be static, whereas in practice, trucks arrive late, equipment fails, etc., so uncertainty and variability should be taken into account. Therefore, future research should incorporate these issues in the truck scheduling problem in order to increase the applicability.

References

Alpan, G., Larbi, R.,& Penz, B. (2011).A bounded dynamic programming approach to schedule operations in a crossdocking platform. *Computers & Industrial Engineering*, 60(3):385–96.

Apte, UM., Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics: Research and Applications,* 3(3):291–302.

Bartholdi, III. JJ., Gue, KR. (2004). The best shape for a crossdock. *Transportation Science*, 38(2):235–44.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., & Zandieh, M. (2011). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications*, 38, 1964–1979.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., Zandieh, M. (2010). Amulti-criteria cross-docking Scheduling with just-in-time approach. *The International Journal of Advanced Manufacturing Technology*, 49(5–8):741–56.

Boloori Arabani, A. R., Zandieh, M., & Fatemi Ghomi, S. M. T. (2011). Mulit-objective genetic-based algorithms for a cross-docking scheduling problem. *Applied Soft Computing*, 11, 1954–4970.

Boloori Arabani, A. R., Zandieh, M., & Fatemi Ghomi, S. M. T. (2012). A cross-docking scheduling problem with sub-population multi-objective algorithms. *International Journal of Advanced Manufacturing Technology*, 58, 741–761.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1):32–41.

Boysen, N., Briskorn, D., & Tschöke, M. (2013). Truck scheduling in cross-docking terminals with foxed outbound departure. *OR Spectrum*, 35, 479–504.

Boysen, N., Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega*, 38, 413–422.

Boysen, N., Fliedner, M., Scholl, A. (2010).Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*,32(1):135–61.

Chen, P., Guo, Y., Lim, A., Rodrigues, B.(2006). Multiple crossdocks with inventory and time windows. *Computers & Operations Research*,33, 43–63.

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

Joo, C. M., & Kim, B. S. (2013). Scheduling compound trucks in multi-door crossdocking terminals. *International Journal of Advanced Manufacturing Technology*, 64, 977–988.

Kirpatrick, S., Gelatt Jr., C.D., Vecchi, C.P. (1983). Optimization by simulated annealing. *Science,* 220 (4598), 671–680.

Konur, D., Golias, M. M. (2013). Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. *Computers & Industrial Engineering*, 65, 663–672.

Kuo, Y. (2013). Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Systems with Applications*, 40, 5532–5541.

Lee, K., Kim, B. S., & Joo, C. M. (2012). Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance. *Expert Systems with Applications*, 39, 12975–12983.

Liao, T. W., Egbelu, P. J., & Chang, P. C. (2013). Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multidoor cross docking operations. *International Journal of Production Economics*, 141, 212–229.

Maknoon, M.Y., Baptiste, P. (2009). Cross-docking: increasing platform efficiency by sequencing incoming and outgoing semi-trailers. *International Journal of Logistics: research and Applications*, 12, 249–261.

McWilliams, D. L. (2009). A dynamic load-balancing scheme for the parcel hub scheduling problem. *Computers & Industrial Engineering*, 57, 958–962.

McWilliams, D. L. (2009). Genetic-based scheduling to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 56, 1607–1616.

McWilliams, D. L. (2010). Iterative improvement to solve the parcel hub scheduling problem. *Computers & Industrial Engineering*, 59, 136–144.

McWilliams, D. L., Stanfield, P. M., & Geiger, C. D. (2005). The parcel hub scheduling problem: A simulation-based solution approach. *Computers & Industrial Engineering*, 49, 393–412.

McWilliams, D. L., Stanfield, P. M., & Geiger, C. D. (2008). Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers. *Computers & Industrial Engineering*, 54,709–720.

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, 64–79.

Sadykov, R. (2012). Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost. *Annals of Operations Research*, 201, 423–440.

Soltani, R., Sadjadi, S. J.(2010). Scheduling trucks in cross docking systems: a robust meta-heuristics approach. *Transportation Research Part E,* 46, 650–666.

Vahdani, B., Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust metaheuristics. *Computers & Industrial Engineering*, 58, 12–24.

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

Yu, W., Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, 184, 377–396.