Dynamic vs. Static Optimization of Crossdocking Operations[1]

Monique Guignard[a, 2], Peter Hahn[b] and Heng Zhang[b]

[a] University of Pennsylvania, the Wharton School, OPIM Dept, 3730 Walnut St.., Philadelphia, PA 19104, U.S.A.
[b] University of Pennsylvania, Eng. School, ESE Department, 200 S 33rd St. , Philadelphia, PA 19104, U.S.A

Abstract To improve operations commonly found in today's crossdocks, we offer a door assignment optimization tool that will reduce the distance travelled by goods across the crossdock, as well as workload and labor cost. The cross dock door assignment problem (CDAP) minimizes total distance travelled by the goods inside the crossdock where door capacities are limited by the time they can be used in a work day. Experience shows that blindly following CDAP optimization results could negatively impact delivering goods efficiently. We therefore propose a dynamic assignment scheme in which door assignment is reconsidered whenever a new trailer arrives. Repeatedly updating and resolving for door assignments within a simulation experiment is made feasible through the use of our advanced MatHeuristic solver, CHH, and this permits us to compare the static and the dynamic assignments. Using simulation data based on data from an actual crossdock, we demonstrate the superiority of the dynamic approach in terms of operations efficiency and make managerial recommendations.

Keywords crossdock; dynamic optimization; simulation; matheuristic; door assignment; turnaround

Introduction

Crossdocking is a practice in logistics that aims at reducing transportation and inventory holding costs in the supply chain. Crossdocking involves receiving goods from suppliers or manufacturers for several destinations and consolidating these shipments with those of other suppliers for common delivery destinations, then loading them for shipment to their respective destinations at the earliest opportunity. In crossdocking, goods from an incoming trailer are unloaded, sorted and then loaded to outbound trailers, with little or no storage in between.

---

[1] submitted to a special issue of Transportation Research part C: Emerging Technologies, entitled "Advances in transportation and logistics" (guest editors Cristián E. Cortés, Doris Sáez and Pablo Rey)

[2] Corresponding author at: University of Pennsylvania, the Wharton School, OPIM Dept, 3730 Walnut St., Philadelphia, PA 19104, U.S.A. Tel.: +1 215-567-3188.
Email address: guignard_monique@yahoo.fr

Besides reducing transportation costs, crossdocking also focuses on reducing the level of inventory held in the supply chain, and therefore inventory holding costs, and order cycle time.

This paper concentrates on the operations that take place inside a crossdock. It assumes that the decisions about trailer contents, as well as their movements from the origins to the crossdock, and the goods' final destinations, are already made. Inbound doors, also called strip doors, are the dock-doors of the building, where incoming full trailers are parked and unloaded. Outbound doors, also called stack doors, are those where outgoing empty trailers are sent to collect freight according to specific destinations. Often, strip doors are found on one side of a crossdock and stack doors on another side. In practice, however, incoming trailers are sometimes unloaded on the stack door side of the crossdock and outgoing trailers loaded on the strip door side. Consolidation (or staging) areas are locations within the crossdock, where goods wait to be loaded into outbound trailers. Figure 1 depicts a typical crossdock configuration and operations.

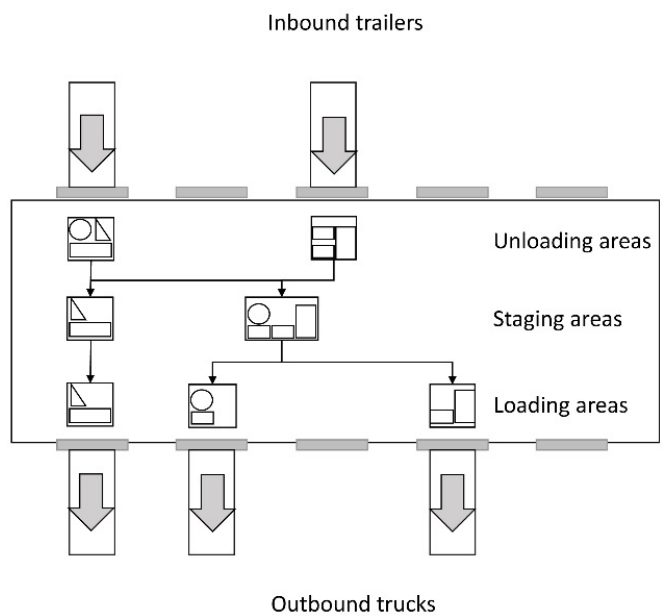


Figure 1. Depiction of Cross-Docking Operations

In the Operations Research and transportation literature, optimization of operations in the crossdock has been either static or dynamic. In static optimization, incoming trailer contents, including destination volumes for the current period, are known and the door assignments are optimized, usually to minimize travel of the goods through the crossdock. Optimized door assignment, while beneficial, does not take into account the realities of the dynamic situation actually encountered during the processing period. In the crossdock research that addresses the changes occurring outside and inside the crossdock over the planning period, usually no more than a day, arrival and/or departures of trailers are scheduled so as to optimize efficiency. Optimization criteria differ (e.g., cost of labor, cost of delays, total busy time, etc.); as well as assumptions (e.g., all incoming trailers are available at the beginning of the work period, or they arrive at random times during that period -- outgoing trailers have a known schedule, or can be called upon when needed). Most optimization approaches rely at least in part on heuristics and/or simulations. We refer the reader to recent surveys [1], [2], [17] and references therein for a more complete picture of the many considerations for crossdock optimization.

Our initial crossdock research focused on minimizing the cost of goods transfer in the crossdock by optimizing the assignment of strip doors to incoming trailers and stack doors to outgoing trailers. This static optimization problem is known as the Cross Dock Door Assignment Problem (CDAP), described in Sections 2, 3 and 4. Different from other CDAP optimization schemes, we imposed a time-related capacity constraint on each door, based on the maximum time to unload or load a trailer. Two exact solvers that we developed have been applied to achieve the optimal solutions of CDAP, solving respectively a Generalized Quadratic 3-dimensional Assignment Problem (GQ3AP), described in [7] and based on [4a], and a Generalized Quadratic Assignment Problem (GQAP) of approximately double size, described in [5], and based on [4]. Beyond a certain size, however, exact solvers, including commercial solvers, are unable to provide optimal solutions in an elapsed time small enough to allow frequent calls to the solver. Thus, we adapted our Convex Hull MatHeuristic (CHH) CDAP solver, based on convex hull relaxation [6] and on simplicial decomposition [6b], to solve the CDAP and provide an approximate solution in much shorter times ([7] and [8]). A brief description of the CHH CDAP solver is provided in [6] and [8], as well as in section 5.

Together with several teams of students from the University of Pennsylvania, the authors have worked closely for several years with two firms managing crossdocks in the Eastern US. The first of these firms handles goods coming from US manufacturers and from nearby seaports and destined for a number of retail outlets. This firm's crossdock is of medium size. The second firm manages a larger crossdock, dedicated to goods from a single supplier, which are then shipped to several distribution centers. Our teams of students analyzed and simulated operations in these crossdocks and were successful in finding operational improvements for both firms. It became clear, in the first few months studying actual crossdock operations for firm one, that their complexity would make a static optimization scheme of little or no use. Optimizing the distance travelled by goods in the crossdock needs to be dynamic, as its solution depends not just on incoming trailer contents, but also on the order in which incoming trailers arrive or are handled. We are facing a large optimization problem with easy constraints related to assigning the doors in a static manner taking door capacities into account, and difficult constraints that have to do with assigning the loading/unloading work over time, in an efficient, timely manner. As explained in section 3, as the work period progresses, the static solutions may become impractical because of the difficult constraints that need to be satisfied in real operations, such as not assigning two trailers to the same door at the same time. The following summarizes our observations that govern the development in this paper.

First of all, the major objective is often to optimize the use of the workforce, and its cost. According to [9], travel inside the crossdock is typically 20-30% of total dock labor costs, which is not insignificant, and it has actually received a lot of attention in one form or the other in the crossdock literature. Instead of directly working on the labor cost, we concentrate on finding the door assignment with the least travel cost of materials inside the crossdock. At the beginning of the day, with the information available on the day's workload, we produce the assignments of all the inbound and outbound doors that minimize total distance by solving the CDAP model. We refer to this scheme as the 'static assignment strategy'.

Secondly, in view of the difficult constraints mentioned above, which are not addressed by the CDAP model or the static assignment strategy, we dynamically update the information and optimize the cross-docking assignments accordingly. To guarantee that we meet the time requirements is our primary concern. Even if minimizing the cost of carrying the goods across the crossdock is not the main objective, it is desirable to assign doors so that the total distance travelled by goods in the crossdock is kept to a reasonable minimum. We are trying to keep a balance between meeting the time requirement and keeping the total goods travel distance reasonable, by minimizing the latter and imposing a limit on how much each door can handle within a shift, this is what we call door capacity. This capacity will naturally decrease during the day.

Our scheme is therefore to apply CDAP optimization repeatedly throughout the work period, taking into account the state of occupancy of all crossdock doors at each new trailer arrival. This is what we refer to as the 'dynamic assignment strategy'. See more on this in particular in sections 3, 4, 6 and 7. In our dynamic approach, the secondary objective of minimizing travel distances is reconsidered repeatedly, with updated data each time. For assigning the next trailer(s) to available doors, we use the heuristic door-trailer assignment (described in section 5) as a guide in our simulation-based experiment. This is explained in detail in section 6.

In summary, on the one hand, we optimize the distance travelled by products within the cross-docking facility. On the other hand, we dynamically reformulate and re-solve the distance minimization problem as needed in real-time. By integrating the CDAP model inside a dynamic scheduling of the crossdock, we overcome the CDAP's static limitations, guaranteeing timely operations without having to know ahead of time when trailers will arrive.

The detailed plan of the paper is as follows:

In section 1, we briefly survey the large body of cross-docking research and identify the literature relevant to our study.

In section 2, we mainly identify the assumptions we will make in our research, define the cross-docking assignment problem, and briefly introduce the CDAP model.

In section 3, we focus on the inadequacy of the CDAP model for real practice and the necessity of a dynamic assignment scheme.

In section 4, we formally define the CDAP model in our context, with the focus on linking its parameters to the static and dynamic assignment strategy to give the reader a clear view of how the CDAP model is utilized in the course of a work day.

In section 5, we consider solving the CDAP model by CHH. Due to its efficiency, we employ this algorithm in our simulation experiments, and recommend it for real-world implementation of a dynamic assignment strategy.

In section 6, we move on to the details of our dynamic assignment strategy, through the introduction of the simulation procedure that mimics the real operation in practice.

In section 7, detailed results the dynamic and the static strategies are presented. Several aspects are considered such as: To what extent do the dynamic assignments sacrifice the distance travelled by goods in order to finish the work in a timely fashion? How many products are left and not shipped in both schemes at the end of the day? How many inbound trailers have not been handled? What are their respective impacts on workers?

1 Literature Survey and Different types of Crossdock Operations

There are many different types of crossdocks and many different management practices. A good practical introduction can be found in Gue's blog [9], and in his 2007 International Commerce Review paper [10]. Part of the following description is taken from there.

The buildings used for crossdocking may have different shapes, in particular rectangular, L-shaped, U-shaped, and T-shaped. Many crossdocks have been transformed into crossdocks from their earlier purposes, and their shapes may be less than optimal. Crossdock sizes vary from a few dozen doors to hundreds. The largest crossdock, of which we are aware, is situated near Dallas and has about 550 doors. Crossdock managers handle differently the question of how to assign outbound trailers to doors and where to put inbound trailers when they arrive. In [10], Gue carefully compares the advantages and disadvantages of crossdocking vs. more traditional forms of warehousing and highlights issues that may predict whether a crossdocking company will fail or succeed. It is clear that careful attention to both costs and to operational strategies is needed for a crossdock operation to succeed.

The crossdock optimization literature initially concentrated on door assignment, as in the early paper by Tsui and Cheng [11]. More recently, attempts at integrating the loading and unloading operations and the transportation of the goods through the crossdock have yielded a variety of exact combinatorial models, which unfortunately cannot be solved optimally even for problems of average size. Heuristic approaches have also been published that will more likely be able to solve approximately problems of average and above average size. Recent survey papers, or papers with an extended literature review, show that interest in crossdocking in the academic literature has been increasing substantially over the years, see [12], [13], [14], [15] for instance, producing a variety of approaches. Yet differences in crossdock layout and operating environment seem to preclude a general solve-everything crossdock algorithm.

As mentioned earlier, we have been involved with two crossdocking firms that operate their crossdocks very differently. In the first firm, trailers with known contents arrive at the facility at irregular, mostly unpredictable, times during the day. For this firm, trying to optimize the crossdock operations as a machine-scheduling problem is not feasible, as trailer arrival times are not known ahead of time. However, at the start of the day, arriving trailer manifests are available for trailers expected to arrive that day. For the second crossdock, trailers arrive in the yard prior to the start of the planning period (typically a day) and thus are available for unloading whenever appropriate, which is most often the assumption made in the literature. Furthermore, trailer content is known precisely ahead of trailer arrival. As these two crossdocking instances show, scheduling trailer unloading, moving goods across the crossdock and then loading outgoing trailers would best be optimized differently for different types of crossdock approaches. We concentrate in this paper on ways to optimize crossdocks of the type of the first firm, where trailers arrive at unknown or only approximately known times during the day and trailer content is known at the start of the working day, but no earlier. A more detailed description is given in the following sections.

Further, our paper draws a lot of emphasis on the simulation in the crossdocking operations in order to compare the static and dynamic assignment strategies. There are some recent studies dealing with the cross-docking simulation problem, for example [3], [27], and [28]. These studies, however, are different from ours in both the purpose and the use of the simulation. In [27] and [28], the authors work from the perspective of the supply chain in the outside of the crossdock, and looks at the supply network which consists of the suppliers, crossdocks and the destinations. The work aims at the study of transportation between a set of suppliers and a set of cross-dock facilities to account for the timely delivery of the randomly generated orders from the suppliers. The relevant metrics in the simulation are the average order cycle, truck utilization and service level in their researches. The operations inside the crossdock as a whole, with which we are directly concerned, are not treated in detail.

Similar to this simulation study, Konur and Golias are concerned with the scheduling and assignments of inbound trailers. They use the total delay time of inbound trailers as the objective function to construct a bi-objective bi-level optimization problem, and solve through Genetic Algorithm. Simulation is used mainly as the way of generating arrival times of inbound trailers to compare First Come First Serve principle and their model. The labor work inside the crossdock, such as loading and unloading, is not directly simulated and accounted for but instead is treated as random numbers. To the best of our knowledge, we are giving the most detailed simulation study of the operations inside the crossdock.

2. Assumptions: The Cross-dock Door Assignment Problem under Consideration

We consider crossdocks where a few trailers may already be in the yard at the beginning of the day, and others arrive at irregular intervals during the day. Doors are not a priori assigned to origins or destinations, and the destination of arriving goods is however known at the start of the day. We specify the assumptions we make in the same style as in [16].

A1: The arriving goods may be packed in boxes of varying size, and we use volume as the standard unit, thus in fact ignoring, as a first approximation, the problem of fitting boxes in outgoing trailers. We hypothesize that all unloading work is done in an identical unit of time, per unit of volume. Similarly, goods that are held in the staging area will take a number of time units proportional to their volume to load into an outbound trailer.

Note that with this assumption we will be able to properly define the capacity of each inbound or outbound door. Obviously, given the time required for fully unloading or loading a truck, as well as the time constraints that must be imposed on each working day, one can schedule only a limited number of trailers at a door. Following this we introduce capacity in the problem to represent this time limitation. More discussion of this issue will be given in sections 3, 4, 6 and 7.

A2: Our prior information on the arrival times of inbound trailers is limited in the sense that we only know in advance which trailers will be arriving that day, but not their actual arriving times. The contents of each trailer are known in advance, but individual item positions in the trailer are not known.

A3: The unloading of the trailers is done using a FIFO policy, i.e., the first inbound trailer arriving in the crossdock yard will be considered for assignment to one of the free inbound doors, if any. Otherwise it will wait in the yard.

A4: The workforce at the crossdock is sized such that it can unload all the inbound trailers and load them in the same work shift even if all the doors are occupied. At this point, we do not try to directly minimize the cost of labor, even though this is of great concern to management. This will be part of our ongoing research. Indirectly, we are trying to minimize the total time the force must remain at work to process the goods arriving in the work period through the timely assignments of trailers. As common in other research of this kind, the workers handling goods in the cross-docking operation are assumed to be homogeneous in the sense that all workers handle the same work with the same productivity.

A5: Goods are loaded into outbound trailers only after having been placed in one of possibly many staging areas. The use of staging (consolidation) areas is a common practice in cross-docking operations. The specific position of staging areas can vary in practice. They may be in the middle of the cross-dock or closer to the stack door side. They may be divided into slots corresponding to the destinations but not always. We do not make specific assumptions on their positions organization. Instead, we only assume two things that should be easily satisfied in practice. First, the amount of materials accumulated in the staging areas is known to us. Second, the door-to-door distances we use already assume that the goods make a stop-over at a staging area.

About the staging areas, we further assume that once the materials accumulated *for one destination* exceed a certain threshold, which we assume to be 90% of a Full Truck load (FTL), we consider the accumulation adequate and one outbound trailer assignment should be needed for this particular destination.

A6: The outbound trailer fleet is well supplied with interchangeable standard trailers, so that no time is lost waiting for the arrival of an outbound trailer.

A7. Note that most of the time, the load will consist of goods coming from several incoming trailers, and one has to wait until the last of these trailers is fully unloaded, as the loading patterns inside incoming trailers are usually not known in advance. Thus for simplicity, the staging areas will not be updated unless an inbound trailer has just finished unloading.

With these assumptions, we have essentially ignored the problem of minimizing labor cost and focused on the distance optimization and on achieving timely operations. We now formally define the problem under consideration. Facing the operation workload of a day, we need to decide, in advance or in the middle of the operation, how these incoming trailers are assigned when they arrive, and how the outbound trailers should be assigned, when the amount of products accumulated at the staging area reaches certain level and requires such an assignment. But we also want to minimize to total distance travelled by materials. If no constraint on the capacity requirement is imposed for doors, we may easily arrange all trailers at one pair of closest strip and stack doors. But this is evidently impossible, since we have very limited operation time in a day. We thus want to optimize the distance with this constraint to produce the assignments, although we may use different rules in terms of how the problem is defined and how the solution will be used, in the static or dynamic assignment strategy.

In terms of distance optimization, we shall consider the CDAP problem. The crossdock door assignment problem (CDAP) will be modelled using the following parameters and data: the number of origins M, the number of destinations N, the number of strip doors I, the number of stack doors J, the number of trips required by the material handling equipment to move items originating from a given origin to the crossdock door where freight destined for a given destination is being loaded, the distance between a given strip door and a given stack door, including the stopover in a staging area, the volume of goods from a given origin to a given destination, the capacity of each strip and stack door and the demand from each destination. The amounts of goods coming from one origin and tagged for one destination are stored in a matrix that we will call the flow matrix in the rest of the paper. The above parameters hold for a period of time (usually a day) during which the originating goods are processed by the crossdock and are loaded onto trailers leaving for destinations. The objective of the CDAP is to minimize the total distance travelled by all goods across the floor of the crossdock for the entire processing period, subject to assignment and capacity constraints. It is a pure 0-1 bilinear generalized assignment-type model.

In the next sections, we first describe in detail the CDAP model, and how we use it to assign trailers to doors. We will then discuss the weakness of making these decisions only once at the beginning of the working day. To deal with this weakness, we then propose a dynamic algorithm that performs door reassignment and crossdock scheduling through the day. To illustrate the concept and processes of the dynamic assignment, we first study a small artificial instance with three incoming trailers in the work period, four destinations, and a crossdock with four doors on both sides.

Then we will look at a realistic simulation problem to compare the dynamic and static schemes, where the trailer loads are taken from volumes processed by the first company mentioned above on a typical busy day. For ease of exposition hereafter, we will assume that trailers start arriving at 7 am, and a standard ten-hour shift starts then and lasts until 5 pm. We also consider the possibility that some trailers may already be stationed in the yard at the beginning of the planning period, and that some goods from the previous day may have been left in the staging areas, not making up sufficient volumes to justify sending out trailers the previous evening.

In our dynamic algorithm, described in section 6, the CDAP problem is re-solved and its data updated, after each new trailer arrival, while in the static model one waits until the infeasibility resolves itself.

3. What is missing in the CDAP model and how we address some of it

The dynamic aspect of assigning trailers to available doors is not addressed by the CDAP, as this optimization model considers only the total capacity of each door, incoming or outgoing, over a given period of time. If we assume that one CDAP model is solved for the entire work period before its start, strictly implementing its solution will usually not meet the timeliness requirement in the cross-docking operations. As a consequence, precious labor resources may be wasted. In addition, as we have learned from practitioners, who are forced by contract to deliver the goods within a short period of time, usually 24 hours, at any rate not more than 48 hours, missing a deadline implies that fines must be paid, and customer satisfaction decreases. Similarly, the time inbound trailers stay in the yard is also not acceptable, and usually results in fines.

Let us consider a simple illustrative example here, as more will be discussed in the experiments in section 6 and 7. At 7 am, we use the information about the contents of all trailers due to arrive that day to solve the static CDAP problem (we refer to the solution as the "planned assignments"). We would like to make actual assignments that adhere strictly to the planned assignments for all trailers that have already arrived before 7 am. This, however, may or may not be possible if we want the work to be completed in a timely manner. If two trailers are planned to be assigned to the same door then we have to either make one trailer-to-door assignment and queue the second trailer, or slightly adjust the assignment such that one of the two conflicting trailers is assigned to a neighboring door. The first choice may be better in the long run, but this fact is not known, as it depends on unpredictable arrivals later in the work period. Crossdock managers would likely make the latter choice, as they care about timeliness. The distance saved, which could be translated into time saved, and therefore labor cost saved, could hardly justify the wait.

A more important concern is what happens after the first set of door assignments are made at 7 am. As stated in section 2, crossdock managers know the arrival times only in an ad-hoc manner, i.e., after the trailers have actually arrived. Let us continue with the previous example. Suppose that at 8 am, more trailers come in, and some of their pre-assigned inbound doors are already occupied, then one could either wait until the planned assignments become feasible, or run a new CDAP optimization with modified flow and distance matrices. In this case, since the doors currently occupied cannot immediately be reassigned, to ensure timeliness, we may form a new distance matrix without these conflict doors and form a new flow matrix with all previously assigned trailers removed. The idea is that when we update the information in real time we could reformulate and resolve the distance minimization problem accordingly, so that de facto we are anticipating and optimizing the future operations without any assumption about the arrival distributions.

What happens on the outbound (stack) door side is similar, thus it is necessary to dynamically change the optimal assignment, or, and that is what we will do, dynamically optimize the assignment rather than fixing them based on the solutions from the CDAP problem. Also, in this part, we do as the first firm we studied does, that is, wait until a full load for a given destination has been accumulated in the consolidation (staging) area before assigning a door to a destination.

Previous algorithmic research papers (for instance [14], [15], [16], [17], [18]) look either at the static door assignment alone, or at the global crossdock problem. In this paper, we take a dynamic approach, re-assessing crossdock door assignments over time, i.e., in practice, over an entire work period, utilizing a CDAP solver. We override with reasonable logic those door assignments that cannot be implemented because (for instance) a CDAP assigned door is occupied, thus avoiding queuing delay. Anticipatory trailer-to-door assignments are re-computed using a CDAP solver each time a new trailer is moved to a strip door. Accordingly, the flow matrix is modified to remove all goods already assigned to strip doors and all goods no longer staged for outgoing delivery. So is the distance matrix due to the currently unavailable doors. The CDAP door capacities are modified to remove door capacity no longer available.

In section 4, we will first present in detail the CDAP model and in section 5, we describe the way the convex hull heuristic (CHH) solves the CDAP problem and assigns trailers to doors. Then in section 6, we show when we dynamically decide to run CDAP using CHH and how we adjust its solution in case of conflicts with the current crossdock configuration.

4. The static CDAP model

We now present the mathematical model of Zhu et al. for the Crossdock Door Assignment Problem (CDAP) [19]. This is a modification of the original model of Tsui and Chang [11].

Parameters:

M number of origins, N number of destinations, I number of strip doors, J number of stack doors

w_{mn} number of trips required by the material handling equipment to move items originating from m to the crossdock door where freight destined for n is being consolidated,

d_{ij} distance between strip door i and stack door j ,

s_m volume of goods from origin m.

S_i capacity of strip door i,

r_n volume of goods destined for destination n,

R_j capacity of stack door j.

Decision Variables:

x_{mi}=1 if origin m is assigned to strip door i, 0 otherwise

y_{nj}=1 if destination n is assigned to stack door j, 0 otherwise.

The new and generalized formulation of the CDAP is:

$$\text{Minimize} \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{n=1}^{N} d_{ij} w_{mn} x_{mi} y_{nj} \tag{4.1}$$

subject to:

$$\sum_{m=1}^{M} s_m x_{mi} \le S_i \qquad i = 1, 2, \ldots, I\,, \tag{4.2}$$

$$\sum_{i=1}^{I} x_{mi} = 1 \qquad m = 1, 2, \ldots, M\,, \tag{4.3}$$

$$\sum_{n=1}^{N} r_n y_{nj} \le R_j \qquad j = 1, 2, \ldots, J\,, \tag{4.4}$$

$$\sum_{j=1}^{J} y_{nj} = 1 \qquad n = 1, 2, \ldots, N\,, \tag{4.5}$$

$$x_{mi} = 0 \text{ or } 1 \qquad m = 1, 2, \ldots, M\,, i = 1, 2, \ldots, I\,,$$

$$y_{nj} = 0 \text{ or } 1 \qquad n = 1, 2, \ldots, N\,, j = 1, 2, \ldots, J\,.$$

(4.1) makes sure that the capacity S_i of strip door i is not exceeded, (4.2) makes sure that each arriving trailer gets assigned to only one receiving (strip) door, (4.3) makes sure that the capacity R_j of outbound (stack) door j is not exceeded, and (4.4) makes sure that each destination is assigned only one stack door. In the original model proposed in [19], in the left hand of (4.2) and (4.4), the parameters such as s_m and r_n are taken as row and column sums of matrix (w_{mn}).

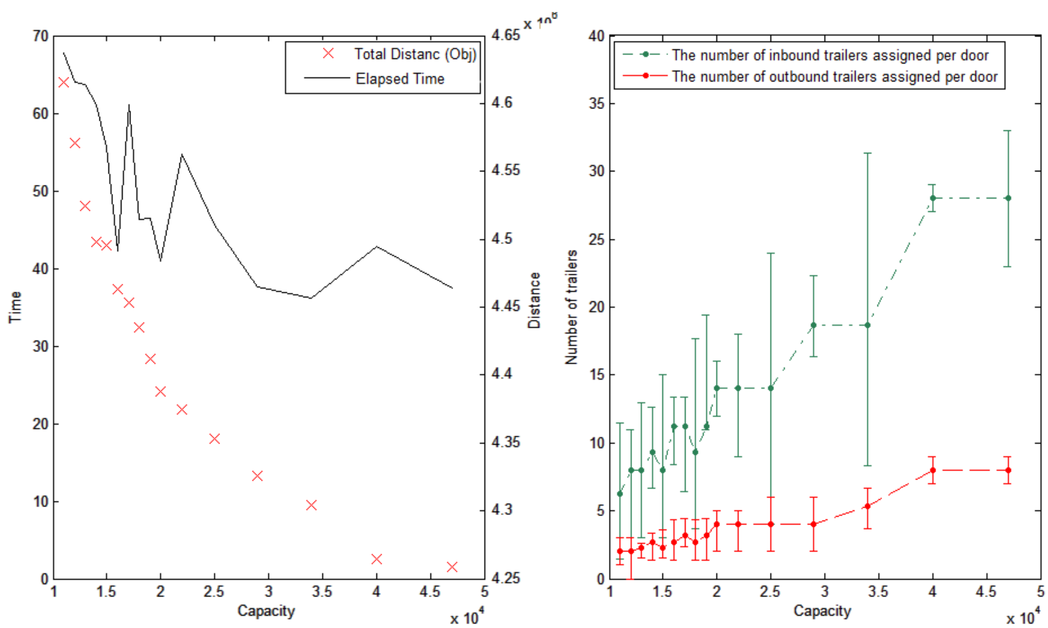


Figure 2. Measures of CHH run versus door capacity

With this model we are able to assign multiple origins to a single strip door, as long as its capacity could accommodate them. It is also possible to assign multiple destinations to one stack door, as long as its capacity could accommodate them.

Note that when we are considering the real-world operations, the door capacities should be determined based on an estimate of total working time, and optimal trailer schedules will depend on the numbers entered. If the number of hours chosen is too large, at the limit only one inbound and one outbound door may be needed, but this is obviously not desirable.

If the capacities are too small, it may not be possible to take care of all shipments within the time allotted. Figure 2 summarizes the results of solving a CDAP instance with 56 incoming trailers, 16 destinations, and 25 inbound and 16 outbound doors. It assumes that all doors have the same capacity, which varies between 10,000 and 47,000 ft^3, the latter corresponding roughly to 24 hours of loading or unloading time. We made the simplifying assumption that one can load or unload 2,000ft^3 in an hour. As can be seen, the distance travelled and CHH running time decreases as capacity increases. On the second picture, one sees the maximum and minimum number of trailers, as well as the average number, per inbound and outbound doors. Clearly, the values taken for the capacities affect the solution of the CADP model. Care should therefore be taken in determining these quantities.

Further, in the implementation of this model, we can determine the quantities such as s_m and r_n in a flexible way. For example, once we are able to estimate the time needed to unload inbound trailer m, we can use it as s_m. This would allow crossdock managers the flexibility of using advanced statistical methods in the estimation, but this is beyond the scope of this paper.

In practice, since we are required to finish all the work within one shift in a day, we take the capacity as the total time of a shift, i.e. 10 hours. Also, due to assumption A1, we set the time needed for inbound trailer m as s'_m proportional to s_m and r'_n proportional to r_n, and use them instead in (4.2) and (4.4). More specifically, we have access to operational data from a cross-docking facility, for instance the real flow matrix in the units of volume, and we learned from the first firm that we studied that, on the average, unloading a trailer takes two hours and loading a trailer takes one and a half hours[3]. Thus, based on the assumptions above, we determine s'_m and r'_n as $s'_m = (2M/\sum_{m=1}^{M} s_m)s_m$ and $r'_n = (2N/\sum_{n=1}^{N} r_n)r_n$.

It is important to point out here that the objective used in the CDAP model could be modified, as the situation requires. Indeed, instead of using total distance travelled by the goods, one could use the total cost of transferring the goods across the crossdock, or any other bilinear objective function. If the objective function were linear, the optimization would be even further simplified. In all cases, its optimization would be called repeatedly by the dynamic system described in section 6.

[3]The reason that unloading takes longer than loading is due to the time it takes to identify destinations of individual goods and assuring that the incoming manifest data is correct for the arriving goods.

5. Solving the static CDAP with the Convex Hull Heuristic (CHH)

As the objective function of the CDAP is a bilinear function, most likely nonconvex, it can be convexified, for instance by SDP-based methods, called generically QCR or quadratic convex reformulation. See for instance [23] for some of the latest results as well as references to earlier work. Another option is to linearize the CDAP model (see for instance [24] and references therein). Unfortunately neither of these approaches can be used to solve within a few minutes a CDAP of practical size, even approximately. We have therefore chosen to adapt a Matheuristic called CHH (short for Convex Hull Heuristic) to solve the CDAP repeatedly within our dynamic approach. CHH is an adaptation to the nonconvex objective function case of the convex hull relaxation, or CHR.

CHR is based on a concept that was independently introduced in [20] and [21] (see [6] for a description of both QCR and CHR). It is an extension of the primal relaxation of [22], which computes both a lower bound on the optimal value of a nonlinear convex integer minimization program, and good integer feasible solutions. Suppose the vector of integer decision variables is y, and the objective function value at y is $f(y)$. A by-product of CHR is that it produces a number of integer feasible solutions, $y(1), ..., y(p)$, and we can compute $f(y(1)), f(y(2)),..., f(y(p))$ and keep the best point, call it $y^{(*)}$, as a best feasible solution. Whether the problem is convex or not, CHR produces integer feasible solutions, and we call this use of it the convex hull heuristic, or CHH for short. In the convex case, CHR produces both a valid bound on the optimum and a best found feasible integer solution $y(*)$. In the nonconvex case, it produces only one best feasible integer solution.

CHR is generic, in the sense that it can be applied to any MINLP problem with linear constraints, as long as linear problems subject to the same constraints are, by comparison, much easier to solve. The algorithm is loosely based on simplicial decomposition, or SD for short, see [6b]. It alternates between a continuous nonlinear problem with one linear constraint, whose number of variables increases by at least one at each iteration, and one linear MIP problem subject to all original constraints. The nonlinear problem (the restricted master problem, MP) optimizes the original objective function over the convex hull of all (mixed-)integer feasible solutions currently kept, and the variables of that problem are the weights associated with each one of these points. The linear MIP problem (the subproblem, SP) optimizes the objective function linearized at the point found by MP over the original constraint set. To speed up the algorithm, SP can make use of advanced software features such as CPLEX's solution pool, for enlarging – and enhancing the quality of – the sample of integer feasible solutions found. Indeed inferior incumbents for the linearized objective function problem may actually be superior for the original nonlinear objective function. In practice, the best solution found in most instances comes from a solution pool. CHR converges to a valid lower bound on the integer optimum for pseudoconvex objective functions.

CHH, as it stands now, is a multi-start matheuristic, designed specifically for (possibly mixed-) 0-1 nonlinear nonconvex optimization problems with linear constraints. In the nonconvex objective function case, CHH uses a pattern similar to that of CHR, alternating between a restricted master problem and a subproblem. The restricted master is an essentially unconstrained nonlinear optimization over the convex hull of all 0-1 feasible solutions found so far, using the original nonlinear objective function. The resulting continuous feasible solution is then used as the linearization point of the next subproblem, which consists in minimizing the linearized objective function over the set of original constraints. Since this algorithm is used as a heuristic whose purpose is to find a sizeable collection of 0-1 feasible solutions of the CDAP, we consider all feasible solutions collected in the solution pool by CPLEX, and do not insist on solving all MIP problems to optimality, since they are only local approximations of the original problem.

In the nonconvex case, we allow at least two changes over a standard implementation of simplicial decomposition. First it is not essential to solve every single intermediate subproblem (SP) to optimality, since we know we cannot guarantee a valid bound. It does happen in practice that out of maybe one hundred calls to the linear MIP solver, one or two take a very long time to prove optimality, and to avoid this unnecessary time increase, we set a limit to the elapsed solution time. Second, the NLP subproblem cannot be solved to guaranteed optimality in the nonconvex case unless one uses a global NLP solver, yet we are simply using the final local optimum provided by a standard NLP solver, such as MINOS or CONOPT. What actually happens is that the optimality criterion of simplicial decomposition tends to be satisfied very quickly, much more so than in the convex case, thus reducing the number of iterations and the number of integer feasible solutions exhibited. We remedy this by restarting the SD algorithm from a pre-selected set of initial linearization points. These were determined by trial and error and may have to be redefined for different types of models.

With these simplifications, CHH runs quite quickly even for relatively large CDAP instances. For the largest instance corresponding to the start of the working day in the first crossdock company, which contains 56x25+16x16 (=1656) 0-1 variables, elapsed solution time on a Core I7 Intel Thinkpad X220 was 48 seconds. This makes it feasible for the dynamic system to call repeatedly CHH to solve a variety of CDAP instances. In an actual crossdock setting, solving approximately the current CDAP instance is actually do-able in real time. In addition elapsed solution times should actually decrease towards the end of the day as the problems become smaller.

6. The simulation study of static and dynamic Assignment

Several critical things could happen in the course of crossdocking operations and require management actions.

B1. The arrival of an inbound truck. A strip door assignment is needed as soon as possible.

B2. The finish of unloading of a particular inbound trailer. Based on Assumption A7, the amount of materials in the staging area increases. As a result, accumulated products at one of the staging areas for a particular destination may exceed 90% of the FTL. This event would require that we try to assign a trailer to a stack door. It is necessary to free the inbound trailer and the strip door, once the trailer is empty. If there is an unassigned inbound trailer in the queue, we may need to make a strip door assignment.

B3. The finish of loading of a particular outbound trailer. When an outbound trailer is fully loaded, it should be taken away from its stack door. In this case, indeed, one more stack door shall be freed. If freeing that stack door occurs when outbound trailers are queued, the first outbound trailer in queue may be moved to the freed door.

These three things will be formally defined as **Events** in the simulation as in Figure 3. To clarify the process, here we have described them in words. It is worth mentioning that in the description of the events above we used words such as ‘may’ instead of saying that an assignment is bound to happen, since whether the door or trailer is assigned immediately depends on not only the current needs from operations, but also the specific strategy, as will be elaborated later. Regardless the nature of these events, we need two actions: inbound or outbound assignment. In static and dynamic assignment scheme, we handle inbound and outbound assignment differently.

Faced with the problems of cross-docking assignment as defined in section 2, one scheme is to use the static assignment strategy which can be simply described as follows. Recall that at the very beginning of the work period, we already have the information needed for solving the CDAP problem and what we lack to schedule a day is only the arrival sequence of incoming trucks. Thus, we solve the CDAP problem by CHH and produce the recommended assignments of both inbound and outbound trailers. Thereafter, whenever we need assignments, inbound or outbound, we look at the particular doors recommended by the CDAP solution, and check for availability. If it is available, then we will make assignment. Otherwise, nothing changes. The inbound truck or the destination will be put into the queue, waiting for the next chances. In case of simultaneous conflict, we use the FIFO principle, or randomly choose one in case of simultaneous events. The unassigned one will be placed into the queue.

In the dynamic assignment scheme, to the contrary, we do the following in a flexible way, depending on whether inbound or outbound assignments are needed. If the inbound assignments are needed, we re-solve the reformulated CDAP problem. The flow matrix shall be changed, and the assigned inbound trailers are removed from consideration. The strip doors currently occupied cannot be accessed, so we disregard them as well. For all the available inbound doors, we calculate their capacity in accordance with the remaining time of the shift. The stack doors being occupied, however, could still be possibly used by the flows currently being considered. But their capacities should be less than the capacity corresponding to the remaining time of the day because of the work going on. We calculate their capacities based on the estimation of the time need for work being done there, and the time remaining in which the door is available. For the unoccupied stack doors, we simply do the same thing as we do for the unoccupied strip doors. We will resolve conflict by assigning the trailer that arrives first to the optimal door, and choose the closest one for the other one if available. In case of outbound door assignments needed, we find it unnecessary to solve a CDAP problem. The part of materials to be shipped out comes from the inbound trailers that have been assigned. Thus, we do what makes the most sense: we simply utilize an available stack door that minimizes the total distances that goods would have to be moved across the crossdock floor. Conflicts are resolved the same way as for the strip door assignments.

For more particulars about how the assignments are done, we will turn to the simulation, which, although conducted under certain assumptions such as the total time needed for loading and unloading is approximately proportional to volume, mimics the real operations in detail.

To compare the dynamic and static optimization of cross-docking operations, we have developed a discrete event simulation, using MatLab©. Our simulation follows the general principles of a standard discrete event simulation [26]. In the simulation, our purpose is to mimic the real operations of cross-docking facilities to demonstrate efficacy of the dynamic optimization compared with the static optimization, in several different aspects of performance. The details of the comparison and some assumptions about the input data could be found in the next section. In this section, we discuss the components of the simulation and the outline of simulation procedure. Instead of giving all the details as pseudo codes, which could be overwhelming, we only indicate the major parts that suffice to describe the dynamic of the system. Note that the simulation of two systems using dynamic and static are essentially the same. The only differences will be highlighted in the descriptions below.

As a dynamic, stochastic system, the cross-docking operations are simulated through a series of events, which happen deterministically or randomly, noticed by Future Event List (FEL). In the course of simulation, the changes of the system are described by the changes of the following main system state variables.

Name	Description
$StripSt(i)_t$	The state of strip door i at time t. It indicates whether strip door i is currently occupied. If not, then $StripSt(i)_t = 0$; if it is occupied by inbound trailer j, then $StripSt(i)_t = j$.
$StackSt(i)_t$	The state of stack door j at time t. It indicates whether stack door i is currently occupied. If not, then $StackSt(i)_t = 0$; if it is occupied by the outbound trailer destined for destination j, then $StackSt(i)_t = j$.
$InTkSt(i)_t$	The state of inbound trailer i at time t. If it is currently being assigned to inbound door j, then it takes value j; if it has been fully unloaded at inbound door j, then it takes value $-j$; if it has not arrived yet, then it takes value 0.
$StageSt(i)_t$	Staging area state for destination i at time t. It indicates the amount of products that have been accumulated for each destination i at the staging area.
$WorkerSt(i)_t$	The state of worker i at time t. $WorkerSt(i)_t = 0$ if he is idle, or $WorkerSt(i)_t = j$ if he is being assigned to door j.

Table 1. The major state variables

Corresponding to the three types of **Events**, there are three types of **Activities** in this simulation system. **Activity 1** is the interarrival of inbound trailers. As described in assumption A2 in section 2, the exact arrival times of inbound trailers are not known to the managers of the crossdock. Therefore we deal with the inter-arrival times as random variables, the distribution of which will be discussed in the next section. **Activity 2** is the unloading of an inbound trailer. Based on assumption A1, the time taken to finish the unloading of a trailer is approximately proportional to the volume of products in the trailer. For more detail please refer to section 4. To keep the random nature of simulation, if the unloading time of trailer m is estimated as s'_m, then we assume that the actual unloading time is $s''_m = s'_m + \varepsilon_m$, where $\varepsilon_m \sim N(0, s'_m/10)$. **Activity 3** is the loading of an outbound trailer. We determine the value of it in the same way above.

Based on these Stage variables and activities, we define the following three events and system dynamic as described in Figure 3. Note that for the strip door assignment, we have two possible options, `AgnIn` or `AgnInFix` in order to determine the proper inbound doors to be assigned. If we are simulating the static assignment model, then in the beginning of a simulated work period, we will solve the CDAP for the whole work period's operation by CHH algorithm, and `AgnInFix` will only allow us to strictly adhere to the door optimizations produced. In case an optimal door is not available, we will leave the inbound trailer waiting in queue until the door is available, which is only possible when **Type 3 Event** happens. All the inbound trailers waiting for the same door will be queued in a FIFO basis.

However, in `AgnIn` we solve the CDAP by invoking a reformulated CHH. All the flows coming from the inbound trailers that have been assigned to the strip doors are removed. That is, we are planning for only the trailers not yet parked at strip doors. All the strip doors currently occupied will be ignored, in the sense that they will be removed from the distance matrix since they are not available to the current strip door assignment. On the stack door side, since the goods contained in trailers not yet parked at a strip door will not immediately be sent to the stack door side of the crossdock, we may still possibly utilize the doors currently occupied. As we explain the capacity as the time left for the day, we need to reduce the capacity for a door by the estimated time for the current loading at that door. After we get the solution, `AgnIn` assigns the inbound trailers to the optimal strip doors if available. Otherwise, we will search for the closest strip door available and make the assignment accordingly. If we can't find a convenient unoccupied strip door, we will queue the trailer.

We use the `AgnOut` and `AgnOutFix` in a similar fashion. `AgnOut` is used in the static assignment, strictly adhering to the initial CHH in the assignment of stack doors. `AgnOutFix` gives flexible assignments in the following way. For all the products accumulated at the staging area, we always keep track of the doors from which they are coming. Once we decide we need a stack door for destination k and unload the products to be shipped to a given destination, amounting to $LoadAmount$, we calculate the total distance that would be travelled by these products, if stack door i' is to be assigned. If stack door i' is already occupied, then we choose the next best stack door. We don't adhere to the solutions produced by solving a CDAP because the

Type 1 Event: Finish of loading at an stack door

Finish of the loading

Free the outbound truck that has been fully loaded and the corresponding stack door i: $StackSt(i)_t = 0$;

Free the corresponding worker j working at stack door i: $WorkerSt(j)_t = 0$.

(Possible) stack door assignment

If $StageSt(k)_t \geq 0.9 \times FTL$ for some k, we need an stack door assignment. The amount of products to be loaded into outbound trucks $LoadAmount = \max(StageSt(k)_t, FTL)$;

run `AgnOut` or `AgnOutFix`;

If stack door i' and worker j' are assigned for loading, then update $StageSt(k)_t = StageSt(k)_t - LoadAmount$, $StackSt(i')_t = k$ and $WorkerSt(j)_t = i'$;

Calculate the time needed for the loading at stack door i', schedule next **Type 1 Event** on FEL.

Event 2: Finish of unloading at an strip door

Finish of the unloading

Free inbound truck i that has been fully unloaded at strip door j: $StripSt(i)_t = 0$, $InTrSt(j)_t = -i$;

Free the corresponding worker k working at inbound door i: $WorkerSt(k)_t = 0$;

Let the flow on truck i to destination l be w_{il}. Then we update the state of staging areas as $Stage(l)_t = Stage(l)_t + w_{il}$.

(Possible) stack door assignment

If $StageSt(k)_t \geq 0.9 \times FTL$ for some k, we need an stack door assignment. The amount of products to be loaded into outbound trucks $LoadAmount = \max(StageSt(k)_t, FTL)$;

Run `AgnOut` or `AgnOutFix`;

If stack door i' and worker j' are assigned for loading, then update $StageSt(k)_t = StageSt(k)_t - LoadAmount$, $StackSt(i')_t = k$ and $WorkerSt(j)_t = i'$;

Calculate the time needed for the loading at stack door i', schedule next **Type 1 Event** on FEL.

(Possible) strip door assignment

If there are inbound trucks i'' that have arrived but not yet assigned, we need an strip door assignment;

Run `AgnIn` or `AgnInFix`;

If strip door j'' and worker k'' are assigned for unloading, then update $StackSt(j'')_t = i''$ and $WorkerSt(k'')_t = j''$;

Calculate the time needed for the unloading at stack door j'', and schedule next **Type 2 Event** on FEL.

Event 3: Arrival of an inbound truck

(Possible) strip door assignment

Inbound truck i has just arrived. If there are strip doors available, we need an strip door assignment;

Run `AgnIn` or `AgnInFix`;

If strip door j and worker k are assigned for unloading, then update $StackSt(j)_t = i$ and $WorkerSt(k)_t = j$;

Calculate the time needed for the unloading at stack door j, schedule next **Type 2 Event** on FEL.

Calculate the time needed for next inbound truck arrival. Schedule next **Type 1 Event** on FEL

Figure 3. The events of simulation

strip door assignments associated with these products are already decided and applied, and we are now considering only the assignment on one side of the crossdock.

The philosophy here for `AgnInFix` and `AgnOutFix`, and hence the dynamic assignment is that at any point of the decision making, we optimize the future without regard to the past. We deem it as sunk cost; having no impact on future operations.

In the simulation, we assume that whenever the amounts accumulated for a destination exceed 90% of FTL, we attempt to make a stack door assignment. We do so in order to fully utilize the trailer capacity as guided by the prevailing cross-docking operation principles. But in practice, we could be more flexible and intelligent and make assignment judiciously by taking into consideration other factors. For example, we may try to fully utilize the trailer capacity in the morning and make assignments only when there is more than 90% of full trailer load accumulated, but in the afternoon we may lower the criterion to quickly clear out the floor before the end of the work period.

Next we use a small example of one work period of operations to illustrate the dynamic assignment scheme. Suppose in a small sized cross-docking facility we have three doors on each side. We assume three inbound trailers will arrive in that work period carrying goods for just four destinations. The flow matrix and arrival times are given by Table 2, and distance matrix given in Table 3, where d_{ij} corresponds to the distance from strip door i to stack door j.

Inbound Trailers	Flow				Arrival times
1	2	3	1	1	0
2	0	1	0	1	0
3	0	2	0	3	1

Table 2. The flow matrix and the arrival times

1	2	3
2	1	2
3	2	1

Table 3. The distance matrix

The initial state of the staging area is given by the $(1, 1, 4, 0)$ which is the amount of products left from yesterday. All these products were coming from strip door 1. Let's assume the FTL is 5. All the loading and unloading takes exactly two units of time. That is, for ease of exposition, suppose we don't have randomness in unloading or loading. We make unbound door assignments only when the accumulated products exceed 90% of FTL. Also, we assume the optimal assignment at time click 0 is given by Table 4.

	Inbound trailers			Destinations			
Door Assignment	2	3	3	2	2	3	3

Table 4. The Optimal Assignment at time 0

Then for the **Type 1 Event** at time click 0, the arrival of trailers 1 and 2, we make actual assignments according to the optimal solution of CDAP, as depicted in Figure 2. Note that no other changes should happen at this time. According to our assumption, we schedule Type 2 Event at time click 2. Then at time click 1, one more inbound trailer is coming in, and is recommended to be assigned to strip door 1 by CDAP. Then one more **Type 2 Event** shall be scheduled at time 3.

At time click 2, we free the trailers 1 and 2 from the strip doors, and therefore we have the state of staging area changing to$(3, 5, 5, 2)$. And the details of their strip doors of origin are in Table 5.

Destination 1		Destination 2		Destination 3		Destination 3	
Amount	Strip Door	Amount	Strip Door	Amount	Strip Door	Amount	Strip Door
1	1	1	1	2	1		
2	2	3	2	1	2	1	2
		1	3	2	3	1	3

Table 5. The details of staging area

Now the products accumulated at the staging area require two outbound trailers, for destinations 2 and 3 respectively. If we assign destination 2 to stack door 1, the total distance travelled will be $1 \times 1 + 3 \times 2 + 1 \times 3 = 10$. Similar calculation tells us that the total distance will be 7 and 10 respectively if we assign the outbound trailer going to destination 1 to stack door 2 and 3. Similar calculation says that stack door 1 is optimal for destination 3. Thus these calculations result in the indicated stack door assignments at time 3 in Figure 4. The unhandled materials left in the staging area will therefore become $(3, 0, 0, 2)$. We schedule **Type 3 Event** at time click 4 and release these two outbound trailers.

At time click 4, we finish the unloading of inbound trailer 3. The staging area state therefore becomes$(3, 2, 0, 5)$. We need one outbound door assignment for destination 4. The total distances will be 8, 9 and 12 for three possible stack assignments. Since the first and second best assignments are not available, we use stack door 3. So finally the staging area state becomes $(3, 2, 0, 0)$. Since there will be inbound trailers coming in, and no further increase of the materials at the staging area is possible, we will no longer have stack door assignment.

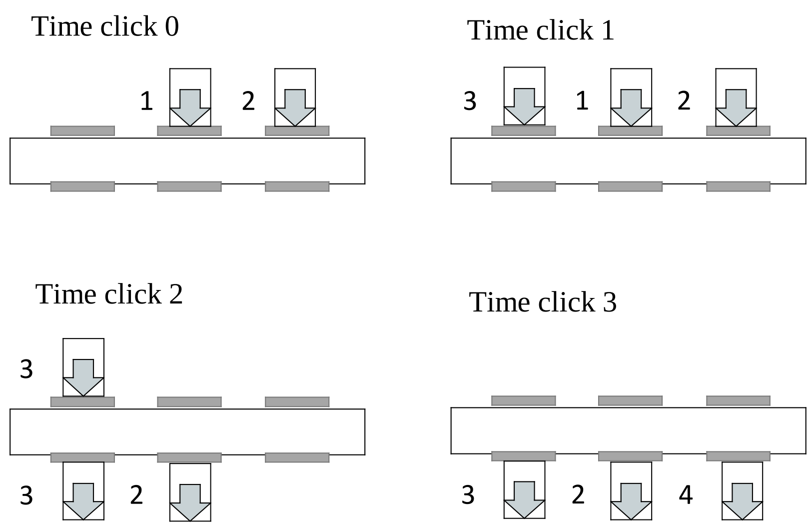


Figure 4. A simple illustration

7. Simulation experiment protocols and results

We compare the impact of our dynamic model and static CDAP model through controlled experiments and the resultant operations statistics. The data used in the experiment are taken and modified from the one-day operational data of the first cross-docking firm with which we were working. The original data set specifies the 25 inbound and 16 outbound doors available, as well as the size 25x16 distance matrix. Also, the flow matrix of size 56x16 as given in the original data represents the typical operation of the cross-docking facility of the first firm.

Besides the flow matrix, we still need to construct other inputs of experiments under certain assumptions. To simulate and demonstrate the superiority of our dynamic model in the real operations, we consider the cumulative effect in 5 consecutive days of both schemes in a controlled experiment. The control group – the group with static assignment – has exactly the same flows and arrival times as the group of dynamic assignment in each single day. In each day, we introduce random disturbance on the original flow matrix to mimic the reality. For example, based on $(F)_{ij}$ in the original flow matrix, at day *t*, we let $(F^t)_{ij} = max((F)_{ij} + \varepsilon_{ij}^t, 0)$ be the flow from *i* to *j*, where ε_{ij}^t is a random normal distribution with mean 0 and standard deviation $(F)_{ij}/10$. This arrangement allows us to keep the pattern of the original data, while the random nature of real operations is taken into account. The communication with operations managers in the cross-docking facility indicates that, although we have the full knowledge about the content of the arriving trailers prior to a day, we are uncertain about their arrival times. Recall that we assume that one work period's shift consists of ten working hours. In our first experiment, we assumed that the arrival time of all the incoming trailers is uniformly distributed in the first eight hours, as can be seen in Figure 5.

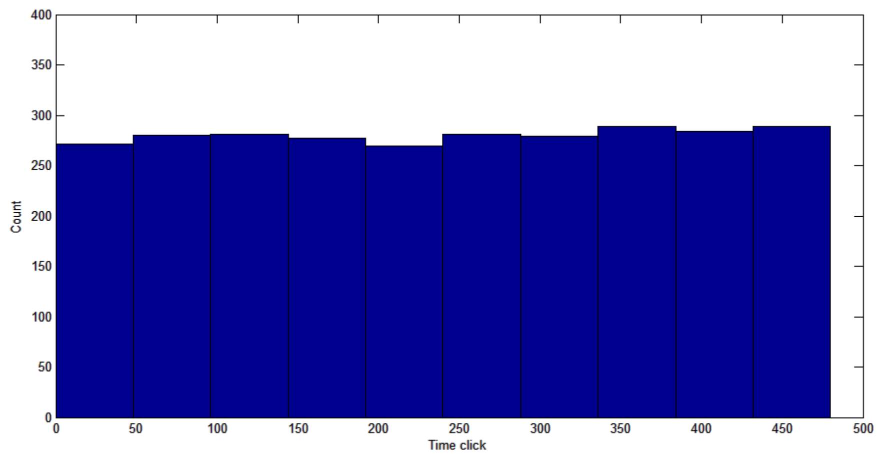


Figure 5. The arrival times of inbound trailers

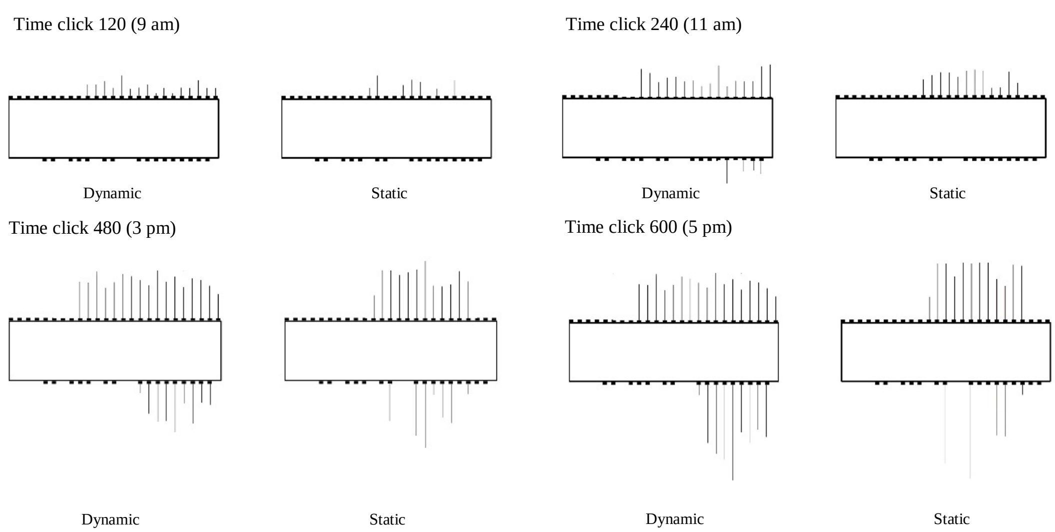


Figure 6. The flows assigned to doors

Then our experiment leads us to typical materials accumulated at each door as presented in Figure 6, which shows the material accumulation at both strip and stack doors at four different time clicks of a work period. Once an inbound trailer with certain amount of materials has been assigned to a particular strip door, the amount of materials at this door will be immediately increased by this amount. Therefore these flows would be reflected in the figure thereafter. We employ similar calculation to the stack doors. The left represents the dynamic strategy well the right represents the static strategy.

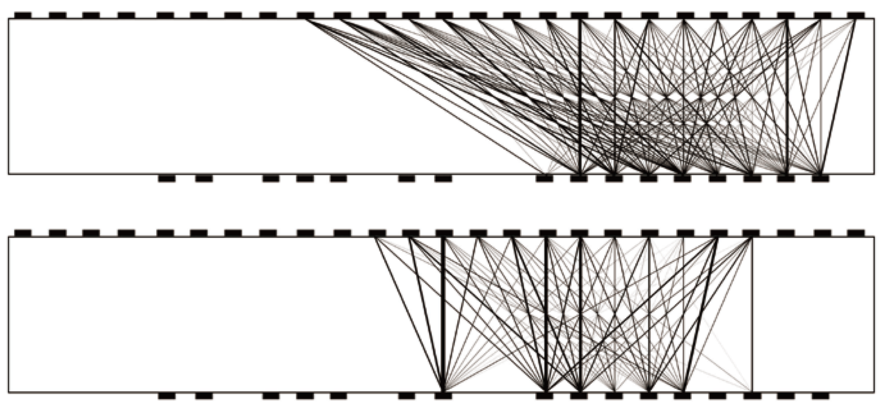
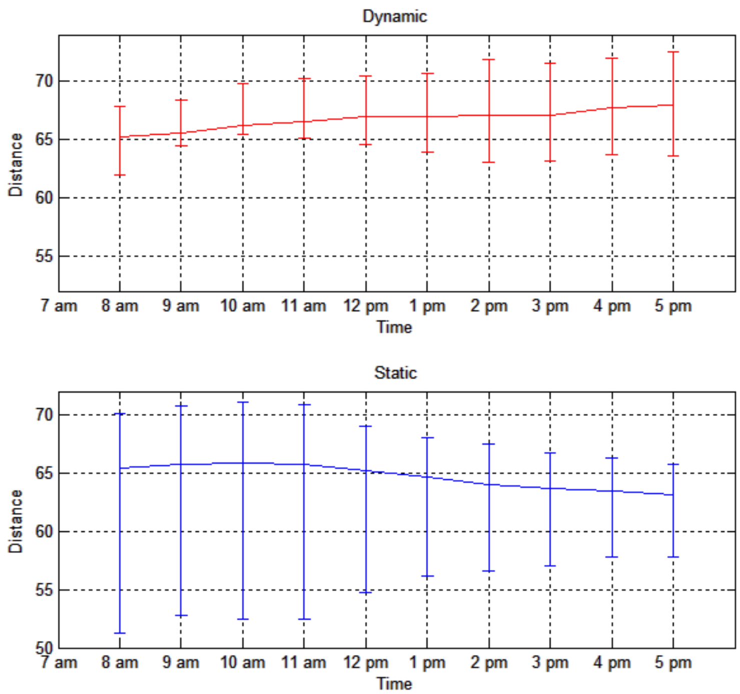


Figure 7. Actual flow movement diagrams

It first can be seen that both strategies are helpful in achieving our secondary goal -- saving the distance travelled by products. Note that our cross-dock has this special structure -- on the left side, the stack doors are relatively far from each other and the strip doors. Both strategies help us escape from that region and make more assignments on the right side. Further, it is easy to observe from the figure that the dynamic CHH CDAP solver better utilizes the doors, as the flows are more evenly distributed to both inbound and outbound doors. This allows us to better use door capacity to work efficiently. In fact, from time click 480 to time click 600, no new inbound assignments are produced in the dynamic assignment strategy, since all trailers have been assigned to the strip doors before time click 480 (as opposed to what happens in the static strategy). This flow pattern resulted from dynamic strategy not only helps us to better utilize the doors so that work can be finished in a timely manner, it also avoids the over-crowded operations in one small region. [10a]

Figure 7 gives another representation of the operations of a work period in our experiment. The line widths are proportional to the magnitude of the flow. The diagram on the top shows how the flows travel inside the cross-dock when dynamic assignment strategy is assumed while the bottom shows what happens in static assignment strategy is employed. It can clearly be observed from the static assignment strategy that the CDAP problem helps us to keep the large flows from long distance travel. The salient thick lines almost directly connect the doors directly opposite to each other. In the dynamic assignment strategy however, although the thick lines still appear to be vertical or nearly vertical, the flows are more evenly dispersed across doors. (But note that in fact



more flows, to be exact three more inbound trucks, are handled this day in the dynamic assignment.)

Figure 8. The actual distance travelled by each unit of material

Intuitively, with the observation of flow diagrams, it seems that the static assignment helps us to save distance travelled. We naturally wonder if this is truly the case. In Figure 8, we present the average distance travelled by each unit of goods. The bar in the top and the bar in the bottom represent the maximum and minimum value respectively. For example, at time t, the total amount of flow from strip door i and stack door j is given by f_{ij}^t while the distance of these two doors is given by d_{ij}, we have the average distance equals to $\sum_{i,j} f_{ij}^t d_{ij} / \sum f_{ij}^t$. The bar on the top indicates the maximum value, and the bar on the bottom is the minimum. One can see that the average distance is relatively steady for both the static and dynamic strategy, and that indeed the static strategy appears to be superior. But with the passage of time, the average distance travelled by materials is gradually growing. This is intuitively clear: we have many stack doors occupied, as more and more materials accumulated at the staging area, and more and more outbound operations required. To facilitate the operations, (to achieve our primary goal), we have to sacrifice the distance to some extent. But to the contrary, in the static situation, we have the opposite trend since at the beginning of the work period, we still have some materials left from an earlier work period in the staging area not cleared out. While the CDAP solver CHH is optimizing the distance travelled by the products from all the inbound trucks to be handled today, these leftovers are unaccounted for. With time passing, this effect wears out and we see the average distance go down. It is also interesting to notice larger variations of distance in the simulations with static assignment. This seems to indicate that the average distance travelled depends a lot on the quality of the solution of CDAP in the static assignment strategy, which might not be stable because 1) CHH in essence is a heuristic algorithm, and 2) it does not take into account the material left from the day before. The dynamic assignment strategy on the other hand, on the average is less sensitive to what happened before and to the quality of the solution.

For comparison purposes, to get a sense of the distance saving of the static strategy, one can look at Figure 9. The distance from strip door 1 to stack door 1 is 53 while the distance from strip door 1 to stack door 3 is 64. It seems that, practically speaking, the difference of these two strategies is not very significant in terms of the distance being saved.

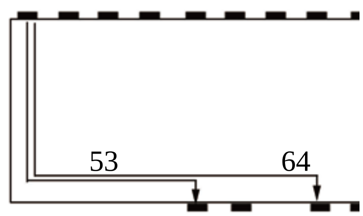


Figure 9. Door distances

We now move on to other aspects of the comparison of the static and dynamic strategies. Recall that our primary goal is to finish the work in a timely manner. We display the relative measures in Table 6. All results are the average across of ten repetitions of five work periods of operation. Note that from the table the superiority of the dynamic assignment strategy is clear in terms of our primary objective. Flows move faster inside the cross-dock. More materials and inbound trucks are left unhandled in the static assignment strategy. The door capacity of the cross-dock in the dynamic assignment strategy is better utilized.

With more doors being utilized in the dynamic assignment strategy, naturally more workers are needed in the operations. Our communication with firms one and two indicate the following facts. Efficiently employing the human resources is important. In the typical cross-docking operations of the two companies we are working with, about 80% of the workers are temporary and reserved from agencies. Once workers are called in for crossdocking work, the crossdock managers hope these workers could be better scheduled, so they could be better utilized in their shift, since they still need to be paid even when idle. According to Table 6, our dynamic assignment helps us to better achieve these goals. The workers worked longer on average and were less idle. The worker utilization rate in the dynamic situation is better than the one in the static situation.

Measure	Dynamic	Static
Average percentage of material unshipped at the end of the day	13.04%	23.94%
Average number of trucks unhandled at the end of the day	0.08	2.88
Average inbound trailer turn-around time[4]	3.42hr	5.23hr
Maximum inbound trailer turn-around time	19.23hr	20.84hr
Average material turn-around time[5]	5.23hr	8.17hr
Maximum material turn-around time	23.31hr	25.89hr
Average door utilization rate[6]	0.25	0.29
Average worker number	28.60	18.00
Average worker idle time	1.41hr	1.57hr
Average worker working time	5.70hr	4.84hr
Average worker utilization rate[7]	0.82	0.78

Table. 6 Comparison of different measures of simulation of dynamic and static assignment strategy

The above results correspond only to uniformly distributed arrival times. We postulate that the arrival time distribution could also be an important factor. If the arrival times are more concentrated at certain hours of a day, we expect the dynamic assignment strategy to perform even better, as in such situation the cross-dock would be more jammed with inbound trucks awaiting for handle, and the dynamic assignment strategy could more efficiently use the door capacities. To demonstrate this point, we perform two additional sets of experiments with different arrival time distributions of inbound trucks. In one set of experiment, we make the assumption that the arrival time distribution is truncated from normal distribution $N(240, 120)$ so

[4] The inbound truck turn-around time is defined as the time starting from the truck arriving at the facility to the finish of its unloading.
[5] The material turn-around time starts from the arrival to the facility, and ends when it's been loaded to the outbound truck.
[6] The door utilization rate is the ratio defined by occupation time/time of a shift (= 10 hours)
[7] The worker utilization rate is the ratio defined by woking time/(idle time + working time)

that the support of the distribution is $[0, 480]$, as displayed in Figure 10. In another set of experiments, we only allow arrival times uniformed distributed in the segment$[180, 300]$. We call these two groups of experiments Experiment 2 and 3. Similar to Experiment 1, in both groups of experiments we repeat the simulations of five days of operation for ten times.

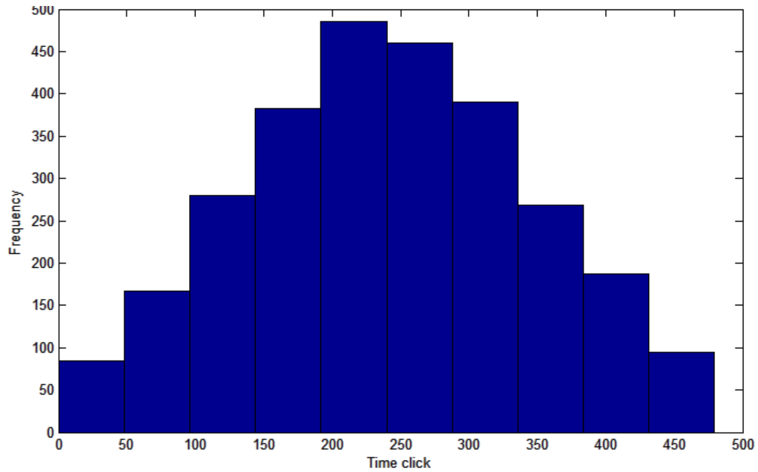


Figure 10. The arrival time distribution of inbound trucks in experiment 2.

Measure	Experiment 2		Experiment 3	
	Dynamic	Static	Dynamic	Static
Average percentage of material unshipped at the end of the day	13.67%	25.18%	14.01%	27.33%
Average number of trucks unhandled at the end of the day	0.08	3.44	0.12	4.62
Average inbound truck turn-around time	3.48hr	6.02.hr	3.93hr	7.52hr
Maximum inbound truck turn-round time	19.21hr	22.45hr	20.54hr	23.28hr
Average material turn-around time	5.38hr	8.83hr	5.82hr	10.95hr
Maximum material turn-around time	28.31hr	28.89hr	29.31hr	29.06hr
Average door utilization rate	0.29	0.23	0.26	0.23
Average worker number per day	30.00	17.20	34.42	18.32
Average worker idle time	1.38hr	1.54hr	1.15hr	1.57hr
Average worker working time	5.16hr	4.64hr	4.14hr	4.23hr
Average worker utilization rate	0.78	0.75	0.80	0.73

Table. 7 Comparison of different simulation measures of dynamic and static assignment strategy of Experiment 2 and 3

It is clear from Table 7 that the more concentrated the arrival times, the more difficult the operations would be. The congestion resulting from the large number of arrivals in a short time interval has a negative impact on various aspects of operations. For both static and dynamic assignment schemes, more trailers and materials are left unhandled at the end of the day when more concentrated arrivals take place. Operations get slower, as the inbound truck turn-around time and material turn-around time increase. This effect is more prevalent in the static situation, as can be readily observed in Table 7.

Dynamic assignment strategy, through flexibly using the capacity of the crossdock, handles this congestion better. But this comes at a certain price: we can observe the salient increment in the number of workers needed in the dynamic assignment strategy in experiment 2 and 3. But the number of workers required and working for the static assignment scheme remains steady. Especially, workers work for considerably shorter time on average in the dynamic scheme in Experiment 3. This observation may suggest precaution to the crossdock managers in considering using the dynamic assignment scheme. Only in Experiment 3, we observe for the first time that a dynamic assignment may result in shorter labor hours than for the static strategy.

When we are facing operations where the arrival times tend to be concentrated at certain hours of a day, and the efficient run of cross-dock is the most important concern, a dynamic assignment strategy becomes even more valuable.

One final result we are going to present, in Table 8, is the average running times of the CHH algorithm in simulations. We are only presenting the running times for some problem sizes as an illustration. The size of the instances is measured by the number of origins (inbound trailers) by the number of destinations, by the number of strip doors by the number of stack doors. These running times are small enough to allow operations managers to solve these problems on the fly whenever an inbound trailer assignment is needed.

Size	Running time	Size	Running time
63x16x25x16	219.54	34x16x8x16	144.40
60x16x25x16	211.09	30x16x15x16	149.33
56x16x25x16	208.31	24x16x11x16	110.03
51x16x20x16	230.22	21x16x14x16	125.62
48x16x19x16	191.23	16x16x10x16	35.39
44x16x16x16	171.12	13x16x15x16	19.91
40x16x15x16	158.40	8x16x9x16	7.00
37x16x13x16	115.71	4x16x15x16	2.04

Table 8. The average running time of CHH algorithm in the simulation

8. Conclusions

We are proposing to handle the dynamic assignment of doors to trailers in a crossdock by using repeated solutions of the static CDAP model obtained by running the CHH MatHeuristic and reassigning doors dynamically, whenever there are real-time changes in the cross-dock and the CDAP problem previously defined is no-longer representative of the situation in the crossdock. We briefly recap the necessity and essence of the dynamic assignment strategy in the following.

To improve the experience-based operations commonly found in today's crossdocks, we offer a door assignment optimization tool for crossdock managers that will reduce the distance travelled by goods across the crossdock as well as the workload and even the labor cost. CDAP optimization is defined to minimize the total distance travelled by the materials inside the crossdock subject to the constraint that one trailer can only be assigned to one door, and the door capacities are limited by the time we can utilize them in a working day. While CDAP is a useful guide, our close cooperation with two firms managing crossdocks indicates that to blindly follow CDAP optimization by itself could negatively impact delivering goods efficiently.

Some key guiding principles regarding this are the following rules-of-thumb from crossdock managers. First of all, materials need to be sent out from the facility efficiently. Sometimes crossdock managers will have to pay fines for goods not shipped out within 48 hours. When the customers evaluate the services provided by the crossdock, they sometimes consider the efficiency of processing the materials as the most important factor. In either case, our communications with the two firms with which we have worked indicates that this is always the most important performance measure in their operations. The inbound trailers should also be returned efficiently. Delay in doing this may result in fines. In view of this, and considering that the situation changes constantly in the course of cross-docking operations, maybe to the extent that it becomes very different from our original planning, we introduce a dynamic assignment scheme to efficiently utilize the solutions from CDAP as the guide of the door assignment.

In summary, we do as follows in the dynamic assignment scheme. Whenever a strip door assignment is needed, we look at our cross-dock and check for available strip doors by removing all the doors that cannot be of immediate use, calculate their remaining capacities, and solve the CDAP problem in real-time. On the stack door side, however, when an assignment is required, since all the origin doors of the flows that need to go across the cross-dock are already assigned, we only search the best stack door in terms of distances. Note that this scheme is made possible by the Convex Hull Heuristic (CHH) to solve the CDAP problem, which is a nonlinear integer problem and is NP-complete. Our dynamic scheme requires solving the problem for a dozen times in practice and no delay of assignment should happen due to the long time of solving CDAP, thus to implement this scheme we need a good heuristic such as CHH to find a satisfactory door assignment in a very short time. The reassignment rules are simple to implement and the CH heuristic takes only one or two minutes on a PC. This dynamic scheme thus gives us a holistic solution to the trailer-to-door assignment problem in the cross-dock.

To compare both schemes, static and dynamic, we resort to simulations and test this approach on a fictitious but realistic example crossdock. Our simulation input data are taken and modified from operations observed at a freight company with whom we are collaborating, and we make different assumptions on the inbound trailer arrival distributions. We do observe that the average distance travelled by the materials is not increasing much in the dynamic scheme, compared to the static scheme. It shows that the dynamic assignments remain to be a practical guidance of the real operations to reduce the distance travelled by the products. To our delight, we find out that the saving on other aspects in the dynamic scheme is evident: no matter what arrival distribution we are assuming, only about a half of the materials are left to handle for each day. Nearly no inbound trucks are left and unhandled. Both trucks and materials are staying far much shorter times. The advantage becomes even more obvious when we assume more concentrated arrivals of inbound trucks.

It is observed in all the experiments that more workers are needed for the dynamic assignment strategy. When the arrival distribution is assumed to be uniform, we see that workers, once called in, work for a relatively long time, which is a desired operations feature of cross-docking. But when a more concentrated inbound distribution is introduced, we see workers work for shorter times in the cross-dock. In the most extreme case, as we considered in Experiment 3, workers on average work for shorter times than their counterparts in the static assignment scheme. This may suggest precaution for managers to judiciously choose the right scheme for the operations.

With the ongoing cooperation with practitioners, which allows us the access to more of the accurate operations data and the opportunity to apply our model, we are considering the following aspects.

Within the framework of our current distance-based dynamic CDAP model, we are considering developing a more complete solution. Workers in our simulation are called in on an ex-post basis. We produce the number of workers needed in the course of the simulation. But in reality, practitioners will have to plan ahead. Based on the results of simulation, we will try to control the worker numbers and move our research to a more direct approach -- a labor cost based approach, -- while the time related constraints are still satisfied and the distance is still being optimized.

The relationships between distances and time, and between distances and labor costs, are not directly defined in our current research. With the access to the operations data, and with certain statistics, we will be able to solve the model with these relationships clearly defined. Another promising direction to pursue is that, without changing the structure of the CDAP problem, we might alter the meaning of coefficients of the CDAP model. Instead of the flow timed by distances travelled, the coefficients could directly be the time taken by the operations once certain assignment is hypothesized.

The CDAP problem could also be modified in other directions. Instead of assuming one destination could only be handled at one stack door, we may relax the problem and allow fractional numbers. In this fashion, the CDAP problem will better mimic the real operations. Inside congestion is also an important problem to be considered, as discussed in [10a]. Directly modeling the congestion based on empirical methods and having the CDAP problem modified accordingly are also possible. We should mention to the reader that these changes should cause no difficulty for solving. CHH is designed to tackle general non-linear objectives [21a] [21b] [22]. We could modify the algorithms accordingly to suit these formulations.

Future research might also involve an extended CDAP model with time-indexed variables. Such a model may be impossible to solve exactly, but the CH heuristic can likely be extended to handle this enlarged model, albeit requiring longer solution time. However it will need to be executed only if there is an unexpected change in the arrival schedule.

References

[1] D. Agustina, C. Lee, R. Piplani, "A Review: Mathematical Models for Cross Docking Planning," *International Journal of Engineering Business Management*, 2(2), 47-54, 2010.

[2] N. Boysen and M. Fliedner, "Cross dock scheduling: Classification, literature review and research agenda," *Omega*, vol. 38(6), 413-422, 2010.

[3] D. Konur and M. Golias, "Cost-Stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach," *Transportation Research Part E*, 49, 71-91, 2013.

[4] P. M. Hahn, "A note on the relationship between the CDAP and the GQAP," *Private Communication*, University of Pennsylvania, ESE Department, 2006.

[4a] P. M. Hahn, "A note on the relationship between the CDAP and the GQ3AP," *Private Communication*, University of Pennsylvania, ESE Department, 2007.

[5] P. M. Hahn, B. J. Kim, M. Guignard, J. M. Smith, and Y.-R. Zhu, "An algorithm for the generalized quadratic assignment problem," *Computational Optimization and Application*, 40(3) 351–372, 2008.

[6] A. Ahlatçioglu, M. Bussieck, M. Esen, M. Guignard, J. Jagla, A. Meeraus, "Combining QCR and CHR for Convex Quadratic Pure 0-1 Programming Problems with Linear Constraints," *Annals of Operations Research*, vol. 199, 33–49 , 2012.

[6b] B. von Hohenbalken, "Simplicial decomposition in nonlinear programming algorithms," Mathematical Programming 13, 49-68, 1977.

[7] Y-R. Zhu, P. M. Hahn, Y. Liu and M. Guignard, "New approach for the crossdock door assignment problem," *Proceedings of the XLI Brazilian Symposium on Operations Research*, Porto Seguro, Bahia, Brazil , 2009.

[8] M. Guignard, P. M. Hahn, A. Alves Pessoa and D. Cardoso da Silva, "Algorithms for the Crossdock Door Assignment Problem," *Proceedings of the Fourth International Workshop on Model-Based Metaheuristics* – Angra dos Reis, Rio de Janeiro, Brazil, September 17 to 20, 2012.

[9] K. Gue, http://kevingue.wordpress.com/research/crossdocking/

[10] K. Gue, “Warehouses without inventory,” *International Commerce Review*, Volume 7 (2), 2007.

[10a] J. Bartholdi III and K. Gue, “Reducing Labor Costs in an LTL Crossdocking Terminal,” *Operations Research* 48 (6), 823-832, 2000.

[10b] K. Gue, “The Effects of Trailer Scheduling on the Layout of Freight Terminals,” *Transportation Science* 33 (4), 419-428, 1999.

[11] L.Y. Tsui, and C.-H. Chang, “An optimal solution to a dock door assignment problem,” *Computers & Industrial Engineering*, 23(1-4)283-286, 1992.

[12] I. F.A. Vis and K.J. Roodbergen, “Layout and control policies for crossdocking operations”, *Computers & Industrial Engineering*, 61, 911–919, 2011.

[13] D. L. McWilliams and M.E. McBride, “Exploring mathematical approximation for the time spans of transfer operations in parcel transshipment terminals,” *Computers & Industrial Engineering* 64, 342–356, 2013.

[14] T.W. Liaoa, P.J. Egbelu and P.C.Chang, “Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door crossdocking operations,” *Int. J. Production Economics* 141, 212–229, 2013.

[15] G. Alpan, R. Larbi, B. Penz, “A bounded dynamic programming approach to schedule operations in a crossdocking platform,” *Computers & Industrial Engineering* 60, 385–396, 2011.

[16] G. Alpan, A.-L. Ladier, R. Larbi, B. Penz, « Heuristic solutions for transshipment problems in a multiple door crossdocking warehouse,” *Computers & Industrial Engineering* 61, 402–408, 2011.

[17] A. Shuib, Wan Nor Ashikin Wan Ahmad Fatthi, “A Review on Quantitative Approaches for Dock Door Assignment in Crossdocking,” *International Journal on Advanced Science Engineering Information Technology*, 2(5), 2012.

[18] J. Van Belle, P. Valckenaers, D. Cattrysse, “Crossdocking: State of the art,” *Omega*, 40, 827–846, 2012.

[19] Y.-R. Zhu, P.M. Hahn, Y. Liu and M. Guignard, “New approach for the crossdock door assignment problem,” *Anais do XLI Simposio Brasileiro de Pesquisa Operacional*, Porto Seguro, Bahia, Brazil, 2009.

[20] V. Albornoz, “Diseño de Modelos y Algoritmos de Optimización Robusta y su Aplicación a la Planificación Agregada de la Producción,” Doctoral Dissertation, Universidad Catolica de Chile, Santiago, Chile, 1998.

[21a] A. Ahlatçıoğlu and M. Guignard, “The convex hull relaxation (CHR) for convex and nonconvex MINLP problems with linear constraints,” OPIM Dept. Report, Univ. of Pennsylvania, 2010.

[21b] M. Guignard and A. Ahlatçıoğlu, “The Convex Hull Relaxation for Nonlinear Integer Programs with Convex Objective and Linear Constraints,” European Workshop in Mixed-Integer

Nonlinear Programming, Marseille, France, May 2010.

[22] M. Guignard, “A new, solvable, primal relaxation for convex nonlinear integer programming problems,” Optimization Online, *http://www.optimization-online.org/DB_FILE/2011/01/2904.pdf*, 2011.

[23] S. Ji, X. Zheng, X. Sun, "An Improved Convex 0-1 Quadratic Program Reformulation for Quadratic Knapsack Problems,” *Pacific Journal of Optimization*, 8(1) 75-87, 2012.

[24] H.D. Sherali, W.P. Adams, “A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems”, *SIAM J. Discrete Math*. 3 (3) 411-430, 1990.

[25] F. Abiprabowo, N. Harinsuit, S. Lim, W. Zhang, “Designing a Simulation Tool for Commercial Cross-Docking Application”, Report, Senior Design Project, 2012-2013, ESE Department, University of Pennsylvania, Peter Hahn and M. Guignard, advisors, see also *https://www.informs.org/Recognize-Excellence/INFORMS-Prizes-Awards/Undergraduate-Operations-Research-Prize*, also *http://www.dvewc.org/student-papers2014/*

[26] J. Banks, J. S. Carson, B. L. Nelson, and D. M. Nicol, *Discrete-Event System Simulation*, Prentice Hall, 2010

[27] G. Arnaout, E. Rodriguez-Velasquez, G. Rabadi and R. Musa, “Modeling cross-docking operations using discrete event simulation”. in *Proceedings of the 6th International Workshop on Enterprise & Organizational Modeling and Simulation*, 2010.

[28] E. Rodriguez-Velasquez, G. Arnaout, G. Rabadi and R. Musa, “Modeling cross-dock networks with time constraints using simulation,” in *Systems and Information Engineering Design Symposium, IEEE*, 2010.