Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

SIMULATION MODELING OF ALTERNATIVE STAFFING AND TASK PRIORITIZATION IN MANUAL POST-DISTRIBUTION CROSS DOCKING FACILITIES

David Alan Cox
Manuel D. Rossetti

University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701, USA

ABSTRACT

Many supply chains have grown increasingly complex, which has led to the development of different facility types. One such facility is known as a post-distribution cross docking system (Post-C). In these facilities, bulk sorted product is received from various suppliers. Each product has its own destination, so the bulk package is broken, sorted by destination, and staged by destination. Typical processing includes: sort received goods by product type; break bulk and sort out goods by destination; move palletized goods to the staging areas of their respective destinations. This paper compares a global staffing policy (in which all workers may perform any task) to a dedicated staffing policy (in which groups of workers are assigned specific tasks). Through comparisons of the two models, the results indicated that the dedicated worker model's benefits from reduced change-over outweigh the lower worker utilization it experiences.

1 INTRODUCTION

Logistics has become a crucial field to many companies throughout the world. Within these advancing supply chains, companies utilize warehouses, distribution centers (DCs), and cross-docking facilities (CFs). Each of these are specifically designed to minimize costs within the company's supply chain.

CFs are the focus of this paper, so it is important to define a CF. For the sake of this paper's study, a facility is considered a CF if all received goods are shipped out within 48 hours of receipt. CFs have been adapted to fit the needs of various industries, which is why it is important to see how these adaptations effect previously studied CF models. One adaptation seen is an intermediate sorting process, carried out between the time of receipt and shipment. These intermediate sorting processes involve taking bulk-packaged goods, breaking bulk, and re-packing the goods by destination.

A generic CF simulation model was created by Magableh, Rossetti, and Mason (2005), which can be used to individually analyze CFs. "The generic CF simulation model incorporates the following problem aspects: resource contention for dock doors, flexible assignment of loads to in-bound (IB) and out-bound (OB) doors, worker resource requirements, material handling contention, and outbound load building" (Magableh, Rossetti, and Mason 2005). This study aims to use this simulation modeling approach to analyze a CF with intermediate sorting involved.

The next section of this paper provides the motivation and background for the development of this simulation model. An outline of the operations will be provided, and then the conceptual modeling will be outlined. The development of this model will be described before conclusions are made based on investigating staffing policies for allocating workers to tasks.

2 BACKGROUND

CFs are seen in several different stages of supply chains. For example, a CF may be used to consolidate orders shipped from suppliers to DCs in retail supply chains (Magableh, Rossetti, and Mason 2005), which Tang and Yan (2009) define as a pre-distribution cross-docking system (Pre-C). In a similar supply chain, a post-distribution cross-docking system (Post-C) may be utilized between the suppliers and several department stores. In these Post-Cs, inbound cases are deconsolidated, sorted according to customer requests, and reconsolidated into outbound cases (Fanti, Stecco, and Ukovich 2016). These Post-Cs contrast from a typical CF, where goods arrive at a cross-dock from suppliers, are counted, get consolidated, and are loaded onto outbound vehicles (Tang and Yan 2009). The primary difference between these two CF types stems from where sorting operations take place. Unlike Post-Cs, the suppliers of all Pre-Cs sort product before shipping it.

An existing Post-C was observed, and it was found that the reason for its methods for operation were not well known. All operations were manually carried out, and its sorting was performed by the staff using a put-to-light system. For this facility, all workers were responsible for the completion of all tasks, yet it was not known whether or not the facility would benefit from dedicating its workers to specific tasks. Literature based on studies pertaining to Post-Cs were found, and it was found that other studies failed to address the comparison of global workers and dedicated workers within a Post-C.

As found in Tang and Yan's study (2009), Post-Cs are advantageous for companies who experience an unstable demand and/or long lead times from its suppliers, when compared to standard DCs and Pre-Cs. Although these systems do enjoy the benefits of risk-pooling, they suffer from higher operations cost, due to the intensive sorting processes. In Tang and Yan's study, the primary work investigated the overall cost of the system, which included overstock, backorders, inventory holding, and shortage costs. However, it did not address the impacts of various approaches to the sorting process.

In Fanti et al. (2016), a Post-C with a sorting machine was analyzed to determine the optimal scheduling of internal operations. This Post-C used a sorting machine to pack outbound boxes with mixed product, and packed boxes were consolidated onto outbound pallets. Mixed integer linear programs and a heuristic algorithm were created in order to solve the complex problem of scheduling the internal operations in Post-Cs. Although this work provided models capable of determining the optimal schedule for a Post-C's internal operations, it was designed based on an automated sorting system.

One study established more well defined types of cross-docks, and it defined Post-Cs as Cross-Dock Managed-Load (CML) facilities. This study surveyed facilities and found nine factors all facilities considered to be valid success factors. These success factors included efficient physical facility design and layout, as well as work balancing and minimization, but the Post-Cs, or CMLs, considered were automated. This contributed to the research gap found in manually operated Post-Cs (Vogt 2010).

Many other studies have also addressed generic warehouses, but do not acknowledge the intermediate operations, such as sorting, inevitable in some facilities. In one such study, it was found that of the 67 papers reviewed pertaining to order picking, only 4 pertained to sorting. This only accounts for approximately 3% of all reviewed papers pertaining to warehouse operations (Gu, Goetschalckx, and McGinnis 2007).

Upon determining a research gap existed in manually operated Post-Cs, the method for analyzing the impact of different staffing policies was considered. Given the ability to obtain first hand observations of a manually operated Post-C, and the desire to model an alternative staffing policy, simulation was the chosen approach for analysis.

Simulation is a great method for handling the variability within supply chains. It allows companies to observe the effectiveness of a Post-C within their supply chain without physically implementing it (Schunk and Plot 2000). One such source of variability is the customer demand, which has been shown to be a big source of uncertainty within supply chains (Yee 2002). The success of a model's prediction of future behavior depends on the system's modeling. Through the modeling and simulation of different

alternatives, a heightened understanding of the system is generated, and the best modeled alternative is found. (Venkateswaran and Son 2002).

With the research gap in manually operated Post-Cs found, it was determined to be beneficial to develop a simulation model of a manual Post-C facility for further analysis and study.

3 SIMULATION MODELING

The system being modeled includes the goods being transported within the supply chain and the interface with the workers, and with the Post-C. It begins with the arrival of semi-trailers containing bulk packaged goods, and it ends with the staging of goods now packaged with goods sharing its final destination. The system is modeled after an existing Post-C that supports perishable products. Figure 1 displays the layout of the facility.

There are two primary types of entities that flow within this system. The first acts as customized logic to control the creation parameters of inbound trailers, and the second is the worker, which performs all Post-C actions. In this system, units of product are bulk packaged in the form of stacks of baskets. The number of baskets contained in a stack is consistent, so only the quantity of stacks throughout the system and the quantity of product within a given stack are relevant. Each stack also has one of three (3) demand levels associated with it (i.e. high, medium, and low).

As trailers arrive to area 1 in Figure 1, the quantity of stacks at the inbound dock adjusts, accordingly. Workers move to the inbound dock to grab a stack, which they take to the bulk sorting zone (area 3).

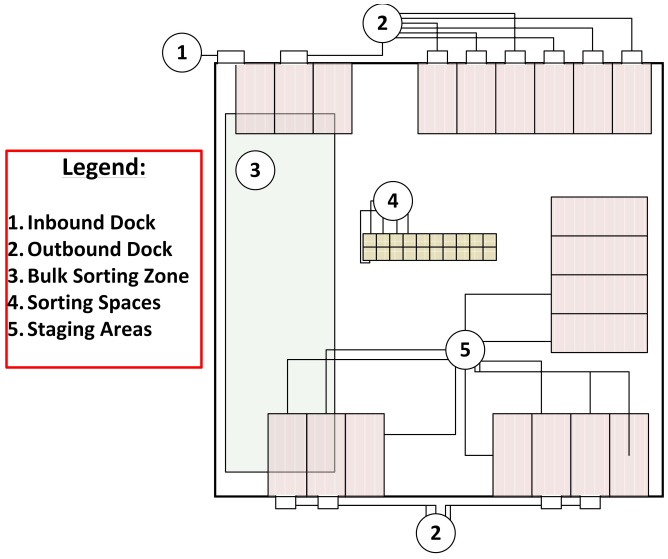

Figure 1: Layout of a post-distribution cross-dock system.

Workers take stacks placed in the bulk sorting zone and begin sorting them by their destination. To begin this, they move a stack over to the first sorting space (area 4) and place a quantity of product corresponding with how much product is needed to fulfill the destination's order. If product remains in their stack, they take the stack to the next sorting space and place product into it. This process is repeated until the stack no longer has product. Once this happens, the worker returns to the bulk sorting zone to retrieve another stack, and they continue the stack sorting process. However, they return to the space after their last sorting space, instead of the first sorting space. The entire sorting process is repeated until all destination orders are filled.

As workers sort product out, they are simultaneously breaking a stack down and building a new stack up. Once the new stack has reached the stack capacity for its product type, the completed stack is pushed

aside for staging so a new stack can begin. Once a stack is pushed aside, a worker takes it to the staging area (area 5) associated with the stack's destination.

All three of the aforementioned actions are carried out, in parallel, until all trailers have arrived and all product has reached the staging area of its destination. Figure 2 provides illustrations of the three actions.

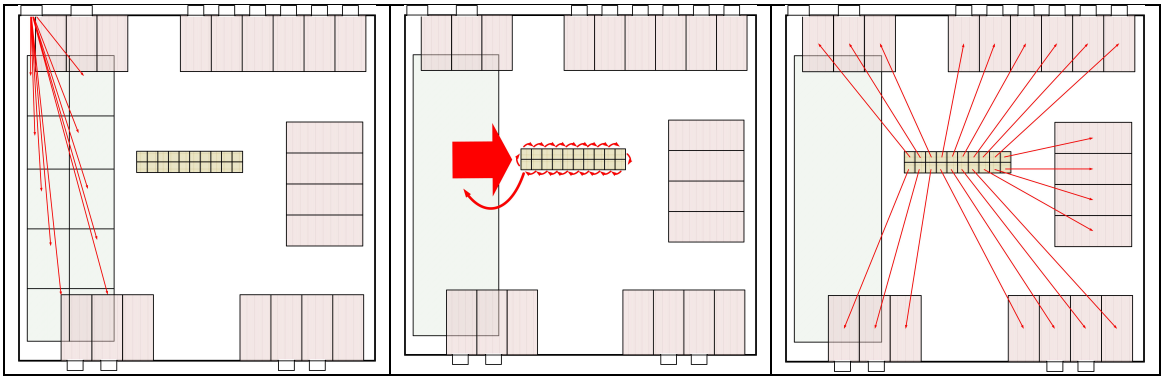

Figure 2: Sequence of Tasks: Bulk Sorting Placement, Sorting, Staging.

3.1 Conceptual Modeling

The Post-C has 17420 square feet of usable space and twelve dock doors. However, only one of these doors is used for receiving inbound trailers. The only permanent structure within the facility space is a put-to-light system, which occupies 320 square feet in the center of the facility floor. This put-to-light system has twenty (20) spaces for the twenty delivery routes the Post-C services. Each route has its own staging area marked off around the perimeter of the floor.

The bulk sorting zone is divided into ten (10) equal sized sub spaces. All sub spaces have the same capacity, but each sub space is associated with one of the three (3) levels of demand. The capacity of one sub space is sixteen (16) stacks. It is assumed that workers can freely walk through these sub spaces with negligible obstruction.

Trailers arrive at the Post-C based on a EXPO(.25) hr. distribution, with the first arrival occurring after EXPO(0.25) hr. The number of trailers scheduled to arrive for the day is based on a Triangular(#T – 1, #T, #T + 2) distribution, where #T represents the average number of trailers scheduled for arrival per day. In the base model, #T is equal to ten (10). Trailers will contain Triangular(20, 35, 60) stacks, and each stack has a 50% chance of being a high demand product, 35% chance of being a medium demand product, and 15% chance of being a low demand product.

Four (4) worker entities arrive at the inbound dock at the start of each replication, and each worker entity has a worker transporter allocated to it. This worker transporter moves at a velocity of Triangular(2.93, 4.4, 5.84) feet per second (Hobbs, Rossetti, and Faas 2006) when the worker is not carrying a stack, and he moves at Triangular(2.7, 4.17, 5.61) feet per second, which is 0.23 feet per second slower than the unobstructed velocity (Kong and Chua 2014).

When workers retrieve a stack from the bulk sorting zone, they must determine if product is new to the light sorting system. To model the frequency of a new product being entered into the system, the probability of the product being new is DISC(0.07, 1, 1.0, 0), where a value of 1 represents a new product. If the product is new, the worker will be delayed for TRIA(10, 15, 30) seconds. Once the worker arrives at a sorting space, there is a 15% chance the destination will not require any product from the stack being sorted. If the destination does, the worker will be delayed TRIA(5, 15, 30) seconds. Once the sorting space capacity is reached, the worker is delayed TRIA(7, 10, 15) seconds to move the stack out of

the sorting space, so it can be retrieved for staging. It is assumed there is no capacity on the number of stacks waiting for staging along the perimeter of the sorting spaces.

As workers complete actions, they search for more work to be done. At the end of any given action, all workers first look to the dock for stacks waiting to be placed in the bulk sorting zone. If there are none, workers will then look to the bulk sorting zone for stacks waiting to be sorted by destination. If there are none, workers will finally look to the sorting spaces for stacks waiting to be placed in their respective staging area. If a worker is still unable to find work, the worker will check if more trailers are scheduled for arrival, if no trailers are scheduled for arrival, the worker ends its shift. Each replication ends when all workers have ended their shift.

The primary difference in this base model and the alternative is the logic workers use to decide what to do next. The primary question raised in modeling was the impact on performance of dedicating workers to each action performed. In the alternative model, workers were divided into three (3) separate types. Type 1 workers were responsible for taking stacks from the dock to the bulk sorting zone. Type 2 workers were responsible for taking stacks from the bulk sorting zone and sorting out products by destination. Type 3 workers were responsible for moving completed stacks from their sorting space to their respective staging area.

If any worker of any type finds no stack to be moved or sorted, they waited for a stack to arrive in his area of responsibility before continuing. In the case of a worker finding no stack to be moved or sorted and no trailers scheduled for arrival, they would move to the next highest type (e.g. a type 1 worker finding no stacks to move into bulk sorting and no trailers set for arrival becomes a type 2 worker). Each replication ends when all trailers have arrived, and all stacks have been moved into their final staging area.

The primary benefit of specializing worker tasks is seen in the reduced effects of change-over. Change-over can be defined as any instance when a worker changes to a task different from the previously performed task. In the context of this Post-C, change-over is seen in cases such as when a worker arrives to a staging area with a stack and must walk over to the dock to place newly arrived stacks in the bulk sorting zone.

However, this benefit is seen at the sacrifice of overall worker utilization. For example, when type 1 workers are done moving stacks into the bulk sorting zone, they must wait for the next trailer to arrive, even though there is work to be done in sorting the stacks they just placed in the bulk sorting zone.

The effects of change-over are modeled through a factor that adjusts the time workers are delayed when placing product in a sorting space or moving a stack out of a sorting space for staging. This factor represents a worker's ability to improve their performance through repeated practice. For initial experimentation and comparison, it is assumed workers are delayed for only 90% of the normal delay to perform sorting tasks. All walking velocities are the same.

3.2 Arena Modeling

Arena Rockwell Software and discrete-event simulation were used to develop the two simulation models. This section discusses the key modelling aspects in each of the two models.

The quantities of stacks throughout the facility were modeled as variables and variable arrays. The worker entities would re-assign the values of these variables as actions were performed. Attributes respective of each worker entity also represented whether a worker was carrying a stack, what demand level the stack was, and the quantity of products contained in the stack. This modeling approach became most complex when stacks were taken through sorting.

During the sorting process, workers move to a bulk sorting sub space to retrieve a stack. Figure 3 displays the pseudo-code for the re-assignment of variables and attributes when a stack is retrieved from the bulk sorting zone for sorting.

```
ASSIGN
    myStackSize = DISC(0.6, 180, 0.8, 108, 1.0, 72)
    RandomOrder = ANINT(UNIF(1, v#Stops))
    DestinationOrders(RandomOrder) = DestinationOrders(RandomOrder) + myStackSize
```

Figure 3: Pseudo-code for picking up a stack from bulk sorting.

Through this re-assignment, the quantity of products in the stack (i.e. myStackSize) is determined, and the quantity needed to fill the order of a random destination is increased by this amount. Because we are only interested in the impacts of the staffing policy, we can afford to randomly determine the quantity of products in each stack, based on the distribution of stack sizes. Additionally, it is irrelevant when, or how the demand of each destination is determined. Rather, the act of sorting out products among a controlled number of destinations is what must be modeled. Therefore, the demand of each destination can be randomly determined, as well. Note that this does not assign all products within the stack to go to this destination. Instead, it ensures the exact total quantity delivered to the Post-C is equal to the quantity shipped out.

When the worker arrives at a sorting space with a stack, a quantity needed for the order is generated. Figure 4 displays the pseudo-code for ensuring this quantity does not exceed the remaining capacity of the sorting space, nor the remaining quantity of product in the stack being sorted. This ASSIGN module does so by randomly generating the number of products to be placed in the sorting space. This number of multiplied by a factor, depending on the demand level of the stack. Upon randomly generating this number, it compares the value generated with the remaining capacity of the sorting space and the remaining quantity of product in the stack being sorted. If the value generated exceeds the remaining product, but does not exceed the remaining capacity of the sorting space, it adjusts its value to equal the remaining product. If the value does not exceed either the remaining product nor the remaining capacity of the sorting space, it remains unchanged. If both the value and the remaining product exceeds the remaining capacity, then the value becomes the remaining capacity of the sorting space. Once any necessary changes are made to the generated value, the attribute representing the remaining product is decreased by the adjusted value, and both the total number sorted and the quantity of product in the sorting space is increased by the adjusted value.

```
ASSIGN
    #Picked = ANINT(TRIA(5, 20, 90)) * ((myStackDLvl == 1) * 2 + (myStackDLvl == 2) * 1.5 +
(myStackDLvl == 3) * 1)
    #Picked = (#Picked > myStackSize && myStackSize <= (SSCapacity(myNextSS) - #inSS(myNextSS))) *
            myStackSize + (#Picked <= myStackSize && #Picked <= (SSCapacity(myNextSS) -
            #inSS(myNextSS))) * #Picked + (#Picked > (SSCapacity(myNextSS) – #inSS(myNextSS)) &&
            myStackSize > (SSCapacity(myNextSS))) * (SSCapacity(myNextSS) - #inSS(myNextSS))
    myStackSize = myStackSize - #Picked
    TotalPicked(myNextSS) = TotalPicked(myNextSS) + #Picked
    #inSS(myNextSS) = #inSS(myNextSS) + #Picked
```

Figure 4: Pseudo-code for putting product in a sorting space.

Placing Stack in Bulk Sorting	**Sorting out a Stack of Products**	**Placing a Stack in a Staging Area**
SEARCH for a sub space at full capacity **If found:** **GO TO** the sub space, and sort out a stack of product **If not found:** **DECIDE** if stacks are at the dock **If True:** **GO TO** the dock and retrieve a stack **If False:** **SEARCH** for a sub space with a stack **If found:** **GO TO** sub space, and sort out a stack of product **If not found:** **SEARCH** for a sorting space with a stack ready to be staged **If found:** **GO TO** the sorting space and stage the stack **If not found:** **DECIDE** if all trailers have arrived **If true:** **DISPOSE** **If false:** **HOLD** for next trailer to arrive	**SEARCH** for a sub space with a stack **If found:** **GO TO** sub space, and sort out a stack of product **If not found:** **DECIDE** if stacks are at the dock **If True:** **GO TO** the dock and retrieve a stack **If False:** **SEARCH** for a sorting space with a stack ready to be staged **If found:** **GO TO** the sorting space and stage the stack **If not found:** **DECIDE** if all trailers have arrived **If true:** **DISPOSE** **If false:** **HOLD** for next trailer to arrive	**DECIDE** if stacks are at the dock **If True:** **GO TO** the dock and retrieve a stack **If False:** **SEARCH** for a sub space with a stack **If found:** **GO TO** sub space, and sort out a stack of product **If not found:** **SEARCH** for a sorting space with a stack ready to be staged **If found:** **GO TO** the sorting space and stage the stack **If not found:** **DECIDE** if all trailers have arrived **If true:** **DISPOSE** **If false:** **HOLD** for next trailer to arrive

Figure 5: Pseudo-code for determining next action after completing each action.

The priority workers give to the three actions changes depending upon which action they have completed. Figure 5 shows the comparison between the logic worker entities undergo after completing each of the three actions. The task prioritization modelling approach presented in Figure 5 has a number of interesting points. In the first case, the worker ensures the sub space never goes over capacity by immediately addressing any sub space at capacity. In the second case, the worker attempts to continue performing the action he just completed, but in the last case, the worker attempts to move a stack from the dock into the facility. This is due to the completion of the first two actions allowing the next action to take place, while the completion of the last action has no influence on the ability to perform the first two actions. Therefore, the workers attempt to help with the first action, first. After the first, special attempt is made by workers in the first two cases, the workers then search for an action to complete as early in the product flow process as possible. If no work is to be done and all scheduled trailers have arrived, the worker moves to dispose itself, and end its shift.

Lastly, the controls and responses of the two models are obtained through the use of several constructs. ASSIGN modules, attributes, variables, and RECORD modules are used to record the total time taken to perform each action, as well as the total time taken to unload, sort, and stage all stacks. Throughout the logic displayed in Figure 5, DECIDE modules determine if a new action has begun, or if an old action has come to an end. If the condition is met, the worker passes through an ASSIGN module to set the variables used to report the time stamp of each action's beginning and end. These time stamps are what are used to compute the total time taken to perform each action. The worker hours of the entire staff is then computed using these values and the number of workers, for each type. For example, the type

3 worker hours needed to run the alternative model is the number of type 3 workers multiplied by the total time taken to stage the stacks. Figure 6 illustrates the time line of when processes start and stop.

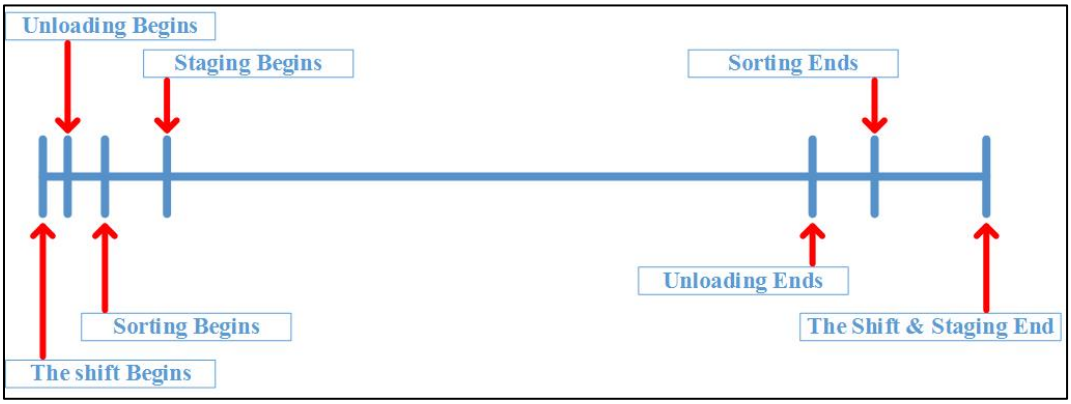

Figure 6: A timeline of when the three actions start and stop throughout a shift.

Several key performance measures were collected from each model to evaluate their performance. These performance measures can be seen below.

- Total time to complete all actions
- Time to complete each action
- Average number of busy workers
    - For the dedicated worker model, average number of busy workers by type
    - For global worker model, average number of workers busy by action
- Average time a worker is busy
    - For the dedicated worker model, average time a worker is busy by type
    - For global worker model, average time a worker is busy by action

In the context of this problem, workers are considered busy if they are moving toward a stack to retrieve, or moving/sorting a stack. Therefore, the only time workers are not busy is when they are waiting for stacks to move/sort.

4 OPTIMIZATION MODELING

Optimization models were created to answer the following questions: what is the minimum total number of workers needed to complete all actions within a certain time frame (model 1); what is the shortest amount of time all actions can be completed with a given number of workers (model 2); what is the minimum total number of workers needed to complete all actions within a certain time frame, while constraining the amount of time each action can take (model 3); what is the shortest amount of time all actions can be completed with a given number of workers, while constraining the amount of time each action can take (model 4). The models are shown in Tables 1 and 2.

- Model 1: Program solved to minimize the number of workers needed to complete all actions.
- Model 2: Program solved to minimize the total time needed to complete all actions.
- Model 3: Program solved to minimize the number of workers needed to complete all actions, while constraining the amount of time each action can take and the total time to complete all actions.
- Model 4: Program solved to minimize the total time needed to complete all actions, while constraining the amount of time each action can take and the total number of workers.

For both the global and dedicated worker models, the only decision variable was the number of workers available. Due to the three different worker types, the number of workers available is a set of variables for the dedicated worker model.

Using these optimization models, analysts can make more informed decisions concerning the needed number of workers (in total and by type) with which to staff their Post-Cs, while under time constraints, as well as the optimal throughput, while under staffing constraints.

The mathematical programs used to answer each of the four questions can be seen in Tables 1 and 2. In the following models, W_i represents the number of workers of type i, which is equal to the total number of workers in the global worker model, $W_{allowed}$ represents the limited number of workers the system is constrained to, T_{total} represents the total time taken to complete all actions, and $T_{allowed\ total}$ represents the limited amount of time T_{total} is constrained to.

Table 1: Models for Global Workers.

Model 1	Model 2
$Minimize \quad \sum_{i=1}^{3} W_i$ $s.t. \quad T_{total} \leq T_{allowed\ total}$ $\{W_i \in \mathbb{N} \mid W_i > 0\}$	$Minimize \quad T_{total}$ $s.t. \quad \sum_{i=1}^{3} W_i \leq W_{allowed}$ $\{W_i \in \mathbb{N} \mid W_i > 0\}$

Table 2: Models for Dedicated Workers.

Model 3	Model 4
$Minimize \quad \sum_{i=1}^{3} W_i$ $s.t. \quad T_{total} \leq T_{allowed\ total}$ $T_i \leq T_{allowed\ i} \quad for\ i = 1,2,3$ $\{W_i \in \mathbb{N} \mid W_i > 0\}$	$Minimize \quad T_{total}$ $s.t. \quad \sum_{i=1}^{3} W_i \leq W_{allowed}$ $T_i \leq T_{allowed\ i} \quad for\ i = 1,2,3$ $\{W_i \in \mathbb{N} \mid W_i > 0\}$

Using OptQuest in Arena's Visual Designer tool, the optimization models were investigated. This tool finds the optimal value of each decision variable in the model through experimentation. The user is able to set bounds for each decision variable and a number of replications for each experiment. The tool then runs the model using suggested values for each decision variable. It adjusts the decision variables with each passing test and uses the average value of the objective function to evaluate the results.

At the end of its run, the tool provides the user with a table of the best simulations with their respective decision variable values and objective function value. Using this tool, the models were solved, and experiments were conducted on the effects of adjusting other factors.

5 DATA COLLECTION AND MODELS VALIDATION

The methods used to collect data and validate the two models are presented in this section. For the purposes of this paper, reasonable ranges of data were obtained. The following parameters were obtained through first hand observations of a Post-C: the arrival rate of trailers; the probability distribution function for the number of stacks on each trailer; the probability distribution function for the number of trailers scheduled for arrival; the probability distribution function for the number of products contained within each stack; the distances between locations; the number of destinations serviced; the number of workers; the probability distribution function for the number of products placed in each sorting space; the probability distribution function for the frequency of new products being sorted; the probability distribution function for destinations not requiring product from a stack; the delays for entering a new product into the put-to-light system, placing product in a sorting space, and moving a product out of a

sorting space. Assumptions were made for the data that was missing, such as an unlimited capacity of stacks waiting to be staged along the perimeter of the sorting spaces. Statistical analysis on the validity of these assumptions is out of the scope of this paper.

6 EXPERIMENTATION AND RESULTS

The experimentation for this study consisted of comparing the dedicated worker model to the global worker model. This comparison looked to the difference in performance measures under current conditions, and how the difference changed as certain conditions changed. The optimization models were also solved under current conditions. In the context of this problem, current conditions constitute all parameters outlined in Section 3.

Under current conditions, the global worker model was able to complete all actions in 320.19 minutes. The average number of workers unloading stacks was 0.4515, the average number of workers sorting stacks was 2.4943, and the average number of workers staging stacks was 0.8208. Although it took 253.29 minutes to complete all unloading, only 36.0023 minutes was spent with a worker busy unloading. Of the 241.3 minutes it took to complete all sorting, a worker was busy sorting for 199.33 minutes, and of the 148.24 minutes it took to complete all staging, a worker was busy staging for 61.7529 minutes.

Before being able to compare the dedicated worker model to the global worker model, the optimal worker arrangement had to be found. Using the second model outlined in Section 4, it was found that beginning with one type 1 worker and three type 2 workers yielded a total time of 314.58 minutes. The average number of workers unloading was 0.5505, the average number of workers sorting was 2.1938, and the average number of workers staging was 0.8326. Of the 228.15 minutes it took to unload all product, type 1 workers were only busy for 170.68 minutes. Type 2 workers were only busy for 212.91 minutes of the 238.24 minutes it took to sort all product, and type 3 workers were busy for all 67.3445 minutes of the 67.3445 minutes it took to stage all product. All four workers were only busy for 279.48 minutes of the 314.58 minutes it took to complete all actions.

The first optimization model was then solved by constraining the total time to a standard shift of 8 hours, or 480 minutes. Under this constraint, and removing the worker quantity constraint, it was found that the global worker model could complete all actions in 398.13 minutes with only 3 workers. The dedicated worker model could complete all actions with only 3 workers, but it took 423.42 minutes to complete, which indicates the dedicated worker model will not always outperform the global worker model under the same constraints.

To further investigate this finding, the third optimization model was solved under two different sets of constraints. The first set represents a limited unloading window where workers only have 200 minutes to complete all unloading before the dock is closed. The second set represents a limited sorting window where workers only have 200 minutes to complete all sorting. No experiment was done to investigate the effects of limiting the time to stage, as this is identical to constraining the time to complete all actions. Table 3 shows the result of the these two solutions for both models.

As shown in Table 3, the dedicated worker model is far more conducive for systems placed under action based time constraints. To further illustrate this point, the models were optimized using the fourth optimization model, and the dedicated model outperformed the global worker model, once again. This is shown in Table 4.

In another comparison of the two models, a sensitivity analysis was performed on the parameter of the delay between trailer arrivals. The relationship between this delay and the total time to complete all actions was compared between the two models. The relationship between the delay and the average number of busy workers was also compared. This analysis was conducted under current conditions, and can be seen in Figure 7.

Table 3: Minimum workers needed for each worker model, while constrained to 480 minutes total.

Constraint	Workers Needed for Global Model	Workers Needed for Dedicated Model
Time to Unload ≤ 200 minutes	7	5
Time to Sort ≤ 220 minutes	7	6

Table 4: Minimum total time needed for each worker model, while constrained to 7 workers.

Constraint	Total Time of Global Model	Total Time of Dedicated Model
Time to Unload ≤ 200 minutes	252.13	223.96
Time to Sort ≤ 220 minutes	252.13	223.96

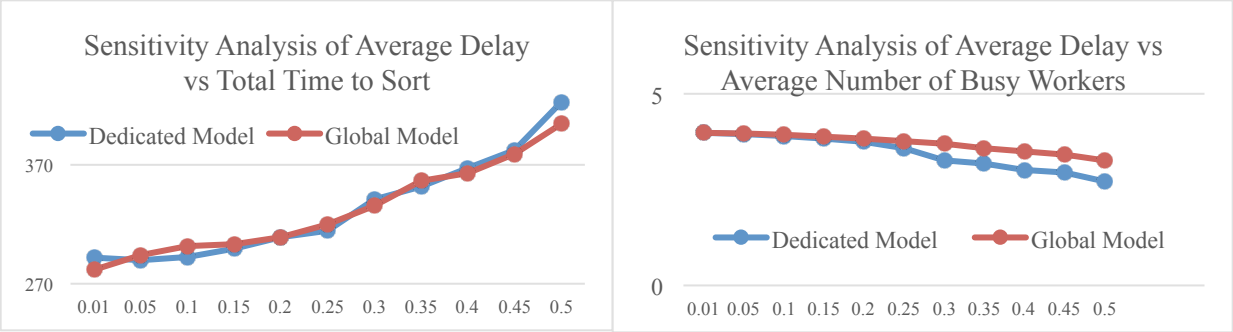

Figure 7: Results of a sensitivity analysis relating the parameter of the delay distribution with the total time to complete all actions and the average number of busy workers when deploying four workers.

As one can see in Figure 7, the dedicated worker model is capable of performing similarly to the global worker model. However, the average number of workers busy in the dedicated worker model steadily decreases at rate higher than that of the global worker model.

7 CONCLUSIONS AND FUTURE WORK

With the research gap in manually operated Post-Cs, the field is ripe with opportunity to provide new insight on its inner workings and most significant factors. Several papers have addressed other warehousing issues and cross-docking problems, but very few have addressed the needs of facilities performing intermediate sorting. From this analysis we have seen that the dedicated worker model suffers from lower worker utilization. However, the benefits it experiences from reduced change-over allow it to perform similarly to the global worker model.

This work leaves room for significant expansion and development. For example, future work may address the layout of the facility and the distances separating each action. Other future work may investigate the effects of various demand levels and bulk sorting space partitions. However, it can be concluded that the dedicated worker model performs comparably to the global worker model.

REFERENCES

Fanti M. P., G. Stecco, and W. Ukovich. 2016. "Scheduling Internal Operations in Post-Distribution Cross Docking Systems". *IEEE Transactions on Automation Science and Engineering*. 13:296-297.

Gu J., M. Goetschalckx, and L. F. McGinnis. 2007. "Research on warehouse operation: A comprehensive review". *European Journal of Operational Research*. 177:1-21.

Hobbs B., M. D. Rossetti, and P. Faas. 2006. "An Object-oriented Framework for Simulating Automatic Data Collection Systems". In *Proceedings of the 2006 Winter Simulation Conference*, edited by L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1545-1553. Piscataway, New Jersey: Institute of Electrical and Electronic Engineers,Inc.

Kong P. W. and Y. K. Chua. 2014. "Start-Up Time and Walking Speed in Older Adults under Loaded Conditions during Simulated Road Crossing". *Experimental Aging Research*. 40:593-594.

Magableh G., M. Rossetti, and S. Mason. 2005. "Modeling and Analysis of a Generic Cross-docking Facility". In *Proceedings of the 2005 Winter Simulation Conference*, edited by M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, 1613-1620, Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Schunk D. and B. Plot. 2000. "Using Simulation to Analyze Supply Chains". In *Proceedings of the 2000 Winter Simulation Conference,* edited by J. A. Joines, R. R. Barton, K. Kang and P. A. Fishwick, 1211-1216. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Venkateswaran J. and Y. J. Son. 2002. "Investigation of Influence of Modeling Fidelities on Supply Chain Dynamics". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucesan, C. H. Cehn, J. L. Snowdon, and J. M. Charnes, 1183-1191. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Vogt J. J. 2010. "The Successful Cross-docking Based Supply Chain". *Journal of Business Logistics* 31(1):99-119

Yan H. and S. L. Tang. 2009. "Pre-distribution and Post-distribution Cross-docking Operations". *Transportation Research Part E*. 45:843-844.

Yee S. T. 2002. "Establishment of Product Offering and Production Leveling Principles via Supply Chain Simulation under Order-To-Delivery Environment". In *Proceedings of the 2002 Winter Simulation Conference*, edited by E. Yucesan, C. H. Cehn, J. L. Snowdon, and J. M. Charnes, 1260-1268. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

AUTHOR BIOGRAPHIES

DAVID COX is an undergraduate student at the University of Arkansas pursuing his B. S. in Industrial Engineer. He will obtain his bachelor's degree in the Spring of 2017. He plans to begin working as a business consultant at Applied Predictive Technologies in July of 2017. His research interests involve simulation modeling, transportation logistics, and supply chain facility staffing policies. David can be reached at d.a.cox1964@gmail.com.

MANUEL D. ROSSETTI is a Professor in the Industrial Engineering Department at the University of Arkansas. He received his Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research and teaching interests involve simulation modeling, logistics optimization, and inventory analysis applied to manufacturing, distribution, and health-care systems. He serves as an Associate Editor for the International Journal of Modeling and Simulation and is active in IIE, INFORMS, and ASEE. He served as co-editor for the WSC 2004 and 2009 conference, the Publicity Chair for the WSC 2013 Conference, and was the 2015 WSC Program Chair. Dr. Rossetti is the author of Simulation Modeling and Arena by Wiley, now in its 2nd edition. He can be contacted at rossetti@uark.edu and http://www.uark.edu/~rossetti/.