Final Report

Cross-docking Operations

"Anticipating Labor and Processing Needs of Cross-dock Operations"

December 28, 2012

Frederick Abiprabowo

Table of Contents

1. Introduction

In the supply chain literature, there are generally three kinds of studies regarding cross-docking: fundamentals of cross-dock, distribution planning, and operations inside the facility. Studies of cross-dock fundamentals take a high-level perspective and discuss the issues of cross-docking in relations to a company's distribution process, management, etc. Apte and Visnawathan [1] presented how cross-docking can improve distribution efficiency while also maintaining customer satisfaction level. Cross-docking can also be viewed as a method in the management of a supply chain [2]. Distribution planning problems pertain to scheduling of trucks, vehicle routing, and network navigation. Kailei Song [3] formulated the scheduling problem as an optimization problem minimizing the time difference between the start and finish of the sequence of jobs. Min Wen et al [4] addressed the variation of the Vehicle Routing Problem known as Vehicle Routing Problem with Cross-docking. Finally, there are many research papers regarding the operations inside cross-docks. The Cross-docking door assignment problem comes into play in finding truck assignments that minimize the distances traveled by forklifts, which can be solved by the formulation of Generalized Quadratic 3-dimensional Assignment Problem [5,6]. Kevin Gue [7] compares different protocols of items staging and proposes pros and cons of each. Zhengping Li [8] designed a product ranking formula that can be used for the sorting process.

Due to high carrying costs of inventory, it is often not possible to stock all the items in a warehouse distribution center. Sending trucks directly from the suppliers to the customers would eliminate the need of warehousing, though this would result in multiple deliveries with substantial transportation costs. To tackle these problems, most suppliers implement cross-

docking facilities into their operations strategy due to the fact that cross-docking consolidates transportation while reduces inventory at the same time. Most cross-docks are in the shape of an I—though other exotic shapes such as E, H, or U do exist [9]—and consist of a set of strip doors on one side and stack doors on the other. Incoming trucks from suppliers are assigned to strip doors for the unloading process. On the other side of the cross-dock, there are usually empty trailers sitting on the stack doors ready to be filled with items.

Zongze's ExtendSim CDAP simulation report dealt with the operations inside the cross-docking facility and discussed the formulation of goods movement inside a cross-dock as a GQ3AP. Then, a simulation model was designed after one of the cross-docking facilities of a retail company in New Jersey. This model was then tested to compare the cost of goods movement in two different scenarios: GQ3AP-optimized door assignment and manual door assignment. Based on the results of this test, the scenario with optimized door assignment did yield a lower cost than the one without.

This report continues where the previous report left off and addressed its proposed future plans for the model. Below are the future plans as enumerated in the previous report (in quotation marks), with additional comments after each item.

A. "Refine Model"
   1. "A cross-dock in the real life has far more doors than we build (4-by-4) here. We should make it similar to the real life one."
      - The model has been expanded and now is of the size 120-by-120.
   2. "In a cross-dock, the customer truck at the stack door should have a certain capacity."

- This feature has been added to the current model with the assumed capacity of 2800 cubic feet. Outgoing trucks can only leave the cross-dock when this capacity is reached.

3. "In the future work there may exist dynamic batching and dynamic scheduling."

- The batching process is now done by the items' destination and each batch can contain up to 80 cubic feet of items. The schedule of truck arrival is assumed to be random throughout the day, as no additional information regarding the schedule was made available.

B. "Expand Model"

- This was addressed in Part A point 1.

C. "Experiment with Typical Realistic Simulation"

- The data used in the current simulation is based on a real, 1-day worth of data given by the said retail company.

In the previous model (see Progress Report #3), items and trucks were created randomly according to a specific set of constraints, as well as the given random distributions. Although this method of creation was useful for experimental purposes, two problems were evident in the model. Firstly, the obtained result was not realistic enough, i.e., the variation of the items sent to the destinations among multiple simulation days was not deemed as significant. This was expected since the creation process was heavily dependent on the constraints and random distributions, and they remained unchanged between multiple runs. Secondly, the model had very limited user interface capabilities. This was a concern as this research is aimed to possibly aide cross-docking floor managers and other occupations alike.

With those in mind, improvements and adjustments were made to the model.

The changes since Progress Report #3 include:

- size expansion from 16x16 to 120x120,

- data import from Excel worksheets,

- item and truck creation based on the imported demand data,

- use of ExtendSim database to manipulate data,

- items staging before being loaded to the outgoing trailers, and

- a different logic for items palletization.

This report describes how simulation can be used as a design and validation tool for cross-docking facilities. A step-by-step walkthrough of the model, testing parameters, and its results are included. Finally, some ideas for the future model are proposed.

Note: snapshots of diagrams are included in this report just for clarity. Since the layout of some H-Block/Excel worksheet is normally very large, the diagrams here are only small, cut-off fragments that constitute a bigger one. Thus, some blocks, words and connections in each diagram may seem incomplete or disconnected.

2. Step-by-step Walkthrough

2.1. Importing demand data from an Excel file

	A	B	C	D	E	F	G
1							
2							
3			1600	0	0	0	0
4			1000	0	0	0	0
5			0	0	0	0	1000
6			0	0	0	0	0
7			0	0	0	875	0
8			0	0	0	0	0

Fig. 1: Items schedule sheet

	A	B	C	D
1			Demand	
2				
3	1		1	
4	2		1	
5	3		1	
6	4		1	
7	5		1	
8	6		1	

Fig. 2: Demand satisfiability

Before the simulation can run, an Excel file containing the demand schedule for the day has to be present. The size of the table is 50x50 (50 origins and 50 destinations) and the data points are presented with origins as the X axis and destinations as the Y axis. The demand values shown on this sheet are in cubic feet, which will be represented as the volume property in the simulation. This will be explained in more details later. In addition, there is also a sheet that keeps track whether the demand (X axis) for a particular destination (Y axis) is satisfied. At the start of the simulation, these values are set to 1 (destination still waits for items). Later, when all the demanded items of the destination have been sent, the demand value will change to -1 (do not send items to this destination anymore). These sheets are then imported to the model as the ExtendSim internal database.

Record #	1	2	3	Field 4	Field 5
1	1600	0	0	0.00	0.00
2	1000	0	0	0.00	0.00
3	0	0	0	0.00	1000.00
4	0	0	0	0.00	0.00
5	0	0	0	875.00	0.00
6	0	0	0	0.00	0.00
7	0	0	0	0.00	1000.00
8	0	0	0	0.00	0.00
9	0	0	0	0.00	0.00
10	0	0	450	0.00	0.00
11	750	0	0	0.00	0.00
12	0	0	0	0.00	0.00

Record #	Field 1
1	1.00
2	1.00
3	1.00
4	1.00
5	1.00
6	1.00
7	1.00
8	1.00
9	1.00
10	1.00
11	1.00
12	1.00

Fig. 3 and 4: The Excel sheets after being imported as ExtendSim DB

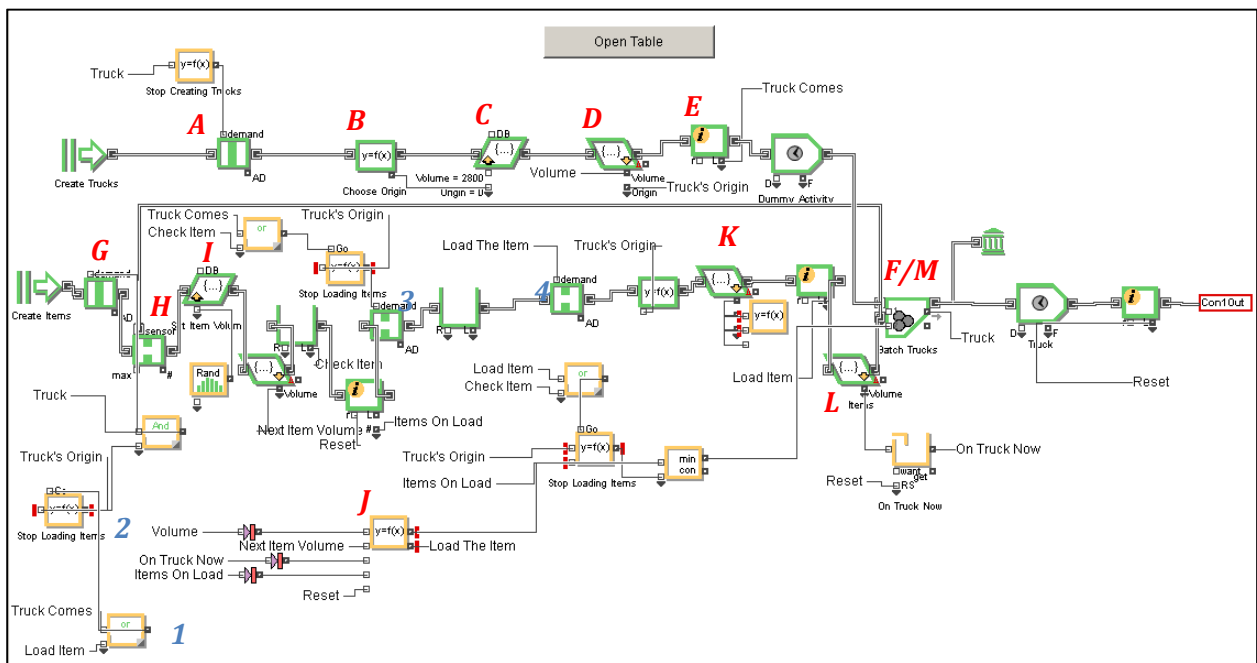## 2.2. Creating items and trucks according to the database

Fig. 5: "Create Trucks and Items" H-Block

The red alphabets shown above represent the important points of the creation process, which are ordered sequentially. The explanations provided below correspond to these letters. In addition,

the numbers in blue on the diagram facilitate the reader in locating the blocks in interest, though they do not represent an additional step in the process.

A. Trucks are created infinitely, but the create block is directly linked to a demand gate. An equation block is feed into the demand gate. The logical table representation of the block is:

Truck variable	Check demand on *Figure 4*	Demand gate
1 (a truck is present)	-	0 (close)
0 (not present)	1 (at least one destination still has demand)	1 (open
0 (not present)	-1 (no more demand)	1 (open)

Fig. 6: Logical table for point B

B. An item equation block chooses a random destination on *Figure 4* whose value is 1.

C. The truck is assigned a volume. In this case, the value is the constant 2800.

D. The volume and origin of the truck are stored in the variables *Volume* and *Truck's Origin*, respectively.

E. This information block is used for counting how many trucks are created for that day. Also, the "passing" of the truck is stored in the variable *Truck Comes*.

F. The truck goes to the batch block. The presence of the truck in this block is stored in the variable *Truck*.

G. Like trucks, items are created infinitely and the create block is linked to a demand gate as well, but with a much stricter set of constraints. Let's start from the very outer part of the

blocks that feed into the demand gate: the "Or" block (*number 1*). Since this block goes

to the "Go" input of the "Stop Loading Items" block (*number 2*), the latter block

calculates its equation only if one of the former block's inputs is detected. The equation

itself determines if there is no more demand for the given truck's destination, i.e., there is

no combination of X (the destination of the item to be created, ranging from 1-50) and a

fixed Y (the given truck's origin) such that the value for any point (X, Y) is greater than 0

on *Figure 3*. The values on *Figure 3* can indeed be all less than or equal to 0 within a

given row (the exact case when a particular origin has fulfilled all its shipments), as they

will later be decremented by volume of each item created for the truck.

H. Area gating from this point up to *F*; this ensures that only one item may be present in this

area at a time.

I. The item is assigned a volume, which can be 1, 2, or 6. These values are all equally likely.

J. For simplicity, this part also includes all the related value blocks that are connected to

block explained in part *F*. The main purpose for these blocks is to check whether the

incoming item should be loaded into the current truck. The resulting output is the

batching size and it is the minimum of two disparate values depending on different

scenarios: origin demand is satisfied or truck capacity is filled. The former checks the

demand table for the given origin and outputs the current batching size if all the demand

is satisfied. Otherwise, it outputs an arbitrary large number. The corresponding demand

gate that opens the way for items to go through is block *number 3*. On the other hand, the

latter keeps track of the current items volume in the truck and outputs the current

batching size if the truck's capacity is exceeded should it load the next item. Similarly, it

outputs an arbitrary large number otherwise. The corresponding demand gate is block *number 4*.

K. This block carries out the database updating process for each item that passes through. For example, there is an item with volume 2 and destination 1 for truck with origin 3 and the current demand value in the corresponding coordinate (1, 3) is 1600. Then this process decrements said value is decremented by 2, that is, it becomes $1600 - 2 = 1598$. Moreover, if all the demand is satisfied for a given origin, then this process will also change the origin's value on *Figure 4* to -1.

L. This block keeps track of the total volume of items in the truck.

M. The item proceeds to the batch block. The truck and items batch together and leave the block if the origin's demand is satisfied or truck reaches its capacity.

In other words, the items and truck process can be explained as follows. First, a truck is created with a volume of 2800 and a random origin (from 1 to 50) whose demand is still 1 on *Figure 4*; a truck would never be created for an origin whose demand is already -1 (satisfied). Now, the items may be created. The destination of the item is obtained from the X axis on *Figure 3*—with the given truck's origin as the Y axis—for which the value of the coordinate is larger than 0. Afterwards, the item is assigned a volume (equally probable between 1, 2, or 6). The item may be loaded to the truck if one of these 2 cases holds: demand is not satisfied or truck has not reached its capacity. Then, this item creation process continues until one of the aforementioned cases is violated. At this point, the truck (filled with items) leaves the H-Block and another truck is created. Truck creation repeats until the demand for all destinations is satisfied, meaning that all values on *Figure 4* are -1s.
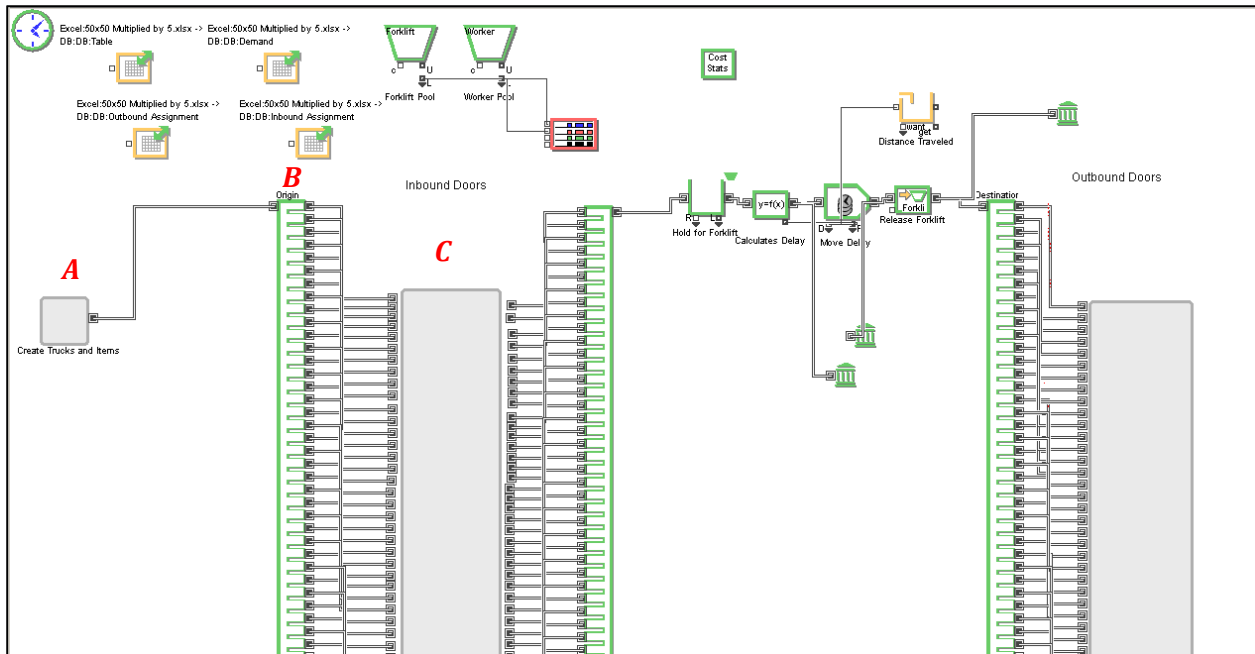
2.3. Arriving in the cross-dock

Fig. 6: Model Overview

A. This "Create Trucks and Items" H-block outputs trucks filled with items.

B. Trucks then proceed to their respective inbound doors, which can be randomly assigned or predetermined.

C. An H-block of inbound doors. There are 120 inbound doors in the current model. The unloading process of trucks will be explained more in detail in the next section.
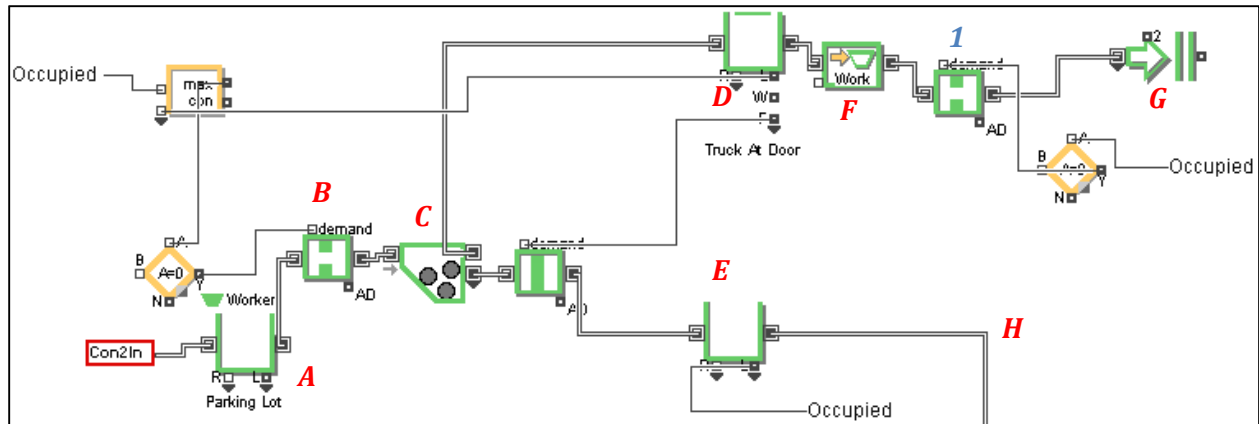
2.4. Unloading items

Fig. 7: Inbound Door Part 1

A. An adequate amount of resources—in this case, workers—is required to unload a given truck. Otherwise, the truck has to wait in the queue block until resource is available.

B. Demand gate closes if there is a truck that is being unloaded at the given moment; it opens otherwise.

C. The truck is unbatched and it is ready for the unloading process.

D. The truck waits until all its items are unloaded, and only until then the following demand gate (*number 1*) opens. Also, only 1 truck can be present and be unloaded at a given door at a time.

E. Items are unloaded one by one and are sorted by destination in order to facilitate the palletizing process later.

F. After all the items are unloaded, demand gate (*number 1*) opens and the used resources are released.

G. The truck exits the simulation.

H. The items follow the path to the palletizing process.
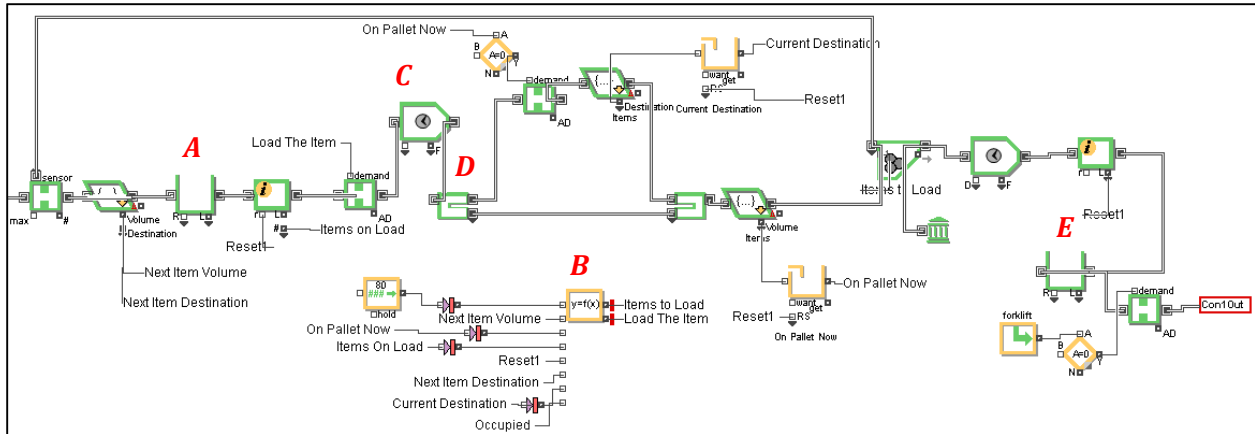
2.5.   Batching into pallets

Fig. 7: Inbound Door Part 2

A. Since each pallet has to contain items going to the same destination, this queue block sorts the items by destination in order to facilitate the palletizing process later on.

B. The accompanying equation blocks here are similar to the ones used to fill the trucks depicted in *part J section 2*. That said, two adjustments were made. The first one is that the capacity is now 80 cubic feet, which is the assumed maximum capacity for each pallet. The second one is the introduction of the variable "Current Destination," which tracks the current destination of the items being palletized. Following from the previous step that the items are sorted by destination, the change of Current Destination signifies that there are no more items going to that particular destination. Therefore, the pallet shall proceed even though it has not reached its capacity yet.

C. A delay is applied to every item and is proportional to the item's volume, i.e., the larger the item, the longer the delay will be.

D. There are two routes in this Select Item Out block. The upper route opens when the number of items in the current pallet is exactly zero. The destination of the first item that

goes through the very route is recorded in a Holding Tank as Current Destination. At this

moment, the variable Current Destination is already known, the item in the pallet is not

zero, and thus the upper route closes. The subsequent items (with the same destination)

then go through the lower route and fill the pallet. As mentioned in *part B* above, the

pallet proceeds given that either one of these conditions is met: the pallet's capacity is

reached or there are no more items going to the destination recorded in Current

Destination.

E. Pallets wait in this queue block until there are enough resources (in this case, forklifts). If

so, they are ready to be moved across the floor.
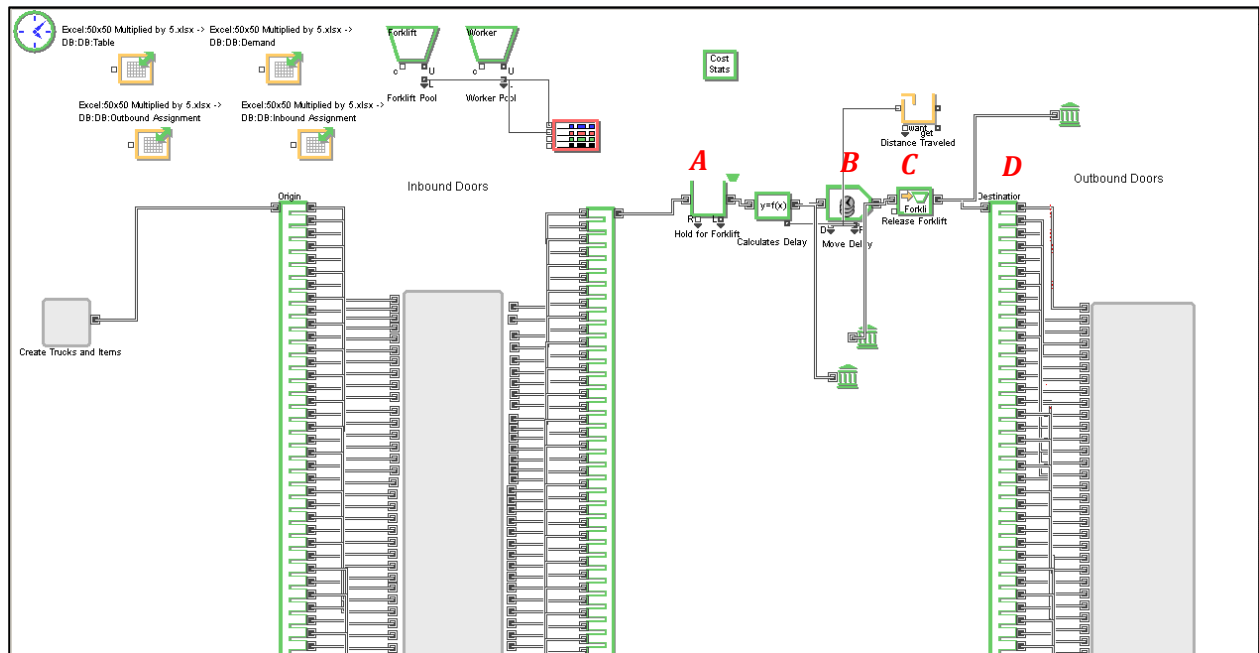
2.6.   Moving pallets across the floor

Fig. 8: Model Overview revisited

Figure 6 from *section* 3 is reproduced above for your convenience.

A. The pallet has to wait until at least a forklift is available.

B. A delay is applied to each pallet, which is calculated as the time required for the forklifts to move from the inbound door to the outbound door using the Manhattan distance.

C. The forklift resource is released back to the resource pool.

D. The pallet now is distributed to its corresponding outbound door.

2.7. Staging the pallets
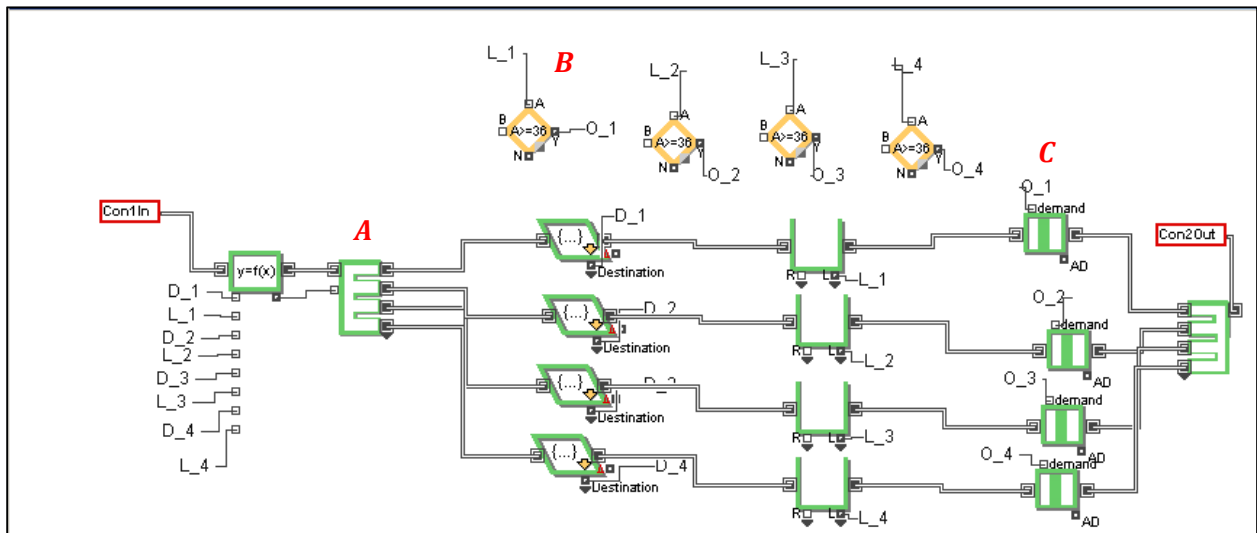
Fig. 9: Staging

A. Currently each inbound door can hold staged items going to 4 different destinations. Each pallet is staged together with other pallets of the same destination.

B. The staged pallets are released if there are 36 pallets of the same destination in order to ensure that they altogether have 3000 cubic feet of volume.

C. Once the condition in *B* is satisfied, this demand gate opens and lets the items proceed to the loading process.

2.8. Loading items
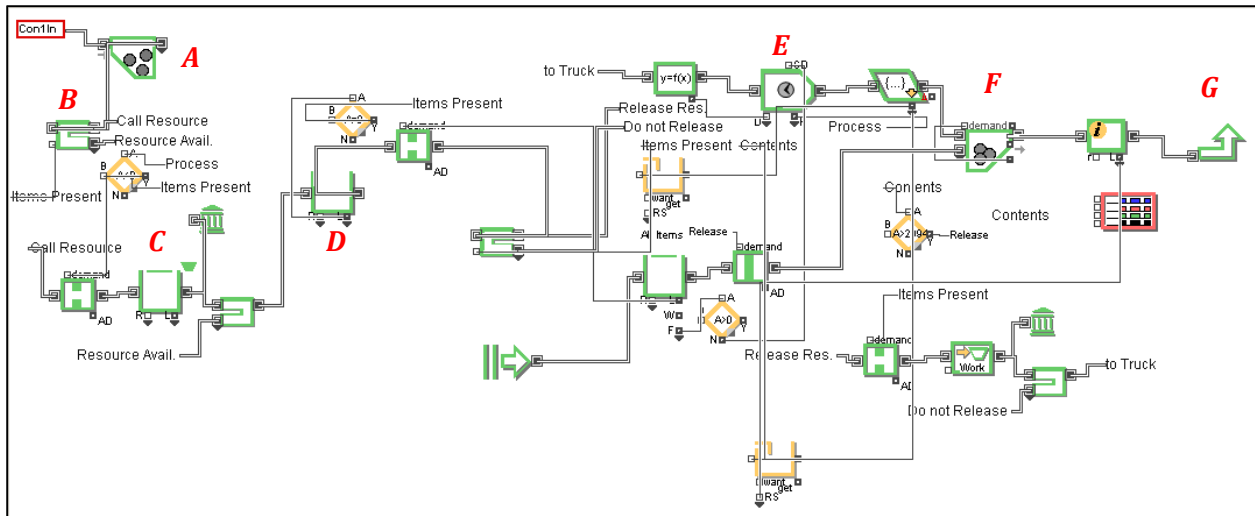
Fig. 9: Outbound door

A. Each pallet is unbatched back into items.

B. If no worker is working on the loading process, the item proceeds to the Call Resource route, which will then in turn call a worker. Otherwise, the item proceeds to the Resource Available route, which will add the item to the queue of items the worker is working on.

C. The Call Resource route waits until at least a worker is available.

D. The items are ready to be loaded to the truck as long as there is a truck on the outbound door.

E. A delay is applied to every item, which is directly proportional to the item's volume.

F. Once the truck has hit its capacity, i.e., 3000 cubic feet, the truck and its items are batched together.

G. The truck exits the facility.

2.9. Calculating item cycle time
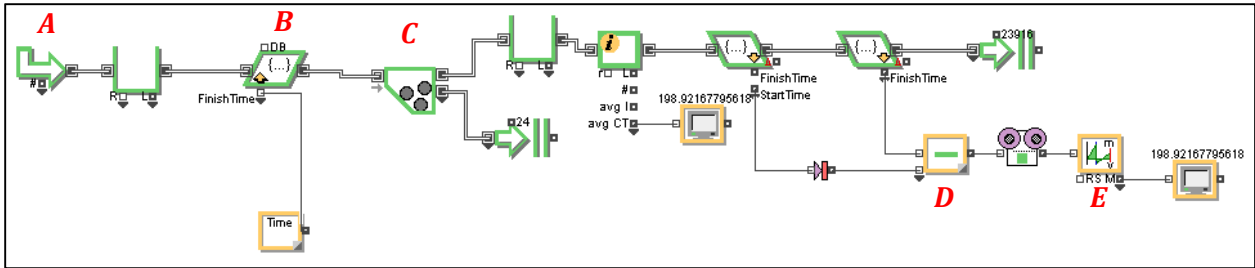
Fig. 10: Cycle item calculation

A. Each truck exiting the simulation is caught by this Catch block.

B. A variable called FinishTime is assigned to the truck and all the items contained in it.

C. The truck is unbatched to items and an empty truck, the former of which proceeds to the next steps while the latter exits the simulation.

D. The cycle time of each item is calculated by taking the difference between the variables FinishTime and the StartTime.

E. This block takes and records the average of cycle times of all the items exiting the facility.

3. Testing the Model: Input Parameters

3.1. Origin-Destination Matrix

	A	B	C	D	E	F	G
1							
2							
3			1600	0	0	0	0
4			1000	0	0	0	0
5			0	0	0	0	1000
6			0	0	0	0	0
7			0	0	0	875	0
8			0	0	0	0	0

Fig. 11: Origin-Destination Matrix

The Origin-Destination Matrix used in the model is derived from a one-day data obtained from a cross-docking facility in New Jersey. Since the given data was not sufficient, a more detailed origin-destination matrix was constructed for the purpose of the model. The result was a 50x50 matrix, where each cell represents how many items need to be sent from the given origin to the given destination.

3.2. Door Assignment Tables

	A	B	C	D	E	F	G	H
1	Optimized Door Assignments							
2	To Block	Origin	Inbound Door		To Block	Destination	Outbound Door	
3	1035	1	17		8997	1	19	44
4	4503	2	18		9727	2	12	37
5	4719	3	6		9873	3	7	32
6	7	4	14		10019	4	4	29
7	220	5	13		10165	5	6	31
8	438	6	20		10311	6	10	35

Fig. 12: Optimized Door Assignments

	A	B	C	D	E	F	G	H
1	Random Door Assignments							
2	To Block	Origin	Inbound Door		To Block	Destination	Outbound Door	
3	1035	1	24		8997	1	3	44
4	4503	2	23		9727	2	4	37
5	4719	3	11		9873	3	8	32
6	7	4	1		10019	4	9	29
7	220	5	8		10165	5	1	31
8	438	6	4		10311	6	13	35

Fig. 13: Randomized Door Assignments

Two different door assingment tables were used, namely the G3QAP-optimized and randomized door assignments. *To Block* refers to the block number as represented in the simulation model. *Origin* refers to the item's origin and *Inbound Door* refers to the item's assigned door. Likewise for *Destination* and *Outbound Door*. Since the simulation model can only assume a set of assignments at a time, the model was run twice: one with optimized assignments and the other with randomized assignments.

3.3.　Facility dimension, resources available, and other parameters

The rest of the parameters are not imported from an Excel sheet, but rather inputted directly to the corresponding blocks inside the model. For your convenience, the summary of the rest of the parameters and where to change them is given in the following table.

Summary of Parameters

Name	Value	Corresponding block
Forklift Speed	30 m/minute	6248 Equation in Model Overview
Distance between opposite doors	37.5 m	6249 Equation in Model Overview
Distance between adjacent doors	4 m	6250 Equation in Model Overview
Available Forklifts	20	222 Resource Pool in Model Overview
Available Workers	30	223 Resource Pool in Model Overview
Maximum capacity of incoming trucks	2800 ft^3	4912 Set in Create Trucks and Items
Volume of incoming items	1, 2, or 6 ft^3 (equally probable)	4909 Random Number in Create Trucks and Items
Maximum capacity of pallets*	80 ft^3	1295 Constant in Inbound Door 1
Unloading Time*	0.01 minute/pallet	1430 Activity in Inbound Door 1
Maximum capacity of outgoing trucks*	3000 ft^3	9491 Constant in Outbound Door 1
Loading Time*	2*Volume / 60 minutes	9479 Equation in Outbound Door 1

*for changes in inbound / outbound doors, make sure to choose Open Structure and update other doors in current model

Fig. 14: Summary of Parameters

4. Testing the Model: Outputs

Sample outputs of the current model are summarized in the following table.

Sample Outputs

Output	Optimized	Random
Distance Traveled (m)	4713.9	7114.97
Forklift Utilization	0.47139	0.71
Workers Utilization	0.312	0.31
Average Item Cycle Time (minute)	198.92	297.63

Fig. 15: Sample Outputs

As expected, the total distance traveled by the forklifts inside the facility under the optimized door assignments is much less than that under the randomized assignments. Moreover, because the simulation finishes earlier in the former scenario, the forklift utilization is also less than that of the latter. Consequently, average item cycle time is also less in the former scenario, as less distance needs to be traveled and thus items exit the simulation faster.

5. Future Model

The following implementations should be considered in refining the model in the future:

1. Optimizing the door assignments internally in ExtendSim.

2. Introducing the "Weight" attribute. This will then act as another constraint for truck's capacity, in addition to the "Volume" attribute.

3. Enlarging the model to 200x200.

4. Adding more performance measures and tabulating them on the display for presentation purposes.

5. Tabulating all the input parameters on the model overview to facilitate any parameter adjustments, thereby reducing the need to browse through the respective blocks in order to change a given parameter.

6. Adding the "adjacent truck loading" mechanism, which puts the outgoing truck next to an inbound truck whose items mostly go to a particular destination.

7. Enabling the user to toggle between different staging, sorting, or any kinds of operation that have multiple implementations.

6. References

[1] - U.M. Apte and S. Viswanathan (2000), "Effective cross docking for improving distribution efficiencies," International Journal of Logistics, Vol.3, pp.291-302.

[2] - U.M. Apte and S. Viswanathan (2002), "Strategic and technological innovations in supply chain management," International Journal of Manufacturing Technology Management, 4: 264-282.

[3] - F. Chen, K. Song (2009), "Minimizing makespan in two-stage hybrid cross docking scheduling problem," Computers & OR, pp.2066~2073.

[4] - M. Wen, J. Cordeau, G. Laporte, and J. Larsen, "The Dynamic Multi-Period Vehicle Routing Problem," October 2009, CIRRELT-2009-43.

[5] - L.Y. Tsui and C.H.Chang (1992), "An optimal solution to a dock door assignment problem," Computer Industrial Engineering, 23: 283-286.

[6] - M. Guignard, P. Hahn, A. Pessoa, and D. C. da Silva (2012), "Algorithms for the Cross-dock Door Assignment Problem", Matheuristics 2012.

[7] - J. J. Bartholdi III, K. R. Gue, and K. Kang (2001), "Staging Freight in a Cross Dock," Proceedings of the International Conference on Industrial Engineering and Production Management, (May), Quebec City, Canada, 10 pages.

[8] - Z. Li, M.Y.H. Low, Y.G. Lim, and B. Ma "Optimal Decision-making on Product Ranking For Crossdocking/Warehousing Operations," Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference.

[9] - J. J. Bartholdi III and K. R. Gue (2004), "The Best Shape for a Cross Dock," Transportation Science, Vol. 38, No. 2, pp.235–244