CDAP TEAM

CDAP Simulation Report

Zongze Chen

2010/5/13

Contents

Acknowledgement

First and foremost, I would like to show my deepest gratitude to my supervisor, Peter Hahn, a respectable, responsible and resourceful Professor, who has provided me with valuable guidance in every stage of the writing of this report for my independent study. Without his enlightening instruction, impressive kindness and patience, I could not have completed this report.

I shall extend my thanks to Professor Monique Guignard-Spielberg for all her kindness and help. The courses she taught provide me a foundation in solving problems. I would also like to thank Ying Liu for her thesis work and the dataset we used in simulation. My sincere appreciation also goes to my simulation project partner Yao Xiao. Also, I want to give my great thanks to Imagine That Inc. for providing us ExtendSim to simulate.

I. Introduction

A cross-dock facility enables the moving of products from a manufacturing plant and delivers them directly to their customers with little or no material handling in between. Cross-docking not only reduces material handling, but also reduces the need to store the goods in a warehouse. A typical cross-docking structure includes **strip doors** and **stack doors**. Strip doors represent the doors where full trailers are parked and unloaded. Any incoming trailer can be unloaded to any strip door. Stack doors represent the doors where empty trailers are put to collect freight for specific destinations. There are many issues with which to be concerned in the processing of goods inside a cross-dock. Based on our previous work on the assignment of facilities to locations, we are concentrating on the issues that determine the cost of processing and moving goods inside a cross-dock. We have shown in our earlier work [1] that these costs can be minimized by appropriate assignment of receiving doors to incoming trailer trucks and stack doors to outgoing trailer trucks and have formulated this last problem as a Generalized Quadratic 3-dimensional Assignment Problem. We have also shown [2] that the problem is also solved exactly by the simpler Generalized Quadratic Assignment Problem.

In this project we collaborate with National Retail Systems (NRS) who provides us with realistic cross-dock situations and data to study and to evaluate our modeling and optimization result. NRS is a retail cross-docking giant in the Northeastern U.S. Their North Bergen facility receives goods from multiple vendors, sorts and loads them onto outbound trailer trucks for a number of retail stores. We will develop a simulation model of two cross-docks in operation of their New Jersey facility - Buildings A and B. Since last November we visited NRS twice and got familiar with the daily processing in the cross-dock.

For my TCOM 899 independent study project, I will develop a simulation based on ExtendSim software in order to model the processing within the cross-dock operations performed by NRS. There are several main objectives to be achieved:

1. Develop the model of cross-dock process. For simplicity, we start the development with a 4×4 cross-dock. Later, this will be expanded to realistic dimensions.

2. Use the data generated from the GQ3AP algorithm to simulate the process and analyze the total cost under the situation and discuss the optimization.

3. Improve our optimization models to take into account the impact of truck arrival and departure times. Determine how one can improve cross-docking operations and what costs could be reduced through improved operational control.

Simulation nowadays is becoming a very useful tool when analyzing and testing the algorithm adapted in real situation. In this independent study period, we successfully received a research grant from ExtendSim who provide the main software for the simulation.

This report describes how simulation helps ensure success of cross-docking systems by determining optimal routing costs. Modeling methods and issues are also discussed as they apply to cross-docking. This report includes discussion of the actual processes employed by NRS, description of our models, simulation results and comparisons, and our conclusions.

II. Cross-dock Process

1. General Process

In logistics, cross-docking is described as the flow of material directly from receiving to shipping, where the goal is minimal handling and virtually no storage.
Cross-docking applies to case and pallet handling done either automatically or manually. Cross-docking is done to change the type of transportation being used, to sort materials intended for different destinations, or to combine material from different shipments. Cross-docking is used to decrease inventory storage by streamlining the flow between the supplier and the manufacturer. Figure 1 shows the basic process of a cross-dock.

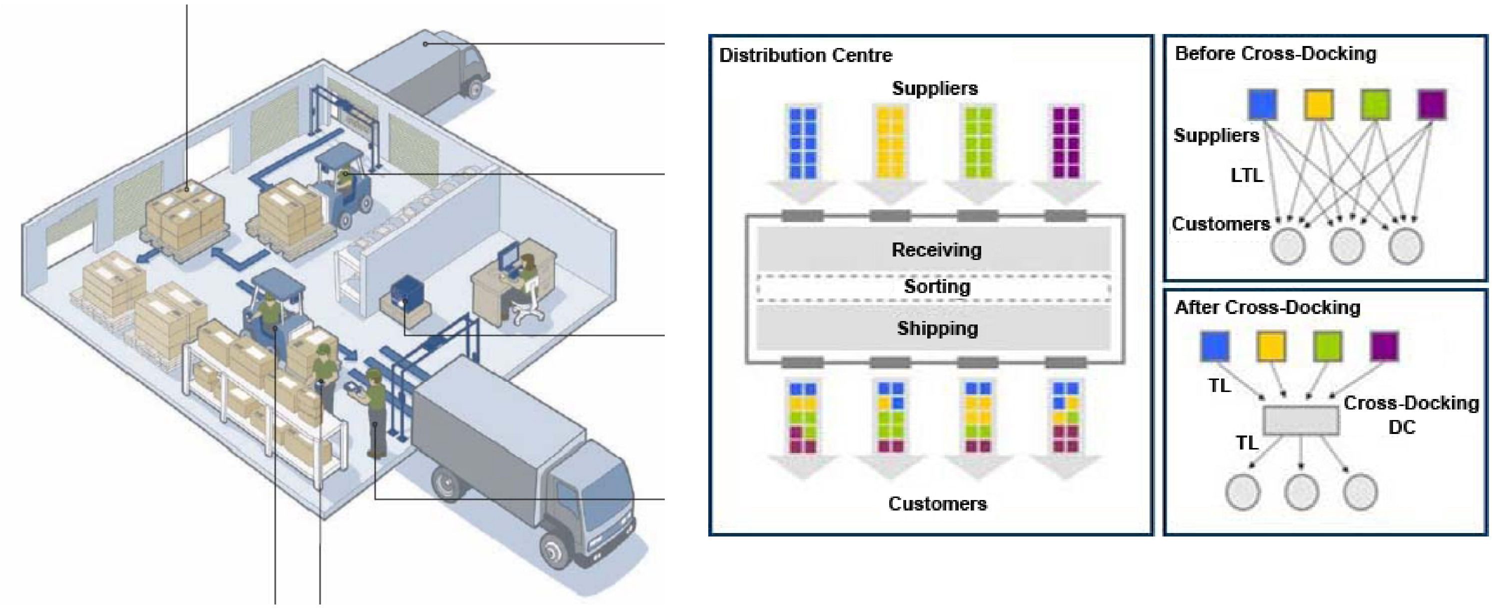


Figure 1 Cross-docking operations and benefit

On the left of Figure 1, we can see inside a cross-dock, with the incoming traffic. Inside the cross-dock, the products will be merged and sorted to the different destination exits. This significantly improves the efficiency of the logistics process. Observe the situation depicted in the top right-hand corner of Figure 1, if there is no intermediate cross-dock, it would take 12 trailer trucks to transfer all the products between suppliers and customers. Moreover, these trailer trucks would be partially empty. Now, observe the situation depicted in the lower right-hand corner of the figure. Only seven trailer trucks are needed and these are more efficiently utilized.

2. NRS Cross-dock Process

To be more realistic, we visited NRS twice and studied the process in Building A and B, C. Among them, building C contains a complex conveyor system that sorts the transferred goods via RFID sensing. The working system in Building C is highly automatic. This is not a cross-dock in the classic sense. Thus, we mainly analyze the building A and B where the process can be seen from the Figure 2 below:

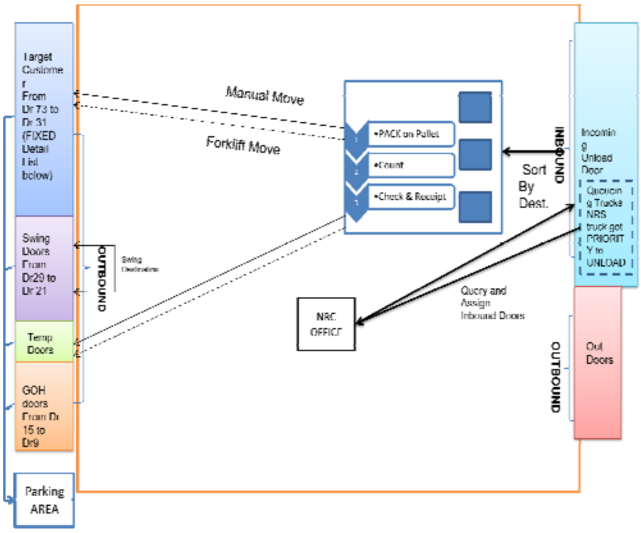


Figure 2 NRS Building A&B structure

In building A and B, there are several stack doors and strip doors lying opposite to each other. Also, there are some temporary doors to be assigned when necessary in order to meet high demands during the day. On the Left corner of the cross-dock, NRS provides some GOH (Garment on Hanger) doors to transfer clothes to the target customers like Macy`s. The processing performed by NRS mainly consists of:

Initial Phase: Suppliers and customers should contact with NRS office to schedule drop-off and pick-up times in advance.

1st The incoming trailer trucks come in according to the pre-scheduled time.

2nd By contacting the NRC office, each incoming trailer truck will be assigned a strip door and corresponding stack doors (one input to many outputs). The goods unloaded from the trailer trucks will experience several procedures including: sorting by destination, stacking on a pallet, counting the number of goods and checking the received products with the receipt. A receipt contains the information to be confirmed such as goods number, goods destination and the priority level or special requirements for the goods.

3rd If confirmed, the pallets will be transferred to stack doors either by forklift truck or manual forklift. In building A, most pallets are transferred manually. In building B, most pallets are carried by forklift trucks.

4th There will be a trailer waiting at the stack doors, and if the outbound trailer is full, then, it will leave for the destination or go to the parking area to wait for a departing tractor, according to the schedule.

In the NRS processing system, they have a special system based on Google Maps to track the location of incoming trailer trucks and make necessary adjustments for door assignment.

III. Model Development

1. General Explanations

To achieve the goal of simulating realistic cross-dock operations, we need to build a simple model to which we can easily add necessary more complex features later. Also, we should use simulation software to represent the cross-docking operations precisely and vividly. Thus, based on the above criteria, we chose to use ExtendSim software to do the simulation. This is because on the one hand, ExtendSim is powerful and has many functions and on the other hand, it consists of blocks that can be easily expanded to add new processing steps. In Feb 2010, I qualified for a research grant from ExtendSim Company, which included the license for ExtendSim OR version 7.

In this Section, I will go through the model development and the introduction to ExtendSim software.

A general model should take the details of the following sub-sections into consideration: The inbound services, the transportation route, and the outbound exit. I will discuss the three parts in detail for the model we put together so far.

1. Inbound Services:

The inbound services includes the incoming trailer truck schedule, the unload process, the procedure in packing to a pallet and also, the destination for each pallet.

In the freight industry, every company keeps a record of its pre-scheduled trailer truck information. Thus, when the new customers call in to schedule a new slot to load/unload the products, the company can easily assign them based on available time slots and avoid unnecessary collisions.

Each of the pallets will go to their own destination based on the mark on it. In the cross-dock, the products from one strip door can be routed to different stack doors, as the receivers can be different companies. For example, one supplier called SONY provides TVs and this company gets many customers like Target, Bestbuy, etc. Usually, the person supervising the cross-dock operations assigns the incoming and outgoing doors with little analysis, and this is likely to result in high processing costs.

2. Transportation Route:

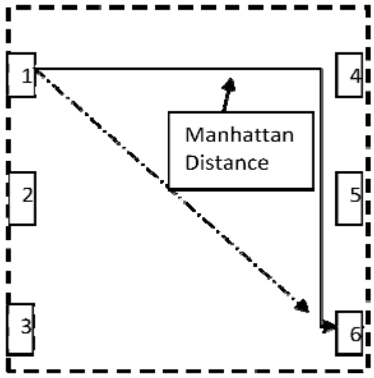


Figure 3 Manhattan Distance Example

We are currently developing a model for general cross-docking layout, where doors are symmetrically located in the two terminals shown in Figure 3. We set the distance between two adjacent doors within one terminal (e.g., door 1 and door 2) to be 1 *Distance Unit* and the distance between two parallel doors in different terminals (e.g., door 1 and door 4) to be 3 *Distance Units*. To avoid collisions of forklifts, we assumed that a forklift can move only along axial paths, i.e., if a forklift wants to transfer one pallet from door 1 to door 6, the distance moved is 1+1 + 3 = 5 *Distance Units* (thus the distances are rectilinear). This concept of movement is called Manhattan Distance. Manhattan Distance means the distance between two points measured along rectilinear paths at right angles. In a plane with point 1 at (x_1, y_1) and point 2 at (x_2, y_2), it is

$| x_1 - x_2| + | y_1 - y_2|$. By using Manhattan metrics, we can avoid the collision as much as is possible and also achieve a high efficiency.

Also, there is an important factor for the optimization of routing, which is to consider the flow capacitance. For example, a path with large traffic capacitance should hold more traffic flow than other paths.

3. Outbound Exit:

At the stack door, workers will load the pallets onto outbound trailers. In this process, the workers will collect all the pallets from different incoming doors, and merge them together. They will also arrange them properly inside the trailer. Once the trailer is full and if the schedule leaving time is up, then the trailer truck will proceed directly to its destination. If there is still more time left for departure, the trailer will be taken to parking area and wait for the destination tractor to pick it up.

The facts above include the most general requirements for the typical cross-docking in our simulation. For a cross-docking solution to succeed, the current and future operations must be understood completely. Because simulation includes building a virtual distribution facility, it can provide additional insight beyond other design techniques. Thus, by simply building a simulation model, project team members will better understand their current and future operations. Thus, we chose a typical 4-by-4 assignment cross-docking to begin our simulation development. And, we change the factors such as different schedules, products amount and type, route capacity during the simulation. Here, I will introduce the simulation software used in our project:

4. Simulation Software:

We are using ExtendSim to do the simulation mainly for the following reasons:

1. It is "open" software so that modelers can customize it infinitely, if needed. Most often, the ExtendSim's existing prebuilt blocks or icons of ExtendSim meet the requirements. So, customization is seldom required. Simplified C language is used to build models. The API (Applications Programming Interface) is free.
2. It is easy to use. Just draw the flow chart of the system you have in mind. After populating with data and other information, it automatically morphs into an animated simulation model of the system. It is that simple, because of its advanced messaging system, transparent to the user.
3. Productivity of the modeler is enhanced considerably and hence the cost of ownership (TCO) of the simulation model comes down dramatically.
4. Data interface to external data sources such as Excel, SAP, Oracle etc is available through ActiveX, Com and ODBC technologies. Each model can have its own memory resident database, so that even a novice can operate the model. It is a standalone application that does not require any other tool for its operation.
5. ExtendSim incorporates animation for model communication, debugging, and presentation. By default, models are animated on the worksheet, providing a quick view of the workings of the simulation. A more sophisticated worksheet animation can be created by animating the hierarchical blocks and changing the pictures that represent the products. This is all done within the standard ExtendSim drag and drop interface. In the simulation, ***item*** is the terminology of the simulation for a ***product***.

Our ExtendSim research grant provides ExtendSim version OR - Operations Research to us. This version uses Discrete Event Simulation, which is appropriate for tracking and analyzing the behavior of physical or logical entities when events cause them to change state or move through a system. Using ExtenSim OR, we develop the model correspond to the requirements mentioned above. Figure 4 reveals a basic overview of the simulation model.

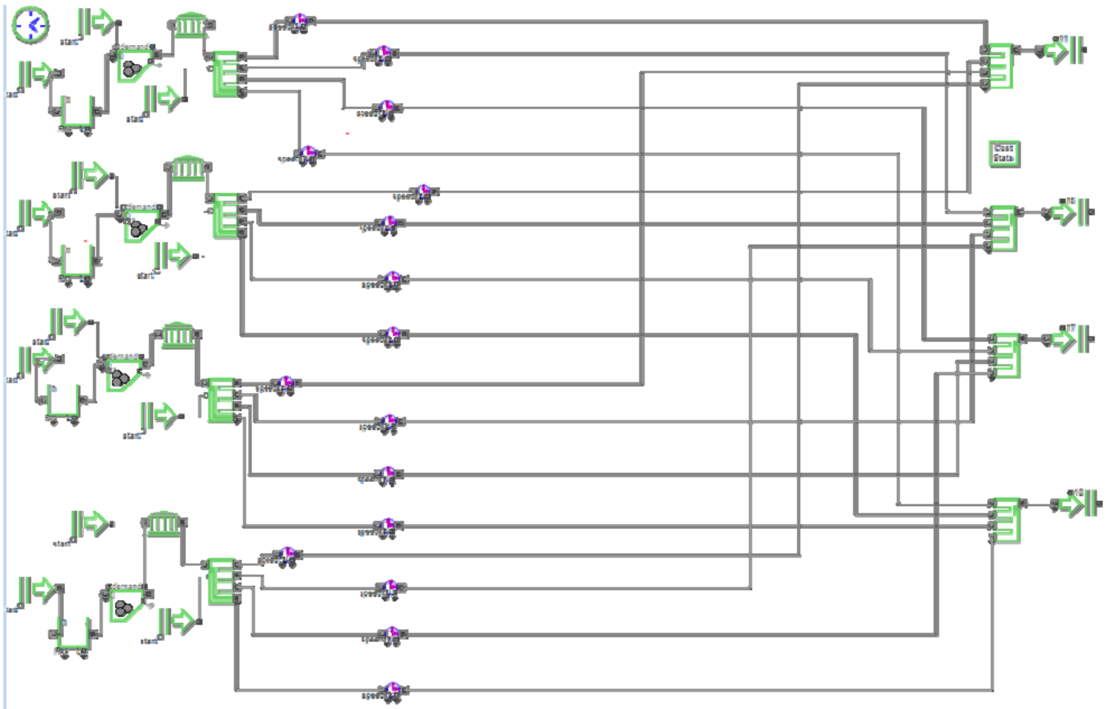


Figure 4 Four Strip doors-by Four stack doors Model

The details of the diagram are introduced in the next three sub-sections (Inbound Services, Transportation Route and Outbound Exit.

2. Inbound Services

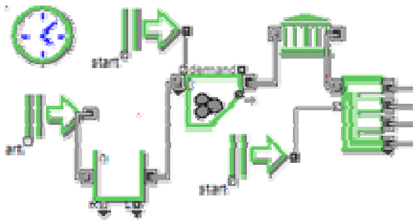


Figure 5 Strip Door Model

As we mentioned above, we choose to simulate a general model that can be expanded later. In our initial simple model, we build a 4 strip doors × 4 stack doors layout cross-dock. We represent each strip door separately and build an inside view to see the process of each door. Each strip door can either work simultaneously or process with its own schedule. In ExtendSim OR, each function is represented by a block. Figure 5 above shows the details of one strip door.

Create Block:

The *Create* block creates items randomly, by schedule, or infinitely and also can be used to create values randomly or by schedule. It can initialize newly created items with properties, such as *attributes* or *priorities*.

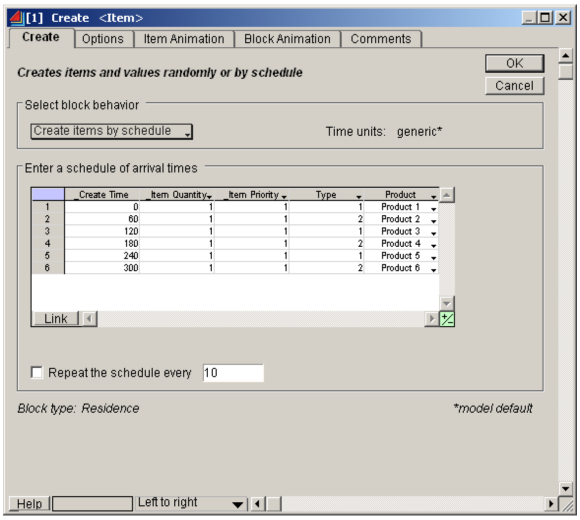
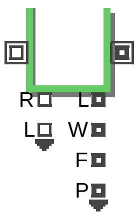


Figure 6 Create block dialog box

Randomly: A random distribution causes Items/Values to be generated with a random or constant inter-arrival time. The distribution determines the time between item/number arrivals; a smaller inter-arrival time indicates that items/number will arrive more frequently.

By schedule: Creating items/Values by schedule causes an item/number to be generated at a specific arrival time. The schedule defines when the item/number will arrive and the time between arrivals is fixed. (Shown in Figure 6)

As we said, the *Create* block can generate items and also values and here we use both. The difference between item and value is that, item means a product, like TV, Clothes, and cannot be quantized, while the value means real numbers, like a pulse or a number “1”/ “0” to represent the true or false, or the numbers indicate the demand or need.



Queue Block:

A *Queue* block provides a buffer or waiting line to store items waiting further processing. *Queue* blocks can have simple behavior, such as holding items in first in, first out (FIFO) order, or more complex behavior, such that items are held and released in groups based on their attributes, also we can set an option in the *Queue* block's dialog (Figure 7) to specify how long an item will wait until it reneges, or prematurely leaves.

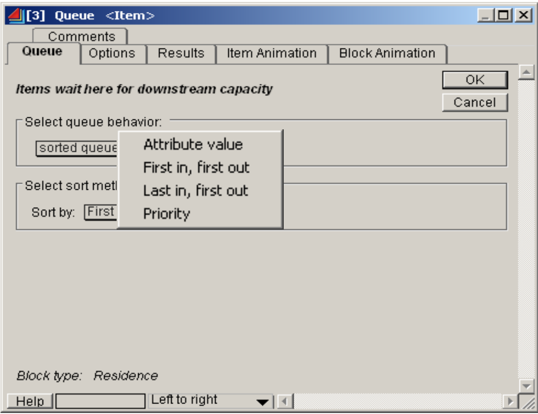


Figure 7 Queue block dialog box

We use the *Queue* block to perform the buffer function in cross-docking, when the workers build the pallet, there may exist a waiting time, and the incoming boxes will be stored at that time. A *Queue* block can simulate that process very well.

Select Item Out Block:

The *Select Item Out* block selects which output gets items from the input, based on a decision like the dialog box shows in Figure 8. The decision can be made automatically such as random picking. And also the decision can be manually set by the system designer such as to put a schedule inside the dialog box.

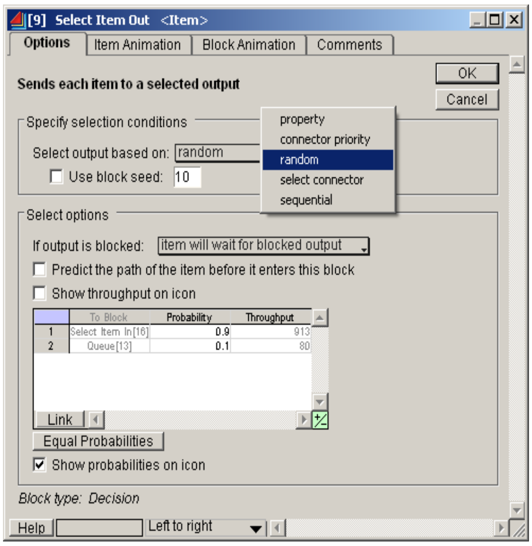
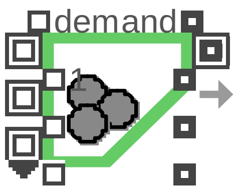


Figure 8 Dialog box for the Select Item Out block

We use *Select Item Out* block as a strip door of the model, it actually decides which stack door a pallet should go to. In our model, there will be four stack doors, and thus, four outgoing options corresponded. There are several criteria to make the route decision such as random picking or select connector based on schedule. Obviously we will choose the latter option, as in our simulation, each destination has a certain amount demand for the products. Thus, we should make the decision according to the demand of each customer. To make the decision, we then use the combination of two blocks : *Create*“Values”and the *Select Item Out*.

The *Create* block will send a scheduled order to the *select item out* block, in order to control which way to go, for example, at Time “2”, it will send a “0”to the *Select Item Out* block which means, "Now, when you receive any products just directly transfer them to the top connector". Here, the default value “0” means the top connector which links to the stack door 1. The rest of the connectors are in sequence as value “1” for stack door 2 and etc. So if the received value is “1”, all the products will be transferred to the second connector.



Batch Block:

Batching allows multiple items from different sources to be joined as one new item for simulation purposes (processing, routing, and so on). The *Batch* block accumulates items from each source up to a specified count, and then releases a single item that represents the batch. In this process, the original input items are destroyed and replaced by one new output item. The number of items required for a batch is called the "*batch size*". Items can be permanently batched together as one new item that flows through and exits the model, or they can be temporarily joined for some specific purpose and un-batched at a later point in the process.

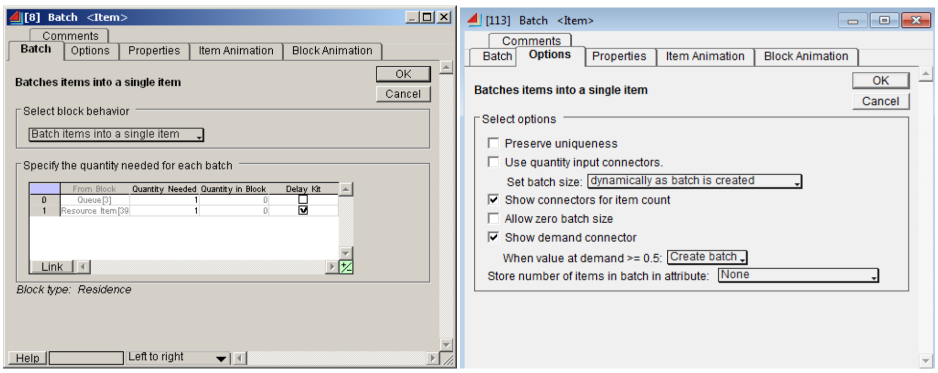


Figure 9 Batch block dialog box

Figure 9 shows the options that exist in the *Batch* block. Here we should take notice of the batch sizes, which would likely be different for different products. In real life, some of the products are large like TVs or fridges, and some are small, like food or rice cookers. Thus, due to different sizes, the pallets should have different sizes. For example, it may take just two to build a pallet for the large items like TVs, while it would take 15 units to build a pallet for small appliances, such as rice cookers.

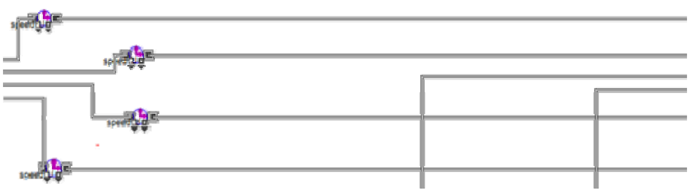
Also, the process in the simulation may bring up a new question which is, when we set a “batch size”the *Batch* block will wait until the size is achieved. However, if the number is not achieved and meanwhile all the products are unloaded, then they will not be transferred and stayed at the *Batch* block. To solve this, we set a demand function which is also controlled by the value generated by *Create* block, the design method is the same as it in “select routing”portion of the simulation.

In detail for this specification: Sometimes the number of items required to create a batch changes during the course of the simulation. For instance, outside factors could determine how many items go into each batch, or you may want batches to be made in a time-dependent fashion. As shown in the Figure 9, the *Batch* block’s Options tab allows you to manipulate the size of batches through quantity input connectors or through a demand connector.

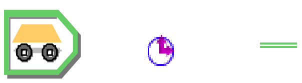
Create batch when value at demand $\geq$ 0.5. Items are allowed into the *Batch* block as they are available, up to the required number. However, the batched item will not leave the block as long as the demand connector has a value < 0.5. When the demand connector becomes $\geq$ 0.5, the batched item leaves the block. With this option, a batch can consist of fewer items than the number in the Quantity Needed column, because the batched item will have been created by joining whichever items were available when the demand connector got a value $\geq$ 0.5.

To sum up, we can monitor the status and simulate the process in the unloading portion of the simulation. By making decision to different routes, we can assign the target products to the customers.

3. Transportation Route



We talked about the Manhattan Distance in the in *General Explanations* Section of this report. Thus, when building our model for cross-docking, we use the vertical and horizontal lines to indicate the path. As we started our simulation development with a model of 4 strip and 4 stack doors, therefore, there would be 16 possible combinations. Each of the strip doors has four options for choosing where to go. For convenience, we always make the top connector link to stack door 1, the next connector link to stack door 2, and so on.



Transport Block:

Transport blocks move items from the start of a path to the end of the path based on distance and speed information. When the model is animated at a later stage in its development, we can display multiple items travelling a certain distance together.

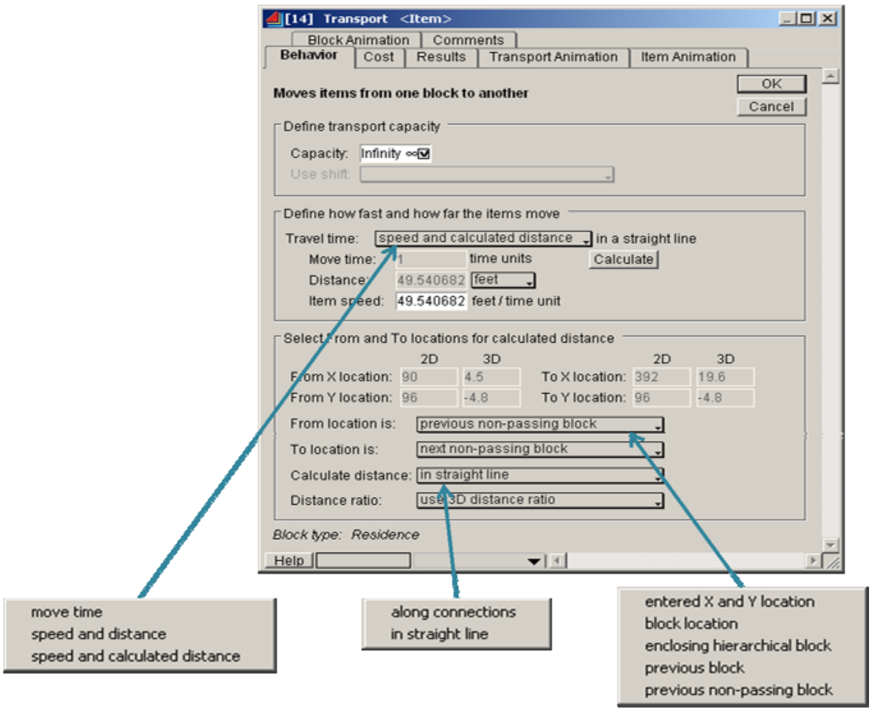


Figure 10 Transport block dialog box

The *Transport* block's Behavior tab in Figure 10 has options that specify how to calculate travel time – the time it will take an item to travel from the starting point to the ending point. Travel time can be based on: *Move Time*, *Speed and Distance*, and *Speed and calculated distance*. In our model we chose the second option.

Speed and Distance is described in the following detail: A. how fast the item is traveling, and B. how far the item must travel to reach its destination. These details are entered in the fields in the *Behavior* tab or received at input connectors. The calculated *move time* is displayed in the dialog. This option is most often used if the transportation pathway is centered on this block.

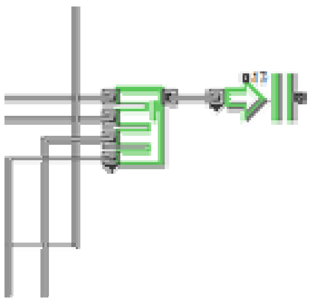
As we have been given the layout of cross-dock, thus the distances should be known to us.

Here, like we said above in *General Explanations* Section, the vertical and the horizontal distance are made in a certain proportion in order to simplify calculations. In our model, we set the horizontal distance at 1200 *Distance Units* and the vertical distance between two adjacent doors at 400 *Distance Units*.

As a very important factor in the optimization, cost is always the main parameter to be minimized. Simulation on the one hand shows the process inside a cross-dock and also, on the other hand, keeps track of the total cost in this process. The major factor of cost in the transportation process is time delay. We manually set a speed element of 20 *Distance Units* for the workers to move a pallet per *Time Unit*. And, to calculate the cost we set a factor of 10 *Cost Units* for each *Time Unit*.

Another important factor is the flow capacity. We are about to set up a capacity requirement in the *Transport* block, too. In the real situation, NRS told us, normally they would have sufficient capacity to move the pallets.

4. Outbound Exit



The outbound service consists of two main components. One component is outbound loading and the other is trailer truck departure. In Figure 1 (Section II), we can see the function of cross-dock, which is to unload, store and combine different products from different suppliers and send them to the various customers. Thus, each customer in this model will receive a combination of products to fit his, her or its demand. For example, a customer herein named Macy`s may need TVs, rice cookers, and also clothing from the corresponding three different suppliers herein called SONY, Panasonic and Calvin Klein. So when we build the model in this section, the main function should be to merge different flow paths into one.

The *Select Item In* block can combine the inputs from any number of sources into one stream of output items. The *Select Item In* block will accept items from any of the four inputs. Its dialog is set to select input based on: merge. If the *Select Item In*'s output is blocked, the block will force items to drop or to wait in queue. When the *Select Item In* becomes unblocked, it will check each input in turn to try to pull an item through to the *Exit*.

According to the real situation, the trailer truck will leave for the destined customer, based on the schedule. Thus, we are about to set up a *Decision* block and a *Timer* to determine the schedule and whether the truck is full or not and then decide where to go.

5. Other Additional Blocks

History Block:

Each item that passes through the *History* block (if it is connected in series) or is viewed by the block (if it is connected in parallel) is allocated a row in the *History* block's table. The table's first column displays the item's arrival time. Popup menus at the top of the other columns are for selecting additional information to display, such as the value of an attribute, an item's property, and so forth.

Statistics Block:

The *Statistics* block (Value library) accumulates data and calculates statistics for a particular type of block using a specified statistical method. In addition to the block number, block name, and the time the information was observed, this block displays metrics that are specific to the block type, such as utilization or average wait time for activity-type blocks or the mean, variance, and standard deviation of all the *Mean & Variance* blocks in the model.

Cost Stats Block:

The *Cost Stats* block (Item library) records in a table the input costs and total cost generated by each costing block in a model. This block reports cost information for item-based blocks, such as for a discrete event model. Information can be exported to a spreadsheet. The block can be placed anywhere in the model.

IV. Test Example

After building all the modules in the ExtendSim, we try to perform the simulation under the sample dataset. The previous student Ying in this project generates a very simple sample data for us to use, which is quite helpful. The data is generated by optimization algorithm.

INPUT DATA					
Flow matrix	(number of pallets, not boxes)				
		Customer 1 (Macy's)	Customer 2 (Target)	Customer 3 (Costco)	Customer 4 (Walmart)
Supplier 1	(TV)	5	3	3	4
Supplier 2	(refrigerator)	2	4	6	3
Supplier 3	(cloths)	6	2	5	3
Supplier 4	(food)	2	1	3	1

Distance matrix	(Using Manhattan distances, horizontal distance of the grid is 3, and vertical distance of the grid is 1.)			
	outbound door 1	outbound door 2	outbound door 3	outbound door 4
inbound door 1	3	4	5	6
inbound door 2	4	3	4	5
inbound door 3	5	4	3	4
inbound door 4	6	5	4	3

Door capacity	Maxmium 20 pallets per door

SOLUTION								
Minimum total cost =213, with the following optimized door assignments :								
Supplier 1	(TV)	>>>----->>>	inbound door 1	Cross-dock	outbound door 1	>>>----->>>	Customer 4	(Walmart)
Supplier 3	(cloths)	>>>----->>>	inbound door 2		outbound door 2	>>>----->>>	Customer 1	(Macy's)
Supplier 2	(refrigerator)	>>>----->>>	inbound door 3		outbound door 3	>>>----->>>	Customer 3	(Costco)
Supplier 4	(food)	>>>----->>>	inbound door 4		outbound door 4	>>>----->>>	Customer 2	(Target)

Figure 11 Sample Test Data

The dataset includes several aspects, here are the details:

1. Four virtual suppliers with different products and four virtual customers with different needs of certain products are generated.
2. Four products TV, fridge, clothes, food are generated.
3. A simple 4-by-4 cross-dock is generated, with vertical and horizontal ration 3:1.
4. Different customers have different needs of products, for example, Macy`s needs 5 units TVs, 2 Units fridges and 6 Units clothes, and 2 Units food. (The Units here

can be pallets)

5. By using the GQ3AP crossdock door assignment algorithm, we assigned Supplier 1 (TV products) to use door 1, Supplier 3 (clothes products) to use door 2, Supplier 2 (fridge products) to use door 2 and Supplier 4 (food products) to use door 4, in the inbound side. The stack door assignments are customer 4, 1, 2, 3 in sequence.
6. As we said, different size of the products should have a different batching size. Thus we assumed that

TV:	Two TVs make one Pallet	Total of 30 TVs are needed
Fridge:	One Fridge makes one pallet	Total of 15 Fridges are needed
Clothes:	Ten Clothes make one pallet	Total of 160 Clothes are needed
Food:	Five foods make one pallet	Total of 35 foods are needed

With this dataset, we have set up our simulation model. And when we perform the simulation, in order to test the optimized solution, we manually set up several other routes to compare with.

For example, we changed the sequence of strip door1 and door2, which means, Clothes will go through door1 and TV will go through door2. And we did other similar changes too.

V. Simulation Result

<u>Optimized</u>

We simulate the optimized routes which are given by the data provided in the GQ3AP optimization of the 4 by 4 example described in *General Explanations* Section that we are using for this first rudimentary simulation effort . Using the *Cost Stats* block, we can easily monitor the total cost of the process.

	Block	Block Name	Cost/Item	Cost/Time Unit	Total Cost	Time
0	1	Transport	0	10	2400	1300
1	13	Transport	0	10	4000	1300
2	23	Transport	0	10	3000	1300
3	31	Transport	0	10	3600	1300
4	43	Transport	0	10	2400	1300
5	48	Transport	0	10	1200	1300
6	53	Transport	0	10	4800	1300

Link

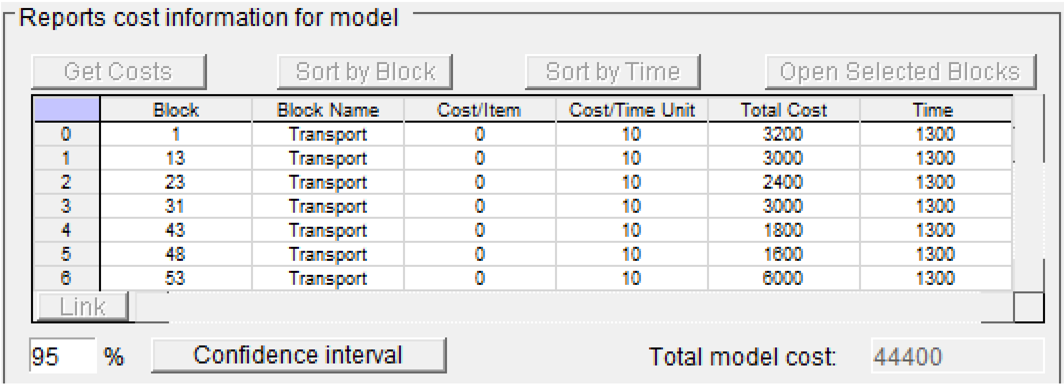
95 % Confidence interval Total model cost: 44000

Figure 12 Optimized Route Cost

Figure 12 shows the 44000 *cost units* in total, and enumerates the cost in each block. In fact, the calculated cost is the sum of cost from each functional block like *Queue* block and *Transportation* block.

As mentioned above, we also simulated some other routing scheme to see whether the proposed route is the most optimized one. And Figures 13 and 14 below are examples of the schemes after making changes to the route. From the costs monitor, we can see the cost jumped from 44000 *Cost Units* to 44400 *Cost Units* and 45800 *Cost Units* respective. After making all the possible changes, we can conclude that the route we proposed is the most optimized.

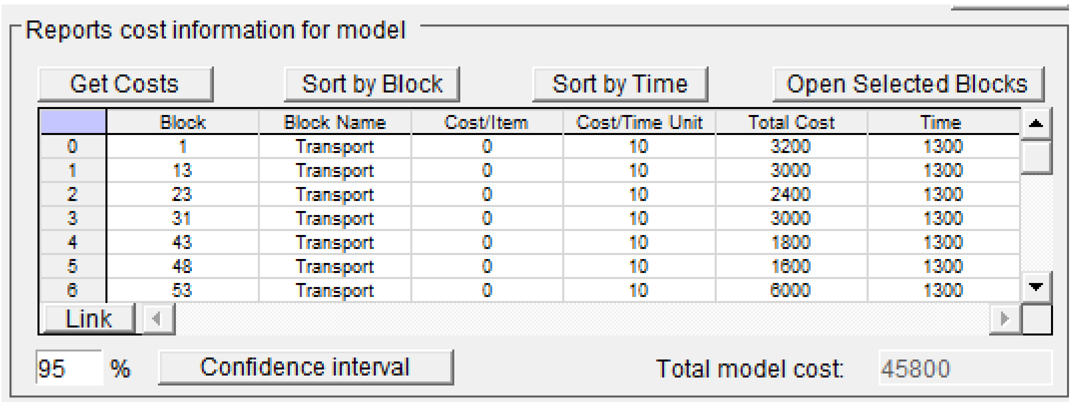
<u>Strip door1 &2 exchanges</u>



	Block	Block Name	Cost/Item	Cost/Time Unit	Total Cost	Time
0	1	Transport	0	10	3200	1300
1	13	Transport	0	10	3000	1300
2	23	Transport	0	10	2400	1300
3	31	Transport	0	10	3000	1300
4	43	Transport	0	10	1800	1300
5	48	Transport	0	10	1600	1300
6	53	Transport	0	10	6000	1300

Figure 13 Modified Route Cost I

<u>Strip door1& door2 door3 & door4 exchange</u>



	Block	Block Name	Cost/Item	Cost/Time Unit	Total Cost	Time
0	1	Transport	0	10	3200	1300
1	13	Transport	0	10	3000	1300
2	23	Transport	0	10	2400	1300
3	31	Transport	0	10	3000	1300
4	43	Transport	0	10	1800	1300
5	48	Transport	0	10	1600	1300
6	53	Transport	0	10	6000	1300

Figure 14 Modified Route Cost II

VI. Plans for the summer

So far we have finished building a general model of cross-dock and we also determined most of the blocks that are necessary for an adequate simulation of the operations in Buildings A and B. The first phase is achieved by simulating the sample data and the process. Nevertheless, we still have to refine the model, expand it to deal with whatever specific situation interests NRS and experiment with a typical realistic situation suggested by NRS requirements.

A. Refine Model

Since we made a general model of cross-dock, we should be able to expand it to simulate a more complex structure of cross-dock. The aspects to be expanded are as follows:

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.
2. In the current model, it still lacks some of the functions to simulate the whole process. Two main functions are to be added.
   a. In a cross-dock, the customer truck at the stack door should have a pre-determined schedule to decide when they should leave. In the future work, we will add a *Timer* block to track the current time and make a decision of whether to leave or not.
   b. In a cross-dock, the customer truck at the stack door should have a certain capacitance. If the truck is full, it should leave to the parking place and use another truck instead. In the future work, we will certainly add a block to decide whether the truck is full or not. We will set volume limits for the truck and compare the current loads with these limits.

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

B. Expand Model

Normally a cross-docking station has far more strip and stack doors than the model we proposed, i.e. 4-by-4. So in the future, we need to add more stack and strip doors. For example, we should build a 16 strip doors-by-16 stack doors model like the structure inside NRS cross-docking. That requires us to use techniques to combine small blocks as a whole in order to save space on our worksheet.

C. Experiment with Typical Realistic Situation

After refining and expanding the model, we should experiment with a typical realistic situation suggested by NRS requirements. For example, we should deal with the daily schedule provided by NRS for a variety of customers.

VII. References

[1] Zhu, Y-R., “Recent advances and challenges in quadratic assignment and related problems”, PhD dissertation, University of Pennsylvania, 2007

[2] Liu, Y., "A Comparison of Methods for Solving the Crossdock Door Assignment Problem," A thesis in Electrical and Systems Engineering, University of Pennsylvania, 2009

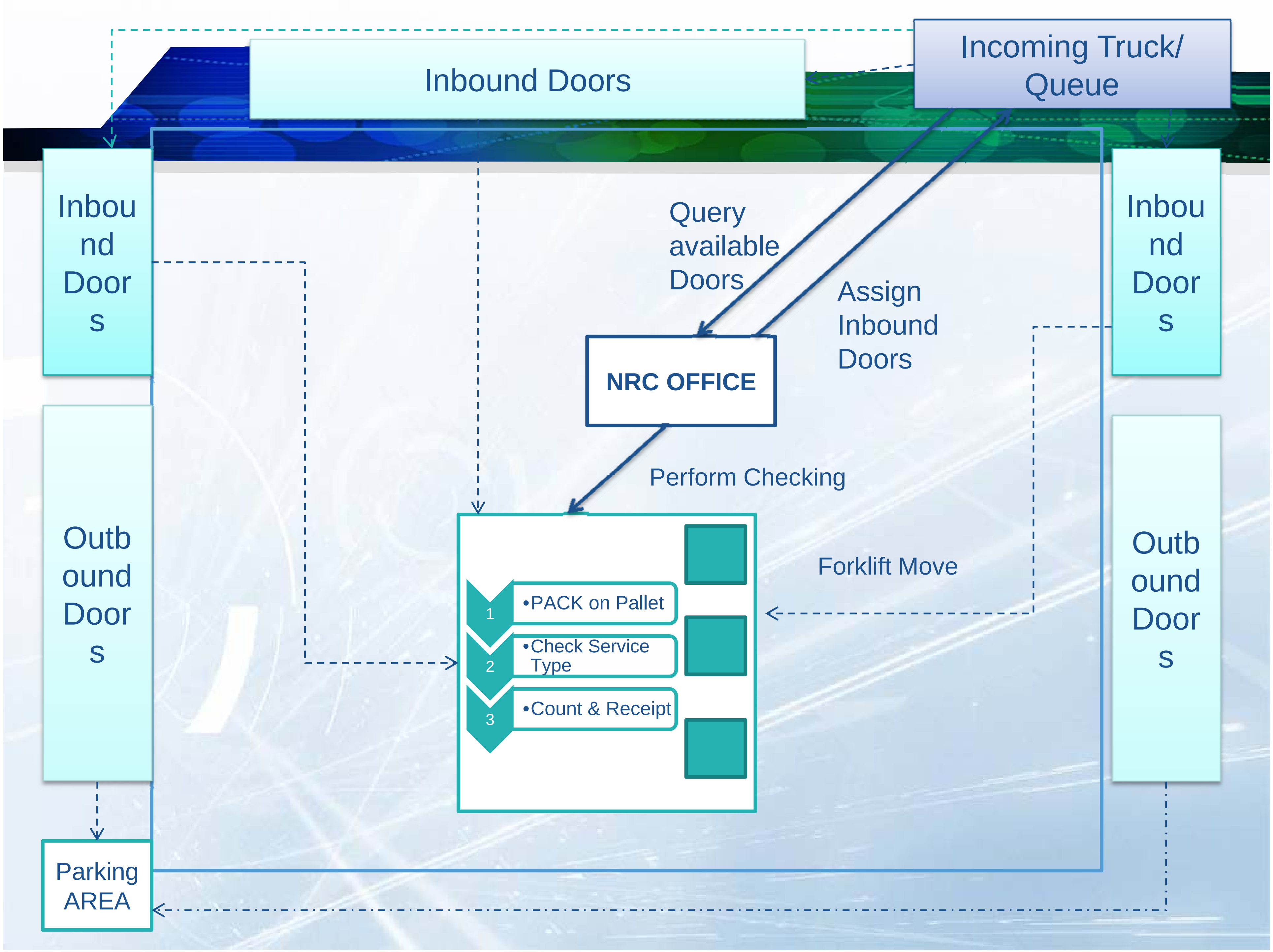
[3] ExtendSim User Guide for Version 7

Simulation for Cross-dock

by Zongze Chen and Yao Xiao
University of Pennsylvania

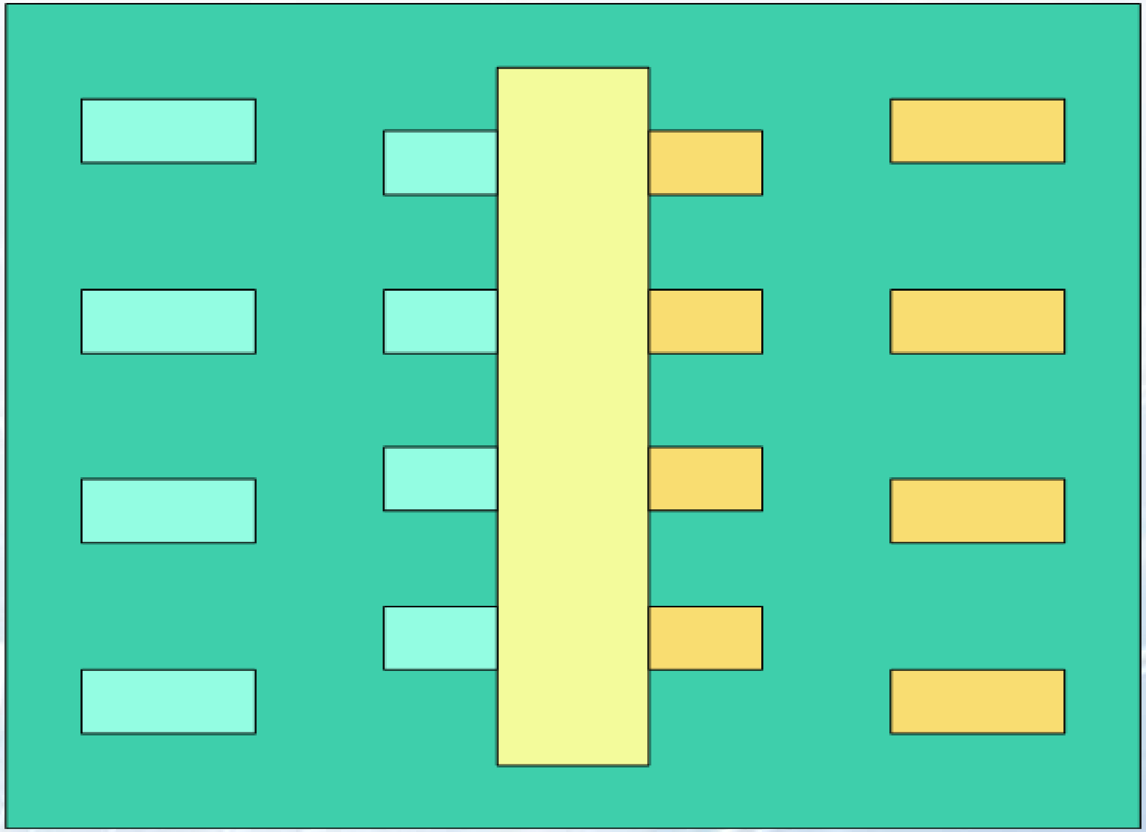
Content

- Background and Real Situation
- Model in Simulation
- Data to Simulate
- Schedule and Route Setup
- Simulation Results



Incoming Truck/ Queue
Inbound Doors
Inbound Doors
Inbound Doors
Query available Doors
Assign Inbound Doors
NRC OFFICE
Perform Checking
Forklift Move
Outbound Doors
Outbound Doors
1 •PACK on Pallet
2 •Check Service Type
3 •Count & Receipt
Parking AREA

CDAP Parameters



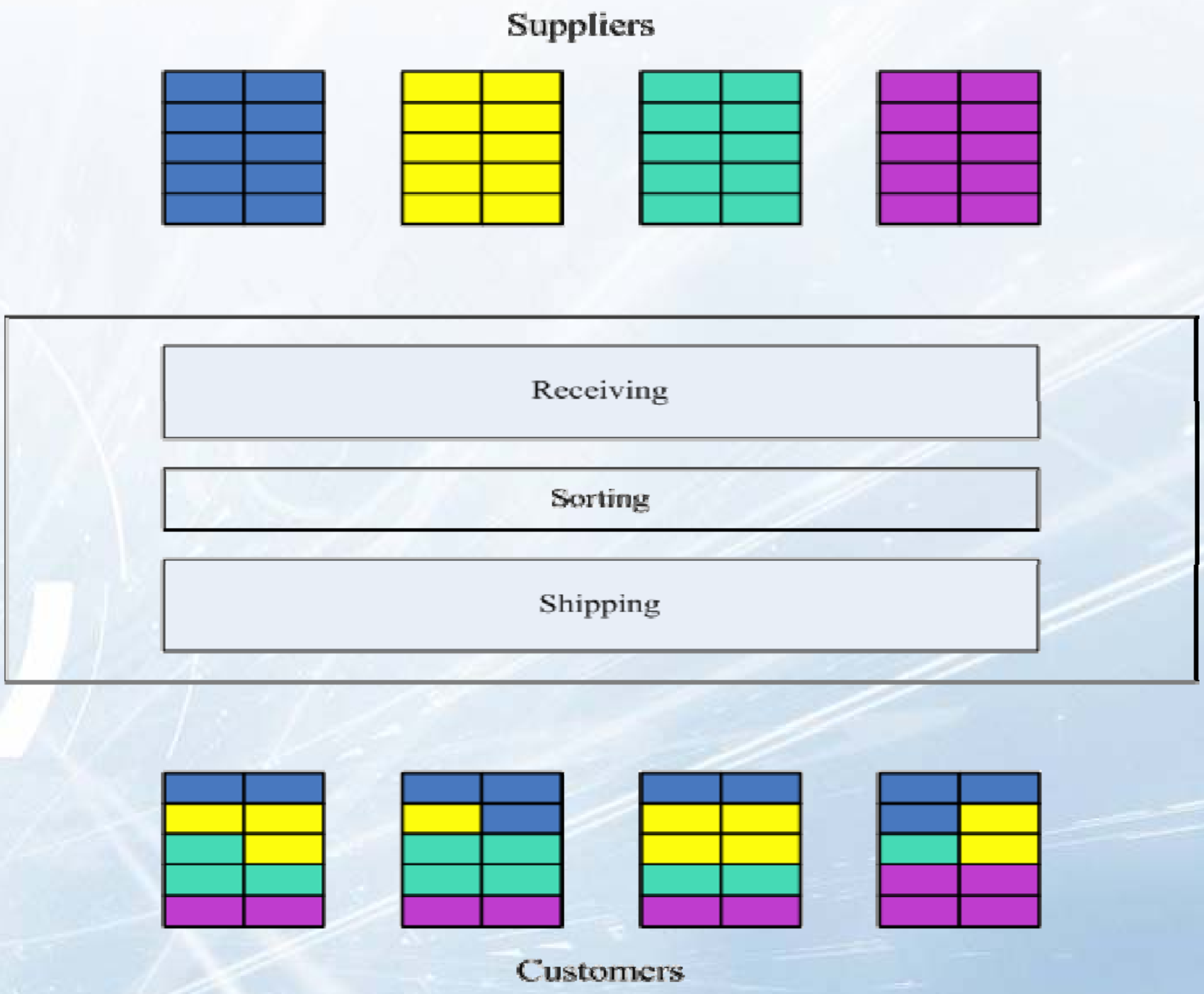
Origin

Strip
(Receiving)

Stack
(Shipping)

Destination

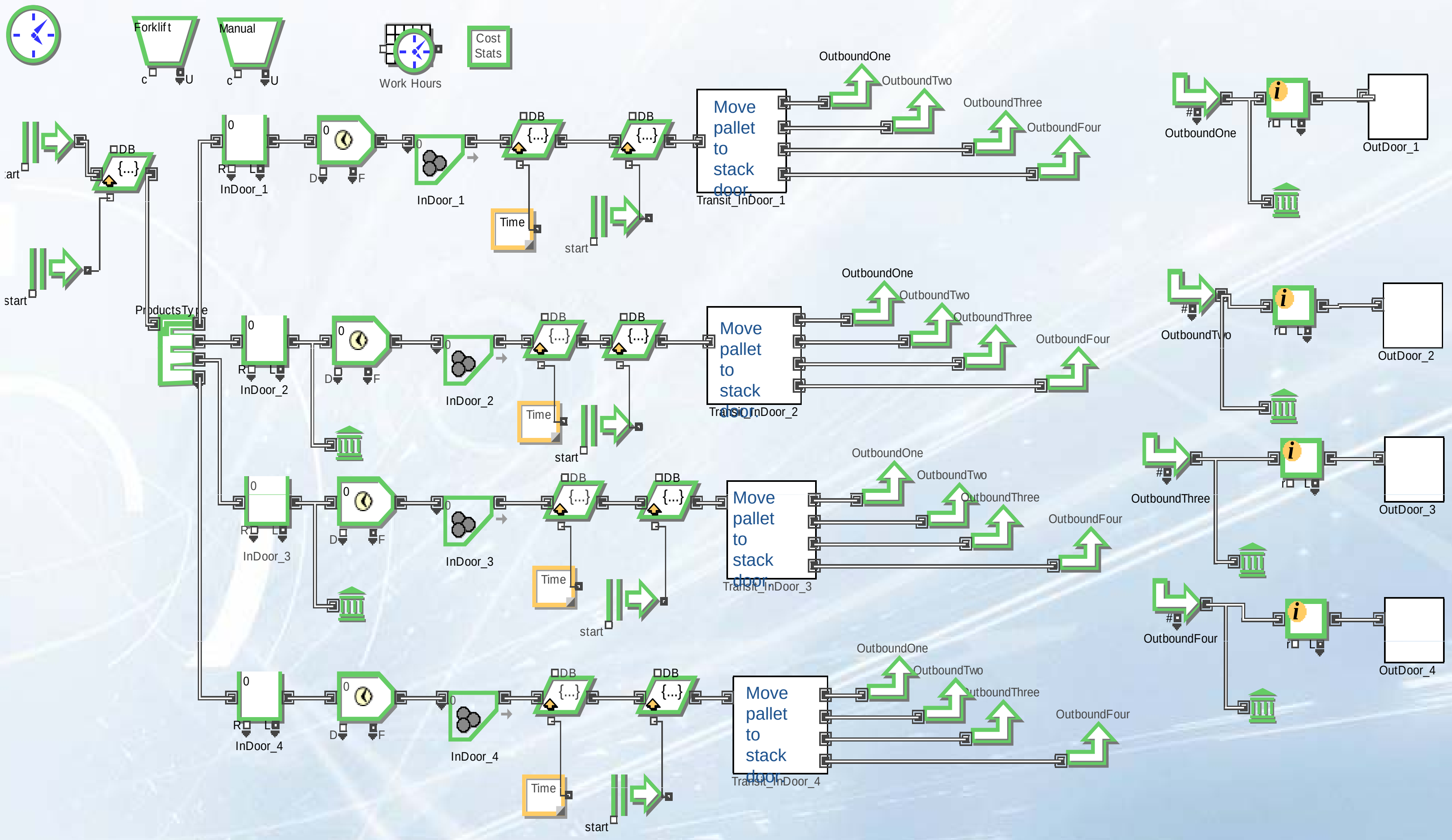
Distribution Center



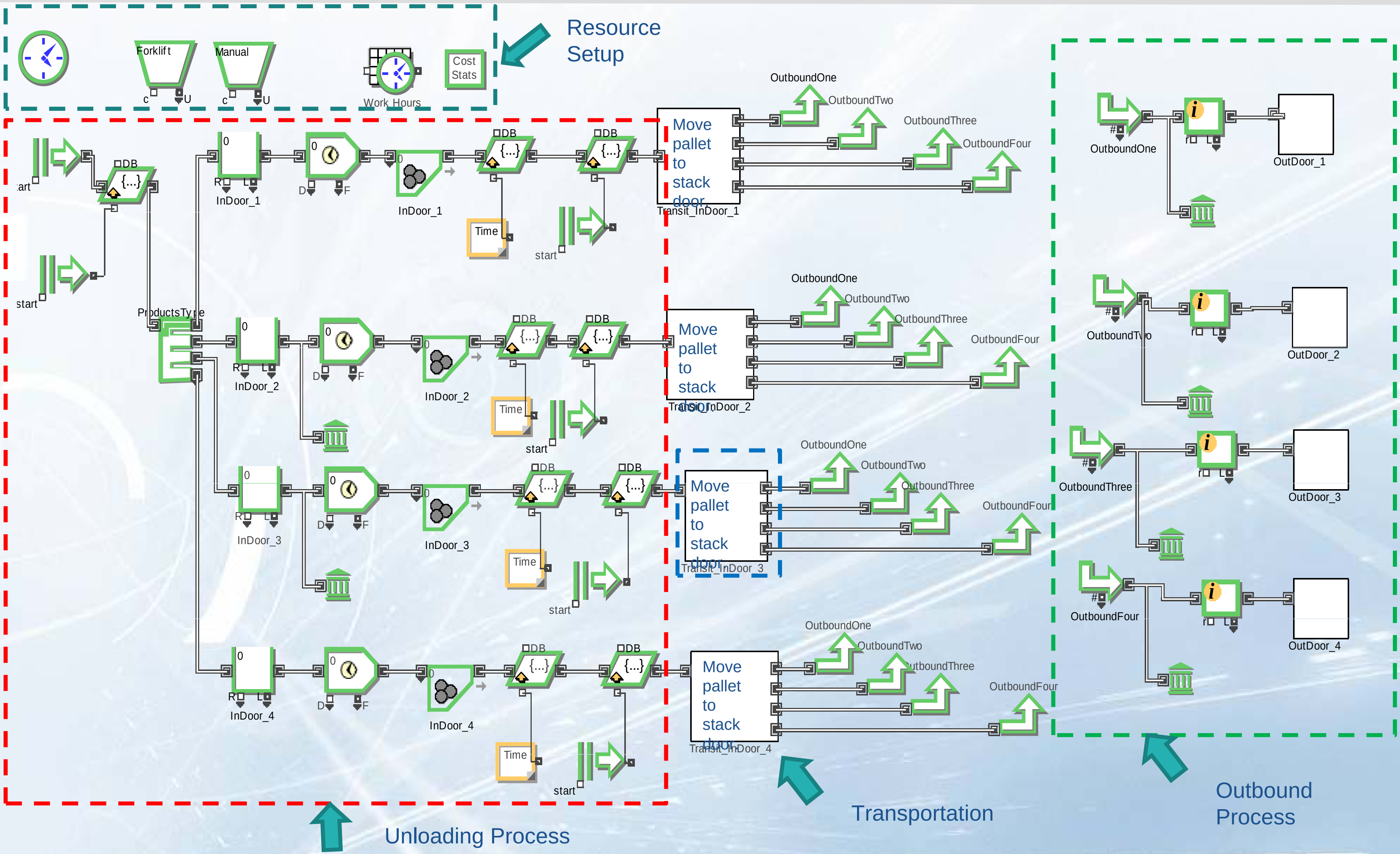
Features of the model

- Limits number of manual and motor forklifts.
- Labels each pallet by destination.
- Calculates the cost through moving time.
- Sets up trailer capacity; introduces a spare trailer when first one is fully loaded.
- Allows breaks in the work schedule.

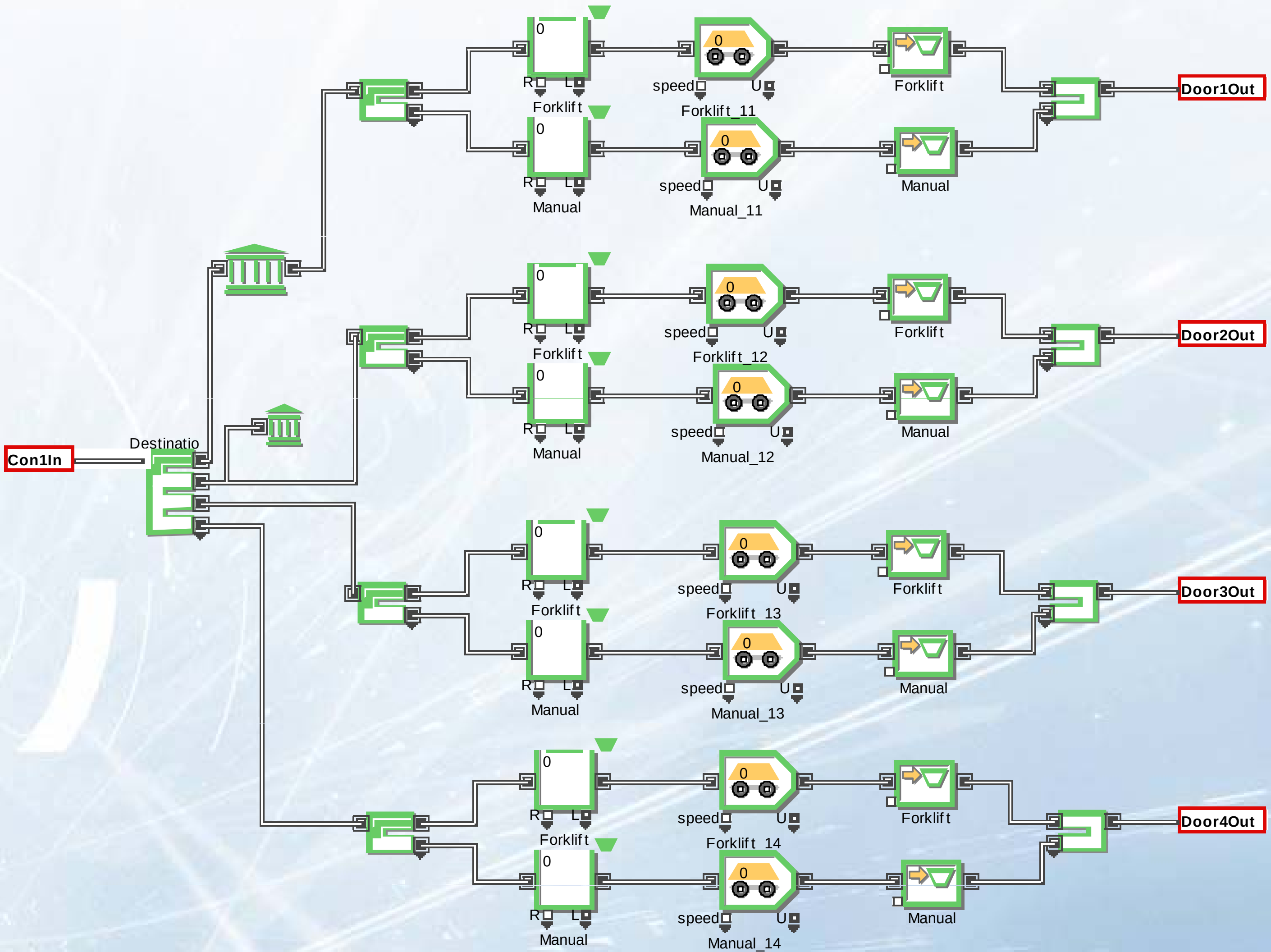
Model in Simulation



Model in Simulation

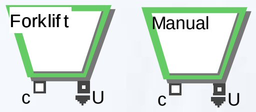


Transportation Details

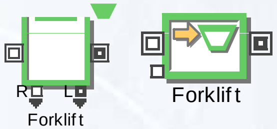


Key Block Function(i)

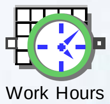
- Resources Blocks:



A tank contains work resource, “Forklift” or “Manual”.

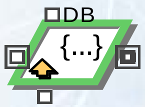


represent the “Resource Using & Releasing” process.



controls daily work schedule.

- Property Blocks:

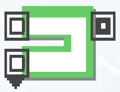


is used to label designed properties, such as “destination”, “current time” etc.

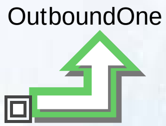
Key Block Function(ii)

- Routing Blocks:

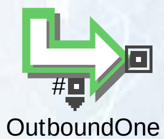
is used to select different choices, such as “ the route to the destination”.



is used to merge all the items from different routes.

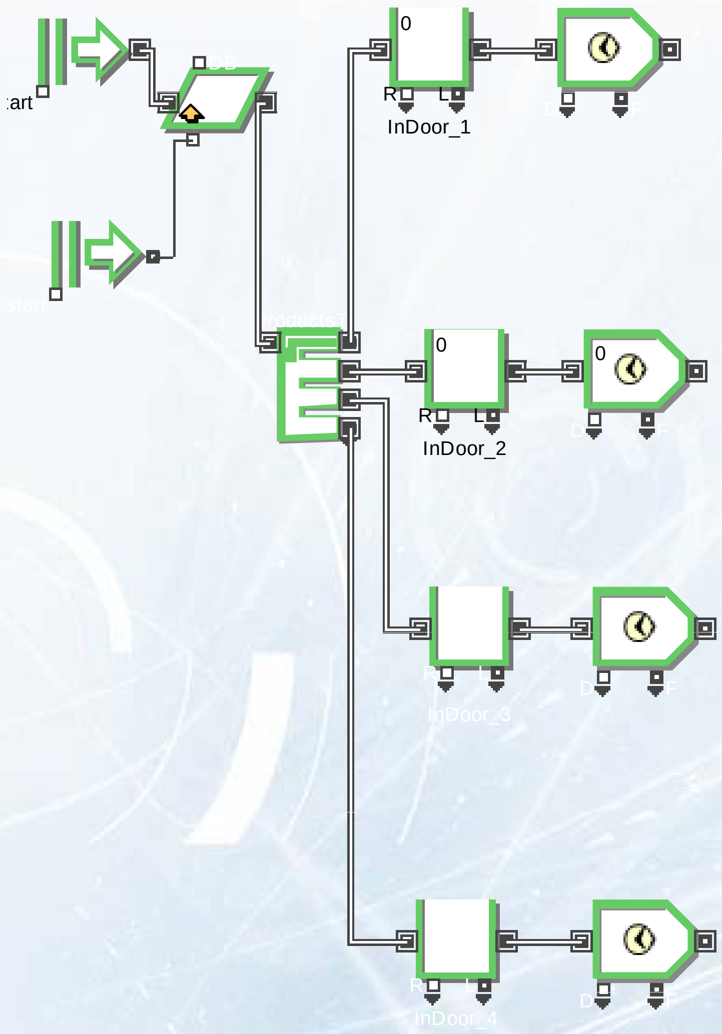


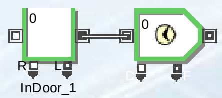
is used to choose one outbound door.



is used to receive all pallets from different inbound doors.

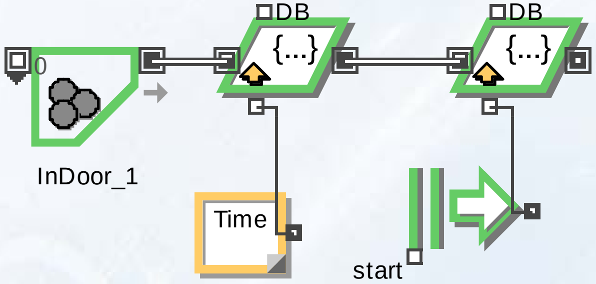
Part One: Unloading Setup



1. Main functions include Control start and end time, Generate products as scheduled, Unload products from incoming trailers
2. The block  is used to generate products.
3. The block 

   represents the incoming trailer and the entire unloading process.

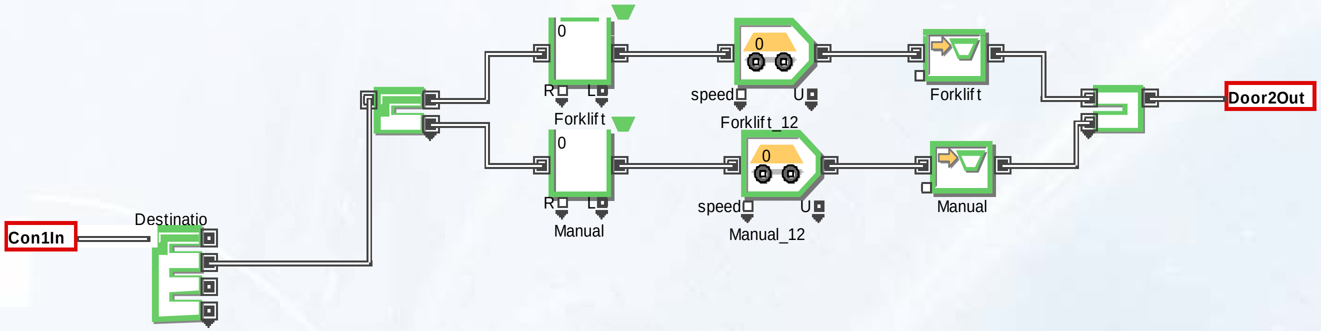
Batching Pallets

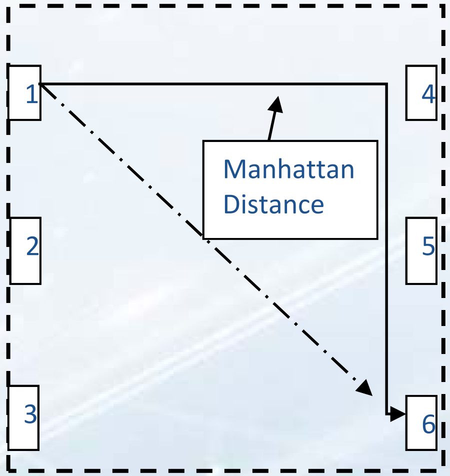


Batch products into pallets with different ratios according to specific item; and label pallets with destinations and current time.

<u>TV: 3:1; Fridge: 2:1; Clothes: 10:1; Food: 5:1.</u>

Part Two: Transportation Setup

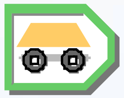




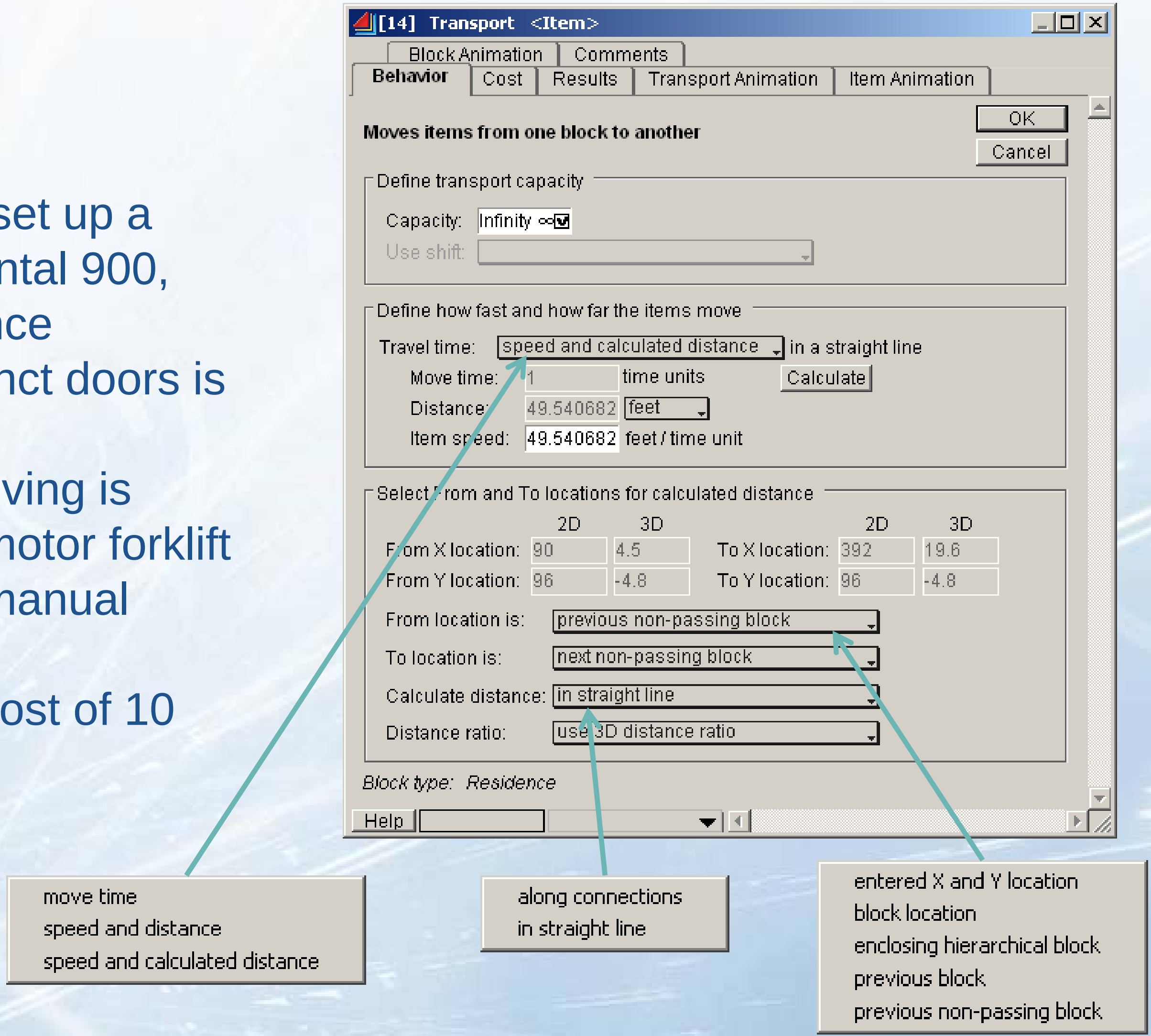
1. Main functions include Route in Manhattan distance, Control the pallet speed, Choose the moving type (manual or motor forklift).

2. Additional Function includes Calculate the cost by unit(10/time unit) as pre-setup.

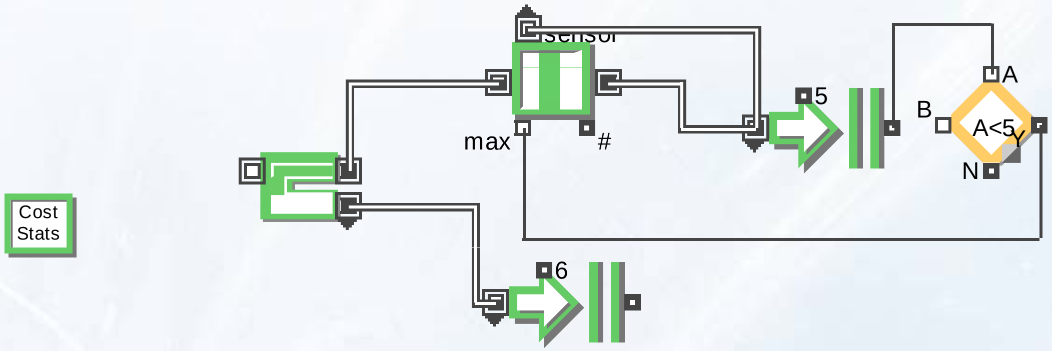
Transportation



1. For our work, we set up a distance of horizontal 900, and vertical distance between two adjunct doors is 300.
2. The Speed for moving is 400/time-unit for motor forklift 200/time-unit for manual forklift
3. And we set up a cost of 10 per time unit



Part Three: Outbound Setup



1. Main functions include Merging products from different inbound doors to outbound doors, Using spare trailers.

2. Additional Functions include Calculate the total cost.

Data in the Simulation

INPUT DATA

Flow matrix (number of pallets, not boxes)

		Customer 1 (Macy's)	Customer 2 (Target)	Customer 3 (Costco)	Customer 4 (Walmart)
Supplier 1	(TV)	5	3	3	4
Supplier 2	(fridge)	2	4	6	3
Supplier 3	(clothes)	6	2	5	3
Supplier 4	(food)	2	1	3	1

Distance matrix

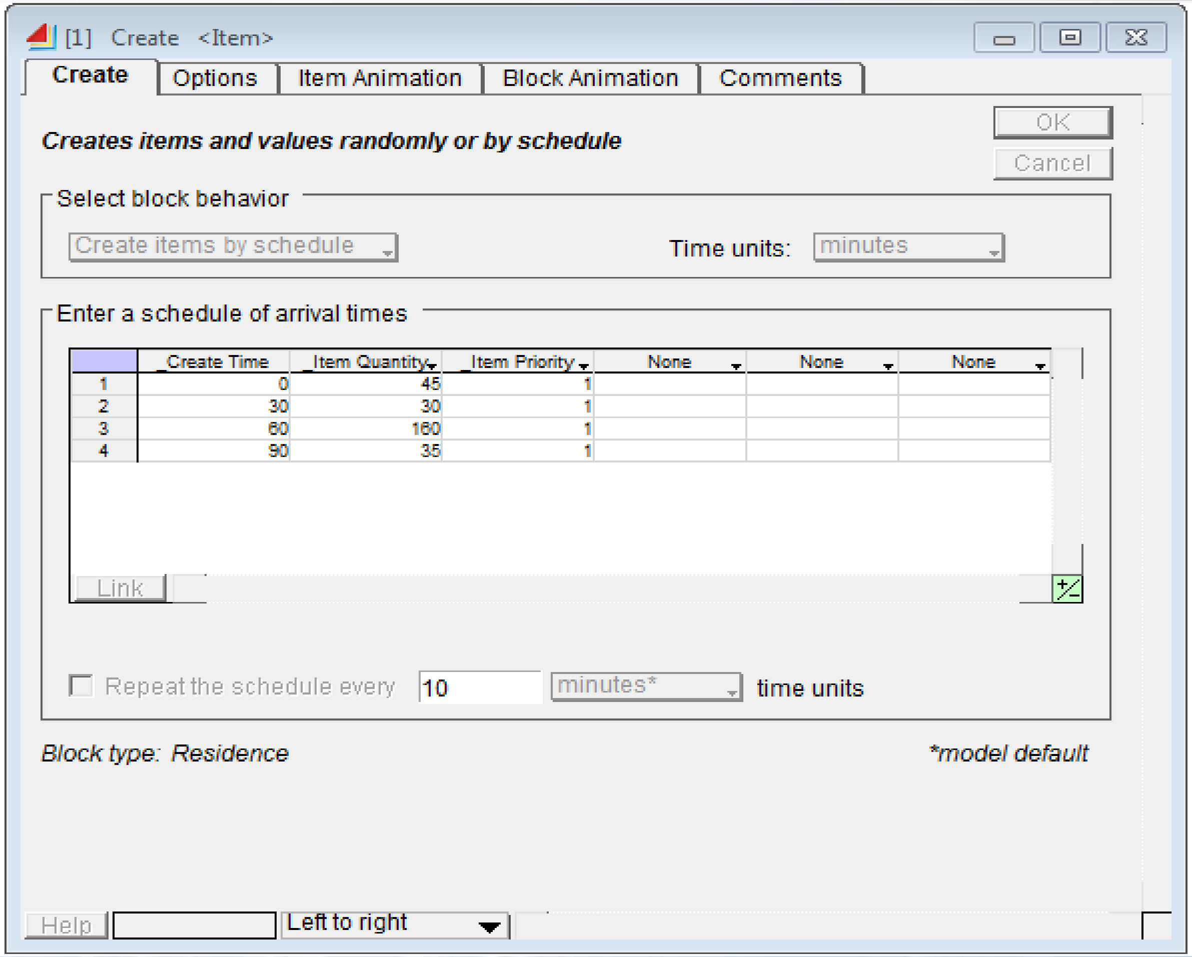
	outbound door 1	outbound door 2	outbound door 3	outbound door 4
inbound door 1	900	1200	1500	1800
inbound door 2	1200	900	1200	1500
inbound door 3	1500	1200	900	1200
inbound door 4	1800	1500	1200	900

Door capacity Maxmium 20 pallets per door

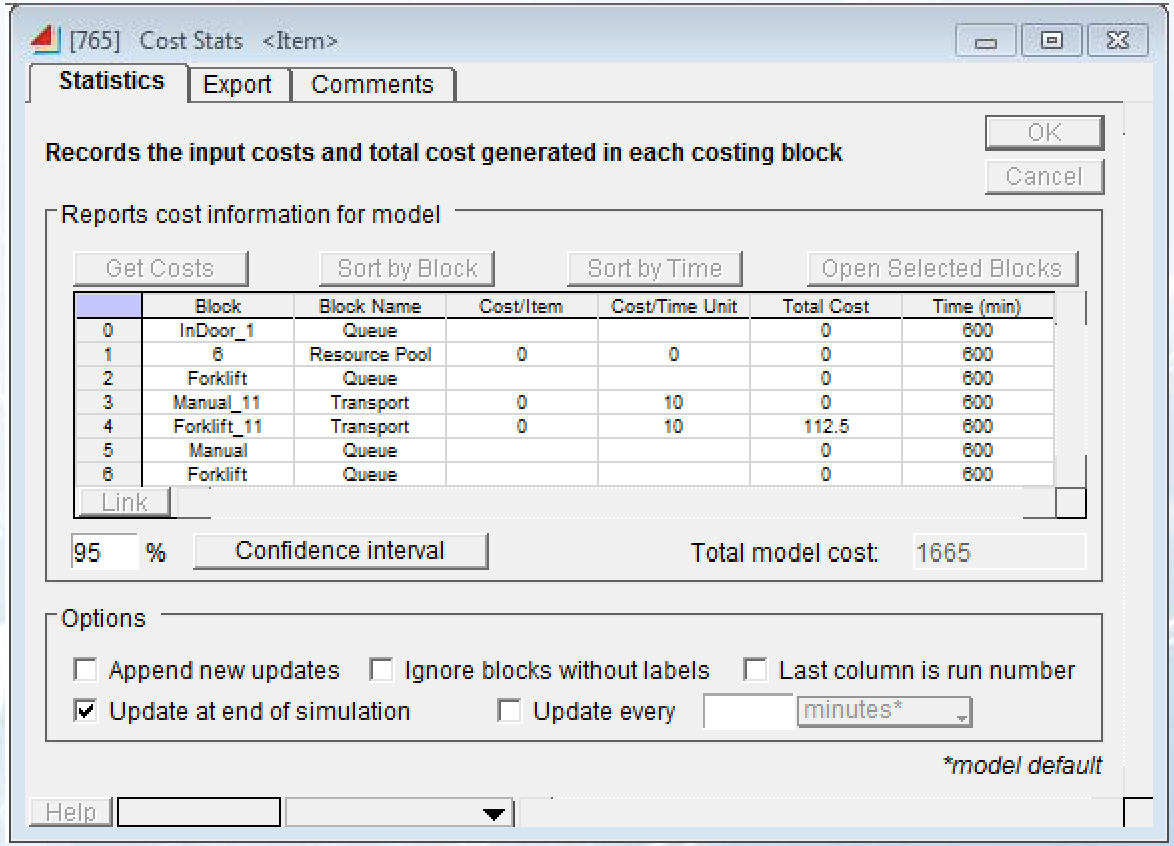
SOLUTION

Supplier 1	(TV)	>>>---->>>	inbound door 1	Cross-dock	outbound door 1	>>>---->>>	Customer 4 (Walmart)
Supplier 3	(clothes)	>>>---->>>	inbound door 2		outbound door 2	>>>---->>>	Customer 1 (Macy's)
Supplier 2	(fridge)	>>>---->>>	inbound door 3		outbound door 3	>>>---->>>	Customer 3 (Costco)
Supplier 4	(food)	>>>---->>>	inbound door 4		outbound door 4	>>>---->>>	Customer 2 (Target)

Simulation Setup and Results



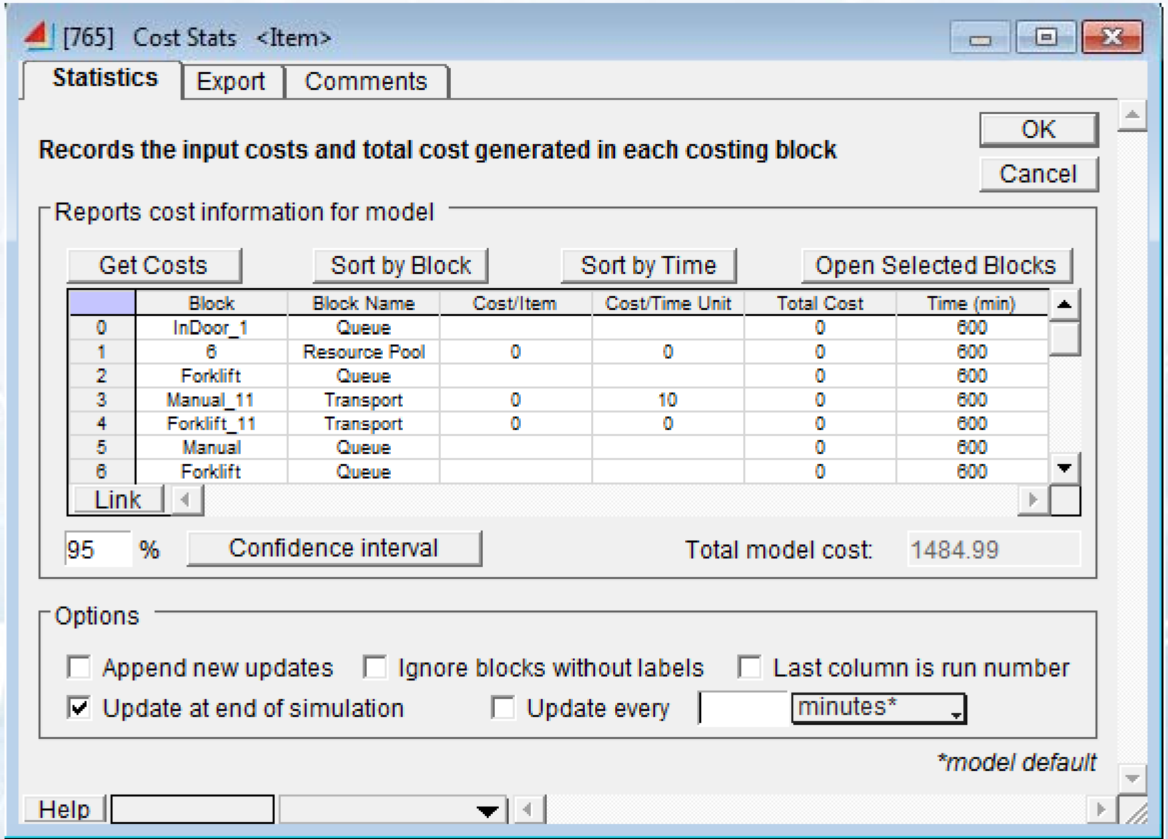
Simulation Setup and result



	Block	Block Name	Cost/Item	Cost/Time Unit	Total Cost	Time (min)
0	InDoor_1	Queue			0	600
1	6	Resource Pool	0	0	0	600
2	Forklift	Queue			0	600
3	Manual_11	Transport	0	10	0	600
4	Forklift_11	Transport	0	10	112.5	600
5	Manual	Queue			0	600
6	Forklift	Queue			0	600

Manual Door Assignment

Simulation Setup and result



[Optimized Door Assignment]