Mathematical Model for Cross Docking Systems without Temporary Storage

- 임시 보관 장소를 보유하지 않은 크로스도킹 시스템을 위한 수학적 모델 -

Yu Woo Yeon *

Abstract

크로스도킹이란 창고나 물류센터에 하역된 물품이 저장됨이 없이 도착지별로 재분류되어서 직출하되는 물류 시스템이다. 크로스도킹은 물류비용의 큰 비중을 차지하는 보관비용을 감소시킬 수 있을 뿐만 아니라 고객의 요구에 빠른 대응을 할 수 있다는 장점을 지니고 있다. 크로스도킹이 성공적으로 수행되기 위해서는 창고나 물류센터의 입고에서부터 출고까지의 모든 작업들이 계획적이고 원활하게 수행되어져야만 한다. 본 연구에서는 임시보관 장소를 보유하지 않은 크로스도킹 시스템의 총 운영시간을 최소화하기 위한 입고 트럭과 출고 트럭의 일정계획 수립을 위한 수학적 모델을 개발하였다.

KEYWORD : SCM, CROSS-DOCKING, OPTIMIZATION, Scheduling, MAKESPAN

1. INTRODUCTION

In today's distribution environment, customers are demanding better service, which translate into more accurate and timely shipments. Because it is difficult to ship directly from the manufacturers to the customer in most manufacturing environments, a certain type of intermediate points is necessary to connect between manufacturers and customers. One type of intermediate point in a supply chain system is the distribution center. Cross docking is a method of distribution management that helps companies better control their distribution operations.

* 삼성 SDS의 전자/제조 컨설팅팀

A cross docking operation involves multiple trucks (known as receiving trucks) that deliver products or items from suppliers to a distribution center and multiple trucks (known as shipping trucks) that ship items from the distribution center to customers. Based on customer demands, a receiving truck may have its items transferred to multiple shipping trucks. Similarly, a shipping truck can receive its consignments from multiple receiving trucks. A unique characteristic of a cross docking system is the absence or prohibition of long term storage of items at the distribution center. Items delivered to the distribution center from suppliers are shipped to customers as soon as possible without being placed in storage in the distribution center. The system can be operated with or without temporary storage. Ultimately, at the end of schedule period (e.g, one day), no item is left in the temporary storage. The advantage of cross docking is that the turnaround times for customer orders, inventory management cost, and warehouse space requirements are reduced.

The objective of this research is to find the best truck spotting sequence for both receiving and shipping trucks in order to minimize total operation time where a temporary storage area is not available in a distribution center. The allocations of the items from receiving trucks to shipping trucks are decided simultaneously as well as the spotting sequence of the receiving and shipping trucks

2. MODEL DESCRIPTION

Depending on the number of docks available at the distribution center, the dock holding pattern for trucks, and the existence of temporary storage, various cross docking scenarios or models can be generated (Yu, 2002). The cross docking system of this study is assumed to operate as follows:

1. Receiving trucks arrive at the receiving docks and unload products onto the receiving dock.
2. Products move from the receiving dock to the shipping dock on a conveyor.
3. Shipping trucks load products from shipping docks and leave shipping docks.

The cross docking system in this research does not consider the operation inside the warehouse or distribution center, such as scanning and sorting operations, etc. Therefore, the arrival sequence of the products at the shipping dock is the same as their unloading sequence at the receiving dock.

In the model studied in this research, it is assumed that there is no temporary storage in the warehouse or distribution center. However, both the receiving truck and the shipping truck can move in and out of the docks repeatedly during their tasks until their tasks are finished. Therefore, it is possible that a receiving truck unloads some of its products to the receiving dock, moves out, waits and goes into the receiving dock again to unload its remaining products. When the truck is out of the dock and waiting, another receiving truck could enter the dock to unload its products. This sequence can be similarly applied to the shipping truck. However, the conveyor connecting the receiving dock and the shipping dock may need to stop if a shipping truck is not available when a product arrives at the shipping dock. This is necessary because of the absence of a temporary storage.

In addition to the operation conditions above, the following assumptions are applied to the model.

1. All receiving and shipping trucks are available at time zero.
2. The total number of receiving products for each type of products is the same as the total number of shipping products for each type of products.
3. The following information is assumed to be previously known.
   i) Product types and the number of products loaded in a receiving truck.
   ii) Product types and the number of products needed for a shipping truck.
   iii) Loading and unloading times for the products.
   iv) Moving times of products from a receiving dock to a shipping dock.
   v) Delay time (i.e., truck change time) due to truck changes.

In a sequencing or scheduling problem, total operation time is often called makespan. In this research, makespan is defined as follows: Makespan is the total operating time of the cross docking operation. The total operating time is from the moment when the first product of the first scheduled receiving truck is unloaded onto the receiving dock to the moment when the last product of the last scheduled shipping truck is loaded from the shipping dock. The objective of this research is to find the best sequence for truck spotting for both the receiving and the shipping trucks to minimize total cross docking operation time or to maximize the throughput of the cross docking system. The product routing (i.e., the allocation of products from receiving trucks to shipping trucks) is also decided simultaneously as well as the spotting sequences of the receiving and shipping trucks.

3. MODEL DEVELOPMENT

In an operating condition assumed in this study, delay time occurs when the shipping truck changes or when the shipping truck is not loading any products from the shipping dock and waits for its needed products to arrive at the shipping dock. The change of receiving trucks at the receiving dock may cause the waiting of the shipping truck at the shipping dock. Therefore, both types of the delay times for this model are related to truck changes. From the above characteristics, it is evident that the makespan can be minimized if the number of truck changes is minimized. Minimizing the number of truck changes is equivalent to minimizing the number of matching pairs of the receiving trucks and shipping trucks. The receiving truck and the shipping truck are said to be paired if any product moves from the receiving truck to the shipping truck. Consequently, makespan can be minimized if the number of matching pairs of the receiving and shipping trucks is minimized.

To solve the cross docking problem for this model, two approaches were developed. For the first approach, a mixed integer programming model was developed with the objective of minimizing makespan of a cross docking operation. However, the use of mixed integer programming is not considered suitable for modeling the problem because of the exponential growth in variables and constraints as the number of receiving trucks, shipping trucks, and products increase.

The second approach also applied mathematical programming model using a different objective function. As mentioned earlier, minimizing the makespan is equivalent to minimizing the number of matching pairs of the receiving and shipping trucks for this model. Therefore, the second integer programming model was developed to minimize the number of matching pairs of the receiving truck and the shipping truck while product requirements are satisfied. By changing the objective of the mathematical model, the number of variables and constraints of the second integer programming model were drastically decreased in comparison with the first mixed integer programming model.

3.1 Mathematical Model I

For Mathematical Model I of the cross docking problem, it is assumed that unloading time from a receiving truck and loading time into a shipping truck are the same for all products and that it takes one unit of time in duration to unload or load one unit of product. Additionally, it is assumed that all operations can be carried out simultaneously. In other words, unloading operations from a receiving truck, loading operations into a shipping truck, or receiving and shipping truck changes can be carried out at the same time. With the above assumptions, the following mixed integer programming model was developed for the problem with the objective of minimizing makespan of a cross docking operation.

3.1.1 Notations

The following notations are used in Model I:

Continuous Variables:

T = Makespan,

U_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j start to unload from receiving truck i onto the receiving dock,

L_{ij} = Time at which the variable t_{ij} transferring from receiving truck i to shipping truck j finished loading from the shipping dock into shipping truck j,

Integer Variables:

x_{ijk} = Number of units of product type k which transfer from receiving truck i to shipping truck j,

t_{ij} = Total number of units of products which transfer from receiving truck i to shipping truck j, where $\left(t_{ij} = \sum_{k=1}^{N} x_{ijk}\right)$,

Binary Variables:

$$v_{ij} = \begin{cases} 1, & \text{If any products transfer from receiving truck } i \text{ to shipping truck } j \\ 0, & \text{Otherwise} \end{cases}$$

$$y_{iji'j'} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ immediately or directly precedes the variable } t_{i'j'} \text{ in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$y_{00i'j'} = \begin{cases} 1, & \text{If the variable } t_{i'j'} \text{ is placed at the first position in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

$$y_{ij00} = \begin{cases} 1, & \text{If the variable } t_{ij} \text{ is placed at the last position in the receiving or shipping sequence} \\ 0, & \text{Otherwise} \end{cases},$$

Data:

R= Number of receiving trucks in the set,

S= Number of shipping trucks in the set,

N = Number of product types in the set,

r_{ik} = Number of units of product type k which is initially loaded in receiving truck i,

s_{jk} = Number of units of product type k which is initially needed for shipping truck j,

D= Delay time for truck change,

V = Moving time of products from the receiving dock to the shipping dock,

M = Big number.

3.1.2 Mixed Integer Programming Model (Model I)

The mixed integer programming model for the cross docking problem with the objective of minimizing makespan of a cross docking operation is presented below.

Mixed Integer Programming Model I for the Cross Docking Problem

Min T

Subject to

$$T \geq L_{ij}, \quad for\ all\ i, j \tag{1}$$

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad for\ all\ i, k \tag{2}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad for\ all\ j, k \tag{3}$$

$$\sum_{k=1}^{N} x_{ijk} = t_{ij}, \quad for\ all\ i, j \tag{4}$$

$$t_{ij} \leq M\, v_{ij}, \quad for\ all\ i, j \tag{5}$$

$$v_{ij} = \sum_{i'=1}^{R} \sum_{j'=1}^{S} y_{iji'j'} + y_{ij00}, \quad for\ all\ i, j \tag{6}$$

$$v_{i'j'} = \sum_{i=1}^{R} \sum_{j=1}^{S} y_{iji'j'} + y_{00i'j'}, \quad for\ all\ i', j' \tag{7}$$

$$\sum_{i'=1}^{R} \sum_{j'=1}^{S} y_{00i'j'} = 1, \tag{8}$$

$$\sum_{i=1}^{R} \sum_{j=1}^{S} y_{ij00} = 1, \tag{9}$$

$$y_{ijij} = 0, \quad for\ all\ i, j \tag{10}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} - M\left(1 - y_{iji'j'}\right), \quad for\ all\ i, j, i', j'\ and\ where\ i = i' \tag{11a}$$

$$U_{i'j'} \geq U_{ij} + t_{ij} + D - M\left(1 - y_{iji'j'}\right), \quad for\ all\ i, j, i', j'\ and\ where\ i \neq i' \tag{11b}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} - M\left(1 - y_{iji'j'}\right), \quad for\ all\ i, j, i', j'\ and\ where\ j = j' \tag{12a}$$

$$L_{i'j'} \geq L_{ij} + t_{i'j'} + D - M(1 - y_{iji'j'}) \quad \textit{for all } i, j, i', j' \textit{ and where } j \neq j' \tag{12b}$$

$$L_{ij} \geq U_{ij} + V + t_{ij}, \quad \textit{for all } i, j \tag{13}$$

$$U_{i'j'} \geq L_{ij} - V - M(1 - y_{iji'j'}) \quad \textit{for all } i, j, i', j' \textit{ and where } i \neq i' \textit{ or } j \neq j' \tag{14}$$

all variables 0.

Constraint (1) ensures that makespan is equal to or greater than the time the last product is loaded onto the last scheduled shipping truck. Constraint (2) ensures that the total number of units of product type k that transfer from receiving truck i to all shipping trucks is exactly the same as the number of units of product type k which was initially loaded in receiving truck i. Similarly, constraint (3) ensures that the total number of units of product type k that transfer from all receiving trucks to shipping truck j is exactly the same as the number of units of product type k needed for shipping truck j. Constraint (4) defines the t_{ij} variables which is used in constraints (11) to (13) in order to calculate the unloading and loading times. Constraint (5) just enforces the correct relationship between the t_{ij} variables and the v_{ij} variables.

Constraint (6) ensures that only one of the t_{ij} variables can immediately or directly precede another $t_{i'j'}$ variable in the receiving or shipping sequence when $v_{ij}=1$. Constraint (7) ensures that only one of the $t_{i'j'}$ variables can immediately or directly follow another t_{ij} variable in the receiving or shipping sequence when $v_{i'j'}=1$. Constraint (8) ensures only one of the $t_{i'j'}$ variables can be placed at the first sequence position of the receiving or shipping sequence. Constraint (9) ensures that only one of the t_{ij} variables can be placed at the last sequence position of the receiving or shipping sequence. Constraint (10) ensures that there are no consecutive sequences that transfer products from the same receiving truck to the same shipping truck.

Constraint (11a) and (11b) make a valid sequence of unloading times of the t_{ij} variables, based on their order. If there is no receiving truck change between consecutive unloading sequence (in case of $i = i'$), constraint (11a) is applied. However, if there is a receiving truck change between the

consecutive unloading sequences (in case of $i \neq i'$), the delay time for receiving truck change must be considered, thus constraint (11b) is applied. Similar to constraints (11a) and (11b), constraints (12a) and (12b) make a valid sequence of loading times of the t_{ij} variables, based on their order. If there is no shipping truck change between the consecutive loading sequences (in case of j = j′), constraint (12a) is applied. However, if there is a shipping truck change between consecutive loading sequences (in case of $j \neq j'$), the delay time for shipping truck change must be considered, thus constraint (12b) is applied. Finally, constraints (13) and (14) establish the proper relationship between the unloading time and the loading time of the t_{ij} variables.
The number of decision variables for this integer programming model is *RS(RS+N+6)+1*. The decision variables consist of *RS(RS+3)* of binary variables, *RS(N+1)* of integer variables and *(2RS+1)* of continuous variables. The number of constraints is *3RS(RS+2)+N(R+S)+2* including *RS(3RS+2)* of inequality constraints and *(4RS+RN+SN+2)*of equality constraints.

3.2 Mathematical Model II

As mentioned ealier, minimizing makespan is equivalent to minimizing the number of matching pairs of the receiving and shipping trucks for this cross docking model. The objective of mathematical model II is to minimize the number of matching pairs of the receiving trucks and the shipping trucks while product requirements are satisfied. With the same assumptions as in Model I, Model II integer programming model was developed as follows. All notations used for Mathematical Model II are the same as in Model I.
The integer programming model with the objective of minimizing the number of matching pairs of the receiving truck and the shipping truck is presented below.

<u>Integer Programming Model II for the Cross Docking Problem</u>

Min $$\sum_{i=1}^{R}\sum_{j=1}^{S} v_{ij}$$

Subject to

$$\sum_{j=1}^{S} x_{ijk} = r_{ik}, \quad for\ all\ i,k \tag{15}$$

$$\sum_{i=1}^{R} x_{ijk} = s_{jk}, \quad for\ all\ j,k \tag{16}$$

$$x_{ijk} \le M\, v_{ij}, \quad for\ all\ i,j,k \tag{17}$$

all variables 0.

This mathematical model has two decision variables. The first decision variable, v_{ij}, is for pairing. It shows whether receiving truck i and shipping truck j are paired or not. If the variable v_{ij} equals one, it implies that some products transfered from receiving truck i to shipping truck j. The second decision variable, x_{ijk}, represents the number of units of product type k that transfers from receiving truck i to shipping truck j. In other words, the variable x_{ijk} shows product routing.

Constraints (15) and (16) are exactly the same as Constraints (2) and (3), respectively, of Model I. Constraint (17) enforces the correct relationship between the x_{ijk} variables and the v_{ij} variables, and ensures that there is a product transfer between a receiving truck and shipping truck if and only they are paired. The number of decision variables for this integer programming model is $RS(1+N)$, including RS of binary variables and RSN of integer variables. The number of constraints is $N(RS+R+S)$ and includes RSN of inequality constraints and $N(R+S)$of equality constraints.

One example problem was applied to Model II and solved by LINDO. The example problem has four receiving trucks, three shipping trucks, and seven product types. Information about each receiving truck and shipping truck is presented in Table 1. For this example problem, there are a total of 96 variables and 133 constraints. The solution obtained for the problem using Model II is presented in Table 2. Table 2 shows the matching pairs for the receiving and shipping trucks and the product routing between them. The minimum number of matching pairs required for this example is eight as shown in Table 2.

< Table 1. Example Problem to Illustrate Mathematical Model II >

Receiving **Truck**

Shipping Truck

Truck	Product	Quantity
1	1	150
	2	50
	3	50
2	4	200
	5	50
3	1	150
	3	150
4	6	100
	7	50

Truck	Product	Quantity
1	1	100
	3	50
	5	50
2	2	50
	4	200
	6	100
3	1	200
	3	150
	7	50

< Table 2. Number of Matching Pairs and Product Routing generated by Mathematical Model II for Example Problem >

Receiving Truck	Shipping Truck	Product Type	Number of Produc
1	2	2	50
1	3	1	150
		3	50
2	1	5	50
2	2	4	200
3	1	1	100
		3	50
3	3	1	50
		3	100
4	2	6	100
4	3	7	50

4. CONCLUSION

The cross docking model studied in this research assumes that there is no temporary storage in the warehouse or distribution center. However, both the receiving truck and the shipping truck can move in and out of the docks as many times as needed until their unloading or loading tasks are completed. To solve the cross docking problem, two approaches were developed; a mixed integer programming model and an integer programming model.

The objective of the first mixed integer programming (Model I) model is to minimize makespan. For the Model I model, the number of variables and constraints grow exponentially as the number of receiving trucks, number of shipping trucks, and number of product types increase. Computationally, the approach is not effective for solving the test problems, including the smallest one. A different view point to the problem led to the development of a second integer program model (Model II).

Model II is relatively simple. The objective of the integer programming model or Model II is to minimize the number of matching pairs of the receiving and shipping trucks. A receiving truck and a shipping truck are said to form a matching pair if there is material exchange between them. With the objective of minimizing the number of matching pairs, the number of variables and constraints are dramatically decreased.

5. References

[1] Cooke, James Aaron (1995), Do you have what it takes to Cross Dock?, *Logistics Management*, September, 47-50.

[2] Rohrer, Matthew (1995), Simulation and Cross Docking, *Proceedings of the 1995 Winter Simulation Conference*, 846-849.

[3] Schaffer, Burt (1998), Cross Docking can Increase Efficiency, *Automatic I.D. News, 14(8) July*, 34-37.

[4] Schwind, Gene F (1995), Considerations for Cross Docking, *Material Handling Engineering, 50(12) November*, 47-51.

[5] Schwind, Gene F. (1996), A Systems Approach to Docks and Cross Docking, *Material Handling Engineering, 51(2) February*, 59-62.

[6] Witt, Clyde E. (1998), Crossdocking: Concepts Demand Choice, *Material Handling Engineering, 53(7) July*, 44-49.

[7] Yu, Wooyeon (2002) Operational Strategies for Cross Docking Systems, *Iowa State University, Ph.D Dissertation.*

저 자 소 개

유우연: 현재 삼성 SDS의 전자/제조 컨설팅팀에서 근무중이다. 명지대학교 산업공학과 학부를 졸업후(1992) 아이오아 주립대학교 산업공학과에서 석사(1997) 와 박사(2002) 학위를 받았다. 그의 관심 분야는 물류와 공급망사슬 그리고 IT 관련 분야이며 현재 국내 유수기업을 대상으로 하여 상기 분야에 대한 컨설팅을 수행중이다.