Scheduling Problem of Receiving and Shipping Trucks for Cross Docking Systems

-크로스도킹시스템을 위한 하역과 선적 트럭의 일정계획-

유 우 연*
Yu Woo yeon

Abstract

Cross docking is a material handling and distribution concept in which products move directly from receiving dock to shipping dock, without being stored in a warehouse or distribution center. Depending on the facility and operating conditions or strategies employed, it is possible to generate various cross docking scenarios or models. The cross docking model, which is studied in this research, assumes there are a separate receiving dock and a separate shipping dock. It is also assumed that the products contained in a receiving truck and the products needed for a shipping truck are known in advance. Furthermore, the study is restricted to scenarios where there is only one receiving dock and only one shipping dock at the warehouse. The research objective is to find the best truck spotting sequence for both receiving and shipping trucks to minimize total operation time (i.e., the makespan) of the cross docking system.

크로스도킹이란 물류와 분배의 개념으로서 창고나 분배 센터에서 물품이 재고로 남겨짐이 없이 곧바로 하역 창구에서 적재 창구로 이동 되어지는 것을 말한다. 창고시설이나 운영 조건이나 운영 전략에 의해서 다양한 크로스도킹 모델이 생성될 수 있다. 본 연구에서 고려 되어지는 크로스도킹 모델은 하나의 독립된 하역 창구와 독립된 적재 창구를 가정하고 있다. 또한 하역 트럭에 적재되어 있는 물품이나 적재 트럭에 실려질 물품의 수와 종류는 사전에 알려져 있다는 것을 가정한다. 또한 고려되어진 창고나 분배창구는 단지 하나의 하역 창구와 하나의 적재 창구를 가지고 있다고 가정된다. 본 연구의 목적은 고려되어진 크로스도킹 시스템의 총 운영 시간을 최소화 하기 위한 하역 트럭과 적재 트럭의 일정 계획을 찾는데 있다.

* 삼성 SDS 컨설팅 Div.

1. Introduction

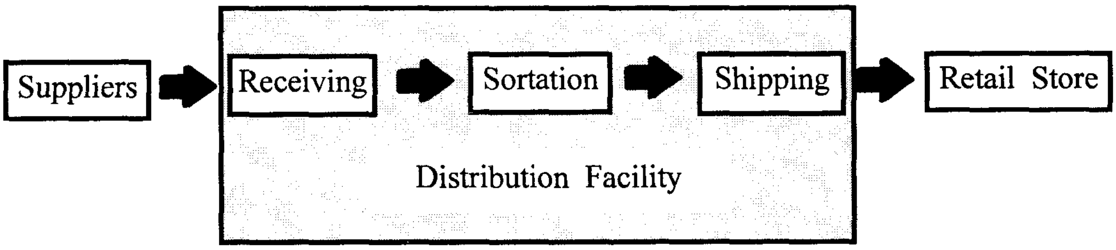
A typical warehouse is a dynamic and intelligent distribution center in which products and packages are processed in real time and moved in and out on schedule. A dynamic and intelligent warehouse is also a place where all distribution and logistic functions are tied together and where inventory storage is minimal. Operations of the distribution center consist of five basic functions: receiving, sorting, storing, picking and shipping. If the way these five elements cooperate is improved, costs can be reduced and productivity can be improved. However, the best way to reduce cost and improve efficiency is not by simply improving a function but by eliminating it if feasible. Cross docking has the potential of eliminating storage and picking, the two most expensive warehousing operations.

Cross docking is a material handling and distribution concept in which products move directly from receiving dock to shipping dock, without being stored in a warehouse or distribution center. In general, it seems that cross docking works best for companies either distributing large volumes of merchandise or serving a large number of stores. Cross docking systems handle a high volume of products in a short amount of time. The advantages of cross docking systems include increased inventory turn, thus reduced inventory, increased customer responsiveness, and better control of the distribution operation.

Figure 1 shows the flow of material in a typical cross docking operation [4]. As shown in Figure 1, the cross docking system generally operates as follows:

1. Products (packages, boxes, cartons, etc.) arrive on the receiving docks.
2. Products are scanned and verified at the receiving docks. In some cross docking systems products are also weighed, sized and labeled at the receiving dock.
3. Products are placed on the sortation systems, which sort by destinations.
4. Products are processed to the proper location on the shipping docks.

Figure 1. Typical Cross Docking Flow [4]



2. Literature Review

In many warehouses or distribution centers, cross docking systems are implemented successfully. Nevertheless, relatively few research papers or articles are published on cross docking systems. As a result, no systematic or generally accepted approach for planning cross docking operation has emerged.

The first and only technical paper found on cross docking systems was presented by Rohrer [4]. He discussed modeling methods and issues as they apply to cross docking systems. When a cross docking system is considered, there are several points to remember. Schwind [6] discussed considerations for cross docking systems. A few articles in trade magazines report the successful implementation of cross docking systems. Forger [2] discussed the success of cross docking operation of Chicago Area Consolidation Hub (CACH) of UPS. He explained how cross docking applied to CACH. Some considerations on the equipment and procedures for cross docking success are presented in Schwinds article [7]. The concepts of cross docking are found in many articles. Among these articles, Witts article [8] presented the concepts of cross docking most thoroughly. Schaffer [5] explained the requirements for successful cross docking.

Drawing from published work, it is clear that cross docking systems can greatly reduce inventory, shorten the product flow time between the manufacturer and the customer, and produce better control of the distribution operation. To implement cross docking successfully, the appropriate software should be developed as well as the hardware. Nevertheless, there is no reported research on these aspects of cross docking system. This lack of research motivated the study undertaken in this research.

3. Model Descriptions

In this research, the following cross docking system is considered. The cross docking system of this study is operated 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 other words, the order in which the products are unloaded at the receiving dock is the same as the order they arrive at the shipping dock.

In the cross docking model studied in this research, there is temporary storage in front of the shipping dock. If a product that arrives at the shipping dock does not need to be loaded into shipping truck currently at the dock, the product can be stored in the temporary storage until the appropriate shipping truck comes into the shipping dock. In this model, both the receiving and the shipping trucks must stay in docks until they finish their task once they come into docks. Therefore, a receiving truck cannot leave the receiving dock until all of its products are unloaded onto the receiving dock. Similarly, a shipping truck cannot leave the shipping dock until all of its needed products are loaded.

The objective of this research is to find the best truck spotting sequence for both the receiving and the shipping trucks to minimize total cross docking operation time (i.e. makespan) or to maximize the throughput of the cross docking system. 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.

3.1 Assumptions

The following main assumptions are made in this research.

1. All receiving and shipping trucks are available at time zero.
2. All products received must be shipped. Long term storage is not allowed.
3. 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.
4. The unloading sequence of the products from a receiving truck can be determined.
5. It can be unloaded only the necessary amount of products from a receiving truck. In other words, any products loaded in a receiving truck are accessible so that only the necessary amount of products can be unloaded from a receiving truck.
6. Only one unit of a product can be loaded into the shipping truck at a time. Therefore, loading products simultaneously from a receiving truck and the temporary storage into a shipping truck is prohibited.
7. The operations inside the warehouse or distribution center such as scanning and sorting operations are not considered. Therefore, the arrival sequence of the products at the shipping dock is maintained as the unloading sequence of the products at the receiving dock.
8. The capacity of temporary storage is unlimited. Therefore, it can be assumed that the capacity of the temporary storage can be as high as the total number of products presented in a model.
9. 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.

Table 1 shows the information required for the cross dockingproblem. This example set has three receiving trucks, four shipping trucks and nine product types.

Table 1. Information Required for Cross Docking Problem

Truck	Product	Quantity
1	1	64
	2	58
	3	19
	4	38
	5	19
	6	58
	7	19
	8	7
	9	58
2	5	132
	9	118
3	2	49
	7	97
	8	49
	9	145

Receiving Truck

Truck	Product	Quantity
1	4	38
	6	29
	8	28
2	5	151
	6	10
	7	12
	8	28
3	2	41
	6	19
	7	61
	9	229
4	1	64
	2	66
	3	19
	7	43
	9	92

Shipping Truck

4. Model Development

In this model of the cross docking problem, there are two types of delay times. The first type of delay time occurs when there is a shipping truck change. The second type of delay time occurs when the shipping truck currently in dock does not load any products from a certain receiving truck in dock or temporary storage, and waits until its needed products arrive at the shipping dock. The change of receiving trucks at the receiving dock or the unloading products from a receiving truck to temporary storage may cause the second type of delay time. For this problem, the first type of delay time is the same regardless of the shipping truck spotting sequences because all shipping truck sequences have the same number of shipping truck changes which is *(S-1)*, where *S* represents the total number of shipping trucks in the problem. Similarly, the number of receiving truck changes are the same regardless of the receiving truck sequences. Therefore, the only factor, which can affect makespan, is the number of unloaded products from receiving trucks that transfer to the temporary storage. If the number of products sent to temporary storage decreases, the waiting time of the shipping truck at the shipping dock may decrease, thus makespan may decrease.

From the above characteristics, it can be seen that makespan will be minimized if delay time or idle time is minimized. For this cross docking problem, the main factor that causes idle time is the number of products passing through temporary storage. Therefore, minimizing the number of products passing through temporary storage seems to be a good strategy for minimizing makespan. To solve the cross docking problem, two approaches were developed; branch and bound method and tabu search method. Both approaches find a solution based on the minimization of the total number of products passing through temporary storage.

4.1 Branch and Bound Method

Branch and bound is a useful method for solving many combinatorial optimization problems. As its name implies, the approach consists of two fundamental procedures. Branching is the process of partitioning a large problem into two or more subproblems, and bounding is the process of calculating a lower bound on the optimal solution of a given subproblem (Baker, 1974).

The branching procedure replaces an original problem by a set of new problems that are
(i) mutually exclusive and exhaustive subproblems of the original problem,
(ii) partially solves versions of the original problem, and
(iii) smaller problems than the original.
Furthermore, the subproblems can themselves be partitioned in a similar

fashion.

For the cross docking problem, the total number of products passing through the temporary storage is calculated in each node. In the typical scheduling problem, only one schedule needs to be made for the problem. However, in the cross docking problem, two sequences need to be made; one is the receiving truck sequence and the other is the shipping truck sequence. Therefore, the typical branch and bound method needs to be modified to construct two sequences.

For the cross docking problem, the following branch and bound method is suggested.

1. Set $\pi = \infty$.
2. Solve the problem using branch and bound and compute the total number of products, σ, that pass through the temporary storage at each node based on the solution up to that node.
3. Compare the total number of products passing through the temporary storage, σ, with the current best solution, π.
   a) If σ is equal to or greater than π, fathom the node.
   b) If the node is a leaf node and σ is lower than the current best solution, π, update the current best solution, $\pi \leftarrow \sigma$.
4. Continue until all nodes are fathomed.

The nodes indicate the trucks being scheduled. The sequence of the nodes indicates the order in which the trucks are scheduled at the docks. The following branching strategy was developed for the cross docking problem.

1. The first node branched from the root node is used for the shipping trucks.
2. If all products for the last scheduled shipping truck are filled from the scheduled receiving trucks, the next branch will be used for a shipping truck. Otherwise, the next branch will be used for a receiving truck.

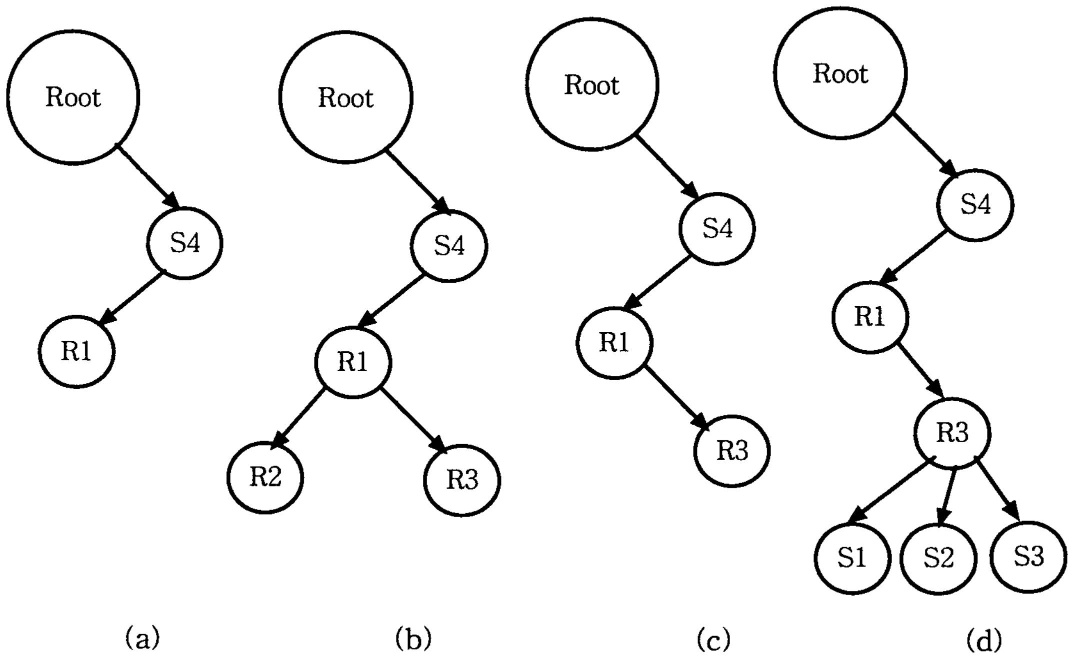


Figure 2. Explanation of Branching Strategy for the Cross Docking Problem

In order to explain the above strategy, consider the problem set in Table 1. Suppose node *R1* as shown in *(a)* of Figure 2 needs to be branched from at a given iteration. From the branch and bound tree, it can be seen that receiving truck 1 and shipping truck 4 are scheduled up to this point. At this point, the shipping truck 4 does not load all of its needed products yet. In other word, the shipping truck 4 needs to load more products after it loads products from receiving truck 1. Therefore, the tree will branch out next on the remaining receiving trucks again as shown in *(b)* of Figure 2. Now, suppose node *R3* as shown in *(c)* of Figure 2 needs to be branched in a given iteration. Then, the shipping truck 4 is ready to leave the shipping dock because it has loaded all of its needed products from receiving truck 1 and receiving truck 3. Therefore, the system will branch out next on the remaining shipping trucks as shown in *(d)* of Figure 2. The remainder of the branching process continues in a similar fashion until all nodes are fathomed and the optimal solution is found.

4.2 Tabu Search

The tabu algorithm is a search technique aimed at building extended neighborhood procedures, with particular emphasis on avoiding being caught in a local optimum. The idea of the tabu search is quite simple. In a tabu search, the best move available is always taken, even if this makes the objective value somewhat worse. This is basically a diversification move, because intensification is momentarily of no advantage. Now, if the move gets out of the local optimum on the very next move, the objective can probably be decreased the most by moving right back to the same local optimum. Therefore, the search has to be forced to continue diversifying for a few moves. The approach that the tabu search employs to prevent returning to the same local optimum is to keep a list of the last m moves and not to allow moves in the list to be repeated while they remain on the list (they are currently "tabu"). Glover gave the number of moves, m, in the list typically to be set equal to 7 (i.e., $m = 7$).

Tabu search for the cross docking problem was developed to minimize the number of products that pass through temporary storage. Therefore, the solution of the tabu search is presented as the number of products that pass through temporary storage. Tabu search for the cross docking problem used the following basic elements of the neighborhood search procedure:

1. *Initial Seed* - There are a number of ways of obtaining the initial seed. The initial seed is randomly picked for the cross docking problem.
2. *Neighborhood of the Current Solution* - The adjacent pairwise interchange operation is used to generate a neighborhood of a current solution.
3. *Selection Criterion* - To select the next solution after the adjacent pairwise interchange, all solutions in the neighborhood are evaluated and the best solution among all neighborhood solutions is chosen as the next solution even if this makes the objective function value somewhat

worse.

4. *Termination* - Tabu search will stop if there is no improvement of the objective for a maximum number of iterations specified by the user. In other words, if a certain number of consecutive solutions do not improve the current best solution, the algorithm will stop. For the cross docking problem, 1000 were used as the maximum number of iterations.

5. *Number of Tabu List* - As Glover suggested, a list seven tabu points were used for this algorithm.

In order to explain the tabu search algorithm for the cross docking problem, the following notations are used:

i = Number of iteration,

K = Maximum number of iterations allowed which was set by a user,

T^C = Ordered set of the current receiving and shipping truck sequences,

T = Best neighborhood of the current receiving and shipping truck sequences,

T^* = Ordered set of the best receiving and shipping truck sequences,

The tabu search algorithm used for the cross docking problem is as presented below.

TABU SEARCH ALGORITHM FOR THE CROSS DOCKING PROBLEM

STEP 1

Generate the initial receiving and shipping truck sequences randomly. Set the current sequence as the initial sequence; (i.e. set T^c).

STEP 2

Select the next receiving and shipping truck sequences as follows:

(2a) For each adjacent neighborhood sequence of the current sequence, if there is the same sequence of the neighborhood sequence in the tabu list, do not consider the neighborhood sequence as the next sequence. Otherwise, calculate the total number of products that pass through temporary storage for the neighborhood sequences of the current

sequence.

(2b) Among all neighborhood sequences considered in *Step (2a)*, choose the next receiving and shipping truck sequences as the neighborhood sequence that has the smallest total number of products that pass through temporary storage; (i.e. Choose T).

STEP 3

Set the current sequence as the next sequence; (i.e. set $T^c \leftarrow T$).

STEP 4

If the current sequence is the best solution found so far, set the best sequence as the current sequence (i.e. $T^* \leftarrow T^c$) and set the number of iteration to 1; (i.e. set $i = 1$). Go to *Step 2*. Otherwise, increase the number of iterations by 1; (i.e., set $i \leftarrow i+1$).

STEP 5

If the number of iteration is greater than the maximum number of iteration (i.e. $i>K$), stop. Choose the best sequence as the best solution found so far. The best receiving and shipping truck sequences are found. Otherwise, go to *Step 2*.

5. Implementation and Results

Ten sets of test problems were randomly generated to test the performances of the tabu search. The range for the number of receiving trucks and shipping trucks is three to five, respectively. Total numbers of products in the test set are between 890 and 1030 units. Table 2 shows the detailed information about the ten test problem sets.

After applying the tabu search to the ten sets of test problems, the solutions were obtained and they are as presented in Table 3. Because the tabu search can possibly find different solutions to a given problem based on the initial receiving and shipping truck sequences specified, the tabu search was run ten times for each test problem to test the quality of tabu solution and reduce

Table 2. Detailed Information for the Ten Test Problem Sets

Problem Number	Number of Receiving Truck	Number of Shipping Truck	Number of Product Types	Total Number of Products
1	4	5	4	990
2	5	4	6	1030
3	3	3	8	890
4	5	5	8	1000
5	5	3	8	960
6	4	4	5	1020
7	5	4	6	980
8	3	5	7	890
9	4	4	8	900
10	3	4	9	930

Table 3. Tabu Search Solutions for the Cross Docking Problem

Problem Number	Optimal Solution	Worst Solution	Best Tabu Solution	Average Tabu Solution	Worst Tabu Solution
1	93	645	93	115.1	164
2	147	772	165	178.5	203
3	233	461	233	233.0	233
4	265	792	286	343.1	427
5	180	546	217	247.1	327
6	151	789	151	157.8	219
7	127	681	127	140.3	158
8	241	553	241	266.4	352
9	204	611	204	222.0	265
10	155	611	155	175.8	259

the chances of obtaining inferior solution or local optimum. Each run employed a different starting solution that was generated randomly.

Table 3 shows the global optimal solutions obtained from the branch and bound method, the worst solution for the problem set obtained from the enumeration method, the best tabu search solutions, the worst tabu search solutions and the average tabu search solutions of ten runs. The solutions are presented as the total number of products that pass through temporary storage. When the best tabu search solutions are examined, the tabu search found the optimal solution in seven out of the ten test sets. As presented in Table 3, the tabu search performed well for the cross docking problem.

6. Conclusions

Cross docking has many advantages such as reducing inventory and improving customer responsiveness etc. Nevertheless, there is no reported research on the cross docking problem. In this paper, two solution approaches were developed to solve the cross docking problem; the branch and bound method and the tabu search method.

The first approach suggested for the cross docking problem was the branch and bound method. For the branch and bound approach, the new branching strategy was developed in order to make two separate sequences; one is for a receiving truck sequence and the other is for a shipping truck sequence. The use of a good starting upper bound will help to reduce the computational time required to solve and obtain the optimal solutions to the problems. The second approach is the tabu search. Comparison of the solutions obtained from the tabu search with the optimal solutions indicated that the tabu search solutions were close to the optimal solutions.

7. References

[1] Baker, R. Kenneth (1974), *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York, USA.

[2] Forger, Gary, (1995), UPS starts Worlds Premiere Cross-Docking Operation, *Modern Materials Handling, November*, 36-38.

[3] Glover, Fred and Laguna, Manuel (1997) *Tabu Search*, Kluwer Academic Publishers, Boston, USA.

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

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

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

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

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

저 자 소 개

유우연: 명지대학교 산업공학과 학부를 졸업후 (1992) 아이오아 주립대학교 산업공학과에서 석사 (1997) 와 박사 (2002)학위를 받았다.
현재 삼성 SDS 컨설팅 Div. 에서 근무중이다.
주요 관심 분야는 물류, 공급망사슬 과 정보 기술 분야이다.
그는 현재 IIE, INFORMS, 대한 산업 공학회, 한국 경영 과학회와 안전 경영 과학회의 회원이다.