

© 2016, TextRoad Publication

Truck scheduling in a cross-dock system with multiple doors and uncertainty in availability of trucks

Mahdi Rajabi¹ and Mohsen Akbarpour Shirazi²

¹ M.Sc. student, Department of industrial engineering and managerial systems, Amirkabir University of Technology, ² Member of board of science, Department of industrial engineering and managerial systems, Amirkabir University of Technology

> Received: March 31, 2016 Accepted: May 20, 2016

ABSTRACT

Aim of this study was investigating a truck scheduling problem with uncertainty in both inbound and outbound problems. Scheduling authorities at cross-dock should consider these uncertainties and propose a scheduling & sequencing plan which is robust against these uncertainties. As we know, trucks don't necessarily assign to receiving or shipping doors and should wait a while till a proper dock door is free and ready to service. The objective of this study is to minimize these waiting times by proper assignment and sequencing trucks. This schedule should be ready before trucks enter the cross-dock terminal and thus we must use expected arrival times of trucks. And since these expected arrival times may differ from real arrival times of trucks, scheduling should be performed in a way to include these differences. If the difference between real and expected arrival time of a truck is less than its waiting time, no difference is made to the scheduling. But if the lag time is greater than waiting time, the schedule would be inefficient. As a result, it's necessary to schedule in a way that these waiting times be at most the size of difference between real and expected arrival times of trucks to make it useable and efficient. Moreover, this paper considers uncertainty in mathematical model of the problem by use of chance constraints to consider uncertainty and tackle it and also multi-product model and product assignment between inbound and outbound rucks.

KEYWORDS: Cross-dock, scheduling-assignment, stochastic, uncertainty, scheduling with chance constraints.

1. INTRODUCTION

Products are transferred rapidly in today's customer-oriented economy and cost efficiency creates a competitive advantage for organizations. Organizations have understood that cross-dock operations have a great role in their distribution model to achieve this advantage which can replace current warehousing policies or to be merged with traditional models. As a result, organizations have planned to use cross-dock operations in their distribution models to reduce costs and hence achieve this competitive advantage. Cross-docking is a special warehousing policy which products are transferred directly from inbound trucks to outbound trucks without a temporary storage in between [1].

Use of cross-docks in a supply chain can be profitable for organizations through rapid transshipment of products and reduction of inventory costs. Internal processing operations of a cross-dock consist of unloading products from inbound trucks at inbound doors, sorting, temporary storage, consolidation and loading them to outbound trucks at outbound doors. Cost reduction and better efficiency can be achieved through improvement of these processes [2].

Cross-docks are intermediate nodes in a distribution network which are assigned particularly to loading trucks. Crossdock in distribution warehousing dictionary includes trucks and doors at two sides (inbound & outbound) with minimum storage area. Cross-dock name describes the process of unloading products from inbound doors and then transshipment to outbound doors. A cross-dock mainly is a consolidation point in a distribution network where smaller shipments can be consolidated to achieve economy of scale in transportation.

Proper integration of inbound and outbound trucks plays a great role in efficiency and effectiveness of cross-docks systems and also in whole supply chain. Truck scheduling operation considers assignment of inbound trucks to inbound doors where shipments are unloaded and assignment of outbound trucks to outbound doors where shipments are loaded. Scheduling strategies can affect the efficiency of cross-dock operations because they change delivery and waiting times. Moreover congestion or truck failure often result in fluctuation in truck arrival times [3]. This uncertainty in truck arrival times makes it irrational and inappropriate to use scheduling policies based on predicted truck arrival times [4].

Therefore we will consider a simultaneous inbound and outbound truck scheduling problem under uncertainty in truck arrival times. The purpose of this study is to handle a truck scheduling problem under uncertainty which consists of both scheduling and sequencing of inbound and outbound trucks to achieve an efficient transshipment time, reduced truck waiting time and to ensure the on-time delivery of shipments. In this study inbound truck arrival times are assumed uncertain which is a rational assumption in a real world situation.

Corresponding author: Mahdi Rajabi, Department of industrial engineering and managerial systems, Amirkabir University of Technology, Iran; Email: mehdi.rjb90@yahoo.com

2. BACKGROUND AND LITERATURE REVIEW

At the beginning cross-dock operations were used in transportation industry of United States in 30's and since then are used in less-than-truckload (LTL) operations. The U.S. army began to use these operations in 50's and Walmart is also utilizing it since 80's. No products are stored in these warehouses and items are transshipped in less than 48 hours. The use of cross-dock facilities in Wal-Mart have led to decreased inventory costs through implementing full-truckload (FTL) policy besides reduced transportation costs [3].

As mentioned before, cross-dock scheduling decisions determine the allocation of resources through planning horizon. These decisions for truck service types are taken in a way to ease the transshipment of products from inbound to outbound doors and mostly consider the doors as a constraint which means the number of available dock doors exceeds the number of trucks. Therefore detailed scheduling and sequencing is presented for minimizing waiting time of trucks and shipments. Truck scheduling can be completed at the beginning of planning horizon or a dynamic forward scheduling which is updated during the planning horizon which are called offline and online scheduling respectively. Most of the previous papers focused on operations of the facilities which is independent of location and layout of cross-docks. As

Agustina et al. [5] have mentioned: cross-dock operations include product allocation (choosing products to be transferred by cross-dock), dock service mode (assignment of dock doors to destinations), vehicle routing, transshipment and scheduling.

Chen & Lee [8] considered a cross-dock problem as a two-machine flow-shop. Their objective is to determine sequence of inbound and outbound trucks to minimize the make span (the time from starting the unloading of first inbound truck till finishing the loading of last outbound truck). They assumed that all the trucks are available at time zero. Unloaded shipments can be held in a temporary storage area until a proper outbound truck enters the shipping dock. The authors have proved that their model is strongly NP-hard and thus proposed a meta-heuristic approach based on Johnson's rule.

Van Belle et. al. [7] classified the truck scheduling problem into three categories. First category defines a simple crossdock with one inbound door and one outbound door. Scheduling in this case reduces to sequencing inbound and outbound trucks. Cross-docks with multiple inbound and outbound doors are discussed in second category but only inbound or outbound trucks are subject of scheduling. In the third category, simultaneous inbound and outbound truck scheduling is considered in a multiple inbound and outbound door cross-dock.

Yu et. al. [8] studied a truck scheduling problem of first category. They proposed a mixed-integer programming model to minimize the makespan. They developed a heuristic algorithm to solve the problem. Later on Zandieh (2011) [9] have developed several meta-heuristic algorithm to solve the same model.

Li et. al. [10] also studied a problem in the third category to maximize the transshipped products within planning horizon. If a shipment can't be loaded and delivered in a planning period, it will be handled at the next period. An intelligent genetic algorithm (IGA) is developed to solve the problem.

Larbi et. al. [11] considered the scheduling of internal operations in a single door cross-dock under three policies. Then they compared the performance of system under each policy. In the first policy it is assumed that we have full information about incoming shipments and the content of inbound trucks. In the second and third policies, we have assumed that no and partial information is known from incoming trucks. In this paper different solution approaches for scheduling outbound trucks in a cross-dock is presented.

Knour and Golias [12] studied truck scheduling problem in a multiple dock door cross-dock with uncertainty in arrival times of trucks. Thus a time window is considered for entering time of trucks and genetic algorithm (GA) is used to solve the model. The objective of this study is to minimize the average of total service and tardiness costs in the situation which cost of truck services are varied. This study formulates the scheduling problem as a bi-objective bi-level optimization problem.

They also used three approaches to determine the strategic planning while there is no extra information about crossdock operations: deterministic, optimistic and pessimistic approaches. They formulated a bi-objective bi-level optimization for optimistic and pessimistic approaches. The objective is to achieve a proper approach for truck scheduling problem at a cross-dock with uncertainty in truck arrival times [13].

Miao et. al. [14] studied both inbound and outbound truck scheduling problem in a multiple dock door cross-dock for the first time. They assumed that each dock door while it's available, can be used both as an inbound or outbound door and also trucks enter the cross-dock terminal at a per-defined time.

Van Belle et. al. [17] scheduled both inbound and outbound trucks. They considered the time needed to transfer shipments between cross-dock doors and delay of trucks based on pre-defined entering and leaving time of trucks.

3. Problem description

Truck scheduling is one of the operational problems of cross-docks which is related to assignment of trucks to crossdock doors [15]. These doors are considered resources that should be scheduled through planning horizon.

In this study we consider sets of inbound trucks (R) pairing with inbound doors (K) and sets of outbound trucks (S)pairing with outbound doors (L) in a multiple door cross-dock. We assume that type and quantity of products coming from inbound trucks and needed in outbound trucks are known beforehand. Different types of products $t \in N$ can be transported by each truck. r_{ik} is the amount of product type $k \in N$ which is initially loaded on inbound truck $i \in I$ and s_{ik} is the amount of product type $t \in N$ to be delivered by outbound truck $j \in J$. Truck changeover time just like most papers is assumed

to be constant (D). Each shipment's quality will be examined electronically after unloading then sorted, consolidated and transferred to shipping dock doors. Transferring process is dependent to the product type and related inbound and outbound trucks. As a matter of fact, the distance between inbound door k and outbound door l defines the transfer time of each shipment. This time is represented by t_{kl} . Objective of the problem is to minimize the probable waiting time of inbound and outbound trucks due to uncertainty in entering times which is calculated by sum of waiting times in receiving and shipping dock doors separately.

Following notations are used to formulate the mathematical model:

Sets:

- R Number of inbound trucks
- S Number of outbound trucks
- N Types of products
- L Number of receiving dock doors
- K Number of shipping dock doors

Parameters:

 f_{ijt} Number of products type t to be transferred from inbound truck i to outbound truck j

- r_{it} Number of products type t initially loaded on inbound truck i
- s_{jt} Number of products type t to be loaded on outbound truck j
- t_{kl} Transfer time of products from receiving door k to shipping door l
- D Truck changeover time
- a_i Expected entering time of inbound truck i to dock terminal
- b_{i} Expected entering time of outbound truck j to dock terminal

Continuous decision variables:

- \hat{a}_i Real entering time of inbound truck i to dock terminal
- b_i Real entering time of outbound truck j to dock terminal
- r_i Time which inbound truck i starts unloading at a receiving dock door
- e, Time which inbound truck i leaves receiving dock door
- s_i Time which outbound truck j starts loading at a shipping dock door
- f_{i} Time which outbound truck j leaves shipping dock door

Continuous decision variables:

 X_{ik} 1, if inbound truck i is assigned to receiving door k; 0, otherwise.

 Y_{jl} 1, if outbound truck j is assigned to shipping door 1; 0, otherwise.

- V_{ii} 1, if any product is transferred from inbound truck i to outbound truck j; 0, otherwise.
- p_{ii} 1, if inbound truck i precedes inbound truck j in inbound trucks sequence; 0, otherwise.

 q_{ii} 1, if outbound truck i precedes outbound truck j in outbound trucks sequence; 0, otherwise.

In this study, waiting time of trucks is considered as a negative factor in scheduling trucks. Thus the problem should be scheduled in a way to minimize any probable waiting times. We use and to represent waiting times of inbound and outbound trucks respectively. We define and as the beginning of unloading and loading operations of inbound truck i and outbound truck j respectively, then the waiting time of inbound and outbound trucks are calculated as follows:

$$Z_i = r_i - a_i \tag{1}$$
$$Z_i = s_i - b_i \tag{2}$$

With the above explanations, mixed-integer programming model with the objective of minimizing waiting times is can be described as follows:

$$Min \sum_{i=1}^{R} Z_{i} + \sum_{j=1}^{S} Z_{j}$$
(3)

s.t:

$$P \left\lfloor Z_i \ge T_i \right\rfloor \ge \rho \qquad \forall i \in R \tag{4}$$

$$P \left\lfloor Z_i \ge T_i \right\rfloor \ge \rho \qquad \forall i \in S \tag{5}$$

$$\sum_{k=1}^{K} X_{ik} = 1 \qquad \forall i \in \mathbb{R}$$
(6)

$$\sum_{s}^{L} Y_{jl} = 1 \qquad \forall j \in S$$
(7)

$$\sum_{j=1}^{S} f_{ijt} = \mathbf{r}_{it} \qquad \forall i \in \mathbb{R}, \forall t \in \mathbb{N}$$
(8)

$$\sum_{i=1}^{n} f_{ijt} = \mathbf{s}_{jt} \qquad \forall j \in S, \forall t \in N$$
(9)

$$\sum_{l=1}^{L} \sum_{k=1}^{n} Z_{ijkl} = \mathsf{V}_{ij} \qquad \forall i \in \mathbb{R}, \forall j \in S$$

$$\tag{10}$$

$$Z_{iikl} \leq X_{ik} \qquad \forall i \in \mathbb{R}, \forall j \in S, \forall k \in K, \forall l \in L$$
(11)

(12)

(27)

$$Z_{ijkl} \leq Y_{jl} \qquad \forall i \in R, \forall j \in S, \forall k \in K, \forall l \in L$$

$$Y_{ijkl} = V_{ijkl} = V_{ijkl} = V_{ijkl} = V_{ijkl} = V_{ijkl} = V_{ijkl}$$

$$X_{ik} + X_{jk} - 1 \le p_{ij} + p_{ji} \qquad \forall i, j \in \mathbb{R}, i \ne j, \forall k \in \mathbb{K}$$

$$p_{ij} + p_{ji} \le 1 \qquad \forall i, j \in \mathbb{R}$$
(13)
(14)

$$Y_{il} + Y_{il} - 1 \le q_{ii} + q_{ii} \qquad \forall i, j \in S, i \ne j, \forall l \in L$$

$$(15)$$

$$q_{ij} + q_{ji} \le 1 \qquad \forall i, j \in S \tag{16}$$

$$r_i \ge a_i \qquad \forall i \in R \tag{17}$$

$$\begin{aligned} z_i + a_i &\geq e_j + D - M \ (1 - p_{ji}) &\forall i, j \in K \end{aligned}$$

$$\begin{aligned} s_i &\geq b_i &\forall i \in S \end{aligned}$$

$$(18)$$

$$\begin{aligned} S_i &\geq b_i & \forall j \in S \\ Z_i &+ b_i \geq f_j + D - M \ (1 - q_{ji}) & \forall i, j \in S \end{aligned}$$

$$(19)$$

$$e_i \ge Z_i + a_i + \sum_{i=1}^{N} r_{ii} \qquad \forall i \in R$$
(21)

$$f_j \ge Z_j + b_j + \sum_{t=1}^N s_{jt} \qquad \forall j \in S$$

$$(22)$$

$$f_{j} \ge e_{i} + \sum_{l=1}^{L} \sum_{k=1}^{K} Z_{ijkl} t_{kl} + \sum_{t=1}^{N} f_{ijt} - M \cdot (1 - V_{ij}) \qquad \forall i \in R, \forall j \in S$$
(23)

$$X_{ik}, Y_{jl}, V_{ij}, p_{ij}, q_{ij} \in \{0, 1\} \qquad \forall i \in \mathbb{R}, \forall j \in S, \forall k \in \mathbb{K}$$

$$(24)$$

$$\hat{a}_i, \hat{b}_j, r_i, \mathbf{e}_i, s_j, \mathbf{f}_j \ge 0 \qquad \forall i \in \mathbb{R}, \forall j \in S$$
(25)

Equation (3) is the objective function of minimizing total makes pan. Equations (4) and (5) show the uncertainty of model. Equations (6) and (7) state that each truck is only assigned to one dock door which prevents the twice processing operation of trucks or more. Equation (8) ensures that number of products of type t transferred from each inbound truck to outbound trucks are exactly equal to the number of products of type t in that specific inbound trucks. Equations (10)-(12) ensures the correct relation between variables , and is set. Sequence of inbound and outbound trucks are shown in equations (13)-(16). Starting time of unloading inbound trucks and loading outbound trucks are shown in equations (17)-(20). Equations (21) and (22) state the finish time of loading outbound trucks. Equation (23) shows the finish time of outbound trucks which is greater than start time of all related inbound trucks plus transfer times from receiving docks to shipping dock plus unloading time of all the products transferred between related inbound trucks.

Starting time of un/loading operations is after arrival of trucks to cross-dock terminal. If dock doors are occupied when a truck enters the cross-dock, it should wait until one door is free. Thus start time of each truck according to equations (17) and (18) is after finish time of previous truck. As a result, two equations are obtained from equations (17) and (18) and also (19) and (20):

$$r_{j} = \max\{a_{j}, p_{ij}.(e_{i} + D)\}$$
(26)

 $s_{j} = \max \{ b_{j}, q_{ij} . (f_{i} + D) \}$

D

And equations (22) and (23) formulate the finish time of outbound trucks as follows:

$$f_{j} = \max\left\{s_{j} + \sum_{t=1}^{N} s_{jt}, \mathbf{V}_{ij} \left(e_{i} + \sum_{l=1}^{L} \sum_{k=1}^{K} Z_{ijkl} t_{kl} + \sum_{t=1}^{N} f_{ijt}\right)\right\}$$
(28)

4. Scheduling With Chance Constraints

Goal of chance-constrained mathematical programming (CCMP) is to find the optimal solution of problems which probability of events in it have deterministic bounds. This concept was first used by Charnes Luward. They used the deterministic equivalent of the problem to solve it.

Perkopa [31] studied combination of CCMP with independent random variables and proposed an equivalent deterministic problem which is convex on right hand distribution parameters under deterministic conditions.

Luedtke et al. [32] showed that for most statistical distribution functions, average of sample approximation is suitable to reach a good and feasible solution.

Zhang et. al. [33] studied a scheduling model with multi-level chance constraints and verified the equivalent deterministic model and observed that decomposition algorithm is proper for large scale problems.

5. Solution Approach with Chance Constraints

To solve the uncertain model by scheduling with chance constraints, equivalent deterministic constraints are made by methods describe in last section. Thus equations (1) and (2) are replaced with the following ones:

$$P \left\lfloor \alpha_i - a_i \ge T_i \right\rfloor \ge \rho \tag{29}$$

$$P\left[\frac{\alpha_{i} - a_{i} - \overline{T_{i}}}{\sqrt{\operatorname{var}(T_{i})}} \ge \frac{T_{i} - \overline{T_{i}}}{\sqrt{\operatorname{var}(T_{i})}}\right] \ge \rho$$
(30)

$$P\left[Z \leq \frac{\alpha_i - a_i - \overline{T_i}}{\sqrt{\operatorname{var}(T_i)}}\right] \geq \rho \tag{31}$$

In the standard normal distribution we have: $\Phi(E) = \rho$

$$\Phi\left[\frac{\alpha_{i} - a_{i} - \overline{T_{i}}}{\sqrt{\operatorname{var}(T_{i})}}\right] \ge \Phi[E]$$
(33)

(32)

$$\alpha_i - a_i - \overline{T_i} - E\left(\sqrt{\operatorname{var}\left(T_i\right)}\right) \ge 0 \tag{34}$$

The same approach is used for equation (2):

$$\beta_j - b_j - \overline{T_j} - E\left(\sqrt{\operatorname{var}(T_j)}\right) \ge 0 \tag{35}$$

6. Meta-Heuristic Algorithm

Exact approaches cannot find the optimal solution of large scale problems (see section 5). To overtake this difficulty, a meta-heuristic algorithm namely variable neighborhood search (VNS) is used.

6.1 Variable Neighborhood Search Algorithm

VNS algorithm was proposed by Hensen and Mladenovi'c [41] which is a meta-heuristic approach to solve combinatorial optimization problems. Systematic change of neighborhood structure in a local search is the basic idea of this algorithm. Basically, a local search method searches a limited portion of solution space. VNS is a simple and effective method based on systematic change of neighborhood through the search operation. This method escapes from local optimum by changing the neighborhood structure. It starts from an initial solution and uses two involute loops to continue the search. In the first loop, shaking operator creates a solution at the neighborhood of current solution. This random solution is the key to avoid getting stuck in local optimum. In the second loop, it uses the random solution found at last step to do the local search and the local optimum. The local search, seeks an improved solution in the neighborhood of current solution while shaking relocates the solution to a new neighborhood structure. Local search is applied until the solution keeps improving or a stopping criterion is met. After finishing the inner loop, outer loop creates a solution which is used by inner loop for the next iteration. This continues until the stopping criterion of the whole algorithm is met. Because neighborhood structures play an important role on the performance of VNS algorithm, it is crucial to choose these structures wisely to achieve an efficient algorithm.

Main steps to implement the VNS algorithm are as follows:

- 1. Start: a set of neighborhood structures Nk(k=1,..., kmax) is defined. k=1 and an initial solution x is defined and a stopping criterion is selected.
- 2. Shaking: a random solution x' is generated from the kth neighborhood of x.
- 3. Local search: a local search method is applied to x' to find the local optimum, x''.
- 4. Move: if x'' is better than the current solution x, current solution is replaced with x'' and k=1. Else, k=k+1.
- 5. Stop: if the stopping criterion is not met, algorithm continues on step 2, otherwise it stops.

Solutions gained by VNS algorithm are highly dependent to factors like initial solution, chosen neighborhoods, local search methods and the sequence of neighborhoods.

6.2 Initial Solution of the Algorithm

A random initial solution is considered for the VNS algorithm. Each inbound or outbound truck is assigned randomly to a receiving or shipping dock door. Then a random sequence is assigned to the trucks that are assigned to a same dock door. The algorithm uses this solution to continue the search.

6.3 Neighborhood Structures

In this study we consider four neighborhood structures as follows:

- 1. Replacing the door of two inbound trucks together: in this structure, two inbound trucks are selected and their assigned door are replaced with each other.
- 2. Replacing the door of two outbound trucks together: in this structure, two outbound trucks are selected and their assigned door are replaced with each other.
- 3. Changing the assigned door of an inbound truck: in this structure, one inbound truck is selected and is assigned to a different receiving dock door.
- 4. Changing the assigned door of an outbound truck: in this structure, one outbound truck is selected and is assigned to a different shipping dock door.

6.4 Stopping Criterion

Stopping criterion for VNS algorithm in this study is the number of iterations between two improvements in the solution. This means that if in 70 consequent iterations no improvement is reached, algorithm stops.

7. Computational Results

Instance problems were generated in two groups with small and large scales and were solved by mentioned method. Scale of each problem is defined by number of inbound and outbound trucks, number of receiving and shipping dock doors and number of product types which for each size, five instances were generated. The results of small-scaled instances were compared to optimum results gained by exact solution method. For this, relative error of algorithm is calculated by following equation. Optimum solution in this equation is calculated by GAMS software. Also in cases which no optimum solution is gained by our algorithm is used.

Relative error percentage = $\frac{\text{Solution of algorithm-optimum solution}}{\text{optimum solution}} \times 100$

7.1 Computational Results In Small-Sized Instances

In this section the performance of algorithm in small-sized instances are check. To do this, 10 problem sizes are defined and in each size, 3 instances are generated and each instance is solved 5 times by VNS algorithm. Compared results from exact solution and our algorithm are shown in table 1.

Table 1 shows the performance of the algorithm in average, worst and best cases. From the 8th size on, GAMS software couldn't solve the problem in a reasonable time. As could be observed, relative error increases with increase in problem size but this growth doesn't have a specific trend. the relative error of the algorithm compared to optimum solution is acceptable. Fig 1 illustrates the performance of algorithm in small scales by means of relative error percentage.

In the following section performance of algorithm is evaluated by time elapsed till end of algorithm compared to GAMS software. In the table 2, information of problem sizes, time of GAMS software to reach optimum solution and VNS algorithm time are shown. Because stopping criteria of algorithm is the iterations between two improvements, two different times is reported for this algorithm; the time when algorithm reached the best found solution for the first time and also the finish time of it.

As observed in table 2, CPU times for small sizes are near zero and it gradually increases by increase in problem size. This trend in CPU time increase is shown in figure 2. The finish time of proposed algorithm is illustrated in figure 3 and it shows that the same trend for increase in CPU time to reach the best solution for the first time also exists in the finish time of algorithm.

Size No.	No. of receiving	No. of shipping	No. of inbound	No. of outbound	No. of product	Error			
	doors	doors	trucks	trucks	types	Varia	ble neighborhood	l search	
						Worst	Mean	Best	
1	2	1	2	2	2	0.00031	0.00031	0.00031	
2	3	3	4	4	2	0.00027	0.00027	0.00027	
3	3	3	4	5	2	0.00016	0.00016	0.00016	
4	3	3	6	6	3	0.00011	0.00011	0.00011	
5	3	3	7	6	3	4.9913	2.1436	0.00018	
6	4	4	7	6	3	6.1126	5.4593	5.3384	
7	4	4	7	8	4	11.8374	7.1127	5.1123	
8	4	4	8	8	4	7.1923	5.8923	5.4190	
9	4	5	8	9	4	12.1107	10.7293	9.1035	
10	5	5	9	9	4	12.1020	6.4210	3.0384	
		Avera	age error			5.4347	3.7759	3.0012	





Figure 1. Average error of proposed algorithm compared to GAMS results in small scale

Size No.	No. of receiving	No. of shipping	No. of inbound	No. of outbound	No. of product		Solution time				
	uoors	doors	trucks	trucks	types	GAMS	GAMS VNS				
							Best time	Finish time			
1	2	1	2	2	2	0.524	0.0568	0.5301			
2	3	3	4	4	2	0.598	0.0943	2.9782			
3	3	3	4	5	2	1.929	0.0934	2.6509			
5	3	3	6	6	3	5.947	0.1951	3.8997			
5	3	3	7	6	3	11.703	0.2574	5.0803			
6	4	4	7	6	3	870.228	1.6161	10.694			
7	4	4	7	8	4	665.023	1.4643	10.19			
8	4	4	8	8	4	1385.930	2.9716	12.511			
9	4	5	8	9	4	2473.245	9.9325	28.237			
10	5	5	9	9	4	2849.307	15.7602	39.0856			

Table 2. Time elapsed by proposed algorithm compared to GAMS software in small scale



Figure 2. Time to reach the best solution for the first time by proposed algorithm in small scale



Figure 3. Finish time of proposed algorithm in small scale

7.2 Computational Results In Large-Sized Instances

After evaluating the performance of VNS algorithm in small-sized problems, we solve large-sized problems with our proposed algorithm. Just like last section, we defined 10 sizes and for each size, 3 instances were generated and each instance was solved 5 time by VNS algorithm. Considering that GAMS software is unable to solve these large-sized problems, the best solution found by algorithm is considered as the optimum solution to evaluate the overall performance of the algorithm. Table 3 shows the performance of the algorithm based on objective function in large-sized problems.

As could be observed, VNS algorithm performs well from the relative error percentage viewpoint. This algorithm has 2.29% error in average compared to best results obtained. Also its performance remains acceptable in last three sizes which are very large problems. Figure 4 shows the relative error percentage of this algorithm in large scale. Now we consider CPU times to reach the solutions of the algorithm. As seen in table 4, time elapsed to solve the problem increases as the problem size increases which in the largest problem it takes 2300 seconds to report the solution.

Figure 5 shows the CPU time to find the best solution by proposed algorithm for the first time in large scale problems. Time growth trend is almost linear proportional to increase in problem size.

T	ab	le .	3.]	Re	elati	ve	erro	r p	erc	ent	age	of	pro	posed	l a	lgori	itł	nm	in	large	e sca	ıle

Size	No. of	No. of	No. of	No. of	No. of	Error				
No.	receiving	shipping	inbound	outbound	product	VNS				
	doors	doors	trucks	trucks	types	Worst	Average	Best		
1	6	6	10	10	2	0.0023	0.0023	0.00		
2	6	7	12	12	2	5.0389	1.8325	0.00		
3	7	7	12	14	3	0.9285	0.3309	0.00		
5	8	8	15	15	3	1.019	0.4902	0.00		
5	8	9	15	18	3	0.8927	0.1108	0.00		
6	10	9	20	18	4	5.9028	1.7905	0.00		
7	10	10	20	25	4	8.2173	3.9088	0.00		
8	10	10	25	25	4	14.770	5.0378	0.00		
9	10	12	25	30	5	10.743	5.0923	0.00		
10	12	12	30	30	5	7.9374	4.3190	0.00		
		Ave	rage error			5.4347	2.2915	0.00		



Figure 4. Average error of proposed algorithm in large scale

Table 4. Time elapsed by proposed a	algorithm in	large scale
-------------------------------------	--------------	-------------

Size	No. of receiving	No. of	No. of	No. of	No. of	CPU time VNS		
No.	doors	shipping	inbound	outbound	product			
		doors	trucks	trucks	types	Best time	Finish time	
1	6	6	10	10	2	0.9031	25.6821	
2	6	7	12	12	2	9.1147	46.4875	
3	7	7	12	14	3	40.8819	78.9901	
5	8	8	15	15	3	63.385	125.1145	
5	8	9	15	18	3	106.1092	339.5118	
6	10	9	20	18	4	336.1203	771.069	
7	10	10	20	25	4	669.5992	1002.001	
8	10	10	25	25	4	840.0127	139.0818	
9	10	12	25	30	5	1716.3124	2090.473	
10	12	12	30	30	5	1828.0175	2310.8509	



Figure 5. Time to reach the best solution for the first time by proposed algorithm in large scale

8. Conclusion

In this study we considered a truck scheduling problem with uncertainty in both inbound and outbound problems. Scheduling authorities at cross-dock should consider these uncertainties and propose a scheduling & sequencing plan which is robust against these uncertainties. As we know, trucks don't necessarily assign to receiving or shipping doors and should wait a while till a proper dock door is free and ready to service. The objective of this study is to minimize these waiting times by proper assignment and sequencing trucks. This schedule should be ready before trucks enter the cross-dock terminal and thus we must use expected arrival times of trucks. And since these expected arrival times may differ from real arrival times of trucks, scheduling should be performed in a way to include these differences. If the difference between real and expected arrival time of a truck is less than its waiting time, no difference is made to the scheduling. But if the lag time is greater than waiting time, the schedule would be inefficient. As a result, it's necessary to schedule in a way that these waiting times be at most the size of difference between real and expected arrival times of trucks to make it useable and efficient. Moreover, this paper considers uncertainty in mathematical model of the problem by use of chance constraints to consider uncertainty and tackle it and also multi-product model and product assignment between inbound and outbound rucks.

To implement and evaluate the results of the model, some problem instances were generated based on number of inbound and outbound trucks, receiving and shipping doors and also product types. Each problem in small and large scales were solved by our proposed algorithm. The results show that by increasing number of trucks and iterations per problem, CPU times will increase and the objective function gets worse. It's notable that increase in number of dock doors will smoothen the flow of products and also the assignment of trucks and thus lead to improvement in objective function but at the other hand it will increase the volume and time of computations.

REFERENCES

- 1. Zäpfel & Wasner, Warehouse sequencing in the steel supply chain as a generalized job shop model 2009. International Journal of Production Economics. P. 482-501
- 2. Wooyeon Yu a, Pius J. Egbelu, Scheduling of inbound and outbound trucks in cross docking systems with temporary storage 2008. European Journal of Operational Research. P. 377–396
- Boysen, N. Fliedner, M, Cross dock scheduling: Classification, literature review and research agenda 2010. Omega. P. 413-422
- 4. Boysen, N. Fliedner, M. Scholl, A, Scheduling inbound and outbound trucks at cross docking terminals 2010. OR spectrum. P. 135-161
- 5. Agustina, D. Lee, C. K. M. Piplani, R, A Review: Mathematical Modles for Cross Docking Planning 2010. International Journal of Engineering Business Management. P. 47-54
- 6. CHEN, F. LEE, C.-Y, Minimizing the makespan in a two-machine cross-docking flowshop problem 2009. European Journal of perational Research. P. 59-72
- 7. Van Belle, J. Valckenaers, P. Cattrysse, D, Cross-docking: State of the art 2012. Omega. P. 827-846
- 8. Yu, W. Egbelu, P. J, Scheduling of inbound and outbound trucks in cross docking systems with temporary storage 2008. European Journal of perational Research. P. 377-396
- 9. Boloori Arabani, A. R. Fatemi Ghomi, S. M. T. Zandieh, M, Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage 2011. Expert systems with Applications. P. 1964-1979
- Lee, Kangbae. Byung Soo Kim. Cheol Min Joo, Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance 2012. Expert Systems with Applications. P. 12975-12983
- 11. Larbi, R. Alpan, G. Baptiste, P. Penz, B, Scheduling cross docking operations under full, partial and no information on inbound arrivals 2011. Computers & Operations Research. P. 889-900
- 12. Konur, D. Golias, M. M, Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A metaheuristic approach 2013. Transportation Research Part E: Logistics and Transportation Review. P. 71-91
- 13. Konur, D. Golias, M. M, Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty 2013. Computers & Industrial Engineering. P. 663-672
- 14. Miao, Z. Lim, A. Ma, H, Truck dock assignment problem with operational time constraint within crossdocks 2009. European Journal of Operational Research. P. 105–115
- 15. Van Belle, J. Valckenaers, P. Cattrysse, D, Cross-docking: State of the art 2012. Omega. P. 827-846
- 16. Nenad Mladenovi'c, Pierre Hansen, Variable neighborhood search 1997. Computers and Operations Research. P. 1097–1100
- 17. Nenad Mladenovi'c, Pierre Hansen, Variable neighborhood search: Principles and applications 2001. European journal of Operational Research. P. 449–467