DOCK ASSIGNMENT AND TRUCK SCHEDULING PROBLEMS AT CROSS-DOCKING TERMINALS

Ching-Jung Ting
Department of Industrial Engineering and Management, Yuan Ze University, Taiwan
ietingcj@saturn.yzu.edu.tw

Amanda G. Rodríguez López
Parque Tecnológico, Universidad Nacional de Ingenieria, Nicaragua
amandaguadaluper@gmail.com

Abstract

In this paper, we consider the integration of dock assignment and truck scheduling problem at cross-docking terminals. The problem is first formulated as a 0-1 integer programming model. Since both dock assignment and truck scheduling problems are NP-hard, its integration is more difficult to solve. Thus we propose reduced variable neighborhood search (RVNS) algorithms to solve the problem. Computational experiments are carried out on four set of instances. The results show that RVNS is capable of finding good solutions in a much shorter computation time when it is compared with optimization solver Gurobi's solutions.

1. Introduction

Cross-docking is the process of receiving product and shipping it out the same day or overnight without putting it into storage [5][6]. Therefore the storage and retrieval cost, the two most expensive among those five warehousing operations (receiving, sorting, storing, retrieving and shipping), are removed. Nowadays, cross-docking has become an increasingly popular distribution strategy implemented by organizations to improve supply chain efficiency and minimize distribution cost [30]. A December 2010 survey of 219 logistics professionals conducted by Saddle Creeks [42] showed that more than two thirds (68.5%) of respondents currently cross dock and 15.1% plan to begin in the next 18 to 24 months. Cross-docking is not a new practice; it has seen a resurgence of interest in recent years due to the benefit provided by the cross-docking systems, such as improving service levels, reducing transportation costs, etc. It requires advanced knowledge of the inbound product, its destination, and a system for routing the product to the proper outbound vehicle. Wal-mart [32] and Toyota [38] are two well-known examples that reported the successful implementation of cross-docking.

To the best of our knowledge, three papers present a review of cross-docking research. Boysen and Fliedner [11] structured a classification scheme for the cross-docking truck scheduling problem. Agustina et al. [1] provided a general picture of the mathematical models used in cross-docking planning. Van Belle et al. [36] presented an extensive overview of the cross-docking concept and described several characteristics. Several future research opportunities are discussed.

Dock assignment problems are defined to find the best truck-to-dock assignment pattern to minimize the travel distance of exchanging products and hence the total operational time within the terminal. If the number of trucks arrives at the cross-docking yard is more than the number of docks, some of them have to wait in a queue until the assigned trucks complete their operations and release the docks. In this context, the objective of the truck scheduling problem is to sequence the waiting trucks in a way so that the total operational time is minimized. As the objective of both problems is to minimize total operational time, they can be addressed in an integrated way.

Most research discussed both dock assignment and truck scheduling problems separately. For the dock assignment problem, Tsui and Chang [33] used a bilinear program to determine the inbound and outbound dock allocation in a cross-docking that would minimize the total material handling effort. The dock assignment problem is a special case of the quadratic assignment problem (QAP), and like all QAP problems, this bilinear problem is NP complete [18]. This formulation was latter solved in Tsui and Chang [34] using a branch and bound algorithm. Bartholdi and Gue [5] proposed a non-linear model to minimize the total labor cost (travel costs and congestion cost), subject to an additional door pressure constraint. A simulated annealing procedure that swaps pairs of trucks sequentially to solve the assignment problem. Yu [39] worked in the application of a dock assignment problem in a real cross-docking facility, solving the model through an on-line algorithm, a local search heuristic and a greedy genetic algorithm for the outbound dock assignment problem.

Oh et al. [31] considered the mail distribution center in which the different doors are clustered into groups. A non-linear mathematical model is developed with the objective of minimizing the internal travel distance and a decomposition heuristic and a genetic algorithm are proposed to solve the problem. Bozer and Carlo [12] considered the inbound and outbound trailer-to-door assignments in crossdocks without taking into account the congestion. A simulated annealing-based heuristic is proposed to determine the door assignments. Yu et al. [40] developed an on-line myopic policy that assigns arriving inbound trucks on a real-time basis to minimize the total expected travel time. However, the policy only minimized the considered inbound trucks; it may worsen the processing time of future arriving trucks. Cohen and Keren [16] discussed the existing approaches for assigning docks to trucks. They proposed a non-linear mixed integer programming model and developed a heuristic algorithm to solve the problem.

In the case of the truck scheduling problem, earlier studies assumed a single shipping and a single receiving dock. Truck scheduling problem reduces to the sequencing problem in such an assumption. Yu and Egbelu [41] addressed a truck scheduling problem where the product assignments from inbound trucks to outbound trucks are determined simultaneously with the docking sequences of the inbound and outbound trucks. Chen and Lee [14] studied the truck scheduling problem as a two-machine flow shop scheduling problem and showed that the problem is strongly NP-hard. Vahdani and Zandieh [35] presented an exhausted analysis of the performance of five metaheuristic algorithms for the truck scheduling problem. According to their results variable neighborhood search algorithm (VNS) is recommended to solve truck scheduling in cross-docking system problems. Boloori Arabani et al. [8] also presented five metaheuristics to tackle same problem. Boysen et al. [11] introduced a base model for scheduling trucks at cross-docking terminals, where a "one inbound dock serves one outbound dock" problem is considered, the problem was also solved through a stochastic approach by Baptiste and Maknoon [4] and through a graph based model by Larbi et al. [20]. Boloori Arabani et al. [7] addressed another particular case of truck scheduling problem with just-in-time approach that outbound trucks have a due date. Larbi et al. [21] considered on the outbound truck scheduling problem that outbound trucks are available at any time and preemption is allowed.

McWilliams and his coworkers conducted studies devoted to the parcel hub scheduling problem (PHSP) involving the scheduling of a set of inbound trailers loaded with a batch of heterogeneous parcels to a set of shipping docks, with the objective of minimizing the time span of the transfer operation. McWilliams et al. [24] proposed a simulation based scheduling algorithm utilizing a genetic algorithm (GA) to guide the search. It was assumed that the batch sizes of the inbound trucks are equal, while this assumption was relaxed in McWilliams et al. [25]. McWilliams et al. [25] proposed a simulation-based scheduling approach with an embedded genetic algorithm to solve the problem. To reduce the demand for simulation evaluations, McWilliams [26] proposed a minimax programming model resembling a multi-knapsack problem, and used a genetic algorithm to solve the large-scale PHSP. McWilliams [27] developed local search and simulated annealing algorithms and showed that both algorithms outperformed GA. The results show that the new approach is superior to the GA.

Miao et al. [28] considered a truck dock assignment problem with an operational time constraint in cross-dockings where the number of trucks exceeds the number of docks available. Wang and Regan [37] compared the performance of different strategies for processing trucks at a cross-docking facility through dynamic simulation models. Boysen [10] also worked in a real world scheduling setting for the food industry considering zero-inventory cross-dockings in the food industry. Chen and Song [15] extended Chen and Lee [14] to the two-stage hybrid cross-docking scheduling problem which at least one stage has more than one parallel machine. Li et al. [22] considered a multiple dock cross-docking where all docks are the same and could be used either as inbound or outbound, the problem is formulated as a parallel machine scheduling problem, where there is no differentiation between inbound and outbound operations.

Alpan et al. [3] also solved the multiple inbound and outbound dock configuration problem by a bounded dynamic programming. Forouharfard and Zandieh [17] developed an imperialistic competitive algorithm (ICA) for a scheduling problem. Alpan et al. [2] considered a multiple inbound and outbound dock configuration where the objective is to find the best schedule of transshipment operations to minimize the sum of inventory holding and truck replacement cost. Liao et al. [23] studied the simultaneous dock assignment and sequencing of inbound trucks for a multi-door cross docking operation under a fixed outbound truck departure schedule. The objective is to minimize the total weighted tardiness. The problem was solved by six different metaheuristic algorithms which are simulated annealing (SA), tabu search (TS), ant colony optimization (ACO), differential evolution (DE), and two hybrid differential evolution algorithms (DE). They found that ACO is the best among all the six metaheuristic algorithms tested.

In this paper, we present a reduced variable neighborhood search (RVNS) to solve the integrated problem with multi doors for both inbound and outbound trucks. The variable neighborhood search (VNS) algorithm proposed by Mladenović and Hansen [29] is a new approach for solving combinatorial optimization problems. VNS is easy to implement and has been applied to several optimization successfully [19]. By adopting the basic strategy of VNS, the RVNS can obtain good performance on dock assignment and truck scheduling problem considered, being able to provide good solutions for the testing instances under different scales.

The remainders of the paper are organized as follows. Section 2 presents the problem description and the model formulation. Section 3 proposes reduced variable neighborhood search algorithms for the model presented in section 3. Section 4 tests the proposed models in Gurobi Optimizer and the effectiveness of the RVNS algorithms. Section 5 concludes this research and suggests future work.

2. Problem Description

2.1 Assumptions

In this paper a cross-docking system where outbound trucks have to deliver goods to different destinations is considered. The objective is to minimize the total delay time of outbound trucks by the efficient scheduling of inbound trucks and assignment of inbound and outbound trucks to their respective docks. During the process inbound trucks can be scheduled at any time since no restrictions on arrival time are considered. Service times vary from truck to truck, which depends on the flow each truck carries/demands. The facility layout considered allows an intermediate storage with unlimited capacity in front of each outbound dock. Different from other researches, transshipment times depend on the docks between which a shipment is moved. As soon as an outbound truck loads its predefined set of products, it leaves the terminal.

The assumptions in this research are as follows.

1. One side of the dock is exclusively designated to inbound trucks and the other side to outbound trucks.
2. The number of inbound/outbound trucks is larger than the number of inbound/outbound docks. Such a configuration is quite common and realistic in the real world.
3. No preemption of trucks is allowed. Once docked, trucks will not leave the dock until their loading or unloading operations is finished.
4. Unit loading time for a unit pallet is identical to any outbound truck, same for the unit unloading time for any inbound truck.
5. Transshipment time between docks depends on the distance between the inbound dock and the outbound dock. That is, the transshipment time between each pair of docks is given.
6. Each outbound truck has a predetermined departure time.
7. The flow between inbound and outbound trucks is known. The flow is dedicated to a specific outbound truck and is not interchangeable.
8. Each outbound dock is capable to storage the outbound truck's freight.

Based on these assumptions, a mixed integer programming will be formulated to minimize the total delay times by considering the scheduling and assignment on both inbound and outbound trucks simultaneously. Since the dock assignment is a NP-hard problem, consider both scheduling and assignment problem is NP-hard. An exact solution approach is difficult to find solution within reasonable time for real world large problem. We propose a reduced variable neighborhood search algorithm to solve this problem.

2.2 Mathematical Formulation

Following notations are used for the model.

Input

a_i: Arrival time of inbound truck *i*

b_k: The starting available time of receiving dock *k*

d: Dummy last truck

f_{ij}: Flow between inbound truck *i* and outbound truck *j*

g_l: The starting available time of shipping dock *l*

I: Set of inbound trucks

J: Set of outbound trucks

K: Set of inbound docks

L: Set of outbound docks

M: A large number

o: Dummy starting truck

p_j: Planned departure time of outbound truck *j*

S_i: Associated outbound trucks to inbound truck *i*

S_j: Associated inbound trucks to outbound truck *j*

t_{kl}: Transportation time between docks k and l
α: Unloading time per unit of flow
β: Loading time per unit of flow
Decision Variables
c_i^k: Time at which inbound truck i enters the receiving dock k
E_j^l: Time at which outbound truck j leaves the shipping dock l
F_i^k: Time at which inbound truck i leaves the receiving dock k
H_j^l: Time at which outbound truck j enters the shipping dock l

$$x_{im}^k = \begin{cases} 1 & \text{if inbound truck } m \text{ immdeiately succeeds } i \text{ at receiving dock } k \\ 0 & \text{otherwise} \end{cases}$$

$$y_{jn}^l = \begin{cases} 1 & \text{if outbound truck } n \text{ immdeiately succeeds } j \text{ at shipping dock } l \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ij}^{kl} = \begin{cases} 1 & \text{if inbound truck } i \text{ is assigned to dock } k \text{ and outbound truck } j \text{ to dock } l \\ 0 & \text{otherwise} \end{cases}$$

The mathematical model for the problem is as follow.

$$Min \quad \sum_{j\in J}\sum_{l\in L}(E_j^l - p_j)^+ + \sum_{i\in I}\sum_{k\in K}(c_i^k - a_i) \tag{1}$$

$$S.T. \quad \sum_{i\in I\cup\{o\}} x_{im}^k = \sum_{i\in I\cup\{d\}} x_{mi}^k \qquad \forall m \in I, k \in K \tag{2}$$

$$\sum_{k\in K}\sum_{m\in I\cup\{d\}} x_{im}^k = 1 \qquad \forall i \in I \tag{3}$$

$$\sum_{i\in I\cup\{d\}} x_{oi}^k \le 1 \qquad \forall k \in K \tag{4}$$

$$\sum_{j\in J\cup\{o\}} y_{jn}^l = \sum_{j\in J\cup\{d\}} y_{nj}^l \qquad \forall n \in J, l \in L \tag{5}$$

$$\sum_{l\in L}\sum_{n\in J\cup\{d\}} y_{jn}^l = 1 \qquad \forall j \in J \tag{6}$$

$$\sum_{j\in J\cup\{d\}} y_{oj}^l \le 1 \qquad \forall l \in L \tag{7}$$

$$z_{ij}^{kl} \le \sum_{m\in I\cup\{d\}} x_{im}^k \qquad \forall i \in I, j \in J, k \in K, l \in L \tag{8}$$

$$z_{ij}^{kl} \le \sum_{n\in J\cup\{d\}} y_{jn}^l \qquad \forall i \in I, j \in J, k \in K, l \in L \tag{9}$$

$$z_{ij}^{kl} \ge \sum_{m\in I\cup\{d\}} x_{im}^k + \sum_{n\in J\cup\{d\}} y_{jn}^l - 1 \qquad \forall i \in I, j \in J, k \in K, l \in L \tag{10}$$

$$F_i^k \ge c_i^k + \alpha \sum_{j\in S_i} f_{ij} \qquad \forall i \in I, k \in K \tag{11}$$

$$c_m^k \ge F_i^k - (1 - x_{im}^k)M \qquad \forall i, m \in I \cup \{o\} \cup \{d\}, k \in K \tag{12}$$

$$c_o^k = F_o^k \geq b_k \qquad \forall k \in K \qquad (13)$$

$$h_o^l = E_o^l \geq g_l \qquad \forall l \in L \qquad (14)$$

$$c_i^k \geq a_i \qquad \forall i \in I, k \in K \qquad (15)$$

$$E_j^l \geq h_j^l + \beta \sum_{i \in S_j} f_{ij} \qquad \forall j \in J, l \in L \qquad (16)$$

$$h_n^l \geq E_j^l - (1 - y_{jn}^l) M \qquad \forall j, n \in J \cup \{o\} \cup \{d\}, l \in L \qquad (17)$$

$$E_j^l \geq (c_i^k + t_{kl}) z_{ij}^{kl} + (\alpha + \beta) f_{ij} \qquad \forall j \in J, i \in S_j, k \in K, l \in L \qquad (18)$$

$$c_i^k, F_i^k \geq 0 \qquad \forall i \in I, k \in K \qquad (19)$$

$$h_j^l, E_h^l \geq 0 \qquad \forall j \in J, l \in L \qquad (20)$$

$$x_{im}^k = 0,1 \qquad \forall i, m \in I, k \in K \qquad (21)$$

$$y_{jn}^l = 0,1 \qquad \forall j, n \in J, l \in L \qquad (22)$$

$$z_{ij}^{kl} = 0,1 \qquad \forall i \in I, j \in J, k \in K, l \in L \qquad (23)$$

The objective function (1) sums up the delay time for all outbound trucks and the waiting time of all inbound trucks. Constraint (2) ensures the flow conservation for all the inbound trucks. Constraint (3) states that each inbound truck must be assigned to exactly one receiving dock *k*. Since a dock can be left unused, constraint (4) enforces that every receiving dock serves at most one truck at a time. Constraint (5) ensures the flow conservation for all the outbound trucks. Constraint (6) guarantees that each outbound truck is assigned to exactly one shipping dock *l*. Constraint (7) enforces that every shipping dock serves at most one truck at a time. Constraints (8), (9) and (10) jointly define variable *z* which represent the logic relationship among *x* and *y*. z_{ij}^{kl} is set to 1 if both inbound truck *i* is assigned to receiving dock *k* and outbound truck *j* is assigned to shipping dock *l*. Constraint (11)-(12) make a valid sequence for arriving and departing times for the inbound trucks assigned to the same dock. Constraints (13) and (14) ensure the start and end time of the first truck at each dock should be larger than the dock's starting available time. Constraints (15) states that the inbound truck can only be served after its arrival time. Constraints (16)-(17) function in a similar manner for the outbound trucks. Constraint (18) connects the departure time for an outbound truck to the arrival time of an inbound truck if a flow must be transferred from that inbound truck. Constraints (19)-(20) are the nonnegativity constraints. Constraints (21)-(23) are the integrality constraints.

3. Reduced Variable Neighborhood Search Algorithm

Variable neighborhood search (VNS) algorithm initiated by Mladenović and Hansen [29] is a top-level methodology for solving combinatorial and global optimization problems. Its basic idea is to successively explore a set of predefined multi-neighborhoods to provide a better solution. It explores either at random or systematically a set of neighborhoods to get different local optima and to escape from local optima. VNS exploits the fact that using various neighborhoods in local search may generate different local optima and that the global optima is a local optima for a given neighborhood. Different neighborhood structures can be exploited in both deterministic and stochastic ways.

Reduced variable neighborhood search (RVNS) is a variant of VNS. This method is obtained if random points are selected from $N_k(x)$ without being followed by the time-consuming descent exploration of the neighborhood. RVNS is useful for very large instances for which local search is costly. It is observed that the best value for the parameter is often 2. In addition, the maximum number of iterations between two improvements is usually used as stopping condition (Hansen et al., 2010). The procedure of RVNS starts when a set of neighborhood structures N_k ($k = 1, \ldots, k_{max}$) are defined. Then a random initial solution is generated. For each iteration a solution x' is generated by applying the first move operator. The current solution is replaced by the new solution x' if and only if the objective function value is improved by the new solution then $x = x'$. If the solution is not improved, the algorithm moves to the next neighborhood N_{k+1}. Then another neighborhood structure generates a new neighbor x'', which will replace the current solution if and only if the objective function value is improved. The stopping criterion of the algorithm can be a user-defined computing time or a maximum number of iterations. In our experiments reported in the next section, we use the latter criterion.

Three different RVNS algorithms are proposed in order to test which one performs better for the dock assignment and truck scheduling problem. The difference among these three RVNS is the neighborhood search mechanism. Since our problem tackles on both inbound and outbound truck scheduling and dock assignment. The neighborhood for inbound and outbound truck scheduling and dock assignment can be searched either sequentially or simultaneously. The first RVNS (RVNS1) tries to search both inbound and outbound trucks simultaneously. The second RVNS (RVNS2) will find the neighborhood for the inbound trucks and fixed the outbound trucks. Once the inbound truck neighborhood search is done, its solution is fixed and then move to the outbound truck neighborhood. The third RVNS (RVNS3) will randomly choose inbound or outbound trucks to search. The solution representation and neighborhood structure are described in details in the next sections. Note that each local search could use first improvement (a move made as soon as an improvement is found) or best improvement strategy (a move to the best solution in the neighborhood). For these three RVNS, we use the first improvement approach.

3.1 Solution Representation

Each solution is broken down into two permutations, one associated to inbound trucks and the other one to outbound trucks. Both permutations represent the assignment of a truck to a dock and a position in the sequence to be served. Docks are separated from each other by a zero. It is important to notice that in a feasible solution each dock serves at least one truck, therefore the solution cannot contain a zero in the beginning or at the end of the permutation, neither have two zeros in a row. The trucks assigned to the same dock will be served based on the sequence in the solution.

Figure 1 shows an example for six trucks and 3 docks. The solution represents that trucks 1 and 2 are assigned to dock 1 and truck 1 precedes truck 2. Trucks 4 and 6 are assigned to dock 2 and truck 4 immediately precedes truck 6. Trucks 3 and 5 are assigned to dock 3 and truck 5 will be served right after truck 3 is processed.

1	2	0	4	6	0	3	5

Figure 1. Scheme of a solution

3.2 Neighborhood Structures

In a local search algorithm a neighborhood structure is designed by introducing moves from one solution to another. The neighborhood structures with which the neighboring solutions are determined to move to is one of the key elements of algorithms that use neighborhood structures. Therefore the performance of the algorithm significantly depends on the efficiency of the neighborhood structures. For permutations representing sequencing and scheduling problems position-based neighborhoods are largely used. Two position-based neighborhood structures were selected: swap and modified insert. In the case one of the neighborhood structures is about to make a violation, then the move is forbidden and a new neighbor out of the same neighborhood structure is generated.

When applying a neighborhood structure two possible effects in the permutation can be seen, one is if the movement involve merely trucks (not dock separators), then the dock and/or position trucks occupy are interchanged, in this case, any change in the number of trucks a dock serves is performed. On the other hand, if the movement involves trucks and dock separators, then the number of trucks a dock serve can be decreased or incremented, having the possibility of having some docks idle meanwhile others are busy.

3.2.1 Swap

In the swap operator two random positions are generated, and then the corresponding trucks (or docks separators) associated to those positions are exchanged. This neighborhood structure is illustrated in figure 2.

1	0	2	4	6	0	3	5	Before
1	4	2	0	6	0	3	5	After

Figure 2. Swap operator

As it can be seen in the example a dock separator (which is zero) and truck 4 exchange their positions. This swap of positions changed completely the solution, since not only truck 4 was moved to dock 1 but also trucks 1 and 2 are now served at dock 1.

3.2.2 Modified Insert

Through this structure, three random numbers are generated. The first two represent the two elements in the permutation that will be subject to a backward or forward insert operation. The third number is uniform randomly generated between zero and one, if its value is less than or equal than 0.5 then a forward insertion operation is performed, on the other hand if it is greater than 0.5 a backward insertion operation is performed. The forward insertion move tries to find a better solution by moving a truck from its current dock to those docks on the left side or from its current sequence to earlier sequence. The backward insertion operator work in a similar way, but this time the selected truck is moved to a later sequence at the same dock or to other docks on the right side.

1	0	2	4	6	0	3	5	Before
1	0	3	2	4	6	0	5	After (3^{rd} RAN= 0.15)
1	0	4	6	0	3	2	5	After (3^{rd} RAN= 0.70)

Figure 3. Modified insert operator

This neighborhood structure is illustrated in figure 3. In the example it can be seen a permutation where the first two random numbers generated correspond to positions 3 and 7 (which are associated to trucks 2 and 3). If the third number generated is 0.15 then the truck occupying the greatest position is inserted right behind the number occupying the smaller position. If the third random number is 0.70 then the truck associated with the smaller position is inserted right in front of the truck occupying the larger position generated.

4. Results

The proposed reduced variable neighborhood search algorithms were coded in Visual C++ 2008 and run on a computer with Inter Core 2 Duo @ 3.00GHz, 1.99GB of RAM and Windows XP operating system. The instances tested were generated based on the flow pattern in Carlo [13]. Each one of the RVNS algorithm frameworks presented was tested and compared with the solution given by Gurobi optimizer when the running time is limited to two hours. Each instance is run for 30 times.

Each RVNS algorithm was tested and compared with the solution given by Gurobi optimizer with the running time limited of two hours. The instance size is represented by $a \times b \times c \times d$, where a and b represent number of inbound trucks and outbound trucks, respective, and c and d represent the number of inbound docks and outbound docks, respectively. The number of inbound and outbound trucks ranges from 8 to 48, while the number of receiving and shipping docks ranges from 4 to 16. The number of iterations for the RVNS is 10000 and the number of neighborhoods is 2.

Table 1-4 show the results for three different RVNS algorithms in different sizes of instances. Table 1 presents the results for the smallest size instances. The column headings are the instance number n, the optimal solution and CPU time in seconds of Gurobi, the best result and average CPU time for each RVNS algorithm out of 30 runs. All three RVNS can find 19 optimal solutions out of 20 instances. RVNS2 provide the best average solutions. The computational times are much smaller than the time used by Gurobi. This indicates that the RVNS can be an efficient and effective algorithm to find good solutions.

Table 1. Results for the 8×8×4×4

n	**Gurobi**		**RVNS1**		**RVNS2**		**RVNS3**	
	f(x)	**CPU (Sec)**	**Best**	**CPU (Sec)**	**Best**	**CPU (Sec)**	**Best**	**CPU (Sec)**
1	310	11946	315	0.11	**310**	0.19	315	0.09
2	456	10438	**456**	0.11	**456**	0.19	**456**	0.09
3	489	11836	**489**	0.13	**489**	0.16	**489**	0.09
4	268	10655	**268**	0.08	**268**	0.20	**268**	0.09
5	352	8779	**352**	0.09	**352**	0.19	**352**	0.11
6	439	8319	**439**	0.09	**439**	0.17	**439**	0.09
7	436	10432	**436**	0.09	**436**	0.19	**436**	0.09
8	431	8167	**431**	0.09	**431**	0.17	**431**	0.09
9	463	9201	**463**	0.11	**463**	0.20	**463**	0.09
10	377	10288	**377**	0.13	**377**	0.2	**377**	0.09
11	537	8153	**537**	0.13	541	0.16	**537**	0.08
12	471	10525	**471**	0.09	**471**	0.19	**471**	0.09
13	827	11635	**827**	0.09	**827**	0.19	**827**	0.09
14	878	8115	**878**	0.09	**878**	0.20	**878**	0.08
15	282	8379	**282**	0.09	**282**	0.22	**282**	0.09
16	579	8290	**579**	0.09	**579**	0.19	**579**	0.09
17	535	9449	**535**	0.11	**535**	0.19	**535**	0.09
18	367	11459	**367**	0.11	**367**	0.17	**367**	0.09
19	528	9969	**528**	0.11	**528**	0.19	**528**	0.09
20	305	10316	**305**	0.09	**305**	0.20	**305**	0.08
Avg.	466.50	9817.55	466.75	0.10	**466.70**	0.19	466.75	0.09

Table 2 shows the results for the instances with 16 trucks and 8 docks. Gurobi cannot solve the problem within 2 hours computation time limit. We only provide the solution that Gurobi obtain after the time limit. Our RVNS algorithms provide better solutions than those obtained by solving the integer programming formulation with optimization software Gurobi. The Gurobi solver ran out of memory for larger size instances in tables 3 and 4 which have more than 32 trucks and 16 docks for both inbound and outbound operations. We only list the results for all RNVS algorithms. RVNS with sequential search for inbound and outbound neighborhood (RVNS2) needs longer time to implement. For the 32×32×16×16 problems, three algorithms do not show significant difference in terms of average solution. However, for the largest set of instances 48×48×16×16, RVNS2 provides much better results compared with the other two RVNS algorithms.

Table 2. Results for the 16×16×8×8

n	**Gurobi (2hrs.)**	**RVNS1**		**RVNS2**		**RVNS3**	
		Best	**CPU (Sec)**	**Best**	**CPU (Sec)**	**Best**	**CPU (Sec)**
1	1269	**1175**	0.24	**1175**	0.45	**1175**	0.23
2	1245	1,162	0.27	**1160**	0.50	1162	0.23
3	1061	**842**	0.25	847	0.42	**842**	0.25
4	1024	**966**	0.28	**966**	0.44	**966**	0.27
5	1111	1,069	0.22	1070	0.42	**1065**	0.22
6	1359	**1318**	0.25	**1318**	0.52	**1318**	0.25
7	954	**893**	0.24	**893**	0.45	**893**	0.25
8	856	**831**	0.25	**831**	0.47	**831**	0.25
9	783	**759**	0.24	**759**	0.52	**759**	0.22
10	991	**961**	0.25	**961**	0.50	**961**	0.22
11	1353	1,278	0.25	**1276**	0.50	**1276**	0.27
12	910	**827**	0.24	**827**	0.48	**827**	0.25
13	1335	**1297**	0.25	**1297**	0.50	1301	0.25
14	1688	1,559	0.25	1574	0.50	**1555**	0.24
15	1395	**1318**	0.22	**1318**	0.50	**1318**	0.25
16	991	**949**	0.28	**949**	0.47	**949**	0.27
17	917	**733**	0.28	**733**	0.41	**733**	0.23
18	1916	**825**	0.24	**825**	0.45	**825**	0.27
19	1171	**1057**	0.25	1063	0.45	**1057**	0.22
20	534	**402**	0.27	**402**	0.49	**402**	0.25
Avg.	1143.15	1011.05	0.25	1012.20	0.47	**1010.75**	0.24

In order to compare the results yielded by the three different frameworks tested, a t-test was performed among the three frameworks (RVNS1 vs. RVNS2, RVNS1 vs. RVNS3, and RVNS2 vs. RVNS3). Where the null hypothesis (H_0) states that there is no significant difference between the means of the samples under comparison. It is important to notice that H_0 is accepted when the P-Value is greater or equal than 0.05.

Table 3. Results for the 32×32×16×16

n	**RVNS1**		**RVNS2**		**RVNS3**	
	Best	**CPU (Sec)**	**Best**	**CPU (Sec)**	**Best**	**CPU (Sec)**
1	**1583**	0.75	1601	1.45	1591	0.75
2	1202	0.78	1201	1.47	**1198**	0.77
3	**2560**	0.80	**2560**	1.70	**2560**	0.77
4	2200	0.69	**2187**	1.28	2198	0.77
5	**1574**	0.70	**1574**	1.47	1579	0.72
6	**1922**	0.70	1926	1.45	1923	0.72
7	1608	0.69	**1605**	1.38	**1605**	0.72
8	18310	0.80	18310	1.61	**18309**	0.84
9	**1743**	0.75	**1743**	1.50	**1743**	0.77
10	1813	0.75	1812	1.36	**1805**	0.69
Avg.	3451.50	0.74	3451.90	1.47	**3451.10**	0.75

Table 4. Results for the 48×48×16×16

n	**RVNS1**		**RVNS2**		**RVNS3**	
	Best	**CPU (Sec)**	**Best**	**CPU (Sec)**	**Best**	**CPU (Sec)**
1	3924	1.03	**3889**	2.28	3948	1.00
2	3792	1.05	**3718**	2.03	3800	1.06
3	3329	1.06	**3255**	2.02	3306	1.03
4	4756	1.03	**4656**	1.98	4712	1.03
5	4461	1.14	**4407**	1.95	4458	1.05
6	3417	1.05	**3378**	2.31	3409	1.09
7	3887	1.06	**3781**	2.16	3859	1.08
8	5306	1.14	**5147**	2.00	5312	1.03
9	3665	1.08	**3621**	2.23	3648	1.08
10	4734	1.20	**4671**	2.20	4740	1.06
Avg.	4127.10	1.08	**4052.30**	2.12	4119.20	1.05

The tests were performed in Minitab 15.1. The summary of the results are presented in the Table 5. The difference in the performance of the three algorithms is not significant for small and medium instances, however for large instances, there is a significant difference between the performance of RVNS1 compared with RVNS2, and RVNS2 compared with RVNS3. In both cases RVNS2 presented best solution among three RVNS algorithms.

Table 5. t-test performance comparison among the three RVNS frameworks

Instance Set	**RVNS_1 vs. RVNS_2**	**RVNS_1 vs. RVNS_3**	**RVNS_2 vs. RVNS_3**
8x8x4x4	Accept H_0	Accept H_0	Accept H_0
16x16x8x8	Accept H_0	Accept H_0	Accept H_0
32x32x16x16	Accept H_0	Accept H_0	Accept H_0
48x48x16x16	Reject H_0	Accept H_0	Reject H_0

5. Conclusion

In this research a new mixed integer formulation is proposed to solve the multi-door cross-docking truck scheduling and door assignment problem. The problem involves simultaneous door assignment and sequencing of inbound trucks subject to known inbound arrival schedule and a fixed outbound truck departure schedule. The objective is to minimize the total waiting time for inbound trucks and tardiness of outbound trucks. Since this problem encompasses the combination of two NP-Hard problems it cannot be solved in large instances within a reasonable computational time. The problem is solved reduced variable neighborhood search (RVNS) algorithm with different neighborhood structures. Four different size instances are tested for the proposed algorithm and compared with solutions found by Gurobi solver. For the small size instances, the proposed RVNS can find the optimal solutions within short computational times. For the medium to large size instances, the RVNS algorithms outperform Gurobi in terms of solution quality and computing time.

Future work should be conducted from the following perspectives. First, the arrival schedule could be also the decision to match the outbound truck schedule. Second, integrate truck sequencing problem with time window consideration on outbound truck schedule. Third, combining the scheduling and assignment problem with other cross-docking related problem such as vehicle routing problem is also desirable.

References

[1] Agustina, D., Lee, C. K. M. and Piplani, R., “A Review: Mathematical Models for Cross Docking Planning,” *International Journal of Engineering Business Management*, 2, 47-54 (2010).

[2] Alpan, G., Ladier, A. L., Larbi, R. and Penz, B., “Heuristic Solutions for Transshipment Problems in a Multiple Door Cross Docking Warehouse,” *Computers & Industrial Engineering*, 61, 402-408 (2011).
[3] Alpan, G., Larbi, R. and Penz, B., “A Bounded Dynamic Programming Approach to Schedule Operations in a Cross Docking Platform,” *Computers & Industrial Engineering*, 60, 385-396 (2010).
[4] Baptise, P. and Maknoon M. Y., “Cross-docking: Scheduling of Incoming and Outgoing Semi Trailers,” *19th International Conference on Production Research-ICPR*, (2009).
[5] Bartholdi, J. J. and Gue, K. R., “Reducing Labor Cost in an LTL Crossdocking Terminal,” *Operations Research*, 48, 823-832 (2000).
[6] Bartholdi, J. J. and Gue, K. R., “The Best Shape for a Crossdock,” *Transportation Science*, 38, 235-244 (2004).
[7] Boloori Arabani, A. R., Fatemi Ghomi, S. M. T. and Zandieh, M., “A Multi-criteria Cross-docking Scheduling with Just-in-time Approach,” *International Journal of Advanced Manufacturing Technology*, 49, 741-756 (2010).
[8] Boloori Arabani, A. R., Fatemi Ghomi, S. M. T. and Zandieh, M., “Meta-heuristics Implementation for Scheduling of Trucks in a Cross-docking System with Temporary Storage,” *Expert Systems with Applications,* 38, 1964-1979 (2011).
[9] Boysen, N., Fliedner, M. and Scholl A., “Scheduling Inbound and Outbound Trucks at Cross Docking Terminals,” *OR Spectrum,* 32, 135-161 (2010).
[10] Boysen, N., “Truck Scheduling at Zero-inventory Cross Docking Terminals,” *Computers & Operations Research*, 37, 32-41 (2009).
[11] Boysen, N. and Fliedner, M., “Cross Dock Scheduling: Classification, Literature Review and Research Agenda. *Omega*, 38, 413-422 (2009).
[12] Bozer, Y. A. and Carlo, H. J., “Optimizing Inbound and Outbound Door Assignments in Less-than-truckload Crossdocks,” *IIE Transactions*, 40, 1007-1018 (2008).
[13] Carlo, H. J., *Door Assignment and Sequencing Problems in Crossdocks and Container Terminals,* Ph.D. dissertation, Industrial and Operations Engineering Department, University of Michigan, MI, USA (2007).
[14] Chen, F. and Lee, C. Y., “Minimizing the Makespan in a Two-machine Cross-docking Flow Shop Problem,” *European Journal of Operational Research,* 193, 59-72 (2009).
[15] Chen, F. and Song, K., “Minimizing Makespan in Two-stage Hybrid Cross Docking Scheduling Problem,” Computers & Operations Research, 36, 2066-2073 (2009).
[16] Cohen, Y. and Keren, B., “Trailer to Dock Assignment in a Synchronous Cross-dock Operation,” *International Journal of Logistics System and Management*, 5, 574-590 (2009).
[17] Forouharfard, S. and Zandieh, M., “An Imperialist Competitive Algorithm to Schedule of Receiving and Shipping Trucks in Cross-docking Systems,”

International Journal of Advanced Manufacturing Technology, 51, 1179-1193 (2010).
[18] Garey, M. R. and Johnson, D. S., *Computers and Intractability*: *A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, NY, (1979).
[19] Hansen, P., Mladenović, N. and Moreno-Perez, J. A. "Variable Neighborhood Search: Methods and Applications," *4OR*, 6, 319–360 (2010).
[20] Larbi, R., Alpan, G., Baptiste, P. and Penz B., "Scheduling of Transshipment Operations in a Single Strip and Stack Docks Crossdock," *19th International Conference on Production Research, Chile* (2009).
[21] Larbi, R., Alpan, G., Baptiste, P. and Penz B., "Scheduling Cross Docking Operations under Full, Partial and No Information on Inbound Arrivals," *Computers & Operations Research,* 38, 889-900 (2011).
[22] Li, Z. P, Low, M. Y. H, Shakeri, M. and Lim, Y.G., "Crossdocking Planning and Scheduling: Problems and Algorithms," *SIMTech Technical Reports*, 10, 159-167 (2009).
[23] Liao, T. W., Egbelu, P. J. and Chang, P. C., "Simultaneous Dock Assignment and Sequencing of Inbound Trucks under a Fixed Outbound Truck Schedule in Multi-door Cross Docking Operations", *International Journal of Production Economics*, (2012), http://dx.doi.org/10.1016/j.ijpe.2012.03.037.
[24] McWilliams, D. L., Stanfield, P. M. and Geiger, C. D., "The Parcel Hub Scheduling Problem: A Simulation-based Solution Approach," *Computers & Industrial Engineering*, 49, 393-412 (2005).
[25] McWilliams D. L., Stanfield, P. M. and Geiger, C. D., "Minimizing the Completion Time of the Transfer Operations in a Central Parcel Consolidation Terminal with Unequal-batch-size Inbound Trailers," *Computers & Industrial Engineering*, 54, 709-20 (2008).
[26] McWilliams, D. L., "Genetic-based Scheduling to Solve the Parcel Hub Scheduling Problem," *Computers & Industrial Engineering*, 56, 1607-1616 (2009).
[27] McWilliams, D. L., "Iterative Improvement to Solve the Parcel Hub Scheduling Problem," *Computers & Industrial Engineering*, 59, 136-144 (2010).
[28] Miao, Z., Lim, A. and Ma, H., "Truck Dock Assignment Problem with Operational Time Constraint within Crossdocks," *European Journal of Operational Research*, 192, 105-115 (2007).
[29] Mladenović, N. and Hansen, P., "Variable Neighborhood Search," *Computers & Operations Research*, 24, 1097-1100 (1997).
[30] Napolitano, M., *Making the move to cross docking: A practical guide to planning, design, and implementing a cross dock operation.* Warehousing Education and Research Council, Oak Brook, Illinois (2000).
[31] Oh Y, Hwang H, Cha CN, Lee S. "A Dock-door Assignment Problem for the Korean Mail Distribution Center," *Computers & Industrial Engineering*, 51, 288-96 (2006).

[32] Stalk, G., Evans, P. and Shulman, L. E., "Competing on Capability: The New Rules of Corporate Strategy," *Harvard Business Review*, 70, 57-69 (1992).
[33] Tsui, L. Y. and Chang, C. H., "A Microcomputer Based Decision Support Tool for Assigning Dock Doors in Freight Yards," *Computers & Industrial Engineering*, 19, 309-312 (1990).
[34] Tsui, L. Y. and Chang, C. H., "An Optimal Solution to a Dock Door Assignment Problem," *Computers & Industrial Engineering*, 23, 283-286 (1992).
[35] Vahdani, B. and Zandieh, M., "Scheduling Trucks in Cross-docking Systems: Robust Meta-heuristics," *Computers & Industrial Engineering,* 58, 12-24 (2010).
[36] Van Belle, J., Valckenaers, P. and Cattrysse, D., "Cross-docking: State of the Art," *Omega*, 40, 827-846 (2012).
[37] Wang, J. F and Regan, A., "Real-time Trailer Scheduling for Crossdock Operations," *Transportation Journal*, 47, 5-20 (2008).
[38] Witt, C. E., "Crossdocking: Concepts Demand Choice," *Material Handling Engineering*, 53, 44-49 (1998).
[39] Yu, F. T., *Door Allocation Problem at Intermediate Consolidation Terminals of Less-than-Truckload Motor Carriers,* Ph.D dissertation, Industrial and Operations Engineering Department, University of Michigan, MI, USA (2004).
[40] Yu, V. F., Sharma, D. and Murty, K. G., "Door Allocations to Origins and Destinations at Less-than-truckload Trucking Terminals," *Journal of Industrial and Systems Engineering*, 2, 1-15 (2008).
[41] Yu, W. and Egbelu, P. J., "Scheduling of Inbound and Outbound Trucks in Cross Docking Systems with Temporary Storage," *European Journal of Operational Research*, 184, 377-396 (2008).
[42] 2011 Cross-Docking Trends Report. Whitepaper series. Saddle Creek Corporation.