Large Neighborhood Search for Outbound Truck Scheduling Problem in a Cross-docking

Warisa Wisittipanich and Nattapong Kamsura
Department of Industrial Engineering
Excellence Center in Logistics and Supply Chain Management
Faculty of Engineering, Chiang Mai University
warisa.o@gmail.com, kamsura01@gmail.com

Abstract

This paper studies the multi-objective outbound truck scheduling problem in a multi-door cross-docking terminal. The key characteristic of the problem is that each truck can travel multiple trips to deliver products to different sets of customers. The goal is to find the door assignment, truck sequencing, and the departure time of outbound trucks so that each outbound truck leave the dock door as close to their due time as much as possible. The problem is NP-hard in which an exact solution method consumes high computational time to obtain optimal solution especially in the large-size problems. For this reason, this paper proposes an application of a metaheuristic called Large Neighborhood Search (LNS) to deal with the problem complexity. In LNS, three destroy and two repair strategies are proposed with an aim to improve the solution quality. The numerical results show that the proposed LNS yields solutions closed to those obtained from an exact method in small-sized problems with fast computing time. For the larger-size problems, the proposed LNS can find solutions in relatively fast computing time while an exact method cannot find optimal solution in an acceptable time.

Keywords
Truck Scheduling, Outbound Truck, Total Earliness, Total Tardiness, Large Neighborhood Search

1. Introduction

Cross-docking is one of the logistics strategies to increase customer response and reduces lead time. The main concept of cross-docking is to directly transfer incoming products from incoming vehicles to outgoing vehicles without storing them in between so that storing and retrieving functions are minimized. The problem in cross-docking is mainly due to the coordination of operations to ensure rapid transshipment. Truck scheduling becomes one of the most difficult and challenging decision problems since the number of dock doors in the real-world practice can be up to 500 doors. Therefore, mataheuristic approaches are required to deal with such high-complexity of the problem.

Various research have been conducted on different aspects to deal with truck scheduling problem in cross-docking. Miao et al. (2009) presented an application of Tabu search (TS) and Genetic algorithm (GA) for solving truck scheduling problems in cross-docking with two objectives; minimization of the operational cost and minimization of total number of unfulfilled shipments. They used ILOG CPLEX to find optimum solutions and compared solutions derived from the two metaheuristics. The results showed that the metaheuristics, especially TS, outperform the CPLEX solver in nearly all test cases adapted from industrial applications. Soltani and Sadjadi (2010) presented an application of two hybrid metaheuristics named Hybrid Simulated Annealing and Hybrid Variable Neighborhood Search to solve inbound and outbound truck scheduling problem in a cross-docking with the objective to minimize the total flow time. The proposed algorithms were compared with TS, and the results showed that their proposed metaheuristics were better than TS. Alpan et al. (2011) proposed the Dynamic Programming (DP) and three metaheuristics for the purpose to compare the solutions quality in truck scheduling problem. It was found that the results obtained from DP yielded higher solution quality but consumed higher computational time than obtained by metaheuristics. Arabani et al. (2011) proposed five metaheuristics to minimize total makespan in truck scheduling problem. They also investigated the effects of using simulation on the success of a multi-door cross-docking. Agustina et al. (2014) focused on the integration of truck scheduling and vehicle routing problem in cross-docking to ensure that food can be delivered just-in-time at minimum cost of delivery, including inventory holding and transportation costs, and the penalty costs of early or tardy deliveries. The problem was modeled as a mixed integer linear program in CPLEX, but only small size instances can be solved within reasonable time. In large size problem, the authors recommended that the size of the solution space should be reduced by modifying the customer zones and using hard- instead of soft-time windows. Wisittipanich and Hengmeechai, (2015) presented the door assignment and truck scheduling problem in cross-docking terminals with multi-door according to Just-In-Time concept. This study presented an implementation of a multi-objective differential evolution (MODE) to obtain a Pareto frontier, with the objective to minimize total earliness and total tardiness of all trucks. Keshtzari et al. (2016) proposed the improved version of the mixed integer programming model for solving small-size truck scheduling problems in the cross-docking terminals. The new model showed its efficiency over the previous ones. In addition, the novel metaheuristics based on Particle Swarm Optimization (PSO) was also proposed for solving the larger size problems. The results showed that the new algorithm outperformed the other two state-of-the-art metaheuristics. Assadi and M. Bagheri, (2016) studied the truck scheduling problem in the multiple door cross-docking system with unequal ready times with the objective to minimize the weighted number of tardy trucks. Two metaheuristics, Simulated Annealing (SA) and GA were used to solve the problem, and the results showed the efficiency of the proposed metaheuristics. Behnamian et al. (2018) studied location-allocation and truck scheduling in multiple cross-dockings where the loads are transferred from origins to destinations through cross-docking facilities. In addition, this paper considered the breakdown of trucks. The author modelled the problem in two stages; the first stage is for the location of cross-docking centers and the second stage is for truck scheduling in multiple cross-dockings. However, due to the difficulty of obtaining the optimum solutions in medium and large scale problems, four metaheuristic algorithms, i.e., GA, SA, Differential Evolution (DE), and hybrid algorithms were proposed. The result showed that SA was the best algorithm among the four algorithms.

This paper focuses on outbound truck scheduling problem in a multi-door cross-docking terminal. The goal is to find the door assignment, truck sequencing, and the departure time of outbound trucks so that each outbound truck leave the dock door as close to their due time as much as possible. The key characteristic of the problem is that each truck can travel multiple trips to deliver products to different sets of customers. Thus, the assignment and sequencing of trucks needs to be made for each delivery. The problem is formulated as mixed integer programming (MIP) model as referred to the previous study of Wisittipanich and Kamsura (2018). The objective of the model is to find truck schedule with minimum total earliness and minimum total tardiness. Since the problem is NP-hard, the mathematical model can only find solution in small size problems. For this reason, this paper proposes Large Neighborhood Search (LNS) algorithm with different destroy and repair strategies to handle large size problems.

2. Problem Description

This paper studies the outbound truck scheduling problem in multi-door cross-docking system by considering two important criteria. First, the departure due time of each outbound truck k ($k = 1, 2, 3, \ldots, K$) is predetermined in advance in order to ensure on-time delivery. Thus, each outbound truck is expected to leave the dock door h ($h = 1, 2, 3, \ldots, H$) at its predetermined due time as close as possible. Second, each outbound truck can make multiple delivery trips i ($i = 1, 2, 3, \ldots, I$) to different sets of customers due to the limitation of truck capacity. The scheduling problem composes of two sub-problems; the assignment of all trucks to dock doors and the sequencing of trucks at each door. The objective of the problem is to find outbound truck schedule to minimize the total earliness and the total tardiness of all outbound trucks. The problem is formulated based on the following assumptions:

- Outbound trucks are heterogeneous with different capacities.
- Products are either loaded to the full truck capacity or subjected to customer demand.
- The total loading time of products to an outbound truck is set according to the truck capacity.
- The delivery route of each truck is predetermined in advance. Thus, the traveling time of each delivery trip is known.
- The numbers of delivery trips for all trucks are assumed to be equal.

According to the previous study of Wisittipanich and Kamsura (2018), the outbound truck scheduling problem is formulated as mixed integer programming (MIP) model as follows.

Indices

I : A set of delivery round i, where $i \in I$, $i = 1, 2, ..., I$

K : A set of outbound truck k, where $k \in K$, $k = 1, 2, ..., K$

H : A set of outbound door *h*, where $h \in H$, $h = 1, 2, ..., H$

Parameters

LT_{ik} : Loading time of outbound truck k round i

UT_{ik} : Unloading time an outbound truck k round i to customers

TV_{ik} : Travel time of an outbound truck k round i

due_{ik} : Due time of an outbound truck k round i

M : Big Number

Decision variables

et_{ik} : Arrival time of an outbound truck k round i

dt_{ik} : Departure time an outbound truck k round i

$X_{ii'kk'}$: { 1, if an outbound truck *k* delivering for round *i* is assigned to the dock door after an outbound truck *k'* delivering for round *i'*; 0, otherwise

P_{ikh} : { 1, if an outbound truck *k* delivering for round *i* is assigned to an outbound door *h*; 0, otherwise

The mathematical model of the problem can be formulated as follows.

Objective function

Two objective functions are considered in this study; minimization of the total earliness and minimization of the total tardiness of outbound trucks as shown in equation (1) and (2), respectively.

$$\text{Min } z_1 = \sum_{i=1}^{I}\sum_{k=1}^{K}\max(0, er_{ik}) \qquad (1)$$

$$\text{Min } z_2 = \sum_{i=1}^{I}\sum_{k=1}^{K}\max(0, ta_{ik}) \qquad (2)$$

Where

er_{ik} : Earliness time of outbound truck *k* traveling round *i*

ta_{ik} : Tardiness time of outbound truck *k* traveling round *i*

Constraints

Door assignment: Constraint (3) ensures that each outbound truck is assigned to an outbound door.

$$\sum_{h=1}^{H} P_{ikh} = 1; \forall_{i,k}\{i = 1, 2, ..., I; k = 1, 2, ..., K\} \qquad (3)$$

Truck sequences: Constraints (4) - (5) represent the conflict constraints to ensure that each outbound door can operate only one outbound truck at a time.

$$et_{ik} \geq et_{i'k'} - M(3 - X_{ii'kk'} - P_{ikh} - P_{i'k'h}); \forall_{i,i',i\neq i',k,k',k\neq k',h} \{i = 1, 2, ..., I; k = 1, 2, ..., K; h = 1, 2, ..., H\} \qquad (4)$$

$$et_{i'k'} \geq et_{ik} - M(X_{ii'kk'} + 2 - P_{ikh} - P_{i'k'h}); \forall_{i,i',i\neq i',k,k',k\neq k',h} \{i = 1, 2, ..., I; k = 1, 2, ..., K; h = 1, 2, ..., H\} \qquad (5)$$

Arrival time of truck to the dock door: Constraint (6) states that the arrival time of an outbound truck *k*, round *i* must be greater than or equal to the departure time of its previous round (an outbound truck *k*, round *i*-1).

$$et_{ik} \geq dt_{i-1k} + TV_{i-1k} + UT_{i-1k;} \forall_{i,i>1,k} \quad \{i = 1, 2, ..., I; k = 1, 2, ..., K\} \tag{6}$$

Departure time of truck from the dock door: Constraint (7) states that the departure time of truck *k* must be greater than or equal to its arrival time plus loading time.

$$dt_{ik} \geq et_{ik} + LT_{ik;} \forall_{i,k} \quad \{i = 1, 2, ..., I; k = 1, 2, ..., K\} \tag{7}$$

Earliness and tardiness conditions: Constraint (8) and (9) specify the earliness time (er_{ik}) and tardiness time (ta_{ik}) of outbound truck *k* traveling round *i*, respectively.

$$er_{ik} \geq due_{ik} - dt_{ik}; \forall_{i,k} \{i = 1, 2, ..., I; k = 1, 2, ..., K\} \tag{8}$$

$$ta_{ik} \geq dt_{ik} - due_{ik}; \forall_{i,k} \{i = 1, 2, ..., I; k = 1, 2, ..., K\} \tag{9}$$

Constraint (10) states that all variables are non-negative variables.

All variable $\geq$ 0 (10)

3. Proposed Algorithm

3.1 Large Neighborhood Search (LNS)

Large neighborhood search (LNS), proposed by Shaw in 1998, is a metaheuristic approach in which an initial is gradually improved by destroy and repair methods. While a destroy method destruct part of a solution, a repair method reconstructs the destroyed solution to obtain the better solution. The main concept behind LNS is that the large neighborhood allows the algorithm to navigate in the solution space easily even though the instance is tightly constrained (Pisinger and Ropke, 2010). The LNS pseudo-code is shown in Figure 1.

Algorithm 1 Large neighborhood search

```
input: a feasible solution x
x^b = x;
repeat
    x^t = r(d(x));
    if accept(x^t, x) then
        x = x^t;
    end if
    if c(x^t) < c(x^b) then
        x^b = x^t;
    end if
until stop criterion is met
return x^b
```

Figure 1. Pseudo-code of Large Neighborhood Search (Pisinger and Ropke, 2010)

According to Figure 1, the original LNS begins with an initial feasible solution that has already been found, for example by a simple construction heuristic. The variable x^b is the best solution obtained during the search, *x* is the current solution, and x^t is a temporary solution. The function *d(x)* is the destroy method that destructs parts of a solution and the function *r(x)* repairs the destroyed solution and return the feasible solution, x^t. If the new solution is accepted with better objective function $c(x^t)$, the best solution is updated. These processes are repeat until stopping criterion is met.

3.2 Application of LNS to the outbound truck scheduling problem

This paper proposes an application of LNS to the multi-objective outbound truck scheduling problem in cross-docking system. The goal is to find a set of non-dominated solutions of truck schedules with minimum total earliness and minimum total tardiness. The LNS framework for the truck scheduling problem is shown in Figure 2.

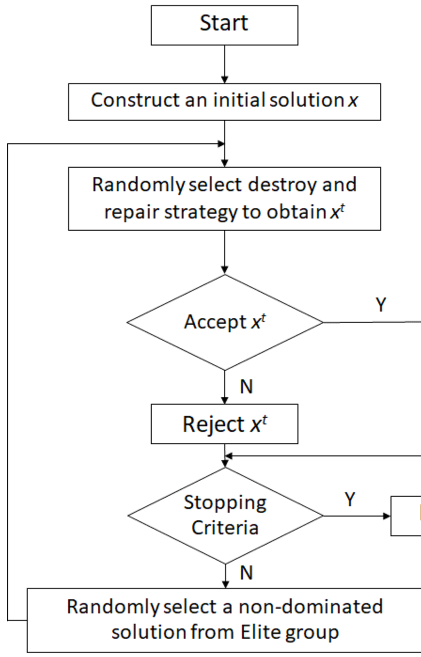

Figure 2. Proposed LNS framework

An example of a cross-docking with three outbound doors, four outbound trucks, and each truck travels two trips is used to illustrate the application of the proposed LNS. According to Figure 3, the LNS algorithm begins with constructing an initial solution. To construct an initial solution, the string with dimensions equal to the number of trucks multiplied by the number of trips is generated. Each dimension value is generated with uniform random number in range [0, 1]. Thus, in this example, a string with eight dimensions is generated. Then, the roulette wheel rule is adopted to assign each outbound truck to its corresponding outbound door. After achieving the assignment of trucks to doors, the earliness and the tardiness of each truck are computed as shown in Figure 4.

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	0.81	0.91	0.24	0.52	0.18	0.27	0.69	0.32

⬇

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	3	1	2	1	1	2	1

Figure 3. Solution Representation

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	3	1	2	1	1	2	1
Earliness Value	3	0	5	3	13	0	24	0
Tardiness Value	0	25	0	0	0	5	0	0

Figure 4. An initial solution with earliness and tardiness values

In this study, three destroy strategies and two repair strategies are developed with an aim to improve the quality of solutions. One destroy strategy and one repair strategy are randomly selected to obtain a new solution x^t. The details of how each destroy and repair strategy are applied to a solution is shown with the following example.

Destroy Strategies 1 (DS1)

The concept of DS1 is to destruct the assigned doors which contribute to earliness or tardiness. The destruction is made at the highest earliness or tardiness value at some degrees of destruction. Figure 5 shows an example of DS1 results at 50% degree of destruction. When there are more than one door with the same value of earliness or tardiness, the ties are broken arbitrarily. According to Figure 4, since the total number of doors are eight, four assigned doors are destroyed which are doors of truck 2 in the first round, truck 1, truck 2, and truck 3 in the second round..

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	~~3~~	1	2	~~1~~	~~1~~	~~2~~	1
Earliness Value	3	0	5	3	13	0	24	0
Tardiness Value	0	25	0	0	0	5	0	0

Figure 5. Illustration of DS1

Destroy Strategies 2 (DS2)

Since the assignment of trucks to doors in the first round highly affect the overall truck schedule in the next rounds, the idea of DS2 is to randomly destroy the assigned doors only in the first round at some degree of destruction. Figure 5 shows result after applying DS2 with 50% degree of destruction. As shown in Figure 6, since the position 2 and 4 are randomly selected, door 3 of truck 2 and door 2 of truck 4 in the first round are destroyed.

	Round 1	
Random Position	2	4

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	~~3~~	1	~~2~~	1	1	2	1

Figure 6. Illustration of DS2

Destroy Strategies 3 (DS3)

The idea of DS3 is to randomly destroy the assigned doors in each round equally. Figure 7 shows result after applying DS3 with 100% degree of destruction (50% degree of destruction for each round). According to Figure 7, door 3 of truck 2 and door 2 of truck 4 in the first round are destroyed. Similarly, door 1 of truck 2 and door 2 of truck 3 in the second round are destroyed.

	Round 1		**Round 2**	
Random Position	2	4	2	3

	Round 1				**Round 2**			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	~~3~~	1	~~2~~	1	~~1~~	~~2~~	1

Figure 7. Illustration of DS3

Repair Strategies 1 (RS1)

Repair strategy 1 (RS1) is designed to repair all destroyed solutions from DS1, DS2, and DS3. The concept of RS1 is to simply assign the new doors to the destroyed doors according to roulette wheel rule. As seen in Figure 8, the destroyed doors are repaired and new truck schedule is obtained.

Destroy Solution	Round 1				Round 2			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	~~3~~	1	~~2~~	1	~~1~~	~~2~~	1

	Round 1				Round 2			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Repair Door		2		1		2	3	

New Solution	Round 1				Round 2			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	2	1	1	1	2	3	1

Figure 8. Illustration of RS1

Repair Strategies 2 (RS2)

The idea of RS2 is to repair the destroyed solution by reassign the destroyed door to trucks based on random position rule. Similar to RS1, RS2 can be used with all destroyed strategies. Figure 9 illustrates the result after applying RS3. As shown in Figure 9, the random position is generated for each destroyed door, and the doors are assigned to the corresponding trucks. Then, new truck schedule is obtained.

Destroy Solution	Round 1				Round 2			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Door Assign	3	~~3~~	1	~~2~~	1	~~1~~	~~2~~	1

	Truck	Destroy Door	Random Position
Round 1	2	3	3
	4	2	1
Round 2	2	1	4
	3	2	2

Position	Repair Door	Truck	
1	2	2	**Round 1**
2	2	4	
3	3	2	**Round 2**
4	1	3	

New Solution	Round 1				Round 2			
Truck	**1**	**2**	**3**	**4**	**1**	**2**	**3**	**4**
Solution	3	2	1	2	1	3	1	1

Figure 9. Illustration of RS2

Once a new solution x^t is obtained from the selected destroy and repair methods, the objective function of new solution is checked according to acceptance criteria. If x^t is a non-dominated solution of x, the x^t is kept in the non-dominated solution set, called Elite group. Otherwise the solution x^t is rejected. Then, a non-dominated solution from Elite group is randomly selected to perform destroy and repair operations to obtain a new solution until the stopping criteria is met.

4. Computational Experiment

To evaluate the performance of the proposed LNS, 15 instances are generated. Each instance is characterized by its parameters *I*, *K* and *H*, which are the number of delivery rounds, the number of outbound trucks, and the number of outbound doors, respectively. The algorithm is implemented on the Microsoft Visual Studio version 2017, and the computational experiment is executed using a personal computer of Intel® Core™ i7-7500U CPU 2.70 GHz processor with 8GB RAM memory.

The proposed LNS for multi-objective outbound truck scheduling problem is applied on the test instances. The goal is to find truck schedule with minimization of earliness and minimization of tardiness simultaneously. In LNS, the stopping criteria is set to 1000 iterations. For each iteration, one destroy strategy and one repair strategy are randomly selected to obtain a new solution. The solutions obtained from the proposed LNS are expressed as a set of non-dominated solutions and compared with optimal solution obtained from LINGO optimization solver. The experimental results are illustrated in Table 1.

Table 1. Experimental results of the proposed LNS

Instance	Problem size			LINGO Solution		LNS Solution (Earliness, Tardiness)
	I	K	H	Minimize Earliness	Minimize Tardiness	
1	2	2	2	(0,0)	(48,0)	(18,30) (48,0)
2	3	2	2	(0,0)	(121,0)	(61,30) (121,0)
3	2	3	2	(0,131)	(47,30)	(18,151) (47,30)
4	2	3	3	(0,0)	(77,0)	(18,151) (47,30) (77,0)
5	3	3	2	(0,104)	(108,10)	(60,183) (120,63) (132,30)
6	2	4	2	(0,287)	(23,61)	(18,311) (38,61)
7	2	4	3	(0,50)	(48,30)	(18,311) (38,61) (67,30)
8	2	4	4	(0,0)	(97,0)	(18,311) (38,61) (67,30) (97,0)
9	3	4	2	(0,248)	N/A	(47,388) (112,183) (119,145) (122,103) (125,61)
10	3	4	4	(0,48)	N/A	(47,388) (112,183) (119,145) (122,103) (125,61) (184,30) (244,0)
11	2	5	3	(0,206)	N/A	(18,533) (36,161) (56,61)
12	2	7	3	N/A	N/A	(36,442) (55,191)
13	2	20	15	N/A	N/A	(90,2228) (108,1736) (126,944) (144,842) (145,753) (162,650) (189,488) (198,326) (216,284) (234,242) (252,200) (271,159)
14	2	20	10	N/A	N/A	(108,1226) (126,1004) (144,752) (162,530) (180,488) (181,459)
15	2	30	20	N/A	N/A	(163,2) (180,2024) (181,1845) (191,1735) (198,1532) (199,1293) (200,1474) (208,1302) (216,1160) (234,1028) (235,969) (253,897) (254,838) (270,734)

As mentioned earlier, since the problem is NP-hard, LINGO can only find optimal solution in small to medium size problem. When the problem size increases, LINGO cannot find solution in a reasonable time. However, the proposed LNS can find solution in all problem sizes. In addition, while LINGO can solve the problem for one objective at a time, LNS yields a set non-dominated solutions providing an advantage for decision makers to make decision based on their preference. Figure 10 illustrates a set of non-dominated solutions obtained from LNS for instance 13. In term of solution quality, LNS can find solutions closed to those obtained from LINGO in the case of tardiness minimization. Nevertheless, solutions obtained from LNS are inferior to those obtained from LINGO in the case of earliness minimization. This can be explained that, LINGO simply set the truck to depart after its due time so that the earliness in minimized whereas LNS allows trucks to load products after the assigned door is available. This may be a drawback of the proposed LNS which needs further improvement especially in the case of earliness minimization.

The performance of the proposed LNS is also evaluated in term of computational time. The comparison of computational time between LINGO optimization solver and the proposed LNS is shown in Figure 11. For small to medium size problems (problem 1 - 9), both solution methods can easily find the solution with fast computing time. However, in the large-size problems, the proposed LNS can find solution in few seconds while LINGO cannot find solution in an acceptable time (20 hours).

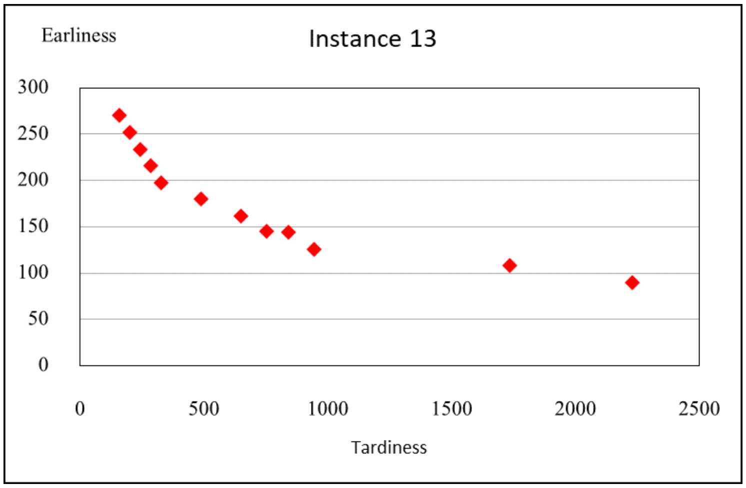

Figure 10. Non-dominated solutions for Instance 13

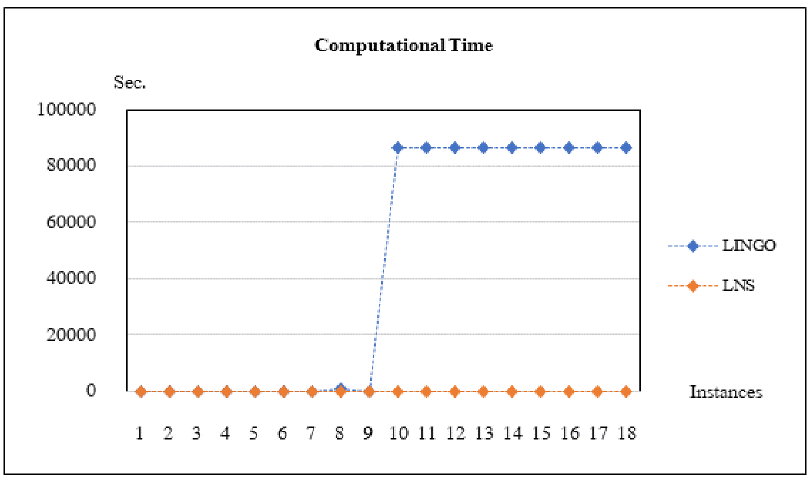

Figure 11. Comparison of computational time

5. Conclusion

This paper focuses on the multi-objective outbound truck scheduling problem in a multi-door cross-docking terminal. The key characteristic of the problem is that each truck can travel multiple trips to deliver products to different sets of customers. Thus, the assignment and sequencing of trucks needs to be made for each delivery. The objective of the problem is to find outbound truck schedule to minimize the total earliness and the total tardiness of all outbound trucks. The mixed integer programming (MIP) model for the model is presented. However, since the problem is NP-hard, the mathematical model can only find solution in small size problems. For this reason, this paper proposes Large Neighborhood Search (LNS) algorithm with different destroy and repair strategies to handle large size problems. In the proposed LNS, three destroy and two repair strategies are proposed to improve the solution quality. All combinations of destroy and repair strategies can be used to obtain new truck schedule. The numerical results show that the proposed LNS yields solutions closed to those obtained from an exact method in small-sized problems with fast computing time. For the larger-size problems, the proposed LNS can find solutions in relatively fast computing time while an exact method cannot find optimal solution in an acceptable time. Further research include improvement of the algorithm and incorporate other constraints in cross-docking operations to more reflect real-world practices.

Acknowledgements

This work was supported by Excellence Center in Logistics and Supply Chain Management, Chiang Mai University, Thailand.

References

Agustina, D., Lee, C .K .M., and Piplani, R., Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics*, vol. 152, pp .29-41, 2014.

Arabani, A .B., Ghomi, S .F., and Zandieh, M., Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert systems with Applications*, vol. 38, no. 3, pp. 1964-1979, 2011.

Alpan, G., Larbi, R., and Penz, B., A bounded dynamic programming approach to schedule operations in a cross docking platform .*Computers & Industrial Engineering*, vol. 60, no. 3, pp. 385-396, 2011.

Assadi, M .T., and Bagheri, M., Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems .*Computers & Industrial Engineering*, vol. 96, pp. 149-161, 2016.

Behnamian, J., Ghomi, F., Jolai, S.M.T., and Heidary, F. Location-allocation and scheduling of inbound and outbound trucks in multiple cross-dockings considering breakdown trucks. *Journal of Optimization in Industrial Engineering*, vol. 11, no. 1, pp. 51-65, 2018.

Keshtzari, M., Naderi, B., and Mehdizadeh, E. An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems .*Computers & Industrial Engineering*, vol. 91, pp.197-204, 2016.

Miao, Z., Lim, A., and Ma, H., Truck dock assignment problem with operational time constraint within cross docks .*European Journal of Operational Research*, vol. 192, no. 1, pp .105-115, 2009.

Pisinger, D. and Ropke, S.*Large neighborhood search*. In M. Gendreau & J.-Y. Potvin (Eds.), Handbook of Metaheuristics, 2nd Edition , 2010.

Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. *In CP-98 (Fourth International Conference on Principles and Practice of Constraint Programming)*, vol. 1520 of Lecture Notes in Computer Science, pages 417–431, 1998.

Soltani, R. and Sadjadi, S .J., Scheduling trucks in cross-docking systems :A robust meta-heuristics approach. *Transportation Research Part E :Logistics and Transportation Review*, vol. 46, no. 5, pp. 650-666, 2010.

Wisittipanich, W. and Hengmeechai, P., A Multi-Objective Differential Evolution for Just-In-Time Door Assignment and Truck Scheduling in Multi-Door Cross Docking Problems. *Industrial Engineering & Management Systems*, vol. 4, no. 3, pp. 299-311, 2015.

Wisittipanich, W. and Kamsura, N., (2017) A Mathematical Model for Outbound Truck Scheduling with Multiple Trips in Multi-Door Cross Docking System, *Proceedings of the International Conference on Industrial Engineering and Operations Management (8th IEOM)*, Bandung, Indonesia, March 6-8, 2018.

Biographies

Warisa Wisittipanich received the Doctor degree of Engineering from Asian Institute of Technology, Thailand in 2012. Since then, she has been working as an assistant professor at the Department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, Thailand. Her areas of interest and research are operations research, production scheduling and sequencing, inbound and outbound truck scheduling, vehicle routing problem, supply chain and logistics management, and metaheuristic application for optimization.

Nattapong Kamsura received his Bachelor degree of Science form the department of Modern Management and Information Technology, College of Art Media and Information Technology, Chiang Mai University in 2014 and a Master degree of Engineering from a department of Industrial Engineering, Faculty of Engineering, Chiang Mai University, Thailand in 2016. His research interests include operations research, truck scheduling, supply chain and logistics management, and computer science.