

A Truck Scheduling Problem for Multi-Crossdocking System with Metaheuristics

Phan Nguyen Ky Phuc¹, Nguyen Van Thanh^{2,*} and Duong Bao Tram¹

¹International University-Vietnam National University, Vietnam National University, HoChiMinh City, 70000, Vietnam

²Faculty of Commerce, Van Lang University, Ho Chi Minh City, 70000, Vietnam

*Corresponding Author: Nguyen Van Thanh. Email: thanh.nguyenvan@vlu.edu.vn

Received: 29 January 2022; Accepted: 23 March 2022

Abstract: The cross-docking is a very important subject in logistics and supply chain managements. According to the definition, cross-docking is a process dealing with transshipping inventory, in which goods and products are unloaded from an inbound truck and process through a flow-center to be directly loaded onto an outbound truck. Cross-docking is favored due to its advantages in reducing the material handing cost, the needs to store the product in warehouse, as well decreasing the labor cost by eliminating packaging, storing, pick-location and order picking. In cross-docking, products can be consolidated and transported as a full load, reducing overall distribution costs. In this paper, we focus on a truck scheduling at the multi-door, multi-crossdocking network with inventory constraints and process capability constraints. In this model, a truck can visit several docks for loading or unloading many types products. This situation is very common in reality. This study also developed an exact mathematical model using mixed-integer linear programming (MILP) with the objective of minimizing the makespan to obtain the benchmark in small scale problems. Large scale problems are solved through Simulated Annealing (SA) algorithm and Tabu Search (TS) algorithm. Performance of these algorithms will be compared to benchmarks obtained from solver as well as to each other.

Keywords: Truck scheduling; multi-door; multi-crossdocking network; simulated annealing; tabu search

1 Introduction

As the global markets on supply chain has seen an influx of competitors during the past few years, it is pertinent that manufactures, retailers and distributors strive to optimize costs to increase their competitiveness. Driven by such demand, the idea of cross docking was hailed. It was defined by [1] that cross-docking is a process dealing with transshipping inventory, in which goods and products are unloaded from an inbound truck and process through a flow-center to be directly loaded onto an outbound truck. The impact of cross-docking was proved to be highly beneficial in reducing warehousing cost, which takes up around 30% of the product sales cost, when Walmart first



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

pioneering its implementation. In 1992, by putting the cross-docking model into effects across 82% of its inventory, Walmart became the most profitable retailer globally, successfully reduced its cost 2%–3% [2]. One of the earliest works addressing short-term scheduling in cross-dock belongs to [3] whose work is renowned for considering 32 models. The general aim was to generate a sequence of receiving inbound and outbound truck at each door to minimize the makespan. Later on, [4] applied and concluded that the TS metaheuristics could effectively solve a cross-docking problem. The literatures regarding truck scheduling are quite well documented over the year. Williams in [5] was the first who contributed his work to this topic's literature. The problem, with the objective of minimizing the makespan, was solved using genetic algorithm in combination with simulation model. Another study was of [6], which focused on generating sequences of trucks entering the door, and solutions are developed based on different assumptions to the problem. It was concluded that heuristic algorithm performed better or as well as dynamic programming with or without prior assumption of known truck sequencing. Reference [7] tackled a simplified cross-dock model to gain the underlying complexity of truck sequencing problem, which was split into sub problems of inbound and outbound sequencing separately.

Reference [8] investigated the truck scheduling problem with constraint of time window and deadline for truck departure. Though the paper successfully touched on real-world constraint, the model was simplified to only cross dock with single inbound/outbound door, which may not be applicable to a real situation [8]. The methodology was developed using a hybrid metaheuristic between SA and Keshtel algorithm. Reference [9] also tackled the truck scheduling problem with time window constraints but expanded the problem to multi-door cross docking system. The objective was minimizing tardiness of outbound truck and proposed TS and SA for generating the solution. For multi-door cross-dock, a few studies have taken the approach similar to that of flow shop with parallel machines. Reference [10] is one of the first research which followed such route. Similarly, [11] also applied the idea to their cases with the addition of time-indexed variables. The problem was approached by using constructive polynomial-time algorithm and more traditional scheduling algorithm like Johnson's rule-based algorithm [12]. The topic also consists of works from works from [11–14]. The work of [14] was constructed predicated on the work of [10] on the two-stage hybrid cross-docking scheduling. The new work appends that of [10] as the authors used time-indexed model as opposed to the original completion time and precedence model. The study went on to develop the solution using compressed differential heuristic and compared the result coined from both models [14]. The problem continued to be expanded to multi cross-docking system, or cross-dock network. Reference [15] first contributed to this topic with a multi cross-dock model to minimize the operational cost. Aside from the trucking scheduling constraint, inventory balance was calculated to deduce the holding cost and inventory level against the capacity. The solution was generated by TS and SA metaheuristics, which were compared against the simple greedy algorithm [16]. Reference [16] presented their work on multi-cross dock which intimately adhered to the previously proposed notations by [3], therefore shared similarity to that of [13]. The work approached the problem of truck scheduling by using sequencing variable. The limitation, however, is the failure to regard the capacity and the increased complexity from the approach. The problem was solved using firefly and SA metaheuristics. The most recent work was that of [17], in which the problem of truck scheduling to find the minimum makespan was done for a multi-serviced/purposed crossdocking network. The problem was solely approached by devising a MILP model [17]. Other extension of cross-docking and its application can be shown in the works [18], where the authors applied particle swarm optimization (ω -PSO) to minimize the makespan. A cosine decreasing strategy of inertia weight was applied in this study to balance between exploit and explore. Furthermore, crossover strategy is presented to prevent the algorithm from falling

into local optimum. The integrated model of routing inbound vehicles between suppliers and cross-docks and outbound vehicles between cross-docks and retailers was considered in [19]. Different to our work, this study aimed to minimize the total cost by optimizing assignment of products to suppliers and retailers instead of scheduling trucks consideration. The problem was solved through endosymbiotic evolutionary algorithm. Multi-objectives for cross-docking problem was studied by [20]. In this study, the authors investigated truck scheduling in a rail–road physical internet cross-docking hub considering energy consumption. The two main objectives were to minimize the energy consumption and cost of outbound truck. The problem was different to this study since main variables were only whether a truck should be assigned to a dock. The multiple-dock visit was not allowed and dock capacity was also not considered in this study. Other extension and related works of cross-dock problems were presented in [21,22]. Though cross-docking problem has been rigorously explored since the 90s, scheduling problems do not take much proportion in the literary vault. For problem regarding multi-cross dock alone, there have only been two papers publicly released, to the best of our knowledge. The aim of this paper is to devise a mathematical model and well as suitable approaches for solution development of large-sized problems to obtain the aim and satisfy all requirement from the company. The model should reflect on the real condition to a certain extent to acquire a level of applicability, which can serve as a foundation for future development. The scope of the problem will fall within the spectrum of crossdocking operational planning through scheduling. However, it will only concern with the exterior operations involving coordinating the trucks, the unloading and loading. This study has some resemblance to the work of [17], however; it differentiates itself with other researches by expanding the problem to multi-door, multi-crossdocking network. Furthermore, the model also allows multi visiting of shipping and receiving trucks to other docks. At each dock, trucks will load or unload some kinds of products which are specified by the dock. Load or unload splitting are also permitted in this model. Dock capacity is also considered here so that the model is very similar to real-practice case. The rest of the paper is organized as follows. Section 2 elaborates on the problem description and crossdocking system, with the corresponding mathematical model, followed by the presenting of algorithms to solve large-sized problems in Section 3. Result analysis is presented in Section 4 and lastly, section 5 recapitulates the paper in discussion and conclusion.

2 Mathematical Model

In this paper, the study will focus on the cross-docking process which involve separate, multiple docks that have the capability to handle different types of products. All cross docks allow temporary storages, but at the end of the day, the inventory in all cross docks has to be zero. In addition, the layout is symmetrical, meaning there are an equal number of inbound and outbound doors for each dock. We also assume that the inbound doors and outbound doors are separate, meaning each set has single purpose. At all times, each door can only process one truck and preemption is not allowed. Furthermore, the number of loaded products has to be equal or larger than the demand.

In our model, the super scripts R and S represent for variables relating to process of receiving and shipping, respectively.

Indices $\begin{matrix} \{L\} \\ \{SEP\} \end{matrix}$

r index of receiving trucks, $r \in R$

s index of shipping trucks, $s \in S$

d index of docks, $d \in D$ $\begin{matrix} \{L\} \\ \{SEP\} \end{matrix}$

p index of product types, $p \in P$

t index of time periods, $t \in T$

Parameters

ω_{rp}^R In-transit inventory of unit product p on the receiving truck r

ω_{sp}^S Number of unit product p demanded by the shipping truck s

τ_r^R Soonest time receiving truck r enter any dock d

τ_s^S Soonest time shipping truck s enter any dock d

ρ_d Capacity of dock

η_d Number of inbound/outbound doors at dock d

α Time for handling an unit item

β_{dp} If dock d can handle product p , $\beta_{dp} = 1$, otherwise; $\beta_{dp} = 0$

γ Transition time of the truck between the docks

$BigM$ A very large number

Variables

C_{max} The makespan

E_{rd}^R Entering time of receiving truck r at dock d

L_{rd}^R Leaving time of receiving truck r at dock d

Q_{rdp}^R Quantity of product p receiving truck r unloaded at dock d

X_{rdh}^R Binary variable, $X_{rdh}^R = 1$ if receiving truck r enters dock d before dock h ; otherwise $X_{rdh}^R = 0$

Y_{rdp}^R Binary variable, $Y_{rdp}^R = 1$ if item p is unloaded by receiving truck r at dock d

Z_{rd}^R Binary variable, $Z_{rd}^R = 1$ if receiving truck r enter dock d ; otherwise, $Z_{rd}^R = 0$

U_{rdt}^R Binary variable, $U_{rdt}^R = 1$ if $t \geq E_{rd}^R$; otherwise $U_{rdt}^R = 0$

V_{rdt}^R Binary variable, $V_{rdt}^R = 1$ if $t \leq L_{rd}^R$; otherwise $V_{rdt}^R = 0$

R_{rdt}^R Binary variable, $R_{rdt}^R = 1$ if $E_{rd}^R \leq R_{rdt}^R \leq L_{rd}^R$; otherwise $R_{rdt}^R = 0$

N_{trdp}^R Number of product p being unloaded by receiving truck r at dock d up to time t

E_{sd}^S Entering time of shipping truck s at dock d

L_{sd}^S Leaving time of shipping truck s at dock d

Q_{sdp}^S Quantity of product p shipping truck s loaded at dock d

X_{sdh}^S Binary variable, $X_{sdh}^S = 1$ if shipping truck s enters dock d before dock h ; otherwise $X_{sdh}^S = 0$

Y_{sdp}^S Binary variable, $Y_{sdp}^S = 1$ if item p is loaded by shipping truck s at dock d

Z_{sd}^S Binary variable, $Z_{sd}^S = 1$ if shipping truck s enter dock d ; otherwise, $Z_{sd}^S = 0$

U_{sdt}^S Binary variable, $U_{sdt}^S = 1$ if $t \geq E_{sd}^S$; otherwise $U_{sdt}^S = 0$

V_{sdt}^S Binary variable, $V_{sdt}^S = 1$ if $t \leq L_{sd}^S$; otherwise $V_{sdt}^S = 0$

R_{sdt}^S Binary variable, $R_{sdt}^S = 1$ if $E_{sd}^S \leq R_{sdt}^S \leq L_{sd}^S$; otherwise $R_{sdt}^S = 0$

N_{tsdp}^S Number of product p being loaded by shipping truck s at dock d up to time t

I_{dpt} Inventory of product p at dock d at time t

$\min C_{max}$

(1)

Subject to

$$C_{max} \geq L_{sd}^S, \forall s, d \tag{2}$$

The Eq. (2) claims that the makespan must be greater or equal to the leaving dock time of all shipping trucks

$$Y_{rdp}^R \leq \beta_{dp}, \forall r, d, p \tag{3}$$

$$Y_{rdp}^R \leq Z_{rd}, \forall r, d, p \tag{4}$$

$$Y_{rdp}^R \geq \beta_{dp} + Z_{rd} - 1, \forall r, d, p \tag{5}$$

$$Q_{rdp}^R \leq BigM \times Y_{rdp}^R, \forall r, d, p \tag{6}$$

The Eqs. (3)–(6) ensure that a receiving truck can only visit a dock if it is allowed and its unloaded quantity to this dock is zero in case of no visitation

$$\tau_r^R \leq E_{rd}^R + BigM(1 - Z_{rd}^R), \forall r, d \tag{7}$$

This imposed the time window constraints on receiving trucks. The receiving truck cannot enter the dock before its allowed soonest enter time in Eq. (7).

$$L_{rd}^R \geq E_{rd}^R + \alpha \sum_{p \in P} Q_{rdp}^R - BigM(1 - Z_{rd}^R), \forall r, d \tag{8}$$

$$L_{rd}^R \leq E_{rd}^R + \alpha \sum_{p \in P} Q_{rdp}^R + BigM(1 - Z_{rd}^R), \forall r, d \tag{9}$$

$$X_{rdh}^R \leq Z_{rd}^R, \forall r, d \tag{10}$$

$$X_{rdh}^R + X_{rhd}^R \leq 1, \forall r, d, h \tag{11}$$

$$E_{rh}^R \geq L_{rd}^R + \gamma - BigM(1 - X_{rdh}^R), \forall r, d, h \tag{12}$$

The Eqs. (10)–(12) state that if a receiving truck visits dock d then dock h , the associating variable Z_{rd}^R will be one. Furthermore, the enter time at dock h must be greater or equal to the leaving time of dock d plus traveling time between two docks.

$$\sum_{d \in D} Q_{rdp}^R = \omega_{rp}^R, \forall r, p \tag{13}$$

The Eq. (13) says that the total unloaded quantity at all docks must be equal to the quantity the receiving truck carrying

$$t \geq E_{rd}^R - BigM(1 - U_{rdt}^R), \forall r, d, t \tag{14}$$

$$t \leq E_{rd}^R - 1 + BigM \times U_{rdt}^R, \forall r, d, t \tag{15}$$

The Eqs. (14) and (15) ensure that if t is greater than the entering time dock d of receiving truck r , $U_{rdt}^R = 1$, otherwise; $U_{rdt}^R = 0$

$$t \leq L_{rd}^R + BigM(1 - V_{rdt}^R), \forall r, d, t \tag{16}$$

$$t \geq L_{rd}^R + 1 - BigM \times V_{rdt}^R, \forall r, d, t \tag{17}$$

The Eqs. (16) and (17) guarantee that if t is smaller than the leaving time dock d of receiving truck r , $V_{rdt}^R = 1$, otherwise; $V_{rdt}^R = 0$

$$R_{rdt}^R \leq U_{rdt}^R, \forall r, d, t \quad (18)$$

$$R_{rdt}^R \leq V_{rdt}^R, \forall r, d, t \quad (19)$$

$$R_{rdt}^R \leq Z_{rdt}^R, \forall r, d, t \quad (20)$$

$$R_{rdt}^R \geq U_{rdt}^R + V_{rdt}^R + Z_{rdt}^R - 2, \forall r, d, t \quad (21)$$

The Eqs. (18)–(21) forces the constraint that if t is in the range of entering time and leaving time dock d and the receiving truck r also visits the dock $R_{rdt}^R = 1$, otherwise; $R_{rdt}^R = 0$

$$\sum_{d \in D} R_{rdt}^R \leq 1, \forall r, t \quad (22)$$

$$\sum_{r \in R} R_{rdt}^R \leq \eta_d, \forall d, t \quad (23)$$

The Eqs. (22) and (23) show that and at any time a receiving truck can only be served by one dock and total number of receiving trucks is served by a dock d cannot be greater than the number of its door.

$$Y_{sdp}^S \leq \beta_{dp}, \forall s, d, p \quad (24)$$

$$Y_{sdp}^S \leq Z_{sd}^S, \forall s, d, p \quad (25)$$

$$Y_{sdp}^S \geq \beta_{dp} + Z_{sd}^S - 1, \forall s, d, p \quad (26)$$

$$Q_{sdp}^S \leq \text{BigM} \times Y_{sdp}^S, \forall s, d, p \quad (27)$$

$$\tau_s^S \leq E_{sd}^S + \text{BigM} \times (1 - Z_{sd}^S), \forall s, d \quad (28)$$

$$L_{sd}^S \geq E_{sd}^S + \alpha \sum_{p \in P} Q_{sdp}^S - \text{BigM}(1 - Z_{sd}^S), \forall s, d \quad (29)$$

$$L_{sd}^S \leq E_{sd}^S + \alpha \sum_{p \in P} Q_{sdp}^S + \text{BigM}(1 - Z_{sd}^S), \forall s, d \quad (30)$$

$$X_{sdh}^S \leq Z_{sd}^S, \forall s, d \quad (31)$$

$$X_{sdh}^R + X_{shd}^R \leq 1, \forall s, d, h \quad (32)$$

$$E_{sh}^S \geq L_{sd}^S + \gamma - \text{BigM}(1 - X_{sdh}^S), \forall s, d, h \quad (33)$$

$$\sum_{d \in D} Q_{sdp}^S = \omega_{sp}^R, \forall s, p \quad (34)$$

$$t \geq E_{sd}^S - \text{BigM}(1 - U_{sdt}^S), \forall s, d, t \quad (35)$$

$$t \leq E_{sd}^S - 1 + \text{BigM} \times U_{sdt}^S, \forall s, d, t \quad (36)$$

$$t \leq L_{sd}^S + \text{BigM}(1 - V_{sdt}^S), \forall s, d, t \quad (37)$$

$$t \geq L_{sd}^S + 1 - \text{BigM} \times V_{sdt}^S, \forall s, d, t \quad (38)$$

$$R_{sdt}^S \leq U_{sdt}^S, \forall s, d, t \quad (39)$$

$$R_{sdt}^S \leq V_{sdt}^S, \forall s, d, t \quad (40)$$

$$R_{sdt}^S \leq Z_{sdt}^S, \forall s, d, t \quad (41)$$

$$R_{sdt}^S \geq U_{sdt}^S + V_{sdt}^S + Z_{sdt}^S - 2, \forall s, d, t \quad (42)$$

$$\sum_{d \in D} R_{sdt}^S \leq 1, \forall r, t \quad (43)$$

$$\sum_{s \in S} R_{sdt}^S \leq \eta_d, \forall d, t \quad (44)$$

We also apply the same physical constraints for the shipping trucks, which creates Eqs. (24)–(44).

$$N_{trdp}^R \leq \text{BigM} \times (1 - V_{rdt}^R), \forall r, d, t \quad (45)$$

$$N_{trdp}^R \leq Q_{rdp}^R + \text{BigM} \times V_{rdt}^R, \forall r, d, t \quad (46)$$

$$N_{trdp}^R \geq Q_{rdp}^R - \text{BigM} \times V_{rdt}^R, \forall r, d, t \quad (47)$$

The total unload at a dock is assumed to be affected right after the receiving truck r enter the dock. If $V_{rdt}^R = 0$, $N_{trdp}^R = Q_{rdp}^R$ otherwise $N_{trdp}^R = 0$. So it can be interpreted as if t is higher than leaving time, i.e., $V_{rdt}^R = 0$, the total unload quantity of receiving truck r up to t is equal to its total unload quantity. If t is smaller than leaving time, the total unload quantity at dock is zero. These constraints are shown in Eqs. (45)–(47)

$$N_{tsdp}^S \leq \text{BigM} \times U_{sdt}^S, \forall r, d, t \quad (48)$$

$$N_{tsdp}^S \leq Q_{sdp}^S + \text{BigM} \times (1 - U_{sdt}^S), \forall t, s, d, p \quad (49)$$

$$N_{tsdp}^S \geq Q_{sdp}^S - \text{BigM} \times (1 - U_{sdt}^S), \forall t, s, d, p \quad (50)$$

The same idea is applied for constructing the constraints of shipping trucks. However; the total load to a shipping truck up to time t is calculated immediately after the shipping trucks enter the docks, $U_{sdt}^S = 1$, i.e., t is higher than entering time of shipping truck as shown in Eqs. (48)–(50)

$$I_{dpt} = \sum_{r \in R} N_{trdp}^R - \sum_{s \in S} N_{tsdp}^S, \forall d, p, t \quad (51)$$

$$\sum_{p \in P} I_{dpt} \leq \rho_d, \forall t, d \quad (52)$$

By forcing the inventory level is always greater or equal to zero and smaller than capacity through Eqs. (51) and (52), we ensure that the receiving trucks only visits the dock when the dock has enough product for satisfying their load demands.

3 Solution Approach

To solve the small-scale problems, CPLEX Optimizer engine which is developed by IBM company was used to create the benchmark. However, due to the NP-hard property of the original problems,

when the size increases metaheuristic algorithms must be adopted. In this study, TS and SA are also implemented and results obtained from CPLEX are used as benchmarks. During TA and SA, this study applies two common following algorithms for creating initial solutions and assignment process. The Fig. 1 shows how the initial solutions are created while Fig. 2 explains about the assignment process.

Algorithm 1: Algorithm for creating initial solution

```

Input:
  R: set of receiving truck, S: set of shipping truck
  D: set of docks, P: set of products
  Refin = [ ]: finishing list of receiving trucks
  Shipfin = [ ]: finishing list of shipping trucks

Initialize:
  Calculate gap between the unloading target amount and available
  unloading capacity
  Select trucks r greedily with smallest positive gap and earliest
  available time
  Assign trucks r to docks sequentially
  Refin.push(r)

while Shipfin ≠ S do
  Find CanShip list that can be satisfied by current inventory
  if CanShip = [ ] then
    Calculate gap between the unloading target amount and
    available unloading capacity
    Select truck r greedily with smallest positive gap and earliest
    available time
    Assign truck r to docks with earliest possible assigning time as
    priority
    Update inventory and docks' occupation vectors
  else
    Calculate gap between the loading target amount and available
    inventory
    Select truck s greedily with smallest positive gap and earliest
    available time
    Assign truck s to docks with earliest possible assigning time as
    priority
    Update inventory and docks' occupation vectors
  Cmax = max (departure time of all trucks s)
return Cmax

```

Figure 1: Algorithm for creating initial solution

Algorithm 2: Algorithm for assignment

Input:

re_seq: input receiving truck sequence
ship_seq: input shipping truck sequence
dock_seq: input dock sequence
ShipList = []: list of unassigned shipping truck satisfied by the current inventory

Initialize:

Step 1: Assign the first receiving truck to the according input dock sequence calculate the properties of the trucks and docks, update dock inventory and occupation status, remove the truck from the sequence.

Step 2: Check the inventory level. If any shipping truck is satisfied, move to step 3. If not, move to step 4

Step 3: Pick the first shipping truck in the sequence. Check if its demand is satisfied by the inventory. If so assign the truck to the docks with earliest possible assigning time as priority and remove the truck out of the sequence. If not, push the truck to the end of the sequence and pick the next truck in line.

Step 4: Pick the next receiving truck in line. Check if the inventory of the docks is enough to store the required unload product. If so, assign the truck with earliest possible assigning time as priority and remove the truck out of the sequence. If not, push the truck to the end of sequence and pick the next in line truck.

Step 5: Check if there is any shipping truck left in the sequence. If yes, go back to step 2. If no, end the program.

Step 6: Calculate C_{max} as maximum departure time of all shipping trucks.

return *The feasible solution*

Figure 2: Algorithm for assignment

TS and SA are chosen is due to its simplicity in the process of creating new solution in the process of exploring and exploiting, and the foundation of such process is the neighborhood search method. In this paper, the neighborhood search is implemented through two swapping methods in Fig. 3. The object of swapping is the sequence of receiving trucks, the sequence of shipping trucks and the sequence of docks for the first truck. These are also the input into the evaluation function to calculate the corresponding makespan.

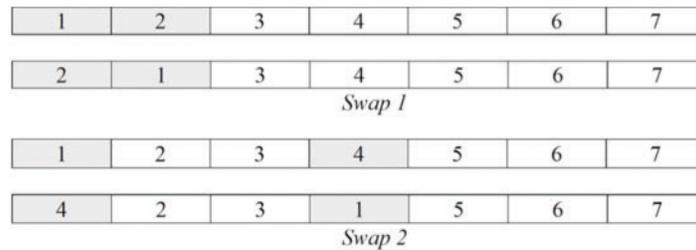


Figure 3: Swapping methods

3.1 Tabu Search

The pseudo code of TS is described in Fig. 4.

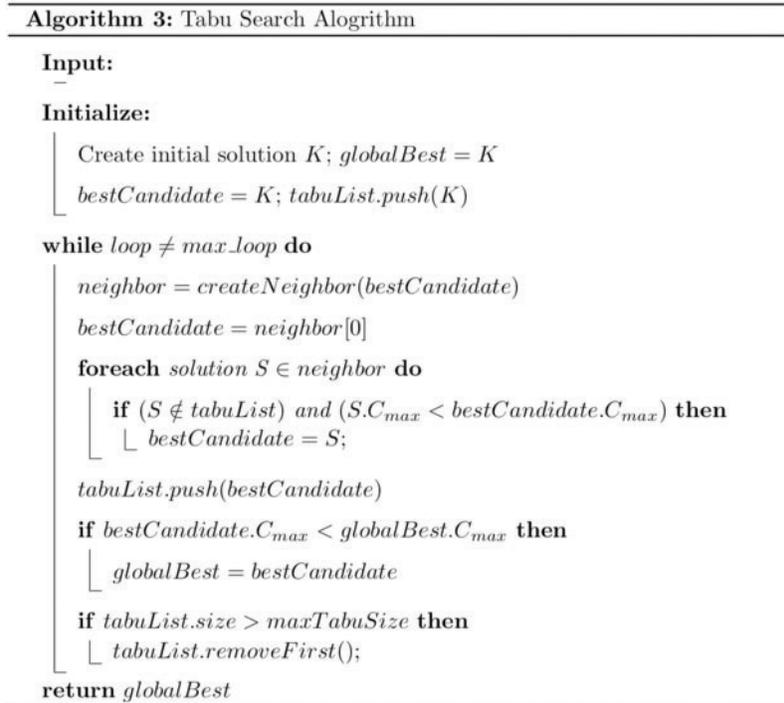


Figure 4: Pseudo code of Tabu Search algorithm

3.2 Simulated Annealing

For simulated annealing algorithm, we verify its performance with two versions. The first one employs the sigmoid function which is presented in Fig. 5. The second one applies the metropolis function which is described in Fig. 6.

Algorithm 4: SA Sigmoid Function

Input:
 _

Initialize:
 Create initial solution K ; $current = K$; β ; $globalBest = K$;
 $neighbor = \emptyset$; $run = 1$
 β the coefficient of temperature decrease, $\beta \in [0, 1]$

while $T > minT$ and $run \leq numrun$ **do**
 $neighbor = createNeighbor(current)$
if $neighbor.C_{max} < globalBest.C_{max}$ **then**
 $globalBest = neighbor$
else
 $\delta = \alpha(neighbor.C_{max} - current.C_{max})$
 $P = \frac{1}{1+e^{-\delta/T}}$
if chances to move $< P$ **then**
 $current = neighbor$
 $run = run + 1$
 $T = \beta \times T$
return $globalBest$

Figure 5: SA Sigmoid function

Algorithm 5: SA Metropolis Function

Input:
 _

Initialize:
 Create initial solution K ; $current = K$
 $globalBest = K$; $neighbor = \emptyset$; $run = 1$
 β the coefficient of temperature decrease, $\beta \in [0, 1]$

while $T > minT$ and $run \leq numrun$ **do**
 $neighbor = createNeighbor(current)$
if $neighbor.C_{max} < current.C_{max}$ **then**
 $current = neighbor$
if $current.C_{max} < globalBest.C_{max}$ **then**
 $globalBest = current$
 $run = run + 1$
else
 $\delta = \alpha(neighbor.C_{max} - current.C_{max})$
 $P = \frac{1}{1+e^{-\delta/T}}$
if chances to move $< P$ **then**
 $current = neighbor$
 $run = run + 1$
 $T = \beta \times T$,
return $globalBest$

Figure 6: SA Metropolis function

4 Result Analysis

To conduct result analysis, ten data sets with different scales are considered. The data set information and the results obtained from CPLEX are given in [Tab. 1](#).

Table 1: Data sets' information and results from CPLEX

Problem	Receiving truck	Shipping truck	Product	Total quantity	Cmax	Runtime (second)
1	3	4	4	140	112	20
2	3	4	5	64	36	37
3	3	4	8	180	110	57
4	3	4	9	188	110	81
5	3	4	10	444	277	217
6	5	4	6	1030	476	9478
7	6	4	8	491	221	9193
8	5	7	10	430	208	10812
9	6	7	10	2020	Unsolved	
10	6	11	15	1252	Unsolved	

For small-scaled problems, CPLEX works quite well in terms of run time, which only takes less than 2 minutes to solve. When there is increase in the number of trucks and product quantity, the run time grows exponentially as can be seen from the data set 6 to 10.

The comparison between results for both TA and SA and CPLEX are shown in the [Tab. 2](#).

Table 2: The results comparison between CPLEX and metaheuristic algorithms

Problem	TA	SA (Sigmoid)	SA (Metropolis)
1	0.00%	0.00%	0.00%
2	0.00%	0.00%	0.00%
3	0.00%	0.00%	0.00%
4	0.00%	0.00%	0.00%
5	7.94%	7.94%	7.94%
6	9.03%	10.08%	9.03%
7	9.50%	13.57%	9.50%
8	0.48%	1.92%	0.48%

In the [Tab. 2](#) the percentage is calculated as [Eq. \(53\)](#)

$$P = \frac{\text{makespan of algorithm} - \text{makespan of benchmark}}{\text{makespan of the benchmark}} \quad (53)$$

On an overall viewpoint, SA Metropolis algorithm yields most promising results when comparing with 2 other methods in gap.

5 Conclusions

In conclusion, to solve the problem of truck scheduling in crossdocking network, 3 approaches are taken. The first is using MILP in conjunction with CPLEX to solve for the exact solution. However, because of its restriction to small-sized problems, TS and SA are implemented to search for the makespan of large-sized problems. The two metaheuristics exhibit the tradeoff between producing a consistent and good result and having short run time. In general, the results from the approaches proved to be not only optimal and feasible to the constraints of the system, but also managed to adhere and comply to several practical conditions. The result also proves the credibility and feasibility of the model as well as the algorithm. Regarding the all-encompassing and real-life adherent nature of the proposed model, not only does it make a solid contribution to the topic's literature but also serve as a foundation for further development of the program into software. Further study on this topic can be expanded to include the interior operations of the crossdocking network. Another direction is to expand the problem downstream by combining the truck scheduling problem with the vehicle routing problem to the customers. Although the algorithm obtained reliable results, this study still encountered some challenges in handling the most difficult constraint in the crossdocking problem, the concurrency of load and unload. This concurrency creates challenges in ensuring the feasibility of system state as well as the solutions deriving from the neighborhood. The feasibility is only assured through very carefully checked and revised mechanism. This process sometime takes long time for specific cases.

Funding Statement: The authors wish to express their gratitude to International University-Vietnam National University, Van Lang University, Vietnam for financial support for this research.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] A. L. Ladier and G. Alpan, "Cross-docking operations: Current research versus industry practice," *Omega*, vol. 62, no. 1, pp. 145–162, 2016.
- [2] G. Stalk, P. Evans and L. Shulman, "Competing on capabilities: The new rules of corporate strategy," *Harvard Business Review*, vol. 70, no. 2, pp. 57–69, 1992.
- [3] W. Yu, "Operational strategies for cross docking systems," Ph.D. dissertation, Iowa State University, 2002.
- [4] W. Yu and P. J. Egbelu, "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage," *European Journal of Operational Research*, vol. 184, no. 1, pp. 377–396, 2008.
- [5] D. L. McWilliams, P. M. Stanfield and C. D. Geiger, "The parcel hub scheduling problem: A simulation-based solution approach," *Computers & Industrial Engineering*, vol. 49, no. 3, pp. 393–412, 2005.
- [6] M. Y. Maknoon and P. Baptiste, "Cross-docking: Scheduling of incoming and outgoing semi trailers," *International Journal of Logistics Research and Applications*, vol. 12, no. 1, pp. 249–261, 2009.
- [7] N. Boysen, M. Fliedner and A. Scholl, "Assembly line balancing: Which model to use when," *International Journal of Production Economics*, vol. 111, no. 2, pp. 509–528, 2008.
- [8] A. Golshahi-Roudbaneh, M. Hajiaghahi-Keshteli and M. M. Paydar, "Cross-dock scheduling considering time windows and deadline for truck departures," *Scientia Iranica*, vol. 28, no. 1, pp. 532–546, 2021.
- [9] G. Ozden and I. Saricicek, "Scheduling trucks in a multi-door cross-docking system with time windows," *Bulletin of the Polish Academy of Sciences-Technical Sciences*, vol. 67, no. 1, pp. 349–362, 2019.
- [10] F. Chen and K. Song, "Minimizing makespan in two-stage hybrid cross docking scheduling problem," *Computers & Operations Research*, vol. 36, no. 6, pp. 2066–2073, 2009.

- [11] G. B. Fonseca, T. H. Nogueira and M. G. Ravetti, "A hybrid Lagrangian metaheuristic for the cross-docking flow shop scheduling problem," *European Journal of Operational Research*, vol. 275, no. 1, pp. 139–154, 2019.
- [12] T. H. Nogueira, F. B. Coutinho, R. P. Ribeiro and M. G. Ravetti, "Parallel-machine scheduling methodology for a multi-dock truck sequencing problem in a cross-docking center," *Computers & Industrial Engineering*, vol. 143, no. 1, pp. 1–20, 2020.
- [13] W. Wisittipanich and P. Hengmeechai, "Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization," *Computers & Industrial Engineering*, vol. 113, no. 1, pp. 793–802, 2017.
- [14] P. M. Cota, B. M. R. Gimenez, D. P. M. Araujo, T. H. Nogueira, M. C. Souza *et al.*, "Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre," *Computers & Industrial Engineering*, vol. 95, no. 1, pp. 135–143, 2016.
- [15] P. Chen, Y. Guo, A. Lim and B. Rodrigues, "Multiple crossdocks with inventory and time windows," *Computer & Operations Research*, vol. 33, no. 1, pp. 43–63, 2006.
- [16] M. M. Isfahani, R. T. Moghaddam and B. Naderi, "Multiple cross-docks scheduling using two metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 74, no. 1, pp. 129–138, 2014.
- [17] G. C. Issi, R. Linfati and J. W. Escobar, "Mathematical optimization model for truck scheduling in a distribution center with a mixed service mode dock area," *Journal of Advanced Transportation*, vol. 2020, no. 1, pp. 1–13, 2020.
- [18] Y. Ye, J. Li, K. Li and H. Fu, "Cross-Docking truck scheduling with product unloading/loading constraints based on an improved particle swarm optimisation algorithm," *International Journal of Production Research*, vol. 56, no. 16, pp. 5365–5385, 2018.
- [19] K. Y. Lee, J. S. Lim and S. S. Ko, "Endosymbiotic evolutionary algorithm for an integrated model of the vehicle routing and truck scheduling problem with a cross-docking system," *Informatica*, vol. 30, no. 3, pp. 481–502, 2019.
- [20] T. Chargui, A. Bekrar, M. Reghioi and D. Trentesaux, "Multi-Objective sustainable truck scheduling in a rail-road physical internet cross-docking hub considering energy consumption," *Sustainability*, vol. 11, no. 11, pp. 1–23, 2019.
- [21] O. Theophilus, M. A. Dulebenets, J. Pasha, O. F. Abioye and M. Kavooosi, "Truck scheduling at cross-docking terminals: A follow-up state-of-the-art review," *Sustainability*, vol. 11, no. 19, pp. 1–23, 2019.
- [22] R. K. Mavi, M. Goh, N. K. Mavi, F. Jie, K. Brown *et al.*, "Cross-docking: A systematic literature review," *Sustainability*, vol. 11, no. 11, pp. 1–19, 2020.