Truck Scheduling Model in the Cross-docking Terminal by using Multi-agent System and Shortest Remaining Time Algorithm

Purba Daru Kusuma
Computer Engineering, Faculty of Electrical Engineering
Telkom University, Bandung, Indonesia

***Abstract*—One most important and critical problem in a cross-docking system is truck scheduling. Many studies in it assumed that the temporary storage is unlimited which is in the real world, the temporary storage is limited. Many studies focus on minimizing total completion time. Meanwhile, studies that focus on minimizing temporary storage are hard to find, although this aspect is very important. Due to its complexity, especially in the cross-docking system with multiproduct characteristics, manual scheduling is almost impossible to achieve its goals. Many studies used several techniques, such as genetic algorithm (GA) and mixed integer programming where these methods are computationally expensive. Based on this problem, in this work, we propose new truck scheduling model in a cross-docking terminal with limited temporary storage constraint. This model is developed by using multi-agent system. The main contribution of this work is proposing the multi-agent-based truck scheduling model with limited temporary storage capacity constraint and temporary truck changeover permit. In it, there are three agents: inbound-trucks scheduler agent, outbound-trucks scheduler agent, and material handler agent. The shortest remaining time (SRT) algorithm is adopted in every agent. Based on the simulation result, this proposed model is proven competitive compared with the existing FIFO based models and integer-programming based model. Compared with the integer-programming model, it creates 41.8 percent lower in maximum inventory level. Compared with the FIFO based model, it creates 52.1 to 55.1 percent lower in maximum inventory level. In total time aspect, it creates 0.2 to 2.2 percent lower than the FIFO based model. It creates 7.2 percent higher in total time compared with the integer-programming based model.**

***Keywords*—Truck scheduling; cross-docking system; multi agent system; shortest remaining time; intelligent supply chain**

I. Introduction

Cross-docking system is one popular supply chain management (SCM) system which is implemented in a distribution center [1,2]. This system is different from the conventional supply chain management system. In the conventional system, the arriving products are stored first in a big warehouse or storage area [1]. Then, these products are sorted and delivered to the customers [1]. Meanwhile, in the cross-docking system, the arriving products from the inbound vehicles will be transferred immediately to the outbound vehicles. The temporary storage is used as a sorting and consolidation area, and it is usually small [2]. In the conventional system, products can stay in the warehouse in days, weeks, or months. In the other side, in the cross-docking system, products are in the terminal usually in less than 24 hours [2].

Based on its characteristics, the cross-docking system has several advantages. The first advantage is improving service level [1] by reducing total completion time [1] in the terminal and delivery time [3,4] so that the products will be delivered to the customers faster. The second advantage is the cross-docking system can reduce the cost of warehousing [3], such as inventory-holding cost [3,4], handling cost [3], transportation cost [3,4], labor cost [3], storage space cost [3], and order picking cost [4]. The third advantage is the cross-docking system may reduce or avoid product quality degradation, or risk of product damage or obsolescence [3], especially for the perishable products due to its less time in a warehouse. Because of its advantages, the cross-docking system has been adopted by some largest chained retailers, such as Walmart, Target, COSTCO, and Auchan [5]. This system has also been adopted by some big shipping companies, such as FedEx, UPS, USPS, and DHL [5].

One important problem in the cross-docking system is the truck scheduling problem. Because the fast-processing time is achieved by transferring the products from the inbound vehicles to the outbound vehicles as immediately as possible, the accuracy of the truck scheduling must be high [1]. This scheduling model means matching the inbound trucks and the outbound trucks which are docked at the doors at the same time [1]. In the cross-docking system, although it usually consists of some doors in both inbound and outbound sides, the number of doors is still less than the number of vehicles [1]. Because of its complexity, many studies used computational solution to solve this problem because this problem is almost impossible to be solved manually. Many studies used integer programming method [5-7]. Other studies also used metaheuristic method [8], such as genetic algorithm [1,3] due to its characteristics as a combinatorial solution. Meanwhile, some other studies implemented FIFO method [4,7] in one side (inbound or outbound).

There are several problems due to these existing studies in modeling the truck scheduling in the cross-docking system. First, many studies assumed that the temporary storage is unlimited [1,9,10]. This assumption was taken because in these previous models, every vehicle (usually truck) will be docked at the door until it completes its process. For the inbound truck, completion means that all carried products have been unloaded. Meanwhile, for the outbound truck, completion means that all requested products have been loaded. By reducing temporary changeover potential during loading and unloading process, the total completion time can be minimized. Unfortunately, in the real world, the temporary storage has limited capacity. Second, implementing metaheuristic or integer programming methods is a resource consuming process because of its computationally expensive characteristics.

Based on these problems, this work aims to propose truck scheduling model in the cross-docking system with the constraint is a limited temporary storage capacity. In this work, the inbound and outbound vehicles are trucks with same size or identical. Due to this constraint, in our work, temporary truck changeover is permitted. The objective of our proposed model is minimizing the total time and the inventory level.

This model is developed by using multi-agent system. This method is adopted due to the characteristics of the truck scheduling in the cross-docking system consists of three sub systems: inbound trucks scheduling, material handling, and outbound trucks scheduling. In every sub system, the shortest remaining time algorithm is adopted rather than metaheuristic method so that this process is computationally light.

This work has several novelties. These novelties are as follows.

- The new multi agent-based truck scheduling is proposed where the inbound scheduling, outbound scheduling, and material handling processes are conducted autonomously.
- The limited temporary storage capacity is applied in this model.
- The temporary truck changeover is allowed. It means that the trucks can be shifted temporarily during its loading or unloading process when other trucks are more available.

This paper is organized as follows. In the first section, the background, problem, research purpose, novelty, and paper organization are explained. In the second section, the latest previous works in truck scheduling are explored. In the third section, the proposed model is described. In the fourth section, the simulation, result, and findings are discussed. In the fifth section, the main result of this work related to the research purpose is concluded.

II. Related Works

Cross-docking system is a supply chain system that is popular in a distribution center [1,2]. In it, the incoming products are directly transferred from the incoming vehicles to the outcoming vehicles without storing them [2]. This mechanism is different from the conventional or traditional distribution center. In the conventional one, the incoming products are stored in the warehouse first. Then, when there is request or order for these products, they are packed and then sent to the customers. In this conventional way, the activities include receiving, storing, order picking, and shipping [2]. These stages occur due to the mismatch in supply and demand which usually happens in the conventional system. Different from it, in the cross-docking system, precise synchronization of the inbound and outbound vehicles plays critical role [2]. Besides, although it is almost impossible to be achieved, perfect synchronization between supply and demand is very important. That is why there are many studies in the cross-docking system that assumed that supply and demand are equal [11].

The performance of the cross-docking system can be divided into two aspects: design aspect and operational aspect [1]. The design aspects include location, terminal layout, number of docks, and temporary storage capacity [1]. Meanwhile, the operational aspects include sorting, consolidation, and truck scheduling [1]. The structure of the cross-docking terminal can be I-shape, X-shape, L-shape, or T-shape [10]. The illustration of the cross-docking terminal is shown in Fig. 1.

The scenario of the cross-docking system is usually as follows [1]. The cross-docking terminal consists of multiple receiving (inbound) docks and multiple shipping (outbound) docks. Inside the terminal, there are sorting and consolidating facilities, and temporary storage. The inbound trucks arrive and then are assigned to any available (empty) inbound dock. If there is not any available inbound dock, then, this truck should wait until at least one inbound dock is available. When a truck arrives in an inbound dock, then it unloads all the products it carries. After the unloading process completes, this truck then leaves the dock as soon as possible so that this dock can be used by other trucks. In the sorting and consolidating facilities, the unloaded products are split and merged with other products that are carried by other trucks. Then, they are loaded to the designated outbound trucks. After the loading process completes, this outbound truck also leaves the area as soon as possible so that this outbound dock can be used by other outbound trucks.

In the cross-docking system, temporary storage is needed to store the unloaded products that are not needed by the current docked outbound trucks but will be loaded to the future outbound trucks. The unloaded products usually stay in the temporary storage no more than 24 hours [1,2]. If they stay longer than 24 hours, then the system cannot be called as cross-docking system [2].

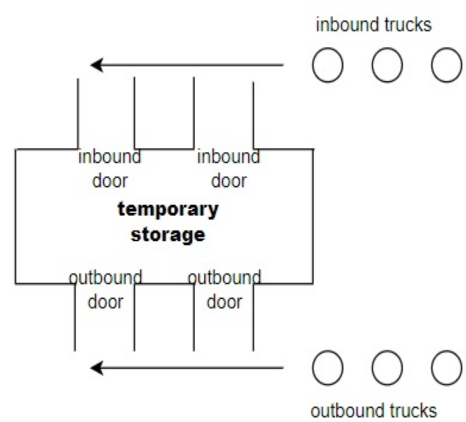

Fig. 1. Cross-docking Terminal.

There are many studies that are conducted in the truck scheduling in the cross-docking system. It is because truck scheduling plays important role to achieve the main goal of the cross-docking system. The objective of the most studies is minimizing the total time [8-10]. Besides, several other objectives are minimizing total cost [10,11], additional penalty [11] and delivery cost [7], flow time [12], processing time [12], tardiness of the outbound trucks [12], waiting time [10,13], completion time [1], inventory level [14], total product stay time [14], and total truck loading and unloading time [12]. Most of studies assumed that the temporary storage is unlimited [1,9,10] although it is impossible to be implemented in the real world. Several methods that were used are FIFO [4,7], integer programming [5-7], and genetic algorithm [1,3]. Most of studies conducted single inbound dock and single outbound dock [1,4]. Meanwhile, other studies conducted multiple inbound docks and multiple outbound docks.

Yu, Ha, and Park [1] used genetic algorithm in arranging the sequence of the inbound and outbound trucks. In their work, the goal is minimizing the maximum completion time. In it, the cross-docking terminal consists of multiple inbound docks and multiple outbound docks. The temporary storage was assumed unlimited. The simulation process consisted of 8 receiving docks, 8 shipping docks, 20 inbound trucks, 30 outbound trucks, 18 product types, 8,460 unit of products. The changeover time was assumed 75 time-units while the product moving times was assumed 100 time-units.

Issi, Linfati, and Escobar [2] used mixed-integer linear programming to solve this scheduling problem. This work aimed to minimize total time (makespan). This work used a multinational food company in Chile as a study case. In it, the products routing was excluded in the scheduling algorithm. Meanwhile, they proposed that warehouse tasks and vehicle routing problem should be included in the scheduling model for the future research potential.

Mohammadzadeh, Sahebjamnia, Fathollahi-Fard, and Hahiaghaei-Keshteli [8] focused on optimizing the total time in the truck scheduling in the cross-docking system. In this work, they used and compared three nature inspired metaheuristic models: red deer algorithm (RDA), virus colony search (VCS), and water wave optimization (WWO).

Lee, Lim, and Ko [11] used endosymbiotic evolutionary algorithm to solve the vehicle routing and truck scheduling problem in the cross-docking system. This work aimed to minimize the total cost. In it, the truck scheduling is integrated with the vehicle routing. This work implemented zero temporary storage. All vehicles are assumed identical. Tardiness or earliness is allowed with some penalty. The changeover time is fixed.

Fathollahi-Fard, Ranjbar-Bourani, Cheikhrouhou, and Hajiaghaei-Kesthteli [9] used social engineering optimizer algorithm to solve the truck scheduling problem in a cross-docking system. The cross-docking facility has I-shaped structure. The assumption was that all trucks are available at time zero. The changeover time is same for all vehicles. The temporary storage is unlimited.

Dulebenets [5] combined the diploid evolutionary algorithm with the mixed integer mathematical model to solve the truck scheduling in the cross-docking system. In this work, temporary storage was assumed unlimited. Rather than all trucks are available at time zero, in this work, each truck was expected arrives in certain time with zero arrival time.

Molavi, Shahmardan, and Sajadieh [7] used FIFO method in the loading mechanism in the truck scheduling in a cross-docking system. Meanwhile, the mixed-integer programming is used in the unloading mechanism. This work aimed to minimize the total cost due to the penalty and additional delivery cost because of the delayed shipment.

Larbi, Alpan, Baptiste, and Penz [4] used FIFO method in the truck scheduling in the cross-docking system. As FIFO is implemented strictly in the inbound side, the arrangement was conducted in the outbound side. This work aimed to minimize the total cost. This work implemented three scenarios. First, the system has complete and precise information about the order and the contents of the inbound trucks. Second, the system does not have the information of the incoming trucks but only knows the daily quantities of the products that must be shipped to every destination. Third, the inbound trucks sequence is known but the information of the content is only known after the inbound truck arrives to the receiving dock.

TABLE I. PREVIOUS WORKS SUMMARY

Authors	Objective Parameters	Method
Yu, Ha, and Park [1]	total completion time	genetic algorithm
Issi, Linfati, and Escobar [2]	total completion time	mixed integer programming
Mohammadzadeh, Sahebjamnia, Fathollahi-Fard, and Hahiaghaei-Keshteli [8]	total completion time	red deer algorithm. virus colony search algorithm
Lee, Lim, and Ko [11]	total completion time	evolutionary algorithm
Dulebenets [5]	total completion time	evolutionary algorithm
Molavi, Shahmardan, and Sajadieh [7]	total cost	first-in-first-out
Larbi, Alpan, Baptiste, and Penz [4]	total cost	first-in-first-out
Fathollahi-Fard, Ranjbar-Bourani, Cheikhrouhou, and Hajiaghaei-Keshteli [9]	total completion time	social engineering optimizer
Chargui, Bekrar, Reghioui, and Trentesaux [6]	energy consumption, total cost	simulated annealing, tabu search
Khorasani, Keshtzari, Islam, and Feizi [15]	delivery lead time	mixed integer linear programming
Ye, Li, Li, and Fu [16]	total completion time	particle swarm optimization

The summary of these previous works is shown in Table I. Based on this summary, it is shown that most of the studies in the truck scheduling in the cross-docking system focused on minimizing the total completion time. The other objective is minimizing total cost. Unfortunately, Research which their objective is maintaining the inventory level in the temporary storage is not popular although the temporary storage becomes important part in the cross-docking system. Based on it, this work, which focuses on minimizing the inventory level while maintaining low total time becomes very relevant.

III. Proposed Model

In this work, we propose a truck scheduling model that is implemented for the cross-docking terminal based on multi agent system. Its objective is to minimize the inventory level and the total completion time. It is developed by combining the multi-agent system and shortest remaining time algorithm. This cross-docking system implements single-inbound-single-outbound. Multiple products are handled in this system. There is temporary storage capacity constraint.

The detailed assumptions in the proposed model are as follows. The terminal has one inbound door for receiving and one outbound door for shipping [4]. The temporary storage has limited capacity [13]. This is a multiproduct logistic system so that it handles multiple products with various quantity [1]. The inbound trucks carry products from suppliers. The outbound trucks carry products to be delivered to customers. Each truck, either inbound or outbound, carries several products. All trucks have same capacity [11]. The inbound trucks carry less products, but the quantity of each product is higher. The outbound trucks carry more products, but the quantity of each product is less. As a just-in-time model, the total amount of the received products is equal to the shipped products in both product variation and quantity [11]. It is guaranteed that the temporary storage is empty in the end of scheduling process. In the beginning of the scheduling process, all inbound trucks and outbound trucks has been arriving in the cross-docking terminal [9].

This proposed model is developed by using multi agent system. Wooldridge defined an agent as a software or entity that can observe its environment, make decision, and perform actions that affect its environment and its own or other agents' internal states [17]. As a multi agent system, this model consists of three agents: inbound trucks scheduler agent, outbound trucks scheduler agent, and material handler agent. Every agent cannot be interfered or subordinated by other agent based on the autonomous concept of the multi agent system [18]. The role of the inbound-trucks scheduler is to conduct the inbound-trucks traffic flow in the cross-docking terminal. The role of the outbound-trucks scheduler is to conduct the outbound-trucks traffic flow in the cross-docking terminal. The role of the material handler is to organize the material flow among inbound truck, outbound truck, and temporary storage. This model is illustrated in Fig. 2. As an agent-based model, each agent has its own goal that must be achieved which is explained later. These three agents are developed as rule-based agents since they behave based on specific rules or mechanisms [19]. This rule-based concept is chosen due to its simplicity as a collection of conditional statements which is stored and selected based on the condition to achieve better performance [20].

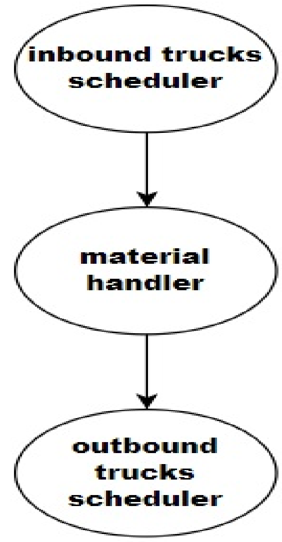

Fig. 2. Multi Agent Architecture in the Cross-docking System.

The explanation of Fig. 2 is as follows. The inbound trucks scheduler arranges the inbound truck sequence or traffic based on the temporary storage and outbound door condition. Then, the material handler has responsibility in managing product traffic among inbound door, outbound door, and temporary storage. Finally, the outbound trucks scheduler arranges the outbound truck sequence or traffic based on the temporary storage and inbound door condition. These three agents work autonomously but share information among them.

Before we explain further, the notations used in this model are as follows.

i	inbound truck index
j	outbound truck index
n_{pv}	product variety
p	product
P	set of products
q_{av}	available quantity
q_{re}	requested quantity
s	Status
s_{so}	inbound-outbound contribution status
s_{sw}	inbound-storage contribution status
t_i	inbound truck
t_{is}	selected inbound truck
t_o	outbound truck
t_{os}	selected outbound truck
w	temporary storage
Δ	gap

The mechanism of the inbound-trucks scheduler is as follows. When the inbound door is empty, the inbound-truck scheduler will check whether there exists inbound truck in the queue then determine this truck to go to the inbound door. When there are several inbound trucks in the queue, then the scheduler will decide which truck is assigned to the door for docking process. During the docking process, this inbound truck unloads its products sequentially. If this inbound truck cannot unload its products but still has products inside it, this truck will be assigned back to the queue. This policy is taken due to the limited capacity of the temporary storage. This concept is different from other previous studies about cross-docking where the inbound truck will stay at the outbound door until all the carried products are completely unloaded to the outbound trucks or the temporary storage [1]. When all the carried products have been unloaded, this truck will leave the inbound door immediately [1]. The activities of the inbound truck are illustrated in Fig. 3. Meanwhile this mechanism is formalized by using (1) to (6). This process can also be seen in algorithm 1.

algorithm 1: inbound trucks scheduler

```
while inbound queue > 0 do
  find the fittest inbound truck
  if unloading is possible then
    unloading
  else
    if truck payload > 0 then
      back to queue
    else
      leave the system
    end if
  end if
end while
```

$$s(t_i) = \begin{cases} 0, t_i \neq t_{is} \wedge q_{av}(t_i) > 0 \\ 1, t_i = t_{is} \\ 2, t_i \neq t_{is} \wedge q_{av}(t_i) = 0 \end{cases} \quad (1)$$

$$t_{is} = \begin{cases} min\big(\Delta(t_i, t_{os})\big), \exists t_{os} \\ max\left(n_{pv}(t_i, w)\right), \nexists t_{os} \end{cases} \quad (2)$$

$$\Delta(t_i, t_{os}) = \sum_{p \in P} \Delta\left(q_{av}(t_{i,p}), q_{re}(t_{os,p})\right) \quad (3)$$

$$\Delta\left(q_{av}(t_{i,p}), q_{re}(t_{os,p})\right) = \begin{cases} q_{re}(t_{os,p}) - q_{av,p}, q_{re}(t_{os,p}) > q_{av,p} \\ 0, else \end{cases} \quad (4)$$

$$q_{av,p} = q_{av}(t_i, p) + q_{av}(w_p) \quad (5)$$

$$n_p(t_i, w) = \sum_{p \in P} p, q_{av}(t_{i,p}) > 0 \wedge q_{av}(w_p) = 0 \quad (6)$$

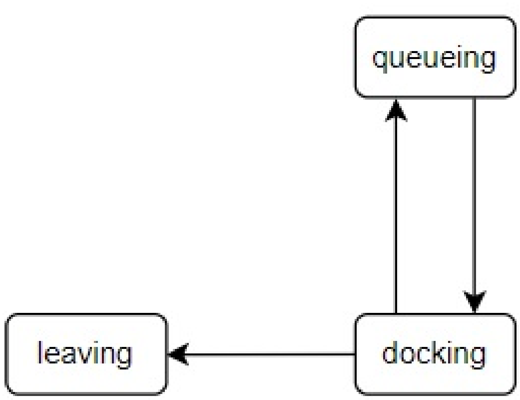

Fig. 3. Inbound Truck State Diagram.

Equation (1) shows that there are three possible values of the inbound truck status. Status 0 indicates the inbound truck is in the queue. Status 1 indicates that the inbound truck is at the inbound door. Status 2 indicates that the inbound truck leaves the cross-docking system.

Equation (2) shows that there are two options in determining the selected inbound truck. It depends on whether there exists outbound truck at the outbound dock. In case there exists an outbound truck at the outbound dock, the shortest remaining time scheduling is adopted so that the outbound truck can be served as fast as possible. The shortest remaining time scheduling method is chosen due to its characteristics that prioritize the fastest job that can be executed due to current condition [21]. It is done by choosing the most suit inbound truck that can provide the selected outbound truck. In case there does not exist an outbound truck at the outbound dock, the pre-emptive of the shortest remaining time is adopted. It is done by selecting the most suit inbound truck that can improve the product variety in the temporary storage. Higher product variety may improve product movement possibility [22].

The explanation of (3) to (6) is as follows. Equation (3) shows that the gap between the inbound truck and the selected outbound trucks is accumulation of all products. Eq. (4) shows that the gap is calculated only if the quantity of the requested products in the selected outbound truck is more than the quantity of the available product. Equation (5) shows that the available product is the accumulation of products in the inbound truck and the temporary storage. Equation (6) formalized the calculation of the number of product variation in the temporary storage that can be supported by the inbound truck. The result of (6) is used in (2).

The inbound truck still stays at the inbound door only if it can contribute to the cross-docking terminal. First, this truck still has product in its container. Empty truck cannot contribute so that it must leave the inbound door immediately. If there is selected outbound truck, this inbound truck must have product needed by the selected outbound truck. If there does not exist selected outbound truck, then the inbound truck still can contribute only if the temporary storage current capacity is still less than its maximum capacity. This mechanism is formalized by using (7) to (9).

$$s(t_{is}) = \begin{cases} 1, s_{so}(t_{is}) = 1 \vee s_{sw}(t_{is}) = 1 \\ 0, else \end{cases} \quad (7)$$

$$s_{so}(t_{is}) = \begin{cases} 1, \exists t_{os} \wedge \exists(q_{av}(t_{is}, p) > 0 \wedge q_{re}(t_{os}, p) > 0) \\ 0, else \end{cases} \quad (8)$$

$$s_{sw}(t_{is}) = \begin{cases} 1, \nexists t_{os} \wedge \exists\big(q_{av}(t_{is}, p)\big) \wedge q(w) < q_{max}(w) \\ 0, else \end{cases} \quad (9)$$

The explanation of (7) to (9) is as follows. Equation (7) formalizes the selected inbound truck status, whether it still can contribute or not. It depends on two aspects. The first aspect is whether it still can contribute to the selected outbound truck, which is formalized in (8). The second aspect is whether it still can contribute to the temporary storage which is formalized in (9).

The mechanism of the outbound-trucks scheduler is as follows. When the outbound door is empty, the outbound truck scheduler will check whether there exists an outbound truck in the queue so that it can move to the outbound door. If there are several outbound trucks in the queue, then selection process runs. The scheduler selects the most possible truck to be provided by the temporary storage without observing whether there exists an inbound truck at the inbound door. This flow is illustrated in Fig. 4 and algorithm 2.

algorithm 2: outbound trucks scheduler

while outbound queue > 0 **do**
  find the fittest outbound truck
  loading
  leave the system
end while

This concept is adopted based on the shortest remaining time scheduling [21]. Different from the inbound truck which can be transferred back to the queue, once the outbound truck is in the outbound door, it remains there until all its requested products are fulfilled [1]. Once all its requested products are fulfilled, this outbound truck then leaves the cross-docking system [1]. This mechanism is formalized by using (10) to (13).

$$s(t_o) = \begin{cases} 0, t_0 \neq t_{os} \wedge q_{re}(t_o) > 0 \\ 1, t_o = t_{os} \\ 2, t_o \neq t_{os} \wedge q_{re}(t_o) = 0 \end{cases} \quad (10)$$

$$t_{os} = t_o, min\big(\Delta(t_o, w)\big) \quad (11)$$

$$\Delta(t_o, w) = \sum_{p \in P} (q_{re}(t_{is,p}) - q_{av}(w_p)), q_{re}(t_{is,p}) > q_{av}(w_p) \quad (12)$$

$$s(t_{os}) = \begin{cases} 1, \exists q_{re}(t_{os}) > 0 \\ 0, else \end{cases} \quad (13)$$

The explanation of (10) to (13) is as follows. Equation (10) shows that there are three possible values of the outbound truck. Status 0 indicates that the outbound truck is in the queue. Status 1 indicates that the outbound truck is at the outbound door. Status 2 indicates that the outbound truck leaves the cross-docking system. Equation (11) and (12) indicate that the selected outbound truck is the outbound truck with the minimum gap between its requested products and the available products in the temporary storage. Equation (14) shows that the selected outbound truck is still active until its request is completely fulfilled.

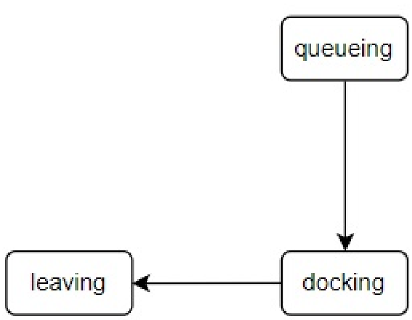

Fig. 4. Outbound Truck State Diagram.

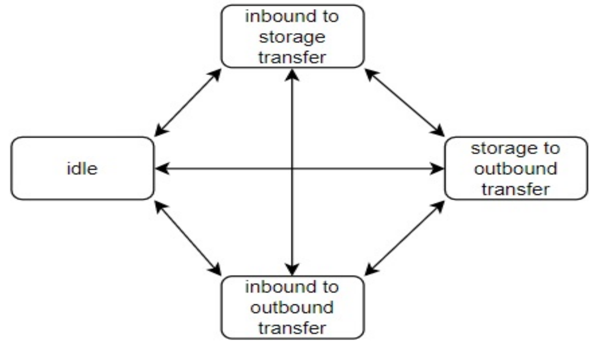

Fig. 5. Material Handling State Diagram.

The primary goal of the material handler agent is serving the selected outbound truck as fast as possible. Meanwhile, the secondary goal is transferring the products in the inbound truck container as fast as possible while maintaining low temporary storage or inventory level [14] so that it is not exceeded. It means that the material scheduler observes the existence of the selected outbound truck first before the selected inbound truck. Fig. 5 illustrates the product movement in the cross-docking terminal.

The material scheduler prioritizes to fulfill the outbound truck request from the inbound truck [2] rather than from the temporary storage so that the container of the inbound truck will be empty faster so that the waiting time in the inbound queue can be minimized [13]. The request is fulfilled by the temporary storage only if the request cannot be fulfilled by the inbound truck. If there does not exist an outbound truck in the outbound door, the product moves from the selected inbound truck to the temporary storage only if its current capacity is still less than its maximum capacity. This mechanism is formalized by using (15). This process is also formalized in algorithm 3.

algorithm 3: material handler

if inbound-to-outbound is possible **then**
  move product from inbound to outbound
else
  if storage-to-outbound is possible **then**
  move product from storage to outbound
  else
  if inbound-to-storage is possible **then**
   move product from inbound to storage
  else
   idle
  end if
  end if
end if

$$A(w) = \begin{cases} m(t_{is}, t_{os}), \exists t_{is} \wedge \exists t_{os}, \exists\big(q_{av}(t_{is,p}) > 0 \wedge q_{re}(t_{os,p}) > 0\big) \\ m(w, t_{os}), \nexists t_{is} \wedge \exists t_{os}, \exists\big(q_{av}(w_p) > 0 \wedge q_{re}(t_{os,p}) > 0\big) \\ m(t_{is}, w), \exists t_{is} \wedge \nexists t_{os} \wedge q(w) < q_{max}(w) \end{cases} \quad (15)$$

IV. Result and Discussion

This proposed model is then implemented into truck scheduling simulation so that its performance can be evaluated. In this simulation, there are two observed variables: total time and maximum inventory level. Total time is one of the most important operational parameters in the cross-docking system. It is also widely observed in many studies [2,8-10]. Maximum inventory level is observed due to one of the goals in implementing cross-docking system is reducing inventory level [14]. In this simulation, this proposed multi agent based (MAS) model is compared with three previous truck scheduling models: inbound FIFO (I-FIFO) model [4], outbound-FIFO-integer programming (O-FIFO-IP) model [7], and mixed-integer programming (MIP) model [2]. In this work, there are three simulations. The first simulation is conducted to observe the relation between the changeover time and the observed parameters. The second simulation is conducted to observe the relation between the maximum storage capacity and the total time. The third simulation is conducted to observe the relation between the number of trucks and the total time per truck.

In this simulation, there are several default variables. These default variables are applied in these three simulations. The number of products is 20 units. The average number of products carried by the inbound trucks is 5 units. The average number of products carried by the outbound trucks is 10 units. The products distribution in every truck follows normal distribution. The truck capacity is 50 units. The product moving time is 1 time-unit. In this work, the time-unit is used as unit for time related variables, such as total time and changeover time, to generalize the simulation.

In the first simulation, we observe the relation between changeover time and the observed variables. The reason is that the changeover time affects the total time so that in several studies, the truck stays at the dock until it completes its work [1,2]. In this simulation, the number of inbound trucks is 4 units. The number of outbound trucks is 4 units. The temporary storage capacity is 50 units. The changeover time ranges from 5 to 15 time-unit. The result is shown in Fig. 6 and Fig. 7.

Fig. 6 shows that the increasing of the changeover time makes the total time increases. This trend tends to be linear. This trend occurs in all models. Compared among models, the proposed model is competitive enough. The MIP model [2] performs as the best model in creating low total time. Compared with the MIP model, the proposed model performs 7.2 percent higher in total time. Meanwhile, compared with the I-FIFO [4] and the O-FIFO-IP [7] models, the proposed model performs better. Compared with the I-FIFO model [4], the proposed model performs 2.2 percent lower in total time. Compared with the O-FIFO-IP model, the proposed model performs 0.2 percent lower in total time.

The explanation of Fig. 7 is as follows. The increasing of the changeover time increases the maximum inventory level in all previous works [2,4,7]. Meanwhile, in the proposed model, the maximum inventory level does not change. Compared with the previous models, the proposed model performs as the best model in minimizing the maximum inventory level. Compared with the I-FIFO model [4], it performs 55.1 percent lower. Compared with the O-FIFO-IP model [7], it performs 52.1 percent lower. Compared with the MIP model [2], it performs 41.8 percent lower.

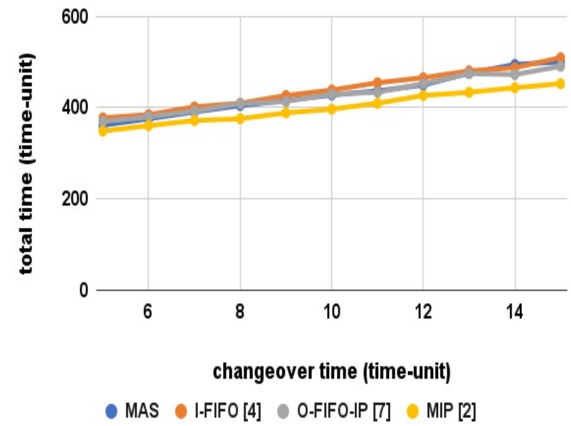

Fig. 6. Relation between Changeover Time and Total Time.

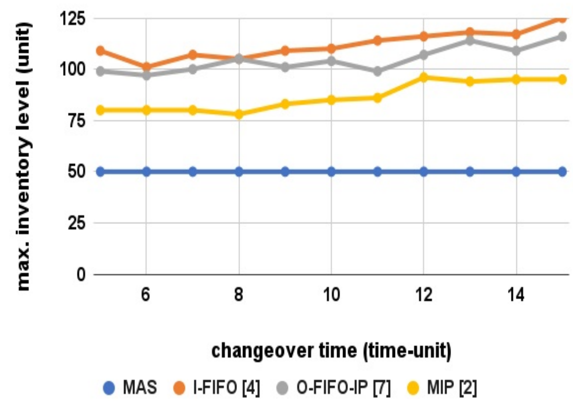

Fig. 7. Relation between Changeover Time and Maximum Inventory Level.

In the second simulation, we observe the relation between the temporary storage capacity and the observed variables. In this simulation, the number of inbound trucks is 4 units. The number of outbound trucks is 4 units. The changeover time is 5 time-unit. The temporary storage capacity ranges from 50 to 100 units. The result is shown in Fig. 8 and Fig. 9. This simulation occurs only for the proposed model because in the compared models [2,4,7], the temporary storages capacity does not become a constraint. The simulation result is then compared with the previous compared models.

Fig. 8 shows that in the proposed model, the increasing of the temporary storage capacity does not affect the total time. The total time tends to be stable. The total time ranges from 353 time-unit to 366 time-unit with the average value is 358.7 time-unit. Compared with the previous models, the proposed model is still worse than the MIP model [2] but better than the I-FIFO [4] and the O-FIFO-IP [7] models.

Fig. 9 shows that in the proposed model, the increasing of the temporary storage capacity makes the maximum inventory level increase too. When the temporary storage capacity ranges from 50 to 65 units, the maximum inventory level is equal to the temporary storage capacity. After that the maximum inventory level still increases but it is below the temporary storage capacity.

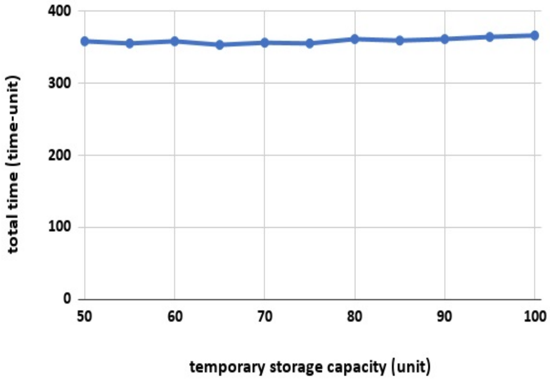

Fig. 8. Relation between Temporary Storage Capacity and Total Time.

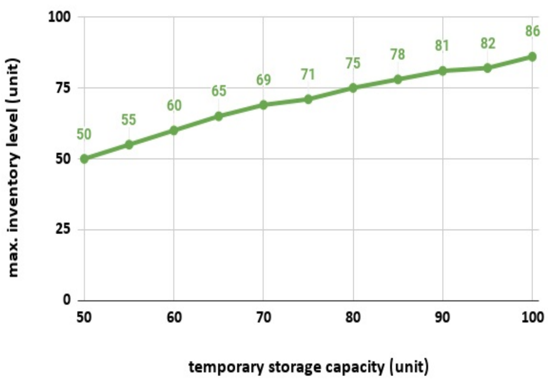

Fig. 9. Relation between Temporary Storage Capacity and Maximum Inventory Level.

Comparison between the proposed model and the previous models is as follows. Compared with the previous models [2,4,7], when the temporary storage capacity is high (100 units), the proposed model still performs as the best model in creating low maximum inventory level. Compared with the I-FIFO model [4], it performs 31.2 percent lower. Compared with the O-FIFO-IP model [7], it performs 25.8 percent lower. Compared with the MIP model [2], it performs 10.4 percent lower.

In the third simulation, we observe the relation between the number of trucks and the total time per truck. In this simulation, the changeover time is 5 time-unit. The temporary storage capacity is 50 units. The number of inbound or outbound trucks ranges from 5 to 15 units. The result is shown in Fig. 10.

Fig. 10 shows that the increasing of the number of trucks makes average total time per truck increase too. It is because the scheduling becomes more complicated. Fortunately, this increasing is not significant. When the number of inbound or outbound trucks increases 100 percent (from 5 units to 10 units), the total time per truck increases only 8.8 percent. Meanwhile, when the number of inbound and outbound trucks increases 200 percent (from 5 to 15 units), the total time per truck increases only 10.9 percent. It means that the waiting time also increases with low inclination due to the increasing of the number of inbound or outbound trucks.

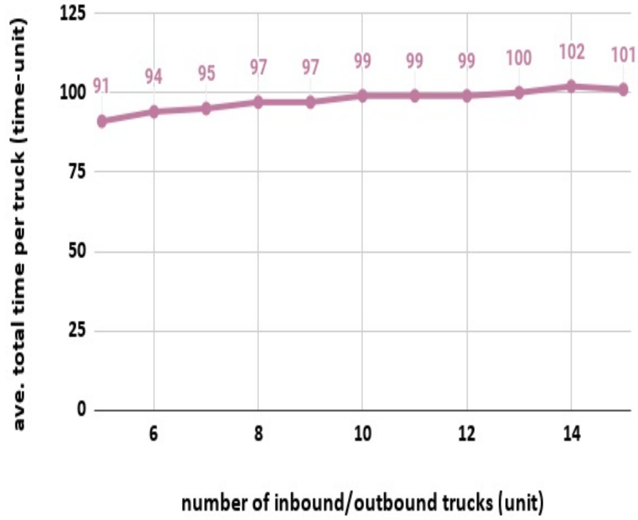

Fig. 10. Relation between Number of Inbound or Outbound Trucks and Average Total Time per Truck.

Based on the result, there are several findings due to this work. First, overall, our proposed model performs as the best model in creating low inventory level as it is one of goals in the cross-docking system [14]. Second, our proposed model is still competitive in creating low total time as it is the main goal of the cross-docking system in making efficient and fast processing mechanism [2,8-10]. The integer programming model still performs as the best model in finding optimal solution [23]. The problem in implementing the integer programming is its computationally expensive due to its polynomial time algorithm characteristic [23]. In the other side, the shortest remaining time algorithm that is adopted in our proposed model is a linear time algorithm so that our proposed model is computationally lighter than the integer programming-based truck scheduling model.

The third finding is that our proposed model still creates low waiting time due to the increasing of the number of trucks because the total time per truck increases but with low inclination. This condition also related with one of the goals of the cross-docking system in minimizing waiting time [10,13].

V. Conclusion

This work shows that the proposed model, which is developed by using multi-agent system and shortest remaining time algorithm, can solve the truck scheduling problem in the cross-docking system. It is proven competitive compared with the existing FIFO based model and integer-programming based model. This proposed model performs as the best model in creating low inventory level. Compared with the integer-programming model, it creates 41.8 percent lower in maximum inventory level. Compared with the FIFO based model, it creates 52.1 to 55.1 percent lower in maximum inventory level. In total time aspect, it creates 0.2 to 2.2 percent lower than the FIFO based model. Although this proposed model creates 7.2 percent higher in total time compared with the integer-programming based model, it is computationally lighter due to its complexity is linear time rather than the integer-programming based model that its complexity is polynomial time. In this proposed model, the waiting time increases with low inclination due to the increasing of the number of trucks.

This work has proposed new truck scheduling model in the cross-docking system by using multi-agent system. In the future, the multi-agent system can be implemented in the cross-docking system not only for the truck scheduling. Moreover, it can be implemented in the material handling and forklift management. Forklift management can be modeled as a swarm robot with collaborative approach. This work uses single-inbound-single-outbound scenario. In the future, it can also be expanded into multiple doors where each door is treated as an autonomous agent and there is collaboration among the doors.

Acknowledgment

This work was supported by Telkom University, Indonesia.

References

[1] W. Yu, C. Ha, and S. Park, "A hybrid genetic algorithm for integrated truck scheduling and product routing on the cross-docking system with multiple receiving and shipping docks", Mathematical Problems in Engineering, article ID: 2026834, pp. 1-17, 2021.

[2] 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, article ID: 8813372, pp. 1-13, 2020.

[3] R. K. Mavi, M. Goh, N. K. Mavi, F. Jie, K. Brown, S. Biermann, and A. A. Khanfar, "Cross-docking: a systematic literature review", Sustainability, vol. 12, article ID: 4789, 2020.

[4] R. Larbi, G. Alpan, P. Baptiste, and B. Penz, "Scheduling cross docking operations under full, partial, and no information on inbound arrivals", Computers & Operations Research, vol. 38, pp. 889-900, 2011.

[5] M. A. Dulebenets, "A diploid evolutionary algorithm for sustainable truck scheduling at a cross-docking facility", Sustainability, vol. 10, article ID: 1333, 2018.

[6] T. Charqui, A. Bekrar, M. Reghioui, and D. Trentesaux, "Multi-objective sustainable truck scheduling in a rail-road physical internet cross-docking hub considering energy consumption", Sustainability, vol. 11, article ID: 3127, 2019.

[7] D. Molavi, A. Shahmardan, M. S. Sajadieh, "Truck scheduling in a cross docking systems with fixed due dates and shipment sorting", Computer & Industrial Engineering, vol. 117, pp. 29-40, 2018.

[8] H. Mohammadzadeh, N. Sahebjamnia, A. M. Fathollahi-Fard, and M. Hahiaghaei-Keshteli, "New approached in metaheuristics to solve the truck scheduling problem in a cross-docking center", International Journal of Engineering-Transactions B: Applications, vol. 31, no. 8, pp. 1258-1266, 2018.

[9] A. M. Fathollahi-Fard, M. Ranjbar-Bourani, N. Cheikhrouhou, and M. Hajiaghaei-Keshteli, "Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system", Computer & Industrial Engineering, vol. 137, article ID: 106103, 2019.

[10] O. Theophilus, M. A. Dubelenets, J. Pasha, O. F. Abioye, and M. Kavoosi, "Truck scheduling at cross-docking terminals: a follow-up state-of-the-art review", Sustainability, vol. 11, article ID: 5245, 2019.

[11] 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.

[12] N. Boysen, "Truck scheduling at zero-inventory cross docking terminals", Computer and Operations Research, vol. 37, no. 1, pp. 32-41, 2010.

[13] K. L. Choy, H. K. H. Chow, T. C. Poon, G. T. S. Ho, "Cross-dock job assignment problem in space-constrained industrial logistics distribution hubs with a single docking zone", International Journal of Production Research, vol. 50, no. 9, pp. 2439-2450, 2012.

[14] A. L. Ladier and G. Alpan, "Cross-docking operations: current research versus industry practice", Omega, vol. 62, pp. 145-162, 2016.

[15] S. T. Khorasani, M. Keshtzari, M. S. Islam, and R. Feizi, "Intravenous fluid delivery time improvement: application of cross-docking system", International Journal of Health Care Quality Assurance, vol. 31, no. 8, pp. 1070-1081, 2018.

[16] Y. Ye, J. Li, K. Li, & 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.

[17] M. Wooldridge, An Introduction to MultiAgent Systems, 2nd ed., John Wiley & Sons, 2009.

[18] K. M. Lhaksmana, Y. Murakami, and T. Ishida, "Role-based modeling for designing agent behavior in self-organizing multi-agent systems", International Journal of Software Engineering and Knowledge Engineering, vol. 28, no. 1, pp. 79-96, 2018.

[19] J. Dietrich, A. Kozlenkov, M. Schroeder, and G. Wagner, "Rule-based agents for the semantic web", Electronic Commerce Research and Applications, vol. 2, no. 4, pp. 323-338, 2003.

[20] Sugiyanto, G. Fernando, and W. K. Tai, "A rule-based AI method for an agent playing big two", Applied Sciences, vol. 11, article ID: 4206, pp. 1-23, 2021.

[21] M. L. Pinedo, Scheduling: Theory, Algorithms, and Systems, 3rd ed., Springer, New York, 2008.

[22] T. J. van Kampen, D. P. van Donk, and D. J. van der Zee, "Safety stock or safety lead time: coping with unreliability in demand and supply", International Journal of Production Research, vol. 48, no. 24, pp. 7463-7481, 2010.

[23] M. Conforti, G. Cornuejols, and G. Zambelli, Integer Programming, Springer, Switzerland, 2014.