

## A study of cross dock with predefined door assignment and truck time windows using Petri net

Lijuan Zhang<sup>1</sup>, Benoît Trouillet<sup>1</sup>, Frédéric Semet<sup>1</sup>

<sup>1</sup>CRISTAL - UMR CNRS 9189, École Centrale de Lille CS 2048  
59651 Villeneuve d'Ascq, France  
{lijuan.zhang, benoit.trouillet, frederic.semet}@ec-lille.fr

**Abstract.** The cross dock management problems (CDMP) are always formulated as a mixed integer programming (MIP) model, and various heuristics and exact approaches have been proposed to solve this kind of problems. Certain authors study the CDMP by simulation, and the most frequently-used tool is Arena. In this paper, we address a resource management problem with predefined door assignment and truck time windows in cross dock. A model, based on Petri net, is proposed. Thanks to the Petri net model, we simulate the behaviors of the terminal with different shared resource numbers. The results obtained can provide helpful advices to improve the original time windows, present work distribution and manage resources in cross dock.

**Keywords:** cross dock, door assignment, resource management, Petri net, time windows.

### 1 Introduction

As an efficient and effective logistic procedure that directly transfer the goods from origins to destinations, compared with the traditional distribution center, cross dock can cut cost, reduce risk of product damages, as well as save storage space and delivery lead time.

The basic concept behind cross-docking is to eliminate the two most costly operations in a distribution center by transferring the incoming goods directly from receiving docks to shipping docks. The activities of cross dock terminal are as follows: goods from suppliers are shipped into the cross dock center, and then they are unloaded at the receiving doors, and sorted according to their destinations and directly transferred to the shipping doors, after that, they are loaded in trucks and delivered to customers.

In this paper, we address a resource management problem with predefined door assignment and truck time windows in cross dock. A model, based on T-timed Petri net is proposed. With the Petri net model, the behaviors of the terminal with different resource numbers are simulated. Certain literatures and methods for solving problems at operational level are reviewed in Section 2, Section 3 details the hypothesis for modelling cross dock using Petri net. The strategy of the modelling are explained in Section 4. The simulation results are reported in Section 5, while conclusions are drawn in Section 6.

### 2 Literature review

The current literatures on cross dock management problems consist of cross dock network design, layout design, dock door assignment, transshipment, vehicle routing and scheduling problems. These studies can be classified according to the three decision levels: strategic, tactical and operational. The door assignment problem and truck scheduling problem are at the operational level.

These kinds of problems are always formulated as a MIP model and various methods have been proposed to solve them. The most frequently-used exact approach is a Branch and Bound (B&B) method, but which is no longer efficient as the problem size grows. There are a considerable number of heuristics which are applied to obtain an approximation value, such as Genetic algorithm, Tabu search, Ant Colony Optimization, etc.

A bilinear programming model is formulated by Y.Louis and C.Chang [1] to describe a basic dock door assignment problem: trucks from the suppliers come in, goods are unloaded at the receiving doors, then goods are loaded in outbound trucks at the shipping doors. The movements of goods between receiving door and shipping door are made by forklift. Their objective is to minimize the travel distance. In this formulation, one door must be assigned to one truck, and one truck must get assigned only one door, that is to say that the number of suppliers must equal to number of receiving doors and the number of customers must be equal to the number of shipping doors. The authors [2] propose the Branch and Bound algorithm to solve this problem, and computational results show that the CPU time increases dramatically as the problem size grows.

Y. Zhu et al. [5] improve the formulation by Y.Louis and C.Chang [1] so that the number of suppliers and receiving doors can be different and the number of customers and shipping doors can also be different. M.Guignard et al. [4] propose two heuristics: local search and Convex Hull, to solve the model by Y. Zhu et al. [5]. Compared with the Branch and Bound method, the two heuristics are more efficient for the big size instances.

A.Lim et al. [6] bring in time windows of trucks. Each truck has an arrival time at the cross dock and a departure time from the cross dock. Trucks have to be assigned according to their time windows, the number of trucks can exceed number of doors and two trucks cannot be assigned to the same door when their time windows overlap. Tabu search and genetic algorithm are proposed. Compared with CPLEX solver, the computational results show that the heuristic algorithms can obtain better solutions within less CPU time.

The literatures about cross dock door assignment and scheduling problems with MIP formulations do always not consider the activity of resources.

Certain authors also study behaviors of cross dock by simulation, and the most frequently-used tool is Arena. During simulation, the door assignment is often not considered and predefined. The time cost by each operation (e.g. Liong and Loo [9]), the personnel (resource) planning (e.g. Liu and Takakuwa, [10]) are always the concerns of authors.

Until now, we find only one article by B.Trouillet [7] which proposes Petri net to study cross dock. The author presents the main operations: unloading, sorting and loading in cross dock with Petri net. With the presentations of Petri net model, the author proposes three classes of constraints to improve the IP model proposed by A.Lim et al. [6]. However, the representation is not detailed enough, the author does not consider the activity of resources and does not carry out the experiments with the Petri net model.

In this article, we detail the strategy of modelling cross dock using Petri net. The door assignment is predefined. The resources are considered and shared by all the operations. A model is built and the simulation is carried out. The performance of cross dock is evaluated and analyzed with different resource number and the total resource number required in cross dock is determined.

### 3 Hypothesis

Before the modelling, we make some hypothesis.

Hypothesis 1: The door assignment for the inbound trucks and outbound trucks are predefined.

The door assignment is obtained with the MIP formulation by Zhang et al. [8]. The inbound trucks *IT1* and *IT4* are assigned in receiving door *RD1*, *IT2* and *IT3* are assigned in *RD2*, the outbound trucks *OT1*, *OT2* and *OT4* are assigned in shipping door *SD1*, and *OT3* is assigned in shipping door *SD2* (the trucks of which the time windows overlap cannot be assigned in a same door).

Hypothesis 2: Pre-emption is not allowed. Unloading or loading of a truck cannot be interrupted.

Hypothesis 3: Before the earliest truck arrives, the cross dock is totally unloaded, there is not any work in cross dock.

The operations in cross dock include unloading, sorting and loading. In the following, we detail how to build the model operation by operation.

## 4 Strategy of modelling

Petri net is a graphic and mathematic tool, created in 1962 by Carl Adam Petri. The proprieties of Petri net can be referred to the paper by Murata [11].

In this section, we explain the steps to build a Petri net model for cross dock using the following example. We consider 3 receiving doors, 3 shipping doors, 4 inbound trucks with their time windows arrival time in cross dock  $a_m = [16 \ 20 \ 39 \ 30]$ , depart time from cross dock  $d_m = [28 \ 36 \ 50 \ 40]$ , and 4 outbound trucks with their time windows, arrival time in cross dock  $a_n = [20 \ 42 \ 25 \ 51]$ , depart time from cross dock  $d_n = [30 \ 50 \ 46 \ 80]$ . The unloading/loading/sorting velocity per resource per unit time is  $p_1=5$ . The goods quantity transferred from inbound truck  $m$  to outbound truck  $n$  is  $w_{mn}$ , and

$$w_{mn} = \begin{bmatrix} 33 & 15 & 52 & 22 \\ 18 & 34 & 25 & 20 \\ 31 & 24 & 14 & 38 \\ 29 & 35 & 25 & 44 \end{bmatrix}$$

### 4.1 Unloading

This operation is modeled ‘door by door’ then ‘truck by truck’ and then ‘block by block’. First, all of the inbound trucks are classified according to the receiving door where they are assigned. Then the inbound trucks which share the same door are separately modeled. Last, the goods in each inbound truck are modeled block by block, where block is a unit for count goods sets in unloading step. For example, there are 33 goods from  $IT1$  to  $OT1$  and unloading velocity per unit time per operator is 5, there will be 7 blocks (33/5).

The model for describing unloading in receiving door  $RDI$  is shown in Figure 1. One column represents the operations on one block, and the goods blocks from inbound truck to outbound truck  $OT1$ ,  $OT2$ ,  $OT3$  and  $OT4$  are listed from left to right. The time above  $t_0$  and  $t_4$  is the arrival time of  $IT1$  and  $IT4$ , and the time above  $t_2$  and  $t_3$  represents that it consumes one unit time for unloading the first block. Unloading can be started when:

- Inbound truck is assigned in receiving door ( $t_1$ );
- There is resource to do the unloading ( $P_{23}$ );

Assignment is described by transition  $t_1$  and  $t_5$ . This activity depends on two conditions:

- Inbound truck arrive in the cross dock facility ( $t_0$  and  $t_4$ );
- There is receiving door available to receive this inbound truck ( $P_{21}$  and  $P_{22}$ );

The assignment order of  $IT1$  and  $IT4$  in  $RDI$  is determined by initial marking which is detailed in section 4.4.

The block from  $P_{10}$  to  $P_{15}$  represent the goods from  $IT4$  to  $OT1$ , and according to their time windows, these goods cannot be shipped and they are transported in storage area.

For each block, at the end of unloading operation, there are two places for describing their following behaviors (eg.  $P_6$  and  $P_7$ ), one place ( $P_6$ ) is for connecting with the sorting step which is presented in section 4.2, and the other one ( $P_7$ ) is for evaluating if the unloading is finished. For the goods blocks which are shipped to storage area, there is only one place to evaluate if they are unloaded.

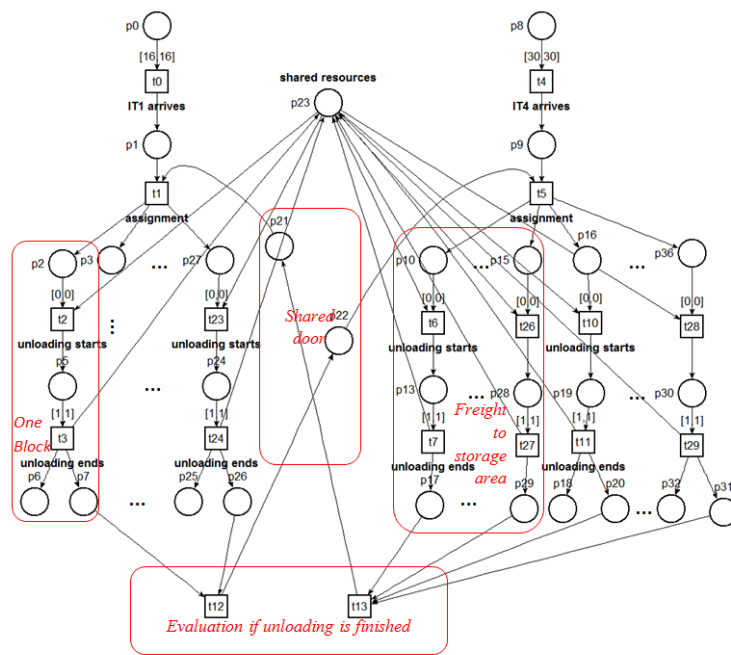


Figure 1: Unloading step at receiving door RD1

## 4.2 Sorting

Sorting operation is after the unloading step. In this section we detail how to build the model for one block, and models for the other blocks are similar. We suppose that transportations from *IT1* and *IT4* to *OT1*, *OT2* and *OT4* are automated by conveyors, those from *IT1* and *IT4* to *OT3* are by operators, as well as those from *IT2* and *IT3* to *OT3* are automated, and those from *IT2* and *IT3* to *OT1*, *OT2* and *OT4* are by operators. We also suppose that compared with transportations by operators, automatic transportation is no time cost and no labor cost.

The first block is transportation automatic from *IT1* to *OT1*, which is represented by a transition. For the blocks transported by operators, the model is shown in Figure.2. The times above  $t_{33}$  and  $t_{35}$  represent that it consumes one unit time for transporting one block.

Sorting step is connected with the operation loading which is represented in section 4.3.

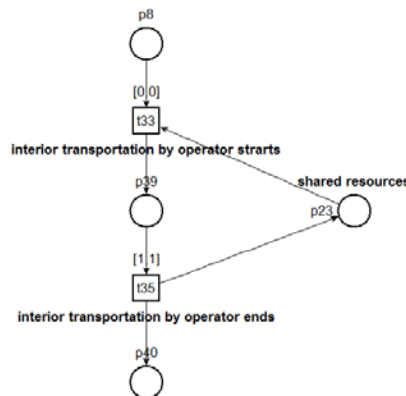


Figure 2: Internal transportation by operators

### 4.3 Loading

In this section, we explain loading of the first block from *IT1* to *OT1*, shown in Figure.3, to explain how to build the model for loading.

The time above  $t_{45}$  is the arrival time of *OT1*, assignment of *OT1* is described by transition  $t_{46}$ , which depends on two conditions:

- *OT1* arrives in cross dock terminal ( $t_{45}$ );
- There is shipping door available ( $P_{49}$ );

Loading process is represented by transitions  $t_{47}$  and  $t_{48}$ , and there are three conditions to begin loading:

- Blocks to *OT1* arrive in shipping door ( $P_{38}$ );
- *OT1* is assigned in a shipping door ( $P_{48}$ );
- There is resource for loading ( $P_{23}$ );

The Petri net for describing door assignment of outbound trucks are shown in Figure.4. In the case cited, *OT1*, *OT2* and *OT4* share the shipping door 1 (*SD1*), and *OT3* is assigned in *SD2*. The assignment order of trucks in the same door is determined by initial marking which is detailed in section 4.4.

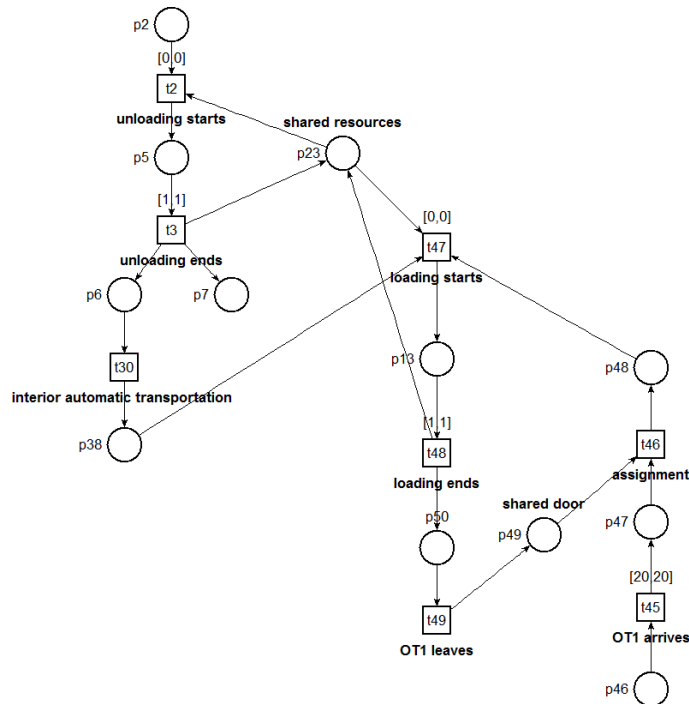


Figure 3: Model for loading

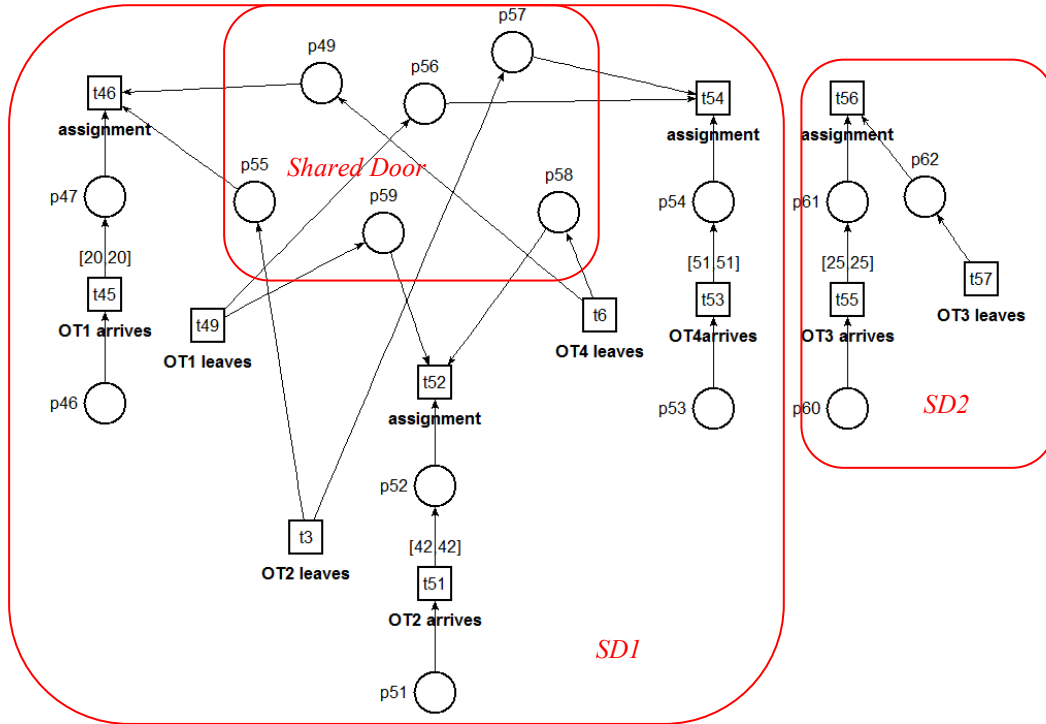


Figure 4: Door assignment of outbound trucks

#### 4.4 Initial marking

To simulate the behavior of cross dock, it is enough to combine the unloading, sorting, and loading model as an entire model and set up the initial marking.

For the trucks which get assigned in the same door, we suppose that the earlier the truck arrives, the earlier it can be assigned. The assignment order in Petri net model is controlled by initial marking. For instance, *IT1* arrives earlier than *IT4*, in Figure.1, and we set up  $M_0(p_{21}) = 1$  and  $M_0(p_{22}) = 0$ . The initial marking setting up is shown in Table 1.

Table 1: Initial marking

Mark	Value	Behaviors
$M(p_0)$	1	evaluate if inbound truck <i>IT1</i> arrives
$M(p_8)$	1	evaluate if inbound truck <i>IT4</i> arrives
	1	similarly, the places for evaluating if <i>IT2</i> & <i>IT3</i> arrives
$M(p_{60})$	1	evaluate if outbound truck <i>OT3</i> arrives
$M(p_{46})$	1	evaluate if outbound truck <i>OT1</i> arrives
$M(p_{51})$	1	evaluate if outbound truck <i>OT2</i> arrives
$M(p_{53})$	1	evaluate if outbound truck <i>OT4</i> arrives
$M(p_{21})$	1	evaluate whether receiving door <i>RD1</i> is available, and in <i>RD1</i> , <i>IT1</i> is earlier assigned than <i>IT4</i> .
	1	similarly, the places for evaluating if <i>RD2</i> is available and determining assignment order of <i>IT2</i> & <i>IT3</i> .
$M(p_{62})$	1	evaluate whether shipping door <i>SD2</i> is available
$M(p_{49})$ $M(p_{55})$ $M(p_{58})$	1	evaluate whether shipping door <i>SD1</i> is available, and decide assignment order in shipping door <i>SD1</i> is firstly <i>OT1</i> , then <i>OT2</i> , lastly <i>OT4</i> .
$M(p_{23})$	<i>N</i>	<i>N</i> is a positive integer number which represent the resource number in cross docks
Others	0	

## 5 Simulation results

The Petri net model for the case is built, and we define the firing rules of transitions. For all the transitions that are simultaneously fired:

- Priority rules of firing: the unloading operations have greater priority than interior transportations, and interior transportations have greater priority than loading operations.
- Unloading firing priority: the earlier the inbound truck arrives in cross dock, the greater priority it has for unloading.
- Loading firing priority: the earlier the outbound truck arrives in cross dock, the greater priority it has for loading.
- For all of the transitions that have same priority, they are randomly fired.

We simulate the behavior of the terminal with different resource number ( $N$ ) with software Tina, and we obtain the results as follows.

### 5.1 Proposition of time windows

We note the beginning time and finishing time of unloading/loading of each inbound/outbound truck and propose these time points as the new time windows shown in Table 2.

**Table 2:** Time windows proposed for trucks with different resource number

Resource number ( $N$ )	New proposed time windows					Resource number ( $N$ )	New proposed time windows				
1	$a_m$ :	16	42	90	62	7	$a_m$ :	16	20	39	30
	$d_m$ :	42	62	113	90		$d_m$ :	20	25	43	34
	$a_n$ :	156	191	167	213		$a_n$ :	26	44	27	51
	$d_n$ :	167	213	191	239		$d_n$ :	28	48	45	55
2	$a_m$ :	16	29	53	39	8	$a_m$ :	16	20	39	30
	$d_m$ :	29	39	64	53		$d_m$ :	20	24	42	34
	$a_n$ :	86	103	91	115		$a_n$ :	25	43	26	51
	$d_n$ :	91	115	103	128		$d_n$ :	27	47	44	55
3	$a_m$ :	16	25	40	31	9	$a_m$ :	16	20	39	30
	$d_m$ :	25	32	48	41		$d_m$ :	19	23	42	33
	$a_n$ :	62	74	66	82		$a_n$ :	24	43	25	51
	$d_n$ :	67	82	75	91		$d_n$ :	26	46	44	54
4	$a_m$ :	16	22	40	31	10	$a_m$ :	16	20	39	30
	$d_m$ :	23	28	46	38		$d_m$ :	19	22	42	33
	$a_n$ :	51	59	53	66		$a_n$ :	23	42	25	51
	$d_n$ :	54	66	60	73		$d_n$ :	25	46	43	54
5	$a_m$ :	16	21	40	31	11	$a_m$ :	16	20	39	30
	$d_m$ :	22	26	45	37		$d_m$ :	19	22	42	33
	$a_n$ :	30	51	39	56		$a_n$ :	23	42	25	51
	$d_n$ :	39	56	51	62		$d_n$ :	25	45	43	54
6	$a_m$ :	16	20	39	30	12	$a_m$ :	16	20	39	30
	$d_m$ :	21	24	43	35		$d_m$ :	19	22	41	33
	$a_n$ :	28	45	29	51		$a_n$ :	22	42	25	51
	$d_n$ :	30	50	46	56		$d_n$ :	24	45	43	54

We review the original time windows:

$a_m$	16	20	39	30
$d_m$	28	36	50	40
$a_n$	20	42	25	51
$d_n$	30	50	46	80

We can see that, with the resource number grows, the depart time of last outbound truck does not always decrease. Under constraints of time windows defined in this case in Section 3, it needs to determine how many resources required. We compare the original time windows with the new proposed time windows to verify if they coincide. It is obvious that keeping 6 resources working simultaneously in this cross dock is an advisable choice: it can maximum stay in step with the original windows with minimum resource. The resource number inferior 6 is not feasible solution, while the resource number superior 6 is the feasible solution. However, to respect the time windows defined, 6 resources are sufficient.

### 5.2 Makespan

Under the firing rules defined, we draw the makespan in this terminal: the working period of each operation at each working station. The makespan within  $N=6$  is shown in Figure.5 and makespan within  $N=10$  is shown in Figure.6. The work distribution at each time in terminal is clearly presented by makespan.

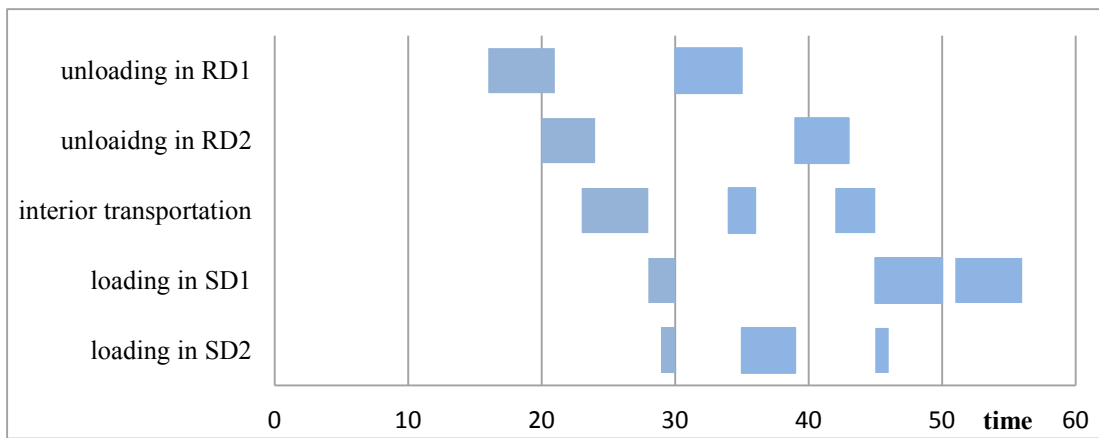


Figure 5: Makespan within N=6

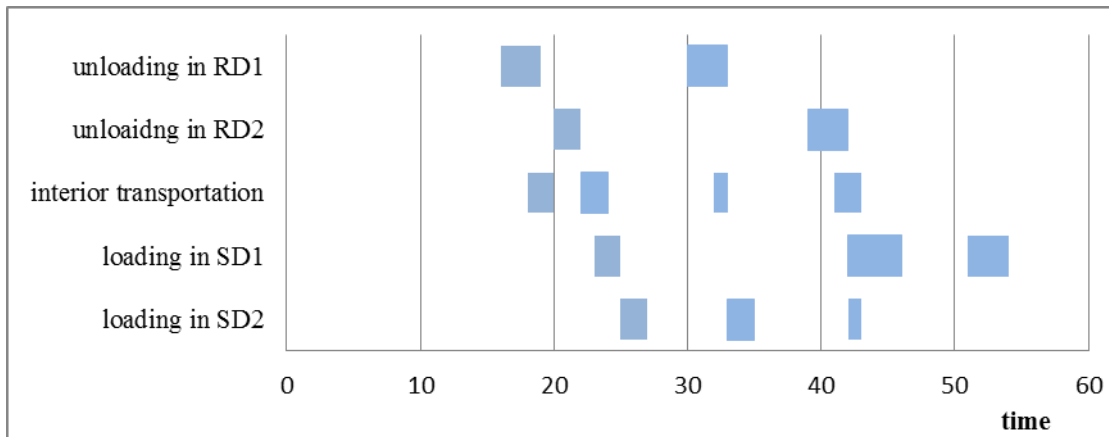


Figure 6: Makespan within N=10

### 5.3 Free resources

Definition: A *Free resource* is a resource which performs no task. The *Free resource number* is equal to the total resource number minus the occupied resource number.





## Acknowledgement

The authors are grateful for the financial support from China Scholarship Council.

## References

1. Tsui, L. Y., & Chang, C. (1990). A microcomputer based decision support tool for assignment dock doors in freight yards. *Computers & Industrial Engineering*, 19, 309–312.
2. Tsui, L. Y., & Chang, C. (1992). An optimal solution to a dock door assignment problem. *Computers and Industrial Engineering*, 23, 283–286.
3. Bermúdez, R., Cole, M. H., & Ph, D. (1999). A genetic algorithm approach to Door Assignments in Breakbulk Terminals. *Technical Report*, 1–19.
4. Guignard, M., Hahn, P. M., Cardoso, D., & Pessoa, A. A. (2012). Algorithms for the Cross-dock Door Assignment Problem. *Optimization Online*, 1–12.
5. Zhu, Y., Hahn, P. M., & Guignard-spielberg, M. (2009). New Approach for the Cross-Dock Door Assignment Problem. *Proceedings of the XLI Brazilian Symposium on Operations Research*, (Dmi), 1226–1236.
6. Lim, A., Ma, H., & Miao, Z. (2006). Truck Dock Assignment Problem with Time Windows and Capacity Constraint in Transshipment Network Through Crossdocks. *Computational Science and Its Applications-ICCSA 2006*, 688–697.
7. Trouillet, B. (2009). Petri Net Model and Mathematical Resolution for Cross Docking. *Conference on Intelligent Manufacturing Systems*, 10(1).
8. Zhang, L., Trouillet, B., & Semet, F. (2016). Door assignment and resource management problem with truck time windows constraints in cross docks, *ROADF2016*.
9. Liong, C.-Y., & Loo, C. S. E. (2009). a Simulation Study of Warehouse Loading and unloading system using ARENA. *Journal of Quality Measurement and Analysis*, 5(2), 45–56.
10. Liu, Y., & Takakuwa, S. (2009). Simulation-based personnel planning for materials handling at a cross-docking center under retail distribution environment. *Proceedings - Winter Simulation Conference*, 2414–2425.
11. Murata.T (1989). Petri nets: properties, analysis and applications. *Proceeding of the IEEE*, 77(4): 541-580, April 1989.