



Research Article



## A Soft Optimization Model to Solve Space Allocation Problems in Breakbulk Terminals

Masood Jafari Kang\*, Pooyan Mobtahej, Atefe Sedaghat, Maryam Hamidi

*Department of Industrial Engineering, Lamar University, Beaumont, TX, 77710, United States*

### Keywords

Freight Transportation,  
System Utilization,  
Space Allocation  
Problems,  
Warehouse,  
Cargo.

### Abstract

In recent decades, freight transportation systems have been developed rapidly. This development leads to using various policies to enhance system utilization. The studies show that an optimized policy related to space allocation benefits the shareholders in freight transportations. The objective of space allocation problems is to find the best arrangement of cargos in warehouse cells to meet the problem aims. In this paper, inspired by the Office Space Allocation problem, we developed a novel model to minimize the handling costs and to maximize available spaces for the next arriving cargo. We first formulate the optimization model and discuss various constraints. We then present an approach to solve the proposed model. Lastly, we analyze a numerical example derived from the data of Port of Beaumont to illustrate the efficiency of the model.

### 1. Introduction

The rapid growth of commercial activities all around the world has been increasing the overseas trades, which arises the critical rule of transshipment systems. Ports, as the joint connection between seaways and land transport, have a significant influence on the performance of the system. Thus, any improvement on the ports to save time and money and to increase the utilization of space and equipment can subsequently promote the whole system efficiency [1]. In the maritime industry, as one of the most interesting topics in transportation planning, many types of research have been conducted in various aspects among which we can refer to warehouse management and container terminals [2]. In cargo and freight transportations the rule of Ports is highlighted to transfer goods from ship to the rails or road modes of transportation, and vice versa. These cargos are classified into two main groups: a substantial amount of commodities, like crude oil and bulk grain, and goods that are packed into standards sizes, like containers and pulps [3]. Being in

standard size makes it possible to present mathematical models for handling the packed cargo. However, in container terminals, stacking is one of the most important problems which directly would affect the performance of the terminals' operators. Because they put containers on top of other containers, while the containers in a column do not have the same departure date [4].

Thanks to machinery and equipment investment, freight transportation and cargo shipment systems have been developing rapidly. Such development highlights the need for port and inventory management to enhance system utilization. Along with the development of cargo and freight transportations, the management of warehouses has become more complex. To deal with such complexity, many types of research were conducted using optimization methods. Software companies are trying to utilize, and also simplify, these methods to provide their users with functional tools. The best tools for warehouse management should consider the user considerations and expectations in the optimization model as much as they could in the shortest possible time.

\* Corresponding Author: M. J. Kang  
E-mail address: [mjafarikang@lamar.com](mailto:mjafarikang@lamar.com)

Received: 18 November 2021; Revised: 13 December 2021; Accepted: 26 December 2021  
<https://doi.org/10.52547/crpase.7.4.2424>

Academic Editor: **Mohammad Yazdi**

Please cite this article as: M. J. Kang, P. Mobtahej, A. Sedaghat, M. Hamidi, A Soft Optimization Model to Solve Space Allocation Problems In Breakbulk Terminals, Computational Research Progress in Applied Science & Engineering, CRPASE: Transactions of Industrial Engineering 7 (2021) 1–7, Article ID: 2424.

As the warehouse status is changing dynamically, the processing time is the most important feature of a tool [5]. Warehouse management has a key role in any organization, especially in the seaport. The more successful strategy we use in a warehouse, the less money we lose, and the more cargo we can handle in the same period of time [6].

In this study, inspired by Office Space Allocation (OSA) problems, we developed an optimization model to minimize handling costs and to maximize available spaces for next arriving cargos. Space allocation problems find the best arrangement of cargos in warehouse cells to meet the problem objectives. Many types of research utilized optimization methods to address the problems. Handling costs refer to the total cost of transporting cargo to cells of warehouse lots and moving them from the cells to corresponding departure modes. The second part is going to increase the available spaces for storing the next arriving cargos by assigning the arriving cargos or reassigning the current cargos with the same or near departure date adjacent to each other. The problem constraints were defined as hard constraints, which must be met all the time, and soft constraints, which can be satisfied or otherwise come up with penalties. Finally, we developed a mathematical optimization model to simultaneously decide on cargos allocation and reallocation to have both minimum handling cost and maximum available space.

## 2. Literature Review

The space allocation problem seeks to assign several entities onto some specific rooms with specific attributes in such a way that both the space underuse and overuse are minimized. Apart from usage consideration, there are other requirements that are related to entities' expectations. Such requirements can be considered as model constraints, known as hard constraints. However, they can be defined as soft constraints which are entered into the objective function [7]. The OSA problems have to do with the multiple knapsack problems and general assignment problems [8], and they are classified into two main groups: the first one is to reassign all entities to the whole rooms considering the entities connections and the space constraints; While the second class is going to allocate the new entities to the available rooms to reduce the disruption between entities. It is obvious that the models of the second class need more constraints, which makes them more difficult [9]. In the present study, we consider the first one to simplify the problem.

Ulker et al applied an integer programming model to solve the Office Space Allocation Problem. The defined two optimization models for both hard and soft constraints. They used CPLEX to solve these models on sample data. By comparing the results of the models, they concluded that considering soft constraints instead of hard ones reduce the problem difficulty [7]. Afterward, in 2011, they carried out another study on OSA problems. In this study, they defined the problem as a 0/1 integer programming model and used Groubi as an (ILP) solver. Their objective function was to minimize the space usage and the summation of a series of penalties. The space usage was the maximum of space underuse and two times the space overuse. They also considered the penalty of not allocation, being in the same room, not sharing a room, being adjacent, and being away

from each other [8]. Pereira et al conducted their research on the OSA problem to maximize the synergies between employees. They defined synergy as a formula of the weighted distance between different employees in a building. They used heuristic algorithms including Tabu search and Greedy, and finally, the result highlighted the capability of Tabu search [9].

Yang et al proposed a mathematical model using the analysis of picking strategy of the warehouse delivery operation. They showed that the model can mitigate the cost of the logistics and raise the revenue by decreasing the cost of good handling and storage and the time lost [10]. Fumi et al developed a storage location assignment problem (SLAP) to minimize the overall warehouse space using a mathematical optimization approach by considering a dedicated storage policy [11]. For minimizing the reshuffles that may occur in the time of stacking of arrived containers in a terminal yard, a decision tree heuristic approach is implemented which utilizes the optimal solution of a stochastic programming model. With this approach, two shared and dedicated stacking policies were compared. The result showed shared-stacking policy has better performance [12]. Navazi et al used a mathematical model to optimize a sustainable multi-objective problem using two meta-heuristic algorithms considering circular economy [13]. Three algorithms including Ordinary least squares, M-estimation, and Tau-estimation methods presented to minimize the loss function of the multiple linear regression model [14], [15]. A distributional robust approach developed under demand uncertainty, and they showed that the delivery lead time is lesser compared to the deterministic environment [16].

While mentioned studies in OSA optimization problems approaches has been considered for methods on space allocation and utilization, in the other hand, similar relevant studies have been considered which could be a baseline for model comparison for defining a novel framework [17] for creating the best optimization model employed in this research. For instance, in Zo et al., a hybrid multi-objective genetic algorithm theory has been employed for a warehouse layout problem in optimization. Practical optimization principles would include first in and first out, which would continue to the end. This proposed optimization model in the mentioned paper is adaptable to other similar challenges and problems [18]. In another relevant study, static stochastic programming model of cargo space allocation for air cargo agents has been defined, and the optimization model has been solved with a robust optimization method, which could help cargo agents in the efficient and profitable allocation of cargo space to airlines which lastly would result in maximizing profits of the optimization model [19]. In another study, to minimize the Automated Parking System (APS) operation time within the space and operational constraints, a mathematical model was designed based on a special genetic algorithm in which feasible allocations are represented by chromosomes, with the retainment process ensuring its viability under crossover or mutation. Lastly, the proposed approach is used to compare with a traditional greedy method by running augmented experiments on a set of numerical problems examples for baseline comparisons [20]. Another study regarding partition-based space

management is proposed for optimizing the use of space on fixed-sized flash pages to store variable-sized values. Based on the achieved results, such method is beneficial for not only achieving optimal storage space utilization, but it could also trade a little space utilization for better performance at last [21]. Finally, in available open-source and modeling software which could be used as a beneficial model [22], using statistical and traditional machine learning models are employed for minimization space functions for the optimizations approach [23].

### 3. The Optimization Model

As we stated in the previous sections, there are many considerations and requirements a manager should know to store units and cargo in a warehouse. Some of them are easy to follow, like minimizing handling costs or weight considerations in stacking. However, some considerations increase the problem complexity because of the constraints they bring into the model. To deal with this model, it is highly recommended to relax constraints or penalize the objective function rather than hard constraints. Thus, in this section, we will go through the model with hard constraints, and afterward, we build the final model by replacing the hard constraints with penalties. The objective of this model is to minimize handling costs, and along with that, to maximize the available space for the next arrival cargos. The model requirements and considerations are listed as follows:

- to assign all cargos to the warehouse cells
- not to assign cargos to occupied cells
- to fully use rows or otherwise leave them empty
- to assign the units with the same departure date to the same rows
- to assign the units with the same departure date to the adjacent rows
- not to stack heavier cargos on lighter ones
- not to stack cargos with a later departure date on cargo with a sooner departure date.

Consider  $I$  as a set of arrival cargos which are supposed to be assigned to warehouse cells. Let  $R$  be the number of rows in a warehouse, and consider each row has  $C$  columns where we can put units on each other in  $S$  stack high. Thus, the decision variable of the model is defined as a binary matrix of  $\chi_{I \times R \times C \times S}$  where each  $\chi_{i,rcs}$  is equal 1 if the cargo  $i$  is allocated to row  $r$  in columns  $c$  at stack  $s$ , otherwise, it is zero.

(1)

$$\chi_{i,rcs} = \begin{cases} 1 & \text{if cargo } i \text{ is assigned to row } r \text{ in columns } c \text{ at stack } s \\ 0 & \text{Otherwise} \end{cases}$$

We defined handling costs as loading costs, for loading the unit to row  $r$  in the warehouse ( $Cost_r^{load}$ ), and unloading costs, for unloading it from the row  $r$  the warehouse ( $Cost_r^{unload}$ ). Thus, the objective function for the hard model is presented as follow:

$$\text{Minimize: } Z = \sum_{i \in I} \sum_{r \in R} \sum_{c \in C} \sum_{s \in S} \chi_{i,rcs} \times (Cost_r^{load} + Cost_r^{unload}) \quad (2)$$

Which is subjected to the following constraints:

- *All allocation:*

As it is presented in Eq. (3), this constraint controls if all cargos are assigned.

$$\sum_{r \in R} \sum_{c \in C} \sum_{s \in S} \chi_{i,rcs} = 1 \quad \forall i \in I \quad (3)$$

- *Stack feasibility:*

It states that units can be stacked two or more as long as the stacks are occupied.

$$\sum_{i \in I} \chi_{i,rc(s-1)} \geq \sum_{i \in I} \chi_{i,rcs} \quad \forall r \in R \quad c \in C \quad s \in S \quad (4)$$

- *Not to overuse:*

It prevents the model from assigning two or more units to a cell.

$$\sum_{i \in I} \chi_{i,rcs} \leq 1 \quad \forall r \in R \quad c \in C \quad s \in S \quad (5)$$

- *Date matching in rows*

The rule of this constraint is to assign the units with the same departure date to same rows.

$$\chi_{i_1,rcs} + \sum_{c' \in C} \sum_{s' \in S} \chi_{i_2,r'c's'} \leq 1 \quad \forall \text{date}_1 \neq \text{date}_2 \quad r \in R \quad c \in C \quad s \in S \quad (6)$$

- *Date matching in adjacency*

Its application is similar to the date matching constraint in rows, but it tries to assign the units with the same departure date not only to same rows but to the adjacent rows as well. Taking  $N_r$  as the set of rows which are adjacent to  $r$ , this constraint can be defined as below:

$$\chi_{i_1,rcs} \leq \sum_{r' \in N_r} \sum_{c' \in C} \sum_{s' \in S} \chi_{i_2,r'c's'} \leq 1 \quad \forall \text{date}_1 = \text{date}_2 \quad r \in R \quad c \in C \quad s \in S \quad (7)$$

- *Weight priority in stacking*

In stacking units on each other, weight consideration is crucially important. Especially, when there is no shelf, and we must put a unit on top of other units. To consider this requirement, we define the weight priority in stacking.

$$\sum_{i \in I} \chi_{i,rc(s-1)} \cdot \text{weight}_i \geq \sum_{i \in I} \chi_{i,rcs} \cdot \text{weight}_i \quad r \in R \quad c \in C \quad s \in S \quad (8)$$

- *Date priority in stacking*

Aside from weight requirements, the other consideration in stacking a unit on top of other units is the departure date of each unit. Although the expectation of warehouse managers is to put units with the same departure date close to each other, it is not all the time possible. In such case, the

managers practically put units with earlier departure date on the units with later departure date in stacks or in front of them in columns. Thus, we define the date priority constraint in stacking and in columns as two constraints presented in Eq. (9) and (10).

$$\sum_{i \in I} \chi_{i,r(c-1)s} \cdot date_i \geq \sum_{i \in I} \chi_{i,r cs} \cdot date_i \quad r \in R \quad c \in C \quad s \in S \quad (9)$$

$$\sum_{i \in I} \chi_{i,r c(s-1)} \cdot date_i \geq \sum_{i \in I} \chi_{i,r cs} \cdot date_i \quad r \in R \quad c \in C \quad s \in S \quad (10)$$

Although we could model the problem with hard constraints, the objective function is not oriented to maximize the available space for next cargos. It is worth mentioning that the constraints for date priority do not affect the results because we define hard constraints to control date matching in rows and their neighborhoods. Thus, the model does not manage a warehouse in an efficient manner, and moreover, the model may not have a feasible solution all the time. To solve this problem, we upgrade the model into a soft model which is more flexible in solving different problems with different consideration and requirements. The objective function is defined as presented in Eq. (11). Here,  $\omega_p$  refers to the penalty of  $p$  whose weight is denoted by  $w_p$ . So, the objective function seeks to minimize both the handling costs and the summation of the product of the penalties and their weights.

$$\begin{aligned} \text{minimize : } Z = & \sum_{i \in I} \sum_{r \in R} \sum_{c \in C} \sum_{s \in S} \chi_{i,r cs} \\ & \times (Cost_r^{load} + Cost_r^{unload}) \\ & + \sum_{p \in P} w_p \times \omega_p \end{aligned} \quad (11)$$

To change a hard constraint to the soft one, we define  $\delta_i^1$  as a Boolean variable which is equal to 1 whenever the constraint 1 for each  $i$  is not fulfilled, otherwise  $\delta_i^1 = 0$ .

$$\delta_i^p = \begin{cases} 1 & \text{if constraint } p \text{ for } i \text{ is violated} \\ 0 & \text{Otherwise} \end{cases} \quad (12)$$

$$\omega_p = \sum_i \delta_i^p \quad (13)$$

Not all the hard constraints we defined previously need to be converted to soft constraints. So, in the new model, the constraints controlling the logic of a solution are considered as hard constraints. These constraints are “*stack feasibility*” and “*avoid rewriting*”. Apart from these, the other constraints are redefined as soft constraints.

### I. Hard constraints:

*Avoid rewriting*

$$\sum_{r \in R} \sum_{c \in C} \sum_{s \in S} \chi_{i,r cs} \leq 1 \quad \forall i \in I \quad (14)$$

*Stack feasibility*

$$\sum_{i \in I} \chi_{i,r c(s-1)} \geq \sum_{i \in I} \chi_{i,r cs} \quad \forall r \in R \quad c \in C \quad s \in S \quad (15)$$

### II. Soft constraints:

*Not allocation penalty:*

$$\begin{aligned} \sum_{r \in R} \sum_{c \in C} \sum_{s \in S} \chi_{i,r cs} &= 1 - \delta_i^{notAllocation} \quad \forall i \in I \\ \omega_1 &= \sum_{i \in I} \delta_i^{notAllocation} \end{aligned} \quad (16)$$

*Overload penalty:*

$$\begin{aligned} \sum_{i \in I} \sum_{c \in C} \sum_{s \in S} \chi_{i,r cs} &\leq 1 + M \cdot \delta_r^{overLoad} \quad \forall r \in R \\ \omega_2 &= \sum_{r \in R} \delta_r^{overLoad} \end{aligned} \quad (17)$$

*Optimal use reward:*

$$\begin{aligned} \delta_r^{fullyuse} &\leq \frac{\sum_{c \in C} \sum_{s \in S} \sum_{i \in I} \chi_{i,r cs}}{|S| \cdot |C|} \\ &\leq 2 - \delta_r^{fullyuse} \quad \forall r \in R \\ \omega_3 &= \sum_{r \in R} \delta_r^{fullyuse} \end{aligned} \quad (18)$$

*Empty rows reward:*

$$\begin{aligned} \delta_r^{fullyempty} &\leq 1 - \frac{\sum_{c \in C} \sum_{s \in S} \sum_{i \in I} \chi_{i,r cs}}{|S| \cdot |C|} \\ &\leq 1 - \varepsilon + \varepsilon \cdot \delta_r^{fullyempty} \\ &\quad \forall r \in R \\ \omega_4 &= \sum_{r \in R} \delta_r^{fullyempty} \end{aligned} \quad (19)$$

*Date Mismatching in rows:*

$$\begin{aligned} (1 + \varepsilon) \delta_r^{dateMatchInRow} &\leq \sum_{c \in C} \sum_{s \in S} \chi_{i_1, r cs} \\ &+ \sum_{c' \in C} \sum_{s' \in S} \chi_{i_2, r c' s'} \\ &\leq 1 + \delta_r^{dateMatchInRow} \end{aligned} \quad (20)$$

$\forall date_1 \neq date_2 \quad r \in R$

$$\omega_5 = \sum_{r \in R} \delta_r^{dateMatchInRow}$$

*Date Mismatching in Adjacent rows:*

$$(21)$$

$$(1 + \varepsilon)\delta_{rr'}^{dateMissMatch} \leq \sum_{c \in C} \sum_{s \in S} \chi_{i_1,rcs} + \sum_{c' \in C} \sum_{s' \in S} \chi_{i_2,r'c's'} \leq 1 + \delta_{rr'}^{dateMissMatch} - \varepsilon$$

$\forall \text{ date}_1 \neq \text{date}_2 \quad r \in R, r' \in N_r$

$$\omega_6 = \sum_{r \in R} \sum_{r' \in N_r} \delta_{rr'}^{dateMissMatch}$$

Weight Priority in stacking:

$$\chi_{i,rc(s-1)} + \sum_{i' \in I} \chi_{i',rcs} \leq 1$$

$$+ M \cdot \delta_r^{weightMissMatch} \quad \forall \text{ weight}_i < \text{weight}_{i'} \quad r \in R \quad c \in C \quad s \in S$$

$$\omega_7 = \sum_{r \in R} \delta_r^{weightMissMatch}$$

Date Priority in stacking:

$$\omega_8 = \sum_{r \in R} \left( \sum_{c \in C} \sum_{s \in S} \sum_{i \in I} \chi_{i,rcs} \cdot \text{date}_i \times \text{datePriority}P_{cs} \right) \quad (23)$$

We considered  $M$ , as a positive big number, (=1000) and  $\varepsilon$ , as positive very small number, (= 0.0001). To enter the date priority into our model, we define a priority matrix for each row with  $c$  columns and  $s$  stacks (datePriority $P_{cs}$ ). In the next section we will go through the model and apply it on sample of data.

#### 4. Experiment and Results

We used a set of 60 arrival cargos at three different weight, which are supposed to be sent on 4 different departure dates. We also took a warehouse with 10 rows in two aisles, where we can store units in three columns and three stacks high. There is one aisle in the middle of the warehouse for loading and unloading units, there is also one enter door and one exit door at both ends of the aisle. Figure 1 presents the layout map of the sample warehouse. Accordingly, the overall capacity of this warehouse is 90 units.

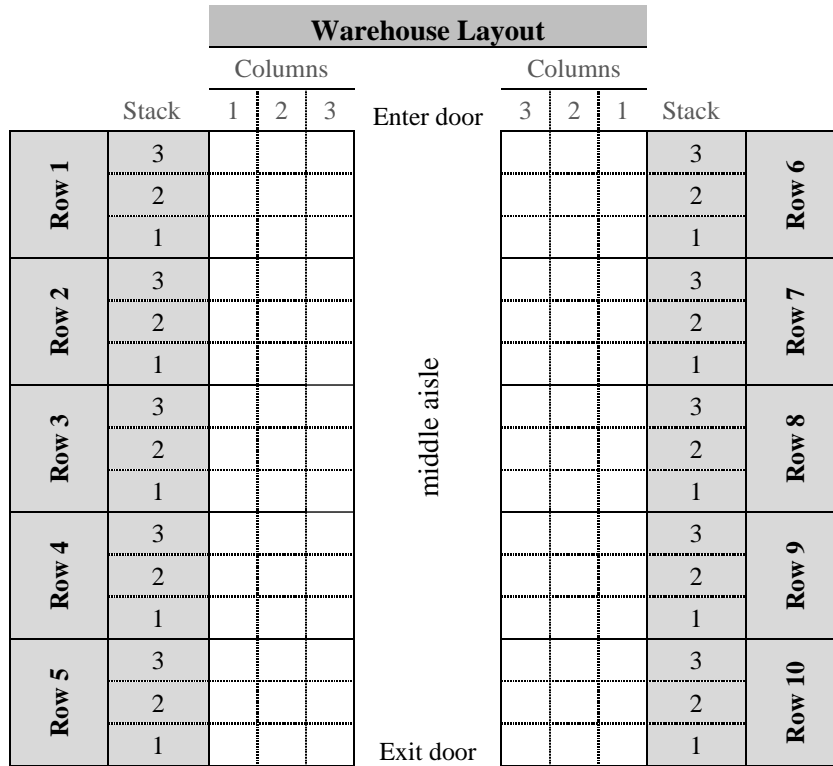


Figure 1. The layout of sample warehouse

To solve the optimization model, we used IP solver of CPLEX 12.8.0. Additionally, we consider the weighted for each penalty as: *Not allocation penalty* = 50, *Optimal use reward* = 10, *Overuse penalty* = 100, *Empty rows reward* = 20, *Date Matching in rows* = 10, *Date Matching in shed* = 5, and *Weight Priority in stacking* = 10. And finally, for *Date Priority in stacking* constraint, we defined the priority matrix of datePriority $P_{cs}$  as:

$$\text{datePriority}P_{cs} = \begin{bmatrix} 0.067 & 0.133 & 0.200 \\ 0.044 & 0.111 & 0.178 \\ 0.022 & 0.089 & 0.156 \end{bmatrix} \begin{matrix} \text{stack 1} \\ \text{stack 2} \\ \text{stack 3} \end{matrix} \quad (24)$$

After running the model, we achieved the final solution, which is the exact optimal point of our problem. All cargos were assigned to the rows of the warehouse. Figure 2 shows

the optimal assignment plan for 60 arrival cargos. According to this plan, the model allocated cargos which are going to be sent on April 10 to rows 1, 2, and 6, which are close to each other. The same pattern is followed in allocation of cargos for April 15 and April 20. The model assigned cargos

with the same departure date on the same rows, except Row 2. Additionally, in row 2 and row 7, the date priority consideration is satisfied. Aside from this, Figure 3 represents the assigned cargos weight and shows that there is no heavy unit on top of a lighter unit.

Warehouse Layout											
		Columns					Columns				
		Stack	1	2	3	Enter door	3	2	1	Stack	
Row 1	3		April 10	April 10	April 10	middle aisle	April 10	April 10	April 10	3	Row 6
	2		April 10	April 10	April 10		April 10	April 10	April 10	2	
	1		April 10	April 10	April 10		April 10	April 10	April 10	1	
Row 2	3		April 20	April 20	April 10			April 20	April 20	3	Row 7
	2		April 20	April 20	April 10			April 20	April 20	2	
	1		April 20	April 20	April 10			April 20	April 20	1	
Row 3	3		April 15	April 15	April 15			April 15	April 15	3	Row 8
	2		April 15	April 15	April 15			April 15	April 15	2	
	1		April 15	April 15	April 15			April 15	April 15	1	
Row 4	3		April 5	April 5	April 5					3	Row 9
	2		April 5	April 5	April 5				2		
	1		April 5	April 5	April 5				1		
Row 5	3								3	Row 10	
	2								2		
	1								1		
					Exit door						

Figure 2. the optimal assignment of 60 units to the warehouse layout classified based on the departure date

Warehouse Layout												
		Columns					Columns					
		Stack	1	2	3	Enter door	3	2	1	Stack		
Row 1	3		1.5	1	1	middle aisle	1	1	1	3	Row 6	
	2		1.5	1	1		1	1	1	2		
	1		1.5	1.5	1.5		1.5	1	1	1		
Row 2	3		1	2	1			2	1	3	Row 7	
	2		2	2	1			2	2	2		
	1		2	2	1			2	2	1		
Row 3	3		1.5	1.5	1.5			1.5	1.5	1.5	3	Row 8
	2		1.5	1.5	1.5			1.5	2	1.5	2	
	1		2	2	1.5			1.5	2	2	1	
Row 4	3		2	1	1					3	Row 9	
	2		2	1	1				2			
	1		2	1	1				1			
Row 5	3								3	Row 10		
	2								2			
	1								1			
					Exit door							

Figure 3. the optimal assignment of 60 units to the warehouse layout classified based on the units weight

## 5. Conclusions

In the present study, by comparing similar related studies, we tried to build an optimization model using integer programming models to solve space allocation problems in warehouses. Then, we defined penalties to relax hard constraints and replace them with soft constraints. The model relaxation not only reduces the problem complexity, but it makes it possible to include more requirements and considerations into the model as well. The final solution on sample data showed that all the considerations are satisfied, and moreover, it produced more adjacent available space for both the present condition and the future condition. While the model with hard constraints was practically unsolvable for the same data. Although the result seems desirable, it is noticeable that the IP model may not be an efficient method when the problem size increases. Aside from this, in real-world problems in warehouse management, there are other considerations and requirements which need to be included in the model, for example, cargo type matching in rows or their neighborhoods. The model presented in this study proofed its capabilities to be flexible for adding any considerations and requirements into the objective function. However, our suggestion for future works is to utilize either heuristic or meta-heuristic algorithms to solve the model with more considerations and for real-world samples.

## References

- [1] B. Rahimikelarijani, A. Abedi, M. Hamidi, J. Cho, and E. Stromberg, Optimal ship channel closure scheduling for a bridge construction, 67th Annual Conference and Expo of the Institute of Industrial Engineers (2017) 530–537.
- [2] B. Rahimikelarijani, A. Abedi, M. Hamidi, and J. Cho, Simulation modeling of Houston Ship Channel vessel traffic for optimal closure scheduling, Simulation Modelling Practice and Theory 80 (2018) 89–103.
- [3] H. Rashidi and E. P. K. Tsang, Novel constraints satisfaction models for optimization problems in container terminals, Applied Mathematical Modelling 37 (2013) 3601–3634.
- [4] B. Borgman, E. van Asperen, and R. Dekker, Online rules for container stacking, OR Spectrum 32 (2010) 687–716.
- [5] J. Karasek, An Overview of Warehouse Optimization, International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems 2 (2013).
- [6] N. Jakobsson and J. Rudälv, Optimization Of The Product Allocation In Warehouses, Umea University (2018).
- [7] Ö. Ülker and D. Landa-Silva, A 0/1 integer programming model for the office space allocation problem, Electronic Notes in Discrete Mathematics 36 (2010) 575–582.
- [8] Ö. Ülker and D. Landa-Silva, Designing difficult office space allocation problem instances with mathematical programming, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6630 (2011) 280–291.
- [9] R. Pereira, K. Cummiskey, and R. Kincaid, Office space allocation optimization, 2010 IEEE Systems and Information Engineering Design Symposium, SIEDS10, (2010) 112–117.
- [10] D. Yang, Y. Wu, and W. Ma, Optimization of storage location assignment in automated warehouse, Microprocessors and Microsystems 80 (2021).
- [11] A. Fumi, L. Scarabotti, and M.M. Schiraldi, Minimizing Warehouse Space with a Dedicated Storage Policy, International Journal of Engineering Business Management 5, (2013).
- [12] A. H. Gharehgozli, Y. Yu, R. de Koster, and J. T. Udding, A decision-tree stacking heuristic minimising the expected number of reshuffles at a container terminal, International Journal of Production Research 52 (2014) 2592–2611.
- [13] F. Navazi, A. Sedaghat, and R. Tavakkoli-Moghaddam, A New Sustainable Location-Routing Problem with Simultaneous Pickup and Delivery by Two-Compartment Vehicles for a Perishable Product Considering Circular Economy, IFAC-PapersOnLine 52 (2019) 790–795.
- [14] M. M. Ahmadi, H. Shahriari, and Y. Samimi, A novel robust control chart for monitoring multiple linear profiles in phase II, Communications in Statistics - Simulation and Computation (2020) 1–12.
- [15] S. Mehri, M. M. Ahmadi, H. Shahriari, and A. Aghaie, Robust process capability indices for multiple linear profiles, Quality and Reliability Engineering International 37 (2021) 3568–3579.
- [16] S. Momen and S. A. Torabi, Omni-channel retailing: A data-driven distributionally robust approach for integrated fulfillment services under competition with traditional and online retailers, Computers & Industrial Engineering 157 (2021) 107353.
- [17] P. Mobtahej, Evaluation of Innovation Maturity Model (IMM) Framework on Assessment of Software Development Projects, Computational Research Progress in Applied Science & Engineering 06 (2020) 251–258.
- [18] Qiaohong Zu, Mengmeng Cao, Fang Guo, and Yeqing Mu, Slotting optimization of warehouse based on hybrid genetic algorithm, in 2011 6th International Conference on Pervasive Computing and Applications, (2011) 19–21.
- [19] C. Zhang, R. Luo, and Z. Chen, An optimization model of cargo space allocation for air cargo agent, in 2010 7th International Conference on Service Systems and Service Management (2010) 1–5.
- [20] M. Zou, Q. Wang, and S. Liu, Optimization of Parking Space Allocation for Automated Parking System of Paternoster Type by Genetic Algorithm, Chinese Control And Decision Conference (CCDC), (2019) 3834–3838.
- [21] Y.-T. Chen, M.-C. Yang, Y.-H. Chang, T.-Y. Chen, H.-W. Wei, and W.-K. Shih, KVFTL: Optimization of storage space utilization for key-value-specific flash storage devices, in 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), (2017) 584–590.
- [22] P. Mobtahej, The Motivation behind investment in development of open source software projects, Computational Research Progress in Applied Science & Engineering, Computational Research Progress in Applied Science & Engineering 07 (2021) 1–4.
- [23] P. Mobtahej, X. Zhang, M. Hamidi, and J. Zhang, Deep Learning-based Anomaly Detection for Compressors Using Audio Data, in 2021 Annual Reliability and Maintainability Symposium (RAMS), (2021) 1–7. DOI: 10.1109/RAMS48097.2021.9605720.