Multi-objective based Simplified Swarm Optimization for Container Loading Optimization with Practical Constraints

Linh-Hoang Truong, Chen-Fu Chien*

***Abstract*— The container loading problem (CLP) plays a crucial role in various industries and commercial applications, especially in logistics and supply chain management. Although there are many studies that have been proposed to deal with CLP, the combination of practical objectives and constraints has not yet received much attention. This study aims to develop multi-objective simplified swarm optimization (MOSSO) algorithm-based UNISON framework for three dimensions single container loading problem. Space utilization and weight distribution are treated as the objectives while satisfying common CLP constraints. In addition, the merging space algorithm is applied to merge small fragment spaces generated after loaded parcels into the container to a larger space to load more parcels. Numerical experiments are designed to validate and compare the results of the proposed MOSSO with existing studies by using a public benchmark dataset. The results have shown the practical effectiveness of the proposed approach improved average space utilization while satisfying center balance and other crucial constraints. Indeed, this study could be used as a digital support system to assist decision-makers and avoid adjusting the patterns to save time.**

***Keywords*:** simplified swarm optimization; container loading problem; digital support system; logistics and transportation.

I. INTRODUCTION

Logistics and transportation are key elements in today's evolving economy where deliver speed and cost are essential criteria. An efficient logistics strategy could help companies reduce warehousing and transportation expenses. There are many benefits of container transportation such as combination of variety of goods, securities, multiple destinations, and large capacity. Thus, an effective container loading plays a potential role to dramatically decrease logistics and transportation costs in the entire transportation system. In addition to transportation costs, emissions of the logistics industry also have a significant impact on the environment. Therefore, enhancing logistics processes is ever more crucial in order to provide better service quality and handle additional goods simultaneously lowering costs and reducing emissions.

A set of items is chosen from warehouses or other sources for each shipment to load into the container and transport to the designated locations. However, selecting rectangular-shaped and loading into containers while maximizing the space utilization with respect to practical constraints are complex and time-consuming. An inefficient loading process often happens a set of parcels could not load into a container while the total volume of parcels is smaller than the capacity of the container. Thus, leading to increase cost and a decrease in the competitiveness of the business. Indeed, the logistics companies gain significant benefits from an effective loading method, which boosts income per shipment even while enhancing safety and preventing damage to commodities inside containers. Furthermore, reducing fuel consumption helps logistics companies lowering the carbon footprint.

Focusing on realistic needs, this study aims to establish a digital support system to support logistics companies in building an effective parcel loading solution for a container. The study develops an UNISON framework that integrates Simplified Swarm Optimization (SSO) technique to optimize the parcel loading solution which maximize the multiple objectives as space utilization and load balancing while satisfying practical constraints. The loading solution includes the location of rectangular parcels and the corresponding orientation which is loaded into containers. Moreover, the UNISON framework is employed to help decision-makers or straight forwarders effectively judge and find the best appropriate decision to clarify the objective. To evaluate the performance of the proposed method, an empirical study is conducted by using the benchmark data set provided by [1]. The empirical results have shown the improvement in space utilization and other crucial criteria as loaded balancing and loaded parcel ratio.

The remainder of the thesis is organized as follows. Chapter 2 explains related studies. Chapter 3 shows the key components of the proposed framework, whereas chapter 4 validates the framework by conducting an empirical study on a public dataset. This study contribution and detailing future work are summarized in Chapter 5.

II. LITERATURE REVIEW

A. Container Loading Problem Formulation

The container loading problem could be referred to three bin packing problem (3D-BPP) which select and load a set of items or parcels into a container with respect to the multiple constraints. Basically, the main goals are to maximize the space utilization and total value of loaded parcels. Given a container with dimensions W, L, H which stand for width, length and height of the container, and a set of parcels $P = \{p_1, p_2, \dots, p_n\}$. The objectives of CLP are expressed as shown in as shown in Equation 1:

$$\max \sum_{i=1}^{n} x_i v_i \quad \text{and} \quad \max \sum_{i=1}^{n} x_i s_i \tag{1}$$

Where v_i is the volume of the parcel b_i. s_i is the value or profit related to parcel p_i. x_i is the decision variable that is 1 if parcel p_i is selected, otherwise is 0. Each parcel has its own characteristics and rotate orientations, as shown in Figure 1. The choice of parcel orientation could cause the value of x_i because of the following some hard constraints:

- The parcel must be fitted inside the container.
- Loaded parcels must not be overlapped.
- The parcel must be support by other parcels below.

Linh-Hoang Truong and Chen-Fu Chien are with the Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Taiwan

- The weight of above parcels should not exceed the maximum stacking ability of below parcels.
- Total weight of parcels inside must not exceed the maximum allowed container weight.

If the parcel p_i meet mentioned constraints, then the value of $x_i = 1$, otherwise could not be selected to load into the container.

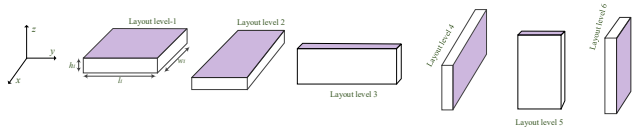


Figure 1. Illustration of the six orthogonal item orientations

B. *Placement Procedures*

There are three kinds of approaches in literatures to deal with the placement procedure of CLP including exact methods, heuristic, and meta-heuristic algorithms. While the third type is the most significant method received attention so far due to several reasons. First, the CLP has been proved as a NP-hard problem. Second, although there are existing studies in which mixed-integer linear programming (MILP) is used to formulate CLP, it remains difficult when consider practical constraints in the same form [2]. Third, due to the time limitation, logistic companies need solutions that generated quickly and effectively which are difficult to achieve by using exact methods [3].

Meta-heuristic algorithms are also popular in CLP in which the solution generated from the heuristic approach is modified to achieve better solutions. These methods run iteratively where the position, layout, or the packing sequence of parcels inside are modified, if the solution has a better fitness value which could be space utilization, the next optimal solution is updated. There are various proposed approaches tackled CLP by using meta-heuristic algorithms such as genetic algorithm [4], beam search approach [5], artificial bee colony algorithm [6], tabu search approach [7].

However, in order to produce packing plans that are appropriate for real-world problems, numerous additional constraints must be addressed in addition to geometric limitations. Much research has recently introduced realistic limitations to the fundamental issue, but much more effort is still needed.

C. *Simplified Swarm Optimization (SSO)*

Simplified Swarm Optimization (SSO) is a kind of meta-heuristics optimization technique that first proposed by Yeh [8]. SSO has showed as a very useful and efficient algorithm in optimization problems in term of both solution quality and updating mechanism. Some of studies could be mentioned as network reliability [9] and deep learning [10]. Owing to its simplicity and efficiency, this study develops a MOSSO to deal with the 3-D container loading problem.

In particular, the proposed SSO contains parameters used in initializing and updating process. Initializing parameters include number of generations (*num_gen*), and population size (*pop_size*). The particle index j at generation t has the form of $X_j^t = \left(x_{j,1}^t, x_{j,2}^t, \dots, x_{j,n}^t\right)$, pBest P_j^t is the j-th solution with the best fitness value which has achieved so far, and gBest G^t is the best solution among P_j^t at generation t, where $j = 1, 2, \dots, pop\_size$. For overall, the set of all particles are denoted as $X = \left(X_1, X_2, \dots X_{pop\_size}\right)$, set of pBest of all particles $P = \left(P_1, P_2, \dots P_{pop\_size}\right)$. Updating parameters include C_g, C_p and C_w which are probabilities of different updating mechanism of particle components in each generation. A random number, $\rho \in [0,1]$ is a probability of choosing updating mechanism, as shown in Equation 2:

$$x_{j,l}^{t+1} = \begin{cases} gBest^t, & \text{if } \rho \in [0, C_g] \\ pBest_j^t, & \text{if } \rho \in [C_g, C_p] \\ x_{j,l}^t, & \text{if } \rho \in [C_p, C_w] \\ \chi, & \text{if } \rho \in [C_w, 1] \end{cases} \quad (2)$$

Where j,l is the index of particle and component in a particle, respectively. $x_{j,l}$ result is $gBest^t$, $pBest_j$ the value of previous generation, and a random value with predefine range for condition 1 to 4, respectively. Note that, the cumulative sum of C_g, C_p and C_w should be 1.

III. RESEARCH FRAMEWORK

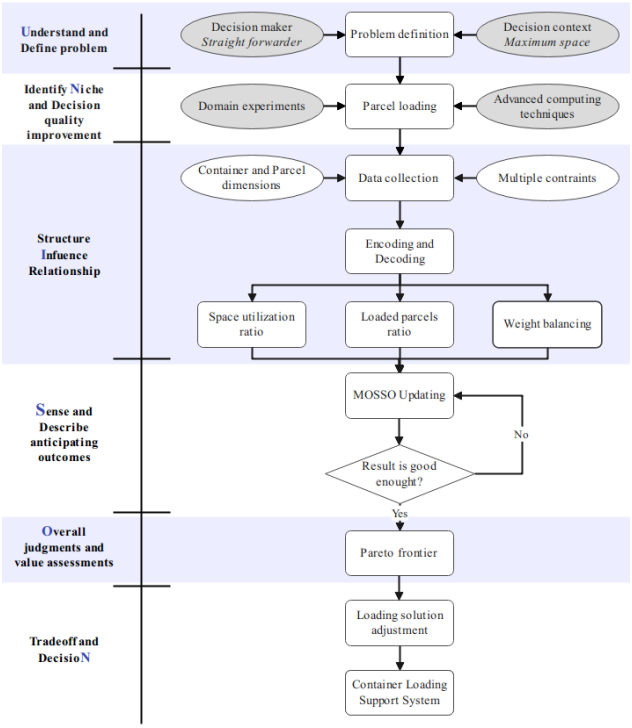


Figure 2. UNISON framework for CLP

This study proposes a MOSSO based UNISON framework to determine an appropriate loading sequence for CLP while satisfying practical constraints, as shown in Figure 2. The proposed framework is explained in six phases: (1) understand and define the problem; (2) identify the niche for decision quality improvement; (3) structure the objective hierarchy; (4) sense and describe expected outcomes; (5) perform overall judgment and value assessments; and (6) trade-off among the attributes and decide. Indeed, UNISON framework has been applied into various contexts as a systematic framework for enhancing solution quality, such as quality control [11], machine maintenance [12], time series forecasting [13].

A. *Understand and define the problem*

In the beginning, the problem definition and problem structuring are defined in which the decision-maker and decision context are related. In this study, the decision context is the process of loading several rectangular parcels into a certain container. The decision-maker is the

transportation manager or straight forwarder. To satisfy the customers while minimizing the operational costs, the logistics companies should transport as many packages as possible for each shipment. The main problem is that decision-makers often rely on experience without the support of calculation tools, which leads to inefficient shipments. Thus, the decision-maker needs an efficient and robust method to maximize parcels per-transportation and at the same time reduce emissions to the environment.

B. Identifies the niche for decision quality improvement

CLP problem has proven to be an NP-hard problem due to the difficulty trying to find the optimal loading solution [3]. In many cases, simply changing the orientation of a package could produce more remaining space inside the container, resulting in more loads that can be loaded. Considering the competitiveness in the marketplace, an efficient loading process in a certain amount of time becomes a crucial issue. Indeed, by combining a decision maker's experience with the integration of advanced techniques, logistics companies can make an intelligence decision.

C. Structure influence relationship

After knowing the problem and identifying the niches, related data and loading methodologies are investigated to solve the container loading problem.

1) Notations

Before the loading methodology is presented, there are some notations should be declared, as shown in TABLE 1.

TABLE 1. Notations

Notation	Definition
p_i	Parcel index i, $i = 1,2, \dots n$ where n is total number of parcels
W, L, H	The width, length, and height of the container.
M	The maximum loaded weight of the container
K	Number of parcel types
w_k, l_k, h_k	The dimension of parcel type k , $k = 1,2 \dots K$
m_k	Weight of box type k

2) Encoding

There are two components in the encoding solution which are loading sequence and corresponding layout of the parcels to load into the container. As shown in Figure 3, the first row is the loading sequence of each parcel by uniform random and are sorted follow the ascending order. For example, at the beginning, parcel 1 have the random loading order is 0.10 and is loaded first compared with loading orders of the remaining parcels. The second row is the encoding of the loading layout of each parcel. Since specific parcel has different possible layouts, for instance, parcel 1 could be rotate in 6 orientations but only 2 orientations for parcel 2. The total probability of choosing layout is 1 while each possible layout has equal probability, that is 1 / total possible layouts.

3) Decoding

The loading sequence and the corresponding layout level are initialized at the first generation. The decoding procedure loads all parcel follow the loading sequence with the corresponding layout.

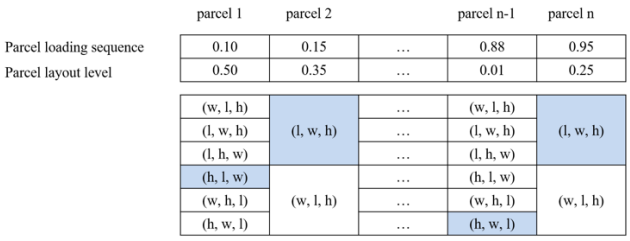


Figure 3. Loading sequence and the corresponding layout of the encoding

The detail of decoding procedure is listed in the ALGORITHM 1. At the beginning, if there is no parcel inside the container, then the first parcel is loaded as the position (0, 0, 0) or most bottom, left and container floor. Three corresponding spaces are front, right and upper space are updated based on the dimension of loaded parcel and chosen space, as shown in Figure 4. From the next generation, the algorithm finds fitted space for the parcel with priority of most left, bottom, and the back vertices of the container. Each time a parcel is loaded into the container, the remaining allowed weight of the container and 3 corresponding spaces are updated.

For the loaded layout, each parcel has l possible layouts, and the value of layout chromosome is between [0, 1] which is used to determine which layout is chosen. For instance, as shown in Figure 3, the layout of parcel 1 is (*h, l, w*) or (height, length, width) since its chromosome value is 0.5 which corresponding the 4-th layout.

During the loading process, the parcels are loaded using the following heuristic strategies:

- *Position*: the space with the minimum *x, y* and *z* coordinates is chosen for loading the parcel. After a parcel are loaded, the remaining space inside the container are updated as three-side spaces, as shown in Figure 4.
- *Orientation*: the face with the largest area is chosen first as the base to maximize support and stability.
- *Space*: the next parcels are then loaded into the space with the priority of upper, right and front space, respectively.

ALGORITHM 1. Decoding procedure

Input: Chromosome of parcel loading and its layout
Output: parcel loading solution

```
FOR parcel i = 1 to n:
  IF parcel i weight ≥ container remaining weight:
    Continue
  END IF
  IF container is empty:
    Parcel position = (0, 0, 0)
    Update container remaining allowed weight
    Update space j to front, right and upper spaces
  ELSE
    FOR space j in Spaces inside container:
      IF parcel i space fit space:
        Parcel position = space j position
        Update container remaining allowed weight
        Update space j to front, right and upper spaces
        Break
    Merging and ascending sort Spaces inside container
```

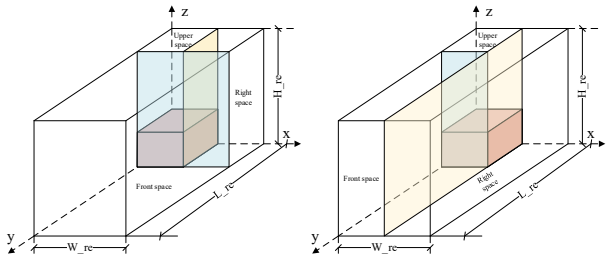


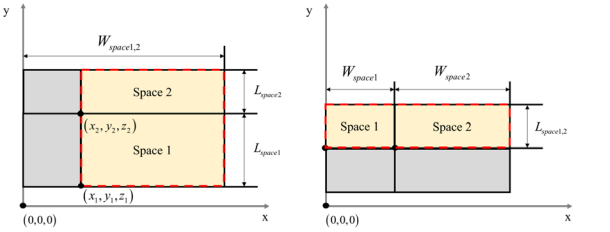
Figure 4. Partitioning layouts with full support from below for a remaining space.

4) Merging remaining space

When parcels are loaded into the container, they create three remaining spaces. The more parcels are loaded, the more remaining spaces are generated, which can be very small in base and volume and make difficult or impossible to load other parcels that cause of the result is low space utilization. Therefore, inspired by the study in [7], the merging remaining space algorithm is used to combine small remaining spaces, which are unlikely to load rows into larger remaining spaces.

Every time a parcel is loaded into the container, the side-spaces are generated, as shown in Figure 4. Then, the merging remaining space algorithm merge these spaces along x and y axis. Note that, all possible merging spaces must have the same z-coordinate and height:

- *Merging two adjacent spaces with the same length or width*: two remaining spaces are merged if they have the same x and equal width, or y coordinate and have the same length, as shown in Figure 5.
- *Merging two adjacent space that have different of length or width*: two remaining spaces are merged if they satisfy the Equation 3. The gray areas and red rectangular in the figure refer to the parcel that already loaded into the container and the merged space, respectively, as shown in Figure 6.



a) Two adjacent space of the same width b) Two adjacent space of the same length

Figure 5. Merging two adjacent spaces with the same length or width

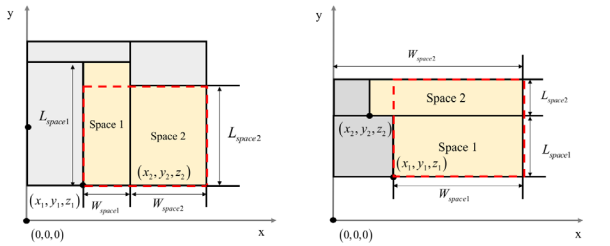


Figure 6. Merging two adjacent spaces with different lengths and widths

$$\begin{cases} y_{space1} & \geq y_{space2} \\ x_{space1} + w_{space1} & = x_{space2} \\ y_{space1} + l_{space1} & \geq y_{space2} + l_{space2} \end{cases} \quad (3)$$

$$\begin{cases} x_{space1} & \geq x_{space2} \\ y_{space1} + l_{space1} & = y_{space2} \\ x_{space1} + w_{space1} & \geq x_{space2} + w_{space2} \end{cases}$$

D. Sense and describe anticipating outcomes

This study extend the SSO algorithm to deal with multi objectives in CLP optimization.

1) Initialization

The proposed MOSSO first encodes the loading sequence of parcels in the form of two random variables, parcel loading sequence and its corresponding layout. The loading sequence is uniformly random while layout of the parcels is first selected as largest base for ensuring the stability. More detail, the range for random loading sequence is [0, 1] and the loading layout is $p \in [0, 1]$, which corresponding with the level of layout with largest base area.

2) Evaluate fitness function

After the initial solutions are generated, each solution in the population is evaluated by the fitness function. In this problem, two objectives which are space utilization, and weight balancing are taking into account. Space utilization is calculated by summary the volume of all parcels loaded into the container. Weight distribution is the distance of the gravity center (*GC*) of the full loaded container with the idea *GC*. Since the larger the value of two objectives in which the weight distribution could be reverse by *1 - distance*, the better the solution is.

3) Define the pBest P_i for X_i and identify the set of non-dominated solutions Q

i. Define *P*: Let $P_i = X_i, i = 1,2, \dots n$ at the beginning where *n* is size of the population.

ii. Define non-dominated solution $Q = \emptyset$. From the population, all solutions are sorted and assign to Q if it is non-dominate solution, $Q = (q_1, q_2, .. q_n)$ where q_n is the total non-dominated solution in Q.

iii. The sorting algorithm for sorting and creating Pareto frontier is presented in [11]. Basically, the algorithm returns the set of solution which is not dominated by any other solutions and creating a frontier, other solutions are removed. In order to maintain the diversity of the non-dominated solutions, the crowding distance of all solutions are also calculated. The solutions with minimum crowding distance which has no contribution to the convergence of the algorithm are eliminated [11].

4) Update the solution

Choose a non-dominated solution in Q for updating each solution X_i: gBest in SSO must be replaced by one of the solutions in the Pareto frontier solution set. In this study, random method is applied in which a random solution in Q is chosen as the gBest input for SSO updating each particle in the population.

5) Update pBest and Q

i. From the generation $t = 2$, if the particle fitness value of the particle dominates the previous one, then the particle *pBest* is updated by the current solution. However, in the situation where neither of them is dominated by the other, one is randomly selected to be the particle *pBest*.

ii. Create *new_Q* set of non-dominated solution: similar to the step 3.3 which is create new Pareto frontier by new *pBest*.

iii. Update the non-dominated solutions in Q: let $Q = Q \cup new\_Q$ and remove dominated solutions from Q.

6) Check the termination criterion

If the generation t < num_gen, let $t = t + 1$ and go back to step 4, otherwise halt and return the Pareto frontier set of solution.

E. Overall judgments and value assessments

1) Space utilization

Given a container with dimensions W, L, H, maximum weight support M are used to load a set of n parcels which vary in dimension w_i, l_i, h_i, and weight m_i respectively. The objective is producing a parcel loading sequence solution such that space utilization for a container is maximum. Denoted that the ratio of space utilization comparing between total volume of all parcels and the container, R, as shown in Equation 3.2:

$$\text{Space utilization (\%)} = \frac{\sum_{i=1}^{n} w_i l_i h_i}{W \times L \times H} \times 100 \qquad (4)$$

Where the numerator and denominator are denoted for the total volume of loaded parcels and the total volume of the container, respectively.

2) Weight distribution

Weight distribution in road transport concerns the distribution of the weight of the loaded vehicle among its axles. The center of gravity (CG) of the container should be lied close to the geometrical midpoint of the container floor. In more details, the ideal *CG* is at half-width, half-length and floor of the container. Therefore, the load balancing criteria, denoted as *LB*, is formulated as follow:

$$\text{Distribution} = (1\text{-dist}) \times 100 \qquad (5)$$

Where,

$$\text{dist} = \frac{1}{2} \times \left(\frac{|CG_x - W/2|}{W/2} + \frac{|CG_y - L/2|}{L/2} \right) \qquad (6)$$

Where,

$$CG_x = \frac{\sum_{i=1}^{n} \text{weight}_i \times (x_i + w_i/2)}{\sum_{i=1}^{n} \text{weight}_i}$$
$$CG_y = \frac{\sum_{i=1}^{n} \text{weight}_i \times (y_i + l_i/2)}{\sum_{i=1}^{n} \text{weight}_i} \qquad (7)$$

In Equation 3.6, CG_x, CG_y are the center of gravity of the loaded parcels along width, length of the container; i, n is the index of loaded parcel and the total number of loaded parcels in the container, respectively; x_i, y_i, are x-, y-coordinate values of the front-left-bottom vertex of loaded parcels.

F. Trade-off and decisions

Finally, proposed method returns parcel loading sequence that achieve highest criteria. Furthermore, the decision maker can modify the solution so that standards such as ease of loading and unloading, ease of delivery to be more satisfied. Therefore, the proposed method is not only offer solutions that meet the highest criteria but also provide many solutions to help decision makers choose the most suitable solution. The transportation process of the company could be enhanced, increase the satisfying of customers and thus the competitiveness of the company in the market share. The proposed decision support system for CLP is presented in Figure 7.

IV. EMPIRICAL STUDY

A. Research problem

An empirical study was conducted to evaluate the performance of the proposed framework for CLP. The proposed MOSSO begins with describing and defining the problem. To adapt the needs of customers and increase their position in market share, transportation companies are required to reduce the shipping time but maintain associated costs. In addition, a proper loading product sequence could reduce the complexity of the loading process and increase

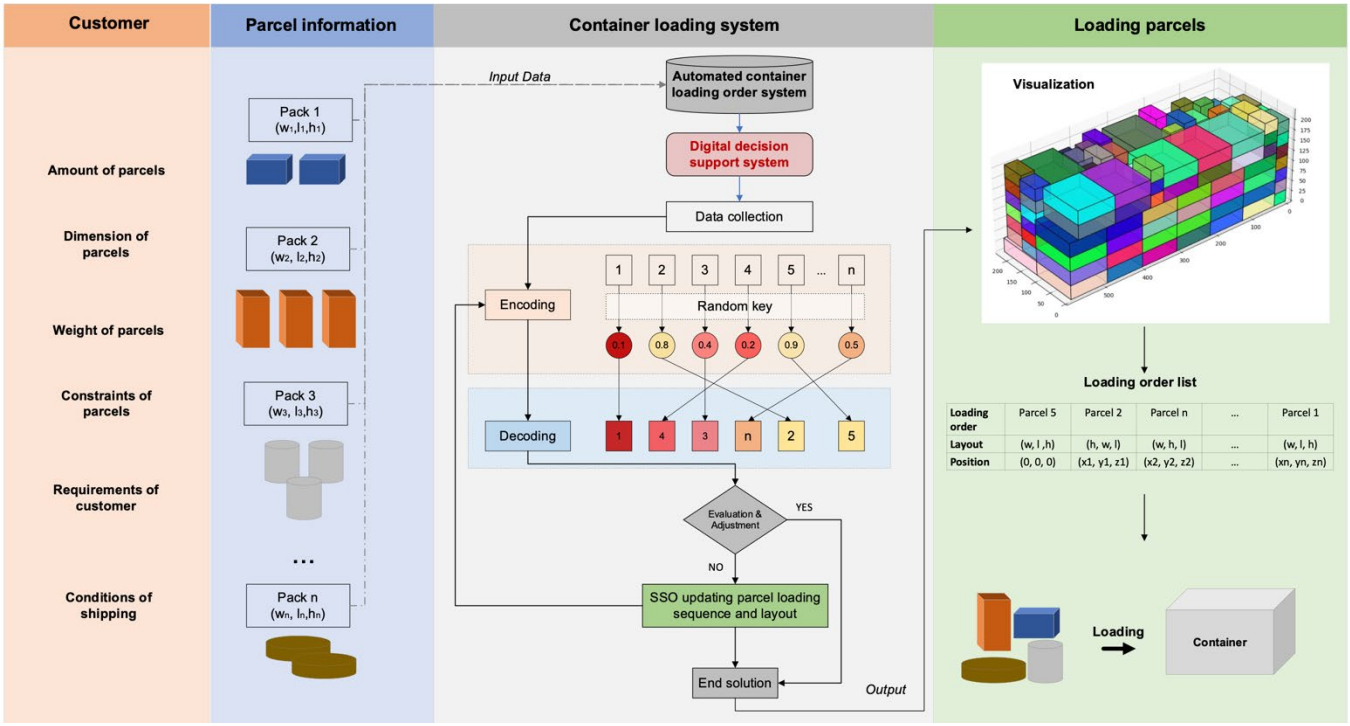


Figure 7. Flow chart of Digital Decision Support System for CLP

number of products for each shipment. Motivated by these gaps, this study develops an effective container loading support system for transportation companies by finding the loading sequence of products while satisfying practical constraints.

B. Benchmark dataset

To evaluate the performance of the proposed framework, a common benchmark dataset generated in [1] is used. The dataset contains 3 test classes in total, from pack1 to pack3, with 100 test instances each. However, due to the large number of test instances in each test class, this study chooses a part of all test instances as a test class sample. Since parcels within test class share common characteristics such as dimensions, then the small number of test instance could be used to evaluate the performance of the proposed method.

C. MOSSO for CLP

In this study, the algorithm is deployed on Windows 10 operating system using Python 3.10 programming language, and the matplotlib graphics library to simulate parcel loading solutions. Hardware used includes Intel i5-9400 CPU at 2.90GHz, 16GB RAM.

1) Parameter configurations

The proposed method is examined using parameter settings in TABLE 2.

TABLE 2. Parameter settings

Number of generations	50, 100, 200
Number of replications	10
Number of particles	20, 30, 50
C_g, C_p and C_w	(0.55, 0.75, 0.85)

2) Experiment results

At the beginning, the loading order of parcels follow its base area, from parcels with largest to smallest base area. This loading strategy ensure the maximum support and stability of the above parcels. After initialize loading solution, parcels are then loaded into the container following the corresponding loading sequence. The space utilization, loaded parcel ratio, loaded balancing are used to evaluate the performance of the first loading solution. From the next generation, the loading sequence and layout of each parcel are updated follow the updating mechanism of MOSSO algorithm.

a) Space merging

After the parcel is loaded into the container, three spaces are created, these are front, upper, and right space. More parcels are loaded into the container, more fragment the space inside the container and thus caught some parcels cannot loaded although the actual remaining space still fit.

TABLE 3 is the comparison of the container loading with and without consider merging space after a parcel have been loaded. In order to evaluate the performance between using merging and no using merging algorithms, space utilization and load parcel ratio metrics are used. Besides, two test instances are randomly selected from each test class and evaluated for 10 replications to reduce the effect of noise.

As shown in TABLE 3, the result of the proposed method generates better performance than the algorithm without using merging space. Space utilization ratio is calculated by Equation 4. The larger number of space utilization ratio, the better results the algorithm generated.

TABLE 3. Space utilization ratio of MOSSO with and without merging space algorithm

Test pack	Test instance	Without space merging (%)	With merging space (%)
1	3	91.50 ± 0.87	**94.76** ± 0.00
	10	83.15 ± 0.46	**89.21 ± 1.64**
	20	89.97 ± 0.90	**94.78 ± 0.93**
2	5	88.41 ± 1.01	**91.83 ± 1.14**
	13	87.63 ± 1.29	**93.38 ± 0.11**
	33	86.37 ± 1.16	**90.18 ± 1.38**
3	4	89.67 ± 1.16	**93.97 ± 0.62**
	20	86.84 ± 1.49	**89.88 ± 0.71**
	26	91.06 ± 1.07	**93.67 ± 0.43**

There are some test instances in pack 3 in which the performance of the proposed method is equal to without using merging space. There are 8 parcel types with different dimensions, which generate more fragments in remaining spaces. Since these fragment spaces are varied in shape and volume, the algorithm not considered them as the mergeable space. As a result, even some fragment spaces could be merged, but the merged space still too small in volume to fit other parcels, that generate in poor performance.

b) Multi-objectives optimization

This section is dedicated to the analysis of the performance of the proposed method when considering space utilization and loaded balance criteria. Similar with the conducted experiments with and without merging space algorithm, the same data are used for evaluating the performance of the proposed method. The result of the MOSSO is shown in TABLE 4.

The loaded balance metrics are calculated based on the weight and position of each parcel inside the container. The container type used in the benchmark dataset has the dimension of 233, 587, 220 for width, length, and height, respectively. The idea gravity center of the container is $\left[{}^{233}/_{2}, {}^{587}/_{2}\right]$ and if the value of Cg_x, Cg_y are equal to the idea gravity center for corresponding axes, then the container has idea loaded balance with the value of balance G is 100.

The column without balance and with balance are the average results of space utilization, loaded balance ratio and gravity center of the proposed method with the parameter settings as in TABLE *2*. Column "Trade-off" present the tradeoff between space utilization and the balance ratio when consider loaded balance ratio. The positive value means the criteria of considering balance has higher result and the negative value means contrast. The results have shown that, the space utilization often slightly decrease while balance ratio is enhanced.

The Pareto frontier convergence of MOSSO for test pack 1-case 20, pack 2-case 5 and pack 3-case 26 are examples illustration in Figure 8. The MOSSO found near optimal solution at the first test iteration 35. Next iteration, the solution are slightly update. Thus the proposed method can find the optimal solution in short amount of time.

TABLE 4. Results of MOMO-SSO with and without consider loaded balancing

Test class	Test instance	Criteria	Without balance	With balance	Trade-off
pack1	3	Space	**94.76** ± 0.00	**95.76 ± 0.10**	**1.00**
		Balance	90.76 ± 2.29	**97.41 ± 0.42**	**6.65**
	10	Space	**89.21 ± 1.64**	88.35 ± 0.43	-0.86
		Balance	79.72 ± 2.53	**94.28 ± 1.03**	**14.56**
	20	Space	**94.78 ± 0.93**	92.85 ± 0.74	-1.93
		Balance	**74.55 ± 5.17**	**95.63 ± 1.14**	**21.08**
pack2	5	Space	**91.83 ± 1.14**	90.97 ± 0.52	-0.86
		Balance	**78.72 ± 3.76**	**93.40 ± 1.35**	**14.68**
	20	Space	**93.38 ± 0.11**	**91.77 ± 0.41**	-1.61
		Balance	**84.54 ± 2.40**	**94.14 ± 0.91**	**9.60**
	33	Space	**90.18 ± 1.38**	**88.78 ± 0.77**	-1.40
		Balance	**78.87 ± 3.00**	**92.76 ± 1.24**	**13.89**
pack3	4	Space	**89.97 ± 0.62**	87.93 ± 0.48	-2.04
		Balance	75.25 ± 2.65	93.12 ± 1.56	**17.87**
	16	Space	**89.88 ± 0.71**	88.15 ± 0.21	-1.73
		Balance	69.85 ± 2.32	**93.42 ± 1.42**	**23.57**
	26	Space	**93.67 ± 0.43**	93.44 ± 0.15	-0.23
		Balance	**69.03 ± 2.89**	**94.10 ± 0.19**	**25.07**

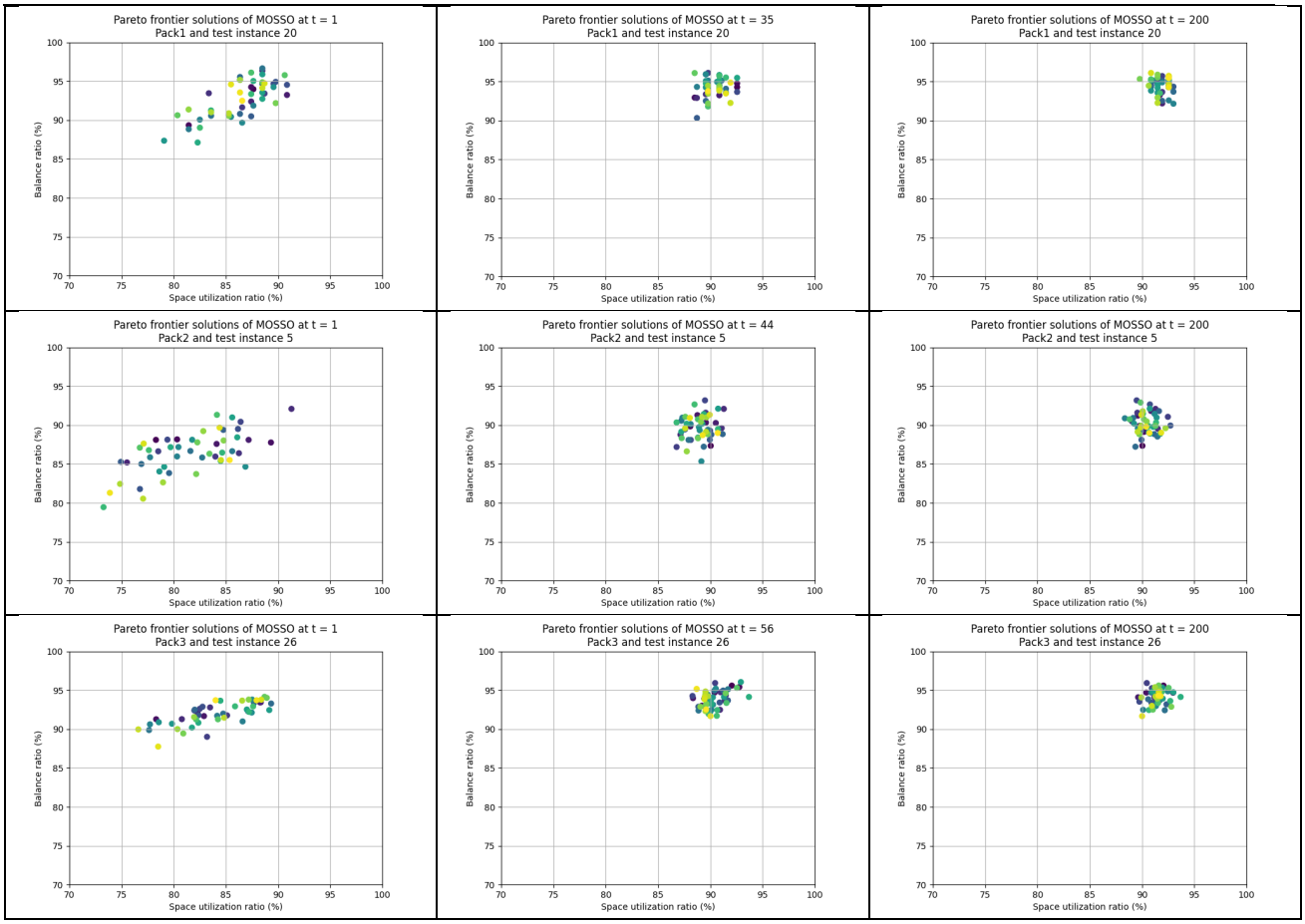


Figure 8. Pareto updating results of benchmark dataset

3) Result comparison

In order to evaluate the performance of the proposed MOSSO with existing methods in the literature, this section compare the results in term of space utilization. The data used to validate the proposed method are similar with existing methods TABLE 5, TABLE 6 and TABLE 7 show the references, the corresponding objectives, and the considered constraints.

TABLE 5. List of existing methods to compare

References	**Loading procedure**
[12]	Heuristic
[13]	Heuristic
[14]	Heuristic – Tabu search
[15]	Heuristic – Genetic algorithm
[16]	Genetic algorithm

TABLE 6. Considered objectives of the methods in TABLE 5

	[12]	[13]	[14]	[15]	[16]	MOSSO
Space utilization	✔	✔	✔	✔	✔	✔
Load balancing	✔		✔		✔	✔

TABLE 7. The constraints of the methods in TABLE 6

	[12]	[13]	[14]	[15]	[16]	MOSSO
Weight limitation	✔		✔	✔	✔	✔
Orientation		✔		✔		✔
Overlap	✔	✔	✔	✔	✔	✔
Full support						✔
Stability						✔

The fitness value comparison between proposed MOSSO and existing methods is presented in TABLE *8*. Since in this study, only three first test classes in the benchmark dataset are used to validate the proposed method, only results of first three test classes in the existing methods are mentioned. The last row is the average result of each method.

TABLE 8. Fitness value comparison

Test class	[12]	[13]	[14]	[15]	[16]	MOSSO
Pack1	80.77	87.54	88.14	**94.34**	91.31	**94.05**
Pack2	79.77	89.12	90.43	**94.88**	92.40	**93.25**
Pack3	81.07	90.32	90.86	**95.05**	93.32	**92.58**
Average	80.54	88.99	89.81	**94.76**	92.34	**93.29**

Although the fitness value of the proposed method is lower than only method [15], other objectives and crucial constraints are achieved. For instance, in the top result method, that is [15], they only consider the objective of space utilization but not fully support and stability of the parcel inside. It is worth to mentioned that the fully support, stability constraints and load balance criteria have equal or even more important than space utilization, especially with the high value products. The proposed method satisfies not only space, loaded balance criteria but also other objectives and constrains that listed as crucial aspects in practice. The visualization of the proposed method result is shown in Figure 9 where each figure illustrates for each test instance and test class.

D. Trade-off and Decisions

In the era of industry 4.0, many studies are proposed to enhance the performance of logistics and transportation, especially in reduce costs for each shipment by optimizing container loading. Indeed, to maintain the competitiveness of the companies in the market, advanced computing techniques should be applied to support decision makers in term of quality and time. Therefore, this study applies MOSSO algorithm to support decision maker decide loading sequence of set of products into a container. In addition, decision maker could adjust the solution to adapt with different cases and product characteristics. Following the Industry 3.5 concept, the quality of container loading solution is enhanced then increase the productivity and more satisfy customers.

V. CONCLUSION

For logistics and supply chains, CLP issues play a crucial role in important industrial and shipment applications. This study develops a UNISON framework for container loading optimization by considering multi-practical constraints. A multi-objectives and multi-populations-based simplified swarm optimization algorithm are proposed to enhance the performance of the container loading solution. This study concentrated on the loading problem which places a set of 3D rectangular shaped into a container to maximize the container space utilization and consider other practical constraints simultaneously such as loaded balancing and total loaded parcels. An effective container loading system should meet numerous complex criteria in practice and make the transportation process more efficient, stable, and safe. Besides, study framework proposes a digital decision support system to help decision-makers visualize loading solution and update better results quickly. Therefore, the logistics firm may optimize the value of items in a constrained container by maximizing space usage. The workers can get improved outcomes from a digital support system with loading visualization and prevent wasting time.

To validate the performance of the proposed methods, empirical data is used. The result shows that the proposed methods are effective to deal with practical constraints and criteria when compared with the simple loading approach which is without considering merging space algorithm and existing methods. The loading space utilization is improved compared with other studies. Although the proposed method has not reached the highest utilization of space, this study has considered multiple constraints. In future work, the realistic transportation process is much more complex and requires consideration. Based on the result of the proposed method and domain knowledge, the quality of the solution could be further improved to adapt to practical cases. Furthermore, more arising constraints can be investigated to optimize the loading solution.

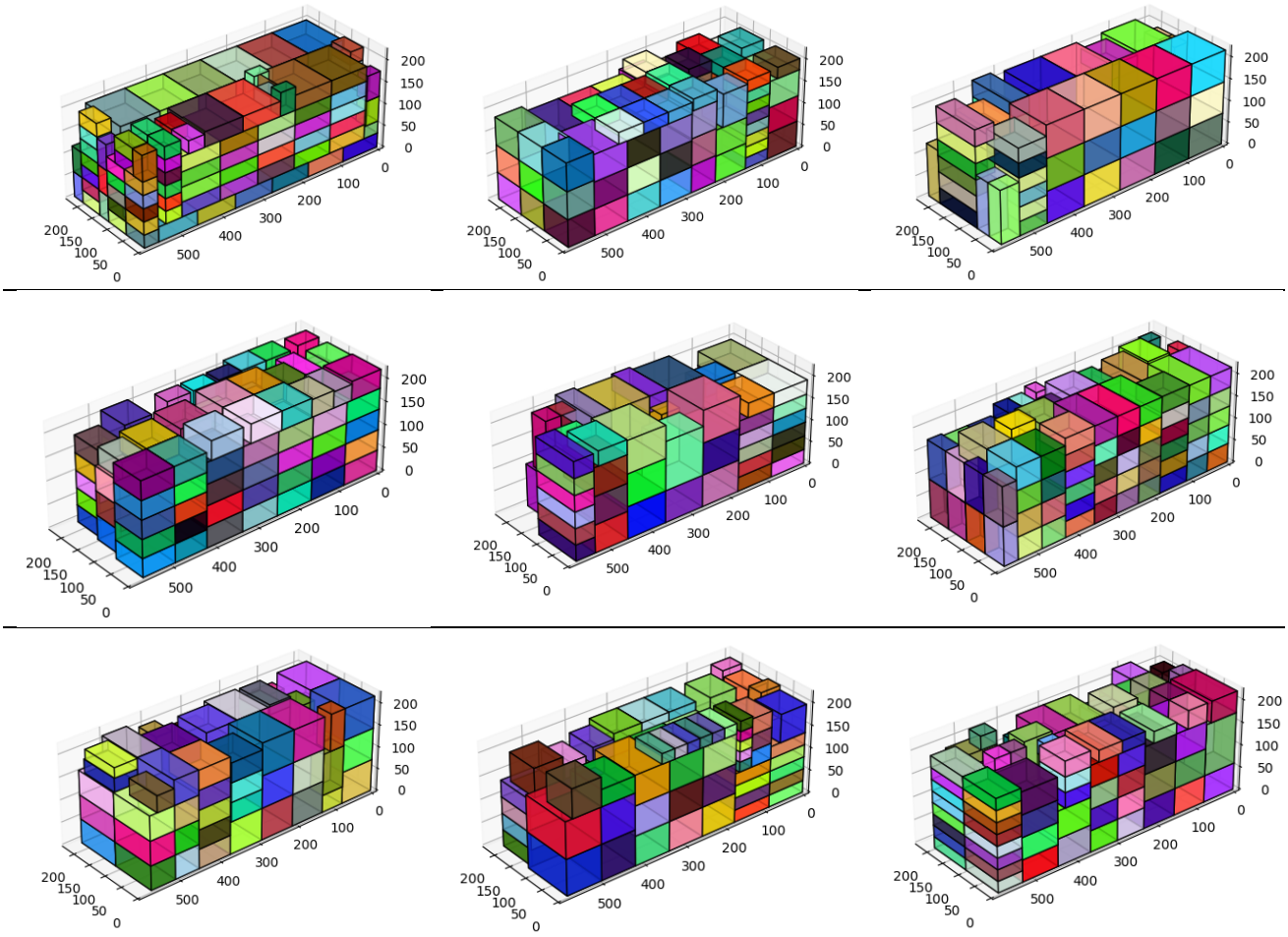


Figure 9. Visualization of the proposed MOSSO for benchmark dataset

Although the proposed method shows effective result, there are gaps to be further research and improve, as mentioned in TABLE 9. Indeed, to satisfy the realistic needs, the solution should consider more constraints and criteria.

TABLE 9. Further considerations in the container loading problem

Aspects		Details
Loading procedure	Merging space	Merging for more than two adjacent spaces
Parcel related	Stackability	Maximum stack weight of a product
Transport related	Multiple drops	A container transports products to multiple destinations
	Multiple customers	Product set of different customers should be placed together and separate with other customer products
	Loading complexity	The loading solution should not be too complicated that make difficult and time consuming to the loader

REFERENCES

[1] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," Journal of the operational research society, vol. 41, no. 11, pp. 1069-1072, 1990.

[2] X. Z. Zhao, J. A. Bennell, T. Bektas, and K. Dowsland, "A comparative review of 3D container loading algorithms," (in English), International Transactions in Operational Research, vol. 23, no. 1-2, pp. 287-320, Jan-Mar 2016, doi: 10.1111/itor.12094.

[3] M. Gajda, A. Trivella, R. Mansini, and D. Pisinger, "An optimization approach for a complex real-life container loading problem," Available at SSRN, 2020.

[4] J.-N. Zheng, C.-F. Chien, and M. Gen, "Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem," Computers & Industrial Engineering, vol. 89, pp. 80-87, 2015.

[5] I. Araya, M. Moyano, and C. Sanchez, "A beam search algorithm for the biobjective container loading problem," (in English), European Journal of Operational Research, vol. 286, no. 2, pp. 417-431, Oct 16 2020, doi: 10.1016/j.ejor.2020.03.040.

[6] T. Bayraktar, F. Ersöz, and C. Kubat, "Effects of memory and genetic operators on Artificial Bee Colony algorithm for Single Container Loading problem," Appl Soft Comput, p. 107462, 2021.

[7] J. M. Liu, Y. Yue, Z. R. Dong, C. Maple, and M. Keech, "A novel hybrid tabu search approach to container loading," (in English), Computers & Operations Research, vol. 38, no. 4, pp. 797-807, Apr 2011, doi: 10.1016/j.cor.2010.09.002.

[8] W.-C. Yeh, "A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems," Expert Syst Appl, vol. 36, no. 5, pp. 9192-9200, 2009.

[9] W.-C. Yeh, "A novel boundary swarm optimization method for reliability redundancy allocation problems," Reliability Engineering & System Safety, vol. 192, p. 106060, 2019.

[10] W.-C. Yeh, "New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series," IEEE Transactions on Neural Networks and Learning Systems, vol. 24, no. 4, pp. 661-665, 2013.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE transactions on evolutionary computation, vol. 6, no. 2, pp. 182-197, 2002.

[12] F. Parreño, R. Alvarez-Valdés, J. M. Tamarit, and J. F. Oliveira, "A maximal-space algorithm for the container loading problem," Informs J Comput, vol. 20, no. 3, pp. 412-422, 2008.

[13] K. He and W. Huang, "Solving the single-container loading problem by a fast heuristic method," Optimization Methods & Software, vol. 25, no. 2, pp. 263-277, 2010.

[14] J. Liu, Y. Yue, Z. Dong, C. Maple, and M. Keech, "A novel hybrid tabu search approach to container loading," Computers & Operations Research, vol. 38, no. 4, pp. 797-807, 2011.

[15] J. F. Gonçalves and M. G. Resende, "A parallel multi-population biased random-key genetic algorithm for a container loading problem," Computers & Operations Research, vol. 39, no. 2, pp. 179-190, 2012.

[16] A. G. Ramos, E. Silva, and J. F. Oliveira, "A new load balance methodology for container loading problem in road transportation," European Journal of Operational Research, vol. 266, no. 3, pp. 1140-1152, 2018.