Research Article

ROMAYA - Researches on Multidisciplinary Approaches 2024, 4(2): 184-194 ISSN:2791-9099

Hybrid Tabu Search Algorithm for Container Loading Problems

Konteyner Yükleme Problemleri için Hibrit Tabu Arama Algoritması

Canan Hazal Akarsu / Lect. Dr.
Istanbul University-Cerrahpaşa, Institute of Graduate Studies
hazalakarsu@iuc.edu.tr

Tarık Küçükdeniz / Assoc. Prof. Dr.
Istanbul University-Cerrahpaşa, Faculty of Engineering, Department of Industrial Engineering
tkdeniz@iuc.edu.tr

Abstract

The increasing impacts of globalization and the COVID-19 pandemic underscore the critical role of the logistics sector in global trade and national economies. Effective container loading plans enhance logistics and shipping efficiency while boosting customer satisfaction. The container loading, a complex combinatorial optimization problem, significantly affects the economy, environment, and safety. In modern logistics, developing solutions that ensure high volume utilization and meet a range of practical constraints is crucial. This paper introduces a hybrid tabu search algorithm that employs an iterative use of tabu search and heuristic methods to solve three-dimensional container loading problems, considering high volume utilization and practical constraints. The approach focuses on two main stages of the problem. In the first stage, the tabu search algorithm determines the loading sequence of box types; in the second stage, the loading heuristic is used for placing boxes into the container based on the established sequence. The developed hybrid tabu search algorithm has been coded and implemented using the Python programming language. The performance of the proposed algorithm has been evaluated using test problems provided by the OR-library. The algorithm aims to maximize volume utilization by minimizing the spaces between boxes during loading and optimizing the arrangement of box stacking. The results demonstrate that the algorithm provides high-quality solutions by achieving high volume utilization in problems with heterogeneous structures. The successful outcomes obtained from test sets characterized by strong heterogeneity and various practical constraints highlight the potential of the hybrid tabu search algorithm to enhance efficiency in logistic processes. This underscores the algorithm's ability to offer valuable practical applications in the logistics sector, particularly in reducing transportation costs and optimizing loading processes.

Keywords: Container Loading, Bin Packing, Tabu Search, Heuristic, Load Stability.
JEL Codes: C61,C44,L91

Özet

Küreselleşmenin artan etkileri ve Koronavirüs salgınının gösterdiği gibi, lojistik sektörü, dünya ticaretinde ve ulusal ekonomilerde hayati bir role sahiptir. Etkin konteyner yükleme planları, lojistik ve nakliye maliyetlerini azaltırken müşteri memnuniyetini artırmaktadır. Konteyner yükleme problemi, ekonomi, çevre ve güvenlik üzerinde önemli etkileri olan karmaşık bir kombinatoryal optimizasyon problemidir. Modern lojistikte, yüksek hacim kullanımı ve çeşitli pratik kısıtları karşılayacak çözümler geliştirmek büyük önem taşımaktadır. Bu çalışmada, üç boyutlu konteyner yükleme problemlerinin çözümü için yüksek hacim kullanımı ve pratik kısıtları dikkate alan, tabu arama ve sezgisel algoritmanın iteratif olarak kullanıldığı bir hibrit tabu arama algoritması geliştirilmiştir. Bu yaklaşım, problemin iki temel aşamasına odaklanır. İlk aşamada, kutu tiplerinin yükleme sırasını belirlemek için tabu arama algoritması; ikinci aşamada ise, belirlenen sıraya göre kutuların konteynere yerleştirilmesi için geliştirilmiş bir yükleme sezgiseli kullanılmıştır. Geliştirilen hibrit tabu arama algoritması, Python programlama dili kullanılarak kodlanmış ve uygulanmıştır. Önerilen algortimanın

Submission Date / Yayıma Geliş Tarihi : 13.02.2024
Acceptance Date / Yayıma Kabul Tarihi : 07.05.2024

performansı, OR-library kütüphanesinden sağlanan test problemleri ile değerlendirilmiştir. Algoritma, yükleme sırasında kutular arasındaki boşlukları minimize ederek ve kutuların istiflenme düzenini optimize ederek, hacim kullanımını maksimize etmeyi amaçlamaktadır. Sonuçlar, algoritmanın heterojen yapıdaki problemlerde yüksek hacim kullanımı sağlayarak kaliteli çözümler sunduğunu göstermektedir. Özellikle güçlü heterojen yapıdaki ve çeşitli pratik kısıtları içeren test setleri üzerinde elde edilen başarılı sonuçlar, hibrit tabu arama algoritmasının lojistik süreçlerdeki verimliliği artırma potansiyelini gözler önüne sermekte ve algoritmanın lojistik sektöründe, özellikle de taşıma maliyetlerini düşürme ve yükleme süreçlerini optimize etme açısından değerli pratik uygulamalar sunabileceğine işaret etmektedir.

Anahtar Kelimeler: Konteyner Yükleme, Kutu Paketleme, Tabu Arama, Sezgisel, Yük Stabilitesi.
JEL Kodları: C61,C44,L91

Introduction

The logistics sector, influenced by globalization, has begun to play an increasingly active role in international trade and the economies of countries. Especially in recent years, the COVID-19 pandemic has once again highlighted the critical importance of the logistics sector worldwide. It has become evident that production must continue and products must be delivered to our homes for life to be sustainable indoors. In a globalized world, borders have essentially disappeared, and the necessity to transport the cheapest goods at the lowest cost has emerged in global competition. To date, no alternative solution has been found to replace transportation. Therefore, developing fast and effective solutions that can reduce transportation costs will be one of the most important tools to benefit the national economy. For Turkey, which serves as a bridge in logistics, this issue holds even greater strategic importance. Reducing process costs and providing more efficient services to customers have become increasingly important for businesses in the face of growing competition. Logistics is one of the most critical processes in the supply chain for businesses. In the logistics process, companies primarily deal with the container loading problem (CLP). Efficient container loading strategies play a key role in minimizing logistics and transportation costs, while simultaneously enhancing customer satisfaction (Erbayrak et al., 2021). The CLP addressed in this paper is a significant combinatorial optimization (CO) problem with substantial implications for economic efficiency, environmental sustainability, and safety in practical applications (Erbayrak et al., 2021; Ramos et al., 2018). The problem of loading three-dimensional rectangular items into a three-dimensional rectangular container under specified constraints is known as the container loading problem (Zhu et al., 2021) and is typically an NP-hard problem (Scheithauer, 1992). In today's logistics industry, the complexity of constraints, the variety of cargo types, and the volume of orders are expanding rapidly, leading to significant difficulties in designing loading plans (Zhu et al., 2021). Despite advances in computer technologies, no optimal solution has yet been found for this complex and challenging problem encountered in real life.

The key to gaining an edge in global competition is the efficient management of supply chains and successful logistics services. On a global scale, logistics activities have significantly increased in recent years. As reported by Statista (2020), over 100 billion parcels were shipped in 2019, and is predicted to reach 200 billion by 2025. In line with this growing demand, shipping companies are increasingly focusing on optimizing product logistics management. It has been observed that the efficient use of shipping vehicles has a significant impact on reducing logistics costs. Reflecting this reality in the scientific literature, CLPs are extensively studied. These problems cover numerous real-world scenarios and must satisfy practical constraints while also aiming for maximum space utilization. These problems are particularly important for industrial sectors where effective loading of items into airplanes, ships, trailers, or trucks is required (Liu et al., 2011).

Container loading problems are also referred to as 3D bin packing problems or pallet loading problems in the literature. Over the past 10 years, Bortfeldt and Wascher (2013), Zhao et al. (2016), and Ali et al. (2022) have reviewed the three-dimensional container loading literature, presenting and categorizing different problems and methods. As foreseen by Bischoff and Ratcliff (1995), the focus of research in this field is on considering various constraints in real-world scenarios. Zhao et al. (2016) contributed to the classification of algorithmic approaches necessary for solving more complex versions of the problem by focusing on the heterogeneity of boxes and containers. Gimenez-Palacios et al. (2021) examined the problem from a broader perspective by addressing additional constraints encountered in logistics processes, alongside those proposed by Bischoff and Ratcliff (1995). The comprehensive literature review conducted by Ali et al. (2022) provided a thorough evaluation of the existing solution approaches in the field and highlighted potential directions for future studies. Container loading problems fall under the category of NP-hard CO problems (Scheithauer, 1992), and the literature offers numerous approaches for solving them. These methods include exact algorithms, constructive heuristics, metaheuristics, tree search algorithms, hyperheuristics and machine learning (ML) techniques (Ali et al., 2022). Ali et al. (2022) classified CLPs into off-line and on-line problems. Off-line problems are those in which all items to be loaded are ready for loading at time t0, and all information about the items to be loaded, such as width, length, height, weight, etc., is known in advance. In on-line problems, however, the items to be loaded arrive sequentially for loading, and information about the items, such as width, length, height, weight, etc., is only known when the item arrives for loading. The approaches proposed in the literature for solving off-line problems include exact algorithms, constructive heuristics, metaheuristics, tree search algorithms, hyperheuristics and ML algorithms. For on-line problems, the methods proposed in the literature are primarily constructive heuristics and ML techniques.

In practical scenarios, container loading problems require not only maximizing space utilization by finding the best geometric arrangement of items but also adhering to various real-world constraints. These constraints include factors such as item orientation, load stability, weight restrictions, and the load-bearing capacity of the container. Ensuring the stability of the load is particularly vital, where each box needs sufficient support from beneath to maintain balance during transportation. Addressing both the goal of space optimization and these constraints makes it difficult to rely solely on mathematical models, prompting the use of heuristic methods to find feasible solutions. Given the complexity of real-world constraints, mathematical approaches have proven insufficient, and heuristic solution methods have been preferred instead. Heuristic methods based on human knowledge, metaheuristic methods using intelligent search strategies such as genetic algorithms and simulated annealing, and hybrid approaches that combine these with other methods like tree search have come to the forefront. Although exact algorithms have also been proposed, their practical use is limited due to computational requirements. Trivella and Pisinger (2016) proposed a MILP model for a 3D packing problem that aims to minimize the sum of deviations from the desired center of gravity and the number of boxes used. Erbayrak et al. (2021) extended the work of Trivella and Pisinger (2016) by introducing new constraints and an objective function, proposing a new mathematical model. The datasets introduced in the study of Trivella and Pisinger (2016) were used to validate the model. Additionally, a real-world case study of a Turkish filter company's CLP was solved using the proposed mathematical model. While the authors noted that the proposed mathematical model produced effective results, they also emphasized that for solving large-scale container loading problems encountered in real life, developing metaheuristic algorithms would be more feasible than a mathematical modeling approach.

Bio-inspired methods, such as metaheuristics, offer an efficient approach for obtaining approximate solutions to container loading problems, while avoiding the complexities involved in sophisticated software implementations. Leon et al. (2019) proposed a tabu search algorithm to solve the CLP. Tijjani and Ozkaya (2014) compared five different reinforcement learning algorithms and two different evolutionary algorithms for solving the CLP. A study that used a deep learning algorithm in a hybrid manner with a heuristic method was conducted by Zhu et al. (2021). Although tree search algorithms have been proven to be a successful paradigm for solving container loading problems, applying them to large-scale problems is very time-consuming. To overcome this challenge, Zhu et al. (2021) integrated a deep learning algorithm into the tree search algorithm. Saikia et al. (2018) proposed an approach using evolutionary strategies and reinforcement learning techniques for the problem of loading containers from the dock to the ship. Ali et al. (2022) point out that most published studies utilizing ML have focused solely on basic geometric constraints, while practical constraints are frequently oversimplified in these approaches.

This paper proposes a hybrid approach combining the tabu search algorithm with heuristic algorithms, aiming to improve both high space utilization and practical constraints such as the full support requirement for each box from below. This paper is organized into five sections. In the introduction section, the impacts of globalization and the COVID-19 pandemic on the logistics sector, along with the importance of the problem, are explained, and studies on container loading problems and solution methods are reviewed. In the second section, the definition of the container loading problem, its constraints, and the objective function are explained. The third section details the proposed hybrid tabu search algorithm developed for CLPs. The fourth section presents the scenarios in which the algorithm was tested and the analysis of the results obtained. In the conclusion, the findings are summarized, and suggestions for future studies are provided.

Problem Definition

Loading three-dimensional rectangular items into three-dimensional rectangular containers and transporting them along the supply chain under specified constraints is known as the container loading problem (Zhu et al., 2021). CLP is a common NP-hard problem (Scheithauer, 1992). Despite advances in computer technologies, no optimal solution has yet been found for this problem at the scale encountered in real life.

Dyckhoff (1990) and Wäscher (2007) typologies are two methods used to classify cutting and packing (C&P) problems. The Dyckhoff typology classifies problems according to size (one, two or three-dimensional), type of assignment (selection of small items or selection of containers), variety of large objects (single, same, different), and variety of small items (few, medium, many). Wäscher (2007) later extended this typology by focussing on the variety of small items and the constancy of the sizes of large items. These classifications are critical for better understanding the problems and developing appropriate solution methods. In particular, the variety of small items and the fixed size of large items are the main factors in determining packaging strategies. The problem considered in this paper is labeled as Single Large Object Placement Problem (SLOPP) in the typology of Wäscher et al. (2007) and 3/B/O/R in the classification of Dyckhoff (1990).

In the CLP considered in this paper, n different types of rectangular boxes $(B=\{b_1,b_2,...,b_n\})$ are to be placed in a container of length L, width W and height H. The length, width and height dimensions (l_i,w_i,h_i) specified for each box type are available and there are a certain number of boxes $(m_i, i=\{1,2,...,n\})$ of each type. The process of placing the boxes in the container starts from a virtual starting point inside the container, i.e. the right rear bottom corner. This corner is defined as $O(0,0,0)$ in 3D Cartesian coordinate system and the boxes are aligned along the x (length), y (width) and z (height) axes relative to this starting point. The position of the box b_{ij} *(box j of type* b_i, $i=\{1,2,...,n\}$ *and* $j=\{1,2,...,m_i\}$*)* inside the container is expressed by the coordinates (x_{ij},y_{ij},z_{ij}) of the right rear bottom corner (See Figure 1).

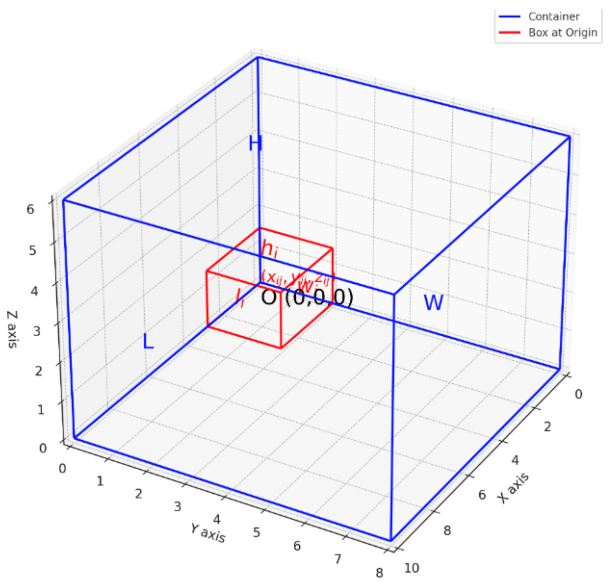

Figure 1. Container and box layout in 3D coordinate system

The proposed model takes into account several key constraints, including:

- Product positioning constraints: Boxes are arranged vertically, ensuring that they do not overlap with one another during loading.
- Weight limit constraints: The cumulative weight of the boxes loaded must remain within the container's maximum weight capacity.
- Grouping constraints: Wherever feasible, similar types of boxes are grouped together to streamline the loading and unloading processes.
- Orientation constraints: Boxes can be rotated into one of six possible orientations for loading. Some boxes have restricted orientation options, while others are allowed to be positioned in any of the six orientations.
- Load stability constraints: Boxes must be fully supported and cannot be suspended in mid-air. The bottom face of each box must rest securely either on the container floor or on other boxes beneath it. Additionally, at least one side of each box should be in contact with the container walls or adjacent boxes.

The objective function of the model is the maximization of volume utilization. Volume utilization (VU) is the ratio of the total volume of loaded boxes to the container volume.

Hybrid tabu search algorithm

Metaheuristic algorithms can offer high volume efficiency but sometimes fall short in overcoming practical constraints. On the other hand, heuristic methods based on human experience can give superior results compared to metaheuristics, especially in respect to these practical limitations (Liu et al., 2011). In this context, this paper proposes an iterative approach using tabu search and heuristic algorithm to solve CLPs. This approach focuses on two main phases of the problem: Firstly, the problem of optimizing the ordering of the types of boxes to be loaded into the container and secondly, the problem of placing these boxes into the container in the most efficient way. In the first stage, a tabu search algorithm is used to determine the loading order of the box types; in the second stage, a loading heuristic is used to place the boxes in the container. The pseudo-code of the hybrid tabu search algorithm is given in Figure 2.

```
Algorithm: Hybrid Tabu Search

Input: Set of boxes B = {b1, b2, ..., bn}
Output: Best solution s_best for the container loading problem

Initialize:
   Create s0 as the loading order of box types.
   s_best ← s0.
   Initialize tabu_list.

while the termination criterion is not met do:
   Generate neighbor solutions s' from s.
   Add the neighbor solutions whose actions are not in the tabu list to the valid neighbors list.
   Perform loading with the loading heuristic for valid neighbor solutions and calculate the value
of f(s').
   Select the solution with the highest f(s') value among the valid neighbors and set s ← s'.
      if f(s') > f(s_best):
         s_best ← s.
      endif
   Update tabu_list.
endwhile

Result:
   Return s_best.
```

Figure 2. Pseudocode of the hybrid tabu search algorithm

The solution is encoded as the sequence of box types to be loaded. The initial solution is created as an ascending order of box types. For example, in a problem with 5 different box types, the initial solution is [1, 2, 3, 4, 5]. This initial solution represents the order in which the boxes will be placed and serves as the baseline that the algorithm will attempt to improve. This arrangement, assumed to be the best solution, is referred to as s_best. The algorithm iteratively generates new solutions while maintaining a tabu list to avoid repeating forbidden moves. In each iteration, 10 potential solutions are generated to increase diversity and expand the exploration space. Valid solutions whose actions are not in the tabu list are sent to the loading heuristic. Boxes are placed into the container using the loading heuristic according to the new loading sequence. The objective function value of the solution (s) is calculated using Formula 1. The objective function of the model is to maximize VU, defined as the ratio of the total volume of the loaded boxes to the container's volume.

$$VU(s) = \frac{\sum_{i=1}^{K} l_i \times w_i \times h_i \times k_i}{L \times W \times H} \quad (1)$$

In Formula 1, l_i, w_i, h_i represent the length, width and height of the type box b_i; k_i is the number of loaded boxes of type b_i where $1 \leq k_i \leq m_i$; K is the number of loaded box types where $1 \leq K \leq n$; L, W, H are the length, width and height of the container.

Among the valid solutions whose actions are not in the tabu list, the best one is accepted as the current solution, and if the current solution is better than the best solution, s_best is updated. This process continues until the specified termination condition is met. When the algorithm terminates, the obtained s_best is used as a guide for how to load the container most efficiently.

Loading Heuristic:

After determining the loading order of the box types by tabu search, the boxes are placed into the container using the loading heuristic. The pseudo-code for this heuristic is presented in Figure 3. Initially, the empty (loadable) space is considered to be the entire container. The loading heuristic then follows a specific sequence and method to determine the appropriate block:

1. Calculating Possible Orientations: The possible_orientation method calculates the different orientations in which the box can fit into the remaining space $S_{remaining}$. For each orientation, it checks whether the box's height (h_i), width (w_i) and length (l_i) dimensions fit within the dimensions of the available space ($L_{remaining}$, $W_{remaining}$, Hz) and whether they comply with the box's orientation constraints.

2. Block Strategy: Boxes are placed in blocks to achieve high volume utilization. These blocks are formed by combining the same or different box types and are typically rectangular in shape. The majo-

rity of loading approaches used in the literature load boxes into the container one by one. Loading individual boxes frequently creates fragmented and irregular gaps within the container, which in turn reduces overall space efficiency. Additionally, individually loaded boxes generally create a less stable loading arrangement, making the loading and unloading processes more time-consuming and complex. On the other hand, the block strategy combines boxes to form larger and more regular blocks, which maximizes space utilization by reducing unused spaces in the container. Large, regular blocks create a more stable structure within the container, enhancing the stability of the load (Liu et al., 2011). The block strategy also offers time savings in logistics and transportation operations by enabling quick loading and unloading of blocks of the same boxes. Moreover, blocks allow boxes to support each other, reducing the load on individual boxes and increasing their load-bearing capacity. The algorithm generates a series of blocks for each possible orientation and calculates how efficiently each block fits into a specific area of the container using the Evaluation Function $f(i)$.

3. Evaluation Function $f(i)$: During the placement of each box block, the function $f(i)$ is used to calculate the volume of the remaining space. This function evaluates how well the block to be placed fits into the remaining space of the container.

$$f(i) = L_{remaining} \times W_{remaining} \times H_{remaining} - l_i \times w_i \times h_i \times k_i \quad (2)$$

$L_{remaining}$, $W_{remaining}$, $H_{remaining}$: The length, width and height of the remaining space $S_{remaining}$

l_i, w_i, h_i: The length, width, and height of the boxes of type b_i

k_i: The number of boxes of type b_i loaded into the remaining space $S_{remaining}$

```
Algorithm: Loading Heuristic

Input: Set of boxes B sorted according to the order determined by Tabu search,
Container dimensions (L, W, H)

Output: Loaded container C

Initialize:
   Start with an empty container C of dimensions (L, W, H).
   Initialize the initial empty space list remain_list with ((0, 0, 0, L, W, H)).
   Determine the first loading space S* as (0, 0, 0, L, W, H).
   Sort the boxes in set B according to the Tabu search order and create a Box object for each
box.

Loading Loop:
while (remain_list is not empty and there are unloaded boxes) do
   Select the most suitable empty space as S*.
   for each box type in set B:
      Take the box as current_box.
      if (current_box has not been loaded yet and it fits into S*) then
         Determine the most suitable block for current_box and place it into S*.
        After placement, update the remaining empty spaces and add the new spaces to remain_list.
      else
         Move on to the next box type.
      endif
   if (S* can be merged and there is a suitable space in the waste_list) then
      Merge S* with the appropriate waste_list space and obtain a new S*.
   endif
endwhile

Result:
- Return the loaded container C.
```

Figure 3. Pseudocode of the loading heuristic

4. Rules: The placement of boxes is subject to certain rules:

- Rule 1: If $f(i)$ is zero, it means that this block completely fills the space and this block is selected.
- Rule 2: If $f(i)$ is not zero, meaning the block does not fill the space completely in 3 dimensions, the block that fully fills the space in two dimensions is selected.
- Rule 3: If a block does not meet the other rules, the block with the smallest $f(i)$ value is selected.
- Rule 4: If multiple blocks have the same $f(i)$ value, the blocks' surface areas are considered, and the block with the largest surface area is selected.

5. Managing the Remaining Empty Spaces: When suitable blocks are selected and loaded into the container, non-rectangular empty spaces may occur. To ensure that every remaining empty space retains a rectangular shape and to optimize space usage, the available space is divided into sections. Once a box is placed into a given rectangular area, the remaining space is subdivided into three new rectangular regions: the space to the right, the space above, and the space in front of the loaded box. There are six possible ways to create these three remaining areas, as shown in Figure 4. The partitioning scheme in Figure 4(a), which takes into account the stability and support of the boxes, is used in this paper.

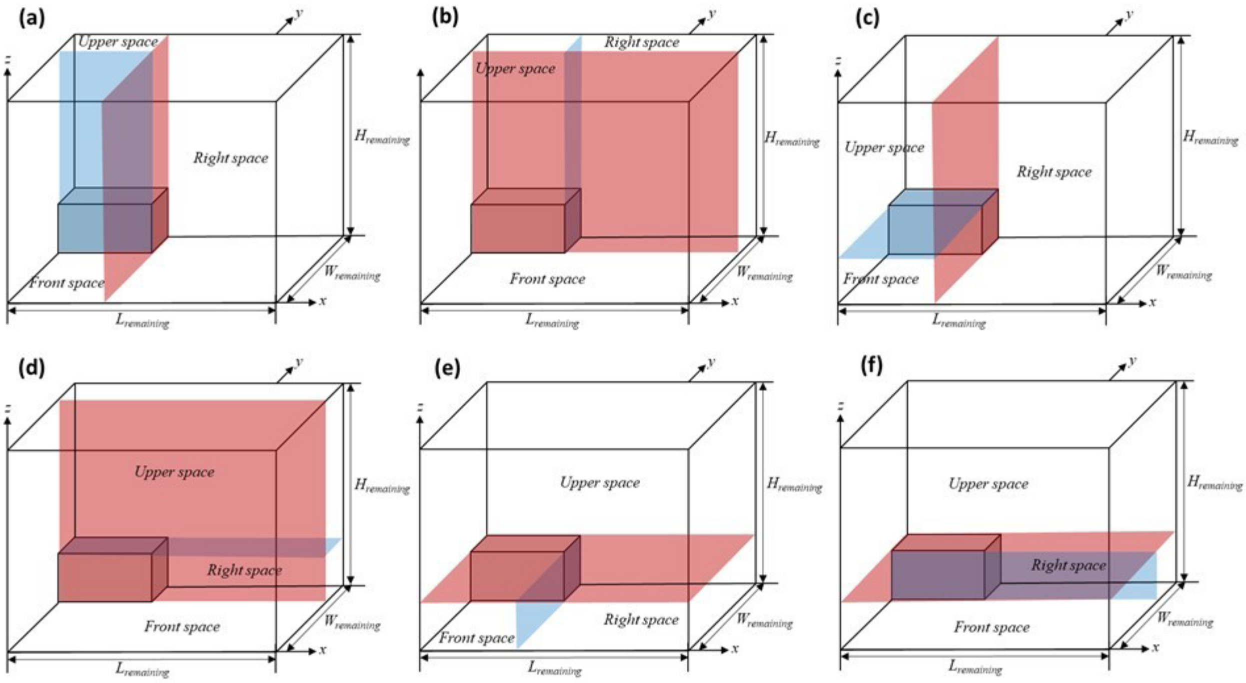

Figure 4. Six possible ways to partition the remaining space

If these empty spaces are directly considered as waste spaces, the loading may result in low volume utilization. To effectively use the empty spaces, they need to be merged according to specific rules:

- Rule 1: If two adjacent spaces share the same height and are aligned along the x or y axis, and have either the same length or width, they should be merged, as shown in Figure 5(a).
- Rule 2: If the lengths and widths of two adjacent spaces differ, but their heights are the same, and their total length or width equals the length or width of the current space, they should be merged, as illustrated in Figure 5(b), to create a new remaining space.
- Rule 3: If the lengths and widths of three adjacent spaces differ, but their total length and width match the dimensions of the current space, these areas should be merged, as depicted in Figure 5(c).

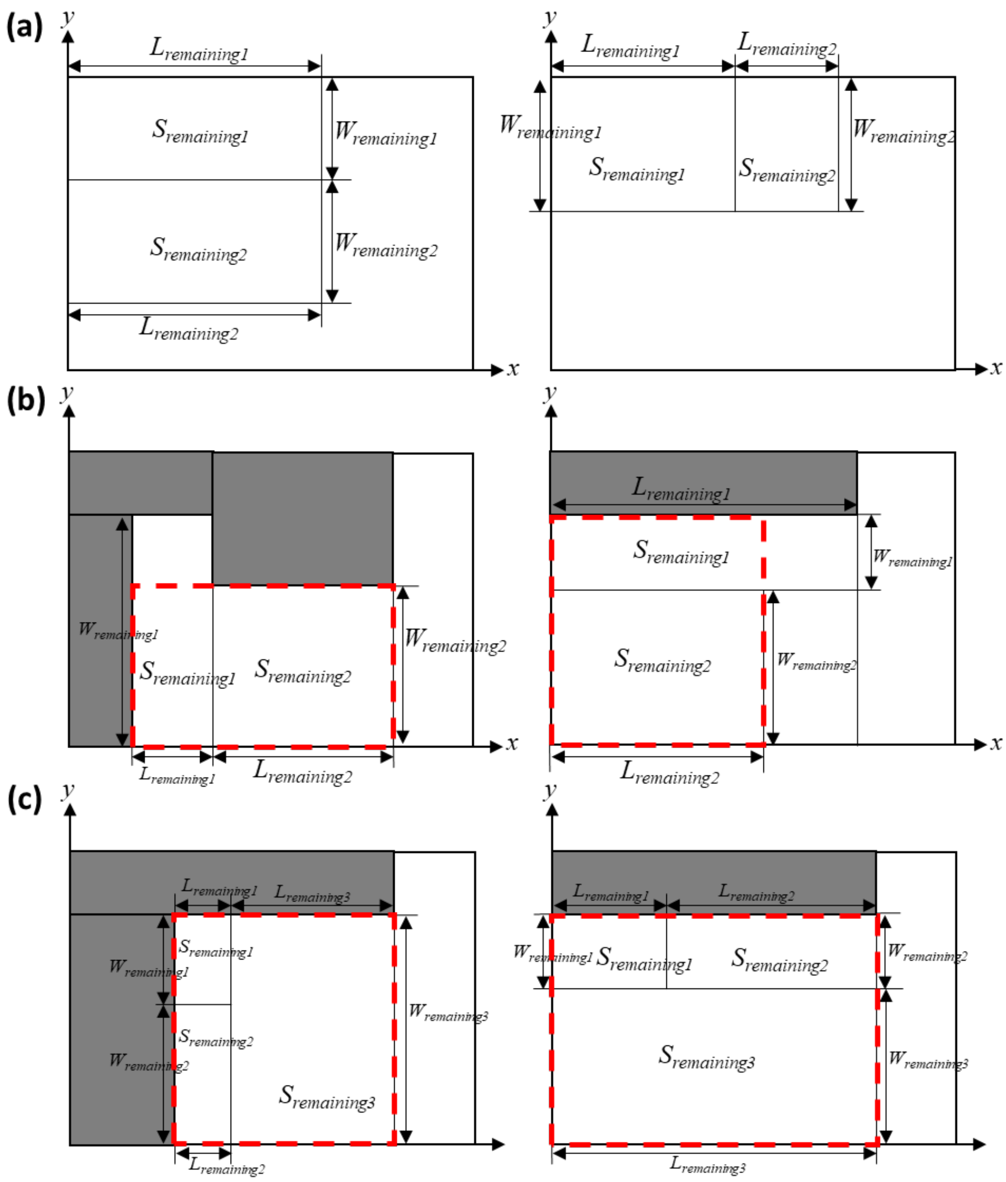

Figure 5. Merging of the remaining spaces

Experimental Results

The aim of this paper is to develop a hybrid tabu search algorithm capable of effectively addressing container loading problems, providing high-quality solutions in a reasonable time, and to test the algorithm's convergence performance using benchmark problems. The developed hybrid tabu search algorithm was coded and implemented using the Python programming language. The performance of the proposed approach was evaluated using a well-known benchmark dataset from the OR-library (http://people.brunel.ac.uk/~mastjjb/jeb/info.html), originally provided by Bischoff and Ratcliff (1995). The dataset includes 7 different test classes (thpack1, thpack2, ..., thpack7) ranging from weakly heterogeneous to strongly heterogeneous, with each test class containing 100 test problems. The test problems involve placing varying numbers of boxes into a standard ISO container with dimensions of 587×233×220. The objective function is to maximize the volume utilization of a single container. Volume utilization is defined as the ratio of the total volume of the loaded boxes to the container's volume.

The performance of the developed hybrid tabu search algorithm was evaluated on a total of 700 problems across 7 different test classes. For each of the 100 problems within each test class, the algorithm was run for 300 iterations, and the average volume utilization achieved by the algorithm in each test class was analyzed. This comprehensive analysis reveals the algorithm's ability to adapt to different problem structures and its performance across varying degrees of heterogeneity. Examining the averages of the problems in each test class, as shown in Table 1, it is observed that the algorithm is capable of producing consistent and high-quality solutions even in test classes where complexity and diversity increase. These results demonstrate the algorithm's capacity to provide an end-to-end solution for container loading problems and its adaptability to practical applications.

Table 1. Average Volume Utilization in Test Classes

Test Class	Volume Utilization Rate (%)
1	84,34
2	86,52
3	88,33
4	88,16
5	87,99
6	87,26
7	86,18

To test the convergence performance of the hybrid tabu search algorithm, the first 3 problems of the most strongly heterogeneous test class, thpack7, from the dataset were selected from the 700 problems solved by the algorithm. The graphs showing the convergence performance of the algorithm for these problems are presented in Figures 6-8. These problems were specifically chosen to test the algorithm's ability to handle complexity in problem-solving. Each problem contains 20 different box types with a certain number of boxes for each type. This provides an ideal scenario to evaluate the algorithm's capability to effectively place boxes of varying sizes and quantities.

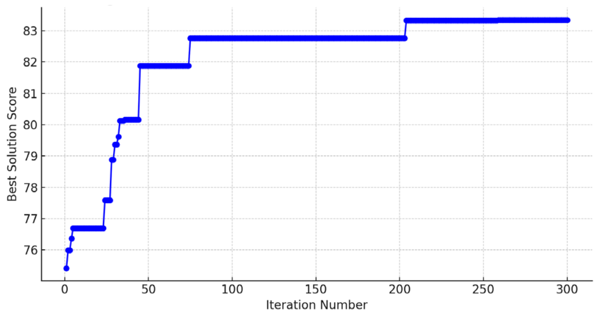

Figure 6. Convergence Performance of the Algorithm for Problem 7-0

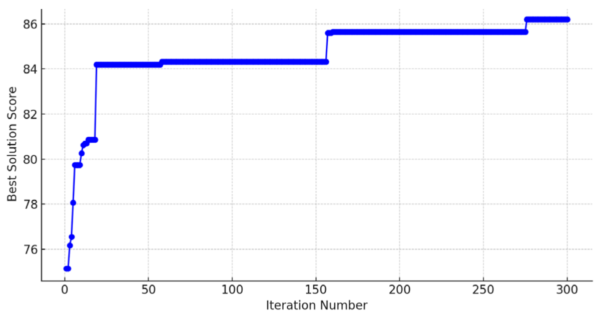

Figure 7. Convergence Performance of the Algorithm for Problem 7-1

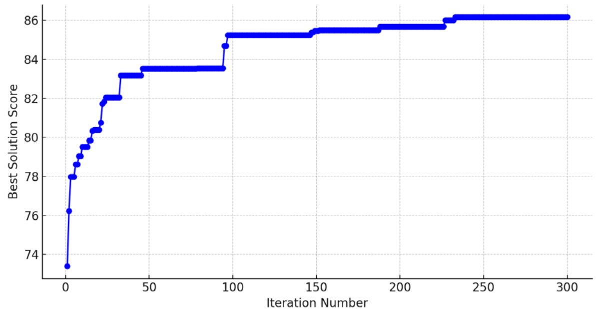

Figure 8. Convergence Performance of the Algorithm for Problem 7-2

As shown in Figures 6-8, the algorithm rapidly converges to high-quality solutions and continues to improve as iterations progress. Detailed analyses of the first 3 problems of the most strongly heterogeneous test class, thpack7, reveal that the algorithm performs highly even in complex scenarios involving 20 different box types with a certain number of boxes for each type. These findings demonstrate the algorithm's adaptability and overall effectiveness across various scenarios of volume utilization and box type diversity.

Table 2. Best Achieved Solution and Loading Order for Three Problems

Test Problem	Volume Utilization Rate (%)	Loading Order of Box Types
7-0	83,34	[14, 16, 0, 15, 11, 4, 17, 2, 6, 5, 9, 18, 7, 12, 1, 8, 10, 13, 19, 3]
7-1	86,20	[7, 16, 5, 2, 3, 13, 18, 1, 10, 14, 19, 17, 4, 12, 15, 9, 6, 8, 0, 11]
7-2	86,17	[16, 11, 8, 12, 17, 15, 0, 10, 1, 6, 18, 7, 2, 9, 19, 14, 3, 4, 13, 5]

The best solution (volume utilization rate) achieved after 300 iterations for the three problems and the optimal loading order of box types are summarized in Table 2. In all three problems, the algorithm achieved high volume utilization rates and effectively placed various box types. The loading arrangements for the container based on the solutions of the three problems are shown in Figures 9-11.

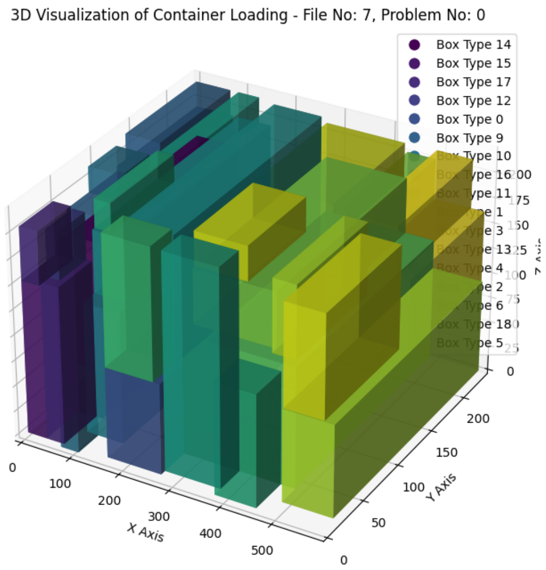

Figure 9. Optimal Loading Arrangement for Problem 7-0

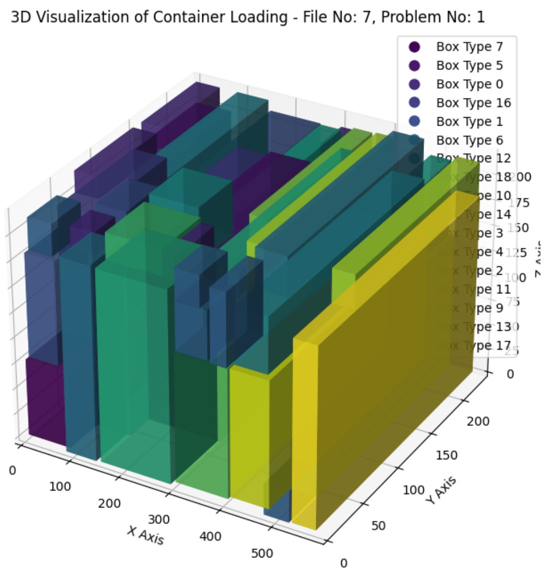

Figure 10. Optimal Loading Arrangement for Problem 7-1

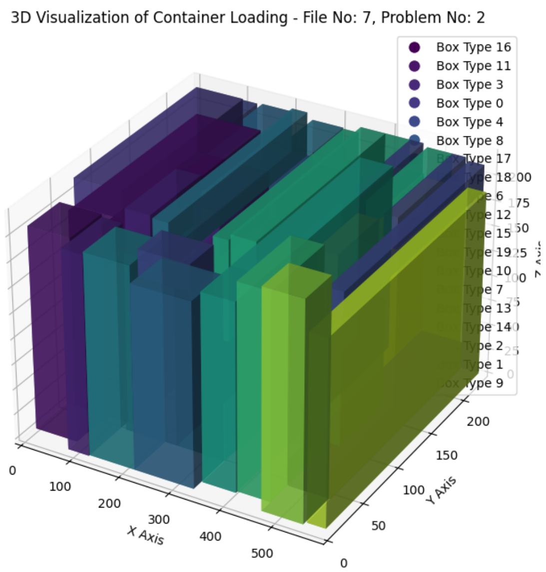

Figure 11. Optimal Loading Arrangement for Problem 7-2

Conclusion

The container loading problem is of great economic and operational significance, especially in the logistics sector, which is a critical component of global trade and supply chain management. Solving this problem can provide a competitive advantage for companies aiming to reduce costs and increase customer satisfaction. The primary objective of this paper is to address the CLP, which is one of the key challenges faced by the logistics industry, and to provide an effective solution. In this paper, a hybrid tabu search algorithm was developed for container loading problems, proposing a hybrid approach that combines the tabu search algorithm with heuristic algorithms to improve high space utilization and address practical constraints, such as the full support requirement for each box from below. The algorithm was implemented using the Python programming language, and the benchmark dataset from the OR-library was used to evaluate the performance of the algorithm. The findings obtained from 300 iterations for each of the 700 problems in seven different test classes showed that the algorithm offers high-quality solutions by achieving high volume utilization rates even in problems with strongly heterogeneous structures. Specifically, performance analyses conducted on the first three problems of the most strongly heterogeneous test class, thpack7, within the benchmark dataset, demonstrated that the proposed algorithm can effectively place boxes of varying sizes and quantities into the container and achieve high volume utilization rates in solving heterogeneous problems.

For future studies, it is recommended to further enhance the hybrid tabu search algorithm and perform an in-depth analysis of its performance by comparing it with other metaheuristic approaches. Additionally, integrating artificial learning techniques can aim to enhance the predictive and adaptive features of the algorithm, allowing it to respond more rapidly and effectively to real-time changes encountered in dynamic logistics environments. In this context, the integration of ML and artificial intelligence technologies with the algorithm will further improve automation and intelligent decision-making processes. Finally, investigating the role of the hybrid tabu search algorithm in developing sustainable logistics solutions will be an important step towards maximizing economic and environmental efficiency.

References

Ali, S., Ramos, A. G., Carravilla, M. A., & Oliveira, J. F. (2022). On-line three-dimensional packing problems: a review of off-line and on-line solution approaches. Computers & Industrial Engineering, 108122.

Bischoff, E.E., and Ratcliff, M.S.W. (1995). Issues in the development of approaches to container loading. OMEGA, 23(4), 377-390.

Bortfeldt, A., & Wascher, G. (2013). Constraints in container loading–a state-of-the-art review. European Journal of Operational Research, 229(1), 1–20.

Dyckhoff H. (1990). A typology of cutting and packing problems. European Journal of Operational Research, 44, 145–159.

Erbayrak, S., Özkır, V., & Yıldırım, U. M. (2021). Multi-objective 3D bin packing problem with load balance and product family concerns. Computers & Industrial Engineering, 159, 107518.

Gimenez-Palacios, I., Alonso, M., Alvarez-Valdes, R., & Parren o, F. (2020). Logistic constraints in container loading problems: the impact of complete shipment conditions. TOP: An Official Journal of the Spanish Society of Statistics and Operations Research, 29(1), 177-203.

Leon, P., Cueva, R., Tupia, M., & Dias, G. P. (2019). A taboo-search algorithm for 3d- binpacking problem in containers. In World Conference on Information Systems and Technologies (pp. 229–240). Springer.

Liu, J., Yue, Y., Dong, Z., Maple, C., & Keech, M. (2011). A novel hybrid tabu search approach to container loading. Computers & Operations Research, 38(4), 797–807.

Ramos, A. G., Silva, E., & Oliveira, J. F. (2018). A new load balance methodology for container loading problem in road transportation. European Journal of Operational Research, 266(3), 1140–1152.

Saikia, S., Verma, R., Agarwal, P., Shroff, G., Vig, L., & Srinivasan, A. (2018). Evolutionary RL for Container Loading. arXiv preprint arXiv:1805.06664.

Scheithauer, G. (1992). Algorithms for the container loading problem. In Operations Research Proceedings 1991 (pp. 445–452). Springer.

Statista (2020). Global parcel shipping volume between 2013 and 2026. URL https:// www.statista.com/statistics/1139910/parcel-shipping-volume-worldwide.

Tijjani, S., & Ozkaya, A. (2014). A comparison of reinforcement learning and evolutionary algorithms for container loading problem. In 2nd International Symposium on Engineering, Artificial Intelligence and Applications.

Trivella, A., & Pisinger, D. (2016). The load-balanced multi-dimensional bin-packing problem. Computers & Operations Research, 74, 152-164.

Wäscher, G., Haußner, H., Schumann, H. (2007). An improved typology of cutting and packing problems. European Journal of Operational Research, 183(3), 1109– 1130.

Zhao, X., Bennell, J. A., Bektas¸, T., & Dowsland, K. (2016). A comparative review of 3d container loading algorithms. International Transactions in Operational Research, 23 (1–2), 287–320.

Zhu, Q., Li, X., Zhang, Z., Luo, Z., Tong, X., Yuan, M., & Zeng, J. (2021, October). Learning to Pack: A Data-Driven Tree Search Algorithm for Large-Scale 3D Bin Packing Problem. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 4393-4402).