Advances in Artificial Intelligence Research (AAIR)
ISSN 2757-7422
www.dergipark.com/aair/

Vol. 3 (No. 1), pp. 12-18, 2023
doi: 10.54569/aair.1216400
Published online: Jan 8, 2023

A Block-Building Based GRASP Method for Solving Container Loading Problem

Merve ÖZDEMİR[1*], Tuncay YİĞİT [1]

[1] Süleyman Demirel University, Faculty of Engineering, Department of Computer Engineering, Isparta, Turkey

Abstract

The importance of container transportation is constantly increasing. For this reason, lower cost transportation is of great importance for companies in transportation by air, land, rail and sea in domestic and international markets. One way of reducing the costs is to utilize the container volume effectively. In this study, a block-building based GRASP method is proposed for solving the container loading problem. The proposed method achieved a loading factor over 91% on the BR dataset. The results are compared with other GRASP methods and other heuristic or meta-heuristic algorithms in the literature. The results show improvements in comparison to the other methods.

Keywords: *Container loading problem, optimization, heuristic, GRASP.*

1. Introduction

There is a significant increase in the need for logistics because of the growth in trade volumes locally and globally thanks to the rapid development in the field of transportation. Land, air, sea and rail transportations gain value and the investments in these areas are increasing. With the globalization of trade in the world, the volume of trade is also growing. The growing trade volume has brought competition to the fore in domestic and international businesses. Countries have started to invest more in the logistic sectors to perform better in the competitive environment.

The container loading problem (CLP) is one of the most important problems in terms of providing better and faster transportation. In today's competitive conditions in domestic and international markets, every organization aims to deliver its product at the lowest cost. One way of reducing the costs is to use the container volume at the maximum.

The CLP is a packaging problem where a large container needs to be filled with smaller boxes. As the number of box types and the number of boxes increase, the variety of settlements increases gradually. These types of problems are defined as NP-Hard problems which require exponential time to solve. Therefore, heuristics and meta-heuristics are important approaches in solving NP-Hard problems [1].

Heuristic algorithms imitate natural phenomena to accomplish any purpose. There is no guarantee that the best solution will be found. They aim to find solutions close to the best by deciding the effective ones among various alternatives. They usually reach almost the best solution efficiently. Heuristic algorithms are needed when the exact solutions of optimization problems have an undefined structure, and they can be used as a part of the process of finding the exact solution [2].

General purpose heuristic methods can be examined in six groups: Biological-based, physics-based, swarm-based, social-based, music-based and chemistry-based. The genetic algorithm, differential evolution algorithm, ant colony algorithm, cuckoo search algorithm, artificial neural networks, bee colony algorithm and artificial immune systems are biological-based methods. The imperialist competitive algorithm, parliamentary optimization algorithm and tabu search algorithm are social-based algorithms. The artificial chemical reaction algorithm is chemical-based; the harmony search algorithm is music-based; the simulated annealing algorithm, big bang big crunch algorithm, gravitational search algorithm, central force optimization, intelligent water drop algorithm and electromagnetism algorithm are physics-based heuristic methods. The particle swarm optimization is a swarm-based algorithm [3].

In this paper, a GRASP method is proposed to solve the container loading problem. The details of the proposed method are explained and other heuristic algorithms in the literature are compared with the proposed method and the results are presented in the following sections.

2. Literature Review

In literature, we can find several heuristic and meta-heuristic methods proposed to solve the container loading problem.

Parreno *et al.* proposed a GRASP method for solving the container loading problem. This method is based on a constructive block heuristic and tested on Bischoff and Ratcliff (BR) data set. Compared to parallel

*Corresponding author
E-mail address: merve.aydemir90@gmail.com

Received: 8/Dec/2022; Received in revised form: 12/Jan/2023; Accepted: 14/Jan/2023.

algorithms, it performs better on average but not for every problem of the dataset [4].

Moura and Oliveira presented a GRModGRASP algorithm based on the wall-building approach for solving container loading problem. Results were tested on BR and LN (Loh and Nee) datasets [5].

Gehring and Bortfeldt proposed a parallel genetic algorithm for the CLP with a single container. The parallel genetic algorithm follows a migration pattern. The quality of the parallel genetic algorithm has been demonstrated by the tests performed [6].

Dereli and Daş proposed two algorithms for the CLP using the ant colony optimization approach. The performances of these algorithms, whose parameters are determined by factorial design, have been tested for dataset given in the literature. The KKS-2 algorithm, one of the two proposed algorithms, gave better results than the KKS-1 algorithm [7].

Koyuncuoğlu examined customer priority and assignment rules in combination in his study. They proposed a mixed integer mathematical model for the simultaneous assignment of one or two forklifts to containers. The results of the model were analyzed using an approach based on genetic algorithm, tabu search, nearest neighbor and Lin-Kernighan heuristics [8].

Ceschia and Schaerf used local search methodologies as a solution technique in their study. Local search works on arrays of boxes to load, and the actual load is obtained by calling a special procedure the loader at each iteration. The loader places the boxes in the container using a deterministic heuristic that generates a suitable load according to the constraints [9].

Can and Sahingoz used simulated annealing meta-heuristics in their study. Their method was compared with Bortfeldt's test scenarios of the CLP [10].

Sheng *et al*. proposed a method using simulated annealing and tree-graph search procedure to solve the container loading problem. They examined the results by testing the obtained results with the BR datasets [11].

With the algorithm proposed by Zhou and Liu, it regards the loading arrangement as the position of individuals in the swarm, and as individuals interact with each other and with loading constraints, most of them will meet with the good ones and eventually stop at the best position. The algorithm can solve the three-dimensional container loading problem with or without pallets [12].

3. Material and Methods

3.1. Container loading problem

The container loading problem is solved in a three-dimensional space consisting of the X, Y, Z axis as shown in **Figure 1**. The X axis is parallel to the right and left directions, the Y axis is parallel to the back and front directions, and the Z axis is parallel to the up and down directions. The point where the three points intersect (0, 0, 0) is the origin point.

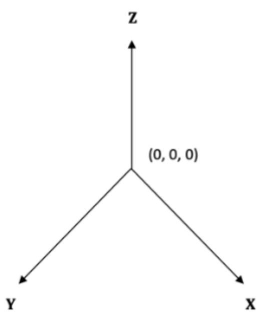

Figure 1. *X Y Z axis*

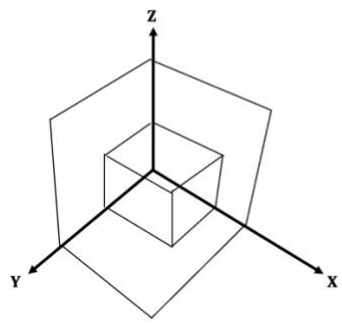

Figure 2. *Box placement in container*

As in **Figure 2**, there are *n* boxes of different sizes with width w_i, length l_i, and height h_i that are intended to be placed inside a container of width W, length L, and height H. Various constraints are defined for the container loading problem. They can be grouped into two categories: Mandatory and non-mandatory constraints [13]. We list the main constraints for the container loading problem as follows.

- All boxes must be completely inside the container.
- There must be no overlap between boxes.
- Each box must be placed parallel to the boundary surface of the container [14].

We can name the basic constraints as overflow, intersection and support constraints. In the overflow constraint, it is checked whether the dimensions of the box to be placed overflow from the container.

$$0 \leq xi \leq W - wi \tag{1}$$

$$0 \leq yi \leq L - li \tag{2}$$

$$0 \leq zi \leq H - hi \tag{3}$$

In Eqs. (1)-(3), xi is the width of the box to be placed; yi is the length of the box to be placed; zi is the height of the box to be placed. If inequality in Eqs. (1)-(3) are not satisfied, the box overflows the container. In the support constraint, it is checked whether the box to be placed is supported by the container floor or other boxes.

$$(yj \leq yi \leq yj + lj) \wedge (xj \leq xi \leq xj + wj) \tag{4}$$

$$(yj \leq yi \leq yj + lj) \wedge (xj \leq xi + wi \leq xj + wj) \tag{5}$$

$$(yj \leq yi + li \leq yj + lj) \wedge (xj \leq xi \leq xj + wj) \tag{6}$$

$$(yj \leq yi + li \leq yj + lj) \wedge (xj \leq xi + wi \leq xj + wj) \tag{7}$$

$$(yj \leq (yi + li)/2 \leq yj + lj) \wedge (xj \leq (xi + wi)/2 \leq xj + wj) \tag{8}$$

In Eqs. (4)-(8), xj is the width of the placed box; yi is the length of the placed box. If the box to be placed is on the container floor, the box is supported, but if the box to be placed is not on the container floor, the box is supported if at least three of the five inequalities in Eqs. (4)-(8) are met. In intersection constraint, it is checked whether the box to be placed intersects with the other boxes that are placed.

$$(xj - xi \geq wi) \vee (xi - xj \geq wj) \vee (yj - yi \geq li) \vee (yi - yj \geq lj) \vee (zj - zi \geq hi) \vee (zi - zj \geq hj) \tag{9}$$

It is checked whether the inequality given in Eq. (9) is satisfied between the box to be placed and all the boxes that are placed. If at least one of the inequalities is not satisfied, an intersection has occurred between the boxes.

3.2. GRASP algorithm

The GRASP is a multi-start or iterative process that was developed by Feo and Resende for solving hard combinatorial problems [15]. Each GRASP iteration contains a construction phase where a feasible solution is constructed, and a local search phase which works iteratively by replacing the current solution with a better solution found in the neighborhood of the current solution. Each solution constructed in the constructive phase is used in the local search phase as a starting solution. The best overall solution is returned as the result.

In the construction phase, a complete solution is iteratively constructed starting from an empty solution, by adding one element at a time. At each construction step, the greedy function measures the benefit of adding each element in a candidate list and they are sorted by their benefits. At each iteration of the construction step, these measurements are updated with the change brought about by selecting the previous item. Therefore, the heuristic is adaptive. The randomized component of a GRASP is obtained by random selection of one of the best candidates on the list. However, this is not always the best candidate. The list that contains only the best candidates. This list is named as the restricted candidate list (RCL) [16].

The solution generated in the GRASP construction phase cannot be guaranteed to be a locally optimal solution. Therefore, a local search is needed to try to improve each solution from the construction phase. A local search algorithm iteratively replaces the current solution with a better solution in the neighborhood. It stops when local search fails to find a better solution in the neighborhood. If there is no better solution in terms of the objective function values in the neighborhood, this solution is considered as the local optimum [17].

3.3. BB-GRASP method

In this paper, a block-building based GRASP method is proposed for solving the container loading problem. Six key elements are used as the block-building approach defined in [18]. The details of these six key elements are as follows:

(K1) **cover representation** is used to represent empty space
(K2) **general blocks** are used to construct solutions

(K3) free space is selected with **smallest Manhattan distance**
(K4) the evaluation function $VCS_{\alpha}(b,r)$ is used to select blocks
(K5) blocks are placed at the **anchor corner**
(K6) as a search strategy, the proposed **BB-GRASP method** is used

K1, represents the empty space inside the container. When any box is placed at the corner of the container, the residual empty space is polyhedron. Three overlapping cuboids are used to represent the residual space. Each cuboid is made of a large rectangular box that is internally discrete from the placed block, and this is called cover representation.

K2 includes the construction of blocks. The general block consists of boxes of different types. It can also contain boxes that are located in different directions.

In K3, Manhattan distance is employed to select the next residual space to be placed. The corner with the smallest distance is corresponding to the anchor corner of cuboid, and the residual space with the smallest Manhattan distance is selected for placing the selected blocks.

In element K4, the evaluation function shown in Eq. (10) is used to decide which block to choose for placing as defined in [19].

$$VCS_{\alpha}(b,r) = V(b) \times CS(b)^{\alpha} \tag{10}$$

Given a residual space r, each b block is sorted using Eq. (10). $V(b)$ represents the total volume of the boxes in block b. $CS(b)$ considers that a b block of dimensions $(l \times w \times h)$ is located at coordinate (x, y, z) of the container. Next, the covered surface area of block b by adjacent blocks and container walls are calculated. Then, normalized cover surface of block, $CS(b)$ in the range [0, 1], is computed as the covered surface of the block over the total block surface. α is a parameter that gives weight to the CS function.

In element K5, the blocks are placed in the following order: Near the corners, the edges, the faces of the container.

In element K6, as a search strategy, the GRASP method is used. The GRASP method has two stages: (1) Construction stage, (2) Local search stage. A solution is created by placing the blocks in the container step by step during the GRASP construction phase. Which block will be selected is determined by an evaluation function. A randomization strategy has been added to obtain different solutions. At each block selection, $VCS_{\alpha}(b,r)$ is calculated as the evaluation function for each of the blocks that can be placed in the residual space. $VCS_{\alpha}(b,r)$ provides an assessment of the block's fit to the residual space. A block is randomly selected among the blocks in the RCL with the highest evaluation function value among the blocks that can be placed at each step. RCL is created with the parameter δ ($0 < \delta < 1$) from within the blocks that can be placed. The $100\delta\%$ blocks with the highest $VCS_{\alpha}(b,r)$ value is included in the list. This iteration continues until there are no more blocks that can be placed. This procedure corresponds to the element K4 in the block-building method. In the local search phase, a neighboring solution is obtained similar to a solution created during the construction phase. At this phase, the last placed $k\%$ blocks in the solution are removed from the container. The GRASP construction phase is repeated for the remaining space, but this time RCL consists of the best *m* blocks in that step.

During *t* time, GRASP construction phase and local search phase are repeated to find new solutions. The objective function of a solution is the ratio of total volume of the placed boxes to the volume of the container. The highest volume ratio is the output of the algorithm.

4. Results and Discussion

The proposed method is implemented in C++ language. Experiments were performed on a computer with an Intel Core i5 processor - 2.3 GHz Quad Core and 8 GB 2133 MHz memory.

BR datasets are used to solve this problem [20], [21]. The BR datasets consist of 15 tests. Since there are 100 problems in each dataset, there are 1500 problems in total. **Table 1** shows the variety of box types that increases from BR1 to BR15.

Table 1. *Box Types in Datasets*

Datasets	Number of Box Types
BR-1	3
BR-2	5
BR-3	8
BR-4	10
BR-5	12

BR-6	15
BR-7	20
BR-8	30
BR-9	40
BR-10	50
BR-11	60
BR-12	70
BR-13	80
BR-14	90
BR-15	100

The parameters used in the proposed BB-GRASP method are presented in **Table 2**.

Table 2. *Parameters for BB-GRASP Method*

t	10 seconds
δ	0.05
k	50
m	2
Maximum number of solutions	1000

The results of the BB-GRASP method are displayed in Table 3.

Table 3. *Results for BB-GRASP Method*

Dataset	Minimum Volume Ratio (%)	Maximum Volume Ratio (%)	**Average Volume Ratio (%)**
BR1	84.07	97.01	**93.46**
BR2	89.14	95.82	**93.51**
BR3	91.39	95.55	**93.58**
BR4	91.50	95.52	**93.34**
BR5	91.40	94.65	**93.17**
BR6	92.01	94.28	**93.11**
BR7	91.19	93.73	**92.60**
BR8	91.10	93.41	**92.38**
BR9	91.00	93.44	**92.27**
BR10	90.85	93.30	**92.02**
BR11	90.66	92.98	**91.88**
BR12	90.96	92.56	**91.80**
BR13	90.98	92.88	**91.80**
BR14	90.65	92.52	**91.63**
BR15	90.64	92.84	**91.62**

Table 3 shows the minimum, maximum and average volume ratio results of the BB-GRASP method. In the results, the maximum volume ratio varies between 93.58% and 91.62% for the BR datasets. Although different results are seen for different box types in minimum and maximum values, when the average is looked at, it is seen that there is a decrease in the maximum volume ratio as the box type increases.

Table 4. *Comparison with other heuristics*

Datasets	**BB-GRASP**	GRASP [3]	GRModGRASP [4]	SOA [11]	TS_CLP [10]
BR1	**93.46**	93.27	89.07	92.67	90.62
BR2	**93.51**	93.38	90.43	93.19	91.51
BR3	**93.58**	93.39	90.86	93.44	92.43
BR4	**93.34**	93.16	90.42	93.21	92.35
BR5	**93.17**	92.89	89.57	92.94	92.45
BR6	**93.11**	92.62	89.71	92.70	92.37
BR7	**92.60**	91.86	88.05	92.31	92.13

BR8	**92.38**	91.02	86.13	91.92	91.95
BR9	**92.27**	90.46	85.08	91.50	91.64
BR10	**92.02**	89.87	84.21	91.23	91.42
BR11	**91.88**	89.36	83.98	90.85	91.14
BR12	**91.80**	89.03	83.64	90.59	90.98
BR13	**91.80**	88.56	83.54	90.17	90.60
BR14	**91.63**	88.46	83.25	89.70	90.27
BR15	**91.62**	88.36	83.21	89.20	89.84

In **Table 4**, the BB-GRASP method is compared with the reported results of the SOA [11] which uses swarm optimization problem, TS_CLP [10] which uses simulated annealing algorithm, GRASP [3] and GRModGRASP [4] which uses the Grasp method in the literature. It is seen that better results are obtained for each dataset. Compared to other results, an improvement of at least 0.19% for BR1, 0.13% for BR2, 0.14% for BR3, 0.13% for BR4, 0.23% for BR5, 0.41% for BR6, 0.29% for BR7, 0.43% for BR8, 0.63% for BR9, 0.6% for BR10, 0.74% for BR11, 0.82% for BR12, 1.2% for BR13, 1.36% for BR14 and 1.78% for BR15 is observed. Even if the maximum volume ratio decreases as the box types increase, it is seen that a better volume ratio is obtained compared to other heuristics.

5. Conclusion

We proposed a block-building based GRASP method for the container loading problem. The block-building method was used for the placement of boxes and the GRASP method was applied as the search strategy. The performance of the proposed method was evaluated on the BR datasets and the results were compared with the results of some heuristic algorithms applied in the literature. According to the results, minimum 0.13% and maximum 1.78% improvement were obtained. The proposed method can help to reduce the costs of transportation which is crucial for domestic and international markets. Reducing the cost will increase the satisfaction between customers and organizations to higher levels. As a future work, some constraints that can be used in the CLP can be included or other heuristic algorithms can be used as a search strategy to increase the maximum volume ratio.

References

[1] Sheng L, Hongxia Z, Xisong D, Changjian C. "A Heuristic Algorithm for Container Loading of Pallets with Infill Boxes ". European Journal of Operational Research 252 (2016) 728-736.

[2] Karaboğa D. Yapay Zeka Optimizasyon Algoritmaları. Nobel Yayın Dağıtım, 2011.

[3] Alataş B. Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme. Doktora Tezi, Fırat Üniversitesi, Elazığ, Türkiye, 2007.

[4] Parreno F, Alvarez-Valdes R, Tamarit JM, Oliveira JF. "A maximal-space algorithm for the container loading problem ". INFORMS Journal on Computing, 20(3) 2008 412-422.

[5] Moura A, Oliveira JF. "A GRASP approach to the container-loading problem ". IEEE Intelligent Systems, 20(4) 2005 50-57.

[6] Gehring H, Bortfeldt A. "A Parallel Genetic Algorithm for Solving the Container Loading Problem ". International Transactions in Operational Research 9, 2002 497-511.

[7] Dereli T, Daş GS. "Konteyner Yükleme Problemleri için Karınca Koloni Optimizasyonu Yaklaşımı ". Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi, 25(4) 2010 881-894.

[8] Koyuncuoğlu MU. Bir Konteyner Terminalinde İstif Vinçlerinin Meta Sezgisel Yöntemler Kullanarak Çizgelenmesi. Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli, Türkiye, 2012.

[9] Ceschia S, Schaerf A. "Local search for a multi-drop multi-container loading problem ". Journal of Heuristics, 19(2) 2013 275-294.

[10] Can O, Sahingoz OK. "Solving container loading problem with simulated annealing algorithm ". 15th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, 19-21 November 2014.

[11] Sheng L, Xiuqin S, Changjian C, Hongxia Z, Dayong S, Feiyue W. "Heuristic Algorithm for the Container Loading Problem with Multiple Constraints ". Computers & Industrial Engineering 108, 2017 149-164.

[12] Zhou Q, Liu X. "A Swarm Optimization Algorithm for Practical Container Loading Problem ". IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 Oct – 1 Nov 2017.

[13] Gehring H, Bortfeldt A. "A Genetic Algorithm for Solving the Container Loading Problem ". International Transactions in Operational Research 4, 1997 401-418.

[14] Huang Y, Hwang FJ, Lu H. "An effective placement method for the single container loading problem ". Computers & Industrial Engineering 97, 2016 212-221.

[15] Feo T, Resende MGC. “A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem “. Operations Research Letters 8, 1989 67-71.

[16] Feo T, Resende MGC. “Greedy Randomized Adaptive Search Procedures “. Journal of Global Optimization 6, 1995 109-133.

[17] Festa P, Resende MGC. “GRASP: basic components and enhancements “. Telecommunication Systems, 46(3) 2011 253-271.

[18] Zhu W, Oon W, Lim A, Weng Y. “The six elements to block-building approaches for the single container loading problem “. Applied Intelligence 2012, 37 2012 431-445.

[19] Araya I, Guerrero K, Nunez E. “VCS: A new heuristic function for selecting boxes in the single container loading problem. “. Computers & Operations Research 82, 2017 27-35.

[20] Bischoff EE, Ratcliff MSW. “Issues in the development of Approaches to Container Loading “. Omega-International Journal of Management Science, 23(4) 1995 377-390.

[21] Davies AP, Bischoff EE. “Weight distribution considerations in container loading “. European Journal of Operational Research, 114(3) 1999 509-527.