OPTIMIZATION ERUDITORUM
Volume 2 (2025), No. 1, 70–84
DOI: https://doi.org/10.69829/oper-025-0201-ta06
Tulipa Opera Scholarum

AUGMENTED LAGRANGIAN-BASED HYBRID SUBGRADIENT METHOD FOR ONE-DIMENSIONAL CUTTING STOCK PROBLEM

BANU ICMEN ERDEM[1] AND REFAIL KASIMBEYLI[*,1,2]

[1] *Department of Industrial Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir 26555, Turkey*
[2] *UNEC Mathematical Modeling and Optimization Research Center, Azerbaijan State University of Economics (UNEC), Istiqlaliyyat Str. 6, Baku, 1001, Azerbaijan*

Abstract. This study proposes an enhanced version of the feasible value-based modified subgradient algorithm incorporating two novel placement heuristics for the one-dimensional cutting problem. These heuristics, the Sequential, Controlled, and Pre-Processed Placement heuristic and the Dynamic Placement heuristic are integrated within the algorithm to improve its performance. Additionally, the approach is further refined by combining it with simulated annealing in a hybrid framework. We evaluate the effectiveness of the proposed approach through computational experiments on benchmark problems. The results show that the feasible value-based modified subgradient algorithm, which integrates the Sequential, Controlled, and Pre-Processed Placement heuristic and the Dynamic Placement heuristic, successfully obtains solutions for all test problems. The algorithm achieves optimal solutions using the Dynamic Placement heuristic and outperforms the Sequential, Controlled, and Pre-Processed Placement heuristic in both solution quality and computational efficiency. While the Sequential, Controlled, and Pre-Processed Placement heuristic fails to reach optimal solutions, it enables reasonable runtimes. Conversely, the Dynamic Placement heuristic leads the algorithm to optimal solutions, though it requires longer computation times with the initial parameter settings. This study shows how adaptable and successful the hybrid method and placement heuristics are at handling complex optimization problems. It also shows how well they can solve the one-dimensional cutting problem.

Keywords. One-dimensional cutting problem, Modified subgradient algorithm, Simulated annealing, Hybrid solution approach.

1. Introduction

One-dimensional cutting stock problems are encountered in many businesses that deal with one-dimensional cuts of materials such as pipes, cables, steel shafts, paper, glass, and metal. The stock materials handled may be one-dimensional materials, such as steel shafts, or multi-dimensional materials, such as rectangles or prisms, with cuts evaluated over a single dimension (Fig. 1). The problem is classified as NP-hard in terms of computational complexity. It is challenging to solve this problem due to many decision variables, data size, and constraints when considering the larger problems.

This paper studies the one-dimensional cutting stock problem (1DCSP) with a new constraint. The primary purpose of 1DCSP is to minimize the number of used stocks while meeting the demand.

One-dimensional cutting stock problems have been studied with many aspects since 1961 [20] in the operations research literature. Campello et al. [10] introduced a new heuristic procedure, called the Residual Recombination Heuristic, to the one-dimensional cutting stock problem. The central aspect of the heuristic involves recombining residual cutting patterns in different ways. The purpose of the

*Corresponding author.
E-mail address: bicmen@eskisehir.edu.tr (B.Icmen Erdem), rkasimbeyli@eskisehir.edu.tr (R. Kasimbeyli)
2020 Mathematics Subject Classification: 90C10, 90C27, 90C59.
Accepted: February 02, 2025.

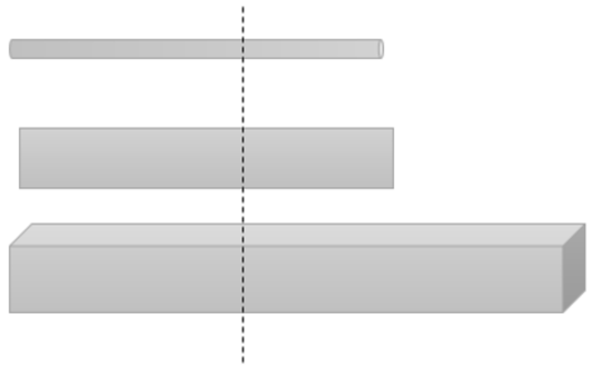

FIGURE 1. One-dimensional cutting examples

study that was proposed by Ayres et al. [1] is to present a new model for integrated lot sizing, one-dimensional cutting-stock, and two-dimensional cutting-stock problems. This integrated model can be used to look at the papermaking process as a three-phase process that begins with the manufacture of jumbos and finishes with the paper sheets we use daily.

Two-dimensional two-stage cutting stock problems are studied in [21, 22]. In these papers, a new mathematical model is proposed, and a new matheuristic solution method is developed. The new method is compared to hybrid metaheuristics. The authors examined the production planning of hollow-core slabs in their work [11]. Two mathematical models were proposed for the problem, which consisted of a one-dimensional multi-period cutting stock problem with innovative aspects regarding the multiple manufacturing modes that can be used to produce the slabs. Cerqueira proposed a modification to the Constructive Greedy Heuristic that involves creating a cutting pattern by sorting items of pair or odd length in descending order, with priority given to those that appear more frequently in the problem and cutting from objects in stock with pair or odd length, to minimize the number of cut objects. Baykasoglu and Ozbel [5] studied the one-dimensional marble plane cutting problem based on the cutting equipment productivity and effective use of marble blocks. For small-size instances, a mixed integer linear programming model was developed for solving the present marble plane cutting problem. For larger size problem instances, a Stochastic Diffusion Search algorithm was developed. Ravelo et al. [38] considered the one-dimensional cutting stock problem in which the non-used material in the cutting patterns may be used in the future if large enough. They gave a heuristic algorithm and two meta-heuristic approaches to the problem. The authors proposed an integer linear programming model for the one-dimensional cutting stock and scheduling problem with heterogeneous orders in their work [37]. They also suggested a novel matheuristic algorithm based on a fix-and-optimize strategy hybridized with a random local search. Wei et al. [47] proposed a new branch-and-price-and-cut algorithm to solve the one-dimensional bin-packing problem. The exact algorithm proposed in that paper was based on the classical set-partitioning model for the one-dimensional bin-packing problems and the subset row inequalities. Burke et al. [9] illustrated how improvements in solution quality could be achieved by the hybridization of the best-fit heuristic together with simulated annealing and the bottom-left-fill algorithm. Chen et al. [13] presented a new simulated annealing approach to solving an integer linear programming formulation of the one-dimensional cutting stock problem. They also discussed a thorough statistical analysis of the effects of various parameters on the efficiency and accuracy of solutions. Li et al. [35] proposed a general particle swarm optimization based on the simulated annealing algorithm for the solution to the multi-specification one-dimensional cutting stock problem. Leung et al. [34] presented a two-stage intelligent search algorithm for a two-dimensional strip packing problem without a guillotine constraint. In the first stage, a heuristic algorithm was proposed. In the second stage, a local search and a simulated annealing algorithm were combined to improve solutions to the problem. In this paper, we studied 1DCSP with a new constraint. This new constraint allows assigning items not wanting to be cut together to different stocks. We propose an exact solution method based on the sharp augmented Lagrangian duality scheme for 1DCSP.

Lagrangian relaxation methods are well-known in the literature. The Lagrangian relaxation strategy is arguably one of the most widely used in large-scale integer programming problems[15]. The Lagrangian relaxation approach is based on the formulation of a Lagrangian dual problem in which some chosen constraints are added to an objective function with a possible penalty term that includes the amount of violation of the constraints and their dual variables [15, 19]. When the classical Lagrangian function is used, the problem becomes nonconvex (and nonsmooth) due to integer variable constraints, and thus it is possible to encounter a duality gap. As a result, selecting the appropriate Lagrangian function to ensure a zero duality gap becomes an essential concern in developing methodologies for nonconvex-constrained optimization problems. In the literature, the sharp augmented Lagrangian duality scheme was used to construct solution methods for constrained optimization problems, where the zero duality gap condition is formed for a wide range of nonconvex problems [2, 3, 17, 18, 31, 32, 39, 16, 40, 41, 14]. The method suggested in this study is based on an improved version of Gasimov's Modified Subgradient (MSG) Algorithm [17]. At each iteration of the MSG method, an unconstrained subproblem is generated, and the global optimal solution is used to update dual variables. The MSG algorithm was proved to provide a strongly monotonically growing sequence of dual values that converges to the standard primal-dual optimal value.

Kasimbeyli et al. introduced the feasible value-based modified subgradient (FMSG) algorithm, an upgraded version of the MSG algorithm. It does not require a global optimal solution to the subproblem at each iteration to update dual variables [25]. Many researchers used this method in the literature to solve various nonconvex optimization problems and develop new solution methods [4, 8, 7, 36, 42, 46, 44].

Because it is unnecessary to (globally) minimize the augmented Lagrangian in the FMSG algorithm at each iteration, we combine this approach with the simulated annealing (SA) metaheuristic to solve the proposed model for 1DCSP in this study. In our method, we first construct the sharp augmented Lagrangian with no length constraints and then use the SA to minimize it. If the produced solution satisfies the length constraint, the procedure is terminated because the acquired solution is theoretically optimal. Otherwise, we formulate the constraint using dual variables (Lagrange multipliers) to add it to the sharp augmented Lagrangian function, which we then minimize using the SA. According to the FMSG algorithm's rules, the dual variables are updated at each iteration using the current solution found by minimizing the updated augmented Lagrangian. This process is repeated until there is no longer any unsatisfied length, at which point the augmented Lagrangian does not need to be updated.

This study proposed two novel placement heuristics for the one-dimensional cutting problem. These placement heuristics are integrated into a new hybrid approach combining the FMSG and SA algorithms. The proposed solution approach successfully achieved optimal solutions for all test problems. Highlights of the study are as follows:

- Two novel placement heuristics were developed for the one-dimensional cutting problem.
- A hybrid solution approach combining the FMSG and SA algorithms was proposed.
- The proposed heuristics achieved optimal solutions for all test problems.
- Computational experiments demonstrated significant time efficiency with the placement heuristics.
- The proposed method outperformed the traditional mathematical model under specific constraints.

The following is how the paper is structured. Section 2 describes the definition and the formulation of the problem under study. Section 3 discusses the solution method for solving the 1DCSP. Section 4 provides computational results on test instances obtained from the literature. These findings are used to analyze and assess the solution method's performance, and Section 5 concludes with a summary of the work and recommendations for future research.

2. Problem Definition and Mathematical Model

The most common mathematical model [20] used for solving one-dimensional cutting stock problems is based on creating cutting patterns in the model. All possible combinations are created before the problem by considering the dimensions of the items requested from a stock. These combinations are called cutting patterns and are treated as parameters in the model. For the solution of this model, cutting patterns must be determined in advance. When the demand list increases, the number of cutting patterns to be created will increase, and it will be challenging to derive each cutting pattern. Hence, it is complicated to solve, especially in large-sized problems. Therefore, the column generation method [20] is used for this problem.

Kasimbeyli et al. [25] studied one-dimensional cutting stock problems without using previously defined cutting patterns. They proposed a mathematical model that does not require a cutting pattern. Since it approaches the problem's solution from a different angle, it provides a new perspective. We give a mathematical model for the one-dimensional cutting stock problem with a new constraint by referencing this model. We accept every item's demand as one in this model, and if the item's demand is more extensive than one, we multiply each item and consider each one a unique item. The set and parameters used in the model are defined as follows:

Set and parameters

n	be the number of items;
m	be the number of stocks;
$i = 1, \dots, n$	set of items;
$j = 1, \dots, m$	set of stocks;
l_i	be the length of the item i;
L	be the length of the stock;

Decision variables

$$x_{ij} = \begin{cases} 1, & \text{if item } i \text{ is cut stock } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_j = \begin{cases} 1, & \text{if stock } j \text{ used,} \\ 0, & \text{otherwise.} \end{cases}$$

Mathematical model

$$\min \sum_{j=1}^{m} y_j \tag{2.1}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} = 1, \ \forall i, \tag{2.2}$$

$$\sum_{i=1}^{n} l_i x_{ij} \leq L y_j, \forall j, \tag{2.3}$$

$$x_{ij} \leq y_j, \quad \forall i, \forall j, \tag{2.4}$$

$$x_{ij} \in \{0,1\}, y_j \in \{0,1\}, \forall i, \forall j. \tag{2.5}$$

The objective function (2.1) minimizes the total number of stocks used. Constraint set (2.2) ensures that the demand for every item has to be met. Constraint set (2.3) imposes the satisfaction of the length restriction for every stock j. Constraint set (2.4) ensures that if stock j is not used, then no item can be assigned to that stock. Finally, the constraint set (2.5) imposes that all decision variables are binary.

Although solving this model with the GAMS solver provides effective solutions, it may fail to find the optimal solution as the problem size increases. Since this is an NP-hard problem, computation time grows rapidly. To overcome this difficulty, we propose a new hybrid solution method and new placement heuristics.

The method for solving the problem described in this section is presented in the following section.

3. Method

The solution method for the 1DCSP, as defined in the previous section, is explained in this section. As previously stated, this method combines the FMSG algorithm, which is based on the sharp augmented Lagrangian duality scheme, and the SA, which is used to solve subproblems.

Now we will go through why this combination is employed in this work and the benefits of the hybrid method. Lagrangian relaxation is a well-known method for solving large-scale combinatorial optimization problems because it allows the creation of a reasonably simple new model by adding (one or more) constraints to the objective function using Lagrange multipliers [15]. The Lagrangian relaxation strategy produces a dual problem, and if its optimal value equals the optimal value of the original (primal) problem, there is no duality gap; otherwise, the difference between the primal and dual optimal values becomes substantially positive, resulting in the duality gap. When a Lagrangian relaxation is utilized, one of the primary issues must be addressed is ensuring that the duality gap vanishes. The zero duality gap criteria, or conditions that guarantee the vanishing of the duality gap, are highly dependent on the Lagrangian function used to describe the related dual problem. The reason for this is that finding supporting points of a set (that is, a set of feasible values of the primal problem, which is the image of the set of feasible solutions under the objective and constraint functions) with respect to the surfaces generated by a Lagrangian function is simplified when solving a dual problem. Because the classical Lagrangian function generates hyperplanes, which can only be used to support convex sets, the accompanying duality scheme may encounter a duality gap when the primal problem is not convex [39, 6]. Because the sharp augmented Lagrangian function produces conical surfaces, it can be utilized as a supporting surface for nonconvex sets, resulting in a duality scheme in which the zero duality gap conditions for a wide range of nonconvex problems are demonstrated (see, e.g.[39, 14]). It should be noted that conical supporting surfaces have been used in the literature to examine a wide range of nonconvex optimization problems and establish optimality conditions (see, e.g.[23, 24, 26, 27, 28, 29, 30]).

All of the theorem conditions ensuring the zero duality gap, recently demonstrated in ([14], Theorem 7), are satisfied for the 1DCSP since the objective and constraint functions are all linear and the decision variables are confined to be binary (0 or 1). This theorem enables us to build a solution strategy for the 1DCSP, a nonconvex and nonsmooth problem, based on a sharp augmented Lagrangian duality scheme (because of integer variables used in the mathematical model). This method is a hybrid subgradient method since it combines the FMSG algorithm [32] and the SA metaheuristic. In this hybrid method, dual variables are updated using the stepsize rule of the FMSG algorithm [4], and subproblems are solved using the SA algorithm at each iteration. A sharp augmented Lagrangian function is used to handle the length limitations. The other constraints are addressed using the SA algorithm because they are no longer challenging to solve using this method. The following are some of the advantages of the hybrid method:

- It is quite simple to put in place;

- The FMSG algorithm does not require global minimums of the subproblems that must be solved at each iteration and instead converges to a global minimum of the original problem (see ([32], Theorems 3.4, 3.6, 3.7) and ([4], Theorems 4.1, 4.2)).

Following a brief description of the sharp augmented Lagrangian and the Simulated Annealing metaheuristic in the following subsections, we discuss the developed hybrid algorithm in Section 3.3 in detail.

3.1. **Sharp augmented lagrangian.** In this part, we briefly review several relevant definitions for the primal problem, which is a general constrained optimization problem:

$$(P) \qquad \text{minimize} \quad f(x) \quad \text{subject to} \quad x \in X, \tag{3.1}$$

where $X = \{x : g(x) = 0\}$ is a compact set of feasible solutions and functions $f : \mathbb{R}^n \to \mathbb{R}, g : \mathbb{R}^n \to \mathbb{R}^p$ are continuous. Denote by $\mathbb{R}_+$ the set of non-negative numbers. The sharp augmented Lagrangian $L : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ associated with (3.7) is defined as follows:

$$L(x, u, c) = f(x) + c\|g(x)\| - \langle u, g(x)\rangle, \tag{3.2}$$

where $x \in \mathbb{R}^n, u \in \mathbb{R}^p, c \in \mathbb{R}_+, \|g(x)\|$ denotes the Euclidean norm of the constraint function $g(x) = (g_1(x), ..., g_p(x))$ and $\langle u, g(x)\rangle = \sum_{i=1}^n u_i f_i(x)$. The solution set of problem (3.7) is denoted by $Sol(P)$. Typically, an element of $Sol(P)$ is denoted by $\bar{x}$. The dual function $H : \mathbb{R}^p \times \mathbb{R}_+ \to \mathbb{R}$ is defined as:

$$H(u, c) = \min_{x \in X}[f(x) + c\|g(x)\| - \langle u, g(x)\rangle], \tag{3.3}$$

where u and c are dual variables (or Lagrange multipliers). Then the dual problem to (3.7) is given by

$$(P*) \quad \text{maximize} \quad H(u, c) \quad \text{subject to} \quad u \in \mathbb{R}^p, c \in \mathbb{R}_+. \tag{3.4}$$

3.2. **Simulated annealing.** Simulated annealing (SA) was proposed by S. Kirkpatrick et al. [33] and V. Cerny [12] for various optimization problems. In the 1980s, SA was used extensively as an efficient and straightforward method for solving combinatorial optimization problems. SA is based on the principle of obtaining a strong crystal structure by first heating the material and then cooling it slowly. The strength of the structure depends on the rate and time of cooling of the metal. If the initial temperature is not high enough or a rapid cooling schedule is applied, defects occur in the material. In this case, the cooled solid cannot reach fundamental equilibrium at every temperature level. Strong crystals can be obtained by careful and slow cooling [45].

The algorithm examines different candidate solutions, starting from a determined initial solution and progressing through many steps. At each step, a neighbor solution is created specifically for the problem. Neighboring solutions ($f(s)$) that improve the objective function value are always accepted. The chance of choosing non-healing neighbor solutions ($f(s')$) depends on the current temperature (T) of the algorithm and the change in the objective function(ΔE). Accepting candidate solutions that do not improve the objective function prevents the algorithm from getting stuck in the local best. As the current temperature decreases as the algorithm progresses, the probability of accepting candidate solutions that do not improve the objective function decreases. The probability of acceptance of bad candidate solutions is calculated using the Boltzmann distribution:

$$P(\Delta E, T) = e^{-\frac{f(s') - f(s)}{T}}. \tag{3.5}$$

By making a certain number of changes at each temperature level, candidate solutions are created and examined. When a certain number of candidate solutions are examined, the equilibrium state is reached, and the temperature is gradually reduced. The temperature reduction is made according to a specific cooling schedule [45].

In the SA algorithm, we represent candidate solutions for 1DCSP by a permutation of whole items as an array that depicts the order in which the selected placement heuristic assigns items to stocks.

The algorithm starts from an initial solution. Different approaches can create the initial solution. The most common initial solution approaches are random and descending order by length. In this paper, these two initial solution creations are used to compare.

The SA algorithm examines different candidate solutions to find the final result. Every candidate solution created at every iteration is called a neighbor solution. In this paper, neighbor solutions are created using a 2-opt exchange operator. With the 2-opt exchange operator, we choose two items randomly from the array of items and then exchange their positions. We send each candidate solution to the selected placement heuristic. With the placement heuristic, the objective function value, the number of used stock, is calculated, and the final item placement, a cutting pattern, is obtained. Proposed placement heuristics are described in section 3.3.

SA algorithm is explained in Algorithm 1 given below.

Algorithm 1 Algorithm of simulated annealing

Require: Cooling schedule ($g(T)$), Initial temperature (T_{max})
Create an initial solution (s_0)
Ensure: $T = T_{max}$ $\quad s = s_0$
while $T \neq T_{stop}$ **do**
  while $i < N$ **do**
    Create candidate solution (s') of s
    $\Delta E \Leftarrow f(s) - f(s')$,
    if $\Delta E \leq 0$ **then**
      $s \Leftarrow s'$
    else
      if $P(\Delta E, T) < \rho$ **then**
        $s \Leftarrow s'$
      end if
    end if
    $i \Leftarrow i + 1$
  end while
  $i \Leftarrow 0$
  $T \Leftarrow g(T)$
end while
Feasible solution s is found

The SA metaheuristic is effective in a variety of well-known combinatorial optimization problems. The majority of problems for which SA metaheuristics have proven effective fall into the following categories: cutting stock, vehicle routing, and assignment problems [43].

3.3. **Placement heuristics.** Depending on the metaheuristic utilized and the problem being addressed, there are a variety of placement heuristics in the literature. Placement heuristics established to reflect the characteristics of the problem and specialized under the restrictions of the problem play a vital role in the solution's success. The suggested 1DCSP placement heuristics allow candidate solutions to be assigned to stock based on specified rules. Candidate solutions are then turned into meaningful cutting patterns that show the items' beginning and finishing places and their sequences.

In this paper, we propose two different placement heuristics for 1DCSP and compare the ability to obtain solutions. The following are the placement heuristics we use:

- Sequential, controlled, and pre-processed placement heuristic
- Dynamic placement heuristic

3.3.1. *Sequential, controlled, and pre-processed placement heuristic.* The sequential, controlled, and pre-processed placement (SCPP) heuristic is based on creating doubles of items with zero trim loss. Because of this pre-processing, the number of items in the sequence reduces by pre-assigning zero-trim loss items to stocks, thereby reducing the sequence. After pre-processing, the remaining items are considered in the given order. Items are assigned to the stock by considering the given length. If an item does not fit the stock, the first item that does fit the remaining length is assigned. SSCP heuristic is described in Algorithm 2 given below.

Algorithm 2 Algorithm of sequential, controlled, and pre-processed placement heuristic

Step 1 The doubles of items with zero trim loss are assigned to the stocks by starting the first item in the items' sequence. These doubles construct a cutting pattern. The remaining items at the end of Step 1 transferred to Step 2.

Step 2 We choose the first unassigned item in the sequence, place it in stock and update the remaining length. Repeat Step 2 until an item does not fit the given remaining length.

Step 3 If an item does fit the given remaining length, it is placed in the stock. Otherwise, start a new stock and repeat Step 2 and Step 3.

Step 4 If there is no item left in the sequence unassigned, terminate the algorithm.

3.3.2. *Dynamic placement heuristic.* Based on the idea of sequential, controlled, and pre-processed placement heuristics explained in the previous section, we propose a dynamic placement heuristic (DP). When considering a sequence of items, it is essential to create zero-trim item combinations first to pre-process the sequence to improve the quality of the solutions. Based on this logic, combinations with zero trim can be determined by testing combinations 2 and 3. Then the remaining items are placed in sequence and with a controlled placement heuristic.

The main challenge here is the computational intensity and time loss of examining all combinations in problems where the number of items increases. In addition, when it is considered that this process should be applied to all neighboring solutions at all steps, difficulties in implementation can be seen. To overcome this difficulty, it has been determined that clustering the items within themselves will significantly reduce the processing intensity. These sets will be created according to the dimensions of the items. The basic logic is that items within a defined set can be cut from the stock at the same amount. The number of sets (Set_{number}) is calculated by the following formulation, while L is the length of the stock and L_{min} is the length of the smallest item:

$$Set_n = \lfloor \frac{L}{L_{min}} \rfloor \tag{3.6}$$

Note that the number of items in each set depends on the problem data. There may be zero items in some sets. Consider the following problem data in Table 1. The given items are to be cut from a stock length of 150 units.

TABLE 1. Item data for the example

item id	1	2	3	4	5	6	7	8
length	26	83	67	100	91	30	77	88
item id	9	10	11	12	13	14	15	
length	71	23	81	25	60	48	50	

The sets defined for the items given in Table 1 are given in Table 2.

TABLE 2. Sets for the example

Sets	Items					
Set_1	2	4	5	7	8	11
Set_2	3	9	13			
Set_3	14	15				
Set_4						
Set_5	1	6				
Set_6	10	12				

According to Table 2, the items in Set_1 can be cut from the stock length of 150 units one time, and the items in the sets can be cut from the same stock corresponding time. We use these sets to create combinations of two or three items. As an example, two items from Set_1 can not be cut together, so we do not have to process these items for a combination of two. Considering the defined sets, the DP heuristic is described in Algorithm 3 given below.

Algorithm 3 Dynamic placement heuristic

Step 1 The doubles of items with zero trim loss are assigned to the stocks by starting the first unassigned item in the items' sequence. This combination of two construct a cutting pattern. The remaining items at the end of Step 1 transferred to Step 2.
Step 2 The triples of items with zero trim loss are assigned to the stocks by starting the first unassigned item in the items' sequence. This combination of three construct a cutting pattern. The remaining items at the end of Step 2 transferred to Step 3.
Step 3 We choose the first unassigned item in the sequence, place it in stock, and update the remaining length. Repeat Step 3 until an item does not fit the given remaining length.
Step 4 If an item does fit the given remaining length, it is placed in the stock. Otherwise, start a new stock and repeat Step 3 and Step 4.
Step 5 If there is no item left in the sequence unassigned, terminate the algorithm.

3.4. **Hybrid subgradient (HSG) algorithm.** This section will go through the HSG algorithm, which was created by merging the FMSG and SA algorithms and taking into account length limits in a unique method.

SA is a general-purpose metaheuristic algorithm that overcomes related constraints and solves such problems with minimal processing effort. Because the FMSG approach allows for local optimums (discovered by the SA algorithm) in updating stepsize parameters, length capacity limits are easily addressed by the HSG algorithm. The approach checks for any unsatisfied length constraints after obtaining a solution to the currently stated problem. If the resulting cutting pattern satisfies length constraints, the program ends. Otherwise, the sharp augmented Lagrangian function is given a length constraint.

We will define the objective, constraint functions, and the set of possible solutions over which the Lagrangian will be minimized by using the following notations.

$$\text{minimize} \quad f(x) \quad \text{subject to} \quad x \in X, \tag{3.7}$$

where $f(x) = \sum_{j=1}^{m} y_j$, $g_i^1(x) = \sum_{j=1}^{m} x_{ij} - 1, \forall i,$, $g_{ij}^2(x) = x_{ij} - y_j, \forall i, \forall j$, and

$$X = \{x = (x_{ij}, y_j) : g_i^1 = 0,\ g_{ij}^2 \leq 0,\ (x_{ij}, y_j) \in \{0, 1\},\ \forall i,\ \forall j\}. \tag{3.8}$$

We define the length constraint (2.3) as the following form to add the sharp augmented Lagrangian function:

$$g_j^3(x,y) = \max_j \{\sum_{i=1}^{n} l_i x_{ij} - Ly_j, 0\} \forall j.$$

We describe the HSG algorithm below, where $\bar{H}$ denotes the optimal value of the (primal) problem.

Algorithm 4 The HSG algorithm

Step 0 (Initialization) Choose the tolerances $\varepsilon_1, \varepsilon_2 > 0$, numbers $\alpha > 0$, $0 < \delta < 2$, $\Delta > 0$. Set $T = 0, p = 0, q = 0$ and go to Step 1.

Step 1 Set $m := 0, k := 0$, choose the initial Lagrange multipliers $(u_k^m, c_k^m) \in R^T \times R_+$, the number H^m and define the augmented Lagrangian function as follows:

$$L(x, u_k^m, c_k^m) = f(x) + c_k^m \|g^3(x)\| - \langle u_k^m, g^3(x) \rangle, \tag{3.9}$$

and go to Step 2.

Step 2 For the given (u_k^m, c_k^m) and H^m, solve the following problem by applying the SA algorithm:

$$\text{minimize } L(x, u_k^m, c_k^m) \text{ subject to } x \in X,\ L(x, u_k^m, c_k^m) \leq H^m. \tag{3.10}$$

Constraints (2.2), (2.4), (2.5) are solved by the SA algorithm (see the definition of the set X defined in (3.8)), where length constraints described in (2.3) are handled in the sharp augmented Lagrangian function, as previously mentioned. SA algorithm is called for a specific number of times to search for a solution to (3.10). If after this specific number of times, a solution can not be found, it is assumed that a solution to (3.10) does not exist, (which means that $H^m < \bar{H}$). If so, go to Step 5. Otherwise, let x_k^m be a solution to (3.10). If $\|g^3(x_k^m)\| \leq \varepsilon_2$ then go to Step 4, otherwise go to Step 3.

Step 3 Update dual variables as follows:

$$u_{k+1}^m := u_k^m - \alpha s_k g^3(x_k^m), \tag{3.11}$$

$$c_{k+1}^m := c_k^m + (1+\alpha) s_k \|g^3(x_k^m)\|, \tag{3.12}$$

where s_k is a positive stepsize parameter defined as follows:

$$0 < s_k = \frac{\delta\alpha(H^m - L(x, u_k^m, c_k^m))}{[\alpha^2 + (1+\alpha)^2]\|g^3(x_k^m)\|^2}. \tag{3.13}$$

Set $k := k + 1$, and go to Step 2.

Step 4 Let x_k^m be a solution to (3.10) with $\|g^3(x_k^m)\| \leq \varepsilon_2$. Set $q = q + 1, m := m + 1$. If $p = 0$ then $\Delta^{m+1} = \Delta^m$, otherwise $\Delta^{m+1} = \frac{1}{2}\Delta^m$. Set $H^{m+1} := min\{f(x_k^m, u_k^m, c_k^m), H^m - \Delta^{m+1}\}$. If $\Delta^{m+1} < \varepsilon_1$ then go to Step 6, otherwise go to Step 2.

Step 5 If the solution to the problem (3.10) does not exist, then set $p = p + 1, m := m + 1$. If $q = 0$ then $\Delta^{m+1} = \Delta^m$, otherwise $\Delta^{m+1} = \frac{1}{2}\Delta^m$. Set $H^{m+1} := H^m + \Delta^{m+1}$ and go to Step 2.

Step 6 x_k^m is a solution to (3.10) with $\|g^3(x_k^m)\| \leq \varepsilon_2$ and $\Delta^{m+1} < \varepsilon_1$.

The selection of the H^0 value is critical to the problem's solution. If this value is set, too high, feasible but not optimal solutions will be obtained. Choosing a value close to the optimal solution may give the algorithm a performance boost.

There are two loops in the HSG algorithm. The solution x_k^m, which is determined for the smallest interval containing the primal-dual optimal value and which satisfies all the length restrictions, can be regarded as an outer loop. If this is possible, the algorithm ends. If not, the augmented Lagrangian function is modified by adding length constraints to it and applying updated Lagrange multipliers, and the method then proceeds to the inner loop, which consists of Steps 2 through 5.

Figure 2 depicts a quick overview of the method.

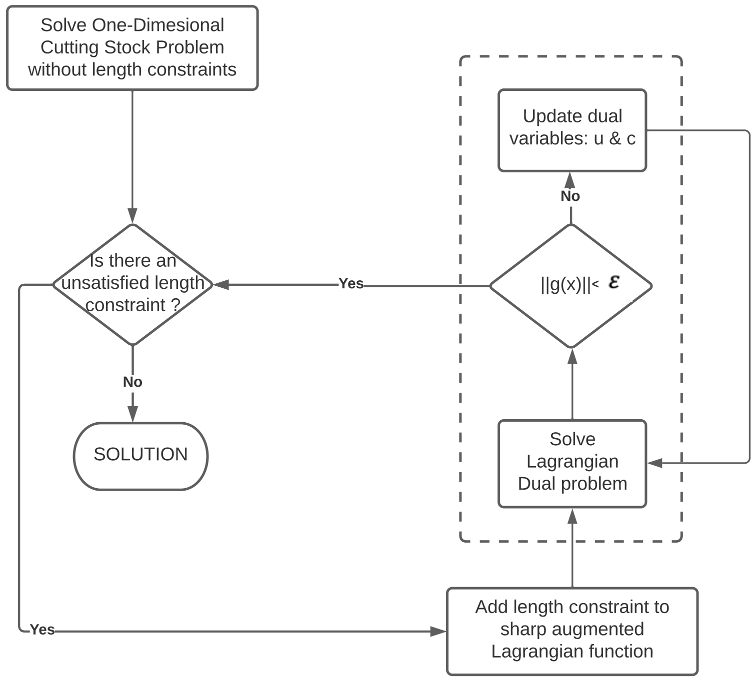

Figure 2. HSG algorithm flow chart

4. Computational Results

A series of computational experiments were conducted to verify the efficiency of the proposed solution method. To compare the performance of the HSG algorithm, all test problems are solved by the mathematical model given in Section 2 and the HSG algorithm. We use two different placement heuristics in the HSG algorithm described in Section 3.3.

The numerical experiments have been carried out on a PC with an Apple MacBook Pro with a 2.8 GHz IntelCore i7 processor and 4 GB RAM. HSG algorithm is implemented in Python 3.9 which is executed on PyCharm. We used GAMS software version 23.3 to solve the test problems with the given mathematical model. We set the CPU time limit to 2000 seconds when solving the problem with the GAMS software.

Algorithms are tested on 20 test problems given in OR-LİB with the label "binpack1". Each problem has 120 items, which will be cut from stocks with a length of 150. Parameters used in the HSG algorithm and within SA are given in Table 3.

T_{max}	$g(T)$	N	ρ		
100	0.8	10	0.25		
H^0	ε_1	ε_2	δ	α	θ
1000	0	0.001	10	0.5	0.5

Table 3. Parameters used in HSG

The comparison of experimental results obtained by the HSG algorithms and mathematical model is presented in Table 4. The Z^* column shows the optimal solutions found in the literature for the given test problems. Column z_m and column T_m show the solutions obtained with the mathematical model and the time it took to obtain these solutions.

Columns z_{HSG}^{sscp} and T_{HSG}^{sscp} represent the solutions obtained using the "Sequential, Controlled, and Pre-Processed Placement Heuristic," along with the times required to obtain these solutions. Similarly, columns z_{HSG}^{dp} and T_{HSG}^{dp} show the solutions and times obtained using the "Dynamic Placement Heuristic" within the HSG algorithm. Additional parameters used in the HSG algorithm are provided in Table 3.

ID	Z^*	z_m	T_m	z_{HSG}^{sscp}	T_{HSG}^{sscp}	z_{HSG}^{dp}	T_{HSG}^{dp}
$U120_{00}$	48	**48**	295.52	49	196.2	**48**	797.7
$U120_{01}$	49	**49**	263.35	50	155.6	**49**	437.0
$U120_{02}$	46	**46**	210.43	47	168.8	**46**	740.8
$U120_{03}$	49	50	2000.36	51	166.9	**49**	1002.9
$U120_{04}$	50	**50**	645.50	51	178.4	**50**	1199.4
$U120_{05}$	48	49	2000.24	49	170.9	**48**	875.1
$U120_{06}$	48	49	2000.22	49	166.0	**48**	797.7
$U120_{07}$	49	50	2000.23	50	156.2	**49**	916.9
$U120_{08}$	51	53	2000.13	52	257.9	**51**	1078.5
$U120_{09}$	46	47	2000.32	48	188.1	**46**	782.6
$U120_{10}$	52	53	2000.06	53	152.7	**52**	1723.1
$U120_{11}$	49	**49**	1651.96	50	171.2	**49**	1265.2
$U120_{12}$	48	49	2000.05	49	148.7	**48**	941.8
$U120_{13}$	49	**49**	156.56	50	165.7	**49**	806.3
$U120_{14}$	50	51	2000.12	51	208.5	**50**	1197.6
$U120_{15}$	48	49	2000.08	49	191.2	**48**	1344.6
$U120_{16}$	52	**52**	171.64	53	148.3	**52**	1363.8
$U120_{17}$	52	53	2000.06	54	161.7	**52**	1403.7
$U120_{18}$	49	**49**	200.59	50	152.1	**49**	1240.0
$U120_{19}$	50	**50**	2000.08	51	194.1	**50**	1244.2

TABLE 4. Computational results

As seen in the table, the optimal solution was obtained for 9 out of 20 problems using the mathematical model. However, for the remaining problems, although the upper time limit was reached, the optimal solution could not be achieved. Using the HSG algorithm with SSCP, the optimal solution could not be achieved for any of the problems. However, the solution time remained under 250 seconds, demonstrating a reasonable use of time. Finally, using the HSG algorithm with the dynamic placement heuristic (DP), the optimal solution was achieved for all problems. However, it was observed that the solution time reached up to 1400 seconds for some problems. Therefore, a new experiment was conducted by modifying the parameters of the HSG algorithm to investigate how to achieve an optimal solution with the exact placement heuristic in a shorter time. The new parameters used in the HSG algorithm are given in Table 5.

T_{max}	$g(T)$	N	ρ		
100	0.8	10	0.25		
H^0	ε_1	ε_2	δ	α	θ
50	0	0.1	10	0.5	0.5

TABLE 5. Parameters used in HSG

The solutions obtained from these calculations, along with their corresponding solution times, are presented in Table 6.

ID	Z^*	z_m	T_m	z_{HSG}^{sscp}	T_{HSG}^{sscp}	z_{HSG}^{dp}	T_{HSG}^{dp}
$U120_{00}$	48	**48**	295.52	49	196.2	**48**	190.102
$U120_{01}$	49	**49**	263.35	50	155.6	**49**	124.42
$U120_{02}$	46	**46**	210.43	47	168.8	**46**	95.52
$U120_{03}$	49	50	2000.36	51	166.9	**49**	139.18
$U120_{04}$	50	**50**	645.50	51	178.4	**50**	175.71
$U120_{05}$	48	49	2000.24	49	170.9	**48**	111.08
$U120_{06}$	48	49	2000.22	49	166.0	**48**	95.53
$U120_{07}$	49	50	2000.23	50	156.2	**49**	144.27
$U120_{08}$	51	53	2000.13	52	257.9	**51**	153.65
$U120_{09}$	46	47	2000.32	48	188.1	**46**	116.53
$U120_{10}$	52	53	2000.06	53	152.7	**52**	212.8
$U120_{11}$	49	**49**	1651.96	50	171.2	**49**	128.53
$U120_{12}$	48	49	2000.05	49	148.7	**48**	83.67
$U120_{13}$	49	**49**	156.56	50	165.7	**49**	67.12
$U120_{14}$	50	51	2000.12	51	208.5	**50**	113.76
$U120_{15}$	48	49	2000.08	49	191.2	**48**	132.92
$U120_{16}$	52	**52**	171.64	53	148.3	**52**	143.02
$U120_{17}$	52	53	2000.06	54	161.7	**52**	234.7
$U120_{18}$	49	**49**	200.59	50	152.1	**49**	129.67
$U120_{19}$	50	**50**	2000.08	51	194.1	**50**	130.31

TABLE 6. Computational results

As shown in Table 6, the optimal solution for all problems was achieved in significantly less time with the newly determined parameters. Achieving the optimal solution for all problems in a shorter time compared to SSCP highlights the effectiveness of the developed placement heuristic.

5. CONCLUSION

The computational experiments conducted in this study demonstrate the efficiency and effectiveness of the proposed HSG algorithm in solving the bin packing problem. Using the dynamic placement heuristic (DP), the optimal solution was achieved for all test problems, showing its superiority over the sequential, controlled, and pre-processed placement heuristic (SSCP). While the SSCP heuristic could not reach the optimal solution for any problem, it demonstrated a reasonable time efficiency with solution times consistently under 250 seconds.

In contrast, the DP heuristic consistently achieved the optimal solution across all test cases. However, under the initial parameter settings, the solution time exceeded 1400 seconds for some problems. We fine-tuned the parameters of the HSG algorithm and obtained a significant reduction in computation time while maintaining the ability to reach optimal solutions. The updated parameters enabled the DP heuristic to achieve the optimal solution for all problems in less time, further highlighting its efficiency and scalability.

These findings highlight the importance of parameter optimization and placement heuristic selection in obtaining high-quality solutions and computing efficiency. The developed placement heuristics, particularly the DP heuristic, demonstrate strong potential for practical applications in large-scale optimization problems where time efficiency and solution accuracy are critical. Future work could explore extending this approach to other combinatorial optimization problems or adapting the algorithm for real-world industrial use cases.

Future studies could focus on extending the proposed HSG algorithm to other combinatorial optimization problems, such as vehicle routing or job scheduling, where efficient packing or allocation is required. Also, searching hybrid approaches that combine the strengths of the dynamic placement heuristic (DP) with other metaheuristic methods, such as genetic algorithms or particle swarm optimization, could further enhance performance. Lastly, integrating machine learning techniques to adjust parameters during runtime dynamically could improve adaptability and efficiency in diverse problem instances.

Statements and Declarations

The authors declare that they have no conflict of interest, and the manuscript has no associated data.

References

[1] A. O. C. Ayres, B. S. C. Campello, W. A. de Oliveira, and C. T. L. S. Ghidini. A bi-integrated model for coupling lot-sizing and cutting-stock problems. *OR Spectrum*, 43:1047–1076, 2019.

[2] A. Azimov and R. Kasimov. On weak conjugacy, weak subdifferentials and duality with zero gap in nonconvex optimization. *International Journal of Applied Mathematics*, 1(2):171–192, 1999.

[3] A. Y. Azimov and R. N. Gasimov. Stability and duality of nonconvex problems via augmented lagrangian. *Cybernetics and Systems Analysis*, 38(3):412–421, 2002.

[4] A. M. Bagirov, G. Ozturk, and R. Kasimbeyli. A sharp augmented lagrangian-based method in constrained non-convex optimization. *Optimization Methods and Software*, 34(3):462–488, 2019.

[5] A. Baykasoğlu and B. K. Özbel. Modeling and solving a real-world cutting stock problem in the marble industry via mathematical programming and stochastic diffusion search approaches. *Computers & Operations Research*, 128:105173, 2021.

[6] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty. *Nonlinear Programming*. John Wiley & Sons Ltd, 2006.

[7] K. G. Bulbul and R. Kasimbeyli. Augmented lagrangian based hybrid subgradient method for solving aircraft maintenance routing problem. *Computers & Operations Research*, 132:105294, 2021.

[8] R. S. Burachik, R. N. Gasimov, N. A. Ismayilova, and C. Y. Kaya. On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian. *Journal of Global Optimization*, 34(1):55–78, 2006.

[9] E. K. Burke, G. Kendall, and G. Whitwell. A simulated annealing enhancement of the best-fit heuristic for the orthogonal stock-cutting problem. *INFORMS Journal on Computing*, 21(3):505–516, 2009.

[10] B. S. C. Campello, C. T. L. S. Ghidini, A. O. C. Ayres, and W. A. de Oliveira. A residual recombination heuristic for one-dimensional cutting stock problems. *TOP*, 30:194–220, 2021.

[11] S. A. d. A. Caroline de Arruda Signorini and G. M. Melega. One-dimensional multi-period cutting stock problems in the concrete industry. *International Journal of Production Research*, 60(8):2386–2403, 2022.

[12] V. Cerný. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

[13] C.-L. S. Chen, S. M. Hart, and W. M. Tham. A simulated annealing heuristic for the one-dimensional cutting stock problem. *European Journal of Operational Research*, 93(3):522–535, 1996.

[14] G. Dinc Yalcin and R. Kasimbeyli. On weak conjugacy, augmented lagrangians and duality in nonconvex optimization. *Mathematical Methods of Operations Research*, 92:192–228, 2020.

[15] M. L. Fisher. An applications oriented guide to lagrangian relaxation. *Interfaces*, 15(2):10–21, 1985.

[16] R. Gasimov and A. Rubinov. The nonlinear and augmented lagrangians for nonconvex optimization problems with a single constraint. *Applied and Computational Mathematics*, 1(2):142–157, 2002.

[17] R. N. Gasimov. Augmented lagrangian duality and nondifferentiable optimization methods in nonconvex programming. *Journal of Global Optimization*, 24(2):187–203, 2002.

[18] R. N. Gasimov and A. M. Rubinov. On augmented lagrangians for optimization problems with a single constraint. *Journal of Global Optimization*, 28(2):153–173, 2004.

[19] A. M. Geoffrion. Lagrangean relaxation for integer programming. In M. L. Balinski, editor, *Approaches to Integer Programming*, pages 82–114. Springer, Berlin, Heidelberg, 1974.

[20] P. C. Gilmore and R. E. Gomory. A linear programming approach to the cutting-stock problem. *Operations Research*, 9:849–859, 1961.

[21] B. Icmen Erdem and R. Kasimbeyli. A two-step math heuristic solution approach for the two-dimensional cutting stock problem. *Journal of Nonlinear and Convex Analysis*, 24:681–699, 05 2023.

[22] B. Icmen Erdem and R. Kasimbeyli. Mathematical programming based heuristic method for two-dimensional two-stage cutting stock problems. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 39:899–908, 2024.

[23] N. Kasimbeyli. Existence and characterization theorems in nonconvex vector optimization. *Journal of Global Optimization*, 62(1):155–165, 2015.

[24] N. Kasimbeyli and R. Kasimbeyli. A representation theorem for bishop-phelps cones. *Pacific Journal of Optimization*, 13(1):55–74, 2017.

[25] N. Kasimbeyli, T. Sarac, and R. Kasimbeyli. A two-objective mathematical model without cutting patterns for one-dimensional assortment problems. *Journal of Computational and Applied Mathematics*, 235(16):4663–4674, 2011.

[26] R. Kasimbeyli. Radial epiderivatives and set-valued optimization. *Optimization*, 58(5):521–534, 2009.

[27] R. Kasimbeyli. A nonlinear cone separation theorem and scalarization in nonconvex vector optimization. *SIAM Journal on Optimization*, 20(3):1591–1619, 2010.

[28] R. Kasimbeyli and M. Karimi. Separation theorems for nonconvex sets and application in optimization. *Operations Research Letters*, 47(6):569–573, 2019.

[29] R. Kasimbeyli and M. Mammadov. On weak subdifferentials, directional derivatives, and radial epiderivatives for nonconvex functions. *SIAM Journal on Optimization*, 20(2):841–855, 2009.

[30] R. Kasimbeyli and M. Mammadov. Optimality conditions in nonconvex optimization via weak subdifferentials. *Nonlinear Analysis: Theory, Methods & Applications*, 74(7):2534–2547, 2011.

[31] R. Kasimbeyli and O. Ustun. Solving the quadratic assignment problem using f-msg algorithm. *Journal of Industrial and Management Optimization*, 3, 05 2007.

[32] R. Kasimbeyli, O. Ustun, and A. M. Rubinov. The modified subgradient algorithm based on feasible values. *Optimization*, 58(5):535–560, 2009.

[33] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science (New York, N.Y.)*, 220:671–80, 06 1983.

[34] S. C. Leung, D. Zhang, and K. M. Sim. A two-stage intelligent search algorithm for the two-dimensional strip packing problem. *European Journal of Operational Research*, 215(1):57–69, 2011.

[35] Y. Li, B. Zheng, and Z. Dai. General particle swarm optimization based on simulated annealing for multi-specification one-dimensional cutting stock problem. In Y. Wang, Y.-m. Cheung, and H. Liu, editors, *Computational Intelligence and Security*, pages 67–76, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[36] F. Ozcelik and T. Sarac. A genetic algorithm extended modified sub-gradient algorithm for cell formation problem with alternative routings. *International Journal of Production Research*, 50:1–13, 01 2011.

[37] A. R. Pitombeira-Neto and B. de Athayde Prata. A matheuristic algorithm for the one-dimensional cutting stock and scheduling problem with heterogeneous orders. *TOP*, 28(1):178–192, 2020.

[38] S. V. Ravelo, C. N. Meneses, and M. O. Santos. Meta-heuristics for the one-dimensional cutting stock problem with usable leftover. *Journal of Heuristics*, 26(4):585–618, Aug. 2020.

[39] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Berlin, Heidelberg, 1998.

[40] A. Rubinov and R. Gasimov. Strictly increasing positively homogeneous functions with application to exact penalization. *Optimization*, 52(1):1–28, 2003.

[41] A. M. Rubinov, X. Q. Yang, A. M. Bagirov, and R. N. Gasimov. Lagrange-type functions in constrained optimization. *Journal of Mathematical Sciences*, 115(4):2437–2505, 2003.

[42] A. Sipahioglu and T. Sarac. The performance of the modified subgradient algorithm on solving the 0-1 quadratic knapsack problem. *Informatica*, 20(2):293–304, 2009.

[43] B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57:1143–1160, 2006.

[44] M. A. Takan and R. Kasimbeyli. A hybrid subgradient method for solving the capacitated vehicle routing problem. *Journal of Nonlinear and Convex Analysis*, 21:413–423, 02 2020.

[45] E.-G. Talbi. *Metaheuristics*, chapter 1, pages 1–86. John Wiley & Sons Ltd, 2009.

[46] O. Ustun and R. Kasimbeyli. Combined forecasts in portfolio optimization: A generalized approach. *Computers & Operations Research*, 39(4):805–819, 2012.

[47] L. Wei, Z. Luo, R. Baldacci, and A. Lim. A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. *INFORMS Journal on Computing*, 32:428–443, 2020.