공학석사 학위논문

Cost Minimization of Reinforcing Bar Order By Applying Bin-Packing Approach

빈 패킹 접근방식을 활용한 철근 주문 비용의 최소화

2023 년 2 월

서울대학교 대학원

건설환경공학부

최 우 석

Cost Minimization of Reinforcing Bar Order By Applying Bin-Packing Approach

빈 패킹 접근방식을 활용한 철근 주문 비용의 최소화

지도교수 김 호 경

이 논문을 공학석사 학위논문으로 제출함

2023년 2월

서울대학교 대학원

건설환경공학부

최 우 석

최우석의 공학석사 학위논문을 인준함

2022년 12월

위 원 장	송 준 호 (인)
부 위 원 장	김 호 경 (인)
위 원	채 윤 병 (인)

ABSTRACT

Cost Minimization of Reinforcing Bar Order By Applying Bin-Packing Approach

Woo Seok Choi
Department of Civil and Environmental Engineering
Seoul National University

Recently, in an environment where the price of domestic reinforcing bars is steadily rising, the cost of rebar construction accounts for about 16% of the total construction cost, so reducing the cost of rebar construction is a very important factor in reducing construction costs. This paper conducts research to reduce rebar ordering costs.

For this purpose, existing representative algorithms of the Column-generation approach and the Bin-packing approach, which are one of the linear programming solutions are introduced.

Revised-Best-fit-decreasing algorithm is proposed, which improve the existing Best-fit-decreasing algorithm. For the newly proposed Revised-Best-fit-decreasing, divide the price of the stock rebar by the length of the stock rebar, reselect the stock rebar with the smallest value as the best rebar, and rearrange the demand rebar in descending order of length. When the demand rebar enters the new stock rebar, it goes into the best rebar. Additionally, it considers two demand rebars at the same time and creates a stock rebar that can contain both demand rebars if the sum of the lengths of the two demand rebars is greater than or equal to the best rebar length.

The algorithm developed above is applied to the caisson structure of Ulsan New Port Development Project. The demand rebars for the above caisson structure consist of 818 types and 36,478 pieces. A quantity of approximately 326.463 tons multiplied by a 6% surcharge for these rebars is the quantity executed for the actual rebar order. As a result of minimizing the rebar order cost with the Revised-Best-fit-decreasing algorithm, the order quantity of the stock rebar was about 322.427 tons, resulting in a reduction of about 4.1 tons and a reduction of about 4.2 million won in the order cost.

The degree of optimization of the algorithm varied with the diameter of the rebar, but the smaller the rebar diameter and the greater the number of demand rebars input to run the algorithm, the greater the minimization effect. Appropriate use of the algorithm proposed in this paper can reduce the order quantity of rebars and reduce the cost of ordering rebars.

Keywords: Rebar; Pattern; Linear programming; Bin-packing approach; Revised-best-fit-decreasing;

Student Number: 2020-28658

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Research Background

Reinforcing bars are members that bear tensile force in reinforced concrete structures and are widely used in civil engineering and building structures. The rebar construction cost accounts for about 16% of the total project cost, making it a very high proportion of the construction cost (Kim, Hong & Joo, 2004). In addition, as the price of reinforcing bars in the Korean market tends to rise recently, there is a risk that the construction cost of reinforcing bars will increase.

Reinforcing bars are required to have different diameters and lengths depending on the shape of the structure, and these reinforcing bars are called demand rebars. Rebars sold in the market are called stock rebars, stock rebars are sold in units of 1m from 6m to 12m according to diameter, so after purchasing the stock rebars, they are processed to make demand rebars and scraps are generated in this process. In other words, the quantity and cost of the stock rebars to be ordered vary depending on how the demand rebars are produced from the stock rebars of a specific length and how the stock rebars and the demand rebars are combined.

The purpose of this study is to analyze the combination of stock rebar and demand rebar to minimize the cost of ordering the entire stock rebars.

However, this paper does not consider other costs such as the cost of processing rebar or the labor of workers. Research is focused solely on minimizing the cost of ordering rebar.

1.2 Definitions and Notations

A particular combination of stock and demand rebars is called a pattern, as shown in Figure 1.1 below.

In this paper, Terms used in the paper follow the definitions below.

Demand rebar = Rebar required for construction

Stock rebar = Rebar sold in the market, usually sold by whole number

Pattern = Any combination of stock rebar and demand rebar(s)

Scrap = Rebar left from the stock rebar after extraction of demand rebars

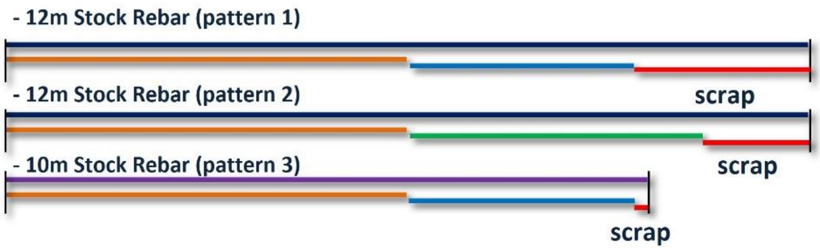


Figure 1.1 Definitions

In this paper, the following notations are used with the following meanings:

l_i = Length of demand rebar (i=1, …, m)

N_i = Required number of demand rebar (i=1, …, m)

L_t = Length of stock rebar (t= 1, …, k)

c_t = Cost of stock rebar (t= 1, …, k)

a_{ij} = Number of pieces of l_i in the pattern j

P_j = Each pattern (j=1, …, n)

x_j = The number of specific pattern j that need to be created (j=1, …, n)

1.3 Structure of the Thesis

This paper consists of five chapters to introduce each part of the proposed methodology.

This chapter describes the background of the research on minimizing the order cost of rebar and expresses definitions and notations used in this paper.

Next, Chapter 2 analyzes a previous paper that conducted similar studies to the rebar order cost minimization study. Specifically, after defining a problem with integer linear programming, describing how to solve it with Column generation method. It also describes the limitations that make it impossible to directly apply this method to this research paper.

Chapter 3 introduces the Bin-packing approach method to complement for the limit point that occurs when applying the Column generation method introduced in Chapter 2 to the rebar ordering case. It also introduces and applies three typical solutions for the Bin-packing approach. Furthermore, it improves the Best-fit-decreasing algorithm and propose a new solving algorithm, Revised-Best-fit-decreasing.

Chapter 4 tests the newly proposed Revised-Best-fit-decreasing algorithm directly on real structures and analyzes the validity and problems of the results.

Finally, Chapter 5 summarizes the main findings and contributions of this study and discusses several additional research topics.

CHAPTER 2

EXISTING METHOD

2.1 Basic Concept

The problem of minimizing the cost of ordering stock rebars can be thought of as the problem of meeting all the demand rebars and minimizing the number of stock rebar orders.

Gilmore and Gomory (1961) proposed this as an integer linear programming problem. Such integer linear programming is the problem of finding optimized values of objective functions that satisfy given inequality conditions.

Looking at Figure 2.1 below, n patterns are generated to meet all demand rebars. A pattern is a specific combination of stock rebar and demand rebars as defined earlier and each pattern consists of one specific stock rebar and multiple specific demand rebar combinations. All n patterns must contain both all kinds of demand rebars and their respective quantities.

For Figure 2.1 below, number of demand rebar l_1 must be greater than or equal to N_1 in n patterns, which is expressed in Inequation (2.1) below. The required number of conditions must be met for every kind of demand rebars.

The number of demand rebars l_i included in the specific pattern P_j can be expressed by a_{ij} based on the notations expressed above.

It is represented by the following Inequation in (2.2).

$$\boldsymbol{2x_1 + 1x_2 + \cdots 0x_n \geq N_1} \tag{2.1}$$

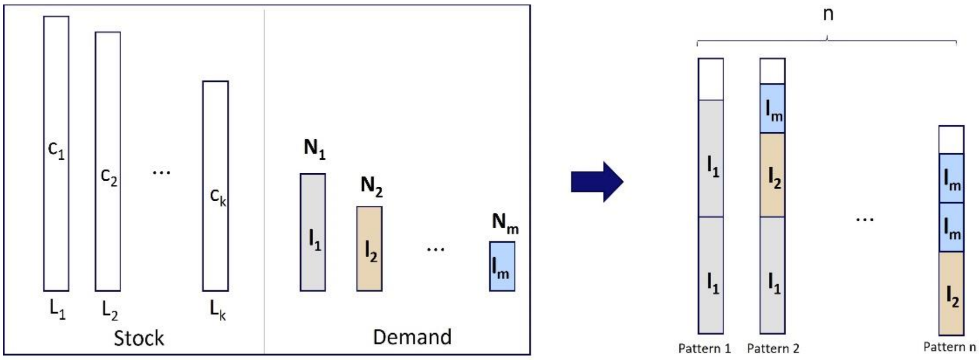


Figure 2.1 Example of rebars and patterns

$$\boldsymbol{a_{i1}x_1 + a_{i2}x_2 + \cdots a_{in}x_n \geq N_i, (i = 1, \cdots, m)} \tag{2.2}$$

By multiplying the price of stock rebars used for each pattern by the number of productions for that pattern, the objective function to be minimized is derived as shown in Function (2.3) below. Since each pattern must be at least 0, the condition of Inequation (2.4) must also be satisfied.

$$c_1x_1 + c_2x_2 + \cdots c_nx_n \tag{2.3}$$

$$x_j \geq 0, (j = 1, \cdots, n + m) \tag{2.4}$$

After all, the problem of minimizing the rebar order cost can be considered as a linear programming problem of minimizing the objective function (2.3) among the values satisfying the conditions of Inequations (2.2) and (2.4).

Gilmore et al. (1961) showed an integer linear programming solution to the Column generation method.

2.2 Example

The Column generation method is explained in the example below.

A total of 5 types of demand rebar are required from 2m to 6m in 1m increments, and 10 of each are required. Market sells only 8m stock rebar at \$10, 9m stock rebar at \$12, 10m stock rebar at \$14. The stock rebar order cost can be minimized in 5 stages.

Step 1. Make basic patterns

The first step is to create the basic patterns. A basic pattern is created with the simplest combination of demand rebars and stock rebar. The total number of patterns is equal to the number of demand rebar types and each pattern consists of only one demand rebar type. With the demand rebars and stock rebars conditions in the example above, basic patterns are constructed as shown in Figure 2.2 below. The basic pattern is constructed by combining the shortest stock rebar and demand that leave no scrap. If there is no combination that separates to fit exactly, construct the basic pattern with the stock rebar with the smallest difference from the length of the demand rebar.

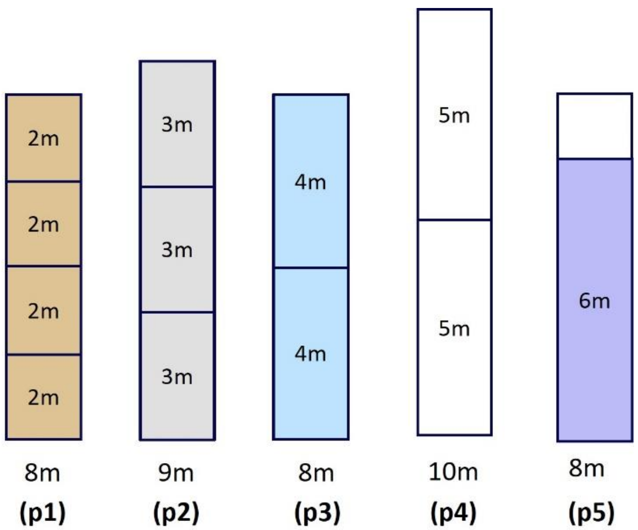


Figure 2.2 Basic patterns

Step 2. Objective function and constraints

Based on the created basic patterns, the objective function and conditional inequation can be derived. The objective function can be expressed as Function (2.5). This cost function consists of the stock rebar price for each pattern multiplied by the unknown quantity produced for each pattern. Conditional inequations are generated for the number of types of demand rebars. Conditional inequations are constructed as in Inequation (2.6) through (2.10) below.

$$\mathbf{10}x_1 + \mathbf{12}x_2 + \mathbf{10}x_3 + \mathbf{14}x_4 + \mathbf{10}x_5 \tag{2.5}$$

$$\mathbf{4}x_1 + \mathbf{0}x_2 + \mathbf{0}x_3 + \mathbf{0}x_4 + \mathbf{0}x_5 \geq \mathbf{10} \tag{2.6}$$

$$\mathbf{0}x_1 + \mathbf{3}x_2 + \mathbf{0}x_3 + \mathbf{0}x_4 + \mathbf{0}x_5 \geq \mathbf{10} \tag{2.7}$$

$$\mathbf{0}x_1 + \mathbf{0}x_2 + \mathbf{2}x_3 + \mathbf{0}x_4 + \mathbf{0}x_5 \geq \mathbf{10} \tag{2.8}$$

$$\mathbf{0}x_1 + \mathbf{0}x_2 + \mathbf{0}x_3 + \mathbf{2}x_4 + \mathbf{0}x_5 \geq \mathbf{10} \tag{2.9}$$

$$\mathbf{0}x_1 + \mathbf{0}x_2 + \mathbf{0}x_3 + \mathbf{0}x_4 + \mathbf{1}x_5 \geq \mathbf{10} \tag{2.10}$$

Step 3. Matrix form

Represent each formula constructed in Step 2 in Matrix format. The objective function (2.5) is represented by a matrix ***C*** of 1 row and 5 columns, and all the left formulas of Inequations (2.6) to (2.10) are bundled and represented by a matrix ***A*** of 5 rows and 5 columns. ***C*** is a matrix that informs the price information of the basic pattern, and ***A*** is a matrix that informs the structure of the basic pattern.

In the example, matrix ***C*** and matrix ***A*** are equal to the values of (2.11) and (2.12) respectively.

After that, this price matrix $\boldsymbol{C}$ and basic pattern matrix $\boldsymbol{A}$ are used to determine whether the introduction of new patterns improve the basic patterns in terms of price.

$$\boldsymbol{C} = [\mathbf{10} \quad \mathbf{12} \quad \mathbf{10} \quad \mathbf{14} \quad \mathbf{10}] \tag{2.11}$$

$$\boldsymbol{A} = \begin{bmatrix} \mathbf{4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{2.12}$$

Step 4. Introduce a new pattern P

The fourth step is to introduce new patterns. The new pattern differs from the basic patterns and consists of any combination of stock and demand rebars. The matrix $\boldsymbol{P}$ representing the new pattern has 5 rows and 1 column, with each row representing the number of each demand rebar in the pattern.

Introducing a new pattern as shown in Figure 2.3 below, the new pattern $\boldsymbol{P}$ can be expressed as (2.13).

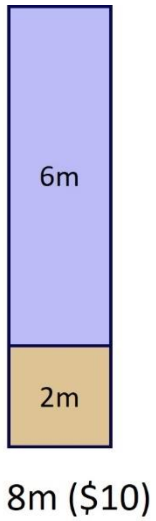


Figure 2.3 Example of a new pattern

After expressing the matrix $\boldsymbol{P}$ for the new pattern, the relationship with the existing $\boldsymbol{C}$, $\boldsymbol{A}$ matrices is derived. There is a specific matrix $\boldsymbol{U}$ of 5 rows and 1 column representing the relationship between the basic pattern matrix $\boldsymbol{A}$ and the new pattern $\boldsymbol{P}$. Their relation can be expressed as (2.14), and the matrix $\boldsymbol{U}$ always exists.

$$\boldsymbol{P} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \tag{2.13}$$

$$\boldsymbol{A} \cdot \boldsymbol{U} = \boldsymbol{P} \tag{2.14}$$

In this example the matrix $\boldsymbol{U}$ is derived (2.15) as follows. The meaning of each element of the matrix $\boldsymbol{U}$ is the proportion of each demand rebar required to make the new pattern $\boldsymbol{P}$ in the basic pattern $\boldsymbol{A}$.

In other words, the basic pattern p_1 originally consisted of 4 2m demand rebars, and only 1/4 of the basic pattern is required to generate a new 2m single rebar.

$$\boldsymbol{U} = \begin{bmatrix} \mathbf{1/4} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \tag{2.15}$$

Multiplying the pattern cost matrix $\boldsymbol{C}$ by the matrix $\boldsymbol{U}$ gives the price value of the new pattern. If this value is greater than c representing the price of the stock rebar used in the new pattern, the new pattern is considered an improvement over the existing patterns.

That is, a new pattern is introduced if the following Inequation (2.16) is satisfied.

$$\boldsymbol{C} \cdot \boldsymbol{U} > \boldsymbol{c} \tag{2.16}$$

Step 5. Find the best fitting new patterns by repeating Step 4.

Repeat Step 4 until no more price-improving patterns are introduced. While repeating Step 4, existing basic patterns are changed to new pattern combinations. Generating a new pattern is the same as generating a new column. Therefore, this method is called column generation method.

Finding the number of orders for the changed pattern yields the overall order cost, which is reduced compared to the existing order cost.

2.3 Limitations of this Method

The above-described method is effective when the length of the demand rebars and the length of the stock rebars are integers and the types of demand rebars are small. As in the previous example, there are 5 demand rebar types and demand rebar lengths are integers, so it is suitable because it is easy to generate basic patterns.

However, in the case of actual civil engineering reinforced concrete structures, there are many types of demand rebars, and the length of the demand rebars is a decimal point, so it is judged that the above method is not suitable.

According to (Jahromi et al., 2012) the one-dimensional cutting stock problem, which considers many combinations of demand rebars and stock rebars, is the Np-hard problem, and the computation time increases exponentially as the number of demand orders increases.

When the number of types of demand rebar is about 1000 or more, it was confirmed that the CPU time took more than 15000 seconds, and when the number of types of demand rebar was more than 5000, an appropriate solution could not be found even after 24 hours (Jahromi et al., 2012).

Column generation method is basically based on simplex method, according to (Fourer, 1988). In the worst case of the simplex method, degeneracy may occur. In simplex method, basically the basic feasible solution proceeds in the direction of decreasing, but if there is a 0 variable in the basic variables and the basic feasible solution does not decrease and the value does not change, infinite cycling is performed (Fourer, 1988).

Therefore, Chapter 3 introduces the Bin-packing approach, which is the core approach of this paper, as a methodology for minimizing the ordering cost of rebars and applies it to a simple example where the column generation method can be applied.

CHAPTER 3

BIN-PACKING APPROACH

The Bin-packing approach is an optimization problem about filling all the demands while minimizing the number of bins used when there is a finite number of bins (Martello & Toth, 1990).

Here, the stock rebar can be regarded as a bin and the problem of minimizing the number of stock rebars generated while putting all the demand rebars into the stock rebar.

The stock rebar order cost is generally related to the number of stock rebar orders, so it fits the overall purpose.

The Column generation method is a method of generating patterns considering many combinations of stock rebars and demand rebars and comparing the prices with existing patterns. In contrast, the Bin-packing approach is concerned with which stock rebars the demand rebar is put into, where the stock rebar is fixed, and does not consider in advance the number of pattern cases. Each pattern is only made up after adding demand rebars to stock rebars.

Solving the bin-packing approach is generally a linear complexity in computational time complexity (Bekesi, Galambos & Kellerer, 2000).

As shown in Figure 3.1 below, the bin-packing approach, whose computational complexity is a linear complexity proportional to the absolute amount of demand rebar, is superior to the Column-generation method, whose computational complexity is exponential complexity. Computational speed is much faster.

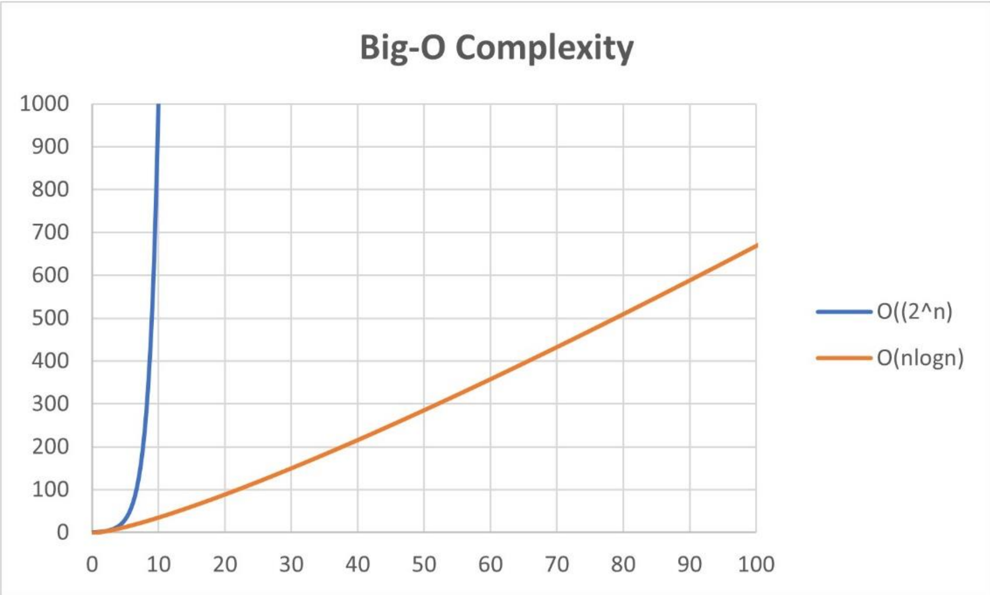


Figure 3.1 Big-O complexity

Due to the above characteristics, it is more advantageous to apply the Bin-packing approach to the problem of minimizing the rebar order cost of real reinforced concrete structures.

3.1 Basic Concept

3.1.1 General Solutions

Martello et al. (1990) introduced algorithms that solves the bin packing approach.

The first is the Best-fit algorithm. The Best-fit algorithm puts the demand into the bin(stock) with the least length of remaining space when it is put into the bin(stock). Here "Best" means the best space, but the best space is the one with the least remaining space when the demand is put into bin(stock).

The second is the Best-fit-decreasing algorithm. The logic of entering demand is the same as the Best-fit algorithm. However, before the demands are put into the bin(stock), they are sorted in descending order of size, here in descending order of length from long to short. The meaning of decreasing added later is to order these demands in descending order.

The third is the First-fit-decreasing algorithm. It is an algorithm that tries to put it in the first bin it meets, unlike the best that puts it in the most optimal space. Similarly, since it is decreasing, sort demand in descending order of length.

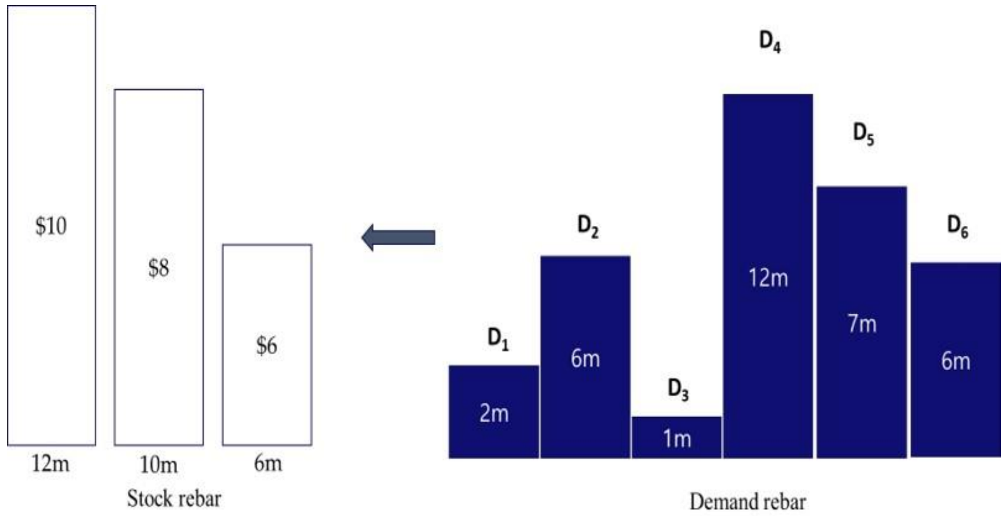
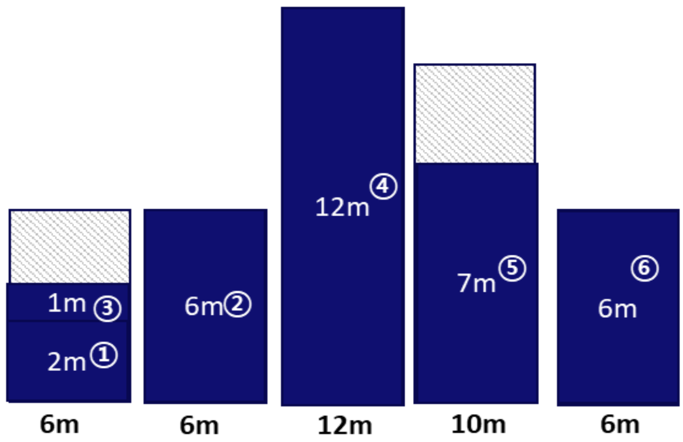


Figure 3.2 An example of the bin-packing approach

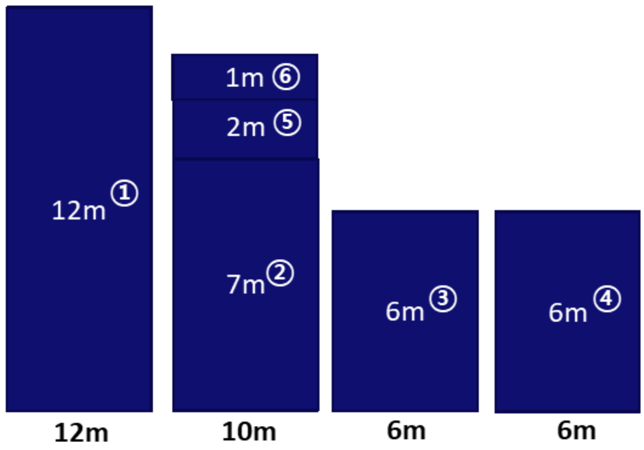
The example in Figure 3.2 illustrates the above three common solutions for the Bin-packing approach.

The stock rebar is available in 12m, 10m and 6m at $10, $8, and $6 respectively. Demand rebar requires 6 as above.

(a)


12m ④
7m ⑤
⑥
6m
1m ③
6m ②
2m ①
6m
6m
12m
10m
6m

(b)


1m ⑥
2m ⑤
12m ①
7m ②
6m ③
6m ④
12m
10m
6m
6m

(c)

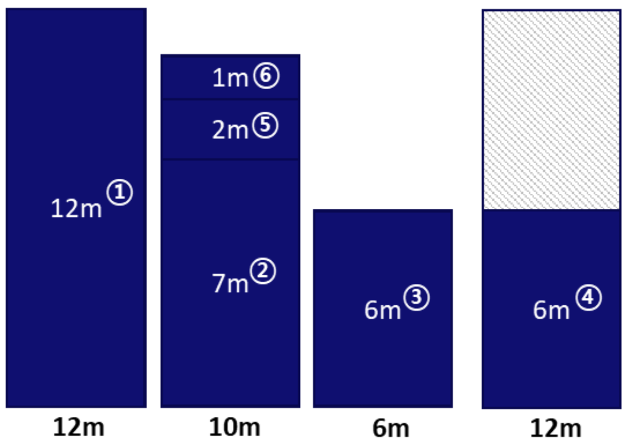

1m ⑥
2m ⑤
12m ①
7m ②
6m ③
6m ④
12m
10m
6m
12m

Figure 3.3 General solutions of the bin packing approach
(a) Best-fit
(b) Best-fit-decreasing (c) First-fit-decreasing

First, in the case of (a) Best-fit, the first demand rebar of 2m is taken into the bin(stock), but according to the best principle of putting it in the smallest space, it goes into the 6m stock rebar at first. For the second 6m demand rebar, the smallest space is the first created 6m stock rebar containing the 2m demand rebar and the remaining 4m space, but the 6m demand rebar cannot enter. So, create a new 6m stock rebar and fill that space with a 6m demand rebar. The third demand rebar, 1m, can fit into the existing minimum space of 4m, so it goes into the first generated stock on the best-fit principle. Using these rules is the Best-fit algorithm.

The second (b) Best-fit-decreasing is an algorithm that arranges the demand rebars in descending order of length in advance and then puts the demands into the minimum space like Best-fit algorithm. So, the first demand rebar in Best-fit-decreasing is 12m, naturally generating a 12m stock rebar to enter. The second demand rebar is 7m and can enter the 12m stock rebar and 10m stock rebar but enters the 10m stock rebar in the best(smallest) space.

The third (c) First-fit-decreasing is an algorithm that sorts the demand rebars in descending order of length and then puts them into the stock rebar where they first meet. The order of stock rebars met at this time is the same as the order of stock rebars specified by the user. The stock rebars are set to 12m, 10m, 6m, so the first 12m demand rebar goes into the 12m stock rebar, the second 7m demand rebar goes into the 10m stock rebar. The difference with the Best-fit-decreasing algorithm comes when inserting the 4th 6m demand rebar, while the Best-fit-decreasing algorithm produces the smallest space, 6m stock rebar, For the first-fit-decreasing algorithm, put a 6m demand rebar on the 12m stock rebar, which is the fourth order of the stock rebar.

For the total rebar order costs derived by the above three algorithms, they are shown in Table 3.1 below. Among the above three algorithms, the Best-fit-decreasing algorithm has the highest degree of cost minimization with a total order cost of $30.

Table 3.1 Total cost for each solution

Method	**Total Cost ($)**
Best-fit	36
Best-fit-decreasing	30
First-fit-decreasing	34

3.1.2 Limitations of General Solutions

The three general algorithms above do not lead to an exact solution of order cost minimization. The minimum cost for the example corresponding to Figure 3.1 is to order two 12m stock rebars and one 10m stock rebar for a total order cost of $28.

First, in the case of the Best-fit algorithm, the overall order volume increases compared to the Best-fit-decreasing algorithm because the demand rebars are not sorted in descending order. The shorter the length of the remaining demand rebar, the more advantageous it is to fill the remaining space of the already used stock rebar. Therefore, sorting the demand rebars in descending order of length is even more effective in minimizing the generation of the number of stock rebars.

The First-fit-decreasing algorithm is related to the stock rebar sort order. This is valued by the algorithm user randomizing the sort order of the stock rebar.

Finally, the Best-fit-decreasing algorithm, which had slightly less order cost of the three methods, failed to derive the exact solution. In Figure 3.2(b), the 3rd and 4th demand rebars each generate a 6m stock rebar. An exact solution would be two 6m demand rebars fitting into one 12m stock rebar, further reducing the overall order cost.

3.2 Revised-Best-Fit-Decreasing

To improve the limitations of the three algorithms mentioned above and derive a solution close to the exact solution of rebar order cost minimization, I modify the existing Best-fit-decreasing algorithm and propose it.

First, the three existing algorithms focus only on the rebar length and set the space with the least remaining space to the best bin (stock). The length and cost of a stock rebar must be considered together to consider not only the number of stock rebar orders, but also the order cost.

Second, two demand rebars need to be considered simultaneously when putting the demand rebar into the stock rebar to improve the Best-fit-decreasing algorithm presented previously.

The Revised-Best-fit-decreasing algorithm added the above two conditions.

3.2.1 Condition to Consider the Price of Stock Rebar

A condition that considers the price of stock rebar relates to how to define the best bin(stock). In the conventional Best-fit algorithm and Best-fit-decreasing algorithm, only the length variable is considered and the space with the minimum remaining length is regarded as the best bin(stock), but the Revised-Best-fit-decreasing algorithm considers length and price simultaneously.

In other words, divide the stock rebar price by the stock rebar length, and the lowest value is the best bin(stock). Because to minimize the overall cost of ordering stock rebars, the cost of stock rebar should be considered at the same time as the length of stock rebar.

Adding the above condition makes the best stock in the example in Figure 3.1 a 10m stock rebar instead of a 6m stock rebar. This means the relative price of 10m stock rebar is cheaper than 6m stock rebar. The 10m stock rebar is relatively inexpensive compared to the 6m stock rebar and has more space for other demand rebars. This is shown in Figure 3.4 below

And considering the price conditions, the stock rebar that has already been used will have a price of zero and will be considered a stock rebar that is cheaper than other stock rebars that are not being used.

Also, if the relative price obtained by dividing the price of the stock rebar by the length of the stock rebar is the same, the longest stock rebar that can contain many demand rebars is the best stock.

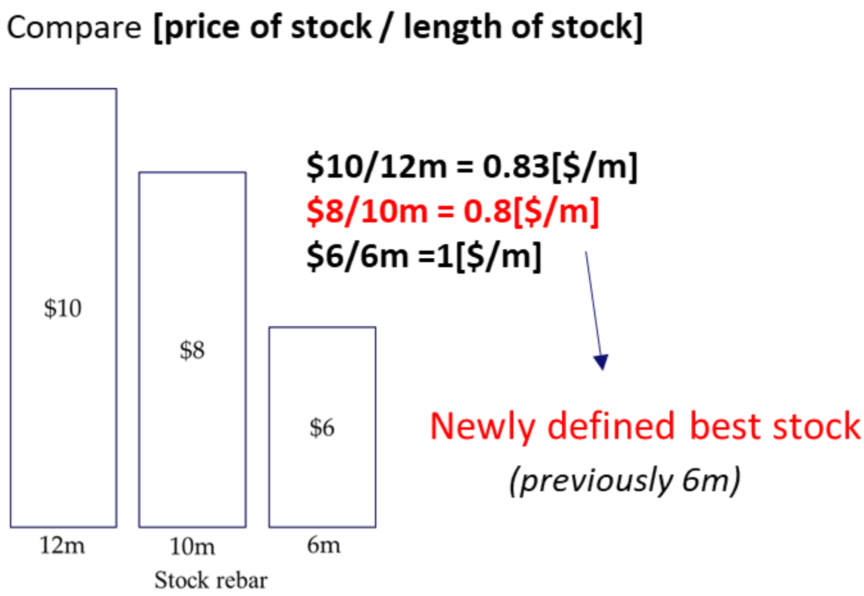


Figure 3.4 Condition to consider the price of stock rebar

3.2.2 Condition to Consider Two Demand Rebars at the same time

The second condition is one that considers two demand rebars simultaneously. The Best-fit-decreasing algorithm in Figure 3.2(b) creates a 6m stock rebar when entering the 3rd demand rebar and creates another 6m stock rebar when entering the 4th demand rebar and inserts it. This is because demand rebars are considered one at a time.

Considering two demand rebars at the same time, if the total length of the two demand rebars is greater than the best stock length set in the first condition, generate a stock rebar that is longer than the combined length of the two demand rebars. This is because it is usually advantageous to have multiple demand rebars in one stock rebar.

Applying this condition, as shown in Figure 3.5, inserting the third demand rebar from the example in Figure 3.1 produces a 12m stock rebar that is more than the combined length of the third and fourth demand rebars, which is 12m.

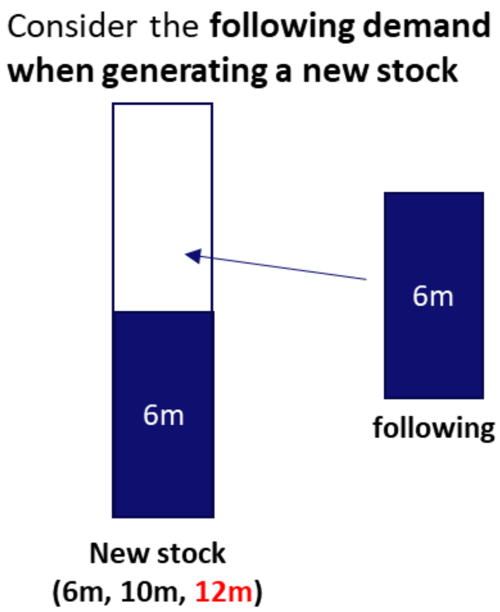


Figure 3.5 Condition to consider two demand rebars at the same time

Adding the above two conditions to the example in Figure 3.1 and applying the Revised-Best-fit-decreasing algorithm results in the pattern shown in Figure 3.6, where the overall order cost is $28, the same as the optimal solution.

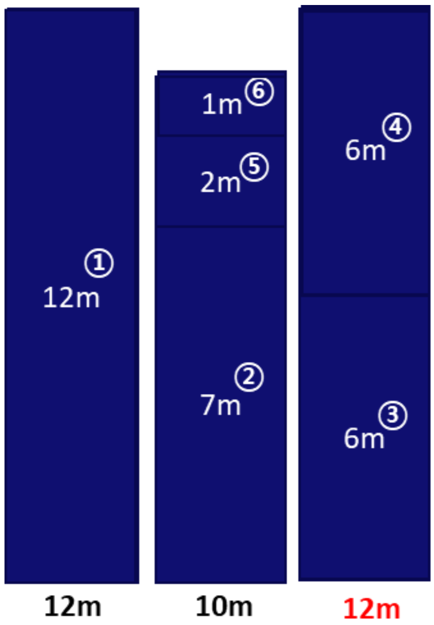


Figure 3.6 Example of Figure 3.1 with Revised-best-fit-decreasing applied

3.3 Simple Test

Gilmore et al. (1961) presented a simple example on minimizing order costs. In this example, I applied the Column generation method of Chapter 2, the existing algorithm of the bin-packing approach introduced in Chapter 3, and the newly proposed Revised-Best-fit-decreasing algorithm.

Gilmore et al. (1961) used as an example a demand of 20 pcs of 2m, 10 pcs of 3m, 20 pcs of 4m and a stock of 5m of 6$, 6m of 7$ and 9m of 10$.

The optimization result values for each method are derived as shown in Table 3.2 below.

Table 3.2 Comparison of total cost incurred by each method

Method	Number of Stock Rebar(s)			Total Cost($)
	5m ($6 each)	6m ($7 each)	9m ($10 each)	
Column Generation	0	10	10	170
Best-fit	40	0	0	240
Best-fit-decreasing	35	0	0	210
First-fit-decreasing	9	9	9	207
Revised-Best-fit-decreasing	0	0	19	190

The solution of the existing Column generation method can be regarded as an exact solution when the types of demand rebar are small, and the lengths are an integer. A simple test result confirms that the Bin-packing approach algorithm increases the overall order cost compared to the Column generation method.

In the simple test above, the total number of demands is 50 and the number of demands is slightly less to find the optimal solution. Due to the algorithmic characteristics of the Bin-packing approach, the greater the number of demands, the greater the chances of filling the remaining space with demands, resulting in greater optimization efficiency.

Among the Bin-packing approach algorithms, the newly proposed Revised-Best-fit-decreasing algorithm is confirmed to be the best in order cost minimization compared to the existing Bin-packing approach algorithms. Therefore, the newly proposed Revised-Best-fit-decreasing algorithm is effective in minimizing the rebar order cost.

The Revised-Best-fit-decreasing algorithm ordered the least number of rebars in stock at 19 out of all the methods. The Revised-Best-fit-decreasing algorithm is expected to be the most effective if the cost of stock rebar does not vary significantly with length.

CHAPTER 4

ACTUAL CASE TEST

Through simple test, the newly proposed Revised-Best-fit-decreasing algorithm derives an optimized solution compared to the existing Bin-packing approach Best-fit, Best-fit-decreasing, and First-fit-decreasing algorithms. After confirming that, I applied this to an actual reinforced concrete structure.

In the actual case test, the number of types of reinforcing bars in demand was very large, so the Column generation method in Chapter 2 is not applied.

The Bin-packing approach algorithm is coded in Python language and the result values were compared.

Before conducting the actual case test, it is necessary to create input data containing various types of information on demand rebars, which was created in the csv format of an excel file. The actual rebar design information can be obtained from the rebar processing shop drawing provided by the rebar design company. The design data for the foundation caisson of the south side wave block construction of Ulsan New Port is provided by the rebar processing shop drawing file provided by Kangshin Development Co., Ltd.

4.1 Test Bed and Market Conditions

Figure 4.1 below shows the caisson structure to which the Bin-packing approach algorithm is applied. The caisson structure for which the actual case test is conducted is a reinforced concrete structure with a width of 38.1m, a length of 32.5m, and a height of 25.0m.

The foundation caisson in Figure 4.1 below is designed as a total of 818 lengths of demand rebar, for a total of 36,478 demand rebars. The types and numbers of rebars required by structure type are shown in Table 4.1 below.

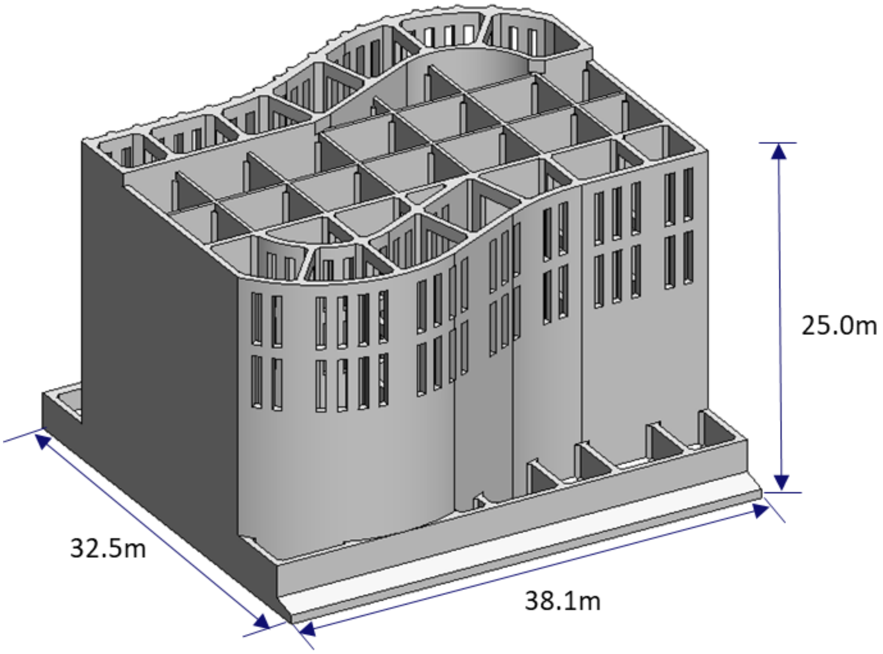


Figure 4.1 Foundation caisson of Ulsan New Port Construction

Table 4.1 Detailed information of the Foundation Caisson

Types	Number of Sub-Types	Pieces
Bottom Plate	151	6905
Front Wall	72	929
Back Wall	84	940
Side Wall	65	1024
Longitudinal Bulkhead	132	3154
Transverse Bulkhead	119	3007
Others	195	20519
Total	818	36478

In addition, in the case of reinforcing bar shop drawing, the rebar data is classified according to the diameter of the rebar, and the required quantity for each diameter is calculated in units of weight. The design quantities by rebar diameter for the foundation caisson in Figure 4.1 above are shown in Table 4.2 below.

Table 4.2 Design quantity by diameter

Diameter	As Drawing(ton)
H13	11.045
H16	86.587
H19	43.789
H22	101.392
H25	31.142
H29	34.029
Total	307.984

The actual construction company orders the rebars by multiplying the design shop drawing values in Table 4.2 above by a 6% design surcharge. If the quantity obtained from the newly proposed Revised-Best-fit-decreasing algorithm is less than that multiplied by the 6% design surcharge, the newly proposed algorithm is a valid algorithm.

It is also an important point to analyze how much the Revised-Best-fit-decreasing algorithm offers significant cost savings compared to the existing Bin-packing approach algorithm.

In fact, in the Korean rebar market, rebars are sold by diameter and length, but the unit of sale is weight. Figure 4.2 below shows how to calculate the rebar order cost. The total rebar order quantity is the product of rebar diameter, rebar length, rebar unit weight, and order quantity. Once the amount of rebars is calculated, multiply it by the unit price for the weight to obtain the order cost for the entire rebars.

As shown in Figure 4.2 below, the variable is the number of orders per length of each rebar to minimize rebar ordering costs. Determining the number of orders for each length of rebar is the core of this research and the core of the Revised-Best-fit-decreasing algorithm.

Finally, for the stock rebar used for the real case test bed, three most common rebars are used: 8m, 10m and 12m.

Dia (mm)	Length (m)	Unit Weight (kg/m)	Number of Order	Total Weight (ton)	Unit Price* (₩/kg)	Total Cost (₩)
13	12	1 (Fixed)	9	0.11	1,045 (Fixed)	100,320
	11		20	0.22		229,900
	⋮		⋮	⋮		
	6		130	0.78		815,100
Sum				**1.12**		**1,170,400**
22	12	3.04 (Fixed)	32	1.17		1,219,891
	11		11	0.37		384,393
	⋮		⋮	⋮		
	6		12	0.22		228,730
Sum				**2.58**		**2,696,100**

Figure 4.2 The process of calculating rebar order cost

4.2 Results and Resolutions

The test bed in Figure 4.1 above is used to calculate the overall order quantity of rebars using the Best-fit, Best-fit-decreasing, and Revised-Best-fit-decreasing algorithms. This value is compared to the existing 6% design surcharge to derive the overall order cost savings.

To compare the total order quantity of rebar derived by the optimization algorithm with the values in the rebar shop drawing, the resulting values are grouped by rebar diameter and summarized in Table 4.3 below.

It is found that the conventional Best-fit algorithm and Best-fit-decreasing algorithm slightly increase the total order volume compared to the values of 6% surcharge. There is no optimization effect as the overall order volume has increased.

However, it is confirmed that the newly proposed Revised-Best-fit-decreasing algorithm reduces the total order volume by about 4 tons compared to the 6% surcharge.

The Revised-Best-fit-decreasing algorithm can save about 4 tons of rebar order quantity in one caisson, resulting in a total order cost reduction of about 4.22 million won.

Table 4.3 Comparison of implementation values and optimization algorithm values

Diameter	Implementation Drawing		Best-fit	Best-fit-decreasing	**Revised-Best-fit-decreasing**	**Savings (₩)**
	Drawing (ton)	**6% surcharge (ton)**	Order quantity (ton)			
H13	11.045	**11.708**	11.701	11.247	**11.235**	**494,285**
H16	86.587	**91.782**	90.642	88.658	**88.040**	**3,910,390**
H19	43.789	**46.416**	49.158	46.764	**45.819**	**623,865**
H22	101.392	**107.476**	108.613	107.543	**107.215**	**272,475**
H25	31.142	**33.011**	37.579	37.579	**33.527**	**-539,220**
H29	34.029	**36.071**	41.378	41.016	**36.590**	**-542,355**
Total	307.984	**326.463**	339.072	332.807	**322.427**	**4,219,710**

From 13mm to 22mm, which has a slightly smaller rebar diameter, the rebar order quantity of the Revised-Best-fit-decreasing algorithm decreased compared to the 6% surcharge. However, at 25mm and 29mm, the total order volume increased with the Revised-Best-fit-decreasing algorithm.

The rebar diameter range in which Revised-Best-fit-decreasing algorithm shows good results in real case test is from 13mm to 22mm.

It is judged that the difference in the results for each diameter is due to the difference in the number of input data for each diameter and the unit weight.

Table 4.4 below shows the number of demand rebars required by rebar diameter and the order saving rates of rebar by diameter. The number of rebars in demand, that is, the diameters of 13mm to 22mm, which have many input data, have positive savings effect. However, there is a negative savings effect at 25mm and 29mm where the input data is small. Revised-Best-fit-decreasing algorithm is effective when there are enough demand rebars and many input data.

The larger the rebar diameter, the smaller the savings. The total weight of the rebar is calculated by multiplying the rebar length, the unit weight and the order quantity. Unit weight of rebar is related to the rebar diameter. As the diameter of the rebar increases, the unit weight of the rebar greatly affects the overall weight of the rebar. In such cases, the effectiveness of the algorithm in minimizing rebar order costs by reducing the number of rebar orders decreases.

The newly proposed Revised-Best-fit-decreasing algorithm is effective when the diameter of rebars is small, the number of demand rebars is sufficiently large, and sufficient input data is secured for optimization.

Table 4.4 Difference in saving rates according to the amount of data per diameter

Dia	**Pieces**	**Design (ton)**	**Revised-Best-fit-decreasing (ton)**	**Saving Rates (%)**
H13	5,183	11.708	11.235	4.21%
H16	20,209	91.782	88.040	4.25%
H19	4,397	46.416	45.819	1.30%
H22	4,583	107.476	107.215	0.24%
H25	1,087	33.011	33.527	-1.54%
H29	1,017	36.071	36.590	-1.42%

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDY

In an environment where rebar material supply is scarce and rebar prices are rising, optimizing rebar order quantity, and reducing rebar order costs can reduce a lot of construction costs.

Through this paper, I improve the Best-fit-decreasing algorithm of the existing Bin-packing approach for the purpose of minimizing rebar orders and propose a Revised-Best-fit-decreasing algorithm.

The Revised-Best-fit-decreasing algorithm is more effective in reducing rebar order cost than the existing Best-fit and Best-fit-decreasing algorithms. In particular, the Revised-Best-fit-decreasing algorithm was applied to the foundation caisson that was used as the actual test bed to optimize the order quantity of rebars, resulting in a quantity reduction effect of about 4 tons and a cost reduction effect of about 4.2 million won.

The Revised-Best-fit-decreasing algorithm is more effective when the quantity of demand rebar is high, and the diameter of rebar is small.

Applying this algorithm selectively where the diameter is small or where the number of rebars in demand is high can maximize the effect of reducing rebar order quantity and reducing rebar order cost.

Running the algorithm required the process of creating the Input data, which took a long time. Therefore, there are advantages in applying this algorithm to highly reproducible or simple structures. Since this algorithm is effective when the quantity of demand rebar is large, the cost reduction effect will be greater if the demand input data of multiple structures are applied to the algorithm at once.

Further study is planned to analyze the results of the algorithm based on the distribution of length of demand rebar. The degree of distribution, such as the standard deviation of the length of demand rebar, will affect the number of stock rebar orders.

Also, I proceed to study the minimum quantity of demand rebar that can derive a valid degree of optimization for the Revised-Best-fit-decreasing algorithm. Since the effectiveness of the algorithm varies with the number of demand rebars, a study of the minimum quantity of demand rebars applicable to this algorithm provides information on the number of input data to which the algorithm can be applied.

Finally, in conducting this research, I did not consider various incidental costs such as labor and transportation costs for processing rebars, but through additional study, I will consider not only the order quantity of simple rebars but also incidental costs.

REFERENCE

Kim, S. K., Hong, W. K., & Joo, J. K. (2004). Algorithms for reducing the waste rate of reinforcement bars. Journal of Asian Architecture and Building Engineering, 3(1), 17-23.

Gilmore, P. C., & Gomory, R. E. (1961). A Linear Programming Approach to the Cutting-Stock Problem. Operations Research, 9(6), 849–859.

Jahromi, M. H., Tavakkoli-Moghaddam, R., Makui, A., & Shamsi, A. (2012). Solving an one-dimensional cutting stock problem by simulated annealing and tabu search. Journal of Industrial Engineering International, 8(1), 1-8.

Fourer, R. (1988). A simplex algorithm for piecewise-linear programming II: Finiteness, feasibility and degeneracy. Mathematical Programming, 41(1), 281-315.

Martello, S., & Toth, P. (1990). Bin-packing problem. Knapsack problems: Algorithms and computer implementations, 221-245.Kato, Y., & Kanda, M. (2014). Development of a modified hybrid aerodynamic vibration technique for simulating aerodynamic vibration of structures in a wind tunnel. Journal of Wind Engineering and Industrial Aerodynamics, 135, 10-21.

Bekesi, J., Galambos, G., & Kellerer, H. (2000). A 5/4 linear time bin packing algorithm. Journal of Computer and System Sciences, 60(1), 145-160.

Dantzig, G. B., & Wolfe, P. (1960). Decomposition principle for linear programs. Operations research, 8(1), 101-111.

Gomory, R. E. (2010). Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem. In 50 Years of

Integer Programming 1958-2008 (pp. 77-103). Springer, Berlin, Heidelberg.

Poldi, K. C., & Arenales, M. N. (2009). Heuristics for the one-dimensional cutting stock problem with limited multiple stock lengths. Computers & operations research, 36(6), 2074-2081.

Johnson, D. S. (1973). Near-optimal bin packing algorithms (Doctoral dissertation, Massachusetts Institute of Technology).

Gilmore, P. C., & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem—Part II. Operations research, 11(6), 863-888.

Vanderbeck, F. (1999). Computational study of a column generation algorithm for bin packing and cutting stock problems. Mathematical Programming, 86(3), 565-594.

Nadoushani, Z. S. M., Hammad, A. W., Xiao, J., & Akbarnezhad, A. (2018). Minimizing cutting wastes of reinforcing steel bars through optimizing lap splicing within reinforced concrete elements. Construction and Building Materials, 185, 600-608.

Berberler, M. E., Nuriyev, U., & Yıldırım, A. (2011). A software for the one-dimensional cutting stock problem. Journal of King Saud University-Science, 23(1), 69-76.

Belov, G., & Scheithauer, G. (2002). A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. European Journal of Operational Research, 141(2), 274-294.

Zheng, C., Yi, C., & Lu, M. (2019). Integrated optimization of rebar detailing design and installation planning for waste reduction and productivity improvement. Automation in Construction, 101, 32-47.

Haessler, R. W. (1975). Controlling cutting pattern changes in one-dimensional trim problems. Operations Research, 23(3), 483-493.

Lee, D., Son, S., Kim, D., & Kim, S. (2020). Special-length-priority algorithm to minimize reinforcing bar-cutting waste for sustainable construction. Sustainability, 12(15), 5950.

Kwon, K., Kim, D., & Kim, S. (2021). Cutting Waste Minimization of Rebar for Sustainable Structural Work: A Systematic Literature Review. Sustainability, 13(11), 5929.

국문초록

최우석
건설환경공학부
서울대학교 대학원

최근 국내 철근의 가격이 꾸준하게 상승추세에 있는 환경 속에서 철근 공사비는 전체 공사비용의 약 16%를 차지하기에 철근공사비를 줄이는 것은 공사비 절감에 매우 중요한 요소이다. 이번 논문은 철근 주문 비용을 줄이기 위한 연구를 수행한다.

이를 위해 선형계획법의 풀이방식 중 하나인 Column-generation approach 와 빈-패킹 접근방식의 기존 대표적인 알고리즘에 대해 살펴보았다. 기존 Best-fit-decreasing 알고리즘을 개선해 Revised-Best-fit-decreasing 알고리즘을 새롭게 제안하였다. 새롭게 제안한 Revised-Best-fit-decreasing 의 경우 재고 철근의 가격을 재고 철근의 길이로 나눠 이 값이 가장 작은 재고 철근을 최적 철근으로 재선정하였고 수요 철근은 길이 순으로 내림차순 정렬하였다. 수요 철근이 새로운 재고 철근에 들어갈 때 먼저 앞서 선정한 최적 철근에 먼저 들어가게 된다. 또한, 한 번에 두 개의 수요 철근을 동시에 고려하여 두 개의 수요 철근 길이의 합이 최적 철근의 길이보다 크거나 같은 경우에는 두 수요 철근을 모두 담을 수 있는 재고 철근을 생성하는 알고리즘을 적용하였다.

위 개발된 알고리즘을 울산 신항 개발 공사의 케이슨 구조물에 적용해보았다. 위 케이슨 구조물의 수요 철근은 818 가지 종류로 구성되어 전체 36,478 개로 설계되어 있는데 이 철근들에 6%의 할증을 가한 약 326.463 톤의 물량이 실제 철근 주문에 실행된 물량이다. Revised-Best-fit-decreasing 알고리즘으로 철근 주문비용 최소화를 진행한 결과 약 322.427 톤의 재고 철근 주문 물량이 나와 약 4.1 톤의 물량 절감효과 그리고 약 420 만원의 주문 비용 절감효과가 발생한다.

철근의 직경 별로 알고리즘의 최적화 정도가 달라졌는데 철근의 직경이 작을수록, 알고리즘을 수행하는 데 투입되는 수요 철근의 개수가 많을수록 최소화 효과가 컸다. 이 논문에서 제안한 알고리즘을 적절하게 활용할 경우, 철근의 발주물량을 절감하고 철근 주문비용을 줄일 수 있다.

주요어: 철근; 패턴; 선형계획법; 빈-패킹 접근방식; Revised-Best-fit-decreasing;

Student Number: 2020-28658