CUTTING PATTERN GENERATION FOR REINFORCEMENT BARS USING INTENSIVE SEARCH ALGORITHM

V. BENJAORAN[1*], and S. BHOKHA[2]

[1]*Institute of Engineering, Suranaree University of Technology, Thailand*
[2]*Faculty of Engineering, Ubonratchathani University, Thailand*

ABSTRACT

Construction industry consumes a lot of steel bars through the reinforced-concrete structures. These reinforcement bars are cut to various demanded lengths and pieces. The cutting process creates a great amount of trim loss and waste. An optimal cutting plan with minimum total trim loss can be arranged through the set of cutting patterns and repetitive cutting times. For a given demand list, there are a vast number of feasible cutting patterns. This research introduces a new algorithm called Intensive Search Algorithm to generate a set of efficient cutting patterns. The algorithm intensively searches for any right demanded length which is lacking and includes it in the current patterns until complete. Test results show that the Intensive Search Algorithm can consistently produce a complete set of cutting patterns which contain all demanded lengths. The generated patterns have restricted trim loss and their diversities are evenly distributed over the different demanded lengths and quantities.

Keywords: pattern cut, reinforcement bars, cutting stock, trim loss.

1. INTRODUCTION

Reinforcement bars are main construction materials. They are available in standard lengths and are needed to be cut before used. The cutting process prepares the demanded items in various lengths and quantities according to the construction design. The process produces an amount of trim loss which depends heavily on the demand lists, the cutting patterns used, and workers' skills. The steel bar is a costly material compared with the others or even compared with workers' wage. Trim loss can increase the project cost without adding any value. Finally, project owners must pay for this through the project prices. Trim loss can also decrease the competitiveness and productivity of the contractors (Kulatunga et al. 2006). De Silva and Vithana (2008) suggested four approaches for the construction waste management such as avoid and reduction, reuse, recycle, and disposal. The best approach is the first one, avoid and reduction, because it addresses on the root cause and it is the cheapest. The cost of the avoid-and-reduction approach is just the planning and designing of materials uses. Therefore, the cutting plan of reinforcement bars should be made to minimize waste.

[*] Corresponding author and presenter: Email: vacharapoom@sut.ac.th

The cutting process of reinforcement bars is classified as One Dimensional Cutting Stock Problem (1D-CSP). The problem definition is to find the cutting plan which satisfies all demanded items and makes the least amount of trim loss. The demanded items are cut from the standard length bar (stock). The stocks (L_s) are in an unlimited amount with the same length of *LS* units. The demand list consists of *n* different lengths (L_i) which are (L_1, L_2, L_3,..., L_n) and any L_i is required with B_i pieces which are (B_1, B_2, B_3,..., B_n). All these demands must be cut.

The demand list of reinforcement bars normally has big *n* and B_i values which indicate a strong degree of heterogeneity of the demand assortment. 1D-CSP of reinforcement bars can be classified as (1/V/I/M) (Dyckhoff 1990) or Single Bin Size Bin Packing Problem (Wäscher et al. 2007). The suitable cutting plan for this type of problem is a pattern-based approach. The cutting patterns are arranged first. Each pattern is an arrangement of demanded lengths and quantities for one stock. A piece which has been left at end is a trim loss of that pattern. These patterns then are repeatedly cut to meet the demanded pieces of any length. The important issues of this assignment are how to make effective patterns and how many of them should be enough. For a set of demand list (1/V/I/M), there are a huge number of feasible cutting patterns. Once a set of effective patterns have been created, a proper optimization model is applied to determine the cutting repetitions of these patterns. This problem model can be defined as follows (Haessler and Sweeney 1991; Vahrenkamp 1996). Decision variables: X_j is the number of cutting times of the pattern P_j, for j = 1 to *m*. T_j is a trim loss of P_j. Objective function is to minimize $\sum_j^m (T_j X_j)$ which is subjected to: $(\sum_j^m (T_j X_j))_i = B_i$, for i = 1 to *n*, where $X_j \geq 0$ and $X_j \in \mathrm{N}$. This paper reports a new method of generating effective cutting patterns.

2. INTENSIVE SEARCH ALGORITHM

The pattern-based cutting plan requires a set of cutting patterns. The total trim loss and number of stocks used depends on these patterns. The best cutting plan makes the least total trim loss and uses the least number of stocks. It results from effective cutting patterns. Any feasible cutting pattern is defined as: P_j is a cutting pattern *j*, $P_j = [A_{1j}, A_{2j}, \ldots, A_{nj}]$, given that A_{ij} is the number of pieces of L_i and A_{ij} is a positive integer, and where $\sum_i^n (L_i \cdot A_{ij}) \leq L_s$. Trim loss of any pattern P_j is T_j which is defined as: $T_j = L_s - \sum_i^n (L_i \cdot A_{ij})$.

Pierce (1964) introduced an algorithm for arranging a large number of various valid cutting patterns. Some patterns may give short trim losses and some give long ones. The algorithm did not evaluate the efficiency of the pattern. If all these patterns are considered, it may be difficult to find the optimal cutting plan. Vahrenkamp (1996) defined a pattern efficient if it can give trim loss shorter than the shortest demanded length ($T_j < \min(L_i)$). He also presented the Random Search Algorithm for generating a set of efficient patterns as needed. The algorithm allows users to specify the acceptable trim loss of a pattern (T_w) which controls the trim losses of all generated patterns. He reported that the Random Search Algorithm could generate a number of efficient patterns within a short time.

However, the efficient patterns as defined above may not give a good cutting plan. The 1D-CSP requires to meet all demanded items. The heterogeneous demand list means that it consists of different demand lengths; some demanded lengths may be needed in several pieces and some in few pieces. The diversity of cutting patterns in the set should increase a chance of getting a good cutting plan. The Random Search Algorithm completely randomly arranges an efficient cutting pattern every time. The result is a set of less diverse patterns. The generated pattern tends to be composed of several small L_i, although they are required in a few pieces. Some L_i may be missing from the set of patterns. This means that the demand list cannot be satisfied completely. This research, therefore, aims to develop an algorithm for generating the efficient cutting patterns. This algorithm is based on the Random Search Algorithm. It is called "the Intensive Search Algorithm" which employs a weighted random wheel (confined random) to pick any L_i and include it into a pattern during the generation process. The weighted random wheel helps control the direction of searching and generating a diversified and efficient pattern according to the demanded pieces (B_i). Figure 1 shows the pseudo-code and the flowchart of the Intensive Search Algorithm.

An important step in the algorithm is to pick one L_i using a weighted random wheel. Any L_i is not completely randomly chosen but the chance of being picked is controlled by the weighted random wheel. This research initiated an index called the availability index of a L_i (V_i).

$$V_i = \begin{cases} B_i / \sum_i^c A_{ij}; \text{if } \sum_i^c A_{ij} > 0 \\ \text{else } 10000 \text{ or big number} \end{cases} \tag{1}$$

The availability index of a L_i is used to construct the weighted random wheel. Any L_i has its own portion of picking chance on this wheel which is proportional to its V_i value. The availability index is the ratio of the demanded pieces and the sum of pieces of L_i appearing in the current set of cutting patterns. If any L_i is still lacking in the current set, its V_i is high and this increases a chance of picking this L_i into the current cutting pattern. On the other hand, any L_i has already been included in the set in many pieces, its V_i decreased and this lessened chance of being picked again. The availability index is dynamic. It changes every time when L_i is put in the current cutting pattern. This index must decrease gradually through the generation process from being a big number at the start. For example, given that L_i = {0.95, 1.40, 1.75, 1.80, 1.88} meters and B_i = {25, 18, 14, 23, 7} pieces, assume that 10 cutting patterns have been already generated, they give the sum result as $\sum_j^c A_{ij}$ = {8, 3, 9, 4, 5}.

Input T_w, *nEffPat*, *nSLoop*
FOR j = 1 TO *nEffPat*
Construct the j^{th} pattern (P_j)
    Construct a current pattern (P_c)
    DO WHILE not exceed *nSLoop*
    $L_c = L_S$; $P_c = 0$; $A_{ic} = 0$
    DO WHILE $L_c >= \mathrm{Min}(L_i)$
        Consider only L_i which $L_i <= L_c$
        Pick one L_i using a weighted random wheel* (using V_i)
        Choose randomly integer k in range $[1, Min(L_c/L_i, B_i – A_{ic})]$
        Update $A_{ic} = A_{ic} + k$
        Update $L_c = L_c – kL_i$
    LOOP
    The current pattern is completed
    Evaluate the current pattern
        IF ($T_c <= T_w$) AND (P_c differs from the existing P_j) THEN
            Accept P_c; $P_j = P_c$
        ELSE
            Reject P_c; Start over P_c
        END IF
    LOOP
NEXT j

Where:
P_j = an efficient cutting pattern j; $P_j = [A_{1j}, A_{2j}, \ldots, A_{nj}]$; $j = 1, 2, 3, \ldots,$ *nEffPat*
A_{ij} = pieces of L_i of a P_j; A_{ij}
T_j = trim loss of a P_j ; $T_j = LS$ –
P_c = a current cutting pattern that is being created
T_c = trim loss of P_c
L_c = current length of a stock being cut
T_w = allowable trim loss of any efficient pattern
nEffPat = number of efficient patterns
nSLoop = number of maximum searching loops
V_i = availability index of a L_i

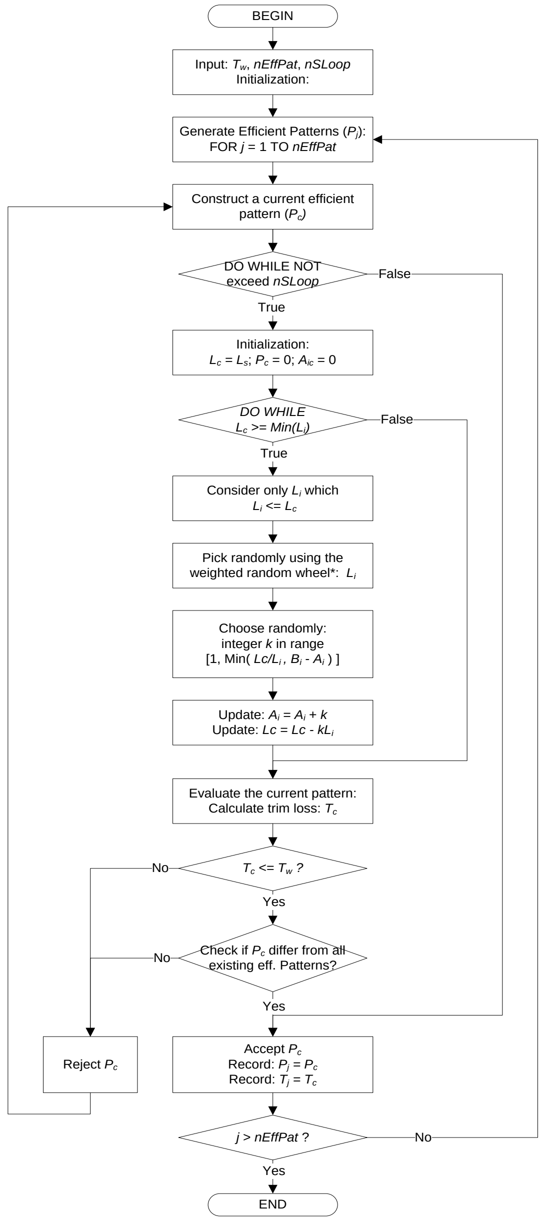

Figure 1: Pseudo-code and flowchart of Intensive Search Algorithm

At the current pattern (11^{th} pattern or $j = 11$), the availability indexes (V_i) can be calculated as = {3.13, 6.00, 1.56, 5.75, 1.40}. In case of any L_i with $\sum_j^c A_{ij} = 0$, its V_i is given as 10000 (or big number) because that length still do not appear in the existing cutting patterns. This big number of V_i is applied to force every L_i to be picked up quickly and included in the set of cutting patterns at least one piece.

Given that L_c = 1.85 meter at this round, therefore, only four L_i = {0.95, 1.40, 1.75, 1.80} which are shorter than L_c can be considered to be picked. The (L_i = 1.88) is excluded at the moment because it is longer than L_c (cannot cut 1.88 meter-long item from the 1.85 meter). Then, the weighted random wheel is created. Any L_i has its own picking probability which is proportional to the V_i. For example,

(L_i = 0.95) with V_i = 3.13 has a picking probability equal to (3.13 / (3.13 + 6.00 + 1.56 + 5.75) = 19%), (L_i = 1.40) with V_i = 6.00 has a picking probability equal to (6.00 / (3.13 + 6.00 + 1.56 + 5.75) = 37%), etc. In this round, the (L_i =1.40) gets the biggest chance to be selected and included in the 11th cutting pattern. Any L_i is selected by chance and its V_i will be updated (reduced) according to Equation (1) in the next round.

The approach for picking and including a L_i in the current cutting pattern in the generation process uses the weighted random wheel which is constructed on the availability index (V_i). The result of the Intensive Search Algorithm is the set of efficient and diverse cutting patterns. Each pattern is composed of various L_i according to the demanded quantities (B_i). This set of cutting patterns should finally give a good cutting plan with minimum trim loss. This Intensive Search Algorithm is a cutting-pattern generating tool which purposefully searches for the right L_i and includes it in the cutting patterns. It is improved on the Random Search Algorithm (Vahrenkamp 1996).

3. TEST AND RESULT

Data from a construction project were used for a test problem. The project was a reinforced concrete structured, high-rise building. The reinforcement details of foundations and beams were reviewed and quantity-taken off. These data gave a demand list of reinforcement bars. The demand list is prepared for one reasonable job and it was used as a test problem. This demand list represented an ordinary job of a reinforced concrete structure construction. The test problem consists of three essential data namely, various demanded lengths (L_i), with demanded pieces (B_i), and stock length (L_s) as shown in Figure 2. The stock length with unlimited supplies is 10 meters-long. The demand combines with 15 different lengths or n = 15, each of which is required in a different number of pieces. It is a total of 261 demanded pieces to be cut and the average is 17.4 demanded pieces per one length. The assortment of this demand list is divided into three ranges using a ratio of L_s/L_i i.e. >5, >2, and <=2. The numbers of demand pieces are distributed uniformly across these three ranges. The total of demanded length is 918.73 meters.

The Intensive Search Algorithm requires some parameters in the cutting pattern generation process. They are the allowable trim loss (T_w), the number of cutting patterns in the set (*nEffPat*), and the maximum searching loop (*nSLoop*). These parameters were specified in the test as T_w = 0.20; *nEffPat* = 30; and *nSLoop* = 500.

Figure 2 shows an example of the set of efficient cutting patterns resulting from the generation process. The $\sum_j^c A_{ij}$ and V_i are calculated and used in the Intensive Search Algorithm. These factors control the diversity of the generated patterns according to the demand. The efficient cutting patterns (P_j) are generated in a specified number of 30 (j = 1 to 30). Each pattern has its own trim loss (T_j) which is limited to be less than T_w.

i	Li	Bi	Pc	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	Sum(Aij)	Vi
1	0.95	25				1		1				1	9	1		1		3									2	1			1	7	1	29	0.86
2	1.40	18			2								1			1	1			2	1		1	2		3				1	2		1	18	1.00
3	1.75	14		2							1						2							4				1						10	1.40
4	1.80	23					1					2		1				1	2											3				10	2.30
5	1.88	7				1									2							1				3							4	11	0.64
6	2.67	22					3				1							2				1			1		1							9	2.44
7	2.88	10						2																					1					3	3.33
8	3.05	36							2						2						1									1	2			8	4.50
9	3.20	4						1															1									1		3	1.33
10	3.75	15							1							2																		3	5.00
11	5.00	26								2							1																	3	8.67
12	5.40	19									1	1									1	1	1				1							6	3.17
13	6.35	7		1															1															2	3.50
14	7.00	23				1														1									1					3	7.67
15	7.19	12			1									1											1			1						4	3.00
			Tc	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28	T29	T30		
			10.00	0.15	0.01	0.17	0.19	0.09	0.15	0.00	0.18	0.05	0.05	0.06	0.14	0.15	0.10	0.01	0.05	0.20	0.15	0.05	0.00	0.20	0.14	0.16	0.03	0.11	0.12	0.15	0.15	0.15	0.13		

Figure 2: A test problem and a set of 30 efficient cutting patterns being generated

The test was arranged into six separate sets. Three sets (1, 2, and 3) used the Random Search Algorithm in the pattern generation process. They were used as control groups. The other three sets (4, 5, and 6) used the Intensive Search Algorithm instead. Each test set would repetitively generate a hundred sets of 30 unique cutting patterns. A total of 3000 cutting patterns were generated in one test set. The results were then analyzed and compared. The first analysis was targeted on the trim loss of the patterns. Table 1 shows the averages and the standard deviations of the patterns' trim loss. It indicates that these six test sets give indifferent averages and standard deviations of the trim loss despite they uses the different algorithm. The results from different consecutive test sets also show that the pattern generation process can give a consistent result. The average of the patterns' trim losses is quite small and so is the standard deviation. Both algorithms can produce an efficient cutting pattern with small trim loss which is bounded by the allowable trim loss (T_w) specified as 0.20.

Table 1: Averages and standard deviations of trim loss, number of patterns with trim loss in different ranges, and number of complete and incomplete sets of cutting patterns and the average richness indices

Testset	Average	S.D.	Testset	=0	(0, 0.1]	(0.1, 0.2]	(0.2, 0.3]	(0.3, 0.4]	>0.4	Sum	Testset	Complete	Incomplete	Avg. H
1	0.097	0.063	1	339	1185	1476	0	0	0	3000	1	86	14	2.367
2	0.097	0.063	2	345	1188	1467	0	0	0	3000	2	83	17	2.379
3	0.099	0.062	3	331	1172	1497	0	0	0	3000	3	79	21	2.379
4	0.099	0.062	4	332	1118	1550	0	0	0	3000	4	100	0	2.526
5	0.099	0.063	5	343	1095	1562	0	0	0	3000	5	100	0	2.530
6	0.099	0.062	6	350	1086	1564	0	0	0	3000	6	100	0	2.530

Table 1 also shows the number of patterns generated from the six test sets classified by the length of the trim loss (T_j). All patterns generated have trim loss smaller than the allowable trim loss. The allowable trim loss is an effective control of the pattern's trim loss. Both algorithms give a slightly different number of patterns in different ranges i.e. $T_j = 0$; $0 < T_j <= 0.1$; and $0.1 < T_j <= 0.2$.

The second analysis was aimed to quantify the diversity of the demanded lengths in the cutting patterns generated. The Shannon-Weiner' richness index (Spellerberg and Fedor 2003) was used in this analysis. This index has been a popular diversity index in the ecological literature. It is a quantitative measure that reflects how many different types there are in a dataset, and simultaneously takes into account how evenly the basic entities are distributed among those types. The value of a diversity index increases both when the number of types increases and when evenness increases. For a given number of types, the value of a diversity index is maximized when all types are equally abundant. The index is calculated as follows.

$$H = -\sum_{i=1}^{n} Q_i \cdot \ln(Q_i) \quad (3)$$

Where: H = the Shannon-Weiner' richness index, and Q_i = the proportion of quantity of the i^{th} type in the dataset.

Not only should a good set of cutting patterns give small trim loss but it should consist of diverse demanded lengths according to the demanded pieces. The richness index of the demand list is calculated as 2.585.

A good set of cutting patterns should give the richness index close to this value which means that set consists of a variety of demanded lengths and pieces similar to the demand list. In this analysis, the richness index of a set of cutting patterns generated is calculated using the availability index (V_i) of any L_i as the Q_i in (3). Each set of the cutting patterns has its own value of the H index; therefore, each test set has one hundred values of the H index. The results of average H index of each test set are shown in Table 1.

Since the Random Search Algorithm constructs a pattern regardless of the demanded pieces, a set of 30 generated patterns could lack some demanded lengths. A set of patterns which does not include all demanded lengths (every demanded length must be included at least one piece) is incomplete. The incomplete set definitely cannot satisfy the entire demand list. It will not cut any missing demanded length. The incomplete set hence is invalid and it cannot make a valid cutting plan. The numbers of complete and incomplete sets of cutting patterns generated in the six test sets are also shown in Table 1. The incomplete set does not have the H index because it cannot be calculated; therefore the average H indices of the test set 1, 2, and 3 are calculated from the complete sets only. The results indicate that the Intensive Search Algorithm can always generate a complete set of cutting patterns while the Random Search Algorithm cannot. The average Hs of test sets 1, 2, and 3 are smaller than test sets 4, 5, and 6. This implies that the sets of cutting patterns from the Intensive Search Algorithm are more diverse than the ones from the Random Search Algorithm. Also, the average Hs of test sets 4, 5, and 6 are close to the one of the demand list. This tells that their diversities are similar.

4. CONCLUSIONS

This research developed the Intensive Search Algorithm for the cutting pattern generation process. This Intensive Search Algorithm was based on the existing Random Search Algorithm. The new algorithm was designed to construct a pattern according to the current demand. It used a weighted random wheel to pick any demand length and include it into the current pattern. This algorithm and the cutting stock problem model were developed on spreadsheet software with embedded macros. The algorithm was tested and compared with the control. Test results showed that the Intensive Search Algorithm could consistently generate a set of efficient cutting patterns which had trim loss bounded by the allowable trim loss. By applying the weighted random wheel in the algorithm, the set of patterns always contains all demanded lengths and the numbers of pieces of these lengths were well distributed according to the demand list. The diversity of the demanded lengths in the set was also similar to the demand list. These results indicated that the Intensive Search Algorithm is superior to the Random Search Algorithm. Therefore, the Intensive Search Algorithm can generate a set of efficient cutting patterns for the demand list of reinforcement bars and this set can then be used to make an optimal cutting plan with minimum trim loss and satisfying the entire demand list.

5. ACKNOWLEDGMENTS

This research is funded by The Commission on Higher Education and The Thailand Research Fund (MRG53-80065).

REFERENCES

Kulatunga U, Amaratunga D, Haigh R, and Rameezdeen R (2006). Attitudes and Perceptions of Construction Workforce on Construction Waste in Sri Lanka. Management of Environmental Quality: An International Journal. 17(1), pp. 57–72.

De Silva N, and Vithana S (2008). Use of PC Elements for Waste Minimization in the Sri Lankan Construction Industry. Structural Survey. 26(3), pp. 188–198.

Dyckhoff H (1990). A typology of cutting and packing problems. European Journal of Operational Research. 44(2), pp. 145–159.

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

Haessler RW, and Sweeney PE (1991). Cutting stock problems and solution procedures. European Journal of Operational Research. 54(2), pp. 141–150.

Vahrenkamp R (1996). Random search in the one-dimensional cutting stock problem. European Journal of Operational Research. 95(1), pp. 191–200.

Pierce JF (1964). Some large-scale production scheduling problems in the paper industry. Prentice-Hall.

Spellerberg IF, and Fedor PJ (2003). A Tribute to Claude Shannon (1916–2001) and a Plea for More Rigorous Use of Species Richness, Species Diversity and the ‘Shannon–Wiener’Index. Global Ecology and Biogeography. 12(3), pp. 177–179.