Journal of Mathematics and Statistics 6 (2): 79-83, 2010
ISSN 1549-3644

Solving One-Dimensional Cutting Stock Problem with Discrete Demands and Capacitated Planning Objective

[1]Sirirat Wongprakornkul and [2]Peerayuth Charnsethikul
[1]Department of Statistics, Faculty of Science, Khon Kaen University, 40002, Thailand
[2]Department of Industrial Engineering, Faculty of Engineering,
Operations Research and Management Science Units, Kasetsart University, 10900, Thailand

Abstract: Problem statement: One-dimensional cutting stock problem with discrete demands and capacitated planning objective is an NP hard problem. **Approach:** The mathematical model with column-generation technique by a branch-and-bound procedure and the heuristic based on the first fit decreasing method are proposed. Then, both approaches were compared and some characteristics were investigated such as upper-bound value, percentage above lower-bound value, computation time, and number of patterns. **Results:** The 24 instances were examined. The proposed heuristic provides the upper-bound value above the lower-bound around 0-16.78%. All upper-bound values from column-generation and integer programming are better than the proposed heuristic but all computation times are higher. **Conclusion:** The proposed heuristic has consistently high performance in computation times.

Key words: Stochastic integer linear programming, cutting stock problems, column-generation technique, heuristics

INTRODUCTION

More than four decades ago, mathematicians studied intensively and developed techniques involved with large-scale problems. The extensive work on large-scale mathematical programming has been initiated by (Dantzig and Wolfe, 1961). One of classical NP-hard problems which could not be solved within the polynomial computation time is a cutting stock problem (CSP). The finite number of cutting patterns may be very large. Gilmore and Gomory (1961; 1963) claimed that if the stock to be cut has length l = 200 inches and there are demands for 40 different lengths from 20-80 inches, the number of possible patterns could be exceed 10-100 million. The more cutting patterns and the more columns of constraints are involved. The most powerful algorithm for solving linear programs with many columns is the Column-Generation Procedure. To apply in the real world problem, uncertainty is inherent in many real combinatorial problems. Uncertainty seems to be an important issue in constraint programming. The Stochastic CSP (SCSP) is a framework that can be used to model combinatorial decision problems involving uncertainty and probabilities recently. Therefore, the number of scenarios depends on the number of random variables and the discrete random choices. The increasing of number of scenarios is tremendously such, 5 variables and 2 discrete demand choices have made 32 scenarios, 10 variables and 2 discrete demand choices have made 1,024 scenarios, but 20 variables and 2 discrete demand choices have made 1,048,576 scenarios. So the specific algorithms must be concerned for solving it.

In this study, we investigate the mathematical model of 1-dimensional cutting stock problem with discrete random demands and propose the column-generation technique for searching the effective cutting patterns. The model is based on “here and now” assumption with minimizing capacitated planning objective. The objectives of this study are to propose a mathematical programming with column-generation technique for solving the stochastic with discrete demands and capacitated planning objective, to propose a heuristic based on the first fit decreasing method and to compare between the mathematical programming method and the proposed heuristic method. In materials and methods shows two algorithms; the mathematical programming with column-generation and the constructive heuristic method. The result compares the computational time of both algorithms on the test problem with various retail items and choices of demands. Sequentially, conclusions are drawn.

MATERIALS AND METHODS

There are two algorithms proposed in this study which are the method of column-generation technique and the proposed heuristic. Both algorithms were coded

Corresponding Author: Sirirat Wongprakornkul, Department of Statistics, Faculty of Science, Khon Kaen University, Khon Kaen, 40002, Thailand

in the C++ programming and run on Pentium 4, 256 MB, 2.4 GHz. The column-generation procedure is linked with LINGO 6.0 solver. Their methodologies can be shown as follows:

The mathematical programming with column-generation technique: Consider the 1D-CSP with discrete demands and minimized capacitated planning objective. The following assumptions are made:

- There is different length of items to be cut from a stock
- Each item has associated a certain length
- Each cutting pattern for a stock are not limited in the number of knives, but the sum of length of items are not exceed a length of stock
- The discrete demands are considered
- The cost of capacity planning such as raw material or stock, inventory and backorder costs are considered

To describe the problem, we introduce the following notations:

k = Index for stock lengths ($k = 1, \ldots, K$)
m = Index for retail items ($m = 1, \ldots, M$)
L_k = The length of stock k; ($k = 1, \ldots, K$)
l_m = The length of item m; ($m = 1, \ldots, M$)
g_k = The cost of stock k
R_k = Amount of stock k available
p = Index for patterns
P_k = The number of feasible patterns for stock k
a_{mkp} = The number of strips of item m cut in pattern p for stock length k, fulfilling $\sum_{m=1}^{M} l_m a_{mkp} \le L_k$
s = Index for scenario of discrete demands ($s = 1,., S$)
F_{ms} = The probability of demand of item m at a scenario s, it equals $\prod_{h=1}^{H} f_{hm}$
h = Index for choices of discrete random demands ($h = 1,\ldots, H$)
f_{hm} = The probability of choice h of item m
I_{ms} = The inventory units of item m at scenario s
T_{ms} = The inventory costs of item m at scenario s
B_{ms} = The backorder units of item m at scenario s
A_{ms} = The backorder units of item m at scenario s
D_{ms} = The demand of item m at scenario s
x_{kp} = Decision variables that represent the number of times pattern p of stock k is used

Minimize:

$$\sum_{k=1}^{K} \sum_{p=1}^{Pk} g_k x_{kp} + \sum_{m=1}^{M} \sum_{s=1}^{S} F_{ms}(T_{ms} I_{ms} + A_{ms} B_{ms}) \qquad (1)$$

Subject to:

$$\sum_{p=1}^{P_k} x_{kp} \le R_k \ \forall k \qquad (2)$$

$$\sum_{k=1}^{K} \sum_{p=1}^{P_k} a_{mkp} x_{kp} - I_{ms} + B_{ms} = D_{ms}, \ \forall m, \ \forall s \qquad (3)$$

$$x_{kp} \ge 0, \ \forall k, \ p \qquad (4)$$

For a large instance, P_k can be extremely large. To reduce the number of P_k, we are not adding all feasible patterns in the model, but selecting some necessary patterns p* that maximized its dual objective function. To determine $a_{mk^*p^*}$ we develop the knapsack based model using the simplex multipliers obtained from (1-4) as follows:

e_k = The simplex multipliers of the used stock k from constraints (2)
r_{ms} = The simplex multipliers of the item m from constraints (3)
$a_{mk^*p^*}$ = The obtained variable that defined number of strips of item m cut in pattern p* for stock length k*
q_k = The used stock length; $q_k \in \{0,1\}$
if q_k = 1 then the stock length k is used. Otherwise, the stock length k is not used

Minimize:

$$V = \sum_{k=1}^{K} (g_k - e_k) q_k - \sum_{s=1}^{S} \sum_{m=1}^{M} r_{ms} a_{mk^*p^*} \qquad (5)$$

Subject to:

$$\sum_{m=1}^{M} l_m a_{mk^*p^*} \le \sum_{k=1}^{K} L_k q_k \qquad (6)$$

$$\sum_{k=1}^{K} q_k = 1 \qquad (7)$$

$$a_{mk^*p^*} \ge 0 \text{ and integer, } m = 1, \ldots, M \qquad (8)$$

$$q_k \ge 0 \text{ and binary, } k = 1,\ldots,K \qquad (9)$$

The obtained output from (5-9) is a new legitimate cutting pattern and introduced into model (1-4) as a new basis column. The new cutting pattern is to be adding until the objective dual values V from (5)≥0, the current column is proven to be globally linear programming optimal point. For a profound description of the column-generation (CG) procedure for cutting stock problems, Lasdon (1970). To illustrate this CG procedure, a flow chart is shown in Fig. 1.

Since the obtained x_{kp} from globally linear programming optimal point usually are not integer therefore, the branch-and-bound procedure is applied for selecting integer values of x_{kp} by the same basis column. The obtained integer solution from the branch-and-bound procedure by the same basis column or necessary patterns are not providing the optimality that proven by solving all the feasible patterns, but it nearly the optimal value than the other heuristics.

The constructive heuristic method: Although the column-generation approach provides the nearly optimal solution, variables and constraints increase drastically when increasing the number of items and the chance of discrete demand. Therefore, the obtained solution from the column-generation approach is not always attainable within the allowable time.

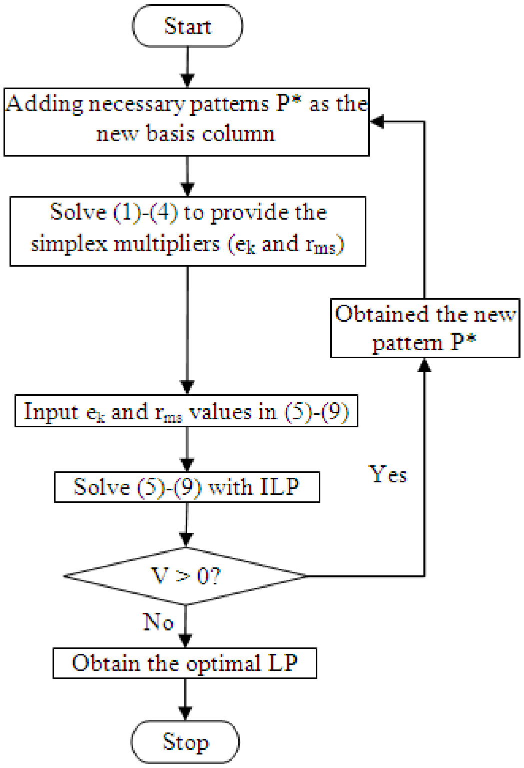
Start
Adding necessary patterns P* as the new basis column
Solve (1)-(4) to provide the simplex multipliers (e_k and r_{ms})
Obtained the new pattern P*
Input e_k and r_{ms} values in (5)-(9)
Solve (5)-(9) with ILP
Yes
V > 0?
No
Obtain the optimal LP
Stop

Fig. 1: Flow chart of column-generation procedure for the problem

The proposed heuristic is based on first fit decreasing method of Bin Packing problem (Brandimarte and Villa, 1995). Items are sorted in such a way that the longer item is selected before others. The outline of the algorithm has been shown in Fig. 2.

The algorithm is presented in the following steps:

Step 1: Selected the longest length item from the item list and the stock size of lowest cost from the stock list. Go to Step 2

Step 2: Knifed the selected item on the selected stock size from step 1 until the remainder length of stock size is not sufficient. Go to Step 3

Step 3: From the remainder length of stock size, searching the longest length item that sufficient from all items and knives it as step 2. Repeated this step until the remainder length is zero or the item are not selected. Go to Step 4

Step 4: Adding one cutting pattern from step 3 to the pattern list. Remove the longest length item from the item list. If the item list is empty go to Step 5, else return to Step 1

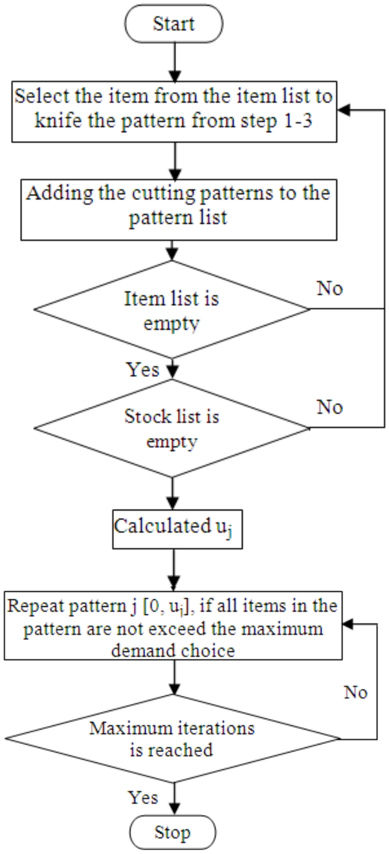
Start
Select the item from the item list to knife the pattern from step 1-3
Adding the cutting patterns to the pattern list
Item list is empty
No
Yes
Stock list is empty
No
Calculated u_j
Repeat pattern j [0, u_j], if all items in the pattern are not exceed the maximum demand choice
No
Maximum iterations is reached
Yes
Stop

Fig. 2: Flow chart of the heuristic for the problem

Step 5: Return all items into the item list but remove the stock size of lowest cost from the stock list. If the stock list is empty go to Step 6, else return to Step 1

Step 6: From all patterns in the pattern list, calculated the maximum usage value of pattern j (u_j) as:

$\bar{d}_z$ = Average demand of item z

n_z = Number of knives of item z

$$u_j = \frac{\bar{d}_z}{n_z} \tag{10}$$

And set the iteration index as 1 then go to Step 7.

Step 7: Sorting the pattern from the pattern list as randomly. If all items in the selected pattern j are not exceed the maximum demand choices, then repeat the selected pattern j from the sorting order randomly between 0 and u_j. If the pattern in the last order is reached, then calculated cost as objective (1). Go to Step 8

Step 8: If the maximum iteration is reached then STOP else increasing the iteration index and go to Step 7

RESULTS

The 24 instances were considered and separated into 3 groups of eight instances. The first group is performed with 10 retail items; all items are 2 choices of demands with 50% of probability. For each instance, all items have the same demand are such (50, 130), (50, 120), (50, 110), (50, 100), (50, 90), (50, 80), (50, 70), (50, 60). All possible scenarios are 1,024. The second group is performed with 6 retail items; all items are 3 choices of demands with probabilities 0.3, 0.3, and 0.4, respectively. For each instance, all items having the same demand are such (50, 90, 130), (50, 90, 120), (50, 90, 110), (50, 80, 130), (50, 80, 120), (50, 80, 110), (50, 70, 130), (50, 70, 120). All possible scenarios are 729. The last group is performed with 5 retail items; all items have 4 choices of demands with probabilities 0.3, 0.3, 0.2, and 0.2, respectively. For each instance, all items have the same demand are such (50, 70, 100, 130), (50, 70, 100, 120), (50, 70, 90, 130), (50, 70, 90, 120), (50, 70, 90, 110), (50, 80, 100, 130), (50, 80, 100, 120), (50, 80, 90, 120). All possible scenarios are 1,024. For all instances, the stock length is 400, the stock cost is 100, the limited of stocks is 1000, inventory cost is 10 and backorder cost is 50.

The lower-bound is computed by solving linear programming model (1-4). The limited time for a branch-and-bound procedure to obtain all integer solutions is 360 seconds. For the proposed heuristic, we set the maximum iterations of randomly generate the repeated cutting pattern to 10.

In order to compare the performance between column-generation technique and proposed heuristic, the upper-bound value and computation times are shown in Table 1-3. From the results, the proposed heuristic provides the upper-bound value above the lower-bound 0.00-16.78% approximately. The upper-bound value from column-generation and integer programming is better than the proposed heuristic but the computation times are higher.

Table 1: Eight instances with 10 retail items and 2 varied demands

			Column generation + ILP				Proposed heuristic			
Problem	Demand 1	Demand 2	UB	LB	Time (sec)	Patterns	UB	Time (sec)	Patterns	Above LB (%)
Test 1	50	130	35700	35687	360	19	39290	29	10	10.10
Test 2	50	120	33250	33212	360	18	36200	29	10	9.00
Test 3	50	110	30620	30620	153	19	33690	29	10	10.03
Test 4	50	100	28140	28140	83	18	31220	29	10	10.95
Test 5	50	90	25670	25670	74	18	28160	29	10	9.70
Test 6	50	80	23200	23200	79	18	26470	29	10	14.09
Test 7	50	70	20720	20720	82	18	23050	29	10	11.25
Test 8	50	60	18240	18240	76	18	21300	29	10	16.78

Note: Time (sec): Computing times in seconds; Patterns: The number of generated cutting patterns

Table 2: Eight instances with 6 retail items and 3 varied demands

				Column generation + ILP				Proposed heuristic			
Problem	Demand 1	Demand 2	Demand 3	UB	LB	Time (sec)	Patterns	UB	Time (sec)	Patterns	Above LB (%)
Test 9	50	90	130	18374	18374	179	11	19596	29	6	6.65
Test 10	50	90	120	17174	17174	179	11	18350	29	6	6.85
Test 11	50	90	110	15974	15974	179	11	17050	29	6	6.74
Test 12	50	80	130	17940	17940	33	11	19586	29	6	9.18
Test 13	50	80	120	18034	18034	33	11	18034	29	6	0.00
Test 14	50	80	110	16656	16656	33	11	16656	29	6	0.00
Test 15	50	70	130	17560	17560	120	11	19010	29	6	8.26
Test 16	50	70	120	16360	16360	120	11	17964	29	6	9.80

Note: Time (sec): Computing times in seconds; Patterns: The number of generated cutting patterns

Table 3: Eight instances with 5 retail items and 4 varied demands

					Column generation + ILP				Proposed heuristic			
Problem	Demand 1	Demand 2	Demand 3	Demand 4	UB	LB	Time (sec)	Patterns	UB	Time (sec)	Patterns	Above LB (%)
Test 17	50	70	100	130	12686	12686	37	9	13956	29	5	10.01
Test 18	50	70	100	120	13192	13192	37	9	13192	29	5	0.00
Test 19	50	70	90	130	12186	12186	37	9	13312	29	5	9.24
Test 20	50	70	90	120	11686	11686	37	9	12756	29	5	9.16
Test 21	50	70	90	110	11186	11186	37	9	12052	29	5	7.74
Test 22	50	80	100	130	13950	13950	37	9	13950	29	5	0.00
Test 23	50	80	100	120	13306	13306	37	9	13306	29	5	0.00
Test 24	50	80	90	120	12870	12870	37	9	12870	29	5	0.00

Note: Time (sec): Computing times in seconds; Patterns: The number of generated cutting patterns

DISCUSSION

From Table 1-3, 24 instances were tested. The computation times of all instances are not always related with various retail items and choices of demands. The column-generation and integer linear programming provides more patterns than the proposed heuristic. The numbers of scenarios for each table are 1,024, 729, and 1,024, respectively. For the mathematical model, the number of variables and constraints after adding the basis column to linear optimal of all groups are (20,498, 10,241), (8,759, 4,375), and (10,249, 5,121) respectively. That causes the computation times of all instances, solved by the proposed heuristic, are almost equal and also less than those solved by the column-generation and integer linear programming.

CONCLUSION

In this study, we have examined the problem of minimizing the number of cutting stocks and capacitated planning cost such inventory and backorder. The 24 instances were experimented into three groups, varying by number of retail items and number of uncertain demands; (10, 2), (6, 3), and (5, 4). All test problems are solved by column-generation technique and the proposed constructive heuristic. For the column-generation technique, the linear programming model is proposed as (1-4) and the new cutting pattern are adding as the new basis column by solving (5-9), the lower-bound is obtained when stop adding the new basis column and the upper-bound is obtained by applied branch-and-bound steps for rounding variables to integer. For the proposed heuristic, it has consistently high performance in computation times but the obtained upper-bounds are not closed to the lower-bound as the column-generation technique. Nevertheless, the proposed heuristic should be more appropriate to solve a large instance because of the quick response requirement.

ACKNOWLEDGEMENT

We would like to acknowledge the Thailand Research Fund (TRF) for providing the grant to support this research.

REFERENCES

Brandimarte, P. and A. Villa, 1995. Advanced Models for Manufacturing Systems Management. 1st Edn., CRC Press, Inc., USA., ISBN: 0849383323.

Dantzig, G.B. and P. Wolfe, 1961. The decomposition algorithm for linear programs. Econometrica, 29: 767-778. http://www.jstor.org/pss/1911818

Gilmore, P.C. and R.E. Gomory, 1961. A linear programming approach to the cutting stock problem. Operat. Res., 9: 849-859. DOI: 10.1287/opre.9.6.849

Gilmore, P.C. and R.E. Gomory, 1963. A linear programming approach to the cutting stock problem-part II. Operat. Res., 11: 863-888. DOI: 10.1287/opre.11.6.863

Lasdon, S.L., 1970. Optimization Theory for Large Systems. 1st Edn., Macmillan Publishing, New York, USA., ISBN: 10: 0486419991.