

Optimization Based Heuristic Approaches for Solving an Integrated One-dimensional Cutting Stock-Transportation Problem

¹Sirirat Wongprakornkul and ²Peerayuth Charnsethikul

¹Department of Statistics, Khon Kaen University, Khon Kaen 40002, Thailand

²OR/MS Units, Department of Industrial Engineering,
Kasetsart University, Bangkok 10903, Thailand

Abstract: In this work, the integration of the one-dimensional cutting stock problem with multiple cutting facilities and the transportation problem was formulated mathematically as a large-scale discrete optimization problem. Benders partitioning approach and the column-generation technique with the direct method and the proposed heuristic method for solving corresponding integer programming (IP) were developed into three approaches and were used to solve a set of various sizes test problems within a controllable computation time. The computation time and the relative-difference percentage between the lower and upper bounds are criterions. The results indicated that the approach based on the column-generation technique with the proposed heuristic method is the most efficient method for solving this studied large-scale problems. Hence, this approach could be used in practical manners; to manage both production and transportation plans simultaneously.

Key words: Large-scale linear programming, column-generation technique, one-dimensional cutting stock problem, transportation problem, Benders decomposition.

INTRODUCTION

In order to solve large-scale problems, the big issue concerned is how long the computational time will be consumed and how effective the obtained solution can be. Absolutely, all industries need to know how to plan the process to reach the goal of minimal cost and meeting all demand requirements. However, the computation to find out the solution should be also within a reasonable time. More than four decades ago, mathematicians studied intensively and developed techniques involved with large-scale problems. Dantzig and Wolfe^[1] initiated the extensive work on large-scale mathematical programming. One of the most efficient technique when applied to linear programs is the column generation technique. It is applied for solving the problems with much fewer rows as compared to many more columns. There are wide variety of situations that could be solved by this approach. Holthaus^[2] proposed the decomposition approaches based on the classical column-generation technique for solving the integer one-dimensional cutting stock problem with different types of standard lengths.

Benders^[3] proposed another partitioning and decomposition technique to divide the original problem into subproblems with smaller sizes. This approach is applied in a variety of applications such as in cellular manufacturing system design^[4]. Benders cuts are effective for fixed-charge capacitated network design problem under heavy traffic load^[5]. Benders decomposition approach is a quick approach and an effective approach for modeling and solving the problems using resource-directive based approaches^[6].

An application of large-scale problem in this study is due to industries that have many plants and each plant has to ship the products to many customers both production plans and transportation plans must be produced cooperatively. In other words, the one-dimensional cutting stock problem (1D-CSP) with multiple cutting facilities and the transportation problem (TP) are integrated. Simply optimizing the cutting operation may not be the best approach as it may result in excessive transportation costs. Therefore, this study considered the 1D-CSP with multiple cutting facilities and the TP simultaneously, which are intractable because of the large number of variables involved. Both problems are combinatorial problems,

Corresponding Author: Peerayuth Charnsethikul, OR/MS Units, IE Department, Kasetsart University, Bangkok 10900, Thailand

which are solvable in non-polynomial time (NP-Hard) for the case of 1D-CSP and within a polynomial time for the case of linear cost TP.

The 1D-CSP can be addressed by finding the best process with the minimum cost to cut stock material (stock) to meet the demand for smaller pieces (retail size) of prescribed dimensions. The column-generation technique is successful for handling the large number of involved variables in the linear programming formulation of the problem^[7,8]. The 1D-CSP is broadly applied in industries such as in the aluminum industry^[9] and the clothing industry^[10]. Originally, cutting/trimming stock problems implicitly assume that the stocks are cut and processed at a single facility. Holthaus^[2] raised decomposition approaches based on the column-generation technique for solving the 1D-CSP with different types of standard lengths or multiple cutting facilities.

The classical TP is to find the minimum transportation cost for shipping units from suppliers to customers so that the demands of all the customers are met without exceeding each supplier capacity. Adlakha and Kowaski^[11] demonstrated the practicality of identifying cases where the paradoxical situation exists in the transportation problems. Sharma and Sharma^[12] proposed a heuristic based approach that runs in polynomial time for enhancing the dual solution of the TP.

In this study, situations where there are multiple cutting facilities that cut stock with various dimension into small retail size pieces are considered. These pieces are shipped to various customers at different locations to satisfy a set of known deterministic demands. In general, it is not the optimum approach to decompose this problem by solving the cutting stock problem and then solve the resulting transportation problem for each retail size. Instead, an efficient procedure is required to make joint cutting and distribution decisions be developed to minimize the total cutting and shipping cost. Therefore, the global objectives of this study are to develop heuristic approaches and to identify the most appropriate approach for solving integrated one-dimensional cutting stock problem-transportation problem, (1D-CSP&TP). The specific supporting objective of this study is to generate an integrated mathematical model for 1D-CSP&TP as the basic foundation for heuristics development.

Problem Formulation

The 1D-CSP&TP can be stated as a distribution system in which there are I suppliers and J

customers. Each supplier has known supply of stocks of length L_1, L_2, \dots, L_K available. Each customer has a demand for small size items of length l_1, l_2, \dots, l_M . The joint cutting stock-transportation problem is to decide the cutting patterns and the associated frequency to be used at each supplier and the number of units of each retail size items to ship from each supplier to each customer in order to minimize the total cost. Here the total cost includes the cost of stocks used as well as the transportation cost. To develop a mathematical model for this problem, some notations used throughout this paper are defined as follows:

i = index for suppliers ($i = 1, 2, \dots, I$),

j = index for customers ($j = 1, 2, \dots, J$),

k = index for stock lengths ($k = 1, \dots, K$),

m = index for retail items ($m = 1, \dots, M$),

L_k = the length of stock k ; $k = 1, \dots, K$,

l_m = the length of item m ; $m = 1, \dots, M$,

g_k^i = the cost of stock k at supplier i ,

R_k^i = amount of stock k available at supplier i ,

p = index for patterns,

P_k = the number of feasible cutting patterns of stock k ,

a_{mkp}^i = the number of strips of item m cut in pattern p

for stock length k by supplier i ,

fulfilling $\sum_{m=1}^M l_m a_{mkp}^i \leq L_k$,

D_{mj} = the demands of item m at customer j ,

c_{ij}^m = the unit cost of shipping one unit of item m from supplier i to customer j ,

x_{kp}^i = decision variables that represent the number of times pattern p of stock k is used at supplier i ,

y_{ij}^m = decision variables representing the amount of items m shipping from supplier i to customer j .

Using the notations above, 1D-CSP&TP can be modeled as:[1D-CSP&TP]:

$$\text{Minimize } \sum_{i=1}^I \sum_{k=1}^K \sum_{p=1}^{P_k} g_k^i x_{kp}^i + \sum_{m=1}^M \sum_{i=1}^I \sum_{j=1}^J c_{ij}^m y_{ij}^m, \quad (1)$$

Subject to

$$-\sum_{p=1}^{P_k} x_{kp}^i \geq -R_k^i, \quad \forall i, k, \quad (2)$$

$$\sum_{k=1}^K \sum_{p=1}^{P_k} a_{mkp}^i x_{kp}^i - \sum_{j=1}^J y_{ij}^m \geq 0, \quad \forall m, i, \quad (3)$$

$$\sum_{i=1}^I y_{ij}^m = D_{mj}, \quad \forall m, j, \quad (4)$$

$$x_{kp}^i \geq 0 \text{ and integer, } \forall i, k, p, \quad (5)$$

$$y_{ij}^m \geq 0, \quad \forall m, i, j. \quad (6)$$

The objective function (1) meets the minimum expenses: stock costs and shipping costs. Constraints (2) ensure that number of stocks cut at each facility does not exceed the availability of each stock type. Constraints (3) ensure that number of units of each item cut at each facility as a function of cutting decisions (x_{kp}^i) provides capacities limitation for transportation decisions (y_{ij}^m). Constraints (4) are the usual supply and demand constraints for transportation decisions for each retail size item. Constraints (5)-(6) ensure that all variables are nonnegative and for all x_{kp}^i variables are integers.

From now on, [1D-CSP&TP] represents the full Mathematical Model of 1D-CSP&TP, and P[I,J,K,M] represents the instance of [1D-CSP&TP] with I suppliers, J customers, K stock lengths, and M retail lengths. The next section, three approaches are schemed for solving [1D-CSP&TP].

Column-Generation Technique: Consider the procedure to apply the column-generation technique to the [1D-CSP&TP] so that at least x_{kp}^i pieces of standard length L_k are furnished. First, the problem is relaxed all integer restrictions in order to provide the simplex multipliers. After that, the problem is decomposed to I subproblems, called [NPP_i]: a new-pattern generated subproblem at supplier i. These subproblems applied the column-generation technique to search the essential cutting patterns for the problem.

The model (1)-(6) is rewritten in the relaxation form by ignoring all integer restrictions. Following the rules to convert from the primal problem to the dual problem, the DUAL model is written in the following (7)-(12).

To develop the corresponding DUAL model, the simplex multipliers used throughout this paper are defined as follows:

- e_{ik} = the simplex multipliers of supplier i and used stock k from constraints (2),
- r_{im} = the simplex multipliers of supplier i and item m from constraints (3), and
- b_{mj} = the simplex multipliers of item m and customer j from constraints (4).

[DUAL]

$$\text{Maximize } \sum_{i=1}^I \sum_{k=1}^K e_{ik} (-R_k^i) + \sum_{m=1}^M \sum_{j=1}^J D_{mj} b_{mj}, \quad (7)$$

Subject to

$$-e_{ik} + \sum_{m=1}^M r_{im} a_{mkp}^i \leq -g_k^i, \quad \forall i, k, p \quad (8)$$

$$-r_{im} + b_{mj} \leq c_{ij}^m, \quad \forall i, j, m, \quad (9)$$

$$e_{ik} \geq 0, \quad \forall i, k, \quad (10)$$

$$r_{im} \geq 0, \quad \forall i, m, \quad (11)$$

$$b_{mj} \text{ unrestricted, } \forall m, j. \quad (12)$$

The simplex multipliers of [DUAL] are used to search a new legitimate cutting pattern. The multi subproblems are decomposed by the column-generation technique to find out the effective cutting patterns for supplier i. For each possible i, the problem [NPP_i] can be modeled as:

$$\text{Maximize } V_{ik^*} = \sum_{m=1}^M r_{im} a_{mk^*p^*}^i - (g_{k^*}^i - e_{ik^*}) \quad (13)$$

Subject to

$$\sum_{m=1}^M l_m a_{imk^*p^*} \leq \sum_{k=1}^K L_k q_k, \quad (14)$$

$$\sum_{k=1}^K q_k = 1, \quad (15)$$

$$a_{imk^*p^*} \geq 0 \text{ and integer, } m = 1, \dots, M,$$

$$q_k \geq 0 \text{ and binary, } k = 1, \dots, K$$

where e_{ik^*} = the simplex multipliers of supplier i and used stock k^* for cutting a new pattern from constraints (2),

$a_{mk^*p^*}^i$ = the number of strips of item m cut in the new pattern p^* for stock length k^* by supplier i, fulfilling

$$\sum_{m=1}^M l_m a_{mk^*p^*}^i \leq L_{k^*}, \text{ and}$$

q_k = the used stock length for stock k; $q_k \in \{0, 1\}$, if $q_k = 1$ then the stock length k is used. Otherwise, the stock length k is not used.

Every supplier is solved to obtain the effective patterns. If $V_{ik^*} = \max \{V_{ik} | k=1, \dots, K\} > 0$ then the new column

$a_{mk^*p^*}^i = (a_{1k^*p^*}^i, \dots, a_{Mk^*p^*}^i)^T$ is introduced into the model as a new basis column. If

$V_{ik^*} \leq 0$ for $i = 1, \dots, I$, the current columns in the model is proven to be globally optimal for the relaxed problem. For a profound description of the column-generation procedure applied to the one-dimensional cutting stock problems, see Lasdon^[13] (section 4.1).

Benders Decomposition: Consider the procedure to apply the Benders decomposition to the [1D-CSP&TP] so that a large-scale problem can be separated to multi subproblems. The relaxed integer [1D-CSP&TP] is decomposed into I subproblems [SP_i] and a master problem [MP]. The [SP_i], $i=1, 2, \dots, I$ is corresponded to the 1D-CSP, whereas [MP] is corresponded to the generalized TP. The models of them are shown as follows: For [SP_i]:

$$\text{Minimize } \sum_{k=1}^K \sum_{p=1}^{P_k} g_k^i x_{kp}^i \quad (16)$$

Subject to

$$- \sum_{p=1}^{P_k} x_{kp}^i \geq -R_k^i, \forall k \quad (17)$$

$$\sum_{k=1}^K \sum_{p=1}^{P_k} a_{mkp}^i x_{kp}^i \geq \sum_{j=1}^J y_{ij}^m, \forall m, \quad (18)$$

$$x_{kp}^i \geq 0, \forall k, p, \quad (19)$$

Constraints (17) correspond to the limitation of capacities of each standard lengths at supplier i. Constraints (18) correspond to the difference between the amount of the products and the amount of shipped products should greater than or equal to zero. In other words, the quantity of available products should be greater than the quantity of the shipped products from the supplier i to the customer j for all required lengths.

The simplex multipliers of constraints (17) – (18) are practicable in the Benders master problem ([MP]). The mathematical model of [MP] is defined as below.

$$\text{Minimize } \sum_{m=1}^M \sum_{i=1}^I \sum_{j=1}^J c_{ij}^m y_{ij}^m + Z \quad (20)$$

Subject to

$$\sum_{i=1}^I y_{ij}^m = D_{mj}, \forall m, j \quad (21)$$

$$Z \geq - \sum_{i=1}^I \sum_{k=1}^K e_{ik} R_k^i + \sum_{i=1}^I \sum_{m=1}^M r_{im} \sum_{j=1}^J y_{ij}^m \quad (22)$$

$$y_{ij}^m \geq 0, \forall m, i, j. \quad (23)$$

The [SP_i], as known in 1D-CSP with different types of standard lengths and the limitation of capacities, is solved by the direct column-generation technique for exactly solving the continuous relaxation of the problem. The simplex multipliers of Constraints (17) and (18) are used to construct the optimality cut in (22). Basically, [SP_i] finds the solution of x_{kp}^i , given

the fixed proportion of allocations on y_{ij}^m . Performing as a coordinator, [MP] then develops its allocations through the subproblem dual price, added more Benders cuts. [MP] is known as a generalized multi-commodity TP. The lower bound value (LB) is obtained from the objective value of [MP]. The upper bound value (UB) is

obtained from $\sum_{m=1}^M \sum_{i=1}^I \sum_{j=1}^J c_{ij}^m y_{ij}^m$ of [MP] plus

$$\sum_{k=1}^K \sum_{p=1}^{P_k} g_k^i x_{kp}^i \text{ of [SP}_i\text{].}$$

To work with hybrid Benders decomposition and column-generation techniques, it is involved with doing loop between the master problem [MP] and subproblems [SP_i]. Obviously, two loops; column-generation loop and Benders loop, are overlapped. If any loop gets struck such as; sluggish convergence or occurred degeneracy situation, the algorithm will take too long computation time. Hence, some rules are stipulated; 1) if the different of UB and LB values between adjacent iterations has less than and equal to 0.5 occurred more than 10 consecutive times, the program will stop. 2) the number of newly generated patterns by the column-generation technique is restricted to 6 times the number of required lengths.

The first rule is raised for protection of the sluggish convergence because the [MP] of Benders decomposition may add the slowly convergent Benders cuts in (22). This rule helps to save the loss computation time; however, it makes the objective value be greater than the optimal solution. The second one is raised for preventing the degeneracy event. If this sub-procedure gets struck, Benders approach will also become stubborn. This stopping rule is created under the idea that the right hand sides obtained from Benders approach each iteration are modified. So some efficient patterns, in which the column-generation technique provided, may not be the efficient patterns in the next iteration. To reduce the computation time, this rule does work and not create much effect occurring in the

objective value in case of using 6*M limitation. Furthermore, the first rule as mentioned above is one of the stopping rules too. In summary, there are two stopping rules in this algorithm;

1. If UB=LB, and
2. If the number of iterations that $UB-LB \leq 0.5$ has been occurred more than 10 consecutive times.

The Proposed Heuristic Method for Solving IP: The key idea to solve 1D-CSP&TP has been outlined using the concept of solving [1D-CSP&TP] model as a continuous problem using both column generation techniques and Benders decomposition to control the number of generated variables and constraints. However, the solution obtained by the approaches may violate the required integer restrictions. This heuristic method is proposed for solving the resulting IP at the end of process. The principle of round down method based on Holthaus^[2] is developed. After solving any corresponding linear programming of [1D-CSP&TP] model, all decision variables are obtained. Consider each x_{kp}^i , there are three categories of the values of

x_{kp}^i such as in the interval; a) greater than or equal to 1, b) between 0 and 1, and c) equal to zero. Hence, three groups of constraints are added under three cases as below:

Case1: If $x_{kp}^i \geq 1$ then the constraint,

$$x_{kp}^i \geq \lfloor x_{kp}^i \rfloor, \text{ is added.}$$

Case2: If $0 < x_{kp}^i < 1$ then the constraint,

$$x_{kp}^i \leq 1, \text{ is added and finally,}$$

Case3: If $x_{kp}^i = 0$ then the constraint,

$$\sum_{k=1}^K \sum_{p=1}^{P_k} x_{kp}^i \leq 1, \text{ is added.}$$

For solving IP with the proposed heuristic method, the constraints (28) - (30) are added with the expectation that the non-relaxed resulting model can be solved within a reasonable time as claimed in Holthaus^[2].

Let $\hat{x}_{kp}^i, \check{x}_{kp}^i$ and \ddot{x}_{kp}^i be x_{kp}^i in Case1, Case2, and Case3, respectively. The mathematical model for solving this related IP can be written as follows:

$$\text{Minimize } \sum_{i=1}^I \sum_{k=1}^K \sum_{p=1}^{P_k} g_k^i x_{kp}^i + \sum_{m=1}^M \sum_{i=1}^I \sum_{j=1}^J c_{ij}^m y_{ij}^m, \quad (24)$$

Subject to

$$-\sum_{p=1}^{P_k} x_{kp}^i \geq -R_k^i, \quad \forall i, k, \quad (25)$$

$$\sum_{k=1}^K \sum_{p=1}^{P_k} a_{mkp}^i x_{kp}^i - \sum_{j=1}^J y_{ij}^m \geq 0, \quad \forall i, m, \quad (26)$$

$$\sum_{i=1}^I y_{ij}^m = \sum_{j=1}^J D_{mj}, \quad \forall m, \quad (27)$$

$$\hat{x}_{kp}^i \geq \lfloor \check{x}_{kp}^i \rfloor \quad (28)$$

$$\check{x}_{kp}^i \leq 1 \quad (29)$$

$$\sum_{k=1}^K \sum_{p=1}^{P_k} \ddot{x}_{kp}^i \leq 1 \quad (30)$$

$$x_{kp}^i \geq 0 \text{ and Integer, } \forall i, k, p, \quad (31)$$

$$y_{ij}^m \geq 0, \quad \forall i, j, m. \quad (32)$$

METHODOLOGY

Based on the approaches described in the previous section, there are 4 related algorithms: 1) Problem Generator, 2) A0-algorithm, 3) A1-algorithm, and 4) A2-algorithm, developed to test efficiency and effectiveness. To code and run all algorithms, the required materials are as follows:

1. Borland C++ version 5.02.
2. LINGO version 6.0.
3. A personal computer with 1.6 GHz, Pentium IV processor with 512 MB RAM under Windows XP Professional.

In order to solve the [1D-CSP&TP], the concept of this study could be divided into two phases. One, the relaxed [1D-CSP&TP] is solved by the column-generation technique or the hybrid Benders decomposition and the column-generation technique. And two, the integer [1D-CSP&TP] is solved by the direct method and the proposed heuristic method.

In the first phase, the idea is to reduce the number of variables by using the two techniques as mentioned previously. Instead of solving the model with all possible patterns, the column-generation approach and the hybrid Benders decomposition and the column-generation provide only essential patterns.

In the second phase, the idea is: if the optimal solution and the computation time cannot go together, we have to choose which one is the most important. This study aims at the computation time, so the

heuristic method is proposed to solve the final resulting IP. By the way, the direct method also uses to solve this IP for computational time and solution quality comparisons.

First of all, the necessary feasible initial-patterns are generated for forming the initial IP. Concerning with various lengths of stock and various costs of stock in each supplier, the suitable stock length is selected under the condition that it can be cut into all retail sizes and that its length has the lowest price compared among all suppliers. This procedure is called “Initial-pattern Generator” as described in the steps below.

Initial-pattern Generator

- Step1: Find the maximum length of the retail item.
- Step2: Find the stock length, which is longer than the one in Step1. Ensure that its cost is the cheapest one from all suppliers
- Step3: Find the maximum number of pieces that the selected stock length from Step2 can cut into each retail item.
- Step4: Generate the initial patterns by using the numbers from Step3.

The relaxed IP is formulated by using obtained initial-patterns. All related decision variables are nonnegative. Now three approaches are designed for solving this problem

A0 Approach: The first approach is called “A0”. This approach applied the column-generation technique (CG) to generate appropriated patterns. These patterns are necessary to help reducing the objective value as much as possible. The x_{kp}^i variables are added more and more until the reduced cost couldn’t decrease in the relaxed discrete restrictions [1D-CSP&TP]. After that, all patterns are gathered for solving the final resulting integer program (IP) by added nonnegative integer variable constraints. This method is called the direct method. In this study, all approaches are limited the computational time at 6 hours. The steps of summarized A0 algorithm are as follows:

A0 Algorithm

- Step0: Formulate the relaxed [1D-CSP&TP].
- Step1: Apply CG at the model from Step0.
- Step2: Solve the resulting IP by the direct method.
- Step3: Test for the stopping rule. If the computational time > 6 hours, stop. The otherwise the optimal solutions are found.

A1 Approach: The second one is called “A1”. After all patterns are generated for solving the final IP. The

proposed heuristic method is applied instead of using the direct method. Therefore, the steps of A1 algorithm can be summarized as follows:

A1 Algorithm

- Step0: The same as A0.
- Step1: The same as A0.
- Step2: Solve IP by the proposed heuristic method.
- Step3: The same as A0

A2 Approach: The last one is called “A2”. This approach applied the Hybrid Benders decomposition and the column-generation technique (HB&CG) to generate patterns and cutting plane simultaneously. Therefore, the new x_{kp}^i variables and the new constraints are added until meeting an optimal solution. After that, all patterns are gathered for solving the IP by the proposed heuristic method. In summary, the steps of A2 algorithm are as follows:

A2 Algorithm

- Step0: Formulate the relaxed[1D-SP&TP].
- Step1: Apply HB&CG to the model from Step0.
- Step2: Solve the final IP by the proposed heuristic method.
- Step3: Test for the stopping rule.

To compare the efficiency and effectiveness of each approach, the IP computation time and the relative-different percentage are used. From the three-approach observations, the IP solving computation times are collected and analyzed. Nevertheless, only two approaches with the proposed heuristic method: A1&A2 are considered to compute the relative-different percentage. Let R1&R2 be the relative-different percentage of the approach A1&A2 respectively. The formulas for computing the relative-different percentage of each approach are as follows:

$$R1 = \left(\frac{IP1 - LB}{LB} \right) * 100,$$

where *IP1* represents the objective value, solved by A1 and *LB* is the lower bound represented by the optimal objective value from Step1 of all approaches and

$$R2 = \left(\frac{IP2 - LB}{LB} \right) * 100,$$

where *IP2* represents the objective value, solved by A2.

For generating different instances of [1D-CSP&TP], the parameters of an instance consist of the

number of suppliers (I), the number of customers (J), the number of stock lengths (K), and the number of retail items (M). For simply calling, the instance is referred as $P[I, J, K, M]$. The needed input data consist of the lengths of stock (L_k), the lengths of retail item (l_m), the material costs (g_k^i), the shipping costs (c_{ij}^m), the capacity (R_k^i), and the demands (D_{mj}). They have been randomly generated in many ways. In this study, the values of $I \in \{2, 3\}$, $J \in \{4\}$, $K \in \{2, 3, 4, 5\}$ and $M \in \{5, 10, 15, \dots, 40\}$ have been investigated. Totally, there are 64 instances in this study. In order to generate all instances, the problem generator is coded with the feasible linear problem guarantee. The total length of all suppliers is greater than 1.25 times of the total needed length of all demands. The assumptions of the problem generator are that a longer stock certainly leads to a higher price and adequate quantities of stocks are assigned for distributions. The algorithm can be shown as follows:

Problem Generator Algorithm

- Step0: Input a seed number, data ranges and the parameters I, J, K , and M .
- Step1: Randomize K standard stock lengths (L_k). Then sort by descending order.
- Step2: Randomize K costs of the standard stock lengths (g_k^i), then sort by descending order, and match them with L_k from Step1.
- Step3: If $i \leq I$ then repeat Step2.
- Step4: Randomize the retail items (l_m) from 1 to the longest L_k .
- Step5: Randomize D_{mj} , c_{ij}^m , and R_k^i .
- Step6: Check the feasibility of the problem.
 - 6.1: Classify l_m into K groups where the longest l_m in each group is less than or equal to L_k .
 - 6.2) If 1.1 times the total needed length of group k is less than the total stock length of group k , the capacity will be added up until the total stock length is greater than or equal to 1.1 times the needed length.
 - 6.3) Consider the overall, if 1.25 times the total needed length is longer than the longest L_k of the first supplier, it will be added up until the total stock length is greater than or equal to this desired value.

RESULTS

Of the three approaches just described, 64 instances have been tested on [1D-CSP&TP]. All

instances were put in the form of $P[I, J, K, M]$ where $I = 2, 3$, $J = 4$, $K = 2, 3, 4, 5$ and $M = 5, 10, 15, 20, 25, 30, 35, 40$. The raw data table is available upon requests. The computational performances of all instances can be discussed as follows.

A0: There were 17 instances, which took the computation time more than 21,600 seconds (> 6 hours). From now on, only 47 instances were considered. To compare with A1, we found that there were 34 instances, which took the computation times less than those of A1 with the maximum gap of 10 seconds, whereas the 13 instances left took the computation times greater than those of A1 with the maximum gap of 18,789 seconds. To compare with A2, we found that there were 43 instances, in which took the computation times less than those of A2 with the maximum gap of 7,016.76 seconds, whereas the 4 instances left took the computation times greater than those of A2 with the maximum gap of 18,689.28 seconds. In term of effectiveness, A0 always provides the best IP objective value as compared to others in all those 47 instances (within the controlled terminated time). However, the method became inconsistent especially when M is large.

A1: 64 instances could be solved within 219.855 seconds. The computation time of A1 were less than those of A2 for all instances with the maximum gap of 55,377.8 seconds.

A2: 63 instances could be solved within 10,726.3 seconds, but the left one took more than 21,600 seconds (or > 6 hours). The computation time of this case, $P[3-4-3-35]$, went up to 55,425.2 seconds. The computation time of A2 did not directly change with the computation time of A0 and A1. For example, $P[2-4-3-20]$: the computation times of A0, A1, and A2 were 11.878, 12.849, and 291.18 seconds, respectively, whereas $P[2-4-3-25]$: the computation times of A0, A1, and A2 were 18,802.7, 13.699, and 113.422 seconds.

It is obvious that A1 could solve the [1D-CSP&TP] within a more reasonable time as compared to the other two. This approach was fast enough for the processors who would like to design some proficient production and transportation plans in real time.

Also, from the computation results of A1, the maximum relative-different percentage of R1 was 0.140256%, whereas the maximum relative-different percentage of R2 was 11.41063%. There were 37 instances that the percentages of R1 were less than

those of R2 with the maximum gap of 11.41058%, 18 instances that the percentages of R1&R2 were both equal and 9 instances that the percentages of R1 were greater than those of R2 with the maximum gap of 0.05163%. In summary, the approach A1 could solve [1D-CSP&TP] test problems within the relative-different gap much lower than those obtained from the approach A2.. For planning the production and transportation strategies, the processors certainly want the smallest relative-different percentage. Therefore, with the reasonable computation time and effective obtained solution statistically in this study, A1 is a more appropriate approach to solve [1D-CSP&TP].

CONCLUSION AND EXTENSION

In this study, three approaches: A0, A1, and A2, based on the concepts of column-generation techniques, Benders decomposition and heuristics for integer linear programming are proposed to solve the integrated model of one-dimensional cutting stock and linear multi-commodity transportation problem.. From the experiments, A1 is the most efficient and effective approach for solving this large scale problem. However, all three approaches can be the alternative approaches to solve [1DCSP&TP] since there exist some heuristics employed in each approach. Each of them certainly has its own advantages (gain) and disadvantages (loss) that can be discussed as follows:

For A0: This approach frequently could not solve the problem in a reasonable time. As to the column-generation technique, this technique obtained only the optimal LP objective value. It is not guaranteed that the IP objective value will also be optimal. Although the IP objective value of A0 is not optimal, but with the direct method for solving the IP, A0 obtained the best IP objective value. And from the experiments, 34 out of 64 instances took the computation times less than those of A1. Therefore, A0 always has a chance to be the best approach but not on average.

For A1: With the proposed heuristic method, this approach could solve the large-scale problems within a reasonable time. But the IP objective value of A1 is worse than the one obtained from A0. However, the experiments show that all instances could be solved within 219.855 seconds (or less than 4 minutes) and the range of the relative-difference percentage of A1 is quite small (less than 0.15%).

For A2: There are one rule for the degeneracy protection in the column-generation technique and two stopping rules in this algorithm. Hence, the obtained LP objective value by A2 will be the best if the operation

runs step by step without jumping with the rule for the degeneracy protection and stop with the first stopping rule (UB=LB). Otherwise, the obtained LP objective value by A2 will be not optimal. Furthermore, the column-generation technique was applied during Benders iteration. For example, suppose A2 obtained the LP objective value after 10 Benders iterations. It means that the column-generation technique was applied 10 times too. Therefore, the computations of A2 takes more time than those of A1. However, the IP objective values of A2 in some instances were less than those of A1 implying that this approach is not all worse.

For the further study, since the [1D-CSP&TP] considered is NP-hard, there is a strong need to continuously develop heuristic approaches to solve the problem efficiently and effectively. Nonato and Scutellà^[14] proposed a linear programming algorithm based on hyper flows for solving the one-dimensional cutting stock problem but not limited the number of stocks. Therefore, their algorithm can be applied as another new alternative approach for solving the [1D-CSP&TP] in the future study.

REFERENCES

1. Dantzig, G. B. and P. Wolfe, 1961. The decomposition algorithm for linear programming., *Econometrica.*, 29: 767-778
2. Holthaus, O., 2002. Decomposition Approaches for Solving the Integer One-Dimensional Cutting Stock Problem with Different Types of Standard Lengths., *European Journal of Operational Research*, 141: 295-312.
3. Benders, J. F., 1962. Partitioning procedures for solving mixed variables programming problems., *Numerische Mathematik*, 4: 238-252.
4. Heragu, S.S. and J. Chen, 1998. Optimal solution of cellular manufacturing system design: Benders decomposition approach., *European Journal of Operational Research*, 107: 175-192.
5. Sridhar, V. and S. J. Park, 2000. Benders-and-cut algorithm for fixed-charge capacitated network design problem., *European Journal of Operational Research* 125: 622-632.
6. Wu, P., C. J. Hartman, C and R. G. Wilson, 2003. A demand-shifting feasibility algorithm for Benders decomposition., *European Journal of Operational Research*, 148: 570-583.
7. Gilmore, P. C. and R. E. Gomory, 1961. A linear programming approach to the cutting stock problem., *Operations Research*, 9: 849-859.

8. Gilmore, P. C. and R. E. Gomory, 1963. A linear programming approach to the cutting stock problem - part II., *Operations Research*, 11: 863-888.
9. Stadtler, H., 1990. A one-dimensional cutting stock problem in the aluminum industry and its solution., *European Journal of Operational Research*, 44: 209-223.
10. Gradišar, M., J. Jesenko and G. Resinovič, 1997. Optimization of roll clothing in clothing industry., *Computer and Operations Research*, 24: 945-953.
11. Adlakha, V., and K. Kowalski, K., 1998. A quick sufficient solution to the more-for-less paradox in the transportation problems., *OMEGA*, 26: 541-547
12. Sharma, R. R. K. and K. D. Sharma, 2000. A new dual based procedure for the transportation problem., *European Journal of Operational Research*, 122: 611-624.
13. Lasdon, S. L., 1970. *Optimization theory for large systems.*, New York:Macmillan publishing.
14. Nonato M. and M. G. Scutellà, 1998. The one-dimensional cutting stock problem: a linear programming algorithm based on hyper flows., *Proceedings of "Algorithms and Experiments"*, Italy,,: 44-50.