Exact and Heuristic Approaches for the Production Planning Problem of a Paper Mill Company

Kyungdoc Kim\textsuperscript{a}, Byung-In Kim\textsuperscript{a*}, Hyunbo Cho\textsuperscript{a}

\textsuperscript{a}Industrial and Management Engineering, Pohang University of Science & Technology (POSTECH)
San 31 Hyoja, Pohang 790-784, Republic of Korea

Abstract

In this paper, we consider a multi-staged two-dimensional cutting stock problem (CSP) in a paper industry. Paper production relies on the sequential processes of various large machines (pulp preparation, paper formation, winding, and sheet cutting), which has various machine related constraints. In addition, there are the operational constraints from the real-world situation. The problem is modelled as a mixed linear integer programming and solved by a heuristic algorithm. Computational experiments are conducted using the test data sets, which are generated based on the real-world data from a large paper mill company. The computational results show the effectiveness of our proposed algorithm.

Keywords: multi-staged cutting stock problem; cutting stock problem; paper mill industry

1. Introduction

A paper production process which includes the production and cutting of paper relies on large heavy machines. The paper is produced by a series of large machines such as paper machine, winder, and sheet cutter. A paper machine produces a large jumbo roll of paper from raw material pulp. Then, the jumbo roll is slit into smaller rolls of paper at the winder. The smaller rolls are cut into a bunch of sheet of paper at the sheet cutter. Figure 1 depicts a general processes of producing paper products.

![Fig. 1. General processes of producing paper products](image)

A cutting stock problem (CSP) is to find efficient cutting patterns by which small items are cut from large materials. It usually comes up when a material is produced in bulk [1], such as paper, steel plate, leather, and glass. All these products are produced in bulk, and cut into ordered items. Throughout the cutting process, trim loss or wasted area is inevitable due to the heterogeneous demanded sizes. Thus, minimizing the waste cost is one of the main objectives of CSP. In this paper, we focus on solving a...
real-world CSP of a paper mill company. According to the topology provided by Wäscher, Haussner, and Schumann [2], this problem can be classified as a variant of a single stock size cutting stock problem.

A production plan includes the sequence of cutting patterns at the winder and sheet cutter. Generating a production plan for the paper mill industry is hard because there are various specifications (width, length, demanded weight and other requirements) of orders that placed by different customers and various constraints of machines.

CSP has been considered in a number of published papers. As an exact approach, Valério de Carvalho [3] compared various linear programming formulations for CSP and Harjunkoski, Westerlund, Pörn [4] modelled CSP as a mixed integer non-linear programming (MINLP) and presented various transformation techniques to handle non-convexity and non-linearity. Delayed column generation has been used popularly in many papers [5, 6, 7, 8, 9, 10]. A new pattern (column) is generated by solving an auxiliary knapsack problem. This approach is very useful to solve a relaxed linear problem but it usually ends up with fractional solutions. Thus, post rounding heuristics for making integer solution is usually followed in the papers [11, 12, 13]. On the other hand, some heuristic approaches are devised to solve CSP. Ferreira, Neves, and Fonseca e Castro [14] solved a roll cutting problem in the steel industry by a heuristic algorithm that is a variant of the repeated exhaustion reduction algorithm of Hinxman [1]. Moreover, Simulated annealing [15, 16], genetic algorithm [17, 18], and random search [19] have been used to solve various CSP.

In this paper, we tackled a two-staged two-dimensional CSP in the paper mill industry. Various machinery constraints and operational constraints are investigated in the company. Due to the large size of customer orders and complex constraints the approaches in the literature could not be applied for our problem directly. Thus, we develop a multiple-choice knapsack-based heuristic algorithm (MCKH) for the problem. The algorithm is tested on the datasets that are generated based on the real-world data obtained from the paper mill company. The solution results are compared with the exact solutions obtained by the mathematical model.

The remainder of this paper is organized as follows: Section 2 describes the problem tackled in this paper and relevant constraints. Section 3 introduces the algorithm details of MCKH. The experiment setup and results are shown in Section 4. Finally, Section 5 discusses the comparison results and provides future research directions.

2. Problem objectives and constraints

2.1. Objectives

Our problem is a multi-objective optimization problem. In the literature, the weighted sum of objectives with proper coefficients is presented and optimized with different coefficient settings. In this paper, four different objectives are developed through the discussion with the engineers at the company. Each objective is expressed in physical quantities of a solution. The first is the sum of absolute deviation of the production weight from the total order weight (SAD). It measures the order fulfillment. The second is the sum of waste weight (WW). It measures the amount of waste in the solution. The third is the total number of patterns (NP) used. It measures the number of setups at the winder. The last is the number of auxiliary rolls (NR) used, which is the unit of roll that can be cut by the sheet cutter. It measures the workload at the sheet cutter. A weighted sum of the objectives is used with appropriate coefficients in the following mathematical model.

2.2. Constraints

The constraints can be classified into two categories. One is machinery constraints that directly come from the specifications of the machines in the company. The other is the operational constraints that come from managerial decisions. The machinery constraints are as follows:

- The maximum width of a jumbo roll at the winder is 4880 mm
- The maximum number of auxiliary rolls to be cut at the winder is 7
- The minimum width of an auxiliary roll at a sheet cutter is 800mm
- The maximum width of an auxiliary roll at a sheet cutter is 2870mm
- The maximum number of sheets in a row to be cut by a sheet cutter is 4

The operational constraints are as follows:
- The length of a jumbo roll should be a multiple of the standard length
- The orders that placed from different group (domestic or international) cannot be produced in a same pattern
- The wrapping style of an order must be considered (single or double wrapping)
- Sheet orders that have same length and width can be grouped and produced together
- An order cannot be rotated during the production
- An order has a lower and upper bound weight to produce

Above constraints are considered and satisfied during the solution approach. Especially, we use a concept of sub-patterns for an order. A sub-pattern is a virtual unit that combines one or more auxiliary rolls of the same order.

2.3. Mathematical model

A mixed integer linear programming (MILP) model is developed to formally present the problem. We explicitly formulate the production planning problem in a model without pre-defined patterns. All the objectives mentioned above are considered in the MILP model, and the constraints on the winder are considered explicitly in the model. The constraints on sheet cutter is considered prior to build the MILP model. The full version of the model can be found in Kim et al [20].

3. Multiple-choice knapsack-based heuristic algorithm (MCKH)

As a solution approach, we develop MCKH. It can be seen as a type of repeated exhaustion reduction algorithm of Hinxman (1980). Following seven steps represent the flow of MCKH. After generating the sub-patterns for whole orders, repeating Step 3 through Step 5 generates a single solution. The solution generation steps (Steps 3 to 5) are repeated and many solutions are generated. Whenever the new solution is found, it is compared with the current best solution and updated if it is better one. MCKH stops when the two of one stopping criteria is met. The two criteria are maximum running time and maximum number of iterations without updating the best solution. More detailed explanations for each step can be found in [20].

Step 1: Merge sales orders of the same specification
Step 2: Generate the sub-patterns for each merged order
Step 3: Generate candidate patterns by solving multiple-choice knapsack problem (MCKP). If there is no possible pattern exist, go Step 6
Step 4: Evaluate candidate patterns and select a pattern as a solution
Step 5: Update remaining orders and sub-patterns, and go to Step 3.
Step 6: Evaluate the generated solution, and update the best known solution if possible.
Step 7: If the stopping criteria are met, then stop. Otherwise, Go to Step 1.

At Step 3, each sub-pattern will be selected as a reference sub-pattern for a new pattern in turn. Then, the best combination of sub-patterns to fill the pattern from the remaining sub-patterns is chosen by a knapsack problem formulation. Since there may exist multiple sub-patterns for an order but at most one sub-pattern for the order can be selected, the knapsack problem become a multiple-choice knapsack problem (MCKP). The MCKP can be formulated as an MIP model as follows:

\[
\begin{align*}
\max \sum_{i=1}^{N} \sum_{j=1}^{M_i} p_{ij} x_{ij} \\
\text{s.t.} \quad \sum_{i=1}^{N} \sum_{j=1}^{M_i} w_{ij} x_{ij} &\leq W \\
\sum_{j=1}^{M_i} x_{ij} &\leq 1 \quad \forall i \\
x_{ij} &\in \{0,1\} \quad \forall i,j
\end{align*}
\]

Where \(x_{ij}\) is 1 if sub-pattern \(j\) for order \(i\) is chosen; 0, otherwise. \(p_{ij}\) is the area (profit in knapsack terminology) of the sub-pattern \(j\) for order \(i\), \(w_{ij}\) is the width (weight in knapsack terminology) of the sub-pattern \(j\) for order \(i\), and \(W\) is the remained pattern width, which is the maximum width of a jumbo
roll minus the width of the reference sub-pattern. MCKP can be solved by extending the matrix-based dynamic programming algorithm for the binary knapsack problem [21].

Selecting a pattern among the candidate patterns in Step 4 is difficult because the selection of a pattern at this stage cannot guarantee the global optimality. Such a problem is typical in any sequential decision algorithm. Therefore, we utilized a stochastic selection strategy. Among the candidate patterns, it is filtered with the total width of a pattern and the length of a pattern by pre-defined heuristic value. Within the filtered patterns, one of them is randomly chosen as a solution pattern at this stage. Note that if there is no patterns remaining after filtering, one of the patterns is randomly chosen from the generated candidate patterns.

4. Experiment results

In this section, the test results are shown for 10 problem instances. Our algorithm is implemented in C++ using Visual Studio 2010. The MILP model is solved by CPLEX v12.4.

4.1. Test data

The test problems were generated based on the real-world problems of the paper mill company. The summarized information of the test problems are shown in Table 1. The number of orders vary from 3 to 6. Total order weights range from 22 ton to 615 ton. The minimum and maximum width of order and standard deviation of orders for each problem are shown in Table 1.

Table 1. Summarized information of test problems

<table>
<thead>
<tr>
<th>Problem ID</th>
<th>Number of orders</th>
<th>Total order weights (Ton)</th>
<th>Stddev. width of orders</th>
<th>Minimum width of order (mm)</th>
<th>Maximum width of order (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>22.798</td>
<td>95.7</td>
<td>665</td>
<td>850</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>33.173</td>
<td>266.5</td>
<td>535</td>
<td>1092</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>246.819</td>
<td>115.9</td>
<td>755</td>
<td>1003</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>615.223</td>
<td>42.2</td>
<td>610</td>
<td>708</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>26.633</td>
<td>170.8</td>
<td>560</td>
<td>980</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>38.098</td>
<td>223.8</td>
<td>640</td>
<td>1194</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>44.201</td>
<td>80.9</td>
<td>510</td>
<td>715</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>67.162</td>
<td>93.7</td>
<td>495</td>
<td>700</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>160.283</td>
<td>179.5</td>
<td>550</td>
<td>921</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>31.730</td>
<td>168.5</td>
<td>640</td>
<td>1092</td>
</tr>
</tbody>
</table>

4.2. Experiment setup

The following parameters were used in the test. All the experiments were conducted on a Dell Precision T7500 (Intel® Xeon® CPU with 2.67 GHz, 20.0GB RAM running windows server 2008 R2 Enterprise.

- Maximum computation time: 300 (seconds)
- Maximum iteration number without updating the best solution: 3000
- Coefficient for SAD: 1000
- Coefficient for WW: 1000
- Coefficient for NP: 10
- Coefficient for NR: 1

4.3. Results

The comparison results for the test data are shown in Table 2. The numbers in bold face represent the optimal solution values. Column 1 indicates the problem ID shown in Table 1. Column 2 shows the lower bound obtained by CPLEX solver. The MILP model found the optimal solutions within the maximum computation time except Problem 4. Note that CPLEX terminated for Problem 3 as reaching the maximum computational time.
MCKH found the optimal solutions for nine out of ten problems. For Problem 9, its objective value is slightly higher than the optimal one. MCKH for all the problems stopped before reaching the maximum computational time. Its computational time ranges from 2 to 11 seconds.

Table 2. Experiment results

<table>
<thead>
<tr>
<th>Problem ID</th>
<th>The solution of MILP</th>
<th>The solution of MCKH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lower bound</td>
<td>Objective value</td>
</tr>
<tr>
<td>1</td>
<td>5481.1</td>
<td>5481.1</td>
</tr>
<tr>
<td>2</td>
<td>4158.9</td>
<td>4158.9</td>
</tr>
<tr>
<td>3</td>
<td>10903.4</td>
<td>10903.4</td>
</tr>
<tr>
<td>4</td>
<td>8582.8</td>
<td>20911.7</td>
</tr>
<tr>
<td>5</td>
<td>6533.3</td>
<td>6533.3</td>
</tr>
<tr>
<td>6</td>
<td>5273.0</td>
<td>5273.0</td>
</tr>
<tr>
<td>7</td>
<td>4355.2</td>
<td>4355.2</td>
</tr>
<tr>
<td>8</td>
<td>5556.4</td>
<td>5556.4</td>
</tr>
<tr>
<td>9</td>
<td>12200.8</td>
<td>12200.8</td>
</tr>
<tr>
<td>10</td>
<td>6300.1</td>
<td>6300.1</td>
</tr>
</tbody>
</table>

5. Conclusion

In this paper, a two-staged two-dimensional CSP in the paper mill industry was tackled. Four different objectives and various constraints from the real-world situation were presented. An MILP model is formulated to formally present the problem. We propose MCKH to solve the problem, and steps for the algorithm is shown in the paper. The test problems were generated from real-world data obtained from a paper mill company. The solutions were obtained by solving problems using MCKH which implemented in C++. The experiment results shows that MCKH generates near optimal solutions within a short time.

This research needs to be improved in various directions, such as adding post optimization algorithms for reducing the number of patterns or the number of rolls using well-known meta-heuristics. Moreover, testing the algorithm on larger dataset is necessary.

References


