Article:

Application of Metaheuristics to Packing Formation Support Systems of Pre-Cut Lumber Factory

Takashi Tanizaki† and Ryohei Yamashita

Kindai University
1 Takaya-Umenobe, Higashi-Hiroshima, Hiroshima 739-2116, Japan
†Corresponding author, E-mail: tanizaki@hiro.kindai.ac.jp

[Received October 27, 2021; accepted January 26, 2022]

In recent years, wood has been considered as a building material from the perspective of SDGs. Pre-cut lumber is often used for wooden houses, including roofs, floors, and pillars. It is preprocessed into a certain shape for the joint parts and has joint brackets if necessary. The use of them has the advantages of shortening the construction period and reducing the construction cost. The number of pre-cut lumber production factories in Japan decreased from 757 in 2001 to 659 in 2011 and then increased to 730 in 2016. Compared with 2011, the number of factories with sales of <500 million yen in 2016 decreased by approximately 30%, while the number of factories with sales of ≥500 million yen increased by 80%, indicating a trend toward a larger scale [1]. Therefore, improving the productivity of these factories is imperative. We have been researching the conversion of factories into smart factories to improve the productivity of pre-cut lumber manufacturing company A. In this company, employees take 1.2 h to prepare packing plans every day, and the company faces the challenge of improving the efficiency of planning operations. Herein, we propose an algorithm using an iterative local search (ILS) with Or-opt (ILS + Or-opt) for packing formation support systems to improve the efficiency of planning operations. This algorithm has the following two features. First, it uses Or-opt to create a neighborhood for a local search. Second, uses exchange neighborhoods when repeating the ILS to achieve diversity in the search.

Keywords: metaheuristics, iterated local search, Or-opt, genetic algorithm, pre-cut lumber

1. Introduction

In recent years, wood has been considered as a building material from the perspective of SDGs. Pre-cut lumber is a wooden house material that is processed into a certain shape for the joint parts of building materials, such as roofs and floors, and has joint brackets attached as necessary. The use of pre-cut lumber has the advantages of shortening the construction period and reducing construction costs. In addition, pre-cut lumber is manufactured by machining based on a building blueprint; therefore, its dimensions are stable and free from deviation. Consequently, the demand for pre-cut lumber in wooden house construction is increasing annually. The number of pre-cut lumber production factories in Japan decreased from 757 in 2001 to 659 in 2011, then increased to 730 in 2016. Compared to 2011, the number of factories with sales of <500 million yen in 2016 decreased by approximately 30%, while the number of factories with sales of ≥500 million yen increased by 80%, indicating a trend toward a larger scale [1]. Therefore, productivity improvement has become an issue in pre-cut lumber production. Accordingly, we conducted this study as part of our efforts to build a foundation for transforming the factory of joint research company A – a pre-cut lumber manufacturer – into a smart factory.

A smart factory is a flexible system that can self-optimize performance across a broad network, self-adapt to and learn from new conditions in real or near-real time, and autonomously run entire production processes [2]. It incorporates various technologies, including but not limited to cyber-physical production systems, IoT, robotics/automation, Big Data analytics, and cloud computing to realize the vision of a data-driven, connected supply network [3]. An example of a smart-factory research project in Japan is value co-creative manufacturing and design via three-dimensional printing technology for producing customized rubber products. A real-virtual fusion manufacturing system was proposed and developed for production planning and scheduling in this project using a scaled model plant system [4]. In the future, we plan to connect pre-cut lumber manufacturing factories, house manufacturers, and house construction sites using a network. As a result, linking data from the design results of house design companies, production and shipping at the factory, and the construction process at the site, we aim to present the construction status to customers and realize advanced arrangements at the construction site. In addition to automating production, we will automate shipping operations using robots in the future. As part of the above, automation of production and shipment planning is essential. Following the development of the production planning algorithm, which was a joint research project, we are now developing a packing formation support system for
In this paper, we propose an algorithm that uses metaheuristics + packing formation simulation as the core of the packing support system. Metaheuristics are used to generate pre-cut lumber permutations for packing formation, and the packing formation simulation checks the constraints and calculates the objective function values. We implemented two metaheuristics: the genetic algorithm (GA) and ILS + Or-opt. In GA, new individuals (i.e., pre-cut lumber permutations) are generated using the widely used permutation crossover and mutation. ILS + Or-opt has the following two features. First, it uses Or-opt to create a neighborhood for a local search. Second, it uses exchange neighborhoods when repeating the iterative local search (ILS) to achieve diversity in the search. Because the algorithm configuration is metaheuristic + packing formation simulation, it is also possible to generate permutations for packing formation simulation using other metaheuristic algorithms.

The remainder of this paper is organized as follows. We describe previous research in Section 2. Next, we define the packing problem for pre-cut lumber in Section 3. We propose an algorithm using metaheuristics + packing formation simulation in Section 4. Finally, we present the results of numerical experiments comparing ILS + Or-opt with GA in Section 5, and conclusions are drawn in Section 6.

2. Previous Research

Multiple pieces of pre-cut lumber are placed on a flat surface to form a tier, which is then stacked and packed, yielding a cuboid packing problem. However, because spacers are inserted between the tiers, there is no interference between pre-cut lumber in the height direction. This problem is modeled as a rectangular packing problem where multiple rectangles with joint brackets are packed such that their total length is less than or equal to the length and width constraints.

The rectangular packing problem is the problem of arranging rectangles (products) of various sizes on a two-dimensional plane (base material) without overlapping them. It has long been studied in the fields of geometry and combinatorial optimization [5]. Typical rectangular packing problems include the two-dimensional knapsack problem (2KP), two-dimensional bin packing problem (2BP), and the strip packing problem (SP). These are NP-hard problems.

Practical solutions to the 2KP have been widely studied investigated since the study of Gilmore and Gomory [6]. In recent years, metaheuristic methods have often been proposed in addition to exact methods. These include GA [7–9], simulated annealing algorithm [10–13], tabu search algorithm [14, 15], and greedy randomized adaptive search procedures [16]. Typical exact methods for solving the 2KP include the tree search [17, 18], and branch and bound methods [19]. Another exact method is dynamic optimization [20].

For the 2BP, Constructive heuristic and metaheuristic methods have been proposed. Lodi et al. [21] examined 2KP and 2SP algorithms in detail. Chung et al. [22] proposed the following two-phase approach. The first step is to apply the first-fit decreasing height algorithm to the 1BP obtained by defining bins of equal heights. In the second step, the first-fit decreasing is applied to this solution. The same idea can be used in conjunction with the next-fit decreasing height [23] and the best-fit decreasing height [24], by using the corresponding algorithms in the second phase. Metaheuristic methods include a simulated annealing algorithm [25] and tabu search algorithms [26].

For the 2SP, constructive heuristic and metaheuristic methods have been proposed. Coffman et al. [27] extended two classical approximation algorithms from the 1BP to the 2SP. They analyzed three “level-oriented” algorithms, i.e., next-fit decreasing height, first-fit decreasing height, and best-fit decreasing height, for packing rectangles into a unit-width, infinite-height bin to minimize the total height of the packing. Baker et al. [28] proposed a different classical approach (bottom-left algorithm), which does not pack the items by levels. This algorithm sorts the items by non-increasing width and packs the current item in the lowest possible position, which is left-justified. Chazelle [29] proposed an efficient implementation of bottom-left. Additionally, several approximate solutions have been proposed [30, 31]. Metaheuristic methods include simulated annealing algorithm [25, 32, 33], and GA [34].

This study addresses the rectangular stuffing problem. Because our research focuses on the packing problem of a rectangle with joint brackets, it is necessary to arrange the rectangles to avoid interference between the joint brackets. The best-fit method and the bottom-left method, which are typical algorithms for the rectangular packing problem, cannot be applied.

Next, we discuss previous research on the irregular strip packing problem. Gomes and Oliveira proposed a hybrid algorithm wherein simulated annealing is used to guide the search over the solution space, while linear programming models are solved to generate neighborhoods during the search process [35]. Burke et al. proposed a bottom-left filling heuristic algorithm using hill climbing and tabu local search [36]. Egeblad et al. proposed a relaxed placement method to reduce the overlap, which can handle irregular polygons with holes. A new geometric approach was utilized, and a guided local search was used to escape local minima [37]. Imamichi et al. proposed an ILS to minimize the total amount of overlap and protrusion of a layout, where the layout may not be completely contained in the container during the application of the algorithm [38]. Leung et al. proposed an extended local search algorithm that adopts two neighborhoods, swapping two given polygons in placement and moving one polygon to a new position. The local search algorithm was used to minimize the overlap according to the aforementioned neighborhoods, and the unconstrained nonlinear programming model was adopted to further reduce the overlap.
during the search process. Moreover, the tabu search algorithm was used to avoid local minima [39]. Peralta et al. proposed a nonlinear programming model that considers the free rotations of polygons and separation lines that separate polygons to ensure non-overlapping. They solved this model using the nonlinear programming solver IPOPT [40]. Thus, metaheuristic algorithms such as simulated annealing and ILS, or formulated as nonlinear programming problems, and using solver software have been proposed to solve the irregular strip packing problem.

The irregular strip packing problem involves packing irregularly shaped polygons to minimize the length in the x-axis direction without overlapping polygons. Therefore, in the packing result, the left vertical axis of each polygon is not aligned. There are two main differences from the pre-cut lumber packing problem, which is the subject of our research. To ensure the stability of transportation, the left side of each column of the pre-cut lumber, excluding the joint bracket, must be aligned in the y-axis direction, as requested by company A. The maximum allowable length in the x-axis direction is given.

Therefore, we propose an algorithm for metaheuristics packing formation simulations. In many previous studies, GA, tabu search, and simulated annealing were used as metaheuristics. In this paper, we propose the ILS + Or-opt method from the viewpoint of inheriting good solutions and GA for metaheuristics.

3. Packing Problem for Pre-Cut Lumbers

3.1. Pre-Cut Lumber Definition

Figure 1 shows the pre-cut lumber definition provided by company A. The joint brackets are attached to the upper and lower surfaces and the left and right surfaces but not to the end surfaces. It is possible to pack multiple pieces of pre-cut lumber side by side in the length direction. This is called “connecting.” Fig. 2 shows a top view of one of the tiers that formed the packing. The tier is defined as each layer that constitutes the packing. The edge is defined as the leftmost and rightmost columns of the pre-cut lumber in each tier. The definitions of the edge/tier length and tier/edge width shown in Fig. 2 are as follows:

- Edge length 1: Total length of the pre-cut lumbers on the leftmost side.
- Edge length 2: Total length of the pre-cut lumbers on the leftmost side.
- Tier width 1: Tier width that does not include joint brackets protruding outward from the edge.
- Tier width 2: Tier width that includes joint brackets protruding outward from the edge.
- Tier length: Total length of the pre-cut lumbers, excluding both edges.

Figure 3 shows a packing image viewed from the direction of the end surface. Spacers are placed between tiers to avoid interference of the joint bracket, in the height direction, including the bottom. The packing height is calculated including the height of the spacer. In addition, packing lumbers are placed between the pre-cut lumber to stabilize the packing and make each tier have the same value of tier width 1. As a result of discussions with company A, our joint research company, packing lumber is not considered for this research.

The problem is to create multiple packages of pre-cut lumber to be built for one house in compliance with all the constraints. The constraints and objective functions of this problem are described below.

3.2. Constraints

There are two types of constraints: hard constraints that must not be violated and soft constraints that can be violated.
Tanzaki, T. and Yamashita, R.

Fig. 4. Constraints of tier.

3.2.1. Hard Constraints

(1) Constraints for the tier

- There are two or more pieces of pre-cut lumber per tier.
- The pre-cut lumber in the bottom tier cannot be connected.
- More than three pieces of pre-cut lumber cannot be connected at the edge.
- \((\text{Maximum tier length excluding both edges}) - \min(\text{edge length 1, edge length 2}) \leq \alpha \) (Fig. 4).
- \(|\text{edge length 1} - \text{edge length 2}| \leq \beta\).
- Tier length \( \leq \gamma \).
- Tier width 1 \( \leq \delta \).
- Tier width 2 \( \leq \varepsilon \).
- Length of the joint bracket protruding outside the tier \( \leq \zeta \) (Fig. 4).

(2) Constraints on the packing

- Packing height \( \leq \eta \).
- Packing weight \( \leq \theta \).

3.2.2. Soft Constraints

- No joint bracket protrudes at both edges of the tier.
- No joint bracket protrudes under the lower surface of the bottom tier.
- No connecting occurs at both edges of the tier above the second tier.
- Tier length 1 of the bottom tier \( > \) tier width 1 above the second tier.
- Pre-cut lumber length of the bottom tier \( \geq \kappa \).
- The weight of the tier is lower than the weight of the tier directly below.

3.3. Objective Function

The objective function minimizes the number of packings and tiers. Furthermore, the number of violations of the soft constraint in Section 3.2.2 is minimized. Therefore, the objective function is as follows:

\[
\text{Minimize } 10000 \ast (a + b) + 1000 \ast c + 100 \ast (d + e + f) + 10 \ast g + 5 \ast h + 2 \ast i, \ldots
\]  

\( a \): The number of packings.
\( b \): The number of tiers.
\( c \): The number of occurrences in which the pre-cut lumber length of the bottom tier is \( < \kappa \).
\( d \): The number of cable ties. A cable tie is a type of fastener used to fix the packing of the pre-cut lumber.
\( e \): Number of occurrences in which the weight of the tier is heavier than the weight of the tier directly below.
\( f \): The number of occurrences that tier width 1 of the bottom tier is equal or less tier width 1 above the second tier.
\( g \): Number of connecting at both edges of the tiers above the second tier.
\( h \): Number of joint brackets protruding under the lower surface of the bottom tier.
\( i \): Number of joint brackets protruding at both edges of the tier.

The weight coefficient of \((a + b)\) was set to 10000 so that it would be larger than the value obtained by multiplying the weights of the other variables. The weight coefficients of \(a\) and \(b\) were set to the same value, by the instructions of the joint research company. The weight coefficients \(c\)–\(i\) were determined according to the degree of the requirement to avoid violation of the soft constraints, following the manufacturer’s instructions.

4. Algorithm

4.1. Overall Structure

Figure 5 shows the overall structure of the proposed algorithm, which consists of a pre-cut lumber ordering unit and a packing simulator. A pre-cut lumber permutation is generated using metaheuristics in the pre-cut lumber ordering unit. According to this pre-cut lumber permutation, the pre-cut lumber is arranged in the packing simulator to compose the tiers and packings, the constraints are checked, and the objective function value is calculated. The aforementioned steps are iterated a specified number of times, and the provisional solution with the smallest objective function value is output, and the algorithm ends.
4.2. Pre-Cut Lumber Ordering

Pre-cut lumber permutations are generated using metaheuristics. GA or ILS + Or-opt is used as metaheuristics.

4.2.1. GA

The GA searches for quasi-optimal solutions through artificial evolution with repeated selection, crossover, and mutation when designing a search algorithm using metaheuristics, it is important to balance between intensification and diversification of the search. We believe that the following algorithm has a high probability of achieving this, because the crossover sequence, inherits the parent solution, and the non-crossover sequence is diversified. Because this algorithm is well-known, a detailed description is omitted. In the following section, only the permutation crossover operation for permutation generation is described.

- S.1. Two parents are randomly selected. Let the selected parents be X and Y and have the following genes.
  \[ X = (a, d, b, e, h, f, g, c), \quad Y = (e, f, c, d, g, a, b, h) \]

- S.2. Two crossover points \(i\) and \(j\) (\(i < j\)) are selected. At this point, the \(i\)-th, \((i + 1)\)-th, \(\ldots\), \((j - 1)\)-th, and \(j\)-th chromosomes are exchanged in a partial permutation (hereinafter referred to as the crossover sequence) (Fig. 6).

- S.3. If the same gene exists in the non-crossover sequence of X and crossover sequence of Y, the gene in the crossover sequence of X is moved to the position of the same gene as Y in the non-crossover sequence of X (Fig. 7). This operation is continued until there are no more same genes in the non-crossover sequence of X and the crossover sequence of Y.

- S.4. The genes in the Y-crossover sequence are moved, as in S.3 (Fig. 8).

4.2.2. ILS + Or-opt

In the permutation crossover of the GA, the same genes of two parents in a crossover sequence are moved to a non-crossover sequence and produce a new individual. Therefore, it is possible that a partial permutation with a good evaluation value will not be inherited by the next generation. This operation may improve the diversity of the solution search but reduce the concentration of the solution search. Therefore, we propose ILS + Or-opt, which generates a new permutation by using insertion and exchange neighborhoods for neighborhood operations when performing ILS. In the insertion neighborhood, a partial permutation is inserted in another place without changing the order, and a new permutation is generated (Fig. 9). In the exchange neighborhood, two partial permutations are exchanged without changing the order, and a new permutation is generated (Fig. 10).

- S.5. The genes in the crossover sequences of X and Y are exchanged, producing X’ and Y’ (Fig. 9).
  \[ X' = (f, e, b, d, g, a, h, c), \quad Y' = (d, a, c, e, h, f, b, g) \]
Tanizaki, T. and Yamashita, R.

Fig. 10. Insertion neighborhood.

Fig. 11. Exchange neighborhood.

• S.4. Perform a simple local search with $X'$ as the initial solution, and obtain the local optimum solution $x$.

• S.5. If $f(x) \leq f(x_{\text{seed}})$, perform (a) with probability $1$; and if $f(x) > f(x_{\text{seed}})$, perform (a) with probability $E(x, x_{\text{seed}})$, in accordance with Eq. (2). If (a) is not performed, (b) is performed.

$$E(x, x_{\text{seed}}) = e^{\frac{f(x) - f(x_{\text{seed}})}{T}}.$$  \hspace{1cm} (2)

Here, $f(x)$ represents the objective function value for $x$.

(a) $x_{\text{seed}} := x$, \hspace{0.5cm} $l := 1$,

(b) $l := \min\{l + 1, l_{\text{max}}\}$.

• S.6. If the end condition iteration is not satisfied, return to S.3. Here, iteration represents the number of ILS iterations.

• S.7. If the end condition maxcnt is satisfied, output a provisional solution, and terminate. Otherwise, update the search solution using the exchange neighborhood, and return to S.2. Here, maxcnt represents the number of iterations for the exchange neighborhood.

4.3. Packing Simulator

The packing simulator determines the pre-cut lumber to be placed in the first tier in the order other than the left end, right end, and both ends in the order determined by the pre-cut lumber ordering unit. When the first tier is completed, the pre-cut lumber to be placed in the second and subsequent tiers is determined sequentially (Fig. 12).

The algorithm is as follows:

• S.1. Arrange the pre-cut lumber in the $L$ direction with the tier length $\leq \gamma$ and $|\text{edge length } 1 - \text{edge length } 2| \leq \beta$.

• S.2. Arrange the pre-cut lumber in the same tier while tier width $1 \leq \delta$ and tier width $2 \leq \epsilon$.

• S.3. Arrange the pre-cut lumber in the same packing while packing height $\leq \eta$.

• S.4. Iterate S.1.–S.3. until there is no more lumber to be placed.

4.4. Pre-Cut Lumber Rotation

By rotating the pre-cut lumber when arranged, there is a possibility that the distance between the pre-cut lumber can be shortened, the filling rate per packing can be improved, and the number of packings can be reduced. Therefore, for pre-cut lumber other than at both ends, four rotations are performed: no rotation, rotation of the upper and lower surface (Fig. 13), rotation of the left and right surface (Fig. 14), and rotation of the end surface (Fig. 15). As a result, the rotation is determined such that the distance from the adjacent left pre-cut lumber is minimized.

5. Numerical Experiments

5.1. Conditions for Numerical Experiments

Numerical experiments were conducted using pre-cut lumber data provided by company A. Table 1 presents the number of pieces of pre-cut lumber in each dataset. In the numerical experiment, the objective function values of ILS + Or-opt and GA and those with and without rotation were compared. The parameters were set so that the calculation time would be within 1 h on a personal computer used for daily work in a manufacturing company similar to company A. To make the number of ILS + Or-opt and GA solution searches the same in the numerical experiment, the parameters were set as follows.

(1) ILS + Or-opt

• Number of local searches: 50.

• Iteration: 10000.

• Maxcnt: 10.

(2) GA

• Number of individuals: 500.

• Number of generations: 10000.
5.2. Results of Numerical Experiment

Tables 2—4 present comparisons of the results of the computer experiments using ILS + Or-opt and GA. The best, worst, and average values represent the results of 30 numerical experiments. The term “with” implies “with rotation,” and the term “without” implies “without rotation.” In each table, the gray cells indicate that the results were better for ILS + Or-opt than for GA.

Table 2 presents comparison results of the objective function values. No hard-constraint violations were encountered in any of the numerical experiments. Regarding the best value, the GA was better than ILS + Or-opt in five out of six cases for both “with rotation” and “without rotation.” Regarding the worst value, ILS + Or-opt was better than the GA in all of six cases for both “with rotation” and “without rotation.” Regarding the average value, ILS + Or-opt was better than the GA in five out of six cases for “without rotation.”

Table 3 presents the comparison results for the number of tiers. Regarding the best value, there was no difference between ILS + Or-opt and the GA for both “with rotation” and “without rotation.” Regarding the worst value, ILS + Or-opt was better than the GA in all of six cases for both “with rotation” and “without rotation.” Regarding the average value, ILS + Or-opt was better than the GA in five out of six cases for both “with rotation” and “without rotation.”

Table 4 presents the comparison results for the number of packings. Regarding the best value, ILS + Or-opt was
better than the GA in one out of six cases for “with rotation,” and there was no difference between ILS + Or-opt and the GA for “without rotation.” Regarding the worst value, ILS + Or-opt was better than the GA in three out of six cases for “with rotation” and in two out of six cases for “without rotation.” Regarding the average value, ILS + Or-opt was better than the GA in five out of six cases for both “with rotation” and “without rotation.”

Figure 16 shows a boxplot of the objective function values. In all cases, the variation in the objective function values of the GA was larger than that of ILS + Or-opt. The best value of the objective function of the GA was better than that of ILS + Or-opt. On the other hand, the worst value of the objective function of the GA was worse than that of ILS + Or-opt. The median of the objective function values was better for ILS + Or-opt than for the GA, except for case 4. Thus, the GA may search for a diversity of solutions, but ILS + Or-opt may be better for concentrating the solution search. These results indicate that the GA is most likely to find the best objective function value; however, on average, ILS + Or-opt is most likely to find a better objective function value than the GA.

Tables 5–7 present comparisons of the computer experiment results for “with rotation” and “without rotation.” We performed the comparisons to investigate the effect of the “with rotation” procedure of the packing simulator.

Table 5 presents a comparison of the results of the objective function values. Regarding the best value, “without rotation” was better than “with rotation” in four out of six cases for both ILS + Or-opt and the GA. Regarding the worst value, “with rotation” was better than “without rotation” in five out of six cases for ILS + Or-opt and in one out of six cases for the GA. Regarding the average value, “with rotation” was better than “without rotation” in four out of six cases for ILS + Or-opt and in three out of six cases for the GA. It was impossible to determine which objective function value was better for the algorithms with and without rotation. Considering Table 4, there was no relationship among the ratio of the width of the joint bracket to that of the pre-cut lumber, the ratio of the height of the joint bracket to that of the pre-cut lumber, and the objective function values of for “with rotation” and “without rotation.”

Table 6 presents the comparison results for the number of tiers. Regarding the best value, there was no difference between “with rotation” and “without rotation” for both ILS + Or-opt and the GA. Regarding the worst value, there was no difference between “with rotation” and “without rotation” for ILS + Or-opt, whereas for the GA, “without rotation” was better than “with rotation” in one out of six cases. Regarding the average value, there was no difference between “with rotation” and “without rotation” for both ILS + Or-opt and the GA. Considering Table 4, there was no relationship among the ratio of the width of the joint bracket to that of the pre-cut lumber, the ratio of the height of the joint bracket to that of the pre-cut lumber, and the number of tiers for “with rotation” and “without rotation.”
Table 5. Objective function values (with vs. without).

<table>
<thead>
<tr>
<th>No.</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>with</td>
<td>without</td>
<td>with</td>
<td>without</td>
</tr>
<tr>
<td>1</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
<tr>
<td>2</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
<tr>
<td>3</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
<tr>
<td>4</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
<tr>
<td>5</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
<tr>
<td>6</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
<td>145282</td>
</tr>
</tbody>
</table>

Table 6. Number of tiers (with vs. without).

<table>
<thead>
<tr>
<th>No.</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>with</td>
<td>without</td>
<td>with</td>
<td>without</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 7. Number of packings (with vs. without).

<table>
<thead>
<tr>
<th>No.</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>with</td>
<td>without</td>
<td>with</td>
<td>without</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 8. Objective function values (‘with’ − ‘without’).

<table>
<thead>
<tr>
<th>No.</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
<th>ILS + Or-Opt</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>with</td>
<td>without</td>
<td>with</td>
<td>without</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

6. Conclusions

We developed an algorithm for packing formation support systems using ILS + Or-opt and performed numerical experiments for comparing it with a GA. Additionally, we also proposed the rotation of pre-cut lumber for improving the filling rate per packing. Numerical experiments were performed for 30 cases, and the following results were obtained:

1. The best objective function value of the GA was better than that of ILS + Or-opt.
2. The worst objective function value of ILS + Or-opt was better than that of GA.
3. The average objective function value of ILS + Or-opt was better than that of GA.
The variation in the objective function value of the GA was larger than that for ILS + Or-opt. Thus, the GA may search for a diversity of solutions, but ILS + Or-opt may be better for concentrating the solution search. These results indicate that the GA is more likely to find the best objective function value; however, on average, ILS + Or-opt is more likely to find a better objective function value than the GA. Because ILS + Or-opt is more likely to inherit a good solution than the GA, it is thought that ILS + Or-opt intensively searches around the good solution. This resulted in less variation in the evaluated values of the solution. Therefore, ILS + Or-opt is more suitable for production planning related to sequential planning for a company than the GA. It was also found that the rotation of the pre-cut lumber was not related to the improvement of the objective function value. This is because the rotation of the pre-cut lumber did not affect the number of tiers or packings, which are large weighting coefficients of the objective function value. In the future, we will investigate the practical application of this algorithm in a factory.

References:


Application of Metaheuristics to Packing Formation Support Systems of Pre-Cut Lumber Factory


Name: Takashi Tanizaki
Affiliation: Professor, Department of Informatics, Faculty of Engineering, Kindai University
Address: 1 Takaya-Umenobe, Higashi-Hiroshima, Hiroshima 739-2116, Japan

Brief Biographical History:
1984 Received M.S. in Engineering from Kyoto University
1984-2009 Sumitomo Metal Industries, Ltd. (Nippon Steel Corporation)
2005 Received Ph.D. in Informatics from Kyoto University
2009- Professor, Kindai University

Main Works:

Membership in Academic Societies:
• Operations Research Society of Japan (ORSJ)
• Society for Serviceology (SfS)
• Japan Association for Management Systems (JAMS)

Name: Ryohei Yamashita
Affiliation: Graduate School of Systems Engineering, Kindai University
Address: 1 Takaya-Umenobe, Higashi-Hiroshima, Hiroshima 739-2116, Japan

Brief Biographical History:
2021 Received Bachelor degree from Faculty of Engineering, Kindai University