Kiln Drying Operations Scheduling with Dynamic Composition of Loading Patterns

Philippe Marier
Jonathan Gaudreault
Thomas Noguer

Abstract

Planning and scheduling wood lumber drying operations is a very difficult problem. The literature proposes different methods aiming to minimize order lateness. They all make use of pre-established kiln loading patterns that are known to offer good physical stability in the kiln and allow full kiln space utilization. Instead, we propose a mixed integer programming (MIP) model, which can be used to generate loading patterns ''on the fly.'' This MIP model can be integrated into existing kiln drying operation planning/scheduling systems in order to improve their solutions. We show how this integration can be done by adapting a state of the art drying operations planning and scheduling methodology from the literature. We compare the solutions obtained by this system using the predefined loading patterns versus the solutions it generates if it is connected to our loading patterns generator MIP model. The study shows it is much better to dynamically create loading patterns than to use predefined ones, as most North American sawmills do.

The problem of planning and scheduling softwood lumber drying operations is very complex. Drying is carried out in batches, each one containing only compatible products that can be dried together. The batch should be made so as to form a stack (rectangular prism) that is geometrically stable and that fills the entire kiln. Because kilns are huge, each batch contains many pieces and contributes to simultaneously satisfy several orders of finished products having different due dates.

Each batch is defined by a loading pattern that specifies the enumeration and specific positioning of lumber bundles in the kiln. Because each kiln can be loaded in millions of different ways (Fig. 1 gives an example of a batch/stack), most North American companies define a set of standard loading patterns in advance that are known to fill the kilns while providing good stack stability. These patterns are mainly based on workers' experience. The task of the planner/scheduler is then to determine when (and for which kiln) each pattern should be used (each pattern can be used repeatedly over the planning horizon) while minimizing order lateness.[1] For this paper, order lateness is defined as the volume not delivered on time times the number of periods during which the back order is present.

For most companies, this planning/scheduling is done manually. A mixed integer programming (MIP) model has already been presented for this problem (Gaudreault et al. 2011), but for instances of industrial size, the MIP model rarely provided good solutions even after long computing time. A constraint programming (CP) model, which provides good solutions for very short computation times, was also introduced in the same article. CP (Bartak 1999b) has the advantage of permitting nonlinear constraints (although this was not a required characteristic in our case). It also allows the expert to tailor the optimization engine based on its knowledge of the problem by selecting the constraint propagation algorithms and the search strategy (Bartak 1999b, Rossi et al. 2006).

[1] This is a combined planning and scheduling problem (Bartak 1999a), since we simultaneously perform process planning (loading pattern decision) and scheduling (Gaudreault et al. 2011).

The authors are, respectively, research professional, professor, and MSc student, FORAC Research Consortium, Univ. Laval, Québec, Canada (Philippe.Marier@forac.ulaval.ca [corresponding author]; jonathan.gaudreault@ift.ulaval.ca; Thomas.Noguer.1@ulaval.ca).

This paper was received for publication in September 2020. Article no. 20-00050.

Forest Prod. J. 71(2):101–110.
doi:10.13073/FPJ-D-20-00050

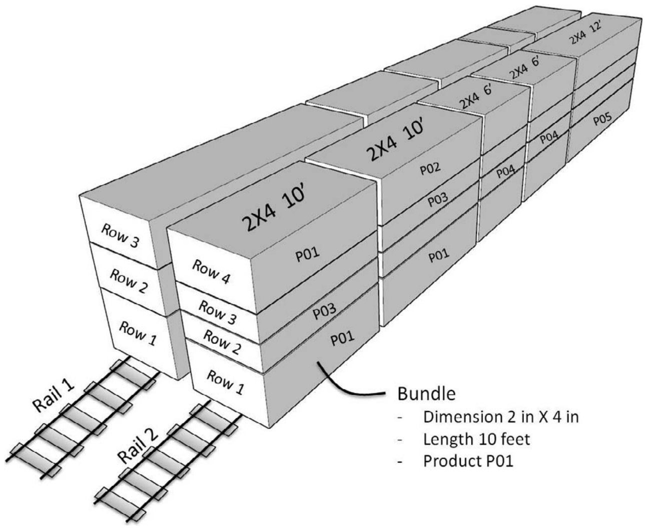


Figure 1.—A kiln loading pattern showing rails, rows, and bundles of different products.

We were approached by a company[2] that sought to determine whether allowing a greater number of kiln loading patterns would increase the company's performance. The general idea was that with a larger number of loading patterns, we would have more chance to plan/ schedule batches that best fit customers' orders.

Rather than using a huge list of predefined loading patterns, we decided to provide the system with the ability to dynamically generate them ''on the fly.'' Our aim is not to compare with actual or manual planning of the kilns. Such comparison has been done in other studies (Gaudreault et al. 2011). We will instead evaluate to what extent the use of a larger number of kiln patterns could improve the solutions obtained by automated scheduling tools.

The paper is organized as follows. We first present preliminary concepts related to the planning and scheduling of wood drying operations. In particular, we present an existing CP solution methodology for the problem as we later improve that approach. We then describe an MIP model we propose in order to dynamically generate loading patterns, and we show how this model can be embedded into the global CP solution methodology. Finally, we present experiments showing how this new model greatly improves the quality of the solutions found (reducing order lateness).

Preliminary Concepts

Wood drying

Wood drying is a process that reduces the moisture content of wood in order to meet industry standards. For softwood lumber, this process can take several days. It is done in batches of several bundles of lumber which are put together in large kilns. The bundles of a given kiln load can be of different lengths, but they must generally be of the same species and of the same section (width and thickness), thus requiring the same drying process. Kiln drying time might be different if the green lumber is staying in the yard for several weeks. In our study, the company policy was to keep inventory low, and thus the time in the yard is not very long. Hence the kiln drying time difference due to different time laying in the yard is not taken into consideration by our partner company. Maturana et al. (2010) give a good description of the overall lumber production process.

On a single site, multiple kilns can be used in parallel, each having its own dimensions, capacity, and drying technologies. Individual kilns may have different drying processes. Since the planning can be done at any time and the drying times are relatively long (24 to 96 hours), it is possible that during planning, some kilns are already drying and, hence, are not available before a certain period.

Bundles to be dried are generally prepared on carriages outside of the kilns. They are pushed on rails into the kiln when they are ready and when the kiln becomes available. The number of rails may vary per kiln. Also, on the same rail, bundles can be stacked in different rows. The number of bundles that can be stacked depends on the height of the kiln and the height of the bundles. In our case study, bundles to be dried come from different sawmills, and thus the height of the bundles may vary. For stacking stability, we have to implement a special constraint enforcing that all bundles of a given row in a kiln must have the same height.

Lumber drying planning and scheduling

Short-term planning/scheduling of lumber drying operations consists of finding a plan showing how each kiln should be used for the next 2 or 3 weeks. Thus, for each kiln, the plan shows which drying process is to be applied, when (start time and end time) and which products compatible with the process are to enter the kiln at that time. Figure 2 shows an example of a solution for the planning of three kilns over a period of 2 weeks (28 periods of 12 h). Each gray block specifies the drying process to be used. It should also be specified for each operation how the kiln must be loaded (in the figure, we show, as an example, the detailed loading for only one of the operations). In fact, for each load, the plan must indicate the number of bundles of each species/dimension/length.

When the number of different products that can be dried is large (most lumber dimensions at our partner company can be available in lengths of 8, 10, 12, 14, and 16 in.), enumerating the various possible product combinations to form loading patterns is unthinkable. For this reason, companies typically work with a small number of predefined loading patterns (a few dozen). However, it would be possible to dynamically build the kiln loading

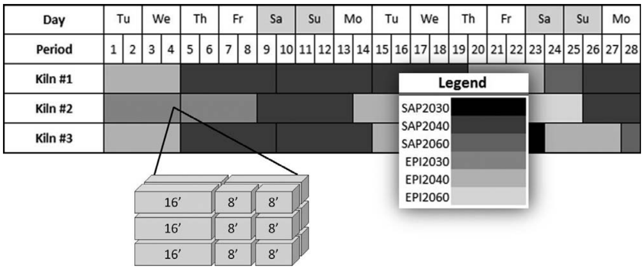
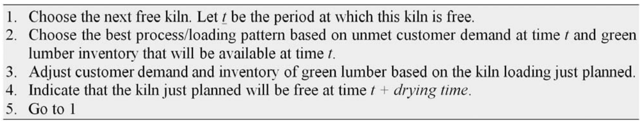


Figure 2.—Production plan for three kilns. Each operation identifies the selected drying process (gray) as well as a loading pattern indicating which specific compatible products will be dried.

[2] Resolute Forest Products is the largest softwood lumber producer in Eastern Canada. They operate about 50 kilns in Quebec and Ontario (Canada).

patterns at the time of planning. This is what we propose in the ‘‘Using a MIP model to define an optimal loading pattern’’ section.

For the detailed planning of softwood drying operations, Gaudreault et al. (2010) formally describe the problem and show a heuristic for solving the problem when preset loading patterns are provided. A heuristic for the multi-period planning of all the kilns has been proposed (Fig. 3).



1. Choose the next free kiln. Let t be the period at which this kiln is free.
2. Choose the best process/loading pattern based on unmet customer demand at time t and green lumber inventory that will be available at time t.
3. Adjust customer demand and inventory of green lumber based on the kiln loading just planned.
4. Indicate that the kiln just planned will be free at time t + *drying time*.
5. Go to 1

Figure 3.—Greedy heuristic to solve the kiln drying operation scheduling problem, from Gaudreault et al. (2010).

In Gaudreault et al. (2011), the same problem is modeled as a MIP problem as well as a CP model. The MIP model does not provide good quality solutions, even after several hours of computing time. The CP model finds very good solutions in a very short time, but optimal solutions have not been obtained for industrial-sized instances.

Interestingly, the search strategy employed by the CP model can be seen as a generalization of the heuristic just presented above (Fig. 3). At step 2, rather than just choosing the best process/loading pattern as proposed by the heuristic (i.e., the one reducing the objective function by the greatest value), we sort all possible processes/patterns in descending order of preference. A loading pattern is eligible (possible) if all of its components are available in inventory (all the quantities for each different product type are available) at the time it is considered for scheduling.

The combination of all these possibilities defines a complete tree (Fig. 4 shows a simple case). We reach a leaf when all the drying kilns have been scheduled over the planning horizon. This corresponds to a solution. In Figure 4, reaching leaf node 12 corresponds to the following solution (through nodes 1, 8, 9, 12): at time t = 0, schedule kiln 1 using loading pattern R3. At time t = 0, schedule kiln 2 using loading pattern R5. At time t = 8, schedule kiln 2 using loading pattern R5. At time t = 10, schedule kiln 1 (note: the last loading pattern is not depicted in Fig. 4). At that point, if computation time is still available, we can resume search in the hope of finding better solutions. To do so, we can step back to the previous visited node (this process is called ‘‘backtracking,’’ and it is precisely what is done in classical ‘‘depth-first search’’).

Exploring the complete tree would ensure finding the optimal solution to the problem, but it is such a large tree that it is computationally impossible for real industrial problems.

After reaching a leaf, instead of systematically backtracking to the previous visited node, we could also backjump to any node that has process/loading patterns still unexplored.

Actually, the search strategy employed to decide in what order the nodes must be visited during backtracking has an important impact on the performance of the algorithm. Gaudreault et al. (2011) have shown that the search strategy called limited discrepancy search (LDS; Harvey and Ginsberg, 1995) is much more efficient than basic strategies (e.g., depth-first search)[3] for this problem.

Other authors propose partial solutions to this problem. Gascon et al. (1998) worked on the development of an integrated system for the management of drying hardwood. In their case, loading pattern issues do not apply because there are no different bundle sizes and there are few different products (species). Aggarwal et al. (1992) also present a decision support system for planning kiln operations for a plant producing hardwood furniture. Their planning model makes wood supply/drying decisions at minimum cost for a furniture production plan. The problem of loading patterns is not present in the case that they studied. Cheng Huang et al. (1998) worked on a problem similar to the one of Aggarwal et al. Their heuristic considers lumber procurement and drying at the same time. As seen in many other papers, kiln charge is tested against global kiln capacity, and no physical validation is made to ensure bundle stacking is feasible and can physically enter the kiln. Arman et al. (2001) also have developed a kiln drying planning model for hardwood used in furniture production. The authors mention that due to the nondeterministic polynomial (NP)-completeness of the problem, the use of an MIP is inefficient. They developed a heuristic for this problem, but it does not look at the operational feasibility of kiln loading.

The method of Gaudreault et al. (2011) still seems to be state of the art, since it has been chosen again this year by a large integrated company from Quebec, which is implementing it (with our support) for a group of five sawmills.

Using a MIP Model to Define an Optimal Loading Pattern

The approaches used to solve the kiln drying operation scheduling problem, whether manual planning, greedy heuristic (Gaudreault et al. 2010), the MIP or CP model (Gaudreault et al. 2011), all share a common subproblem consisting in choosing which loading pattern (from a list of preset patterns) should be used to fill a specific kiln at a specific time. As we mentioned, most North American companies define a set of standard loading patterns in advance, which are known to fill the kilns while providing good stack stability, and select among them.

In this section, we propose the use of a MIP model that allows dynamically generating a loading pattern/optimal stacking. These loading patterns generated ‘‘on the fly’’ can be used in place of the predefined ones in any of the previously mentioned approaches to global kiln planning and scheduling.

Literature review—creating optimal loading pattern

The formulation of the problem can, in some ways, be related to a 2D strip packing problem (2DSP) with tardiness objective function and also with spatial scheduling problems. We reviewed the approaches presented by Bekrar and Kacem (2009), Boschetti and Montaletti (2010), Côté et al. (2013), and Kenmochi et al. (2009). Despite the resemblance, the approaches to the 2DSP do not well apply to our problem for the following reasons: (1) 2DSP do not enforce

[3] Best-first search does not work because the computation of the bounds in the nodes is very poor, as a result of the highly combinatorial structure of the problem. We believe this also explains why the MIP does not perform well for these types of problems.

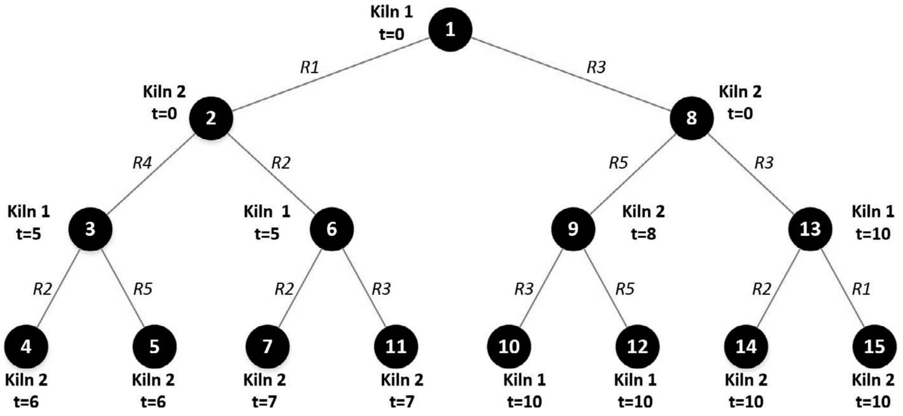


Figure 4.—Example of a search tree representing the solution space for the kiln drying planning problem. Each node corresponds to a choice point where you have to choose a drying process/loading pattern.

restriction on how the stacking is done, that is, elements of any size can be stacked on top of each other; (2) Although empty spaces are normally minimized in the 2DSP problem, solutions with holes are valid, whereas this is not acceptable in kiln loading patterns. For the spatial scheduling problem, we reviewed the approaches given by Garcia and Rabadi (2013, 2016). Even though spatial scheduling problems have some commonalities to our problem, there is quite a difference in the problem definition: (1) in spatial scheduling, no jobs have to be scheduled at the same time; (2) in making kiln loading patterns, choosing a product for a given location imposes a constraint on the candidate products that can be chosen for the position above/below and on the same row: products on the same row must have the same height; products stacked must be of same length; (3) all products to dry at the same time within a kiln must require the same drying process (drying time and drying parameters); and (4) multiple kilns must be scheduled at the same time.

Generating a kiln loading pattern has some resemblances to the problem of container loading. For these problems, Bischoff and Ratcliff (1995) gave a review of many practical requirements, which are not always well considered in the solution methodologies. According to them, these problems may be divided into two categories: on the one hand, cases where the whole of a given consignment of goods is loaded, and on the other hand, cases where it is possible to leave some of the cargo behind. Our problem clearly falls into the category where we can leave some of the cargo behind, since we have to select the bundles among the planned available inventory. Among the different constraints for these problems, loading priorities, although important, are hardly ever explicitly considered in container loading algorithms (Bortfeldt and Wäscher 2013).

When boxes or bundles are to be stacked, stability is of great importance and beyond container space utilization, constraints related to stability are in fact often considered to be one of the most important issues. Despite that, Bortfeldt and Wäscher (2013) found that these concerns are often not considered explicitly, since authors argue that stability is a consequence of load compactness when high container space utilization can be guaranteed, which is typically true for problems where only a subset of the items can be packed.

Allocation constraints (sometimes referred to as connectivity constraints or separation constraints) demand that certain items or classes of items not be loaded into the same container. Such constraints exists in kiln drying softwood lumber, since only the bundles requiring the same drying process can go together in the kiln at the same time. Of the 163 papers reviewed by Bortfeldt and Wäscher (2013), 8 percent of them consider these types of constraints, much of them as soft constraints (in our case, it needs to be a hard constraint).

Container loading problems typically involve many more constraints, which may involve box orientation, load bearing strength of items, multidrop situations, container weight limit, weight distribution within a container, loading priorities, relative positioning constraints, and loading complexity (Bischoff and Ratcliff 1995, Bortfeldt and Wäscher 2013). To tackle these problems with real-life applicability, many different approaches have been used. As mentioned by Sorensen et al. (2016) and Junqueira et al. (2013), these problems are NP-hard, and for real-life sized problems, approaches other than optimization should be used (Dereli and Sena Das 2010, Gonzalez et al. 2013, Yu et al. 2014, Zeineldin and Morsy 2015).

One can see that despite the appearance of resemblance between the creations of kiln loading patterns and container loading problems, there are a lot of practical requirements specific to each application. Since many of the constraints pertaining to container loading problems are not present in our context, we developed a MIP model, which is easy to solve for the dynamic generation of kiln loading patterns.

Proposed model

For a particular kiln that is empty at a specific time, the MIP model we developed identifies: (1) which drying process will be used (this defined compatible products); (2) for each rail, how many bundles of different lengths will be placed on each row; (3) the height of the bundles in each row of each rail of the kiln; and (4) which specific product constitutes each bundle. The model takes into account the physical dimensions of the kiln, the inventory that will be available to be dried at the time of starting the drying process, and the demand for the different finished products (volume required in each period, per product). This might be a combination of firm orders, projected sales and production targets, depending on the company's objective for this planning. The objective of the model is to configure a valid stacking which minimizes order lateness. Figure 1 shows a valid loading pattern for a kiln with two rails. The stacking is independent from one rail to the others. Bundles within a row must be of the same height, and only bundles of same length can be stacked on top of each other.

The MIP solution does not specify the exact location of each bundle selected to make the loading. Decision variable specifies the number of bundles of each product to put in each row on each rail. With that information, it is up to the operator to decide where on the row to put the products of different length, as long as only bundles of the same length can be stacked on top of each other. A constraint ensures there is the same amount of lumber of a given length in each row of a rail, guaranteeing a feasible stacking.

For the company involved in this project, a high proportion of their lumber production is made to order, and its objective is thus to reduce lateness. For other companies, the objective can be linked to some production target. In both cases, this is modeled as a demand that is characterized by a volume of a specific product with a required date.

We recall that for this paper, lateness is defined as the volume not delivered on time times the number of periods during which the back order is present. This computation is done for all of the products having demand over the planning horizon *T*. Since the planning is not done globally in a single mathematical model, the heuristic aims at reducing the lateness by favoring, while planning for a given kiln, the drying of the products which, if not dried immediately, would cause the most lateness. We thus try to maximize the reduction of the lateness by drying the right products in time.

Let *T* be the planning horizon, and let $w_{v,t}$ be the volume of product *v* required at time *t*. Let d_s be the kiln dry time when the drying process *s* is used. In drying a product (which may generate, after planing, a final product *v* required at time *t*), we reduce the lateness by $\Phi_{v,t,s} \times \min\{T-t,\ T-d_s\}$, where $\Phi_{v,t,s}$ is the decision variable that represents the volume of the final product *v* required at time *t* that will be satisfied by this solution for the kiln loading according to drying process *s*. The objective function minimizing the lateness can thus be expressed as follows:

$$\text{maximize} \sum_{v\in V}\sum_{t=1}^{T}\sum_{s\in S}\Phi_{v,t,s}\min\{T-t, T-d_s\}$$

Sets, parameters, and decision variables

Sets

- $\boldsymbol{L}$ Set of bundles length *l*
- $\boldsymbol{P}$ Set of green lumber products *p*
- $\boldsymbol{P}_l$ Set of green lumber products *p* that are packed in bundles of length *l*. $\boldsymbol{P}_l \subseteq \boldsymbol{P}$
- $\boldsymbol{V}$ Set of finished lumber products *v*
- $\boldsymbol{H}$ Set of bundles height *h*. This set also includes the height $h = 0$
- $\boldsymbol{S}$ Set of drying processes *s* (compatible with the kiln being planned)
- $\boldsymbol{S}_p$ Set of drying processes *s* which can be used to dry product $p \in \boldsymbol{P}$. $\boldsymbol{S}_p \subseteq \boldsymbol{S}$

Parameters

- Ψ Large number (we use total volume of products in inventory) (unit: pmp)
- a Maximum number of stacking rows in the kiln
- $\max h$ Maximum stacking height in the kiln (unit: inches)
- n Number of rails for the kiln
- $n_{\max}$ Maximum use of kiln length (unit: feet)
- $n_{\min}$ Minimum use of kiln length (unit: feet)
- i_p Volume of product *p* available for drying (when kiln is scheduled to start its drying process) (unit: pmp)
- u_p Volume of one bundle of product *p* (unit: pmp)
- h_p Height of one bundle of product *p* (unit: inches)
- $w_{v,t}$ Volume of final product *v* required for period *t* (unit: pmp)
- d_s Kiln dry time, in number of periods, when process *s* is used. (unit: number of periods)
- $o_{p,v}$ Volume of finished product *v* obtained from the planing of product *p* after kiln drying. (unit: pmp)
- T Planning horizon, in number of periods

Decision variables

- R_s = 1if process *s* is used for the kiln, 0 otherwise (binary)
- $N_{l,r}$ Number of bundles of length *l* (in feet) put on any one row of rail *r* (positive integer)
- $X_{p,r,g}$ Volume of product *p* assigned to rail *r* in row *g* (positive real, unit: pmp)
- $Q_{p,r,g}$ Number of bundles of product *p* on rail *r* in row *g* (positive integer)
- $I_{r,h,g}$ = 1 if height *h* is chosen for the bundles of row *g* for rail *r*, 0 otherwise (binary)
- $G_{r,g}$ = 1 if row *g* on rail *r* is used, 0 otherwise (binary)
- $E_{r,g,l}$ Slack variable allowing to consider empty rows (positive real)
- J_v Volume of final product *v* that will be obtained from the kiln drying plan (positive real, unit: pmp)
- $\Phi_{v,t,s}$ Satisfied volume of product *v* required at time *t* from production of finished products done using drying process *s* (positive real, unit:pmp)

Constraints

Constraints only allow us to dry at the same time those products that call for the same drying process. Other constraints also require that all bundles on the same row have the same height and that each row of the same rail has the same number of bundles of each length. This ensures stability of the stacking. Each kiln also has specific dimensions, so the length of the bundles put together in a row should not exceed these dimensions. Furthermore, in order to maximize the use of space, the sum of the lengths of bundles of one row must be greater than a specified minimum length. Below are the constraint details.

We select only one drying process *s* for the kiln.

$$\sum_{s\in S} R_s = 1 \tag{1}$$

We fill the kiln such that the total length of all bundles is within acceptable range. Hence, for each rail, the sum of bundles length on a row must be between $n_{\min}$ and $n_{\max}$.

$$n_{\min} \leq \sum_{l\in L}(l \cdot N_{l,r}) \leq n_{\max} \quad \forall r \in \{1..n\} \tag{2}$$

Only products compatible with the selected drying process can be dried.

$$X_{p,r,g} \leq \Psi \sum_{s \in \mathrm{S}_p} R_s \quad \forall r \in \{1..n\}, g \in \{1..a\}, p \in \mathrm{P} \tag{3}$$

We cannot use more products than what are available in inventory.

$$\sum_{r=1}^{n} \sum_{g=1}^{a} X_{p,r,g} \leq i_p \quad \forall p \in \mathrm{P} \tag{4}$$

The MIP model allows for a maximum number of rows in the kiln (parameter *a*), but according to the height of the selected bundles to be dried, it is possible that not that many rows can be used to fill the kiln up to the ceiling. A slack variable ($E_{r,g,l}$) is used in Equation(6) to ensure constraint satisfaction when not all rows are used. Equation (5) makes sure this slack variable is null for rows that are used.

$$E_{r,g,l} \leq \Psi(1 - G_{r,g}) \quad \forall r \in \{1..n\}, g \in \{1..a\}, l \in \mathrm{L} \tag{5}$$

For any given row, bundle selection must be such that it complies with the chosen length assortments for the rail ($N_{l,r}$): there must be the same number of bundles of each length in every row of the rail.

$$\left(\sum_{p \in \mathrm{P}_l} \frac{X_{p,r,g}}{u_p} \right) + E_{r,g,l} = N_{l,r} \quad \forall r \in \{1..n\}, g \in \{1..a\}, l \in \mathrm{L} \tag{6}$$

Equation (7) computes the integer number of bundles of each product used in the loading plan.

$$Q_{p,r,g} = \frac{X_{p,r,g}}{u_p} \quad \forall p \in \mathrm{P}, r \in \{1..n\}, g \in \{1..a\} \tag{7}$$

All the bundles on the same row of a rail must have the same height. Equation (8) guarantees that only one height is chosen per row per rail.

$$\sum_{h \in \mathrm{H}} I_{r,h,g} = 1 \quad \forall r \in \{1..n\}, g \in \{1..a\} \tag{8}$$

All products on a row must be of the selected height.

$$X_{p,r,g} \leq \Psi I_{r,h_p,g} \quad \forall p \in \mathrm{P}, r \in \{1..n\}, g \in \{1..a\} \tag{9}$$

A row is set as not being used if the selected height for that row is null (Eq. 10). It is assumed that bundle height is ≥ 1 for any bundles in the system. Equation(11) assures that height of unused row is zero.

$$\sum_{h \in \mathrm{H}} (h \cdot I_{r,h,g}) \geq G_{r,g} \quad \forall r \in \{1..n\}, g \in \{1..a\} \tag{10}$$

$$\sum_{h \in \mathrm{H}} (h \cdot I_{r,h,g}) \leq \Psi G_{r,g} \quad \forall r \in \{1..n\}, g \in \{1..a\} \tag{11}$$

For each rail, in the stacking of the bundles, one cannot exceed the maximum stack height given by max *h*.

$$\sum_{g=1}^{a} \sum_{h \in \mathrm{H}} (h \cdot I_{r,h,g}) \leq \max h \quad \forall r \in \{1..n\} \tag{12}$$

The decision variable J_v is the volume of finished product *v* that is obtained from the drying plan. Since the planing process is like a disassembly (different finished products are produced at the same time from a given bundle), $o_{p,v}$ is a parameter that gives the volume of finished product *v* obtained from the planing of one unit of volume of product *p* dried.

$$J_v = \sum_{r=1}^{n} \sum_{g=1}^{a} \sum_{p \in \mathrm{P}} X_{p,r,g} o_{p,v} \quad \forall v \in \mathrm{V} \tag{13}$$

Let the decision variable $\Phi_{v,t,s}$ be the portion of the final product *v* required at time *t* satisfied from the drying plan using drying process *s*, and let $w_{v,t}$ be the volume of final product *v* required at time *t*. Equation (14) states that we cannot satisfy more than the product required quantity, and Equation (15) that we cannot use more than the produced quantity to satisfy the final product requirement. Equation (16) makes sure no final product requirement is satisfied from a drying process that is not selected.

$$\sum_{s \in \mathrm{S}} \Phi_{v,t,s} \leq w_{v,t} \quad \forall v \in \mathrm{V}, t \in \{1..T\} \tag{14}$$

$$\sum_{t=1}^{T} \sum_{s \in \mathrm{S}} \Phi_{v,t,s} \leq J_v \quad \forall v \in \mathrm{V} \tag{15}$$

$$\Phi_{v,t,s} \leq \Psi R_s \quad \forall s \in \mathrm{S}, v \in \mathrm{V}, t \in \{1..T\} \tag{16}$$

Integrating the MIP model into the global solution procedure

The sections ''Proposed model,'' ''Sets, parameters, and decision variables,'' and ''Constraints'' describe the MIP model used to generate an optimal loading pattern for an available kiln giving current material availability and unsatisfied demand. The model will select the lumber to dry in order to maximize the reduction of order lateness. It is straightforward to change step 2 of the greedy heuristic (Fig. 3) in order for it to use the process/loading pattern generated by the MIP instead of choosing the best process/loading pattern from a predetermined list.

To integrate the MIP into the procedure involving the search tree (see the ''Lumber drying planning and scheduling'' section) we solve the MIP each time we enter a node, and this generates the loading pattern (according to a drying process) and a first branching from that node. At a later time, if backtracking is done up to this node, the MIP is run again to generate another loading pattern (and another branching). Since the algorithm remembers the previously selected drying processes, we remove from the input data the ability of the MIP to select any of the previously selected drying processes. This ensures a different solution will emerge from the new branching.

When integrating our MIP model within the constrained programming model proposed by Gaudreault et al. (2011), each branching from one level to the next below is done using the MIP which selects a process (A1, A2, A3, A4, and A5 in Fig. 5) and creates a new loading pattern using that process.

Looking at Figure 5, say we are at node 3a and we have determined the next available kiln is kiln No. 2 at time 6. We run the MIP model with the set of drying processes $\boldsymbol{S} =$

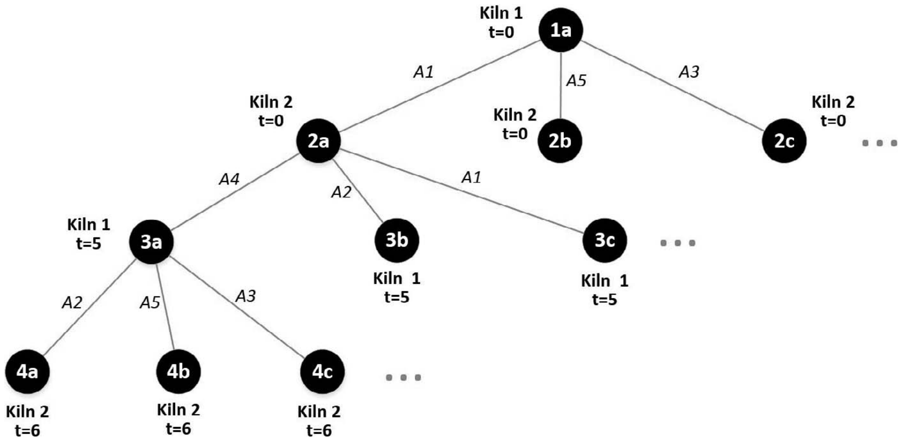


Figure 5.—Search tree dynamically constructed using the MIP model.

{A2, A3, A5, . . .}. At that point, the MIP determines that the best process is A2 (as shown in Fig. 5, leftmost branch) and the model also has generated the loading pattern (selecting the right products for the unsatisfied demand). Having selected process A2, we are now at node 4a. Let's say we want to explore an alternative solution. We have to backtrack in the tree and make a different choice with regard to drying process/loading pattern for the node being considered. If we backtrack by only one level, we are at node 3a and need to run the MIP again to determine the best drying process/loading pattern combination for that level. This time, we need to remove process A2 from the set S of drying processes, since this is the one that was chosen previously. The model would then run with $\boldsymbol{S} = \{A3, A5, \ldots\}$, ensuring a different [partial] solution.

Of course, a search strategy must be defined in order to select the node to backtrack to each time a solution is found. We propose the use of the LDS strategy, since it is the one that gave the best results in Gaudreault et al. (2011).

Evaluation Using Industrial Data

We have shown that all effective approaches to the problem have a subproblem that consists of choosing a loading pattern. In this section, we use these different methods, but we replace the step of selecting the loading pattern from a predefined set, by a step which dynamically builds a loading pattern using the MIP model from the ''Using a MIP model to define an optimal loading pattern'' section.

The different approaches we compare are (1) the greedy heuristic from Gaudreault et al. (2010) using a set of predefined patterns (hereafter called ''Heuristic-Fixed''), (2) the CP model solved using an LDS search strategy, as in Gaudreault et al. (2011) (''LDS-Fixed''), (3) the greedy heuristic modified to dynamically select the loading patterns at each step (''Heuristic-Dynamic''), and (4) search space alternatives with LDS while creating loading patterns dynamically (''LDS-Dynamic'').

The models were tested with real data from a representative average-sized sawmill in Quebec (Canada) using data from four different periods. We use the same cases as in Gaudreault et al. (2011). In all four cases, 40 different products types can be dried in either of the two identical kilns. Although this number of products may seem small, it is typical and representative of what we find at the drying stage in the industry. We look for a plan that will minimize lateness over a horizon of 60 periods of 12 hours. For methods with predefined loading patterns, 150 preset loading patterns are available to meet between 52 and 77 orders. To solve the sub problems of constructing loading patterns, the solver used is CPLEX (version 12.6) with a gap of 0 percent. The experimentations were run on a 64-bit operating system (Windows Server 2008 R2) with 6 GB of RAM and processor Intel Xeon X5675 at 3.07 GHz.

Figure 6 compares the results for the different methods according to computing time. Recall that the heuristic solutions (method 1 and 3) correspond to the first solution found by the corresponding LDS search (methods 2 and 4). Therefore, in the following charts, the first value (left) is also the value of the heuristic.

In each case, the first solution is found in less than 30 seconds. We gave 300 minutes for the LDS search, and graphs show the improvement of the solution (reduction of the objective function) over time. The graph in Figure 6 shows that the majority of the gain with the LDS method is obtained in the first few minutes of computation.

Table 1 compares the quality of the solutions obtained after 300 minutes. The percentage is the lateness reduction compared with the reference method (Heuristic-Fixed, i.e., with pre-established loading patterns). Comparing the two methods with predefined loading patterns (Heuristic-Fixed vs. LDS-Fixed) shows a result consistent with those obtained in Gaudreault et al. (2010). The search method with LDS improves the solution by 5 to 10 percent in cases 1, 2, and 4 and by more than 45 percent in case 3. If we look at the impact of generating patterns dynamically, we notice that the heuristic (Heuristic-Dynamic) is more effective than when it uses the predefined patterns (it allows reducing lateness by 43% on average), and it beats LDS-Fixed (reducing lateness by 33% on average) even if the latter took 5 hours to compute. This clearly shows the positive industrial impact associated with the dynamic creation of loading patterns. The gains are related to the fact that our model can create new loading patterns that help to further reduce lateness. The loading patterns generated by the model are focused on the need to minimize lateness, and more opportunities are explored than when using predefined loading patterns.

Using the LDS strategy to explore alternative solutions (LDS-Dynamic) we get the best solutions found to date for

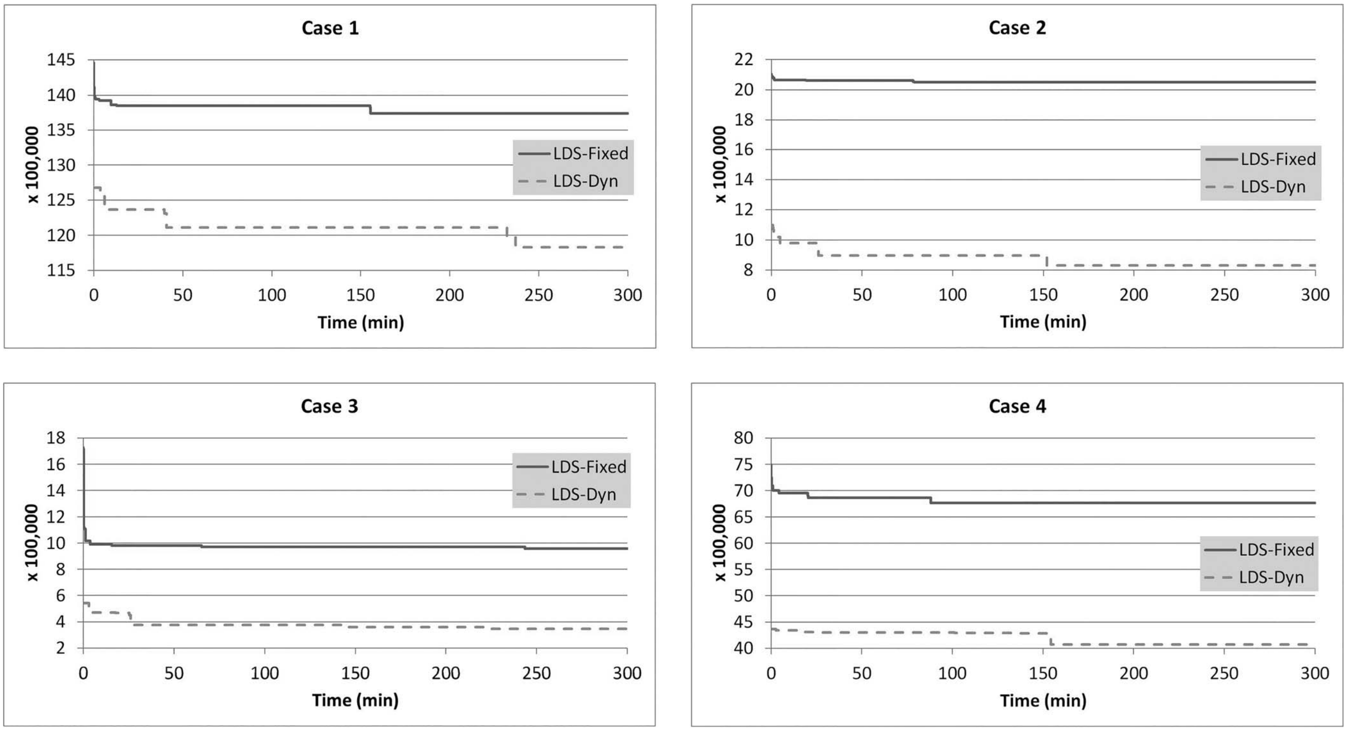


Figure 6.—Order lateness according to computation time for the various methods. The first point on the left of each curve matches the solution value that would be obtained with the corresponding heuristics (fixed and dynamic).

this problem. This approach allows an average gain of 51 percent compared with the heuristic exploiting predetermined fixed patterns, and 8 percent compared with the heuristic using dynamic loading patterns.

We should mention that the LDS-Dynamic method requires much more time to explore each node than the LDS-Fixed method. Hence, this method explored an average of 14,000 nodes over 5 hours, while the LDS-Fixed method explored 4,000,000 nodes in the same period of time. However, nodes from the LDS-Fixed are not the same as the ones from the LDS-Dynamic. Indeed, at each node of the LDS-Dynamic tree, the MIP creates the best loading pattern among the vast quantity of possible ones, while, in a given node of the LDS-Fixed tree, only one loading pattern is evaluated.

Validation Using Toy Problems

In the ‘‘Evaluation using industrial data’’ section, we compared different methods to solve the kiln drying operation scheduling problem. Although the LDS-Dynamic method gives the best known results, it is not possible, with actual large industrial instances, to determine how far the LDS-Dynamic method is from the optimality. To get a more precise idea, we have derived smaller toy problems from the industrial ones (the original industrial instances had 60 periods, while our reduced version has 29 periods), and we developed an MIP model to solve the whole problem as a single step (we call it Big MIP in Fig. 7). Having this MIP solve the whole problem (although it is intractable for the real industrial instances) results in a lower bound on the solution, which we use for comparison purposes.

Most of the constraints in the Big MIP are the same than the ones defined in the ‘‘Using a MIP model to define an optimal loading pattern’’ section. The main difference resides in the fact that the MIP defined in the ‘‘Using a MIP model to define an optimal loading pattern’’ section makes one loading plan for one kiln at one point in time, while the Big MIP makes multiple loading plans for each kiln in time. This can be achieved by adding an index for the loading to most of the decision variables. Moreover, new decision variables define the time at which a loading starts and the time at which a loading ends. Also, binary variables indicate whether or not a given loading starts/ends at a given period on a given kiln. New constraints are set so that no loadings overlap in time on any given kiln.

Figure 7 shows the gap evolution over time for the Big MIP and for the LDS-Dynamic method. Both methods were executed for 5 hours. The LDS-Dynamic method rapidly

Table 1.—Value of the objective function (minimization) for the greedy heuristics versus LDS tree search methods, with (fixed) or without (dynamic) preset loading patterns.

Method	Case 1		Case 2		Case 3		Case 4		Average
(1) Heuristic-Fixed	14,455,348		2,101,861		1,729,067		7,489,769		
(2) LDS-Fixed	13,741,655	↓ 5%	2,049,925	↓ 2%	956,644	↓45%	6,766,709	↓10%	↓15%
(3) Heuristic-Dynamic	12,678,085	↓12%	1,100,762	↓48%	544,041	↓69%	4,367,626	↓42%	↓43%
(4) LDS-Dynamic	11,830,206	↓18%	832,171	↓60%	346,896	↓80%	4,074,191	↓46%	↓51%

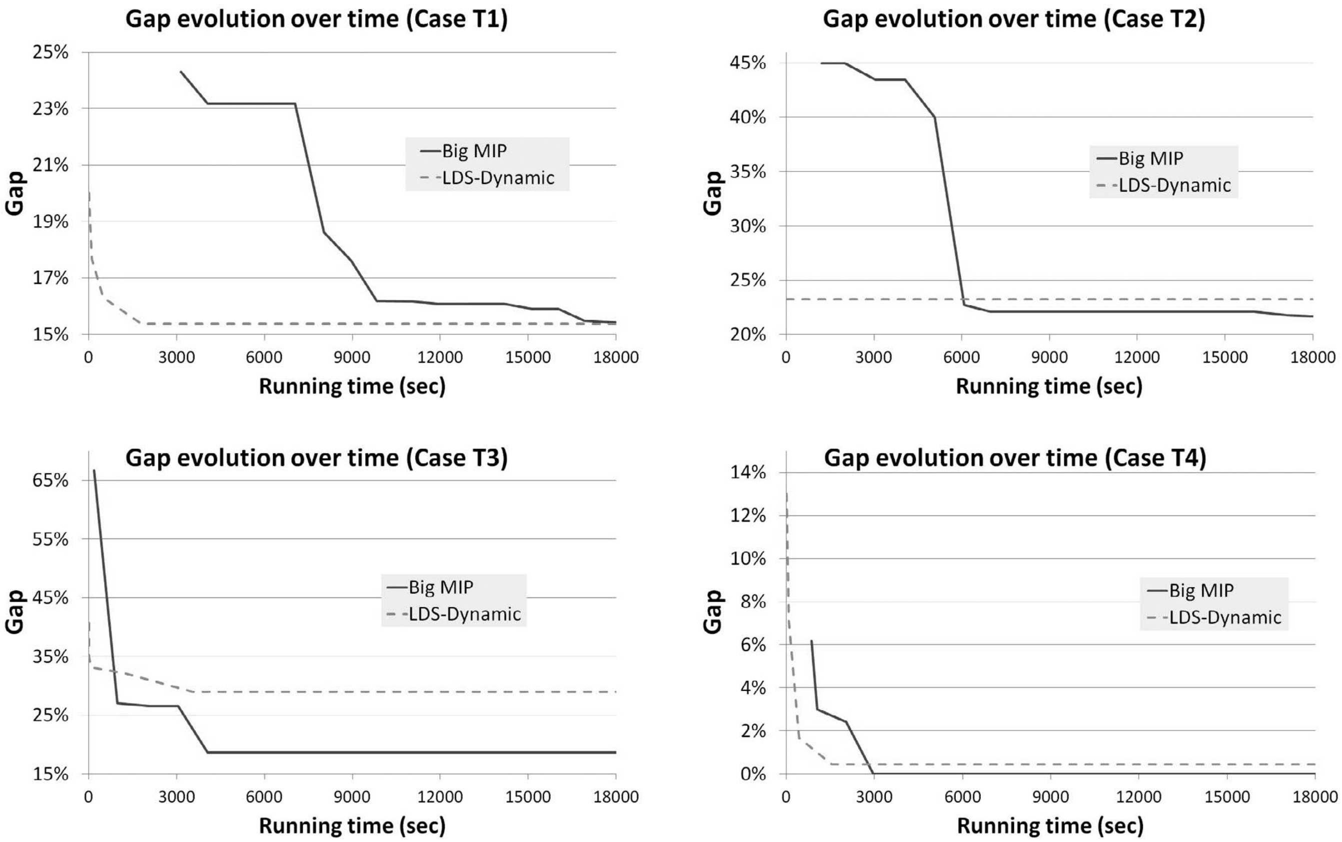


Figure 7.—Evolution of the gap for four toys problems using an exact approach (Big MIP model with CPLEX) versus the LDS-Dynamic approach.

finds good solutions and has difficulties improving the solution after 30 minutes of running time. The gap of the solutions found with our heuristic range between 0.5 and 29 percent (average 17%). For the Big MIP solved using CPLEX, the range for the gap is between 0 and 22 percent (average 14%).

We can see from these results that it takes more time for the Big MIP to converge. Although after 5 hours Big MIP obtained better solutions on average, the results are very close for three out of the four cases. We conclude that even though the solutions obtained are not optimal, we could not do much better in a reasonable amount of time using an exact algorithm solution. We have to mention again Big MIP cannot be used on real-sized instances.

Conclusion

Planning and scheduling of lumber kiln drying operations is a difficult problem that cannot be solved to optimality in a reasonable time. Various heuristics have been presented in the literature, but they either have to use predefined loading patterns or they apply to hardwood and do not address the loading patterns selection. With this project, we wanted to improve automated scheduling methods that apply to softwood lumber drying operations. The method proposed in this article allows the creation of new loading patterns dynamically when planning.

We have shown how the proposed MIP model that generates loading patterns can be integrated into any existing planning and scheduling methodology for kiln drying of softwood lumber. In this case, it was integrated to both a greedy heuristic and to an existing CP approach using an LDS search. We showed that the use of dynamic loading patterns in place of standard existing loading patterns is more effective when considering the objective of minimizing order tardiness. Using four real industrial cases, we computed an average improvement of 40 percent of order lateness when compared with a method using standard loading patterns as our industrial partners currently use, and compared with the same approaches using dynamic generation of loading patterns.

Manufacturers with whom we conducted our project validated that our loading patterns and kiln plans obtained for drying lumber are valid. For the personnel in charge of planning the kiln loading operations, the new approach is particularly useful when green lumber inventory is low, and it is difficult to fill up kilns using existing standard loading patterns. With short computation time, our method allows manufacturers to quickly make new plans according to variation in actual drying times, expected availability of green lumber, or urgency of new orders.

We have improved the quality of the solutions for a problem that was intractable by replacing it by an even more complex problem to model and solve. Moreover, from an industrial point of view, we have shown the forest products sector that planning kiln drying operations using a predefined set of loading patterns limits the quality of the solutions that can be obtained.

Our partner company has integrated our optimization methodology in its operation scheduling suite and is now using this tool to plan about 50 kilns in the provinces of

Quebec and Ontario in Canada. Their planning is based on current and planned short-term upcoming green lumber inventory level. One assumption of any kiln drying scheduling tool is that green lumber inventory is known. By the nature of the sawing process, a lot of uncertainty is involved, and having a good knowledge of available inventory for the coming weeks is quite difficult. This is why rescheduling of drying operations will have to be done on a regular basis, at least once a week.

Considering a specific objective (such as minimizing order lateness), having a computer-generated schedule with this aim will in most circumstances beat any human schedule. But in the end, this is not necessarily what makes a kiln drying scheduling tool strong and interesting for a company to use. Given the experience we have with our partner, we see three key elements for the adoption of such a tool by the industry: (1) ability to automatically connect to up-to-date data (inventory/orders/current planning); (2) ability to reschedule as needed with a few minutes optimization time; (3) ability to use or generate loading patterns that are easy and realistic for the operators. So although we might be closer to optimality considering order lateness minimization in only using dynamic loading patterns, the companies that implement this solution preferred to first use their standard loading patterns and only fall back to dynamic loading pattern when no standard pattern can be used given the planned available inventory at a specific time in the schedule.

References

Aggarwal, A. K., R. R. Vemuganti, and W. Fetner. 1992. A model-based decision support system for scheduling lumber drying operations. *Prod. Operations Manag*. 1(3):320–328.

Arman, R. Y., T. J. Hodgson, and J. A. Joines. 2001. Dry-or-buy decision support for dry kiln scheduling in furniture production. *IIE Trans*. 33:131–136.

Bartak, R. 1999a. Conceptual models for combined planning and scheduling. *In:* Proceedings of CP99 workshop on large scale combinatorial optimisation and constraints. Alexandria, VA, USA, October 16, pp. 2–14.

Bartak, R. 1999b. Constraint programming—What is behind? *In:* Department of Automation, Electronics and Computer Science (Wydawnictwo Pracowni Komputerowej). Proceedings of the constraint programming for decision control workshop. Gliwice, Poland, June 28-29, pp. 7–15.

Bekrar, A. and I. Kacem. 2009. An exact method for the 2D guillotine strip packing problem. *Adv. Operations Res*. 2009, Article ID 732010. DOI: 10.1155/2009/732010

Bischoff, E. E. and M. S. W. Ratcliff. 1995. Issues in the development of approaches to container loading. *Omega* 23(4):377–390.

Bortfeldt, A. and G. Wäscher. 2013. Constraints in container loading – a state-of-the-art review. *Eur. J. Operational Res*. 229(1):1–20.

Boschetti, M. A. and L. Montaletti. 2010. An exact algorithm for the two-dimensional strip-packing problem. *Operations Res*. 58(6):1774–1791. doi:10.1287/opre.1100.0833

Cheng Huang, J. C., C. T. Culbreth, J. A. Joines, R. E. King, and T. J. Hodgson. 1998. Decision support system for lumber procurement and dry kiln scheduling. *Forest Prod. J*. 48(9):51–59.

Côté, J. F., M. Dell'Amico, and M. Iori. 2013. Combinatorial benders' cuts for the strip packing problem. CIRRELT research document No. CIRRELT-2013-27, April 2013.

Dereli, T. and G. Sena Das. 2010. A hybrid simulated annealing algorithm for solving multi-objective container-loading problems. *Appl. Artif. Intelligence* 24(5):463–486.

Garcia, C. and G. Rabadi. 2013. Exact and approximate methods for parallel multiple-area spatial scheduling with release times. *OR Spectrum* 35:639–657.

Garcia, C. and G. Rabadi. 2016. Approximation algorithms for spatial scheduling. international series in operations research & management science, Chapter 1. *In:* Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling. Springer, New York. p. 236, DOI: 10.1007/978-3-319-26024-2_1

Gascon, A., P. Lefrançois, and L. Cloutier. 1998. Computer-assisted multi-item, multi-machine and multi-site scheduling in a hardwood flooring factory. *Computers Industry* 36:231–244.

Gaudreault, J., P. Forget, J. M. Frayret, A. Rousseau, S. Lemieux, and S. D'Amours. 2010. Distributed operations planning in the softwood lumber supply chain: Models and coordination. *Int. J. Ind. Eng.: Theory Appl. Practice* 17(3):168–189.

Gaudreault, J., J. M. Frayret, A. Rousseau, and S. D'Amours. 2011. Combined planning and scheduling in a divergent production system with co-production: A case study in the lumber industry. *Comput. Operations Res*. 38:1238–1250.

Gonzalez, Y., G. Miranda, and C. A. Leon. 2013. A multi-level filling heuristic for the multi-objective container loading problem. *In:* Herrero Á. et al. (Eds.) Proceedings of the International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, Salamanca, Spain, September 11-13. Advances in Intelligent Systems and Computing, vol 239, pp. 11–20. Springer, Cham. https://doi.org/10.1007/978-3-319-01854-6_2

Harvey, W. D., and M. L. Ginsberg. 1995. Limited discrepancy search. *In:* Proceedings of the International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canada. August 20-25. pp. 607–613.

Junqueira, L., R. Morabito, and D. S. Yamashita. 2013. MIP-based approaches for the container loading problem with multi-drop constraints. *Ann. Operation Res*. 199:51–75. DOI: 10.1007/s10479-011-0942-z.

Kenmochi, M., T. Imamichi, K. Nonobe, M. Yagiura, and H. Nagamochi. 2009. Exact algorithms for the two-dimensional strip packing problem with and without rotations. *Eur. J. Operational Res*. 198:73–83.

Maturana, S., E. Pizani, and J. Vera. 2010. Scheduling production for a sawmill: A comparison of a mathematical model versus a heuristic. *Comput. Ind. Eng*. 59:667–674.

Rossi, F., P. Van Beek, and T. Walsh. 2006. Handbook of Constraint Programming, section 6.4, Elsevier, Amsterdam.

Sorensen, T., S. Foged, J. M. Gravers, M. N. Janardhanan, P. Nielson, and O. Madsen. 2016. 3D pallet stacking with rigorous vertical stability. *In:* Proceedings of the 13th International Conference on Distributed Computing and Artificial Intelligence (DCAI'16), Sevilla, Spain, June 1-3. Advances in Intelligent Systems and Computing 474, pp. 535–543. Springer, Cham. https://doi.org/10.1007/978-3-319-40162-1_57

Yu, S., J. Yan, and S. Zhu. 2014. A research on hybrid simulated annealing algorithm for cargo loading problem. *Adv. Intelligent Syst. Comput*. 279:823–830. DOI: 10.1007/978-3-642-54927-4_78.

Zeineldin, R. A., and A. M. Morsy. 2015. A modified artificial bee colony for solving the container loading problem. *Int. J. Compu. Appl*. 114(3):19–24.