Kiln Drying Operation Scheduling with Dynamic Composition of Loading Patterns

Philippe Marier, Jonathan Gaudreault and Thomas Noguer,

FORAC Research Consortium, 1065, av. de la Médecine,
Université Laval, Québec, Canada G1V 0A6
{philippe.marier, jonathan.gaudreault}@forac.ulaval.ca

Abstract. Planning and scheduling wood drying operations is a very difficult problem. Literature proposes different methods aiming at minimising 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 utilisation. We proposed extensions to these methods in order for them to dynamically generate kiln loading patterns during planning/scheduling. The resulting system is an MIP/Constraint programming hybrid method that provided the best known solutions for this industrial problem.

Keywords: Optimisation, mathematical programming, kiln scheduling, lumber, drying.

1 Introduction

The problem of planning and scheduling softwood lumber drying operations is very complex. Drying is carried out by 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) which is geometrically stable and which fills the entire kiln. Because kilns are huge, each batch contains many products 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, most North American companies set in advance a number of standard loading patterns which are known to fill the kilns while providing good stack stability. These patterns are built mainly based on workers' experience. The task of the 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 minimising orders lateness. For example, Figure 1 shows a schedule for a period of two weeks, for three kilns.

For most companies, this planning is done manually. A mixed integer mathematical model has already been presented for this problem [5], but for instances of industrial size, the mathematical model rarely provided good solutions even after long computing time. A constraint programming model (CP) which obtains good solutions for very short computation times was also introduced in the same article. Letting the CP model run for a long time can in theory achieve the optimal solution, but this has never been possible for industrial-sized problems.

We were approached by a company that sought to determine whether allowing a greater number of kiln loading patterns than those normally allowed 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 orders. Thus, rather than using a list of predefined loading patterns, we would enable the optimisation model to dynamically generate stacks of lumber to load the dryer. Of course, the constraints of geometrical stability of the stacks has to be incorporated into the model. Previous work had shown the inability to solve the original problem optimally. In making the problem even more complex, we would be moving away from the idea of theoretical optimality, but generated solutions could be more interesting than those generated by the previously proposed systems.

The paper is organised as follows. We first present preliminary concepts related to the planning and scheduling of wood drying operations. We then briefly describe a mixed integer mathematical model (MIP) we propose in order to dynamically generate loading patterns. Finally, we present experiments showing how this new model greatly improved the quality of the solution found.

2 Preliminary concepts – wood drying

Bundles to be dried are generally prepared on carriages outside of the kilns. They are pushed on rails into the kiln at the time 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.

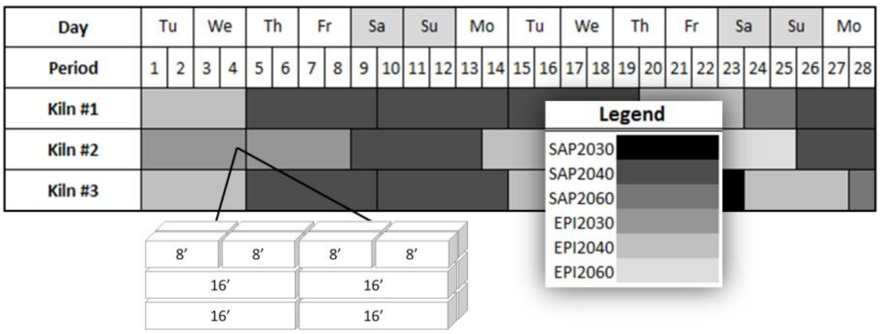

Fig. 1. Production plan for three kilns. Each operation identifies the selected drying process (gray) as well as a loading pattern indicating which specific compatible products will be dried.

Short-term planning of lumber drying operations consists in finding a plan showing, for the next three or four weeks, how each kiln should be used for drying. Thus, for each kiln, the plan shows which drying process to be applied, when (start time and end time), and which products compatible with the process to enter the kiln at that time. Figure 1 shows an example of a solution for the planning of three kilns over a period of two weeks (28 periods of 12 hours). 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 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 very large, enumerating the various possible product combinations to form loading patterns is unthinkable. For this reason, companies typically work with a small number of pre-defined loading patterns (a few dozen). However, it would be possible to dynamically build the kiln loading patterns at the time of planning. This is what we propose in Section 3.

For the detailed planning of softwood drying operations, Gaudreault et al. [4] 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:

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

Fig. 2. Greedy heuristic to solve the kiln drying operation scheduling problem, from [4].

In Gaudreault et al. [5], the same problem is modelled as a mixed integer programming problem (MIP) as well as a constraint programming model (CP). 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. Moreover, leaving the algorithm to run for a longer period of time can theoretically achieve the optimal solution (the algorithm is called complete) although for industrial size instances optimal solutions have not been obtained.

Interestingly, the search strategy employed by the CP model can be seen as a generalisation of the heuristics just presented above (Figure 2). In step 2, rather than choosing the best process/loading pattern (i.e. the one reducing the objective function by the greatest value) we sort all processes/patterns in descending order of preference. The combination of all these possibilities defines a search tree (Figure 3). The first leaf of the tree (on the left) corresponds to the solution that would be obtained with the heuristic. If the computation time allows for it, backtracking can be used to explore alternative branches with the hope of finding better solutions. Of course, the constraints of the CP model dynamically prune the tree. 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.

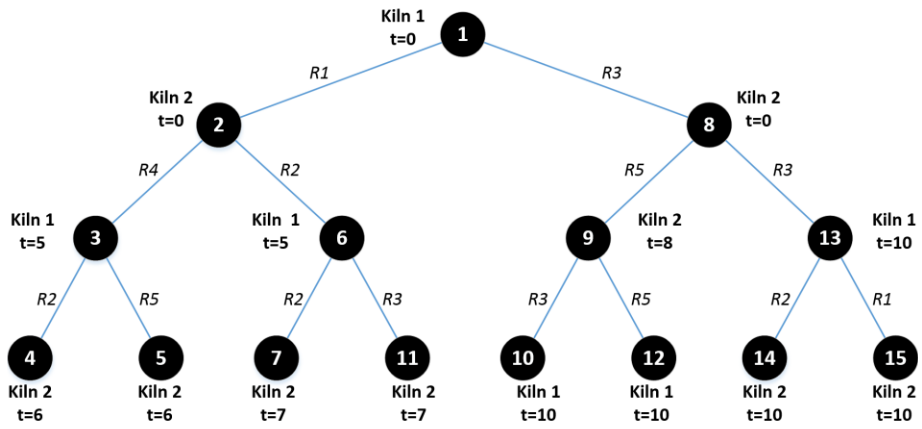

Fig. 3. Example of a search tree representing the solution space for the kiln drying planning problem. Each node corresponds to a choice point where you have to choose a drying process/loading pattern.

Gaudreault et al [5] have shown that the search strategy LDS (Limited Discrepancy Search [6]) in the tree was much more efficient than basic strategies (e.g. Depth-First Search).

Other authors propose partial solutions to this problem. Gascon et al. [1] worked on the development of an integrated system for the management of drying hardwood. This system includes a multi-site kilns planning module which uses a heuristic based on the concepts of control point and safety stock to minimise backorders. In their case, loading patterns issues do not apply because there are no different bundles sizes and there are few different products (species).

Agarwal et al. [2] also present a decision support system for planning kiln operations for a plant producing hardwood furniture. The planning model is an MIP which includes the ability to buy dry wood ready to be used when the dry wood needed for furniture production must be available at minimum cost. As in the case studied by Gascon et al., drying time is relatively long and can vary between 11 and 18 days depending on the time of the year, weather conditions and the time spent pre-drying. Because of the complexity of the problem, the authors had to use long periods in their model (5 days) and implement a version of the relaxed linear program. The problem of loading patterns is not present either in the case that they studied.

Arman et al. [3] also have developed a kiln drying planning model for hardwood used in furniture production. The opportunity to buy dried wood or to dry it is considered by the model whose objective is to satisfy the demand at the lowest cost. The authors mention that due to the 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.

3 Using an MIP model to define an optimal loading pattern

The approaches used to solve the kiln drying operation scheduling problem, whether manual planning, greedy heuristic [4], the MIP or CP model [5], all have a sub problem consisting in choosing which loading pattern (from a list of preset patterns) should be used to fill a specific kiln at a specific time. In this section, we propose the use of an MIP model that allows dynamically generating a loading pattern/optimal stacking.

For a particular kiln that is free at a specific time, the MIP model we developed identifies: (1) what drying process will be used; (2) for each rail, how many bundles of different lengths will be placed on each row; (3) what the height is of the bundles of each row of each rail of the kiln; (4) what product constitutes each bundle. The model takes into account the physical dimension 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 dried products (volume required in each period, per product). The objective of the model is to configure a valid stacking which minimise the order lateness.

Constraints only allow us to dry at the same time those products that have the same drying parameters. Other constraints also require that all bundles on the same row have the same height and that in each row of the same rail, there is 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 maximise the use of space, the sum of the lengths of bundles of one row must be greater than a specified minimum length.

The greedy heuristic from [4] (see Figure 2) may be adapted to make use of the proposed sub-model. In step 2, rather than choosing the best process/loading pattern from a predetermined list, we simply use the mathematical model to generate one.

The same goes for the constrained programming model proposed by Gaudreault et al. [5]. In their search tree, each node corresponds to moments where we need to select a drying process and a loading pattern to insert into the plan. This can be done using our MIP model. When a first solution to the overall problem is obtained (first leaf, bottom left) we can backtrack to a previously visited node to choose an alternative drying process/loading pattern. This is done not only by running the model again, but also by removing the ability of the model to choose for that node the same process as those previously selected for that node. 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 to use the LDS strategy since it is the one that gave the best results in [5].

4 Experiments using industrial data

We have shown that all effective approaches to the problem have a sub-problem which consists of choosing a loading pattern. In this section, we take these different methods, but we replace the step of selecting the loading pattern among those predefined, by a step which dynamically builds a loading pattern using the MIP model presented in the previous section.

The different approaches we are to compare are: (1) heuristic with a list of preset patterns [4] (fixed heuristic), (2) search space alternatives with LDS strategy [5] (fixed LDS), (3) greedy heuristic with dynamic loading patterns generation (dynamic heuristic), and (4) search space alternatives with LDS while creating loading patterns dynamically (dynamic LDS).

The models were tested with real data from a representative average size sawmill in Quebec (Canada) using data from four different periods (February-April 2005). In all four cases, 40 different products can be dried in any of the two identical kilns. We look for a plan that will minimise lateness over a horizon of 60 12-hour periods. 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%.

Figure 4 compares the results for different methods according to computing time. Recall that the heuristic solutions (1 and 3) correspond to the first solution found by the corresponding search (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.

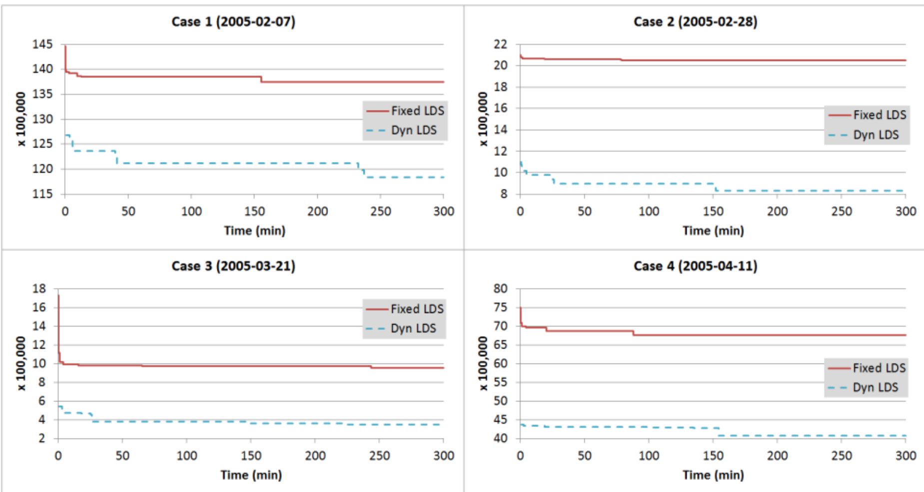

Fig. 4. Graphs of the reduction in lateness as a function of computation time for the various methods

Table 1 compares the results of the heuristic method and the search with LDS (after 300 minutes). The percentage is the lateness reduction compared with the reference method (fixed heuristic, i.e. with pre-established loading patterns). Comparing the two methods with predefined loading patterns (fixed heuristic VS fixed LDS) shows a result consistent with those obtained in Gaudreault et al. [4]. The search method with LDS improves the solution by 5% to 10 % in cases 1, 2 and 4 and by more than 45% in case 3. The graph in Figure 4 shows that the majority of the gain with the LDS method is obtained in the first few minutes of computation.

If we look at the impact of generating patterns dynamically, we notice that the heuristic (dynamic heuristic) is more effective than when it uses the predefined patterns (it allows reducing lateness by 43% on average) and it beats LDS (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 focussed on the need to minimise lateness, and more opportunities are explored than when using predefined loading patterns.

Using the LDS strategy to explore alternative solutions (dynamic LDS) we get the best solutions found to date for this problem. This approach allows an average gain of 51% compared to the heuristic exploiting predetermined fixed patterns, and 8% compared to the heuristic using dynamic loading patterns.

We should mention that the dynamic LDS method requires much more time to explore each node than the fixed LDS method. Hence, this method explored an average of 14 000 nodes over 5 hours while the fixed LDS method explored 4 000 000 nodes in the same period of time. However, nodes from the fixed LDS are not the same as the one from the dynamic LDS.

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

Method	Case 1 (2005-02-07)		Case 2 (2005-02-28)		Case 3 (2005-03-21)		Case 4 (2005-04-11)		Average
(1) Fixed heuristic	14 455 348		2 101 861		1 729 067		7 489 769		
(2) Fixed LDS	13 741 655	↓5%	2 049 925	↓2%	956 644	↓45%	6 766 709	↓10%	↓15%
(3) Dynamic heuristic	12 678 085	↓12%	1 100 762	↓48%	544 041	↓69%	4 367 626	↓42%	↓43%
(4) Dynamic LDS	11 830 206	↓18%	832 171	↓60%	346 896	↓80%	4 074 191	↓46%	↓51%

5 Validation using toy problems

In Section 4, we compared different methods to solve the kiln drying operation scheduling problem. Although the *Dynamic LDS* method gives the best known results, it is not possible, with these huge industrial instances, to determine how far the *Dynamic LDS* method is from the optimality. To get a more precise idea, we have derived smaller problems from the industrial ones and we developed an MIP model to solve the *whole problem* (we call it *Big MIP* in Table 2) closer to optimality. Having an MIP solving the whole problem (although it is intractable for the real industrial instances) allows getting a lower bound on the solution, which we will use for comparison purposes.

Table 2 synthesises the results. The first solutions of *Dynamic LDS* has an average GAP of 21.8% and the first solutions of Big MIP an average GAP of 35.5%. However, *Dynamic LDS* gets its first solution in 12 seconds on average compared to 1423 seconds for *Big MIP*. At that time, *Dynamic LDS* now reaches a GAP of 18.4% on average. It is only when we let both methods run for a long period of time (5 hours), that *Big MIP* surpasses *Dynamic LDS*. After 5 hours, the average GAP for *Dynamic LDS* is 16.9% in comparison to 13.9% for *Big MIP*.

Table 2. Indicators of the quality of the solution (toy problems).

		Case T1		Case T2		Case T3		Case T4		Average	
	Method	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap
When Dynamic LDS gets its first solution	Big MIP	-	-	-	-	-	-	-	-	-	-
	Dynamic LDS	2	17.7%	18	23.2%	11	33.4%	16	13.0%	12	21.8%
When Big MIP gets its first solution	Big MIP	3261	24.3%	1342	45.0%	225	66.7%	864	6.2%	1423	35.5%
	Dynamic LDS	3261	15.4%	1342	23.2%	225	33.4%	864	1.7%	1423	18.4%
After 5 hours	Big MIP	18000	15.4%	18000	21.7%	18000	18.6%	18000	0.0%	18000	13.9%
	Dynamic LDS	18000	15.4%	18000	23.2%	18000	28.8%	18000	0.4%	18000	16.9%

6 Conclusion

Planning and scheduling of lumber kiln drying operations is a difficult problem that is not possible to solve 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. The method proposed in this article allows the creation of new loading patterns dynamically when planning, which allows being more effective when considering the objective of minimising orders tardiness. Manufacturers with whom we conducted our project validated that our loading patterns and kiln plans obtained for drying lumber are valid. We are currently pursuing a project with a company for the integration of the planning and scheduling model to their ERP system.

References

1. Gascon, A., Lefrançois, P., Cloutier, L. Computer-assisted multi-item, multi-machine and multi-site scheduling in a hardwood flooring factory. Computers in Industry, vol. 36, pp. 231--244 (1998)
2. Aggarwal, A.K., Vemuganti, R.R., Fetner,W.: A model-based decision support system for scheduling lumber drying operations. Production and Operations Management, vol. 1, no. 3, pp. 320--328 (1992)
3. Arman, R.Y., Hodgson, T.J., Joines, J.A. Dry-or-buy decision support for dry kiln scheduling in furniture production. IIE Transactions, vol. 33, pp. 131--136 (2001)
4. Gaudreault, J., Forget, P., Frayret, JM., Rousseau, A., Lemieux, S., D'Amours, S. Distributed operations planning in the softwood lumber supply chain: Models and coordination. International Journal of Industrial Engineering : Theory Applications and Practice, vol. 17, no. 3, pp. 168--189 (2010)
5. Gaudreault, J., Frayret, JM., Rousseau, A., D’Amours, S. Combined planning and scheduling in a divergent production system with co-production: A case study in the lumber industry. Computers & Operations Research, vol. 38, pp. 1238--1250 (2011)
6. Harvey, W. D., Ginsberg, M.L. Limited discrepancy search. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 607--615 (1995)