A Preliminary Study to Tackle Sawmill Production Scheduling with ASP

Giacomo Da Col[1], Philipp Fleiss[1], Alice Tarzariol[2], Erich C. Teppan[1,2], and Elena Wiegelmann[1]

[1] University of Klagenfurt, Klagenfurt, Austria
[2] Fraunhofer Austria, Innovationszentrum für Digitalisierung und Künstliche Intelligenz (KI4LIFE), Lakeside B13a, 9020 Klagenfurt, Austria

1 Problem Description and ASP Modelling

The forest industry plays a major role in Austria's economy [1]. The scheduling process in the sawmill industry involves determining which logs should be directed to the sawing line to produce the necessary lumber to meet order deadlines. The sawmill use case modelled in this paper is based on the requirements of an industrial partner whose primary concern for production is the fulfilment of the orders in time, while also minimizing waste. Every day, a list of *demands* determines which products are needed. If the sawing line is not capable of meeting the demand for the day, the missing products are added to a *backlog* to be produced the following day. Ideally, the backlog of every day should be empty, therefore this is the first optimization criterion to be minimized. On the contrary, if the sawline produces more than what is demanded, the exceeding products are added to the *inventory*. The inventory products are not thrown away but are available to fulfil the demands of the following days. Since maintaining a large inventory increases operational costs, the second optimization criterion is to minimize it. Lastly, the cutting pattern for every log should be selected in order to minimize the number of logs needed to satisfy the demand. We took inspiration from one of the most cited papers in the field [6], and we wrote a model in ASP [2] to reflect the specifics of our industrial partner.

ASP Modelling and Benchmark. Each instance of the considered scheduling problem is defined through the following facts: (i) `availability(m,t,a)` stating that, at the period t, a units of material m are available to be processed; (ii) `yield(m,p,y)` stating that y units of product p can be obtained from processing the material m; (iii) `capacity(t,c)` defining the capacity, c, at the period t; (iv) `demand(p,t,d)` defining the demand of the product p is d units, at the period t; (v) `inventory(p,0,q)` defining the initial amount, q, of product p in the inventory; (vi) `backlog(p,0,q)` defining the initial amount, q, of product p in the backlog. To simulate different scenarios, we created a benchmark of instances consisting of five types of *demands* and five random seeds, for a total of 25 problem instances. For every instance, the scheduling *period* corresponds to one working day, resulting in five periods per week; the *raw materials* (i.e., logs) available for each period corresponds to five different sizes of spruce, the most common variety of wood in Austria [1], the raw availability is 50 logs per size on every period; the *capacity* (i.e., the maximal amount of material that can be processed per period) is 200 logs; and lastly, we used a timber-sawing professional software[5] to identify five cutting patterns on the considered logs to *yield* boards of various dimensions (i.e., products). We assessed the performance of four different ASP encodings using the standard solving strategy for CLINGO [4], where the one obtaining the best results for the benchmark is the following:

```
1  0 {processed(M, T, 1..A): availability(M,T,A)} C :- capacity(T,C).
2  processed(M, T, N-1) :- processed(M, T, N), N>1.
3  result(P,0,I)  :- inventory(P,0,I), I>=0.
4  result(P,0,-I) :- backlog(P,0,I), I>0.
5  result(P,T,R1) :- demand(P,T,D),  result(P,T-1,R), T>0,
6    S=#sum{Y,M,X : processed(M,T,X), yield(M,P,Y)}, R1=S-D+R.
7  inventory_lin(P,T,1..I) :- result(P,T,I), I>1.
8  backlog_lin(P,T,1..-B)  :- result(P,T,B), B<-1.
9  :~ backlog_lin(P,T,B).   [1@3,P,T,B]
10 :~ inventory_lin(P,T,I). [1@2,P,T,I]
11 :~ processed(M,T,N).     [1@1,M,T,N]
12 upperbound(-X) :- X = #max{D : demand(P,T,D)}.
13 :- backlog_lin(P,T,X), upperbound(X).
```

Following the model proposed in [6], we define the schedule according to the availability of each period. The choice rule in line 1 generates the candidate solutions containing at most c processed material at the period t. When an atom $processed(m, t, q)$ is guessed to be true for an interpretation, then the rule in line 2 requires that all the atoms $processed(m, t, i)$ for $i \in [1..q]$ must also be included in it. As the backlog and inventory of a product are mutually exclusive, we can project this information in the predicate $result(p, t, q)$, representing the quantity q of product p in excess (when $q > 0$) or in shortfall (when $q < 0$) at time t. From lines 9 to 11, we defined the optimization criteria for an optimal solution, where the weak constraint with the highest priority uses the auxiliary predicate *backlog_lin* to minimize the amount in the backlog in every period. In line 10, the constraint with the subsequent priority level uses the auxiliary predicate *inventory_lin* to minimize the amount of each product in the inventory in every period, while lastly, line 11 contains the constraint with the lowest priority, minimizing the number of materials processed. The constraint in line 13 was introduced to add bias observed for the instances of our benchmark in order to cut the search space and guide the ASP solver towards better solutions faster. The idea is to prevent accumulating backlog, forcing the avoidance of solutions exceeding the maximal demand of the instance. By including the bias and scaling down the benchmark by a factor of ten, CLINGO obtained promising results, finding solutions within a timeout of 30 minutes for all the 25 instances, and being able to find and prove optimality in seven of them. The ASP model in this paper represents an initial prototype for compactly representing the sawmill production scheduling, and, in future work, we aim to investigate further domain knowledge and ASP extensions (e.g., multi-shot [3]) to improve the solving performance.

References

[1] Anna-Maria Walli Christian Lackner Marianne Schreck. *Austrian Forest Report 2023*. Tech. rep. Stuben1, 1010 Vienna, Austria: Federal Ministry of Agriculture, Forestry, Regions and Water Management, 2023.

[2] M. Gebser et al. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[3] Martin Gebser et al. “Multi-shot ASP solving with clingo”. In: *Theory and Practice of Logic Programming* 19.1 (2019), pp. 27–82.

[4] Martin Gebser et al. “Potassco: The Potsdam answer set solving collection”. In: *Ai Communications* 24.2 (2011), pp. 107–124.

[5] Tekl Studio LTD. *Cutlog*. `https://www.cutlog.com/`. Accessed: 2024-10-11.

[6] Sergio Maturana, Enzo Pizani, and Jorge Vera. “Scheduling production for a sawmill: A comparison of a mathematical model versus a heuristic”. In: *Computers & Industrial Engineering* 59.4 (2010), pp. 667–674.