# A Hybrid Approach for the Artificial Teeth Scheduling Problem

Felix Winter and Nysret Musliu

Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, DBAI, TU Wien, Favoritenstraße 9, 1040 Vienna, Austria
{felix.winter,nysret.musliu}@tuwien.ac.at

**Keywords:** Artificial Teeth Scheduling Problem, Hybrid Approach, Metaheuristics, Constraint Programming

## 1 Introduction

Modern-day manufacturing of artificial teeth relies on a highly automated production process that utilizes complex machine environments. Therefore, a large number of teeth products are manufactured daily to fulfill orders from customers all around the world. Due to these large-scale requirements, efficient automated production scheduling methods are required to minimize costs and consider all the constraints arising in the complex machine environment.

Previously, we introduced the artificial teeth scheduling problem (ATSP) originating from the industry in [4]. In addition to a formal specification of the problem, we provided an exact constraint-modeling approach to solve the problem. Further, we proposed a simulated annealing approach to tackle large-scale instances from the industry. An experimental evaluation on real-life instances showed that the exact approach can find optimal cost results for some small instances. However, the heuristic method was required to provide solutions for large instances within a reasonable time.

In [5], we further proposed a set of low-level heuristic operators that can be utilized with hyper-heuristic approaches. Experiments showed that hyper-heuristics could efficiently solve realistic instances and can improve results over the previous heuristic results in many cases. The existing heuristic approaches can produce high-quality solutions for practical instances. However, optimal solutions are still unknown for all large real-life instances.

This work proposes a novel hybrid approach for the ATSP that combines exact constraint-modeling techniques with a heuristic approach. In particular, the proposed technique decomposes the problem into two phases. In the first phase, optimized job patterns are determined for a given instance using an exact approach based on constraint-modeling. Afterwards, the optimized patterns are used to build an initial job sequence as a starting point for the second phase, where a heuristic further optimizes the solution.

Solving machine scheduling problems in two phases has been successfully applied in the past (e.g., [6,3]). However, existing decomposition methods cannot directly be applied to solve the ATSP, as they assume that jobs are part of the input.

## 2 A 2-Phase Approach for Artificial Teeth Scheduling

The ATSP can be viewed as a complex single-machine scheduling problem originating from industrial teeth manufacturing. However, in contrast to traditional machine scheduling problems, the jobs that need to be scheduled are not given as input to the problem. Instead, problem instances specify demands for various teeth products, and a part of the decision-making is to group demanded teeth products into jobs. Several complex constraints impose restrictions on how products can be grouped. Thus, in practical instances, it is often required to overproduce certain products to fulfill job capacity constraints.

The aim of the ATSP is to find schedules that minimize three objective criteria: Makespan, Waste, and Tardiness. While the first two objectives are minimized by finding efficient jobs that have a short duration and keep overproduction as low as possible, job tardiness is mainly influenced by the sequence of the jobs. For the publicly available real-life instances, all three objectives are weighted uniformly. A comprehensive problem specification of the ATSP can be found in [4].

The existing exact- and heuristic approaches solve the problem in a single phase, i.e., the problem is modeled to simultaneously consider decisions on creating the jobs and the scheduling aspect. In this work, we propose to decompose the problem in two phases. In the first phase, the method aims to create a set of jobs for all given product demands to minimize total job duration. Afterwards, the approach focuses on job scheduling in phase 2. Note that the 2-phase solution approach is incomplete, even if we reach optimal results in both phases, as the first phase entirely neglects the tardiness objective. However, as finding high-quality solutions for large-scale instances is challenging, the decomposition approach can potentially find improved upper bounds compared to existing techniques.

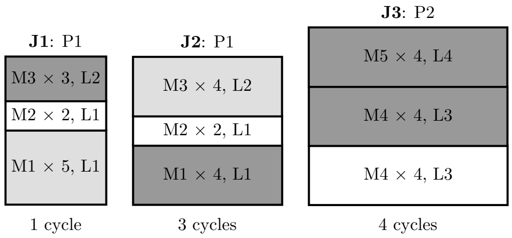
Figure 1 illustrates solutions to phases 1 and 2. In Figure 1a we can see three jobs (J1-J3), each including various different product configurations (Production molds M1-M5, different colors, production lines L1-L4, and production programs P1-P2). Each job in the example also uses a different number of production cycles which directly determine the job's length. The second phase schedules the jobs that were created in phase 1. In the example in Figure 1b, the jobs have been scheduled one after the other, although a different sequence would have been possible. All three jobs have precise start and ending time points ($t1 - t6$), and the arrows between the jobs visualize the setup time length between jobs.

## 3 Solution Method & Preliminary Experimental Results

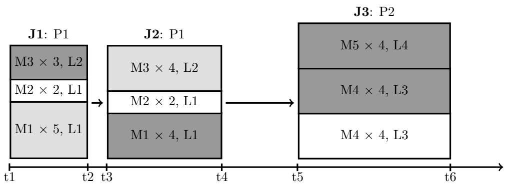
For the first phase, we have implemented a constraint model using the high-level modeling language MiniZinc [1], which can be used with constraint programming and mixed integer programming solvers as an exact approach. For the first set of preliminary experiments, we used the solver CP-SAT [2] in version 9.8.

To solve the second phase, we use a variant of the local search-based simulated annealing approach from [4], that only activates the job swap neighborhood. Thus, the heuristic focuses on optimizing the job sequence without modifying the jobs.

We conducted experiments with the proposed approach using the benchmark instances from [5]. Instances 1-6 include small scenarios, whereas instances 7-20 consist




(a) An example solution for the first phase consists of three jobs with different durations (i.e., production cycles).




(b) An example solution of the second phase consists of the three jobs created in phase 1 (See Figure 1a).

Fig. 1: Visual representation of example solutions for the 2-phase approach.

of large-scale real-life instances. The experimental environment was similar to the one used in [5] using a time limit of 1 hour. Note that we dedicated the most time to the exact solver and left the heuristic only the last 10 seconds of the time limit to find an optimized job sequence. We chose these time restrictions, as finding efficient jobs in phase 1 is particularly challenging. The exact technique could find optimal solutions only for a few small instances in this phase.

Table 1 gives an overview of the cost results achieved by existing methods and the proposed approach. Columns 2 (LB) & 3 (Exact) display the best lower bounds and upper bounds of the cost results achieved with exact techniques for all instances in [4]. Column 4 (SA) shows the best upper bounds achieved with the simulated annealing approach from [4], whereas Column 5 (HH) shows upper bounds achieved by the hyper-heuristic approach from [5]. Finally, Column 6 (Hybrid) displays the results of the hybrid approach proposed in this work (i.e., the final objective cost results after phase 2). Best cost results are formatted in boldface. A - indicates no solution was achieved within the time limit.

The results show that the proposed 2-phase hybrid approach cannot produce competitive results compared to existing techniques on small instances. In these cases, existing methods can likely reach solutions not explored by the decomposition approach. However, the novel approach produces improved results for most real-life instances, finding new upper bounds in 12 cases. These results indicate that the proposed hybrid approach can be a promising technique, especially for large-scale instances that lead to a vast search space for existing methods.

| Instance | LB | Exact | SA | HH | Hybrid |
|---|---|---|---|---|---|
| **Instance 1** | 2.08 | **2.53** | **2.53** | **2.53** | 2.54 |
| **Instance 2** | 1.25 | 1.96 | 1.96 | **1.94** | 2.04 |
| **Instance 3** | 2.23 | **2.23** | **2.23** | **2.23** | 2.24 |
| **Instance 4** | 2.54 | **2.54** | **2.54** | **2.54** | **2.54** |
| **Instance 5** | 1.63 | **2.1** | 2.13 | 2.12 | 2.20 |
| **Instance 6** | 3 | **3** | **3** | **3** | **3** |
| **Instance 7** | 0.5 | - | 2.95 | 2.85 | **2.72** |
| **Instance 8** | 0.15 | - | 2.38 | 2.47 | **1.77** |
| **Instance 9** | 0.59 | - | 2.99 | 2.85 | **2.43** |
| **Instance 10** | 0.53 | - | 2.67 | 2.66 | **2.02** |
| **Instance 11** | 0.34 | - | 2.76 | 2.78 | **2.43** |
| **Instance 12** | 1.02 | - | 2.97 | 2.91 | **2.53** |
| **Instance 13** | 0.6 | - | 2.97 | 2.85 | **2.57** |
| **Instance 14** | 0.46 | - | 2.99 | 2.84 | **2.65** |
| **Instance 15** | 0.56 | - | 2.99 | 2.81 | **2.06** |
| **Instance 16** | 0.37 | - | 2.98 | **2.67** | 2.76 |
| **Instance 17** | 0.2 | - | 2.94 | **2.79** | 2.89 |
| **Instance 18** | 0.39 | - | 2.98 | 2.72 | **2.22** |
| **Instance 19** | 0.18 | - | 2.94 | 2.7 | **2.14** |
| **Instance 20** | 0.18 | - | 2.98 | 2.78 | **2.37** |

Table 1: Cost results achieved by existing methods and the proposed approach.

**Acknowledgements** The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association is gratefully acknowledged.

## References

1. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: Minizinc: Towards a standard CP modelling language. In: CP. Lecture Notes in Computer Science, vol. 4741, pp. 529–543. Springer (2007)
2. Perron, L., Didier, F.: Cp-sat, https://developers.google.com/optimization/cp/cp_solver/
3. Tang, T.Y., Beck, J.C.: CP and Hybrid Models for Two-Stage Batching and Scheduling. In: Integration of Constraint Programming, Artificial Intelligence, and Operations Research. pp. 431–446. Lecture Notes in Computer Science (2020)
4. Winter, F., Mrkvicka, C., Musliu, N., Preininger, J.: Automated Production Scheduling for Artificial Teeth Manufacturing. Proceedings of the International Conference on Automated Planning and Scheduling **31**, 500–508 (May 2021)
5. Winter, F., Musliu, N.: An investigation of hyper-heuristic approaches for teeth scheduling. In: MIC. Lecture Notes in Computer Science, vol. 13838, pp. 274–289. Springer (2022)
6. Zhao, Z., Liu, S., Zhou, M., Guo, X., Qi, L.: Decomposition Method for New Single-Machine Scheduling Problems From Steel Production Systems. IEEE Transactions on Automation Science and Engineering **17**(3), 1376–1387 (Jul 2020)