TU WIEN Informatics

dbai Databases and Artificial Intelligence Group

Solving the Test Laboratory Scheduling Problem with Variable Task Grouping

Philipp Danzinger, Tobias Geibinger, Florian Mischek, Nysret Musliu

Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling

The Test Laboratory Scheduling Problem (TLSP)

Setting

In an industrial test laboratory, a large number of diverse tests from multiple projects have to be scheduled. Tests can be performed in different modes, and require time, qualified personnel and specialized equipment. Schedules have to fulfill several legal, structural and operational constraints.
Among those are restrictions on which units of a resource can be assigned to any particular task, due to qualification or equipment specifications. Further, projects have release dates and deadlines as well as internal precedence constraints between tasks. These and other constraints must be satisfied in all feasible schedules.
The quality of a schedule is judged via a combination of business goals, as well as measures aimed at reducing overhead and introducing robustness against changes.

Task Grouping

In TLSP, tasks are not scheduled directly, but instead have to be grouped into larger units, called *jobs*. These jobs derive their properties (e.g. duration, resource requirements, precedence,...) from the tasks they contain. Only jobs have a mode, timeslots, and resources assigned to them.

Motivation:

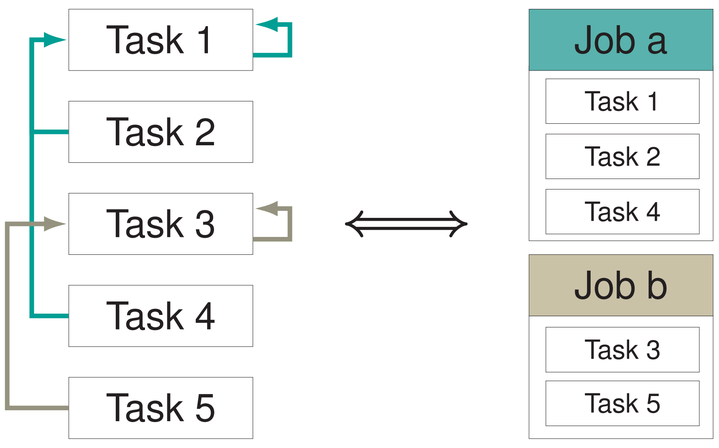
- Test assemblies can be reused between similar tasks
- Tasks have widely different durations, down to small fractions of timeslots
- Batching of tasks reduces overhead and complexity

Job j, containing tasks A_j

- Resource requirements of j: $r_j^{\mathcal{R}} = \max_{a \in A_j} r_a^{\mathcal{R}}$
- Available resource units: $\mathcal{R}_j = \bigcap_{a \in A_j} \mathcal{R}_a$
- ...

Constraint Programming Model

- Grouping tasks amounts to forming partitions
- Few restrictions, most tasks of a family could appear in the same job
- Goals: few decision variables, respect symmetries
- Each job is identified by a representative task
- Each task is treated as a potential job
- Tasks point at their job's representative task
- Symmetry breaking: representative task must be the smallest task in a job



Decision variables, for each task t:

$\mathtt{repr}(t) \in \mathtt{Tasks}$	representative pointed at by t
$s_t \in \{1, \ldots, T\}$	starting time slot
$m_t \in M$	assigned mode
$a_{rt}^{\mathcal{R}} \in \{0, 1\}$	whether resource unit r of type $\mathcal{R}$ was assigned

Time, mode and resource assignments are set to 0 for tasks that are not job representatives.

Optimizations

Valuable optimizations include:

- The smallest task of each family must point to itself
- Include the number of jobs in search, starting with fewer jobs
- Combine functionally identical resources into equivalence classes

Example Constraint: Resource assignment

The CP model is considerably more complex than our previous model for TLSP-S [1]. TLSP-S is a variant of TLSP where the grouping of jobs into tasks is fixed and part of the input.

$$\sum_{e \in Eq} a_{ej}^{Eq} = r_j \quad \forall j$$

TLSP-S

$$\sum_{e \in Eq} a_{et}^{Eq} = \max_{t_2 \text{ s.t. } \mathtt{repr}(t_2) = t} r_{t_2} \quad \forall t \text{ s.t. } \mathtt{repr}(t) = t$$

TLSP

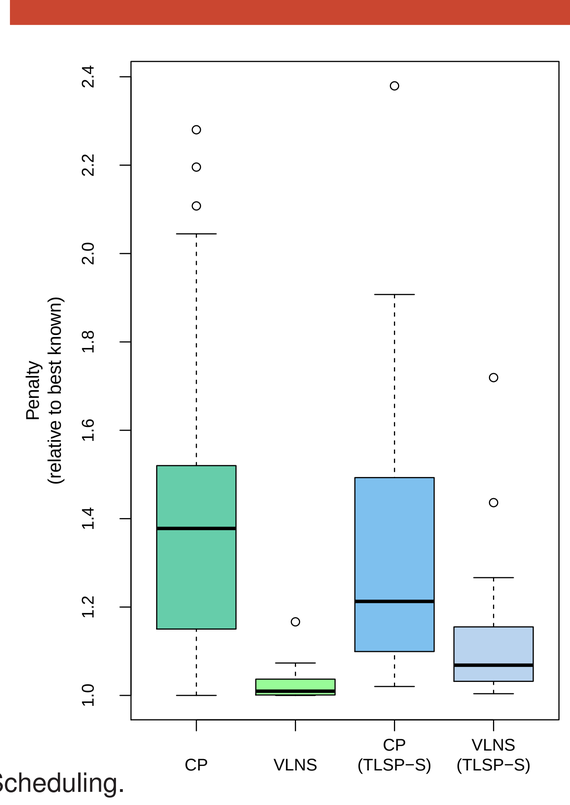
The resource assignment constraint is exemplary of how the model differs from the TLSP-S model. Task grouping requires dynamically adapting the scheduling requirements and increases conceptual and computational complexity.

Very Large Neighborhood Search

- Hybrid algorithm
- Repeatedly solve small subproblems using the CP model
  - Fix all but k projects to current values
  - Tabu list to avoid redundant work
  - Increase k if no improvement is possible
- Adapted from VLNS for TLSP-S [1]
- Initial solution produced by finding any feasible schedule with CP
- Choose between CP models for TLSP and TLSP-S at each step with a given probability

[1] Tobias Geibinger, Florian Mischek, and Nysret Musliu. Investigating Constraint Programming for Real World Industrial Test Laboratory Scheduling. In *CPAIOR*, pages 304–319, 2019.

Computational results



- The CP model with grouping is significantly more complex then the one for TLSP-S
  - Modeled in MiniZinc, solved by Chuffed
  - Better solutions than fixed grouping for instances with up to 10 projects (∼40 jobs)
  - Optimal solutions possible for up to 5 projects (∼20 jobs, ∼50 tasks)
  - Feasible solutions found for $30/33$ instances
- Incorporating both CP models into VLNS yields best known results for TLSP
  - Despite the advantage of known good grouping for TLSP-S
  - Finding an initial schedule quickly is crucial for good results
- Successfully deployed in industrial test laboratory

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.

Contact: pdanzing@dbai.tuwien.ac.at