Chalmers University of Technology
University of Gothenburg
Mathematical Sciences
Ann-Brith Strömberg

TMA521
MMA511
Large-scale optimization
February 8, 2018

Project 2: Production scheduling by column generation

1 The project task

The project task is to define and implement a decomposition algorithm for the scheduling problem described below using the *time indexed model*. The algorithm that should be used is a *Dantzig-Wolfe reformulation solved by column generation*.

1.1 Problem files and implementation

At your service, an AMPL model file (.mod) for the time-indexed mathematical machining models is provided on the course homepage together with a script file (.run) and a set of data files (.dat). These files can be used to solve the problem instances by the Cplex solver, to get key solutions for comparisons.

The decomposition algorithm chosen should be implemented in AMPL or Matlab (or C/C++ if you prefer). The sub and master problems may be solved by Cplex MILP solver. Instructions for writing scripts in AMPL are found at, e.g., `http://www.ampl.com/NEW/LOOP1` and `http://www.ampl.com/NEW/LOOP2`. Instructions for the AMPL and Matlab interfaces to Cplex are found at, e.g., `http://www.math.chalmers.se/Math/Grundutb/CTH/mve165/1516/Assignments/LP_Exc_1603.pdf`.

To access the currently installed version of AMPL at Chalmers Linux system, you must type (in a command window): `vcs-select -p ampl-20171212`

At your service are also time limited student licences for AMPL and several MILP solvers to download from PingPong and install on you own computer.

1.2 Deadlines and examination

Your report should be submitted in PingPong (and emailed to your opponents) on Tuesday, 6th of March. The program codes should be submitted in PingPong on Thursday, 8th of March (afternoon; see the course plan). After that the "competition instance and tasks" will be released and you will have a few hours to compute as good solutions as possible to these instances.

The examination of the project includes an oral presentation of the same and an opposition to another groups report, during the seminar on Friday, 9th of March, when the competition will also take place (the performance of your algorithm on the competition instances will not be a basis for the examination).

1.3 Applying the algorithm to the models

The machining problem (see Section 4) should be solved by column generation while the feasibility problem may be solved by a MILP-solver, without decomposition.

In the column generation algorithm applied to the machining problem, a column is suitably defined by one schedule (composed by a sequence of operations) for each machine. When a final set of columns has been generated (what “final” means is your decision), the corresponding restricted master problem, with integer requirements on the appropriate variables, may be solved by Cplex MILP-solver.

If any of these instructions or the AMPL-files are unclear or seem unsuitable, please inform me as well as your fellow students. Note that there may be misprints in the .mod-file, such as the very last “-1” in (5j) being missing. You are also welcome to discuss any difficulty or indistinct instruction in this assignment with me, but don’t forget to discuss also with your fellow students.

1.4 Presentation of results

The types of results to be presented in the report and at the seminars include upper and lower bounds on the optimal value as functions of number of iterations as well as of CPU-time, the best solution (i.e., schedule) found, the value of the best solution found compared with the key solution (found using the AMPL-files supplied).

In order to receive solutions (schedules) with different properties, different objective functions should be formulated and tried out in the computations.

The specific competition task will be revealed on the 8th of March (late afternoon).

2 Definition of the problem

2.1 Indices and sets

The queue of jobs j to the multitask (MT) cell go through three different phases:

- Planned orders not yet released, i.e., existing only in the planning system.
- Released jobs, or so called production orders, i.e., physical parts being processed elsewhere on their way to the MT cell.
- Jobs checked in into the MT cell, i.e., parts inside the MT cell waiting to be processed.

$\mathcal{J}$ denotes the whole set of jobs to be done during the planning period. Some jobs are to be processed on the same part, and the pairs of two such jobs adjacent in the routing form the set $\mathcal{Q} \subset \mathcal{J} \times \mathcal{J}$. For the part, the routing of which is illustrated in Figure 1, the pairs (j,q) and (q,l) belong to the set $\mathcal{Q}$.

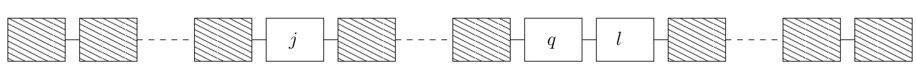

Figure 1: *Routing of a part with jobs j, q, and l. Here, $(j,q),(q,l) \in \mathcal{Q}$.*

Each job j consists of n_j operations $i \in \mathcal{N}_j = \{1,\ldots,n_j\}$ to be processed inside the MT cell. A typical routing for a job is listed in Table 1

In order to fix the order in the schedule between two jobs of the same type for the same type of parts, the set $\mathcal{P} \subset \mathcal{J} \times \mathcal{J}$ is populated by pairs (j,q) of these jobs, for which the release date of job j is less than or equal to the release date of job q. The set $\mathcal{K} := \{1,\ldots,10\}$ denotes the resources $k \in \mathcal{K}$ in the MT cell.

i	Description
1	Mounting into a fixture
2	Turning/Milling/Drilling
3	Manual deburring
4	Automatic deburring
5	Demounting

Table 1: The different route operations i in the MT cell.

k	Description
MC 1–5	Multitask machines
Man Gr	Manual deburring station
DBR	Automatic deburring machine
M/DM 1–3	Mount/demount stations

Table 2: The resources k of the MT cell

2.2 Parameters

$$\lambda_{ijk} = \begin{cases} 1, & \text{if operation } (i,j) \text{ can be processed in resource } k, \\ 0, & \text{otherwise.} \end{cases}$$

a_k : the time when resource k will be available the first time.

d_j : the due date of job j, i.e., the point in time when the last operation n_j of job j is planned to be completed.

r_j : the release date of job j.

p_{ij} : the processing time for operation i of job j.

w : the transportation time for a product inside the MT cell.

v_{jq} : the interoperation time between the jobs j and q, where $(j,q) \in \mathcal{Q}$.

M : a sufficiently large positive number (larger than the planning horizon).

All dates described above are given in hours relative to a time point, t_0, which is the starting time of the schedule to be calculated.

2.3 Realistic release dates and interoperation times

If a job j is checked in into the MT cell, i.e., the part is ready to be processed at time t_0, then r_j is set to 0. Release dates for the other two phases, i.e., released jobs and planned orders (see Section 2.1) are not that easy to get hold on. In the planning system of the MT cell, there are a planned latest release date for each job, denoted ϱ_j. This means that the job j in the MT cell is planned to be started at the latest at the time ϱ_j. The desired release date, r_j, is, however, the realistic point in time when the part arrives at the MT cell. A good guess is given by

$$r_j := \max\left\{\varrho_j - t_0 - 0.8\vartheta_j; \nu_j^0\right\},$$

where ν_j^0 is the standard lead time from the operation where the part is about to be processed at time t_0 until it arrives at the MT cell. Let μ_{act} denote this actual operation. Then, the standard lead time is given by

$$\nu_j^0 := \sum_{\mu=\mu_{\mathrm{act}+1}}^{\mu_{m_j-1}} (\rho_\mu + \varsigma_\mu + \vartheta_\mu) + 0.2\vartheta_j, \tag{1}$$

where ρ_μ, ς_μ, and ϑ_μ denote the process, setup, and queue times of operation μ, which is processed elsewhere, i.e., not in the MT cell; see Figure 2.

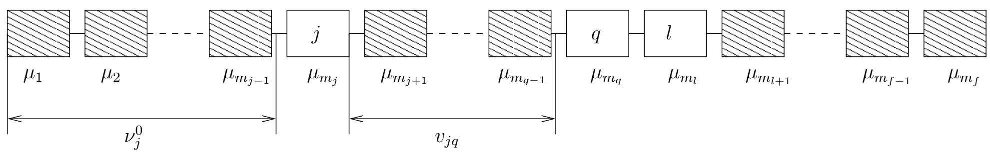

Figure 2: *The interoperation time, v_{jq}, and the standard lead time, ν_j^0, are denoted, for the case of a planned order, i.e., when $\mu_{\mathrm{act}} = \mu_1$.*

In order to prevent the jobs j and q from being scheduled too close in time— when these jobs are to be performed on the same physical part— the parameter v_{jq} is introduced, representing the planned interoperation time (for jobs done outside the MT cell) from the completion of job j to the start of job q. It is defined as

$$v_{jq} := \sum_{\mu=\mu_{m_j+1}}^{\mu_{m_q-1}} (\rho_\mu + \varsigma_\mu + \vartheta_\mu) + 0.2\vartheta_q,$$

where ρ_μ, ς_μ, and ϑ_μ denote the process, setup, and queue times of operation μ, which is processed elsewhere (cf. (1)).

3 The engineer's mathematical model

3.1 Variables

$$z_{ijk} = \begin{cases} 1, & \text{if operation } (i,j) \text{ is allocated to resource } k, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{ijpqk} = \begin{cases} 1, & \text{if operation } (i,j) \text{ is being processed before operation } (p,q) \text{ on resource } k, \\ 0, & \text{otherwise.} \end{cases}$$

$t_{ij} =$ the starting time of operation (i,j).

$s_j = t_{n_j,j} + p_{n_j,j}$, the completion time of job j.

$$h_j = \begin{cases} s_j - d_j, & \text{if } s_j > d_j, \text{ i.e., the tardiness of job } j, \\ 0, & \text{otherwise.} \end{cases}$$

3.2 Objective functions

The objective is to minimize a weighted sum of the job finishing times and tardiness, and the time used in the fixure (the weights fulfilling $A_j > 0$, $B_j > 0$, and $\varepsilon \in [0,1)$):

$$\sum_{j \in \mathcal{J}} (A_j s_j - \varepsilon t_{1j} + B_j h_j).$$

3.3 The optimization model

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j\in\mathcal{J}} (A_j s_j + B_j h_j), && && \text{(2a)}\\
\text{subject to} \quad & \sum_{k\in\mathcal{K}} z_{ijk} = 1, && i\in\mathcal{N}_j,\ j\in\mathcal{J}, && \text{(2b)}\\
& z_{ijk} \le \lambda_{ijk}, && i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(2c)}\\
& y_{ijpqk} + y_{pqijk} \le z_{ijk}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(2d)}\\
& z_{ijk} + z_{pqk} - y_{ijpqk} - y_{pqijk} \le 1, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(2e)}\\
& t_{ij} + p_{ij} - M(1-y_{ijpqk}) \le t_{pq}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(2f)}\\
& t_{ij} + p_{ij} + w \le t_{i+1,j}, && i\in\mathcal{N}_j\setminus\{n_j\},\ j\in\mathcal{J}, && \text{(2g)}\\
& t_{1j} \ge r_j, && j\in\mathcal{J}, && \text{(2h)}\\
& t_{ij} - a_k z_{ijk} \ge 0, && j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(2i)}\\
& t_{1q} - s_j \ge v_{jq}, && (j,q)\in\mathcal{Q}, && \text{(2j)}\\
& s_j - t_{n_j j} = p_{n_j j}, && j\in\mathcal{J}, && \text{(2k)}\\
& s_j - h_j \le d_j, && j\in\mathcal{J}, && \text{(2l)}\\
& h_j \ge 0, && j\in\mathcal{J}, && \text{(2m)}\\
& t_{ij} \ge 0, && i\in\mathcal{N}_j,\ j\in\mathcal{J}, && \text{(2n)}\\
& z_{ijk} \in \{0,1\}, && i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(2o)}\\
& y_{ijpqk} \in \{0,1\}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(2p)}
\end{aligned}
$$

where (2b) ensures that every operation is processed exactly once, and (2c) makes sure that each operation is scheduled on a resource allowed for that operation. The constraints (2d) and (2e) determine an ordering of the operations that are processed on the same resource. The constraints (2d) make sure that at most one of the variables y_{ijpqk} and y_{pqijk} may attain the value 1, and the constraints (2e) regulates that at least one of the variables y_{ijpqk} and y_{pqijk} must have the value 1 if operations (i, j) and (p, q) are to be performed on the same resource.

The constraints (2f) ensure that the starting time of operation (p, q) is scheduled after the completion of the previous operation on the same resource. Generally, in scheduling problems, symmetry preventing constraints $t_{pq} + p_{pq} - My_{ijpqk} \le t_{ij}$ are required, but this is redundant here since the variables y_{ijpqk} and y_{pqijk} are controlled by the inequalities (2d)–(2e). The constraints (2g) ensure that the operations within job j are scheduled in the right order and that each operation starts after the previous operation is completed and the goods is transported to the next resource.

The inequality (2h) regulates the starting times of the first operation of every job, so that no job is scheduled before its release date. The inequality (2i) makes sure that no operation is scheduled on resource k before this resource is available for the first time. The constraint (2j) regulates that any pair of jobs to be processed on the same physical part is scheduled in the right order. The constraints (2k)–(2m) determine the finishing times and the tardiness for the objective function. The constraints (2n) are redundant due to the inequalities (2g)–(2i), provided that $r_j \ge 0$ for all j.

4 Decomposition into machining and feasibility problems

4.1 Sets and variables

$\tilde{\mathcal{K}} =$ the set of multitask machines, $\tilde{\mathcal{K}} \subset \mathcal{K}$

$$z_{jk} = \begin{cases} 1, & \text{if job } j \text{ is allocated to resource } k, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{jqk} = \begin{cases} 1, & \text{if job } j \text{ is being processed before job } q \text{ on resource } k, \\ 0, & \text{otherwise.} \end{cases}$$

$t_j =$ the starting time of the machining operation of job j.

$s_j = t_j + p_j^{\mathtt{m}} + p_j^{\mathtt{pm}}$, the completion time of job j,

where $p_j^{\mathtt{pm}}$ is the sum of the post-machining route operations.

$$h_j = \begin{cases} s_j - d_j, & \text{if } s_j > d_j, \text{ i.e., the tardiness of job } j, \\ 0, & \text{otherwise.} \end{cases}$$

4.2 The machining problem

$$\begin{aligned}
&\text{Minimize} && \sum_{j\in\mathcal{J}} (A_j s_j^{\mathtt{m}} + B_j h_j^{\mathtt{m}}), && && \text{(3a)}\\
&\text{subject to} && \sum_{k\in\widetilde{\mathcal{K}}} z_{jk}^{\mathtt{m}} = 1, && j \in \mathcal{J}, && \text{(3b)}\\
& && z_{jk}^{\mathtt{m}} \le \lambda_{jk}^{\mathtt{m}}, && j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, && \text{(3c)}\\
& && y_{jqk}^{\mathtt{m}} + y_{qjk}^{\mathtt{m}} \le z_{jk}^{\mathtt{m}}, && j, q \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, j \ne q, && \text{(3d)}\\
& && y_{jqk}^{\mathtt{m}} + y_{qjk}^{\mathtt{m}} + 1 \ge z_{jk}^{\mathtt{m}} + z_{qk}^{\mathtt{m}}, && j, q \in \mathcal{J},\ j \ne q,\ k \in \widetilde{\mathcal{K}}, && \text{(3e)}\\
& && t_j^{\mathtt{m}} + p_j^{\mathtt{m}} - M(1 - y_{jqk}^{\mathtt{m}}) \le t_q^{\mathtt{m}}, && j, q \in \mathcal{J}, j \ne q,\ k \in \widetilde{\mathcal{K}}, && \text{(3f)}\\
& && t_j^{\mathtt{m}} \ge r_j^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(3g)}\\
& && t_j^{\mathtt{m}} \ge a_k z_{jk}^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(3h)}\\
& && t_q^{\mathtt{m}} \ge s_j^{\mathtt{m}} + v_{jq}^{\mathtt{m}}, && (j, q) \in \mathcal{Q}, && \text{(3i)}\\
& && s_j^{\mathtt{m}} = t_j^{\mathtt{m}} + p_j^{\mathtt{m}} + p_j^{\mathtt{pm}}, && j \in \mathcal{J}, && \text{(3j)}\\
& && h_j^{\mathtt{m}} \ge s_j^{\mathtt{m}} - d_j^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(3k)}\\
& && h_j^{\mathtt{m}} \ge 0, && j \in \mathcal{J}, && \text{(3l)}\\
& && t_j^{\mathtt{m}} \ge 0, && j \in \mathcal{J}, && \text{(3m)}\\
& && z_{jk}^{\mathtt{m}} \in \{0, 1\}, && j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, && \text{(3n)}\\
& && y_{jqk}^{\mathtt{m}} \in \{0, 1\}, && j, q \in \mathcal{J},\ j \ne q,\ k \in \widetilde{\mathcal{K}}, && \text{(3o)}
\end{aligned}$$

where $v_{jq}^{\mathtt{m}} = v_{jq} + t_{1q}$ and $p_j^{\mathtt{pm}} = \sum_{i=3}^{n_j} p_{ij}$. To simplify the implementation of the column generation algorithm, you may disregard the constraints (3i) (a pair of jobs to be processed on the same physical part scheduled in the right order).

4.3 The feasibility problem

The aim of the feasibility problem is to produce good feasible schedules for the remaining resources of the MT cell, i.e., the three setup and the two deburring stations. The objective function consists of three terms. The first term is the total processing lead time, i.e., the time the fixture for each job is occupied, $s_j - t_{1j}$. The second term is the total tardiness. The third term is a weight ω_k times the variable

z_{ijk}, inserted in order to avoid large computation times caused by symmetric solutions for the three setup stations.

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{j\in\mathcal{J}} \Big(A_j s_j - 0.001 t_{1j} + B_j h_j + \sum_{i\in\mathcal{N}_j} \sum_{k\in\mathcal{K}} \omega_k z_{ijk} \Big) && && \text{(4a)}\\
\text{subject to} \quad & \sum_{k\in\mathcal{K}} z_{ijk} = 1, && i\in\mathcal{N}_j,\ j\in\mathcal{J}, && \text{(4b)}\\
& z_{ijk} \le \lambda_{ijk}, && i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(4c)}\\
& y_{ijpqk} + y_{pqijk} \le z_{ijk}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(4d)}\\
& z_{ijk} + z_{pqk} - y_{ijpqk} - y_{pqijk} \le 1, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(4e)}\\
& t_{ij} + p_{ij} - M(1 - y_{ijpqk}) \le t_{pq}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(4f)}\\
& t_{ij} + p_{ij} + w \le t_{i+1,j}, && i\in\mathcal{N}_j \setminus \{n_j\},\ j\in\mathcal{J}, && \text{(4g)}\\
& t_{1j} \ge r_j, && j\in\mathcal{J}, && \text{(4h)}\\
& t_{ij} - a_k z_{ijk} \ge 0, && j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(4i)}\\
& t_{1q} - s_j \ge v_{jq}, && (j,q)\in\mathcal{Q}, && \text{(4j)}\\
& s_j - t_{n_j j} = p_{n_j j}, && j\in\mathcal{J}, && \text{(4k)}\\
& s_j - h_j \le d_j, && j\in\mathcal{J}, && \text{(4l)}\\
& h_j \ge 0, && j\in\mathcal{J}, && \text{(4m)}\\
& t_{ij} \ge 0, && i\in\mathcal{N}_j,\ j\in\mathcal{J}, && \text{(4n)}\\
& y_{2j2qk} = y^m_{jqk}, && j,q\in\mathcal{J},\ k\in\mathcal{K}, && \text{(4o)}\\
& z_{2jk} = z^m_{jk}, && j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(4p)}\\
& z_{ijk} \in \{0,1\}, && i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, && \text{(4q)}\\
& y_{ijpqk} \in \{0,1\}, && i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, && \text{(4r)}
\end{aligned}
$$

where y^m_{jqk} and z^m_{jk} are the solutions obtained from the machining problem. Analogously with the previous section, you may disregard the constraints (4j).

5 The time indexed machining model

5.1 Time intervals

The time horizon of the schedule is divided into $\mathtt{T}+1$ time intervals. The index $u \in \mathcal{T} = \{0, 1, \ldots, \mathtt{T}\}$ denotes a time interval starting at u and ending at $u+1$ and having the length ℓ.

5.2 Variables

$$
x_{jku} = \begin{cases} 1, & \text{if job } j \text{ is to start at the beginning of time interval } u \text{ on} \\ & \text{resource } k, \\ 0, & \text{otherwise.} \end{cases}
$$

$s_j =$ the completion time of job j.

$$
h_j = \begin{cases} s_j + p_j^{\mathtt{pm}} - \tilde{d}_j, & \text{if } s_j + p_j^{\mathtt{pm}} > \tilde{d}_j, \text{ i.e., the tardiness of job } j, \\ 0, & \text{otherwise.} \end{cases}
$$

5.3 The time indexed optimization model

$$\text{Minimize} \quad \sum_{j\in\mathcal{J}} (A_j s_j + B_j h_j), \tag{5a}$$

$$\text{subject to} \quad \sum_{k\in\tilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} x_{jku} = 1, \qquad j \in \mathcal{J}, \tag{5b}$$

$$\sum_{u\in\mathcal{T}} x_{jku} \le \lambda_{jk}, \qquad j \in \mathcal{J},\ k \in \tilde{\mathcal{K}}, \tag{5c}$$

$$\sum_{j\in\mathcal{J}} \sum_{\nu=[u-\tilde{p}_j+1]_+}^{u} x_{jk\nu} \le 1, \qquad k \in \tilde{\mathcal{K}}, u \in \mathcal{T}, \tag{5d}$$

$$\sum_{k\in\tilde{\mathcal{K}}} \left(\sum_{\mu=0}^{u} x_{jk\mu} - \sum_{\nu=0}^{u+\tilde{v}_{jq}^{\texttt{pm}}} x_{qk\nu} \right) \ge 0, \qquad u = 0, \ldots, \texttt{T} - \tilde{v}_{jq}^{\texttt{pm}},\ (j,q) \in \mathcal{Q}, \tag{5e}$$

$$x_{jku} = 0, \qquad u = \texttt{T} - \tilde{v}_{jq}^{\texttt{pm}}, \ldots, \texttt{T},\ (j,q) \in \mathcal{Q}, \tag{5f}$$

$$\sum_{k\in\tilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} u x_{jku} + \tilde{p}_j^{\texttt{pm}} = s_j, \qquad j \in \mathcal{J}, \tag{5g}$$

$$s_j - h_j \le \tilde{d}_j, \qquad j \in \mathcal{J}, \tag{5h}$$

$$h_j \ge 0, \qquad j \in \mathcal{J}, \tag{5i}$$

$$x_{jku} = 0, \qquad u=0, \ldots, \max\{\tilde{r}_j^{\texttt{m}}, \tilde{a}_k\}-1, j\in\mathcal{J}, k\in\tilde{\mathcal{K}}, \tag{5j}$$

$$x_{jku} \in \{0,1\}, \quad j \in \mathcal{J},\ k \in \tilde{\mathcal{K}},\ u \in \mathcal{T}, \tag{5k}$$

where $\tilde{v}_{jq}^{\texttt{pm}} = \tilde{p}_j^{\texttt{pm}} + v_{jq}$ is the interoperation time between the jobs j and q including the processing time of the machining and post-machining operations of job j. You may disregard the constraints (5e) and (5f).

It is possible to reformulate the objective function using solely the x_{jku}-variables. With this reformulation, the continuous variables s_j and h_j are no longer needed and hence the constraints (5g), (5h), and (5i) can be removed (as well as the constraints (5e) and (5f)). The machining problem is hence reformulated as to

$$\text{minimize} \sum_{j\in\mathcal{J}} \sum_{k\in\tilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} \left(A_j(u+\tilde{p}_j^{\texttt{pm}}) + B_j \left[u+\tilde{p}_j^{\texttt{pm}} - \tilde{d}_j \right]_+ \right) x_{jku},$$

$$\text{subject to} \quad \sum_{k\in\tilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} x_{jku} = 1, \qquad j \in \mathcal{J},$$

$$\sum_{u\in\mathcal{T}} x_{jku} \le \lambda_{jk}, \qquad j \in \mathcal{J},\ k \in \tilde{\mathcal{K}},$$

$$\sum_{j\in\mathcal{J}} \sum_{\nu=[u-\tilde{p}_j+1]_+}^{u} x_{jk\nu} \le 1, \qquad k \in \tilde{\mathcal{K}}, u \in \mathcal{T},$$

$$x_{jku} = 0, \qquad u=0,1,\ldots,\max\{\tilde{r}_j^{\texttt{m}}, \tilde{a}_k\} - 1, j\in\mathcal{J}, k\in\tilde{\mathcal{K}},$$

$$x_{jku} \in \{0,1\}, \quad j \in \mathcal{J},\ k \in \tilde{\mathcal{K}},\ u \in \mathcal{T}.$$