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

TMA521
MMA510
Optimization, project course

September 22, 2011

Project 2: Production scheduling

1 The project task

The project task is to define and implement a decomposition algorithm for the scheduling problem described below. For this, there are two alternatives, composed by the two models *the engineer's model* and *the time indexed model*. The algorithm that should be used is the *Dantzig-Wolfe reformulation solved by column generation*.

1.1 Problem files and implementation

At your service, AMPL model files (.mod) for the two mathematical models are provided at the course homepage together with a script file (.run) and a set of data files (.dat) for each model. These files can be used to solve the problem cases by the Cplex solver, to get key solutions for comparisons. All files named *MTC3* correspond to the engineer's model and all files named *MTC4* correspond to the time indexed model.

The decomposition algorithm chosen should be implemented in AMPL or Matlab (or C/C++ if you prefer). The sub and master problems should be solved by Cplex MILP solver.

Instructions on how to write scripts in the AMPL command language can be found, e.g., at the web page `www.ampl.com/NEW/index.html#LoopTest` under "Looping and Testing 1 & 2". Instructions for the AMPL and Matlab interfaces to Cplex are found, e.g., in `www.math.chalmers.se/Math/Grundutb/CTH/mve165/0910/Exercises/LP_Exc_1003.pdf`

1.2 Deadlines and examination

A preliminary report should be sent to anstr.chalmers@analys.urkund.se on Friday, 7th of October and the final report should be sent in on Tuesday, 11th of October. The program codes should be sent in on Tuesday, 11th of October.

The examination of the project includes an oral presentation of the same and an opposition to another groups project, during the seminar on Wednesday, 12th of October. The competition will also take place at this seminar.

1.3 Applying the algorithm to the models

The machining problem (see Section 4) should be solved using the column generation method, but the feasibility problem may be solved directly, without decomposition, using the Cplex MILP-solver.

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 (and for the time indexed model also the corresponding allocation of operations to machines). 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 MIP-solver.

If any of these instructions or the AMPL-files are unclear or seem unsuitable, please tell me and your fellow students.

You are also welcome to discuss any difficulty or indistinct instruction in this assignment with me or Karin, but don’t forget to discuss also with your fellow students.

1.4 Presentation of results

The types of results that should be presented in the report and at the seminars include upper and lower bounds on the optimal value as functions of number of iterations and 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 at the 11th of October.

2 Definition of the problem

2.1 Indices and sets

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

- Planned orders not yet released, i.e. exist 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 $\mathcal{Q} \subset \mathcal{J} * \mathcal{J}$. For the part, which routing is described by Figure 1, the pairs (j, q) and (q, l) belongs to the set $\mathcal{Q}$.

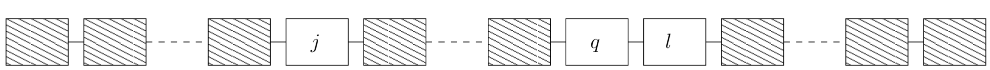

Figure 1: *Routing of a part with jobs j, q and l.* $(j, q), (q, l) \in \mathcal{Q}$.

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

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

Table 1: The possible route operations i in the multitask cell.

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 where the release date of job j is less than or equal to the release date of job q. The set $\mathcal{K}$ denotes the 10 resources k that the multitask cell consists of.

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 multitask cell

2.2 Parameters

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

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 for job j.
p_{ij} : the processing time in hours for operation i of job j
w : the transportation time for a product inside the multitask cell
v_{jq} : the interoperation time between the jobs j and q, where $(j, q) \in \mathcal{Q}$
M : a sufficiently large positive number, i.e. greater 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, 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, let us denote it ϱ_j. This means that the job j in the MT cell is planned to be started at the latest at this point in time. The desired release date, r_j, we want to get hold on in the optimization model is however the realistic point in time when the part arrives to the MT cell. The best guess we can make is that this is given by

$$r_j = \max(\varrho_j - t_0 - 0.8\vartheta_j; \nu_j^0),$$

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

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

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

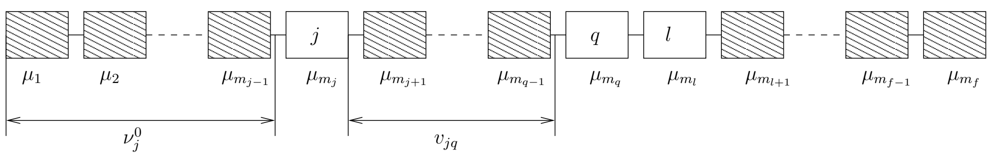

Figure 2: *The interoperation time, v_{jq}, is marked out together with ν_j^0 for the case of a planned order, i.e. $\mu_{act} = \mu_1$.*

In order to prevent the possibility that job j and q are scheduled too close in time, when both jobs are to be performed on the same physical part, the parameter v_{jq} is introduced. This is the planned interoperation time between the completion time of job j and the starting time of job q for jobs done outside the multitask cell. The definition of v_{jq} is

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

where ρ_μ, ς_μ and ϑ_μ denote the process, setup, and queue times of operation μ processed elsewhere as in (2.3).

3 The engineer's mathematical model

3.1 Variables

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

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

$t_{ij} =$ 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 function, is chosen so that it minimizes the job finish times together with the total tardiness, according to

$$\sum_{j\in\mathcal{J}}(s_j+h_j).$$

If also the time used in the fixure is considered, the objective function becomes

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

where $\varepsilon\in[0,1)$.

3.3 The optimization model

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

where (1b) ensures that every operation is processed exactly once, and (1c) makes sure that each operation is scheduled on a resource allowed for that operation. The constraints (1d) and (1e) determine an ordering of the operations that are processed on the same resource. The constraints (1d) make sure that at most one of the variables y_{ijpqk} and y_{pqijk} may attain the value 1, and the constraints (1e) 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 constraint (1f) makes sure 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, the symmetry preventing constraint $t_{pq}+p_{pq}-My_{ijpqk}\le t_{ij}$ is required, but this becomes redundant here since the variables y_{ijpqk} and y_{pqijk} are regulated by the inequalities (1d) and (1e). The constraints (1g) ensure that the operations within the same 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.

Equation (1h) regulates the starting times of the first operation of every job, so that no job is scheduled before its release date. Equation (1i) makes sure that no operation is scheduled on resource k before this resource is available for the first time. The constraint (1j) regulates that any pair of jobs to be processed on the same physical part is scheduled in the right order. The constraints (1k) through (1m) calculate the finish times and the tardiness for the objective function. The constraints (1n) are the nonnegativity constraints of the starting times. This is redundant due to equations (1g) through (1i) provided that r_j is nonnegative for all j. The constraints (1o) and (1p) are the binary constraints on the variables.

Note: In order to simplify the implementation of the column generation algorithm, you may disregard the constraints (1j) (a pair of jobs to be processed on the same physical part is scheduled in the right order).

4 Decomposition into a machining and a feasibility problem

4.1 Variables

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

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

$t_j =$ 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

$$\begin{aligned}
\text{Minimize} \quad & \sum_{j\in\mathcal{J}} (s_j^{\mathtt{m}} + h_j^{\mathtt{m}}), && && \text{(2a)}\\
\text{subject to} \quad & \sum_{k\in\widetilde{\mathcal{K}}} z_{jk}^{\mathtt{m}} = 1, && j \in \mathcal{J}, && \text{(2b)}\\
& z_{jk}^{\mathtt{m}} \le \lambda_{jk}^{\mathtt{m}}, && j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, && \text{(2c)}\\
& y_{jqk}^{\mathtt{m}} + y_{qjk}^{\mathtt{m}} \le z_{jk}^{\mathtt{m}}, && j,q \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, j \ne q, && \text{(2d)}\\
& y_{jqk}^{\mathtt{m}} + y_{qjk}^{\mathtt{m}} + 1 \ge z_{jk}^{\mathtt{m}} + z_{qk}^{\mathtt{m}}, && j,q \in \mathcal{J},\ j \ne q,\ k \in \widetilde{\mathcal{K}}, && \text{(2e)}\\
& t_j^{\mathtt{m}} + p_j^{\mathtt{m}} - M(1 - y_{jqk}^{\mathtt{m}}) \le t_q^{\mathtt{m}}, && j,q \in \mathcal{J}, j \ne q,\ k \in \widetilde{\mathcal{K}}, && \text{(2f)}\\
& t_j^{\mathtt{m}} \ge r_j^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(2g)}\\
& t_j^{\mathtt{m}} \ge a_k z_{jk}^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(2h)}\\
& t_q^{\mathtt{m}} \ge s_j^{\mathtt{m}} + v_{jq}^{\mathtt{m}}, && (j,q) \in \mathcal{Q}, && \text{(2i)}\\
& s_j^{\mathtt{m}} = t_j^{\mathtt{m}} + p_j^{\mathtt{m}} + p_j^{\mathtt{pm}}, && j \in \mathcal{J}, && \text{(2j)}\\
& h_j^{\mathtt{m}} \ge s_j^{\mathtt{m}} - d_j^{\mathtt{m}}, && j \in \mathcal{J}, && \text{(2k)}
\end{aligned}$$

$$
\begin{aligned}
h_j^{\mathtt{m}} &\geq 0, & & j \in \mathcal{J}, & \text{(2l)}\\
t_j^{\mathtt{m}} &\geq 0, & & j \in \mathcal{J}, & \text{(2m)}\\
z_{jk}^{\mathtt{m}} &\in \{0,1\}, & & j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, & \text{(2n)}\\
y_{jqk}^{\mathtt{m}} &\in \{0,1\}, & & j,q \in \mathcal{J},\ j \neq q,\ k \in \widetilde{\mathcal{K}}, & \text{(2o)}
\end{aligned}
$$

where $v_{jq}^{\mathtt{m}} = v_{jq} + t_{1q}$ and $p_j^{\mathtt{pm}} = \sum_{i=3}^{n_j} p_{ij}$.

Note: Analogously with the previous note, you may disregard the constraints (2i).

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 minimizing the total processing lead time, i.e. the time the fixture for each job is occupied, $s_j - t_{1j}$. The next term is minimizing the total tardiness and the last term is a weight ω_k times the variable z_{ijk}, in order to avoid large computation times caused by the computation of symmetric solutions for the three setup stations.

$$
\text{Minimize} \quad \sum_{j\in\mathcal{J}} \Big(s_j - 0.001 t_{1j} + h_j + \sum_{i\in\mathcal{N}_j} \sum_{k\in\mathcal{K}} \omega_k z_{ijk} \Big) \qquad \text{(3a)}
$$

$$
\begin{aligned}
\text{subject to} \quad \sum_{k\in\mathcal{K}} z_{ijk} &= 1, & & i\in\mathcal{N}_j,\ j\in\mathcal{J}, & \text{(3b)}\\
z_{ijk} &\leq \lambda_{ijk}, & & i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, & \text{(3c)}\\
y_{ijpqk} + y_{pqijk} &\leq z_{ijk}, & & i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, & \text{(3d)}\\
z_{ijk} + z_{pqk} - y_{ijpqk} - y_{pqijk} &\leq 1, & & i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, & \text{(3e)}\\
t_{ij} + p_{ij} - M(1 - y_{ijpqk}) &\leq t_{pq}, & & i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, & \text{(3f)}\\
t_{ij} + p_{ij} + w &\leq t_{i+1,j}, & & i\in\mathcal{N}_j \setminus \{n_j\},\ j\in\mathcal{J}, & \text{(3g)}\\
t_{1j} &\geq r_j, & & j\in\mathcal{J}, & \text{(3h)}\\
t_{ij} - a_k z_{ijk} &\geq 0, & & j\in\mathcal{J},\ k\in\mathcal{K}, & \text{(3i)}\\
t_{1q} - s_j &\geq v_{jq}, & & (j,q)\in\mathcal{Q}, & \text{(3j)}\\
s_j - t_{n_j j} &= p_{n_j j}, & & j\in\mathcal{J}, & \text{(3k)}\\
s_j - h_j &\leq d_j, & & j\in\mathcal{J}, & \text{(3l)}\\
h_j &\geq 0, & & j\in\mathcal{J}, & \text{(3m)}\\
t_{ij} &\geq 0, & & i\in\mathcal{N}_j,\ j\in\mathcal{J}, & \text{(3n)}\\
y_{2j2qk} &= y_{jqk}^m, & & j,q\in\mathcal{J},\ k\in\mathcal{K}, & \text{(3o)}\\
z_{2jk} &= z_{jk}^m, & & j\in\mathcal{J},\ k\in\mathcal{K}, & \text{(3p)}\\
z_{ijk} &\in \{0,1\}, & & i\in\mathcal{N}_j,\ j\in\mathcal{J},\ k\in\mathcal{K}, & \text{(3q)}\\
y_{ijpqk} &\in \{0,1\}, & & i\in\mathcal{N}_j, p\in\mathcal{N}_q, j,q\in\mathcal{J}, (i,j)\neq(p,q), k\in\mathcal{K}, & \text{(3r)}
\end{aligned}
$$

where y_{jqk}^m and z_{jk}^m are the solutions obtained from the machining problem.

Note: Analogously with the previous notes, you may disregard the constraints (3j).

5 The time indexed machining model

5.1 Time intervals

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

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

5.3 The time indexed optimization model

$$\text{Minimize} \quad \sum_{j\in\mathcal{J}} (s_j + h_j), \tag{4a}$$

$$\text{subject to} \quad \sum_{k\in\widetilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} x_{jku} = 1, \qquad j \in \mathcal{J}, \tag{4b}$$

$$\sum_{u\in\mathcal{T}} x_{jku} \le \lambda_{jk}, \quad j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}}, \tag{4c}$$

$$\sum_{j\in\mathcal{J}} \sum_{\nu=(u-p_j+1)_+}^{u} x_{jk\nu} \le 1, \qquad k \in \widetilde{\mathcal{K}}, u = 0, \dots, \mathtt{T}, \tag{4d}$$

$$\sum_{k\in\widetilde{\mathcal{K}}} \Big(\sum_{\mu=u}^{\mathtt{T}} x_{jk\mu} + \sum_{\nu=0}^{u+v_{jq}^{\mathtt{ext}}-1} x_{qk\nu} \Big) \le 1, \qquad (j,q) \in \mathcal{Q},\ u = 0, \dots, \mathtt{T} - v_{jq}^{\mathtt{ext}} + 1, \tag{4e}$$

$$x_{jku} = 0, \qquad j \in \mathcal{J}, k \in \widetilde{\mathcal{K}}, u = 0,1,\dots,\max\{r_j, a_k\}, \tag{4f}$$

$$\sum_{k\in\widetilde{\mathcal{K}}} \sum_{u\in\mathcal{T}} u x_{jku} + p_j + p_j^{\mathtt{pm}} = s_j, \qquad j \in \mathcal{J}, \tag{4g}$$

$$s_j - h_j \le d_j, \qquad j \in \mathcal{J}, \tag{4h}$$

$$h_j \ge 0, \qquad j \in \mathcal{J}, \tag{4i}$$

$$x_{jku} \in \{0,1\}, j \in \mathcal{J},\ k \in \widetilde{\mathcal{K}},\ u \in \mathcal{T}, \tag{4j}$$

where $v_{jq}^{\mathtt{ext}} = p_j + p_j^{\mathtt{pm}} + v_{jq}$. The constraints (4e) can be equivalently expressed as (this version is implemented in the 2011 version of the AMPL mod-file)

$$\sum_{k\in\widetilde{\mathcal{K}}} \Big(\sum_{\mu=0}^{u} x_{jk\mu} - \sum_{\nu=0}^{u+v_{jq}^{\mathtt{ext}}} x_{qk\nu} \Big) \ge 0, \quad (j,q) \in \mathcal{Q},\ u = 0, \dots, \mathtt{T} - v_{jq}^{\mathtt{ext}}, \tag{5}$$

Note: Analogously with the previous notes, you may disregard the constraints (4e) (as well as (5)).