FLOW-TIME ESTIMATION IN DYNAMIC JOB SHOPS WITH PRIORITY SCHEDULING USING A HYBRID MODELLING APPROACH

Jörg SIGRIST, Christoph HEITZ
Zurich University of Applied Sciences
Institute of Data Analysis and Process Design
Rosenstrasse 3, 8400 Winterthur
Switzerland
E-mail: sigj@zhaw.ch

Abstract:
A new approach for due date assignment in dynamic job shops with priority scheduling is presented. The future temporal development of the production system, eventually determining the flow-time of a job, is governed by both the processing of the jobs already present in the system as well as the processing of future arriving jobs. We combine a simulation-like approach for the already known jobs with a stochastic model describing the influence of future arriving jobs. The resulting model is a hybrid system dynamics model that can be solved numerically, leading to estimates for the flow-time of all available jobs.

In a simulation study, we compare the new approach with other popular methods known in literature. Our results indicate that the new method significantly outperforms all other studied methods in terms of accuracy of the estimates, in most cases by at least a factor of two. Furthermore, the effect of priority scheduling can be modelled correctly, yielding good estimates for jobs of different priorities.

Keywords:
Dynamic job shop, flow-time estimation, simulation, scheduling

1. INTRODUCTION
We consider a dynamic job shop environment with local priority scheduling, where the priority can either be static (e.g. a property of the job) or dynamic, such as for due-date related scheduling rules. Our goal is to estimate a flow-time for each job at the time instant of its arrival, under the assumption that the job properties (including the due date) are known at this time.

The completion time of a given job depends on how this particular job is processed, how the other jobs already present in the system are processed, and how jobs that are not yet known but will arrive during the processing of the considered job do interfere. This is a rather complex problem, and different attempts have been made to reduce this complexity and to identify a set of key information numbers that is sufficient for a good estimate. Prior investigations have shown that there are at least three main determinants for the completion time [7], [12]: the properties of the job itself (type, processing time, priority, route information), the general state of the production system (load balance, congestion) as well as the loading of the stations on the job's route. Most methods in literature are based on regression models that combine several of these indicator variables to estimate a flow-time.
2. DYNAMICAL MODEL

2.1. Basic idea
Basically, our model is a forward-simulation which explicitly models and includes future arriving jobs. Our approach contains both a deterministic and a stochastic element, which are defined as follows: the deterministic element consists of all the jobs which are present at simulation start. All information of these jobs (in the following called real jobs) are known, and every simulation yields a completion time for each of them. The stochastic part of our approach is modelled by the so-called virtual work, which continuously arrives at and flows through the system. These two types of work compete against each other for the same resources. As a result, the completion times of the real jobs will be delayed by the virtual work.

2.2. Considered system and modelling approach
We consider a production system consisting of several working stations \( j = 1, \ldots, M \), each station having only one server (no parallel processing capabilities) with a processing rate \( \mu_j \). Jobs arrive randomly as a Poisson process with rate \( \lambda_j \) at station \( j \), the overall arrival rate being \( \lambda = \sum_{j=1}^{M} \lambda_j \). An arriving job \( i \) has a random sequence of stations, described by routing probabilities \( p_{ij} \). The processing times at each station \( j \) are random with an average of \( \tau_j \).

Each job has a priority \( p \) and scheduling is done according to the priority. Jobs arriving from outside have a random priority \( p \) which is described by a function \( g(p) \), where \( g(p) \) is the probability that the arriving job has a priority higher than \( p \). Throughout the paper we assume non-preemptive schemes, i.e. jobs that are currently processed are finalized even if a job with a higher \( p \)-value arrives.

2.3. Modelling of virtual work
At each working station, real jobs as well as virtual work are present. The waiting virtual work at any time instant \( t \) is described by the function \( G(p, t) \), which is defined as follows: \( G(p, t) \) is the amount of queued virtual processing time with priority larger than \( p \), where the priority is allowed to take arbitrary values between \( -\infty \) and \( +\infty \). The value \( G(-\infty, t) \) corresponds to the total virtual work amount in the queue at time \( t \). Furthermore, we always have \( G(\infty, t) = 0 \). For the processing, we assume that the virtual work can be split into arbitrarily small portions. This, of course, is an approximation to the real discrete dynamics, and additional mechanisms are introduced to make sure that basic queuing properties are correctly reproduced.

The processing of real jobs and virtual work at a server follows the following rules:

- Virtual work arrives at and flows through the system constantly. Thus, the virtual work behaves like a fluid, using resources and delaying real jobs.
- The servers may process either real jobs or virtual work.
- If the server processes a real job, the server is busy until the job is fully completed (non-preemptive scheme). Virtual work in contrast is processed in arbitrary small portions (flow approximation) and the processing can be interrupted at any time.
• If a server processes virtual work and a real job with priority \( p' \) is present in the queue, no virtual work with priority lower than \( p' \) is processed. This reflects the fact that virtual and real work are treated identically with respect to the scheduling process.

2.4. Job arrivals at queue

We assume that virtual work arrives continually in small work packages \( \Delta G(p) \), where the function \( G(p) \) describes the priority distribution as defined above. Thus, \( G(-\infty) \) equals the total amount of arrived virtual work in the considered time increment \( \Delta t \). The dynamics of \( G(p, t) \) due to arrival of virtual work is given by

\[
G(p, t + \Delta t) = G(p, t) + \Delta G(p)
\]

For virtual work that arrives from another station of the job shop, \( \Delta G(p) \) is given by the processing of the upstream station and the routing (see 2.5 and 2.6). For the arrival of a virtual job from outside of the system, we follow the approach in [8]: let \( \lambda \) be the arrival rate of the jobs at the considered station, \( \tau \) the average processing time, and \( g(p) \) the probability that the priority of an arriving job is larger than \( p \). Then we get:

\[
\Delta G(p) = \lambda \cdot \tau \cdot \Delta t \cdot g(p)
\]

Here, \( \lambda \cdot \Delta t \) is the expected number of arriving jobs, and \( \lambda \cdot \Delta t \cdot \tau \) is the expected total amount of arriving work during any time interval \( \Delta t \).

2.5. Processing

Every time a server is either idle or processing virtual work, a decision has to be made what kind of work has to be processed next. Let's define the priority \( p' \) as the highest priority of all jobs waiting in the queue. We have to distinguish between the following cases:

1. There is no virtual work available with priority higher than \( p' \)
2. There is virtual work available with priority higher than \( p' \)
3. There is no real job in the queue

For case 1, the server starts to process the real job with the highest priority. All the virtual work remains in the container and \( G(p, t) \) is unchanged.

For case 2, the server will start to process virtual work. While for all priorities \( p' \) a uniform processing with respect to the priority is assumed, no virtual work with a smaller priority than \( p' \) is being processed. The effect on the virtual work container of the server's queue can be formulated as follows:

\[
G(p, t + \Delta t) = \begin{cases} 
G(p, t) - \Delta t, & p \leq p' \\
G(p, t) - \frac{G(p, t)}{G(p', t)} \Delta t, & p > p'
\end{cases}
\]

For case 3, only virtual work is present, which in particular will be the case for \( t \to \infty \). In order to yield a consistent model, we require \( \rho(t \to \infty) = \bar{\rho} \), and \( \bar{W}_q(t \to \infty) = \bar{W}_q \), where \( \bar{\rho} \) is the average utilization of the server and \( \bar{W}_q \) is the average waiting time for a job at this station. This consistency can only be achieved by introducing a dynamic utilization \( \rho(t) \). For
this purpose, we start with the well known M/M/1 queuing model relationship (see e.g. [3])
\[ \bar{\rho} = \frac{\mu \cdot \bar{W}_q}{\left(1 + \mu \cdot \bar{W}_q\right)} \] which establishes a relationship between the mean utilization \( \bar{\rho} \), the
mean queuing time \( \bar{W}_q \) and the processing rate \( \mu \). This relationship is only valid for long
term observations. We nevertheless can use it as an approximation when we’re dealing with
virtual work, as virtual work is an expected mean value itself. As the M/M/1 assumption isn’t
valid in an arbitrary system, we simply expand the above relationship by a scalar, which is
set such as the initially mentioned consistency for \( t \to \infty \) is reached. This results in:
\[ \rho(t) = \frac{1}{\bar{W}_q} + \mu \frac{1}{\bar{W}_q(t)} + \mu \]
where \( \rho(t) \): current utilization, \( W_q(t) \): current waiting time at the server and \( \mu \): mean
processing rate at the server. With the same heuristics as above for case 2, \( G(p, t + \Delta t) \) can
be expressed for all \( p \) as \( G(p, t + \Delta t) = G(p, t) + \Delta G(p, t) \) with \( \rho(t) \) according to (4) and:
\[ \Delta G(p, t) = \frac{G(p, t)}{G(-\infty, t)} \cdot \rho(t) \cdot \Delta t \]
The virtual work outflow from server \( i \) at any time \( t \) is denoted by \( G_{\text{out}, i}(p, t) \). As no virtual
work will be processed in case 1., no virtual work will leave the server and thus
\( G_{\text{out}, i}(p, t) = 0 \). For cases 2. and 3., the outflow of virtual work exactly matches the amount
of virtual work which is being processed: \( G_{\text{out}, i}(p, t) = \Delta G_i(p, t) \). This is consistent with the
fluid approximation for the virtual work.

2.6. Routing
The flow of virtual work from station \( i \) to station \( j \) depends not only on the above discussed
outflow, but also on the routing probability \( \rho_{ij} \) and the average processing times at the
sending and receiving stations:
\[ \Delta G_j(p, t) = \rho_{ij} \cdot \frac{G_{\text{out}, i}(p, t)}{\tau_i} \cdot \tau_j \]
where \( \rho_{ij} \): routing probability from station \( i \) to \( j \) (similar to Jackson Networks [3]), \( \frac{G_{\text{out}, i}(p, t)}{\tau_i} \):
dimensionless outflow, representing a fraction of an average job, \( \tau_j \): processing time of an
average job at station \( j \)

3. ESTIMATION OF FLOW-TIME
Flow-times are estimated using a time discrete forward simulation. Each run is initialized
with the current system state, and time is incremented in steps of \( \Delta t \). At each simulation
step, the virtual work containers and real jobs are updated according to the method
described in 2. This procedure continues as long as real jobs remain in the system. At the
end of the simulation, a completion time will be assigned to each job.
4. SIMULATION

Simulations are made in a Discrete-Event (DE) simulation environment. Each time a new job enters the system, its flow-time is estimated according to the method described in 3. This estimation is stored and compared with the job’s real duration of stay in the system. Statistics then are made with the difference between the estimates and the real values.

The investigations are made on a traditional job shop system as used in [12] with five workstations. The new flow-time estimation method (VWS: virtual work simulation) is tested and compared to 12 other popular methods, including:

OBE [12], ADRES and LDP [2], DTWK and DPPW [4], TWK [13], NOP [5], SLK [1], PPW [9], JIQ [6], WIQ and JIS [11]. All methods are tested under different loads using both FIFO and SSTF policy.

The performances are measured in respect of different standard performance measures: mean lateness (ML), mean absolute lateness (MAL), mean tardiness (MT), mean squared lateness (MSL) and mean semi quadratic lateness (MSQL).

5. RESULTS

Similar to some methods proposed in literature, our method is slightly biased, which is reflected in a ML value unequal to 0. This bias is a result of the fluid characteristics of the virtual work which doesn’t match correctly the behaviour of real future jobs.

In terms of accuracy (MAL criterion), the VWS method outperforms each other studied method as shown in table 1: in case of SSTF policy and 85% utilization, VWS yields a MAL value of 3.626. The next best method OBE generates a value of 10.6302 and thus is three times less accurate than VWS. This difference is not that large but still statistically relevant for FIFO policy (benefit: 28%) and lower utilization (65% utilization leads to a superiority of 100% (SSTF) and 25% (FIFO)).

<table>
<thead>
<tr>
<th>PM</th>
<th>ML FIFO</th>
<th>ML SSTF</th>
<th>ML FIFO</th>
<th>ML SSTF</th>
<th>MT FIFO</th>
<th>MT SSTF</th>
<th>MSL FIFO</th>
<th>MSL SSTF</th>
<th>MSQ FIFO</th>
<th>MSQ SSTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>VWS</td>
<td>0.8550</td>
<td>1.5043</td>
<td>6.3974</td>
<td>3.5608</td>
<td>3.6262</td>
<td>2.5025</td>
<td>110.4811</td>
<td>34.2713</td>
<td>79.4416</td>
<td>29.2224</td>
</tr>
<tr>
<td>OBE</td>
<td>0.2048</td>
<td>4.3097</td>
<td>8.2391</td>
<td>10.6302</td>
<td>4.2220</td>
<td>7.4700</td>
<td>170.2803</td>
<td>221.9802</td>
<td>100.5640</td>
<td>175.4298</td>
</tr>
<tr>
<td>LDP</td>
<td>5.8353</td>
<td>5.4071</td>
<td>18.7346</td>
<td>18.6552</td>
<td>9.7649</td>
<td>12.0430</td>
<td>376.5371</td>
<td>644.8649</td>
<td>293.3706</td>
<td>420.7435</td>
</tr>
<tr>
<td>DPPW</td>
<td>0.0000</td>
<td>0.6491</td>
<td>15.0989</td>
<td>14.9951</td>
<td>7.5949</td>
<td>7.5221</td>
<td>412.1497</td>
<td>404.9258</td>
<td>222.0446</td>
<td>217.1754</td>
</tr>
<tr>
<td>DTWK</td>
<td>-0.0861</td>
<td>-0.0983</td>
<td>24.1134</td>
<td>23.3225</td>
<td>12.0068</td>
<td>11.6421</td>
<td>1259.8985</td>
<td>1146.9708</td>
<td>497.0048</td>
<td>432.7304</td>
</tr>
<tr>
<td>JIS</td>
<td>-1.9211</td>
<td>-1.0908</td>
<td>21.1005</td>
<td>15.7323</td>
<td>9.5042</td>
<td>7.3483</td>
<td>742.7265</td>
<td>433.8340</td>
<td>413.8501</td>
<td>193.6483</td>
</tr>
<tr>
<td>PPW</td>
<td>0.1296</td>
<td>2.9677</td>
<td>21.5739</td>
<td>20.1193</td>
<td>10.8518</td>
<td>11.5425</td>
<td>835.8065</td>
<td>702.7653</td>
<td>554.3024</td>
<td>500.9241</td>
</tr>
<tr>
<td>NOP</td>
<td>0.1279</td>
<td>2.9663</td>
<td>21.7639</td>
<td>20.3825</td>
<td>10.9459</td>
<td>11.6744</td>
<td>849.7887</td>
<td>720.5836</td>
<td>560.6232</td>
<td>509.7629</td>
</tr>
<tr>
<td>SLK</td>
<td>0.0965</td>
<td>-0.1948</td>
<td>27.5518</td>
<td>22.5071</td>
<td>13.8241</td>
<td>11.5662</td>
<td>1237.0023</td>
<td>802.6649</td>
<td>835.5040</td>
<td>492.6285</td>
</tr>
<tr>
<td>TWK</td>
<td>0.2425</td>
<td>11.0200</td>
<td>26.3740</td>
<td>24.7801</td>
<td>17.8083</td>
<td>18.2001</td>
<td>1333.9095</td>
<td>1088.4116</td>
<td>1017.3047</td>
<td>875.0342</td>
</tr>
<tr>
<td>CON</td>
<td>0.0887</td>
<td>-0.2025</td>
<td>29.5706</td>
<td>24.1684</td>
<td>14.8296</td>
<td>11.9830</td>
<td>1405.2670</td>
<td>917.5130</td>
<td>926.3967</td>
<td>545.9710</td>
</tr>
</tbody>
</table>

Table 9: Performance measures at 85% load, mean flow-time ~= 50
6. CONCLUSIONS

In this study we present a new approach for assigning due dates in a dynamic job shop. The proposed method differs from other popular due date assignment methods in several aspects: First, it explicitly models the processing capabilities of the system as a network of workstations, leading to a much more detailed model of the production system. Second, it is based on the full information on the present status of the production system. Third, the effect of future arriving jobs is taken into account by explicit modelling them as a flow of virtual work with statistical properties including priorities.

This new method is compared with the most popular and best performing methods known in the literature of flow-time estimation according to several performance measures. It is shown that the VWS method outperforms every other method in respect of the estimate's accuracy.

7. REFERENCES

[8] Heitz, C., Roithner, T., Estimation of job completion times in a dynamic job shop based on static job shop solution for IMS sessions, Paper 166