Noname manuscript No.
(will be inserted by the editor)

Using linear programming to analyze and optimize stochastic flow lines

Stefan Helber, Katja Schimmelpfeng, Raik Stolletz and Svenja Lagershausen[*]

Leibniz Universität Hannover, Institut für Produktionswirtschaft, Königsworther Platz 1, 30167 Hannover, Germany
e-mail: {raik.stolletz,stefan.helber}@prod.uni-hannover.de,
katja.schimmelpfeng@tu-cottbus.de,

February 14, 2008, revised February 25, 2009

Abstract This paper presents a linear programming approach to analyze and optimize flow lines with limited buffer capacities and stochastic processing times. The basic idea is to solve a huge but simple linear program that models an entire simulation run of a multi-stage production process in discrete time, to determine a production rate estimate. As our methodology is purely numerical, it offers the full modeling flexibility of stochastic simulation with respect to the probability distribution of processing times. However, unlike discrete-event simulation models, it also offers the optimization power of linear programming and hence allows to solve buffer allocation problems. We show under which conditions our method works well by comparing its results to exact values for two-machine models and approximate simulation results for longer lines.

1 Modeling flow lines with limited buffer capacities and random processing times

Stochastic processing times at the stations of a flow line with limited buffer capacities can lead to blocking or starvation of the line's bottleneck. In this case the throughput of the line falls below the production rate of the bottleneck operating in isolation (Gershwin, 1994, p. 117). In the design process for a flow line, one needs to quantify this impact of processing time variability on the line's production rate and inventory level to efficiently allocate machines and buffers.

In practice, discrete-event simulation (DES) is usually used to analyze the performance of a planned flow line. Several software packages with graphical user interfaces allow the planner to easily model a system at an arbitrary level of detail (Swain, 2007). DES offers a great degree of modeling flexibility with respect to probability distributions and other details of the line's mode of operation. However, while modeling a flow line via DES is easy, a systematic optimization of the flow line design is not. A simple and relevant question is how to allocate a given total number of identical buffer spaces in a flow line so that the production rate is maximized. This question can usually not be answered efficiently using DES because of the long computation times of the simulation runs and the combinatorial nature of the decision problem (Gershwin und Schor, 2000).

It is also possible to use analytic queueing models to derive exact or approximate closed-form solutions or decomposition algorithms for flow lines with (un-)limited buffer capacities. The numerical effort for these methods is often negligible so that a systematic optimization of the line is possible (Gershwin und Schor, 2000). However, the mathematical assumptions required for these analytic models often restrict their use in practice. In addition, even a slight

[*] The authors thank the anonymous referees for their helpful comments and suggestions.

modification of such an analytic model may easily exceed the capabilities of a practitioner who may therefore resort to DES.

As a result, one rarely finds flow lines with limited buffer capacity that have been systematically optimized, as analytic queueing models are rarely understood and optimization based on DES is often too time-consuming.

For the special problem of analyzing and optimizing flow lines with limited buffer capacity we propose a methodology that is about as simple as DES, but uses the optimization potential of mixed-integer linear programming. The key idea is to work with a discrete-time dynamic production-inventory model with continuous production quantities. This model approximates the behavior of a discrete-material production system operating in continuous time. Among the parameters of this model is the production capacity of a production stage during a (discrete) time period. It stems from a hypothetical simulation run in continuous time. In other words, the realizations of the stochastic processing times of the different jobs at a given production stage are transformed via sampling into corresponding realizations of production capacities for that production stage and the corresponding time period. If the number of these periods in the model is sufficiently large and some other conditions (to be explored in this paper) hold, the discrete-time model leads to a surprisingly accurate prediction of the production rate of the original flow line that operates in continuous time. To determine the production rate estimate within the context of our multi-stage discrete-time production-inventory model, we use the simplex algorithm of linear programming (LP). Our linear model can easily incorporate buffer allocation and/or machine selection decisions. In this case, additional integer decision variables for the buffer sizes are introduced. This leads to a mixed-integer problem that can be solved via branch&bound or branch&cut algorithms. Our approach therefore combines the flexibility of DES with respect to probability distributions of stochastic processing times with the optimization power of (mixed-integer) linear programming. The contribution of this paper is to describe how the method works and under which conditions it can be expected to yield precise production rate estimates.

The literature on DES of production system is huge. Law und Kelton (1991) and Kelton et al. (2006), among others, give introductions to the methodology and describe simulation models of flow lines. Ho et al. (1979) introduced the concept of infinitesimal perturbation analysis were a single sample path (in continuous time) of a DES is used to determine a gradient of a performance measure for optimization purposes. A survey of the literature on analytic queueing models of flow lines with limited buffer capacity is given by Dallery und Gershwin (1992). The recent development in the field is presented in the book edited by Liberopoulos et al. (2006). Several monographs treat the analysis of manufacturing systems via queueing models (Buzacott und Shanthikumar, 1993; Gershwin, 1994; Tijms, 1994; Altiok, 1996). The literature on linear-programming based simulation of flow lines using a sample path of processing times is more limited. Abdul-Kader (2006) presents a linear programming (evaluation) model of an unreliable flow line in continuous time that is based on an earlier model by Johri (1987). In this model, the buffer capacity cannot be made a decision variable and only a fixed and given buffer allocation can be treated. The situation is similar for a model by Matta und Chefson (2005) which is based on an earlier model by Schruben (2000). In the model of a closed flow line by Matta und Chefson, the buffer capacity determines the number of constraints of the continuous time LP. We are not aware of continuous time LP models of stochastic flow lines in which the buffer size is a decision variable. However, in order to optimize the design of a flow line, the buffer size must be allowed to be a decision variable. For this reason we developed our discrete-time model which is presented in this paper. From a methodological point of view, our approach is very similar to the one presented by Helber und Henken (2007) for shift scheduling in contact centers.

The remainder of this paper is structured as follows: In Section 2 we present and compare LP-based simulation approaches for stochastic flow lines. Section 3 presents results of a systematic numerical study to assess the accuracy of the proposed method. We summarize our results and give directions for further research in Section 4.

2 Continuous vs. discrete time linear programming models of stochastic flow lines

2.1 Continuous time evaluation model

As stated above, several modeling approaches have been proposed for continuous time LP models of flow lines. The key idea is to use for a sample part of processing times a set of real-valued decision variables to model the time at which processing of a workpiece w at a station k starts and/or ends. An example of such a model can be formulated as follows using the notation in Table 1, see also Matta und Chefson (2005):

$$\text{Minimize} \sum_{k=1}^{K}\sum_{w=1}^{W}(XS_{kw}+XF_{kw}) \tag{1}$$

subject to

$$\begin{aligned} XS_{kw}+d_{kw} &= XF_{kw}-B_{kw}, && \forall k,\ \forall w && (2)\\ XS_{k+1,w} &= XF_{kw}+W_{kw}, && \forall k \le K-1,\ \forall w && (3)\\ XS_{k,w+1} &= XF_{k,w}+S_{k,w+1}, && \forall k,\ \forall w \le W-1 && (4)\\ XF_{k,w+b_k} &\ge XS_{k+1,w}, && \forall k \le K-1,\ \forall w \le W-b_k && (5) \end{aligned}$$

The objective function (1) ensures that each workpiece w will be transferred to the next station or buffer as soon as possible. The time workpiece w spends at station k consists of the processing time d_{kw} and the blocking time B_{kw}. The buffer behind the last station K is assumed to be infinitely large so that no workpiece can be blocked at the last station, i.e., $B_{Kw} = 0$ holds for all workpieces w. If the workpiece w starts being processed at station k at time XS_{kw}, it leaves the station $d_{kw} + B_{kw}$ time units later, see Equations (2). Equations (3) define the waiting time W_{kw} of workpiece w in the buffer behind station k. The starting time of workpiece $w + 1$ is determined in Equations (4) by the finishing time of the preceding workpiece w and the starving time $S_{k,w+1}$. Therefore, only one workpiece can be processed at the same time at station k. Due to the limited buffer capacity b_k, the workpieces w and $w + b_k$ cannot be in the buffer behind station k at the same time. For this reason Equations (5) state that workpiece $w + b_k$ cannot be transferred to a buffer before workpiece w has left this buffer. Once this model has been solved for a given realization of processing times, an estimate for the production rate of the line can be computed.

In this model, the size of the buffers between the stations determines the number of constraints of the LP. The buffer allocation is therefore exogenous to the model. Matta und Chefson (2005) discuss how the solution to such a model can be incorporated into an approach to optimize the buffer allocation. If an identical sequence of realizations of random processing times for workpieces w at stations k is fed into an LP like the one presented above and into a DES, both modeling approaches lead to the same cycle time and processing rate (Schruben, 2000; Matta und Chefson, 2005). A continuous time LP model can therefore be used, at least in principle, to simulate a stochastic flow line. The resulting LP can become huge if tight confidence intervals for performance measures are required, but the power of computers and LP solvers keeps increasing, so this problem should eventually be eliminated by technological progress. However, there does not appear to be an obvious way to incorporate buffer allocation decisions into a continuous time model. For this reason, we propose a different approach.

2.2 Transformation of stochastic processing times into processing rates via periodic sampling

In order to transform the continuous processing times into the modeling context of a discrete time model, we can use a simple sampling approach. Consider a sequence of processing times or durations d_{kw} for an ordered set of workpieces w that is processed at a station k. Assume that the station is operating in isolation so that it can never be blocked or starved. The

Table 1 Notation for the continuous time model

Sets and indices	
$w = 1, ..., W$	workpieces
$k = 1, ..., K$	stations in the flow line
Parameters	
d_{kw}	processing time or duration at station k for workpiece w
b_k	capacity of the buffer behind station k
Real-valued decision variables	
XS_{kw}	starting time at station k for workpiece w
XF_{kw}	finishing time at station k for workpiece w
W_{kw}	waiting time for workpiece w in the buffer behind station k
B_{kw}	blocking time at station k for workpiece w
S_{kw}	starving time of station k before processing workpiece w

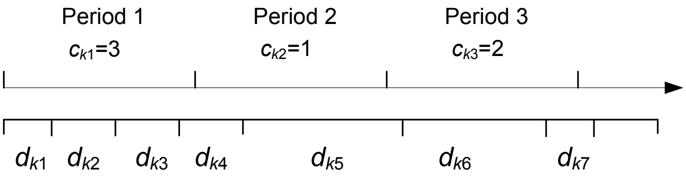

Fig. 1 Sampling of discrete time processing rates

idea of the sampling approach is to count the number of events per period as depicted in Figure 1.

The upper row of this figure shows three discrete time periods 1 to 3 and the lower, the processing times of 7 workpieces successively processed at that station so that it never idles until the last piece is processed. In the example, three workpieces can be finished in period 1, only one in period 2 and two in period 3. The shorter the processing times relative to the period lengths are, the higher is the number of workpieces that can be processed during a period. As in each digital representation of analogous signals, information is lost if the sampling frequency is too low as stated by the Nyquist-Shannon-sampling theorem (Isermann (1987, pp. 31)). Loosely speaking, in order not to loose information, the length of a period must, according to this theorem, be shorter than half of the length of the shortest possible processing time. If the shortest possible processing time can be arbitrarily close to zero (for example, because processing times are assumed to be exponentially distributed), such a perfect sampling without any loss of information is already theoretically impossible.

However, there is an additional problem: As the processing or duration times d_{kw} are considered to be realizations of random variables, the sampled counts of processing capacities c_{kt} for the discrete time periods are also realizations of random variables. In order to provide a reasonable characterization of the workstation in a discrete time model, a sufficiently large number of workpieces must be considered and a sufficiently large number of realizations of the processing capacities c_{kt} at station k for different periods t is required.

For this reason it is clear that on the one hand one would like to have both a very high sampling frequency and a very large number of sampled processing times of different workpieces (and therefore a very large number of periods t), but on the other hand one needs to be able to solve a discrete time LP of limited size in limited time. This tradeoff will be explored in detail in our numerical study.

Table 2 Notation for the discrete time model

Sets and indices	
$k = 1, ..., K$	stations in the flow line
$t = 1, ..., T$	periods
T_0	last period of the warm-up phase
Parameters	
c_{kt}	potential processing capacity of station k in period t, realization of an integer random variable obtained by sampling
b_k	exogenously given capacity of the buffer behind station k
b_{tot}	exogenously given total buffer capacity between all stations
Real-valued decision variables	
Q_{kt}	production quantity of station k in period t
Y_{kt}	end-of-period inventory level of station k in period t
Integer decision variables	
X_k	endogenously determined capacity of the buffer behind station k
Performance measures and auxiliary quantities	
Inv_k	average inventory in buffer k and at machine k
PR	production rate estimate
Y'_{kt}	inventory in buffer k and at machine k in period t

2.3 Discrete time evaluation and optimization model

Our discrete time production-inventory model of a stochastic flow line is based on the following assumptions for the case of *given* buffer sizes:

- The flow line consists of stations $k = 1, ..., K$.
- Behind each but the last station there is a buffer that can hold b_k parts. This buffer size is exogenously given. The material supply to the first station and the space behind the last station is unlimited.
- Time is divided into discrete periods t of equal length.
- The maximum number of parts that can be processed at station k in period t is c_{kt}. It is the realization of a stochastic counting process obtained via sampling, see Section 2.2.
- The objective is to maximize the production rate of the system. The production rate is the number of workpieces processed at the last station divided by the length of the observation period. The observation period starts after the first t_0 periods (warm-up phase).

Using the above assumptions and the notation in Table 2, the (evaluation) model in discrete time can be stated as follows:

$$\text{Max} \sum_{k=1}^{K} \sum_{t=1}^{T} (10T - t) Q_{kt} \tag{6}$$

subject to

$$Y_{k,t-1} + Q_{kt} = Y_{kt} + Q_{k+1,t+1}, \qquad k = 1, ..., K, t = 1, ..., T, \tag{7}$$

$$Q_{kt} \leq c_{kt}, \qquad k = 1, ..., K, t = 1, ..., T, \tag{8}$$

$$Q_{kt} = 0, \qquad k = 1, ..., K, t < k, \tag{9}$$

$$Y_{kt} \leq b_k, \qquad k = 1, ..., K, t = 1, ..., T, \tag{10}$$

In the objective function (6), the total production is maximized. The coefficient $10T - t$ in the objective function ensures that early production at each station is rewarded in order to model a push system. Equations (7) state that the end-of-period buffer level Y_{kt} of stage k at period t is the buffer level from the previous period plus the current production quantity of stage k minus the production quantity at the next stage $k+1$ in the next period $t+1$. It is a standard inventory equation for a dynamic multi-stage production system with the convention that those variables that are not defined (e.g., "$Q_{K+1,T+1}$") are omitted. The constraint (8) states that the production capacity for each period and station may not be exceeded. Production at downstream production stages can only start if material can be available. Due to the lead time of one period between adjacent production stages in Equations (7), Station k cannot start production earlier than in period $t = k$, see Equations (9). Equations (10) state that the inventory level must not exceed the buffer capacity. The model has $2 \cdot K \cdot T$ decision variables and $3 \cdot K \cdot T + (K-1)K/2$ constraints.

The production rate PR in the observation period is determined as the number of workpieces processed at the last station after a warm-up phase of T_0 periods divided by the number of periods after this warm-up phase:

$$PR = \frac{1}{T - T_0} \cdot \sum_{t=T_0+1}^{T} Q_{Kt} \tag{11}$$

Note that the end-of-period inventory Y_{kt} as used in balance equations (7) does not account for workpieces at the machines. We therefore determine from the solution of the model a modified inventory Y'_{kt} which includes the inventory at the machines as the difference of cumulated production of adjacent machines

$$Y'_{kt} = \sum_{\tau=1}^{t} Q_{kt} - \sum_{\tau=1}^{t} Q_{k+1t} \tag{12}$$

and the average inventory Inv_k related to buffer k as

$$Inv_k = \frac{1}{T - T_0} \cdot \sum_{t=T_0+1}^{T} Y'_{kt} \tag{13}$$

for the periods following the warm-up phase. In this base (evaluation) variant of the discrete-time model, we assume that the buffer allocation is given. We now turn to the buffer optimization variant of the model and assume that only the *total* number b_{tot} of buffer spaces is given. We further assume that these buffer spaces can be freely located between the machines and the production-rate maximizing buffer allocation is sought. Then the constraints (10) have to be replaced by the following two constraints:

$$Y_{kt} \leq X_k, \qquad k = 1, ..., K-1, t = 1, ..., T \tag{14}$$

$$\sum_{k=1}^{K-1} X_k = b_{tot}, \tag{15}$$

The constraints (14) state that the end-of-period inventory level must not exceed the now endogenous buffer size X_k. The equation (15) guarantees that all the available buffer spaces are allocated in the line. This minor modification allows to incorporate the buffer allocation problem and hence to optimize the flow line. If we aim at the production-rate maximizing buffer allocation, the original linear program (6)-(10) therefore turns into a mixed-integer program as buffer sizes must be integer.

2.4 Error sources of simulation and optimization in discrete time

Our approach to simulate and optimize flow lines using linear programming models in discrete time leads to two conceptually distinct types of errors:

- **Simulation error:** Like a conventional simulation of a flow line in continuous time, our approach is based on a limited number of realizations of random variables, i.e., a sample path, that models processing times. Therefore any production rate (or inventory level) estimate for the flow line is also a realization of a random variable. In order to get tight confidence intervals for the estimates of these random variables, it may be necessary to perform long runs that simulate the processing of many work pieces. For large flow lines with many machines or for effective processing times with a very high coefficient of variation (including potential repair times, Gaver (1962)), this can lead to linear programs that are still too large to be solved with the hard- and software currently available if precise performance measures are required.
- **Time discretization errors:** In the inventory balance equations (7), a production quantity processed at station k in period t cannot be processed earlier than in period $t+1$ at the next station $k+1$. In general, this induces an artificial time lag between the moment a workpiece has completed its operation at one station and is available to start production at the succeeding station. Therefore, blocking and starving of stations (and also the inventory level) is implicitly overestimated in the discrete time model and hence the production rate is underestimated. This effect should increase as the period length increases.

While the simulation error always occurs, the time discretization error occurs “only” if the effective processing times are not integer multiples of the period length. If the effective processing times follow a distribution in continuous time, both error types always occur.

One might consider to isolate these two error types by using an identical sample path of effective processing times in a continuous time simulation and in a discrete time simulation and attribute any difference of the estimated performance measures solely to the time discretization error. In this approach, one would have to solve for the same sample path the continuous time linear program (1) - (5) and, after conversion of processing times into production capacities, the discrete time linear program (6) - (10).

However, due to time discretization it is not possible to use exactly the same sample path of effective processing times in both models and therefore it is also not possible to truly isolate the two error types using this approach. Assume, for example, that in a two-machine line the second machine operates faster than the first machine. If production capacities are sampled for T periods in the discrete time model, then more processing times have to be sampled for the second machine than for the first. However, the second machine cannot process more parts than the first machine. In the continuous time simulation, the required number of realizations of processing times is identical for all stations and does not depend on the speed of the machines. In general, the solution to the discrete time model is therefore based to some extent on partially different sets of realizations of random variables. If non-identical sample paths of processing times have to be used in a specific continuous time simulation and a “corresponding” discrete time simulation, then it is impossible to isolate the simulation error from the time discretization error in the attempt to explain the differences in the simulation results.

Even if all machines operate on average at the same speed, but buffer sizes are limited and processing times are realizations of random variables, a simulation in continuous time will report some blocking and starving for a simulation run and the number of sampled processing times in this continuous time simulation still equals the number of processed work pieces. However, in order to simulate this line over the same time horizon in discrete time, again a *larger* number of sampled processing times is needed to determine production capacities c_{kt} and therefore there is no consistent one-to-one mapping of sampled processing times to workpieces in both the continuous *and* the discrete time simulation. It is therefore impossible to use identical sample paths in both simulations and attribute differences of the estimated performance measures directly to the time discretization error.

To make things worse, only few exact results for the performance analysis of stochastic flow lines with limited buffer capacity are available. For this reason, one may be forced to compare the results from the discrete time simulation to results from a separate continuous time simulation to assess the quality of the discrete time approach. Then the differences in the performance measure estimates are due to

i) the simulation error of the continuous time simulation,
ii) the simulation error of the discrete time simulation and
iii) the time discretization error.

3 Numerical evaluation of the approximation

3.1 Outline of the numerical study

In order to evaluate the accuracy and the numerical effort of our method, we performed a numerical study. The measure of accuracy used in the study is the relative deviation of the production rate and the inventory level estimates of the discrete time linear program from the true value (or a hopefully precise estimate of this true value). In the design of our numerical experiments we tried to reflect the errors and problems mentioned in Section 2.4 that are due to simulation in discrete time, to time discretization and to simulation in continuous time (in order to determine reference values).

In the first part of the study we start by treating Markovian two-machine lines for which exact results are available. This way we can avoid errors that are due to simulation in continuous time in order to determine the "true" values of the performance measures. Such a system can be easily analyzed via a $M/M/1/N$ queueing model in either discrete or continuous time. The idea is to interpret the arrival process of the queueing model as the production process of the first machine of the line and the service process of the queueing model as the production process of the second machine. The system size N represents the number of buffer spaces between the machines plus the space at the second machine.

In our first step, we try to isolate those errors that are solely due to simulation in discrete time from the time discretization error as described in Section 2.4. To this end, we start with a discrete time Markovian model of a flow line where the time discretization error cannot occur as all processing times are geometrically distributed multiples of the period length. In the next step, we consider a continuous time Markovian model of a two-machine line with exponentially distributed processing times. The comparison with the exact results lumps together the discrete time simulation error and the time discretization error as explained in Section 2.4.

The second part of our numerical study is devoted to longer lines. Only very limited results are available for lines with more than two machines, see Gershwin und Schick (1983) and Tan (2003) as examples for the rare exceptions. For this reason we compare our simulation in discrete time to a separate simulation in continuous time and are aware of the three above-mentioned sources of differences between the discrete time simulation and the continuous time simulation.

In our study, we ask for the impact of the following features of problem instances on the accuracy of our method:

- Number of periods in the discrete time model and (average) processing rates at the machines
- Number of buffer spaces for each buffer between the machines in the flow line
- Location of the bottleneck (if any) in the line

For lines with more than two machines, we added the following problem features:

- Number of stations
- Variability of the processing times
- Exogenously given even distribution of buffer spaces vs. endogenously determined (production rate maximizing) allocation of buffer spaces

The processing rates of the machines and the number of periods in the discrete time linear program determine the number of work pieces that are processed, i.e., the total production quantity. We conjecture that the accuracy of our discrete time simulation should increase with that quantity, just as it would in continuous time.

However, a given number of (potentially) processed workpieces can be modeled in discrete time by different combinations of period numbers and processing rates. In order to minimize

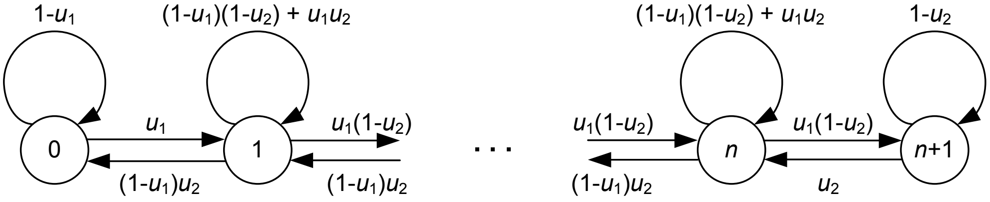

Fig. 2 Transition diagram of a discrete time Markov chain for a two-machine line

the time discretization error, it is attractive to use many (short) periods and low processing rates. Unfortunately, this increases the size of the linear program. On the other hand, if few (long) periods and high processing rates are used, so that the resulting discrete time linear programs are smaller and easier to solve, the time discretization error is bigger and hence the results might be less accurate.

To create comparable conditions in our experiments, we set the number of simulated periods (after a warm-up period) and the processing rate of the machines so that comparable expected numbers of workpieces could be processed by the line (see below).

With respect to the buffer size we expect to find more precise results for problem instances with larger buffers. Large buffers reduce blocking and starving. For larger buffers, the modeling discrepancy between the discrete flow of material in the real system and the continuous flow in the linear program should hence be less relevant.

It was not clear to us whether the location of a potential bottleneck in the system has a significant effect on the accuracy of the method.

For lines with more than two machines, we compare our results to those from a discrete-event simulation model originally coded in C (Helber, 1999). We expect to find larger deviations as the number of stations increases, because the number of “modeling defects” due to the discretization increases. With respect to the variability of the processing times, we conjecture to find an increasing accuracy with decreasing variability as the production rate of a zero-variability discrete material flow line can be determined exactly via a continuous flow model. It was not clear to us whether the method should be more precise if the (even) buffer allocation is exogenously given or an uneven distribution endogenously determined so that the production rate is maximized.

For all of these parameter types, we systematically explore a range of parameter values which we consider to be relevant, in order to find out under which conditions the method appears to yield reasonably precise production rate estimates.

3.2 Comparison with exact results for Markovian two-machine lines

3.2.1 Geometrically distributed processing times In this first step of the numerical study, we consider a two-machine line with discrete processing times that are integer multiples of the period length used in the discrete time linear program. In this situation a time discretization error does not occur.

To create a Markovian model that can be analyzed exactly, we assume that during each time period a machine $i \in \{1, 2\}$ that is neither blocked nor starved completes its operation on a workpiece with a probability $0 < u_i < 1$. Then processing times are geometrically distributed with mean $\frac{1}{u_i}$ and the squared coefficient of variation (SQV) is $1 - u_i$ (Tran-Gia, 1996, p. 39). If the buffer can hold n workpieces and one workpiece can occupy the second machine, the corresponding Markov chain has $n + 2$ states describing the number of workpieces in the buffer and on the second machine. Figure 2 shows the transition diagram for this system. It is straightforward to determine the steady-state probabilities of this Markov chain and from these the expected values of the production rate and the inventory level, see, e.g., Gershwin (1994).

We used the testbed presented in Table 3. The processing probabilities u_i of the machines are the product of the base processing probability (for example, 0.5 in the first base processing probability case) and the bottleneck factor for the respective machine (e.g. 0.8 for machine M_1 in bottleneck factor case 1, leading to a production probability

Table 3 Test bed for the analysis of two-machine lines with geometrically distributed processing times

Parameter type	Cases	Parameter value per case
Buffer spaces per buffer	4	1, 4, 8, 16
Base processing probabilities (corr. SQV)	2	0.5 (0.5), 0.9 (0.1)
Bottleneck factor	3	(M$_1$: 0.8; M$_2$: 1.0), (M$_1$: 1.0; M$_2$: 1.0), (M$_1$: 1.0; M$_2$: 0.8)

Table 4 Results for geometrically distributed processing times

u_1	u_2	BS	PR*	PR^{ds}	RDP	Inv*	Inv^{ds}	RDI
0.40	0.50	1	0.33	[0.32, 0.32, 0.33]	-0.6	0.8	[0.7, 0.7, 0.7]	-13.6
0.40	0.50	4	0.39	[0.38, 0.38, 0.39]	-0.2	1.7	[1.5, 1.6, 1.7]	-3.6
0.40	0.50	8	0.40	[0.39, 0.40, 0.41]	0.1	2.2	[1.9, 2.1, 2.2]	-5.0
0.40	0.50	16	0.40	[0.39, 0.40, 0.40]	-0.9	2.4	[2.1, 2.4, 2.6]	-1.4
0.50	0.50	1	0.38	[0.37, 0.38, 0.38]	0.1	1.0	[0.9, 0.9, 0.9]	-13.5
0.50	0.50	4	0.45	[0.44, 0.45, 0.45]	-0.4	2.5	[2.3, 2.4, 2.5]	-3.7
0.50	0.50	8	0.47	[0.46, 0.47, 0.47]	-1.3	4.5	[4.2, 4.5, 4.8]	0.0
0.50	0.50	16	0.49	[0.47, 0.48, 0.49]	-0.3	8.5	[8.0, 8.8, 9.6]	3.9
0.50	0.40	1	0.33	[0.31, 0.32, 0.33]	-1.8	1.2	[0.9, 0.9, 0.9]	-21.1
0.50	0.40	4	0.39	[0.38, 0.38, 0.39]	-0.7	3.3	[3.1, 3.1, 3.2]	-5.7
0.50	0.40	8	0.40	[0.39, 0.39, 0.40]	-0.8	6.8	[6.7, 6.8, 7.0]	0.2
0.50	0.40	16	0.40	[0.39, 0.40, 0.40]	-0.4	14.6	[14.2, 14.5, 14.8]	-0.9
0.72	0.90	1	0.68	[0.70, 0.71, 0.71]	4.3	0.8	[0.9, 0.9, 0.9]	14.3
0.72	0.90	4	0.72	[0.71, 0.72, 0.72]	-0.5	1.1	[1.1, 1.1, 1.2]	3.1
0.72	0.90	8	0.72	[0.71, 0.72, 0.73]	0.3	1.1	[1.1, 1.1, 1.2]	0.9
0.72	0.90	16	0.72	[0.71, 0.72, 0.72]	-0.3	1.1	[1.1, 1.1, 1.2]	1.9
0.90	0.90	1	0.83	[0.86, 0.86, 0.87]	4.3	1.0	[1.3, 1.3, 1.4]	34.2
0.90	0.90	4	0.88	[0.88, 0.88, 0.89]	0.5	2.5	[2.5, 2.7, 2.9]	7.7
0.90	0.90	8	0.89	[0.88, 0.89, 0.89]	0.0	4.5	[4.0, 4.6, 5.3]	3.2
0.90	0.90	16	0.89	[0.89, 0.89, 0.90]	-0.3	8.5	[7.3, 9.7, 12.1]	14.0
0.90	0.72	1	0.68	[0.69, 0.70, 0.71]	3.3	1.2	[1.5, 1.5, 1.5]	24.8
0.90	0.72	4	0.72	[0.71, 0.72, 0.73]	-0.1	3.9	[4.3, 4.3, 4.4]	10.7
0.90	0.72	8	0.72	[0.71, 0.72, 0.73]	0.2	7.9	[8.2, 8.3, 8.3]	5.2
0.90	0.72	16	0.72	[0.72, 0.72, 0.73]	0.4	15.9	[16.2, 16.3, 16.3]	2.3

of $u_1 = 0.5 \cdot 0.9 = 0.40$ for machine M$_1$). The corresponding squared coefficient of variation is given in brackets behind each base processing probability. These production probabilities were then used to sample production capacities c_{kt} required in Equations (8) as illustrated in Figure 1.

The different combinations of parameters led to a full factorial design of the experiment with $4 \cdot 2 \cdot 3 = 24$ different two-machine lines that were both analyzed via our numerical method and via the exact solution of the discrete time Markov chain model depicted in Figure 2.

We decided to set the number of periods in our discrete time simulation model (6) - (10) so that the production of a comparable number of work pieces could be modeled after an initial warm-up period of $T_0 = 500$ periods. This number of was set to 10,000 work pieces. The number of periods in the discrete time linear program was hence determined as follows:

$$T = T_0 + T_1 = 500 + \lceil 10,000 \max\{\frac{1}{u_1}, \frac{1}{u_2}\}\rceil \qquad (16)$$

The T_1 periods after the warm-up phase were subdivided into 10 time segments of identical lengths in order to compute rough 95% confidence intervals for the performance measures by treating these time segments as if they were independent.

In Table 4 we report for each combination of processing probabilities $u_i, i \in \{1, 2\}$ and buffer size (BS) the true (expected) production rate (PR*) and the true (expected) inventory

Table 5 Test bed for the analysis of two-machine lines with exponentially distributed processing times

Parameter type	Cases	Parameter value per case
Buffer spaces per buffer	4	1, 4, 8, 16
Base processing rates	3	0.5, 1.0, 2.0
Bottleneck factor	3	(M_1: 0.8; M_2: 1.0), (M_1: 1.0; M_2: 1.0), (M_1: 1.0; M_2: 0.8)

level (Inv*) from the exact analysis of the Markovian model. From the discrete time simulation ("ds") we also report 95% confidence intervals for both the production rate and the inventory level. The three numbers in brackets are the lower limit of those 95% confidence intervals, the average, and the upper limit of those confidence intervals. Finally, we report the relative deviation (in %) of the production rate estimate (RDP) and the inventory level estimate (RDI). It was computed as (100*(discrete time simulation value - true value)/(true value)). The relative deviations of the production rate estimates are mostly quite small, in particular for buffers with four or more spaces. The inventory level estimates are less accurate. However, most of the larger relative deviations occur in cases where the absolute inventory levels are small due to small buffer sizes.

For this set of instances, the discrete time linear program has between 46452 and 102004 columns (variables) and between 58064 and 127504 constraints. The CPU time to solve the linear program using Cplex 11.0.0 on a Dual Core Pentium IV machine with 2.8 GHz and 2 GB RAM ranges from 1 to 4 seconds. The time to build the matrix of the linear program using the GAMS interpreter is always substantially longer than the time to solve the linear program.

3.2.2 Exponentially distributed processing times If the processing times at the machines are exponentially distributed, a standard continuous time $M/M/1/N$ queuing model can be used to determine the exact values of the average production rate and inventory levels. In this setting, each machine $i \in \{1, 2\}$ is characterized by its processing rate μ_i. The average processing time is $\frac{1}{\mu_i}$ and the squared coefficient of variation of the processing times is 1. Processing at the first machine is interpreted as the arrival process in the queuing model. If this system is analyzed via the discrete time linear program (6) - (10), both the discrete time simulation error and the time discretization error occur.

The testbed in Table 5 led to 36 distinct flow lines. The number of periods was again determined as shown in Equation (16) to allow for about 10,000 workpieces to be processed at the bottleneck station if it were never starved or blocked.

The results in Table 6 are organized like those for geometrically distributed processing times in the previous section. Unless buffers are very small, the discrete time simulation is apparently fairly accurate. It tends to underestimate the production rate slightly (-1.169% over the 36 cases), which may be due to the time discretization error. The inventory level estimates are larger than for the lines where the time discretization error does not occur. They appear to be smallest if the processing time is about as long as the period length in the discrete time simulation. Again, the largest relative deviations of the inventory level estimates occur if the absolute inventory levels are low.

3.3 Comparison with approximate simulation results for longer lines

To evaluate the performance of our method for longer lines we compared its results to those obtained by a discrete-event simulation for the testbed consisting of the 864 cases described in Table 7. The last line in Table 7 indicates that for each line we first evaluated an assumed even distribution of the buffer spaces in the line and then sought the production-rate maximizing buffer allocation as described in Section 2.3.

We only asked for the accuracy of the production rate estimates (as opposed to inventory level estimates) for three reasons: First, we already know from the results for two-machine

Table 6 Results for exponentially distributed processing times

μ_1	μ_2	BS	PR*	PR^{ds}	RDP	Inv*	Inv^{ds}	RDI
0.40	0.50	1	0.30	[0.25, 0.26, 0.27]	-12.5	0.9	[0.7, 0.7, 0.7]	-19.1
0.40	0.50	4	0.36	[0.36, 0.36, 0.36]	-0.9	1.9	[1.8, 1.9, 2.0]	1.0
0.40	0.50	8	0.39	[0.37, 0.38, 0.38]	-2.9	2.8	[2.7, 2.8, 3.0]	1.5
0.40	0.50	16	0.40	[0.39, 0.40, 0.40]	-0.5	3.7	[3.3, 3.8, 4.3]	3.1
0.50	0.50	1	0.33	[0.29, 0.29, 0.29]	-12.9	1.0	[0.8, 0.8, 0.8]	-21.5
0.50	0.50	4	0.42	[0.40, 0.41, 0.41]	-2.6	2.5	[2.3, 2.3, 2.4]	-6.0
0.50	0.50	8	0.45	[0.43, 0.44, 0.45]	-2.1	4.5	[4.0, 4.3, 4.7]	-3.6
0.50	0.50	16	0.47	[0.47, 0.47, 0.48]	0.1	8.5	[8.0, 8.5, 9.1]	0.2
0.50	0.40	1	0.30	[0.25, 0.26, 0.27]	-11.9	1.1	[0.8, 0.8, 0.8]	-27.7
0.50	0.40	4	0.36	[0.35, 0.36, 0.36]	-2.2	3.1	[2.8, 2.8, 2.9]	-9.3
0.50	0.40	8	0.39	[0.38, 0.39, 0.40]	0.5	6.2	[5.8, 5.9, 6.0]	-4.8
0.50	0.40	16	0.40	[0.39, 0.40, 0.40]	-0.6	13.3	[12.6, 13.0, 13.4]	-2.4
0.80	1.00	1	0.59	[0.55, 0.56, 0.58]	-4.4	0.9	[1.0, 1.0, 1.0]	15.5
0.80	1.00	4	0.73	[0.70, 0.72, 0.74]	-1.1	1.9	[2.1, 2.2, 2.3]	16.4
0.80	1.00	8	0.78	[0.76, 0.78, 0.79]	0.0	2.8	[3.0, 3.1, 3.3]	12.1
0.80	1.00	16	0.80	[0.78, 0.80, 0.82]	0.6	3.7	[3.6, 4.3, 5.0]	17.2
1.00	1.00	1	0.67	[0.64, 0.65, 0.66]	-2.8	1.0	[1.1, 1.1, 1.2]	14.7
1.00	1.00	4	0.83	[0.81, 0.82, 0.83]	-1.6	2.5	[2.8, 2.9, 2.9]	14.0
1.00	1.00	8	0.90	[0.89, 0.90, 0.91]	0.1	4.5	[4.5, 4.7, 4.9]	3.6
1.00	1.00	16	0.94	[0.91, 0.94, 0.96]	-0.9	8.5	[7.7, 8.8, 9.9]	3.9
1.00	0.80	1	0.59	[0.55, 0.57, 0.58]	-4.2	1.1	[1.1, 1.1, 1.2]	-0.1
1.00	0.80	4	0.73	[0.72, 0.73, 0.74]	0.3	3.1	[3.2, 3.2, 3.3]	3.2
1.00	0.80	8	0.78	[0.76, 0.78, 0.79]	0.3	6.2	[6.2, 6.4, 6.7]	3.8
1.00	0.80	16	0.80	[0.78, 0.79, 0.81]	-0.2	13.3	[12.4, 13.0, 13.5]	-2.7
1.60	2.00	1	1.18	[1.23, 1.25, 1.27]	5.9	0.9	[1.6, 1.6, 1.7]	93.1
1.60	2.00	4	1.46	[1.46, 1.49, 1.52]	2.3	1.9	[2.9, 2.9, 3.0]	57.0
1.60	2.00	8	1.55	[1.51, 1.54, 1.57]	-0.6	2.8	[3.7, 3.9, 4.1]	40.4
1.60	2.00	16	1.59	[1.55, 1.58, 1.62]	-0.7	3.7	[4.5, 4.9, 5.4]	34.3
2.00	2.00	1	1.33	[1.39, 1.43, 1.46]	7.1	1.0	[1.9, 1.9, 2.0]	92.2
2.00	2.00	4	1.67	[1.65, 1.68, 1.71]	0.8	2.5	[3.6, 3.7, 3.9]	49.2
2.00	2.00	8	1.80	[1.76, 1.80, 1.84]	0.0	4.5	[5.6, 5.9, 6.1]	30.0
2.00	2.00	16	1.89	[1.86, 1.91, 1.95]	0.9	8.5	[7.7, 9.0, 10.3]	6.1
2.00	1.60	1	1.18	[1.20, 1.23, 1.27]	4.4	1.1	[1.8, 1.8, 1.9]	59.7
2.00	1.60	4	1.46	[1.42, 1.45, 1.48]	-0.4	3.1	[4.0, 4.1, 4.2]	31.3
2.00	1.60	8	1.55	[1.53, 1.56, 1.58]	0.3	6.2	[7.1, 7.3, 7.5]	18.0
2.00	1.60	16	1.59	[1.57, 1.59, 1.62]	0.1	13.3	[13.9, 14.4, 14.9]	7.9

lines, that our method does not yield precise inventory level estimates. Second, while our discrete event simulations of the longer lines are all very precise with a relative width of the production rate confidence intervals below 1%, the inventory level estimates from these simulations happen to have much wider confidence intervals, often wider than 50%. We therefore don't have precise reference values for inventory levels of longer lines. Third, a financial analysis of the investment in a flow line shows that the investment in work in process of a flow line can safely be ignored if compared to those cash flows that are due to the investment in machines, buffers, and the continuous processing of material over several month or years, see Helber (1999, 2001). We are aware that this result contradicts the current "philosophy" of "lean production". However, it justifies to concentrate on precise production rate estimates as opposed to inventory level estimates.

We used the Erlang-k-distribution to generate processing times with a squared coefficient variation (SCV) below 1.0 and the balanced-mean variant of the Cox-2-distribution to generate processing times with a SCV of 1.0 or above (Buzacott und Shanthikumar, 1993, p. 542). These distributions allow to model processing times with the given coefficients of variation easily. Figure 3 shows the frequency diagram of absolute values of relative deviations. The maximum deviation was about 55% and the mean value of the absolute values of relative deviations was 8.7%. The mean value of the relative deviations of the produc-

Table 7 Test Bed for the analysis of longer lines (864 cases); "f. m." means "first machine", "l. m." means "last machine","o. m." means "other machines"

Parameter type	Number of cases	Parameter value per case
Buffer spaces per buffer	4	1, 4, 8, 16
Base processing rates	3	0.5, 1.0, 2.0
Bottleneck factor	3	(f. m.: 0.9; o. m.: 1.0), (all machines 1.0), (l. m.: 0.9; o. m.: 1.0)
Number of stations	3	5, 7, 9
Processing time variability	4	0.25, 0.5, 1.0, 2.0
Buffer allocation	2	even vs. production-rate maximizing

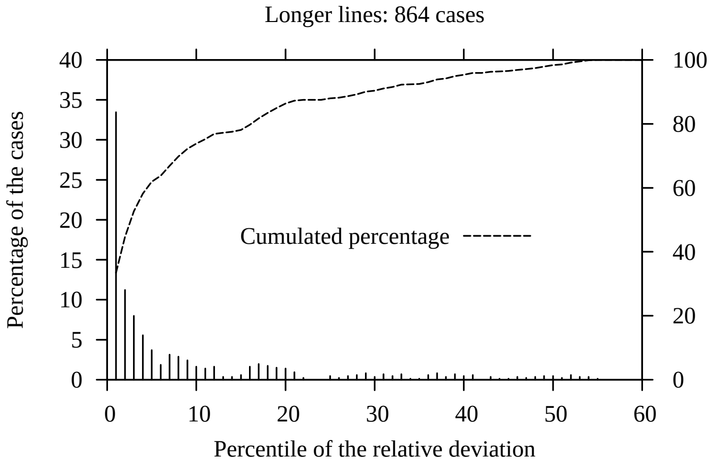

Fig. 3 Percentage of cases over percentiles of relative deviations for all 864 cases of longer lines

tion rate estimates was -8.6% which indicates that the discrete time simulation tends to underestimate the production rate, which can be explained by the time discretization error.

In Tables 8 to 12 we report the impact of the flow line parameters introduced in Table 7. We always report the average of the relative deviation of the production rate estimate (RelDevPR), the respective average over absolute values of relative deviations (AbsRelDevPR) and the CPU time (in seconds) to solve the linear program. The upper part of the tables report the results for a given (even) distribution of the buffer spaces in the line and the lower part the results for the production rate maximizing (optimized) buffer allocation.

The results in Table 8 show that the method is quite inaccurate for very small buffer sizes. The accuracy increases with the buffer sizes. The linear programs for systems with small buffers appear to be particularly difficult to solve.

Table 9 suggests that the method is both faster and more accurate for higher base production rates. As for the two-machine lines (see Equation (16)), the number of periods was adjusted to allow for an expected value of 10,000 workpieces produced at the bottleneck machine of each line if it were operating in isolation. Therefore, higher base processing rates lead to smaller linear programs which explains the shorter CPU times, but not the apparent higher accuracy of the method. Due to the time discretization error, one would rather expect more accurate results for cases with lower base production rates. We do not have any explanation for this observation.

The location of the bottleneck does not appear to have a major impact on the performance of our method, see Table 10. However, the number of stations in the flow line is important. Table 11 shows that the computation times increase and the accuracy decreases with the number of stations. The variability of the processing time is very important, see

Table 8 Impact of the number of buffer spaces per buffer

Buffer spaces per buffer	1	4	8	16
Even buffer allocation:				
RelDevPR [%]	-25.3	-6.0	-2.4	-0.7
AbsRelDevPR [%]	25.3	6.1	2.4	0.9
CPU [sec.]	80.5	20.5	23.7	25.3
Optimized buffer allocation:				
RelDevPR [%]	-25.2	-6.1	-2.3	-0.7
AbsRelDevPR [%]	25.2	6.1	2.4	0.9
CPU [sec.]	285.8	86.1	64.3	59.4

Table 9 Impact of the base processing rate

Base processing rate	0.5	1.0	2.0
Even buffer allocation:			
RelDevPR [%]	-11.4	-8.9	-5.5
AbsRelDevPR [%]	11.4	9.0	5.6
CPU [sec.]	58.4	37.5	16.6
Optimized buffer allocation:			
RelDevPR [%]	-11.5	-8.9	-5.4
AbsRelDevPR [%]	11.5	8.9	5.5
CPU [sec.]	187.5	120.7	63.5

Table 10 Impact of the bottleneck location

Bottleneck location	f.m.	bal. line	l.m.
Even buffer allocation:			
RelDevPR [%]	-8.6	-8.6	-8.6
AbsRelDevPR [%]	8.7	8.7	8.6
CPU [sec.]	40.6	30.8	41.1
Optimized buffer allocation:			
RelDevPR [%]	-8.5	-8.7	-8.5
AbsRelDevPR [%]	8.6	8.7	8.6
CPU [sec.]	131.0	102.7	137.9

Table 11 Impact of the number of stations in the line

Stations	5	7	9
Even buffer allocation:			
RelDevPR [%]	-7.9	-8.7	-9.2
AbsRelDevPR [%]	8.0	8.8	9.3
CPU [sec.]	16.0	34.5	62.1
Optimized buffer allocation:			
RelDevPR [%]	-7.9	-8.7	-9.2
AbsRelDevPR [%]	7.9	8.7	9.3
CPU [sec.]	60.5	115.0	196.2

Table 12. The accuracy of the method decreases substantially as the squared coefficient of variation of the processing times increases. The computation times, however, appear to decrease with increasing variability.

A consistent result over Tables 8 to 12 is that the accuracy of our method does not appear to depend on whether the buffer allocation is given or endogenously determined. The time to determine a production-rate maximizing buffer allocation is on average three to four times the time to evaluate the flow line for a given buffer allocation. This is a significant result as it suggests that our method can be used to quickly optimize a flow line, at least with respect to the buffer allocation. If one uses a discrete event simulation to evaluate a given buffer allocation, one can expect to simulate substantially more than three or four different buffer allocations.

Table 12 Impact of the squared coefficient of variation

SCV	0.25	0.5	1.0	2.0
Even buffer allocation:				
RelDevPR [%]	-3.1	-5.2	-9.4	-16.8
AbsRelDevPR [%]	3.2	5.3	9.4	16.8
CPU [sec.]	54.5	43.1	31.1	21.4
Optimized buffer allocation:				
RelDevPR [%]	-3.0	-5.2	-9.4	-16.7
AbsRelDevPR [%]	3.1	5.3	9.4	16.7
CPU [sec.]	151.2	136.2	110.1	98.0

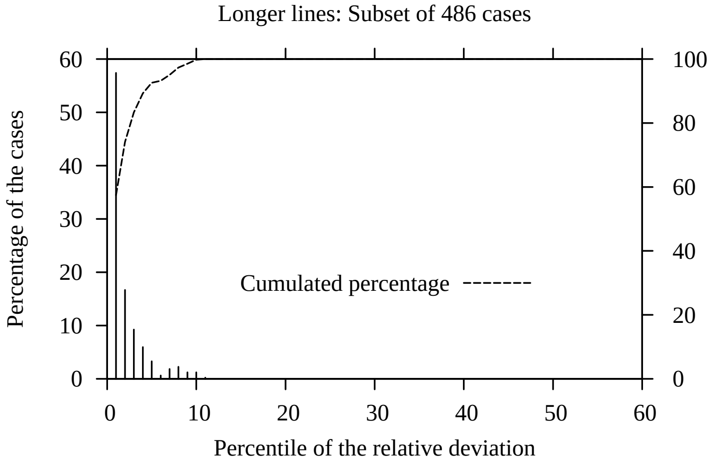

Fig. 4 Percentage of cases over percentiles of relative deviations for the subset of 486 cases of longer lines

In a final step of our numerical analysis we studied a subset of the testbed in Table 7. To create this subset, we excluded all the lines with only one buffer space per buffer and all the lines with a squared coefficient of variation of 2.0. This led to 486 different lines. Figure 4 reports the absolute values of relative deviations. For this subset, the maximum absolute value of the relative deviation of the production rate estimate was less than 11%, the average over these absolute values was 1.64% and the average over the relative deviations was -1.51%. Unless buffer sizes are very small or processing times are highly variable, our method produces reasonably accurate results.

The discrete time linear program had up to 432,000 columns and 591,000 rows, the maximum CPU time to solve the program to an optimality gap of at most 0.5% using Cplex 11.0 on a (different) 3.0 GHz Dual Core Pentium 4 PC with 4 GB of RAM was 1071 seconds.

In our eyes the main benefit from the linear programming approach is the ability to simulate and optimize a flow line simultaneously. As long as one is only interested in the evaluation of a single configuration, a conventional discrete-event simulation in continuous time might be just as efficient. However, this changes if the line has to be optimized. In this paper, we only addressed the question of an optimized buffer allocation and found that this increased the computation times only by a factor of three or four. One might therefore as well decide about different configurations of machines. Then our approach appears to be valuable due to its flexibility and the limited additional computation time for the optimization. The alternative to combine a discrete-event simulation with optimization will usually lead to very long computation times due to the combinatorial nature of the optimization problems, see Gershwin und Schor (2000).

4 Conclusion and further research

In this paper we presented a novel approach to incorporate simulation into linear programming optimization models of flow lines with limited buffer capacity. The key idea was to use a discrete-time modeling framework and to transfer sampled processing times of workpieces into sampled processing capacities of workstations. Within a mixed integer programming software, a flow line can be simulated without using a dedicated discrete event simulation package. All that is needed is a random number generator to create a stream of realizations of stochastic processing times, which are turned into sampled processing capacities. The advantage of the method is that it allows to incorporate the buffer allocation problem into the analysis and optimization of a flow line. The method yields reasonably precise production rate estimates, unless the buffers between the machines are very small and/or the variability of the (effective) processing times at the machines is rather high. However, a system with both a very high variability of the effective processing times and very few buffer spaces will usually not operate efficiently anyway. In an economically efficient flow line, the bottleneck is rarely starved or blocked and under these conditions our method generally performs well. We conclude that it may be a powerful tool to analyze and optimize flow lines with low to moderate processing time variability, in particular, if the available computation power keeps increasing rapidly.

We are currently extending this work to flow lines with closed loops, for example due to a ConWiP production control system. First results are promising. It should also be possible to analyze re-entrant lines using our method. Based on these results we will extend our analysis to the investment problem of designing lines such that the net present value from the investment is maximized (Helber, 2001), including the decision about alternative machines for the production stages.

References

Abdul-Kader, W. (2006). Capacity improvement of an unreliable production line–ananalytical approach. *Computers & Operations Research 33*, 1695–1712.

Altiok, T. (1996). *Performance Analysis of Manufacturing Systems.* New York et al.: Springer.

Buzacott, J. A. und J. G. Shanthikumar (1993). *Stochastic Models of Manufacturing Systems.* Englewood Cliffs, NJ: Prentice Hall.

Dallery, Y. und S. B. Gershwin (1992). Manufacturing flow line systems: A review of models and analytical results. *Queuing Systems Theory and Applications 12*(1-2), 3–94. Special issue on queuing models of manufacturing systems.

Gaver, D. (1962). A waiting line with interrupted service, including priorities. *Journal of the Royal Statistical Society 24*, 73–90.

Gershwin, S. und I. Schick (1983). Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers. *Operations Research 31*(2), 354–380.

Gershwin, S. B. (1994). *Manufacturing Systems Engineering.* Englewood Cliffs, New Jersey: PTR Prentice Hall.

Gershwin, S. B. und J. E. Schor (2000). Efficient algorithms for buffer space allocation. *Annals of Operations Research 93*, 117–144.

Helber, S. (1999). *Performance Analysis of Flow Lines with Non-Linear Flow of Material*, Band 473 of *Lecture Notes in Economics and Mathematical Systems.* Berlin et al.: Springer-Verlag.

Helber, S. (2001). Cash-flow-oriented buffer allocation in stochastic flow lines. *International Journal of Production Research 39*, 3061–3083.

Helber, S. und K. Henken (2007). Profit-oriented shift scheduling of inbound contact centers with skills-based routing, impatient customers, and retrials. Technical Report dp-379, Leibniz Universitaet Hannover, Institut fuer Produktionswirtschaft, Koenigsworther Platz 1, D-30167 Hannover.

Ho, Y., M. Eyler, und T. Chien (1979). A gradient technique for general buffer storage design in production line. *International Journal of Production Research 17*(6), 557–580.

Isermann, R. (1987). *Digitale Regelsysteme. Band I: Deterministische Regelungen.* Berlin, Heidelberg, New York: Springer.

Johri, P. K. (1987). A linear programming approach to capacity estimation of automated production lines with finite buffers. *International Journal of Production Research 25*, 851–866.

Kelton, W. D., R. P. Sadowski, und D. T. Sturrock (2006). *Simulation with Arena with CDROM* (4 Aufl.). McGraw Hill Higher Education.

Law, A. M. und W. D. Kelton (1991). *Simulation Modeling and Analysis* (2 Aufl.). New York et al.: McGraw-Hill.

Liberopoulos, G., C. T. Papadopoulos, B. Tan, J. M. Smith, und J. M. Gershwin (Hrsg.) (2006). *Stochastic Modeling of Manufacturing Systems. Advances in Design, Performance Evaluation, and Control Issues*, Berlin, Heidelberg, New York. Springer.

Matta, A. und R. Chefson (2005). Formal properties of closed flow lines with limited buffer capacities and random processing times. In: J. M. Felix-Teixera und A. E. C. Brito (Hrsg.), *The 2005 European Simulation and Modelling Conference*. Porto. S. 190–198.

Schruben, L. W. (2000). Mathematical programming models of discrete event system dynamics. In: J. A. Joines, R. R. Barton, K. Kang, und P. A. Fishwick (Hrsg.), *Proceedings of the 2000 Winter Simulation Conference*, S. 381–385.

Swain, J. J. (2007). Simulation software survey. *OR/MS Today 34*(5).

Tan, B. (2003). State-space modeling and analysis of pull-controlled production systems. In: S. B. Gershwin, Y. Dallery, C. T. Papadopoulos, und J. MacGregor Smith (Hrsg.), *Analysis and Modeling of Manufacturing Systems*, Chapter 15, S. 363–398. Kluwer.

Tijms, H. C. (1994). *Stochastic Models*. Chichester: Wiley.

Tran-Gia, P. (1996). *Analytische Leistungsbewertung verteilter Systeme. Eine Einführung.* Springer.