Helber, Stefan; Schimmelpfeng, Katja; Stolletz, Raik; Lagershausen, Svenja

Working Paper

Using linear programming to analyze and optimize stochastic flow lines

Diskussionsbeitrag, No. 389

Provided in Cooperation with:
School of Economics and Management, University of Hannover

Suggested Citation: Helber, Stefan; Schimmelpfeng, Katja; Stolletz, Raik; Lagershausen, Svenja (2008) : Using linear programming to analyze and optimize stochastic flow lines, Diskussionsbeitrag, No. 389, Leibniz Universität Hannover, Wirtschaftswissenschaftliche Fakultät, Hannover

This Version is available at:
https://hdl.handle.net/10419/27199

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

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 and 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 and Schor, 2000). However, the mathematical assumptions required for these analytic models often restrict their use in practice. In addition, even a slight modification of such an analytic model may easily extend 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 contribu-

tion 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 unmanageable. Law and Kelton (1991) and Kelton et al. (2006), among others, give introductions to the methodology and describe simulation models of flow lines. A survey of the literature on analytic queueing models of flow lines with limited buffer capacity is given by Dallery and 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 and Shanthikumar, 1993; Gershwin, 1994; Tijms, 1994; Altiok, 1996). The literature on linear-programming based simulation of flow lines 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). The buffer capacity in the model by Abdul-Kader (2006) determines an upper limit of a summation index. Therefore, 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 and Chefson (2005) which is based on an earlier model by Schruben (2000). In the model of a closed flow line by Matta and 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 and 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 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 and Chefson (2005):

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

subject to

$$XS_{kw} + d_{kw} = XF_{kw} - B_{kw}, \qquad \forall k,\ \forall w \tag{2}$$
$$XS_{k+1,w} = XF_{kw} + W_{kw}, \qquad \forall k \leq K-1,\ \forall w \tag{3}$$
$$XS_{k,w+1} = XF_{k,w} + S_{k,w+1}, \qquad \forall k,\ \forall w \leq W-1 \tag{4}$$
$$XF_{k,w+b_k} \geq XS_{k+1,w}, \qquad \forall k \leq K-1,\ \forall w \leq W-b_k \tag{5}$$

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 Equation (2). Equation (3) defines the waiting time W_{kw} of workpiece w in the buffer behind station k. The starting time of workpiece $w+1$ is determined in Equation (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 Equation (5) states 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 and 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 and 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.

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

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

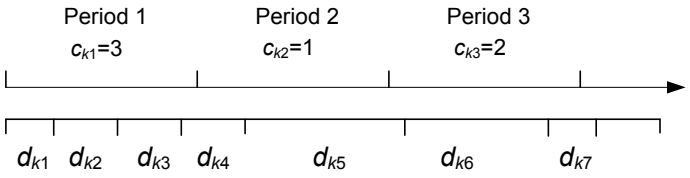

Fig. 1 Sampling of discrete time processing rates

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 processing rates 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 rates 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.

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).

Table 2 Notation for the discrete time model

Sets and indices	
$k = 1, ..., K$	stations in the flow line
$t = 1, ..., T$	periods
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
PR	production rate estimate
Integer decision variables	
X_k	endogenously determined capacity of the buffer behind station k

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

$$\text{Max } PR = \frac{1}{T - t_0} \cdot \sum_{t=t_0+1}^{T} Q_{Kt} \tag{6}$$

subject to

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

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

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

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

The production rate in the observation period is determined in the objective function (6). It is computed 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. Equation (7) 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 when material can be available, see Equation (9). Equation (10) states the the inventory level must not exceed the buffer capacity.

In this base (evaluation) variant of the discrete-time model, we assume that the buffer allocation is given. We now turn to the buffer optmization 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 constraint (10) has to be replaced by the following two constraints:

$$Y_{kt} \leq X_k, \qquad k = 1, ..., K-1 \tag{11}$$

$$\sum_{k=1}^{K-1} X_k = b_{tot}, \tag{12}$$

The constraint (11) states that the end-of-period inventory level must not exceed the now endogenous buffer size X_k. The equation (12) 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.

3 Numerical evaluation of the approximation

3.1 Outline of the numerical study

In order to evaluate the accuracy of our method, we performed a numerical study. The measure of accuracy used in the study is the relative deviation of the production rate estimate PR from Equation (6) of the linear program from the true value (or a hopefully precise estimate of this true value). In the first part of this study, we compare our results to exact results for reliable two-machine one-buffer systems with exponential processing times. Such a system can be easily analyzed via the $M/M/1/N$ queueing model. 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 is the number of buffer spaces between the machines plus the space at the second machine. It appears natural to start with two-machine flow lines as these are the smallest possible flow lines with limited buffer capacity. This initial analysis should provide first insights which help to design experiments for the analysis of the method for flow lines with more than two machines.

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

- Number of periods in the discrete time linear program
- Number of buffers spaces for each buffer between the machines in the flow line
- (Average) processing rates at the machines
- Location of the bottleneck (if any) in the line

We conjectured that the accuracy should increase with the number of periods in the model as the "simulation run" gets longer. With respect to the buffer size we expected to find more precise results for problem instances with larger buffers. The modeling discrepancy between the discrete flow of material in the real system and the continuous flow in the linear program should be less relevant, if blocking and starving is reduced due to larger buffers. The impact of the processing rate at the workstations is more subtle to analyze. On the one hand, very low rates (or long processing times) lead to very small numbers of events per period, so that the sampling error due to the discretization (see Section 2.2) is minimized. The accuracy of the method should hence increase as the processing rates decrease. On the other hand, if for a *given* number of periods the processing rate decreases, the number of events within the studied time also decreases. This makes the production rate estimate more variable and less accurate. For a given number of periods that can be treated within the linear program, we therefore expect to find a processing rate that maximizes the accuracy of the method by balancing these two opposing effects. 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.

Exact results are not available for flow lines with limited buffer sizes and more than two machines. For this reason, we compare our results to those from a discrete-event simulation model coded in C (Helber, 1999). In this second part of the study, we added the following problem features:

- Number of production stages
- Variability of the processing times
- Allocation of the buffer spaces

We expected to find larger deviations as the number of production stages increases, because the number of "modeling defects" due to the discretization increases. With respect to the variability of the processing times, we conjectured 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 when the buffer allocation is exogenously given or endogenously determined.

For all of these parameter types, we systematically explored a wide range of parameter values which we considered to be relevant, in order to find out under which conditions the method appears save to use or not.

Table 3 Test bed for the analysis of two-machine lines

Parameter type	Number of cases	Parameter value per case
Number of periods	5	2500, 5000, 10000, 20000, 40000
Buffer spaces per buffer	6	1, 2, 4, 8, 16, 32
Base processing rates	3	0.1, 1.0, 10.0
Bottleneck factor	5	(M_1: 0.7; M_2: 1.0), (M_1: 0.9; M_2: 1.0), (M_1: 1.0; M_2: 1.0), (M_1: 1.0; M_2: 0.9), (M_1: 1.0; M_2: 0.7)

Table 4 Impact of the number of periods

Number of periods	2500	5000	10000	20000	40000
MAD [%]	6.2	6.0	5.2	5.2	5.0

Table 5 Impact of the number of buffer spaces per buffer

Number of buffers	1	2	4	8	16	32
MAD [%]	16.3	8.4	3.5	2.1	1.4	1.4

3.2 Comparison with exact results for two-machine lines

In this first part of the numerical study, we used the design of the test bed presented in Table 3. In each case, we added $t_0 = 500$ time periods to those reported in Table 3 as a "warm-up" phase. The processing rates of the machines are the product of the base processing rate (for example, 0.1 in the first base processing rate case) and the bottleneck factor for the respective machine (e.g. 0.9 for machine M_1 in bottleneck factor case two, leading to a production rate of $0.1 \cdot 0.9 = 0.09$ for machine M_1).

The different combinations of parameter types led a to full factorial design of the experiment with $5 \cdot 6 \cdot 3 \cdot 5 = 450$ different two-machine lines that were both analyzed via our numerical method and via the exact solution of the $M/M/1/N$ queueing model. The average overall deviation ((estimate-true value)/true value) of our production rate estimate from the true value was 0.338%. However, this measure only indicates that there appears to be no major systematic deviation as negative deviations are compensated by positive deviations. The more relevant measure of accuracy is the mean absolute deviation (MAD) over the 450 instances which was 5.517%. Figure 2 shows that in about 80% of the cases the production rate estimation error did not exceed 10% while the maximum error exceeded 25%.

Table 4 presents the MAD for each number of periods in the linear program. The accuracy appears to increase moderately as the number of periods increases. It seems to increase strongly with increasing buffer sizes, just as we conjectured, see Table 5.

Table 6 indicates that indeed there appears to be a base processing rate that maximizes the accuracy of our method. For this rate, on average one

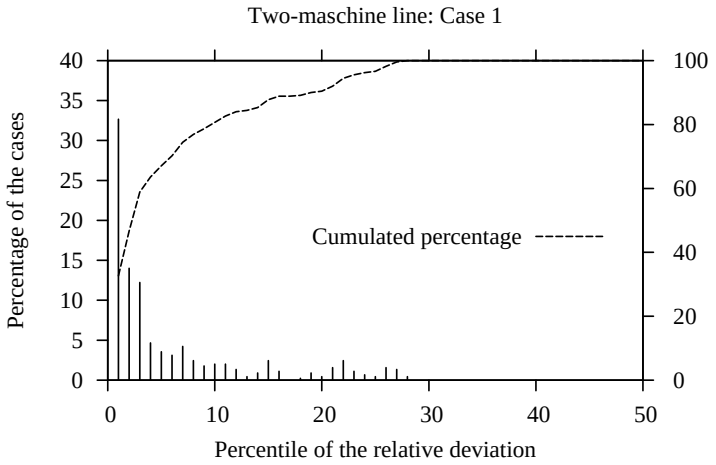

Fig. 2 Percentage of cases over percentiles of relative deviations for all 450 cases of two-machine lines

Table 6 Impact of the base processing rates

Base processing rate	0.1	1.0	10.0
MAD [%]	7.4	1.5	7.7

Table 7 Impact of the bottleneck location

Bottleneck location case	1	2	3	4	5
MAD [%]	5.1	6.0	5.8	5.5	5.2

part is processed per period if non-bottleneck machines operate in isolation. This confirms our hypothesis that for a given number of periods in the linear program a too low base processing rate leads to too few events in the "simulation run" and hence a too high variability of the production rate estimate from the linear program. Table 7 indicates that the method is more accurate for systems with clear bottlenecks (cases 1 and 5 in Table 7) than for balanced lines. This is also a common result for other approximation methods like two-machine decompositions, see Helber (1999, 2005). (A very clear bottleneck essentially determines a systems performance, so that these tend to be relatively easy to analyze.)

In a next step of our analysis (out of the original 450 instances), we asked for the largest possible "compact" subset of instances within a given accuracy limit. The four parameter dimensions in Table 3 with the parameter values given in this table define a complete four-dimensional cube that represents the full-factorial design of our experiment. We now asked for the largest four-dimensional *sub-cube* which is complete in the sense that all its instances meet a given accuracy limit.

Table 8 Compact subspace for the analysis of two-machine lines with a maximum deviation of 10% (240 cases)

Parameter type	Number of cases	Parameter value per case
Number of periods	4	5000, 10000, 20000, 40000
Buffer spaces per buffer	4	4, 8, 16, 32
Base processing rates	3	0.1, 1.0, 10.0
Bottleneck factor	5	(M_1: 0.7; M_2: 1.0), (M_1: 0.9; M_2: 1.0), (M_1: 1.0; M_2: 1.0), (M_1: 1.0; M_2: 0.9), (M_1: 1.0; M_2: 0.7)

The idea is that this sub-cube or compact four-dimensional subset of instances can be constructed by combining four compact partial sets of the parameters that describe the factorial design of our experiment. For example, with respect to the buffer sizes $\{1, 2, 4, 8, 16, 32\}$ introduced in Table 3, the *partial sets* of buffer size parameters $\{\{1, 2, 4, 8\}, \{1, 2\}, \{2, 4, 8, 16, 32\}$ and $\{4, 8\}\}$ are all compact. However, the partial set $\{1, 4, 8\}$ is not compact because the buffer size "2" is missing between "1" and "4" and the subset $\{4, 8, 32\}$ is not compact because the buffer size "16" is missing between its neighbors "8" and "32". We developed an additional integer program (not reported here) to identify the largest possible sub-cube out of the parameter space with a maximum production rate estimation error of 10%.

Table 8 shows the partial parameter sets for this compact subset. If the minimum buffer size was 4 and a minimum of 5000 periods were used within the linear program to estimate the production rate, the production rate estimation error of the remaining 240 cases never exceeded 10% and the MAD was only 1.88%. (Note that there were indeed many more cases with an absolute value of the production rate estimation error below 10% outside of this compact subspace. Figure 2 indicates that 80% or 360 out of 450 cases led to a deviation of at most 10%, but about 120 of them were outside of this four-dimensional sub-cube.) When we reduced the accuracy limit to 5%, the compact subspace reported in Table 9 was detected. In more than 150 cases, the MAD was only 0.909%.

In a last step of our analysis of two-machine lines with exponential processing times, we decided to discard the extremely difficult cases of very low processing rates or very small buffers. We studied the subset of 200 cases out of those introduced in Table 3 with a minimum base processing rate of 1.0 and a minimum buffer size of 4. This led to the frequency diagram in Figure 3 with a MAD of 1.54% and a maximum error of 7.48%. We concluded that the method works reasonably well for exponential two-machine lines unless base processing rates or buffer sizes are very small.

Table 9 Compact subspace for the analysis of two-machine lines with a maximum deviation of 5% (150 cases)

Parameter type	Number of cases	Parameter value per case
Number of periods	5	2500, 5000, 10000, 20000, 40000
Buffer spaces per buffer	3	8, 16, 32
Base processing rates	2	1.0, 10.0
Bottleneck factor	5	(M_1: 0.7; M_2: 1.0), (M_1: 0.9; M_2: 1.0), (M_1: 1.0; M_2: 1.0), (M_1: 1.0; M_2: 0.9), (M_1: 1.0; M_2: 0.7)

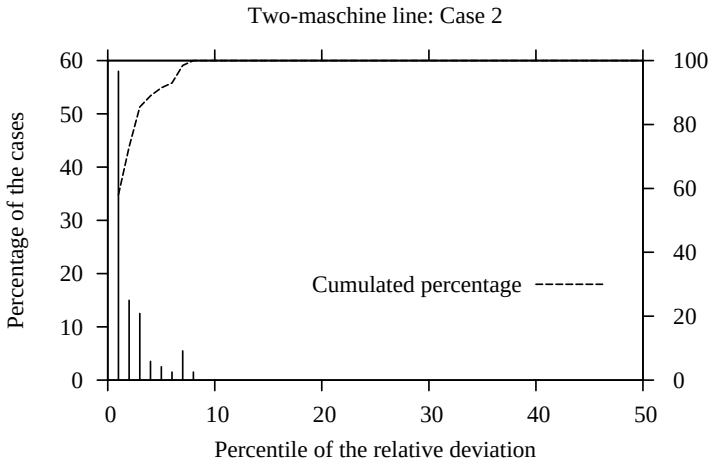

Fig. 3 Percentage of cases over percentiles of relative deviations for 200 cases of two-machine lines with minimum base processing rate of 1.0 and buffer size of 4

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 test bed consisting of the 1944 cases described in Table 10.

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 and Shanthikumar, 1993, p. 542). Figure 4 shows the frequency diagram of relative deviations. The maximum deviation was 40.82% and the MAD was 5.57%. With respect to the number of periods in the linear program, the number of buffer spaces per buffer and the bottleneck location, we found results very similar to those for the two-machine lines. However, the procedure seemed to do best for high processing rates, see Table 11. Table 12 shows that the accuracy slightly deteriorates as the line

Table 10 Test Bed A for the analysis of longer lines (1944 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
Number of periods	3	5000, 10000, 20000
Buffer spaces per buffer	3	2, 10, 50
Base processing rates	3	0.1, 1.0, 10.0
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	3, 5, 7
Processing time variability	4	0.25, 0.5, 1.0, 2.0
Buffer allocation	2	even vs. production-rate maximizing

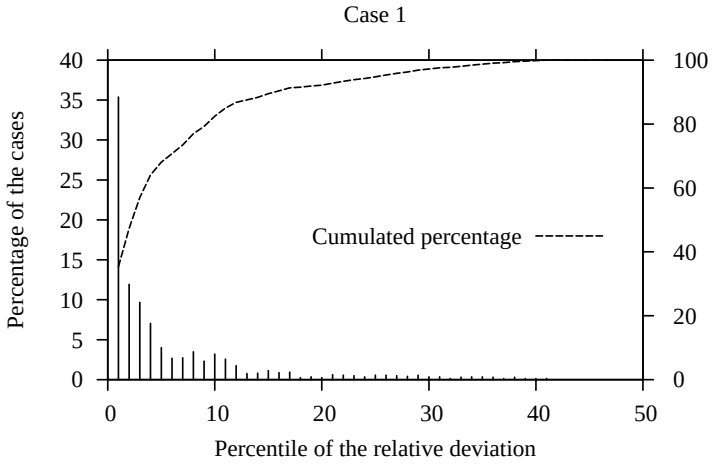

Fig. 4 Percentage of cases over percentiles of relative deviations for all 1944 cases of Test Bed A for longer lines

Table 11 Impact of the base processing rates

Base processing rate	0.1	1.0	10.0
MAD [%]	8.52	5.02	3.156

gets longer. The accuracy of the method did not seem to differ significantly for systems with given or optimized buffer allocation.

The results in Table 13 show that the performance of our method degrades as the variability of the processing times increases.

Tables 14 and 15 summarize the results for parts of the original parameter space from Table 10. Case 7 in Table 15, for example, shows that for

Table 12 Impact of the number of stations in the line

Number of stations	3	5	7
MAD [%]	4.844	5.798	6.056

Table 13 Impact of the squared coefficient of variation of the processing times

SCV	0.25	0.5	1.0	2.0
MAD [%]	2.549	3.781	6.227	9.706

Table 14 Partial analysis for Test Bed A; "a. c." means "all cases"

Case	1	2	3	4	5
Number of periods	a. c.	5000	a. c.	a. c.	a. c.
Buffer spaces p. buffer	a. c.	a. c.	2	a. c.	a. c.
Base processing rates	a. c.	a. c.	a. c.	0.1	a. c.
Bottleneck factor cases	a. c.	a. c.	a. c.	a. c.	a. c.
Line length cases	a. c.	a. c.	a. c.	a. c.	a. c.
Variability cases	a. c.	a. c.	a. c.	a. c.	2.0
Buffer allocation cases	a. c.	a. c.	a. c.	a. c.	a. c.
Number of cases	1944	648	648	648	486
Average deviation [%]	-2.75	-2.45	-8.12	-7.11	-6.62
MAD [%]	5.57	5.93	13.3	8.52	9.71
Maximum absolute deviation [%]	40.82	40.82	40.82	40.82	40.82

Table 15 Partial analysis for Test Bed A (continued)

Case	6	7	8	9
Number of periods	5000	5000	10000, 20000	10000, 20000
Buffer spaces per buffer	a. c.	2	a. c.	10, 50
Base processing rates	0.1	0.1	1.0, 10.0	1.0, 10.0
Bottleneck factor cases	a. c.	a. c.	a. c.	a. c.
Line length cases	a. c.	a. c.	a. c.	a. c.
Variability cases	2.0	2.0	0.25, 0.5, 1.0	0.25, 0.5, 1.0
Buffer allocation cases	a. c.	a. c.	a. c.	a. c.
Number of cases	54	18	648	432
Average deviation [%]	-14	-34.17	0.06	0.27
MAD [%]	15.52	34.17	2.92	0.62
Maximum absolute deviation [%]	40.82	40.82	16.79	3.25

lines with a base processing rate of 0.1, two spaces per buffer and a SCV of 2.0, the MAD is 34.17% which indicates that in this part of the parameter space the method fails. Case 9 in the same table, however, shows that for a base processing rate of at least 1.0, a buffer size of at least 10 and a SCV of the processing times of at most 1.0, the method leads to a MAD of 0.62% with a maximum deviation of 3.25%. Figure 5 shows the frequency diagram. In this area, the method seems to work quite reliably.

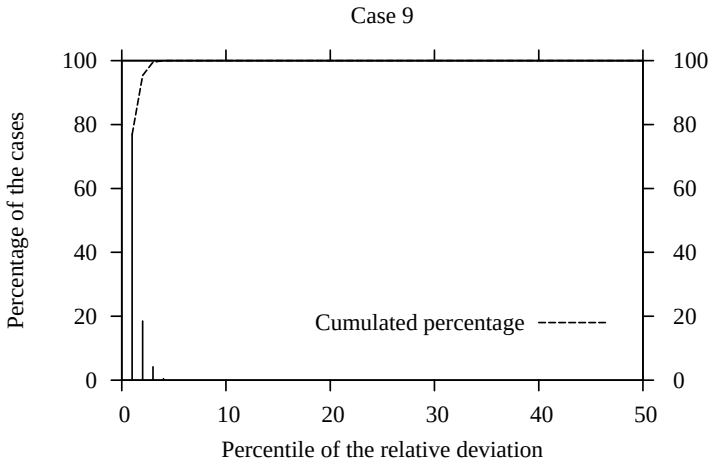

Fig. 5 Percentage of cases over percentiles of relative deviations for all 432 instances of case 9 of test bed

Table 16 Test Bed B for the analysis of longer lines (216 cases)

Parameter type	Number of cases	Parameter value per case
Number of periods	1	10000
Buffer spaces per buffer	4	1, 2, 4, 8
Base processing rates	1	1.0
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	3, 5, 7
Processing time variability	3	0.25, 0.5, 1.0
Buffer allocation	2	even vs. production-rate maximizing

In the final step of our analysis we again turned our attention to smaller buffers in the Test Bed B described in Table 16, as this part of the parameter space is both practically important and most challenging to analyze.

Table 17 shows that the overall performance of the procedure is relatively poor, if only one or two buffer spaces per buffer exist. With four or more buffer spaces, the approach works quite well. Once again, the performance degrades as flow lines get longer, see Table 18. The analysis of the effect of processing time variability in Table 19 confirms the previous observation that the method works well for low and medium variability of the processing times. Figure 6 shows the frequency diagram for Test Bed B.

Table 17 Impact of the number of buffers spaces per buffer (Test Bed B)

Number of buffers	1	2	4	8
MAD [%]	16.72	7.949	2.849	0.956

Table 18 Impact of the number of stations in the line (Test Bed B)

Number of stations	3	5	7
MAD [%]	5.562	7.474	8.319

Table 19 Impact of processing time variability (Test Bed B)

SCV	0.25	0.5	1.0
MAD [%]	3.222	6.232	11.902

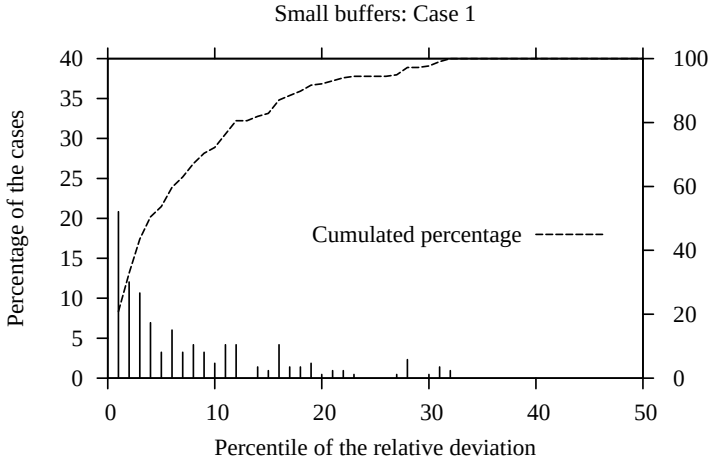

Fig. 6 Percentage of cases over percentiles of relative deviations for all 216 instances of Test Bed B

Table 20 Partial analysis for Test Bed B

Case	1	2	3	4
Number of periods	10000	10000	10000	10000
Buffer spaces per buffer	a. c.	1	2	4
Base processing rates	a. c.	a. c.	a. c.	a. c.
Bottleneck factor cases	a. c.	a. c.	a. c.	a. c.
Line length cases	a. c.	a. c.	a. c.	a. c.
Variability cases	a. c.	a. c.	a. c.	a. c.
Buffer allocation cases	a. c.	a. c.	a. c.	a. c.
Number of cases	216	54	54	54
Average deviation [%]	-7.09	-16.72	-7.95	-2.84
MAD [%]	7.12	16.72	7.95	2.85
Maximum absolute deviation [%]	31.67	31.67	17.07	7.46

Table 21 Partial analysis for Test Bed B (continued)

Case	5	6	7
Number of periods	10000	10000	10000
Buffer spaces per buffer	8	2, 4, 8	1
Base processing rates	a. c.	a. c.	a. c.
Bottleneck factor cases	a. c.	a. c.	a. c.
Line length cases	a. c.	a. c.	a. c.
Variability cases	a. c.	0.25, 0.5	1.0
Buffer allocation cases	a. c.	a. c.	a. c.
Number of cases	54	108	36
Average deviation [%]	-0.84	-2.28	-14.17
MAD [%]	0.96	2.35	14.17
Maximum absolute deviation [%]	3.63	8.43	31.67

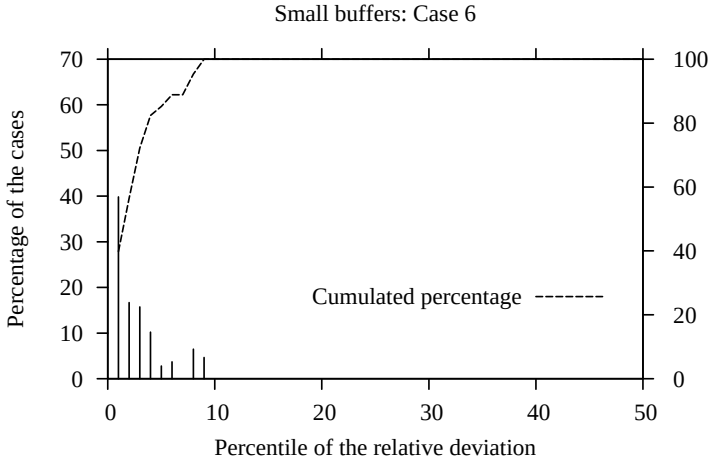

Fig. 7 Percentage of cases over percentiles of relative deviations for all 108 instances of Case 6 of Test Bed B

We again studied parts of the parameter space of Test Bed B as reported in Tables 20 and 21. The result for Case 6 in Table 21 shows that the method works well, if at least two buffer spaces per buffer are available and the SCV of the processing times does not exceed 0.5. The frequency diagram for this case is presented in Figure 7.

The solution times for the linear program were usually within the area of seconds up to a maximum of a few minutes on a 3 GHz Intel Pentium IV machine with 4 GB of RAM and the CPLEX 10.0 solver from ILog. Sometimes it appeared to take more time to construct the matrix of the linear program for a problem instance than to determine the optimal solution of the linear program. With respect to the computation times, incorporating the buffer allocation problem did not seem to make a major difference.

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.

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

Dallery, Y. and 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.

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

Gershwin, S. B. and 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*, Volume 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. (2005). Analysis of flow lines with cox-2-distributed processing times and limited buffer capacity. *Operations Research Spectrum 27*, 211–242.

Helber, S. and 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.

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, and D. T. Sturrock (2006). *Simulation with Arena with CDROM* (4 ed.). McGraw Hill Higher Education.

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

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

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

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

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

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