INT. J. PROD. RES., 1988, VOL. 26, NO. 4, 691–700

An improved algorithm to approximate the behaviour of flow lines[†]

M. B. M. DE KOSTER[‡]

In this paper, an approximation method is given for the analysis of flow lines with a continuous product flow. The algorithm is an improvement of an earlier developed method. It is applied here to lines in which the machines are unreliable with exponentially distributed life and repair times. All intermediate buffers have finite capacity. The algorithm consists of repeated decomposition and aggregation steps, in which two-stage lines are approximated by a single machine. The method appears to be fast and performs well for lines which are not too long.

1. Introduction

Recently some algorithms have been developed to estimate the throughput, average buffer contents and blocking and starvation probabilities of flow lines. There are two different approaches. The first one is the open queueing network approach, where each server has its own random processing time. Examples of such papers are Hillier and Boling (1967), Takahashi *et al.* (1980), Altiok (1982), Brandwajn and Jow (1985), Boxma and Konheim (1981), Gershwin (1987), Suri and Diehl (1986) (they consider a closed flow line), and Perros and Altiok (1986).

Most of the approximation methods are quite accurate, although they are sometimes computationally complex. This complexity is among other things due to the fact that the number of states depends linearly on each of the buffer capacities. In approximating the performance of such a line a number of two server systems have to be solved numerically which may give computational difficulties if the state space is large.

In the other approach, the goods flow is continuous and the machines have production rates instead of service times. Two such papers are De Koster (1987) and Glassey and Hong (1986). In both papers, the machines are unreliable with exponentially distributed life and repair times. This second approach is justified when cycle times are small compared to downtimes and runtimes. A more thorough discussion of the justification of fluid models can be found in Mitra (1986). The only difference between De Koster (1987) and Glassey and Hong (1986) is that failure rates in De Koster are time dependent, which means that the failure rates are constant, independent of the machine states. The failure rates in Glassey and Hong are state dependent, which means here that a machine that is up cannot fail during the time that it is blocked by a full downstream buffer and a stagnating downstream machine or during the time that it is starved, which means that it received no input. The failure rates do not change if the machine is slowed down, but not forced down. The continuous flow model studied here has been solved first for the two-stage case by Wijngaard (1979), who also discussed both time dependent and state dependent failures.

Received March 1987.

[†] Research supported by the Netherlands Organization for the Advancement of Pure Research (ZWO).

[‡] Eindhoven University of Technology, Department of Industrial Engineering and Management Science, P.O. Box 513, 5600 MB Eindhoven, Netherlands.

In this paper the approximation method for flow lines with a continuous goods flow as introduced in De Koster (1987) is improved and compared with the results of Glassey and Hong (1986). The approximation algorithm is tested only for machines with two states (an up and a down state), however, the algorithm is easily adapted for multi-state machines, where each state has its own machine speed and an exponentially distributed sojourn time. Transitions between states have to follow an irreducible Markov process. In De Koster (1986) the algorithm is used to approximate complex networks of machines and buffers.

2. The approximation method

An example of the flow lines studied in this paper is given in Fig. 1. In the line of Fig. 1, the machine speed of production unit i (PU_i) is denoted by v_i, its failure rate by λ_i and its repair rate by μ_i. The capacity of buffer i (B_i) is denoted by K_i, the buffer content is X_i. The throughput of the line of Fig. 1 is denoted by $v(K_1, \ldots, K_{N-1})$. Due to the finiteness of the buffer capacities it may happen that production units are blocked by lack of storage space for finished products. Since the speeds of the PU's are different they may also be slowed down. The method to approximate the behaviour of this production line is clarified in Fig. 2. The random variable $\tilde{X}_i$ denotes the buffer content of the corresponding two-stage line in the ith step. E denotes the expectation operator.

In approximation step i in Fig. 2 the parameters of the ith aggregate PU are calculated as well as $E\tilde{X}_i$. The aggregate PU's are determined to be of the same type as the original PU's (that is with a production rate v, failure rate λ and repair rate μ, with both lifetimes and repairtimes exponentially distributed). The exact calculation of these

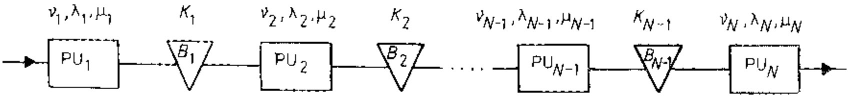

Figure 1. N-stage flow line.

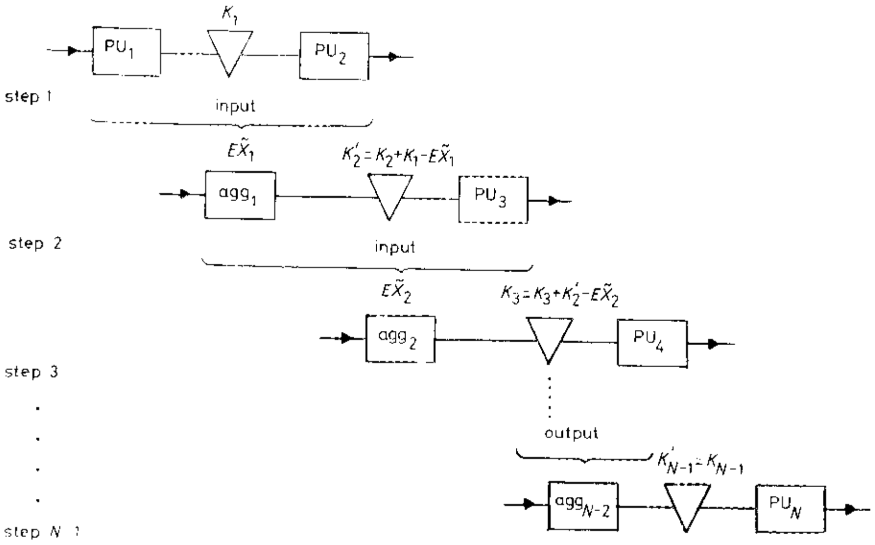

Figure 2. Approximation 'from the left' of the line of Fig. 1.

parameters is based on the two-stage line analysis of Wijngaard (1979). In Wijngaard (1979), regeneration points are distinguished and a cycle is defined as the time between two subsequent regenerations. During such a cycle the total costs can be calculated. By altering the costs per unit of time in each state various two-stage line performance parameters can be calculated, such as throughput, mean cycle time, mean blocking time of PU_1, mean buffer content and many more (see also De Koster (1987)).

For a given two-stage line the failure and repair rate of the aggregate PU are determined as the failure and repair rate of the whole line output (or input). They are determined similarly as in De Koster (1987). For instance, the repair rate of the line output, μ is determined as U/S_0, where U is the average number of uptimes and downtimes of the line output per cycle and S_0 is the average downtime of the line output per cycle. In the $(N-1)$st approximation step the parameters of the aggregate PU agg_{N-2} are determined by using the output behaviour of the two-stage line of step $N-2$, in all other steps the input behaviour is used.

The major difference with the algorithm described in De Koster (1987) is that the buffer capacity in the ith step $(2 \leq i \leq N-2)$ equals $K_i' = K_i + K_{i-1}' - E\tilde{X}_{i-1}$. The buffer capacity in the first step equals K_1 and in the $(N-1)$st step K_{N-1}. An explanation for the determination of the aggregate PU and for the used buffer capacities is that in step $i (1 < i < N-2)$ the input behaviour is such that agg_{i-1} 'sees' that the average maximum buffer capacity downstream equals $K_i' = K_i + K_{i-1}' - E\tilde{X}_{i-1}$. The average content of agg_{i-1} is $E\tilde{X}_{i-1}$, so $K_{i-1}' - E\tilde{X}_{i-1}$ can be stored additionally to K_i. The output of the $(N-2)$nd two-stage line 'sees' a maximum buffer capacity downstream of K_{N-1}.

The approximation from the left, as sketched in Fig. 2 is of course only one of the different possible aggregation sequences. This sequence is chosen to estimate the behaviour of the content of B_{N-1}. If we are interested in the behaviour of the content of B_1, then we have to approximate from the right to the left. In such an approximation from the right, a similar scheme as in Fig. 2 can be used. However, instead of input and output behaviour we then have to use output and input behaviour, respectively.

If the behaviour of the content of another buffer is desired, say B_i, for $2 \leq i \leq N-2$, then a combination of approximation from the left and from the right has to be used. Suppose for instance $3 \leq i \leq N-3$, then approximation from the left can be applied on the line $\mathrm{PU}_1 - B_1 - \ldots - B_{i-1} - \mathrm{PU}_i$, but K_{i-1}' can best be defined as $K_{i-1} + K_{i-2}' - E\tilde{X}_{i-2}$, instead of K_{i-1}. In a similar way approximation from the right can be applied on the line $\mathrm{PU}_{i+1} - B_{i+1} - \ldots - B_{N-1} - \mathrm{PU}_N$. K_{i+1}' is defined as $K_{i+1} + E\tilde{X}_{i+2}$. Combining the results of both approximation procedures yields an approximation of the behaviour of the content of B_i.

3. Approximations of three-stage lines

A drawback of the method of aggregation for three-stage lines is that buffer correlation is not taken into account. In the whole three-stage line it may happen that PU_2 is starved and blocked simultaneously. By applying the approximation of the previous section, that is, by first aggregating over PU_1, B_1 and PU_2 with respect to the output behaviour and then calculating the throughput of $\mathrm{agg}_1 - B_2 - \mathrm{PU}_3$, this throughput estimate will often be too small. If it happens often that B_1 is empty and B_2 is full, simultaneously, then this will be the case. This is most likely to happen if $\mathrm{eff}_2 > \mathrm{eff}_1$, where

$$\mathrm{eff}_i = (v_i \mu_i)/(\lambda_i + \mu_i),$$

the efficiency of PU_i. Since the only interesting lines are those lines where B_2 can become full indeed, this phenomenon will then certainly arise. This misestimation of the throughput can be corrected for instance by adding some constant to the capacity of B_1 in the approximation from the left. However it is difficult to quantify this correction constant.

Therefore we propose the following approach in approximation from the left in case $\mathrm{eff}_2 > \mathrm{eff}_1$, or in case $\mathrm{eff}_2 = \mathrm{eff}_1$ and PU_2 is more stable than PU_1. See also Fig. 3. In step 1 in Fig. 3 the average buffer content, $E\tilde{X}_1$, as well as the parameters of agg_1 are calculated (from the input behaviour, as in §2). In step 2 as buffer capacity $K_2 + K_1 - E\tilde{X}_1$ is taken since the input of the line in step 2 can still store $K_1 - E\tilde{X}_1$ in agg_1 and additionally K_2 in B_2. This gives us a good estimate for the throughput, $\hat{v}$. That this estimate for the throughput is a good one in case $\mathrm{eff}_2 > \mathrm{eff}_1$ can be understood from the following two special cases.

(*a*) Suppose PU_2 is perfect and $v_2 \geq \max\{v_1, v_3\}$. In this case the line is equivalent with the two-stage line $\mathrm{PU}_1 - B - \mathrm{PU}_3$, with B a buffer of capacity $K_1 + K_2$. Since in the approximation method of Fig. 3, agg_1 equals PU_1 and $E\tilde{X}_1 = 0$ the throughput estimate $\hat{v}$ equals $v(K_1, K_2)$.

(*b*) Suppose PU_3 is perfect and $v_3 \geq v_2$. It is easy to see that PU_2 will never be blocked or slowed down by PU_3. Hence the line is equivalent with the line $\mathrm{PU}_1 - B_1 - \mathrm{PU}_2$. In the approximation of Fig. 3, the throughput estimate $\hat{v}$ again equals $v(K_1, K_2)$.

If only a throughput estimate of the line is needed, then we are finished, since $\hat{v}$ is a good estimate of v. If the behaviour of the buffer content of B_2 is needed then an iterative procedure can be started by adding a constant C_1 to K_1, then applying the approximation of Fig. 2, comparing the resulting throughput and $\hat{v}$, then adapting C_1 and repeating this process until sufficient convergence is reached. As a rule an underestimate and an overestimate of $\hat{v}$ is needed for such an iterative procedure. For an underestimate of $\hat{v}$, C_1 can be chosen 0, for an overestimate (in most cases) $C_1 = \max\{K_1, K_2\}$ is in most cases sufficient.

In approximation from the right to the left a similar procedure is followed if $\mathrm{eff}_2 > \mathrm{eff}_3$.

4. Numerical results and comparisons

In this section numerical results of the approximation are given. They are compared with simulation results on the one hand and with results of De Koster (1987) and Glassey and Hong (1986) on the other hand. The algorithm described in Glassey and Hong (1986) is the continuous flow version of the algorithm of Gershwin (1983). In Glassey and Hong (1986), the algorithm was tested only for lines with state dependent

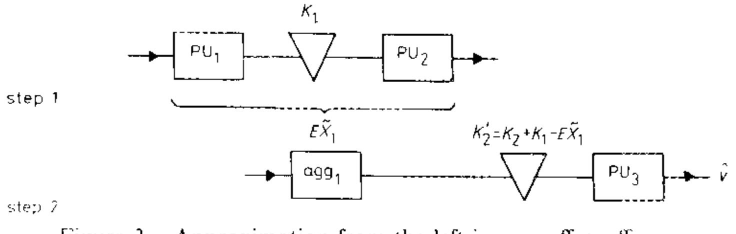

Figure 3. Approximation from the left in case $\mathrm{eff}_2 > \mathrm{eff}_1$.

Case	No. PU's	ν	λ	μ	K	eff
1.	3	(1, 0·9, 1)	(0·01, 0·01, 0·01)	(0·09, 0·07, 0·09)	(45·49, 20)	(0·9, 0·7875, 0·9)
2.	3	(0·9, 10, 1)	(0·01, 0·01, 0·01)	(0·09, 0·09, 0·09)	(3·17, 2)	(0·81, 9, 0·9)
3.	3	(1, 1·1, 1·2)	(0·01, 0·02, 0·03)	(0·09, 0·09, 0·09)	(50, 50)	(0·9, 0·9, 0·9)
4.	4	(1·05, 1, 1·05, 1)	(0·01, 0·01, 0·01, 0·01)	(0·07, 0·07, 0·07, 0·09)	(15, 9, 12)	(0·919, 0·875, 0·919, 0·9)
5.	4	(10, 10, 10, 1)	(0·01, 0·01, 0·01, 0·01)	(0·09, 0·09, 0·09, 0·09)	(21.01, 10·91, 20)	(9, 9, 9, 0·9)
6.	5	(0·9, 1·05, 1, 0·9, 1)	(0·01, 0·01, 0·01, 0·01, 0·01)	(0·09, 0·07, 0·09, 0·07, 0·09)	(10·50, 9·63, 15·29, 12)	(0·81, 0·919, 0·9, 0·7875, 0·9)
7.	5	(1·05, 1, 1·1, 0·9, 0·9)	(0·01, 0·01, 0·02, 0·01, 0·01)	(0·07, 0·09, 0·09, 0·09, 0·07)	(10·50, 9·63, 15·29, 12)	(0·919, 0·9, 0·9, 0·81, 0·7875)

Table 1. Seven multi-stage cases.

failures, here it is tested also for lines with time dependent failures. The first seven cases are listed in Table 1. In this table three three-stage lines, two four- and two five-stage lines are considered, all with time-dependent failures. For the three-stage lines, dependent on the ratios of the machine efficiencies, the method of § 2 or § 3 is used. Although for the third case the efficiencies are equal, since the variance in machine speed increases from the left to the right, PU_2 is more stable than PU_3 and therefore in the approximation from the right the method of § 3 is used. All lines in Table 1 were simulated 10 runs of 500 000 units of time per run on a Burroughs B7900 computer. The results are listed in Table 2.

In Table 2 p_i stands for the probability of buffer i full, q_i is the probability of buffer i empty and L_i is the average content of buffer i. De K (left) and (right) stand for

Case		$v(K)$	p_1	q_1	p_{N-1}	q_{N-1}	L_1	L_{N-1}
1.	simulation	0·7650	0·4998	0·0045	0·0213	0·5360	38·11	4·380
	de K (left)	0·7648			0·0216	0·5329		4·423
	de K (right)	0·7652	0·5037	0·0044			38·22	
2.	simulation	0·7007	0·1286	0·6472	0·2589	0·5820	0·720	0·681
	de K (left)	0·7048			0·0750	0·7449		0·349
	de K* (left)	0·6902			0·0743	0·7757		0·308
	de K (right)	0·6948	0·1352	0·6494			0·773	
	de K* (right)	0·6919	0·1386	0·5622			0·940	
3.	simulation	0·8288	0·0753	0·0790	0·0495	0·1636	30·60	22·63
	de K (left)	0·8204			0·0434	0·1804		21·30
	de K (right)	0·8289	0·0751	0·0680			30·99	
	de K* (right)	0·8242	0·0801	0·0623			31·71	
4.	simulation	0·7323	0·5536	0·0490	0·1869	0·3393	10·96	4·161
	de K (left)	0·7238			0·1606	0·1875		4·099
	de K* (left)	0·7179			0·1446	0·1938		3·647
	de K (right)	0·7385	0·5586	0·0482			11·11	
	de K* (right)	0·7158	0·5864	0·0456			11·40	
5.	simulation	0·8788	0·9032	0·0164	0·8545	0·0220	20·08	18·67
	de K (left)	0·8743			0·8178	0·0271		18·35
	de K* (left)	0·8716			0·8006	0·0300		18·19
	de K (right)	0·8683	0·9019	0·0132			20·11	
	de K* (right)	0·8409	0·9057	0·0129			20·14	
6.	simulation	0·6769	0·2164	0·2881	0·0303	0·6609	5·322	2·167
	de K (left)	0·6858			0·0306	0·6614		2·130
	de K* (left)	0·6467			0·0278	0·6975		1·879
	de K (right)	0·6610	0·1758	0·2000			6·039	
	de K* (right)	0·6419	0·1984	0·1627			6·599	
7.	simulation	0·6671	0·6478	0·0546	0·2074	0·4745	8·135	4·300
	de K (left)	0·6652			0·1876	0·4997		4·088
	de K* (left)	0·6431			0·1571	0·5448		3·521
	de K (right)	0·6629	0·6795	0·0516			8·534	
	de K* (right)	0·6284	0·6996	0·0492			8·599	

Table 2. Numerical results of the two approximation methods.

Case	No. PU's	ν	λ	μ	K	eff	SD/TD
8.	3	(1, 1·1, 1·2)	(0·01, 0·02, 0·03)	(0·09, 0·09, 0·09)	(50, 50)	(0·9, 0·9, 0·9)	SD
9.	3	(1·5, 1, 1·1)	(0·05, 0·02, 0·03)	(0·1, 0·08, 0·07)	(30, 70)	(1, 0·8, 0·77)	SD
10.	4	(1·1, 1, 1, 0·9)	(0·02, 0·01, 0·01, 0·01)	(0·08, 0·07, 0·09, 0·05)	(40, 50, 60)	(0·88, 0·875, 0·9, 0·75)	SD
11.	3	(1, 0·9, 1)	(0·01, 0·01, 0·01)	(0·09, 0·07, 0·09)	(18·29, 8)	(0·9, 0·7875, 0·9)	TD
12.	5	(0·9, 1·05, 1, 0·9, 1)	(0·01, 0·01, 0·01, 0·01, 0·01)	(0·09, 0·07, 0·09, 0·07, 0·09)	(6·96, 6·38, 10·29, 8)	(0·81, 0·919, 0·9, 0·7875, 0·9)	TD

Table 3. Five lines with time dependent and state dependent failures.

approximation from the left and from the right respectively with the algorithm of this paper, de K* stands for the method of De Koster (1987). For some three-stage lines (case 1 and case 3 in the approximation from the left) the approximation of this paper coincides with the approximation of De Koster (1987). In such cases the results of De Koster (1987) are not given.

From Table 2 it appears that the algorithm described in this paper is a substantial improvement compared with the method of De Koster (1987) especially as far as the throughput estimate is concerned. The computation times are worse, since in each two-stage line evaluation the average buffer content has to be calculated additionally. However, these time differences are small. Note that for the second case the relative error in the estimation of L_{N-1} is big. This is caused by the fact that PU_2 is much faster than both PU_1 and PU_3 and $eff_2 > eff_1$.

In Table 3 five lines are listed to be compared with the method of Glassey and Hong (denoted by G&H). Two lines have time dependent failures and three lines have state dependent failures (denoted by TD and SD, respectively). They were all simulated 10 runs of 500 000 units of time per run. The results are given in Table 4.

From the results of Table 4 it appears that the algorithm of G&H is best for state dependent failures. For time dependent failures the throughput estimate of de K is best. Although often the algorithm of G&H performs better than the one of this paper (especially for state dependent failures), a major drawback is the long and unpredictable runtimes needed. In Table 4, the number of two-stage line evaluations for both algorithms (M) is listed. For an N-stage line the algorithm of this paper takes $N-1$ two-stage line evaluations for the throughput and less than $(N-1)^2$ two-stage

Case		$v(K)$	p_1	q_1	p_{N-1}	q_{N-1}	L_1	L_{N-1}	M
8.	simulation	0·8366	0·0738	0·0916	0·0538	0·2042	30·71	22·49	
	G&H	0·8356	0·0715	0·0655	0·0513	0·1770	32·76	22·81	44
	de K (left)	0·8242			0·0452	0·2042		21·16	2
	de K (right)	0·8322	0·0754	0·0738			31·06		10
9.	simulation	0·7314	0·4388	0·0355	0·0660	0·1369	22·55	35·72	
	G&H	0·7280	0·3926	0·0340	0·0560	0·1220	22·27	34·67	34
	de K (left)	0·7228			0·0556	0·1363		33·78	2
	de K (right)	0·7348	0·3909	0·0376			21·96		12
10.	simulation	0·7441	0·3958	0·0304	0·5155	0·0068	28·14	49·64	
	G&H	0·7476	0·3683	0·0256	0·4732	0·0031	28·60	50·37	198
	de K (left)	0·7362			0·2579	0·0184		40·22	3
	de K (right)	0·7294	0·3948	0·0244			29·16		3
11.	simulation	0·7300	0·5841	0·0214	0·0457	0·6438	14·68	1·626	
	G&H	0·7384	0·6046	0·0199	0·0463	0·6488	15·05	1·575	40
	de K (left)	0·7284			0·0461	0·6453		1·618	2
	de K (right)	0·7298	0·5906	0·0212			14·76		2
12.	simulation	0·6446	0·2781	0·2990	0·0400	0·7080	3·649	1·353	
	G&H	0·7098	0·1179	0·2892	0·0434	0·6962	3·806	1·333	592
	de K (left)	0·6516			0·0404	0·7137		1·313	4
	de K (right)	0·6293	0·2132	0·2128			4·071		4

Table 4. Comparison with the results of Glassey and Hong.

line evaluations for the acquisition of all average buffer contents. The number of two-stage line evaluations of Glassey and Hong is unpredictable (it is based on a standard algorithm for the iterative solution of a system of $6N-5$ non-linear equations). It depends on the initial trial-solution, but the procedure may also fail to converge.

In experiments with longer lines (up to 15 machines), it has appeared that adapting the buffer contents of all buffers overestimates the throughput. Therefore it is suggested for lines consisting of more than five machines, *not* to adapt the buffer contents in each fourth or fifth aggregation step of the approximation scheme of Fig. 2, depending on the amount of correlation between buffers. The more correlation between buffers in a line, the more frequent buffers need to be adapted. However, more research has to be devoted to such large systems in order to obtain good approximation methods.

References

ALTIOK, T., 1982, Approximate analysis of exponential tandem queues with blocking. *European Journal of Operational Research*, **11**, 390–398.

BOXMA, O. J., and KONHEIM, A. G., 1981, Approximate analysis of exponential queueing systems with blocking. *Act. Inform.*, **15**, 19–66.

BRANDWAJN, A., and JOW, Y. L., 1985, An approximation method for tandem queues with blocking. *Amdahl Corporation Technical Report.*

DE KOSTER, M. B. M., and WIJNGAARD, J., 1987, On the equivalence of multi-stage production lines and two-stage lines. To appear in *IEE Transactions.*

DE KOSTER, M. B. M., 1987, Estimation of line efficiency by aggregation. *International Journal of Production Research*. **25**, 615–626.

DE KOSTER, M. B. M., 1986, Approximate analysis of production systems. Report BDK/ORS/86/09, Eindhoven University of Technology.

GERSHWIN, S. B., 1987, An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. *Operations Research*, **35**, 291–305.

GLASSEY, C. R., and HONG, Y., 1985, The analysis of behavior of an unreliable N-stage automatic transfer line with $(N-1)$ interstage buffer storages. Working paper ORC 86-10, Operations Research Center, University of California, Berkeley, U.S.A.

HILLIER, F. S., and BOLING, R., 1967, Finite queues in series with exponential or Erlang service times—A numerical approach. *Operations Research*, **15**, 286–303.

MITRA, D., 1986, Stochastic theory of a fluid model of multiple failure-susceptible producers and consumers coupled by a buffer. Working paper, AT&T Bell Laboratories, Murray Hill, U.S.A.

PERROS, H. G., and ALTIOK, T., 1986, Approximate analysis of open networks of queues with blocking: Tandem configurations. *IEEE Transactions of Software Engineering*, **12**, 450–461.

SURI, R., and DIEHL, G. W., 1986, A variable buffer-size model and its use in analyzing queueing networks with blocking. *Management Science*, **32**, 206–224.

TAKAHASHI, Y., MIYAHARA, H., and HASEGAWA, T., 1980, An approximation method for open restricted queueing networks. *Operations Research*, **28**, 594–602.

WIJNGAARD, J., 1979, The effect of interstage buffer storage on the output of two unreliable production units in series, with different production rates. *AIIE Transactions*, **11**, 42–47.

Dans cet article une méthode d'approximation est presentée pour l'analyse des lignes de déroulement d'un courant continu. L'algorithme, qui est une amélioration d'une méthode antérieure, est appliquée sur des lignes consistant de plusieures étapes, dans lesquelles les machines ne sont pas sûr, avec longévité et durée de réparation distribuées exponentiellement. La méthode est composée de pas de décomposition et d'aggregation à plusieurs reprises. Dans un tel pas d'aggregation une parcelle de ligne consistant de deux machines, est remplacée par une seule machine. La méthode fait preuve d'être rapide et efficace en cas des lignes de longueur limitée.

In dieser Abhandlung wird eine Methode gegeben für die Annäherung von mehrstufigen Fliessbänder mit einem stetigen Güterstrom. Das Algorithmus ist eine Verbesserung von einer früheren entwickelten Methode. Es wird hier angewendet auf Fliessbänder mit unzuverlässige Maschinen die exponentielle Betriebslebensdauer und Reparaturzeiten haben. Alle zwischenliegende Puffer haben eine endliche Kapazität. Das Algorithmus besteht aus wiederholte Dekompositions und Aggregationsschritte, in den zwei Maschinen angenähert werden von einer einzigen Maschine. Es zeigt sich dass die Methode schnell ist und gut wirkt für nicht zu lange Linien.