Stochastic Modeling and Applications
Vol.20, No. 2 (December, 2016), 91-109

MUK PUBLICATIONS
Open Access Publisher

INVESTIGATING THE POTENTIAL OF STATE-TRUNCATION APPROXIMATIONS FOR PRODUCTION LINES

DENIZ DURMUS, SENCER YERALAN, AND AYBEK KORUGAN

Abstract. We present preliminary investigative work into alternative approximations to model of production and service systems. Exact Markovian formulations of production and service systems are known to yield unimaginably large state spaces. The literature is ripe with various decomposition approximations for the evaluation of the various performance measures, e.g., the production rate. Almost universally, approximations are validated by simulation. This practice has been entrenched in the dominant paradigms of operations research. It is well known that simulation runs yield acceptable estimates in spite of visiting only a very small subset of the system states. Our investigation seeks to test analytical approximations that mimic simulation by truncating exact state-space models in order to reduce the computational difficulties. The work presented here should not be taken as an approximation technique, but rather as an early investigative effort to see if such an approach exhibits enough promise to develop and refine practical techniques.

1. Synopsis

We take a rather phenomenological approach to reviewing a fundamental component of the various stochastic models developed within the operations research community over the past half century ([5]). We dwell on the essence of simulation studies, which are ubiquitously employed as verification tools in most of the related research. Due to the sheer size of the accompanying state-space-based stochastic models (e.g. Markovian models), studies reported in the literature offer many approximation techniques. The ontological aspects of very large state-space-based stochastic models are questioned ([18]) in the recent literature. For instance, production line models may easily have state spaces with cardinalities in the order of 10^100 or more. Since seeking exact solutions is practically prohibitive, in case of serial production lines, one finds the literature ripe with many decomposition approaches ([7], [4], [20]). The validation of these approximation techniques are usually done through simulation studies. One is content with approximations that agree with simulation results within a few percentage points.

In this paper, we intentionally take a rather uncomplicated system, namely a tightly coupled serial production line with station breakdown. Past work by one of the authors at an automotive plant ([11]) involves serial assembly lines with over 160 workstations. The line consists of a series of segments separated by buffers.

2000 *Mathematics Subject Classification.* Primary 60J99; Secondary 15A24.

Key words and phrases. Approximation Techniques, Production and Service Systems, Simulation, Stochastic Models, Markovian Models.

There are no inter-station buffers within the segments. As an approximation, first the segments were replaced by a single equivalent station, and then a decomposition method ([20]) was implemented. The equivalent station representing the tightly coupled (bufferless) line segment was developed from the initial principles, and its validity was shown by simulation.

Philosophically speaking, there is no a priori watertight evidence that simulation would provide unbiased and accurate estimates of the performance measures of the line segments. That is, before one employs simulation as a validation tool, simulation itself should be validated. How is it possible that simulation is a justifiable validation tool, when it is obvious that a regular simulation run could never visit all of the states of the Markovian model with, say 10^{100} states? This conundrum is rarely addressed in the literature. Moreover, as simulation constitutes a fundamental component of the dominant paradigm, it behooves us to scrutinize its role and its limitations, both qualitatively and quantitatively. A recent study ([6]) along these lines of thought provide the inspiration of our approximation framework.

1.1. Production Lines. In almost all cases, focus is on the evaluation of two primary performance measures: the production rate, and the expected number of items in the buffers. The studies of [9], [14], [15], and [10] are early examples of discrete -time Markovian models. Studies conducted by [19], and [20] provide examples of continuous-time models. There are many extensions to these basic models. For instance, [13] present closed-loop systems. These may be considered as queueing networks. Such closed-loop systems are seen in industries where hot pallets are used to hold the workpiece while it progresses through the manufacturing system. Once the workpiece is completed, the hot pallets are returned to the beginning of the line. Thus, the models track the hot pallet. There are several textbooks on the subject, to which the reader is referred for detailed information ([1], [2], [3], [8], [12], [16], [17]).

Our work does not focus on a particular new extension to the production line models. We take a uncomplicated model based on a well-known set of assumptions (MY81 [15]) and investigate if a truncation approximation holds promise.

2. Motivation

Simulation is most often used as a tool of verification. We do not question the basic premise that simulation may be used in this capacity. However, many models are developed in such a brute-force manner that the meaning of the very premise of a model becomes vulnerable to criticism. A recent report by Yeralan and Buyukdagli [18] mentions an automotive plant with 168 robotic workstations arranged in a serial manner ([11]). Even without inter-station buffers, given that each workstation is subject to breakdown, the system modeled as a discrete-time discrete-space Markov chain has over 10^{50} states. If we were to visit a different state every nanosecond, a complete tour of all the states would take 10^{27} times the age of the universe. This is an incomprehensible number so incomprehensible that Yeralan and Buyukdagli calls into question the ontology of such a model. Given that there are an inordinate number of system states in typical Markovian models, how is it the case that simulation gives us answers which would take total enumeration a practically endless amount of time?

As an attempt to clarify, consider a system with the number of states even greater than the 10^{50} mentioned for the automotive plant model. The continuous-time M/M/1 queue with given arrival (λ) and service (μ) rates. The number of customers in the system uniquely determines the state of the system. It is clear that the number of customers in the system is unbounded. Thus, the system state space is unbounded, with an infinite number of elements. A complete tour of all the states of the system is, by definition, impossible in a finite amount of time, irrespective of how quickly we visit each state. The system is stable as long as the traffic intensity $\rho = \lambda/\mu$ is less than unity. We often investigate two performance measures: the expected number of customers in the system, and the utilization of the server. The expected number of customers in a stable queue is 1/(1-ρ). Similarly, the server is busy with probability ρ and idle with probability (1-ρ) (Hillier and Lieberman, 2010).

A simple simulation (written in SciLab) of the M/M/1 queue quickly provides the two performance measures, as shown below. Figure 1 plots the average number of customers in the system, and server utilization for the case λ=9 and μ=10, (ρ=0.9).

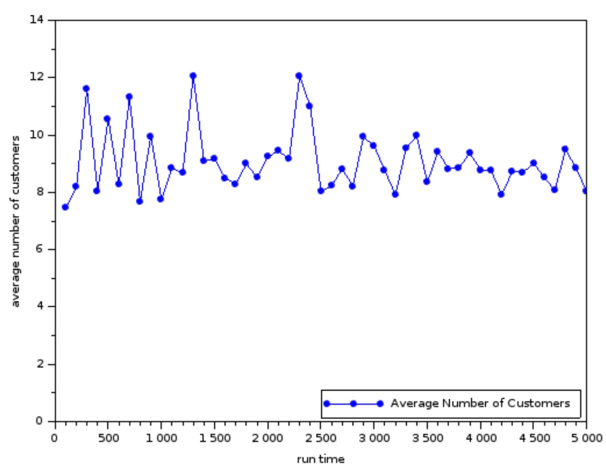

FIGURE 1. The Average Number of Customers in the System as a Function of the Simulation Runtime.

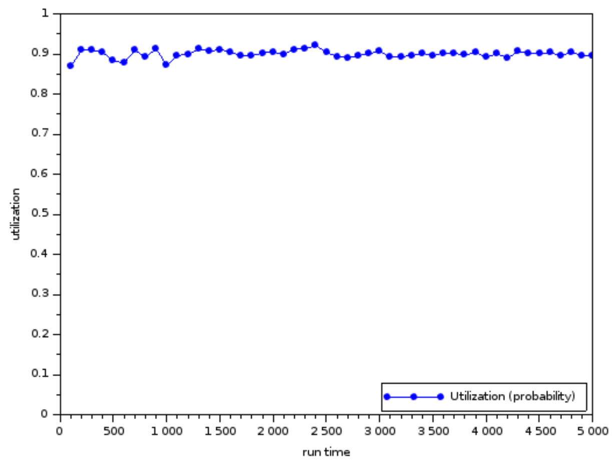

FIGURE 2. Server Utilization as a Function of the Simulation Runtime.

Now, the question remains, as to how is it possible for a simulation run of only a few thousand transaction to yield performance measures so close to the theoretical values (relative errors in the range of a few percent), given that a total enumeration of the states is impossible. After all, the M/M/1 queue has an infinite number of states. It is clear that the simulation does not visit all possible states.

Clearly, the M/M/1 queue may only make a transition to an adjacent state. That is, if there are $N > 0$ customers in the system, the next transition would be to either state N+1 or to state N-1. Hence, the number of customers in the system throughout the simulation run is bounded by a maximum and a minimum. It is customary to start the system at state 0 (empty system). Then the maximum number of customers in the system, N_{max}, during the simulation run is a finite number. Again, it is quite clear that N_{max} is a function of the simulation run. As the simulation run time increases, we may expect N_{max} to also grow, as there is more time for visiting states with a higher number of customers.

We modify the simulation code keep track of N_{max}. It is then plotted as a function of the runtime.

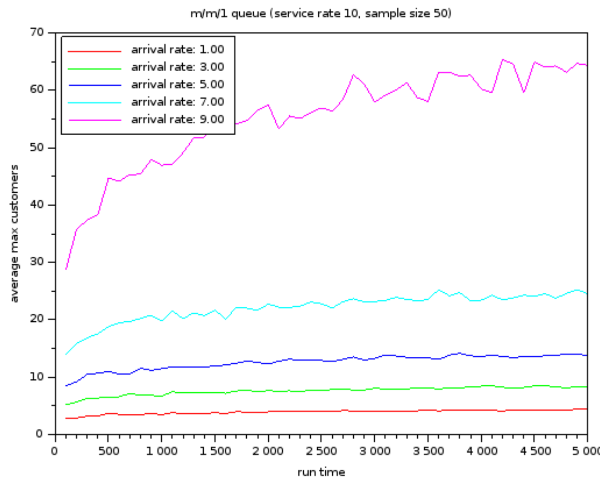

FIGURE 3. N_{max} as a Function of Runtime, as Obtained by Simulation.

Figure 3 shows the results of the simulation runs. Each simulation run has a service rate of 10, and a runtime between 100 and 5000 time units. Five different arrival rates are used: 1, 3, 5, 7, and 9. The simulation was run and the maximum number of customers in the system (N_{max}) throughout the runs were recorded. Each point on the graph is actually an average of 50 runs with identical parameters. It is interesting to observe that, in each case, N_{max} is an increasing function, although its growth slows down considerably.

It is possible to analytically compute the expected value of N_{max} as a function of the system parameters and the length of time the system is observed. Such computation falls into the domain of transient analysis. The Appendix 1 gives the transient analysis for an M/M/1 queue which starts with an idle server and runs for a given period of time. The analytical work shows how N_{max} may be computed. Here we will suffice with simply graphing the theoretical values of N_{max} and comparing them to the figure above.

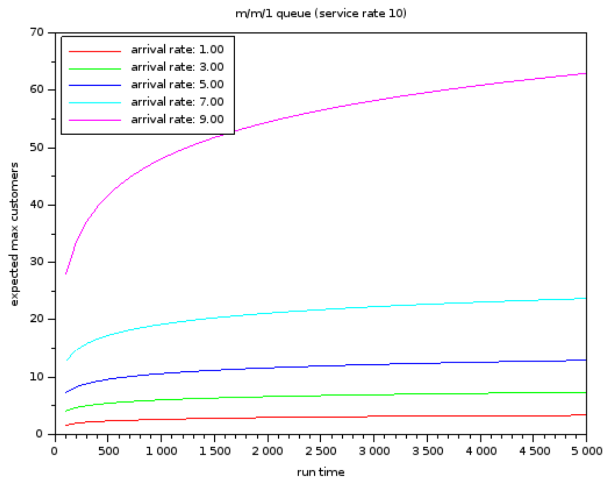

FIGURE 4. N_{max} as a Function of Runtime, as Computed Theoretically.

With a simulation runtime of 5000 time units, theoretically, the expected maximum number of customers in the system is about 65.

2.1. Observations. In effect, the simulation runs successfully evaluate the two performance measures, the utilization probability and the expected number of customers in the system, by only visiting a handful (say 100) of states out of the possible infinite number of states. The implications of this observation are quite significant in many ways.

First, it shows that the value of simulation as an investigative tool is not only in its utility to collect data and obtain statistics, but also in delineating and concentrating on the more likely states and ignoring (completely) the states which have negligible effect on the performance measures. In effect, the simulation only considers a truncated system, where among the infinite number of states, only a few hundred states are dealt with. In other words, if we were to build a simulation model of the M/M/1 queue and another for a modified M/M/1 queue where arrivals to the system with 100 customers in it were lost (M/M/1/100), the two models would give us exactly identical numerical results.

In a related observation, we see that simulation initial conditions are of importance. For example, if we were to start the given M/M/1 queue simulation with 1,000,000 customers in the system and ran it for a few thousand time units, we would always get the utilization to be 100%, since the simulation run would not be long enough for the system to reach a steady-state.

A similar case could be made for different performance measures. Suppose the probability that there are 1000 customers or more in the system is taken as a performance measure. For the system discussed above, our simulation runs would always return this value to be zero, since there will never be enough time for the runs to observe such states with 1000 or more customers. Theoretically, however, these performance measures are readily available.

Finally, our observations point to some insights concerning the modeling of production lines. Inspired by the discussions above, one may attempt to build a truncated model of the system and solve it. Of the 168 stations of the automotive line, how many of the stations could be down at any given time? Clearly, the state where all 168 stations are down is very very unlikely. If the probability of breaking down is in the range [0, 0. 01], then in the worst case, the probability that all 168 stations break down is about 10^{-336}. Granted there are more ways to enter the state where all 168 stations are down, but still, the argument is quite strong that the state with all 168 stations down will almost certainly never be observed.

We entertain the idea of truncated state space approximations next.

3. A Three Station Line with No Buffers

Consider a three-station production line with no buffers. The line is assumed to operate in discrete time, as explained in [15]. There are five station states: up and operating (U), up but blocked (B), up but starved (S), down and under repair (D), down, blocked and under repair (X). With three stations and no buffers, the Markov chain has 32 states. These states are listed below.

TABLE 1. System States of the Production Line.

Index	Station States	Index	Station States	Index	Station States	Index	Station States
0	DDD	8	DSS	16	UUS	24	BDS
1	**DDU**	9	DBD	17	USD	25	BBD
2	DDS	10	DXD	18	**USU**	26	BXD
3	DUD	11	UDD	19	USS	27	XDD
4	**DUU**	12	**UDU**	20	UBD	28	**XDU**
5	DUS	13	UDS	21	UXD	29	XDS
6	DSD	14	UUD	22	BDD	30	XBD
7	**DSU**	15	UUU	23	**BDU**	31	XXD

Having only 32 states, the Markov chain can be solved exactly. Given the station breakdown repair probabilities, we easily compute the steady-state probabilities. The production rate is obtained as the steady-state probability that the last station is in the state up and operating (the production rate is also available as functions of the other station probabilities, however, such details are not the focus of our study. The reader is referred to the literature (see for instance [8]) for these details). In this case, there are eight such states, marked in bold in Table 1.

An illustrative simple case is when we have identical stations, each with a breakdown probability of q and a repair probability of r. The production rate for such a production line is illustrated by the graphic below.

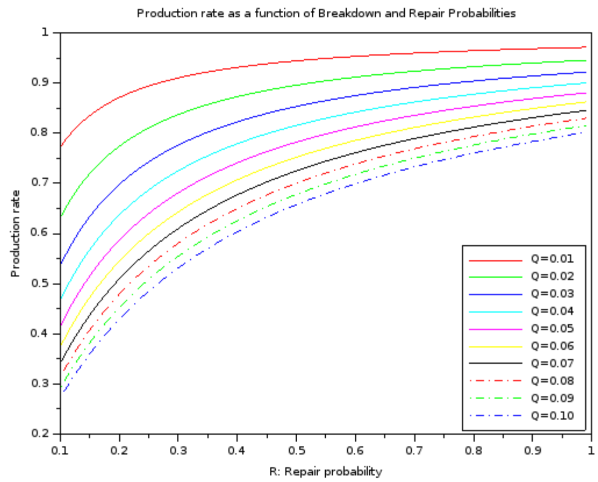

FIGURE 5. Production Rate as a Function of Breakdown and Repair Probabilities.

The insights into the why simulation works well for the M/M/1 queue led us to conclude that certain states are never visited. Those states which are pertinent to the performance measures are mostly visited. The performance of simulation, of course, also depends on the initial state. We next list explicitly the steady-state probabilities of each of the system states to see if there are any states with negligible effects on the performance measure. We select rather realistic parameters. A breakdown probability of 0.01 means that the mean time between failures is 100 cycles. The repair probability is selected to be an order of magnitude larger, corresponding to a mean time to repair of 10 cycles.

TABLE 2. Steady-State Probabilities of the Production Line Model (q=0. 01, r=0. 1).

Index	Station States	Steady-State Probability	Index	Station States	Steady-State Probability
15	**UUU**	0.74867172	2	DDS	0.00028882
25	BBD	0.07514296	5	DUS	0.00011103
24	BDS	0.06805575	14	UUD	0.00007917
8	DSS	0.06089789	28	**XDU**	0.00007585
16	UUS	0.015000090	1	**DDU**	0.00006823
4	**DUU**	0.00752210	20	UBD	0.00003962
23	**BDU**	0.00752203	17	USD	0.00003574
19	USS	0.00676627	13	UDS	0.00003209
7	**DSU**	0.00676627	12	**UDU**	0.00000758
18	**USU**	0.00075181	31	XXD	0.00000280
30	XBD	0.00039688	10	DXD	0.00000252
26	BXD	0.00039618	27	XDD	0.00000228
22	BDD	0.00035895	0	DDD	0.00000204
9	DBD	0.00035656	3	DUD	0.00000065
29	XDS	0.00032296	21	UXD	0.00000028
6	DSD	0.00032170	11	UDD	0.00000023

The states are arranged so that their steady-state probabilities are in decreasing order. Again, we mark in bold those states where the last machine is productive. It is seen that the steady-state probabilities display a great range of values. The state with the largest probability is UUU with a probability of almost 0.75. The state with the least probability is UDD with a probability of $2.3\text{x}10^{-7}$. The difference between the largest and smallest steady-state probability is over six orders of magnitude. That is, the ratio is on the order of a million to one. Clearly, if a simulation runs shorter than a few million cycles, the state UDD with the smallest probability is likely never be visited. This is analogous to not visiting states with more than, say, 100 customers during the simulation of an M/M/1 queue with a traffic intensity of 0.9.

3.1. A Truncated Model of the Three Station Production Line with No Buffers. Inspired by our findings in regarding the M/M/1 queue, now we consider a truncated model of the bufferless three-station line. We truncate the model by disregarding the states in which more than a single station is under repair. In effect, we are making the seemingly unrealistic assumption that once a station breaks down, the breakdown probability of the other stations drops down to zero. This may seem unreasonable, but it does follow the truncated M/M/1 queue case. There, we make the assumption that once there are some K customers in the system, the arrival rate drops down to zero.

The three-station case can easily be modified to find the production rate of the truncated model. One approach is to start with the steady-state probabilities as computed above. Then, we may remove those states with more than one station under repair, and re-normalize the steady-state vector. Afterwards, we re-compute the production rate.

The systems states with at most one station under repair are listed below.

TABLE 3. States with at Most One Station Under Repair.

Number of Stations Under Repair	Index	Station States
0	15	**UUU**
	16	UUS
	18	**USU**
	19	USS
1	4	**DUU**
	5	DUS
	7	**DSU**
	8	DSS
	12	**UDU**
	13	UDS
	14	UUD
	17	USD
	20	UBD
	22	BDD
	23	**BDU**
	24	BDS
	25	BBD

The state transition diagram illustrates the transitions among the systems states of the truncated model.

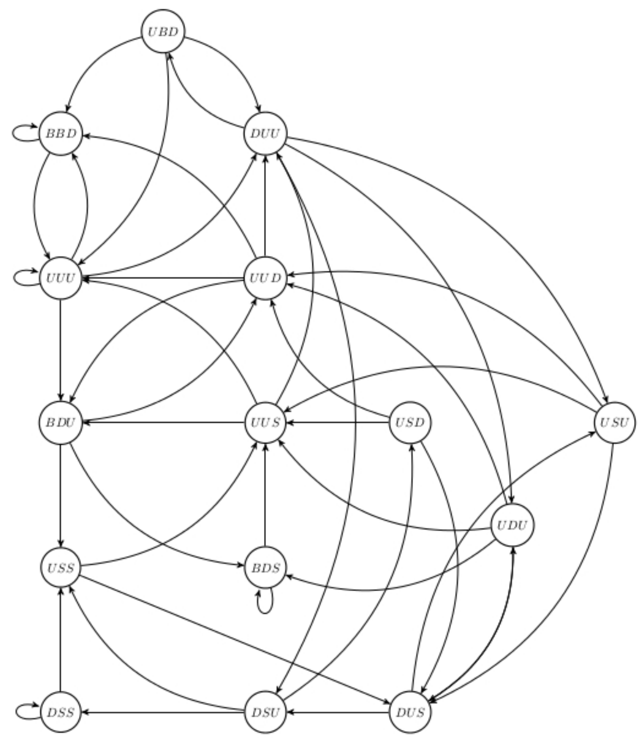

Figure 6. The State Transition Diagram of the Truncated Bufferless Three Station Model.

The states where the last station is operational are marked by bold letters. Note that of the eight such states, we now have only six. The normalization of the steady-state probabilities means we add the steady-state probabilities of the seventeen states shown in Table 3 and then normalize the vector by multiplying it with the reciprocal of the sum of its elements. The production rate is then computed as the sum of the normalized steady-state probabilities of the states shown in bold in Table 3.

The production rate of the truncated model will of course differ from that of the complete model. The question is by how much. We computed the difference for a series of parameters of a three-station bufferless line with identical stations. The differences are given as absolute percentage errors (APE). The absolute percentage error is computed as:

$$APE = 100 \cdot \left| \frac{\text{(production rate)} - \text{(production rate of truncated model)}}{\text{(production rate)}} \right|$$

The absolute percentage errors are plotted below.

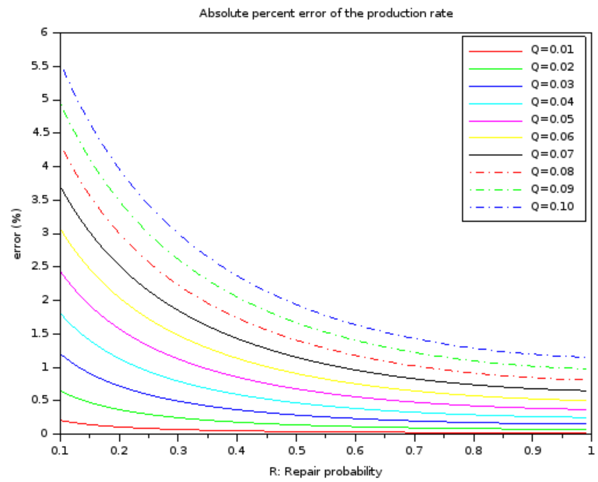

FIGURE 7. Absolute Percentage Errors as Functions of Model Parameters.

Figure 7 is rather remarkable. First, observe that the maximum error is about 5.5%. This is actually an extreme case, where both the breakdown and the repair probabilities are 0.1. In this extreme case, the stand-alone availability of a station is 50%. Clearly, in actual implementations, a station would be expected to be operational more than 50% of the time. For realistic cases, that is, where the stand-alone availability is around 90%, the error less than 1%. This is a remarkable phenomenon, that serves not only as an eye-opener, but as motivation to develop practical approximate production line models which are then to be solved algebraically.

3.2. A More Detailed Truncated Models of Production Lines with No Buffers. Once again, inspired by the results of the preceding section, we now proceed to construct models of production lines where the number of down machines are limited. Analogous to the truncated M/M/1 queue (the M/M/1/K queue), we make the auxiliary assumption that once K of the N stations are down, the remaining stations become perfectly reliable. Agreeably, this seems like an unjustifiable and quite counter-intuitive assumption. The justification lies in the insights hitherto developed and it is these insights which constitute the contribution of this thesis. In short, we want to remove some of the system states that have probabilities orders of magnitude smaller than others. Obviously, there are many ways to do this pruning. The assumption made here is one approach. It does have the advantage, however, that it is relatively straightforward to model this truncated system, just as it was relatively straightforward to implement the truncated M/M/1 queue.

It should be noted that, by its nature, the topic required dealing with larger sets of data, whose exact solutions are needed for the conceptual analysis. Accordingly, much effort was involved in code development. In particular, it is noteworthy that some source code are in the order of 1 gigabyte. Clearly, a gigabyte of code is not to be written manually. The approach here was to write code that in turn generates source code, later to be incorporated into downstream compiling and linking. The code development itself may also be considered as a contribution, as it provides a template or framework to the generation of multi-echelon software. It should also be noted that code development was done in open-source environments using only open-source tools.

Software was written to automatically generate the states and the transition probability matrix of bufferless production lines with N stations, but where at most K of the N stations are allowed to be down. The algorithm is explained in Appendix 2. We call this the N/K-Truncated-Model. Effectively, once K of the stations are down, the remaining N-K stations are assumed to be perfectly reliable. The states were explicitly obtained by the developed software. The software is explained in [6].

Table 4. Number of System States (N/K models, N stations, at most K down).

N/K	1	2	3	4	5	6	7	8	9	10
3	15	26	32							
4	40	92	116	128						
5	103	314	435	488	512					
6	257	1027	1594	1882	2000	2048				
7	623	3218	5665	7133	7833	8096	8192			
8	1476	9656	19454	26389	30267	31992	32576	32768		
9	3435	27858	64555	95041	114780	125140	129399	130688	131072	
10	7882	77694								524288

As seen in Table 4, the number of system states drop considerably when the maximum allowed number of down stations (K) is small. In the extreme case of K=1, we allow only one station to be down at a time. In case of a bufferless production line with 10 stations (N=10), the number of system states goes from 524288 to 7882. This is a 66-fold reduction in the number of system states.

The next investigative question, of course, is "*how well does the truncated model approximate the original model?*". Here, once again, we develop software to compute not only the transition probability matrices for the N/K truncated models, but also the production rates. The N/N model gives the production rate of the original models where the number of system states are several orders of magnitude larger. The N/1 truncated models are the approximations. Numerical results are given in the table below.

Table 5. Average Percentage Error of the N/1 Truncated Models.

N	Production Rate (q=0.01, r=0.1)		APE
	N/1 (truncated)	N/N (full model)	
3	0.5166273929	0.5142570532	0.46
4	0.5431647628	0.5388719616	0.80
5	0.5451422158	0.5387804217	1.18
6	0.5375021638	0.5290534472	1.60

As seen from Table 5 the N/1 truncated models provide good approximations to the production rate. The truncated models seem to always over-estimate the production rate. Moreover, the error term seems to increase almost linearly with the number of stations. This can also be used to fine tune the estimates, if our focus were to develop computational methods for the estimation of the production rate. However, our interest in this thesis is more on the conceptual side of the methodologies, as we investigate the ramifications of the models and their use.

4. Conclusions and Future Research

It was mentioned at the outset that our work is a preliminary investigation into the feasibility of approximations based on truncated state spaces. The inspiration comes from the apparent success of simulation. It seems that simulation naturally focuses on the system states that have a greater influence on the performance measures. Removing the system states which have negligible effects on the performance measures, we were able to duplicate the results of simulation.

Our work shows that very good approximations of the performance measures are possible with significant pruning of state spaces (66 fold reduction for a 10-station line). Thus, the current investigation provides the necessary justification to continue with the development of approximations based on truncated state spaces.

References

1. T. Altiok, *Performance analysis of manufacturing systems*, Springer, Science & Business Media, 2012.
2. R. G. Askin and C. R. Standridge, *Modeling and analysis of manufacturing systems*, Wiley, John & Sons Inc, 1993.
3. J. A. Buzacott and J. G. Shanthikumar, *Stochastic models of manufacturing systems (vol. 4)*, Prentice Hall, Englewood Cliffs, NJ, 1993.
4. Y. Dallery, R. David, and X. L. Xie, *An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers*, IIE Transactions **20** (1988), no. 3, 280–283.
5. Y. Dallery and S. B. Gershwin, *Manufacturing flow line systems: A review of models and analytical results*, Queueing Systems **12** (1992), no. 1-2, 3–94.
6. D. Durmus, *On the applicability of simulation as a verification tool for markovian models of production systems*, Master's thesis, Yaar University, Izmir, Turkey, 2016.
7. S. B. Gershwin, *An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking*, Operations Research **35** (1987), no. 2, 291–305.
8. ______, *Manufacturing systems engineering*, Prentice Hall, 1994.

9. S. B. Gershwin and I. C. Schick, *Analysis of transfer lines consisting of three unreliable stages and finite storage buffers*, Complex Materials Handling and Assembly-Systems **9** (1978).
10. ———, *Modeling and analysis of three-stage transfer lines with unreliable machines and finite buffers*, Operations Research **31** (1983), no. 2, 354–380.
11. S. Kayaligil, S. Yeralan, and K. Demirta, *Approximate performance analysis of a closed line with random failures by aggregation of segments*, Annals of the Faculty of Engineering Hunedoara-International Journal of Engineering **10** (2012), no. 1.
12. J. Li and S. M. Meerkov, *Production systems engineering*, Springer, Science & Business Media, 2008.
13. N. Maggio, A. Matta, S. B. Gershwin, and T. Tolio, *A decomposition approximation for three-machine closed-loop production systems with unreliable machines, finite buffers and a fixed population*, IIE Transactions **41** (2009), no. 6, 562–574.
14. E. J. Muth, *The reversibility property of production lines*, Management Science **25** (1979), no. 2, 152–158.
15. E. J. Muth and S. Yeralan, *Effect of buffer size on productivity of work stations that are subject to breakdown*, Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on . IEEE, 1981, pp. 643–648.
16. H. T. Papadopolous, C. Heavey, and J. Browne, *Queueing theory in manufacturing systems analysis and design*, Springer, Science & Business Media, 1993.
17. H. Tempelmeier and H. Kuhn, *Flexible manufacturing systems: Decision support for design and operation (vol. 12)*, John Wiley & Sons, 1993.
18. S. Yeralan and O. Buyukdagli, *The ontology of large-scale Markovian models*, SSMO **2015** (2015), 287.
19. S. Yeralan, W. E. Franck, and M. A. Quasem, *A continuous materials flow production line model with station breakdown*, European Journal of Operational Research **27** (1986), no. 3, 289–300.
20. S. Yeralan and B. Tan, *Analysis of multistation production systems with limited buffer capacity part 1: The subsystem model*, Mathematical and Computer Modelling **25** (1997), no. 7, 109–122.

Appendix 1 – The Maximum Length of the M/M/1 Queue

We consider the M/M/1 queue which starts from state 0 (empty system) at time 0 and runs for a given duration T. The number of customers in the system in the time interval [0, T] will vary as a function of the system parameters. We are interested in the distribution and the expected value of $N_{max}(T)$, the maximum number of customers in the M/M/1 queue during the time interval [0, T], given that the system is empty at time 0.

The transition rate matrix of the M/M/1 queue with an arrival rate of λ and a service rate of μ is given below.

$$\Lambda = \begin{bmatrix} -\lambda & \lambda & & & \\ \mu & -(\lambda+\mu) & \lambda & & \\ & \mu & -(\lambda+\mu) & \lambda & \\ & & \ddots & & \end{bmatrix}$$

Note that Λ is a square matrix of infinite size. The element (j, k) of Λ is the transition rate from state j to state k, where j differs from k. The diagonal elements of Λ are set to -(λ+μ) so that the rows sum to zero. The steady state probability row vector Π is computed as the normalized solution to the linear equation

$$\Lambda\Pi = [0, 0, 0, ...]$$

Normalization refers to setting the length of the vector Π so that its elements sum to unity. In vector notation, we may write,

$$\Pi u = 1$$

where u is a column vector consisting of all ones as shown below.

$$u = \begin{bmatrix} 1 \\ 1 \\ 1 \\ . \\ . \\ . \end{bmatrix}$$

The transient solution is easily obtained for the general Markov process. Let Π(t) be the state row probability vector at time t. Then,

$$\Pi(t) = \Pi(0)e^{\lambda t}$$

Given that the process starts in a state between 0 and N, the probability that the process stays with this range ([0, N]) is easily computed using the truncated state transition matrix. Let,

$$\Lambda_N = \begin{bmatrix} -\lambda & \lambda & & & \\ \mu & -(\lambda+\mu) & \lambda & & \\ & \mu & -(\lambda+\mu) & \lambda & \\ & & \ddots & & \\ & & & & \\ & & & & \mu - (\lambda+\mu) \end{bmatrix}$$

be a square matrix corresponding to the states 0 to N. Hence Λ_N has dimension N+1. Now the probability that the system is in state k where k is in the range [0, N] at time t is given as,

$$\Pi_N(t) = \Pi_N(0)e^{\lambda_N t}$$

Here, the subscript N, as in Λ_N indicates that this is a truncated vector, containing N+1 elements, corresponding to states from 0 to N. Let us assume that,

$$\Pi(0) = \Theta = [1, 0, 0, ...]$$

Then, the probability that the system is still in the range [0, N] at time t, given that it started in state 0 at time 0 is given by,

$$\Theta e^{\lambda_N t} u$$

where u is the column vector of ones as described above. This probability is the same as the probability that the state with the maximum index visited, N_{max} is in the range [0, N]. In other words,

$$P[N_{max}(t) \leq N] = \Theta e^{\lambda_N t} u$$

which is the probability distribution function of the random variable N_{max}. Successive differences of the distribution function yields the probability mass function of N_{max}. We compute the probability distribution and probability mass functions of N_{max}(t) using the built-in matrix exponentiation functions in Scilab. Below are the graphs showing the probability distribution and mass functions for the case λ=9, μ=10, t=1000.

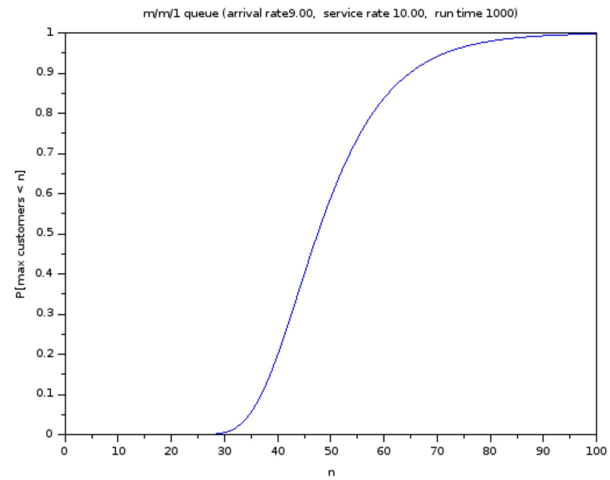

FIGURE 8. The Probability Distribution Function of N_{max} (λ=9, μ=10, t=1000).

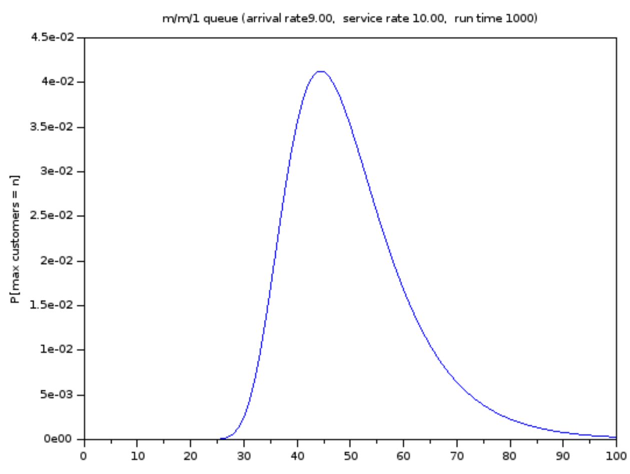

FIGURE 9. The Probability Mass Function of N_{max} (λ=9, μ=10, t=1000).

As expected, the probability mass function of N_{max} is skewed to the right, since the maximum has a lower bound of 0, but no upper bound.

The expected value of N_{max} is easily computed, as the probability distribution function is evaluated. Since N_{max} is a non-negative random variable, its expected value is simply the integral of its reliability function. Thus, keeping a tally of successive values of

$$P[N_{max}(t) > N] = 1 - \Theta e^{\lambda_N t} u$$

for N=1, 2, until the value reaches a threshold (say, 10^{-10}) gives the first moment of the random variable.

Appendix 2 – The Algorithm

The algorithm to generate the states and the state transitions is rather straightforward. It relies on the notion of what we call "intrinsic" or "direct" state transitions and "indirect" station transitions. Station state changes due to breakdown or repair constitute direct transitions. State transitions due to state changes of neighbouring stations are called indirect transitions. As an example, consider a two station line, where the system state makes a transition from UU to DU. Here, the transition of the first station is a direct transition, since the station experiences a breakdown. Note that the second station does not change its state. However, the system state transition from US to UU displays a indirect station transition of station 2. Here, the state of station 1 does not change. The state of station 2 changes because a part is now available. This is an example of indirect transitions.

At an step of the algorithm, we keep a set of possible system states. We step through these system states, considering breakdowns and repairs. These give rise to new system states. If the new system state is not already in our list, it is appended to the list we step through. As expected, the list initially grows quite rapidly, but then settles into its final set. Since there are a finite number of stations and station states, there are a finite number of system states. When we step through all of the system states and there are no new system states to append to the list the algorithm terminates.

The algorithm is modified for the truncated systems. Here we put an artificial limit on the total number of stations that are allowed to be down at any given time. This additional constraint, although complicates bookkeeping, does not pose any further intellectual challenges.

A summary of the algorithm steps is available in [6].

Deniz Durmus: Izmir,Turkey
E-mail address: denizdurmus91@gmail.com

Sencer Yeralan: Industrial Engineering, Yasar University, Izmir,Turkey
E-mail address: sencer.yeralan@yasar.edu.tr
URL: senceryeralan.org

Aybek Korugan: Industrial Engineering, Bogazici University, Istanbul, Turkey
E-mail address: aybek.korugan@boun.org.tr