

Production networks with stochastic machinery default

Simone Göttlich, Stephan Martin, and Thorsten Sickenberger

Abstract We present a model of production networks that includes random break-downs of individual processors. The defaults of processors are exponentially distributed and the time-continuous formulation of network dynamics yields a coupled PDE-ODE system with Markovian switching. Its solution is a piecewise deterministic process, which allows us to use a modified stochastic simulation algorithm to trace stochastic events and to simulate the deterministic behavior of the network between them. The impact of stochastic default is illustrated with an exemplary Monte-Carlo simulation.

1 Stochastic network model

Real world production planning in many cases no longer is a deterministic problem, but includes the consideration of stochastic events, such as machine failure (see e.g. [7]). Even though random failures are unavoidable by their very nature, the gathering of statistical data on e.g. mean availability of machines helps planners to deal with unexpected failure and to optimize production. In this work, we present a method to include stochastic failure into a mathematical network model for time-continuous supply chains.

Simone Göttlich

Dept. of Mathematics, TU Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany,
e-mail: goettlich@mathematik.uni-kl.de

Stephan Martin (Corresponding author)

Dept. of Mathematics, TU Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany,
e-mail: smartin@mathematik.uni-kl.de

Thorsten Sickenberger

Maxwell Institute and Heriot-Watt University, Dept. of Mathematics, Edinburgh EH14 4AS, UK,
e-mail: t.sickenberger@hw.ac.uk

We start from the network model originally discussed in [5]. Here, differential equations are used to model and study the dynamics of a production network. Each machine (processor) is represented by an arc e of given length l^e , and each distribution knot of the supply chain is represented by a vertex connecting at least two arcs. The topology of the supply chain is thus modeled using a directed, connected graph. The flow of products is modeled using a continuous product density rather than tracing individual items (see [2] for an up-to-date introduction to continuous models). The propagation of product density through a processor can be interpreted as monitoring the level of completion of the particular production process. Properties of processors $e \in \{1, \dots, M\}$ are modeled with (i) a finite processing capacity μ^e , and (ii) a finite processing speed v^e . In front of each processor, there is a buffer (queue) q^e , where products that temporarily cannot enter a processor are stocked.

In order to model machine default, we assume that breakdown times of individual processors as well as their repair times are exponentially distributed, such that the parameters τ_{on}^e and τ_{off}^e describe mean time between failures (MTBF) and mean repair time (MRT), respectively. From that, one obtains a stochastic process r^e taking values in $\{0, 1\}$ to model the evolution of operating and in-operating time periods of a processor, which depends on the time $t > 0$ as well as on the random sample $\omega \in \Omega$, whose argument ω is usually dropped. This process has been discussed in [3] in a related model for chains of processors. A realization or scenario of the multivariate state process $\mathbf{r}(t) = (r^1(t), \dots, r^M(t)) \in \{0, 1\}^M$ is given by a set of sampled switching points \tilde{t}_i^* , where one of the M processors changes its current state from operating to in-operating (1 to 0) or vice versa. These points can be sampled via an exponential distributed random variable with rate parameter $\lambda^{\text{sum}} = \sum_{e=1}^M \lambda^e$ where $\lambda^e = 1/\tau_{\text{on}}^e$ or $\lambda^e = 1/\tau_{\text{off}}^e$, respectively. It describes the length of the time period up to the next switching and the sampled switching points create a set $\mathcal{T}^* = \{\tilde{t}_0^*, \tilde{t}_1^*, \dots\}$ of strictly increasing times. Between these times the multivariate state process $\mathbf{r}(t) = (r^1(t), \dots, r^M(t)) \in \{0, 1\}^M$ is a random constant (see [1], [8]).

Having the signal of on and off states available, we include it into the network model by setting the capacity μ^e to zero, if the respective processor is currently in off-state. This means, that no more products are processed - no matter of their current level of completion inside the processor - and that no more products enter the processor. Arriving product inflow is stored in the respective queue. Once the processor gets back to work, production is restarted at the same stage it was stopped and no products are lost.

The network model reads as follows. Let $t \in [t_0, T]$. The density $\rho^e(x, t)$ of goods inside processor e satisfies the hyperbolic PDE

$$\partial_t \rho^e(x, t) + \partial_x f^e(\rho^e(x, t)) = 0, \quad x \in [a^e, b^e], l^e = b^e - a^e \quad (1a)$$

and the transport of goods along the arc e is described by the flux function

$$f^e(\rho^e(x, t)) = \min(v^e \cdot \rho^e(x, t), \mu^e \cdot r^e(t)), \quad (1b)$$

The queue-load $q^e(t)$ of each processor is modeled by the ODE

$$\partial_t q^e(t) = g_{\text{in}}^e(t) - g_{\text{out}}^e(t), \quad (1c)$$

describing the difference of all incoming and outgoing fluxes, which are designed as

$$g_{\text{in}}^e(t) = \begin{cases} A^{s(e),e}(t) \sum_{\bar{e} \in \delta_{s(e)}^-} f^{\bar{e}}(\rho^{\bar{e}}(b^{\bar{e}}, t)) & \text{if } e \text{ has predecessors} \\ G_{\text{in}}^{s(e)}(t) & \text{if } e \text{ has no predecessor} \end{cases} \quad (1d)$$

and

$$g_{\text{out}}^e(t) = \begin{cases} \min\{g_{\text{in}}^e(t), \mu^e \cdot r^e(t)\} & \text{if } q^e(t) = 0, \\ \mu^e \cdot r^e(t) & \text{if } q^e(t) > 0, \end{cases} \quad (1e)$$

respectively. $G_{\text{in}}^v(t)$ is the network inflow, s denotes a map identifying any arc with its vertex of origin, and $A^{s(e),e}(t)$ are control functions setting distribution rates at vertices. The PDE boundary condition couples the ODE and PDE via the outgoing flux function

$$\rho^e(a^e, t) = g_{\text{out}}^e(q^e(t))/v^e \quad (\text{PDE boundary conditions (b.c.)}). \quad (1f)$$

See [6] for a more detailed description and discussion of the model. In the present work, we restrict ourselves to constant distribution rates $A^{s(e),e}(t) := \alpha^{s(e)}$, e.g. the naive control, where outgoing products are equally distributed on all successive processors.

2 Numerical aspects

Realizations of the stochastic supply chain model are computed by a subsequent iteration of two algorithms: the simulation of switching points of the multivariate state-process \mathbf{r} and the deterministic computations between switchings. It is therefore related to the stochastic simulation algorithm introduced, see [4], to compute a scenario of the Chemical Master Equation, which is a hybrid ODE. However, in our approach the deterministic dynamics is modeled by a coupled ODE-PDE system. A sketch of the simulation algorithm reads as follows:

- (i) The sampling of the next switching points w.r.t. the exponential distribution of the multivariate state process can be done by drawing two random variables: First, an exponential distributed random variable to sample the next switching point and, second, a uniform distributed one to sample the next switching processor.
- (ii) Between sampled switching points $t \in [\bar{t}_i^*, \bar{t}_{i+1}^*]$ the solution of the coupled ODE-PDE system is approximated by an interaction of an ODE and PDE solver. The hyperbolic PDE is discretized by an upwind scheme, whereas the ODE is solved by the explicit Euler method.

An interesting part of the algorithm is the coupling of both equations through (1d), (1e): Careful attention has to be taken in sequencing the numerical iteration in a correct order, such that no mass is lost in during a switch. For details, we again refer

to [6]. Let us note, that the deterministic algorithm is completely parallelizable and can treat any network topology, including cycles and loops.

3 A case study: The diamond network

For the scope of this article, we want to illustrate the stochastic network model and the impact of default with a case study of averaged quantities obtained from a Monte-Carlo simulation. To do so, we study the diamond network illustrated in Fig. 1, which consists of 7 processors and two diversion controls α_1 , α_2 located in front of processors 2, 3 and 4,5 respectively. Processing rates are $\mu_1 = 40$, $\mu_2 = 30$, $\mu_3 = 20$, $\mu_4 = 20$, $\mu_5 = 5$, $\mu_6 = 10$, $\mu_7 = 10$, all processors share common length $l^e = 1$ and velocity $v^e = 1$. Stochastic default is set to $\tau_{\text{on}}^e = 0.95$,

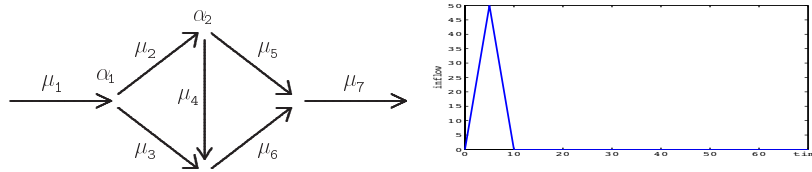


Fig. 1: Diamond network: 7 processors and two controls.

$\tau_{\text{off}}^e = 0.05$, except for processors 2 and 4, where $\tau_{\text{on}}^e = \tau_{\text{off}}^e = 0.5$. We fix the controls to $\alpha_1 = 0.6$, $\alpha_2 = 0.4$. In particular, the chosen setting creates several bottlenecks in the network, where the total capacity behind a network vertex is smaller than the possible maximum inflow. In total, we have a setting where the influence of stochasticity on the network's production performance can be hardly predicted without the use of simulation. One intuitive guess to predict the dynamics of the network could be as follows: Since we know the long-run average availability of each processor, the expectational capacity of each processor is given by

$$\mu_{\text{guess}}^e = \mathbb{E}[r^e \mu^e] = \frac{\tau_{\text{on}}^e}{\tau_{\text{on}}^e + \tau_{\text{off}}^e} \mu^e$$

and we can compute the stochastic dynamics using these capacities in a purely deterministic simulation. We include this computation in our example to demonstrate, that this intuitive guess clearly does not work. We simulate $N = 1000$ scenarios on a time interval $[0, 100]$ to ensure that all incoming product flow of $\int G_{\text{in}}(t) dt = 250$ units are completely processed, even in extreme events. In Fig. 2, we show queue loads in front of the processors over time. In order to illustrate the dynamics of the processor in Figure 3, we pick the middle point $\frac{b^e - a^e}{2}$ and show the time evolution of all quantities evaluated at that point. Looking at Fig. 2, we see that stochastic

queues can either significantly overtop deterministic ones (q2, q3, q4) or stay below deterministic peaks (q5, q6, q7), and take longer to be emptied. The intuitive guess predicts the emergence of queuing in q2 but fails to do so in q3, q4. In Fig. 3, processor emptying sees delay caused by stochasticity, which is considerably large in p2 and p4. The intuitive guess is never an approximation to the stochastic case and can even be worse than the deterministic setting, c.f. p2 and p4. Finally, the impact

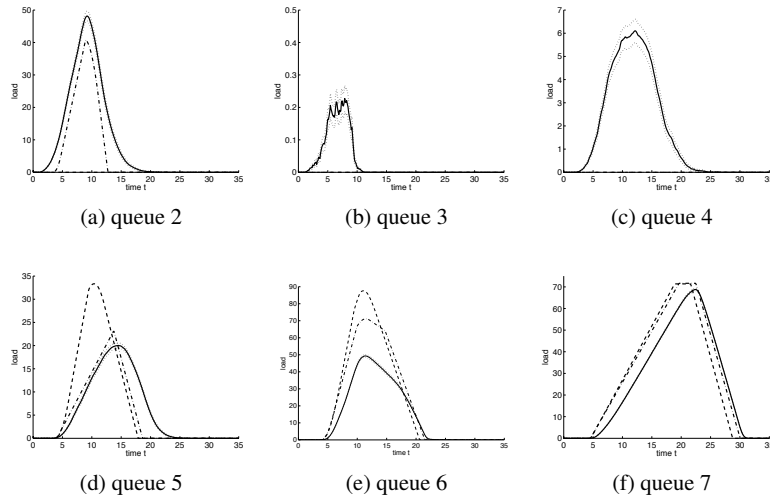


Fig. 2: Simulation of queue loads: (- -) deterministic, (-·-) intuitive guess, (-) stochastic MC, (··) 75% confidence interval.

of stochastic dynamics can be observed in the total production time. Figure 4 shows the probability distribution of total production time over 1000 scenarios. We can see that both the deterministic as well as the intuitive guess production time lie outside the stochastic distribution.

Acknowledgements

This work was supported by the BC/DAAD ARC project "Robust simulation of networks with random switching" (1349/50021880), the DFG grant HE 5386/6-1, and Leverhulme Trust grant F/00 276/K.

References

1. Davis, M.H.A.: Markov Models and Optimisation. Monograph on Statistics and Applied Probability 49, Chapman & Hall, London (1993)

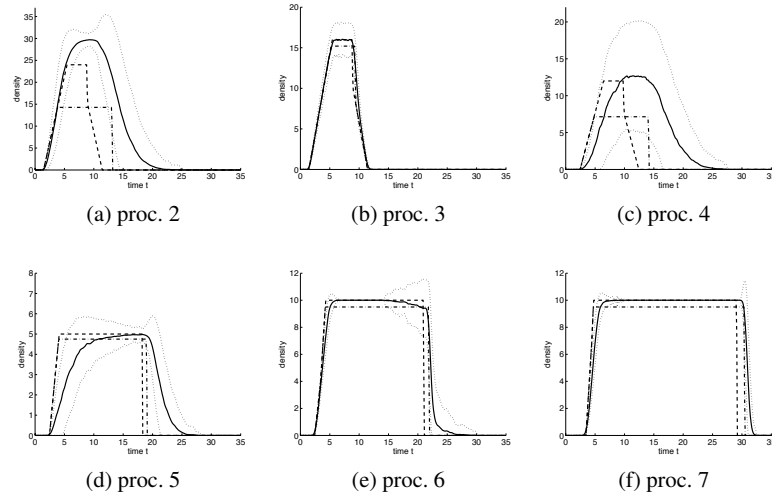


Fig. 3: Simulation of product densities: (--) deterministic, (-·-) intuitive guess, (-) stochastic MC, (··) 75% confidence interval.

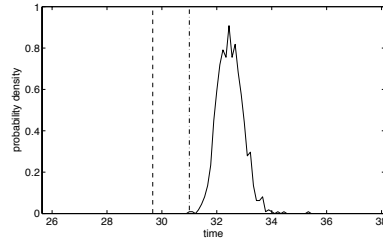


Fig. 4: Total production time: (--) deterministic, (-·-) intuitive guess, (-) stochastic MC

2. D'Apice, C., Göttlich, S., Herty, M., Piccoli, B.: Modeling, Simulation, and Optimization of Supply Chains: A Continuous Approach. SIAM (2010)
3. Degond, P., Ringhofer, C.: Stochastic dynamics of long supply chains with random breakdowns. SIAM J. Appl. Math. **68**(1), 59–79 (2007)
4. Gillespie, D.T.: A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Phys. Chem. A **104**, 403–434 (1976)
5. Göttlich, S., Herty, M., Klar, A.: Network models for supply chains. Comm. Math. Sci. **3**(4), 545–559 (2005)
6. Göttlich, S., Martin, S., Sickenberger, T.: Time-continuous production networks with random breakdowns. Heriot-Watt Mathematics Report HWM10-9, 21 pages (2010)
7. Kelly, F.P., Zachary, S., Ziedins, I. (Eds.): Stochastic Networks: Theory and Applications. Oxford University Press (2002)
8. Mao, X., Yuan, C.: Stochastic Differential Equations with Markovian Switching. Imperial College Press (2006)