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

A decomposition method for assembly/disassembly systems with blocking and general distributions

Jean-Sébastien Tancrez

Received: March 2018. Revised: October 2018, January 2019.

Abstract A modelling methodology is presented for assembly/disassembly systems with general processing time distributions and finite buffers. The approach combines the distributions discretization and a decomposition technique to analyze large manufacturing systems in a reasonable computational time and with good accuracy. In the decomposition technique, the system is decomposed into two station subsystems and the processing time distributions of the virtual stations are iteratively modified to approximate the impact of the rest of the network, adding estimations of the blocking and starving distributions. To analyze each subsystem, the general processing time distributions are discretized by aggregation of the probability masses, and the subsystem is then analytically modeled using a discrete Markov chain. We first show that this approach allows an accurate estimation of the subsystems cycle time distributions, which is crucial in the decomposition technique. Using computational experiments, we show that our decomposition method leads to accurate performance evaluation for large manufacturing systems (relative error on the order of one percent) and that the fine distribution estimation indeed seems to bring an improvement. Furthermore, we show on examples that, using decomposition, the cycle time distributions can be approximated reliably for large systems.

Keywords Manufacturing systems · Stochastic Model · General distributions · Finite buffers · Decomposition · Queueing networks.

Jean-Sébastien Tancrez
Université catholique de Louvain, CORE, Louvain School of Management
Chaussée de Binche, 151, 7000 Mons, Belgium
Tel.: +32 65 323 538
E-mail: js.tancrez@uclouvain.be

1 Introduction

In this paper, we are interested in the modelling of manufacturing systems with finite buffers and with general processing time distributions. We focus on the analysis of assembly/disassembly manufacturing systems (which can be modelled by fork-join queueing networks). An assembly/disassembly system is a feedforward open system (not closed) in which the stations are linked arbitrarily without forming loops. A station can have several input or output stations, but each buffer has exactly one upstream station and one downstream station (Dallery et al., 1994). When a processing ends in a station, one job is taken from each upstream buffer and one job is sent to each downstream buffer. We suppose that the system is manufacturing a single product type, with discrete parts (as opposed to continuous material). An example of assembly/disassembly system is given in Figure 1 (e.g. S_2 is a disassembly station and S_6 is an assembly station).

The processing time at a station (i.e. the time spent processing one job) is random, following a general probability distribution with a finite support (it does not have to be exponential or phase-type). The variability of the processing times may come from manual operations or from failures in unreliable stations (in which case, the time to failure and repair time can be included to the "effective process time", see Hopp and Spearman (1996)). However, by default, we suppose manual and reliable stations as it applies better when discretizing the distribution (see Section 3). We suppose an asynchronous transfer system, i.e. the stations do not have to start or stop their operations at the same time. Furthermore, we assume finite buffers. Blocking can thus occur: a station gets blocked if one of the next buffers is full when the station finishes its job (we suppose a blocking-after-service policy). Starvation of a station occurs when one of the previous buffers is empty, preventing the starting of a new job in the station. The system is supposed to be saturated, i.e. the first station is never starved and the last is never blocked (fictitious stations can be used to mimic the arrival or demand processes). Note that a pure flow line (which can be modelled by a tandem queue) is a particular case of an assembly/disassembly system, where stations follow each other sequentially, i.e. without assembly and disassembly stations. Our approach thus applies to pure flow lines. We do not consider split-and-merge systems, where items are routed rather that assembled/disassembled, and where one unit entering the system

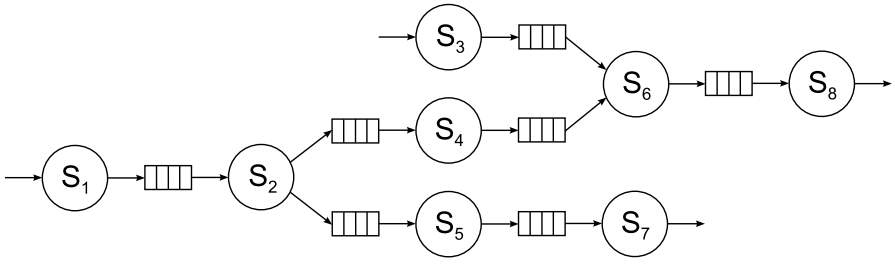

Fig. 1 Example of a assembly/disassembly system with finite buffers.

leads to exactly one unit leaving the system (see Bitran and Dasu (1992) or Govil and Fu (1999) for reviews and Tancrez (2009) for the application of PMF to split-and-merge systems).

In practice, the processing time in stations may follow a large range of distributions, depending on the particular application and on the collected data (in the form of histograms in particular). No exact model exists for manufacturing systems with general processing time distributions. The modelling process of such systems can be seen as made of three steps: data collection, distribution fitting, and analytical modelling. From the collected data, an analyst has to build distributions which are tractable for the analytical method, most often phase-type (PH) distributions so that the Markov theory can then be applied (Johnson and Taaffe, 1989; Bobbio and Telek, 1994; Bobbio et al., 2005). In our work, in order to transform the general distributions, we discretize them, applying probability mass fitting (PMF, Tancrez et al. (2011)). In the next step, i.e. analytical modelling, we apply a decomposition approach. In the decomposition technique, the manufacturing system is decomposed into virtual two station subsystems and the processing time distributions of the stations are iteratively modified to approximate the impact of the rest of the network, adding estimations of the blocking and starving distributions. This approach allows to efficiently analyze large systems (Dallery and Frein, 1993). We argue that PMF distribution fitting is particularly well suited for decomposition as it allows accurate estimations of the distributions and conserves their shape, and as these distribution estimations are crucial in the decomposition technique. Our aim in this paper is to combine two techniques (probability mass fitting and decomposition) and to show that these two techniques combine well, leading to accurate performance evaluation of large general manufacturing systems.

The remainder of the paper is structured as follows. In the next Section, we review the main analytical models for stochastic manufacturing systems. In Section 3, we present the PMF discretization of distributions and show how small manufacturing systems can then be modelled. In Section 4, we show how cycle time distributions can be computed in the PMF modified system and argue that they provide accurate estimations in the original system. The decomposition approach for assembly/disassembly manufacturing systems is then presented in Section 5. Its efficiency is shown using computational experiments in Section 6, and the estimation of the cycle time distributions is illustrated. Finally, we conclude in Section 7.

2 Literature Review

Various analytical methods have been proposed to model stochastic manufacturing systems and queueing networks. Several comprehensive reviews are available (Dallery and Gershwin, 1992; Buzacott and Shanthikumar, 1993; Altiok, 1996; Balsamo et al., 2001; Papadopoulos and Heavey, 1996; Govil and Fu, 1999; Li et al., 2009).

Most often, before applying the analytical method to evaluate the performance of a system, **distribution fitting** is needed. The processing time distribution are usually fitted to phase-type distributions. The first fitting method, historically and in popularity, is moments fitting.

This method computes the two (or three) first moments from the original distributions and the phase-type distributions are then built in order to get the same first moments. Various methods have been proposed, matching two or three moments, with various types of PH distributions, and for original distributions showing various properties. Sauer and Chandy (1975) (using hyperexponential and generalized Erlang distributions), Marie (1980) (using mixed generalized Erlang distributions) and Whitt (1982) (using hyperexponential and mixed Erlang distributions) were the first to propose closed-form two moments fittings. Altiok (1985) proposed to add the third moment, using mixed Erlang distributions with two stages, and gave the necessary conditions. Botta and Harris (1986) (using generalized hyperexponential distributions) and Johnson and Taaffe (1989) (using mixtures of Erlang distributions), extended the applicability of three moments fitting. Osogami and Harchol-Balter (2006) proposed a closed-form three moments fitting for any original distributions, using Erlang-Coxian distributions with an almost minimal number of phases. Subsequently, Bobbio et al. (2005) found a fitting with a minimal number of phases (number which depends on the properties of the distributions), using acyclic PH distributions. The second main type of method to approximate distributions is based on the computation of the maximum likelihood. In 1992, Bobbio and Cumani (1992) proposed such a method, fitting a Coxian distribution which maximizes the likelihood with the original general distribution. Later, Asmussen et al. (1996) made a significant contribution by applying the EM (expectation maximization) algorithm for the maximum likelihood estimation. The fitting of discrete phase-type distributions has been less investigated than the fitting of continuous PH distributions. Concerning moments fitting, Adan et al. (1994) proposed a two first moments fitting method and Telek (2000) gave the necessary conditions in terms of the second moment. The likelihood maximization was applied by Bobbio et al. (2003) to fit acyclic discrete PH distributions (with infinite support). In this paper, we apply probability mass fitting (PMF, Tancrez et al. (2011)) which aims at mimicking the shape of the distribution with a discrete phase-type distribution (see Section 3).

Using the phase-type processing time distributions as an input, two kinds of analytical models can be applied: exact or approximate. **Exact analytical models** are the richest since they allow a direct and exact understanding of the influence of a decision variable on the performance of interest. Closed-form models are only available for very simple configurations, e.g. two station lines. State models build continuous (sometimes discrete) Markov chains to exactly analyze systems with exponential or phase-type (PH) processing time distributions. Based on the identified state space, a transition matrix is derived and the stationary equations are solved numerically to obtain the steady-state probabilities and infer the performance measures. The reader is referred for example to Hillier and Boling (1967) for an early contribution, or Gourgand et al. (2005) for the study of lines with exponential distributions and infinite buffers. However, these models' applicability is limited to small instances as the state space size of the Markov chain increases quickly with the system size. State models of small networks serve as a building block

for the decomposition method (see below). Such two machine models are provided e.g. by Tan and Gershwin (2009); Colledani (2013); Liu et al. (2012).

However, most stochastic manufacturing systems are too complex to be modeled exactly. Exact models suffer from their complexity, which makes their applicability limited to small systems. As a result, **approximate analytical models** have been proposed. The system to be analyzed is simplified in order to be analytically modeled. While closed-form models exist (Buzacott and Shanthikumar, 1993; Hopp and Spearman, 1996), as well as methods such as the generalized expansion method (Kerbache and Smith, 1987, 1988) and the aggregation method (Terracol and David, 1987; de Koster, 1987), the most popular and successful are based on the idea of decomposing the system into smaller subsystems, and then including back the interdependencies between the subsystems. The decomposition techniques decompose the network into smaller subsystems (made of one, two or three stations), analyze them in isolation and then mix the results iteratively. Solving more, but much easier, subproblems, allows to approximately analyze the global system much more quickly and with good accuracy. A set of equations that determines the unknown parameters and the links between the subsystems is first derived. An iterative procedure is then used to solve the equations. This method was initially created for systems with exponentially distributed processing times (see Dallery and Frein (1993); Huisman and Boucherie (2011) for surveys and Gershwin (1987); Brandwajn and Jow (1988); Helber (1999) for early examples), and was then extended in many ways. Altiok (1989); van Vuuren and Adan (2006); and Helber (2006) extended the method for continuous phase-type distributions, while Gun and Makowski (1987) did for discrete PH distributions. Helber (1998) and Jeong and Kim (1998) extended the decomposition method to assembly/disassembly systems, with unreliable machines.

After these early contributions, many studies followed, improving the decomposition methodology, and applying it to many different settings. Tempelmeier and Burger (2001) proposed a decomposition method for flow production systems with finite buffers and generally distributed processing times. Krieg and Kuhn (2002) developed a decomposition method for kanban controlled production systems with multiple products and setup changes between products, where setup times and processing times are exponentially distributed. Kuhn (2003) investigated the interdependency between production and maintenance on an automated flow line system. Levantesi et al. (2003) studied production lines with multiple failure modes, and considered real and virtual failure modes for each building block. Manitz (2008) studied assembly systems with general production time distributions (G/G/1/N stations), and relied on estimations of the arrival rates, service rates,and coefficients of variation to build the decomposition equations. Manitz (2015) then extended the approach to assembly/disassembly systems. Colledani and Tolio (2011) proposed a decomposition technique for production systems where the quality control is integrated, in order to improve the design of such systems. Kim (2011) adapted the decomposition technique to take autocorrelation and cross correlation into account. Colledani and Gershwin (2013) used two-stage Markovian fluid models as a building block, to propose a decomposition technique for

continuous flow lines with machines characterized by general Markovian fluid models and finite capacity buffers.

Most of this literature, on the modelling of stochastic manufacturing systems, focuses on the estimation of first moment performance measures. In this paper, we also study the evaluation of distributions, besides expected values. More specifically, we look at the distribution of the cycle time, i.e. the time passing between two jobs exiting a given station (also called the inter-departure time). Papers of note which have investigated the estimation of variances and distributions in other contexts include the following. Leemans (2001) and Tan (2003) were among the firsts to propose methodologies to evaluate distributions, relying on state models. Leemans (2001) applied matrix-geometric methods to derive the queue length distribution of a Markovian two-class two-server queue, and then proposed an algorithm to compute the waiting time distribution. Tan (2003) proposed an algorithm to generate the state space model of unreliable production lines controlled by various pull policies (kanban, constant WIP, control point or base stock), and then derived the distributions of various performance measures, notably the cycle time. More recently, Assaf et al. (2014) focused on the variance of performance measures such as the cumulated output or the cycle time on small production systems with unreliable machines. Lagershausen and Tan (2015) proposed a method to compute the exact distributions of the cycle time for closed queueing networks with phase-type distributions. Shi and Gershwin (2016) studied the distribution of the time spent by one part in a two-station one-buffer line with unreliable machines, and then extended their method to longer lines using a decomposition approach.

In this paper, we propose a decomposition method for assembly/disassembly manufacturing systems with general distributions. The main contribution of this paper is in matching-up the PMF discretization with the decomposition method to evaluate the performance of large systems, and in showing that these two approaches indeed combine well, leading to accurate performance evaluation. The reason is that PMF discretization allows fine cycle time distribution estimations, and that these distributions are crucial in the application of the decomposition technique. Furthermore, another originality of this research is in the accurate cycle time distribution estimation itself, and in showing that it stays mostly true after applying the decomposition. The cycle time distribution offers a much more detailed information than the usual average cycle time.

3 Probability Mass Fitting

As explained previously, in order to model manufacturing systems with general processing time distributions, tractable distributions have first to be built, before applying analytical methods (e.g. a decomposition method). A discrete Markov chain can be built if the processing time distributions are discrete with regular intervals. Accordingly, the idea of probability mass fitting (PMF) is simply to aggregate on regular discrete values the original probability mass around them. In few words, PMF transforms a given distribution into a discrete one by aggregating the probability mass distributed in the interval $((j-1)\tau+\alpha, j\tau+\alpha]$ on the point $j\tau$, for $j = 2, \ldots, a$,

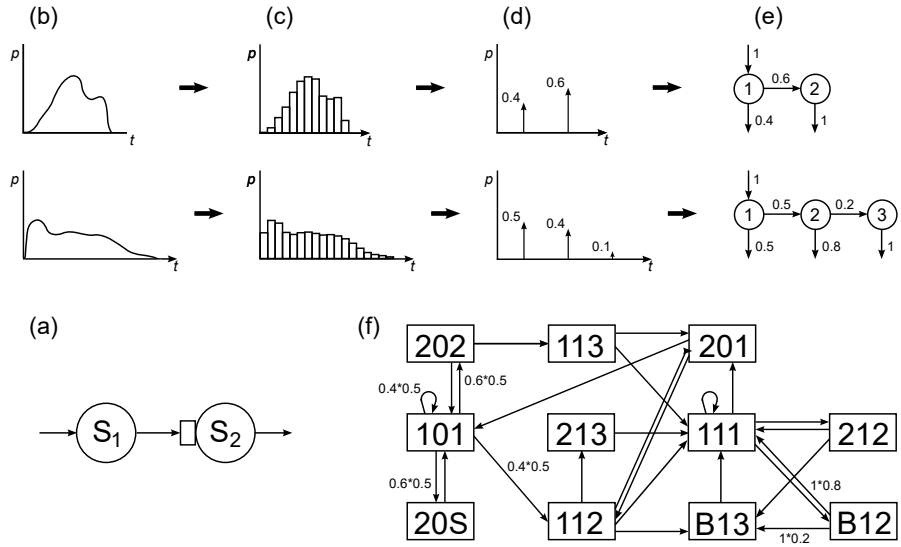

Fig. 2 Modelling of a two station line with a buffer of size one (a), with processing times (b) collected in the form of histograms (c): construction of the tractable discrete distributions by PMF (d, with $a = 2$ for the first distribution, $a = 3$ for the second), PH representation (e) and Markov chain modelling the evolution of the system (f).

and the mass in $[0, \alpha]$ on τ. Increasing the number of discrete values (parameter a) will improve the accuracy of the probability fitting but deteriorate the tractability of the model.

PMF is illustrated in Figure 2 on a two station line (Figure 2.a). An original distribution (Figure 2.b) is transformed into a discrete distribution (Figure 2.d) simply by aggregating the probability masses on discrete values ($a = 2$ for the first distribution, $a = 3$ for the second) with regular distances between them (parameter τ). Here, the probability masses are aggregated in the middle of the interval, i.e. $\alpha/\tau = 0.5$.

From the transformed discrete distributions, the behavior of the system can be modeled by a discrete Markov Chain, i.e. using a state model. The Markov chain given in Figure 2.f lists all the possible recurrent states of a two station line and the transitions between these states. The first symbol of a state refers to the first station, the second to the buffer and the third to the second station. Each station can be starved (S), blocked (B) or in some stage of processing (for example, 1 means that the station already spent one time step working on the current job). Each buffer is described by its content (0 or 1 here, as the line has a buffer of size one). For example, two transitions are possible from 113, depending if the first station continues the same job or finishes it. In the first case, the new state will be 201 (the second station ends and takes the job from the buffer). The probability of this transition is 0.6, given in the PH representation of Figure 2.e. In the second case, the first station ends its job and places it in the buffer, the new state is thus 111. From the Markov chain and its stationary probabilities, the performance measures can be computed. For example, the throughput of a station is given by the production rate of this station (known from the processing time distribution of the station) times the probability

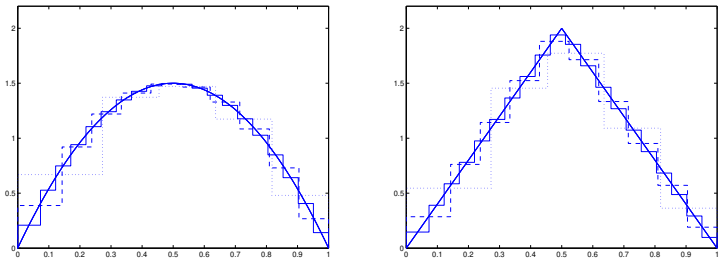

Fig. 3 Original continuous distributions (beta(2,2) and triangular(0,1,0.5)) and discretized distributions by PMF ($\alpha/\tau = 0.5$), with $a = 5$, 10 and 20 discretization steps.

of the station to be working (which is simply the sum of the stationary probabilities of all the states where the station is working, not starved or blocked). In the example of Figure 2, the throughput of the second station is its production rate times the probability it is working (one minus the stationary probability of state 20S). This methodology can be applied to model virtually any manufacturing system, but is limited to small networks (less than five stations) due to the explosion of the Markov chain state space and the prohibitive computational effort required for larger systems (Tancrez et al., 2011).

Probability mass fitting has various advantages. First, it is simple and intuitive. It can be refined (decreasing the interval size τ, or, equivalently, increasing the number of discrete values a) and, at the limit, it tends to the exact distribution. The accuracy of the approximation can be chosen according to the affordable computational effort. Other characteristics of PMF come from the fact that it builds discrete phase-type distributions (and not continuous). Bobbio et al. (2004) summarized the advantages of discrete PH distributions: their ability to approximate original distributions with a low coefficient of variation, with abrupt changes, with finite support, or with deterministic values. Neuts (1981) argues that continuous phase-type distributions are better suited to model heavy tails. He however notes that continuous PH distributions are not well suited to model delayed distributions and distributions with steep increases or decreases. These characteristics imply that PMF is better suited to model reliable stations with manual operations. This source of uncertainty leads to processing time distributions on a rather small finite support, which can be efficiently fitted using PMF (with a small number of discrete values a). On the opposite, failures and effective process times including the repair time (Hopp and Spearman, 1996) will lead to a larger distribution support and will thus be less efficiently modelled using PMF. This is why we suppose manual reliable stations by default (see Section 1).

In this paper, we profit in particular from another valuable advantage: probability mass fitting preserves the shape of the distribution. In fact, it could have been called “shape fitting”. This point can even be thought as a motivation of the idea of PMF. It is illustrated in Figure 3. Moreover, in the next Section, we show that this property stays true after applying the state model: the shape of the cycle time distribution is conserved. This characteristic, the shape

conservation, makes a valuable difference with moments fitting methods, for example, which sums up the information in two or three numbers and looses information about the shape of the distribution or the percentiles (Pearson et al., 1979). In the literature, visual comparisons of the distribution functions is a common way to evaluate the adequacy of the fitting (see for example the extensive distribution fitting methods comparison study by Lang and Arthur (1997)).

4 Cycle Time Distributions

In this Section, we show that, using probability mass fitting, shape conservation does not only hold for the original processing time distributions of the stations but also holds for their cycle time distributions (i.e. including blocking and starvation times). While the literature focuses on the expectation of performance measures, complementary information regarding their distribution can be very useful in order to reflect the stochastic nature of a manufacturing system. Furthermore, a detailed and reliable estimation of the cycle time distribution of a station is crucial in the application of the decomposition technique (see Section 5).

4.1 Computation

To begin, we explain how the cycle time distribution of stations of small manufacturing systems can be computed from the modelling presented in the previous section. By the cycle time distribution of a station, we mean the distribution of the time passed between two jobs exiting a given station. The cycle time distribution is possibly different for each station, and differs from the processing time as it includes the blocking and starvation times. In the discretized time, the cycle time distribution is a discrete phase-type distribution, and it can be modelled as the time until absorption of a discrete Markov chain with one absorbing state. It is fully characterized by the transition matrix of the Markov chain and the initial probabilities for starting in any of the states of the chain (Neuts, 1981).

The transition matrix characterizing the distribution of the cycle time of a station is deduced from the Markov chain modelling the evolution of the system (see Figure 2.f). In this Markov chain, a transition from a state where station i is working or blocked to a state where station i is starved or in its first stage of processing (symbol 1) corresponds to one unit leaving station i. These transitions, from $[1, 2, \ldots, a$ or $B]$ to $[S$ or $1]$, are highlighted (dashed) in the left-hand side of Figure 4, for the second station. They are the transitions that have to be modified to get the transition matrix of the cycle time PH distribution of station $i = 2$. The departure state of these transitions stays the same but the transition goes to the absorbing state of the Markov chain characterizing the PH distribution. Informally said, reaching the absorbing state means that one item leaves the station, and that the cycle time thus ends. The Markov chain of the PH cycle time distribution (for the second station) is given in Figure 4 for a two station line. This

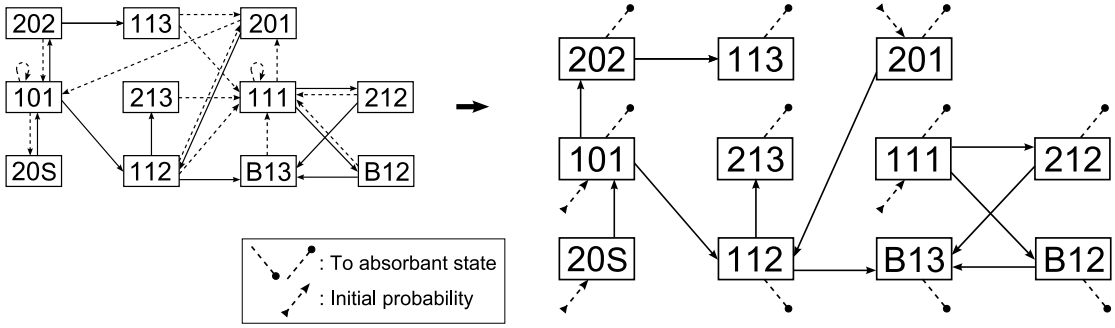

Fig. 4 Markov chain modelling the evolution of two station line with a buffer of size one (left-hand side), and Markov chain associated with the phase-type distribution of the cycle time of the second station (right-hand side).

illustrates how the Markov chain modelling the evolution of the system, given in the left-hand side, is modified to get the phase-type distribution.

Besides the transition matrix, the other information needed to characterize a phase-type distribution is the initial probabilities of starting in any of the chain states. In our case, the transitions mentioned previously (from $[1, 2, \ldots, a$ or $B]$ to $[S$ or $1]$) correspond to the end of one cycle time and also to the beginning of the next one. The initial probabilities are thus positive for the destination states of these transitions, and equal the probability for a cycle time to start in these particular states. The initial probability of a state s is the sum, for all these transitions, of the stationary probability of the departure state of the transition multiplied by the transition probability to s. The vector of initial probabilities $\mathbf{p_{init}}$ can be computed as follows:

$$\mathbf{p_{init}} = \frac{\boldsymbol{\pi}\ \mathbf{P_{out}}}{\boldsymbol{\pi}\ \mathbf{P_{out}}\ \mathbf{e}}\ ,$$

where $\mathbf{P_{out}}$ is the matrix of the transition probabilities from $[1, 2, \ldots, a$ or $B]$ to $[S$ or $1]$ (on station i), i.e. corresponding to the beginning of a cycle time, $\boldsymbol{\pi}$ is the vector of the stationary probabilities of the states of the Markov chain modelling the evolution of the system, and $\mathbf{e}$ is a column vector of ones.

The cycle time distribution of any station can thus be computed in the PMF modified system. It is a discrete phase-type distribution, characterized by its transition matrix and its initial probabilities. The computation does not require any additional matrix inversion. Thanks to a mature theory (Neuts, 1981), the probability density function can be formulated, and the moments can be computed, for example.

4.2 Shape Conservation

In the previous Section, we show that the cycle time distribution of any station can be computed in the modified system with PMF discretized distributions. We now argue that it is a good approximation of the cycle time distribution in the original system with general distributions. In

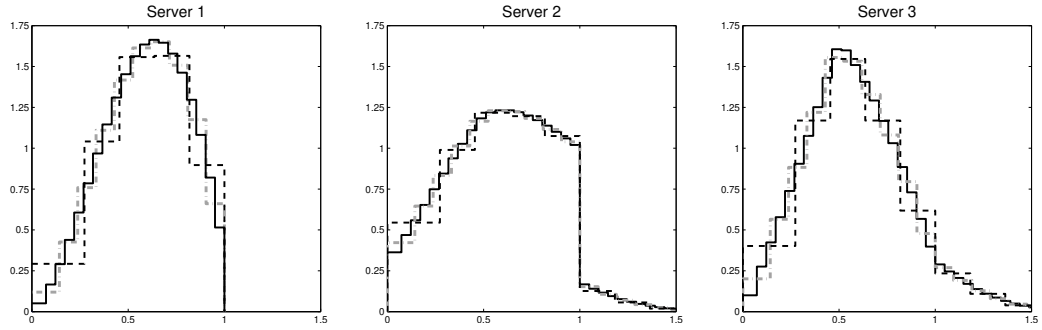

Fig. 5 Cycle time distributions computed, using PMF ($\alpha/\tau = 0.5$), for the three stations of a line (beta(2,2), uniform(0,1) and triangular(0,1,0.5) processing time distributions), with 5 (dashed), 10 (dash-dot, gray) and 20 (solid) discretization steps.

other words, the shape conservation of the processing time distributions (see Figure 3) extends to the cycle time distributions.

We first have a look at the following illustrative manufacturing system: a three station line with buffers of size one and processing time distributions beta$(2, 2)$, uniform$(0, 1)$ and triangular$(0, 1, 0.5)$. Figure 5 depicts, for each station, the computed discrete cycle time distribution. The three graphs show how starving and blocking impact the cycle time. The distributions are different from the beta, uniform and triangular processing time distributions (see Figure 3). The cycle time distributions are computed with 5, 10 and 20 discretization steps. As our PMF distribution fitting refines and converges when the number of steps increases, it can be supposed that the cycle time distribution computed with 20 steps is accurate. Figure 5 reveals that the shape of a cycle time distribution appears to be independent of the number of discretization steps used. The distribution is of course more detailed with 20 steps but the distributions computed with 5 or 10 steps show the same shape. It tends to show that the cycle time distribution estimations with 5 or 10 steps are already good.

To further convince of the accuracy of the cycle time distribution approximation, we compare the computation(PMF with $\alpha/\tau = 0.5$ and $a = 10$) to the result of a simulation, on a three station line (uniform distributions, zero storage space), for the third (end) station. The results are given in Figure 6. The distributions, their shapes and values, show to be remarkably similar. As another example, let us consider an assembly system with four stations. The two first stations are in series, and the fourth station assembles the jobs coming from the second and third stations. The buffer sizes equal one except for the buffer between the third and fourth station, whose size is zero. The processing time distributions are beta(2,2), uniform(0,1), triangular(0.1,0.9,0.6) and beta(5.5,6). Figure 7 shows that the cycle time distribution estimation (fourth station) is very close to the simulation result. Note that many other examples were tested, showing the same behavior (see Tancrez (2009)).

In conclusion, we may say that the cycle time distributions computed by our modelling method have a shape close to the original ones (with general processing time distributions). The

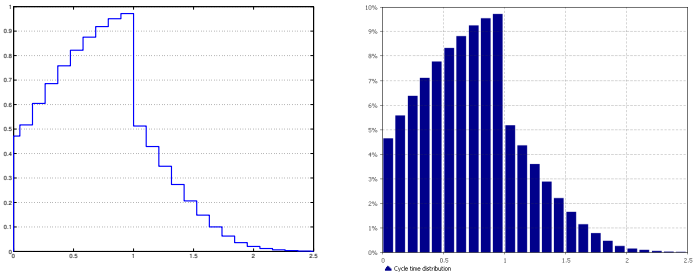

Fig. 6 Cycle time distribution of the third and last station of a line, with uniform processing time distributions, computed using PMF (left) and computed via simulation (right).

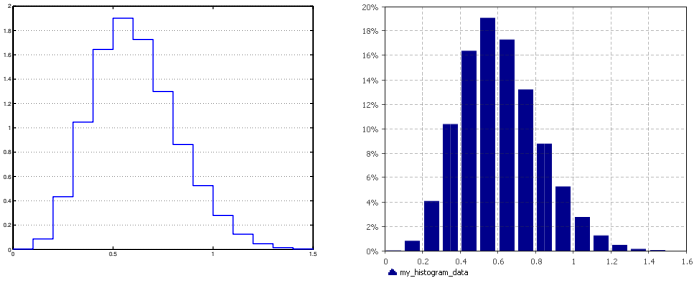

Fig. 7 Cycle time distribution of the fourth and last station of an assembly system, with various processing time distributions, computed using PMF (left) and computed via simulation (right).

computation of a good cycle time distribution estimation is a significant advantage, compared to other fitting methods which are unable to do so, such as moments fitting in particular (Pearson et al., 1979). The distribution offers more detailed information on the behavior of the system, compared to the isolated expectation. It allows to estimate measures such as the variance or the percentiles, and is crucial in the application of the decomposition technique, as explained in the next Section.

5 Decomposition

As any exact model, the modelling method presented in Section 3 and illustrated in Figure 2 has one main weakness: its complexity, which limits the applicability of the method. To model larger systems, we study the application of decomposition (see Section 2) after probability mass fitting. With decomposition, the computational time is drastically decreased, compared to a state model. We implement the method, following standard principles for decomposition methods, for lines and assembly/disassembly manufacturing systems. A system is decomposed into two (virtual) station subsystems, which are analyzed using the state model presented in Section 3. Then, the processing time distributions of the virtual stations are iteratively modified to approximate the impact of the rest of the network, adding estimations of the blocking and starving distributions.

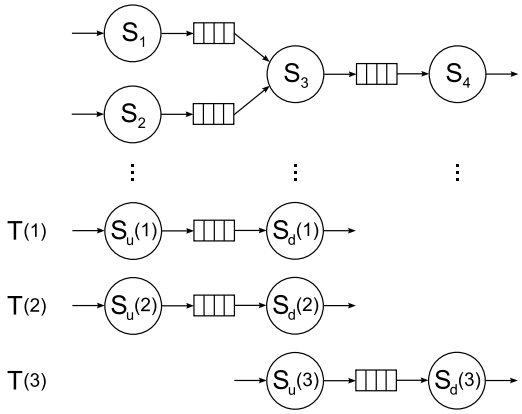

Fig. 8 Decomposition of an assembly system.

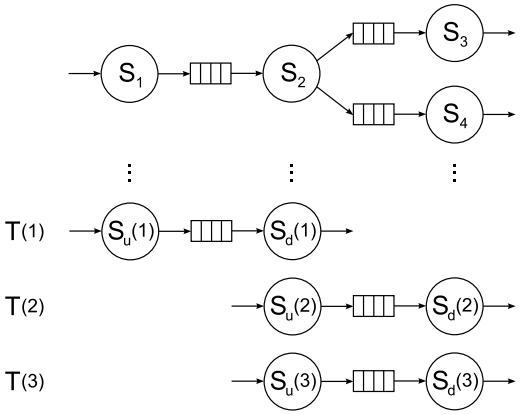

Fig. 9 Decomposition of a disassembly system.

In this context, the ability (shown in Section 4) of PMF to reliably approximate the cycle time distributions is a valuable feature of PMF for the application of the decomposition method. The cycle time of one station is the average time between the departures of two units from the station. The cycle time distribution thus derives from the processing time distribution but is modified due to the rest of the system (for example, the cycle time distribution in Figure 6 is a modification of the uniform processing time distribution). Starvation time is added due to the previous stages and blocking time is added due to the blocking time. From the estimation of the cycle time distribution of a station, and from its processing time, it is thus possible to deduce estimations of the blocking and starvation time distributions (even conditional distributions). As mentioned previously, those distributions are used to update the processing time distributions of the virtual stations in the decomposition technique. The reliable estimations of the blocking and starvation time distributions provided by PMF are thus valuable when applying the decomposition technique. It hints that PMF should well with the decomposition technique (it is shown with experimental results in Section 6).

The decomposition technique we implement decomposes the systems into two station subsystems (similar to Dallery and Frein (1993)), as illustrated in Figures 8 and 9. The original system is denoted T and the subsystems are denoted $T(i)$ ($i = 1, 2, \ldots, m_b$, with m_b the number of buffers). There is one subsystem for each buffer in the original system. The buffer sizes of the subsystems are kept equal to the corresponding original buffer sizes. A subsystem $T(i)$ is made of two "virtual" stations: an upstream station $S_u(i)$ and a downstream station $S_d(i)$. The processing times of the virtual stations are modified so that the flow of items through the subsystem buffer is close to the flow in the original buffer (almost the same arrivals and departures, starvations, blockages, and buffer levels). In other words, upstream and downstream stations of each two station subsystem summarize the effects on the buffer of the entire upstream portion of the original system and the entire downstream portion of the original system, respectively (Dallery and Gershwin, 1992).

Compared to an isolated two station subsystem, in the original system the upstream portion may cause starvation in a station, if this portion has been comparatively slow. Consequently, we have to modify the processing time distribution of the virtual station $S_u(i)$ to include the omitted starvation time. The latter can be deduced from the previous subsystem $T(i-1)$ (potentially more than one for an assembly station). Indeed, a starvation of station S_i in the original system corresponds to a starvation of the second virtual station $S_d(i-1)$ in the previous subsystem $T(i-1)$. The processing time distributions of stations $S_u(i)$ are thus iteratively updated, starting from the beginning of the system to its end ($i = 1, 2, \ldots, m_b$), using the information from the previously computed system $T(i-1)$ and station $S_d(i-1)$. In the particular case of an assembly station, the starvation may be caused by several stations (each of its predecessors). The omitted starvation time to be added is thus deduced from several subsystems, namely the subsystems corresponding to the preceding buffers. An example is given in Figure 8. On this example, the algorithm first analyzes subsystems $T(1)$ and $T(2)$. This allows to evaluate the starving caused by S_1 and S_2 on the assembly station S_3. Subsequently, the starving caused by both predecessors can be consistently[1] added to the processing time of the virtual station $S_u(3)$. The subsystem $T(3)$ can then be analyzed, and so on.

Similarly, compared to an isolated two station subsystem, in the original system the downstream portion may cause blocking in a station. The processing time distribution of $S_d(i)$ has thus to be modified in order to include the omitted blocking time, which corresponds to the blocking time in the first virtual station $S_u(i+1)$ of the next subsystem $T(i+1)$. The processing time distributions of stations $S_d(i)$ are thus iteratively updated, starting from the end of the system ($i = m_b, m_b - 1, \ldots, 1$), using the information from the previously computed system $T(i+1)$ and station $S_u(i+1)$. In particular, a disassembly station can be blocked by each of its successors, and the omitted blocking time to be added is thus deduced from the subsystems corresponding to all the succeeding buffers. In Figure 9 for example, the algorithm analyzes subsystems $T(2)$ and $T(3)$ in order to estimate the blocking caused by S_3 and S_4 on the disassembly station S_2. Subsequently, the blocking caused by both successors is consistently[2] added to the processing time of the virtual station $S_d(1)$. The subsystem $T(1)$ can then be analyzed, and so on.

All in all, the processing time distributions of the virtual stations are thus updated iteratively, in a loop. The algorithm first goes forward and updates the processing times of the upstream stations $S_u(i)$ (from the starvation time of previous S_d stations). It then goes backward and updates the processing times of the downstream stations $S_d(i)$ (from the blocking time of next S_u stations). The processing time distributions of the subsystems $T(i)$ are thus progressively updated, in several loops, and the algorithm converges to an approximate model of the original system T. This approximate model (each of the resulting subsystems) can then be used to

[1] If S_3 is starved by two preceding stations, the starving ends when each station has ended its job. The starvation time to be added to $S_u(3)$ is thus the maximum of the starvation times computed on $S_d(1)$ and $S_d(2)$.

[2] If S_2 is blocked by two successors, the blocking ends when each of them has ended its job. The blocking time to be added to $S_d(1)$ is thus the maximum of the blocking times computed on $S_u(2)$ and $S_u(3)$.

a	$m=2$	$m=3$	$m=4$	$m=5$	$m=10$	$m=20$
4	0.1 *(0.1)*	0.2 *(0.1)*	0.2 *(0.3)*	0.3 *(4.8)*	0.7	1.1
6	0.1 *(0.1)*	0.2 *(0.2)*	0.3 *(1.2)*	0.4 *(41)*	0.8	1.3
8	0.1 *(0.1)*	0.3 *(0.3)*	0.4 *(4.7)*	0.4 *(240)*	0.8	1.6

Table 1 Computational time (in seconds) of the decomposition method, using PMF, for lines with a total storage space equal to two. The number m of stations and the number a of discretization steps vary. The computational time of the exact state model is given under parenthesis.

estimate the performance measures of the original system. For example, the throughput of the last station (when the algorithm converged, the throughput is equal for each of the subsystems), can be computed on the second virtual (modified) station of the last subsystem $T(m_b)$, exactly as for an exact state model (see Section 3).

In our case, the processing time distributions are discrete phase-type. This allows to compute finer estimations of the starving and blocking times to be added in the virtual station processing times. We are able to compute a discrete distribution of the starving (resp. blocking) time, which basically gives the probabilities for the starving (or blocking) time to equal $n\tau$ ($n = 1, 2, \ldots, a$). Furthermore, the estimation can still be refined, as we can in fact compute the conditional probability of a discrete starving (blocking) time, knowing the value of the preceding processing time. In other words, we are able to compute the probability for the starvation time to equal $n\tau$, knowing that the preceding processing time equals $n'\tau$. In short, we are able to compute fine conditional blocking and starvation time distributions, while these distributions are crucial in the decomposition technique, and, as seen in Section 4.1, probability mass fitting leads to accurate estimations of these distributions. PMF discretization and decomposition should thus combine well and lead to accurate performance evaluation.

The motivation behind the decomposition technique is to decrease the computational cost for modelling large systems. To assess the complexity of the method, let us denote n_{iter} the number of decomposition loops/iterations and a the number of PMF discrete values, and suppose, to keep it simple, that b represents the average buffer size and that $m \approx m_b$ (the number of stations is similar to the number of buffers). We get that the computational cost is proportional to $2\, n_{iter}\, m\, b\, a^2$ using decomposition, as m subsystems are analyzed twice in each iteration, and each two-stations subsystem is analyzed using a state model with a number of Markov chain states proportional to $b^{2-1}\, a^2$. We thus see that the increase of the computational cost with the system size (a, b, m) is much less steep than for an exact state model of the full system with m stations ($b^{m-1}\, a^m$). In Tables 1 and 2, we give the actual computational time needed by the decomposition method using PMF on production lines . Table 1 shows the impact of the number of stations while Table 2 shows the impact of the total storage space. On both Tables, it can be seen that the computational time stays low even with many stations, or with large buffers. Compared to the state model (under parenthesis), the computational cost is much smaller for large systems. Table 5 gives the computational time reached for large assembly/disassembly

a	$B_\Sigma = 2$	$B_\Sigma = 4$	$B_\Sigma = 6$	$B_\Sigma = 8$	$B_\Sigma = 10$	$B_\Sigma = 20$	$B_\Sigma = 40$
4	0.2 *(0.1)*	0.2 *(0.1)*	0.2 *(0.1)*	0.2 *(0.3)*	0.2 *(0.4)*	0.2	0.3
6	0.2 *(0.2)*	0.3 *(0.3)*	0.3 *(0.5)*	0.3 *(1.4)*	0.3 *(2.3)*	0.4	0.6
8	0.3 *(0.3)*	0.3 *(0.6)*	0.4 *(2.9)*	0.4 *(8.1)*	0.4 *(44)*	0.7	1.2

Table 2 Computational time (in seconds) of the decomposition method, using PMF, for three station lines. The total storage space $B_\Sigma = \sum b_i$ and the number a of discretization steps vary. The computational time of the exact state model is given under parenthesis.

systems ($m = 8$), with various buffer sizes. Even if the computational times are a bit longer, and increases with the buffer sizes, they stay low (in seconds). Finally, note that, in all our experiments, the number of iterations of the decomposition method is quite small: it is around five, independently of the system size parameters (a, b, m).

6 Computational Experiments

In this Section, we perform numerical experiments to show the efficiency of the decomposition method combined with PMF. We focus on the computation of the cycle time, i.e. the time between the departures of two units from the system. It is the inverse of the throughput, which gives the number of items served in one time unit. We first analyze the accuracy of the expected cycle time estimation on small systems and then on larger assembly/disassembly systems. We also provide some comparison points with existing methods from the literature. Finally, we show that cycle time distributions can also be accurately estimated using our decomposition approach.

6.1 Results on Small Systems

To begin, we analyze a set of small systems in order to allow the comparison with the exact state model (see Section 3). The analyzed networks have three or four stations (line, disassembly and assembly topologies). The total storage space of a network goes from zero to four and is balanced among the buffers. The processing time distributions are chosen randomly from ten distributions with various shapes (uniform(0,1), beta(1.3,1), beta(2,2), beta(4,4), beta(5.5,6), beta(8,8), beta(10,9), triangular(0,1,0.5), triangular(0.2,1,0.3) and triangular(0.1,0.9,0.6)). In total, 1500 systems (50 for 5 storage space and 6 topologies) were analyzed. The number a of PMF discretization steps equals 4, 6, 8 or 20 (20 is only possible with decomposition), and $\alpha/\tau = 0.5$. The expected cycle time of the last station is estimated and then compared to simulation results. The average relative errors (computed as $|result_{simu} - result_{decompo}|/result_{simu}$) are given in Tables 3 and 4. Note that the errors presented here are the errors made by the global modelling method, on manufacturing systems with general processing time distributions. The errors have thus two components: the error brought by the distribution fitting (PMF), and the error brought

	Line		Disassembly		Assembly	
a	$m = 3$	$m = 4$	$m = 3$	$m = 4$	$m = 3$	$m = 4$
4	0.69% *(.47%)*	1.14% *(.46%)*	0.96% *(.51%)*	1.38% *(.46%)*	0.93% *(.50%)*	1.37% *(.46%)*
6	0.53% *(.20%)*	1.05% *(.21%)*	0.78% *(.22%)*	1.30% *(.21%)*	0.76% *(.22%)*	1.28% *(.21%)*
8	0.51% *(.12%)*	1.01% *(.13%)*	0.74% *(.13%)*	1.28% *(.13%)*	0.71% *(.13%)*	1.25% *(.13%)*
20	0.52%	1.02%	0.74%	1.29%	0.71%	1.27%

Table 3 Average relative error when estimating the expected cycle time, in percent, reached by the decomposition using PMF, the number of steps a and the number of stations m varying. The error reached by the state model is given under parentheses.

a	$B_\Sigma = 0$	$B_\Sigma = 1$	$B_\Sigma = 2$	$B_\Sigma = 3$	$B_\Sigma = 4$
4	2.75% *(0.48%)*	0.97% *(0.41%)*	0.55% *(0.45%)*	0.57% *(0.53%)*	0.56% *(0.53%)*
6	2.93% *(0.23%)*	0.90% *(0.19%)*	0.32% *(0.20%)*	0.23% *(0.23%)*	0.29% *(0.23%)*
8	2.93% *(0.14%)*	0.90% *(0.12%)*	0.28% *(0.12%)*	0.22% *(0.14%)*	0.24% *(0.13%)*
20	2.99%	0.93%	0.25%	0.22%	0.23%

Table 4 Average relative error when estimating the expected cycle time, in percent, reached by the decomposition using PMF, the number of steps a and the storage space changing (B_Σ stands for $\sum b(i, j)$). The error reached by the state model is given under parentheses.

by the analytical model (decomposition). It can be compared with the errors made by the state model (under parentheses), that give the errors coming only from the distribution fitting (PMF).

Table 3 shows that the proposed decomposition method leads to good accuracy. The relative error on the expected cycle time is in the order of one percent (with general processing time distributions). It can also be seen that the topology (line, disassembly, assembly) has little impact. On the opposite, the accuracy level tends to deteriorate when the number of stations increases (we show in Section 6.2 that it is not confirmed with larger systems). Table 4 shows the impact of the buffer sizes on the expected cycle time estimation. The decomposition approximation improves when the storage space increases. This is not surprising as the blocking and starvation times decrease when the storage space increases, and as the main approximation concerns them when a system is decomposed.

The impact of the number of discretization steps a is quite low. The global error is composed of the PMF distribution fitting error (revealed by the error made by the state model) and of the decomposition error, and the former is significantly smaller than the latter. From $a = 8$, and even $a = 6$ in most cases, the PMF error seems to have no impact on the global error anymore. In these cases, the relative errors shown on Tables 3 and 4 are nearly equal to the decomposition error (revealed with $a = 20$). When the decomposition is used, the distributions may thus be discretized in 6 or 8 steps.

a	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
4	1.39%	0.48%	0.65%	0.70%	0.70%
	3.1 sec.	*3.6 sec.*	*3.9 sec.*	*3.7 sec.*	*4.0 sec.*
6	1.40%	0.26%	0.30%	0.33%	0.29%
	3.5 sec.	*5.3 sec.*	*5.5 sec.*	*6.6 sec.*	*7.5 sec.*

Table 5 Average relative error when estimating the expected cycle time, in percent, reached by the decomposition using PMF on large assembly/disassembly systems ($m = 8$), the number of steps a and the storage space varying (i refers to all buffer sizes being in the interval $[2(i-1)\ 2i]$). The computational time is given in italic.

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Smallest error (min)	0.03%	0.00%	0.00%	0.00%	0.00%
10^{th} smallest error	0.17%	0.05%	0.08%	0.05%	0.02%
50^{th} error (median)	0.92%	0.19%	0.29%	0.35%	0.31%
10^{th} largest error	3.27%	0.57%	0.48%	0.57%	0.50%
Largest error (max)	6.74%	1.32%	0.86%	0.98%	0.90%

Table 6 Spread of errors when estimating the expected cycle time, among 100 experiments for each storage space (i), in percent, reached by the decomposition method using PMF on large assembly/disassembly systems ($m = 8$), with 6 steps ($a = 6$) and the storage space varying (i refers to all buffer sizes being in the interval $[2(i-1)\ 2i]$).

6.2 Larger assembly/disassembly systems

In this Section, we show how the decomposition method using PMF behaves on larger assembly/disassembly manufacturing systems. For this, we analyze systems with the topology shown in Figure 1, with eight stations and seven buffers, a disassembly station and an assembly station. We run 5 sets of 100 experiments. For each experiment, the buffers sizes are randomly chosen in the interval $[2(i-1)\ 2i]$, where $i = 1, 2, \ldots, 5$ is the index of the set of experiments (i.e. the buffer sizes are chosen among 0, 1 or 2 in the first set, 2, 3 or 4 in the second set, etc.). The distributions are chosen among the ten used in the previous experiments (see Section 6.1). For these 500 networks, we apply the decomposition method, using PMF ($\alpha/\tau = 0.5$) with $a = 4$ or 6, to estimate the expected cycle time, and compare the results to simulation results.

The accuracy level reached on the expected cycle time is given in Table 5. The Table reveals that, even for large networks, the accuracy is good (note again that these are the errors made by the global modelling method, with general processing time distributions). In fact, the accuracy does not seem to deteriorate when the system size increases (see Table 4). Table 5 also confirms that the accuracy of the estimations tends to improve when the buffer sizes increase, while it is less good with very small buffers (size 0, 1 or 2; $i = 1$). To further assess the reliability of our decomposition method, Table 6 displays the spread of the errors (with $a = 6$). Each column corresponds to one storage space configuration, i.e. 100 experiments. The errors are sorted in ascending order: the 1^{st} (minimum), 10^{th}, 50^{th}, 90^{th} and 100^{th} (maximum) are given. With

buffers of size 2 or larger (from $i = 2$), an error smaller than one percent is basically guaranteed (except for few cases when $i = 2$), while an error smaller than 0.5 percent is very likely (with a probability close to 90%). We also see that errors can reach very small values (close to zero). However, Table 6 also confirms that our method is less accurate and less reliable with very small buffers ($i = 1$), as the error may become quite large in some cases (several percents).

On all experiments, with any storage space, we get an average relative error of 0.52% (resp. 0.78%) with 6 (resp. 4) discretization steps. For buffer sizes larger than two, the relative errors is around 0.3% with six discretization steps. These experiments confirm the value of our decomposition approach: it allows to accurately evaluate the expected cycle time of large networks with general distributions in a short computational time (given in italic in Table 5).

6.3 Comparison with Available Methods

In this Section, our goal is to assess the quality of our results by comparing to results from the literature. Unfortunately, the comparison reveals to be limited, as few results from the literature are readily comparable, with a limited number of papers analyzing systems with phase-type distributions and similar assumptions (finite buffers in particular).

Altiok (1989) provides four examples of three station lines under saturation, with buffers of size three. His decomposition leads to average errors ranging from 2.31% to 6.87% (see Table 4, in Altiok's examples $B_\Sigma = 6$). Altiok (1989) also provides three other examples of unsaturated lines (Poisson arrivals and finite buffer in front of the first station). The relative errors are 1.3% (buffer sizes of 8, 5 and 3), 4.26% (buffer sizes of 5, 3 and 2), and 3.69% (5 stations, buffer sizes larger than 2). In his book, Altiok (1996) gives two other examples of unsaturated lines (Poisson arrivals and infinite buffer in front of the first station). On these examples, his decomposition method leads to relative errors of 0.35% (three stations, buffer sizes of three) and 1.02% (five stations, buffer sizes larger than four). Gun and Makowski (1987) do not give any results on the cycle time approximation. van Vuuren and Adan (2006) propose and test a decompositon method for assembly systems (without stations in series, and without disassembly stations). They apply the method on a set of experiments with 768 assembly systems, with 3, 5 or 9 stations and buffers of sizes 0, 2, 4 or 8, and report an average relative error of 1.5%. This can be compared to the 0.52% average relative error found in Section 6.2 with 6 PMF discretization steps (recall that it includes the distribution fitting error). To allow better comparison with Table 5, the average relative errors found by van Vuuren and Adan (2006) are 2.35%, 1.68%, 1.05% and 0.82%, for buffers of sizes 0, 2, 4 or 8 (distribution fitting error not included). Bierbooms et al. (2013) analyse lines with finite buffers and breakdowns, and report results on a large test set : 0.57%, 1.03%, 1.65% and 2.18%, with 4, 8, 12 and 16 stations (buffer sizes range for 1 to 50). Some other references can be cited as limiting cases, even if the system assumptions are pretty different. Brandwajn and Jow (1988) were among the first authors to propose a decomposition method. They studied lines with blocking and exponential processing time distributions. On nine

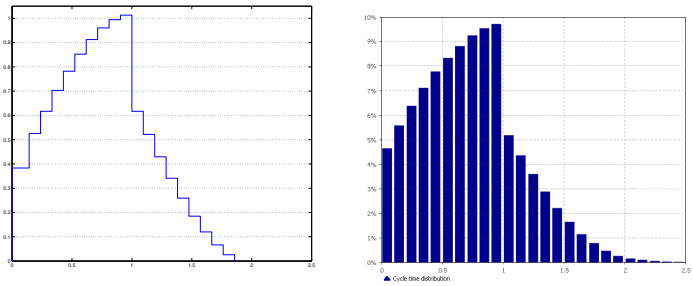

Fig. 10 Cycle time distribution of the third and last station of a production line (uniform processing time distributions, no storage space). The left-hand side gives the distribution computed by the decomposition using PMF ($\alpha/\tau = 0.5$) with $a = 10$ discretization steps. The right-hand side gives the distribution computed via simulation.

examples, their decomposition technique lead to a 0.89% average relative error, with buffer of sizes larger than 2 on all examples. The results brought by the generalized expansion method can also be compared. Andriansyah et al. (2010) analyse bufferless multi-server manufacturing systems with exponential processing time distributions. Their networks are not saturated, but are fed by an arrival process. They report an average relative error of 2.5% on about twenty examples. Manitz (2015) studies assembly/disassembly systems with general production time distributions (G/G/1/N stations), and compares the estimations of the throughput (the inverse of the cycle time) to simulation results, assuming gamma distributions. He reports estimations "within a 5% error interval" and "even better for most cases". The detailed results are displayed but the average error is not given. It appears that a good share of the errors are over 1%.

Even if this analysis is limited, our results compare favorably. It tends to show that PMF indeed brings an improvement in the application of decomposition. As shown in Section 4, PMF allows to accurately approximate the cycle time distributions, and thus the starving and blocking time, leading to a more accurate decomposition of the system.

6.4 Cycle Time Distributions

One of the advantages of the proposed modelling method comes from its ability to compute good cycle time distribution approximations (see Section 4.2). In this Section, we test whether this ability remains true when decomposition is used instead of a state model. The cycle time distributions are simply computed on the decomposed subsystems with the modified processing time distributions. The subsystems are analyzed by a state model and the computation explained in Section 4.1 can thus be applied.

To begin with, we analyze a three station line with uniform processing time distributions and no storage space, as in Figure 6. Figure 10 shows the cycle time distribution estimation for the last station, using decomposition (left-hand side). It can be compared to the result of a simulation (right-hand side). We see that the cycle time distribution estimation is still valuable when using

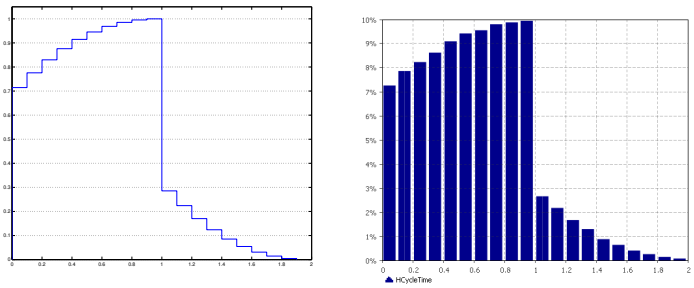

Fig. 11 Cycle time distribution of the last station of a large assembly/disassembly system (uniform distributions, buffers of size 2), computed using decomposition (left) and computed via simulation (right).

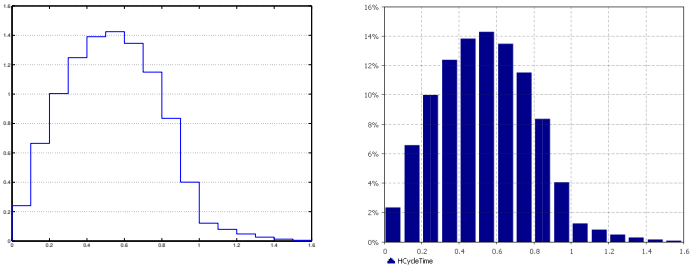

Fig. 12 Cycle time distribution of the last station of a large assembly/disassembly system (beta(2,2) distributions, buffers of size 4), computed using decomposition (left) and computed via simulation (right).

decomposition. However, it is not as good as the result obtained with a state model, shown in Figure 6. For example, the values just before one are: 0.972 with simulation, 0.972 with the exact PMF state model, 1.013 with decomposition. The values just after one are: 0.519 with simulation, 0.512 with the exact model, 0.617 with decomposition. The less accurate approximation is not surprising, as we already observed that the decomposition is less accurate, and particularly with buffers of size zero.

We now illustrate the accuracy of the cycle time distribution estimation on larger systems, with larger buffers. We look at two instances of the assembly/disassembly system used in the previous experiments set, and shown in Figure 1. In the first case (Figure 11), the buffer sizes equal two and the processing times are uniformly distributed, while in the second case (Figure 12), the buffer sizes equal four and the processing times distributions are beta(2,2). For both Figures, the left-hand side shows the cycle time distribution estimation using decomposition (and PMF with $a = 10$ and $\alpha/\tau = 0.5$), while the right-hand side gives the result of a simulation. We see that the cycle time distribution estimation is still a reliable approximation, in its shape and values, when using decomposition. For example, in Figure 11, the values just after one are: 0.268 with simulation, and 0.286 with decomposition. In Figure 12, the maximum values of the distributions are: 1.431 with simulation, and 1.425 with decomposition.

The ability to compute a reliable approximation of the cycle time distribution (except with very small buffers) is a valuable feature of probability mass fitting, compared to other concurrent

methods (moments fitting in particular). It is true when a state model is coupled to PMF and also mostly true when the decomposition technique is applied. The distribution provides more information on the behavior of the system, compared to the isolated expectation. Percentiles, for example, help the manager to evaluate the risk and the service level of the system.

7 Conclusion

In this paper, we propose an efficient methodology to evaluate the performance of stochastic manufacturing systems with general characteristics: assembly/disassembly layout, general processing time distributions, finite storage spaces, and a potentially large number of stations. To model these general manufacturing systems, we first apply probability mass fitting (PMF) to fit general distributions to discrete phase-type distributions. Then, we apply decomposition: the system is decomposed into two station subsystems and the processing time distributions of the virtual stations are iteratively modified to approximate the impact of the rest of the network, adding estimations of the blocking and starving distributions. The ability of PMF to reliably approximate the cycle time distributions (and thus the blocking and starvation time distributions) is an important advantage of PMF for the application of the decomposition method.

Our computational experiments show that the decomposition method coupled to PMF provides accurate estimations. The accuracy improves when the storage space increases (it is sensibly less good with zero buffers). The global error is made of the PMF error and the decomposition error. Both errors are shown to be limited, and from 8 discretization steps the PMF error is negligible. On a set of large assembly/disassembly manufacturing systems, the average global error equals 0.3% with buffer sizes larger than two, and 1.4% with buffer sizes smaller than two ($a = 6$). Comparing with results available in the literature, it seems that probability mass fitting has a positive impact in the application of the decomposition technique, due to its ability to accurately approximate cycle time distributions. Furthermore, we showed that using PMF the decomposition approach allows to compute reliable estimations of the cycle time distributions (except with very small buffers). This is a valuable feature as it offers more detailed information than the usual expected performance measures.

The main contribution of this paper is in matching-up the PMF discretization with the decomposition method to evaluate the performance of large general assembly/disassembly manufacturing systems, and in showing that these two approaches indeed combine well. Future research could extend these results to other configurations such as split-and-merge networks, closed manufacturing systems, or multiple-server networks. It could also aim at coupling the performance evaluation with optimization methods (for buffer sizing for example). Finally, the distributions approximations could be further explored, beyond the cycle time distribution, in particular looking at the distribution of the time spent by a part in the entire production system.

References

Adan I, van Eenige M, Resing J (1994) Fitting discrete distributions on the first two moments. Tech. rep., Department of Mathematics and Computer Science, Eindhoven University of Technology

Altiok T (1985) On the phase-type approximations of general distributions. IIE Transactions 17(2):110–116

Altiok T (1989) Approximate analysis of queues in series with phase-type service times and blocking. Operations Research 37(4):601–610

Altiok T (1996) Performance Analysis of Manufacturing Systems. Springer, New York

Andriansyah R, van Woensel T, Cruz FRB, Duczmal L (2010) Performance optimization of open zero-buffer multi-server queueing networks. Computers & Operations Research 37(8):1472–1487

Asmussen S, Nerman A, Olsson M (1996) Fitting phase-type distributions via the em algorithm. Scandinavian Journal of Statistics 23:419–441

Assaf R, M MC, Matta A (2014) Analytical evaluation of the output variability in production systems with general markovian structure. OR Spectrum 36(3):799–835

Balsamo S, de Nitto Personé V, Onvural R (2001) Analysis of Queueing Networks with Blocking. Kluwer Academic Publishers, Dordrecht, The Netherlands

Bierbooms R, Adan IJBF, van Vuuren M (2013) Approximate performance analysis of production lines with continuous material flows and finite buffers. Stochastic Models 29(1):1–30

Bitran GR, Dasu S (1992) A review of open queueing network models of manufacturing systems. Queueing Systems 12(1-2):95–134

Bobbio A, Cumani A (1992) Ml estimation of the parameters of a ph distribution in triangular canonical form. In: Balbo G, Serazzi G (eds) Computer Performance Evaluation, Elsevier Science Publishers, pp 33–46

Bobbio A, Telek M (1994) A benchmark for ph estimation algorithms: results for acyclic-ph. Stochastic Models 10:661–677

Bobbio A, Horváth A, Scarpa M, Telek M (2003) Acyclic discrete phase type distributions : Properties and a parameter estimation algorithm. Performance Evaluation 54:1–32

Bobbio A, Horváth A, Telek M (2004) The scale factor: a new degree of freedom in phase-type approximation. Performance Evaluation 56(1-4):121–144

Bobbio A, Horváth A, Telek M (2005) Matching three moments with minimal acyclic phase type distributions. Stochastic Models 21:303–326

Botta RF, Harris CM (1986) Approximation with generalized hyperexponential distributions: weak convergence results. Queueing Systems 1(2):169–190

Brandwajn A, Jow Y (1988) An approximation method for tandem queues with blocking. Operations Research 36(1):73–83

Buzacott J, Shanthikumar J (1993) Stochastic Models of Manufacturing Systems. Prentice-Hall, Englewood Cliffs, New Jersey

Colledani M (2013) Performance evaluation of two-stage buffered production systems with discrete general markovian machines. In: Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control International Federation of Automatic Control, pp 1638–1643

Colledani M, Gershwin S (2013) A decomposition method for approximate evaluation of continuous flow multi-stage lines with general markovian machines. Annals of Operations Research 209:5–40

Colledani M, Tolio T (2011) Integrated analysis of quality and production logistics performance in manufacturing lines. International Journal of Production Research 49(2):485–518

Dallery Y, Frein Y (1993) On decomposition methods for tandem queueing networks with blocking. Operations Research 41(2):386–399

Dallery Y, Gershwin S (1992) Manufacturing flow line systems: a review of models and analytical results. Queueing Systems 12:3–94

Dallery Y, Liu Z, Towsley D (1994) Equivalence, reversibility, symmetry and concavity properties in fork/join queueing networks with blocking. Journal of the Association for Computing Machinery 41:903–942

de Koster M (1987) Estimation of line efficiency by aggregation. International Journal of Production Research 25(4):615–625

Gershwin S (1987) An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking. Operations Research 35(2):291–305

Gourgand M, Grangeon N, Norre S (2005) Markovian analysis for performance evaluation and scheduling in m machine stochastic flow-shop with buffers of any capacity. European Journal of Operational Research 161:126–147

Govil M, Fu M (1999) Queueing theory in manufacturing : A survey. Journal of manufacturing systems 18(4):214–240

Gun L, Makowski A (1987) An approximation method for general tandem queueing systems subject to blocking. SRC Technical Report, Electrical Engineering Departement and Systems Research Center, University of Maryland

Helber S (1998) Decomposition of unreliable assembly/disassembly networks with limited buffer capacity and random processing times. European Journal of Operational Research 109:24–42

Helber S (1999) Performance analysis of flow lines with non-linear flow of material. In: Lecture Notes in Economics and Mathematical Systems, vol 473, Berlin, Heidelberg, New York: Springer

Helber S (2006) Analysis of flow lines with cox-2-distributed processing times and limited buffer capacity. In: Liberopoulos G, Papadopoulos C, Tan B, Smith J, Gershwin S (eds) Stochastic Modeling of Manufacturing Systems, Springer, pp 55–76

Hillier F, Boling R (1967) Finite queues in series with exponential or erlang service times–a numerical approach. Operations Research 15(2):286–303

Hopp W, Spearman M (1996) Factory Physics : Foundations of Manufacturing Management. Irwin, Burr Ridge, Illinois

Huisman T, Boucherie RJ (2011) Decomposition and aggregation in queueing networks. In: Richard J Boucherie NMv (ed) Queueing Networks: A Fundamental Approach, Springer US

Jeong KC, Kim YD (1998) Performance analysis of assembly/disassembly systems with unreliable machines and random processing times. IIE Transactions 30(1):41–53

Johnson M, Taaffe M (1989) Matching moments to phase distributions: Mixtures of erlang distributions of common order. Stochastic Models 5(4):711–743

Kerbache L, Smith JM (1987) The generalized expansion method for open finite queueing networks. European Journal of Operational Research 32:448–461

Kerbache L, Smith JM (1988) Asymptotic behavior of the expansion method for open finite queueing networks. Computers & Operations Research 15(2):157–169

Kim S (2011) Modeling cross correlation in three-moment four-parameter decomposition approximation of queueing networks. Operations Research 59(2):480–497

Krieg G, Kuhn H (2002) A decomposition method for multi-product kanban systems with setup times. IIE Transactions 34(7):613–625

Kuhn H (2003) Analysis of automated flow line systems with repair crew interference. In: Gershwin S, Dallery Y, Papadopoulos C, Smith JM (eds) Analysis and Modeling of Manufacturing Systems, Kluwer, pp 155–179

Lagershausen S, Tan B (2015) On the exact inter-departure, inter-start, and cycle time distribution of closed queueing networks subject to blocking. IIE Transactions 47(7):673–692

Lang A, Arthur J (1997) Parameter approximation for phase-type distributions. In: Chakravarthy S, Alfa A (eds) Matrix Analytical Methods in Stochastic Models, Marcel Dekker, New York, pp 151–206

Leemans H (2001) Waiting time distribution in a two-class two-server heterogeneous priority queue. Performance Evaluation 43(2-3):133–150

Levantesi R, Matta A, Tolio T (2003) Performance evaluation of continuous production lines with machines having different processing times and multiple failure modes. Performance Evaluation 51:247–268

Li J, Blumenfeld DE, Huang N, Alden JM (2009) Throughput analysis of production systems: recent advances and future topics. International Journal of Production Research 47(14):3823–3851

Liu J, Yang S, Wu A, Hu S (2012) Multi-state throughput analysis of a two-stage manufacturing system with parallel unreliable machines and a finite buffer. European Journal of Operational Research 219:296–304

Manitz M (2008) Queueing-model based analysis of assembly lines with finite buffers and general service times. Computers & Operations Research 35(8):2520–2536

Manitz M (2015) Analysis of assembly/disassembly queueing networks with blocking after service and general service times. Annals of Operations Research 226(1):417441

Marie R (1980) Calculating equilibrium probabilities for λ(n)/ck/1/n queues. In: Performance'80: Proceedings of the 1980 international symposium on Computer performance modelling, measurement and evaluation, ACM, New York, NY, USA, pp 117–125

Neuts M (1981) Matrix-Geometric Solutions in Stochastic Models: an Algorithmic Approach. Dover Publications, New York

Osogami T, Harchol-Balter M (2006) Closed form solutions for mapping general distributions to quasi-minimal ph distributions. Performance Evaluation 63(6):524–552

Papadopoulos H, Heavey C (1996) Queueing theory in manufacturing systems analysis and design : A classification of models for production and transfer lines. European Journal of Operational Research 92:1–27

Pearson E, Johnson N, Burr I (1979) Comparisons of the percentage points of distributions with the same first four moments, chosen from eight different systems of frequency curves. Communications in Statistics - Simulation and Computation 8(3):191–229

Sauer C, Chandy K (1975) Approximate analysis of central server models. IBM Journal of Research and Development 19(3):301–313

Shi C, Gershwin SB (2016) Part sojourn time distribution in a two-machine line. European Journal of Operational Research 248(1):146–158

Tan B (2003) State-space modeling and analysis of pull-controlled production systems. In: Gershwin S, Dallery Y, Papadopoulos C, Smith JM (eds) Analysis and Modeling of Manufacturing Systems, Kluwer, pp 363–398

Tan B, Gershwin S (2009) Analysis of a general markovian two-stage continuous-flow production system with a finite buffer. International Journal of Production Economics 120:327–339

Tancrez JS (2009) Modelling queueing networks with blocking using probability mass fitting. PhD thesis, Catholic University of Louvain

Tancrez JS, Chevalier P, Semal P (2011) Probability masses fitting in the analysis of manufacturing flow lines. Annals of Operations Research 182(1):163–191

Telek M (2000) Minimal coefficient of variation of discrete phase type distributions. In: 3rd International Conference on Matrix-Analytic Methods in Stochastic models, MAM3, Notable Publications Inc., Leuven, Belgium, pp 391–400

Tempelmeier H, Burger M (2001) Performance evaluation of unbalanced flow lines with general distributed processing times, failures and imperfect production. IIE Transactions 33:293–302

Terracol C, David R (1987) An aggregation method for performance evaluation of transfer lines with unreliable machines and finite buffers. In: IEEE Conference on Robotics and Automation, Raleigh, NC

van Vuuren M, Adan I (2006) Performance analysis of assembly systems. In: A N Langville WJS (ed) MAM 2006: Markov Anniversary Meeting, Boson Books, Raleigh, North Carolina, USA, pp 89–100

Whitt W (1982) Approximating a point process by a renewal process, i: Two basic methods. Operations Research 30(1):125–147