2008/28

■

Probability masses fitting in the analysis of manufacturing flow lines

Jean-Sébastien Tancrez, Philippe Chevalier
and Pierre Semal

CORE
Voie du Roman Pays 34
B-1348 Louvain-la-Neuve, Belgium.
Tel (32 10) 47 43 04
Fax (32 10) 47 43 01
E-mail: corestat-library@uclouvain.be
http://www.uclouvain.be/en-44508.html

CORE DISCUSSION PAPER
2008/28

Probability masses fitting in the analysis of manufacturing flow lines

Jean-Sébastien TANCREZ[1], Philippe CHEVALIER[2]
and Pierre SEMAL [3]

May 2008

Abstract

A new alternative in the analysis of manufacturing systems with finite buffers is presented. We propose and study a new approach in order to build tractable phase-type distributions, which are required by state-of-the-art analytical models. Called "probability masses fitting" (PMF), the approach is quite simple: the probability masses on regular intervals are computed and aggregated on a single value in the corresponding interval, leading to a discrete distribution. PMF shows some interesting properties: it is bounding, monotonic and it conserves the shape of the distribution. After PMF, from the discrete phase-type distributions, state-of-the-art analytical models can be applied. Here, we choose the exactly model the evolution of the system by a Markov chain, and we focus on flow lines. The properties of the global modelling method can be discovered by extending the PMF properties, mainly leading to bounds on the throughput. Finally, the method is shown, by numerical experiments, to compute accurate estimations of the throughput and of various performance measures, reaching accuracy levels of a few tenths of percent.

Keywords: stochastic modelling, flow lines, probability masses fitting, discretization, bounds, performance measures, distributions.

[1] CORE, Université catholique de Louvain, Belgium and LSM, Facultés Universitaires Catholiques de Mons, Belgium. E-mail: js.tancrez@uclouvain.be
[2] CORE and Louvain School of Management, Université catholique de Louvain, Belgium. E-mail: philippe.chevalier@uclouvain.be
[3] Louvain School of Management, Université catholique de Louvain, Belgium. E-mail: Pierre.semal@uclouvain.be

This paper presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

1 Introduction

In the present economic situation, companies are submitted to an ever more competitive environment. Investments have to bring ever higher profit margin. In view of this, inventory has become a crucial question in production, due to the investment it requires and ties up. It is essential for companies to get high productivity, while limiting the investments used by stocks. In this context, the analysis of manufacturing systems with finite storage capacity is of practical interest. Models of such systems needs to be realistic, accurate and complete in order to allow to design and manage them efficiently, and, in particular, to find the best balance between productivity benefits and inventory costs.

In the last decades, various analytical models for manufacturing systems with finite buffers have thus been proposed. The only exact models of practical interest are called state models. They build a continuous (rarely discrete) Markov chain that models the evolution of the system and from which the performance measures can be inferred (see [11] for example). However, these models' applicability is limited to small instances as the state space size of the Markov chain increases quickly with the system size. As a result, approximate models have been proposed, based on the idea of decomposing the system. First, the decomposition method breaks the system into smaller subsystems, analyze them in isolation and then adds back the interdependencies between the subsystems iteratively (see [6] for a good review). Second, the generalized expansion method also decomposes the system, but it adds the concept of an artificial node which registers the blocked jobs (see [12]). For further details, several comprehensive reviews of analytical models for manufacturing systems are available, such as [5, 9, 14, 1, 7, 13, 10].

An important assumption of these analytical models (exact and approximate) is that they suppose phase–type distributed processing times, so that the theory of Markov processes can be applied. No analytical model is directly available for generally distributed processing times. Consequently, in practice, in order to analyze real systems with general distributions, one has to build tractable phase–type distributions previously to the application of the analytical models. This preliminary step is part of the modelling pro-

The work of J.–S. Tancrez has been funded by the European Social Fund and the Walloon Region of Belgium. It is made in collaboration with ArcelorMittal Fontaine.
This text presents research results of the Belgian Program on Interuniversity Poles of Attraction initiated by the Belgian State, Prime Minister's Office, Science Policy Programming. The scientific responsibility is assumed by the authors.

cess, which should be seen as a whole. The most classical method to build tractable distributions is moments fitting: the two (or three) first moments are computed from the original distributions and the phase–type distributions are then built in order to get the same first moments (see [17] for example). Other statistical techniques can be used, based on the maximum likelihood principle for example (see [4]).

Our originality lies in this step of the modelling process. In this paper, we present a new alternative to build tractable distributions, called "probability masses fitting" (PMF). PMF is quite intuitive: the probability masses on regular intervals are computed and aggregated on a single value in the corresponding interval, leading to a discrete (phase–type) distribution. PMF is particularly natural when the data about the processing time distributions is collected in the form of histograms, the most common form in practice. Note that the method is essentially designed with continuous original distributions in mind (even if discrete original distributions could be considered). After PMF, from the discrete distributions, a state model is applied, modelling the evolution of the system by a Markov chain, from which the performance measures can be computed. The goal of this paper is to present probability masses fitting and to study its effects on the global modelling process and on its results. PMF is presented in a general fashion: the particular value in the interval where the probability mass is aggregated is a parameter. In a previous communication [16], we introduced a particular PMF, called "grouping at the end", which aggregates the probability masses at the end of the intervals, and studied its effect on the productivity estimation. The present paper introduces the general PMF and its properties, and presents the estimation of various performance measures.

The rest of the paper is organized as follows. Probability masses fitting is presented in details in section 2. We show various properties of PMF, such as bounding and monotonicity properties. We present two slightly different alternatives (generating instant jobs or not) and discuss PMF's characteristics. In section 3, the global modelling method (PMF followed by a state model) is presented for manufacturing flow lines. Then, our first point is to translate the PMF properties to the productivity of the modeled system. This is done in section 4. In section 5, we show, by computational results, how our methods behaves. We show that our method leads to accurate estimations of the performance measures and we study how the results change when the system configuration changes. Finally, we conclude in section 6.

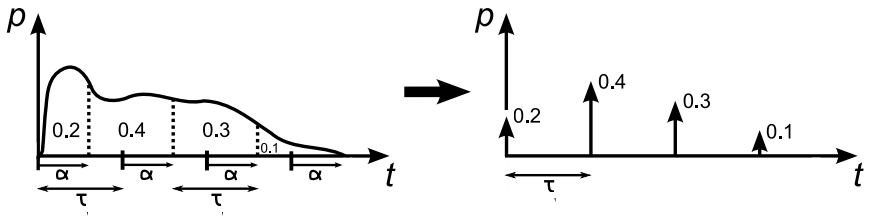

Figure 1: Discretization by probability masses fitting, with $a = 3$.

2 Probability Masses Fitting

In this section, we present the proposed method to build tractable discrete phase–type distributions, called "probability masses fitting", and denoted PMF. A general definition is first given, followed by some interesting properties. Then, an alternative PMF is presented. Finally, we discuss the advantages and weaknesses of PMF.

2.1 Definition

Let us suppose we have a positive finite processing time distribution and want to transform it into a discrete phase–type distribution by probability masses fitting. To begin, we choose the number of non–zero discrete values, denoted a. Moreover, as we want to infer a discrete Markov chain from the obtained distribution, the size of the interval between two consecutive discrete values, denoted τ, has to be constant. Knowing this, the idea behind probability masses fitting is quite simple: the probability of each discrete value equals the original probability mass around this discrete value. The original distribution is transformed into a discrete one by aggregating the original probability mass distributed in an interval around each discrete value. In order to fix the shift of the intervals, we introduce the parameter α, with $0 \leq \alpha \leq \tau$, which gives the size of the part of the interval located after the discrete value. In summary, 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 each $j = 1, 2, \ldots, a$, and the mass in $[0, \alpha]$ on 0. PMF is illustrated on figure 1. Note that τ gives the interval between two discrete values as well as the interval on which the probability mass is computed.

In practice, the parameters a, i.e. the number of non–zero discrete values, and the factor α/τ, i.e. the proportion of the interval which is located

after the discrete value, are first chosen. The space τ between two discrete values, or, equivalently, the probability mass interval size, is then deduced from these parameters so that the finite width of the original distribution is covered. In other words, we want the maximum original time max to be equal to $a\tau + \alpha$, so that τ is chosen as follows:

$$\tau = \frac{max - \alpha}{a}. \tag{1}$$

Before showing some properties of PMF, let us introduce some notations. The random variable representing the processing time of $w_{i,k}$, the job k on station i, is denoted $l(w_{i,k})$. The realization of $l(w_{i,k})$ in a sample production run r_P (producing P units), i.e. the time the job $w_{i,k}$ takes in this particular run, is denoted $l^r(w_{i,k})$. After PMF with parameter α, we get the corresponding discrete processing times $l_\alpha(w_{i,k})$ and the realizations $l^r_\alpha(w_{i,k})$. Note that, with these notations, the PMF discretization can be formulated as follows:

$$l^r_\alpha(w_{i,k}) \triangleq \left\lceil \frac{l^r(w_{i,k}) - \alpha}{\tau} \right\rceil \tau, \qquad \forall\, r_P, i, k.$$

2.2 Properties

One of the main advantages of probability masses fitting lies in the strong control it offers on the transformation of the individual processing times. The transformation of each original processing time to a discrete processing time is exactly known. Notably, we know that, when discretized, a processing time will not be shifted by more than α to the left neither by more than $\tau - \alpha$ to the right (see figure 1). It leads to the following **bounds** on the processing time.

Proposition 1. *The original processing time of a job, $l^r(w_{i,k})$, can be bounded using its discretized value, $l^r_\alpha(w_{i,k})$.*

$$l^r_\alpha(w_{i,k}) - (\tau - \alpha) \leq l^r(w_{i,k}) \leq l^r_\alpha(w_{i,k}) + \alpha, \qquad \forall\, r_P, i, k. \tag{2}$$

Proof. There are two limit cases. The lower bound corresponds to a processing time of length 0 which is increased to τ, with $j = 1, \ldots, a$. The upper bound corresponds to a processing time of length $j\tau + \alpha$ which is decreased to $j\tau$, with $j = 0, \ldots, a$. □

The main interest of these bounds on processing times comes from their extension to bounds on the productivity, which will be shown in section 4.1.

The non–conservation of the distribution expectation, in general, is another characteristic of probability masses fitting. The error made on the expectation is a function of the characteristics of the original distribution and of the parameter α, i.e. the shift of the interval on which the probability mass is computed. Consequently, it is quite interesting to study the evolution of the discretized processing time according to the parameter α. It can be checked that this evolution is a **monotonic**, more precisely decreasing, function. Moreover we show that a particular α can always be found so that the expectation of the discrete distribution equals the expectation of the original distribution, if the latter is continuous. In other words, the conservation of the expectation can be obtained by choosing the right parameter α.

Proposition 2. *When discretizing by PMF with two parameters α such that $\alpha_1 \leq \alpha_2$, and a constant, the first discretized processing time will always be larger than the second one.*

$$l^r_{\alpha_1}(w_{i,k}) \geq l^r_{\alpha_2}(w_{i,k}), \qquad \forall\, r_P, i, k.$$

The expectation of a distribution discretized by probability masses fitting is thus decreasing when α is increasing. Moreover, if the original cumulative distribution function $F(t)$ is continuous, the evolution in α of the expectation of the discretized distribution is also continuous. Consequently, a parameter α can always be computed so that the expectation is conserved, i.e. so that the expectation of the discretized distribution is the same as the original one.

Proof. To begin, note that $j\tau_1 + \alpha_1 \leq j\tau_2 + \alpha_2$, $\forall\, j$, as, by equation (1), $j\tau + \alpha = j((max - \alpha)/a) + \alpha = (j\, max + \alpha(a - j))/a$, and as $j \leq a$. In order to prove the first part of the proposition, two cases have thus to be considered. In the first case, the original processing time lies between $(j-1)\tau_2 + \alpha_2$ and $j\tau_1 + \alpha_1$. We thus get $l^r_{\alpha_2}(w_{i,k}) = j\tau_2$ and $l^r_{\alpha_1}(w_{i,k}) = j\tau_1$ and, as $\tau_1 \geq \tau_2$ ($\tau = (max - \alpha)/a$), the proposition inequality is valid in this case. In the second case, the original processing time lies between $j\tau_1 + \alpha_1$ and $j\tau_2 + \alpha_2$. We thus get $l^r_{\alpha_2}(w_{i,k}) = j\tau_2$ and $l^r_{\alpha_1}(w_{i,k}) = (j+1)\tau_1$ and the proposition inequality is also valid in this second case.

To prove the continuity in α of $E[t_\alpha]$, the expectation of a discretized distribution, we can simply formulate $E[t_\alpha]$ as a sum of continuous functions. If $f(t)$ and $F(t)$ are the probability density function and the cumulative distribution function of the original

distribution, $E[t_\alpha]$ can be written as follows:

$$\begin{aligned} E[t_\alpha] &= \sum_{j=0}^{a} j\tau \int_{(j-1)\tau+\alpha}^{j\tau+\alpha} f(t)\,dt = \sum_{j=1}^{a} j\tau \left(F(j\tau+\alpha) - F((j-1)\tau+\alpha)\right) \\ &= \tau\big(F(\tau+\alpha) - F(\alpha)\big) + 2\tau\big(F(2\tau+\alpha) - F(\tau+\alpha)\big) + \ldots \\ &\qquad \ldots + a\tau\big(F(a\tau+\alpha) - F((a-1)\tau+\alpha)\big) \\ &= -\tau F(\alpha) - \tau F(\tau+\alpha) - \ldots - \tau F((a-1)\tau+\alpha) + a\tau F(max) \\ &= a\tau - \tau \sum_{j=0}^{a-1} F(j\tau+\alpha). \end{aligned}$$

This proves the second statement of the proposition. Finally, in order to prove that an α leading to expectation conservation can be found, we just have to show that this expectation is larger than the expectation of the original distribution when α equals zero and smaller when α equals τ. As it is continuous, the expectation of the discretized distribution will thus equal the expectation of the original distribution for a given α. From equation (2), it can be seen that, when α equals zero, $l^r(w_{i,k}) \leq l_0^r(w_{i,k})$, in other words every processing time is lengthened by the discretization. The expectation of the discretized distribution is thus larger than the original one. Similarly, from (2), when α equals τ, $l_\tau^r(w_{i,k}) \leq l^r(w_{i,k})$, and the expectation of the discretized distribution is thus smaller than the original one. This ends the proof. □

Note that the distribution function $F(t)$ is indeed continuous if the processing times distributions are given in the form of histograms.

2.3 Probability Masses Fitting Without Instant Job

In the previous section, we introduced the basic probability masses fitting. This discretization potentially generates discrete processing times of length zero with non–zero probability of occurrence (see figure 1). In discretized time, jobs with length zero can thus occur. We call them "instant jobs". It is not difficult to guess that instant jobs complicate the modelling of the system (see section 3). It leads us to propose an alternative to the basic PMF, an alternative which does not generate jobs with length zero. We call it "probability masses fitting without instant job", denoted PMF/nIJ. Moreover, in order to avoid confusion, we will, from now on, call the basic PMF presented previously "probability masses fitting with instant job", and note it PMF/IJ. PMF will be used for the general concept, without differentiation between both alternatives.

In this section, we briefly present PMF/nIJ and its properties. Probability masses fitting without instant job is very similar to its sister with instant jobs. It is just modified in order to remove the possibility of instant jobs. The small difference thus only concerns the first interval (from 0 to α). The original probability mass on this interval $[0, \alpha]$ is aggregated on τ

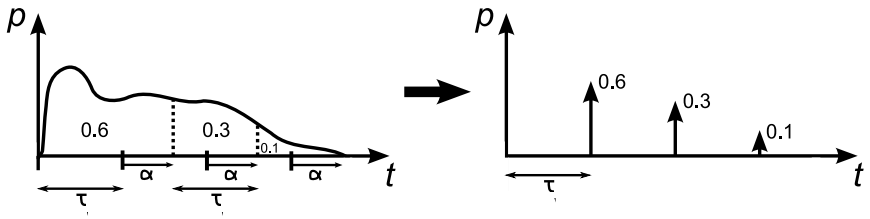

Figure 2: Discretization by probability masses fitting without instant job, with $a = 3$.

instead of 0. In summary, PMF/nIJ 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 $j = 1, 2, \ldots, a$, and the mass in $[0, \alpha]$ on τ. PMF/nIJ is illustrated on figure 2. Compared to figure 1, the differences lie in the width of the interval on which the first probability mass is computed and in the absence of probability in zero. Using PMF/IJ or PMF/nIJ, the discretized distributions will have the same values for all but the first interval. Note that PMF/IJ and PMF/nIJ are equivalent when $\alpha = 0$.

The parameters a, α and τ are defined exactly the same way as for PMF/IJ. The space τ between two discrete values is deduced from a and α/τ using (1). The processing times discretized by PMF/nIJ are denoted $l_{\alpha'}(w_{i,k})$ and the realizations $l^r_{\alpha'}(w_{i,k})$. With these notations, PMF/nIJ can be formulated as follows:

$$\begin{array}{rcll} l^r_{\alpha'}(w_{i,k}) & \triangleq & \tau, & \text{if } l^r(w_{i,k}) \leq \tau + \alpha, \quad \forall\, r_P, i, k, \\ & \triangleq & \left\lceil \frac{l^r(w_{i,k}) - \alpha}{\tau} \right\rceil \tau, & \text{if } l^r(w_{i,k}) > \tau + \alpha, \quad \forall\, r_P, i, k. \end{array}$$

The properties of probability masses fitting without instant job are similar to the properties of PMF with instant jobs (see section 2.2). We give them shortly. As the first interval is larger using PMF/nIJ (see figure 2), the **bounding** property is less good than the one of PMF/IJ. The lower bound is less tight.

Proposition 3. *The original processing time of a job, $l^r(w_{i,k})$, can be bounded using its PMF/nIJ discretized value, $l^r_{\alpha'}(w_{i,k})$.*

$$l^r_{\alpha'}(w_{i,k}) - \tau \leq l^r(w_{i,k}) \leq l^r_{\alpha'}(w_{i,k}) + \alpha, \qquad \forall\, r_P, i, k. \tag{3}$$

Proof. The lower bound corresponds to a processing time of length 0 which is increased to τ. The upper bound corresponds to a processing time of length $j\tau + \alpha$ which is decreased to $j\tau$, $\forall j = 1, \ldots, a$. □

Identically to PMF/IJ, it can be shown that the processing time discretized by PMF/nIJ is a **monotonic**, decreasing, function of α. However, in contrast to PMF/IJ, an α cannot always be found in order to fit the original expectation. The continuity can be proved under the same assumption but an α cannot always be found so that the discretized expectation is smaller than the original one (it happens if the expectation of the original distribution is smaller than τ).

Proposition 4. *Discretizing by PMF/nIJ, with two parameters α such that $\alpha_1 \leq \alpha_2$, and a constant, the first discretized processing time will always be larger than the second one.*

$$l^r_{\alpha'_1}(w_{i,k}) \geq l^r_{\alpha'_2}(w_{i,k}), \qquad \forall\, r_P, i, k.$$

Proof. As for Proposition 2, $j\tau_1 + \alpha_1 \leq j\tau_2 + \alpha_2$, $\forall\, j$, as $j\tau + \alpha = j((max - \alpha)/a) + \alpha = (j\, max + \alpha(a - j))/a$, and as $j \leq a$. In order to prove the proposition, three cases have to be considered. In the first case, the original processing time lies between 0 and α_2. We thus get $l^r_{\alpha'_2}(w_{i,k}) = \tau_2$ and $l^r_{\alpha'_1}(w_{i,k}) = \tau_1$ (since $\tau_1 + \alpha_1 > \alpha_2$) and, as $\tau_1 \geq \tau_2$ ($\tau = (max - \alpha)/a$), the inequality is satisfied. In the second case, the original processing time lies between $(j-1)\tau_2+\alpha_2$ and $j\tau_1+\alpha_1$ (where $j = 1 \ldots a$). We thus get $l^r_{\alpha'_2}(w_{i,k}) = j\tau_2$ and $l^r_{\alpha'_1}(w_{i,k}) = j\tau_1$ and, as $\tau_1 \geq \tau_2$, the proposition inequality is valid in this case. In the third case, the original processing time lies between $j\tau_1 + \alpha_1$ and $j\tau_2 + \alpha_2$. We thus get $l^r_{\alpha'_2}(w_{i,k}) = j\tau_2$ and $l^r_{\alpha'_1}(w_{i,k}) = (j + 1)\tau_1$ and this ends the proof. □

2.4 Advantages and weaknesses

We end the presentation of probability masses fitting by a short discussion presenting its main advantages and weaknesses (especially compared to moments fitting) and the differences between PMF with or without instant job. The main **advantages** of probability masses fitting are the following.

- The idea of PMF is *simple* and intuitive. It is natural and easy to understand.
- It preserves the *shape* of the distribution. In fact, it could have been called "shape fitting". Moreover, we show further on that this stays true for the computed throughput distribution.
- PMF is particularly natural when the data about the processing time distributions is collected in the form of *histograms*, the most common form in practice.
- It is *intelligible*: it offers a strong control on the transformation of the individual processing times. The transformation of each original processing time to a discrete processing time is exactly known.

- As a result of the previous advantage, PMF allows to get *bounds* on the productivity.
- It is *refinable*: the accuracy improves when the number a of discrete values increases. Moreover, at the limit, when the step size decreases, PMF leads to the exact original distribution.
- As will be shown later on, PMF leads to *accurate* estimations of the performance measures of the manufacturing system.

However, probability masses fitting has, of course, some **weaknesses**.

- PMF does not, in general, conserve the *moments* of the distribution. However, we showed that, using PMF with instant jobs, the parameter α can be chosen in order to conserve the expectation, if the original cumulative distribution function is continuous (see section 2.2).
- For the PMF to be applied as defined here, the distributions have to have a *finite* domain. Note that this is always the case in practice.
- PMF is badly suited for processing time distributions including *rare events*. Indeed, rare events extend the width of the distribution, leading to many zero discrete values in the discrete distribution, and increase the complexity without improving the accuracy.

Finally, to compare both PMF alternatives, with or without job, we can say that instant jobs have a positive contribution on the accuracy of the distribution fitting, but at the expense of a more complex modelling. This trade–off will be studied in more details by computational experiments (see section 5). Instant jobs also lead to better bounds, in particular lower bounds. Moreover, PMF/IJ has the advantage that it allows, by choosing the adequate shift parameter α, to preserve the expectation of the original distribution.

3 Method

In the introduction, we described the modelling process of a manufacturing system as being composed of two steps: a tractable distributions building preempting the analytical modelling. Our originality lies in the first step, where we use probability masses fitting. PMF has now been well defined in the previous section. We are in a position to present the complete modelling method. In very few words, from the discrete processing time

distributions built by probability masses fitting, the evolution of the manufacturing system is described by a Markov chain, using a state model, and the performance measures of the system can then be estimated from the chain.

We present the modelling method for **flow lines**. However, note that the approach could be applied to more complex manufacturing systems (assembly/disassembly systems in particular). We are interested in flow lines with asynchronous part transfer and stochastic processing times. They are made of a series of m stations separated by $m-1$ finite buffers. The items are processed sequentially by the stations and stored in the buffers between them when necessary. Such lines experience productivity losses due to blocking and starving. First, a station is said to be blocked when it cannot get rid of an item because the next buffer is full. Second, a station is said to be starved when it cannot begin to work on a new item because the previous buffer is empty. Increasing buffer sizes allows to limit these productivity losses. We make a few classical assumptions on the production lines we analyze. None of them is restrictive in practice. First, the processing times are generally distributed but finite. Second, the buffer sizes are finite. These finiteness assumptions are not restrictive since they are always satisfied in practice. Third, we suppose that the line operates under saturation, i.e. the first station is never starved and the last station is never blocked. This assumption can easily be relaxed using an initial and a last station modelling the real arrival process and the real demand. Finally, the blocking policy is chosen to be blocking after service (a blocked job stays in station, after having been processed), the most common in manufacturing. The model could easily be modified in order to follow another policy.

The manufacturing systems it models being well defined, we now present the proposed global **modelling method**. The first step of the method builds discrete distributions by probability masses fitting (see section 2). To begin, the number of discrete values, a, and the parameter α are chosen (the number a is chosen for the longest distribution, it is consistently smaller for shorter distributions). The interval width τ is inferred from (1). Note that τ has to be the same for each station's distribution. Every processing time distribution is then transformed into a discrete distribution by PMF, i.e. by aggregating the probability mass in the interval $((j-1)\tau+\alpha, j\tau+\alpha]$ on the point $j\tau$, for $j = 1, 2, \ldots, a$, and the mass in $[0, \alpha]$ on 0 if PMF/IJ is used or τ if PMF/nIJ is used.

The discrete distribution can easily be formulated as a discrete phase–type distribution. An analytical model can then be used. Here, we use

a state model, which has the advantage to be exact. Approximate models (decomposition or expansion) could also be applied. The evolution of the production system is described by a Markov chain whose states are the possible combinations of the stages of the various stations and the utilizations of the various buffers. The performance of the production system can then be estimated from the analysis of the Markov chain. Transient performances, like the throughput at some time t, can be derived from the matrix of transition probabilities. The steady–state performances (the cycle time, the work in progress or the flow time for example) can be computed from the steady–state probabilities derived from the Markov chain.

The method has been named "Bounding Discrete Phase–type" (BDPH) since it relies on a discrete phase–type approximation of the various distributions of the system to be studied and since it leads to bounds [15]. If necessary, we specify if PMF/IJ or PMF/nIJ is used by the notations BDPH/IJ and BDPH/nIJ.

The method can be more clearly understood when explained on a simple **example**. Let us consider a line made of two stations separated by a buffer of size one. Figure 3 shows the application of the BDPH method, with both probability masses fitting alternatives: PMF/IJ and PMF/nIJ. Figure 3.a shows the real processing time distributions. The method can be applied directly to the distributions or on the histograms collected from them, as it is often the case in practice (figure 3.b). Our method works as follows. To begin, PMF transforms the distributions into the discrete distributions shown in figure 3.c, with two non–zero discrete values ($a = 2$) and $\alpha/\tau = 1/2$, i.e. aggregating the probability mass in the middle of the interval. The discrete processing time distributions can easily be represented as phase–type distributions (figure 3.d). Then, the behavior of the system can be modeled by a Markov Chain, i.e. using a state model. The Markov chains given in figure 3.e list all the possible recurrent states of the system 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 service (for example, 1 means that the station already spent one time step working on the current job). Each buffer is described by its utilization (0 or 1 for a buffer of size one). For example, state B12 means that the first station is blocked, that the buffer is full and that the second station already worked during two time steps on the current job. From a Markov chain and its stationary probabilities, the performance measures can be computed. For example, the idle (blocked/starved) probability of a machine is easily computed and the

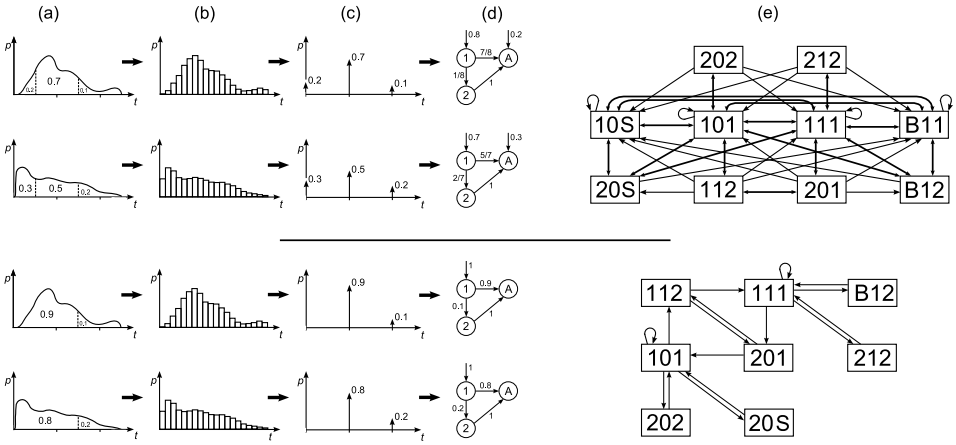

Figure 3: Steps of the BDPH method (with instant jobs in the upper part and without in the lower part) applied to a two–station line with processing times (a) collected in the form of histograms (b): construction of the tractable discrete distributions by PMF (c), PH representation (d) and Markov chain modelling the evolution of the line (e).

productivity easily deduced from it.

In figure 3, the difference between probability masses fitting with or without instant job appears, of course, in the processing time distributions (the first step of the PMF/nIJ case gathers two intervals of the PMF/IJ case) and, more interestingly, in the Markov chains that model the evolution of the system. The chain obtained when instant jobs are possible has more states and is clearly more dense. This will affect the speed of the method, in the construction of the chain as well as in its resolution, making the BDPH/IJ method slower (see section 5 for computational times).

The difference comes from the fact that instant jobs make some transitions possible, while they are impossible with PMF/nIJ. For example, using PMF without instant job, four transitions are possible from 101, depending if the stations continue to work on the same job or end. If the first station ends and the second one continues, the new state will be 112 as the first station will put its finished item in the buffer and begin to work on another one, while the second station will work on the same item for a second time step. If both stations finish their jobs, the first station will pass the item to the second station and both of them will begin a new job, leading to the state 101 again. The two other transitions from 101, and, finally, the whole

Markov chain can be found following this logic. If PMF with instant jobs is used, more transitions are possible. As can be seen on figure 3, eight transitions are possible from state 101, compared to four when PMF/nIJ is used. Let us have look at the transitions from 101, as we did for PMF/nIJ. If the first station ends and the second one continues its job, a possible transition leads to 112 (as for PMF/nIJ). However, if an instant job occurs on the first station, it will block the station and lead to B12. In a second case, if both stations finish their jobs, the transition from 101 to 101 is also possible. It takes place if no instant job occurs, as well as if an equal number of instant jobs occur on each station (an infinite number of them is possible, as they can occur simultaneously on every station). A transition from 101 to B11, for example, is also possible if two more instant jobs occur in the first station, filling the buffer and blocking the station. The Markov chain and its transition probabilities can be found applying this logic for every state.

To conclude this section, we comment on the **complexity** of the proposed method. The size of the Markov chain is, in first approximation, proportional to the number of individual stations and buffer states combinations, i.e. $\prod_{i=1}^{m}(a_i+2)\cdot\prod_{i=2}^{m}(b_i+1)$ with m the number of stations, a_i the number of non–zero discrete values in the discretized processing time distribution of station i, and b_i the size of buffer i. Note that this is a pessimistic estimation as a non–negligible amount of these individual combinations are impossible. It can be seen from the formula that the complexity increases when the number of discrete values a increases, in other words when the discrete time step τ decreases (leading to better accuracy). There is thus a clear trade–off between complexity and accuracy, which can be directly controlled by the parameter a. The size of the Markov chain also quickly increases with the complexity of the system's configuration (number of stations and buffer sizes). The explosion of the state space is the main limitation of the approach, to deal with realistic cases. It is essentially due to the fact that a state model is used in the last step of the method, and is quite independent of the discretization by probability masses fitting. This weakness can be overcome as usual when exact methods become too complex: using approximate methods. Decomposition or expansion models could be applied after probability masses fitting (see section 1).

4 Properties

In this section, our goal is to translate the properties we observed on probability masses fitting to the global modelling method. First, we reach one of the main results obtained thanks to probability masses fitting: upper and lower bounds on the productivity. Second, we show the monotonicity of the computed productivity in the shift parameter α. Third, we argue that the distribution shape conservation stays true for the computed productivity.

4.1 Bounding Methodology

In this subsection, we present the bounding methodology that shows that the BDPH method leads to bounds on the productivity. This methodology relies on two ideas. First, it involves an intelligible transformation, i.e. with good knowledge of its effect on the processing times. Second, the concept of critical path allows to translate the effect of the transformation to the global time.

In order to present the bounding methodology in a general form, we use the following formalism. We consider a probability masses fitting which transforms the original processing time $l^r(w_{i,k})$ to the discrete processing time $\tilde{l}^r(w_{i,k})$, such that $l^r(w_{i,k})$ can be bounded using its discretized value as followss:

$$\tilde{l}^r(w_{i,k}) - \delta_- \leq l^r(w_{i,k}) \leq \tilde{l}^r(w_{i,k}) + \delta_+, \qquad \forall\, r_P, i, k. \tag{4}$$

The terms δ_- and δ_+ stands for the maximum value by which a processing time can be increased and decreased, respectively, when discretized. Using PMF with instant jobs, we have $\delta_- = (\tau - \alpha)$ and $\delta_+ = \alpha$. Using PMF without instant job, we have $\delta_- = \tau$ and $\delta_+ = \alpha$.

4.1.1 Critical path

The notion of critical path is the key idea that allows to translate the properties of probability masses fitting on processing times to properties on the global production time. The notion of critical path has been introduced in a previous communication [16]. We recall its definition and a useful property. The critical path relies on the structural property of the manufacturing system. This property can be found for example in [8] for fork–join queueing networks with blocking. We state it for the particular case of flow lines in the following lemma. The moments job $w_{i,k}$ starts and ends are denoted $t_{start}(w_{i,k})$ and $t_{end}(w_{i,k})$.

Lemma 5. (Structural property of a flow line) *Given an m–station flow line including a buffer of size b_i before each station i, the moment a job starts is given by the following equation, $\forall\, i, k$:*

$$
\begin{aligned}
t_{start}(w_{i,k}) = max[\, & t_{end}(w_{i-1,k}), && (5)\\
& t_{end}(w_{i,k-1}), && (6)\\
& t_{end}(w_{i+1,k-b_{i+1}-2}), && (7)\\
& t_{end}(w_{i+2,k-b_{i+1}-b_{i+2}-3}), && (8)\\
& \ldots,\\
& t_{end}(w_{m,k-\sum_{j=i+1}^{m} b_j-(m-i+1)})\,]. && (9)
\end{aligned}
$$

Where, for notation purpose, $t_{end}(w_{0,k}) = 0, \forall\, k$.

This equation can easily be understood. In a few words, because of the line structure, a job k can only be started on station i: (5) if its processing in the previous station $i-1$ is ended, (6) if the processing of the previous job $k-1$ in station i is ended and (7-9) if this previous job $k-1$ is not blocked in station i by some unfinished jobs downstream. Furthermore, since there is no reason to wait once all these conditions are satisfied, $t_{start}(w_{i,k})$ will be given by the maximum of the right hand sides of Lemma 5.

The **critical path** of r_P, denoted $cp(r_P)$, is defined as the sequence of jobs that covers the production run r_P (producing P units) without gap and without overlap. By definition, the length of a run r_P (in other words the time to produce P units in this particular run) can thus be written as a sum of job lengths:

$$l(r_P) = \sum_{w_{i,k} \in cp(r_P)} l^r(w_{i,k}). \tag{10}$$

The critical path can be built quite easily. Starting with the last job that leaves the system, $w_{m,P}$, we can look which job end, in this precise run, has triggered its start. Repeating this process, we can proceed backwards in time until the start of the run, as every job start is triggered by the end of another job. It follows that every run r_P has at least one critical path.

The notion of critical path is well illustrated on the Gantt chart associated to a particular run. An example is given in figure 4. It can be seen that the critical path covers r_{11} and that each couple of successive jobs corresponds to one of the terms (5–9). The "most perceptive readers" can also observe on figure 4 that the predecessor of a job in the critical path can be deduced from the state of the same station just before this job. If the station is previously working, the predecessor is obviously on the same station. If the station is previously starved, the predecessor is on the previous

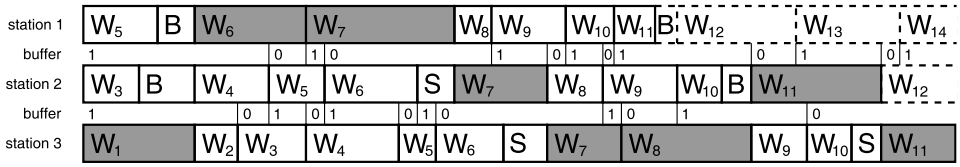

Figure 4: Gantt chart of a run r_{11} on a three station line with buffers of size one. The critical path is given in gray. The time goes from left to right. The state of a station at a given time is represented either by a letter (B for blocked, S for starved) or by the job currently processed. The state of a buffer is represented by its current utilization.

station ($w_{3,11}$ to $w_{2,11}$ for example). If the station is previously blocked, the predecessor is on a later station ($w_{2,11}$ to $w_{3,8}$). In the second case, the same item is twice in the critical path (item 11 in the example). In the third case, the $\sum(b_i + 1)$ items between both stations are "omitted" (items 9 and 10). From these observations, an upper bound can be inferred on $|cp(r_P)|$, the cardinality of the critical path, i.e. the number of jobs in it. Indeed, this upper bound corresponds to the worst case where the critical path goes up from the last station to the first one, without going down.

Lemma 6. *Let r_P be a production run on an m–station flow line, the number of jobs in the critical path satisfies the following upper bound:*

$$|cp(r_P)| \leq P + m - 1.$$

4.1.2 In a Particular Run

Here, we give the lemma on which the bounding methodology relies. It shows that, if the processing time is bounded, the time needed to achieve a given production in a particular run can also be bounded. The run obtained from the PMF discretization of the original run r_P is denoted $\tilde{r}_P$, and its length is denoted $l(\tilde{r}_P)$.

Lemma 7. *Given a PMF discretization of the processing times $l^r(w_{i,k})$ to $\tilde{l}^r(w_{i,k})$ verifying (4), the time a m–station line takes to produce P units in a production run r_P can be bounded as follows:*

$$l(\tilde{r}_P) - \delta_-(P + m - 1) \leq l(r_P) \leq l(\tilde{r}_P) + \delta_+(P + m - 1). \qquad (11)$$

Proof. These bounds follow from equations (4) and (10). However, these equations are not sufficient since the critical path is not the same in continuous and in discretized time.

The equation of Lemma 5 is valid for any run: a job $w_{i,k}$ cannot be started before all the jobs on the right hand side are finished. The absence of overlap in the critical path is thus independent of the considered run. However, which precise job end will trigger the start of job $w_{i,k}$ depends on the processing times and thus on the particular run we consider. In another run, gaps could appear between the job of the sequence $cp(r_P)$. The sequence $cp(r_P)$ is thus just a non–overlapping path (possibly with gaps) in the discretized run $\tilde{r}_p$ and its length is smaller than the length of the critical path $cp(\tilde{r}_p)$ in $\tilde{r}_p$. We get:

$$\begin{aligned} l(r_P) &\overset{(10)}{=} \sum_{w_{i,k} \in cp(r_P)} l^r(w_{i,k}) \overset{(4)}{\leq} \sum_{w_{i,k} \in cp(r_P)} \left(\tilde{l}^r(w_{i,k}) + \delta_+\right) \\ &\leq \sum_{w_{i,k} \in cp(\tilde{r}_p)} \tilde{l}^r(w_{i,k}) + \delta_+ |cp(r_P)| \overset{(10)}{=} l(\tilde{r}_P) + \delta_+ |cp(r_P)|. \end{aligned}$$

As, by Lemma 6, $|cp(\tilde{r}_P)| \leq P+m-1$, we get the right inequality of the present lemma. For the left inequality, using the same equations and the fact that $cp(\tilde{r}_p)$ is a non–overlapping path in the original run r_P, we get:

$$\begin{aligned} l(r_P) &\overset{(10)}{=} \sum_{w_{i,k} \in cp(r_P)} l^r(w_{i,k}) \geq \sum_{w_{i,k} \in cp(\tilde{r}_P)} l^r(w_{i,k}) \\ &\overset{(4)}{\geq} \sum_{w_{i,k} \in cp(\tilde{r}_P)} \left(\tilde{l}^r(w_{i,k}) - \delta_-\right) \overset{(10)}{=} l(\tilde{r}_P) - \delta_- |cp(\tilde{r}_P)|. \end{aligned}$$

As $|cp(\tilde{r}_P)| \leq P + m - 1$ (Lemma 6), this ends the proof. □

Lemma 7 provides a major result. Considering any sample production run, bounds on the individual processing times can be extended to bounds on the time it takes to produce a given production. Unfortunately, this result cannot yet be directly used since it refers to a sample production run. For the results to be useful, we need to be able to say something about an average production run. This point is tackled in the following.

4.1.3 Transient Throughput

We first extend lemma 7 to the expected time T_P necessary to reach a given production P. By definition, T_P ($\tilde{T}_P$) equals the sum of the lengths of all possible original (discretized) runs r_P ($\tilde{r}_P$), weighted by their probabilities. Consequently, we simply have to check that each of the three terms of equation (11) is weighted in the same way. That comes from the definition of probability masses fitting. As it aggregates the probability masses, the total probability in continuous time of all the runs r_P which discretize to the same run $\tilde{r}_P$ is equal to the probability of this run $\tilde{r}_P$ in discretized time. As this is true for each discrete run and as an original run has only one discrete correspondent, we can extend Lemma 7 to the following proposition.

Proposition 8. *Given a PMF discretization of the processing times $l^r(w_{i,k})$ to $\tilde{l}^r(w_{i,k})$ verifying (4), the expected time T_P a m–station line takes to produce P units can be bounded using $\tilde{T}_P$, the expected time to produce P units computed in the discretized time. We have:*

$$\tilde{T}_P - \delta_-(P+m-1) \leq T_P \leq \tilde{T}_P + \delta_+(P+m-1). \tag{12}$$

Proof. By definition, the expected time to produce P, T_P, equals the sum of the lengths of all possible runs r_P, weighted by their probabilities. More formally, we have: $T_P = \int f(r_P)l(r_P)dr_P$, where $f(r_P)$ is the density function of the runs r_P. To get (12) from (11), we have to check that each of the three terms of (11) is weighted in the same way. We thus relate r_P to its discretized correspondent, $\tilde{r}_P$ (which also produces P). Let us note $\gamma(\tilde{r}_P)$ the set of original runs r_P (in infinite number) which have the same discretized correspondent $\tilde{r}_P$. We can decompose the previous integral and use lemma 7, to get:

$$T_P = \sum_{\tilde{r}_P} \int_{r_P \in \gamma(\tilde{r}_P)} f(r_P)l(r_P)dr_P \leq \sum_{\tilde{r}_P} \left(l(\tilde{r}_P) + \delta_+(P+m-1)\right) \int_{r_P \in \gamma(\tilde{r}_P)} f(r_P)dr_P.$$

As the PMF discretization simply aggregates the probability masses in intervals, $\int_{r_P \in \gamma(\tilde{r}_P)} f(r_P)dr_P$ gives the probability of the run $\tilde{r}_P$, i.e. $P[\tilde{r}_P]$. As $\sum_{\tilde{r}_P} P[\tilde{r}_P] = 1$, we get:

$$T_P \leq \sum_{\tilde{r}_P} l(\tilde{r}_P)P[\tilde{r}_P] + \delta_+(P+m-1) = \tilde{T}_P + \delta_+(P+m-1).$$

The way to the lower bound is very similar. Using lemma 7, we get:

$$T_P \geq \sum_{\tilde{r}_P} \left(l(\tilde{r}_P) - \delta_-(P+m-1)\right) \int_{r_P \in \gamma(\tilde{r}_P)} f(r_P)dr_P = \tilde{T}_P - \delta_-(P+m-1).$$

□

Note that bounds on the production reached in a fixed time can be quite easily derived from the previous proposition.

4.1.4 Steady–State Productivity

When focusing on the steady–state productivity, the results get even simpler. The cycle time c is defined as the average time, in steady–state, between the completion of two units by the line, i.e. $c = \lim_{P\to\infty} T_P/P$. The cycle time in the discretized time is denoted $\tilde{c}$, i.e. $\tilde{c} = \lim_{P\to\infty} \tilde{T}_P/P$.

Proposition 9. *Given a PMF discretization of the processing times $l^r(w_{i,k})$ to $\tilde{l}^r(w_{i,k})$ verifying (4), the cycle time c can be bounded using $\tilde{c}$, the cycle time computed in the discretized time:*

$$\tilde{c} - \delta_- \leq c \leq \tilde{c} + \delta_+. \tag{13}$$

Proof. These bounds straightforwardly follows from proposition 8 by dividing (12) by P and making $P \to \infty$. □

Propositions 8 and 9 show that our bounding methodology allows to extend bounds on the individual processing times to prove bounds on the productivity, in transient or in steady–state, from below and from above. The similarity of equations (4) and (13) is remarkable in the present case of flow lines. We believe that this bounding methodology is an interesting contribution. It is clear that the approach is not restricted to the present application. It could be extended to more complex systems (as a critical path can be defined for them too). Moreover, note that bounds can easily be inferred on an other performance measure, namely the idle (blocked/starved) time. Indeed, the latter equals the cycle time minus the processing time, on which both we have bounds.

The bounding methodology presented in this subsection can be compared to the bounding methodology proposed in [3]. The latter also use the structural property of the system, in the form of a recursion equation. It is based on the stochastic ordering of the random variables representing the processing times. However, direct relations between the processing times are required, i.e. the additional terms (δ_- and δ_+) are not allowed.

4.1.5 Bounds by Probability Masses Fitting

The results given earlier can readily be instantiated to probability masses fitting with or without instant job. We just give the bounds on the cycle time. Obviously, the bounds on the transient throughput could also be instantiated.

Corollary 10. *The BDPH/IJ method allows to compute upper and lower bounds on the cycle time. With c_α the cycle time in the time discretized by PMF/IJ, we have:*

$$c_\alpha - (\tau - \alpha) \leq c \leq c_\alpha + \alpha.$$

Proof. This corollary is straightforward from propositions 1 and 9 ($\delta_- = (\tau - \alpha)$ and $\delta_+ = \alpha$). □

Corollary 11. *The BDPH/nIJ method allows to compute upper and lower bounds on the cycle time. With $c_{\alpha'}$ the cycle time in the time discretized by PMF/nIJ, we have:*

$$c_{\alpha'} - \tau \leq c \leq c_{\alpha'} + \alpha.$$

Proof. This corollary is straightforward from propositions 3 and 9 ($\delta_- = \tau$ and $\delta_+ = \alpha$). □

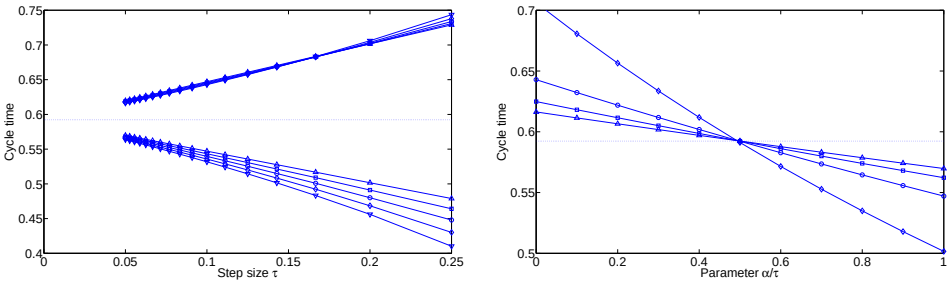

Figure 5: The BDPH/IJ method is applied an example. On the left–hand side, the bounds on the cycle time are computed for various step sizes τ, i.e. various numbers a of steps, with $\alpha = 0$ ($\bigtriangledown$), 0.25 ($\Diamond$), 0.5 ($\odot$), 0.75 ($\Box$), 1 ($\bigtriangleup$). On the right–hand side, the cycle time c_α is computed for various parameters α/τ and with 5 ($\Diamond$), 10 ($\odot$), 15 ($\Box$), or 20 ($\bigtriangleup$) time steps.

These results show that the BDPH method allows to bound the productivity, from above and from below. Furthermore, the accuracy of the bounds is related to the selected step size τ. The bounds thus become tighter, and converge, when the discretization step is decreased, i.e. when the number of discrete values is increased. In other words, the bounds are refinable. Moreover, it allows to a priori choose the desired accuracy of the results. Of course, every accuracy improvement will require additional computational efforts caused by the increase of the state space size (which can be estimated, see section 3).

It can also be seen from these corollaries that probability masses fitting with instant jobs offers better bounds than PMF without instant job. The gap between both bounds is smaller. In particular, PMF/IJ allows to compute better lower bounds. This is natural since the absence of instant job requires the interval on which the first probability mass is collected to be larger (see figure 2), and thus the lower bound on processing times to be worse. On the left–hand side of figure 5, we draw the bounds obtained by the method with instant jobs, for 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). It can be seen that the bounds get sharper when the step size τ decreases. Moreover, the figure reveals that PMF/IJ with the shift parameter α equal to one, i.e. aggregating the probability mass in the beginning of the interval, leads to the best bounds. It comes from the fact that the step size τ, and thus the distance between both bounds, decreases

when α increases ($\tau = (max - \alpha)/a$, see equation (1)). More precisely, the lower bound shows to be better with $\alpha = 1$, while nothing can be said about the upper bound. As will be shown in Section 5 by computational experiments, this observation appears to be not particular to the present example.

4.2 Monotonicity

In the previous section, we extended the bounding property of probability masses fitting to the global modelling method. Similarly, the monotonicity in α of the discretized processing times can also be extended. It can be shown using the critical path as well but it is more easily shown as a corollary of a result given in [2], which uses the fact that the discretized processing times can be stochastically ordered, and follows the bounding methodology proposed in [3]. The monotonicity in α of the transient throughput could be shown in the same way.

Proposition 12. *Using the BDPH method with two parameters α such that $\alpha_1 \leq \alpha_2$, the first cycle time will always be larger than the second one.*

$$c_{\alpha_1} \geq c_{\alpha_2} \qquad \textit{and} \qquad c_{\alpha'_1} \geq c_{\alpha'_2}.$$

Proof. The discretized random variables using α_1 or α_2 can be compared stochastically. Proposition 2 (and 4) tells that a realization of the random variable $l^r_{\alpha_1}(w_{i,k})$ (and $l^r_{\alpha'_1}(w_{i,k})$) is always larger than the corresponding realization $l^r_{\alpha_2}(w_{i,k})$ (and $l^r_{\alpha'_2}(w_{i,k})$). We thus have the stochastic ordering inequality $l^r_{\alpha_1}(w_{i,k}) \geq_{st} l^r_{\alpha_2}(w_{i,k})$ (and $l^r_{\alpha'_1}(w_{i,k}) \geq_{st} l^r_{\alpha'_2}(w_{i,k})$). The following increasing convex ordering inequalities are straightforwardly deduced:

$$l^r_{\alpha_1}(w_{i,k}) \geq_{icx} l^r_{\alpha_2}(w_{i,k}), \qquad \text{and} \qquad l^r_{\alpha'_1}(w_{i,k}) \geq_{icx} l^r_{\alpha'_2}(w_{i,k}).$$

Consequently, as flow lines with finite buffers can be modeled by stochastic decision free Petri nets, Corollary 5.1 of [2] applies and proves the proposition. □

These results reveal the evolution of the results of the BDPH method in function of the shifting parameter α. When α is increased, the cycle time estimation decreases. Unfortunately, such a monotonic property cannot be proved on the bounds (corollaries 11 and 10). In the right–hand side of figure 5, we illustrate the evolution of the cycle time c_α in α. The decreasing characteristic can be clearly observed.

4.3 Shape Conservation

Another interesting property of the probability masses fitting approach lies in the preservation of the shape of the original distribution (see section 2.4).

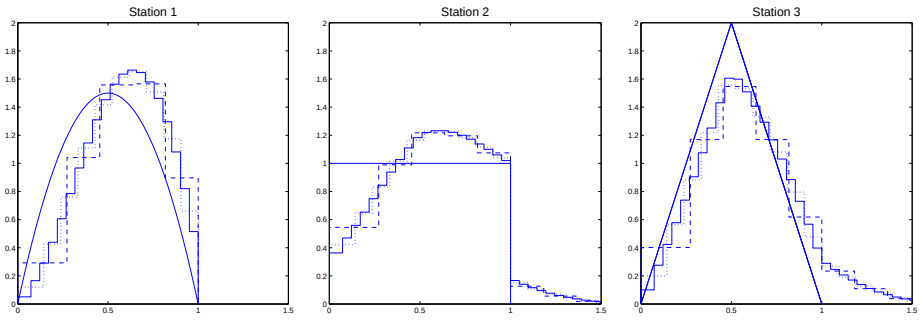

Figure 6: Cycle time distributions for the three stations of a line, with 5 (dashed), 10 (dotted) and 20 (solid) discretization steps (with PMF/nIJ). The original processing time distributions are also depicted.

Here, like in previous subsections, we want to extend this PMF property to the global modelling method result. We argue that the shape conservation extends to the cycle time distributions, i.e. the distributions of the time between two job exits from a given station.

Let us have a look at the example already analyzed in the previous subsections: a three station line with buffers of size one and processing time distributions beta$(2, 2)$, uniform$(0, 1)$ and triangular$(0, 1, 0.5)$. Our method allows to compute discrete estimations of the cycle time distributions. Figure 6 depicts, for each station, the original processing time distributions and the computed discrete cycle time distributions. The three graphs clearly show how starving and blocking impact the cycle time compared to the raw processing time. The cycle time distributions are computed with 5, 10 and 20 steps. As our method refines and converges when the number of discretization steps increases, it can be supposed that the cycle time distribution computed with 20 steps is accurate. Figure 6 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, that their shape is realistic.

In conclusion, we may say that the distribution shape conservation stays true for the cycle time. Even with 5 or 10 steps, the shape of the cycle time distribution estimation is already realistic. The computation of a good cycle time distribution estimation is a significant advantage of our method, notably compared to moments fitting. 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 quantiles. Note that other distributions, such as the distribution of the flow time, can also be computed.

5 Computational Results

In this section, we show by computational experiments how our modelling method behaves. The impact of the number of discretization steps and of the shift parameter α is shown for the bounds as well as for the estimation of the cycle time. Both probability masses fitting alternatives, with or without instant job, are distinguished. Moreover, the accuracy levels reached for the work–in–progress and for the flow time are given. We then compare these results to the computation time needed to reach them. After that, we study the impact of the configuration of the line, i.e. the number of stations and the buffer sizes, on the accuracy of the performance evaluation.

To begin, we study an assortment of 500 three–station flow lines. The buffer configurations vary from [0 0] up to [2 2]. They are equally shared out according to the global storage space (sum of both buffer sizes) and the space is uniformly divided, i.e. we analyze 100 [0 0] configurations, 100 either [1 0] or [0 1] configurations, 100 [1 1] configurations, 100 either [2 1] or [1 2] configurations, and 100 [2 2] configurations. The processing time distributions are randomly chosen among the 10 following distributions: 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). These distributions have various expectations and various coefficients of variation. The 500 lines are analyzed using both PMF alternatives (with or without instant job). Five different shift parameters α are used ($\alpha/\tau = 0, 0.25, 0.5, 0.75, 1$) as well as, with PMF/IJ, the parameters α that conserve the expectations of the original distributions (see proposition 2). The number a of non–zero values in the discretized distributions varies from five up to ten. We thus made a total of 33000 (500 lines $\cdot$(5 nIJ +6 IJ) $\cdot$ 6 a) experiments. The 500 three–station lines were also analyzed by simulation, in order to assess the accuracy of our method. In the following, we give the average accuracy gap, in percents, between the results of our method and the results of the simulation (i.e. $|res_{bdph} - res_{simu}|/res_{simu}$). It is given for various parameter configurations and for various performance measures.

To begin, we are interested in the **bounds** on the cycle time. In subsection 4.1.5, we argued that probability masses fitting with instant jobs

a	5	6	7	8	9	10
$\alpha/\tau = 0$	17.4	14.5	12.4	10.9	9.7	8.7
$\alpha/\tau = 0.25$	16.3	13.7	11.8	10.4	9.3	8.4
$\alpha/\tau = 0.5$	15.3	12.9	11.2	9.9	8.8	8
$\alpha/\tau = 0.75$	14.3	12.2	10.6	9.4	8.4	7.6
$\alpha/\tau = 1$	**13.4**	11.4	10	8.9	8	**7.3**
Exp. cons.	15.5	13.1	11.3	10	8.9	8.1

Table 1: Average accuracy gap, in percent, reached on the lower bound by the BDPH/IJ method. The number of non–zero values in the discrete distribution increases from left to right and the shift parameter α increases from top to bottom. The last row stands for α chosen in order to conserve the expectation of the original distributions.

offers better bounds. Moreover, we showed on a simple example that the lower bound appears to be better with $\alpha/\tau = 1$, i.e. the probability mass is aggregated in the beginning of the interval. This observation stays true on the computational experiments, as illustrated in Table 1. This table shows the accuracy of the lower bound computed with PMF/IJ as a function of the shift parameter α and the number a of non–zero values in the discretized distribution. It can be seen that the bound improves when a or α increases. However, as expected, the bounds are not accurate (it is known that the gap between both bounds equals the step size τ).

Of course, estimations of the **cycle time** can also be computed by the BDPH method. They should lead to better accuracy. The cycle time computed directly in the discretized time is not the best possible estimation (except whit shift parameters α that conserve the expectation). Indeed, as previously explained, in general, the distribution discretized by PMF does not have the same expectation as the original processing time distribution. But the error on the average processing time is known, as the expectations of both original and discretized distributions are known. It can thus be subtracted in order to get a better estimation of the cycle time, which is composed of the average processing time plus the average idle time. This estimation thus only includes the error on the idle (blocked/starved) time. Table 2 shows the accuracy reached with various parameters α and various number a of non–zero values in the discretized distributions. First of all, this table show that our method leads to accurate estimations of the cycle time. The error made by the method amounts to a few tenths of percent.

a	5	6	7	8	9	10
$\alpha/\tau = 0$	0.63	0.53	0.46	0.42	0.37	0.33
$\alpha/\tau = 0.25$	0.35	0.26	0.22	0.20	0.18	0.16
$\alpha/\tau = 0.5$	**0.31**	0.22	0.18	0.14	0.12	**0.1**
$\alpha/\tau = 0.75$	0.51	0.39	0.33	0.28	0.25	0.22
$\alpha/\tau = 1$	0.78	0.63	0.53	0.46	0.41	0.37
Exp. cons.	**0.21**	0.15	0.13	0.1	0.09	**0.08**
Comp. time	*1.4 s*	*2.1 s*	*3.5 s*	*5.7 s*	*10 s*	*18 s*

a	5	6	7	8	9	10
$\alpha/\tau = 0$	0.63	0.53	0.46	0.42	0.37	0.33
$\alpha/\tau = 0.25$	0.52	0.38	0.31	0.27	0.23	0.21
$\alpha/\tau = 0.5$	**0.31**	0.21	0.16	0.13	0.11	**0.09**
$\alpha/\tau = 0.75$	0.86	0.6	0.47	0.38	0.31	0.27
$\alpha/\tau = 1$	1.21	0.87	0.68	0.56	0.48	0.41
Comp. time	*0.2 s*	*0.3 s*	*0.7 s*	*1.7 s*	*2.2 s*	*5.8 s*

Table 2: Average accuracy gap, in percent, reached on the cycle time estimation by the BDPH/IJ method (upper table) and by the BDPH/nIJ (lower part). The number of non–zero values in the discrete distributions increases from left to right and the shift parameter α increases from top to bottom (in each part). The row "Exp. cons." stands for α that conserve the expectation. The computational times (in seconds) for the [2 2] buffer configuration are given in italic.

Of course, the accuracy improves when the number of discretization steps increases. A 0.1 percent accuracy is reached with $a = 10$, while a 0.3 percent accuracy is already obtained with five non–zero values in the discretized distribution. Furthermore, it can be seen that the best results are obtained with $\alpha/\tau = 0.5$ and with α which conserves the expectation. This is quite natural: aggregating the probability masses in the middle of the interval offer a better approximation of the distribution. This choice leads to a better estimation of the cycle time, which means a better estimation of the idle time. It can also be seen in table 2 that the best absolute results are obtained choosing the shift parameter α in order to conserve the expectation of the original distribution, using PMF/IJ. However, when the computational time[1] is also taken into account, the best option shows to be PMF/nIJ with

[1]Using the Gaussian elimination implemented in MATLAB® on a 2.16 GHz usual PC, 2 GB RAM.

a	5	6	7	8	9	10
PMF/IJ, $\alpha/\tau = 0.5$	0.38	0.27	0.21	0.17	0.15	0.13
PMF/IJ, Exp. cons.	0.2	0.16	0.13	0.12	0.11	0.1
PMF/nIJ, $\alpha/\tau = 0.5$	**0.36**	0.23	0.18	0.15	0.12	**0.11**
a	5	6	7	8	9	10
PMF/IJ, 0.5	0.48	0.34	0.26	0.21	0.18	0.16
PMF/IJ, Exp. cons.	0.26	0.2	0.17	0.14	0.13	0.12
PMF/nIJ, 0.5	**0.41**	0.27	0.21	0.17	0.14	**0.13**

Table 3: Average accuracy gap, in percent, reached on the work–in–progress estimation (upper part) and on the flow time estimation (lower part), using PMF/IJ as well as PMF/nIJ. The number of non–zero values in the discrete distribution increases from left to right.

$\alpha/\tau = 0.5$, i.e. probability masses fitting aggregating in the middle of the interval, without instant job. Indeed, for example, an average gap of 0.21% is reached in 0.3 seconds with PMF/nIJ while PMF/IJ needs 1.4 seconds. Similarly, PMF/nIJ leads to an accuracy gap of 0.13% in 1.7 seconds while PMF/IJ takes 3.5 seconds.

Table 3 show the accuracy gap obtained on two other performance measures: the **work–in–progress** (WIP) and the **flow time**. The WIP is easily computed from the occupations of the buffers and of the stations while the flow time is deduced from the Little's law. The table shows that our method leads to accurate estimations of these performances measures too. As for the cycle time, the level of accuracy expresses in tenths of percent. Again, the PMF/IJ with α that conserve the expectation leads to the better absolute results. When the computation times given in table 2 are taken into account, it can be seen that, like for the cycle time, PMF/nIJ aggregating in the middle of the interval reveals to be the best choice. In the computation of the work–in–progress, for example, an accuracy gap of 0.12% is reached in 2.2 seconds, compared to 5.7 seconds with PMF/IJ. Concerning the flow time, a gap of 0.17% is obtained in 1.7 seconds, compared to 3.5 seconds.

At this point, it is interesting to study how the method behaves when the **configuration** of the line changes, i.e. when the number of stations and the buffer sizes vary. For this, we run a new set of experiments: 500 other flow lines are analyzed. Half of them is made of 2 stations and the other half is made of 4 stations. As for the three station lines previously studied, the total storage space (sum of every buffer sizes) vary from zero to four, and

$\sum b_i$	0	1	2	3	4	Total
2 stations	0.11	0.11	0.13	0.14	0.14	0.13
	0.1 s	*0.1 s*	*0.1 s*	*0.1 s*	*0.1 s*	
3 stations	0.15	0.12	0.13	0.12	0.11	0.13
	0.1 s	*0.2 s*	*0.3 s*	*0.4 s*	*1.7 s*	
4 stations	0.15	0.12	0.12	0.13	0.12	0.13
	0.9 s	*1.6 s*	*4.7 s*	*54 s*	*200 s*	
Total	0.14	0.12	0.13	0.13	0.12	0.13

Table 4: Average accuracy gap, in percent, reached on the cycle time estimation with PMF/nIJ aggregating in the middle of the interval. The results are given for lines with various number of stations and various total storage space (sum of every buffer size). The computational times are given in italic.

is uniformly shared out. The processing time distributions are randomly chosen among the 10 distributions listed previously. For this experiment, we focus on the cycle time estimation computed by the BDPH/nIJ method with $\alpha/\tau = 0.5$, i.e. aggregating the probability mass in the middle of the interval, which showed to be the best option in the previous experiments. Moreover, we chose a, the number of non–zero values in the discretized distributions, to be equal to eight. This choice leads to excellent accuracy in previous experiments (0.13%, see Table 2), while still increasing a only offers marginal benefit. An other choice of a would lead to the same conclusions. Again, the 500 new flow lines are also analyzed by simulation, in order to assess the accuracy of our method. We give the average accuracy gap, in percents, between the results of our method and the results of the simulation.

From table 4, the first conclusion is obvious: the accuracy of the estimations reveals to be remarkably stable. Whatever the dimension which varies, the number of stations or the buffer sizes, the accuracy gap observed reveals to be nearby constant ($0.13 \pm 0.02\%$). In other words, the accuracy of the results does not deteriorate when the configuration gets more complex. This is another advantage of our method. Table 4 also shows the computational time needed. It illustrates what we already revealed previously: the complexity, due to the state model, is the main weakness of the global modelling method. The computational time increases very quickly when the system gets more complex.

In summary, interesting observations can be drawn from the computational experiments. First, the best lower bound is offered by PMF/IJ

aggregating in the beginning of the interval ($\alpha = \tau$). Second, and most important, our method computes accurate estimations of the performance measures (cycle time, work–in–progress and flow time). Average accuracy gaps express in a few tenths of percent. Even with five non–zero values in the discretized distributions, i.e. with five phases in the phase–type distributions, excellent accuracy levels are reached (0.3%). Third, the accuracy level reached by the estimations shows to be remarkably stable when the system's configuration changes.

6 Conclusion

We introduce a new approach in order to build tractable phase–type distributions which are required by analytical models of stochastic manufacturing systems. Called "probability masses fitting" (PMF), its principle is simple: the probability masses on regular intervals are computed and aggregated on a single value in the corresponding interval, leading to a discrete distribution. The place where the value lies in the interval is chosen by the shift parameter α. Moreover, two alternative PMF are proposed, generating potential instant jobs, i.e. processing times of length zero, or not. Two characteristics follow directly from the definition. First, PMF is refinable: the approximation becomes more and more accurate when the interval size is decreased or, equivalently, when the number of intervals is increased. Second, PMF conserves the shape of the original distribution. However, PMF does not, in general, conserve the first moments. Two other properties of probability masses fitting were shown in the text. First, the discretized times allow to bound the original times. Second, the evolution of the discretized times in the shift parameter α is a monotonic, decreasing, function. Moreover, we showed that, using probability masses fitting with instant jobs, the shift parameter α can be chosen in order to conserve the mean of the original distribution.

After probability masses fitting, various state–of–the–art analytical models can be applied. In this paper, we chose a state model. In other words, from the discrete phase–type distributions built by PMF, the evolution of the system is modeled by a Markov chain and the performances are evaluated from it. A state model has the advantage to be exact but has a high complexity. The Markov chain size increases quickly when the system size increases. Here, the method is applied to manufacturing flow lines. The properties shown on probability masses fitting can be extended to the global modelling method. The first remarkable result concerns bounds on the pro-

ductivity. Using the concept of critical path, we prove refinable upper and lower bounds on the transient throughput and on the cycle time. PMF with instant jobs leads to tighter bounds and, in particular, a better lower bound is obtained when the probability masses are aggregated in the beginning of the interval. Moreover, the monotonocity of PMF can be extended to the cycle time computed by the global method. Finally, interestingly, the proposed method allows to compute a realistic distribution of the cycle time, what is a far more detailed information than the isolated expectation and cannot be computed by other methods.

On computational experiments, our modelling method proves to offer accurate estimations of various performance measures, namely the cycle time, the work–in–progress and the flow time. Using only five discretization steps, an accuracy level of 0.3% is reached. Comparing various parameters options, PMF without instant job aggregating the probability mass in the middle of the interval appears to be the best choice, in the sense of the trade–off between accuracy and computational time. When the configuration of the line (number of stations and buffer sizes) is changed, the results' accuracy reveals to be remarkably stable.

In conclusion, we believe that probability masses fitting can be thought as a valuable alternative in order to build tractable distributions for the analytical modelling of manufacturing systems. In order to improve its applicability, and avoid the complexity of state models, approximate analytical models could be applied after PMF. In particular, decomposition methods could take advantage of the computation of detailed and realistic distributions of the cycle time of each station. Moreover, the probability masses fitting approach could of course be applied to more complex manufacturing systems, such as assembly/disassembly systems.

References

[1] T. Altiok. *Performance Analysis of Manufacturing Systems*. Springer, New York, 1996.

[2] F. Baccelli and Z. Liu. Comparison properties of stochastic decision free petri nets. *IEEE Trans. On Automatic Control*, 37:1905–1920, 1992.

[3] F. Baccelli and A.M. Makowski. Queueing models for systems with synchronization constraints. *Proceedings of the IEEE*, 77:138–161, 1989.

[4] A. Bobbio, A. Horváth, M. Scarpa, and M. Telek. Acyclic discrete phase

type distributions : Properties and a parameter estimation algorithm. *Performance Evaluation*, 54:1–32, 2003.

[5] J.A. Buzacott and J.G. Shanthikumar. *Stochastic Models of Manufacturing Systems*. Prentice-Hall, Englewood Cliffs, New Jersey, 1993.

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

[7] Y. Dallery and S.B. Gershwin. Manufacturing flow line systems: a review of models and analytical results. *Queueing Systems*, 12:3–94, 1992.

[8] Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility, symmetry and concavity properties in fork/join queueing networks with blocking. *Journal of the ACM*, 41:903–942, 1994.

[9] S.B. Gershwin. *Manufacturing Systems Engineering*. Prentice-Hall, Englewood Cliffs, New Jersey, 1994.

[10] M.K. Govil and M.C. Fu. Queueing theory in manufacturing : A survey. *Journal of manufacturing systems*, 18(4):214–240, 1999.

[11] F.S. Hillier and R.W. Boling. Finite queues in series with exponential or erlang service times—a numerical approach. *Operations Research*, 15(2):286–303, 1967.

[12] L. Kerbache and J. MacGregor Smith. The generalized expansion method for open finite queueing networks. *European Journal of Operational Research*, 32:448–461, 1987.

[13] H.T. Papadopoulos and C. Heavey. 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, 1996.

[14] H.G. Perros. *Queueing Networks with Blocking : Exact and Approximate Solutions*. Oxford University Press, 1994.

[15] J.-S. Tancrez and P. Semal. The bounding discrete phase-type method. In A. N. Langville and W. J. Stewart, editors, *MAM 2006: Markov Anniversary Meeting*, pages 257–269, Raleigh, North Carolina, USA, 2006. Boson Books.

[16] J.-S. Tancrez, P. Semal, and P. Chevalier. Histogram based bounds and approximations for production lines. *European Journal of Operational Research*, to appear.

[17] Marcel van Vuuren, Ivo J.B.F. Adan, and Simone A.E. Resing-Sassen. Performance analysis of multi-server tandem queues with finite buffers and blocking. *OR Spectrum*, 27:315–338, 2005.

Recent titles

CORE Discussion Papers

2007/85. Jean-Pierre FLORENS, Jan JOHANNES and Sébastien VAN BELLEGEM. Identification and estimation by penalization in nonparametric instrumental regression.
2007/86. Louis EECKHOUDT, Johanna ETNER and Fred SCHROYEN. A benchmark value for relative prudence.
2007/87. Ayse AKBALIK and Yves POCHET. Valid inequalities for the single-item capacitated lot sizing problem with step-wise costs.
2007/88. David CRAINICH and Louis EECKHOUDT. On the intensity of downside risk aversion.
2007/89. Alberto MARTIN and Wouter VERGOTE. On the role of retaliation in trade agreements.
2007/90. Marc FLEURBAEY and Erik SCHOKKAERT. Unfair inequalities in health and health care.
2007/91. Frédéric BABONNEAU and Jean-Philippe VIAL. A partitioning algorithm for the network loading problem.
2007/92. Luc BAUWENS, Giordano MION and Jacques-François THISSE. The resistible decline of European science.
2007/93. Gaetano BLOISE and Filippo L. CALCIANO. A characterization of inefficiency in stochastic overlapping generations economies.
2007/94. Pierre DEHEZ. Shapley compensation scheme.
2007/95. Helmuth CREMER, Pierre PESTIEAU and Maria RACIONERO. Unequal wages for equal utilities.
2007/96. Helmuth CREMER, Jean-Marie LOZACHMEUR and Pierre PESTIEAU. Collective annuities and redistribution.
2007/97. Mohammed BOUADDI and Jeroen V.K. ROMBOUTS. Mixed exponential power asymmetric conditional heteroskedasticity.
2008/1. Giorgia OGGIONI and Yves SMEERS. Evaluating the impact of average cost based contracts on the industrial sector in the European emission trading scheme.
2008/2. Oscar AMERIGHI and Giuseppe DE FEO. Privatization and policy competition for FDI.
2008/3. Wlodzimierz SZWARC. On cycling in the simplex method of the Transportation Problem.
2008/4. John-John D'ARGENSIO and Frédéric LAURIN. The real estate risk premium: A developed/emerging country panel data analysis.
2008/5. Giuseppe DE FEO. Efficiency gains and mergers.
2008/6. Gabriella MURATORE. Equilibria in markets with non-convexities and a solution to the missing money phenomenon in energy markets.
2008/7. Andreas EHRENMANN and Yves SMEERS. Energy only, capacity market and security of supply. A stochastic equilibrium analysis.
2008/8. Géraldine STRACK and Yves POCHET. An integrated model for warehouse and inventory planning.
2008/9. Yves SMEERS. Gas models and three difficult objectives.
2008/10. Pierre DEHEZ and Daniela TELLONE. Data games. Sharing public goods with exclusion.
2008/11. Pierre PESTIEAU and Uri POSSEN. Prodigality and myopia. Two rationales for social security.
2008/12. Tim COELLI, Mathieu LEFEBVRE and Pierre PESTIEAU. Social protection performance in the European Union: comparison and convergence.
2008/13. Loran CHOLLETE, Andréas HEINEN and Alfonso VALDESOGO. Modeling international financial returns with a multivariate regime switching copula.
2008/14. Filomena GARCIA and Cecilia VERGARI. Compatibility choice in vertically differentiated technologies.
2008/15. Juan D. MORENO-TERNERO. Interdependent preferences in the design of equal-opportunity policies.
2008/16. Ana MAULEON, Vincent VANNETELBOSCH and Wouter VERGOTE. Von Neumann-Morgenstern farsightedly stable sets in two-sided matching.
2008/17. Tanguy ISAAC. Information revelation in markets with pairwise meetings: complete information revelation in dynamic analysis.

Recent titles

CORE Discussion Papers - continued

2008/18. Juan D. MORENO-TERNERO and John E. ROEMER. Axiomatic resource allocation for heterogeneous agents.
2008/19. Carlo CAPUANO and Giuseppe DE FEO. Mixed duopoly, privatization and the shadow cost of public funds.
2008/20. Helmuth CREMER, Philippe DE DONDER, Dario MALDONADO and Pierre PESTIEAU. Forced saving, redistribution and nonlinear social security schemes.
2008/21. Philippe CHEVALIER and Jean-Christophe VAN DEN SCHRIECK. Approximating multiple class queueing models with loss models.
2008/22. Pierre PESTIEAU and Uri M. POSSEN. Interaction of defined benefit pension plans and social security.
2008/23. Marco MARINUCCI. Optimal ownership in joint ventures with contributions of asymmetric partners.
2008/24. Raouf BOUCEKKINE, Natali HRITONENKO and Yuri YATSENKO. Optimal firm behavior under environmental constraints.
2008/25. Ana MAULEON, Vincent VANNETELBOSCH and Cecilia VERGARI. Market integration in network industries.
2008/26. Leonidas C. KOUTSOUGERAS and Nicholas ZIROS. Decentralization of the core through Nash equilibrium.
2008/27. Jean J. GABSZEWICZ, Didier LAUSSEL and Ornella TAROLA. To acquire, or to compete? An entry dilemma.
2008/28. Jean-Sébastien TANCREZ, Philippe CHEVALIER and Pierre SEMAL. Probability masses fitting in the analysis of manufacturing flow lines.

Books

Y. POCHET and L. WOLSEY (eds.) (2006), *Production planning by mixed integer programming*. New York, Springer-Verlag.
P. PESTIEAU (ed.) (2006), *The welfare state in the European Union: economic and social perspectives*. Oxford, Oxford University Press.
H. TULKENS (ed.) (2006), *Public goods, environmental externalities and fiscal competition*. New York, Springer-Verlag.
V. GINSBURGH and D. THROSBY (eds.) (2006), *Handbook of the economics of art and culture*. Amsterdam, Elsevier.
J. GABSZEWICZ (ed.) (2006), La différenciation des produits. Paris, La découverte.
L. BAUWENS, W. POHLMEIER and D. VEREDAS (eds.) (2008), *High frequency financial econometrics: recent developments*. Heidelberg, Physica-Verlag.
P. VAN HENTENRYCKE and L. WOLSEY (eds.) (2007), *Integration of AI and OR techniques in constraint programming for combinatorial optimization problems*. Berlin, Springer.

CORE Lecture Series

C. GOURIÉROUX and A. MONFORT (1995), Simulation Based Econometric Methods.
A. RUBINSTEIN (1996), Lectures on Modeling Bounded Rationality.
J. RENEGAR (1999), A Mathematical View of Interior-Point Methods in Convex Optimization.
B.D. BERNHEIM and M.D. WHINSTON (1999), Anticompetitive Exclusion and Foreclosure Through Vertical Agreements.
D. BIENSTOCK (2001), Potential function methods for approximately solving linear programming problems: theory and practice.
R. AMIR (2002), Supermodularity and complementarity in economics.
R. WEISMANTEL (2006), Lectures on mixed nonlinear programming.