COMMUN. MATH. SCI.
Vol. 8, No. 4, pp. 909–930
© 2010 International Press

A LEVEL SET APPROACH TO MODELING GENERAL SERVICE RULES IN SUPPLY CHAINS*

CHRISTIAN RINGHOFER†

Abstract. The need for service rules, or policies, in supply chains arises if not all the parts processed in the chain are considered identical, but are distinguished by certain attributes. We develop and analyze a methodology to model arbitrary service rules in large supply chains based on a kinetic (traffic flow like) theory and a level set approach. The final result is a system of hyperbolic conservation laws for the densities of parts, grouped by their attributes. The validity of the model is verified against discrete event simulations for several test cases.

Key words. supply chains, traffic flow models, level sets, Boltzmann equation, fluid limits.

AMS subject classifications. 65N35, 65N05.

1. Introduction

The goal of this paper is to model a quite general set of scheduling policies in the framework of hyperbolic conservation law models for supply chains. Conservation law models for supply chains and supply networks use an analogy to traffic flow models and treat parts passing through a supply chains as vehicles traveling on a virtual road (the stage of the process). Thus, parts enter the chain as raw product at stage $x=0$ and leave the supply chain as finished product at stage $x=X$. Their evolution through the stages is described by a conservation law. The flux function governing this law can either be derived from heuristic considerations [1], elementary queueing theory (using so called clearing functions [8, 12]) or from microscopic models describing the individual behavior of each part (i.e., from a kinetic theory) [5, 2, 7]. In the latter case, macroscopic conservation laws are obtained by employing large time averages or moment closures to the underlying kinetic equations. If the microscopic model takes into account the random behavior of the nodes in the supply chain, the goal is to derive macroscopic equations for the means and the variances of the probability density that a given part is at stage x at time t.

This paper is concerned with the inclusion of scheduling policies in general supply chain models. The need for such policies arises when not all parts in the chain are treated as identical, but are distinguished by features such as, e.g., a product type, i.e., more than one type of product is produced at the same time, or by individual delivery dates, i.e., each part is supposed to be delivered at a certain due date, or by the time they have already spent in the process [4, 11]. The latter criterion is especially relevant if the parts in the supply chain represent a perishable good, that is, their value decreases over time. In all these cases the parts are not necessarily processed in sequence and each node in the chain decides on which parts to process first according to a certain policy or service rule [6].

The general model considered in this paper is based on a priority function. Let $x\in[0,X]$ denote the stage of the process and let $y\in\mathbb{R}^d$ denote the (vector valued) attribute of an individual part. The Newton equations for an individual part are then

*Received: January 1, 2009; accepted (in revised version): January 11, 2010. Communicated by Pierre Degond.
This work was supported by the NSF under grants DMS-0604986 and DMS-0757309.
†Department of Mathematics, Arizona State University, Tempe, AZ 85287-1804, USA (ringhofer@asu.edu).

of the form

$$\frac{dx}{dt}=v, \quad \frac{dy}{dt}=E$$

(for the examples considered in this paper, it will be convenient to let the attribute y of an individual part change continuously over time). Let $f(x,y,t)$ denote the density of parts with attribute y which are at stage x at time t. The Newton equations imply that the density function $f(x,y,t)$ evolves according to the Vlasov equation

$$\partial_t f(x,y,t)+\partial_x[vf(x,y,t)]+\nabla_y\cdot[Ef(x,y,t)]=0,$$

expressing the conservation of parts. Modeling the evolution of the supply chain under a given service rule then reduces to choosing a velocity v, depending on the state of the chain and the individual supplier. We introduce a priority function $p(x,y,t)$ which denotes the priority the node at stage x assigns to a part with attribute y at time t. The basic concept of priority scheduling is, that the velocity v of a part depends on the number of parts with a higher priority at the same node. We compute the number of parts at (x,t) with a priority higher than q as

$$\phi_f(x,t,q)=\int H(p(x,y,t)-q)f(x,y,t)\ dy, \tag{1.1}$$

where H denotes the usual Heaviside function. So, we assume that the velocity of a part with attribute y is dependent only on the number of parts with higher priority, and set $v=v(x,t,\phi(x,p(y),t))$. The kinetic model we consider in this paper is therefore of the form

$$(a)\ \partial_t f(x,y,t)+\partial_x[v_f(x,t,p(x,y,t))f(x,y,t)]+\nabla_y\cdot[E(x,t)f(x,y,t)]=0, \tag{1.2}$$

$$(b)\ v_f(x,t,q)=v(x,t,\phi_f(x,t,q)),$$

with the function ϕ_f given by (1.1).

So, modeling the flow in the supply chain under priority scheduling rules requires that we choose

- the priority function $p(x,y,t)$, modeling the scheduling policy,
- the velocity $v_f(x,t,q)$, describing the flow picture.

To give the paper a somewhat more concrete setting we will discuss several possible choices for the priority function and the velocity in the next section.

Equation (1.2) still has to be supplemented by initial and boundary conditions, which we write as

$$v_f f(x=0,y,t)=F^B(y,t), \quad f(x,y,t=0)=f^I(x,y), \tag{1.3}$$

where $F^b(y,t)$ denotes the influx of parts with attribute y and $f^I(x,y)$ denotes the initial state of the system. The kinetic problem (1.2)–(1.3) is potentially rather high dimensional. The goal of this work is therefore to find a macroscopic approximation to the transport problem (1.2)–(1.3). We do so by finding a multi-phase approximation to the kinetic density function $f(x,y,t)$ of the form

$$f(x,y,t)=\sum_{n=1}^{N}\rho_n(x,t)\delta(y-Y_n(x,t)), \tag{1.4}$$

where $\rho_n(x,t)$ denotes the macroscopic density of parts with attribute $y=Y_n$. And find an approximate solution by deriving a system of hyperbolic conservation laws for the densities ρ_n and the attributes Y_n. This paper is a generalization, and an alternative, to the work presented in [3], where only one dimensional attributes ($y\in\mathbb{R}^1$, $p(x,y,t)=y$) and a particular choice of the velocity profile v_f is considered. In [3] the equations for the macroscopic densities ρ_n and Y_n in (1.4) are obtained by computing moments of the kinetic equation, using the standard multi-phase approach [9]. This paper treats general velocity profiles v_f, which as will be seen, allows also for the study of the influence of service rules on stochastic supply chain models. Moreover, the present paper uses a reformulation of the kinetic equation (1.2) via a level set approach which allows for almost exact multi-phase solution of the form (1.4), in the sense that the only discretization error occurs when discretizing the initial and boundary conditions.

This paper is organized as follows. In section 2 we establish the general relations between the phase velocity v_f and a macroscopic flux function for the whole ensemble. In section 3 we derive the governing macroscopic conservation laws via a multiphase approach for the level sets of parts of equal priority and discuss the approximation properties of the approach. Section 4 is devoted to numerical experiments, verifying the approximation for the case of a simple FIFO policy and illustrating the method for a more complex policy. The proofs of the theorems in section 3 are given in the appendix in section A.

2. Velocities, priorities and level sets

In this section we present some choices for the velocity profile v_f and the priority function p. The purpose of this section is twofold. First it will give a somewhat more concrete background to the derivation in the following sections, and second we will choose the velocity profile v_f according to a general principle, relating the microscopic velocity v_f to the flux function for the macroscopic flow of the the densities ρ_n. This relation will be essential for the formulation of the approximation of the kinetic equation by the multi- phase solution (1.4).

2.1. Velocity profiles. We start by observing a general relation between the microscopic velocity profile v_f and the macroscopic flux function. We define the macroscopic part density $\bar\rho(x,t)$ as

$$\bar\rho(x,t)=\int f(x,y,t)\ dy.$$

Integrating the kinetic equation (1.2) over the attribute space gives the macroscopic conservation law

$$\partial_t\bar\rho(x,t)+\partial_x F(x,t)=0,\quad F(x,t)=\int v_f(x,t,p(x,y,t))f(x,y,t)\ dy=0.$$

Because of the definition (1.1) of the aggregate density ϕ_f we have that

$$\begin{aligned}\int\psi(q)\partial_q\phi(x,t,q)\ dq&=-\int\psi(q)\delta(p(x,y,t)-q)f(x,y,t)\ dydq\\&=-\int\psi(p(x,y,t))f(x,y,t)\ dy\end{aligned}$$

holds for any function $\psi(q)$. Setting $\psi(q)=v_f(x,t,q)$ gives

$$F(x,t)=-\int v_f(x,t,q)\partial_q\phi_f(x,t,q)\ dq=-\int v(x,t,\phi_f(x,t,q))\partial_q\phi_f(x,t,q)\ dq$$

$$= -V(x,t,\phi_f(x,t,q))|_{q=-\infty}^{\infty},$$

where V denotes the antiderivative of the velocity profile v with respect to the third variable. On the other hand, $\phi_f(x,t,\infty)=0$ and $\phi(x,t,-\infty)=\bar{\rho}(x,t)$ holds, giving

$$F(x,t)=V(x,t,\bar{\rho})-V(x,t,0).$$

Thus the kinetic equation (1.2) implies the macroscopic conservation law

$$\partial_t\bar{\rho}+\partial_x F=0, \quad F(x,t)=V(x,t,\bar{\rho}(x,t))-V(x,t,0), \quad \partial_z V(x,t,z)=v(x,t,z). \tag{2.1}$$

Remark 2.1. We note that the relation between the kinetic equation (1.2) and the macroscopic model (2.1) is a consequence of the fact that the velocity v_f in (1.2) solely depends on the number of parts with higher priority, i.e., on $\phi_f(x,t,p)$. This, in turn, is a consequence of the priority scheduling policy.

Of course, the solution of (2.1) will not provide any information about the impact of the service rules, i.e., the total density $\bar{\rho}$ does not contain any information about which parts have been processed first. The significance of (2.1) lies rather in the modeling aspect. We will usually assume a given macroscopic conservation law, i.e., a flux function F, and will be interested in the impact of a given policy on the flux of parts with different attributes. The relation (2.1) tells how to choose the microscopic velocity v, namely as the derivative of the macroscopic flux, with respect to the density, i.e., as the phase velocity of the macroscopic flux. For instance, in [2] a flux function of the form

$$F(x,t)=\min\{c(x),v_0(x)\bar{\rho}(x,t)\} \tag{2.2}$$

has been derived from a deterministic automaton model, where v_0 denotes the raw velocity (the inverse of the cycle time) of the individual node and $c(x)$ denotes the capacity (the maximal service rate). This corresponds to a microscopic velocity v in (1.2) of the form

$$v(x,t,\phi)=v_0(x)H(c(x)-v_0(x)\phi),$$

which is precisely the microscopic velocity used in [3] — but derived from different considerations. On the other hand, in [5] a model for the same automaton has been derived which includes the effect of random breakdows of suppliers, governed by Markov processes. This results into a macroscopic flux function F of the form

$$F(x,t)=a(x)c(x)\left[1-\exp\left(-\frac{v_0(x)\bar{\rho}(x,t)}{a(x)(c(x)}\right)\right]$$

where $a(x)$ denotes the average availability of supplier x, and $\bar{\rho}$ is now the expectation of the density. Translating this to a microscopic policy model corresponds to

$$v(x,t,\phi)=v_0(x)\exp\left(-\frac{v_0\phi}{ac}\right).$$

2.2. Attributes and policies. How to choose the attributes of individual parts, and how to choose service rules, will of course depend on the particular application. For demonstration purposes we will restrict ourselves in this paper to three attributes, namely

- y_1: the cycle time, i.e., the time elapsed since entering the system,
- y_2: the time to a due date, i.e., the time the part is supposed to be delivered at a certain date and y_2 measures the time left to this date,
- y_3: the type of the part.

Since the type of the part will usually be integer valued, and will not change over time the kinetic equation (1.2) will have a solution of the form

$$f(x,y,t)=\sum_{k=1}^{K} f_k(x,y_1,y_2,t)\delta(y_3-k).$$

This is, however, merely a notational detail. The attributes will change over time according to

$$\frac{dy}{dt}=E=\begin{pmatrix}1\\-1\\0\end{pmatrix},$$

i.e., the cycle time will grow linearly and the time to due date will decay linearly with time. The influx $F^B(y,t)$ will be given by

$$F^B(y,t)=\sum_{k=1}^{K} d_k(t,y_2)\delta(y_1)\delta(y_3-k)$$

with $d_k(t,y_2)$ the distribution of the time to due date at entry of parts of type k. (All parts will have zero cycle time y_1 upon arrival.)

If we order the parts strictly according to type, then $p(x,y,t)=y_3$ would hold, i.e., parts of type K have the highest priority and will always be served first, regardless of the other attributes. If we employ a FIFO (first in first out) policy then the parts will be ordered according to their cycle time and $p(x,y,t)=y_1$ will hold. In the same way, if we only care about on time delivery $p(x,y,t)=-y_2$ would be chosen. The model allows however for more sophisticated policy choices. Assume that we are dealing with perishable goods, then we might choose a priority of the form

$$p(x,y,t)=-y_2H(T(y_3)-y_1)+y_1H(y_1-T(y_3)).$$

That is we schedule according to the delivery date up to a certain time $T(y_3)$ (dependent on the type) when the item gets too close to perishing. In this case we switch to a FIFO policy to minimize cycle time from here onward.

Note, that if the priority function p does not depend on a certain component of the attribute vector y, then this component can be integrated out of the whole kinetic model (1.2), reducing the dimension of the problem. We might however choose not to do so, in order to observe this component anyway. In this case the corresponding attribute becomes just additional information carried along by the flow, without influencing it.

3. Multi-phase approximations

We now turn to deriving a multi-phase approximation for the initial boundary value problem (1.2)–(1.3). So we approximate the kinetic density $f(x,y,t)$ by a superposition of Dirac delta functions in attribute space of the form (1.4) and the goal is to derive a system of partial differential equations for the evolution of the densities

$\rho_n(x,t)$ and the attributes $Y_n(x,t)$. The standard approach [9] would be to compute moments of the kinetic equation (1.2) in the attribute dimensions and close the moment equations by the ansatz (1.4). This is essentially the methodology employed in [3]. We will follow a different route in this paper, which essentially gives an exact solution of the kinetic equation (1.2). So, the only approximation error arises from approximating the initial and boundary conditions in a way that is compatible with (1.4). The principal problem we have to overcome, is that the kinetic equation (1.2) is not well defined for multi-phase solutions of the form (1.4). Assuming a solution of the form (1.4), the cumulative density ϕ_f in (1.1) becomes

$$\phi_f(x,t,q) = \sum_{n=1}^{N} \rho_n(x,t) H(p(x,Y_n(x,t),t) - q),$$

and the corresponding velocity is given by

$$v_f(x,t,q) = v(x,t,\sum_{n=1}^{N} \rho_n(x,t) H(p(x,Y_n(x,t),t) - q).$$

Evaluating the velocity v_f at $q = p(x,Y_m(x,t),t)$, which is necessary to evolve the multi-phase solution (1.4), implies that the $\phi_f(x,t,q)$ has to be evaluated precisely at the values of q where ϕ_f is discontinuous. The key to deriving an exact multi-phase approximation is to formulate the kinetic equation (1.2) appropriately. This is achieved by deriving an appropriate formulation of the equation for the level sets of parts with equal priority.

3.1. Level set evolution. We will address the problem of evaluating the velocity v_f at the points where ϕ_f is discontinuous by reformulating the problem as a shock problem for the hyperbolic conservation law governing the evolution of the level set function of all parts with equal priority. This is, in a sense, the reverse approach to the general use of level set functions, where the higher dimensional equation is used to deal with the nonlinearity in the lower dimensional equation [10, 13]. We define the level set function $\Lambda(x,t,q)$ by

$$\Lambda(x,t,q) = \int \delta(p(x,y,t) - q) f(x,y,t)\ dy. \tag{3.1}$$

Accordingly, we factor the kinetic density $f(x,y,t)$ into

$$f(x,y,t) = \Lambda(x,t,p(x,y,t)) g(x,y,t),$$

where the function $g(x,y,t)$ denotes the distribution of attributes within the level set of equal priorities. So,

$$\int \delta(p(x,y,t) - q) g(x,y,t)\ dy = 1$$

holds for all values of x,q,t.
The evolution equation for Λ is given by the following

LEMMA 3.1. *The level set function $\Lambda(x,t,q)$, defined by (3.1), satisfies the initial boundary value problem*

$$(a)\ \partial_t \Lambda(x,t,q) + \partial_x[v_f(x,t,q)\Lambda] + \partial_q[A_f(x,t,q)] = 0, \tag{3.2}$$

$$(b)\ A_f(x,t,q) = \int \delta(p(x,y,t)-q) f(x,y,t)[\partial_t p + v_f(x,t,q)\partial_x p + E(x,t)\nabla_y p]\ dy,$$

with initial and boundary conditions given by

$$(c)\ \Lambda(x,0,q) = \Lambda^I(x,q) = \int \delta(p(x,y,0)-q) f^I(x,y)\ dy,$$

$$(d)\ v_f(0,t,q)\Lambda(0,t,q) = F^B_\Lambda(t,q) = \int \delta(p(0,y,t)-q) F^B(y,t)\ dy.$$

The proof of Lemma 3.1 is deferred to the appendix in section A.

Note that if the priority p and the attribute y does not vary in space and time, i.e., for $p=p(y)$ and $E=0$, we obtain a closed initial boundary value problem for the level set function Λ.

A multi-phase approximation of the form (1.4) for the kinetic density f would correspond to an ansatz of the form

$$\Lambda(x,t,q) = \sum_{n=1}^{N} \rho_n \delta(p(Y_n)-q)$$

for the level set function Λ. Note that we still face the same problem as before, since the cumulative density ϕ_f would have to be evaluated at the points $q=p(x,Y_n,t)$ where it is discontinuous. Also note that because of the definition (3.1) the level set function Λ is just the derivative of the cumulative density ϕ_f with respect to the variable q. So, because of the definition of the velocity profile v as the phase velocity of the macroscopic flow, we have the relations

$$\Lambda(x,t,q) = -\partial_q \phi_f(x,t,q), \quad v(x,t,\phi) = \partial_\phi F(x,t,\phi).$$

This means that the term $v_f(x,t,\phi_f(x,q,t))\Lambda(x,t,q)$ can be written as a derivative, namely as $-\partial_q F(x,t,\phi_f(x,t,q))$. This means we can integrate equation (3.2)(a) once with respect to the variable q, yielding the following

Corollary 3.2. *A solution of the level set equation (3.2)(a) can be obtained by solving*

$$\partial_t \phi_f(x,t,q) + \partial_x[F(x,t,\phi_f(x,t,q))] - A_f(x,t,q) = 0, \quad \partial_z F(x,t,z) = v(x,t,z) \tag{3.3}$$

and defining Λ *as* $\Lambda = -\partial_q \phi_f$.

Remark 3.1. Corollary 3.2 establishes the link between the model presented in [3] and the macroscopic model derived in [2]. For the special case $p(x,y,t)=y$ the macroscopic model in [2], which uses the 'min' function as flux, is obtained by integrating the Heaviside function in the kinetic model [3] with respect to y.

The advantage of the form (3.3) of the level set equation (3.2)(a) in the context of multi-phase approximations is the following Using the multi-phase ansatz (1.4), $\phi_f(x,t,q)$ will be a piecewise constant function of the priority q. In this formulation, the velocity of the shocks (the discontinuities of ϕ_f) will be determined by a Rankine-Hugoniot condition in terms of the size of the shocks and the size of the jump discontinuities in the flux function F. In order to define the value $v_f(x,t,p(x,y,t))$ at the

jump discontinuity, we construct a distributional solution of the problem (3.2) of the form

$$\Lambda(x,t,q)=\tilde{\Lambda}(x,t,q)+a(x,t)\delta(P(x,t)-q) \tag{3.4}$$

where $\tilde{\Lambda}$ is a smooth function, $q=P(x,t)$ is the location of the discontinuity of ϕ_f, and $a(x,t)$ is the size of the jump in ϕ_f. We formulate this in terms of

THEOREM 3.3. *A function $\Lambda(x,t,q)$ of the form (3.4) is a weak solution of the equation (3.2)(a) if the velocity $v_f(x,t,q)$ is defined as*

$$v_f(x,t,q)=\lim_{\varepsilon\to 0+}\frac{F(x,t,\phi_f(x,t,q+\varepsilon))-F(x,t,\phi_f(x,t,q-\varepsilon))}{\phi_f(x,t,q+\varepsilon)-\phi_f(x,t,q-\varepsilon)}. \tag{3.5}$$

The proof of Theorem 3.3 is deferred to appendix A.

As a consequence of Theorem 3.3, we replace the definition of $v_f(x,t,q)$ by the definition (3.5) and solve the kinetic equation

$$(a)\ \partial_t f(x,y,t)+\partial_x[v_f(x,t,p(x,y,t))f]+\nabla_y\cdot[Ef]=0 \tag{3.6}$$

$$(b)\ v_f(x,t,q)=\lim_{\varepsilon\to 0+}\frac{F(x,t,\phi_f(x,t,q+\varepsilon))-F(x,t,\phi_f(x,t,q-\varepsilon))}{\phi_f(x,t,q+\varepsilon)-\phi_f(x,t,q-\varepsilon)}$$

$$(c)\ \phi_f(x,t,q)=\int H(p(x,y,t)-q)f(x,y,t)\ dy.$$

We approximate the solution of the problem (3.6) by a multi-phase ansatz of the form

$$f(x,y,t)=\sum_n \rho_n\delta(Y_n-y),$$

which results in the cumulative density $\phi_f(x,t,q)$ being of the form

$$\phi_f(x,t,q)=\sum_n \rho_n H(P_n(x,t)-q),\quad P_n(x,t)=p(x,Y_n(x,t),t).$$

A rather straight forward calculation yields that an *exact* solution of equation (3.6) is obtained by evolving the amplitudes $\rho_n(x,t)$ and the attribute vectors $Y_n(x,t)$ are evolved according to the equations

$$(a)\ \partial_t\rho_n+\partial_x[w_n\rho_n]=0,\quad \partial_t Y_n+w_n\partial_x Y_n-E=0, \tag{3.7}$$

where the velocities $w_n(x,t)$ are computed as

$$(b)\ w_n(x,t)=v_f(x,t,P_n(x,t)),\quad P_n(x,t)=p(x,Y_n(x,t),t)$$

and v_f is evaluated via (3.6)(b). Of course, in order to obtain a multi-phase solution of equation (3.6), the initial and boundary conditions have to be formulated compatibly with with the multi-phase ansatz. Therefore, we have to replace the initial and boundary conditions (1.3) by conditions of the form

$$v_f f(x=0,y,t)=\sum_{n=1}^N F_n^B(t)\delta(Y_n^B(t)-y),\quad f(x,y,t=0)=\sum_{n=1}^N \rho^I(x)\delta(Y^I(x)-y). \tag{3.8}$$

which leads to the appropriate initial and boundary conditions for the conservation laws in (3.7). We reiterate that the multi-phase ansatz gives an exact solution of equation (3.6). So the only approximation arises by approximating the initial and boundary conditions in (3.8). How many terms to use in the expansion of $f(x,y,t)$ becomes therefore solely a question of how well the initial and boundary functions can be approximated by the a multi-phase expansion of the form (3.8).

3.2. Computation of the phase velocity. We conclude this section by elaborating on the actual computation of the velocities $w_n(x,t)$ in (3.7), since they are given formally only in terms of the limit in (3.6)(b). It turns out that (3.6)(b) uniquely determines the velocities even in the case of crossing priorities, i.e., if not all of the values $P_n(x,t)=p(x,Y_n(x,t),t)$ are distinct for a given point in space and time. The basic principle is that in the multi-phase approximation the function $\phi_f(x,t,q)$ is piecewise constant in the variable q with discontinuities at $q=P_n$. Therefore, the function $F(x,t,\phi_f(x,t,q))$ is piecewise constant as well with discontinuities at the same points $q=P_n$, and the difference quotient in (3.6)(b) becomes constant for ε sufficiently small. This gives rise to the following

Algorithm:

- For a fixed x,t, given Y_n, compute $P_n(x,t)=p(x,Y_n(x,t),t)$.
- Define a permutation $n\to\sigma(n)$, such that the permuted priorities are ordered, i.e.,
$$P_{\sigma(1)}\geq P_{\sigma(2)}\geq\ldots\geq P_{\sigma(N)}$$
holds.
- Assume that we have α_k of the values of P_n which coincide for a fixed (x,t). So we have
$$P_{\sigma(1)}=\cdots=P_{\sigma(\alpha_1)}>P_{\sigma(\alpha_1+1)}=\cdots=P_{\sigma(\alpha_2)}>P_{\sigma(\alpha_2+1)}\cdots$$
$$>P_{\sigma(\alpha_{K-1}+1)}=\cdots=P_{\sigma(\alpha_K)}.$$
- Since the priority values $P_{\sigma(\alpha_{k-1}+1)}=\cdots=P_{\sigma(\alpha_k)}$ all coincide, the values $w(\sigma(j))=v_f(x,P_{\sigma(j)},t)$ have to be all equal for $j=\alpha_{k-1}+1,\ldots,\alpha_k$.
- The cumulative density $\phi_f(x,t,P_n\pm\varepsilon)$ is now given by
$$\phi_f(x,t,P_{\sigma(n)}\pm\varepsilon)=\sum_{k=1}^{K}H(P_{\sigma(\alpha_k)}-P_{\sigma(n)}\mp\varepsilon)\gamma_k,\quad \gamma_k=\sum_{m=\alpha_{k-1}+1}^{\alpha_k}\rho_{\sigma(m)}$$
or
$$\phi_f(x,t,P_{\sigma(\alpha_j)}+\varepsilon)=\sum_{k=1}^{j-1}\gamma_k,\quad \phi_f(x,t,P_{\sigma(\alpha_j)}-\varepsilon)=\sum_{k=1}^{j}\gamma_k$$
$$\phi_f(x,t,P_{\sigma(\alpha_j)}+\varepsilon)-\phi_f(x,t,P_{\sigma(\alpha_j)}-\varepsilon)=-\gamma_j.$$
- At the same time, we have
$$F(x,t,\phi_f(x,t,P_{\sigma(\alpha_j)}+\varepsilon))-F(x,t,\phi_f(x,t,P_{\sigma(\alpha_j)}-\varepsilon))$$
$$=F(x,t,\sum_{k=1}^{j-1}\gamma_k)-F(x,t,\sum_{k=1}^{j}\gamma_k)$$

and because of (3.6)(b)

$$w_{\sigma(\alpha_j)} = \frac{1}{\gamma_j}[F(x,t,\sum_{k=1}^{j}\gamma_k) - F(x,t,\sum_{k=1}^{j-1}\gamma_k)] = w_{\sigma(\alpha_j-1)} = \cdots = w_{\sigma(\alpha_{j-1}+1)}. \tag{3.9}$$

- This determines the values $w_{\sigma(n)}$, $n=1,\ldots,N$ uniquely, and w_n is then obtained by inverting the permutation σ.

The above algorithm expresses the following simple strategy. After reordering the attributes according to their priority with the permutation σ, and grouping them together according to equal priorities, the highest priority $P_{\sigma(\alpha_1)}$ moves with the group velocity

$$w_{\sigma(\alpha_1)} = \frac{1}{\gamma_1}F(x,t,\gamma_1) = \frac{F(x,t,\sum_{m=1}^{\alpha_1}\rho_{\sigma(m)})}{\sum_{m=1}^{\alpha_1}\rho_{\sigma(m)}}$$

of the total flow. Subsequent lower priorities receive whatever is left of the flux of the higher priorities. Note, that the total flux over all priorities satisfies

$$\sum_{n=1}^{N} w_n\rho_n = \sum_{n=1}^{N} w_{\sigma(n)}\rho_{\sigma(n)} = \sum_{j=1}^{K} w_{\sigma(\alpha_j)}\gamma_j = F(x,t,\sum_{k=1}^{K}\gamma_k) = F(x,t,\sum_{n=1}^{N}\rho_n)$$

because of the telescopic sum in (3.9). Therefore, after deciding on a global flux function $F(x,t,\phi)$, and using the the phase velocity $v_f(x,q,t) = \partial_\phi F(x,t,\phi_f(q))$ for the kinetic model, the multi-phase approximation reproduces exactly the correct macroscopic flux for the total density. This total flux is distributed onto parts of different priorities according to the above algorithm, and the evolution of the attribute vectors $Y_n(x,t)$ rearranges the ordering of the priorities as the system evolves.

4. Numerical experiments

In this section we perform some numerical experiments on the multi-phase approximation from section 3. We verify the model on a simple test case against a discrete event simulation in the deterministic as well as the stochastic case, including random breakdown of individual nodes. Sections 4.1 and 4.2 deal with a straight FIFO (First In First Out) policy and section 4.3 uses a more complex mixed policy. Since we compare the FIFO results directly to discrete event simulation results, it is necessary to precisely define the type of discrete event simulation used. The simple discrete event algorithm used in this paper is given in section 4.2.

4.1. The basic setup. We restrict ourselves essentially to the three dimensional attribute vector of section 2.2. So, $y=(y_1,y_2,y_3)$ holds, with y_1 the elapsed cycle time, y_2 the time to a given due date, and y_3 the (integer valued) type of the part. We will use two types. So we have $y_3 \in \{1,2\}$.

Before we proceed, we have to make a slight modification to this setup. The need for this modification arises from the fact that the phase velocity $v(x,t,\phi) = v_0(x)H(c(x)-v_0(x)\phi)$, corresponding to the flux $F(x,t,\bar\rho) = \min\{c(x),v_0(x)\bar\rho\}$, in section 2.1 (2.2) is discontinuous. This gives rise to the following problem: since by design all parts with the same priority $p(x,y,t)$ move with the same velocity $v_f(x,t,p)$, all parts with priority p will be held back if the corresponding flux would exceed the capacity $c(x)$, while, in reality, a certain number of them would still be processed, up to the point where the capacity is exceeded. We point out that this is a modeling issue

of the kinetic model (1.2), rather than one of the theory in section 3, and arises only if the level sets of equal priority have a nonzero measure in the space of attributes. This problem, in turn, arises only in the case where the macroscopic flux function F in section 2.1 is not continuously differentiable with respect to the density ρ, as is the case for the min function in (2.2).

We remedy this problem by introducing an additional — artificial — attribute y_4, which will remain constant over space and time, and prioritizes the parts slightly within the same level set. So, we modify the priority function $p(x,y_1,y_2,y_3,t)$ to

$$\tilde{p}(x,y_1,y_2,y_3,y_4,t)=p(x,y_1,y_2,y_3,t)+\varepsilon y_4. \tag{4.1}$$

This means we spread parts of equal priority p over the additional dimension y_4, and the correct portion of the parts would still be processed, or, in other words, if the level set has a nonzero measure in (y_1,y_2,y_3) space, it still has zero measure in the enlarged (y_1,y_2,y_3,y_4) space. Parts with different priority p will still have different priority $\tilde{p}$, if the parameter ε is chosen small enough. Since y_4 is an artificial attribute, it might as well be chosen uniformly distributed in the interval $[0,1]$. According to (3.7), this gives the equations

$$\partial_t\rho_n+\partial_x[w_n\rho_n]=0, \quad \partial_t Y_n+w_n\partial_x Y_n-E=0, \quad E=\begin{pmatrix}1\\-1\\0\\0\end{pmatrix}. \tag{4.2}$$

The velocities w_n have to be computed according to the algorithm in section 3.2, with the priority function p replaced by $\tilde{p}$ from (4.1). From the results in section 3, we know that the approximation of the kinetic solution (1.2) by the multi-phase solution (4.1) is given solely in terms of the approximation quality of the boundary conditions. Thus, we have to approximate the boundary flux density

$$v_f f(0,y,t)=\sum_{k=1}^{2}\lambda(y_3,t)\delta(y_1)\delta(y_2-d(y_3))\delta(y_3-k)\chi_{[0,1]}(y_4) \tag{4.3}$$

by the appropriate flux density for the multi-phase solution in (3.7). The boundary flux (4.3) states that particles of type $y_3=k$, $k=1,2$ enter at a rate $\lambda(y_3,t)$. The elapsed cycle time y_1 at entry is zero, the time to the due date at entry is given by the functions $d(y_3)$, dependent on the type, and the density is spread uniformly in the artificial attribute y_4. The kinetic influx (4.3) is, according to (3.8), approximated by a multi-phase influx of the form

$$v_f f(0,y,t)\approx\sum_{n=1}^{N}F_n^b(t)\delta(Y_n^b(t)-y). \tag{4.4}$$

We do so by replacing the indicator function χ in (4.3) by the approximation

$$\frac{1}{M}\sum_{m=1}^{M}\delta\left(y_4-\frac{m}{M}\right),$$

and the index n by the double index $n=(k,m)$, $k=1,2$, $m=1,\ldots,M$, $N=2M$. Setting F_{km}^b and Y_{km}^b in (4.4) to

$$v_f f(0,y,t)\approx\sum_{k=1}^{2}\sum_{m=1}^{M}F_{km}^b(t)\delta(Y_{km}^b(t)-y), \quad F_{km}^b=\frac{1}{M}\lambda(k,t), \tag{4.5}$$

$$Y^b_{km}(t) = \left(0, d(k), k, \frac{m}{M}\right)$$

gives (4.3), with the indicator function replaced by the superposition of $\delta-$ measures. So, essentially we have replaced one particle with weight $\lambda(y_3,t)$ entering the system, by M particles with weights $\frac{\lambda(y_3,t)}{M}$. We note that the last two attributes (y_3,y_4) are just indices, i.e., the equation (4.2) with the boundary conditions (4.5) has the obvious exact solution $Y^{(3)}_{km}(x,t) = k$ and $Y^{(4)}_{km}(x,t) = \frac{m}{M}$, and the last two components of the attribute vector $Y(x,t)$ do not really have to be solved for. So, altogether, we have to solve $3N = 6M$ one dimensional hyperbolic equations for the densities ρ_{km} and the first two components of the attribute vector $Y_{km},\ k=1,2,\ m=1,\ldots,M$. In order to employ standard numerical methods for hyperbolic conservation laws, we rewrite the system (4.2) in conservative form as

$$\partial_t \rho_n + \partial_x[w_n \rho_n] = 0, \quad \partial_t(\rho_n Y_n) + \partial_x[w_n \rho_n Y_n] - \rho_n E = 0. \tag{4.6}$$

The boundary conditions now take the form that the fluxes $w_{km}\rho_{km}(0,t) = F^b_{km}(t)$ and $w_{km}\rho_{km}Y_{km}(0,t) = F^b_{km}Y^b_{km}(t)$ have to be prescribed at influx at $x=0$. The system (4.6) is solved by a standard Lax-Wendroff scheme, and the velocities w_n are computed via the algorithm in section 3.2.

4.2. Verification for a FIFO policy. We first verify the the multi-phase approximation against a discrete event simulation, i.e., a stochastic automaton. In previous work [2, 5] it has been shown that, in an appropriate scaling limit, the solution of the automaton converges to the solution of a conservation law with the flux functions described in section 2.1. In the case of a pure FIFO policy it is possible to modify the automaton, such that it also simulates the scheduling rule. In the following we briefly describe the precise workings of the stochastic automaton. We consider a system of $M >> 1$ processors which we locate in the interval $x \in [0,1]$. So processor m occupies the cell $[(m-1)\Delta x, m\Delta x),\ m=1,\ldots,M,\ \Delta x = \frac{1}{M}$. Each processor has an individual processing time $T(x) = \frac{\Delta x}{v_0(x)}$ and a capacity $c(x) = \frac{p(x)}{T(x)}$, i.e., it can handle up to $p(x)$ parts at the same time. Each processor breaks down and is repaired according to a Markov process, that is the time intervals between breakdowns and repairs are exponentially distributed. The basic variables involved in the automaton are the following:

- Let $\tau(x,r)$ denote the time part number r arrives at the queue in front of processor number x.
- Let $s(r)$ denote the type of part number r. We consider a system with two types of parts, i.e., $s(r) \in \{1,2\}$.
- Let $\alpha(x,r)$ denote the time part number r arrives at the top of the queue in front of processor x. So, at $t = \alpha(x,r)$ the part could be fed into the processor, if it were running.
- Let $\omega(x,\alpha)$ denote the time it has to wait if the processor is down at time $t = \alpha$. So, $\beta(x,r) = \alpha(x,r) + \omega(\alpha(x,r))$ denotes the time it is actually fed into the processor, and $\tau(x+\Delta x, r) = \beta(x,r) + \frac{\Delta x}{v_0(x)}$, denotes the time it leaves processor x and arrives at the queue in front of the next processor in the chain.

The automaton works according to the following rules:

- $\alpha(x,r) \geq \tau(x,r)$: The part has to have arrived before it can be fed into the processor.

- $\alpha(x,r) \geq \beta(x,r-\Delta r) + \frac{T(x)}{p(x)}$: The processor can only accept parts in time intervals of length $\frac{T(x)}{p(x)}$.

This yields the recursion

$$(a)\ \alpha(x,r) = \max\{\tau(x,r), \beta(x,r-\Delta r) + \frac{T(x)}{p(x)}\}, \quad \beta(x,r) = \alpha(x,r) + \omega(x,\alpha(x,r)), \tag{4.7}$$

$$(b)\ \tau(x+\Delta x,r) = \beta(x,r) + \frac{\Delta x}{v_0(x)}.$$

The waiting time $\omega(x,\alpha)$ is computed as

$$\omega(x,\alpha) =$$

$$\begin{pmatrix} 0, & \text{if processor } x \text{ is up at time } \alpha \\ t_{next} - \alpha, & \text{if processor } x \text{ is down at } \alpha \text{ with } t_{next} \text{ the next time it is up again} \end{pmatrix}$$

In the case of a deterministic automaton, the processors are up all the time, and $\omega(x,\alpha) = 0$ will always hold. In this case α and β can be eliminated via $\beta(x,r) = \alpha(x,r) = \tau(x+\Delta x,r) - \frac{\Delta x}{v_0(x)}$, yielding the simple deterministic recursion

$$\tau(x+\Delta x,r) = \max\left\{\tau(x,r) + \frac{\Delta x}{v_0(x)}, \tau(x+\Delta x, r-\Delta r) + \frac{T(x)}{p(x)}\right\}, \tag{4.8}$$

It has been shown in previous work [2] [5], that, in an appropriate scaling limit, the solution of the recursion (4.7) converges to the solution of the conservation law

$$\partial_t \rho + \partial_x F = 0$$

for $\Delta x \to 0$, $\Delta r \to 0$. In the deterministic case (no breakdowns), corresponding to the recursion (4.8), the flux function is of the form $F(x,\rho) = v_0(x)\min\{p(x),\rho(x)\}$. In the case of exponentially distributed up and down times with means $\langle T_{up}\rangle$ and $\langle T_{down}\rangle$, the flux function is given by $F(x,\rho) = a(x)p(x)v_0(x)[1-\exp(\frac{\rho}{p})]$, with $a(x)$ the average availability, i.e., $a(x) = \frac{\langle T_{up}\rangle}{\langle T_{up}\rangle + \langle T_{down}\rangle}$.

The density $\rho(x,t)$ and the flux F are related to the arrival times τ via the relations

$$\rho(x,t) = \int \delta(t-\tau(x,r))\partial_x \tau(x,r)\ dr, \quad F(x,t) = \int \delta(t-\tau(x,r))\ dr. \tag{4.9}$$

The recursion (4.7) can be used directly to simulate a FIFO policy. The significance of a FIFO (First in First Out) policy is that the parts remain ordered in the sequence of their arrival, as is guaranteed by (4.7). (We have $\beta(x,r) \geq \alpha(x,r) > \beta(x,r-\Delta r)$, and the same relation hods for $\tau(x,r)$.) Therefore we can use the recursion (4.7) to compute partial densities and fluxes belonging to each species by modifying (4.9). So, for each individual species, we first compute the arrival times in the system according to a given influx density. Then we sort the parts according to their arrival time and run the automaton given by the formulas (4.7). At the end of the discrete event simulation, we simply invert the sort and compute the density and flux for each species according to

$$(a)\ \rho_k(x,t) = \int \delta(t-\tau(x,r))\partial_x \tau(x,r)\delta(s(r)-k)\ dr, \tag{4.10}$$

$$(b)\ F_k(x,t) = \int \delta(t-\tau(x,r))\delta(s(r)-k)\ dr$$

for $k=1,2$.

The system:
We consider a chain of forty processors. Each processor has a throughput time $\frac{\Delta x}{v_0}=1$, except for processors $11-20$, which have a throughput time $\frac{\Delta x}{v_0}=2$. Each processor can handle $p(x)=30$ parts at the same time. This yields a bottleneck capacity of $c(x)=15\frac{parts}{time}$ for processors $11-20$ and a capacity $c(x)=30\frac{parts}{time}$ for the rest. Figure 4.1 shows the influx for both species. So, the total influx (the sum of both curves in figure 4.1) exceeds the bottleneck capacity in the interval $40<t<80$.

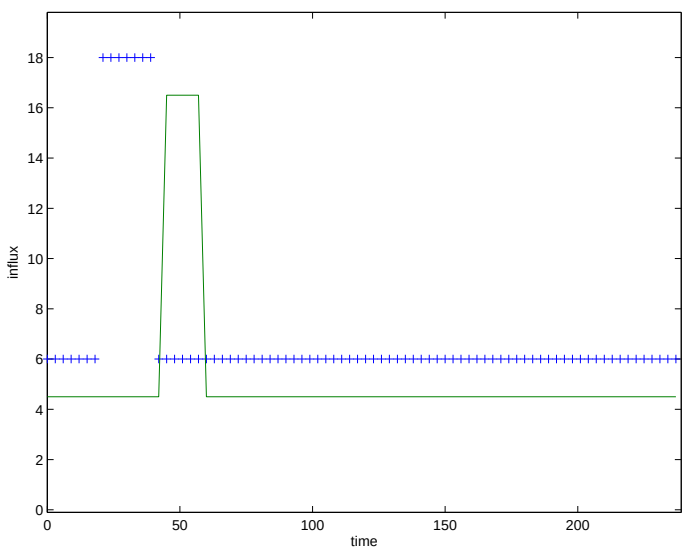

FIG. 4.1. *Influx, +=species 1, solid=species 2,*

The deterministic case:
We first test the deterministic case. So, we set the waiting times $\omega(x,\alpha)$ in (4.7) to zero and solve only the recursion (4.8), computing densities and fluxes, according to (4.10). We compare this to the multi-phase solution, using the deterministic flux function $F(x,\rho)=v_0(x)\min\{p(x),\rho\}$. Figure 4.2 shows the quantity $\Delta x\rho$, i.e., the number of parts corresponding to each processor (those in the queue and the processor itself), for one of the species in a contour plot.

To give a more quantitative comparison, we plot the density over time at various stages. Figure 4.3 shows the densities of the two species in processor 5 (before the bottleneck), processor 11 (the first bottleneck stage), and stage 21 (after the bottleneck). The right panel shows the densities for the sum of both species.

The random case:
Next we test the multi-phase solution on the stochastic system. So, we use the recursion (4.7) where we generate the up and down times from exponential distributions with means $\langle T_{up}\rangle(x)=3T(x)$ and $\langle T_{down}\rangle(x)=T(x)$. So each processor runs on average for three cycle times, and then shuts down for the next cycle time, giving an availability of $a=0.75$. We compute 300 realizations of the automaton (4.7), compute

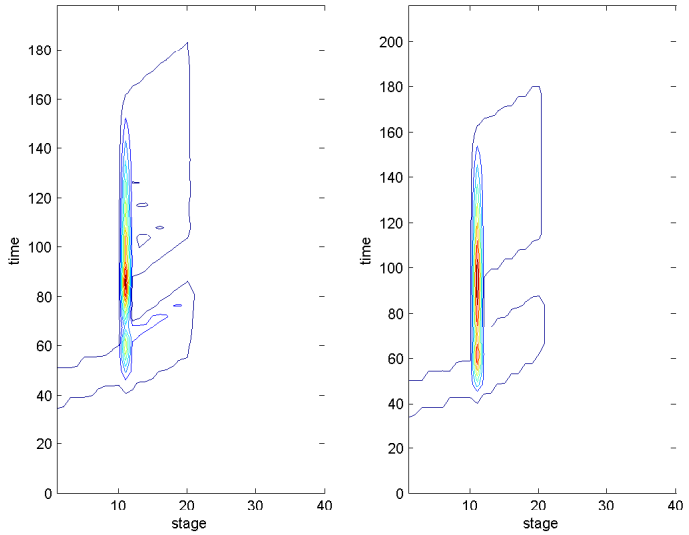

FIG. 4.2. *Parts per processor comparison (contour plot) between the deterministic automaton and the deterministic conservation law for species 2. Left panel: automaton. Right panel: the multiphase solution*

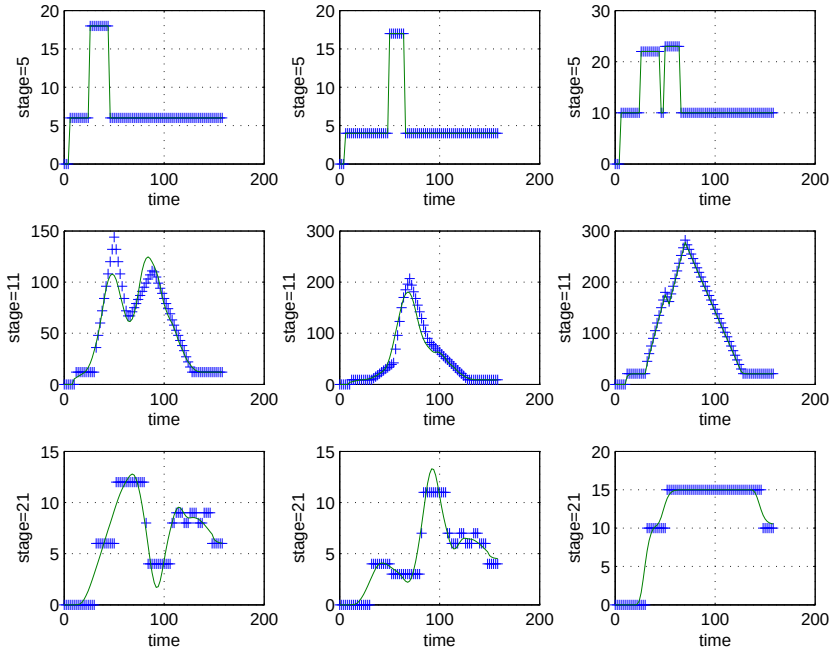

FIG. 4.3. *Parts per processor comparison between the deterministic automaton and the deterministic conservation law for stages 5,11 and 21. Left panel: species 1. Middle panel: species 2. Right panel: both species. solid line=multiphase solution. += automaton.*

the means, and compare this to the mean field multi-phase solution, using the flux function $F(x,\rho) = a(x)p(x)v_0(x)[1-\exp(\frac{\rho}{p})]$. As in the deterministic case, we compare the quantity $\Delta x\rho$, for one of the species in a contour plot in figure 4.4. To give a

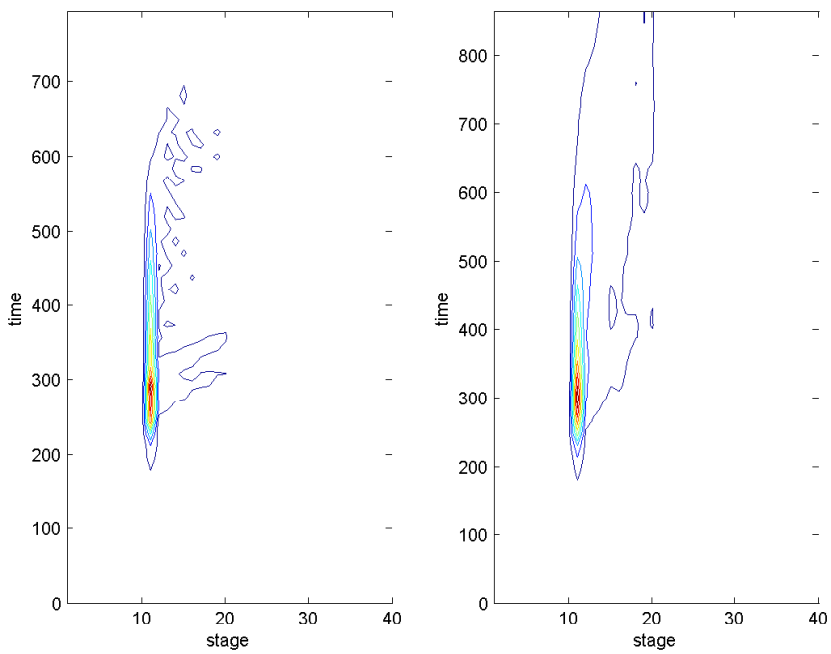

FIG. 4.4. *Parts per processor comparison (contour plot) between the average over 300 realizations of the stochastic automaton and the mean field conservation law for species 2. Left panel: the mean field multiphase solution. Right panel: average over 300 automaton realizations.*

more quantitative comparison again, we plot the density over time at stages $5, 11, 21$ in figure 4.5.

REMARK 4.1. At first glance, it is surprising that the present approach is able to accurately simulate the effects of the FIFO policy on the *expectation* of the density, since we apparently have interchanged the evaluation of the expectation operator with the evolution of a nonlinear stochastic dynamical system. However, as shown in [5], we are using the correct conservation law for the expectation of the density $\bar{\rho}$ of the whole ensemble, and the process of re-ordering the parts according to the elapsed cycle time can apparently be commuted with the expectation operator. Note that the agreement in figure 4.5 is not perfect. In particular, there is some disagreement between the automaton and the conservation law solution at the higher stages. The deviation occurs actually in the total part density (in the right panel of figure 4.5), and is therefore due to the imperfection of the exponential flux function. The exponential flux function in [5] is derived from a molecular chaos assumption on the individual parts. We conjecture that this assumption is violated temporarily in the rather complex situation simulated here, leading to the discrepancy in the total density, and consequently to the discrepancy in the density of the individual species.

4.3. A more complex policy. We conclude the numerical experiments with a numerical study of the influence of different policies. We consider the same system

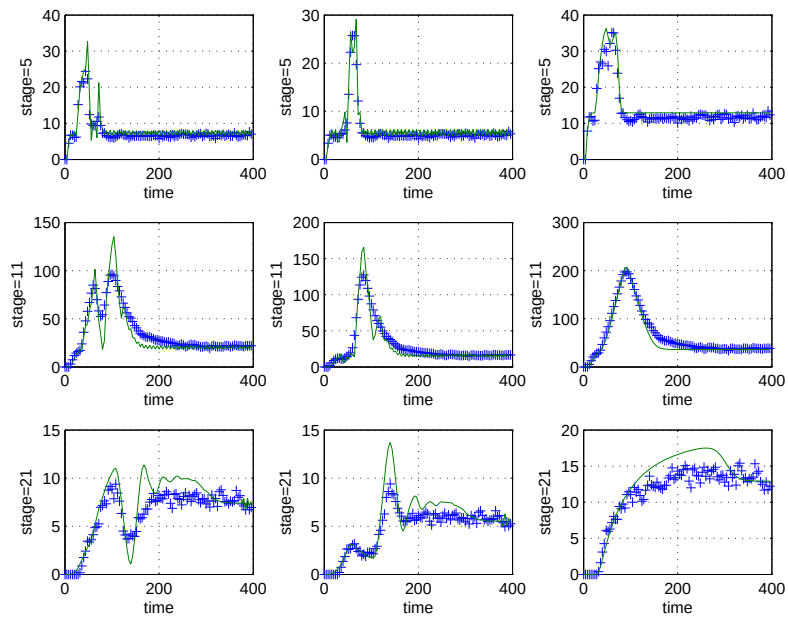

FIG. 4.5. *Parts per processor comparison between 300 realizations of the stochastic automaton and the mean field conservation law for stages 5,11 and 21. Left panel: species 1. Middle panel: species 2. Right panel: both species. solid line=mean field multiphase solution. += automaton expectation.*

as in section 4.2. Each species now has a 'due date', i.e., a certain limit on the cycle time, after which it is delivered late. This models essentially the production of a perishable good which is spoiled and worthless after spending too much time in the system. We compare the FIFO policy from section 4.2, where the priority is set to $p(y)=y_1+\varepsilon y_4$, to a more complex policy with a priority function of the form

$$p(y)=y_1H\left(y_2-\frac{1}{2}d(y_3)\right)-y_2H\left(\frac{1}{2}d(y_3)-y_2\right)+\varepsilon y_4. \tag{4.11}$$

The policy given by the priority function (4.11) can be interpreted in the following way:

- We schedule according to FIFO, using y_1, until the time to due date y_2 has reached half its limiting value d, where d is dependent on the species y_3. From this point on we switch policies, prioritizing the parts according to the time until the part becomes worthless.
- We choose as the as maximal acceptable cycle times $d(1)=75,\ d(2)=200$. So species one 'spoils' after 150% of the raw throughput time, while species number two can spend 400% of the raw throughput time in the system before spoiling.

The left panel in figure 4.6 shows result for the FIFO policy, and the right panel for the policy corresponding to the priority function (4.11). The top row shows the cycle time $Y_k^{(1)}(x=40,t),\ k=1,2$ at exit, i.e., the total time parts have spent in the system. The middle row shows the cycle time $Y_k^{(2)}(x=40,t),\ k=1,2$ at exit, i.e., the time to due date for each system at the exit.

- The cycle times at exit for FIFO (top, left panel in figure 4.6) are identical for both species, as they should be, using a FIFO policy.
- Using FIFO, there is a significant amount of spoiled parts of species 1 at the exit. That is the curve in the middle left panel of figure 4.6 for species 1 dips significantly below 0. Using the policy given by (4.11), essentially all parts of both species can be delivered on time, as seen in the middle right panel of figure 4.6.
- Note, that the attributes vanish for certain periods of time. This is an artifact of the conservative discretization (4.6)(b) of the attribute equations. The primary variable used in the code is $\rho_n Y_n$ and the attributes in the top two panels of figure 4.6 are computed as $\frac{\rho_n Y_n}{\rho_n}$. So, if there are no parts ($\rho_n = 0$), the attribute Y_n is meaningless.
- The bottom panel of figure 4.6 shows the average attribute $\rho_n Y_n$ for each species, for the FIFO policy and the policy given by (4.11). This represents a measure of the cost.

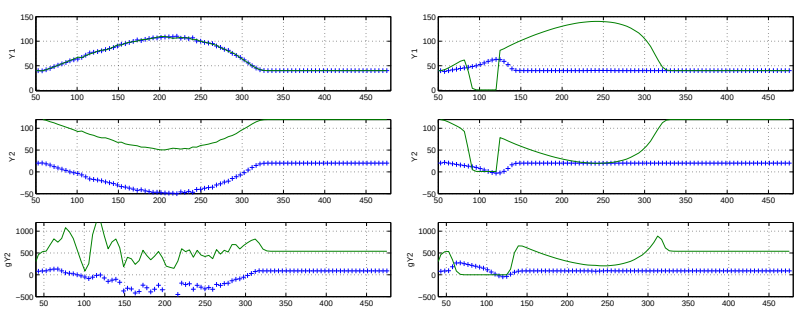

FIG. 4.6. *Attribute comparison. Left panel: FIFO. Right panel: service rule for perishable goods, defined by the priority function (4.11). Top: cycle time at exit. Middle: time to due date at exit. Bottom: attribute density* $\rho_n Y_n^{(2)}$ *of the time to due date. += species 1, solid = species 2.*

5. Conclusions

The evolution of parts in a supply chain, governed by a quite general class of service rules based on prioritizing attributes, can be modeled by a set of hyperbolic conservation laws. These conservation laws yield an exact solution to the underlying kinetic equations, as long as the level sets of parts of equal priority form a set of measure zero in a sufficiently high dimensional attribute space. This situation can be created by artificially inflating the attribute space, essentially breaking down parts of equal priority into subgroups. The resulting macroscopic model can even be used to model stochastic systems as long as the correct flux function for the evolution of the expectation of the whole ensemble is known.

Appendix A. Proof of Lemma 3.1.

Proof. In order to keep the conservative form of the equation it is convenient to formulate the problem (1.2)–(1.3) in a weak form. This weak form is given by

$$\int_0^\infty dx \int_0^\infty dt \int dy \; \times \ldots \tag{A.1}$$

$$f(x,y,t)[\partial_t \psi(x,y,t) + v_f(x,t,p(x,y,t))\partial_x \psi(x,y,t) + E(x,t)\nabla_y \psi(x,y,t)] = B[\psi]$$

for all test functions $\psi(x,y,t)$, with the functional $B[\psi]$ given by

$$B[\psi]=\int_0^\infty dx\int dy\ \psi(x,y,0)f^I(x,y)+\int_0^\infty dt\int dy\ \psi(0,y,t)F^B(y,t),$$

describing the initial and boundary conditions.

The evolution equation for the level set function Λ is derived by using a test function ψ whose dependence on the attribute y is solely through the priority p. So we set

$$\psi(x,y,t)=\psi_1(x,t,p(x,y,t))=\int\psi_1(x,t,q)\delta(p(x,y,t)-q)\ dq$$

and obtain

$$R[\psi_1]+S[\psi_1]=B[\psi_1]$$

with the functionals R, S and B given by

$$R[\psi_1]=\int_0^\infty dx\int_0^\infty dt\int dq\int dy\ \times\ldots$$

$$f(x,y,t)\delta(p(x,y,t)-q)[\partial_t\psi_1(x,t,q)+v(x,t,\phi_f(x,t,q))\partial_x\psi_1(x,t,q)],$$

$$S[\psi_1]=\int_0^\infty dx\int_0^\infty dt\int dy\int dq\ \times\ldots \tag{A.2}$$

$$f(x,y,t)\psi_1(x,t,q)\delta'(p(x,y,t)-q)[\partial_t p+v_f(x,t,p)\partial_x p+E(x,t)\nabla_y p],$$

$$B[\psi_1]=\int_0^\infty dx\int dq\ \psi_1(x,q,0)\Lambda^I(x,q)+\int_0^\infty dt\int dq\ \psi_1(0,t,q)F^B_\Lambda(q,t),$$

and Λ^I, F^B_Λ defined as in (3.2)(b,c). Now, by definition $\int f(x,y,t)\delta(p-q)\ dy=\Lambda(x,t,q)$, $\forall x,q,t$ holds. Therefore, $R[\psi_1]$ becomes

$$R[\psi_1]=\int_0^\infty dx\int_0^\infty dt\int dq\ \Lambda(x,t,q)[\partial_t\psi_1(x,t,q)+v(x,t,\phi_f(x,t,q))\partial_x\psi_1(x,t,q)]$$

integrating by parts with respect to the variable q in (A.2) gives

$$S[\psi_1]=\int_0^\infty dx\int_0^\infty dt\int dy\int dq\ \times\ldots$$

$$f(x,y,t)\partial_q\psi_1(x,t,q)\delta(p(x,y,t)-q)[\partial_t p+v_f(x,t,p)\partial_x p+E(x,t)\nabla_y p]$$

or

$$S[\psi_1]=\int_0^\infty dx\int_0^\infty dt\int dq\ \partial_q\psi_1(x,t,q)A_f(x,t,q),$$

$$A_f(x,t,q)=\int f(x,y,t)\delta(p(x,y,t)-q)[\partial_t p+v_f(x,t,p)\partial_x p+E(x,t)\nabla_y p]\ dy.$$

So the equation $R[\psi_1]+S[\psi_1]=B[\psi_1]$, $\forall\psi_1$ is the weak form of the problem (3.2). □

Proof of Theorem 3.3.

Proof. The weak formulation of (3.2)(a) is given by

$$\int \Lambda(x,t,q)\partial_t\psi\ dxqt+\int \Lambda(x,t,q)v_f(x,t,q)\partial_x\psi\ dxqt+\int A_f\partial_q\psi\ dxqt=0,$$

where $\psi(x,t,q)$ is a smooth test function vanishing at $t=0$ and $x=0$. Inserting the form (3.4) for Λ into the weak form of the equation gives

$$\int \tilde{\Lambda}(x,t,q)[\partial_t\psi+v_f(x,t,q)\partial_x\psi]+A_f(x,t,q)\partial_q\psi\ dxqt+$$

$$\int a(x,t)\partial_2\psi(x,t,P)\ dxt+\int a(x,t)v_f(x,t,P)\partial_1\psi(x,t,P)\ dxt=0,$$

where we denote with $\partial_1\psi$, $\partial_2\psi$ the partial derivative of the test function ψ with respect to its first and second variable (to distinguish from the total derivative). Since the smooth part $\tilde{\Lambda}$ of the solution has to satisfy the equation (3.2) pointwise away from the discontinuity at $q=P(x,t)$, we obtain

$$\int a(x,t)\partial_2\psi(x,t,P)\ dxt+\int a(x,t)v_f(x,t,P)\partial_1\psi(x,t,P)\ dxt=0, \tag{A.3}$$

At the same time the cumulative density $\phi(x,t,q)$, satisfying $\partial_q\phi=-\Lambda$ has to satisfy equation (3.3). The Rankine Hugoniot condition for equation (3.3) implies that if $\phi(x,t,q)$ has a jump discontinuity (a shock) of the form

$$\phi(x,t,q)=\tilde{\phi}(x,t,q)+a(x,t)H(P(x,t)-q),\quad \tilde{\Lambda}=-\partial_q\tilde{\phi},$$

then the function $P(x,t)$ has to satisfy

$$\partial_t P[\phi]_P+[F]_P\partial_x P=0,$$

with the jumps $[\phi]_P$ and $[F]_P$ defined by

$$[\phi]_P(x,t)=\lim_{\varepsilon\to 0+}\phi(x,t,P(x,t)+\varepsilon)-\phi(x,t,P(x,t)-\varepsilon),$$

$$[F]_P(x,t)=\lim_{\varepsilon\to 0+}F(x,t,\phi(x,t,P(x,t)+\varepsilon))-F(x,t,\phi(x,t,P(x,t)-\varepsilon)).$$

So, the jump in ϕ is given by $[\phi]_P=a$. We write the jump $[F]_P$ in the flux as $[F]_P=aw$ with

$$w(x,t)=\frac{[F]_P}{[\phi]_P}=\frac{1}{a}[F(x,t,\tilde{\phi}(x,t,P(x,t)+a))-F(x,t,\tilde{\phi}(x,t,P(x,t)))]$$

and obtain the equation

$$\partial_t P+w\partial_x P=0 \tag{A.4}$$

from the Rankine Hugoniot condition. We rewrite the first term in (A.3) as

$$\int a(x,t)\partial_2\psi(x,t,P)\ dxt=\int a(x,t)\delta(P-q)\partial_t\psi(x,t,q)\ dxqt$$

$$= -\int \partial_t a(x,t)\delta(P-q)\psi(x,t,q)\ dxqt - \int a(x,t)\delta'(P-q)\partial_t P\psi(x,t,q)\ dxqt$$

$$= -\int \partial_t a(x,t)\delta(P-q)\psi(x,t,q)\ dxqt - \int a(x,t)\delta(P-q)\partial_t P\partial_q\psi(x,t,q)\ dxqt$$

$$= -\int \partial_t a(x,t)\psi(x,t,P)\ dxt + \int a(x,t)w(x,t)\partial_x P\partial_3\psi(x,t,P)\ dxt \tag{A.5}$$

where the last equality holds because of (A.4). Similarly, we obtain for the second term in (A.3):

$$\begin{aligned}
&\int a(x,t)v_f(x,t,P)\partial_1\psi(x,t,P)\ dxt \\
&= \int a(x,t)v_f(x,t,P)\delta(P-q)\partial_x\psi(x,t,q)\ dxqt \\
&= -\int \partial_x[a(x,t)v_f(x,t,P)]\delta(P-q)\psi(x,t,q)\ dxqt \\
&\qquad - \int a(x,t)v_f(x,t,P)\delta'(P-q)\partial_x P\psi(x,t,q)\ dxqt \\
&= -\int \partial_x[a(x,t)v_f(x,t,P)]\delta(P-q)\psi(x,t,q)\ dxqt \\
&\qquad - \int a(x,t)v_f(x,t,P)\delta(P-q)\partial_x P\partial_q\psi(x,t,q)\ dxqt \\
&= -\int \partial_x[a(x,t)v_f(x,t,P)]\psi(x,t,P)\ dxt \\
&\qquad - \int a(x,t)v_f(x,t,P)\partial_x P\partial_2\psi(x,t,P)\ dxt.
\end{aligned} \tag{A.6}$$

Summing the expressions (A.5) and (A.6) to give (A.4) we obtain

$$\int \{-\partial_t a - \partial_x[a(x,t)v_f(x,t,P)]\}\psi(x,t,P)\ dxt$$

$$+ \int a(x,t)[w(x,t) - v_f(x,t,P)]\partial_x P\partial_2\psi(x,t,P)\ dxt = 0.$$

Since $\psi(x,t,q)$ is an arbitrary test function, the terms $\psi(x,t,q)$ and $\partial_3\psi(x,t,P(x,t))$ can be chosen independently. Therefore

$$v_f(x,t,P(x,t)) = w(x,t) = \frac{1}{a}[F(x,t,\tilde{\phi}(x,t,P(x,t)+a)) - F(x,t,\tilde{\phi}(x,t,P(x,t)))] \tag{A.7}$$

has to hold for a function Λ of the form (3.4) to be weak solution of the equation (3.2). If we define the velocity $v_f(x,t,q)$ according to (3.5), then $v_f(x,t,q) = \partial_q F(x,t,q)$ holds away from $q = P(x,t)$, where ϕ is smooth. At $q = P(x,t)$ the definition (3.5) reduces to the correct expression (A.7). So, if we redefine $v_f(x,t,q)$ by (3.5) we obtain a solution of the level set equation (3.2) in the weak sense. $\square$

REFERENCES

[1] D. Armbruster, D. Marthaler and C. Ringhofer, *Kinetic and fluid model hierarchies for supply chains*, SIAM J. Multiscale Modeling and Simulation, 2(1), 43–61, 2004.

[2] D. Armbruster, P. Degond and C. Ringhofer, *A Model for the dynamics of large queuing networks and supply chains*, SIAM J. Appl. Math., 66, 896–920, 2006.

[3] D. Armbruster, P. Degond and C. Ringhofer, *Kinetic and fluid models for supply chains supporting policy attibutes*, Bulletin of the Inst. Math., Academica Sinica, 2, 433–460, 2007.

[4] J.G. Dai and G. Weiss, *Stability and instability of fluid models for re-entrant lines*, Math. Oper. Res., 21, 1, (February), 1996.

[5] P. Degond and C. Ringhofer, *Stochastic dynamics of long supply chains with random breakdowns*, SIAM J. Appl. Math., 68(1), 59–79, 2007.

[6] A. Fügenschuh, S. Göttlich and M. Herty, *A new modeling approach for an integrated simulation and optimization of production networks*, H.-O. Gunther, D. Mattfeld, L. Suhl (Eds.), Management logistischer Netzwerke, Physica-Verlag Heidelberg, 45–60, 2007.

[7] S. Göttlich, M. Herty and A. Klar, *Network models for supply chains*, Commun. Math. Sci., 3(4), 545–559, 2005.

[8] W.H.S.C. Graves and D.B. Kletter, *A dynamic model for requirements planning with application to supply chain optimization*, Operations Research, 46(3), 35–49, 1998.

[9] S. Jin and X. Li, *Multi-phase computations of the semiclassical limit of the Schrodinger equation and related problems: Whitham vs. Wigner*, Physica D, 182, 46–85, 2003.

[10] S. Jin, H. Liu, S. Osher and R. Tsai, *Computing multi-valued physical observables for the high frequency limit of symmetric hyperbolic systems*, J. Comput. Phys., 210, 497–518, 2005.

[11] P.R. Kumar and S.H. Lu, *Distributive scheduling based on due dates and buffer priorities*, IEEE Trans. Autom. Control, 36, 1410–1416, 1991.

[12] D. Simchi-Levi, P. Kaminsky and E. Simchi-Levi, *Designing and Managing The Supply Chain*, McGraw-Hill Irwin, New York, 2nd edition, 2003.

[13] R. Tsai and S. Osher, *Level set methods and their applications in image science*, Commun. Math. Sci., 1(4), 623–656, 2003.