SIAM J. CONTROL OPTIM.
Vol. 52, No. 1, pp. 663–686

© 2014 Society for Industrial and Applied Mathematics

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A VARIATIONAL APPROACH FOR CONTINUOUS SUPPLY CHAIN NETWORKS[*]

KE HAN[†], TERRY L. FRIESZ[‡], AND TAO YAO[‡]

Abstract. We consider a continuous supply chain network consisting of buffering queues and processors first proposed by [D. Armbruster, P. Degond, and C. Ringhofer, *SIAM J. Appl. Math.*, 66 (2006), pp. 896–920] and subsequently analyzed by [D. Armbruster, P. Degond, and C. Ringhofer, *Bull. Inst. Math. Acad. Sin. (N.S.)*, 2 (2007), pp. 433–460] and [D. Armbruster, C. De Beer, M. Freitag, T. Jagalski, and C. Ringhofer, *Phys. A*, 363 (2006), pp. 104–114]. A model was proposed for such a network by [S. Göttlich, M. Herty, and A. Klar, *Commun. Math. Sci.*, 3 (2005), pp. 545–559] using a system of coupling ordinary differential equations and partial differential equations. In this article, we propose an alternative approach based on a variational method to formulate the network dynamics. We also derive, based on the variational method, a computational algorithm that guarantees numerical stability, allows for rigorous error estimates, and facilitates efficient computations. A class of network flow optimization problems are formulated as mixed integer programs (MIPs). The proposed numerical algorithm and the corresponding MIP are compared theoretically and numerically with existing ones [A. Fügenschuh, S. Göttlich, M. Herty, A. Klar, and A. Martin, *SIAM J. Sci. Comput.*, 30 (2008), pp. 1490–1507; S. Göttlich, M. Herty, and A. Klar, *Commun. Math. Sci.*, 3 (2005), pp. 545–559], which demonstrates the modeling and computational advantages of the variational approach.

Key words. continuous supply chain, partial differential equations, variational method, mixed integer programs

AMS subject classifications. 35C05, 49J20, 90C11

DOI. 10.1137/120868943

1. Introduction.

1.1. Modeling overview.

Manufacturing systems can be described by a number of mathematical models, which provide powerful tools to study and analyze the behavior of such systems under specific conditions. Among theses mathematical representations, we distinguish between static/stationary models and dynamic models; the latter has an inherent dependence on time and falls within the scope of this paper.

The dynamic models describe and predict time evolution of system states by introducing dynamics to different production steps. These models can be further categorized as *discrete event simulation* (DES) [5] and continuum models [2]. The DES is an exact and computationally intensive method, in which the evolution of the system is viewed as a sequence of significant changes in time, called *events*, for each part (product) separately. A cost-effective alternative to the discrete event models are fluid-based continuum network models represented by *partial differential equations* (PDEs). The continuum models seek to describe the system dynamics from an aggregate level and ignore granularities in local space and time. There has been a significant number of publications on the PDE formulation of traffic dynamics, for example, in [12, 13,

[*] Received by the editors March 6, 2012; accepted for publication (in revised form) December 11, 2013; published electronically February 20, 2014. This work was partially supported by the NSF through grant EFRI-1024707, "A theory of complex transportation network design."
http://www.siam.org/journals/sicon/52-1/86894.html

[†] Department of Civil and Environmental Engineering, Imperial College London, London SW7 2BU, UK (k.han@imperial.ac.uk).

[‡] Department of Industrial and Manufacturing Engineering, Pennsylvania State University, University Park 16802 (tfriesz@psu.edu, tyy1@engr.psu.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

25, 37, 38, 41]. The continuum modeling technique was not applied to supply chain networks until recently, by the seminal work of [1, 2, 3, 4, 11, 15, 33] and [34].

In particular, [2] derived, based on simple rules for releasing parts, a conservation law model for the density and flux (flow) of the parts in the production process:

$$\partial_t \rho(t,\, x) + \partial_x \min\left\{V(x)\rho(t,\, x),\, \mu(x)\right\} = 0, \qquad t \in [0,\, T], \quad x \in [a,\, b], \tag{1.1}$$

where the processor is expressed as a spatial interval $[a,\, b]$, and $\rho(t,\, x)$ denotes the spatial-temporal distribution of the product density. In other words, $\rho(t,\, x)$ measures the local product density (in number of products per unit distance) at time t and location x. $V(x)$ and $\mu(x)$ denote location-dependent processing speed and flow capacity, respectively. The above PDE will be asymptotically valid in regimes where a substantial number of parts are present on the processor. It should be noted that the solution to the conservation law (1.1) can only be considered in the distributional sense, due to the discontinuous dependence of the flux function on x. This is easily seen from an example involving a bottleneck: consider the flow capacity $\mu(a)$ at a point a and assume that $\mu(a^-) > \mu(a^+)$. If the flow at $x = a-$ is saturated, a Dirac δ-distribution will emerge in the density profile $\rho(t,\, \cdot)$ at location $x = a$, which corresponds to an active bottleneck. When this happens, an integral solution of (1.1) does not exist.

To overcome such theoretical difficulty, [23] proposed, in addition to the PDE formulation (1.1), a separate ordinary differential equation for the buffering queues immediately upstream of each processor, thus avoiding direct encounter of the δ-distribution. This finally leads to a system of conservation laws coupled with ordinary differential equations. This supply chain model can be extended to incorporate general network topology [30], certain real-world production features such as multicommodity or due-date production [16], and a class of network control and optimization problems [24].

1.2. Numerical techniques.

In [16, 22, 23] and [24], the numerical methods for solving the aforementioned coupling system of PDEs and ODEs is an upwind finite difference scheme for the conservation laws and a forward Euler scheme for the ordinary differential equations. To guarantee stability of the explicit discretization scheme, the Courant–Friedrichs–Lewy (CFL) condition [36] must hold for the conservation law:

$$\Delta t \leq \min_i \left\{\frac{\Delta x_i}{V^i}\right\}, \tag{1.2}$$

where Δt denotes the time step, index i runs through every processor (link) of the supply chain network, and Δx_i and V^i denote, respectively, the spatial step and the maximum processing speed associated with processor i. The solution method described above encounters several numerical difficulties. First, the ODE representing the buffer queue has a discontinuous dependence on its unknown variable, and such irregularity may result in a negative queue size in a time discretization. Certain modifications were proposed in the literature to remedy such numerical deficiency. For example, [4] proposed a smoothing parameter to revise the dynamics at queues. However, such modification could sometimes suffer from nonphysical solutions to be illustrated in section 5.5. Second, the CFL condition for the conservation law and the stiffness condition for the modified ODE [4] imply a tradeoff between numerical accuracy and computational efficiency, which could potentially increase the computational burden of the solution method.

In view of the above limitations, we propose in this article a reformulation of the network dynamics using cumulative production curves [44] and Hamilton–Jacobi

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

equations. As we shall demonstrate subsequently, this alternative not only eliminates the abovementioned numerical issues but also leads to an efficient "grid-free" algorithm and closed-form solution representation. Here, "grid-free" means the PDE is solved without spatial discretization and without intermediate calculation of densities inside the spatial interval. The solution procedure of the Hamilton–Jacobi equation is a variational method called the Lax formula [7, 14, 18, 19, 35]. The formula was originally proposed as a semianalytic solution representation of the scalar conservation law and the Hamilton–Jacobi equation [18, 35]; its applications to fluid-based traffic modeling are recently investigated in [7, 8, 9, 10, 14]. Using the variational approach, the viscosity solution of the Hamilton–Jacobi equation is formulated as an optimization problem which, depending on the specific form of the Hamiltonian, may be simplified or explicitly instantiated. The variational approach for the continuous supply chains networks is, as we establish in this paper, a powerful analytical and computational tool, and its advantages compared to the finite-difference schemes are listed as follows:

1. The dynamics of the buffer queue and the processor can be simultaneously treated using a single Lax formula, thus avoiding separate modeling of these two model components.
2. The Lax formula yields a much lower numerical error in the solution than the finite-difference schemes.
3. The proposed algorithm is grid-free; in other words, there is no need to discretize the spatial domain of the PDE.
4. The proposed algorithm does not impose any constraints on the time step or uniformity of the time grid. Thus one has more flexibility in choosing the time grid for computational convenience.
5. Our reformulation of the supply chain networks is free of the spatial variables originally appearing in the PDEs. This is arguably more general than the PDE-based system in terms of modeling assumptions and solution methods.

This paper also presents a mixed integer program (MIP) formulation of the supply chain network optimization problem using the variational reformulation. The MIP is one of the main formulations of supply chain optimization in the literature, see [39, 43], and its connection with continuum network models brings new insights to the management of such networks. Optimization problems in the form of MIPs have been investigated in the context of traffic flows and supply chains [21, 22, 27, 26]. This article reveals a new relationship between MIPs and fluid-based continuum models from the point of view of variational principle.

It is natural to compare our proposed MIP with existing ones such as those proposed by [21, 22]. The latter are based on a finite-difference discretization of the PDEs and ODEs. For the same reason mentioned before, the MIP based on the variational approach will allow more efficient computation and yields better solution quality. To reach the same level of numerical precision, the MIP we put forward requires much less (binary) variables than those based on finite-difference schemes; this will be established both theoretically and numerically in this paper.

The rest of the paper is organized as follows. Section 2 briefly reviews the supply chain network model originally proposed by [23], which consists of conservation laws and ODEs. In section 3, a variational approach is proposed to reformulate the system. The solutions are derived in closed form in both continuous and discrete time. Section 4 considers a network flow optimization problem based on the proposed variational approach and derives an MIP. Such an MIP is then compared to the MIP considered in [22]. Finally, several numerical experiments are presented in section 5 to illustrate the advantages of applying the variational approach.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2. Supply chain network model. We begin with the articulation of the network model proposed by [23]. The supply chain model consists of separate modeling of the buffer queues (using ODEs) and processors (using PDEs). A precise description of the network is made via the notion of directed graph $G(\mathcal{A},\, \mathcal{V})$ with the set of edges (or arcs) $\mathcal{A}$ and the set of vertices (or nodes) $\mathcal{V}$.

Definition 2.1 (continuous supply chain network).

1. *A continuous supply chain network is represented as a directed graph* $G(\mathcal{A},\, \mathcal{V})$ *where each edge* $e \in \mathcal{A}$ *corresponds to an individual processor, and each vertex* $v \in \mathcal{V}$ *represents the respective queue upstream of the processor.*
2. *Each processor* $e \in \mathcal{A}$ *is expressed as a spatial interval* $[a^e,\, b^e]$ *with* $L^e = b^e - a^e$ *being the length of the processor.*[1]
3. *Each processor possesses a queue, which is located at the vertex at the upstream end of the processor.*
4. *The flow capacity* μ^e*, processing speed* V^e*, and throughput time* $T^e = L^e/V^e$ *of each processor* $e \in \mathcal{A}$ *are constants.*

For each $e \in \mathcal{A}$, let $\rho^e(t,\, x)$ denote the density of products at time $t \in [0,\, T]$ and location $x \in [a^e,\, b^e]$; let $q^e(t)$ denote the size of the queue upstream of this processor. Assume that products are fed to the buffer queue at the rate $\overline{u}^e(t)$, before they are released to the processor. The dynamics of the processor and the queue are governed by the following advection equation, (2.1), and ODE, (2.2)–(2.3):

$$\partial_t \rho^e(t,\, x) + V^e\, \partial_x \rho^e(t,\, x) = 0, \quad \rho^e(0,\, x) = \rho_0^e(x), \quad (t,\, x) \in [0,\, T] \times [a^e,\, b^e], \tag{2.1}$$

$$\frac{d}{dt} q^e(t) = \overline{u}^e(t) - f^e\big(\rho^e(t,\, a^e)\big), \qquad q^e(0) = q_0^e, \tag{2.2}$$

$$f^e\big(\rho^e(t,\, a^e)\big) = \begin{cases} \min\{\overline{u}^e(t),\, \mu^e\}, & q^e(t) = 0, \\ \mu^e, & q^e(t) > 0. \end{cases} \tag{2.3}$$

Equation (2.3) represents the rate at which products in the queue are released to the processor, i.e., the service rate. Notice that the PDE (2.1) with the flow capacity constraint can be rewritten as a scalar conservation law:

$$\partial_t \rho^e(t,\, x) + \partial_x \phi^e\big(\rho^e(t,\, x)\big) = 0, \tag{2.4}$$

where the flux function satisfies

$$\phi^e(\rho) = \min\big\{V^e \rho,\, \mu^e\big\}. \tag{2.5}$$

A straightforward numerical scheme for solving system (2.1)–(2.3) is to apply space-time discretization and use the finite difference approximation; see [2, 16, 23, 24]. For example, one could use an upwind scheme for conservation law (2.1) and forward Euler method for ODE (2.2). Moreover, to ensure numerical stability, one needs to impose the CFL-type constraints on the time step; see (1.2).

One of the goals of this article is to provide an effective alternative to the numerical scheme mentioned above, namely, the variational method (Lax formula) [7, 14, 18]. The Lax formula was proposed in [35] for the study of scalar conservation laws of the form

$$\partial_t u + \partial_x f(u) = 0, \tag{2.6}$$

[1] Note that the spatial representation of the processor is somewhat imaginary and arbitrary. In an actual manufacturing process, the key quantity of interest is the processing time (or throughput time) of a part, which is assumed to be a constant in this paper. Having a virtual spatial dimension introduced to the dynamics enables us to invoke the PDE formulation.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

of which the PDE (2.1) is just a special case. The Lax formula expresses the weak entropy solution of (2.6) as the solution of a minimization problem which, in our case, can be expressed explicitly. Notice that (2.6) can be equivalently written as the following Hamilton–Jacobi equation:

$$\partial_t U + f\big(\partial_x U\big) = 0, \qquad \text{where} \quad U(t,\,x) = \int_0^t u(s,\,x)\,ds. \tag{2.7}$$

We also wish to consider the following initial condition:

$$U(0,\,x) = U_0(x). \tag{2.8}$$

The classical Lax formula for the Cauchy problem (initial value problem) (2.7)–(2.8) is stated below; the reader is referred to [18] for a detailed derivation.

LEMMA 2.2 (Lax formula). *Assume that $f(\cdot)$ is convex and that $U_0(\cdot)$ is Lipschitz continuous. Then the viscosity solution to the Cauchy problem* (2.7)–(2.8) *is*

$$U(t,\,x) = \inf_{y\in\mathbb{R}} \left\{ t f^*\left(\frac{x-y}{t}\right) + U_0(y) \right\} \qquad t > 0, \tag{2.9}$$

where $f^*(\cdot)$ is the Legendre transform of $f(\cdot)$: $f^*(v) = \sup_u\{vu - f(u)\}$.

3. Variational method.

In this section, we apply the variational formulation to the system governed by (2.1)–(2.3). We will demonstrate the capability of the Lax formula to simultaneously handle dynamics of the queue and processor, in a way consistent with (2.1)–(2.3), and to reduce the complexity of the coupling PDEs and ODEs introduced in [23]. We first consider a single processor. For simplicity of notation, the superscript e will be dropped for now. The following argument will be extended to a general supply chain network in section 3.3.

Denote the flux of products by $u(t,\,x)$, i.e., $u(t,\,x) \doteq \phi(\rho)$, where $\phi(\cdot)$ is defined in (2.5). Recall the density-based scalar conservation law (2.4), which is rewritten as a PDE whose unknown variable is the flux:

$$\partial_x u(t,\,x) + \partial_t g\big(u(t,\,x)\big) = 0, \tag{3.1}$$

where the function $u \mapsto g(u) = \rho$ is defined as the inverse of the function $\rho \mapsto \phi(\rho) = u$ on the interval $[0,\,\mu/V]$. To be precise, we have

$$g(u) = u/V, \qquad u \in [0,\,\mu]. \tag{3.2}$$

Remark 3.1. Notice that such inversion of the unknown in the PDE is made possible only when the flux function $\phi(\cdot)$ is invertible. In addition, to show the equivalence between (3.1) and (2.4), one needs to invoke the definition of weak entropy solution (see [6]). The reader is referred to [7] for more detail on the inversion technique and [20] for a formal proof of equivalence.

Let us fix a time horizon $[0,\,T]$ for any given $T > 0$. The processor of interest is represented by a spatial interval $[a,\,b]$, where $b - a = L$. We introduce the function $U(\cdot,\,\cdot) : [0,\,T] \times [a,\,b] \mapsto \mathbb{R}_+$, defined as

$$U(t,\,x) \doteq \int_0^t u(s,\,x)\,ds, \qquad (t,\,x) \in [0,\,T] \times [a,\,b].$$

The conservation law (3.1) can be equivalently written as a Hamilton–Jacobi equation with unknown $U(t,\,x)$:

$$\partial_x U(t,\,x) + g\big(\partial_t U(t,\,x)\big) = 0. \tag{3.3}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Define function $Q(\cdot) : [0, T] \mapsto \mathbb{R}_+$, which measures the cumulative number of products that have arrived at queue by time t. For what follows, $Q(\cdot)$ is naturally assumed to be nondecreasing and continuous except for countably many t. For a reason that will become clearly later in section 3.1, we extend $Q(\cdot)$ to the time interval $(-\infty, 0)$ with value zero assigned. For the rest of this article, we consider the left-continuous version of $Q(\cdot)$, so that $Q(t) = Q(t-)$, $t \in \mathbb{R}$. Now consider the Lipschitz continuous function:

$$\overline{U}(t) \doteq \inf_{\tau \leq t} \big\{Q(\tau) + \mu(t-\tau)\big\} \leq Q(t). \tag{3.4}$$

Clearly, $\overline{U}(t)$ measures the total number of products that have been released from the buffering queue to the processor by time t. Notice that $Q(t) - \overline{U}(t)$ measures the size of the queue upstream of the processor. We consider the following "initial value" problem:

$$\begin{cases} \partial_x U(t,\, x) + g\big(\partial_t U(t,\, x)\big) = 0, \\ U(t,\, 0) = \overline{U}(t). \end{cases} \tag{3.5}$$

For $t > 0$, the viscosity solution to (3.5) is provided by the following variation of the Lax formula.

Proposition 3.2. *The solution to* (3.5) *is given by the following identities:*

$$U(t,\, x) = \inf_{\tau \in \mathbb{R}} \left\{ x\, g^* \left(\frac{t-\tau}{x}\right) + \overline{U}(\tau) \right\} = \inf_{\tau \in \mathbb{R}} \left\{ x\, g^* \left(\frac{t-\tau}{x}\right) + Q(\tau) \right\}, \tag{3.6}$$

where g^* is the Legendre transform of g,

$$g^*(q) = \sup_{p \in [0,\, \mu]} \big\{q\, p - g(p)\big\} = \begin{cases} 0, & q \leq \frac{1}{V}, \\ \left(q - \frac{1}{V}\right) \mu, & q > \frac{1}{V}, \end{cases} \tag{3.7}$$

where μ denotes the flow capacity of the processor, and V denotes the processing speed.

Proof. Given the functional format of $g(\cdot)$ from (3.2), the Legendre transform can be easily calculated as (3.7). In order to apply the Lax formula given in Lemma 2.2 to the problem (3.5), we readily notice, by switching the roles of t and x, that problem (3.5) is in fact an initial value problem. Adjusting the Lax formula to such a Cauchy problem yields the first identity of (3.6). To prove the second identity, we refer the reader to [7]. $\square$

Equation (3.6) has several significant impacts. (1) The first identity, which is an adjustment of the Lax formula to treat boundary value problems, provides a grid-free and semianalytical solution representation of PDE (3.5). (2) The second identity suggests that the solution can be represented directly in terms of the inflow profile $Q(\cdot)$ without knowledge or intermediate computation of $\overline{U}(\cdot)$. As a result, the separating modeling of processor and buffer queue (2.1)–(2.3) are no longer necessary, and the coupled PDEs and ODEs are replaced by the closed-form solutions provided by (3.6). (3) The assumptions made on $Q(\cdot)$ suggest that our methodological framework can accommodate very general inflow profile, which can even be a distribution.

3.1. Explicit solution in continuous time. Given the simple and explicit functional forms of $g(\cdot)$ and $g^*(\cdot)$, we are able to further simplify the semianalytical expression (3.6) for the solution. The goal of this subsection is to derive closed-form solutions for the system in continuous time. A time-discretization will be introduced in the next subsection to express the solution in discrete time.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

For a reason that will become clear later, we focus our analyses on the family of *piecewise affine* inflow profiles $Q(\cdot)$. Such an assumption is not too restrictive in application since any piecewise continuous function can be approximated, to an arbitrary degree of accuracy, by piecewise affine functions. Notice that by (3.4), $\overline{U}(\cdot)$ will also be piecewise affine. Given a fixed processor with length L, flow capacity μ, and processing speed V, by setting $x = L$ formula (3.6) becomes

$$U(t,\,L) = \min\left\{\inf_{\tau\geq t-L/V} Q(\tau), \qquad \inf_{\tau<t-L/V}\big\{Q(\tau)-\mu\tau+(t-L/V)\mu\big\}\right\}. \tag{3.8}$$

We recall that $Q(\cdot)$ is nondecreasing and left-continuous, and therefore

$$\begin{aligned}\inf_{\tau<t-L/V}\big\{Q(\tau)-\mu\tau+(t-L/V)\mu\big\} &\leq \lim_{\tau\to(t-L/V)^-}\big\{Q(\tau)-\mu\tau+(t-L/V)\mu\big\}\\ &= Q(t-L/V) = \inf_{\tau\geq t-L/V} Q(\tau).\end{aligned}$$

Now we can write (3.8) in a simplified form:

$$U(t,\,L) = \inf_{\tau\leq t-L/V}\big\{Q(\tau)-\mu\tau\big\}+(t-L/V)\mu. \tag{3.9}$$

Remark 3.3. Identity (3.9) has a simple interpretation. Recall that the cumulative number of products that have been released from the queue into the processor at time t is given by (3.4):

$$\overline{U}(t) \doteq \inf_{\tau\leq t}\big\{Q(\tau)+\mu(t-\tau)\big\}.$$

Then for a product that exits the processor at time t, the time of its entry into the processor is $t-L/V$. In view of the natural first-in-first-out assumption, the total products that have exited the processor at time t is equal to the total products that have been released into the processor at time $t-L/V$. Replacing t by $t-L/V$ in (3.4) gives rise to (3.9).

The initial conditions for the buffer queue and the processor can be incorporated into our formula. Let $q(t)$ be the queue size and $\rho(t,\,x)$ be the density of products on the processor. Consider the following initial conditions:

$$q(0)=q_0\geq 0, \qquad \rho(0,\,\cdot\,)=\rho_0(\cdot)\in\mathcal{L}^1([a,\,b]), \tag{3.10}$$

where $\mathcal{L}^1([a,\,b])$ denotes the set of Lebesque integrable functions on the interval $[a,\,b]$. Let $\hat{Q}(\cdot):(-\infty,\,T)\mapsto\mathbb{R}_+$ be the left-continuous function defined by

$$\hat{Q}(t)=\begin{cases}0, & t\leq 0,\\ q_0+Q(t), & t>0.\end{cases} \tag{3.11}$$

We are now ready to state and prove one of the main results of this article, namely, the explicit instantiation of the variational principle (3.6) with piecewise affine boundary profile $Q(\cdot)$ and initial conditions q_0, $\rho_0(\cdot)$.

Proposition 3.4 (continuous-time solution). *Given constants μ, L, V and piecewise affine inflow profile $Q(\cdot):[0,\,T]\mapsto\mathbb{R}_+$ with break points $\{\xi_i\in\mathbb{R}:\ i\in\mathcal{I}\}$, we associate with each time $t>L/V$ the finite set $\Omega_t\doteq\{\xi_i:\ \xi_i\leq t-L/V\}\bigcup\{t-L/V\}$. Given the initial conditions*

$$q(0)=q_0, \qquad \rho(0,\,x)=\rho_0(x) \tag{3.12}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

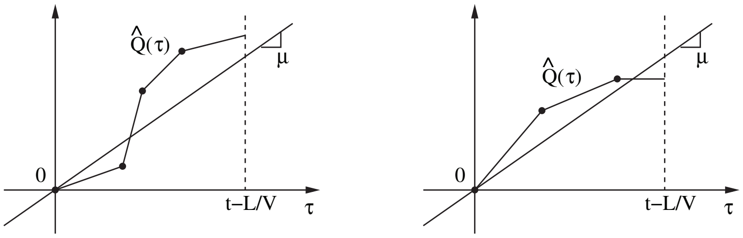

FIG. 1. *Graphical representation of the formula* (3.9). *Since $\hat{Q}(\cdot)$ is piecewise affine, the infimum in the vertical difference between $\hat{Q}(\tau)$ and $\mu\tau$ is obtained either at the break points of $\hat{Q}(\cdot)$ (left) or at $t - L/V$ (right).*

we define $\hat{Q}(\cdot)$ as in (3.11). *Then the exit flow profile $U(t, L)$ can be written as*

$$U(t, L) = \begin{cases} \displaystyle\int_{L-Vt}^{L} \rho_0(\zeta)\, d\zeta, & t \in [0, L/V), \\ \displaystyle\min_{\tau\in\Omega_t} \left\{\hat{Q}(\tau) - \mu\,\tau\right\} + (t - L/V)\mu + \int_0^L \rho_0(\zeta)\, d\zeta, & t \in [L/V, T]. \end{cases} \tag{3.13}$$

Moreover, $U(\cdot, L)$ is Lipschitz continuous.

Proof. Notice that L/V is the minimal time taken to traverse the processor; therefore at any positive time $t < L/V$, $U(t, L)$ is completely determined by the initial distribution $\rho_0(x)$ of products on the processor. A simple calculation using method of characteristics shows that for $0 \le t < L/V$,

$$U(t, L) = \int_0^t V\rho(s, L)\, ds = \int_0^t V\rho_0(L - sV)\, ds = \int_{L-tV}^{L} \rho_0(\zeta)\, d\zeta. \tag{3.14}$$

For time beyond L/V, we argue that an initially positive queue q_0 can be treated as an upward jump in $Q(\cdot)$ at $t = 0$, which is incorporated by replacing $Q(\cdot)$ with left-continuous, piecewise affine and nondecreasing function $\hat{Q}(\cdot)$ defined in (3.11). Finally, in view of (3.9), it remains to show that

$$\inf_{\tau\le t-L/V} \left\{\hat{Q}(\tau) - \mu\tau\right\} + (t - L/V)\mu = \min_{\tau\in\Omega_t} \left\{\hat{Q}(\tau) - \mu\tau\right\} + (t - L/V)\mu. \tag{3.15}$$

This is true since $\hat{Q}$ is piecewise affine, and the infimum in (3.15) must be obtained at either one of the break points or at $t - L/V$. See Figure 1 for an illustration. Since the Lax formula (3.6) gives an viscosity solution to the Hamilton–Jacobi equation, $U(t, L)$ is Lipschitz continuous with Lipschitz constant μ. ∎

The expression (3.13) does not depend on any sort of approximation and is therefore exact. Notice that the feasible set Ω_t is finite, and thus we have converted a continuous optimization problem into a discrete and finite form.

3.2. Explicit solution in discrete form.

3.2. Explicit solution in discrete form. The goal of this subsection is to derive the discrete version of (3.13) for a given time grid. This new algorithm will be applied to network simulation and optimization later in this paper. We consider a time horizon $[0, T]$ for some $T > 0$ and a uniform time grid $0 = t_0 < t_1 < \cdots < t_N = T$ with step size h.

Let $Q(\cdot)$ be any nondecreasing, left-continuous function. For notation convenience, let $Q_i \doteq Q(t_i-)$, $\overline{U}_i \doteq \overline{U}(t_i)$, $0 \le i \le N$. Then $Q(\cdot)$ is approximated by the piecewise-affine function with break points $\{(t_i, Q_i)\}_{i=0}^N$. To derive an explicit

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

numerical scheme, we make one simplification by rounding $t_i - L/V$ to the nearest grid point to the left and define integer constant $\Delta \doteq \lceil \frac{L}{Vh} \rceil$.

We make note of the fact that the presence of the initial density $\rho_0(x)$ induces only integral terms to the Lax formula (3.13), for which the error is quite standard in the literature. Therefore, in the following statements of discrete formula and accompanying numerical error, we assume $\rho_0(x) \equiv 0$.

PROPOSITION 3.5 (Lax formula in discrete form). *Given constants L, V, μ, discrete values Q_i, $i = 0, \dots, N$, and initial data $q(0) = q_0 > 0$, $\rho_0(x) \equiv 0$, define $\hat{Q}_0 = 0$, $\hat{Q}_i = Q_i + q_0$, $1 \le i \le N$. Let $U(t, L)$ satisfy* (3.13). *Then the following discrete version of Lax formula*

$$U_{i,\,L} = \begin{cases} 0, & 0 \le i < \Delta, \\ \min\limits_{0 \le j \le i-\Delta} \left\{ \hat{Q}_j - \mu t_j \right\} + (t_i - L/V)\mu, & \Delta \le i \le N \end{cases} \tag{3.16}$$

satisfies

$$0 \le U_{i,\,L} \le U_{i+1,\,L}, \qquad \Delta \le i \le N-1 \tag{3.17}$$

and

$$0 \le U_{i,\,L} - U(t_i,\,L) \le \left(\left\lceil \frac{L}{Vh} \right\rceil - \frac{L}{Vh} \right) h\mu, \qquad \Delta \le i \le N. \tag{3.18}$$

In particular, if $\frac{L}{Vh}$ is integer, (3.16) *becomes exact.*

Proof. We first verify (3.17). Notice that $\min_{0 \le j \le i-\Delta}\{\hat{Q}_j - \mu t_j\} - \min_{0 \le j \le i+1-\Delta}$ $\{\hat{Q}_j - \mu t_j\}$ is equal to either 0 or $\min_{0 \le j \le i-\Delta}\{\hat{Q}_j - \mu t_j\} - (\hat{Q}_{i+1-\Delta} - \mu t_{i+1-\Delta})$; in the latter case

$$\begin{aligned} &\min_{0 \le j \le i-\Delta} \left\{ \hat{Q}_j - \mu t_j \right\} - \min_{0 \le j \le i+1-\Delta} \left\{ \hat{Q}_j - \mu t_j \right\} \\ &\quad \le \hat{Q}_{i-\Delta} - \mu t_{i-\Delta} - (\hat{Q}_{i+1-\Delta} - \mu t_{i+1-\Delta}) \le \mu\Delta, \end{aligned}$$

and thus $U_{i,\,L} - U_{i+1,\,L} = \min_{0 \le j \le i-\Delta}\{\hat{Q}_j - \mu t_j\} - \min_{0 \le j \le i+1-\Delta}\{\hat{Q}_j - \mu t_j\} - \Delta\mu \le 0$. This shows monotonicity. To show nonnegativity, it suffices to check that $U_{\Delta,\,L} \ge 0$.

For error estimate (3.18), recall $\Delta \doteq \lceil \frac{L}{Vh} \rceil$; it follows from (3.13) and the definition of Ω_t that

$$\begin{aligned} U(t_i,\,L) &= \min_{\tau \in \Omega_{t_i}} \left\{ \hat{Q}(\tau) - \mu\tau \right\} + (t_i - L/V)\mu \\ &= \min \left\{ \min_{0 \le j \le i-\Delta} \left\{ \hat{Q}(t_j) - \mu t_j \right\}, \quad \hat{Q}(t_i - L/V) - \mu(t_i - L/V) \right\} \\ &\quad + (t_i - L/V)\mu \\ &\le U_{i,\,L} \end{aligned}$$

and $U(t_i,\,L) < U_{i,\,L}$ if and only if $\hat{Q}(t_i - L/V) - \mu(t_i - L/V) < \min_{0 \le j \le i-\Delta}\{\hat{Q}(t_j) - \mu t_j\}$. In this case, we have the estimate

$$\begin{aligned} U_{i,\,L} - U(t_i,\,L) &= \min_{0 \le j \le i-\Delta} \left\{ \hat{Q}(t_j) - \mu t_j \right\} - \hat{Q}(t_i - L/V) + \mu(t_i - L/V) \\ &\le \hat{Q}(t_{i-\Delta}) - \mu t_{i-\Delta} - \hat{Q}(t_i - L/V) + \mu(t_i - L/V) \\ &\le \mu(t_i - L/V - t_{i-\Delta}) \\ &= \mu(t_i - L/V - t_i + \Delta h) = \mu h \left(\Delta - \frac{L}{Vh} \right). \end{aligned}$$

This verifies (3.18). □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Remark 3.6. Inequality (3.18) implies the uniform convergence of the numerical method as $h \to 0$. In fact, one may extend such error estimates to piecewise continuous $Q(\cdot)$, using standard results on linear interpolation.

Proposition 3.5 does not impose any constraint on the time grid in terms of step size and uniformity, except that $\Delta \geq 1$. This condition implies that the time step has to be less than or equal to the minimum processing time of the processor. According to (3.18), in order to reduce the numerical error, one may either reduce the step size h, or, less obviously, choose h in a way such that the processing time L/V is a multiple of h.

3.3. Network model. We are now ready to extend the Lax formula to a general supply chain network. In view of Definition 2.1, we introduce additional notation. Given a node $v \in \mathcal{V}$, the set of incoming arcs and outgoing arcs are denoted by $\mathcal{I}^v$ and $\mathcal{O}^v$, respectively. In the case $|\mathcal{O}^v| > 1$, we call v a *dispersive node*. We introduce the flow allocation rate $A^{v,\,e}(t)$ for each node v and $e \in \mathcal{O}^v$:

$$\begin{cases} 0 \leq A^{v,\,e}(t) \leq 1, \qquad \displaystyle\sum_{e \in \mathcal{O}^v} A^{v,\,e}(t) = 1 & \text{if } |\mathcal{O}^v| > 1, \\ A^{v,\,e}(t) \equiv 1 & \text{if } |\mathcal{O}^v| = 1. \end{cases} \tag{3.19}$$

These allocation rates $A^{v,\,e}(t)$ describe the proportion of the flow coming from incoming links that is going to the outgoing link e. They will be later considered as control of the network product flow and are subject to optimization. For a processor $e \in \mathcal{A}$, let $Q^e(t)$ be a nondecreasing, left-continuous function describing the cumulative number of products arriving at the buffer queue of the processor. Let $W^e(t)$ be a nondecreasing and Lipschitz continuous function describing the cumulative number of products that have left the processor. In addition, we fix the processor length L^e, the processing speed V^e, and the flow capacity μ^e. Adapting Proposition 3.4 to network notation, we obtain the following dynamics describing the supply chain network.

Proposition 3.7 (network model based on variational formulation). *Given a supply chain network $G(\mathcal{A},\, \mathcal{V})$, consider, for every $e \in \mathcal{A}$, parameters L^e, V^e, μ^e and initial conditions $q^e(0) = q_0$, $\rho^e(0,\, x) = \rho_0^e(x)$. Then we have the following variational formulations describing the dynamics of the system:*

$$\frac{d}{dt} Q^e(t) = A^{v,\,e}(t) \sum_{\bar{e} \in \mathcal{I}^v} \frac{d}{dt} W^{\bar{e}}(t) \quad \textit{for almost every } t \qquad \forall v \in \mathcal{V},\ e \in \mathcal{O}^v, \tag{3.20}$$

$$\hat{Q}^e(t) = \begin{cases} 0, & t \leq 0, \\ q_0^e + Q^e(t), & 0 < t \leq T, \end{cases} \qquad \forall e \in \mathcal{A}, \tag{3.21}$$

$$W^e(t) = \begin{cases} \displaystyle\int_{L^e - tV^e}^{L^e} \rho_0^e(\zeta)\, d\zeta, & t < \frac{L^e}{V^e}, \\ \displaystyle\inf_{\tau \leq t - \frac{L^e}{V^e}} \left\{ \hat{Q}^e(\tau) - \mu^e \tau \right\} + \left(t - \frac{L^e}{V^e}\right) \mu^e + \int_0^{L^e} \rho_0^e(\zeta)\, d\zeta, & t \geq \frac{L^e}{V^e}. \end{cases} \tag{3.22}$$

Remark 3.8. For any $e \in \mathcal{A}$, $W^e(t)$ is Lipschitz continuous by Proposition 3.4 and thus differentiable almost everywhere. Therefore, the right-hand side of (3.20) is well defined for almost every t. As a consequence, if $e \in \mathcal{O}^v$ for some $v \in \mathcal{V}$, then Q^e is also Lipschitz continuous.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

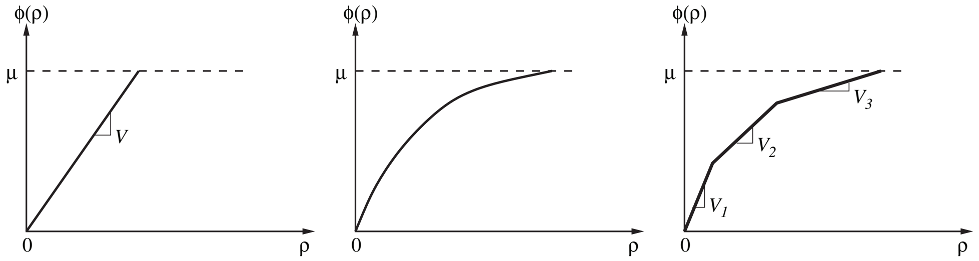

Fig. 2. *Examples of concave and increasing flux function $\phi(\cdot)$, where ρ denotes density, μ denotes flow capacity, and V, V_1, V_2, V_3 denote processing speeds. Left: the linear flux function employed in this paper. Middle: the smooth flux function which leads to a general continuous optimization problem. Right: the piecewise affine flux function which leads to an optimization problem over a finite set with cardinality equal to the number of affine pieces.*

The above system expresses the continuous-time solution of the network model proposed in [23], where a system of coupling PDEs and ODEs were initially employed to describe the dynamics. As we shall see later, besides the closed-form solution, (3.22) also yields better solution precision than the finite-difference schemes under minor condition.

3.4. Some model extensions. The conservation law (2.1) used to represent the dynamic on a single processor is an advection equation and assumes a constant speed of all products regardless of their position or density. Such simple dynamics may not be quite realistic in certain applications, and it is the purpose of this section to introduce several model extensions and the corresponding variational methods.

3.4.1. Location-dependent parameters. A processor may comprise a sequence of individual stages, each one with a different processing speed or flow capacity. Such heterogeneity of parameters within a single processor may be captured by the following conservation law:

$$\partial_t \rho(t,\, x) + \partial_x \min\left\{V(x)\rho(t,\, x),\, \mu(x)\right\} = 0, \qquad (t,\, x) \in [0,\, T] \times [a,\, b], \tag{3.23}$$

where, unlike (2.1), the processing speed and capacity are dependent on the spatial parameter $x \in [a,\, b]$. A conservation law with an x-dependent flux function is often used by traffic analysts to model road heterogeneity such as lane change, curvature, road condition, etc. [32]. We note that there exist variational methods for the conservation law (Hamilton–Jacobi equation) with x-dependent flux function (Hamiltonian); see, for example, [17]. In this paper we propose a simple treatment of (3.23) by assuming piecewise constant approximations of $V(x)$ and $\mu(x)$. By doing so, we decompose the dynamic on the processor into a set of homogeneous ones, each governed by an ODE for the queue and a PDE of the form (2.1) with constant processing speed and capacity. Topologically, this means that we break each link in the network into smaller ones. The variational approach proposed in this paper will apply to such a new network.

3.4.2. Density-dependent processing speed. If the local processing speed is dependent on the density of products, the conservation law becomes

$$\partial_t \rho(t,\, x) + \partial_x \left(V(\rho(t,\, x))\rho(t,\, x)\right) = 0, \qquad (t,\, x) \in [0,\, T] \times [a,\, b], \tag{3.24}$$

which is more in line with the fluid-like traffic dynamics [37, 41]. In particular, it is assumed that the map $\phi(\cdot):\, \rho \mapsto V(\rho)\rho$ is concave and increasing. See Figure 2. Notice

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

that, unlike traffic flow models, backward propagation of kinematic waves is not considered here; therefore, there is no need to include the monotonically decreasing part of the *fundamental diagram* [12, 13, 37]. Similar techniques based on switching the roles of t and x, as we demonstrate in this paper, will apply to conservation law (3.24) and the corresponding Hamilton–Jacobi equation, and the variational solution representation can be consequently obtained. Due to space limitation, we omit technical details and refer the reader to [7] for a specific discussion. Notice that for general functional forms of $\phi(\cdot)$, the variational formulation amounts to a nonconvex optimization problem, for which closed-form solutions are hardly available. In a special case where $\phi(\cdot)$ is piecewise affine, it can be shown that the variational formulation becomes an optimization problem over a set of finite elements; this is directly related to the fact that there is only a finite number of wave speeds. See the right part of Figure 2.

4. Optimizing network flow.

This section defines and solves the following network optimization problem: given a general supply chain network as in Definition 2.1, with dynamics at processors and buffer queues described by the variational formulation (3.20)–(3.22), assume that at each dispersive node one can determine the allocation of flows by controlling $A^{v,\,e}(t)$, so that the network is optimized subject to some other constraints. A general objective functional of such optimization problem is

$$\min_{A^{v,\,e}(t)} \sum_{e\in\mathcal{A}} \mathcal{F}^e(Q^e,\, W^e). \tag{4.1}$$

The continuous-time optimization problem is given by (4.1), subject to (3.20), (3.21), and (3.22). It is shown in the next subsection that by properly time-discretize the system and by introducing binary variables, the optimization problem can be formulated as an MIP.

4.1. Time discretization.

We adapt the "discretize-then-optimize" strategy by time-discretizing system (3.20)–(3.22) and then optimizing the discrete system via mathematical programming techniques. To do this, we prescribe a time grid $\{t_i\}_{i=0}^N$ with step size h and adapt the following notation: $Q_i^e \doteq Q^e(t_i)$, $W_i^e \doteq W^e(t_i)$, $A_i^{v,\,e} \doteq A^{v,\,e}(t_i)$, $\Delta^e \doteq \lceil \frac{L^e}{V^e h} \rceil$.

To avoid differentiation arising from (3.20), we invoke the following lemma.

Lemma 4.1. *Constraint* (3.20) *can be equivalently expressed as*

$$\sum_{e\in\mathcal{O}^v} Q^e(t) = \sum_{\bar{e}\in\mathcal{I}^v} W^{\bar{e}}(t), \tag{4.2}$$

$$Q^e(t_1) \le Q^e(t_2) \qquad \forall\ t_1 < t_2, \quad e\in\mathcal{O}^v. \tag{4.3}$$

Proof. (3.20)⇒(4.2), (4.3). In view of (3.19), summing up (3.20) over $e\in\mathcal{O}^v$ and integrating over time give rise to (4.2). Moreover,

$$\frac{d}{dt}Q^e(t) = A^{v,\,e}(t)\sum_{\bar{e}\in\mathcal{I}^v}\frac{d}{dt}W^{\bar{e}}(t) \ge 0 \qquad \text{for almost every } t;$$

this establishes (4.3).

(4.2), (4.3)⇒(3.20). Differentiating (4.2) gives

$$\sum_{e\in\mathcal{O}^v}\frac{d}{dt}Q^e(t) = \sum_{\bar{e}\in\mathcal{I}^v}\frac{d}{dt}W^{\bar{e}}(t).$$

Define $A^{v,\,e}(t) \doteq \frac{d}{dt}Q^e(t)/\sum_{\bar{e}\in\mathcal{I}} W^{\bar{e}}(t)$, $e\in\mathcal{O}^v$, and then it is straightforward to verify that (3.19) holds. □

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Note that (4.2) and (4.3) guarantee conservation of flux across the junction and nonnegativity of the flux. After applying Lemma 4.1, information of the allocation rates $A^{v,\,e}$ is implicitly described by Q^e; and it is a matter of simple calculation to recover such information from the solution.

The discrete-time forms of (3.21), (3.22) are

$$\hat{Q}^e_0 = 0, \qquad \hat{Q}^e_i = q^e_0 + Q^e_i, \qquad 1 \le i \le N, \tag{4.4}$$

$$W^e_i = \begin{cases} 0, & i < \Delta^e, \\ \min\limits_{0 \le j \le i-\Delta^e} \left\{ \hat{Q}^e_j - \mu^e t_j \right\} + (t_i - L^e/V^e)\mu^e, & i \ge \Delta^e. \end{cases} \tag{4.5}$$

We introduce the variables R^e_i for $e \in \mathcal{A}$, $\Delta^e \le i \le N$, defined recursively via the following:

$$R^e_i = \begin{cases} \hat{Q}^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e}, & i = \Delta^e, \\ \min\left\{ R^e_{i-1}, \quad \hat{Q}^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e} \right\}, & \Delta^e < \ i \le N + \Delta^e. \end{cases} \tag{4.6}$$

Then the second equation of (4.5) becomes

$$W^e_i = R^e_i + (t_i - L^e/V^e)\mu^e, \qquad i \ge \Delta^e. \tag{4.7}$$

By virtue of this new variable R^e_i and (4.6), for each time step i we need to evaluate the "min" function only once, thus reducing the number of variables and operations needed in the problem. In order to linearize the constraints involving the "min" operator, we introduce the binary variables $\beta^e_i \in \{0,\,1\}$, $i = \Delta^e + 1, \dots, \Delta^e + N$, and equivalently write condition (4.6) as

$$R^e_{\Delta^e} = \hat{Q}^e_0 - \mu^e t_0, \quad \begin{cases} R^e_{i-1} + (\beta^e_i - 1)M \le R^e_i \le R^e_{i-1}, \\ \hat{Q}^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e} - M\beta^e_i \le R^e_i \le \hat{Q}^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e}, \end{cases} \tag{4.8}$$

where M is chosen to be a sufficiently large constant. We are now ready to state the main result of this section.

THEOREM 4.2 (MIP formulation of network optimization problem). *Consider a supply chain network as in Definition* 2.1 *with parameters* L^e, V^e, μ^e, *and initial conditions* $q^e(0) = q^e_0$ $\forall e \in \mathcal{A}$. *Define a linear objective function* $\mathcal{F}(Q^e,\, W^e,\, R^e)$. *Given a time grid* $0 = t_0 < t_1 \ldots < t_N = T$ *with step size* h, *the network optimization problem* (4.1), (3.20), (3.21), *and* (3.22) *can be formulated as the following MIP:*

$$\min_{Q^e_i,\, W^e_i,\, R^e_i} \mathcal{F}(Q^e_i,\, W^e_i,\, R^e_i) \tag{4.9}$$

subject to

$$\sum_{e \in \mathcal{O}^v} Q^e_i = \sum_{\bar{e} \in \mathcal{I}^v} W^{\bar{e}}_i, \qquad Q^e_{i-1} \le Q^e_i, \qquad 1 \le i \le N, \tag{4.10}$$

$$R^e_{\Delta^e} = -\mu^e t_0, \quad R^e_{i-1} + (\beta^e_i - 1)M \le R^e_i \le R^e_{i-1}, \quad \Delta^e + 1 \le i \le \Delta^e + N, \tag{4.11}$$

(4.12)
$$q^e_0 + Q^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e} - M\beta^e_i \le R^e_i \le q^e_0 + Q^e_{i-\Delta^e} - \mu^e t_{i-\Delta^e}, \quad \Delta^e + 1 \le i \le \Delta^e + N$$

$$W^e_i = R^e_i + (t_i - L^e/V^e)\mu^e, \qquad \Delta^e \le i \le N, \tag{4.13}$$

$$\Delta^e \doteq \left\lceil \frac{L^e}{V^e h} \right\rceil, \qquad \beta^e_i \in \{0,\,1\}, \qquad e \in \mathcal{A}, \qquad v \in \mathcal{V}, \tag{4.14}$$

where M *is a sufficiently large constant.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Proof. We have already established (4.10) from Lemma 4.1 and (4.13) from (4.7). (4.11) and (4.12) are immediate consequences of (4.4) and (4.8). ▯

Remark 4.3. If $|\mathcal{O}^v| = 1$, i.e., node v is nondispersive, the nonnegative flow condition $Q_{i-1}^e \leq Q_i^e$ in (4.10) is automatically satisfied since $W_i^{\bar{e}}$, $\bar{e} \in \mathcal{I}^v$ are guaranteed to be nondecreasing with respect to the time step i, due to Proposition 3.5. Therefore, the nonnegative flow constraints can be dropped for such a node.

Remark 4.4. The objective function (4.9) is, in its own form, an arbitrary real-valued function of Q_i^e, W_i^e, and R_i^e $\forall 0 \leq i \leq N$, $e \in \mathcal{A}$. Throughout this paper, the objective function is chosen to be linear for ease of computation; see section 5.1 and 5.6. However, in a manufacturing environment with different cost scenarios, the objective function is often nonlinear, nonconvex, and even nonsmooth. If the objective function is smooth and convex, then the resulting program can still be solved efficiently with commercial software. If the objective function is nonsmooth but is convex and admits well-defined subgradients, then using branch and bound, the relaxed problem can still be handled relatively well by the ellipsoidal method, analytic center cutting-plane method, etc.

4.2. Model extensions.

We provide some discussion on two straightforward extensions of the proposed MIP formulation.

4.2.1. Finite buffers.

In a realistic manufacturing process, the capacity of the buffering queue is usually limited. Articulation of this type of condition in the optimization procedure requires the queue size $q^e(t)$ to be expressed explicitly. Recall that the queue size can be written as $q^e(t) = Q^e(t) - \overline{U}^e(t)$, where $\overline{U}^e(t)$ is given by (3.4). Thus we have

$$q^e(t) = Q^e(t) - \overline{U}^e(t) = (Q^e(t) - \mu^e t) - \inf_{\tau \leq t} \left\{Q^e(\tau) - \mu^e \tau\right\}. \tag{4.15}$$

The discrete-time expression for the queue reads

$$q_i^e = Q_i^e - \mu^e t_i - \min_{0 \leq j \leq i} \left\{Q_j^e - \mu^e t_j\right\} = Q_i^e - \mu^e t_i - R_{i+\Delta^e}^e, \qquad i = 0, \ldots, N. \tag{4.16}$$

The finite buffer constraint can then be implemented in the MIP by adding the following linear constraints:

$$Q_i^e - \mu^e t_i - R_{i+\Delta^e}^e \leq C_q^e, \qquad e \in \mathcal{A}, \quad 0 \leq i \leq N, \tag{4.17}$$

where C_q^e denotes the buffer queue capacity on link e.

4.2.2. Inventory cost.

In certain cases the handling of products in the buffer queue may incur additional costs; these may include fixed costs (e.g., warehouse rents) and variable costs (e.g., product degradation). Such a situation can be easily handled in our framework by adding the following cost function to the objective function:

$$\sum_{e \in \mathcal{A}} \alpha_q^e \sum_{i=0}^{N} (Q_i^e - \mu^e t_i - R_{i+\Delta^e}^e), \qquad \alpha_q^e \geq 0, \quad e \in \mathcal{A}, \tag{4.18}$$

where constant α_q^e measures the cost per unit storage. With the revised objective function, the decision variables may further include the network inflow profiles which allow the buffer queues to remain minimum, while maintaining a maximum throughput of the supply chain network.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

4.3. Comparison with existing approaches. The MIP (4.9)–(4.14) differs from the one proposed by [22] in a number of ways. First, the governing equations for the system in our paper are derived from a variational perspective, and the main variables are cumulative product counts, which are related to each other via the Hamilton–Jacobi equations and the junction conditions. In contrast to the model of [23, 24], the dynamics of the processor and the buffering queue are simultaneously handled by the Lax formula. Our approach accepts discontinuous boundary datum $Q(\cdot)$, which implies the presence of a delta-distribution in the flow; such a situation cannot be handled by the conservation law without a separate modeling of the queue. This intuitively explains why in the variational formulation, an explicit treatment of the queue is unnecessary. The Lax formula not only avoids the numerical issue arising from the discontinuity in the ODE (2.2)–(2.3) (a detailed study of such issue will be provided in section 5.5) but also facilitates efficient simulation and optimization of the network by reducing the number of variables needed and memory usage, due to the grid-free nature of the algorithm. The conservation law-based models, on the other hand, rely on a two-dimensional grid and are restricted by the CFL conditions, and thus could potentially lead to large systems which are computationally expensive.

Next we compare the number of variables used in the MIP (4.9)–(4.14) with those presented in [22] with a two-point spatial discretization. Assume the numbers of arcs in the network is $|\mathcal{A}|$ and that the number of time intervals is N. Our proposed MIP has $3N|\mathcal{A}|$ real variables and $N|\mathcal{A}|$ binary variables; the MIP of [22] has the same number of real and binary variables. However, in the latter approach, a two-point spatial discretization may be too coarse to properly represent the PDE. If the spatial grid is to be refined by a factor of n, the CFL condition (1.2) implies that the number of variables needed for the spatiotemporal grid will increase by a factor of n^2, and the number of binary variables will also increase by a factor of n. This fact reveals a trade-off between numerical accuracy and computational efficiency for the conservation law models and the MIP built upon them. The variational approach, on the other hand, does not invoke spatial discretization and has no restrictions on the time step. Therefore, to achieve the same level of numerical precision, our proposed MIP is arguably much smaller in size compared to that of [22] and can be computed more efficiently.

5. Numerical example. In this section, we conduct a sequence of numerical studies of the variational approach and the resulting MIP, with modeling extensions discussed in section 4.2. Consider a test network shown in Figure 3, which consists of seven arcs (processors), $a - g$, and four nodes, $1 - 4$. The primary control variables are the time-varying flow allocation rate $A^{1,\,b}(t)$ and $A^{2,\,e}(t)$ at dispersive nodes 1 and 2, respectively. The time horizon is fixed to be [0, 10]. Network parameters employed in our numerical test are shown in Table 1. In addition, we assume zero initial conditions: $q_0^e = 0$, $\rho_0^e(x) = 0 \ \forall e \in \mathcal{A}$.

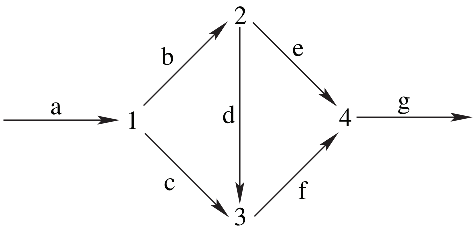

Fig. 3. *A test network consisting of seven arcs and four nodes.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Table 1
Processor parameters.

Processor	a	b	c	d	e	f	g
L^e	2	2	2	2	2	2	2
V^e	2	1	2	4	2	2	2
μ^e	15	6	5	4	3.5	8	14

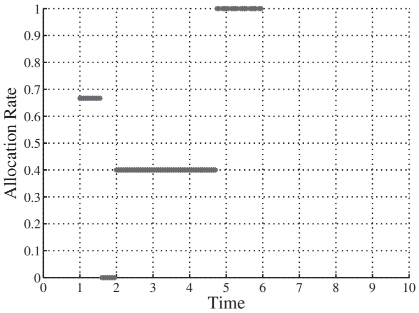

Fig. 4. *Without buffer size constraints: optimal allocation rate* $A^{1,\,b}$.

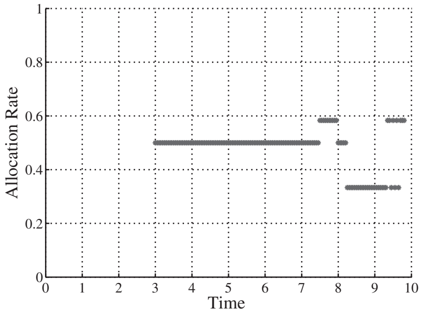

Fig. 5. *Without buffer size constraints: optimal allocation rate* $A^{2,\,e}$.

All MIPs presented in this section were solved with ILOG Cplex 12.1.0 [31] on the HPC platform with Intel Xeon X5675 Six-Core 3.06 GHz processor, provided by the Penn State High Performance Computing Systems [40].

5.1. Without buffer size constraints. As our first scenario, we assume that the flow $f^a(t)$ into the network through processor a is fixed and given by

$$f^a(t) = \begin{cases} 37.5, & 0 \leq t \leq 2, \\ 0 & \text{otherwise.} \end{cases}$$

It is assumed that all the buffers have infinite capacity. The objective is to maximize the total throughput of the network within the time horizon, namely,

$$W^g(10), \tag{5.1}$$

where the notation has the same meaning as before. The optimal allocation rates $A^{1,\,b}(t)$, $A^{2,\,e}(t)$ are computed from the MIP formulation introduced by Theorem 4.2 and are illustrated graphically in Figures 4 and 5. In the optimized network flow profile, only processors a, b, and c have nonempty buffering queues associated with them; this is shown in Figure 6. The optimal value of the throughput $W^g(10)$ is 58.75.

5.2. With finite buffer constraints. We consider additional buffer capacity constraints in the MIP, namely, $q_i^b \leq 10,\ \ q_i^c \leq 10,\ \ i = 0, \ldots, N$. Such constraints are implemented by inserting inequality constraints (4.17) to the MIP. Figure 7 shows the time-dependent queue sizes in the new optimal network flow profile. Notice that the queue associated with processor a cannot be controlled since the inflow $f^a(\cdot)$ is fixed.

In the presence of the finite-buffer constraints, the adjusted allocation rates are depicted in Figures 8 and 9. Interestingly, the optimal value of $W^g(10)$ remains the same as the previous case, without buffer size constraints. This shows the nonuniqueness of the optimal solution of the MIP.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

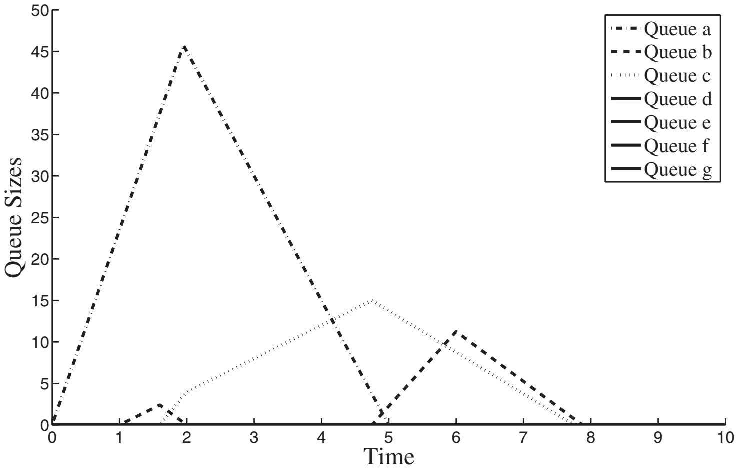

FIG. 6. *Without buffer size constraints: queues upstream of each processor.*

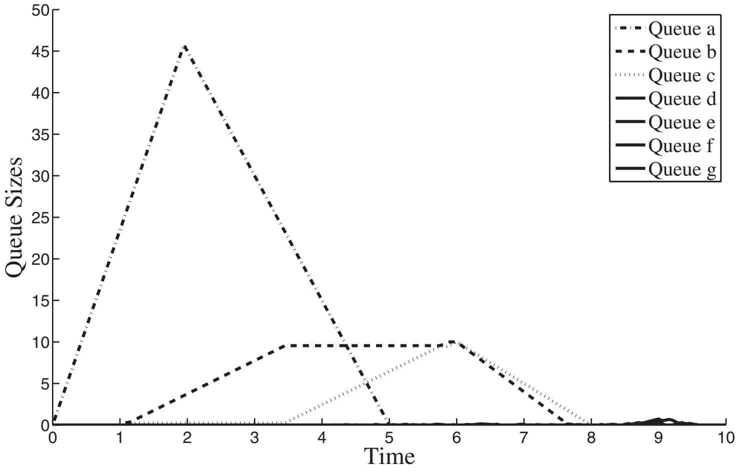

FIG. 7. *With buffer size constraints: queue upstream of each processor.*

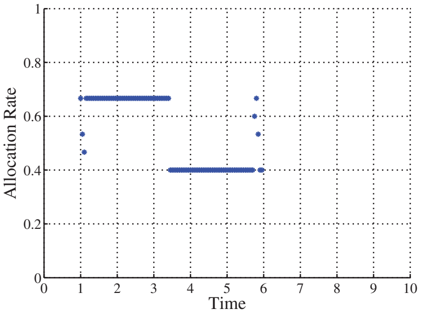

FIG. 8. *With buffer size constraints: optimal allocation rate $A^{1,b}$.*

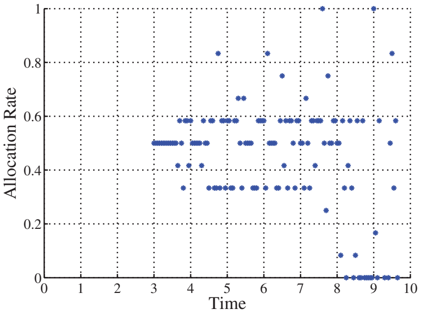

FIG. 9. *With buffer size constraints: optimal allocation rate $A^{2,e}$.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

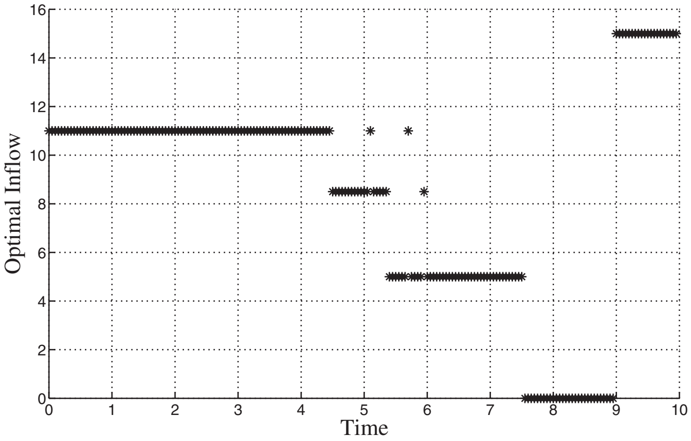

FIG. 10. *Minimum queuing case: optimal inflow* $f^a(t)$.

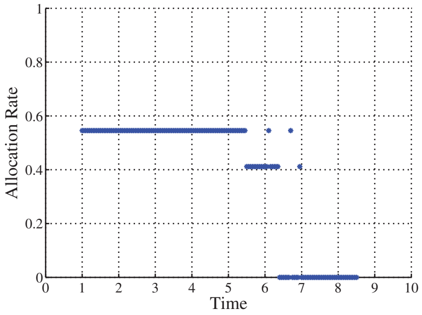

FIG. 11. *Minimum queuing case: allocation rate* $A^{1,\,b}$.

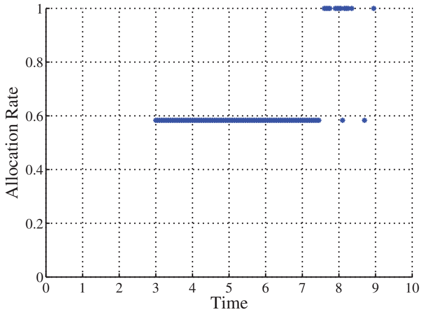

FIG. 12. *Minimum queuing case: allocation rate* $A^{2,\,e}$.

5.3. Minimizing queuing. In this example, we wish to minimize the queuing in the network, by controlling not only the allocation rates but also the inflow profile $f^a(t)$ of the network. In view of our discussion in section 4.2.2, we consider the following revised objective function:

$$\max_{Q_i^a,\,A_i^{v,\,e}} \left\{ W^g(10) - \sum_{e\in\mathcal{A}} c_q^e \sum_{i=0}^{N} (Q_i^e - R_{i+\Delta^e}^e) \right\}, \tag{5.2}$$

where the unit queuing cost c_q^e is set to be equal to one $\forall e \in \mathcal{A}$. By employing the above objective function, we wish to maximize the throughput of the network while keeping the queuing at a minimum level. The resulting optimal inflow from the MIP is shown in Figure 10; the corresponding optimal allocation rates at vertices 1 and 2 are illustrated in Figures 11 and 12, respectively. Under such controls on the inflow and the allocation rates, the queues associated with all the processors remain empty throughout the time horizon.

The final throughput $W^g(10)$ is again 58.75.

5.4. Solution quality. We will analyze the result of our proposed MIP in terms of numerical error and convergence, which will be compare with the finite difference schemes [16, 22, 23, 24]. We use the same network and parameters as before and assume infinite buffer capacities. The time horizon of interest is set to be [0, 80]. The

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

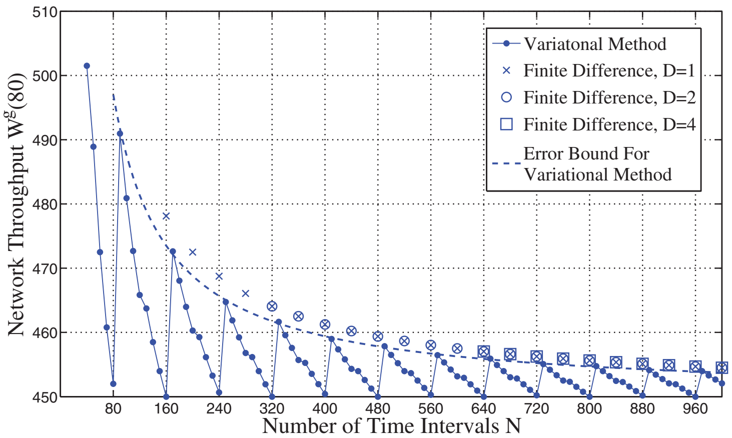

FIG. 13. *Optimal output for different choices of time grid.*

objective is to maximize the throughput W_N^g. The inflow profile is

$$f^a(t) = \begin{cases} 45, & 0 \leq t < 10, \\ 0, & 10 \leq t \leq 80. \end{cases} \tag{5.3}$$

Note that the integration of $f^a(t)$ on the time horizon is 450; it is natural to expect that the total throughput $W^g(80)$ is the same provided the time horizon is large enough. We run instances of the MIP (4.9)–(4.14) for $N = 40, 50, 60, \ldots, 1000$, where N is the number of time intervals. The optimal throughputs produced by these programs are plotted in Figure 13.

Before we explain the figure, let us recall from Proposition 3.5 that the discrete-time Lax formula is exact provided that the throughput time L^e/V^e, $e \in \mathcal{A}$ is a multiple of the time step h, that is, when $N = 160, 320, 480, 640, 800, 960$ in our example. Otherwise, the error is given by estimate (3.18):

$$0 \leq W_N^e - W^e(80) \leq \left(\left\lceil \frac{L^e}{V^e h} \right\rceil - \frac{L^e}{V^e h} \right) h\mu^e, \qquad e \in \mathcal{A}. \tag{5.4}$$

Equation (5.4) suggests that (1) the discrete-time numerical values overestimate the actual value of $W^g(80)$, which is 450 and (2) the error displays an oscillatory pattern with damping, as N increases. The oscillation is caused by the discontinuous function $\lceil \frac{L}{Vh} \rceil$ while the damping is due to the factor h. Notice that the right-hand side of error estimate (5.4) can be further relaxed to $h\mu$. It is not difficult to extend the error estimates for a single processor to a network, noticing that the error in the cumulative production curves adds up linearly through one processor (arc) to another:

$$0 \leq W_N^g - W^g(80) \leq h \max_{p \in \mathcal{P}} \sum_{e \in p} \mu^e, \tag{5.5}$$

where $p = \{e_1, e_2, \ldots, e_m\}$ is any viable path of product flow, $e_i \in \mathcal{A}$, and $W^g(80)$ is the exact value of throughput. The right-hand side of (5.5) is shown in Figure 13 against different values of N.

For comparison purposes, we also implement the MIP proposed by [22], which is based on a finite difference scheme. We experiment with different numbers of spatial intervals D for the discretization, for example, $D = 1$ means the two-point

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

discretization. The optimal throughputs are plotted in Figure 13 as well. Notice that when D increases, the value of N must also increase due to the CFL condition. The numerical results summarized in Figure 13 leads to two observations: (1) the values of $D \in \{1, 2, 4\}$ seem to have no effect on the error as long as the CFL condition is satisfied, and (2) in terms of solution precision, the variational approach outperforms the finite difference scheme even in the worst case scenario.

The above numerical experiment suggests that in order to maintain the same level of numerical error, the size of our proposed MIP can be significantly smaller than the finite-difference approach. For example, to keep the error of W_N^g under 1%, that is, $W_N^g \in [450, 454.5]$, the finite-difference approach requires a time grid of approximately 1000 points and the same amount of binary variables, while with the variational approach, a time grid of 80 or 160 points is sufficient to achieve such solution precision. This observation is clearly supported by Figure 13.

5.5. A case study of a smoothed out version of (2.2)–(2.3). Fügenschuh et al. [22] proposed a smoothed out version of the ODE (2.2)–(2.3) in order to avoid the numerical difficulties caused by the discontinuous right-hand side. The modified ODE reads as follows:

$$\frac{d}{dt}q^e(t) = \overline{u}^e(t) - f^e\big(\rho^e(t,\, a^e)\big), \tag{5.6}$$

$$f^e\big(\rho^e(t,\, a^e)\big) = \min\left\{\mu^e,\, \frac{q^e(t)}{\varepsilon}\right\}, \tag{5.7}$$

where $\varepsilon > 0$ is a smoothing parameter. The modified ODE (5.6)–(5.7) is a practical representation of the original dynamic since it makes the right-hand side of the ODE continuous, and the solution approximates the one of (2.2)–(2.3) well in most cases. To ensure numerical stability of the finite difference discretization, a stiffness condition (see [22]) requires that

$$\Delta t \leq \varepsilon, \tag{5.8}$$

where Δt denotes the time step size.

In this case study, we compare the solutions of (5.6)–(5.7) and the variational formulation to illustrate that the smoothed out version of the original ODE could still yield an ill-behaved solution in certain case.

Let us return to the example in section 5.1 and set the network inflow to be

$$f^a(t) = \begin{cases} 14, & 0 \leq t < 10, \\ 0, & 10 \leq t \leq 80. \end{cases} \tag{5.9}$$

The MIP proposed by [22] employs the smoothed (5.6)–(5.7) for the queue dynamics. Such MIP is solved with $N = 600$, $D = 1$, where N denotes the number of time steps and D denotes the number of spatial intervals for each processor. In other words, the two-point upwind discretization for the conservation law is used. The inflow, exit flow, and queue size of processor a are shown together on the left part of Figure 14. We notice that a discretization of (5.6)–(5.7) yields a nonzero queue even though the inflow is strictly below the processor capacity $\mu^a = 15$. It turns out that such a nonphysical queue is caused by the smoothing parameter ε and can be reduced by choosing smaller ε; however, this once again implies a tradeoff between numerical accuracy and computational burden, due to the stiffness condition (5.8).

The variational method, on the other hand, handles the same problem well with $N = 600$. The right part of Figure 14 shows that the queue stays zero and the exit

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

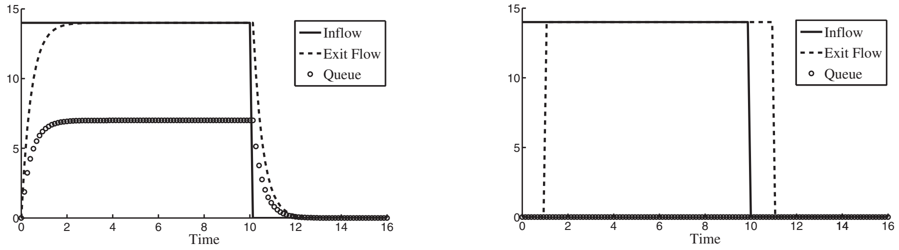

FIG. 14. *A comparison of the smoothed out version of* (2.2)–(2.3) *and the variational approach. Left: solution on processor a obtained by the finite-difference discretization of* (5.6)–(5.7) *with* $\varepsilon = 0.5$*. Right: solution on processor a given by the variational approach.*

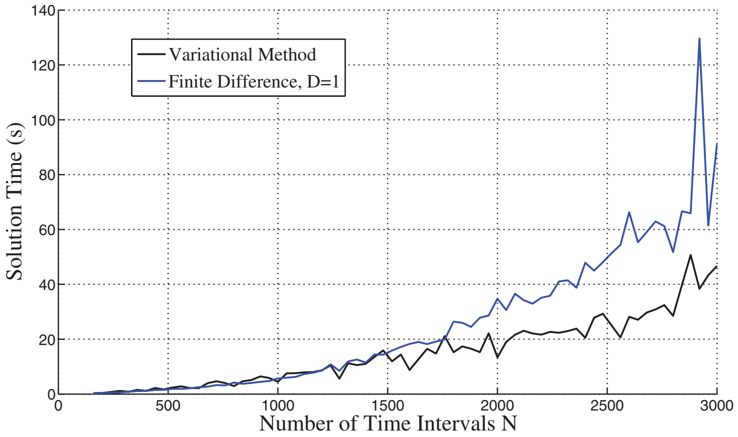

FIG. 15. *Computational times of the two MIPs.*

flow is a a simple time shift of the inflow profile, which is consistent with the physics of the model.

5.6. Solution time. As our final test, the computational times of both MIPs are recorded for the same network optimization problem as in section 5.1 with the following network inflow:

$$f^a(t) = \begin{cases} 30, & 0 \le t \le 5, \\ 0, & 5 < t \le 10. \end{cases}$$

For the MIP of [22], we employ a coarse two-point spatial discretization. For both MIP formulations, the same objective function is chosen to be

$$\max \sum_{i=0}^{N} \frac{w_i^g}{1+t_i}, \tag{5.10}$$

where w_i^g is the exit flow on processor g at time t_i. Choosing such an objective function ensures that the network throughput is maximized at every instance of time; in other words, the products are handled in a way such that they exit the network as early as possible.

The two MIPs are solved with N, the number of time intervals, ranging from 160 to 3000. The corresponding solution times are summarized in Figure 15. Although

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

the two MIPs are similar in size, as we demonstrated in section 4.3, the solution times of the proposed MIP problem is significantly lower than the other MIP. In addition, we observe a nearly linear growth of the computational time when N increases for the variational approach. Possible mechanisms for causing such a significant difference in solution times for large scale problems, involving either internal structure of the discretization or specific settings of the branch-and-bound algorithm, are currently under investigation.

6. Conclusion. This paper proposes a variational method for the modeling, computation, and optimization of a class of continuous supply chain networks. Such supply chain networks are investigated by [23] and can be formulated as a system of partial differential equations and ordinary differential equations. The main contribution made in this paper is an explicit solution representation of the dynamics on both the buffer queue and the processor. Our methodological framework is based on a Hamilton–Jacobi equation and a variational method known as the Lax formula. The closed-form solution is derived in both continuous and discrete time; and the latter leads to an algorithm with provable error estimates. The proposed computational method is grid-free in the sense that it does not require a spatial discretization. Notably, the algorithm requires less computational effort and induces less numerical error than the finite difference method proposed for the coupling PDE and ODE [22]. We also propose, based on the variational formulation, a mixed integer programming approach for the optimization of continuous supply chain networks. As we demonstrate in a series of numerical studies, the appropriate choice of the time grid could lead to a significantly reduced and even zero error, when compared with the MIP of [22]. We also show that the proposed MIP requires much less computational effort than the one based on the PDE-ODE system in order to properly represent the dynamics.

It is worth mentioning that the continuous supply chain model, expressed by an ODE for the buffer queue and a PDE for the processor, is in many ways similar to the famous Vickrey model [42] for dynamic traffic flows. Applications of the variational approach in the venue of traffic modeling are presented in [28] and [29].

Future extensions of the variational formulation will be focused on the modeling of multicommodity supply chain network with more realistic features. To do so, it is desirable to consider inhomogeneous Hamilton–Jacobi equations to account for nonconstant processing time. More sophisticated junction models need to be introduced as well to treat product flows with given origins and destinations.

REFERENCES

[1] D. Armbruster, P. Degond, and C. Ringhofer, *Kinetic and fluid models for supply chains supporting policy attributes*, Bull. Inst. Math. Acad. Sin. (N.S.), 2 (2007), pp. 433–460.

[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 (2006), pp. 896–920.

[3] D. Armbruster, D. Marthaler, and C. Ringhofer, *Kinetic and fluid model hierarchies for supply chains*, Multiscale Model. Simul., 2 (2003), pp. 43–61.

[4] D. Armbruster, C. De Beer, M. Freitag, T. Jagalski, and C. Ringhofer, *Autonomous control of production networks using a pheromone approach*, Phys. A, 363 (2006), pp. 104–114.

[5] J. Banks, J. Carson, and B. Nelson, *Discrete Event System Simulation*, Prentice-Hall, Englewood Cliffs, NJ, 1999.

[6] A. Bressan, *Hyperbolic Systems of Conservation Laws: The One Dimensional Cauchy Problem*, Oxford University Press, Oxford, 2000.

[7] A. Bressan and K. Han, *Optima and equilibria for a model of traffic flow*, SIAM J. Math. Anal., 43 (2011), pp. 2384–2417.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[8] A. Bressan and K. Han, *Nash equilibria for a model of traffic flow with several groups of drivers*, ESAIM Control Optim. Calc. Var., 18 (2012), pp. 969–986.

[9] C. G. Claudel and A. M. Bayen, *Lax–Hopf based incorporation of internal boundary conditions into Hamilton–Jacobi equation. Part* I: *Theory*, IEEE Trans. Automat. Control, 55 (2010), pp. 1142–1157.

[10] C. G. Claudel and A. M. Bayen, *Lax–Hopf based incorporation of internal boundary conditions into Hamilton–Jacobi equation. Part* II: *Computational methods*, IEEE Trans. Automat. Control, 55 (2012), pp. 1158–1174.

[11] J. M. Coron, M. Kawski, and Z. Q. Wang, *Analysis of a conservation law modeling a highly re-entrant manufacturing system*, Discrete Contin. Dyn. Syst. Ser. B, 14 (2010), pp. 1337–1359.

[12] C. F. Daganzo, *The cell transmission model: A simple dynamic representation of highway traffic*, Trans. Res. B, 28 (1994), pp. 269–287.

[13] C. F. Daganzo, *The cell transmission model, part* II: *Network traffic*, Trans. Res. B, 29 (1995), pp. 79–93.

[14] C. F. Daganzo, *A variational formulation of kinematic waves: Basic theory and complex boundary conditions*, Trans. Res. B, 39 (2005), pp. 187–196.

[15] C. D'Apice and R. Manzo, *A fluid dynamic model for supply chains*, Netw. Heterog. Media, 1 (2006), pp. 379–398.

[16] C. D'Apice, G. Simone, M. Herty, and B. Piccoli, *Modeling, Simulation and Optimization of Supply Chains, A Continuous Approach*, SIAM, Philadelphia, 2010.

[17] I. C. Dolcetta, *Representations of solutions of Hamilton–Jacobi Equations*, in Progress in Nonlinear Differential Equations and Applications 54, Birkhauser-Verlag, Boston, MA, pp. 74–90.

[18] L. C. Evans, *Partial Differential Equations*, 2nd ed., AMS, Providence, RI, 2010.

[19] P. Le Floch, *Explicit formula for scalar non-linear conservation laws with boundary condition*, Math. Models Appl. Sci., 10 (1988), pp. 265–287.

[20] T. L. Friesz, K. Han, P. A. Neto, A. Meimand, and T. Yao, *Dynamic user equilibrium based on a hydrodynamic model*, Trans. Res. B, 47 (2013), pp. 102–126.

[21] A. Fügenschuh, M. Herty, A. Klar, and A. Martin, *Combinatorial and continuous models for the optimization of traffic flows on networks*, SIAM J. Optim., 16 (2005), pp. 1155–1176.

[22] A. Fügenschuh, S. Göttlich, M. Herty, A. Klar, and A. Martin, *A discrete optimization approach to large scale supply networks based on partial differential equations*, SIAM J. Sci. Comput., 30 (2008), pp. 1490–1507.

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

[24] S. Göttlich, M. Herty, and A. Klar, *Modeling and optimization of supply chains on complex networks*, Commun. Math. Sci., 4 (2006), pp. 315–330.

[25] M. Garavello and B. Piccoli, *Traffic Flow on Networks. Conservation Laws Models*, AIMS Ser. Appl. Math., Springfield, MO, 2006.

[26] H. Lo, *A novel traffic signal control formulation*, Trans. Res. A, 33 (1999), pp. 433–448.

[27] K. Han, T. L. Friesz, and T. Yao, *A link-based mixed integer LP approach for adaptive traffic signal control*, arXiv:1211.4625, 2012.

[28] K. Han, T. L. Friesz, and T. Yao, *A partial differential equation formulation of Vickrey's bottleneck model, Part* I: *Methodology and theoretical analysis*, Trans. Res. B, 49 (2013), pp. 55–74.

[29] K. Han, T. L. Friesz, and T. Yao, *A partial differential equation formulation of Vickrey's bottleneck model, Part* II: *Numerical analysis and computation*, Trans. Res. B, 49 (2013), pp. 75–93.

[30] M. Herty, A. Klar, and B. Piccoli, *Existence of solutions for supply chain models based on partial differential equations*, SIAM J. Math. Anal., 39 (2007), pp. 160–173.

[31] ILOG CPLEX Division, http://www.cplex.com.

[32] W.-L. Jin and H. M. Zhang, *The inhomogeneous kinematic wave traffic flow model as a resonant nonlinear system*, Trans. Sci., 37 (2003), pp. 294–311.

[33] A. Kotsialos, *A hydrodynamic modelling framework for production networks*, Comput. Manag. Sci., 7 (2010), pp. 61–83.

[34] M. La Marca, D. Armbruster, M. Herty, and C. Ringhofer, *Control of continuum models of production systems*, IEEE Trans. Automat. Control, 55 (2010), pp. 2511–2526.

[35] P. D. Lax, *Hyperbolic systems of conservation laws* II, Comm. Pure Appl. Math., 10 (1957), pp. 537–566.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[36] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, Basel, Switzerland, 1992.

[37] M. J. Lighthill and J. B. Whitham, *On kinematic waves* II: *A theory of traffic flow in long crowded roads*, Proc. Roy. Soc., A229 (1955), pp. 317–345.

[38] G. F. Newell, *A simplified theory of kinematic waves in highway traffic*, Transport. Res. B, 27 (1993), pp. 281–313.

[39] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, Springer Ser. Oper. Res. Financ. Eng., Springer, New York, 2006.

[40] Penn State Research Computing and Cyberinfrastructure, http://rcc.its.psu.edu.

[41] P. I. Richards, *Shockwaves on the highway*, Oper. Res., 4 (1956), pp. 42–51.

[42] W. S. Vickrey, *Congestion theory and transport investment*, Amer. Econom. Rev., 59 (1969), pp. 251–261.

[43] S. Voss and D. L. Woodruff, *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*, 2nd ed., Springer, New York, 2006.

[44] H. P. Wiendahl, *Load-Oriented Manufacturing Control*, Springer, New York, 2012.

Downloaded 02/26/14 to 129.31.182.66. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.