Regulation of the input flow of supply chains to optimize the production

C. D’Apice, C. De Nicola, R. Manzo
Department of Information Engineering, Electrical Engineering and Applied Mathematics
University of Salerno
84084, Fisciano, Salerno, Italy
Email: cdapice@unisa.it cdenicola@unisa.it rmanzo@unisa.it

KEYWORDS
Supply chains; Fluid dynamic model; Simulation; Optimization of input flow

ABSTRACT

The optimal control problem of adjusting the inflow, of piecewise constant type, to a supply chain in order to minimize the queue size and the quadratic difference between the outflow and the expected one is considered. The controls are represented by the duration of injections of different amounts of goods. The supply chain is modelled by a PDE-ODE: the conservation law describes the density of processed parts and the ODE the queue buffer occupancy. The numerical technique is based on the extensive use of generalized tangent vectors to a piecewise constant control, which represent time shifts of discontinuity points.

Introduction

The development of techniques for simulation and optimization purposes of industrial production is of great interest in order to answer questions raising in supply chain planning (optimal processing parameters, minimizing inventories to reduce costs or to ensure fully loaded production lines, and so on). Basically, we distinguish between steady state and instationary models which are time-dependent. A well-known class of stationary models are queuing theory models, which allow the calculation of several performance measures including the mean waiting time of goods in the system, the proportion of time the processors are busy and so on. In contrast, instationary models predict the time evolution of parts and include a dynamic inside the different production steps. The latter can be again divided into two classes: discrete (Discrete Event Simulations, [Forrester 1964]) or continuous (Differential Equations, [Armbruster et al. 2006], [Armbruster et al. 2007], [Helbing et al. 2004]). The latter class, thought in particular for large volume production on networks where a discrete description might fail, includes models based on partial differential equations ([Bretti et al. 2007], [D’Apice et al. 2010], [D’Apice and Manzo 2006], [D’Apice et al. 2009], [Göttlich et al. 2005]). Stochastic inputs to fluid dynamic models can be used to catch real behavior of complex systems.

In this paper, we focus on how to control in some case studies the flow through a supply chain so that a desired amount of goods can be produced and storage costs are minimized. The starting point is a continuous model for supply chains proposed by Göttlich, Herty and Klar in [Göttlich et al. 2005], briefly GHK model. A supply chain consists of processors with constant processing rate and a queue in front of each processor. The dynamics of parts on a processor is described by a conservation law, while the evolution of the queue buffer occupancy is given by an ordinary differential equation, determined by the difference of fluxes between the preceding and following processors.

Various optimal control problems, corresponding to different types of controls, have been analysed for the GHK model (see [D’Apice et al. 2010], [D’Apice et al. 2011], [Göttlich et al. 2010], [Göttlich et al. 2010], [Herty and Klar 2003], [Kirchner et al. 2006]), such as the problem of determining optimal velocities for each individual processing unit or, in the case of networks with a vertex of dispersing type (splitting in more lines), the distribution rate has been controlled to minimize queues. In [D’Apice et al. 2011], piecewise constant controls are considered together with generalized tangent vectors, which represent time shifts of discontinuities of the control. The technique of such generalized tangent vectors was extensively used for conservation laws, see [Bressan et al. 2000], and for the case of network models, see [Garavello and Piccoli 2006], [Herty at al. 2007]. The main result of [D’Apice et al. 2011] is the existence of optimal controls. In [D’Apice et al. 2013], an innovative numerical approach, which builds up on the idea of generalized tangent vectors, is presented, in order to solve the optimal control problem, where the control is given by the input flow to the supply chain and the cost functional \( J \) is the sum of time-integral of queues and quadratic distance from a preassigned desired outflow. In particular the controls are the locations of the discontinuities of the input flow of piecewise constant type, while the flux values are fixed. The numerical method is based on perturbations of piecewise constant controls, obtained by time shifting the discontinuity points. Generalized tangent vectors have a particularly simple evolution in time, which can be conveniently adapted to easily implementable methods such as Upwind-Euler (briefly UE, see [Cutolo et al. 2011 ]). The discretization of the evolution of generalized tangent vectors allows the numerical computation of the cost functional gradient. This can be combined
with classical steepest descent or more advanced Newton methods for the optimization procedure by iterations.

The outline of the paper is the following. In the first section the supply chain model is described together with the optimal control problem. Then the Wave Front Tracking algorithm to construct approximate solutions to the model and the definition of generalized tangent vectors is briefly reported. A section is devoted to the UE numerical algorithm able to find solutions to the ODE-PDE system and also to the numerics for generalized tangent vectors and cost functional derivative. Finally the Euler-Upwind steepest descent algorithm is applied to a case study.

An optimal control problem for supply chains

A supply chain consists of suppliers. Each supplier is composed of a processor for parts assembling and construction. Each processor has an upper limit of parts that can be handled simultaneously. To avoid congestions, each processor has its own buffering area (queue), located in front of the processor, where the parts are possibly stored before the actual processing starts.

Formally a supply chain is a finite directed graph \( G = (V, J) \) with arcs representing processors \( I_j, j \in J = \{1, \ldots, P\} \) and vertices, in \( V = \{1, \ldots, P - 1\} \), representing queues, in front of each processor, except the first. Each processor is parametrized by a bounded closed interval \( I_j = [a_j, b_j] \), with \( b_{j-1} = a_j, j = 2, \ldots, P \). The maximal processing rate \( \mu_j \), and the processing velocity, \( v_j = L_j/T_j \) with \( T_j \) and \( L_j = b_j - a_j \) the processing time and the length of the \( j \)-th processor, are user-defined parameters on each arc. The evolution of density along the \( j \)-th processor is given by a conservation law

\[
\partial_t \rho_j(x,t) + \partial_x f_j(\rho_j(x,t)) = 0, \quad \forall x \in [a_j, b_j], \quad \forall t > 0,
\]

\[
\rho_j(x,0) = \rho_{j,0}(x), \quad \rho_j(a_j,t) = f_{j,\text{inc}}(t) \cdot v_j, \quad \rho_j(b_j,t) = 0,
\]

where the flux function \( f_j(\rho_j(x,t)) \) is given by:

\[
f_j(x,t) = [0, +\infty) \rightarrow [0, \mu_j], \quad f_j(\rho_j(x,t)) = \min\left\{ \mu_j, v_j \rho_j(x,t) \right\},
\]

with \( \rho_j \in [0, \rho_j^{\text{max}}] \) the unknown function, representing the density of parts, while the initial datum \( \rho_{j,0} \) and the inflow \( f_{j,\text{inc}}(t) \) have to be assigned. An input profile \( u(t) \) on the left boundary \( \{a_j, t : t \in \mathbb{R}\} \) is given for the first arc of the supply chain. Each queue buffer occupancy is modelled as a time-dependent function \( t \rightarrow q_j(t) \). The dynamics of the buffering queue is governed by the following equation:

\[
q_j(t) = f_{j-1}(\rho_{j-1}(b_{j-1}, t)) - f_{j,\text{inc}}, \quad j = 2, \ldots, P,
\]

where the first term is defined by the trace of \( \rho_{j-1} \) (which is assumed to be of bounded variation on the \( x \) variable), while the second is defined by:

\[
f_{j,\text{inc}} = \begin{cases} \min\left\{ f_{j-1}(\rho_{j-1}(b_{j-1}, t)), \mu_j \right\} & \text{if } q_j(t) = 0, \\ \mu_j & \text{if } q_j(t) > 0. \end{cases}
\]

This allows the following interpretation. We process as many parts as possible. If the outgoing buffer is empty, then we process all incoming parts but at most \( \mu_j \), otherwise we can always process at rate \( \mu_j \). Summarizing, we obtain a system of partial differential equations governing the dynamics on each processor coupled to ordinary differential equations for the evolution of queues:

\[
\begin{align*}
\partial_t \rho_j(x,t) + \partial_x f_j(\rho_j(x,t)) &= \rho_j(x,t) - \rho_{j,0}(x), \quad j = 2, \ldots, P, \\
q_j(t) &= f_{j-1}(\rho_{j-1}(b_{j-1}, t)) - f_{j,\text{inc}}, \quad j = 2, \ldots, P, \\
\rho_j(a_j,t) &= f_{j,\text{inc}}(t), \quad j = 1, \ldots, P, \\
f_{j,\text{inc}}(t) &= u(t).
\end{align*}
\]

where \( f_{j,\text{inc}} \) is given by (4), for \( j = 2, \ldots, P \).

When we fix a time horizon \([0, T]\), we can define the cost functional:

\[
J(u) = \sum_{j=2}^P \int_0^T \alpha_1(t) q_j(t) dt + \int_0^T \alpha_2(t) \left| v_p \cdot \rho_p(b_p, t) - \psi(t) \right|^2 dt \equiv J_1(u) + J_2(u),
\]

where \( \alpha_1, \alpha_2 \) are weight functions, \( (\rho_j, q_j) \) is the solution to (5) for the control \( u, v_p, \rho_p(b_p, t) \) represents the outflow of the supply chain (assuming the density level is below \( \mu_p \)), while \( \psi(t) \) is a pre-assigned desired outflow. Given \( C > 0 \), we consider the minimization problem

\[
\min_{u \in \mathcal{U}_C} J(u)
\]

where \( \mathcal{U}_C = \{ u : [0, T] \rightarrow [0, \mu_1] ; u \text{ measurable, } T.V.(u) \leq C \} \) (with \( T.V. \) indicating the total variation). In other words, we want to minimize the queues length and the distance between the exiting flow and the pre-assigned flow \( \psi(t) \), using the supply chain input \( u \) as control. In particular, we assume an input flow of piecewise constant type, and fixing the levels we control the discontinuities times, it means that we aim to regulate the injection times of the different amount of goods.

In [D’Apice et al. 2011], the existence of an optimal control was proved for a general problem, which includes the case (5)-(7), while in [D’Apice et al. 2013] a new approach to solve (5)-(7) numerically is provided. The key idea is to focus on piecewise constant controls and perturb the position of discontinuity points. The procedure corresponds to define (generalized) tangent vectors to \( u \) (in the spirit of [Bressan et al. 2000]), taking advantage of the knowledge of time evolution of such tangent vectors, developed in [Herty et al. 2007]. For every \( u \in \mathcal{U} \subset \mathcal{U}_C \), with \( \mathcal{U}_C \) the set of Piecewise Constant controls we indicate by \( \tau_k = \tau_k(u), k = 1, \ldots, \delta(u) \), the discontinuity points of \( u \) (see 1). The perturbation to a piecewise constant control is defined as follows. Given \( u \in \mathcal{U} \), a tangent vector to \( u \) is a vector \( \xi = (\xi_1, \ldots, \xi_{\delta(u)}) \in \mathbb{R}^{\delta(u)} \) representing shifts of discontinuities. The norm of the tangent vector is
Then solve the new RP created by interaction of waves and prolong the solution up to next interaction time, and so on. Notice that, as soon as a boundary datum will achieve a value below \( \mu_j \), then in finite time all values above \( \mu_j \) will disappear from the \( j \)-th processor, see also [Herty et al. 2007]. Therefore, for simplicity, we will assume \((H1)\ \rho_{j,0}(x) \leq \mu_j \) for all \( j \in \mathcal{J} \). Then the same inequality will be satisfied for all times. In this case solutions to RPs are particularly simple, indeed the conservation law is linear, thus given some Riemann data \((\rho_-, \rho_+)\) on the \( j \)-th processor, the solution is always given by a shock travelling with velocity \( v_j \).

**Tangent vectors evolution**

The infinitesimal displacement of each discontinuity of the control \( u \) produces changes in the whole supply chain, whose effects are visible both on processors and on queues. In fact, every shift \( \xi \) generates shifts on the densities and shifts on the queues, which spread along the whole supply chain.

A tangent vector to the solution \((\rho_j, q_j)\) to (5) is given by:

\[
(\beta \xi_j, \eta_j),
\]

where \( \beta \) runs over the set of discontinuities of \( \rho_j \), \( \beta \xi_j \) are the shifts of the discontinuities, while \( \eta_j \) is the shift of the queue buffer occupancy \( q_j \). The norm of a tangent vector is given by:

\[
\| (\beta \xi_j, \eta_j) \| = \sum \| \beta \xi_j \| \Delta^\beta \rho_j + \sum |\eta_j|,
\]

where \( \Delta^\beta \rho_j = \beta \rho_j^+ - \beta \rho_j^- \) is the jump in \( \rho \) of the discontinuity \( \beta \) (where \( \beta \rho_j^+ \), respectively \( \beta \rho_j^- \), is the value on the left, respectively right, of the discontinuity \( \beta \)). Because of assumption (H1), we have no wave interaction inside the processors. Therefore, densities and queues shifts remain constant for almost all times and change only at those times at which one of the following interactions occurs:

- a) interaction of a density wave with a queue;
- b) emptying of the queue.

Assume a wave with shift \( \beta \xi_{j-1} \) interact with the \( j \)-th queue and let \( t \) be the interaction time. The symbols + and − indicate quantities before and after \( t \), respectively. So, \( \rho_j^+ \) and \( \rho_j^- \) indicate the densities on the processor \( I_j \) before and after an interaction occurs and similarly for \( I_{j-1} \). Moreover \( \beta \xi_j \) denote the shift on the processor \( I_j \) and with \( \beta \eta_j^- \) and \( \beta \eta_j^+ \) the shifts on the queue \( q_j \), respectively before and after the interaction. In case a) two subcases have to be distinguished:

a.1) \( q_j(t) = 0 \);

a.2) \( q_j(t) > 0 \).

In case a.1 two further subcases have to be considered:

a.1.1) if \( \rho_j^+ \xi_{j-1} < v_j-1 \rho_{j-1}^- < \rho_j^- \), then \( \beta \xi_j = \frac{v_j-1}{v_j-1-\rho_j^-} \beta \xi_{j-1} \) and \( \beta \eta_j^- = 0 = \beta \eta_j^+ \);

a.1.2) if \( \rho_j^+ \xi_{j-1} > \rho_j^- \), then \( \beta \xi_j = \frac{v_j}{v_j-1-\rho_j^-} \beta \xi_{j-1} \) and \( \beta \eta_j^- = \beta \xi_{j-1} (v_j-1-\rho_j^-) + \beta \eta_j^+ \).

In case a.2 two subcases have to be distinguished:

a.2.1) if \( \rho_j^+ \xi_{j-1} < v_j-1 \rho_{j-1}^- < \rho_j^- \), then \( \beta \xi_j = \frac{v_j}{v_j-1-\rho_j^-} \beta \xi_{j-1} \) and \( \beta \eta_j^- = 0 = \beta \eta_j^+ \);

a.2.2) if \( \rho_j^+ \xi_{j-1} > \rho_j^- \), then \( \beta \xi_j = \frac{v_j}{v_j-1-\rho_j^-} \beta \xi_{j-1} \) and \( \beta \eta_j^- = \beta \xi_{j-1} (v_j-1-\rho_j^-) + \beta \eta_j^+ \).
In case a.2) we have: $\beta \xi_j = 0$, $\beta \eta_j^+ = \beta \xi_{j-1}(\rho_{j-1}^- - \rho_{j-1}^+)$.
Finally in case b) we get: $\beta \eta_j^+ = 0, \beta \xi_{j-1} = 0$ and $\beta \xi_j = -(v_j^-/\rho_{j-1}^-)$. Using the above notations, we indicate with $\beta \xi_P$ the shift to a generic discontinuity of $\rho_P$ and with $\beta \rho_P^+$, respectively $\beta \rho_P^-$, the value of $\rho_P$ on the right, respectively left, of the discontinuity. Considering a control $u \in \mathcal{U}$ and a tangent vector $\xi \in \mathbb{R}^{(3\cdot)}$ to $u$, the gradient of the cost functional $J$ with respect to $\xi$ is given by:

$$\nabla \xi J(u) = Y_1^{WFT} + Y_2^{WFT},$$

where

$$Y_1^{WFT} = \sum_j \int_0^T \alpha_1(t)\eta_j(t)dt,$$

$$Y_2^{WFT} = \sum_{\beta} \left[ \alpha_2(t^\beta) \rho_P^+ + \beta \rho_P^+ - 2\rho(\beta^3) \right] \xi_P \Delta(\beta \rho_P)$$

with $\Delta(\beta \rho_P) = \beta \rho_P^+ - \beta \rho_P^-$ and $t^\beta$ the interaction time of the discontinuity indexed by $\beta$ with $b_P$, the right extreme of the supply chain.

**Steepest descent for the Upwind-Euler scheme**

In this section an Upwind-Euler scheme for the system (5) and then a numerical scheme for the evolution of the tangent vectors to a solution to the PDE-ODE model are introduced. From the latter it is possible to compute numerically the derivative of the cost functional with respect to the discontinuities of the input flow. This, in turn, will be used in steepest descent methods to find the optimal control.

For simplicity we assume:

(H2) The lengths $L_j$ are rationally dependent.

Assumption (H2) allows us to use a unique space mesh for all processors $I_j$, $j = 1, \ldots, P$. Indeed there exists $\Delta$ so that all $L_j$ are multiple of a value $\Delta$ and we will always use time and space meshes dividing by $\Delta$.

In the next section briefly the Upwind-Euler method, analysed in [Cutolo et al. 2011] by: $\mathcal{U}_{\Delta t} = \{ u \in \mathcal{U} : \tau_k(u) = n(u), k \Delta t, n(u) \in N, k = 1, \ldots, \delta(u) \}$. Now for every $u \in \mathcal{U}_{\Delta t}$, shifts $\xi$ are considered so that the obtained time-shifted control is still in $\mathcal{U}_{\Delta t}$. Then every $\xi_k$ is necessarily a multiple of $\Delta t$. Hence from now on we will restrict to the case:

$$\xi_k = \pm \Delta t, \quad k = 1, \ldots, \delta(u).$$

For a generic processor $I_j$ and a discontinuity point $\tau_k$ of the control, $kJ\xi^n_i$ and $kJ\eta^n$ denote the approximations of $kJ\xi_i(t^n)$ and $kJ\eta_i(t^n)$, respectively. Such approximations are defined by a recursive procedure explained in the following.

The tangent vector approximations are initialized by setting:

$$kJ\xi^n_i = 0, \quad n = 1, \ldots, n(k-1, u),$$

$$kJ\eta^n_0 = 0, \quad j = 1, \ldots, P.$$
The definition of $k_{j}^{1} j_{i}^{n(k,u)}$ reflects the fact that the shift $\xi_{k}$ provokes a shift of the wave generated on the first processor.

Now, the evolution of approximations of tangent vectors to $\rho$ inside processors is simply given by:

$$k_{j}^{n+1} j_{i}^{n} = k_{j}^{n} j_{i}^{n-1}.$$ 

On the other side, the approximation of $\xi$ and $\eta$ influence each other at interaction times with queues. More precisely, the four cases described in the previous section are considered, obtaining:

\begin{itemize}
  \item a.1.1) $k_{j}^{1} j_{i}^{n+1} = 0, k_{j}^{1} j_{i}^{n+1} = \frac{\nu_{j}}{\nu_{j-1}} k_{j-1}^{1} j_{i}^{n} N_{i-1}$;
  \item a.1.2) $k_{j}^{n+1} j_{i}^{n} = \frac{\nu_{j}}{\nu_{j-1}} k_{j-1}^{n} j_{i}^{n} N_{i-1} \equiv k_{j}^{n} j_{i}^{n}$;
  \item a.2) $k_{j}^{1} j_{i}^{n+1} = 0, k_{j}^{1} j_{i}^{n+1} = \left( j-1 \rho_{N_{i-1}} \right) - \left( j-1 \rho_{N_{i-1}} + \rho_{N_{i-1}} \right)$;
  \item b) $k_{j}^{1} j_{i}^{n+1} = 0, k_{j}^{n+1} j_{i}^{n} = 0, k_{j}^{n+1} j_{i}^{n}$.
\end{itemize}

Now numerical approximations for $\nabla \xi J$ can be computed. Denote by $k_{j} Y_{1}^{n}$, respectively $k_{j} Y_{2}^{n}$, the numerical approximations of the $k$-th component of $Y_{1}^{WFFT}$, respectively $Y_{2}^{WFFT}$, as defined in (10) on processor $I_{j}$ at time $n$.

Such approximation is initialized by setting:

$$k_{j} Y_{1}^{0} = 0, \quad k_{j} Y_{2}^{0} = 0, \quad j = 1, \ldots, P, \quad k = 1, \ldots, \delta(u).$$

The evolution is determined by the following simple rules. For $Y_{1}$, if $q_{j}^{n+1} > 0$, then we set

$$k_{j} Y_{1}^{n+1} = k_{j} Y_{1}^{n} + \alpha_{1}(n) k_{j} \eta^{n} \triangle t,$$

while for $q_{j}^{n+1} = 0$ two subcases are distinguished:

\begin{itemize}
  \item if $q_{j}^{n} = 0$, then $k_{j} Y_{1}^{n+1} = k_{j} Y_{1}^{n}$;
  \item if $q_{j}^{n} > 0$, then $k_{j} Y_{1}^{n+1} = k_{j} Y_{1}^{n} + \alpha_{1}(n) k_{j} \eta^{n}.$
\end{itemize}

For $Y_{2}$ we set:

$$k_{j} Y_{2}^{n+1} = k_{j} Y_{2}^{n} + \alpha_{2}(n) \nu_{P} \left( P \rho_{N_{i}} - \psi(n) \right)^{2} - \left( P \rho_{N_{i}} - \psi(n) \right)^{2} k_{j} P \eta^{n}.$$

A steepest descent algorithm, denoting with $\vartheta$ the iteration step, is defined by setting

$$\vartheta_{k}^{n+1} = \vartheta_{k}^{n} + \left[ \frac{\Delta t}{\Delta t} \left( \sum_{j} \sum_{n} k_{j} Y_{1}^{n} + \sum_{n} k_{j} Y_{2}^{n} \right) \right.$$

where $h_{\vartheta}$ is a coefficient to be suitably chosen. More precisely the parameter $h_{\vartheta}$ may be chosen to solve an optimization problem to get specific schemes. In [D’Apice et al. 2013] convergence results and error estimates for the Upwind-Euler steepest descent scheme are provided.

Simulations

In this Section we use the Euler-Upwind steepest descent algorithm on a test case. The latter concerns a painting process of the cars bonnet, which consists of the following workflow: the bonnets enter the process and are painted, then they pass to the cooking stage after that they are washed. Finally, the bonnets are packaged and distributed to the retailers. This process can be represented by a sequential graph where the vertices summarize the activities and arcs the transitions between different activities, as shown in the following figure. Consider the following input function:

$$u(t) = \begin{cases} 
  u_{1} & 0 \leq t \leq \tau_{1}, \\
  u_{2} & \tau_{1} < t \leq \tau_{2}, \\
  u_{3} & \tau_{2} < t \leq T.
\end{cases}$$

and fix the wished output flow $\psi(t) = 50$.

We assume that the supply chain (see Figure 2) is initially empty, $v_{j} = 1$ for every $j = 1, \ldots, 6$ and

$$\mu_{1} = 100, \quad \mu_{2} = 70, \quad \mu_{3} = 50, \quad \mu_{4} = 30, \quad \mu_{5} = 40, \quad \mu_{6} = 20.$$ 

For simplicity we set $\alpha_{1} \equiv 1$ and $\alpha_{2} \equiv 1$ and analyze two different cases:

- **Case a)** $u_{1} = 80, u_{2} = 90, u_{3} = 110, T = 35$, initial values $(\tau_{1}, \tau_{2}) = (2, 5)$;
- **Case b)** $u_{1} = 100, u_{2} = 50, u_{3} = 80, T = 35$, initial values $(\tau_{1}, \tau_{2}) = (2, 5)$.

As time and space grid meshes we choose $\Delta x = 0.02$ and $\Delta t = 0.016$ so as to satisfy the CFL condition. We use the condition that $J$ remains unchanged for five runnings of the algorithm as forced stop criterion. The times $\tau_{1}$ and $\tau_{2}$ found by the algorithm are:

- **Case a)** $\tau_{1} \simeq 33.99, \tau_{2} \simeq 34.15$: as expected both $\tau_{1}$ and $\tau_{2}$ run toward $T$; in fact, in order to minimize the queues, the optimization algorithm tends to reduce the inflow levels which increase the queues (i. e. $u_{2}$ and $u_{3}$).
- **Case b)** $\tau_{1} = 0, \tau_{2} \simeq 33.91$: $\tau_{1}$ runs to zero and $\tau_{2}$ runs toward $T$; as in the previous case, the optimization algorithm works to reduce the inflow levels which lead to queues increasing (i. e. $u_{1}$ and $u_{3}$).

In Table I we report the numerical values of $\tau_{1}$, $\tau_{2}$ and $J$ at each iteration of the steepest descent algorithm for Case a).

Figures 3-4 depict the values assumed by $J$, and $(\tau_{1}, \tau_{2})$, at each iteration step for Case a). The behaviour of the cost functional $J$ in the plane $(\tau_{1}, \tau_{2})$ is reported for Case b) in Figure 5, to confirm the goodness of the steepest descent algorithm.
TABLE I: Case a: values of $\tau_1$, $\tau_2$, and corresponding $J$ in 57 iterations of the steepest descent algorithm.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>$\tau_1$</th>
<th>$\tau_2$</th>
<th>$J$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>5</td>
<td>6259043.87</td>
</tr>
<tr>
<td>2</td>
<td>8.39</td>
<td>16.58</td>
<td>5975423.18</td>
</tr>
<tr>
<td>3</td>
<td>13.50</td>
<td>23.53</td>
<td>5840830.02</td>
</tr>
<tr>
<td>4</td>
<td>17.59</td>
<td>27.70</td>
<td>5711361.84</td>
</tr>
<tr>
<td>5</td>
<td>20.87</td>
<td>30.20</td>
<td>5733613.88</td>
</tr>
<tr>
<td>6</td>
<td>23.49</td>
<td>31.70</td>
<td>5711361.84</td>
</tr>
<tr>
<td>7</td>
<td>25.58</td>
<td>32.61</td>
<td>5697897.12</td>
</tr>
<tr>
<td>8</td>
<td>27.26</td>
<td>33.15</td>
<td>5689631.94</td>
</tr>
<tr>
<td>9</td>
<td>28.60</td>
<td>33.48</td>
<td>5680830.02</td>
</tr>
<tr>
<td>10</td>
<td>30.53</td>
<td>33.79</td>
<td>5678934.30</td>
</tr>
<tr>
<td>11</td>
<td>31.21</td>
<td>33.86</td>
<td>5675814.40</td>
</tr>
<tr>
<td>12</td>
<td>31.76</td>
<td>33.90</td>
<td>5676726.63</td>
</tr>
<tr>
<td>13</td>
<td>32.20</td>
<td>33.93</td>
<td>5667195.55</td>
</tr>
<tr>
<td>14</td>
<td>32.55</td>
<td>33.95</td>
<td>565820.15</td>
</tr>
<tr>
<td>15</td>
<td>32.84</td>
<td>33.96</td>
<td>565858.43</td>
</tr>
<tr>
<td>16</td>
<td>33.06</td>
<td>33.97</td>
<td>565437.89</td>
</tr>
<tr>
<td>17</td>
<td>33.24</td>
<td>33.98</td>
<td>565344.61</td>
</tr>
<tr>
<td>18</td>
<td>33.39</td>
<td>33.98</td>
<td>565282.03</td>
</tr>
<tr>
<td>19</td>
<td>33.50</td>
<td>33.99</td>
<td>565237.87</td>
</tr>
<tr>
<td>20</td>
<td>33.59</td>
<td>34.00</td>
<td>565214.97</td>
</tr>
<tr>
<td>21</td>
<td>33.67</td>
<td>34.00</td>
<td>565196.57</td>
</tr>
<tr>
<td>22</td>
<td>33.72</td>
<td>34.00</td>
<td>565187.45</td>
</tr>
<tr>
<td>23</td>
<td>33.77</td>
<td>34.01</td>
<td>565179.41</td>
</tr>
<tr>
<td>24</td>
<td>33.80</td>
<td>34.02</td>
<td>565175.09</td>
</tr>
<tr>
<td>25</td>
<td>33.82</td>
<td>34.03</td>
<td>565171.17</td>
</tr>
<tr>
<td>26</td>
<td>33.84</td>
<td>34.05</td>
<td>565169.43</td>
</tr>
<tr>
<td>27</td>
<td>33.85</td>
<td>34.06</td>
<td>565169.43</td>
</tr>
<tr>
<td>28</td>
<td>33.86</td>
<td>34.07</td>
<td>565167.90</td>
</tr>
<tr>
<td>29</td>
<td>33.88</td>
<td>34.07</td>
<td>565166.61</td>
</tr>
<tr>
<td>30</td>
<td>33.89</td>
<td>34.08</td>
<td>565165.43</td>
</tr>
<tr>
<td>31</td>
<td>33.89</td>
<td>34.09</td>
<td>565165.43</td>
</tr>
<tr>
<td>32</td>
<td>33.90</td>
<td>34.10</td>
<td>565164.42</td>
</tr>
<tr>
<td>33</td>
<td>33.91</td>
<td>34.10</td>
<td>565164.42</td>
</tr>
<tr>
<td>34</td>
<td>33.92</td>
<td>34.10</td>
<td>565163.58</td>
</tr>
<tr>
<td>35</td>
<td>33.92</td>
<td>34.11</td>
<td>565163.57</td>
</tr>
<tr>
<td>36</td>
<td>33.93</td>
<td>34.11</td>
<td>565163.57</td>
</tr>
<tr>
<td>37</td>
<td>33.94</td>
<td>34.12</td>
<td>565162.87</td>
</tr>
<tr>
<td>38</td>
<td>33.94</td>
<td>34.12</td>
<td>565162.87</td>
</tr>
<tr>
<td>39</td>
<td>33.94</td>
<td>34.12</td>
<td>565162.87</td>
</tr>
<tr>
<td>40</td>
<td>33.95</td>
<td>34.12</td>
<td>565162.30</td>
</tr>
<tr>
<td>41</td>
<td>33.95</td>
<td>34.13</td>
<td>565162.30</td>
</tr>
<tr>
<td>42</td>
<td>33.96</td>
<td>34.13</td>
<td>565162.30</td>
</tr>
<tr>
<td>43</td>
<td>33.96</td>
<td>34.13</td>
<td>565162.30</td>
</tr>
<tr>
<td>44</td>
<td>33.96</td>
<td>34.13</td>
<td>565161.85</td>
</tr>
<tr>
<td>45</td>
<td>33.96</td>
<td>34.13</td>
<td>565161.85</td>
</tr>
<tr>
<td>46</td>
<td>33.97</td>
<td>34.13</td>
<td>565161.85</td>
</tr>
<tr>
<td>47</td>
<td>33.97</td>
<td>34.14</td>
<td>565161.85</td>
</tr>
<tr>
<td>48</td>
<td>33.97</td>
<td>34.14</td>
<td>565161.85</td>
</tr>
<tr>
<td>49</td>
<td>33.97</td>
<td>34.14</td>
<td>565161.85</td>
</tr>
<tr>
<td>50</td>
<td>33.97</td>
<td>34.14</td>
<td>565161.85</td>
</tr>
<tr>
<td>51</td>
<td>33.98</td>
<td>34.14</td>
<td>565161.85</td>
</tr>
<tr>
<td>52</td>
<td>33.98</td>
<td>34.14</td>
<td>565161.50</td>
</tr>
<tr>
<td>53</td>
<td>33.98</td>
<td>34.14</td>
<td>565161.50</td>
</tr>
<tr>
<td>54</td>
<td>33.98</td>
<td>34.14</td>
<td>565161.50</td>
</tr>
<tr>
<td>55</td>
<td>33.99</td>
<td>34.14</td>
<td>565161.50</td>
</tr>
<tr>
<td>56</td>
<td>33.99</td>
<td>34.14</td>
<td>565161.50</td>
</tr>
<tr>
<td>57</td>
<td>33.99</td>
<td>34.15</td>
<td>565161.50</td>
</tr>
</tbody>
</table>

Notice that since $J$ decreases when the number of iteration increases, the optimization of $\tau_1$ and $\tau_2$ allows an effective decrement of queues sizes.

The values of $\tau_1$ and $\tau_2$, found by the steepest descent algorithm, seem to be grid independent; in fact choosing different time and space grid meshes the variation of the optimal discontinuities values is not meaningful: for $\Delta x = 0.01$ and $\Delta t = 0.008$ the algorithm stops with values of $(\tau_1, \tau_2)$ in a small neighborhood of $(33.99, 34.16)$ in 109 iterations, respectively $(0, 33.93)$ in 77 iterations, for Case a, respectively Case b.

Notice that since $J$ decreases when the number of iteration increases, the optimization of $\tau_1$ and $\tau_2$ allows an effective decrement of queues sizes.

The values of $\tau_1$ and $\tau_2$, found by the steepest descent algorithm, seem to be grid independent; in fact choosing different time and space grid meshes the variation of the optimal discontinuities values is not meaningful: for $\Delta x = 0.01$ and $\Delta t = 0.008$ the algorithm stops with values of $(\tau_1, \tau_2)$ in a small neighborhood of $(33.99, 34.16)$ in 109 iterations, respectively $(0, 33.93)$ in 77 iterations, for Case a, respectively Case b.

CONCLUSIONS

In this paper, starting by a fluid dynamic model for supply chains, we used a new optimization method based on tangent vectors technique to find the best configuration of input flow in order to minimize the queues formation (representing bottlenecks for the sup-
supply chain) at each processor, and improve the final production. Due to the mathematical difficulties to solve analytically the PDE and ODE equations, we implemented a numerical method using the steepest descent algorithm to get our goal and applied it to some test cases. A further step forward, as future development, it could be done dealing with more complex supply network, where each node could be affected by multiple processors. In this case, incoming and outgoing flows should be considered, and some distribution rules should be implemented.

REFERENCES


CIRO D’APICE is full professor in Mathematical Analysis at the Department of Information Engineering, Electrical Engineering and Applied Mathematics of the University of Salerno, Italy. He is the Director of two research centers: CRMPA - Centro di Ricerca in Matematica Pura ed Applicata, CEMMAC - Centro di Eccellenza su Metodi e Sistemi per Aziende Competitive. He is author of 2 books and more then 150 papers about homogenization and optimal control; conservation laws models for vehicular traffic, telecommunications and supply chains; spatial behaviour for dynamic problems; queueing systems and networks. His e-mail address is cdapice@unisa.it.

CARMINE DE NICOLA is an Electronics Engineer and received his Ph.D. in Mathematics in 2011, at University of Salerno. At the moment he is assistant researcher at University of Salerno. His current research activities focus on queuing theory, operation research and mathematical modeling of complex systems. His e-mail address is cdenicola@unisa.it.

ROSANNA MANZO is a researcher in Mathematical Analysis at the Department of Information Engineering, Electrical Engineering and Applied Mathematics of the University of Salerno, Italy. She received PhD in Information Engineering from University of Salerno. Her research areas include fluid - dynamic models for traffic flows on road, telecommunication and supply networks, optimal control, queueing theory, self - similar processes, computer aided learning. She is author of about 50 papers appeared on international journals and many publications on proceedings. Her e-mail address is rmanzo@unisa.it.