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

Proceedings 28th European Conference on Modelling and Simulation ©ECMS Flaminio Squazzoni, Fabio Baronio, Claudia Archetti, Marco Castellani (Editors)
ISBN: 978-0-9564944-8-1 / ISBN: 978-0-9564944-9-8 (CD)

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, $j\in J=\{1,...,P\}$ and vertices, in $V=\{1,...,P-1\}$, 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 $b_{j-1}=a_j, j=2,...,P$. The maximal processing rate μ_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

$$\partial_t\rho_j(x,t)+\partial_x f_j(\rho_j(x,t))=0,\quad \forall x\in[a_j,b_j],\quad \forall t>0, \tag{1}$$

$$\rho_j\left(x,0\right)=\rho_{j,0}\left(x\right),\quad \rho_j\left(a_j,t\right)=\frac{f_{j,inc}(t)}{v_j},$$

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

$$\begin{aligned} f_j &:& [0,+\infty[\,\to[0,\mu_j]\,, \\ f_j\left(\rho_j\left(x,t\right)\right) &=& \min\left\{\mu_j,v_j\rho_j(x,t)\right\}, \end{aligned} \tag{2}$$

with $\rho_j\in[0,\rho_j^{max}]$ the unknown function, representing the density of parts, while the initial datum $\rho_{j,0}$ and the inflow $f_{j,inc}(t)$ have to be assigned. An input profile $u(t)$ on the left boundary $\{(a_1,t):t\in\mathbb{R}\}$ is given for the first arc of the supply chain. Each queue buffer occupancy is modelled as a time-dependent function $t\to q_j(t)$. The dynamics of the buffering queue is governed by the following equation:

$$\dot{q}_j(t)=f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right)-f_{j,inc},\quad j=2,...,P, \tag{3}$$

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

$$f_{j,inc}=\begin{cases}\min\left\{f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right),\mu_j\right\} & \text{if } q_j\left(t\right)=0,\\ \mu_j & \text{if } q_j\left(t\right)>0.\end{cases} \tag{4}$$

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 μ_j, otherwise we can always process at rate μ_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:

$$\begin{cases}\partial_t\rho_j\left(x,t\right)+\partial_x\min\left\{\mu_j,v_j\rho_j\left(x,t\right)\right\}=0 & j=1,...,P,\\ \dot{q}_j(t)=f_{j-1}\left(\rho_{j-1}\left(b_{j-1},t\right)\right)-f_{j,inc}(t) & j=2,...,P,\\ q_j\left(0\right)=q_{j,0} & j=2,...,P,\\ \rho_j\left(x,0\right)=\rho_{j,0}\left(x\right) & j=1,...,P,\\ \rho_j\left(a_j,t\right)=\frac{f_{j,inc}(t)}{v_j} & j=1,...,P,\\ f_{1,inc}(t)=u(t) & \end{cases} \tag{5}$$

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

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

$$\begin{aligned} J(u) &=& \sum_{j=2}^{P}\int_0^T\alpha_1(t)\,q_j(t)dt+\\ && +\int_0^T\alpha_2(t)\,\left[v_P\cdot\rho_P(b_P,t)-\psi(t)\right]^2dt\\ &\doteq& J_1(u)+J_2(u), \end{aligned} \tag{6}$$

where α_1,α_2 are weight functions, (ρ_j,q_j) is the solution to (5) for the control u, $v_P\cdot\rho_P(b_P,t)$ represents the outflow of the supply chain (assuming the density level is below μ_P), while $\psi(t)\in$ is a pre-assigned desired outflow. Given $C>0$, we consider the minimization problem

$$\min_{u\in\mathcal{U}_C}J(u) \tag{7}$$

where $\mathcal{U}_C=\{u:[0,T]\to[0,\mu_1]\,;\,u$ 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 preassigned 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 at al. 2007]. For every $u\in\widetilde{\mathcal{U}}\subset\mathcal{U}_C$, 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 $u\in\widetilde{\mathcal{U}}$, a tangent vector to u is a vector $\xi=(\xi_1,\ldots,\xi_{\delta(u)})\in\mathbb{R}^{\delta(\approxeq)}$ representing shifts of discontinuities. The norm of the tangent vector is

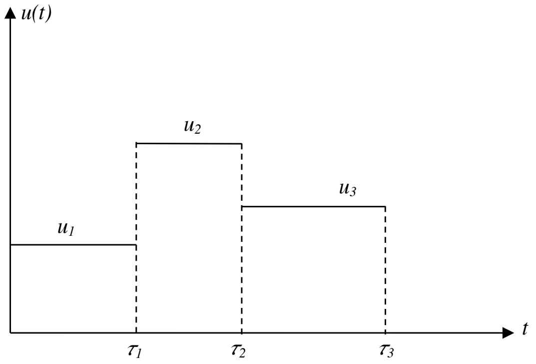

Fig. 1. Input flow.

defined as:

$$\|\xi\| = \sum_{k=1}^{\delta(u)} |\xi_k| \cdot |u(\tau_k+) - u(\tau_k-)| \,,$$

where $(\cdot\, +)$ and $(\cdot\, -)$ indicate the limit from the right and the left. Assume now for simplicity that $\tau_1 > 0$, $\tau_{\delta(u)} < T$, and set $\tau_0 = 0$, $\xi_0 = 0$, $\tau_{\delta(u)+1} = T$, $\xi_{\delta(u)+1} = 0$. Then given a tangent vector ξ to u, for every ε sufficiently small we define the infinitesimal displacement as:

$$u_\varepsilon = \sum_{k=0}^{\delta(u)} \chi_{[\tau_k+\varepsilon\xi_k, \tau_{k+1}+\varepsilon\xi_{k+1}[} u(\tau_k+), \qquad (8)$$

where χ is the indicator function. In other words u_ε is obtained from u by shifting the discontinuity points of $\varepsilon\xi$

In the next sections numerical schemes for the solution of (5) and for the evolution of tangent vectors will be defined. The latter, in turn, will provide the information for the computation of numerical gradient of the cost functional J.

The evolution of tangent vectors is particularly clear for the theoretical numerical scheme given by the WFT algorithm. We thus recall briefly the WFT algorithm and the evolution of tangent vectors along approximate solutions constructed via the WFT algorithm.

The Wave Front Tracking algorithm

In this section we explain how to construct piecewise constant approximate solutions to (5) by WFT method, see [Bressan 2000] for details.

Given a discretization parameter σ and initial conditions $\rho_{j,0}$ in BV, the space of bounded variation functions, a WFT approximate solution is constructed by a procedure sketched by the following steps:

• Approximate the initial datum by a piecewise constant function (with discretization parameter σ) and solve the Riemann Problems (RPs) corresponding to discontinuities of the approximation. In RPs solutions approximate rarefactions by rarefaction shocks of size σ;

• Use the piecewise constant solution obtained piecing together the solutions to RPs up to the first time of interaction of two shocks;

• 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 μ_j, then in finite time all values above μ_j will disappear from the j-th processor, see also [Herty at al. 2007]. Therefore, for simplicity, we will assume

(H1) $\rho_{j,0}(x) \le \mu_j$ 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 (ρ_-, ρ_+) 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 ξ generates shifts on the densities and shifts on the queues, which spread along the whole supply chain.

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

$$({}^\beta\xi_j, \eta_j),$$

where β runs over the set of discontinuities of ρ_j, ${}^\beta\xi_j$ are the shifts of the discontinuities, while η_j is the shift of the queue buffer occupancy q_j. The norm of a tangent vector is given by:

$$\|({}^\beta\xi_j, \eta_j)\| = \sum_\beta |{}^\beta\xi_j||\Delta^\beta\rho_j| + \sum_j |\eta_j|, \qquad (9)$$

where $\Delta^\beta\rho_j = {}^\beta\rho_j^l - {}^\beta\rho_j^r$ is the jump in ρ of the discontinuity β (where ${}^\beta\rho_j^l$, respectively ${}^\beta\rho_j^r$, is the value on the left, respectively right, of the discontinuity β).

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 $\bar{t}$ be the interaction time. The symbols $+$ and $-$ indicate quantities before and after $\bar{t}$, respectively. So, ρ_j^- and ρ_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) $q_j(\bar{t}) = 0$;

a.2) $q_j(\bar{t}) > 0$.

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

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

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

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

$$\nabla_{\xi} J(u) = Y_1^{WFT} + Y_2^{WFT}, \tag{10}$$

where

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

$$Y_2^{WFT} \doteq \sum_{\beta} [\alpha_2(t^{\beta})\, v_P \left({}^{\beta}\rho_P^+ + {}^{\beta}\rho_P^- - 2\psi(t^{\beta})\right) {}^{\beta} \cdot \xi_P\, \Delta({}^{\beta}\rho_P)]$$

with $\Delta({}^{\beta}\rho_P) = {}^{\beta}\rho_P^+ - {}^{\beta}\rho_P^-$ and t^{β} the interaction time of the discontinuity indexed by β 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 Δ so that all L_j are multiple of a value Δ and we will always use time and space meshes dividing by Δ.

In the next section briefly the Upwind-Euler method, analysed in [Cutolo et al. 2011] to construct numerical solutions to the supply chain model (5) is reported.

A. Upwind-Euler scheme for supply chains

Given a space mesh Δx, for each processor I_j, we set $\Delta t_j = \Delta x / v_j$ and define a numerical grid of $[0, L_j] \times [0, T]$ by:
- $(x_i, t^n)_j = (i\Delta x,\ n\Delta t_j)$, $i = 0, ..., N_j$, $n = 0, ..., M_j$ are the grid points;
- ${}^{j}\rho_i^n$ is the value taken by the approximated density at the point $(x_i, t^n)_j$;
- q_j^n is the value taken by the approximate queue buffer occupancy at time t^n.

The Upwind method reads:

$${}^{j}\rho_i^{n+1} = {}^{j}\rho_i^n - \frac{\Delta t_j}{\Delta x} v_j \left({}^{j}\rho_i^n - {}^{j}\rho_{i-1}^n\right) = {}^{j}\rho_{i-1}^n, \tag{11}$$

where $j \in \mathcal{J}$, $i = 0, ..., N_j$ and $n = 0, ..., M_j$. Notice that the CFL condition is given by $\Delta t_j \le \frac{\Delta x}{v_j}$, and thus holds true. The explicit Euler method is given by:

$$\begin{aligned} q_j^{n+1} &= q_j^n + \Delta t_j \left(f_{j-1}^n - f_{j,inc}^n\right), \ j \in \mathcal{J} \setminus \{\infty\}, \\ n &= 0, ..., M_j, \end{aligned} \tag{12}$$

where f_{j-1}^n needs to be defined and

$$f_{j,inc}^n = \begin{cases} \min\left\{f_{j-1}({}^{j-1}\rho_{N_{j-1}}^n), \mu_j\right\} & q_j^n(t) = 0, \\ \mu_j & q_j^n(t) > 0. \end{cases} \tag{13}$$

Now, if $\Delta t_{j-1} \le \Delta t_j$ we set:

$$\begin{aligned} f_{j-1}^n &= \sum_{l=1}^{M(n)-m(n)-1} \Delta t_{j-1} f_{j-1}({}^{j-1}\rho_{N_{j-1}}^{m(n)+l}) = \\ &= \sum_{l=1}^{\gamma} \Delta t_{j-1} f_{j-1}({}^{j-1}\rho_{N_{j-1}}^{\gamma n+l}), \end{aligned} \tag{14}$$

where $m(n)$ and $M(n)$ are defined as:

$$m(n) = \sup\{m : m\Delta t_{j-1} \le n\Delta t_j\},$$

$$M(n) = \inf\{M : M\Delta t_{j-1} \ge (n+1)\Delta t_j\}.$$

Otherwise, that is if $\Delta t_{j-1} > \Delta t_j$, we set:

$$f_{j-1}^n = f_{j-1}\left({}^{j-1}\rho_{N_{j-1}}^{\lfloor \frac{n\Delta t_j}{\Delta t_{j-1}} \rfloor}\right), \tag{15}$$

where $\lfloor\cdot\rfloor$ indicates the floor function. Boundary data are treated using ghost cells and the expression of inflows given by (13). The convergence of the scheme has been proved in [Cutolo et al. 2011] using a comparison with WFT approximate solutions.

B. Numerics for tangent vectors and cost functional

First the control space is discretized via the time mesh Δt:

$$\widetilde{\mathcal{U}}_{\Delta t} = \{u \in \widetilde{\mathcal{U}} : \tau_k(u) = n(u,k)\,\Delta t,\ n(k,u) \in \mathcal{N},\ k = 1, \ldots, \delta(u)\}.$$

Now for every $u \in \widetilde{\mathcal{U}}_{\Delta t}$ shifts ξ are considered so that the obtained time-shifted control is still in $\widetilde{\mathcal{U}}_{\Delta t}$. Then every ξ_k is necessarily a multiple of Δt. 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 τ_k of the control, ${}^{k,j}\xi_i^n$ and ${}^{k,j}\eta^n$ denote the approximations of ${}^{k}\xi_j(x_i, t^n)$, and ${}^{k}\eta_j(t^n)$, respectively. Such approximations are defined by a recursive procedure explained in the following.
The tangent vector approximations are initialized by setting:

$$\begin{aligned} {}^{k,j}\xi_i^n &= 0, \quad \text{for } n = 1, \ldots, n(k-1, u), \\ j &= 1, \ldots, P, \\ {}^{k,1}\xi_1^{n(k,u)} &= v_1(\pm\Delta t), \\ {}^{k,j}\eta^0 &= 0, \quad j = 1, \ldots, P. \end{aligned}$$

The definition of ${}^{k,1}\xi_1^{n(k,u)}$ reflects the fact that the shift ξ_k provokes a shift of the wave generated on the first processor.
Now, the evolution of approximations of tangent vectors to ρ inside processors is simply given by:

$$ {}^{k,j}\xi_i^{n+1} \;=\; {}^{k,j}\xi_{i-1}^{n}. $$

On the other side, the approximation of ξ and η influence each other at interaction times with queues. More precisely, the four cases described in the previous section are considered, obtaining:
a.1.1): ${}^{k,j}\eta^{n+1} = 0,\ {}^{k,j}\xi_1^{n+1} = \frac{v_j}{v_{j-1}}\ {}^{k,j-1}\xi_{N_{j-1}}^{n}$;
a.1.2): ${}^{k,j}\xi_1^{n+1} = \frac{v_j}{v_{j-1}}\ {}^{k,j-1}\xi_{N_{j-1}}^{n},\ {}^{k,j}\eta^{n+1} =$
$= {}^{k,j-1}\xi_{N_{j-1}}^{n}\,\frac{\left(v_{j-1}\ {}^{j-1}\rho_{N_{j-1}}^{n+1} - \mu_j\right)}{v_{j-1}} + {}^{k,j}\eta^{n}$;
a.2): ${}^{k,j}\xi_1^{n+1} = 0,\ {}^{k,j}\eta^{n+1} =$
$= {}^{k,j-1}\xi_{N_{j-1}}^{n}\left({}^{j-1}\rho_{N_{j-1}}^{n+1} - {}^{j-1}\rho_{N_{j-1}}^{n}\right) + {}^{k,j}\eta^{n}$;
b): ${}^{k,j-1}\xi_{N_{j-1}}^{n} = 0,\ {}^{k,j}\eta^{n+1} = 0,\ {}^{k,j}\xi_1^{n+1} = -\frac{v_j\ {}^{k,j}\eta^{n}}{v_{j-1}\ {}^{j-1}\rho_{N_{j-1}}^{n} - \mu_j}$.
Now numerical approximations for $\nabla_\xi J$ can be computed. Denote by ${}^{k,j}Y_1^n$, respectively ${}^{k}Y_2^n$, the numerical approximations of the k-th component of Y_1^{WFT}, respectively Y_2^{WFT}, as defined in (10) on processor I_j at time t^n.
Such approximation is initialized by setting:

$$ {}^{k,j}Y_1^0 = 0,\ {}^{k}Y_2^0 = 0, \quad j = 1,\dots,P,\ k = 1,\dots,\delta(u). $$

The evolution is determined by the following simple rules. For Y_1, if $q_j^{n+1} > 0$, then we set

$$ {}^{k,j}Y_1^{n+1} \;=\; {}^{k,j}Y_1^{n} + \alpha_1(t^n)\ {}^{k,j}\eta^{n}\triangle t, $$

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

- if $q_j^n = 0$, then ${}^{k,j}Y_1^{n+1} = {}^{k,j}Y_1^{n}$;
- if $q_j^n > 0$, then ${}^{k,j}Y_1^{n+1} = {}^{k,j}Y_1^{n} + \frac{1}{2}\ \alpha_1(t^n)\ {}^{k,j}\xi_1^{n+1}\ {}^{k,j}\eta^{n}$.

For Y_2 we set:

$$ {}^{k}Y_2^{n+1} \;=\; {}^{k}Y_2^{n} + \alpha_2(t^n)\,v_P\left(\left({}^{P}\rho_{N_P}^{n} - \psi(t^n)\right)^2 - \left({}^{P}\rho_{N_P}^{n+1} - \psi(t^n)\right)^2\right)\ {}^{k,P}\xi_{N_P}^{n}. $$

A steepest descent algorithm, denoting with ϑ the iteration step, is defined by setting

$$ \tau_k^{\vartheta+1} = \tau_k^{\vartheta} + \lfloor \frac{h_\theta\left(\sum_j\sum_n\ {}^{k,j}Y_1^n + \sum_n\ {}^{k}Y_2^n\right)}{\Delta t} \rfloor \Delta t, $$

where h_θ is a coefficient to be suitably chosen. More precisely the parameter h_θ 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:

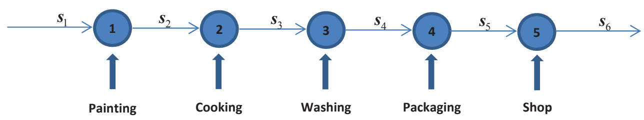

Fig. 2. Painting process of bonnett.

$$ u(t) = \begin{cases} u_1 & 0 \le t \le \tau_1, \\ u_2 & \tau_1 < t \le \tau_2, \\ u_3 & \tau_2 < t \le T. \end{cases} $$

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 $j = 1, ..., 6$ and

$$ \mu_1 = 100,\ \mu_2 = 70,\ \mu_3 = 50,\ \mu_4 = 30,\ \mu_5 = 40, \mu_6 = 20. $$

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 τ_1 and τ_2 found by the algorithm are:
Case a) $\tau_1 \simeq 33.99$, $\tau_2 \simeq 34.15$: as expected both τ_1 and τ_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) $\tau_1 = 0$, $\tau_2 \simeq 33.91$: τ_1 runs to zero and τ_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 τ_1, τ_2 and J at each iteration of the steepest descent algorithm for Case a).

Figures 3-4 depict the values assumed by J, and (τ_1, τ_2), at each iteration step for Case a). The behaviour of the cost functional J in the plane (τ_1, τ_2) is reported for Case b) in Figure 5, to confirm the goodness of the steepest descent algorithm.

Iteration	τ_1	τ_2	J
1	2	5	6259043.87
2	8.39	16.58	5975423.18
3	13.50	23.53	5840830.02
4	17.59	27.70	5771879.74
5	20.87	30.20	5733613.88
6	23.49	31.70	5711361.84
7	25.58	32.61	5697897.12
8	27.26	33.15	5689631.94
9	28.60	33.48	5684376.46
10	29.67	33.67	5681060.69
11	30.53	33.79	5678934.30
12	31.21	33.86	5677581.40
13	31.76	33.90	5676726.63
14	32.20	33.93	5676179.55
15	32.55	33.95	5675820.15
16	32.84	33.96	5675585.43
17	33.06	33.97	5675437.89
18	33.24	33.98	5675344.61
19	33.39	33.98	5675282.03
20	33.50	33.99	5675237.87
21	33.59	34.00	5675214.97
22	33.67	34.00	5675196.57
23	33.72	34.00	5675187.45
24	33.77	34.01	5675179.41
25	33.80	34.02	5675175.09
26	33.82	34.03	5675171.17
27	33.84	34.05	5675169.43
28	33.85	34.06	5675169.43
29	33.86	34.07	5675167.90
30	33.88	34.07	5675166.61
31	33.89	34.08	5675165.43
32	33.89	34.09	5675165.43
33	33.90	34.10	5675164.42
34	33.91	34.10	5675164.42
35	33.92	34.10	5675163.58
36	33.92	34.11	5675163.57
37	33.93	34.11	5675163.57
38	33.94	34.12	5675162.87
39	33.94	34.12	5675162.87
40	33.94	34.12	5675162.87
41	33.95	34.12	5675162.30
42	33.95	34.13	5675162.30
43	33.96	34.13	5675162.30
44	33.96	34.13	5675162.30
45	33.96	34.13	5675161.85
46	33.97	34.13	5675161.85
47	33.97	34.14	5675161.85
48	33.97	34.14	5675161.85
49	33.97	34.14	5675161.85
50	33.97	34.14	5675161.85
51	33.98	34.14	5675161.85
52	33.98	34.14	5675161.50
53	33.98	34.14	5675161.50
54	33.98	34.14	5675161.50
55	33.99	34.14	5675161.50
56	33.99	34.14	5675161.50
57	33.99	34.15	5675161.50

TABLE I: Case a: values of τ_1, τ_2, and corresponding J in 57 iterations of the steepest descent algorithm.

Notice that since J decreases when the number of iteration increases, the optimization of τ_1 and τ_2 allows an effective decrement of queues sizes.

The values of τ_1 and τ_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 (τ_1, τ_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.

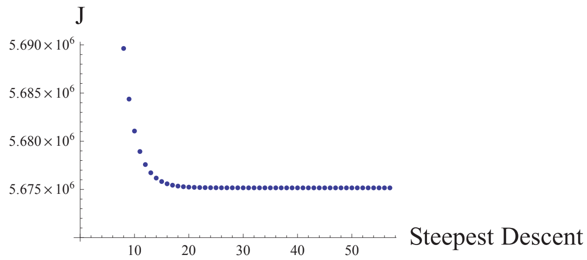

Fig. 3. Case a. J versus iteration steps.

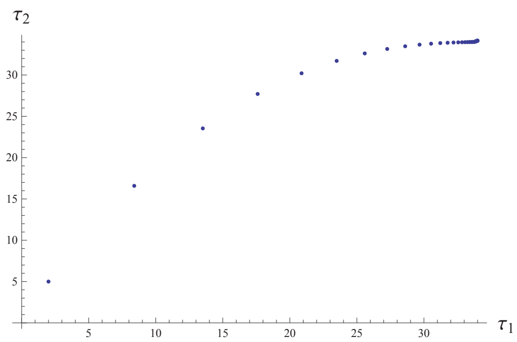

Fig. 4. Case a. "path" followed by the steepest descent algorithm in the plane (τ_1, τ_2).

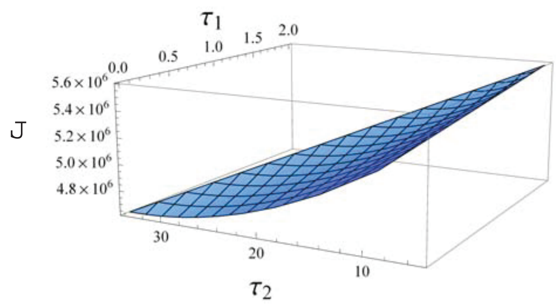

Fig. 5. Case b. Behaviour of J in the plane (τ_1, τ_2).

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 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

D. Armbruster, P. Degond and C. Ringhofer, *A model for the dynamics of large queueing networks and supply chains*, SIAM Journal on Applied Mathematics, 66(3) (2006), pp. 896–920.

D. Armbruster, P. Degond and C. Ringhofer, *Kinetic and fluid models for supply chains supporting policy attributes*, Bulletin of the Inst. Math., Academica Sinica, 2 (2007), pp. 433–460.

A. Bressan, *Hyperbolic Systems of Conservation Laws - The One-dimensional Cauchy Problem*, 250 pages, Oxford Univ. Press, 2000.

A. Bressan, G. Crasta and B. Piccoli, *Well-Posedness of the Cauchy Problem for $n \times n$ Systems of Conservation Laws, Memoirs of the American Mathematical Society*, 146 (694), 2000.

G. Bretti, C. D'Apice, R. Manzo and B. Piccoli, *A continuum-discrete model for supply chains dynamics*, Netw. Heterog. Media, 2(4) (2007), pp. 661–694.

A. Cutolo, B. Piccoli and L. Rarità, *An Upwind-Euler scheme for an ODE-PDE model of supply chains*, Siam Journal on Scientific Computing, 33(4) (2011), pp. 1669–1688.

C. D'Apice, S. Goettlich, H. Herty and B. Piccoli, *Modeling, Simulation and Optimization of Supply Chains*, 216 pages, SIAM, Philadelphia PA, USA, 2010.

C. D'Apice and R. Manzo, *A fluid dynamic model for supply chains*, Netw. Heterog. Media, 1(3) (2006), pp. 379–398.

C. D'Apice, R. Manzo and B. Piccoli, *Modelling supply networks with partial differential equations*, Quarterly of Applied Mathematics, 67(3) (2009), pp. 419–440.

C. D'Apice, R. Manzo and B. Piccoli, *Optimal input flows for a PDE-ODE model of supply chains*, Communication in Mathematical Sciences, 10(36) (2012), pp. 1226–1240.

C. D'Apice, R. Manzo and B. Piccoli: *Numerical schemes for the optimal input flow of a supply-chain, SIAM Journal on Numerical Analysis*, 51(5) (2013), pp. 2634–2650.

J.W. Forrester, *Industrial dynamics*, 464 pages. MIT Press, Cambridge, MA, 1964.

M. Garavello and B. Piccoli, *Traffic flows on networks*, American Institute of Mathematical Sciences, 243 pages, Springfield, MO, 2006.

S. Göttlich, M. Herty and A. Klar, *Network models for supply chains, Communication in Mathematical Sciences*, 3(4) (2005), pp. 545–559.

S. Göttlich, M. Herty and A. Klar, *Modelling and Optimization of Supply Chains on Complex Networks*, Communication in Mathematical Sciences, 4(2) (2006), pp. 315–330.

S. Göttlich, M. Herty and C. Ringhofer, *Optimization of order policies in supply networks*, European J. Oper. Res., 202(2) (2010), pp. 456–465.

M. Herty and A. Klar, *Modeling, Simulation, and Optimization of Traffic Flow Networks*, SIAM J. Sci. Comput., 25(3) (2003), pp. 1066–1087.

M. Herty, A. Klar and B. Piccoli, *Existence of solutions for supply chain models based on partial differential equations*, SIAM J. Math. An., 39(1) (2007), pp. 160–173.

D. Helbing, S. Lämmer, T. Seidel, P. Seba and T. Platkowski, *Physics, stability and dynamics of supply networks*, Physical Review E 70, 066116 (2004).

C. Kirchner, M. Herty, S. Göttlich and A. Klar, *Optimal control for continuous supply network models*, Netw. Heterog. Media, 1(4) (2006), pp. 675–688.

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, CEMSAC - Centro di Eccellenza su Metodi e Sistemi per Aziende Competitive. He is author of 2 books and more them 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, Italy. His current research activities focus on queuing theory, operation research and mathematical modeling of complex systems. His email 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.