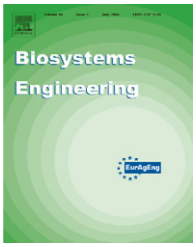
ELSEVIER

Available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/issn/15375110



Research Paper: PH—Postharvest Technology

Optimisation of fresh-food supply chains in uncertain environments, Part I: Background and methodology

F. Dabbene[a], P. Gay[b,*], N. Sacco[c]

[a]IEIIT, CNR, Politecnico di Torino, 24 Corso Duca degli Abruzzi, 10129 Torino, Italy
[b]DEIAFA, Università degli Studi di Torino, 44 Via Leonardo da Vinci, 10095 Grugliasco (TO), Italy
[c]DIMSET, Università di Genova, 1 Via Montallegro, 16145 Genova, Italy

ARTICLE INFO

Article history:
Received 31 January 2007
Received in revised form
29 October 2007
Accepted 23 November 2007
Available online 18 January 2008

The design of the supply chain, and in particular of the distribution phase, for fresh-food products, such as fresh and fresh-cut produce, fruit or meat, cannot be achieved without considering the perishable nature and the variability of the products entering the chain.

Motivated by these considerations, this paper presents a novel approach for the optimisation of fresh-food supply chains that manages a trade-off between logistic costs and some indices measuring the quality of the food itself as perceived by the consumer, such as ripeness, microbial charge or internal temperature. The supply chain and the behaviour of the product during its delivery are described using a hybrid model consisting of two specific parts. The first part takes into account event-driven dynamics (typically product handling) while the second one describes time-driven dynamics (the dynamics of some parameters characterising the food product in the supply chain). The performance of the supply chain, expressed in terms of both logistic costs and final product quality, are then enhanced using a specific optimisation algorithm that uses the model to assure the feasibility of the proposed optimal solutions.

In a companion paper [Dabbene F; Gay P; Sacco N (2008). Optimisation of Fresh-Food Supply Chains in Uncertain Environments, Part II: a Case Study. Biosystems Engineering, accepted], this new methodology is applied to a real-world example concerning meat refrigeration.

1. Introduction

The supply chain, and more specifically the distribution of fresh-food products such as meat, vegetables, fruits and dairy from producer to vendor, is in general a complex process, owing to the perishable nature of these agricultural products. In food supply chains, in fact, there is a continuous change in the quality from the time the raw material leaves the grower (or e.g. the slaughterer for meat products) to the time the product reaches the consumer. This phase contributes considerably to the determination of the final cost of the product as well as to the quality perceived by the consumer. It is therefore of great importance to design and manage the distribution chain in order to deliver the product at the right time, while guaranteeing the desired quality level. Moreover, this should be done keeping as low as possible the costs associated with the handling (storage, cooling, etc.) of the product itself. In addition, the presence of unavoidable biological variability in the products and the uncertainty affecting some aspects of the delivering

[*]Corresponding author.
E-mail addresses: fabrizio.dabbene@polito.it (F. Dabbene), paolo.gay@unito.it (P. Gay), nicola.sacco@unige.it (N. Sacco).

1537-5110/$ - see front matter © 2007 IAgrE. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.biosystemseng.2007.11.011

Nomenclature

a_i	arrival time of the ith job at the first node
C^j	capacity of the jth node
$C(\theta_C,\theta_{NC})$	term of the cost function taking into account the costs due to the adopted operating condition θ_C
$c(k)$	Positive sequence converging to 0
$c_0,\beta_2,$	parameters of the step multiplier $c(k)$
$D(\theta_C,\theta_{NC})$	term of the cost function taking into account the logistic aspects of the chain
$D_E(\theta)$	cost due to earliness
$D_T(\theta)$	cost due to tardiness
d_i	desired due date of the ith job
$E_{\theta_{NC}}\{J(\theta_C,\theta_{NC})\}$	expectation of $J(\theta_C,\theta_{NC})$ with respect the pdf $p_v(v)$
$E(\theta_C)$	compact notation of the expectation $E_{\theta_{NC}}\{J(\theta_C,\theta_{NC})\}$
$\hat{E}_N(\theta_C)$	empirical approximation of $E(\theta_C)$
$\hat{E}_\pm(k)$	empirical value of the cost function taken at the parameters values $\theta_C(k)\pm c(k)\eta(k)$, respectively
$f(x(t),\tau,\theta)$	function, in general nonlinear, giving the first derivative of $x(t)$ knowing $x(t)$, τ, and θ
$g(x(t),\tau,\theta)$	function, in general nonlinear, giving the attributes $y(t)$ knowing $x(t)$, τ, and θ
$\dot{g}(t)=\frac{d}{dt}g(t)$	first derivative of a generic function g with respect of the time t
$J(\theta)$	cost function of the optimisation problem
$J(\theta_C, \theta_{NC})$	cost function of the optimisation problem with explicated arguments
K	maximum number of iteration in the optimisation process
m	number of jobs in the supply chain
$m^j(t)$	number of jobs in the jth cell at time t
N	total number of samples
n	number of nodes of the supply chain
$P(\theta_C,\theta_{NC})$	term of the cost function taking into account the achievement of the product attributes target
$p_i(t)$	position on the ith job at time t
$p_v(v)$	probability density function of the stochastic variable v
q	total number of parameters of the system
q_ω	Number of physical parameters of the system
q_C	Number of controllable parameters
q_{NC}	Number of non controllable parameters
$\mathbb{R}$	Field of real numbers
$\mathbb{R}^p$	Space of the vectors of real numbers with dimension p
S_ω	Hyper-rectangle defining the admissible values of ω
S_T	Set defining the admissible values of T
$s_i^j(T)$	Actual time ith job spends in the jth node, eventually taking into account finite capacity constraints
T	Subset of θ gathering the logistic parameters describing the way the jobs move into the supply chain
T_i^j	Time interval spent by the ith job in the jth cell
T_C	Subset of T gathering the controllable logistic parameters
T_{NC}	Subset of T gathering the non controllable logistic parameters
$\tilde{T}_i^j$	Not necessarily feasible value of the time interval spent by the ith job in the jth cell computed by the optimisation algorithm
$\tilde{T}_i$	Vector gathering the not necessary feasible time intervals $\tilde{T}_i^j$
$\tilde{T}_C(k)$	kth estimate of the optimal, not necessarily feasible, value of the logistic parameters
t	Time
v	Generic stochastic variable with zero mean
$v^{(k)}$	kth sample of the stochastic variable v
$w(k)$	Step size multiplier
w_0, W, β_1	Parameters of the positive sequence $w(k)$ which give the step multipliers
$x(t)$	Vector gathering the state variables of the different products and the states variables describing the interaction with the surrounding environment
$x[i]$	ith entry of the vector x
$\bar{y}_i$	Desired value of $y(t)$ at time τ_i^n
$y_i(t)$	Attribute of the ith job at time t
$y(t)$	Vector $\in \mathbb{R}^m$ gathering the attributes $y_i(t)$ of the m jobs at time t
$\tilde{y}(t)$	Reference values for $y(t)$, $\forall t$
$\delta(t)$	Dirac delta function
Θ_C	Set of admissible values of the controllable parameters
$[\cdot]^T$	Transposing vector operator
γ_C	Weight of the term $C(\theta_C,\theta_{NC})$ in the cost function
γ_D	Weight of the term $D(\theta_C,\theta_{NC})$ in the cost function
γ_P	Weight of the term $P(\theta_C,\theta_{NC})$ in the cost function
ζ_0	Initial value of $x(t)$
$\eta(k)$	Vector gathering the kth sample of q_c random values $\eta_i(k)$, $i=1,\ldots,q_C$
$[\eta^{-1}(k)]$	vector containing the inverse values of $\eta(k)$
$\eta_i(k)$	kth sample of the stochastic variable η_i
$\{\eta(k)\}$	sequence of the random vectors $\eta(k)$, $k=1,\ldots,K$
θ	vector of the parameters representing the operation conditions under which the network is running
θ_C	subset of θ gathering the controllable parameters
θ_C^*	optimal value of the controllable operation condition
$\theta_C(k)$	kth estimate of the optimal feasible value of the controllable operation condition
θ_{NC}	subset of θ gathering the non-controllable parameters
$\bar{\theta}_{NC}$	vector gathering the nominal values of the non-controllable parameters
$\tilde{\theta}_C(k)$	kth estimate of the optimal, not necessarily feasible, value of the controllable operation condition
θ_C^k	approximate optimal value of the controllable parameter computed by the optimisation algorithm
$\Pi_\theta[\cdot]$	projection operator which computes a feasible value of $\tilde{\theta}_C(k)$

$\Pi_{\omega}[\cdot]$	projection operator which computes a feasible value of $\tilde{\omega}_C(k)$
$\Pi_T[\cdot]$	projection operator which compute a feasible value of $\tilde{T}_C(k)$
σ_E, σ_T	weight for the earliness and tardiness costs terms
τ_i^j	Switching time, i.e., the time instant the ith job leaves the jth node
τ_i	Vector gathering the switching times of ith job
τ	Vector gathering the vectors τ_i of the switching times
ψ	Threshold reflecting a trade-off between the number of not useful iterations and the admissible local solution worsening at each iteration
ω	Subset of θ gathering the physical parameters representing some physical properties of the system
ω_C	subset of ω gathering the controllable physical parameters
ω_{NC}	subset of ω gathering the non controllable physical parameters
$\tilde{\omega}_C(k)$	kth estimate of the optimal, not necessarily feasible, value of the physical parameters

process render the management of this phase even more complex.

A supply chain represents the sequence of activities performed in order to deliver the fresh product to a destination with the highest possible quality (Tijskens *et al.*, 2001). Any activity performed in the chain has a potential impact on the product, due to the interaction between the surrounding environment and the product itself (Apaiah *et al.*, 2005; Broekmeulen, 2001). In the general case, the evolution of the product characteristics is governed by the interaction of the product itself with the plant and, eventually, with other products stored in the same place. This is the case, for instance, for apples stored in bins inside a refrigeration cell, for meat carcasses in a chilling tunnel or for the processing and distribution of fresh-cut produce.

With the aim of deriving a mathematical model of the chain, each product can be considered as an “object” described by a dynamical model which takes into account the physiological processes occurring in the product itself. These processes are generally affected by the conditions (e.g. temperature, humidity, etc.) in the immediate surroundings of the product. At the same time, the products may themselves affect the immediate environment. The objective of a modelling process should be to describe the product behaviour as a collection of interacting processes, such that their combined action can describe the observed phenomenon and such that each sub-process can be fully understood in its description. The nature of the sub-process is largely defined by means of fundamental physical laws and the generally accepted rules in a particular discipline. A typical approach, see e.g. Sloof and Everest (2000), consists of separating high-level processes analysing intrinsic product properties that, in general, correspond to quality attributes in the phenomenon under study. Some interesting modelling approaches can also be found in Minegishi and Thiel (2000) and Georgiadis *et al.* (2005).

The design of a supply chain cannot be effected node by node, but the entire chain has to be considered as a whole. As explained well in Gigiel (1996), the poor design encountered in many existing networks is mainly due to a mismatch between the purposes the nodes were originally designed for and how they are actually used. Notice that, in the usual practice, the first step in the design of a node in a supply chain (e.g. a chiller, a storage cell or transportation) is for the user to draw up specifications, in terms of node load and throughput. For determining these quantities, a viable way may consist of developing and tuning a mathematical model for the entire chain, able to capture the dynamics present in the chain. Once such a model has been determined and its parameters have been identified, it is then possible to simulate different scenarios and eventually proceed to an optimisation phase.

Unfortunately, the nominal operating conditions, expressed in terms of product flow, machine loads, plant efficiency, raw food material properties, etc., with respect to which the chain has been designed, can vary in time due to uncontrollable factors affecting the supply chain efficiency and the final product quality. In a real supply chain, in fact, there are many factors that can introduce uncertainty. First of all the natural variance of biological products may render the behaviour (time-driven dynamics) of each food product slightly different. In many applications this kind of variability cannot be neglected and could have a great influence on the observed system (see e.g. Hertog, 2002; Hertog *et al.*, 2004; Peirs *et al.*, 2002). Second, the process conditions in each node can vary and, third, discrete-events dynamics could be perturbed by external and unmodelled factors or disturbances.

The role of uncertainty and the ways to manage it in post-harvest and food engineering processes have been recently considered in many works (see e.g. Van Impe *et al.*, 2001; references therein). In the so-called second-order modelling approach (Nauta, 2000), for example, the perturbations affecting empirical data and/or model parameters are subdivided into uncertainty and/or variability. Uncertainty represents the lack of perfect knowledge of a parameter value, which may be reduced by additional measurements, further improvement of a measurement method (e.g. detection limit, precision) or, if applicable, model structure improvement (Delignette-Muller & Rosso, 2000). Variability, on the other hand, represents a true heterogeneity of the population that is a consequence of the physical system and is irreducible by additional measurements. However, since in a quantitative study it is often difficult to separate variability and uncertainty (Nauta, 2000), especially when both have the same order of magnitude, an alternative is to globally characterise them by associating a probability function to each quantitative parameter (Delignette-Muller & Rosso, 2000). Different methods have been proposed to quantify the effects of the propagation of the uncertainty affecting model parameters on the output of the studied system. One widely used method is the Monte Carlo method (see e.g. Nicolaï *et al.*, 1998; Demir *et al.*, 2003; Poschet *et al.*, 2003). The main drawback of this technique may be the large number of repetitive simulations necessary to obtain an acceptable level of accuracy and the fact that the distribution over the data points must be completely specified in a probabilistic sense.

Other methods have been suggested to provide computationally attractive alternatives for specific applications, like, for example, for distributed parameter systems with parameter fluctuations in space (e.g. the first-order perturbation algorithm; see Nicolaï *et al.*, 2000) or simultaneously in space and time (e.g. the variance propagation algorithm; see Nicolaï *et al.*, 1998 and Scheerlinck *et al.*, 2001). Since the objective of this research is methodology-oriented, only the (overall) perturbation on parameters is considered without attempting to distinguish between uncertainty and variability.

The aim of this paper is to introduce a new modelling framework that allows the discrete-event behaviour of the logistics of the chain and the continuous-time dynamics of some product characteristics to be merged, taking into account uncertainties, and then to propose a dynamic model-based optimisation method that allows the performances of the chain to be improved. The proposed hybrid model is used for developing the optimisation strategy aimed at finding an optimal operation point for the chain, in terms of optimal scheduling and/or proper design of one or more node specifications.

The problem of supply-chain optimisation has been considered in many recent papers (see e.g. Altiparmak *et al.*, 2006; Apaiah & Hendrix, 2005; Beamon, 1998; Mo *et al.*, 2005; Perea-López *et al.*, 2003; Shapiro, 2004). All these contributions deal with the supply chains of generic manufactured or assembled products and do not consider the case of products that can perish or, in general, change some of their attributes while they are managed by the chain. In the specific field of food engineering, the first approaches proposing the use of optimisation algorithms for improving food processes can be traced back to the work of Teixeira and Shoemaker (1988). For a recent survey on the use of modern optimisation in food processing, the reader is referred to Banga *et al.* (2003), and the references therein. Quoting Banga *et al.* (2003): "Model-based simulation of food processing units and/or full plants has received great attention during the past three decades, especially in academic environments (Datta, 1998; Fryer, 1994; Nicolaï *et al.*, 2000, Nicolaï & Van Impe, 1996; Van Impe, 1996). Since most processes are operated in batch mode, these models are usually dynamic in nature, consisting of sets of ordinary and/or partial differential and algebraic equations (…) Thus, these mathematical models usually consist of sets of algebraic, partial and ordinary differential equations (PDAEs), with possible integral equations, and sometimes even logic conditions (modeling discrete events and/or transitions, i.e. hybrid systems)." As suggested also in Barton *et al.* (2000) and García *et al.* (2006), a modern approach to tackle such dynamic optimisation problems is to transform the original problem into a nonlinear programming problem (NLP): this is exactly the approach adopted in this paper. Then, various local/global optimisation techniques can be employed to solve this problem numerically, as discussed in detail in Section 3.

A somewhat similar approach can also be found in the classical literature on integrated manufacturing systems. For instance, Cassandras *et al.* (2001) optimised a manufacturing process consisting of a series of nodes that perform a sequence of operations on a set of jobs. Such operations modify some physical characteristics of the jobs. However, such an approach cannot be directly applied in our framework since it does not allow the single nodes to process more than one job at a time, which is the typical situation arising in fresh-food supply chain. Moreover, this paper—as mentioned previously —considers the situation in which not only the operations performed in the nodes modify some physical characteristic of the single job, but the jobs may influence each other and also the way they influence the node itself. The presence of various variability sources affecting both product characteristics and logistic parameters has been explicitly taken into account.

The paper is structured as follows: in Section 2 a thorough theoretical analysis is carried out and a hybrid dynamic model describing the supply chain is derived. In Section 3 the proposed optimisation algorithm is introduced and its salient features (cost function definition, uncertainty handling and feasibility issues) are discussed. Finally, conclusions are drawn in Section 4. In the second part of this paper (Dabbene *et al.*, 2008), the proposed methodology is applied to a case study of a fresh-meat supply chain, showing the concrete effectiveness of the introduced framework.

2. Theoretical analysis and model development

This section presents the analytic framework for modelling a fresh-food supply chain. The proposed model is hybrid, consisting of a part with time-driven dynamics—the dynamics of some parameters characterising the products in the supply chain—and a part with event-driven dynamics (see e.g. Cassandras & Lafortune, 1999)—the logistic aspects of the chain. To keep the formalism at a general level, the term *job* will be used through the paper to refer to the generic portion or unit of food to be treated individually (e.g. a bin of fresh fruits, a meat carcass, a lot of fresh cut salad, etc.), and the term *nodes* to refer to the servers in which different activities are performed on the jobs (e.g. refrigeration or storage cells, washing or cutting machines, transportation, etc.).

2.1. *Preliminary notation and definitions*

Consider a supply chain consisting of a network formed by n successive nodes in which the different jobs are processed in a sequential way. The total number of jobs that enter the chain is m. The jobs enter the first node at different instants in time (and with possibly different initial conditions of the parameters describing them) and are processed sequentially from node 1 to node n. Every node can process more than one job at a time. The number of jobs processed in a node at a specific instant could be unbounded (infinite process capacities) or, more realistically, bounded (finite process capacities). In the case of unbounded capacity, a job immediately leaves a node and enters the next one as soon as it has been processed. On the other hand, if a node possesses only finite capacity and has no slot available, any new job has to wait in the preceding node until another job leaves the node. Both cases of infinite and finite capacity are addressed in the paper.

Associated with the ith job define a measure $y_i(t)$, called *attribute*, which represents some characteristic (e.g. internal temperature, firmness, ripeness, microbial charge, etc.) that is chosen as representative of the product quality. The attributes $y_i(t)$ of the different jobs vary in time according to a differential equation which depends on the operating conditions of the network and on the different jth node where the job is being processed at time t.

The supply chain is characterised by a vector of parameters $\theta \in R^q$ that represent the operating conditions under which the network is running. Two different types of parameters θ are considered, on the basis of their interpretation. In particular, a distinction is made between *physical* parameters, which represent some physical properties of the nodes in the network (such as power of a refrigeration cell or air composition in controlled atmosphere environments), and the *logistic* parameters of the chain, describing the way in which the various jobs move into the chain. To this end, the vector $\theta \in R^q$ is partitioned in the following way:

$$\theta = \begin{bmatrix} \omega \\ T \end{bmatrix}, \quad \omega \in \mathbb{R}^{q_\omega}, \quad T \in \mathbb{R}^{nm}, \quad q = q_\omega + nm \tag{1}$$

where ω is the *physical parameters vector* and T is the *logistic parameters vector,* which is given by the vector of time intervals

$$T = [T_1^1 \quad \cdots \quad T_1^n \quad \cdots \quad T_i^j \quad \cdots \quad T_m^1 \quad \cdots \quad T_m^n]^{\mathrm{T}} \tag{2}$$

The time interval T_i^j represents the time spent by job i in node j in the case of an infinite capacity network. In the more general case of finite capacity, the precise meaning of T_i^j is clarified in Section 2.3.2.

2.2. Controllable and non-controllable parameters

It is important to note that some of the parameters of the chain (both physical and logistic ones) may be directly imposed by the network manager, and can therefore be considered *control variables*, while other parameters are not directly accessible. In a general setting, these *uncontrollable* parameters may be either fixed (deterministic) and possibly not perfectly known, or stochastic, subject to random variations. For what concerns physical parameters, examples of uncontrollable parameters may be the power of already existing plants, external temperature, etc. Uncontrollable logistic parameters account for the fact that some nodes could require a minimum amount of time to process the job. This is the case, for instance, for transportation times or unit operations for which only information about *minimum processing time* can be given. Following this distinction, the following two vectors are introduced:

$$\theta_C = \begin{bmatrix} \omega_C \\ T_C \end{bmatrix} \in \mathbb{R}^{q_C} \quad \text{and}$$
$$\theta_{NC} = \begin{bmatrix} \omega_{NC} \\ T_{NC} \end{bmatrix} \in \mathbb{R}^{q_{NC}}, \quad q_C + q_{NC} = q, \tag{3}$$

where ω_C, T_C and ω_{NC}, T_{NC} represent respectively the vectors of controllable and uncontrollable physical and logistic parameters.

The controllable parameters, either the physical or the logistic ones, are constrained within specific intervals. In particular, the controllable physical parameters can assume values only in given intervals, that is, ω_C should lie in a given hyper-rectangle S_ω. On the other hand, the logistic parameters can assume only values $T_C \in S_T$, S_T being the set of all possible residence times that do not violate the capacity constraint (see for details Section 2.3.2). The above constraints are summarised by introducing the notation $\theta_C \in \Theta_C$.

2.3. Hybrid model derivation

The following sections provide the theoretical derivation of the hybrid model proposed for describing the dynamical behaviour of a food supply chain. In particular, the discrete-event dynamics of the supply chain are introduced in the first two following sections, while the time-driven dynamics of the attributes $y_i(t)$ are discussed in Section 2.3.3.

2.3.1. Discrete-event dynamics

To describe the dynamics of the discrete-event system, define first the *event-occurrence time* τ_i^j as the time instant in which the ith job leaves the jth node. Then, for notation ease, it is useful to introduce the vectors

$$\tau_i = \begin{bmatrix} \tau_i^0 & \tau_i^1 & \cdots & \tau_i^n \end{bmatrix}^{\mathrm{T}}, \quad i = 1, \ldots, m, \tag{4}$$

which gather the switching times of the ith job, and the composite vector

$$\tau = \begin{bmatrix} \tau_1^{\mathrm{T}} & \tau_2^{\mathrm{T}} & \cdots & \tau_m^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \quad \tau \in \mathbb{R}^{m(n+1)}$$

With this notation settled, the discrete-event dynamics can be formally stated by means of the following recursion, for $i = 1,\ldots,m$ and $j = 1,\ldots,n$

$$\begin{aligned} \tau_i^j &= \tau_i^{j-1} + s_i^j(T) \\ \tau_i^0 &= a_i \end{aligned} \tag{5}$$

where a_i denotes the arrival time of the ith job in the first node. The function $s_i^j(T)$ depends on the logistic parameters T and is introduced to take into account the case of finite capacity. This function provides the *actual time* the ith job spends in the jth node. Therefore, for an infinite capacity network, it results that

$$s_i^j(T) = T_i^j, \tag{6}$$

where T_i^j belongs either to T_C or T_{NC}. Consequently, Eq. (5) simply becomes

$$\begin{aligned} \tau_i^j &= \tau_i^{j-1} + T_i^j \\ \tau_i^0 &= a_i \end{aligned} \tag{7}$$

The explicit derivation $s_i^j(T)$ for the case of networks with finite capacity is discussed in Section 2.3.2.

Some useful parameters that describe the behaviour of the chain and that directly depend on the evolution of the events τ_i^j are now introduced.

First, define the *position* $p_i(t)$ as the node in which the ith job is at time t. Then, the evolution of the variables $p_i(t)$, $i = 1,\ldots,m$ can be represented by means of the following differential equation:

$$\dot{p}_i(t) = \sum_{j=0}^{n-1} \delta(t - \tau_i^j)$$
$$p_i(0) = 0 \tag{8}$$

where $\delta(\cdot)$ is the Dirac delta function. Clearly, the variable $p_i(t)$ can assume only the integer values $0,1,2,\ldots,n$ and is a monotonically non-decreasing function of time. An example of the possible evolution of $p_i(t)$ is depicted in Fig. 1.

Analogously, at each time instant t, $m^j(t)$ denotes the total number of jobs being processed in the jth node at time t. This quantity is governed by the following differential equation:

$$\dot{m}^j(t) = \sum_{i=1}^{m} \delta(t - \tau_i^{j-1}) - \sum_{h=1}^{m} \delta(t - \tau_h^j)$$
$$m^j(0) = 0 \tag{9}$$

This latter equation explicitly shows that the number of jobs in the jth node increases when a generic job i leaves the (j–1)th node at τ_i^{j-1} and enters the jth node and decreases when a generic job h (not necessarily the ith one) leaves the jth node at τ_h^j, as depicted in Fig. 2.

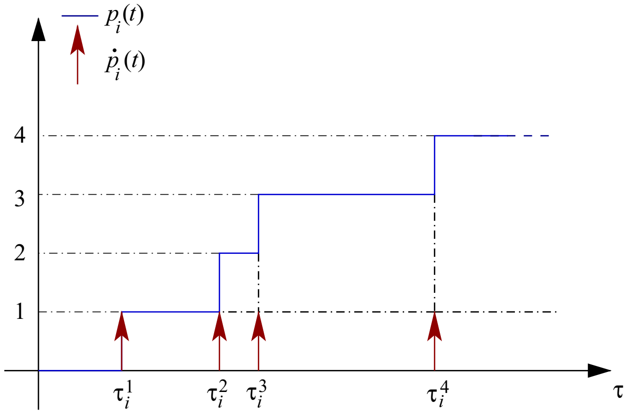


Fig. 1 – Example of the trajectory of the function $p_i(t)$ describing the position of the generic ith job.

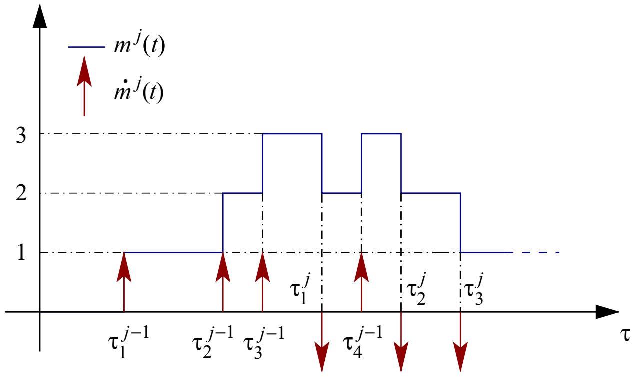


Fig. 2 – Example of a generic trajectory of $m^j(t)$ and $\dot{m}^j(t)$.

Eqs. (8) and (9) fully describe the dynamic behaviour of the network, providing, at each time instant, the number of the jobs present in each node. In the next section, an explicit form for the function $s_i^j(T)$ appearing in (5) is formally derived in the case of networks with finite capacity.

2.3.2. *Finite Capacity Constraints*

As previously mentioned, a more realistic description of the considered systems should take into account the case of finite capacity of the supply chain nodes. To this end, let $C^j > 0$ be the capacity of the jth node, $j = 1,\ldots,n$. Then, it follows that a generic job i can leave the jth node at the time τ_i^jonly if

$$m^{j+1}(\tau_i^j) < C^{j+1} \tag{10}$$

that is, only if there is at least one free slot in the (j+1)th node at time τ_i^j.

Taking into account such capacity constraints, the function $s_i^j(T)$ appearing in the event-driven dynamics described by (5) takes the more involved form

$$s_i^j(T) = \begin{cases} T_i^j & \text{if } m^{j+1}(\tau_i^{j-1} + T_i^j) < C^{j+1} \\ \min\limits_h \left\{ \tau_h^{j+1} | \tau_h^{j+1} > \tau_i^{j-1} + T_i^j \right\} - \tau_i^{j-1} & \text{if } m^{j+1}(\tau_i^{j-1} + T_i^j) = C^{j+1} \end{cases} \tag{11}$$

This function gives the time spent by the ith job in the jth node in both cases when the (j+1)th node has at least a free slot or not. In particular, the first line of (11) coincides with (7) in the case when free space is available in the node (j+1). On the other hand, the second equation in (11) says that the ith job can leave the jth node as soon as another job departs from the (j+1)th node.

2.3.3. *Time-driven dynamics*

Denote by $y(t) = [y_1(t)\ y_2(t) \ldots y_m(t)]^T$ the vector of attributes. As already mentioned, the attribute $y_i(t)$ relative to the ith job evolves in time according to a (usually nonlinear) differential equation. Consequently, $y(t)$ can be seen as the output of a system of differential equations of the type

$$\begin{cases} \dot{x}(t) = f(x(t), \tau, \theta) \\ x(0) = \varsigma_0 \end{cases}$$
$$y(t) = g(x(t), \tau, \theta) \tag{12}$$

$x(t)$ being a vector gathering the state variables of the different products and the state variables that describe the interaction with the surrounding environment. Notice that Eqs. (5) and (12) define in all aspects a hybrid system, where the time-driven dynamics of (12) depend on the vector of events τ, whose dynamics are expressed by the recursion (5). From a different perspective, (12) may be seen as a switching system, whose switching times are regulated by the recursion (5).

2.3.4. *An illustrative example*

In this section, a simple example of fresh-food supply chain is described to clarify the model and notation introduced above. Consider a fresh-cut produce supply chain managing two stocks of different ready-to-eat salads. This supply chain consists of three nodes: the first one represents the producer, where the two stocks of fresh-harvested salad are washed, cut, and packed. The arrival times of the two stocks have been assumed to be $a_1 = 7$ p.m. and $a_2 = 8$ p.m. The second node represents the transportation of the processed stocks of salad from the producer to the retailer, while the third one is the retailer.

For what concerns the intervals T_i^j, $i=1,2$, $j=1,2,3$, we assume the following: the producer processing times are $T_1^1=T_2^1=10\,\text{h}$, the transportation times are $T_1^2=T_2^2=2\,\text{h}$, and, finally, the time the produce may spend in the retailer node is $T_1^3=T_2^3=84\,\text{h}$ (3.5 days). Note that such values fulfil the constraint of keeping the whole process time under 96 h (4 days) to guarantee a commercial quality of the salad, that is

$$\sum_{j=1}^{3} T_i^j \leqslant 96\,\text{h}, \quad i=1,2. \tag{13}$$

Then, given the time the fresh cuts arrive at the producer, it is immediately possible to compute the switching times by means of the recursive Eq. (7), obtaining for both jobs

$$\begin{array}{lcl} \tau_1^0 = 19:00 \ \text{day} \ 1 & & \tau_2^0 = 20:00 \ \text{day} \ 1 \\ \tau_1^1 = 05:00 \ \text{day} \ 2 & \text{and} & \tau_2^1 = 06:00 \ \text{day} \ 2 \\ \tau_1^2 = 07:00 \ \text{day} \ 2 & & \tau_2^2 = 08:00 \ \text{day} \ 2 \\ \tau_1^3 = 19:00 \ \text{day} \ 5 & & \tau_2^3 = 20:00 \ \text{day} \ 5 \end{array} \tag{14}$$

The relevant Gantt diagram is reported in Fig. 3 while the diagrams of $m^j(t)$, $j=1,2,3$, and $p_i(t)$, $i=1,2$, are reported in Figs. 4 and 5, respectively.

3. Optimisation

The model described in the previous section discriminates the possible behaviours that the system can exhibit acting on the controllable parameters θ_C. The role of the network manager consists of choosing the best operating conditions considering different aspects such as operating expenses, product and process conditions and the final product quality.

3.1. *Performance function*

The goal of the optimisation algorithm is to choose the controllable parameters θ_C in order to minimise an objective function that measures the performance of the supply chain. In particular, introduce a performance function J constituted by the sum of three terms

$$J(\theta) = J(\theta_C, \theta_{NC}) = \gamma_C C(\theta_C, \theta_{NC}) + \gamma_P P(\theta_C, \theta_{NC}) + \gamma_D D(\theta_C, \theta_{NC}) \tag{15}$$

The first term takes into account the cost related to the particular operating condition θ_C (e.g. power consumption, transport costs, etc.). The second term accounts for the achievement of a target performance, measured in terms of product attributes. These could be expressed in different ways depending on the specific product. For instance, the

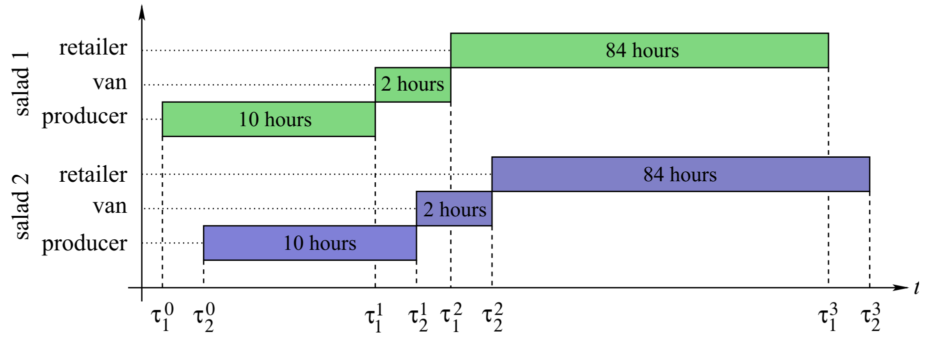


Fig. 3 – Gantt diagram representing the time spent by two stocks of salad in the three different cells of the supply chain. In this example the capacity has been assumed to be infinite.

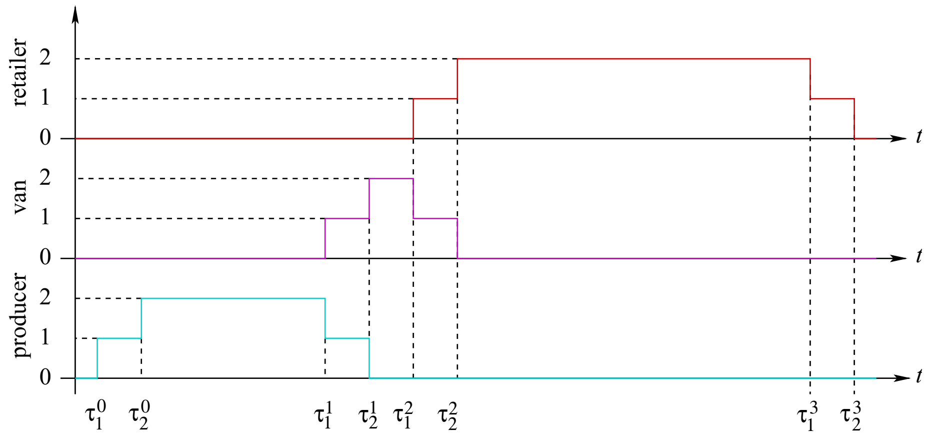


Fig. 4 – Representation of the function $m^j(t)$, $j=1,2,3$ for the considered example.

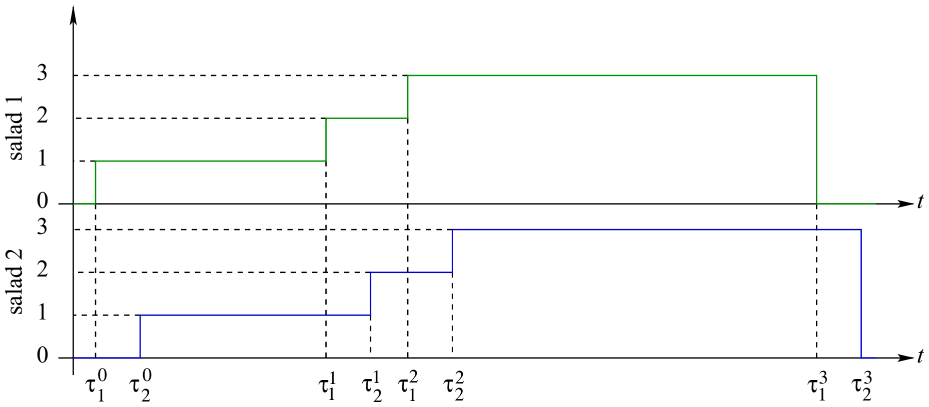


Fig. 5 – Representation of the function $p_i(t)$, $i = 1,2$, of the two stocks of salad for the considered example.

following requirements may be considered:

1. *Trajectory tracking*: $y(t)$ should follow a given reference $\tilde{y}(t)$; namely $P(\theta_C, \theta_{NC}) = ||y(t) - \tilde{y}(t)||$, being $\|\cdot\|$ a norm (for instance $\|\cdot\|_2$, $\|\cdot\|_\infty$, etc.). This is a typical requirement for instance in the case of meat-conditioning processes.
2. *Final value objective*: $y_i(t)$ should achieve the desired value $\bar{y}_i$ when leaving the last node at τ_i^n . This objective can be imposed by setting $P(\theta_C, \theta_{NC}) = ||y_i(\tau_i^n) - \bar{y}_i||$. This is the case, for example, of the ripeness level that the fruit should have when reaching the shelf.

Notice that, since the behaviour of the attribute $y(t)$ directly depends on the operating conditions θ, requirements 1 and 2 translate into a cost term $P(\theta_C, \theta_{NC})$ which is a function of θ_C and θ_{NC}.

The third term in (15) is related to the logistic aspects of the chain and measures the difference between the actual final time τ_i^n and the desired due-date d_i by means of costs of earliness $D_E(\theta) = \sum_{i=1}^{m}(\max\{0, d_i - \tau_i^n\})^{\sigma_E}$ or tardiness $D_T(\theta) = \sum_{i=1}^{m}(\max\{0, \tau_i^n - d_i\})^{\sigma_T}$, where σ_E and σ_T are suitable integers which allow to consider linear, quadratic, etc., earliness and tardiness costs.

An important point that should be taken into consideration is the fact that the cost function depends on the controllable parameters $\theta_C \in \Theta_C$, which are those that should be designed in order to optimise the behaviour of the chain, but also on the uncontrollable parameters θ_{NC}, whose effect should be accounted for. The simple situation when the uncontrollable parameters θ_{NC} assume fixed values $\bar{\theta}_{NC}$ is discussed first. In this case, the optimisation problem simply becomes

$$\min_{\theta_C \in \Theta_C} J(\theta_C, \bar{\theta}_{NC}) \qquad (16)$$

We refer the reader to Banga *et al.* (2003), for an excellent review on the state-of-the-art methodologies for model-based dynamic optimisation in the food processing literature. In particular, the main categories of optimisation techniques can be roughly classified as follows: local gradient-based optimisation techniques and global optimisation techniques. The latter may be *deterministic*, for instance branch-and-bound (see e.g. Floudas, 2000), or *stochastic* methods, which can be further divided into clustering methods, stochastic (adaptive) search and biologically or physically inspired methods (such as genetic algorithms or simulating annealing). It should be noted, however, that, for a generic formulation, no global optimisation algorithm can guarantee to find a global solution with certainty in finite time (Banga *et al.*, 2003). In principle, all these optimisation techniques may be applied to solve the optimisation problem (16). However, it should be noted that the situation considered in this paper is somewhat more general than the ones considered in the cited literature. In fact, we assume that the optimisation algorithm should be able to make decisions in an uncertain environment. Formally, this uncertainty is taken into account assuming that the uncontrollable parameters θ_{NC} are not fixed, but may vary according to a given distribution, as discussed in the next section. In this situation, a direct application of the above-listed techniques is not always possible, since most of them would require some *ad hoc* adjustments (for instance, by incorporating in some way the uncertainty in the cost function as a penalty). We propose instead an algorithm, based on the results in Spall (1992) which can be seen as a stochastic gradient technique combined with ad-hoc gradient estimation, that allows the presence of stochastic uncertainty to be taken into account in a direct way. The choice of this optimisation method has been motivated by the following reasons: (i) reduced computational requirements in terms of cost function evaluations, (ii) capability of handling uncertainty and (iii) ease of implementation. This approach is discussed in detail in the next section.

3.2. *Uncertainty in the supply chain*

In general, the vector of uncontrollable parameters θ_{NC} may be affected by random uncertainty. This is formally written as

$$\theta_{NC} \doteq \bar{\theta}_{NC} + v \qquad (17)$$

where $\bar{\theta}_{NC}$ represents in this case the (known) nominal value of the uncontrollable parameters, while v is a random vector with zero mean and is associated with the given probability density function (pdf) $p_v(v)$. With this assumption, the cost function $J(\theta_C, \theta_{NC})$ becomes a random quantity. Therefore, problem (16) needs to be formulated in a stochastic framework.

A frequently used requirement in this context is to optimise an "average" instance of the problem: in other words, one asks to minimise the expected value of the objective function taken with respect to the random uncertain parameters θ_{NC} (see for instance Kushner & Yin, 1997), that is

$$\min_{\theta_C\in\Theta_C} E(\theta_C), E(\theta_C)\doteq E_{\theta_{NC}}\{J(\theta_C,\theta_{NC})\} \tag{18}$$

where $E(\theta_C)$ is defined as the expectation of $J(\theta_C,\theta_{NC})$ taken with respect to the pdf $p_v(v)$. Notice that the above stochastic optimisation problem is in general very hard to solve, since the mere evaluation of the expected value $E(\theta_C)$ (even if an analytical expression of the cost function were available, which is not our case) would require the solution of a multiple integral. Recently, however, approaches based on uncertainty randomisation have proven their efficacy for the approximate solution of stochastic programs, see e.g. Tempo *et al.* (2004) and Vidyasagar (2002).

In this paper, following a similar philosophy, the expectation in (18) is approximated by introducing its empirical version. To this end, draw N independent identically distributed (iid) random samples of the uncertainty vector v

$$v^{(1)}, v^{(2)}, \ldots, v^{(N)}, \tag{19}$$

according to the density function $p_v(v)$, and construct the so-called *empirical mean*

$$\hat{E}_N(\theta_C)\doteq\frac{1}{N}\sum_{i=1}^{N} J\left(\theta_C,\bar{\theta}_{NC}+v^{(i)}\right) \tag{20}$$

As is well known in the Monte Carlo literature, from the Borel–Cantelli Lemma (see for instance Vidyasagar, 2002) it follows that the empirical mean (20) converges with probability one to the true mean defined in (18) when N goes to infinity. Hence, in the approach of this paper, the empirical approximation $\hat{E}_N(\theta_C)\approx E(\theta_C)$ is employed for building up a solution for the optimisation problem (18). In other words, $\hat{E}_N(\theta_C)$ is taken as "noisy measurements" of the cost function $E(\theta_C)$. Formally, one may write $E(\theta_C)=\hat{E}_N(\theta_C)+\upsilon$, where υ is a random variable with zero mean whose statistics depend on those of the uncertainty vector v.

The solution methodology proposed in this paper for the solution of problem (18) is a modification of a classical gradient descent method, in which the gradient of the cost function is not computed exactly, but is approximated using only a few function evaluations. In particular, the approach adopted here follows the one proposed by Spall (see for instance Spall, 1992, 2003) and tackles the problem via a simultaneous-perturbations stochastic approximation (SPSA) approach. This method approximates the gradient at each iteration using only two evaluations of the cost function. This allows the computation complexity of the problem to be reduced greatly. In detail, the general structure of an SPSA algorithm is based on a recursion in which successive approximations of the optimal value

$$\theta_C^*\doteq\arg\min_{\theta_C\in\Theta_C} E(\theta_C) \tag{21}$$

are sequentially constructed based on noisy observations of the cost function. Formally, let $\theta_C(k)$ denote the kth estimate of the optimal solution, and let $\{\eta(k)\}$ be a random sequence of column random vectors where $\eta(k)=[\eta_1(k)\ \eta_2(k)\ \ldots\ \eta_C(k)]^T$ are not necessarily identically distributed. The two-sided SPSA algorithm to update $\theta_C(k)$ is constructed as follows:

$$\theta_C(k+1)=\theta_C(k)-w(k)[\eta^{-1}(k)]\frac{\hat{E}_+(k)-\hat{E}_-(k)}{2c(k)} \tag{22}$$

where $c(k)$ is a positive sequence converging to zero, $w(k)$ is the step-size multiplier and $[\eta^{-1}(k)]$ is defined as the vector containing the inverses of the elements of $\eta(k)$. Notice that recursion (22) mimics a classical gradient descent method, where the gradient with respect to θ_C of the functional $E(\theta_C)$, which is not available, is approximated at each step using only two noisy evaluations of the cost function.

The values $\hat{E}_\pm(k)$ represent the empirical cost function evaluated at parameter values $\theta_C(k)\pm c(k)\eta(k)$, i.e.

$$\hat{E}_\pm(k)\doteq\hat{E}_N(\theta_C(k)\pm c(k)\eta(k)) \tag{23}$$

Various convergence results for this algorithm have been proven under different hypotheses, see for instance Spall (2003), Gerencsér *et al.* (2001), He *et al.* (2003) and references therein. In particular, it can be shown that the algorithm still converges when the empirical mean is constructed with a very small number of samples. Indeed, even a single sample is sufficient, i.e. $N=1$ in (20), thus allowing the algorithm to be simplified significantly, generating at each step a single instance $v^{(k)}$ of the uncertainty and letting

$$\hat{E}_\pm(k)\doteq\hat{E}_1(\theta_C(k)\pm c(k)\eta(k))=J(\theta_C(k)\pm c(k)\eta(k),\bar{\theta}_{NC}+v^{(k)}) \tag{24}$$

This approach of considering a single realisation of the uncertainty at each step is quite classical in the stochastic approximation literature. In this sense, recursion (22) can be seen as an approximate stochastic gradient approach; see for instance Kushner and Yin (1997).

3.3. *Building feasible solutions*

In general, the optimisation algorithm introduced in Section 3.2 provides solutions which may be not feasible for the problem at hand, since some physical parameter θ_C may not belong to their admissible set Θ_C, and/or the logistic parameters T may violate the finite capacity constraints. This section discusses how to modify the recursion (22) in order to manage such unfeasibility. To this end it is worth noting that, at each step of the recursion, a feasible vector $\theta_C(k+1)$ can be built starting from an unfeasible one $\tilde{\theta}_C(k+1)$ by means of a projection step. To this end, the projection operator $\Pi_\theta[\cdot]:\mathbb{R}^{q_C}\to\Theta_C$ is introduced into Eq. (22) and the recursion modified as follows:

$$\tilde{\theta}_C(k+1)=\theta_C(k)-w(k)[\eta^{-1}(k)]\frac{\hat{E}_+(k)-\hat{E}_-(k)}{2c(k)} \tag{25}$$

$$\theta_C(k+1)=\Pi_\theta[\tilde{\theta}_C(k+1)]. \tag{26}$$

Eqs. (25) and (26) represent a classical projection-based approach to constrained stochastic approximation which is treated extensively in Kushner and Yin (1997).

The construction of the projection operator $\Pi_\theta[\cdot]$ for building a feasible solution $\theta_C(k+1) = \begin{bmatrix} \omega_C(k+1) \\ T_C(k+1) \end{bmatrix}$ is discussed next.

Notice that the operator $\Pi_\theta[\cdot]$ should operate in two different ways for the physical and the logistic parameters, that is

$$\omega_C(k+1) = \Pi_\omega[\tilde{\omega}_C(k+1)] \tag{27}$$

$$T(k+1) = \Pi_T[\tilde{T}(k+1)] \tag{28}$$

where $\tilde{\omega}_C(k+1)$ and $\tilde{T}(k+1)$ are the (possibly unfeasible) solutions provided at the kth step by the SPSA algorithm defined in Eq. (25), and $\omega_C(k+1)$ and $T(k+1)$ are the projected (feasible) solutions of Eq. (26).

For what concerns the operator $\Pi_\omega[\cdot]$, its role is simply to compute the orthogonal projection of $\tilde{\omega}_C(k+1)$ into the set S_ω. Since usually the set S_ω consists of the union of intervals, its implementation is immediate.

Analogously, the operator $\Pi_T[\cdot]$ projects the solution with respect to the finite capacity requirements. However, the implementation of such an operation is in general non-trivial. A possible solution proposed here is to follow the developments of Section 2.3.2. Namely, given a vector $\tilde{T}(k+1)$, a set of feasible residence times may be computed as $T_i^j(k+1) = s_i^j(\tilde{T}_i^j(k+1))$, where the function $s_i^j(\cdot)$is defined in Eq. (11). This operation is reported next for completeness

$$T_i^j = \begin{cases} \tilde{T}_i^j & \text{if} \quad m^{j+1}\left(\tau_i^{j-1} + \tilde{T}_i^j\right) < C^{j+1} \\ \min\limits_h \left\{\tau_h^{j+1} | \tau_h^{j+1} > \tau_i^{j-1} + \tilde{T}_i^j\right\} - \tau_i^{j-1} & \text{if} \quad m^{j+1}\left(\tau_i^{j-1} + \tilde{T}_i^j\right) = C^{j+1} \end{cases}. \tag{29}$$

Actually, two different approaches can be adopted for dealing with the finite capacity constraint. The first methodology is the one explained so far, and consists of performing the operation in (29) at each step of the algorithm. Another approach, which is quite standard in the literature on scheduling, see for instance Luh *et al.* (1998) and Zhang *et al.* (2001), is to carry on the optimisation algorithm without taking into account this constraint, and then perform the operation in (29) on the final solution only. These two approaches are referred to as *each-step projection* and *final projection*, respectively. These issues will be discussed in more detail in *Part II* of this paper.

3.4. *The optimisation algorithm*

In this section, a formal description of the optimisation algorithm previously introduced is given, and its implementation issues are discussed in detail. A pseudo-code of the modified SPSA algorithm for the approximate solution of the optimisation problem (18) is given next.

```
Modified SPSA algorithm

1. Select initial point $\theta(0)$ and maximum number of steps K
2. $k = 0$
3. While $k \leqslant K$
    a. $k = k+1$
    b. Generate a sample $\nu^{(k)}$ according to the pdf $p_\nu(\nu)$ and
       compute
       $\theta_{NC}(k) = \bar{\theta}_{NC} + \nu^{(k)}$
    c. Generate a vector $\eta(k)$ according to a Bernoulli process
    d. Build $\hat{E}_\pm(k) = J(\theta_C(k) \pm c(k)\eta(k), \theta_{NC})$
    e. Construct the (possibly unfeasible) point
       $\tilde{\theta}_C(k+1) = \theta_C(k) - w(k)[\eta(k)^{-1}]\frac{\hat{E}_+(k) - \hat{E}_-(k)}{2c(k)}$
    f. Project the point to build a feasible solution
       $\theta_C(k+1) = \Pi_\theta\left[\tilde{\theta}_C(k+1)\right]$
4. end while
5. return (approximate) optimal value $\theta_C^K = \theta_C(K)$.
```

To be implemented, the algorithm needs the determination of some parameters, namely the gain sequences $w(k)$ and $c(k)$. In this regard, precise guidelines for their choice are given in Spall (1998). These guidelines were developed based on many test cases conducted by the author and others, and form a reasonable starting basis. In particular, the choice of the following sequences is proposed:

$$w(k) = \frac{w_0}{(W+k)^{\beta_1}}, \quad c(k) = \frac{c_0}{k^{\beta_2}}. \tag{30}$$

The asymptotically optimal values for the parameters w_0 and c_0 in (30) are 1.0 and 1/6, respectively. Practically effective and theoretically valid values for these parameters are also suggested as 0.602 and 0.101. Regarding the parameters W, β_1 and β_2, practical guidelines are given in Spall (1998).

Another point that is worth noting is that the innate stochastic nature of the algorithm and the presence of uncertainty do not guarantee that the solution $\theta_C(k)$ decreases at each step. Indeed, in the numerical implementation of the algorithm, this behaviour was frequently observed, and sometimes led to excessive increases of the cost function values. To tackle this problem, a possibility also suggested in Spall (1998) is to introduce at each step a limit to the worsening of the cost function. In detail, the candidate solution $\theta_C(k+1) = \Pi_\theta[\tilde{\theta}_C(k+1)]$ is not considered whenever it gives rise to a cost function value that does not satisfy the relation

$$\frac{J\left(\theta_C(k+1), \bar{\theta}_{NC} + \nu^{(k)}\right) - J\left(\theta_C(k), \bar{\theta}_{NC} + \nu^{(k)}\right)}{J\left(\theta_C(k), \bar{\theta}_{NC} + \nu^{(k)}\right)} \leqslant \psi, \tag{31}$$

where ψ is a threshold that should be selected as a trade-off between the number of not useful iterations, which should be as low as possible, and the admissible local solution worsening. The first constraint should lead to high values of ψ whereas the second one requires low values of this threshold. It should be noted that the introduction of (31) slows the execution of the algorithm slightly, since it requires an additional cost function evaluation at each step.

4. Conclusions and future research directions

In this paper a general framework for describing a fresh-food supply chain and an optimisation methodology to improve the performances of the network preserving the quality of the product are presented. The performance is improved by optimising the values of the physical and the logistic parameters describing the supply chain. Moreover, the presence of uncertainty in the behaviour of the chain is explicitly taken into account in the optimisation phase, leading to results that are robust with respect to parameter changes. It should also be noted that a subject of independent interest is to investigate the possibility of applying different global optimisation approaches for solving the optimisation problem (18). This is the subject of ongoing research by the authors of the present paper, in particular, concentrating on the application of evolutionary strategies, such as differential evolution (Storn & Price, 1997), and adaptive stochastic search methods, such as ICRS (Banga & Casares, 1987).

In the second part of this paper (Dabbene *et al.*, 2008) the proposed approach is extensively illustrated on a case study relevant to a beef meat refrigeration and distribution chain. The results of this case study confirm the effectiveness of the approach. In particular, the solutions provided by the algorithm are proved to be good and robust with respect to uncertainty.

Acknowledgements

The authors would like to thank the anonymous reviewers for their constructive comments that helped us to improve the quality of the paper.

REFERENCES

Altiparmak F; Gen M; Lin L; Paksoy T (2006). A genetic algorithm approach for multi-objective optimization of supply chain networks. Computer and Industrial Engineering, **51,** 197–216

Apaiah K A; Hendrix E M T (2005). Design of a supply chain network for pea-based novel protein foods. Journal of Food Engineering, **70,** 383–391

Apaiah R K; Hendrix E M T; Meerdink G; Linnermann R (2005). Qualitative methodology for efficient food chain design. Trends in Food Science and Technology, **16,** 204–214

Banga J R; Casares J J (1987). Integrated controlled random search: application to a wastewater treatment plant model. Institute of Chemical Engineering Symposium Series, **100,** 183–192

Banga J R; Balsa-Canto E; Moles C G; Alonso A A (2003). Improving food processing using modern optimization methods—a review. Trends in Food Science and Technology, **14**(4), 131–144

Barton P I; Banga J R; Galán S (2000). Optimization of hybrid discrete/continuous dynamic system. Computers and Chemical Engineering, **24,** 2171–2182

Beamon B M (1998). Supply chain design and analysis: models and methods. International Journal of Production Economics, **55,** 281–294

Broekmeulen R A C M (2001). Modelling the management of distribution centres. In: Food Process Modelling (Tijskens L M M; Hertog M L A T M; Nicolaï B M, eds), pp 432–447. Woodhead Publishing Limited, Cambridge

Cassandras C G; Lafortune S (1999). Introduction to Discrete Event Systems. Kluwer Academic Publishers, Boston, USA

Cassandras C G; Pepyne D L; Wardi Y (2001). Optimal control of a class of hybrid systems. IEEE Transactions on Automatic Control, **AC-46**(3), 398–415

Dabbene F; Gay P; Sacco N (2008). Optimisation of fresh-food supply chains in uncertain environments, Part II: A case study. Biosystems Engineering, **99,** 360–371

Datta A K (1998). Computer-aided engineering in food process and product design. Food Technology, **52,** 44–52

Delignette-Muller M L; Rosso L (2000). Biological variability and exposure assessment. International Journal of Food Microbiology, **58,** 203–212

Demir A D; Baucour P; Cronin K; Abodayeh K (2003). Analysis of temperature variability during the thermal processing of hazelnuts. Innovative Food Science and Emerging Technologies, **4,** 69–84

Floudas C A (2000). Deterministic Global Optimization: Theory, Methods, and Applications. Kluwer Academic, Dordrecht

Fryer P (1994). Mathematical models in food processing. Chemistry and Industry London, **13,** 515–518

Georgiadis P; Vlachos D; Iakovou E (2005). A system dynamics modelling framework for the strategic supply chain management of food supply chains. Journal of Food Engineering, **70,** 351–364

García M G; Balsa-Canto E; Alonso A A; Banga J R (2006). Computing optimal operating policies for the food industry. Journal of Food Engineering, **74,** 13–23

Gerencsér L; Hill SD; Vàgò Z (2001). Discrete optimization via SPSA. Proceedings of the American Control Conference, (pp. 1503–1504). Arlington, VA.

Gigiel A (1996). Optimizing the design of conventional carcass chillers. In: New Developments in Meat Refrigeration. Vol. 1, (pp. 1.16–1.24). ECCEAMST—International Institute of Refrigeration.

He Y; Fu M C; Marcus S I (2003). Convergence of simultaneous perturbation stochastic approximation for nondifferentiable optimization. IEEE Transactions on Automatic Control, **AC 48,** 1459–1463

Hertog M L A T M (2002). The impact of biological variation on postharvest population dynamics. Postharvest Biology and Technology, **26,** 253–263

Hertog M L A T M; Lammertyn J; Desmet M; Scheerlinck N; Nicolaï B M (2004). The impact of biological variation on postharvest behaviour of tomato fruit. Postharvest Biology and Technology, **34,** 271–284

Kushner H J; Yin G G (1997). Stochastic Approximation Algorithms and Applications. Springer, New York, USA

Luh P B; Gou L; Zhang Y; Nagahora T; Tsuji M; Yoneda K; Hasegawa T; Kyoya Y; Kano T (1998). Job shop scheduling with group dependent setups, finite buffers, and long time horizon. Annals of Operations and Research, **76,** 233–259

Minegishi S; Thiel D (2000). System dynamics modelling and simulation of a particular food supply chain. Simulation Practice and Theory, **8,** 321–339

Mo J; Qi L; Wei Z (2005). A manufacturing supply chain optimization model for distilling process. Applied Mathematics and Computation, **171,** 464–485

Nauta M J (2000). Separation of uncertainty and variability in quantitative microbial risk assessment. International Journal of Food Microbiology, **57,** 9–18

Nicolaï B M; Scheerlinck N; Verboven P; De Baerdemaeker J (2000). Stochastic perturbation analysis of thermal food processes with random field parameters. Transactions of the ASAE, **43**(1), 131–138

Nicolaï B M; Van Impe J F (1996). Predictive food microbiology: a probabilistic approach. Mathematics and Computers in Simulation, **42,** 287–292

Nicolaï B M; Verboven P; Scheerlinck N; De Baerdemaeker J (1998). Numerical analysis of the propagation of random parameter fluctuations in time and space during thermal food processes. Journal of Food Engineering, **38,** 259–278

Peirs A; Scheerlinck N; Berna Perez A; Jancsók P; Nicolaï B M (2002). Uncertainty analysis and modelling of the starch index during apple fruit maturation. Postharvest Biology and Technology, **26,** 199–207

Perea-López E; Ydstie B E; Grossmann I E (2003). A model predictive control strategy for supply chain optimization. Computers and Chemical Engineering, **27,** 1201–1218

Poschet F; Geerarerd A H; Scheerlinck N; Nicolaï B M; Van Impe J F (2003). Monte Carlo analysis as a tool to incorporate variation on experimental data in predictive microbiology. Food Microbiology, **20,** 285–295

Scheerlinck N; Verboven P; Stigter J D; De Baerdemaeker J; Van Impe J F; Nicolaï B M (2001). A variance propagation algorithm for stochastic heat and mass transfer problems in food processes. International Journal for Numerical Methods in Engineering, **51**(8), 961–983

Shapiro J F (2004). Challenges of strategic supply chain planning and modelling. Computers and Chemical Engineering, **28,** 855–861

Sloof M; Everest B V (2000). Problem decomposition. In: Food Processing Modelling (Tijskens L M M, et al., eds), pp 19–34. CRC Press, Boca Raton

Spall J C (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control, **AC 37,** 332–341

Spall J C (1998). Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Transactions on Aerospace and Electronic Systems, **34,** 817–823

Spall J C (2003). Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control. Wiley, New York, USA

Storn R; Price K (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, **11,** 341–359

Tempo R; Calafiore G; Dabbene F (2004). Randomized Algorithms for Analysis and Control of Uncertain Systems. Communications and Control Engineering Series. Springer, London, UK

Teixeira A A; Shoemaker C F (1988). Computerized Food Processing Operations. Van Nostrand Reinhold Company Inc, New York, USA

Tijskens L M M; Koster A C; Jonker J M E (2001). Concepts of chain management and chain optimization. In: Food Process Modelling (Tijskens L M M; Hertog M L A T M; Nicolaï B M, eds). Woodhead Publishing Limited, Cambridge

Van Impe J F (1996). Power and limitations of model based bioprocess optimisation. Mathematics and Computers in Simulation, **42,** 159–169

Van Impe J F; Bernaerts K; Geeraerd A H; Poschet F; Versyck K J (2001). Modelling and prediction in an uncertain environment. In: Food Process Modelling (Tijskens L M M; Hertog M L A; Nicolaï B M, eds), pp 156–179. Woodhead Publishing Limited, Cambridge

Vidyasagar M (2002). Learning and Generalization: With Applications to Neural Networks. Springer, New York, UK

Zhang Y; Luh P B; Narimatsu K; Moriya T; Shimada T; Fang L (2001). A macro-level scheduling method using Lagrangian relaxation. IEEE Transactions on Robotics and Automation, **17**(1), 70–79