Available online at www.sciencedirect.com
ScienceDirect
Computers and Chemical Engineering 31 (2007) 722–735
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

An agent-based approach for supply chain retrofitting under uncertainty

Fernando D. Mele, Gonzalo Guillén, Antonio Espuña, Luis Puigjaner *

Universitat Politècnica de Catalunya, Chemical Engineering Department, Diagonal 647, E-08028 Barcelona, Spain

Received 23 November 2005; received in revised form 18 December 2006; accepted 19 December 2006
Available online 26 January 2007

Abstract

In this work, decisions that have a long lasting effect on the supply chain (SC) such as the design and retrofit of a production/distribution network are considered. The retrofitting tasks are accomplished by using a SC agent-oriented simulation system, which model each entity belonging to the SC as an independent agent. The starting point is a set of possible design options for the existing SC. For each design alternative a performance index is obtained through the agent-based framework by looking for the best value of the operational variables associated to the resulting network. The proposed methodology allows to address the design of complex SCs which are hard to be modelled otherwise, for example by means of standard mathematical programming tools. Specifically, the multi-agent system is suitable for SCs that are either driven by pull strategies or operate under uncertain environments, in which the mathematical programming approaches are likely to be inferior due to the high computational effort required. The advantages of our approach are highlighted through a case study comprising several plants, warehouses and retailers.

Keywords: Supply chain management; Uncertainty; Multi-agent system

1. Introduction

The concept of the supply chain (SC), which first appeared in the early 1990s, has recently been the focus of much interest, as the possibility of providing an integrated management of a SC can reduce the propagation of unexpected/undesirable events throughout the network and can markedly improve the profitability of all the parties involved.

A SC can be defined as a network of business entities (i.e., suppliers, manufacturers, distributors, and retailers) who work together in an effort to acquire raw materials, transform these raw materials into intermediate and finished products, and distribute the final products to retailers.

Different issues have forced businesses to invest in, and focus attention on their SCs: the fierce competition in today's global markets, the variability in demand, the introduction of products with short life cycles, the rapid development of new products, the wide variety of supply alternatives for similar products/services, the heightened expectations of customers, and so on. These issues, together with continuing advances in communication technologies, has caused SCs and the techniques used to manage them to grow continuously more complex. At present, SCs are highly dynamic environments (Applequist, Pekny, & Reklaitis, 2000).

Supply chain management (SCM) aims to integrate plants with their suppliers and customers so that they can be managed as a single entity and to coordinate all input/output flows (of materials, information, and funds) so that products are produced and distributed in the right quantities, to the right locations, and at the right time (Simchi-Levi, Kamisky, & Simchi-Levi, 2000). Therefore, SCM implies the handling of flows throughout the entire SC, from suppliers to customers while encompassing warehouses and distribution centres (DCs), and usually including after-sales services, returns, and recycling (Silver, Pyke, & Peterson, 1998). The main objective is to achieve acceptable financial returns together with the desired consumer satisfaction levels.

The SCM problem may be considered at different levels depending on the strategic, tactical and operational variables involved in decision-making (Fox, Barbuceanu, & Teigen, 2000). Therefore, a large spectrum of a firm's strategic, tactical and operational activities are encompassed by SCM:

- The strategic level concerns those decisions that will have a long-lasting effect on the firm. It is focused on SC design, and entails determining the optimal configuration for an

* Corresponding author.
E-mail address: luis.puigjaner@upc.es (L. Puigjaner).

0098-1354/$ – see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2006.12.013

entire SC network, including the design of the embedded plants.

- The tactical level encompasses long-/medium-term management decisions, which are typically updated at a rate ranging between once every quarter and once every year. These include overall purchasing and production decisions, inventory policies, and transport strategies.
- The operational level refers to day-to-day decisions such as scheduling, lead-time quotations, routing, and lorry loading.

The complexity of a SC is increased by the high degree of uncertainty brought about by external factors, such as continuously changing market conditions and customer expectations, and internal parameters, such as product yields, qualities and processing times. Although the deterministic analysis requires less computational effort, it is generally worthy to solve the problem stochastically since it provides a valuable insight of the system. With this information managers are able to know the complete spectrum of results they could get, from the worst to the best case, according to the data uncertainty. Instead, the deterministic solution is valid only for the scenario used in the calculations without any information about the risk the decision-maker is taken.

Since the work by Mele, Guillén, Espuña, and Puigjaner (2006) already presented a general multi-agent framework addressing tactical and operational decisions regarding the management of chemical SCs operating under uncertainty, this work aims to extend the capabilities of the multi-agent framework previously presented to contemplate the design and retrofit of chemical SCs. Thus, this paper takes full advantage of the ideas and concepts presented in the previous work to devise a quantitative tool for SC design and retrofit. The use of the multi-agent system as a way to evaluate the different design alternatives allows for properly assessing the impact that the design decisions have in the operation of the network. Furthermore, the application of such novel strategy overcomes the difficulties associated with the application of fairly simple capacity representations that usually either overestimate or underestimate the real SC capacity.

Section 2 of this paper presents a critical review of previous work on SC modelling and design. Section 3 introduces a formal definition of the problem of interest through a motivation example. In Section 4, the strategy introduced to address this problem is described in detail. A numerical example is presented in Section 5 to illustrate the applicability of the proposed framework. Finally, the conclusions of the work are given in Section 6.

2. Literature review

2.1. SCM under uncertainty

A lot of attempts have been made to model and optimise the SC behavior, currently existing a big amount of deterministic (Bok, Grossmann, & Park, 2000; Gjerdrum, Shah, & Papageorgiou, 2000; Timpe & Kallrath, 2000) and stochastic derived approaches. Since the nature of most SCs is characterised by numerous sources of technical and commercial uncertainty, the consideration of all model parameters, such as cost coefficients, production rates, demand, and so forth, as being known is not realistic. Several works deal with uncertainty in SCM at different levels. One part of the effort has been oriented through control theory in which the uncertainty is modelled as disturbances arriving to a dynamic model of the system. The work by Bose and Pekny (2000) looks for the inventory set points that ensure a desired customer service level with a planning tool, and then track them with an model predictive control (MPC) approach. Similarly, Perea-López, Ydstie, and Grossmann (2003) determine the optimal variables that maximise the profit of the system, by optimising a multiperiod mixed-integer linear programming (MILP) problem, and using a rolling horizon MPC approach so as to include the disturbance influence. All these approaches work at an operational level.

Other approaches are able to cope with the uncertainty through fuzzy programming (Sakawa, Nishizaki, & Uemura, 2001) at strategic level. Their limitations are related to the simplicity of the production/distribution models usually used.

A third group and the biggest one includes statistical analysis-based methods in which it is assumed that the uncertain variable follows a particular probability distribution. As in this article, most works apply an *adaptive* strategy in which the SC controls the risk exposure of its assets by constantly adapting its operations to unfolding demand realisations. In the strategy known as *shaper*, the SC aims to restructure the demand distribution contracting agreements with the customer (Anupindi & Bassok, 1999). On the other hand, the most popular model used is the two-stage decision approach. Approaches differ primarily in the selection of the decision variables and the way in which the expected value term, which involves a multidimensional integral accounting for the probability distribution of the uncertain parameters, is computed. The difficulty of continuous distributions is avoided by introducing discrete scenarios, or combinations of discrete samples of all uncertain parameters (Cohen & Lee, 1989; Iyer & Grossmann, 1998; Jung, Blau, Pekny, Reklaitis, & Eversdyk, 2004; Subrahmanyam, 1996; Tsiakis, Shah, & Pantelides, 2001). Pistikopoulos and co-workers (Acevedo & Pistikopoulos, 1998; Bernardo, Pistikopoulos, & Saraiva, 1999) have examined alternative strategies for evaluating the integral term, ranging from cubature methods to sampling methods. Maranas and collaborators (Gupta & Maranas, 2000, 2003; Petkov & Maranas, 1997) convert stochastic features of the problem into a chance-constrained programming problem. Finally, a different approach at strategic level is the work of Applequist et al. (2000), who present a method for evaluating SC projects with the capability to assess the integral values based upon polytope volumes. Literature reveals that the most extensively studied source of uncertainty has been demand (Ahmed & Sahinidis, 1998; Gupta & Maranas, 2000, 2003; Ierapetritou & Pistikopoulos, 1996; Petkov & Maranas, 1997). The emphasis on incorporating demand uncertainty into the planning decisions obeys to the fact that this is one of the most important factors that could affect the capacity of a company to meet the customer demand. However, there are many other factors that could seriously affect this capacity. Hence, in the present work not only demand

uncertainty is considered but also transport and processing times.

2.2. Agent-based modelling

Software agent-based systems have been found to be effective tools for solving certain problems of a complex nature. Since they are built by combining autonomous computer programs or agents in a distributed architecture, they typically perform significantly better than the individual programs do in isolation. The main reason for this improvement lies in the co-operation and distribution of the tasks between the individual agents that constitute the system. The need for coordinating multiple operations is a key feature of many chemical engineering problems, to which multi-agent systems can offer well-suited answers. For instance, Han, Douglas, & Stephanopoulos (1995) decompose a design process into tasks and an agent is assigned to perform each one. Maguire, Scott, Paterson, and Struthers (1995) also propose an agent-based environment for process design. Siirola, Hauan, and Westerberg (2003, 2004) propose a modular framework to implement agent-based systems for engineering design. Through a variety of different algorithmic agents, the key ideas are shown by using a multi-objective optimisation problem as an example.

Within the area of SCM, the earliest attempts to use dynamic simulation was reported by Forrester (1961), who strove to perform a dynamic simulation of industrial systems by means of discrete time mass balances and non-linear delays. However, due to the complexity of the models and the computer limitations at that moment, the work only covers small academic examples.

In the last few years, multi-agent systems have become a promising tool for solving SCM problems. In these approaches, agents are used to emulate the behavior of each of the entities embedded in the SC. The overall multi-agent system, which acts as a discrete-event simulator of the SC, can be also combined with an statistical analysis-based method in order to deal with uncertainty. These statistical methods assume that the uncertain variables follow a set of particular probability distributions and repetitively sample from them to generate a set of possible realisations or scenarios. The deterministic discrete-event simulator is then run for each of these scenarios determined by a combination of values of the uncertain parameters, and provides for each of them a set of output variables. The probability distribution of the performance measure is finally constructed from these values of the output variables. This probability distribution, and not a single point value, is finally used to assess different SC configurations.

Some researchers have studied agents by focusing on co-operation between SC systems. Sauter, Parunak, and Goic (1999) present an architecture that is inspired by insect colonies and human settlements. Fox et al. (2000)[1] simulate the three levels of decision-making: strategic (production allocation and sourcing strategy), tactical (forecasting and planning), and operational (inventory deployment and detailed scheduling). Swaminathan, Smith, and Zadeh (1998)[2] present a modelling and simulation framework for developing customised decision support tools for SC reengineering. A similar more recent contribution in this direction is the work carried out by Julka, Srinivasan, and Karimi (2002). They develop a framework that has two basic elements: object modelling of SC flows and agent modelling of SC entities. The user can configure SC scenarios and compare the effect of different decisions using various performance measures. However, apart from some preliminary works such as those ones by Mele, Guillén, Urbano, Espuña, and Puigjaner (2004) and Mele et al. (2006), these approaches do not include resources for global improvement or optimisation.

2.3. Simulation-based optimisation

Simulation-based optimisation comes as an attractive combined strategy to address optimisation under uncertainty. It deals with the situation in which the analyst would like to find which of possibly many sets of input parameters lead to optimal performance of the represented system. Thus, a simulation model can be understood as a function whose explicit form is unknown that turns input parameters into performance measures (Law & Kelton, 2000). Simulation-based optimisation is an active area in the field of the stochastic optimisation. Reviews of the current research in this area can be found in Fu (2002). In the process system engineering literature, the simulation-based optimisation approaches for SCM have received few attention and are currently waiting for further study. However, the works from Subramanian, Pekny, and Reklaitis (2000) and Jung et al. (2004) are appreciated contributions in this field, in which the authors apply a simulation–optimisation framework while looking for managing uncertainty.

There are a variety of ways for optimising processes represented by means of a simulation model. Many of them have not been used to develop optimisation for simulation software because they often need a considerably amount of technical sophistication on the part of the user, as well as a substantial amount of computation time (Andradóttir, 1998). This is closely related to the fact that some of these techniques are local search strategies and may be strongly problem dependent. On the contrary, the success of the metaheuristics is perhaps their design to seek global optimality and their apparent robust properties in practice even if not completely supported theoretically yet.

The main advantage of evolutionary approaches over other metaheuristics approaches is that they are capable of exploring a larger area of the solution space with a smaller number of objective function evaluations. Since in the context of simulation-based optimisation evaluating the objective function entails running the simulation model, being able to find high quality solutions early in the search is of critical importance. Evolutionary algorithms, among them genetic algorithms (GA), have been used to optimise multi-modal, discontinuous, and non-

[1] The Integrated Supply Chain Management Project, May 30, 2005, www.eil.uto-ronto.ca/iscm-descr.html.

[2] Supply-Chain Modeling and Analysis, May 30, 2005, www-2.cs.cmu.edu/afs/cs/-project/ozone/www/supply-chain/supply-chain.html.

differentiable functions. In the field of SCM, the approaches are supported by the research in GA combined with mathematical programming (Syarif, Yun, & Gen, 2002; Vergara, Khouja, & Michalewicz, 2002; Zhou, Min, & Gen, 2002). However, these works mostly deal with strategic decisions, for instance, combinatorial operation research problems such as the multi-stage facility location/allocation problem in which decisions are the facilities to be opened or the distribution network to be used, rather than tactical or operational ones, and even though progresses made in this area are huge, usually the problem is different from the one undertaken here, that is choosing a SC configuration by taking simultaneously into consideration different variables belonging to the tactical and operational levels.

3. Motivating example

Let us consider a SC consisting of nine interconnected entities as the one shown in Fig. 1. The network comprises two plants (F1 and F2), two DCs (D1 and D2) and five retailers (R1–R5). The plants manufacture final products A–C from raw materials provided by external suppliers. Final products are transported from plants to warehouses and from warehouses to retailers where they become available to customers. In this SC, there is a material flow that moves from the plants to the customers and an information flow (ordering flow) which does it in the opposite direction. The operational policy used is make-to-order from the point of view of the final retailers and make-to-stock in the other entities.

It is assumed that the demand of the system can be modelled as a set of events distributed over the time horizon θ under study. Each of these events occurs at a certain time and involve a certain amount of materials demanded. Both parameters, the amounts and the inter-arrival intervals between events, are assumed to be time-variant. Moreover, processing and transport times are also assumed to follow normal probability distributions. From now on, one realisation of the uncertainty along the time horizon is identified with ω.

The periodic revision strategy known as (R, s, and S) policy is implemented at the DCs, being R the time period between two consecutive inventory reviews, s the reorder point and S the stock level. Thus, the strategy has three parameters whose values have to be determined: R, S, and s. For the retailers, a similar but continuous revision strategy has been applied. It is assumed that the company must assign the parameters to the inventory control laws at the beginning of the time horizon. These parameters together with the capacities of each SC entity constitute the decision variables of the problem; a set of values of these variables is a decision strategy that will be represented by η. With regard to the plants, these are assumed to be multi-purpose batch chemical plants that operate following a set of predefined heuristic rules in order to compute the scheduling decisions that enable the fulfillment of the orders derived from the inventory control policies.

The manager's motivation in this problem is to choose a SC configuration and an inventory control strategy such that the outcomes of the system are as profitable as possible, that is to say optimal. Therefore, it is necessary to define a set of performance measures, or optimality criteria, in order to compare alternative decisions. In this work, the optimality criteria considered is the expected total profit, which is computed as the difference between the revenues and the total cost. The total cost includes the inventory, the manufacturing and the transport costs. Therefore, the expected total profit under a given decision strategy η (SC configuration and inventory control parameters) is

$$\mathcal{F}(\eta) = E[\text{Profit}(\eta, \omega)] \tag{1}$$

where the expectation $E[\cdot]$ is taken with respect to all realisations of the uncertainty ω belonging to a probability space Ω.

As it can be observed in the real-world, the resulting production/distribution network is a complex system which involves the use of different kinds of heuristic rules at the nodes of the network (inventory control policies). Although, it would be pos-

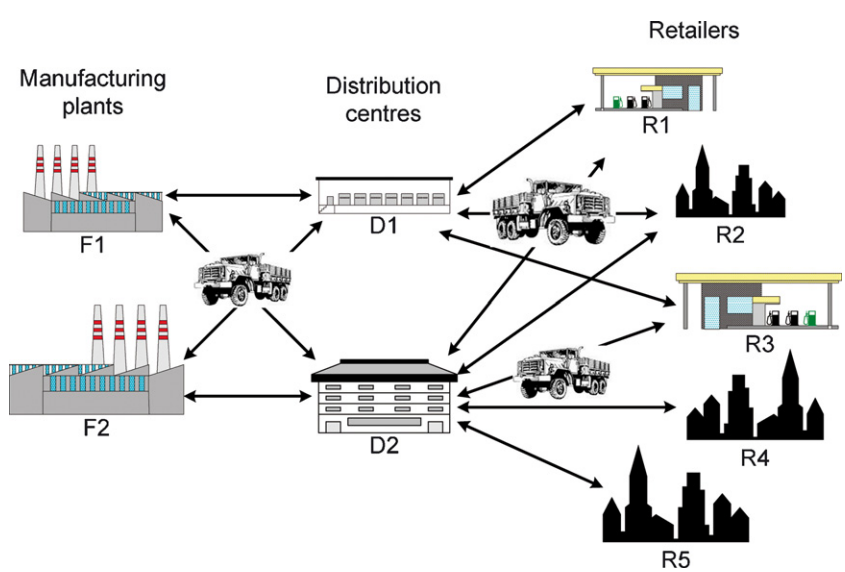

Fig. 1. Structure of the case study.

sible to address the modeling and optimisation of this system by means of standard mathematical programming tools, this would be a highly complex and time-consuming task for two main reasons. In first place, because of the difficulty associated with deriving the explicit form of the analytical function that characterise the system and provides the values of the output state variables given a set of input parameters (Julka et al., 2002). In second place, because of the properties of the resulting mathematical model, if this could be explicitly formulated. Such model would be a non-convex large-scale mixed-integer non-linear programming problem (MINLP) difficult to be solved in reasonable CPU time by the standard mixed-integer programming solvers available nowadays (Jung et al., 2004).

Specifically, the problem previously presented could be formalised as a multi-stage stochastic programming model. For the case presented in this work, the construction of a deterministic model equivalent to the multi-stage stochastic problem using the scenario approximation will create a mathematical formulation much beyond the power of current non-linear (NLP) or MINLP solution techniques owing to its scale and complexity.

4. Proposed approach

To overcome the numerical difficulties associated to the resolution of this deterministic equivalent of the multi-stage stochastic formulation by means of standard mathematical programming tools, the overall SC design problem is hierarchically decomposed into two levels, a higher strategic level and a lower tactical/operational one. Although the optimality is virtually sacrificed, this leads to tractable problems and avoids monolithic formulations that require an extensive computational effort and thus become impossible to solve in the case of large-scale SC problems. At the strategic level, the capacities of the plants and storage sites are considered. At the lower level, tactical (values for the inventory policy parameters) and operational decisions (quantities manufactured in the factories and transported between nodes) are computed for a given configuration. The computation of the tactical variables is carried out by a GA which interacts with the SC agent-oriented simulation system to obtain the fitness of the individuals of the populations generated in each iteration. The simulation system models each entity belonging to the SC as an independent agent and represents the interactions between the components of the SC in a functional way.

The contribution of this work compared to the previous ones is in the hybrid strategy proposed for optimisation and also in the utilisation of an agent-based simulation model to represent the operation of the decentralised SC.

The overall algorithm is depicted in Fig. 2. In first place, a set of design candidates from which the final configuration should be selected is provided. A Monte Carlo sampling is next performed over the probability distributions that represent the uncertain parameters, thus generating a set of instances with given probability of occurrence. For each SC alternative, a GA providing an oriented search mechanism that decreases the computational effort required by rigorous optimisation techniques is applied for optimising the variables associated to the operation of the given SC configuration under the uncertain environment.

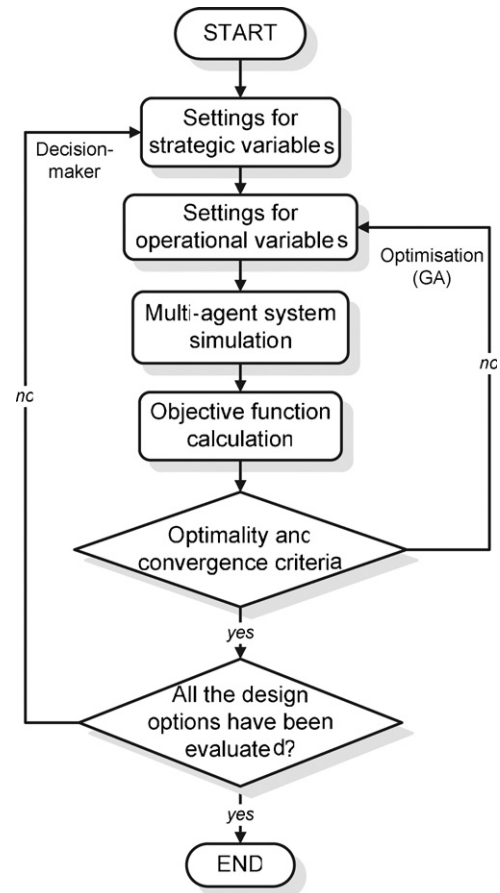

Fig. 2. Decision-making procedure.

Therefore, the fitness function of each chromosome is obtained by running a simulation for each instance or realisation of a set of instances generated by Monte Carlo sampling, that is for a set of values of the uncertain parameters, a set of discrete-event simulations is run and the average value of the objective functions, in this case profits, is computed. If the number of SC configurations to be explored is very high, an additional outer-loop and thus an additional GA can be applied in connection with the procedure explained to evaluate the designs under the uncertain environment.

4.1. Objective function

The different configurations are finally compared in terms of the expected total profit associated to the best set of their operative variables computed by the GA. The profit value computed for each configuration involves the storage, manufacturing, and transport SC costs and the revenues derived from sales over the simulation time horizon θ. The profit model is very simple due to the focus of the article on the solving algorithm.

The revenues (rev) are computed from the sales of products:

$$\text{rev} = \sum_k \sum_t U_{kt}\, S_{kt} \tag{2}$$

where t represents each simulation step; S_{kt} the unit price at retailer node k and time t; U_{kt} is the amount sold at retailer entity

k and time t. Eqs. (3)–(5) represent the storage IC, manufacturing MC and transport TC costs, respectively:

$$\mathrm{IC} = \sum_k \sum_t I_{kt}\,\mathrm{IC}_{kt} \tag{3}$$

$$\mathrm{MC} = \sum_k \sum_t M_{kt}\,\mathrm{MC}_{kt} \tag{4}$$

$$\mathrm{TC} = \sum_k \sum_t T_{kt}\,\mathrm{TC}_{kt} \tag{5}$$

IC_{kt}, MC_{kt}, and TC_{kt} are the unit costs related to holding inventory, manufacturing and transport, respectively, and I_{kt}, M_{kt}, and T_{kt} are the amounts of materials stored, manufactured and transported during the SC operation. Finally, the total profit (profit) over the time horizon is given by

$$\text{profit} = \text{rev} - (\mathrm{IC} + \mathrm{MC} - \mathrm{TC}) \tag{6}$$

The simulation–optimisation strategy applied to assess the different design alternatives under uncertainty, which is the core of this work, is described in more detail in the next section.

Next, the agent-based SC simulator and the two key sub-problems, the tactical level optimisation sub-problem and the simulation sub-problem, are defined, discussing in more detail the various computational details which are needed to link these sub-problems and to drive the computations.

4.2. The agent-based SC simulator

A software agent-based system has been developed so as to implement a SC model. In this paper, it is proposed the use of independent and well-defined agents to model all the entities that belong to the SC. Each agent is represented by a collection of states and a set of rules. The agents were designed following the paradigm proposed by Wooldridge and Jennings (1995): *An agent is an encapsulated computer system that is situated in some environment, and that is capable of flexible, autonomous action in that environment in order to meet its design objectives*.

The agent-based simulator consists of two categories of objects: the *software agents* and the *messages*. All agents communicate with each other by sending objects of message class. Messages are objects that are exchanged between the agents and they represent material, information or cash flows. The behavior of an agent is described in the form of one or more *tasks* classes. An agent may perform multiple tasks with multiple active threads in a task. Among the agents that take part in this system, those agents that represent the physical entities, facilities or nodes in the real-world SC such as external suppliers or customers, manufacturing plants, distribution centers, and so forth, are called *emulation* agents. They include a number of sub-agents and are described in Appendix A. The agents representing external customers or suppliers emulate the behavior of real or hypothetical entities in order to consider them in the simulation runs. Hence, these agents have specific models mimicking this kind of entities. Moreover, the system may have another kind of agents apart from emulation agents. Among them, the most important is the central agent, which do not have correspondence with a physical entity in the chain.

In Fig. 3 it can be seen how the real system is modelled using agents (one for each site plus a central coordination agent), being each of them modelled, in turn, by a set of sub-agents representing the internal departments of these entities. In this figure, the internal structure of an agent factory is shown.

An important point from the implementation perspective is the possibility of defining a *generic* agent or class for every SC site and then generate from this class some children classes, by inheritance, with the specific features for representing factories, distribution centers or whatsoever entity of interest.

The strategies that rule the interaction and the internal behavior of the different agents in the system, making this approach realistic and allowing to obtain meaningful results, are mainly implemented in the demand, inventory and transport models.

4.2.1. Demand model

In order to perform simulations, it is necessary to establish the way in which demand is expressed. In this approach, the demand has been modelled as a set of events distributed over the time horizon of the study, each of these events having an associated amount of material and time of occurrence. Both the amounts and the inter-arrival intervals between events are variable by default. The system can be set to use either historical/forecasted demand data, or values sampled from probability distributions: uniform or normal for the amounts and Poisson for modelling the inter-arrival times between orders (Law & Kelton, 2000). The customer demands are fully backlogged by retailers.

4.2.2. Transport policies

The transport policy accounts for the way in which a repository of "material ready for delivery" is depleted. A number of policies can be used, being their parameters a likely source of uncertainty, for example delivery time, material handling time, and so forth.

4.2.3. Inventory policies

The inventory policy aims to give rules in order to answer the questions: "when should a replenishment order be placed?" and "how large should the replenishment order be?" The system allows to implement a number of inventory control systems (Order-Point (s, k) System, Order-Point Order-Quantity (s, Q) System, Periodic-Review Order-Up-To-Level (R, S) System, etc.) that can be classified as continuous revision strategies or periodic revision ones. In a continuous review, each transaction (shipment, receipt, etc.) triggers an analysis of the inventory status in order to update it. With periodic review, the stock status is determined only every R time units. Obviously, it is possible to devise and implement other strategies more or less complex, including forecasting information and cost optimisation models.

Finally, the agent-based system can eventually use a number of modular tools in order to perform specific tasks or to get the values of specific indexes. There are six different modules: a forecasting module that provides values to simulate the customer demand; a negotiation module that supports the signing up

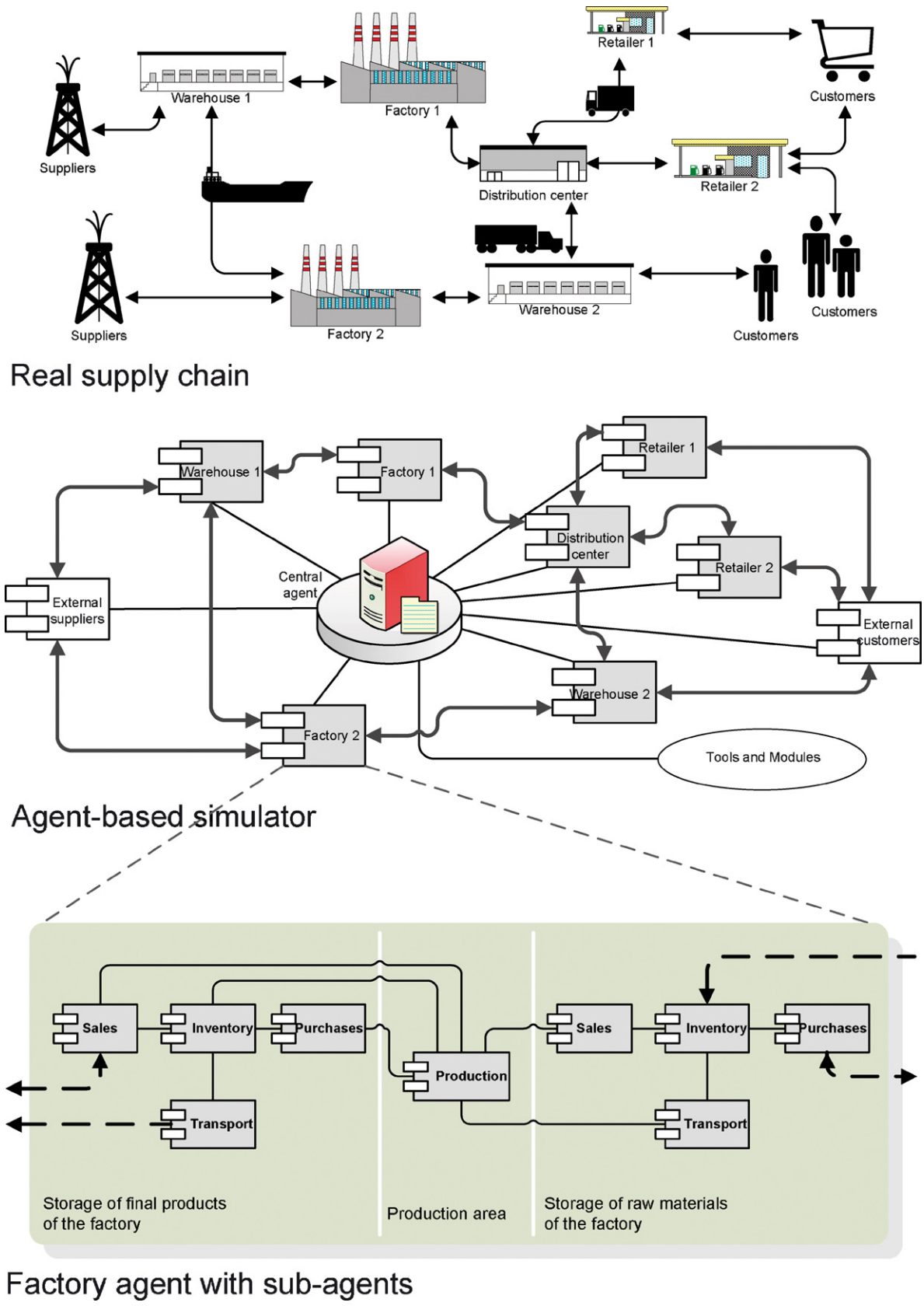

Fig. 3. The agent-based simulation model.

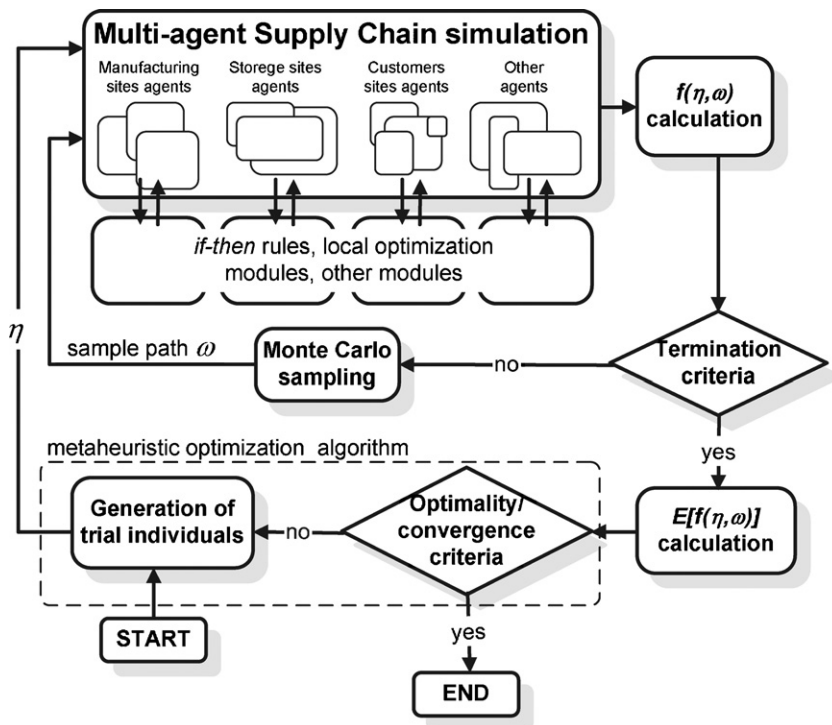

Fig. 4. Proposed framework.

of long-lasting contracts with external suppliers and customers; three modules that provide the value of SC performance indicators, namely: economical indicators (profit, costs), financial indicators and environmental impact indicators; and an optimisation module based on metaheuristic techniques that takes into consideration the values of the performance indicators in order to improve the SC operation.

4.3. Tactical level optimisation sub-problem

The simulation–optimisation strategy comprises two loops. Fig. 4 presents a flow diagram which summarises the overall computational strategy that combines the *simulation* and *tactical level* loop components. The optimiser (GA) uses the parameters of the inventory control laws as decision variables, whereas the inner problem involves a SC simulation. Basically, the set of Monte Carlo driven repetitions of the simulations constitutes a function evaluation for the tactical level optimisation loop. Both loops are next described in more detail.

The objective function of the tactical level optimisation involves a stochastic optimisation in which the profit is maximised by choosing the values of the decision variables, η, associated with the inventory control policies at the facilities. The problem consists of finding the value of η (say η_{opt}) that maximises the expected profit, as expressed in the following equation:

$$\max_{\eta} \mathcal{F}(\eta) = E[\text{Profit}(\eta, \omega)],$$
$$\text{subject to :} \quad \text{strategic and operative constraints} \tag{7}$$

Eq. (7) is evaluated in the simulation loop by using the results of multiple Monte Carlo samplings, which in turn, involves the application of inventory control policies and if-then rules.

The tactical level optimisation algorithm consists of a GA-based strategy, which is applied to improve the values of the inventory control parameters. The use of a GA strategy is motivated by the non-convex nature of the objective function in the solution space and also by the unavailability of explicit equations to describe the SC operation. The scheme of the tactical level optimisation loop is as follows:

- **Step I**. Create an initial population of K randomly generated individuals. Each individual k ($k = 1, 2, \ldots, K$) represents a set of values for the inventory control parameters. Each gene of the chromosome represents the value of an inventory control parameter. Therefore, the number of genes of each chromosome, that is of each individual of the population, is given by the number of products and facilities of the SC. η^k_{gen} denotes the individual k that belongs to the generation gen of the GA, where gen is the counter for the iterations of the tactical level loop.
- **Step II**. Run a *sufficient* number, n, of Monte Carlo samplings to generate different sample paths ω. Each sample path comprises a set of values of the uncertain parameters, which are in this case the demand, the transport times and the processing times.
- **Step III**. Execute the simulator with its embedded rules to obtain a reliable estimate of the expected profit for each individual k, $\mathcal{F}(\eta^k_{\text{gen}})$.

- **Step IV**. Apply genetic operators (selection, crossover and mutation) over the current population of η^k_{gen} values to generate a new one. Until this moment, the best solution found is $\eta^*_{\text{gen}} : \mathcal{F}(\eta^*_{\text{gen}}) = \max\{\mathcal{F}(\eta^k_{\text{gen}}); k = 1, \ldots, N\}$.
- **Step V**. If gen ≥ MaxGen, being MaxGen the maximum number of generations, then stop. Otherwise, set gen = gen + 1 and go to step **II**.

4.4. Simulation sub-problem

The computations inside the tactical level loop optimisation involve the repeated simulation of the SC operation over the planning horizon taking into account the Monte Carlo samples of the uncertain parameters. Within each such simulation, a series of decisions are periodically made in order to replenish inventory in the different nodes and satisfy customer demand. The simulator has the important advantage of being capable of representing the real-world SC operation regarding the replenishment control policies used by the decision-makers at each facility as well as the production and transport rules applied in each node in order to fulfill the orders. Each complete simulation run travels across a so-called timeline, that is a unique future materialisation of uncertainty. Therefore, the procedure for executing a timeline is as follows:

- **Step I**. Run the discrete-event simulation for the planning period θ with demand, transport and processing times outcomes ω. The simulator emulates in each simulation the overall decision-making process regarding the SCM: production scheduling at the manufacturing plants, transport of materials between nodes and inventory replenishment in all the facilities of the network.
- **Step II**. Calculate and record objective function value, Profit(η, ω).

By repeating the above procedure for a sufficient number, n, of Monte Carlo samples, the performance results necessary to compute the expected value, $\mathcal{F}(\eta) = E[\text{Profit}(\eta, \omega)]$, can be collected. Note that the accuracy of the expected value will be governed by the number of analysed timelines and their representativeness.

As it can be observed, the proposed strategy constitutes a decomposition approach for the underlying stochastic multi-stage programming, as it splits the terms of the objective function and the constraints between the two sub-problems. Thus, the solutions provided by the simulation-based approach will be in general sub-optimal. However, the proposed strategy is capable of providing 'good' solutions for problems which are impossible to tackle by means of standard mathematical programming tools. Moreover, the use of building blocks to model the structure of the SC under study makes it easier to implement the proposed approach in a real scenario.

A key point in the proposed methodology is determining the number of samples or replications, n, to take during the Monte Carlo simulation so as to ensure that $E[\text{Profit}(\eta, \omega)]$ is a good enough estimation for the mean of the universe of uncertain variables. As n gets larger, the representativeness of $E[\text{Profit}(\eta, \omega)]$ increases as guaranteed by the central limit theorem, but the exact relationship has to be studied because it depends on the simulation model under consideration.

In this analysis, the aim can be stated as obtaining an estimate of the mean with a relative error γ and a confidence level of $100(1-\alpha)\%$. Then the sequential procedure is as follows:

- **Step I**. Make $n_0 \geq 2$ replications of the simulation and set $n = n_0$.
- **Step II**. Let ω_l be a random variable defined on the lth replication for $l = 1, 2, \ldots, n$. Compute the arithmetic average $\bar{\omega}(n)$ and the confidence-interval half-length $\delta(n, \alpha)$ from $\omega_1, \omega_2, \ldots, \omega_n$.

$$\delta(n, \alpha) = t_{n-1,(1-\alpha)/2}\sqrt{\frac{S^2(n)}{n}} \tag{8}$$

  $S^2(n)$ is the sample variance of ω_l and $t_{n-1,(1-\alpha)/2}$ is the critical point for the t distribution.
- **Step III**. If $\delta(n, \alpha)/|\bar{\omega}(n)| \leq \gamma/(1-\gamma)$, use $\bar{\omega}(n)$ as the point estimate for the mean and stop. Otherwise, replace n by $n+1$, make an additional replication of the simulation and go to Step **II**.

A more detailed explanation about this procedure can be found in Law and Kelton (2000).

4.5. Implementation of the proposed strategy

The computational framework for implementing the proposed strategy consists of the following elements: the computation master module, the agent-based simulator with inventory control systems and scheduling modules, and the objective function calculation module.

The master module provides centralised computation management and tracking of the sequence of actions to carry out the whole algorithm. Its various functional roles include: issuing commands for retrieving data to apply the strategy, triggering events for executing the simulation model, sampling of the uncertainty realisations, and saving the calculation results. This module loads the initial data (inventory levels, values for the inventory control parameters, state of the random number generator), and the settings related to the computation of the proposed strategy (e.g., number of iterations for the optimisation and the Monte Carlo loops, termination criteria).

The discrete-event simulation model is an agent-based simulator that provides specific methods to execute processes simultaneously, to mimic the activities of entities operating in parallel. The complexity of the implementation of this type of discrete-event simulation model increases by the fact that the model must incorporate not only the representation of the active entities but also of various events and logic associated with these events.

There is a database managed by the multi-agent simulator that stores data related to the SC structure (number of product types;

number of processing units in each production facility; production, inventory and transport capacities), SC simulation (length of simulation horizon, length of the horizon and of each period in the scheduling models, step size factor, convergence tolerance), and modelling features of the uncertain information (mean and variance of the demand of each product type, parameters for demand patterns of each product).

The objective function calculation module encloses data to calculate the simulation performance (objective function) from the simulation outputs. It contains data regarding inventory costs, backlogging costs, transport costs, product prices, and so forth.

The master module is a function programmed in *MATLAB* (The Math Works, 2004) that loads the necessary by executing an m-file in *MATLAB*. The discrete-event simulation model is an agent-based simulator constructed with two *MATLAB* toolboxes: *Stateflow* and *Simulink*. *Stateflow* allows representing the system by means of state-transitions diagrams and *Simulink* offers the simulation environment. The objective function calculation module is also an m-file in *MATLAB*.

5. Case study

For the case study presented in this paper, the number of samples or replications, n, to take during the Monte Carlo simulation has been determined using $\gamma = 0.05$ and $\alpha = 0.1$. After 30 simulations, the quotient $\delta(n, \alpha)/|\bar{\omega}(n)|$ falls below $\gamma/(1-\gamma) = 0.0526$, $\gamma = 5\%$. It allows to conclude that, in this case, a relatively small number of simulations is sufficient to obtain statistically significant performance estimates.

Once determined a value for n to accomplish the confidence and error estimation requirements, the motivating example presented before has been solved with the proposed strategy, after tuning the GAs.

Tuning the GA-based strategy is a key task before executing it. This task has required doing sensitivity analysis using a big

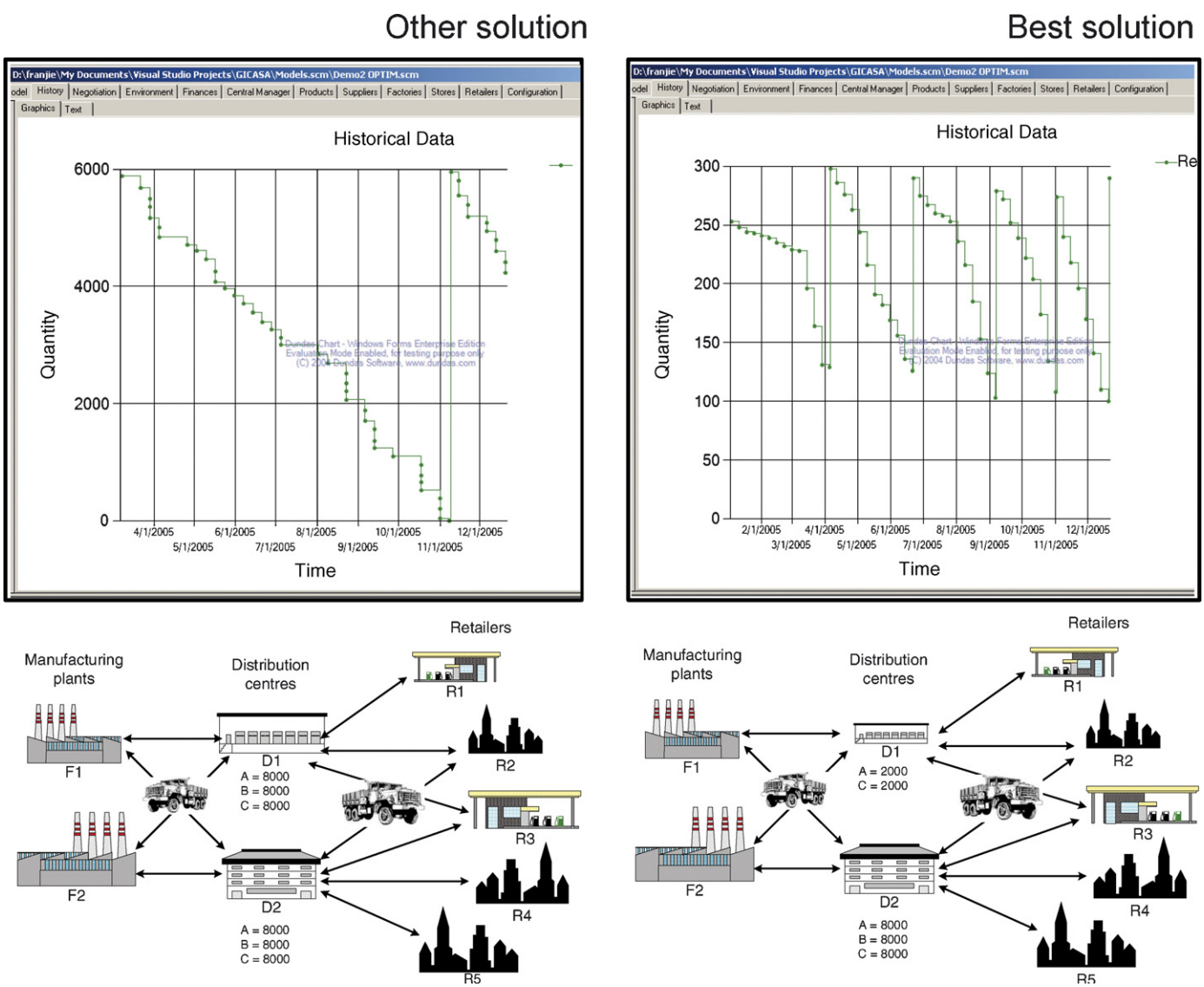

Fig. 5. Inventory level evolution at D2.

Table 1
Capacities of the design alternatives (units)

	Distribution center D1			Distribution center D2		
	A	B	C	A	B	C
Design 1	8,000	8,000	8,000	8,000	8,000	8,000
Design 2	8,000	0	8,000	8,000	8,000	8,000
Design 3	8,000	0	0	8,000	8,000	8,000
Design 4	2,000	2,000	2,000	8,000	8,000	8,000
Design 5	2,000	0	2,000	8,000	8,000	8,000

amount of data. It is not easy since the values of the variables that make the SC to perform properly cannot be analysed in isolation, variable by variable, but taking into account the interaction amongst them. Thus, the analysis has been based on defining a solution space with arbitrary boundaries, sampling different points from this space and then computing the objective function for each of these points. From the analysis of the results, those points with really bad objective function values have been related to some combinations of the parameter values, so as to choose a range for these parameters. The detailed explanation of the procedure and results has been omitted for the sake of clarity.

Five sets of capacity values for the DCs are evaluated (see Table 1). For each set of the design variables, a GA properly designed and tuned is run over the discrete-event simulator. The GA handles 27 variables: the *s* and *R* inventory parameters at the DCs (12 variables), and the *s* parameters at the retailers (15 variables). Real-valued encoding and maximum number of generations as termination criterion are used. A 1-year horizon of time is considered with daily precision for each simulation run. The multi-agent system was run in an AMDK6 computer, 2.16 GHz, 512 MB. With regard to the CPU time, for each configuration only a few hours (3–10 hr) are required to evaluate the expected profit. It is also important to observe that this time depends on the number of simulation runs to be made, which is given by certain tuning parameters of the GA such as the number of generations and the number of individuals in each population. The time requirements are acceptable for such a long-term decision problem, and perhaps it is the cost to be paid for a very realistic representation as it is the multi-agent model.

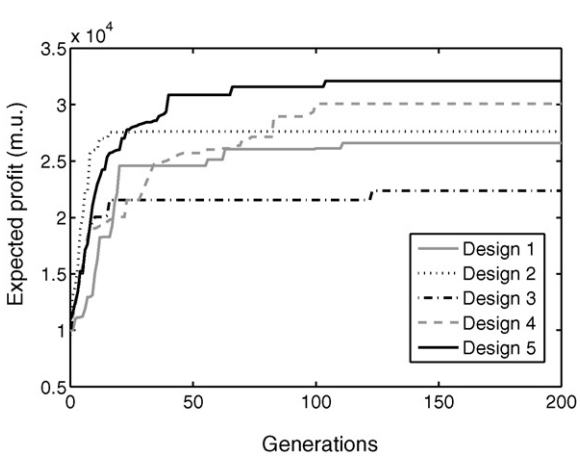

Fig. 6. Average objective function evolution during the GA run.

Fig. 5 shows the inventory level evolution at distribution centre D2 for the best solution (Design 5 in Table 1) and for a solution found using the configuration given by Design 1.

Fig. 6 shows the evolution of the objective function expected profit through 200 generations. The curve represents the average value of profit for about 10 GA runs (each run involves 200 generations). Repeating the same procedure for each configuration, the decision-maker can determine the best strategic decision to be implemented in the network in terms of the resulting expected profit achieved within the time horizon of the analysis.

6. Conclusions

In this work, the SC design and retrofit problem has been addressed. Strategic decisions have been made taking into account their impact at a lower level. The performance of each SC configuration has been assessed through a dynamic multi-agent model that has been coupled with GAs in order to optimise the operation variables associated to each design candidate. The computation cost of the proposed approach depends on the problem dimension and its level of complexity. However, the time requirement is reasonable for such a long-term decision. On the other hand, the use of multi-agent systems allows modelling in a very realistic manner complex distributed SCs, particularly those which imply the combined use of heuristic rules and mathematical programming tools and operate under uncertainty.

Acknowledgements

The authors express their gratitude for the financial support received from the Spanish Ministry of Education, Culture and Sport (FPU research grant to Gonzalo Guillén), the Spanish Ministry of Science and Technology (project DPI2002-00856), the Generalitat of Catalonia (project I-0898 and an FI research grant to Fernando D. Mele) and the European Union (project MRTN-CT-2004-512233).

Appendix A. Brief agent description

The following list briefly describes each type of emulation agent.

A.1. External customer agent

This agent represents outside customers that issue orders to the SC being modelled. Orders are generated on the basis of customer demand, which may be modelled as a sequence of specific customer orders obtained from historical records, as an aggregated demand over a period of time using forecasting techniques or as data sampled from a probability distribution.

A.2. Site agent

A site is defined as a particular location in the SC network that can deliver a set of products from a set of input materials by utilising a set of technologies. This definition is general enough to include a manufacturing plant or simply a warehouse or packaging site. Site agents have sub-agents that perform internal tasks at a site. Fig. 7 shows an example of the internal representation of a generic unit, that is a unit that can represent any SC node, using a state-transition diagram. The boxes represent each of the states among which the system switches. The dotted lined boxes are states that can operate in parallel while the other ones are sequential states. Each circle is a decision node where logical conditions are evaluated and every line represents a logical condition that must be satisfied to go from one state to another.

- The state *Order_reception* operates driven by the *ORin* event. This event makes the diagram to abandon the state *Idle* in order to analyse whether there is inventory enough to satisfy the order. If the order can be satisfied, state *Inv1* is activated and if it cannot, the system returns to the state *Idle* creating a note in a list or queue. In state *Inv1*, the ordered quantity is subtracted from the inventory level *I*, and it is transferred to a repository *Rep* from which it will be ready for delivering. In the case that the generic unit is used to represent the production line behavior, that is at the plants, the state *Order_reception* calls a program which gives the arriving time of the batch of each product to the inventory of finished products as well as generates orders for depleting the inventory of raw materials.
- The state *Material_reception* depends on the event *MAin*, that is, on the material arriving. This event carries the system from the state *Idle2* to state *Inv2* in which material is added to the inventory level *I*.
- The state *Delivering*, by seeing that the content of *H* is greater than a given threshold, sends off the event *MAout* outside the generic unit. The time between two consecutive material deliveries as well as the threshold magnitude is a parameter that can be modified.
- The state *Ordering* implements a control law over the inventory level, that is the criterion in which the request of materials from the suppliers is based on. This state generates the event *ORout* associated to a material quantity.

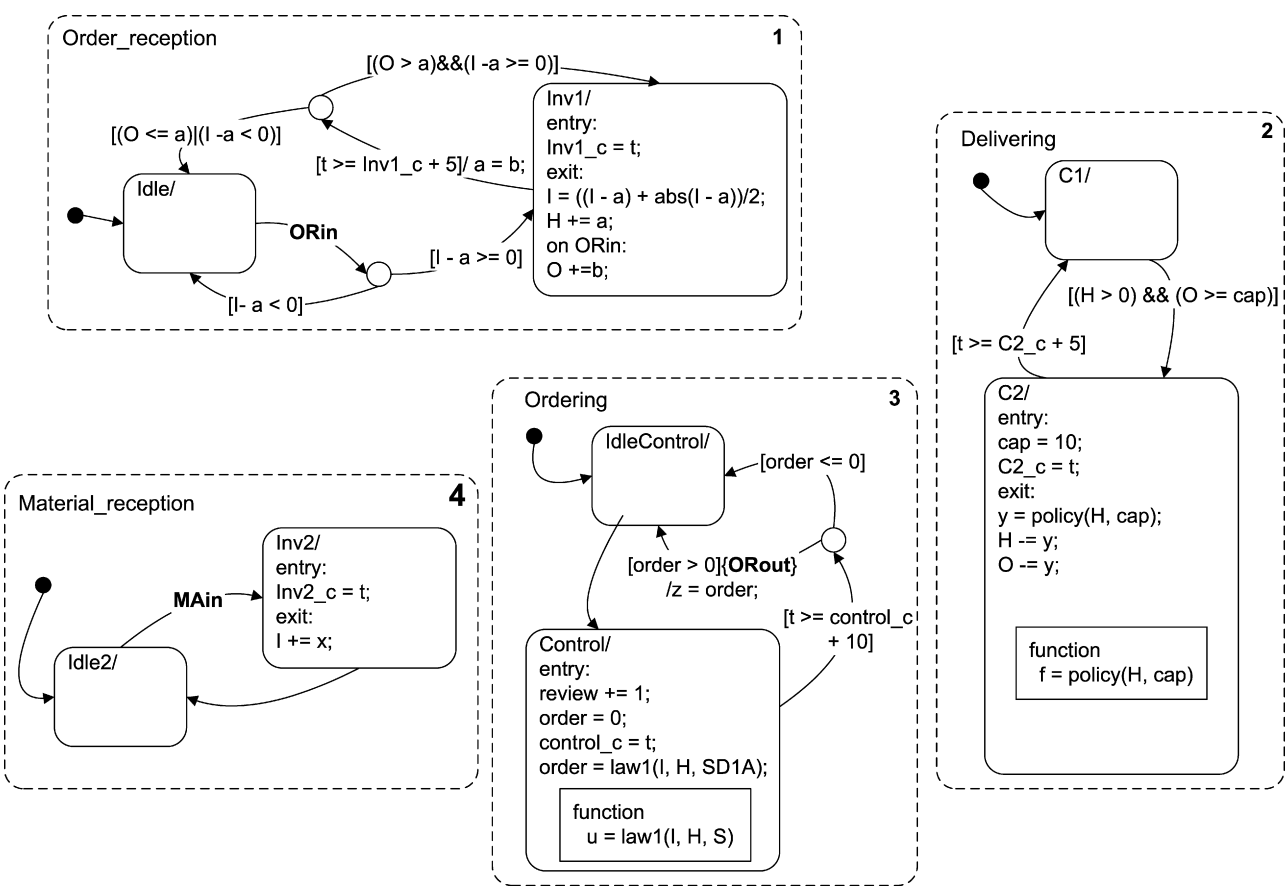

Fig. 7. A SC generic unit in Stateflow.

The inclusion of uncertainty is easily managed as the simulator accepts the definition of parameters and the occurrence of events on the time line as belonging to probability distributions.

Since this generic unit might represent a plant, a warehouse or whatever component of a SC, it can be connected with other ones to represent the entire SC in any case studied. On receiving messages from other sites, this agent decides the further course of action.

A.3. External supplier agent

This agent represents outside suppliers that send materials to the SC being modelled. It has an approximated model of the external suppliers of the chain. In this paper, the model is a conversational one that enables the agent to answer the requests of materials from the SC. It is able to calculate the delivery times as a function of the amounts requested. This model is necessary to represent the supplier behavior and then to be able to study the negotiation processes with suppliers.

References

Acevedo, J., & Pistikopoulos, E. N. (1998). Stochastic optimisation based algorithms for process synthesis under uncertainty. *Computers and Chemical Engineering*, *22*, 647–671.

Ahmed, S., & Sahinidis, N. V. (1998). Robust process planning under uncertainty. *Industrial and Engineering Chemistry Research*, *37*, 1883–1892.

Andradóttir, S. (1998). A review of simulation optimisation techniques. In D. J. Medeiros, E. F. Watson, J. S. Carson, & M. S. Manivannan (Eds.), *Proceedings of the Winter Simulation Conference* (pp. 151–158).

Anupindi, R., & Bassok, Y. (1999). Supply chain contracts with quantity commitments and stochastic demand. In *Quantitative models for supply chain management*. Norwell, MA: Kluwer Academic Publishers.

Applequist, G. E., Pekny, J. F., & Reklaitis, G. V. (2000). Risk and uncertainty in managing manufacturing supply chains. *Computers and Chemical Engineering*, *24*, 2211–2222.

Bernardo, F., Pistikopoulos, E., & Saraiva, P. (1999). Integration and computational issues in stochastic design and planning optimisation problems. *Industrial and Engineering Chemistry Research*, *38*, 3056–3068.

Bok, J. K., Grossmann, I. E., & Park, S. (2000). Supply chain optimisation in continuous flexible process networks. *Industrial and Engineering Chemistry Research*, *39*, 1279–1290.

Bose, S., & Pekny, J. F. (2000). A model predictive framework for planning and scheduling problems: A case study of consumer goods supply chain. *Computers and Chemical Engineering*, *24*, 329–335.

Cohen, M. A., & Lee, H. L. (1989). Resource deployment analysis of global manufacturing and distribution networks. *Journal of Manufacturing and Operation Management*, *2*, 81–104.

Forrester, J. M. (1961). *Industrial Dynamics*. Cambridge, MA: MIT Press.

Fox, M., Barbuceanu, M., & Teigen, R. (2000). Agent-oriented supply chain management. *The International Journal of Flexible Manufacturing Systems*, *12*, 165–188.

Fu, M. C. (2002). Optimisation for simulation: Theory vs. practice. *INFORMS Journal on Computing*, *14*, 192–215.

Gjerdrum, J., Shah, N., & Papageorgiou, L. G. (2000). A combined optimisation and agent-based approach for supply chain modelling and performance assessment. *Production Planning and Control*, *12*, 81–88.

Gupta, A., & Maranas, C. D. (2000). A two-stage modelling and solution framework for multisite midterm planning under demand uncertainty. *Industrial and Engineering Chemistry Research*, *39*, 3799–3813.

Gupta, A., & Maranas, C. D. (2003). Managing demand uncertainty in supply chain planning. *Computers and Chemical Engineering*, *27*, 1219–1227.

Han, C., Douglas, J. M., & Stephanopoulos, G. (1995). Agent-based approach to a design support system for the synthesis of continuous chemical processes. *Computers and Chemical Engineering*, *19*, s63–s69.

Ierapetritou, M. G., & Pistikopoulos, E. N. (1996). Batch plant design and operations under uncertainty. *Industrial and Engineering Chemistry Research*, *35*, 772–787.

Iyer, R. R., & Grossmann, I. E. (1998). A bilevel decomposition algorithm for long-range planning of process networks. *Industrial and Engineering Chemistry Research*, *37*, 474–481.

Julka, N., Srinivasan, R., & Karimi, I. (2002). Agent-based supply chain management-1: Framework. *Computers and Chemical Engineering*, *26*, 1755–1769.

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis, G. V., & Eversdyk, D. (2004). A simulation based optimisation approach to supply chain management under demand uncertainty. *Computers and Chemical Engineering*, *28*, 2087–2106.

Law, A. M., & Kelton, W. D. (2000). *Simulation modeling & analysis* (3rd ed.). New York: McGraw-Hill.

Maguire, P. Z., Scott, D. M., Paterson, W. R., & Struthers, A. (1995). Development of an advanced modelling environment. *Computers and Chemical Engineering*, *19*, S265–S270.

Mele, F. D., Guillén, G., Espuña, A., & Puigjaner, L. (2006). A simulation-based optimization framework for parameter optimization of supply-chain networks. *Industrial and Engineering Chemistry Research*, *45*, 3133–3148.

Mele, F. D., Guillén, G., Urbano, F., Espuña, A., & Puigjaner, L. (2004). A novel agent-based approach for supply chain retrofitting. In *Proceedings of the sixth international conference on foundations of computer-aided process design* (pp. 461–464).

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

Petkov, S. B., & Maranas, C. D. (1997). Multiperiod planning and scheduling of multipurpose batch plants under demand uncertainty. *Industrial and Engineering Chemistry Research*, *36*, 4864–4881.

Sakawa, M., Nishizaki, I., & Uemura, Y. (2001). Fuzzy programming and profit and cost allocation for a production and transportation problem. *European Journal of Operational Research*, *131*, 1–15.

Sauter, J. A., Parunak, H. V. D., Goic, J. (1999). ANTS in the supply chain. *Workshop on agents for electronic commerce at agents 99*, 27, 770.

Siirola, J. D., Hauan, S., & Westerberg, A. W. (2003). Toward agent-based process system engineering: Proposed framework and application to non-convex optimization. *Computers and Chemical Engineering*, *27*, 1801–1811.

Siirola, J. D., Hauan, S., & Westerberg, A. W. (2004). Computing Pareto fronts using distributed agents. *Computers and Chemical Engineering*, *29*, 113–126.

Silver, E. A., Pyke, D. F., & Peterson, R. (1998). *Inventory management and production planning and scheduling*. New York: John Wiley & Sons.

Simchi-Levi, D., Kamisky, P., & Simchi-Levi, E. (2000). *Designing and managing the supply chain. Concepts, strategies, and case studies*. Irwin/McGraw-Hill.

Subrahmanyam, S. (1996). *Issues in the design and planning of batch chemical plants*. PhD Thesis, Purdue University, West Lafayette, USA.

Subramanian, D., Pekny, J. F., & Reklaitis, G. V. (2000). A simulation–optimisation framework for addressing combinatorial and stochastic aspects of an r&d pipeline management problem. *Computers and Chemical Engineering*, *24*, 1005–1011.

Swaminathan, J. M., Smith, S. F., & Zadeh, N. M. (1998). Modelling supply chain dynamics: A multi-agent approach. *Decision Sciences*, *29*, 607–632.

Syarif, A., Yun, Y., & Gen, M. (2002). Study of multi-stage logistic chain network: A spanning tree-based algorithm approach. *Computers and Chemical Engineering*, *43*, 299–314.

The Math Works (2004). MATLAB 7.0 (R14), Simulink 6.0, Stateflow 6.0. User's Manual, The Math Works Inc., Natick, MA.

Timpe, C. H., & Kallrath, J. (2000). Optimal planning in large multi-site production networks. *European Journal of Operational Research*, *126*, 422–435.

Tsiakis, P., Shah, N., & Pantelides, C. C. (2001). Design of multi-echelon supply chain networks under demand uncertainty. *Industrial and Engineering Chemistry Research*, *40*, 3585–3604.

Vergara, F. E., Khouja, M., & Michalewicz, Z. (2002). An evolutionary algorithm for optimizing material flow in supply chains. *Computers and Industrial Engineering*, *43*, 407–421.

Wooldridge, M., & Jennings, N. R. (1995). Intelligent agents: Theory and practice. *Knowledge Engineering Review*, *10*, 115–152.

Zhou, G., Min, H., & Gen, M. (2002). The balanced allocation of customers to multiple distribution centers in the supply chain network: A genetic algorithm approach. *Computers and Industrial Engineering*, *43*, 251–261.