European Journal of Operational Research 199 (2009) 349–358

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Distributed supply chain management using ant colony optimization

C.A. Silva [a,*,1], J.M.C. Sousa [a,1], T.A. Runkler [b], J.M.G. Sá da Costa [a]

[a] Technical University of Lisbon, Instituto Superior Técnico, Department of Mechanical Engineering, CSI/IDMEC, 1049-001 Lisbon, Portugal
[b] Siemens AG, Corporate Technology, Information and Communications, Learning Systems Department, 81730 Munich, Germany

ARTICLE INFO

Article history:
Received 20 November 2007
Accepted 18 November 2008
Available online 30 November 2008

Keywords:
Ant colony optimization
Distributed optimization
Supply chain management
Logistics and operations management

ABSTRACT

Successful supply chain management requires a cooperative integration between all the partners in the network. At the operational level, the partners individual behavior should be optimal and therefore their activities have to be planned using sophisticated optimization tools. However, these tools should take into account the planning of the remaining partners, through the exchange of information, in order to allow some kind of cooperation between the elements of the chain. This paper introduces a new supply chain management technique, based on modeling a generic supply chain with suppliers, logistics and distributers, as a distributed optimization problem. The different operational activities are solved by the optimization meta-heuristic called ant colony optimization, which allows the exchange of information between different optimization problems by means of a pheromone matrix. The simulation results show that the new methodology is more efficient than a simple decentralized methodology for different instances of a supply chain.

1. Introduction

Supply chain (SC) systems are nowadays entering the age of adaptive and intelligent supply chains, a new generation of networks that features collaboration and visibility features across the different partners to deal with the system dynamics, such as supplier failures or demand uncertainty [6,21]. Supply chains systems are a set of separate and independent economic entities more interested in their local objectives than in the global system performance. Therefore, *centralized management* approaches, where a single partner such as the logistic center optimizes the global performance, are becoming less realistic and being replaced by *decentralized management* approaches, where each member optimizes its own performance, albeit knowing that collaboration with other partners can improve the individual and global performance. In any case, the key issue is to align the members objectives and coordinate their decisions to optimize the supply chain performance, but this is particularly more difficult to attain with a decentralized management approach [13].

At the operational level, supply chain management (SCM) is now seeking to determine the stock levels at the logistic centers depending on the demand, or the size and frequency of batches produced at the suppliers to feed the producers in time, or even the delivery planning that minimizes the transportation costs and environmental impacts [8]. The coherence between the different decision making centers in the chain can be easily accomplished by a multi-agent framework [17]. These systems are based on explicit communication between specialized agents assigned to structural elements of the chain (e.g. supplying or logistic agents) about their tasks and using an interaction protocol with a specific message structure, conversation rules, action and reaction behaviors [1].

The research in this field has tackled mainly the interaction between the agents and the optimization issues are usually solved through some simple dispatching rules. However, these methods are usually not sufficient to deal with the complexity of the real-world problems and the agents need to use more powerful optimization techniques [14]. Moreover, to take full advantage of the supply chain framework, the communication protocols should support the possibility of exchanging information during the optimization process, in order to allow agents to react to failures or other type of dynamic disturbances. However, it is necessary to take into account all sort of problems regarding level of disclosure and the asymmetry between the supply chain members that can generate opportunistic behaviors.

This paper introduces a multi-agent supply chain management methodology based on the description of the supply chain as a set of different distributed optimization problems and using the *ant colony optimization* (ACO) meta-heuristic [10] to achieve cooperation between different multiple partners. While optimizing, the ACO algorithm builds a *pheromone matrix*, which is an indirect record of the optimization steps. This matrix can be accessed at all times during the optimization process and contains no private

* Corresponding author.
E-mail address: jmsousa@ist.utl.pt (C.A. Silva).

1 The work is supported by the Project PPCDT/EME/59191/2004, co-sponsored by FEDER, Programa Operacional Ciência e Inovação 2010, FCT, Portugal.

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.11.021

information of any kind. If each system in a supply-chain is optimized by its own ant colony-agent, the pheromone matrices can be used to exchange information between the different systems as a multi-agent system, introducing in this way a coordination mechanism into the supply chain.

The paper proceeds as follows: Section 2 describes the SC model that is used in this paper. Section 3 presents a short literature survey on *decentralized* supply chain management and models the management problem as a distributed optimization problem. Section 4 shows how the ant colony optimization can be used to solve this problem as a multi-agent system. The simulation results are presented in Sections 6 and 7 concludes the paper and presents the guidelines for future work.

2. Decentralized supply chain modeling

We consider a generic supply chain model that comprised three systems: a *logistic* system, its *suppliers* and the *distributers*. A previous study considering only the suppliers and the logistic system was presented in [19]. The suppliers–logistics–distribution system proposed in the paper takes full advantage of the distributed optimization by considering all echelons of the supply-chain. The logistic system collects the orders from the customers, purchases the components from external suppliers and schedules the components delivered by the suppliers as orders. This system is also responsible to manage the cooperation between the partners in the network and for the communication between the clients and the supply chain. The supply system is a network of external suppliers that manufacture the components and deliver them to the logistic system. The distributers are responsible to pick up different items at the warehouses of the *logistic system* and distribute them to clients. Fig. 1 presents a schematic representation of such multiple echelon system.

The modeling approach proposed in this paper consists of describing each of the subsystems of the supply chain by a benchmark optimization problem: the logistic system is described by the general assignment optimization problem [20]; the supplying system by the manufacturing scheduling optimization problem [14]; and the distribution system by the vehicle routing problem [9].

2.1. Logistic system

At each day, the logistic system has an *order list* O of n orders waiting to be delivered. An order $o_j \in O$ with $j = 1,\ldots,n$ is a set of ℓ different types of items, called the components c_i with $i = 1,\ldots,\ell$, in certain quantities q_{ij}. Therefore, an order can be defined as an ℓ-tuple $o_j = (q_{1j},\ldots,q_{\ell j})$. When a new order o_j arrives, it receives two labels: the arrival date or *release date* r_j and the desired delivery date or *due date* d_j, which is the date when the client wishes to receive the order. The order is delivered at the completion date C_j.

Assuming that the system does not deliver orders if they are ready before the due date, the difference between the completion date and the due date is called the *tardiness* $T_j = C_j - d_j$. The objective is to match both dates, i.e. to have for all orders $T_j = 0$. However, two disturbances may influence the system: the fact that suppliers service may not be respected and the fact that some clients ask for desired delivery dates not compatible with the supplier services.

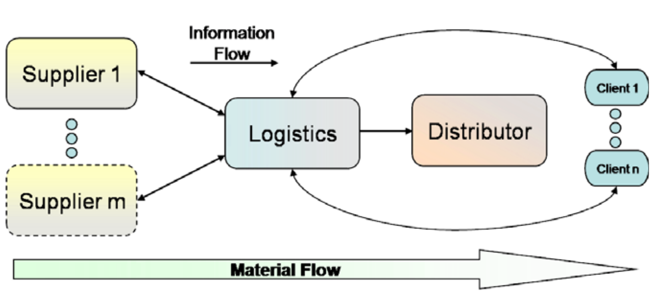

Fig. 1. Generic multi-echelon supply chain system.

In order to define the cost function that best describes the objective of the logistic system, consider the following definitions. Given the set O of orders in the system waiting to be delivered, the subset of orders that are going to be *delivered* is defined as $O_D \subseteq O$. The complementary subset of orders that are *not delivered* and remain in the system is defined as $O_{ND} \subseteq O$, such that $O_D \cup O_{ND} = O$. Consider further that the subset of orders that are delivered at the correct date is defined as $O_D^0 \subseteq O_D$ and the subset of orders that are not delivered and are already delayed is defined as $O_{ND}^d \subseteq O_{ND}$. The optimization objective is to minimize the cost function given by

$$f_L = \frac{w_A\left(\sum_{j\in O} T_j\right) + w_B|O_{ND}^d|}{w_C|O_D^0| + \epsilon}, \tag{1}$$

where $\sum_{j\in O} T_j$ accounts for the minimization of the tardiness of the total set of orders in the system O; $|O_{ND}^d|$ is the cardinality of the subset O_{ND}^d and refers to the minimization of the number of orders that are not delivered and are already delayed; and finally $|O_D^0|$ is the cardinality of the subset O_D^0 and accounts for the maximization of the number of orders delivered at the correct date. The ϵ is a small constant that avoids the infinity value when no orders are delivered at the correct date. The weights w_A, w_B, w_C are used to convert the cost to the same unit, e.g. monetary units, or to balance the different variables in case the order of magnitude of the variables is very different. In this case, $w_A = 1/\text{day}$, $w_B = w_C = 1$. This decision step is done once per day, but different solutions for the same daily problem originate different next-day scheduling problems. The supply chain management is a dynamic succession of daily optimization problems, that are treated independently, even though they are not. Therefore, to evaluate the performance of the supply chain, larger periods of time, such as weeks or months, should be considered.

2.2. Supplying system

The supplying sub-system is a network of m different suppliers or manufacturers M_i with $i = 1,\ldots,m$, each one producing its own set of jobs J_{M_i}, where each job refers to a type of component c_i requested by the logistic sub-system. Each supplier is independent and therefore it optimizes its own problem called the *local supplier* problem. However, from the point of view of the logistic system, the suppliers can be virtually considered as one single entity, and the optimization problem is called the *global supplier* problem.

2.2.1. Local supplier problem

The optimization problem of each M_i supplier, called here the local supplier problem, can be modeled as a single machine scheduling problem [14]. There is one machine that produces all the n_i jobs on the waiting list J_{M_i} of the supplier. The objective function to be minimized is the total tardiness of all the jobs

$$f_{M_i} = \sum_{j\in J_{M_i}} w_T \cdot T_j, \tag{2}$$

where tardiness is defined as in the logistic system and w_T is a weight to convert the cost from days to other unit, such as monetary units.

2.2.2. Global supplier problem

The global supplier problem describes the supply system from the logistics point of view. The problem can be modeled as an open jobshop problem with m machines, where each machine corresponds to a different supplier [14]. In this problem, the set of m machines produces all jobs $J = J_{M_1} \cup \ldots J_{M_m}$, but each machine i can only produce the jobs J_{M_i}. The objective is to minimize the objective function f_L of the logistic system defined in (1). The definition of the supplying problem from the logistic system's perspective is necessary to describe the cooperation mechanism proposed in this paper.

2.3. Distribution system

When the scheduling method has decided which orders must be delivered, a distribution company will pick-up the assigned components and deliver them to the different clients. There is a direct correspondence between clients and orders, but clients are described in this case by their geographical location.

A simple but realistic model of the supply chain distribution problem is the *vehicle routing problem* (VRP)[9], where each vehicle has a limited load capacity F and each truck as a limited distance autonomy R. This problem considers the set O_D of n orders to be delivered by vehicles $i = 1,\ldots,s$. The function to minimize is the sum of the individual travel costs of each vehicle i of the total number of vehicles s required to deliver all the n orders in the set O_D of delivered orders

$$\text{minimize} \quad f_D = \sum_{u=0}^{n}\sum_{v=0}^{n}\sum_{i=1}^{s} w_{D_{uv}} y_{uvi} \quad (\text{route cost}), \tag{3}$$

where y_{uvi} is *1* when vehicle i has traveled between clients u and v and *0* otherwise, and $w_{D_{uv}}$ is the travel cost between client u and client v expressed in Km or monetary units. If a client requests by any chance more than one order, the distribution company considers the existence of more than one client at the same location.

2.4. Performance index for supply chain management

The definition of performance measurements for supply chain has not become yet a mature subject [2]. There are already different approaches that depend on: the variable that is measured, e.g. cost, time or customer responsiveness; the measurement framework, e.g. aggregation expressions or multi-criteria decision measures.

In this paper, we use an aggregation measurement based on the cost of each of the partners in the supply chain, since it the most widely used performance index [2]. The general expression of this index is given by

$$\mathscr{P}_{\text{SCM}} = \sum_{i=1}^{n} \omega_i \times f_i, \tag{4}$$

where $i = 1,\ldots,n$ is the number of partners in the supply chain, ω_i is the weight that measures the importance of the partner in the network and f_i is the contribution of each of the partners to the evaluated cost of the supply chain, e.g. monetary units.

3. Distributed supply chain management

Decentralized supply chain management can be defined as implementing mechanisms that promote the adoption of local actions in a coordinated fashion by each individual member while optimizing their own performance that conduce to the optimization of the global supply chain. However, local system problems and information parameters are very problem-specific, and the cooperation mechanism must be sufficiently general to allow an efficient exchange of information across all systems. As all the subsystems can be described as optimization problems in relatively similar optimization spaces, it is possible to design such a cooperation system based on a distributed optimization paradigm.

3.1. Literature review on decentralized SCM

Decentralized SCM coordination mechanisms usually follow one of three approaches: inventory control, quantity discounts and contracting [16]. Most of the research in this area is based on the classic work of Clark and Scarf that described the problematic of inventory optimal policies on two-echelon systems, the supplier and the distributer [7]. The material flow goes from the supplier to the clients through the distributer, and the echelons cooperate between each other by the exchange of some kind of information. The scope of the optimization problems described in the literature generally refers to the strategic level of the supply chain and the optimality of those policies is derived assuming extended knowledge about the market conditions and the partners' internal information.

Jeuland and Shugan introduced the coordination concept from the marketing point of view in [12], where a one-supplier–one-retailer chain model achieved coordination in pricing decisions with quantity discounts. This concept was extended to the operational research field by Weng in [23], through the introduction of the demand issue into the coordination decision. In that paper, a one-supplier–one-retailer chain achieved maximum joint profit with a retailer optimal ordering decision at an optimum retail price. These works were based on the assumption that the demand is deterministic and that the suppliers have complete market knowledge. Qi et al. have recently developed the analysis to stochastic demand conditions [15]. Schneeweiss and Zimmer proposed in [16] a hierarchical coordination mechanism based on sharing private information between a producer (the logistic center as defined in Section 2) with deterministic demand and the supplier with stochastic demand. All these works describe the coordination as an optimization problem and propose analytical solutions.

The cooperation concept between the different agents in the supply chain gains a considerable strength when analyzed from the *Game Theory* point of view, as introduced by Cachon in [5]. This work describes the supply chain management as a non cooperative game between the different partners. The fact that the cooperation between partners leads to a global optimization is easily explained in the light of the Nash equilibrium concept. This arsenal of new tools led to very interesting results to analyze inventory problems between manufacturers and distributers, especially in competition scenarios or the development of contracting policies between supply chain partners [6].

3.2. General definition of distributed optimization

More than 20 years ago, Bertsekas proposed in [3] the distributed dynamic programming concept to solve large optimization algorithms using parallel computational power provided through the connection of computers over a network. This idea evolved to distributed gradient optimization algorithms [22]. This generalization considers the optimization of multi-dimensional functions using gradient based optimization algorithms. The basic idea of distributed dynamic programming is that it is possible to optimize a function defined in an n-dimensional space using n computation centers. Each one optimizes the function on its own direction and exchanges information periodically with the other computational centers, such that convergence to a global optimum can be achieved under certain conditions. One of the optimization problems originally proposed in [3] was the optimization of a path on a graph.

Based on this concept, we define in this paper the *distributed optimization* paradigm:

Definition 1. Let a certain optimization problem $X = \{X_1, \ldots, X_n\} \in \mathscr{X}^n$ consist of a set of n sub-problems X_k, with $k = 1, \ldots, n$. Let $f = f_1 \circ \cdots \circ f_n$ be the cost function of the problem X, where each of the n cost functions f_k corresponds to the sub-problem X_k and $\circ$ is some aggregation operator. *Distributed optimization* is a methodology where the n optimization (e.g. minimization) processes $\min[f_k]$ are running in parallel, solving cost functions f_k for each sub-problem X_k. The global optimization is defined as

$$\min[f(t)] = \Diamond_{k=1}^{n} \min[f_k(t - l_k)] \\ = \min[f_1(t - l_1)] \Diamond \cdots \Diamond \min[f_n(t - l_n)], \tag{5}$$

where t describes the actual optimization iteration, l_k describes n different and previous optimization iterations and $\Diamond$ describes the information exchange between the different subsystems.

The global problem X, defined on the n-dimensional space $\mathscr{X}^n$ consists of a set of n local problems X_k, where each problem is solving its own local cost function f_k. The sub-spaces $\mathscr{X}_k$ where each problem X_k is defined, are not disjunctive, thus the optimization of the cost function f_k of problem X_k can depend on the solution of the cost function f_h of problem X_h and vice-versa, with $h \neq k$ and $h = 1, \ldots, n$. Therefore, the cost function $f = f_1 \circ \cdots \circ f_n$ consists of a highly nontrivial aggregation $\circ$ of the different local cost functions f_k. To achieve a minimization solution for the global problem X, which in anytime is guaranteed to be optimal, each of the optimization problems has to consider the exchange of information with the remaining problems. The $\Diamond$ indicates this information exchange, which can be *synchronous* or *asynchronous*:

Definition 2. Consider the global optimization process $\min[f(t)] = \min[f_1(t - l_1)] \Diamond \cdots \Diamond \min[f_n(t - l_n)]$. If $\forall_{k=1,\ldots,n} l_k = 0$, the distributed optimization is said to be *synchronous*. If $\exists_{k=1,\ldots,n} l_k \neq 0$, the distributed optimization is said to be *asynchronous*.

This means that at every iteration in the synchronous optimization, the methods access actual information from all the other methods. In the asynchronous method, at every iteration each optimization method can access information from previous iterations and at different moments l_k in time. The asynchronous method is used to avoid convergence problems of each $\min[f_k]$ optimization problem, for example in cases where the computational effort of one iteration is different from problem to problem, as described in [22].

Algorithm 1 Asynchronous distributed optimization.

for $t_1 = 1 : N_{\max}^1$ **do**		**for** $t_n = 1 : N_{\max}^n$ **do**	**for** $t = 1 : N_{\max}$ **do**
Optimize the problem f_1		Optimize the problem f_n	Optimize the problem f
if $\mathrm{mod}(t_1, l_1) = 0$ **then**		**if** $\mathrm{mod}(t_n, l_n) = 0$ **then**	**if** $\mathrm{mod}(t_i, l_i) = 0, i = 1, \ldots, n$ **then**
$\min[f_1(t - l_1)] \diamond \min[f(t)]$	...	$\min[f_n(t - l_n)] \diamond \min[f(t)]$	$\Diamond_{k=1}^{n} \min[f_k(t - l_k)]$
end if		**end if**	**end if**
end for		**end for**	**end for**
Optimize the problem f_1		Optimize the problem f_n	

The generic asynchronous distributed optimization algorithm is given in Algorithm 1. This algorithm shows that each local problem X_k is being optimized by its own procedure $\min[f_k]$ during $N_{\max}^k$ iterations, but every l_k iterations it exchanges information with the other optimization problems. This exchange of information may occur Z times depending on the parameters l_k and $N_{\max}^k > l_k$ parameters, such that $Z = \frac{N_{\max}^k}{l_k}$.

At the end of the distributed optimization, the local problems re-optimize their own problem independently to ensure that their local solution is optimal. However, in cases where the local problems have different optimal solutions, the algorithm drives the local optimizers to the solution that best satisfies the global problem. Therefore, the distributed decision is in general accepted only if it is locally as optimal as if the partner had optimized the process on its own. If it is much worse, the last local optimization procedure will in general drive the local solution back to a better local solution. However, since the best local solution is never saved, it may happen that the final local solution is not the best ever found.

3.3. SCM using distributed optimization

The supply chain problem presented in the previous chapters can be described as the management problem $X_{\mathrm{SCM}} = \{X_L, X_{M_1}, \ldots, X_{M_m}, X_D\}$, where X_L is the logistic scheduling problem, $X_{M_i}, i = 1, \ldots, m$ are the local supplying problems and X_D is the distribution problem. We consider that the supply chain cost function is

$$f_{\mathrm{SCM}} = f_{M_1} \circ \cdots \circ f_{M_m} \circ f_L \circ f_D, \tag{6}$$

where f_L, f_{M_i} and f_D are the expressions proposed in (1)–(3), respectively. Observe that the global supply chain cost function f_{SCM} can be the performance index $\mathscr{P}_{\mathrm{SCM}}$ defined in (4), if the aggregation operator $\circ$ is simply a weighted sum operation.

Note that the supply chain management is done at two distinct moments in time:

- First, the suppliers decide about their own scheduling policy, that may be influenced or not by the logistic system before it receives the new components at the cross-docking center. However, the distribution system is not related to this decision process, since there are no items to distribute.
- Then, the logistic process has to decide which orders to deliver after it receives the new components, under the influence or not of the distribution system. However, the components are already produced and the supplying system is not related to this decision process.

Thus, the daily supply chain management problem $\min[f_{\mathrm{SCM}}]$ is naturally divided into two sequential cooperation problems at each day, which are the *supplying–logistic* problem $\min[f_{\mathrm{SL}}]$ and the *logistic–distribution* problem $\min[f_{\mathrm{LD}}]$:

$$\min[f_{\mathrm{SCM}}] = \min[f_{\mathrm{SL}}] \to \min[f_{\mathrm{LD}}], \tag{7}$$

where the $\to$ operator indicates the time sequence. The supplying–logistic problem is defined as

$$\min[f_{\mathrm{SL}}] = \min[f_{M_1}] \Diamond \ldots \Diamond \min[f_{M_m}] \Diamond \min[f_L]. \tag{8}$$

The logistic–distribution problem is defined as

$$\min[f_{\mathrm{LD}}] = \min[f_L] \Diamond \min[f_D]. \tag{9}$$

Observe that one advantage of this formulation is that it allows a total separation between the suppliers and the distribution systems. It is possible for the logistic system to use the distribution optimization technique with one of the partners and use a completely different coordination mechanism with the other partner.

Since the global supply chain optimization problem can be described as a set of different but coupled distributed problems, the optimization process can allow each problem to use intermediate optimization results of the other problems. In this way, at the end of the management procedure, the global solution is optimized according to the optimal solutions of all the local problems, although this final global solution is not necessarily optimal. Next section shows how this procedure can be easily implemented using the ant colony optimization algorithm.

4. Ant colony optimization

Ants live in colonies and all their actions are towards the survival of the colony as a whole, rather than the benefit of a single individual of the society. Individual ants have no special abilities. They communicate between each other using chemical substances, the pheromones. This type of communication allows the entire colony to perform complex tasks, such as establishing the shortest route paths from their nests to feeding sources. The ACO algorithm mimics the behavior of real ants in a graph environment, with n nodes and arcs between the nest and the food source. ACO has been successfully used to solve the traveling salesman problem (TSP) or jobshop problems [10]. In general, ACO algorithms are a very interesting approach to solve problems that can be translated into seeking the minimum cost of a graph structure, especially when the connection costs can change over time, i.e., when problems are dynamic.

The general framework of the algorithm can be described as follows. Consider a problem that consists of finding the path with the minimum cost on a graph. For a certain ant k placed in node u, the probability of this ant choosing the next trail leading to node v is given by

$$p_{uv}^k(t) = \begin{cases} \frac{\tau_{uv}^{\alpha} \cdot \eta_{uv}^{\beta}}{\sum_{w \in \mathcal{N}^k} \tau_{uw}^{\alpha} \cdot \eta_{uw}^{\beta}}, & \text{if } v \in \mathcal{N}^k, \\ 0 & \text{otherwise}, \end{cases} \tag{10}$$

where τ_{uv} is the pheromone concentration in the edge (u,v), η_{uv} is a *heuristic function* that conducts the search with some valuable information about the problem under optimization, and $\mathcal{N}^k$ is the *admissible nodes list* of ant k that contains all the nodes the ant can choose, and acts as the memory of the ant. The parameters α and β measure the relative importance of trail pheromone intensity and heuristic information, respectively.

A *tour* (or a path on the graph) is a complete route between the nest and the food source performed by one ant. An *iteration t* is the set of tours performed by all the g ants. After completing an iteration, the ants deposit pheromones on the edges corresponding to their individual tours. The update of the pheromone concentration on the trails is given by

$$\tau_{uv}(t+1) = \tau_{uv}(t) \times (1-\rho) + \sum_{k=1}^{g} \Delta\tau_{uv}^k, \tag{11}$$

where $\rho \in [0,1]$ expresses the pheromone evaporation phenomenon and $\sum_{k=1}^{g} \Delta\tau_{uv}^k$ are the pheromones deposited on the trails (u,v) followed by all the g ants after a complete tour, such that

$$\Delta\tau_{uv}^k = \begin{cases} 1/f^k, & \text{if arc } (u,v) \text{ was used by ant } k, \\ 0, & \text{otherwise}. \end{cases} \tag{12}$$

Notice that f_k is the value of the evaluation function of ant k in a minimization problem. Further, the pheromone update may consider only the best ant q of each iteration, instead of all the ants [10]. Then, a new colony of g ants is released to perform another search, using already the pheromone trails left by the previous colony. The time interval taken by the g ants to do a complete tour over the n nodes is $g \times n$ iterations. The algorithm runs for $\mathcal{O}(g \times n \times N_{\max}) \approx \mathcal{O}(N_{\max}^3)$ time, where $N_{\max}$ is the maximum number of iterations.

4.1. Logistic optimization

In the logistic system, the orders waiting to be delivered are the nodes of the graph, the role of the ants is to find the minimum cost path connecting the orders that should be delivered, and the objective function f is the one defined in (1). It is considered that each ant is traveling with a bag containing the available stock and distributing it between the orders that it is visiting. An ant visits only orders that are possible to deliver: e.g., if an order needs 2 components c_i, and the ant only has one ($q_i = 1$), the ant does not visit that order. In this way, ACO only builds feasible solutions. When the stock bag is empty or the remaining components are not enough to deliver any missing order, the search for this ant is finished. Therefore, the algorithm does not have a constant number of iterations, since the number of visited nodes may not be the same from one ant to another. Note that in other implementations, as in TSP, the number of nodes to visit is fixed and equal to the number of nodes to visit [10]. Since the path is not closed, the initial starting point for each ant assumes an important role [20].

The heuristic function η is, in this case, an exponentially biased modification of the earliest due date (EDD) dispatching rule [14]: if an order has already a positive tardiness, the ant feels a stronger attraction to visit it, because the order is already delayed. So, the greater the tardiness (which corresponds to the earliest due date), the more visible to the ants the order is (which corresponds to a higher delivery priority as in EDD). In this case, the heuristic is deposited at the nodes, thus $\eta_{uv} = \eta_v$. As v is a node that represents an order j, and the heuristic function is defined in the interval [0,1], the value 0 corresponds to the order $o_j \in O$ that has the minimum tardiness $\min(T_j)$ and 1 corresponds to the most delayed order $\max(T_j)$. For each order j represented by node v, the heuristic function is given by

$$\eta_v = \frac{e^{\frac{T_j - \min(T_j)}{\max(T_j) - \min(T_j)}} - 1}{e - 1}. \tag{13}$$

The pheromone trails τ_{uv} and the heuristic values η_{uv} are restricted to the interval [0,1] and the pheromones track are initiated with the value of 0.5. When all the orders have the same tardiness, i.e. $\max(T_j) = \min(T_j)$, the heuristic function value is set to 1, since all the orders are equally late.

The admissible nodes list $\mathcal{N}^k$ is the list of orders which were not delivered yet and can still be delivered with the available stock. At the end of each iteration, when all the g ants have delivered all their bags of stocks, the cost function f_L is evaluated and the edges followed by the best ant are updated following the rule in (12), considering only the best ant q at each iteration.

4.2. Supplying optimization

To apply ACO to the supplying system, we follow the approach that consists of finding the optimal path in a graph representing the scheduling problem [10].

Consider the generic problem with a job set $J = \{1, 2, \ldots, n\}$, a machine set $M = \{M_1, M_2, \ldots, M_m\}$ and a set of $n \times m$ operations $H = \{H_{11}, H_{12}, \ldots, H_{nm}\}$ to be done. The first step of the implementation is to define two matrices: the *technological matrix* **T** with size $n \times m$ describing the sequence of operations of the n jobs in the m machines; and the *production time matrix* **P**, also with size $n \times m$, describing the processing time of each operation of the n jobs in the m machines. A null entry indicates for both matrices that job j is not processed in machine i.

The problem graph is a set of $n \times m + 1$ nodes, where node *0* represents the list of jobs. The pheromone matrix τ is a $(n \times m + 1) \times (n \times m + 1)$ matrix representing the weight of all the arcs indicating all possible scheduling sequences. Null entries in the technological matrix **T** correspond to null pheromone matrix τ entries. The heuristic matrix η is also a $(n \times m + 1) \times (n \times m + 1)$ matrix defined using some simple heuristic.

The admissible nodes list $\mathcal{N}^k$ of each ant is initialized with the first operation of all jobs, since all the ants start from the source node 0. At each move of the ants, this matrix is updated in the following way: if an operation of job j as indicated in $\mathbf{T}$ is chosen, then this operation is cleared from the matrices and the next operation of the job j is chosen. The algorithm stops when this matrix is empty, i.e. all the operations have been assigned. Then, the solution of each ant is evaluated in terms of a cost function. The solution is a sequence of all the operations and has to be transformed into a list of operations in each machine.

4.2.1. Local supplying optimization

In this case, we consider that each supplier has only one resource and the jobshop problem consists of finding the best assignment for a single machine problem. The heuristic matrix η is defined again as in the EDD dispatching rule, where orders are scheduled in increasing order of due dates d_j and then normalized to the interval $[0,1]$:

$$\eta_j = \frac{d_j - \min(d_j)}{\max(d_j) - \min(d_j)}. \tag{14}$$

Here, the ACO algorithm has to find the best sequence to produce the different jobs, considering the cost function as defined in (2) and the type of problem can be modeled as a TSP [14].

4.2.2. Global supplying optimization

In this case, we consider that each supplier is a resource of the logistic system and therefore the jobshop problem consists of finding the best assignment of an open jobshop problem with m machines (i.e. suppliers). The heuristic function η is the one defined in (14). The algorithm is the same as for the local supplier problem, and in the end the scheduling result describes the sequence of operations to be performed by each supplier, considering the cost function as defined in (1).

4.3. Distribution optimization

To solve the vehicle routing problem (VRP) that models the distribution process in the logistic supply chain, the artificial ants construct solutions by successively choosing clients to visit until all the clients have been visited, and thus all the orders have been delivered. Whenever the choice of a location leads to infeasible solutions, due to reasons of vehicle capacity or total route length, the depot is chosen as a final location to close the tour and a new tour with a new vehicle is started.

The heuristic information η_{uv} used in this case is the *weighted saving function*, as proposed in [4]:

$$s_{uv} = w_{0u} + w_{0v} - aw_{uv} + b|w_{0u} - \omega_{0v}|,$$

where w_{uv} is the distance cost between clients u and v, and w_{0u}, w_{0v} are the distance costs between clients u and v and the docking center 0. In this implementation, we use $a = 2$ and $b = 1$ as indicated in [4]. The heuristic matrix η is a normalized version of this heuristic:

$$\eta_{uv} = \frac{s_{uv} - \min(s_{uv})}{\max(s_{uv}) - \min(s_{uv})}. \tag{15}$$

All the ants started at the depot node 0. Then, the ants choose the different clients to visit based on (10). When the ant k cannot visit any other client, in order to avoid the violation of any of the constrains (maximal tour cost R or maximum load capacity F), the ant returns to the depot node 0 and updates the admissible nodes list $\mathcal{N}^k$. At the next step, the ants, starting again at node 0, repeat this procedure for the remaining clients until all the clients are visited. At the end, the solution of ant k will be a sequence of the type $(0,u,\ldots,0,v,\ldots,0)$. The number of used trucks will be equal to the number of 0's minus one.

5. Distributed supply chain management using ACO

The ant colony optimization is by definition a multi-agent system: several autonomous agents are solving their own problems, communicating with each other over the pheromone matrix τ and solving the optimization problem through cooperation. The exchange of information between ants of the same colony can be extended to ants of different colonies. This approach is denominated as *parallel implementations* of ACO [10] and has been applied to problems that require considerable computational effort, such as the VRP.

The distributed optimization paradigm, as defined in Section 3, considers that each subsystem of the supply chain can be optimized through a pheromone matrix that indicates the weights on arcs connecting different nodes. The different optimization problems can be described in similar graphs, and therefore, different entities from different problems may be represented by the same nodes and arcs. In this way, it is very easy to exchange the pheromone matrix between different problems. Each colony is solving its problem autonomously taking into consideration relevant information of the colonies that are solving different problems.

Consider now the distributed optimization problem described by (5), introduced in Section 3. The symbol $\diamondsuit$ represents the way in which information is exchanged between the different systems that are being optimized. If the ACO pheromone matrix is used as the vehicle to exchange information between the different optimization problems, then it is possible to introduce the following definition:

Definition 3. The exchange of information in optimization problems (8) and (9), represented by $\diamondsuit$, is done through the pheromone matrix τ.

The literature considers that in ACO parallel implementations it is better to exchange solutions rather than pheromone matrices [10]. However, in this case, the problem being solved by each colony is not the same and a solution cannot be directly exchanged from problem to problem. Therefore, the implementation of the distributed optimization paradigm is done through the exchange of the pheromone matrix. The pheromone exchange approach has also been followed in [11], to solve the VRPWT using multiple ant colonies. The following sections describe the implementation of ACO for each of the supply chain management problems.

5.1. The supplying–logistic problem

Consider the supplying–logistic problem defined in (8):

$$\min[f_{\mathrm{SL}}] = \min[f_{M_1}]\diamondsuit \cdots \diamondsuit \min[f_{M_m}]\diamondsuit \min[f_L].$$

Consider further that suppliers M_1 up to M_m are solving the manufacturing scheduling problems $\min[f_{M_1}],\ldots,\min[f_{M_m}]$ independently, using the ACO method for local supplying optimization, presented in Section 4.2.1. The optimization procedures will result in matrices $\tau_{M_1},\ldots,\tau_{M_m}$, where each one contains indirect information about the optimization results. Note that it is expected that the highest values of τ_{M_i} are the edges describing the path of the best scheduling solution of supplier M_i.

Consider now that all suppliers at certain moments l_k in time provide the logistic system with matrices τ_{M_1} up to τ_{M_m}. The logistic system can now solve the global supplying problem using also ACO, as explained in Section 4.2.2. Recall that this optimization problem is a simple superposition of the local supplying problems. The pheromone matrix that is necessary to solve this problem can be defined as a combination of the local pheromone matrices

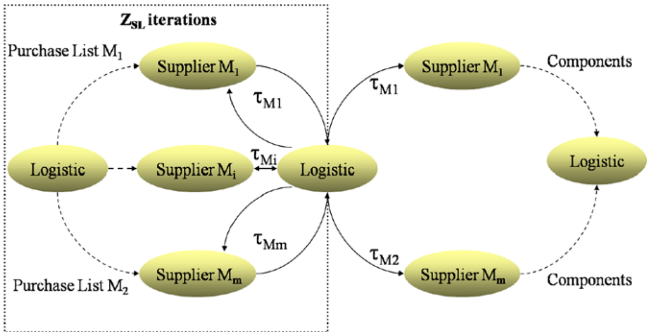

Fig. 2. Information flow between the logistic system and the supplying system.

$$\tau_L = \begin{bmatrix} \tau_{M_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tau_{M_m} \end{bmatrix}. \tag{16}$$

At the end of this optimization process, suppliers M_i receive the information from the logistic system and use it to re-optimize their own processes. The pheromone matrices used by each of the suppliers to solve the local problem are the updated sub-matrices τ_{M_i} of the pheromone matrix τ_L. This exchange of information occurs Z_{SL} times, until suppliers M_i define their last scheduling solution. Notice that the last optimization procedure is always done by the suppliers themselves, guaranteeing in this way that the final solution satisfies their own local objectives. At the end, the suppliers deliver the components to the logistic system, that starts the optimization process of the function f_L defined in (1).

Fig. 2 shows how the information is exchanged between the different systems. The continuous arrows indicated the information exchanged during the distributed optimization process, while the dotted arrows indicate the exchange of other type of information, such as component lists.

5.2. The logistic–distribution problem

Consider the logistic–distribution problem (9):

$$\min[f_{\mathrm{LD}}] = \min[f_L] \diamondsuit \min[f_D].$$

After receiving the components from the suppliers at the cross-docking center, the logistic system starts the optimization process. The solution search space is defined by the $n \in O$ orders waiting in the system to be delivered and the ACO algorithm uses the $n \times n$ matrix τ_L to search for the optimal solution of f_L. In the distribution sub-system, the solution's search space is defined by the $n \cup 0$ nodes, i.e., it is equivalent to the search space of the logistic center plus the cross-docking center 0. In this case, the ACO algorithm uses the $(n+1) \times (n+1)$ matrix τ_D. Note that they both represent a path connecting the clients, although based on different features: the tardiness of the orders and the distance between the clients.

Consider now that after the logistic system has found a scheduling solution, it provides the pheromone matrix τ_L to the distribution system. The distribution system can use this matrix to initialize its own pheromone matrix τ_D, such that

$$\tau_D = \begin{bmatrix} 0 & \mathbf{0}(1 \times n) \\ \mathbf{0}(n \times 1) & \tau_L \end{bmatrix}, \tag{17}$$

where $\mathbf{0}(1 \times n)$ is a row vector of zeros with dimension $1 \times n$, and $\mathbf{0}(n \times 1)$ is a column vector of zeros with dimension $n \times 1$. While the distribution problem is optimizing the solution, it sends the pheromone matrix to the logistic system. The logistic system can use the sub-matrix τ_L of the distribution problem matrix τ_D in order to re-optimize the logistic problem using the information provided by the distribution system. This exchange of information can occur Z_{LD} times until the logistic partner defines its last scheduling solution. Notice that the solution of the logistic system is dominant and at the end the distribution system optimizes the routing solution provided by the logistic system scheduling.

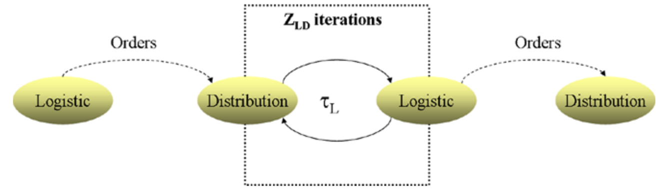

Fig. 3. Information flow between the logistic system and the distribution system.

Fig. 3 shows how the information is exchanged between the different systems. The continuous arrows indicate the information exchanged during the distributed optimization process, while the dotted arrows indicate the exchange of other type of information, such as order lists.

6. Simulation results

This section presents the management results of a multi-echelon supply chain with logistic, supplying and distribution systems, using both decentralized and the distributed optimization approach proposed by this paper. The simulation data emulates supply chain management scenarios of a real-world case of supply chain management at Fujitsu–Siemens Computers [20]. In this case, we do not consider others issues such as pricing or procurement strategy. It is assumed that the orders are placed and agreed on a daily basis with the logistic system and the SC management starts from this point on. No rolling horizon heuristics are used. All the decisions regarding supplying, and distribution are also made with the information available at that day.

The supply chain model, the local optimization algorithms and the distributed optimization methodology were implemented in Matlab 2006$^{\mathrm{TM}}$.

6.1. Supply chain instances

The instances describe a logistic system already running for 30 days, with orders already waiting to be delivered and some existing stock of components at the cross-docking center and some production orders at the suppliers. Therefore, the data describes a steady-state system. The simulation scenarios start at day $D = 31$ and concern 30 days of simulation, until $D = 60$.

The logistic problem can be described by an average number of n arriving orders at each day, following a Poisson distribution [20]. Each order is a set of 1 up to ℓdifferent types of components, and has in average $\mu(\ell)$ different types of components. Each type of component within an order can have a maximum quantity of $q_{\max}$. In this way, a logistic instance problem is defined by the 4-tuple $(n, \ell, \mu(\ell), q_{\max})$. The three different instances considered in this study, describing different sizes, complexity and disturbance degrees are: instances $(5,5,3,10)$, $(10,10,2,30)$ and $(20,10,5,20)$.

The supplying system is composed by two suppliers M_1 and M_2. Each supplier produces different types of components requested by the logistic system and therefore they are not competing with each other to supply the logistic system. When the number of components ℓ is 5, it is considered that supplier M_1 produces 3 types of components $(1,2,3)$ and supplier M_2 produces the remaining types $(4,5)$. When $\ell = 10$, each supplier produces 5 types of components. The components are produced at the suppliers in batches and the time interval between the end of the production and the arrival at the cross-docking center is considered to be null.

The distribution system has an unlimited number of trucks that can be used, but each truck as a maximum quantity of 50 components and a maximum route length of 200. All the clients are considered to be in a 100 × 100 area, where the cross-docking center is located at the geodesic point.

The distributed supply chain management approach considers that the logistic, the supplying and the distribution systems operate independently, while exchanging information in the form of a pheromone matrix during their own optimization processes. All the systems are optimized using the ACO algorithm. The management is done in two sequential steps: the *supplying–logistic problem* and the *logistic–distribution problem*, as defined in Section 3.

6.2. Performance index

The systems' performance within the supply chain using the decentralized management strategy is compared to the systems performance using the distributed management approach, as presented in Section 3. The performance index is the one presented in (4), with $\omega_1 = \omega_2 = \omega_3 = 1$, $\omega_4 = \frac{1}{10}$, and f_i is equal to f_L, f_{M_1}, f_{M_2} and f_D:

$$\mathscr{P}_{\mathrm{SCM}} = f_L + f_{M_1} + f_{M_2} + \frac{1}{10} \cdot f_D. \tag{18}$$

Observe that in this case, for the sake of simplicity, one unit of the costs f_L, f_{M_1}, f_{M_2} and f_D are equivalent to one monetary unit.

6.3. Supply chain performance

As described in Section 3, the supply chain problem can be divided into two sequential problems, the *supplying–logistic* problem and the *logistic–distribution* problem. The results analysis follows this sequence, in order to provide a deeper insight of the dynamics of the proposed management approach. At the end, we discuss the global management results.

6.3.1. Supplying-logistic optimization

The comparison between decentralized and distributed approaches for 1 month (days 31 to 60) is presented in Table 1. This table represents the cost function values for each system: f_{M_1} and f_{M_2} are the suppliers costs, and f_L is the cost function of the logistic system. It also represents the partial index

$$\mathscr{P}_{\mathrm{SL}} = f_{M_1} + f_{M_2} + f_L, \tag{19}$$

that corresponds to the part of the supply chain management performance index (18) covered by the supplying–logistic problem. Due to the stochastic nature of the ACO algorithm, the analysis considers 30 runs of each method and the results are presented in terms of mean μ and standard deviation σ values.

It is possible to observe that the mean value of the suppliers cost functions are very similar for both management methods, for all instances. This means that the suppliers performance does not change, whether the decentralized or the distributed approach is used. The standard deviation values are in general slightly higher for the distributed case, which is explained by the fact that the values of the pheromone matrix are changed during the suppliers optimization process, which introduces some disturbance to the algorithms convergence. Nevertheless, we can conclude that the suppliers performance does not change significantly when the distributed management approach is used. On the other hand, the logistic system performance significantly increases when the distributed optimization method is used, for all instances. In those cases, the mean value of the cost function is clearly lower. Further, as the standard deviations show, the worst solutions obtained with the distributed method are clearly better than the best solutions obtained using the independent approach. The partial supply chain performance index $\mathscr{P}_{ML}$ confirms that the supplying-logistic solution definitely improves, due to the improvement of the logistic system.

6.3.2. Logistic–distribution optimization

Table 2 presents the results for the three instances in terms of distribution system, using both optimization approaches: decentralized and distributed. After solving the supplying-logistic problem, we consider that the initial solution of the logistic system is, in both cases, the solution obtained with the decentralized optimization method. In this way, it is possible to compare both methods on the logistic–distribution problem case only. As performance index for this problem, we use the partial index

$$\mathscr{P}_{\mathrm{LD}} = f_L + \frac{1}{10} \cdot f_D, \tag{20}$$

that corresponds to the part of the supply chain management performance index (18) covered by the logistic–distribution problem.

As it can be seen, the results for the distributed approach are better than the ones following the separate management approach. Again, the logistic system performance is very similar for both approaches, which shows that in general the logistic system only accepts a different scheduling solution when its own performance does not decrease significantly. On the other hand, the distribution system performance increases. In fact, the worst solutions with the distributed optimization (consider the standard deviation) are in general better than the average solutions obtained with the decentralized approach.

Table 2
Distribution results at day $D = 60$ for the different instances using an independent management approach.

Instance	Management	f_L		f_D		$\mathscr{P}_{\mathrm{LD}}$	
		μ	σ	μ	σ	μ	σ
(5,5,13,10)	Decentralized	61.40	15.33	10.03	2.07	71.43	16.82
	Distributed	61.50	16.27	9.57	1.22	71.07	2.03
(10,10,2,30)	Decentralized	41.80	0.57	17.51	1.89	59.31	2.65
	Distributed	42.04	0.92	16.36	2.39	58.40	2.89
(20,10,5,20)	Decentralized	58.48	2.14	36.02	4.50	94.50	5.96
	Distributed	58.43	2.61	33.86	2.49	92.29	5.21

Table 1
Suppliers and logistic systems scheduling results for separate and distributed management approaches at day $D = 60$, for all instances.

Instance	Management	f_{M_1}		f_{M_2}		f_L		$\mathscr{P}_{\mathrm{SL}}$	
		μ	σ	μ	σ	μ	σ	μ	σ
(5,5,13,10)	Decentralized	0.20	0.01	1.37	0.21	61.40	15.33	62.97	15.45
	Distributed	0.21	0.03	1.35	0.34	49.15	1.41	50.71	1.39
(10,10,2,30)	Decentralized	2.20	0.09	9.90	0.32	41.80	0.57	53.90	0.62
	Distributed	2.50	0.12	9.50	0.73	39.56	2.71	51.56	2.57
(20,10,5,20)	Decentralized	0.90	0.31	0.00	0.00	58.48	2.14	59.38	2.25
	Distributed	1.01	0.20	0.02	0.00	53.36	2.69	54.39	2.73

Table 3
Supply chain performance for separate and distributed management approaches at day $D = 60$, for all instances.

Instance	Management	f_{M_1}		f_{M_2}		f_L		f_D		$\mathscr{P}_{\text{SCM}}$	
		μ	σ	μ	σ	μ	σ	σ	σ		
(5,5,13,10)	Decentralized	0.20	0.01	1.37	0.21	61.40	15.33	10.03	2.07	71.43	17.62
	Distributed	0.21	0.03	1.35	0.34	49.15	1.41	9.34	1.33	60.05	2.74
(10,10,2,30)	Decentralized	2.20	0.09	9.90	0.32	41.80	0.57	17.51	1.89	71.41	2.87
	Distributed	2.50	0.12	9.50	0.73	39.56	0.67	161.46	2.60	67.71	4.12
(20,10,5,20)	Decentralized	0.90	0.31	0.00	0.00	58.48	2.14	36.02	4.50	95.40	6.95
	Distributed	1.01	0.20	0.02	0.00	56.36	2.69	33.22	4.01	90.61	8.11

Since the supply chain performance index $\mathscr{P}_{\text{LD}}$ is a plain sum of the costs of both systems, the logistic–distribution solution definitely improves, due to the improvement of the distribution system.

6.4. Global supply chain results

This section presents the results of the global supply chain performance, i.e. when both supplying-logistic and logistic-distribution optimization methods are applied at the same time. The results are presented in Table 3 and compare the performance of the distributed optimization platform with the initial decentralized management supply chain performance, with all subsystems using ACO algorithms. The objective is to emphasize how much the supply chain can improve with the use of the methodology proposed in this paper.

From the results summarized in Table 3, it can be observed that:

- The supplier subsystems are not significantly affected by the use of a distributed management approach. Chronologically, the suppliers are the first system to make the scheduling decisions and, as explained in Section 3, the suppliers change their decision if the proposed solution is also optimal for them. Therefore, the purpose of the distributed supply chain management approach is not to optimize the performance of the suppliers, but solely to assure that their decision is not affected.
- The logistic system performance improves significantly when the distributed optimization is used, because it is able to influence the suppliers scheduling in such a way that diminishes the disturbances caused by tight desired delivery dates or delayed stock arrivals at the cross-docking center. On the other hand, these results are slightly worse when the distribution system proposes a different solution. Also here, the logistic decision takes place before the distribution decision. Therefore, the logistic system only accepts a different solution in cases where the logistic performance is not compromised.
- The distribution system can only influence the decision of the logistic system and it is obvious that the distributed management approach can provide an improvement to the distribution system in terms of routing costs.

The improvement in terms of the global supply chain costs $\mathscr{P}_{\text{SCM}}$ could be different (higher or lower), if the aggregation of costs considered relative weights different than one, as it was considered in this work. However, that would only mean that the relative global improvement would be different, but in any case, it definitely exists.

In conclusion, the distributed management approach is able to improve the logistic and distribution systems performance and maintain the performance of the local supplying systems, which was the main objective of the management methodology proposed in this paper.

7. Conclusions and future work

This paper proposed a new management technique for operational activities of a generic supply chain, with suppliers, logistic and distribution partners. The methodology consists of modeling each of the partners as a combinatorial benchmark optimization problem and optimizing each problem using the ant colony optimization algorithm. This algorithm uses a pheromone matrix to keep an information record during the optimization procedure. With this matrix, it is possible to exchange information between the different optimization processes running in parallel and achieve a cooperation mechanism. The exchange of information with a particular partner may bias the solutions of the remaining partners towards a different but still optimal solution, that suits better the solution of the particular partner. The simulation results showed that this strategy was able to improve the global supply chain performance. The logistic and the distribution systems improved their performance, without compromising the performance of the suppliers.

As future research work, we intend to evaluate the impact of the proposed methodology under different coordination mechanisms, such as contracts with penalty clauses or when the suppliers do not allow a decrease in their individual goal. This aspect can dealt with using fuzzy objectives, for instance [18]. We also aim the generalization of this methodology to different types of optimization algorithms, especially to other meta-heuristics.

References

[1] M. Barbuceanu, M. Fox, Coordinating multiple agents in the supply chain, in: Proceedings of the Fifth Workshops on Enabling Technology for Collaborative Enterprises, WET ICE'96, IEEE Computer Society Press, 1996, pp. 134–141.
[2] B.M. Beamon, Supply chain design and analysis: Models and methods, International Journal of Production Economics 55 (3) (1988) 281–294.
[3] D.P. Bertsekas, Distributed dynamic programming, IEEE Transactions on Automatic Control 27 (3) (1982) 610–616.
[4] B. Bullnheimer, R.F. Hartl, C. Strauss, Applying the ant system to the vehicle routing problem, in: Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization, Kluwer, 1999, pp. 109–120.
[5] G.P. Cachon, Stock wars: Inventory competition in a two-echelon supply chain multiple retailers, Operations Research 49 (5) (2001) 658–674.
[6] G.P. Cachon, S. Netessine, Game theory in supply chain analysis, in: Handbook of Quantitative Supply Chain Analysis: Modeling in the e-Business Era, Kluwer, 2004.
[7] A.J. Clark, H. Scarf, Optimal policies for multi-echelon inventory problem, Management Science 6 (4) (1960) 475–490.
[8] J.M. Cruz, Dynamics of supply chain networks with corporate social responsibility through integrated environmental decision-making, European Journal of Operational Research 184 (2008) 1005–1031.
[9] G.B. Dantzing, R.M. Ramser, The truck dispatching problem, Management Science 6 (1959) 80–91.
[10] M. Dorigo, T. Stützle, Ant Colony Optimization, The MIT Press, 2004.
[11] I. Ellabib, P. Calamai, O. Basir, Exchange strategies for multiple ant colony system, Information Sciences 177 (2007) 1248–1264.
[12] A.P. Jeuland, S.M. Shugan, Managing channel profits, Marketing Science 30 (6) (1983) 239–272.
[13] X. Li, Q. Wang, Coordination mechanisms of supply chain systems, European Journal of Operational Research 179 (2007) 1–16.
[14] M. Pinedo, Scheduling: Theory, Algorithms, and Systems, second ed., Prentice Hall, 2002.
[15] X. Qi, J.F. Bard, G. Yu, Supply chain coordination with demand disruptions, Omega – The International Journal of Management Science 32 (2004) 301–312.

[16] C. Schneeweiss, K. Zimmer, Hierarchical coordination mechanisms within the supply chain, European Journal of Operational Research 153 (2004) 687–703.
[17] W. Shen, Q. Hao, H.J. Yoon, D.H. Norrie, Applications of agent-based systems in intelligent manufacturing: An updated review, Advanced Engineering Informatics 20 (4) (2006) 415–431.
[18] C.A. Silva, J.M.C. Sousa, T.A. Runkler, Optimization of logistics systems using fuzzy weighted aggregation, Fuzzy Sets and Systems 158 (17) (2007) 1947–1960.
[19] C.A. Silva, J.M.C. Sousa, T. Runkler, J.M.G. Sá da Costa, Distributed optimisation of a logistic system and its suppliers using ant colonies, International Journal of Systems Science 37 (8) (2006) 503–512.
[20] C.A. Silva, J.M.C. Sousa, T. Runkler, R. Palm, Soft computing optimization methods applied to logistic processes, International Journal of Approximated Reasoning 40 (2005) 280–301.
[21] H. Stadler, Supply chain management and advanced planning – basics, overview and challanges, European Journal of Operational Research 163 (2005) 575–588.
[22] J.N. Tsitsiklis, D.P. Bertsekas, M. Athans, Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Transactions on Automatic Control 31 (9) (1986) 803–812.
[23] K. Weng, Modeling quantity discount under general price-sensitive demand functions: Optimal policies and relationships, European Journal of Operational Research 86 (1995) 300–314.