An agent-based Decision Support System for resources' scheduling in Emergency Supply Chains

Sarah Ben Othman, Hayfa Zgaya, Mariagrazia Dotoli, Slim Hammadi

Elsevier Editorial System(tm) for Control Engineering Practice

Manuscript Draft

Manuscript Number: CONENGPRAC-D-16-00022R2

Title: An Agent-Based Decision Support System for Resources' Scheduling in Emergency Supply Chains

Article Type: Regular article

Keywords: Emergency Supply Chain, Crisis Management, Decision Support System, Multi-Agent System, Scheduling.

Corresponding Author: Professor Mariagrazia Dotoli,

Corresponding Author's Institution: Politecnico di Bari

First Author: Sarah Ben Othman, Dr.

Order of Authors: Sarah Ben Othman, Dr.; Hayfa Zgaya, Prof.; Mariagrazia Dotoli; Slim Hammadi, Prof.

Abstract: We propose a multi-agent-based architecture for the management of Emergency Supply Chains (ESCs), in which each zone is controlled by an agent. A Decision Support System (DSS) states and solves, in a distributed way, the scheduling problem for the delivery of resources from the ESC supplying zones to the ESC crisis-affected areas. Thanks to the agents' cooperation, the DSS provides a scheduling plan that guarantees an effective response to emergencies. The approach is applied to two real cases: the Mali and the Japan crisis. Simulations are based on real data that have been validated by a team of logisticians from Airbus Defense and Space.

CONTROL ENGINEERING PRACTICE

PAPER CHECKLIST for the preparation of papers (new and revised submissions)

Please ensure that your paper conforms to the following guidelines.

1 PREFERRED FORMAT AND LAYOUT:

X **Title**: Less than 10 words
X Running headline provided

X **Authors' names**

X **Authors' affiliations/addresses**
X **Address** including fax number and email address **of corresponding author**

X **Abstract**: max. 100 words

X **Keywords**: 5–10

X **Font:** Times Roman/Times New Roman (all text, including headings and figure legends)

X **Main headings**: Upper and lower case
X Numbered (1, 2, etc.)

X **Secondary headings**: Upper and lower case
X Numbered (1.1, 1.2, etc.)

X **Tertiary headings**: Upper and lower case
X Not numbered

X **Figures**: Acceptable line quality
X Legible text (spellings, etc., checked)
X Provided within the text

X **Figure legends**: Figure legends do not have to be provided as a separate file, but shall be present in the manuscript

X **Tables**: No vertical lines
X Provided within the text

x **References**: Consistent style
x Correct usage within the text
x **All** references cited in the text
x **All** cited references listed

x **Matrices**: boldface upper case
x **Vectors**: boldface lower case
x **Other variables**: Italics
x **Units**: SI

x **Language**: Spelling checked
x Grammar checked

x **Use of first person ("I/we", etc.) strictly eliminated**

2. REFEREES' SPECIFIC REQUESTS

Not applicable (first submission)

An Agent-Based Decision Support System for Resources' Scheduling in Emergency Supply Chains

[1]Sarah Ben Othman, [2]Hayfa Zgaya, [3]Mariagrazia Dotoli, [1]Slim Hammadi

[1]CRISTAL - Research Centre in Computer Science, Signals and Automatic Systems of Lille, Ecole Centrale de Lille, Lille, France (e-mails: ben.othman.sara@hotmail.fr, slim.hammadi@ec-lille.fr)

[2]ILIS - Lille Institute of Healthcare Engineering, Université Lille 2, Lille, France (e-mail: hayfa.zgayabiau@univ-lille2.fr)

[3]Dipartimento di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Via Re David 200, 70125 Bari, Italy (e-mail: mariagrazia.dotoli@poliba.it).

Abstract: We propose a multi-agent-based architecture for the management of Emergency Supply Chains (ESCs), in which each zone is controlled by an agent. A Decision Support System (DSS) states and solves, in a distributed way, the scheduling problem for the delivery of resources from the ESC supplying zones to the ESC crisis-affected areas. Thanks to the agents' cooperation, the DSS provides a scheduling plan that guarantees an effective response to emergencies. The approach is applied to two real cases: the Mali and the Japan crisis. Simulations are based on real data that have been validated by a team of logisticians from Airbus Defense and Space.

Keywords**: Emergency Supply Chain, Crisis Management, Decision Support System, Multi-Agent System, Scheduling.**

1. Introduction

When a disaster occurs, regardless of its type (natural, such as earthquakes, floods, tsunami, etc., or man-made, e.g., wars), detailed plans should be established for people at risk, so that logistics operators are able to answer the needs of survivors in the affected areas. These operations are called emergency logistics and aim at providing the needed supplies with the minimum cost and time. Hence, emergency logistics refers to a set of interacting and coordinating logistics actors aiming at accomplishing emergency logistics requirements (Sauer, 1999). With respect to classical logistics, emergency logistics is characterized by several distinctive features. First, in the aftermath of a crisis, responsive (or automated) emergency logistics systems are needed in the affected areas for efficient disaster relief supply and recovery (Yongsong et al., 2011). Second, an assessment of supply resources and workforce should be conducted to adjust to the unexpected difficult circumstances. Third, since the crisis environment is typically uncertain, traditional centralized systems cannot deal with the sudden unexpected variations of needs (Maturana et al., 1999). Hence, the Emergency Supply Chain (ESC) objectives for crisis management are: optimal deployment of military units, resources and equipment (personnel, vehicles, planes, etc.); supply of water, food, clothing, etc.; infrastructure reconstruction; medical support.

This paper considers the first two objectives of an ESC, i.e., the optimal allocation of the resources for the supply management during the crisis. In the case of a security threat, immediate operations must be implemented, including the development and maintenance of the ESC to provide logistics support functions. The issue is to implement a suitable procurement policy to deliver the resources avoiding stock-outs that can paralyze the functioning of the whole chain. Therefore, logistics ESC managers have to take into consideration objectives like costs minimizing and constraints such as delivery delays as well as the complexity of the environment characterized by uncertainty and a large number of actors that make the scheduling task highly complex. The described context shows the relevance of

developing an automatic tool to model and optimize logistics solutions to answer emergencies and help decision-makers or authorities make the right choices in real time.

To answer the recalled needs, we propose a Decision Support System (DSS) that solves the ESC resources scheduling problem in a distributed setting. More precisely, the DSS uses an optimization and negotiation scheme to solve the resources scheduling in the areas affected by the crisis while taking into account the requirement for a distributed solution to the ESC management by an agent-based approach. The developed tool not only answers the supply needs during or after emergencies, but also supports the decision maker in scheduling resources in a varying environment. The proposed DSS is developed in cooperation with the logistics department of Airbus Defense and Space, a division of the European Aeronautic Defense and Space Company group now rebranded into Airbus Group. This paper describes the DSS agent-based architecture, where actors provide smart negotiation in order to execute and control the schedule of delivery tasks. Hence, the paper presents a set of tools and approaches for optimizing logistics flows in the ESC, detailing the optimization models used. The application of the DSS to two simulated case studies based on real data (referring to the Mali crisis and the Japan crisis) validated by military logisticians is also shown.

The remainder of the paper is structured as follows. Section 2 reviews the related literature, motivating and positioning our approach. Section 3 and 4 respectively describe the ESC structure and the agent-based DSS for the ESC management. Section 5 defines the scheduling problem and presents the optimization model for its solution. Section 6 explains the behaviour of agents. Section 7 presents the agents' communication protocols and analyses the DSS complexity. A comparison with existing alternatives is presented in Section 8. The developed software and its application to two case studies are presented in Section 9. Section 10 summarizes the paper and presents future research directions.

2. Literature Review and Paper Contribution

2.1 Literature review

The literature is rich in papers related to crisis management and ESC. The reader is referred to the surveys by Altay and Green (2006) who describe the various operational research approaches for disaster supply chain management and by Tang (2006) who presents an overview and perspectives in supply risk management.

The recent papers on ESC management include the following works. Ben-Tal et al. (2011) propose an approach to generate a robust logistics plan that can ease demand uncertainty management in humanitarian supply chains. They apply dynamic linear programming in order to assign emergency response and evacuation traffic flow problems. Sheu and Pan (2014) treat a centralized emergency supply network and propose a method involving three stage multi-objective and mixed-integer linear programming models. Kelle et al. (2014) study pre-positioning decision and response (evacuation and supply) optimization considering the resources of all stakeholders. Guojun and Caihong (2012) introduce the salvable concept for ESCs and address risk management based on urgent relief service, classifying the crisis areas based on their evaluated salvable degree. Barahona (2013) develops an optimization and simulation framework to manage the distributing relief logistics supplies in a multi-tier supply network. A dynamic fuzzy model for disaster relief response in large-scale problems is proposed by Sheu (2010). Nagurney et al. (2011) suggest a supply chain network scheme model for critical needs. Nagurney et al. (2014) study the management of a (humanitarian) disaster relief ESC and develop a network optimization model. Hale and Moberg (2005) propose a decision process for secure storage facilities efficient network supporting multiple supply chain facilities. Asghar et al. (2005) propose an approach to the design and the implementation of a dynamic integrated model for disaster management. Wang (2009) introduces a resource-constrained and decision support workflow model able to specify resource consumption and production while executing a task. Sheu (2007) proposes in his work a hybrid fuzzy clustering-optimization approach for quick response to urgent relief demand in disasters.

The recalled works show that authors typically apply to the ESC management (various different) centralized planning approaches, which are not always appropriate for a distributed chain such as an ESC. In fact, a distributed planning approach is in our opinion much more suitable, since an ESC is typically large and distributed in nature. Hence, a distributed approach allows solving in a more efficient way than a centralized approach the ESC management problem.

The literature is rich in papers dealing with classical supply chain management (Gaonkar and Viswanadham, 2007, Costantino et al. 2012, Dotoli et al, to appear), and different methods have been suggested to solve distributed problems in classical supply chains. In several models, the supply chain is represented by mathematical equations. However, the provided solutions are typically incomplete because the dynamic characteristics of the supply chain are neglected in the models. On the other hand, simulations allow the evaluation of the supply chain considering it as a single centralized actor while the related large number of entities to be modelled is the first limit of this approach (Luder et al., 2004). In various studies, distributed models are constructed and locally maintained and joined during the evaluation (Gupta et al. 2002). Moreover, in (Lee at al., 2002) the authors present a framework for distributed optimization of SC planning using an augmented Lagrangian decomposition and coordination approach.

An interesting distributed approach consists in managing supply chains using a Multi-Agent-System (MAS) framework, since MAS are distributed in nature and as such they cope well with modelling complex systems. Despite their recent emergence, MAS have found their wide application in numerous areas (Chaib-draa, 1995, Luder et

al., 2004). MAS are at the intersection of several scientific fields: distributed computing, software engineering, artificial intelligence, sociology, social psychology, and many more. Hence, not surprisingly, there are several publications on the application of MAS to the logistics industry (Marik and Lazanski, 2007, Morel et al., 2007, Shen et al., 2006). Among the first applications developed based on communicating agents, there is an application for air traffic control (Cammarata et al., 1988). In this application, MAS cooperation strategies have been used to solve conflicts between plans of a group of agents. Moreover, Nguyen et al. (1997) introduce the idea of using generating functions for evaluating the effectiveness of a defence system. They study the quantification of benefits from resource allocation for a naval task group having perfect coordination between its assets. Further, the INGENIAS Development Kit is an application based on MAS to manage a city in which a poisonous material is released, and the central services are not enough to heal all the affected people (Garcia et al., 2009). In addition, St. Germain et al. (2007) address the supply network control in a multi-agent framework.

Despite the abundance of distributed approaches to classical supply chain management, none of the recalled works addresses the peculiar case of supply chain management under emergency. In cooperation with Airbus Defence and Space, our team published several seminal works on ESC management. In (Kaddouci et al., 2009), a first version of an agent-based software tool called OBAC (Optimization Based on Agents Communication) is developed and used to improve logistics planning. Moreover, Zoghlami and Hammadi (2006) propose a linear estimation operator for an ESC. Then, Kaddoussi et al. (2012) integrate an estimator based on an ARMAX model, a model that is proven to be generic and well adapted to crisis situations. In addition, Kaddoussi et al. (2013) treat the problem of scheduling in a crisis supply chain using MAS. Simulations in this work are academic and based on theoretical examples. Subsequently, Othman et al. (2014) propose an adaptive distributed scheduling approach.

2.2 Paper contribution

This paper presents a DSS that solves in a distributed way using the MAS concept the scheduling problem for the delivery of resources to crisis-affected areas of an ESC. According to our previous works presented above, estimation and scheduling tools are integrated into an agent-based crisis management system. The various DSS decisions are the outcome of the reasoning and interactions in the proposed multi-agent system. This paper focuses on the agent-based DSS level structure, the agents' communication protocols and resource scheduling problem formulation and resolution. In previous works by our team, each zone agent uses a local Branch and Bound (B&B) behaviour to resolve the resources delivery problem in interaction with other zone-agents. In this work, instead, a zone-agent is able to choose the best algorithm in an available algorithm package, according, on one hand to the environment characteristics of the internal state of the zone, and, on the other hand, to the data size used for the resources scheduling. Comparisons among different possible algorithm selections' are made by simulation by the agent. The complexity and convergence of the approach are discussed and the contribution is compared with previous works, showing the advancement. Two real case studies obtained by military logisticians (Flichy, 2013) are also presented. With respect to previous contributions, this paper enhances the distributed agent-based architecture previously sketched in (Kaddoussi et al., 2013) as well as the resources scheduling solution through the integration of a scheduling algorithms library and gives more details about the agents' communication. In previous works, no particular attention was paid to the agents' characteristics. Moreover, here the already developed kernel-software for military logistics in cooperation with the Airbus Group (Zoghlami and Hammadi, 2006, Kaddoussi et al., 2012, Kaddoussi et al., 2013, and Othman et al., 2014) is enhanced thanks to the integration of agents. In fact, the previously realized DSS was not able to adapt to the dynamical features of the environment in the crisis areas. Finally, simulations are here made in cooperation with military staff to assess our system, using real data from the SERVAL operation in Mali (Flichy, 2013) and from the Japan crisis.

3. The Emergency Supply Chain

3.1 The ESC Objectives and Management

The ESC has an analogous objective as a commercial supply chain: satisfying the customers' demand since the onset of the crisis. The specific logistics needs are: helping victims, reconstructing a minimum infrastructure, providing food, water, medical support, etc. Attaining these goals requires the involvement of different and separate entities. Indeed, the generic ESC is composed of several dynamically and geographically distributed areas. Such areas' location is a strategic decision, taken following a crisis, and considering distance, ease of access and political stability. Each area or zone must cooperate to satisfy the ESC needs.

When emergency occurs, the optimal routing of flows (information, goods and persons) is one of the keys to success in facing the crisis. The management of the flows of a distributed logistics system such as an ESC spreads on several zones, starting from the resource's supplier and ending at the consumer. The ESC can be considered according to two different approaches. An anticipated action can be maintained, leading to treat a flow pushed from a supplier towards a customer. This flow corresponds to the delivery of a resource based on an estimated need that is not yet expressed. Otherwise, one can consider the problem in the opposite direction, from where suppliers answer the client's demand, in a so-called pulled flows approach (Wang et al., 2011). Generally, in cases of crisis both ESC push and pull modelling methods are used, with a preference for the pushed flow in the upstream areas, and for the pulled flow in the downstream ones. Indeed, a pulled flow procurement policy aims at adapting the shipments to the actual demand expressed in struck zones, in order to avoid overstocking.

However, this flow management cannot cope with the demand that does not correspond to the typically established patterns, and may present a risk in case of delay in the delivery dates. So, the pushed flows procurement policy is generally adopted in the upstream areas in order to provide autonomy to crisis areas and sustain them throughout an operation. This paper adopts such an integrated pushed flows-pulled flows approach. The boundary between the pushed flow and the pulled flow in the supply chain is an important research topic. Such a limit between the two sides is called the equilibrium point. Studying this limit for the ESC shall be of interest in future works.

3.2 The ESC Organization

The general accepted idea to effectively manage logistic flows in an ESC is to route the flows leaving from a regrouping zone via intermediate zones to reach the terminal zones (zones of distribution to the consumers). Hence, an ESC is a distributed system. Moreover, the chain is dynamical and unpredictable: its topology can be modified in real time, in order adapt to the actual situation, and to better organize the flow of resources. Of course, the independent treatment of the ESC zones can generate redundancies of information or erroneous data since every zone has incomplete information and a limited capacity to solve the problem. These limits are able to influence therefore the global behaviour of the system. For this reason, the coordination of the zones' actions proves to be a key element for the ESC effective and reliable management.

The ESC may thus be represented by the following areas in a hierarchical structure respecting military laws (Fig. 1):

- GWZ (Grouping Resources and Waiting Zone) that corresponds to the area where products must be gathered before being shipped;
- GRZ (Grouping Resources Zone) corresponding to the base area from which the resources are transported to the crisis area;
- IZ (Intermediate Zones) positioned for strategic reasons. After satisfaction of its resource requirements, the IZ sends the rest of the resources received from the GRZ to the TZ;
- TZ: Terminal Zone in direct contact with the people affected by the crisis.

This ESC model has been validated by Airbus Group: it reflects what actually happens on a military operational theatre. This ESC scheme, consisting of specialized production units, which are cooperating and geographically distributed, raises the question of the assessment of its own performance. The complexity of these multi-site organizations calls for the use of specialized tools to deal with the dynamic nature of the problem. The idea proposed in this work is the integration of a MAS structure and an optimization model into a tool capable of running operational theatre scenarios. In the next section the architecture of the proposed DSS for the ESC management is described.

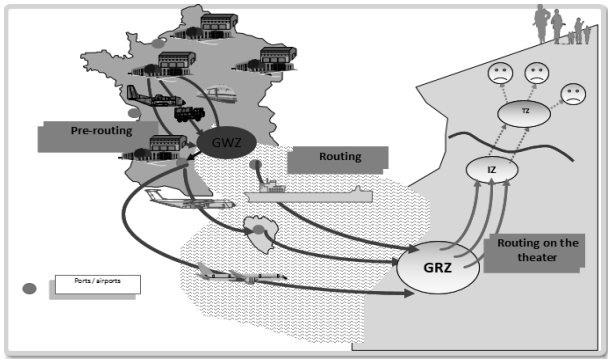

Fig. 1. An example of ESC.

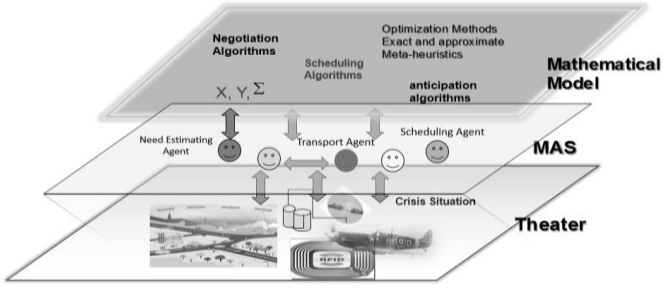

Fig. 2. The proposed three-level architecture of the DSS.

4. The DSS for ESC Management

The main idea of this research is to develop a DSS using a collaborative optimization approach, based on the alliance between optimization algorithms and multi-agent systems. To the best of the authors' knowledge, no author has previously considered such an alliance for the ESC management. In our approach, instead, we integrate polynomial algorithms in the behaviour of a set of autonomous and intelligent entities called agents. These work together in a parallel and distributed manner through interaction protocols, by forming intelligent coalitions in order to solve complex problems.

4.1 The DSS Three Level Architecture

The proposed collaborative optimization approach is implemented in a DSS with a three-level architecture (see Fig. 2). A first layer is composed of mathematical and algorithmical models (exact methods, metaheuristics, etc.) enabling to characterize the reference logistical situation. A second layer (also called agent-based DSS middle level) is based on communicating agents that make it possible to identify the real ground situation. The reference logistical situation should be attained thanks to the collaborative optimization process between agents. The middle level corresponds to the ESC model using the MAS approach. At this level, agents collaborate and negotiate in order to take decisions on estimating and scheduling strategies. A third layer represents the real logistical situation. This varies according to the field: crisis management logistics, transport logistics, hospital logistics, warehouse logistics, etc.

This architecture, innovative and general, is based on communicating agents who represent the different actors in the chain. These agents continually observe the information on the ground layer by comparing the real situation to the reference logistical situation. In function of this information, as well as diverse mathematical models (available at the first layer), these agents adapt their roles and behaviours, using polynomial optimization, in order to better react to different ground disturbances, with the aim of attaining the reference logistical situation as quickly as possible.

Agents work within the MAS and continuously receive information from the theatre of operations. Based on the perceived information and the ESC mathematical model, they adapt their behaviour to respond to the different disturbances that occur in the bottom level. It is interesting to see that the behaviour of agents may suggest different actions and decisions, including the correction and adjustment of the mathematical model. The originality of our approach consists in the fact that agents do not follow the mathematical models blindly but they permanently try to correct and adjust it according to the real data of the environment (see the discussion in Kaddouci et al, 2009). The higher level of the architecture in Fig. 2 contains optimization tools for estimation and scheduling including different mathematical models. As agents are autonomous entities characterized by decision-making capabilities, different algorithms are implemented at this level. According to the complexity of the crisis, agents evaluate the global preference of a proposal in order to identify out in real time the most effective tool to respond to the needs of the lower level, which is of course the theatre of operation. For example, to execute delivery tasks, agents may decide to go for a list algorithm that is particularly suited to the studied ESC due to its dynamic priority rules. This algorithm is characterized by its flexibility and is easy to run in real time and the problem is solved by static or dynamic priority rules. Priority is given to the task with earliest delivery date.

4.2 The DSS Agent-based middle level

This sub-section describes in detail the DSS middle level, which is based on MAS. With the development of artificial intelligence and computer networks, multi-agent technology has been an effective means to solve dynamical and distributed problems (Picco and Baldi, 1998). A multi-agent model of a problem consists in dispersing skills and knowledge into autonomous entities called agents, providing a dialogue between these entities by means of communication and interaction protocols (Woolridge and Jenning, 1995). In a MAS scheme, the change of status of certain objects in the agents' environment can affect their behaviour and decisions through perception (Barbati et al., 2012). Thus, to optimize their choices and to guide their decisions, agents can be equipped with optimization approaches suited to their skills and knowledge. They can take multi-objective decisions based on evaluation criteria. Hence, an alliance between MAS and optimization tools is proposed: these two approaches are different but perfectly complementary thanks to the distribution of the reasoning within complex DSS. In particular, given the features of MAS and the requirement of the stated problem, an agent-based distributed scheduling system is here defined. Actually, the emergency supply chain is naturally composed of autonomous entities, which can be represented by cooperative agents to solve the distributed resources assignment problem. In this context, the proposed multi-agent middle level (Fig. 3) is dynamical and considers each actor in the ESC as an autonomous entity capable of exchanging information with other actors (Kaddouci et al., 2009). In fact, in our ESC, actors are many and various. Multiple models are possible. However, they all involve different areas of the supply chain modelled through one or more agents called "Zone_Agents" (ZAs) corresponding to the ESC theatre areas previously presented in Section 3.B. In our architecture, a ZA can be detailed as a Grouping_Resources_Zone_Agent (GRZA), an Intermediate_Zone_Agent (IZA) or a Terminal_Zone_Agent (TZA). These agents are autonomous entities in a dynamic environment representing and serving each other (Adam, 2000, Luder et al., 2004, Duflos and Vanheeghe, 2000). They can also deal with the dynamic changes of the environment and reversely react. In addition to the created ZAs, software agents named Consumption_Agent (CA) and Weather_Agent (WA) are used to ensure the ESC functioning.

Once the crisis starts, the DSS creates the zone manager agents (ZAs), forming the ESC; the first one that is created is a GRZA which, depending on the situation in the affected area, creates as many IZAs as necessary. The IZAs are responsible for routing resources to the hierarchically superior zones (TZs) that is to say for delivery tasks execution. Then, the created IZAs in their turn create the managers of TZs which are the TZAs in direct contact with Need_Estimating_Agents (NEAs), one for each TZAs, whose role is to provide the estimated needed consumption in the affected areas. Each NEA is directly related to the WA in order to get information about the environment in the relative area. It is also connected to the so-called Post_Coordinator_Agent (PCA) which is responsible for resources distribution and to the CA whose role is to ensure the smooth functioning of the ESC.

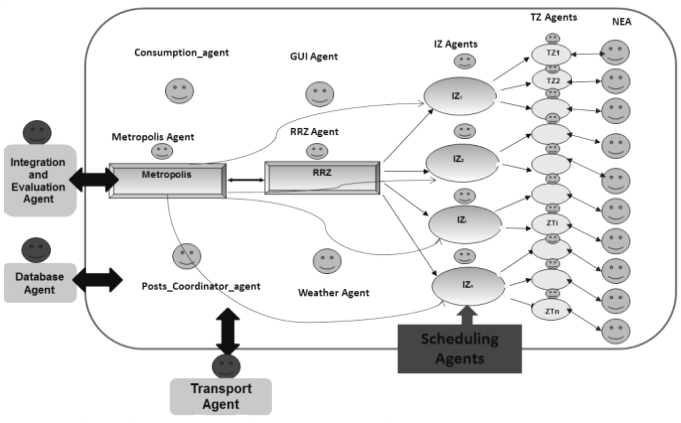

Fig. 3. The agent-based DSS middle level.

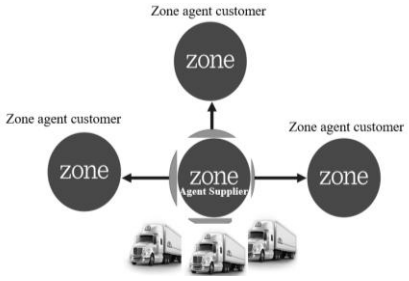

Fig. 4. Multi-customer Multi-transporter general model.

From the information provided by the WA, the NEA determines using fuzzy logic a handicap ratio, which represents the severity of climatic conditions, and corresponds to a need for additional resources. Taking into account the number of people on the theatre and the daily handicap coefficient, using fuzzy logic the estimator gets a first value of resources to deliver to the areas in need. This first quantity of resources is just a recommended value, which is not yet optimized. So, from the previous data, the NEA performs a correction of this value and then gets a second estimated one, called advocated optimized value. Once the day has passed, each TZA notices the amount of consumed resources and informs its NEA of the true value. There is also a GUI_Agent interacting with the system users.

Other agents are also generated as follows. A Transport_Agent (TA) is responsible for providing the means of transport requested by the IZA. A Database_Agent (DB_Agent) communicates with all other agents and is responsible for providing information about resources, transport means and benchmark values of performance indicators. An Integration_Evaluation_Agent (IEA) generates performance indicators after receiving information from IZAs about the distribution of resources to the terminal areas, compares the performance indicators with given reference indicators and decides if the scheduling has succeeded. If the performance indicators exceed the benchmark values, IZAs execute the rescheduling after sending a request to TA to change the allocated means of transport for scheduling. In fact, owing to the complexity and the uncertain dynamic constraints characterizing crisis management environments, rescheduling can have an important role in the success of scheduling problems resolution. Indeed, agent technology provides well suited tools for distributed problems solving that allow, thanks to communication and data exchange between agents, to review and change some previous decisions taken. Message exchanges between agents are formatted according to interaction protocols presented subsequently in Section 7.

5. The Delivery Scheduling Problem in a MAS Setting

The coordination of logistic activities between agents in the DSS for ESC management (Fig. 3) has as a main objective the delivery of the needed resources, i.e., a delivery scheduling problem. In a typical supply chain where the scheduling is managed by agents, resources are shipped to be temporary stored or arrived directly on time to zone agents representing customers. To achieve optimal operational performance, the coordination, agents' interactions and resources delivery and storage are important considerations. In this paper, the delivery scheduling problem is addressed in a MAS setting and for an ESC. Hence, the coordination of agents scheduling activities is studied, which includes storage and delivery due date management. The paper focuses on the coordination of storage of resources in each agent zone which represents a zone agent supplier or a zone agent customer, the coordination of scheduling, zone agent customer(s) orders which require the delivery from the zone agent supplier, and the storage of resources at the zone agent customer(s). This study focuses on single-zone agent supplier/multi-zone agent customer scenario and multi-transporter scenario.

5.1 The Multi-Zone Agent Customers/Transporters and Delivery Scheduling Problem

The ESC optimization problem considers the delivery scheduling decisions. In particular, a multi-zone agent customer/multi-transporter supply chain model (Fig. 4) coordinating delivery-scheduling decisions is studied. The coordination between zone agent supplier and zone agent customer for improving the performance of delivery decisions control has received a great deal of attention and the integrated approach has been recently studied. Typically, the integrated approach focuses on the resources delivery decisions of supply chain partners while minimizing the total cost of the system.

Synchronization of resources delivery scheduling is also essential for the control of global supply chain monitoring and performance, and hence for minimizing the total cost of the system. Research has focused on this area on various assumptions and objective measures that differ from the problem proposed in this paper. In fact, we focus on the coordination of delivery scheduling for the supply of multi-items of resources to more than one zone agent customer. The supplying zone agent transfers resources to one or more zone agent customers by available transporters (Fig. 4), in different delivery scenarios. The aim of this paper is, on the one hand, to determine the best way to construct an optimal scheduling for the delivery of zone agent customers' orders, and, on the other hand, to coordinate the different delivery scheduling decisions. Each zone agent has to specify the arrival time of each resource and to develop an optimizing behaviour based on an algorithmic approach able to establish solution feasibility for instances of various types and difficulty levels. Here the focus is on single zone agent supplier scheduling with multi-resources, multi-zone agent customer delivery-storage problem.

The studied problem is considered in a general way, where a set of heterogeneous transporters are available at the zone agent supplier to serve the zone agent customers. Here, each transporter could serve different zone agent customers without any pre-allotment to any zone agent customer. The general objective is to analyse the efficiency of the results of the optimal delivery scheduling in the studied scenario. In

particular, a B&B algorithm is proposed as an exact method of resolution to solve the general model, and different extensions of the general problem.

5.2 Problem Definition and Formulation

Multiple transporters are considered to deliver the resources from the zone agent supplier to the zone agent customers. Each transporter t has a specific capacity c_t, where the total number of resources cannot exceed the capacity of the vehicle used. Each round trip between the zone agent supplier and a zone agent customer requires a delivery cost as well as a delivery time dependent of the size of the batch b, the transporter t and the location of the zone agent customer c. If the resource r of a zone agent customer c is delivered before its due date d_r, a penalty of earliness, corresponding to a cost equal to β_h, is incurred, it belongs to the zone agent customer c. Sending several resources in batches is able to reduce the transportation costs. The total cost of the DSS includes delivery and storage costs. In Fig. 4 a generic multi-zone agent customer multi-transporter is sketched.

The following terminology is used in this paper.

Parameters

- ZAC: number of Intermediate Zone Agents which are suppliers or customers,
- R: number of resources,
- T: number of transporters,
- SR = {1, 2, ..., R}: set of all resources,
- SC = {1, 2, ..., ZAC}: set of all zone agent customers,
- ST = {1, 2, ..., T}: set of all transporters,
- r : index for resources $r \in SR$,
- c: index for zone agent customers, $c \in SC$,
- k: index for batches,
- $n_{r,c}$: number of resource r by zone agent customer c,
- d_r : due date of the delivery of the resource r,
- cd_r : customer destination of the resource r,
- c_t : capacity of the transporter t,
- TV_{kic}: time for the vehicle i to deliver the batch b_k of resources to zone agent customer c and to return to the supplier location,
- $Cost_{kic}$: delivery cost for vehicle i to deliver batch b_k of resources to customer c and return to the supplier,
- $Cost_{kc}$: delivery cost to deliver the batch b_k of resources to customer c,
- EP_{rc}: customer earliness penalty function for resource r delivery, which may also depend on the customer c,
- DP_{rc}: customer delay penalty function for resource r delivery, which may also depend on the customer c.

Primary variables

- $\beta^1_{r,o}$ =1 if the resource r belongs to the zone agent customer orders o, 0 otherwise,
- $\beta^2_{r,c}$ =1 if the resource r is delivered to the zone agent customer c, 0 otherwise,
- $\beta^3_{r,c,t}$ =1 if the resource r is delivered to the zone agent customer c by the transporter t, 0 otherwise,

Secondary variables

- $AT_{r,c}$: the arrival time of resource r at the zone agent customer c,
- $AB_{k,c}$: the arrival time of batch b_k at the zone agent customer i,
- EB_k: the size of batch b_k,
- EB_k=1(=0) if batch bt_k exists and $b_k \neq \varnothing$ (otherwise),
- Nb_c: the number of delivered batches for zone agent customer c.

Assumptions

The following assumptions are considered.

1. Each Intermediate Zone Agent can, at the same time, be both a supplier and a customer.
2. An Intermediate Zone Agent can be a single-supplier of multiple zone agent customers using multiple transporters available to deliver the resources to the zone agent customers to which they belong. It can be at the same time a zone agent customer for a single zone agent supplier.
3. Resources have to be ready for delivery in batches.
4. Each zone agent can adopt the Need Estimating behaviour in order to estimate individual demand of resources.
5. The round trip time of a transporter from the zone agent supplier to the zone agent customers depends on the size of the batch.
6. The storage cost S_r at the zone agent customers is dependent of the zone agent customer and the considered resources.

Objective function

The problem objective function to minimize is as follows:

$$F = \sum_{c \in SC} \left(\sum_{k=1}^{Nb_c} Cost_{kc} + \sum_{r=1}^{R} EP_{rc} * \max(0, (d_r - AT_{r,c})) + \right.$$
$$\left. + \sum_{r=1}^{R} DP_{rc} * \min((d_r - AT_{r,c}), 0) \right) \quad (1)$$

According to the notation, the three terms of the above objective function represent respectively: the delivery costs of resources, the earliness penalty and the tardiness penalty.

The general model with batch size dependent time and cost

In this paper for each zone agent we study the case of single-supplier/multi-customer with many heterogeneous transporters supplying the zone agent customers. Note that the capacity of transporters is different and the delivery time

and cost depend on the size of the batch and on the transporter. In the sequel the B&B algorithm is detailed.

5.3 The Branch and Bound Algorithm

This section describes the developed B&B algorithm to solve the problem already stated. This algorithm is incorporated into the behaviour of each zone agent and launched when the intermediate zone agent represents a zone agent supplier. When our system attributes the role of supplier to a zone agent, this latter uses a small or medium optimization problem. An exact method as B&B algorithm can be developed and gives a solution in a reasonable time.

This B&B algorithm keeps a list of sub-problems (nodes) whose combination of feasible solutions covers all feasible solutions of the original problem. The list is initialized with the original problem itself. At each iteration, the algorithm picks a current sub-problem from the list of unevaluated sub-problems. This branching appears natural, but for large problems the number of branches will be very large. Thus, if this method is used in the B&B algorithm, it can take a lot of time to find optimal solutions because redundant schedules would be checked repeatedly. However, many sub-problems would already have been removed upon the generation of nodes, since the B&B tree includes redundant solutions. At each node of the search tree, the number of resources that still needs to be delivered with each transporter to each zone agent customer has to be updated. The B&B algorithm characterizes the optimizing behaviour of the several zone agents supplier in our system. Iterations are performed till the list of sub-problems to be processed is empty. The key part of a successful B&B algorithm is the calculation of the lower bounds, which would significantly reduce the time and efforts needed for the B&B method. Based on the major feature of the problem, the lower bound value for the problem is the summation of lower bounds on the total earliness and tardiness costs and the transportation cost. It is assumed that s is a partial batch sequence solution, e(s) is the evaluation of s, and $r_c(s)$ is the number of resources remaining at the customer c for partial solution s. In each zone agent supplier, the solutions are built from the last batch to the first one for each transporter and each zone agent customer. The evaluation of the partial or complete solution is processed with backward equations. The research of a solution starts by constructing a partial solution s. Then, the rest of resources are added in order to generate a complete solution. The objective is to achieve a minimum delivery cost. Thus, the more the transporter will be loaded, the more this lower bound will be effective.

Proposition 1

For a partial solution s to problem min(F), a lower bound for the delivery cost of the remaining resources is given by:

$$\sum_{c \in SC} \left\lceil \frac{r_c(s)}{\max_{t \in ST}(c_t)} \right\rceil * \min_{t \in ST}(Cost_{0ct}) \qquad (2)$$

with b_0 the batch with the minimum size.

Proof. For each zone agent customer c, if $r_c(s)$ is the number of resources remaining to be delivered, the minimum number of round trips is equal to $\left\lceil \frac{r_c(s)}{\max_{t \in ST}(c_t)} \right\rceil$, and the delivery cost of the remaining resources is as denoted in equation (2). The solution found in equation (2) is added to the partial solution s to get the lower bound of the current node under study.

Corollary 1 The lower bound LB(s) of the partial solution s is given as follows: S_r

Once each zone agent supplier has scheduled its tasks through applying B&B algorithm to obtain a local optimal solution in a reasonable time, a global solution is therefore obtained by concatenating the local solutions found. This global solution represents the overall scheduling of the ESC.

The scheduling is interactive through agents' communication in case of perturbation in order to update the scheduling already achieved. So, the agents' interaction allows a real time monitoring of resources scheduling in the theatre thanks to agents' behaviour.

6. The Behaviour of Zone Agents

A zone-agent has to schedule resources. In previous works, a single optimization algorithm integrated in the agent behaviour was used. However, the data size and the environment characteristics change given the ESC dynamical aspect. So, a zone-agent behaviour, represented in Fig. 5, is proposed, which allows a zone to choose dynamically a scheduling algorithm according to the data size and the characteristics of the environment. In this context, it is necessary to identify and represent each agent's rational attitudes by which a zone-agent can reason and deliberate based on what it perceives from its environment. To do that, the informative, motivational and deliberative mental attitudes of the agent is identified, usually representing it by the Belief-Desire-Intention (BDI) model (Rao and Georgeff, 1995; Bruno and Célia, 2015). The behaviour of the zone agent in Fig. 5 is as follows. Initially, a zone-agent has some beliefs (B) corresponding to what it knows about itself and its environment, desires (D) corresponding to what it wants to do at long term (long term goals) and Intentions (I) corresponding to what it wants to do at short term (now). If it's a newly created zone-agent, then it holds: B= **{**disaster**}**, I=Null and D=**{**satisfy myself and my sons**}**. Hence, the following Steps are performed.

Step 1: The zone-agent perceives what happens (Perceptions: p) in its environment thanks to its sensors (weather, total number of zone-agents, etc.).

Step 2: The Beliefs revision. The information perceived updates the beliefs (B) of this agent B = brf(B, p).

Step 3: Based on its new beliefs, the agent knows, according to the opportunities, the size of the problem and the

environment characteristics. So, the Desires and Intentions are updated according to these new Beliefs: D=duf(B,D,I) and I=options(D,I). In fact, the Desires will be more specific, indicating, among other things, amounts of resources to be satisfied and priorities. Intentions correspond to steps to perform to satisfy the demand. So, if the zone-agent holds a suitable algorithm to satisfy demands (i.e., its own needs and those of subordinate areas), it has to update the settings. Otherwise, it selects the most suitable algorithm from the first and higher level (Fig. 2) according to its new information. The strategic learning module allows to the zone-agent to learn according to his previous decisions. In future work, this learning module will allow the zone-agent to update the first level by adding, modifying, or deleting the inside algorithms.

7. The Agents' Protocols and the Complexity of the DSS Middle Level

7.1 The Agents' Communication Protocols

Each protocol is represented by ACL messages exchanged between agents (FIPA). A message M exchanged between agents within an interaction protocol can be represented as follows: M<sender, receivers, performative, content>. In particular: sender is the Global Unique Identifier (GUI) of the sender of the message; receivers is the list of the GUI of all the agents, which will receive the same message simultaneously; performative is the type of the message; FIPA-ACL language propose a list of performatives: (PROPOSE, REQUEST, etc.); content is an optional parameter carrying the content of the message. This content can be a simple text message or an encapsulated object like a list, a matrix, a file, etc.

The communication protocols between agents govern the following exchanges: Resources reception; Resources consumption; Transmission of requests to the zones in charge; Sending resources to the area in need.

These four types of messages summarize the main operations that can be performed every day. In the sequel the proposed protocols are detailed, presenting the messages exchanged between agents within each phase. In the subsequent figures, blue arrows represent INFORM type messages, red ones are for REQUEST, green ones for CONFIRM and orange ones for PROPOSE.

Phase 1: Resources Reception

This first phase (Fig. 6) consists of receiving resources by the zone in need. PCA is the main actor in this phase: it is the agent that informs each zone resources reception. Here, PCA indicates to ZA the resources reception through the message DI_YOURECEIVE, specifying the content of the package and represented in blue since it is an INFORM type message. ZA indicates that it has received the parcel through a DI_OK (green since it is a CONFIRM), it updates its stock, and returns the information to GUI_Agent which can display it on the screen (with a blue message).

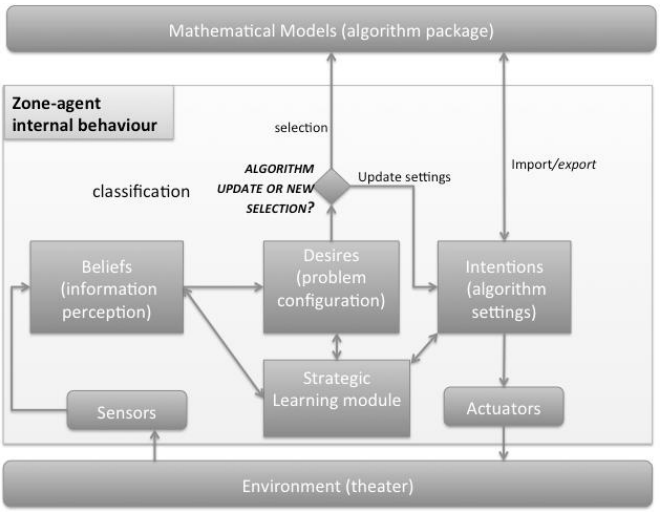

Fig. 5. The zone agent behaviour.

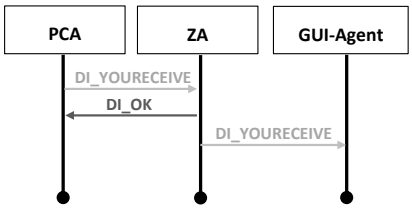

Fig. 6. The resources reception phase.

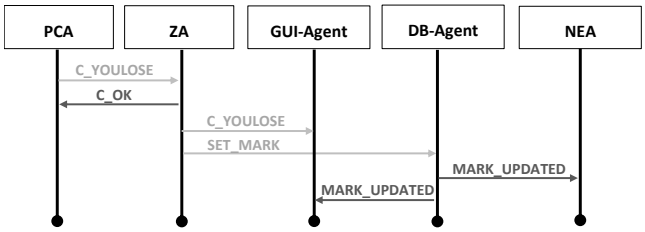

Fig. 7. The resources consumption phase.

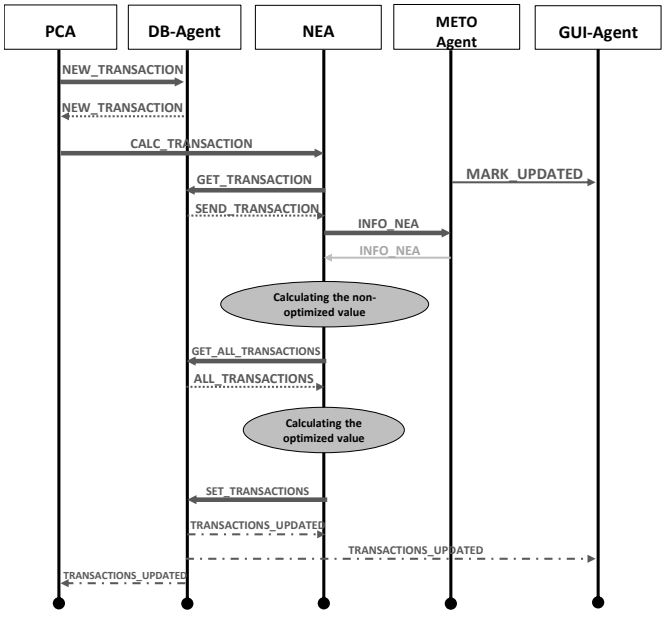

Fig. 8. The request and interaction with estimators phase.

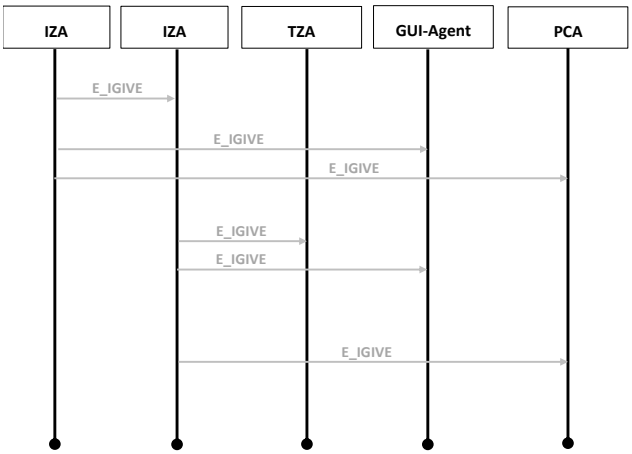

Fig. 9. The sending resources phase.

Phase 2: Resources Consumption

The second phase aims at recording the consumption of resources in each zone. This leads to a decrease in stock for all areas. CA is the main actor in this phase: it is the one who indicates the consumption recorded in each area. Communication between agents is represented in Fig. 7. Basically, the phase comprises two parts: communication and NEA calculation. First there is the communication: CA records the consumption in each zone, sends it to ZA through the C_YOULOSE message and then updates its stocks. It informs GUI_Agent, always to ensure that the information is displayed for the user on the screen. The second part involves the NEA. To function properly, the NEA needs a feedback on the actual consumption, to optimize its estimations. Thus, ZA passes the actual consumption to DB_Agent through the message DB SET_MARK. This agent records the information and sends it to the NEA. It may be noted that DB_Agent also transmits the information to GUI_Agent.

Phase 3: Transmission of requests to the zones in charge

This phase is depicted in Fig. 8. After its own consumption, each area evaluates the needed resources quantities within the next 7 days. These requirements do not meet the daily consumption. Indeed, the action here is to send a query requesting resources. Thus, if the actual stock (excluding safety stock) is not enough within 7 days, then this area sends to its TZA in control a supply request of an amount equal to the amount needed to meet its threshold replenishment. This requires to properly assess the estimates.

Phase 4: Sending resources to the area in need

This phase is depicted in Fig. 9. To respond to requests sent by lower zones, each zone makes an assessment of the resources to be sent on the same day, and what is planned to be sent in the coming days. At this stage a logistics strategy should be applied: what to send, to whom, and when. This strategy aims at defining the quantity of goods to send, based on the following parameters: assessment of all subordinate zones' requirements for each day, and the amount of stock available and stock forecast coming days.

7.2 The MAS Middle Level Complexity

The main behaviour of the zone agent is based on a scheduling algorithm. Two ways of solving scheduling problems may be considered: providing an exact solution using a known strategy, or using a heuristics that can lead to approximate solutions near to the optimal one or to the calculated lower bound. According to the chosen approach, it is easy to show that for small problems, the algorithm converges to an optimal solution. Scheduling agents have at the mathematical level of the proposed 3-level architecture a library of scheduling algorithms. Depending on the situation on the ground and the size of the problem, each scheduler chooses to apply the best suited algorithm to its environment.

Note that each zone agent makes a local decision based on its perception of the environment. The solution therefore becomes highly parallel since all agents make their calculations simultaneously. Limiting decision making at a local situation also reduces the complexity of the solution.

Because the agent is an intelligent and autonomous entity, it can evaluate its local environment regarding data size and decide which polynomial algorithm choose to schedule the delivery tasks to the corresponding zone.

7.3 Discussion on the MAS Convergence

The main issue in the use of multi-agent systems for ESC management is that it is more difficult to control the behaviour and the evolution of the overall system consisting of autonomous agents: for a given problem, it is not assured that all agents converge to a single common solution. The proposed optimization system is based on a distributed architecture with several scheduler agents which cooperate and interact through an exchange of messages, according to a decentralized approach to build a local scheduling at each agent, and so it converges at best towards a global solution that meets the requirements collectively adopted. Initially, and in a society of agents, each agent behaves according to an initial strategy, but with the passing of time, it always emerges a global phenomenon that most agents tend to follow, thus converging towards a behavioural strategy that called collective synchronization.

Note that the collaborative optimization, which is the engine of the proposed DSS, is based on the alliance between optimization algorithms and multi-agent systems. The agent interaction algorithms as well as the optimization algorithms integrated in the behaviour of the agents are polynomial. So our DSS converges and can be applied to other logistics systems (e.g., transport logistics, hospital logistics, warehouse logistics, etc) by changing the ground layer.

8. Discussion and Comparison with Existing DSS

To prove the effectiveness of our approach in comparison to the previously developed ones, the performance and computational time of the DSS are tested by comparing the results given by our approach with those given by the

approaches proposed in previous works (Zoghlami and Hammadi, 2006) and (Kaddoussi et al., 2013).

First of all, a set of test problems is randomly created. The well-known CPLEX solver is used in order to calculate the computational time solutions given by the use of various scheduling approaches. These computation times are therefore compared in order to show the effectiveness of each algorithm and its capacity to solve the studied problem. The details of the example are listed in Table 1. For each order to schedule, there are different zone agent customers, transporter capacity, quantity delivered, due date, transporter time, transporter and storage cost at each zone agent customer. Three cases {A, B and C} are considered for test. For each case, Table 1 displays the number of resources R, the number of zone agent customers ZAC, the number of transporters T, the storage cost S_r at each zone agent customer, the transporter cost $Cost_{kc}$, and the round trip time TV_{kic}. In the class (A), $Cost_{kc}$ is bigger than S_r, where $Cost_{kc}$ and S_r are generated randomly from an unvarying distribution with ranges [20; 60] and [0.2; 0.6]. The parameters for the last two cases are generated by the same way. The solution time results are listed in Table 2, illustrating the computational time for each case with 5 and 20 zone agent customers. As it can be seen, increasing the number of resources, the time of resolution increases accordingly.

Thanks to the dynamical characteristics of the 3-level architecture here proposed, in the first level three scheduling algorithms are implemented: the List algorithm, the B&B algorithm and the Simulated Annealing algorithm. Moreover, multiple agents are in the second level. Compared to previous works, this idea is original and represents our main contribution. In fact, in (Zoghlami and Hammadi, 2006), authors have used a single Glutton algorithm and in (Kaddoussi et al., 2013) a single B&B algorithm was used. It can be seen in the table that our approach has led to the better results thanks to the ability of agents to choose the most suited algorithm to the studied case regarding the environment where zone agent customers evolve. For instance, in case of small size problem one can use an exact algorithm like B&B or a metaheuristics like simulated annealing, however, when the problem becomes more complex with a bigger size, a simple use of List algorithm can lead rapidly to a better solution especially for class C. Therefore, the inconvenient of previous works is that agents do not have the choice to pick up the most adapted algorithm to their environment.

Table 1.
Characteristics of the test cases

Case	R	ZAC	T	S_r	$Cost_{kc}$	TV_{kic}
A	5	5 or 20	4 or 6	[0.2; 0.6]	[20; 60]	[6; 60]
	10	5 or 20	4 or 6			
	15	5 or 20	4 or 6			
B	5	5 or 20	4 or 6	[2; 6]	[20; 60]	[6; 60]
	10	5 or 20	4 or 6			
	15	5 or 20	4 or 6			
C	5	5 or 20	4 or 6	[20; 60]	[20; 60]	[6; 60]
	10	5 or 20	4 or 6			
	15	5 or 20	4 or 6			

Table 2
Comparison of our approach with (Zoghlami and Hammadi, 2006) and (Kaddoussi et al., 2013) for the three classes of tests.

			Class A		
		$(R)\rightarrow$	10	15	20
$(ZAC)\downarrow$			CpuT(s)	CpuT(s)	CpuT(s)
5	Using our approach	List Algorithm	2.3	5.1	26.8
		B&B algorithm	3.5	9.3	67.4
		Simulated annealing	6.7	40.7	360.5
	Using [34]	Glutton algorithm	3.4	10	72.3
	Using [36]	B&B algorithm	4.5	8.2	76.1
20	Using our approach	List Algorithm	20.5	360.7	3400.8
		B&B algorithm	170.8	1200.5	4006.9
		Simulated Annealing	2061.9	760.3	3478.4
	Using [34]	Glutton algorithm	2420.5	3453.7	>4700
	Using [36]	B&B algorithm	3283.4	>4700	>4700
			Class B		
		$(R)\rightarrow$	10	15	20
$(T)\downarrow$			CpuT(s)	CpuT(s)	CpuT(s)
4	Using our approach	List Algorithm	85.4	913.5	4412.4
		B&B algorithm	289.5	2679.5	>4700
		Simulated annealing	450.5	>4700	>4700
	Using [34]	Glutton algorithm	490	3900	>4700
	Using [36]	B&B algorithm	395.4	2896.3	>4700
6	Using our approach	List Algorithm	653.8	3400.8	>4700
		B&B algorithm	3786.4	>4700	>4700
		Simulated annealing	4200.6	>4700	>4700
	Using [34]	Glutton algorithm	3660.7	>4700	>4700
	Using [36]	B&B algorithm	3282.7	>4700	>4700
			Class C		
		$(R)\rightarrow$	10	15	20
$(T)\downarrow$			CpuT(s)	CpuT(s)	CpuT(s)
4	Using our approach	List algorithm	368.4	4200	>4700
		B&B algorithm	472.3	>4700	>4700
		Simulated annealing	738.5	>4700	>4700
	Using [34]	Glutton algorithm	897.2	>4700	>4700
	Using [36]	B&B algorithm	698.7	>4700	>4700
6	Using our approach	List algorithm	1498.5	>4700	>4700
		B&B algorithm	3620	>4700	>4700
		Simulated annealing	4200.6	>4700	>4700
	Using [34]	Glutton algorithm	>4700	>4700	>4700
	Using [36]	B&B algorithm	>4700	>4700	>4700

A final remark is that while solving the problem CPLEX sometimes runs out of memory and cannot solve the instances using the simulated annealing approach, the B&B algorithm or the Glutton algorithm in the case of more than 10 resources in Case B. In fact, using these algorithms for large sized instances, CPLEX is unable to prove optimality

before timeout. However, in the case of large instances we are always able to find good with the most suited algorithm to the crisis situation. The advantage of our approach is instead in being more adapted to the dynamical features of an ESC. In fact, the agents' concept gives the possibility to choose the most adequate algorithm to schedule resources taking into account the varying environment of intermediate zone agents.

9. The Case Studies and Simulation Results

This section first presents the upgraded version of the developed software tool and its main setting functions (originally presented in Zoghlami and Hammadi, 2006), which is then used to highlight the effectiveness of our approach through two realistic scenarios of the SERVAL operation in Mali and of a crisis in Japan (Flichy, 2013).

9.1 The DSS Software Implementation

The DSS is developed with JADE (Java Agent DEvelopment framework) platform (Greenwood, 2005). JADE allows the implementation of a MAS like ours through a middleware complying with the FIPA (Foundation for Intelligent Physical Agents) specifications and has a set of graphical tools supporting the deployment and debugging phases. JADE supports coordination between agents and provides a standard implementation of the communication through ACL messages, which facilitates the communication between agents and complies with FIPA specifications (FIPA). JADE is written in Java supports mobility, evolves rapidly and allows the integration of web services (ALP). In this paper, in addition to the developed tool, a JADE graphical tool is used, which displays in real time message exchanges between agents and is useful to debug agents' conversations. The DSS kernel includes several functionalities accessible from a main interface: registration, authentication, resources setting, zones configuration.

Registration and authentication

When the software is launched, the interfaces to register or identify a user appear just after loading interface. This step is important for information privacy in the military field.

Resources setting

After the registration and the authentication, the different logistics zones (GRZ, IZs and TZs) are set through the interfaces developed and integrated into the software (Fig. 10). The user is prompted to enter the number of IZ and the number of days of simulation, and then import an already set table containing details about the different resources to be routed to the areas in need before moving to zones configuration. This functionality allows the user to rename the resources, to define the average amount of resources consumed by a soldier in time of rest and under emergency. These values are used for estimating areas' needs.

Zones configuration

It consists in setting the number of soldiers and civilians in the different zones as well as their locations. The user provides the name of a geographic location for each zone. The information about weather and geographic position are calculated in real time then displayed on the screen.

For example, the configuration of the logistics zone in Gao (Mali), is shown in Fig. 11. The DSS communicates with a web server via the Internet. This communication takes place in two stages: 1) a first step characterized by sending a request to the web server with the location's name; 2) a second step of receiving the response to the query request by the server. This response has the form of an XML file content. It is then processed by the DSS to get weather information and geolocation (Fig. 12). The positions of the areas are displayed directly on the map.

Main interface

The main DSS interface is displayed in Fig. 13. It allows the administrator to visualize the different transactions and interactions between agents as well as the structure of the studied ESC, the evolution of resource quantities consumption in the different logistics zones, the quantities ordered to avoid stock-outs, the security thresholds and a summary of zones settings. Hence, it provides an overview of stock levels for each resource and each zone, as well as of the variation of resource quantities during delivery.

Fig. 10. Resources settings interface of the DSS.

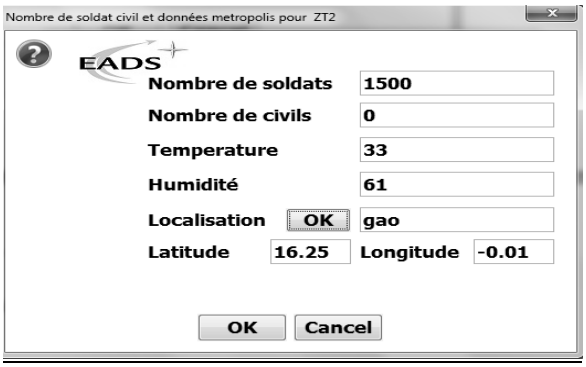

Fig. 11. Setting the number of soldiers and civilians in a TZ using the DSS.

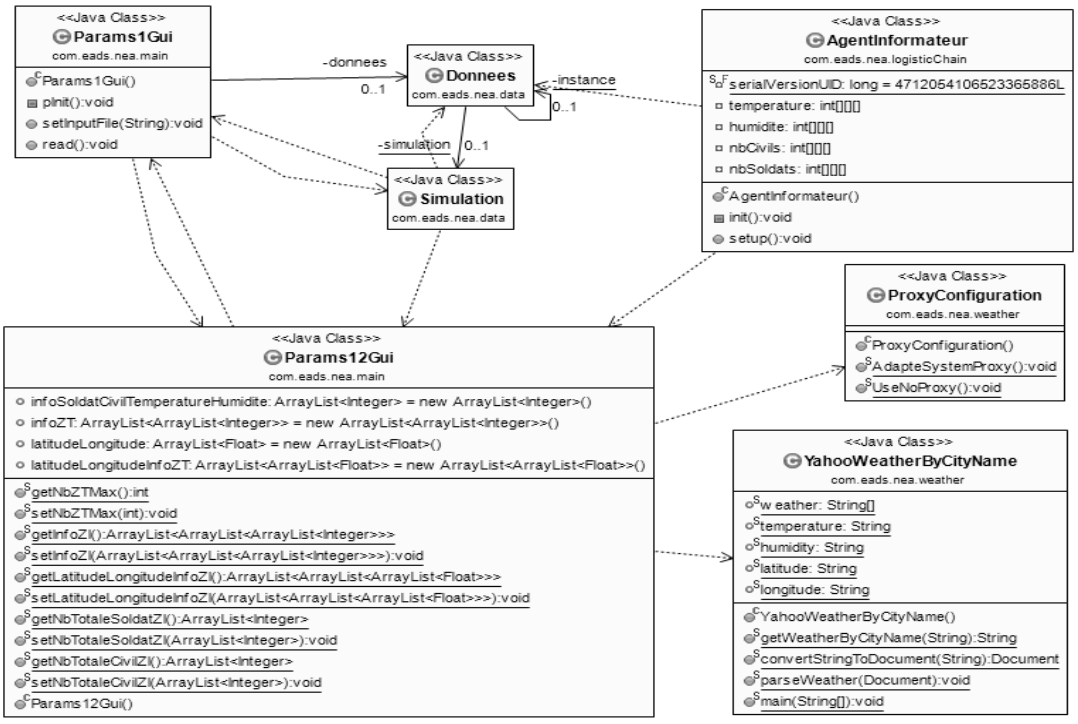

Fig. 12. Class diagram of the used case "Configuration meteorological data and geolocation".

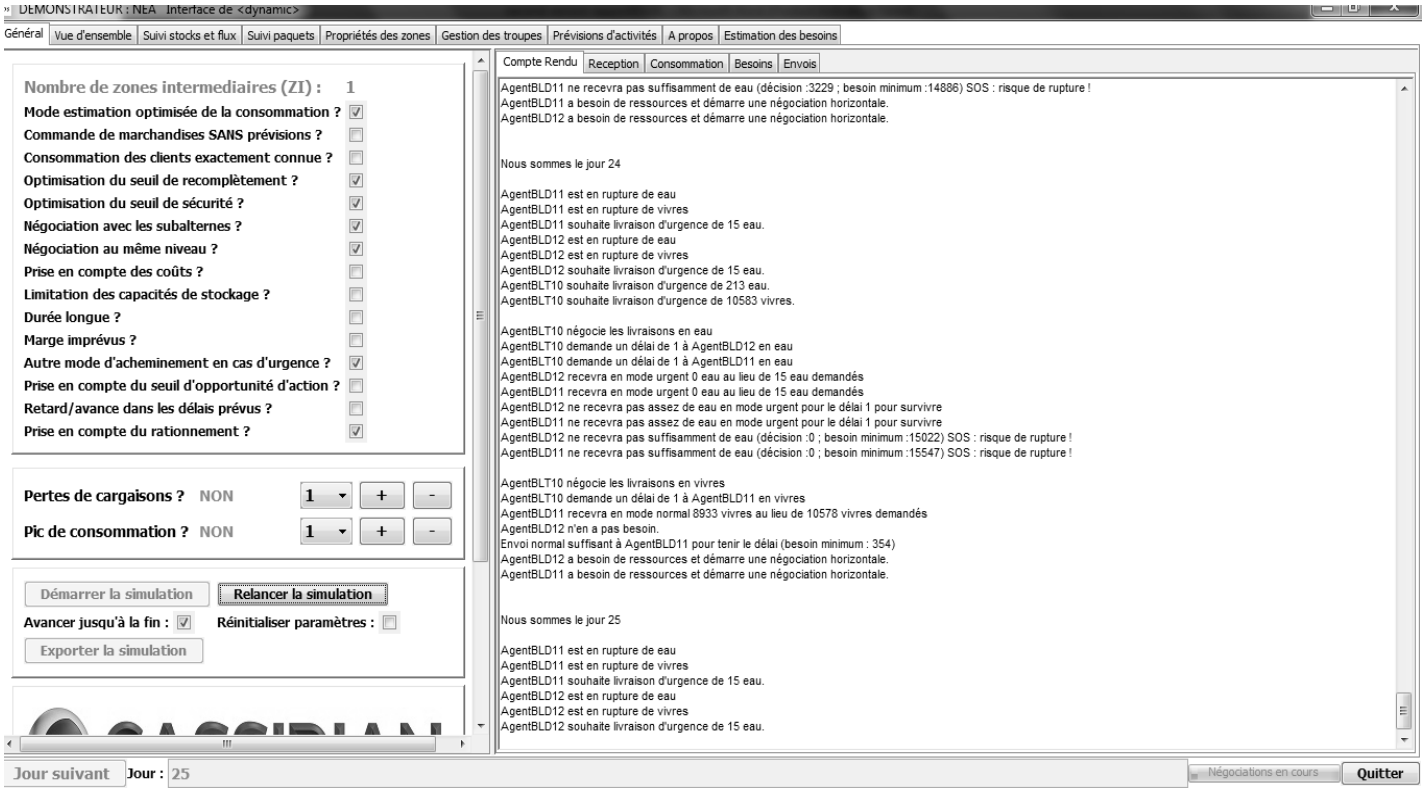

Fig. 13. Main interface of the DSS.

9.2 Case Study 1: the Mali Crisis

To evaluate the proposed methodology, the DSS is applied to an ESC, considering the scenario of a military crisis, that of Mali in the years 2013-2014. Real data related to the Mali operation conducted by the French government are collected. The case study treats this situation defining an ESC that mimics the real theatre of operation (Fig. 14). The GRZ is

defined in Bamako, the IZ is Mopti and two TZs are in Timbuktu and Gao. As part of an operation to restore peace, the considered military intervention addresses soldiers. On average, their numbers (based on real values provided by military logisticians) are as follows: 500 soldiers in the GRZ, 500 soldiers in the IZ, 1500 soldiers in the TZ1, 1500 soldiers in the TZ2. Information about these logistics areas and resources are specified in the DSS kernel software (presented in the previous sub-section), after the registration and authentication steps. A table of resources can be prepared in advance (Table 3) and may be easily imported. Note that water and food supplies are used: their variation depends on weather and is stored as historical data (see Fig. 14).

The ESC is displayed on the main DSS interface. The stocks' variation of the different logistics zones as determined by the DSS is given in the graphs in Figs. 15-17. In each graph, the evolution of stocks in each area is presented (blue curve), with the security threshold (pink curve) and the ordered amounts of resources (green lines) to meet zones needs without being out of stock. The two resources are represented as follows: water (the curves on the left) and food (the curves on the right). Moreover, the DSS provides in output the necessary resources quantities to deliver. It enhances the resources delivery by reducing stock-outs while ordering resources in an optimized way. It orders food and water before reaching resources security threshold, one of the most important parts of the stock. In addition, this tool helps absorb random consumption and its assessment depends on the calculated risk of being out-of-stock.

Note that the initial stock is designed so that there is no immediate stock-out. Consider for example the evolution of the resources stocks for TZ1 computed by the DSS in Fig. 17. The DSS optimizes the resources amounts such that they are never out of stock (except at the beginning of the case study dynamics) while ordering the minimum possible resources quantities. In fact, the system triggers ordering just when the stock gets close to the threshold but never reaches it. An analogous interpretation can be made for IZ (Fig. 16), except when the DSS estimates that it cannot accomplish the supply of TZs without being out-of-stock. It then orders the sufficient amount to hold on till the end (especially for water). Similar considerations stand for the GRZ resource quantities (Fig. 15), except when the DSS calculates that it cannot hold on using the same policy (water, day 46), and then it orders a large amount of resources allowing the survival. The results given by the DSS have been validated in cooperation with military staff.

To clarify the benefits of our DSS based on collaborative optimization scheduling, we compare our approach to the existing DSS used by Airbus Group and based on the constraints programming approach, which is a centralized optimization method. By adding the initial stocks and the routed quantities in GRZ, the amount of food delivered during the actual operation is obtained. A comparison of the delivered resources as obtained by the collaborative approach in our DSS (red curves) with results given by the Constraints Programming approach used by Airbus Group system (blue curves) is shown in Fig. 18.

Table 3
Case study ESC supplies

Supplies	Per soldier	Per soldier in case of emergency	Per civilian	Per civilian in case of emergency	Unit cost
Water	10	10	10	10	5
Rations	1	0.75	1	0.75	10

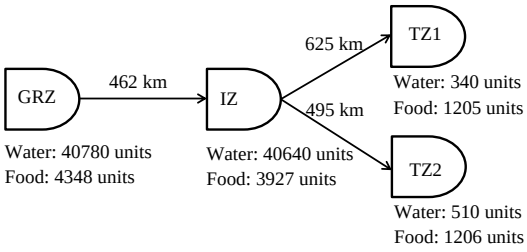

Fig. 14. Structure of the case study 1 ESC.

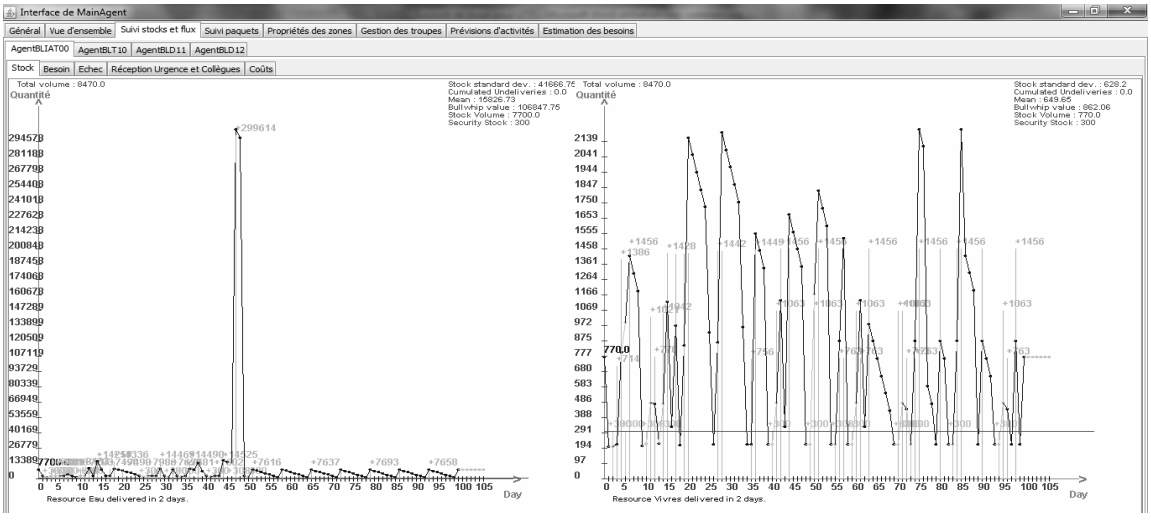

Fig. 15. Class diagram of the used case "Configuration meteorological data and geolocation".

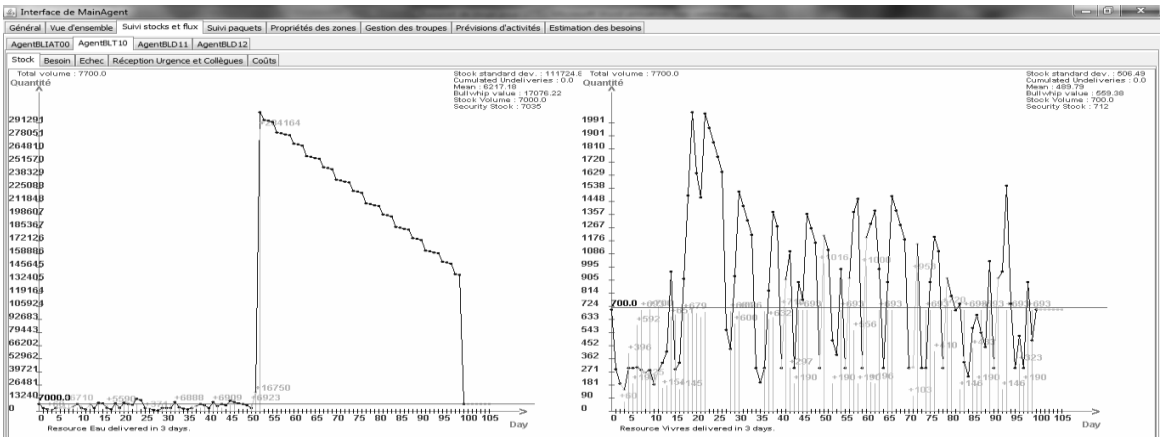

Fig. 16. Class diagram of the used case "Configuration meteorological data and geolocation".

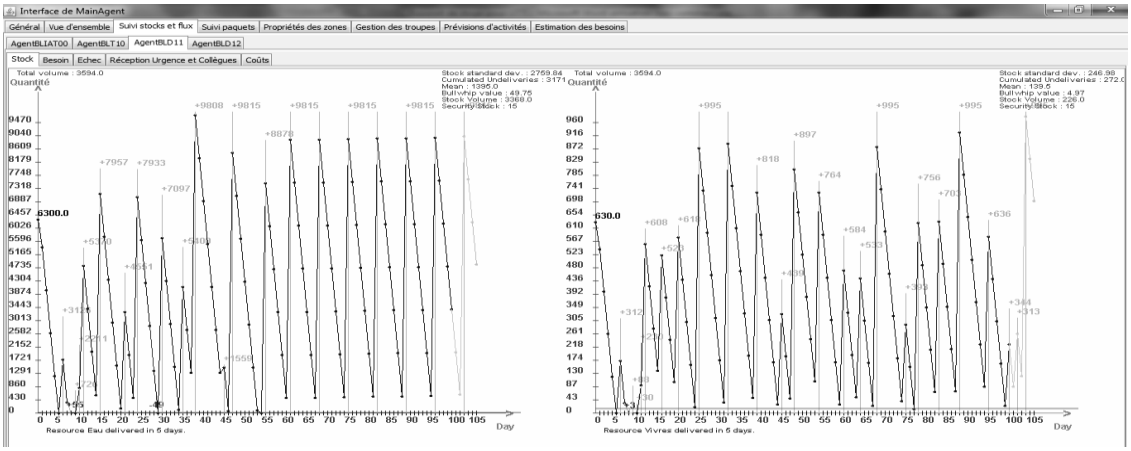

Fig. 17. Main interface of the DSS.

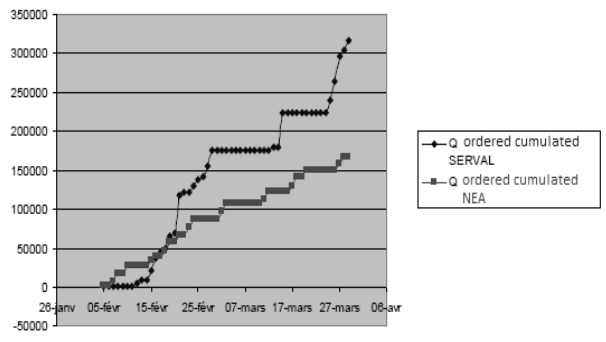

Fig. 18. Comparison with historical data.

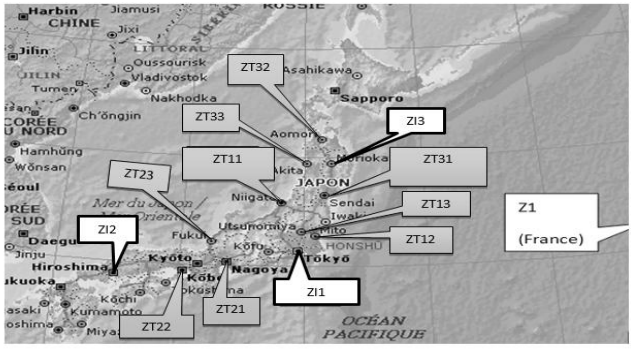

Fig. 19. Zones deployment for crisis management.

At first, the resources amounts of the actual operation are lower because there was not a total of 4000 soldiers directly in theatre. The optimizing behaviour of the DSS is highlighted when the number of soldiers increases (around February 20th: the proposed agent-based system can lead to cost delivery reduction by ordering just the sufficient resources quantities without being in shortage.

9.3 Case Study 2: the Japan Crisis

In this real application situation, we focus on the comparison between non collaborative and collaborative scheduling using different algorithms in order to show the effectiveness of the proposed agent-based architecture. This scenario lasts three weeks: Week 1: Location of emergency due to floods; Week 2: Back to a stationary state; Week 3:

New emergency situation due to nuclear risks. We consider a total of 9 affected areas, a metropolitan area and 3 intermediate zones. The onset of the crisis detected by the metropolis agent leads to the creation of 3 intermediate zones: IZ1, IZ2 and IZ3. Each intermediate zone is managed by an agent. For each intermediate zone we consider 3 TZs to supply in Japan using planes or trucks.

In this case, we use negotiation protocols between agents in a coordination strategy. Zone agents, all working together, negotiate in order to achieve a common goal which is to optimize the delivery date and to avoid stock-outs. After each negotiation cycle, these agents report back to the Integration_Evaluation_Agent that evaluates how well each scheduling agent has done, and issues new instructions accordingly. Each individual agent conducts reasoning about the results of the related local scheduling by improving its own algorithm or using another scheduling algorithm more adapted to the local crisis situation from the first level of the 3-level architecture (Fig. 2). We outline two levels of strategies that can be exercised, the individual negotiation level, and the coordination level.

Figure 21 shows a comparison between delivery costs due to the recommendation of the negotiation process to change the scheduling algorithms in order to be adapted to the crisis zone. This is the result of the negotiation and collaboration between zone agents that delivery scheduling method based on distributed cooperation between agents outperforms in terms of reduction in costs. With the use of a FIFO (First-In-First-Out) algorithm, no priority rules are taken into account. Agents therefore share information about the ESC performance and execute an effective communication algorithm to make a better decision about the choice of the scheduling approach which leads to optimal delivery costs (as explained in Section 6). Because a FIFO algorithm is not suited to solve a crisis management problem, and thanks to the characteristics of the agent, the agent chooses for this case the List algorithm which gives better results and allows reducing delivery costs as shown in Fig. 21. After 14 days of simulations, we compare the evolution of stocks obtained by enabling the collaborative optimization scheduling mode (Fig. 22), with the results obtained with a non-collaborative optimization scheduling approach (Fig. 23). We note that for the same simulation, our approach helps to smooth the evolution of stocks, avoiding falling below the safety threshold. The figures show that, with the use of the collaborative optimization, agents share information about ESC performance. Therefore, there is an effective collaboration between agents to make a better decision about the scheduling execution, which leads to optimal delivery costs.

10. Conclusions

The paper proposes a DSS for crisis management. Using an agent-based architecture and a set of optimization tools, the DSS solves the scheduling problem for the delivery of resources to the crisis-affected areas. After presenting the general DSS architecture and the developed software kernel, two real application situations are simulated and analysed: the Mali and Japan crisis. The results show the flexibility and adaptability of the system to the crisis environment. The provided tool supports logisticians in their decisions: the DSS may either be used off-line to simulate the study of an ESC and analyse a fictional crisis situation or in real time to provide a solution for a real crisis situation of the ESC. It is also to remark that, thanks to the MAS framework, the approach presented for ESC management is flexible and scalable.

Future research directions are as follows. First, a more accurate comparison between the centralised and distributed approaches will be presented, considering the threshold from which centralized solutions are not able to solve large-scale resource allocation problems. Second, the agent kernel, which selects the most adequate optimization algorithms, can be improved thanks to artificial intelligence techniques such as fuzzy logic. Third, the management of the ESC logistics flows shall be improved by developing enhanced user-friendly interfaces. Appropriate agent communication protocols implementing anticipatory mechanisms are also being developed to predict, detect and avoid undesirable situations in the supply chain. A fourth and final perspective is related to the extension of our results to other sectors, including civil logistics, thanks to the genericity of our model.

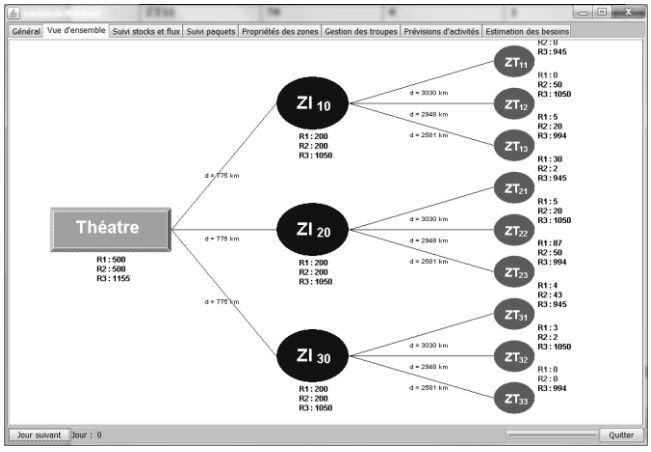

Fig. 20. Structure of the case study 2 ESC.

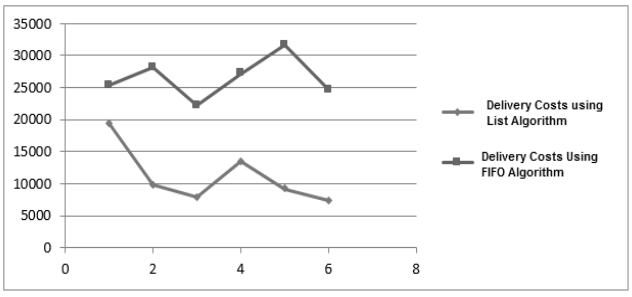

Fig. 21. Comparison of delivery costs using List and FIFO algorithms.

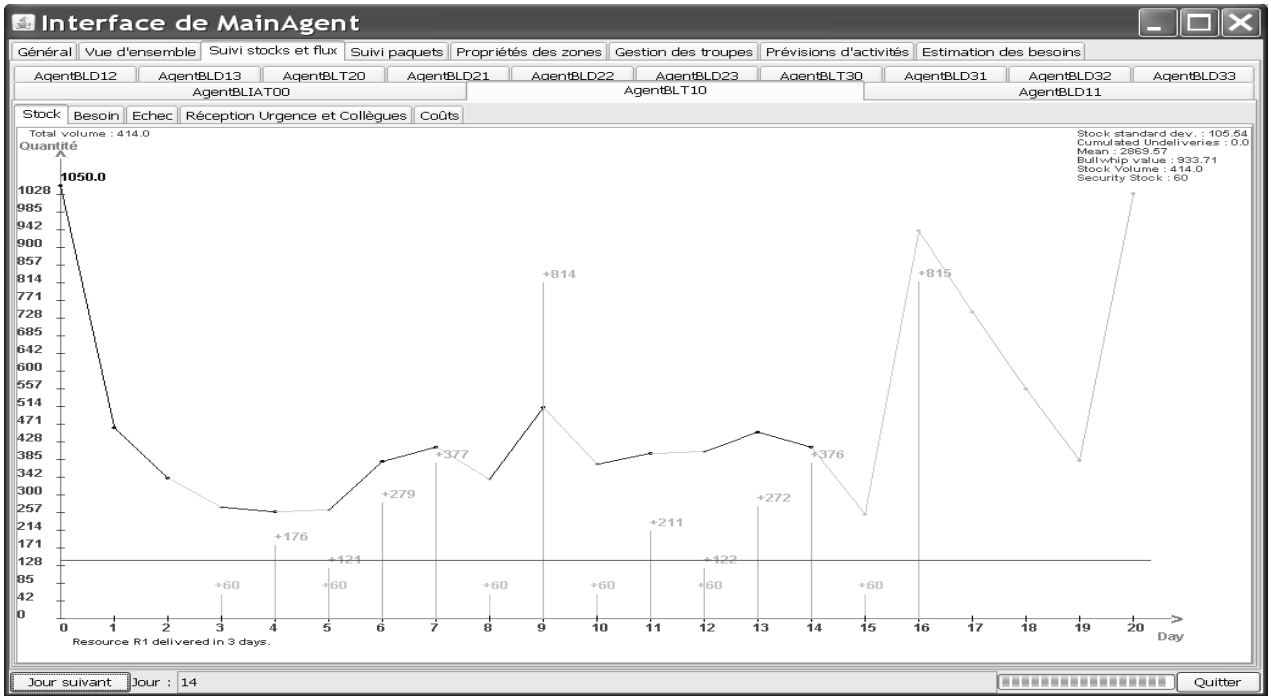

Fig. 22. Evolution of stocks using the collaborative and distributed scheduling.

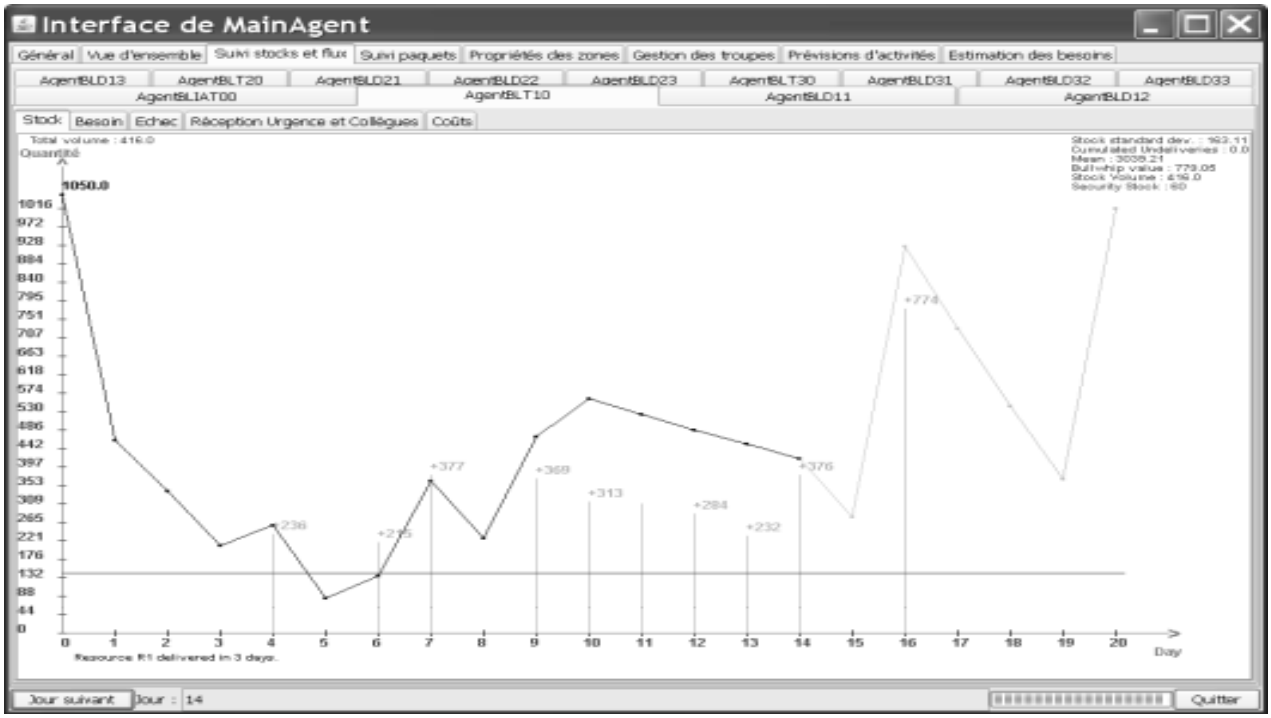

Fig. 23. Evolution of stocks using the non-collaborative optimization scheduling.

11. Acknowledgements

The authors would like to thank the logistics department of Airbus Defense and Space who provided valuable comments and ideas to the writing.

REFERENCES

E. Adam, Model of multi-agent organization to aid cooperative work in business processes: application to complex business systems, PhD Thesis, University of Valenciennes and Hainaut-Cambresis, 2000.

ALP Advanced Logistics Project Homepage, Defense Advanced Research Projects Agency. http://www.darpa.mil/iso2/alp.

N. Altay, W. Green, “OR/MS research in disaster operations management”, Eur. J. Oper. Res., Vol. 175, no. 1, pp. 475–493, 2006.
S. Asghar, D. Alahakoon and L. Churilov, Dynamic Integrated Model for Decision Support Systems, I. J. Of Simulation, Vol. 6 No 10 and 11, 2005.
F. Barahona, “Agile Logistics Simulation and Optimization for Managing Disaster Responses”, Proc. 2013 Winter Simulation Conference, 2013.
M. Barbati, G. Bruno, G. and A. Genovese, “Applications of agent-based models for optimization problems: A literature review”, Exp. Sys. Appls., vol. 39, pp. 6020-6028, 2012.
A. Ben-Tal, B. Do Chung, S. Mandala, T. Yao, “Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains”, Transp. Res. B, vol. 45 pp. 1177–1189, 2011.
W. Bruno and G. Célia, “Towards a cognitive meta-model for adaptive trust and reputation in open multi-agent systems”, Autonomous Agents and Multi-Agent Systems archive, vol 29, no. 6, pp 1125-1156, 2015
S. Cammarata, D. McArthur and R. Steeb, “Strategies of cooperation in distributed problem solving”, in Readings in Distributed Artificial Intelligence, Alan H. Bond and Les Gasser, editors, pp 102-105, Morgan Kaufmann, 1988).
B. Chaib-draa, “Industrial Applications of Distributed AI”, Communication of ACM, vol. 38, no. 11, pp. 49-53, 1995.
N. Costantino, M. Dotoli, M. Falagario, M. P. Fanti, A. M. Mangini, A model for supply management of agile manufacturing supply chains, International Journal of Production Economics, Vol. 135, No. 1, pp. 451-457, 2012.
M. Dotoli, N. Epicoco, M. Falagario, A Fuzzy Technique for Supply Chain Network Design with Quantity Discounts, International Journal of Production Research, Special Issue on “IFAC MIM-INCOM Conferences”, doi: 10.1080/00207543.2016.11784008 (to appear).
E. Duflos and P. Vanheeghe, Estimation Prédiction, Collection Sciences et Technologies dirigée par P. Borne. Edition Technip (in French), 2000.
FIPA, www.fipa.org
T. Flichy, Opération Serval au Mali. L'intervention française décryptée, Paris, Lavauzelle, 2013 (in French).
Gaonkar, R.S.; Viswanadham, N., “Analytical Framework for the Management of Risk in Supply Chains”, IEEE Trans. Aut. Sci. Eng., vol.4, no.2, pp.265,273, 2007.
I. Garcìa-Magarino, C. Gutiérrez and R. Fuentes-Fernàndez, “The INGENIAS Development Kit: a practical application for crisis-management”, Proc. 10th Int. Conf. Artif. Neur. Netw. (IWANN), 2009.
D. Greenwood, “JADE Web Service Integration Gateway (WSIG)”. Whitestein Technologies, Jade Tutorial, AAMAS, 2005.
J. Guojun and Z. Caihong, “A Study on Emergency Supply Chain and Risk Based on Urgent Relief Service in Disasters”, Sys. Engin. Proc., Vol. 5, pp. 313 – 325, 2012.
M. Gupta, H.J. Ko and H. Min, “TOC-based performance measures and five focusing steps in jobshop manufacturing environment”, Int. J. Prod. Res., Vol. 40, no.4, pp. 907-930, 2002.
T. Hale and C. R. Moberg, “Improving supply chain disaster preparedness, A decision process for secure site location”, Int. J. Phys. Distrib. Logist. Man., Vol. 35 No. 3, pp. 195-207, 2005.
A. Kaddouci, H. Zgaya, S. Hammadi and F. Bretaudeau, “PAAN: Partial Agreement Negotiation Network based on Intelligent Agents in Crisis Situation”, Int. J. Math. Comp. Sim., vol. 3, no. 4, 2009.
A. Kaddoussi, H. Zgaya, S. Hammadi, E. Duflos and P. Vanheeghe, “A need estimating agent-based tool for resources forecasting”, Proc. 16th World Multi-Conf.. Systemics, Cybernetics and Informatics: WMSCI 2012. July 17th - 20th, 2012, Orlando, Florida, USA, 2012.
A. Kaddoussi, N. Zoghlami, S. Hammadi, H. Zgaya, “An Agent-Based Distributed Scheduling For Crisis Management Supply Chain”, Int. J. Comp. Intell. Sys., vol. 6, no. 1, pp. 156-173, 2013.
P. Kelle, H. Schneider, H. Yi, “Decision alternatives between expected cost minimization and worst case scenario in emergency supply”, Int. J. Prod. Econ., vol. 157, pp. 250–260, 2014.
Y.H. Lee, S.H. Kim and C. Moon, “Production distributed planning in supply chain using a hybrid approach”, Prod. Plann. Contr., vol. 13, no. 1, pp.35-46, 2002.
A. Luder, J. Peschke, T. Sauter, S. Deter and D. Diep, “Distributed Intelligence for Plant Automation Based on Multi-Agent Systems: The PABADIS Approach”, Prod. Plann. Contr., vol. 15, no. 2, pp. 201-212, 2004.
V. Mařík, J. Lažanský, “Industrial applications of agent technologies”, Contr. Eng. Pract., vol. 15, No. 11, 2007, pp. 1364-1380.
F. Maturana, W.Shen and D. H. Norrie. Multi-agent mediator architecture for distributed manufacturing. J. Intell. Manuf., vol. 7, pp. 257-270, 1999.
G. Morel, P. Valckenaers, J.-M. Faure, C. E. Pereira, C. Diedrich, “Manufacturing plant control challenges and issues”, Contr. Eng. Pract., Vol. 15, No. 11, 2007, pp. 1321-1331.
A. Nagurney, H. Amir Masoumi and M. Yu, “An Integrated Disaster Relief Supply Chain Network Model with Time Targets and Demand Uncertainty”, Springer International, pp 287-318, 2014.
A. Nagurney, M. Yu, Q. Qiang, “Supply chain network design for critical needs with outsourcing”, Papers. Region. Sci., vol. 90, no. 1, pp. 123–143, 2011.
B. U. Nyugen, DF. Reding, and D. Nyugen, “An engagement model to optimize defense against multiple attack assuming perfect kill assessment”, Nav. Res. Logist., vol. 44, no. 7, pp. 687-697, 1997.
S. B. Othman, N. Zoghlami, H. Zgaya and S. Hammadi, “Adaptive Collaborative Agent-based System for Crisis Management”, Proc. IEEE/WIC/ACM Int. Conf. Intell. Ag. Tech., pp. 11-14, Warsaw (Poland), 2014.
G.P. Picco and M. Baldi, “Evaluating the Trade-offs of Mobile Code Design Paradigms in Network Management Applications”, Proc. 20th IEEE Int. Conf. Softw. Eng. (ICSE’97), pp. 146-155, Kyoto, Japan, 1998.
A.S. Rao and M.P. Georgeff , “BDI agents: From theory to practice, In First International Conference on Multi-Agent Systems (ICMAS-95), 1995.
J. Sauer (1999). Knowledge-Based Scheduling Techniques in Industry, CRC Press, Boca Raton.
B. Saint Germain, P. Valckenaers, P. Verstraete, Hadeli, H. Van Brussel, “A multi-agent supply network control framework”, Contr. Eng. Pract., Vol. 15, No. 11, 2007, pp. 1394-1402.
W. Shen, Q. Hao, H. Yoon and D.H. Norrie, “Applications of agent-based systems in intelligent manufacturing: An updated review”, Adv. Engin. Inform., vol. 20, no. 4, pp 415-431, 2006.
J. Sheu, An Emergency Logistics Distribution Approach for Quick Response to Urgent Relief Demand in Disasters, Transp. Res. E, 43 687–709, 2007.
J.B. Sheu, “Dynamic relief-demand management for emergency logistics operations under large-scale disasters”. Transp. Res. E, vol. 46, pp. 1–17, 2010.
J. Sheu, C. Pan, “A method for designing centralized emergency supply network to respond to large-scale natural disasters”, Transp. Res. B, vol. 67, pp. 284–305, 2014.
C.S. Tang, “Perspective in supply risk management. Review”, Int. J. Prod. Econ, Vol. 103, no. 1, pp. 451–488, 2006.
Z. Yongsong, L. Siuming and Y. Kwokkit, “Research of Crisis Management System based on CG and GIS”, Proc. IEEE Int. Conf. Comp. Sci. Aut. Eng., CSAE, pp. 596-599, 2011.
J. Wang, Emergency Response Workflow Resource Requirements Modeling and Analysis, IEEE Trans. Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 39, NO. 3, May 2009.
N. Wang, Y. Chen and L. Zhang, “Design of Multi-agent based Distributed Scheduling System for Bus Rapid Transit”, Proc. 3rd Int. Conf. Intell. Hum.-Mach. Sys Cyb., pp.111-114, 2011.
M. Woolridge and N. Jenning, “Intelligent agents: Theory and practice”, Knowl. Eng. Rev., vol. 2, no. 10, pp. 115-152, 1995.
N. Zoghlami and S. Hammadi, “Estimator Agent approach for distributed logistic chain optimization”, Proc. IEEE Int. Conf. ANIPLA’2006, pp. 13-15, Rome (Italy), 2006.

Paper No: CONENGPRAC-D-16-00022.R2
Status: Provisionally accepted
Title: An Agent-Based Decision Support System for Resources' Scheduling in Emergency Supply Chains
Authors: Sarah Ben Othman Hayfa Zgaya, Mariagrazia Dotoli, Slim Hammadi

Summary of the main changes to the paper

We thank the Editor and Reviewers for their suggestions, which guided us towards a further improvement of the paper.

All changes are signaled in red color in the revised manuscript. In the sequel we discuss in detail the changes made in response to the Editor and Reviewers.

Answers to the Editor

As the final step in the revision process, please take note of the following:
- Please attend to the reviewers' comments below and submit a list of changes or a rebuttal against each point which is being raised by the reviewers when you submit the final manuscript.
We have followed all the Editor's and Reviewers' suggestions.

- Please improve the quality of the figures! Some of the figures are quite pixelated and the text in the figures is sometimes difficult to read.
Thank you for the suggestion. We have completely redrawn figures 6, 7, 8, 9 to improve their resolution. We have removed the old figure 22 which was not legible for lack of space. We have enlarged figures 23 and 24 to make them more readable.

- Make sure that the grammar and spelling is checked.
Thank you, we have carefully checked the English.

We wish to thank the Editor for the time devoted to our manuscript and for his/her appreciation of the contribution. We have followed the Editor's and Reviewer's suggestions to further improve the paper.

Answers to Reviewer 1

The authors are doing a good job.
We deeply thank the Reviewer for supporting us with time and precious suggestions.

Nevertheless the authors should overthink and/or clearly reference their use of the term cognitive. An agent only able to act dynamically or situated cannot be named due to this as cognitive. cognition includes more, mainly the use of internal models used (in combination with external information/stimuli(for everything resulting or executed by/from the agent). from the control field ONLY MPC control can therefore be classified as cognitive or even similar approaches building or using internal models and able to work with this INTERNALLY (Perception (based on the internal model), Interpretation (based on the internal model), Planning (using the internal model), and execution = PIPE). The term cognitive is mainly defined in the field of psychology and in the last year (since 1995) also used by computer scientists and automation engineers (as technical cognition). the key definition is related to the presence of an internal model and the ability to realize cognitive

functions and procedures BASED ON THIS AND MEMORY. To sum up: the authors should use the term more careful. the term is not fixed in the title of the paper, so what. a few related references (if they think the reviewer is wrong) or a more reserved/conservative use may be helpful. yes several engineers use this term in the wrong way (because it's a hype term).

We deeply thank the Reviewer for this suggestion. Although some authors, as acknowledged by the Reviewer, actually use the term cognitive as we did in the previous version of the paper, to avoid confusion in the reader, in this novel revision we have completely removed the use of this term for the paper. We now generically speak of agent behaviour.

An additional check of formats here and there will be helpful and improve the papers quality.

Thank you, we have carefully checked the formats. In particular we have reduced to abstract as required by the journal guidelines. We have also corrected the table format.

Answers to Reviewer 5

The manuscript reports a generous amount of reasonable, sometimes even obvious, prescriptions, without a clear technical content, limited for instance to a few generical position of optimization problems then no dealt with explicitly. It is thus impossible to grasp the possible scientific or technical contribution of the authors, if any,besides generically soundness in the topic. Thus the paper has unfortunately to be rejected.

We thank the Reviewer for the time devoted to our paper. In the sequel we answer his/her specific concerns.

It would be nice to have it rewritten technically sound, but probably security reasons impose not to explain details even in the case the implementation would be technically sound.

Thank you for the comment. We have carefully revised the paper to remove unnecessary material or repetitions and include all possible details. Lack of space prevents us from inserting further material that would in any case refer to the software implementation details and as such are not the main objective of the readers of Control Engineering Practice.

It is someway disconcerting the fact the content is reported to have been developed in collaboration with Airbus group, who is not signing the paper by any of the authors.

Thank you for the comment. We have clearly reported in several parts of the paper and in particular in the acknowledgements section that the Airbus group has supported us. However, this support refers mainly to the validation phase and did not substantiate into a scientific support, which is why no one from the Airbus group is included in the paper authors' list.