



Planning and Optimization of Resources Deployment: Application to Crisis Management

Jason Mahdjoub, Francis Rousseaux

► To cite this version:

Jason Mahdjoub, Francis Rousseaux. Planning and Optimization of Resources Deployment: Application to Crisis Management. The 11th IEEE International Conference on Embedded Software and Systems, Julien Bourgeois, Frédéric Magoulès, Aug 2014, Paris, France. 10.13140/2.1.2213.0568 . hal-01085085

HAL Id: hal-01085085

<https://hal.science/hal-01085085>

Submitted on 20 Nov 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Planning and Optimization of Resources Deployment: Application to Crisis Management

Jason Mahdjoub

CRéSTIC, University of Reims, France
jason.mahdjoub@univ-reims.fr

Francis Rousseaux

CRéSTIC, University of Reims, France
francis.rousseaux@univ-reims.fr

Abstract—Crisis management challenges decision support systems designers. One problem in the decision making is to develop systems able to help the coordination of the different involved teams. Another challenge is to make the system work with a degraded communication infrastructure. Each workstation or embedded application must be able to help to make a decision with a degraded network by taking into account the potential decisions made by other agents. We propose in this article a multi-agent model, based on an ant colony optimization, and designed to manage the complexity in the deployment of resources to solve a crisis. This model is able to manage data uncertainty, and its global goal is to optimize, in a stable way, fitness functions, like saving lives, defined by multiple users. Moreover, thanks to a reflexive process, the model is able to manage the effects into the environment of its decisions, in order to take more appropriate decisions. Thanks to our transactional model, the system is also able to take into account a large data amount without exploring all potential solutions. The graphical interface should be able to make the user defining rules database. Then, if the nature of the crisis is deeply unchanged, users should be able to change rules' databases.

I. INTRODUCTION

Today, crisis management is an important domain throughout the world. It can concern earthquakes, industrial accidents, nuclear crises, etc [1]. Moreover, crises can appear as an imbrication of several emergencies, which can produce more complex crises. One issue is to manage and minimize the effects of this complexity. Our work is centered on the optimization of tasks planning and resources deployment, through an embedded application distributed into an asynchronous network.

This study refers to the project AidCrisis, co-financed by the Champagne Ardenne region and FEDER. The project aims to produce solutions for the decision making, in order to prevent mainly from nuclear, radiological, bacteriologic and chemical risks. Three kinds of aspects have been discriminated [2]. The first one consists of preparing or anticipating potential crises, through classification of circumstances, identification of critical sites, training, scripting events [3], simulation, etc. The second aspect consists of treating an effective crisis, by identifying it, deploying resources and managing the logistic, dealing with localized events, and exposing results. The third aspect consists of analyzing the crisis after its progression, into order to deduce lessons.

During the treatment step, several groups, like first aid agents, police, Doctors, government delegates must collaborate into the working site. Each group has to follow its own

organization, and its own goals, according to a categorized event. Three kinds of groups have been segmented [1], which are the management centers, the hospital centers, and the agents working on the accident area. We have noticed three kinds of problems which can obstruct the coordination of the different teams:

- 1) due to a high quantity of information or queries, decision centers or chiefs can be exceeded. Since teams have to wait for orders and can't always take initiatives, this can have as a consequence dramatic deadlocks. [1] has proposed an adaptive supervision model which enables the decision centers to detect and prevent anomalies into the rescue process.
- 2) during a crisis, a high number of victims or potential victims can be concerned. Moreover, several events can be imbricated, and several future events can appear. Decision center has to manage uncertain information, and make a decision according to an environment which can be very complex. [4] has proposed a model that organizes and summarizes experiences gained by every agent involved by the crisis management, this in order to screen these experiences at the right time.
- 3) the network used for the communication between each agent can be degraded, because of an overloaded network, or because of a material degradation of the network due to the crisis. In this case, every group involved can be isolated. Without communication, no decision can be taken, and no action can be deployed.

Considering this context, we work for the development of an embedded application able to manage three aspects. Firstly, the application must perform into several workstations/smart-phones, through a distributed and asynchronous network. Each workstation must be autonomous if the network is degraded, and must manage the potential decision of every non-connected workstation. As a consequence, every developed algorithm must work in the perspective of these constraints, in order to prevent the problem number 3. Secondly, the application must optimize the assignment of tasks and the deployment of human and material resources by regarding current events and potential future events, described as uncertain data. This aims to prevent the problem number 2. Thirdly, the application must produce an avatar for each final user, so that it can constitute a profile that describes its practices. By an emerging phenomena, this takes into account the hierarchy decisions, even if the network is degraded, or even if the decision center is exceeded. This aims to prevent the problems number 1 and 3.

We present into this article a model that covers the second aspect, centered around optimization of the tasks choice (i.e. the constitution of a strategy to apply considering a crisis) and the deployment optimization of the human and material resources. The choice of the tasks, which are defined by the user, depends on their efficiency, and the choice of the resources depends on their availability in time and space. In the section II, we relate the different algorithms of optimization and we justify our choice. According to a particular data structure articulating events, goals, tasks, and resources, we propose an adaptation of the chosen algorithm into the section III. This model is adapted for large scale applications, manages uncertainty, and by a reflexive process, adapts its decision process according to the effect of its decisions over the time. We end with a conclusion and perspectives into the section IV.

II. RELATED STUDIES

A. Critical of statistical approaches

Statistical approaches have this default that although their predictions seem in average good, they can seriously induce the human and computer deliberation to severe faults. For example, Parunak et al. [5] have demonstrated how a colony of agents, typically a prey/predator ecosystem, can prove wrong a statistical approach. Others objections have been done about more complex statistical approaches. For example, the modern portfolio theory [6], which is a well known theory of finance, was criticized for example by [7], [8]: the theory would be elegant but would consider an environment which does not correspond exactly to the reality, in particular the distribution of observed random variables. [8] proposes to prefer more stable laws like the Pareto law to deal with random variables which does not follow a normal distribution.

The interpretation of a model is often developed according to the model itself, and the only way to develop the criticism and to avoid a kind of fatalism would be to construct new models. By opposition or competition with statistical based models, our way is to develop a simulation based model designed for coordination and affectation of human and material resources. Simulation based model have the advantage to make appear empirically unlikely phenomena, where statistical approaches could consider some potentially important events as insignificant. However, both simulation and statistical based model could accumulate approximations, which should make the system producing incoherent data, then incoherent decisions. To avoid, or at least to limit this phenomena, we suppose that the introduction of the data uncertainty should force the system to have a stable reaction. Then, data should have superposed states, and used fitness functions should return several evaluations, each of them having a probability of apparition. To face up to unstable events which are a signature of crises and social interactions, and to avoid decisions which do not produce dramatic issues as seen in previous examples, we based the combination of our superposed evaluations on the Pareto law.

Simulation, optimizations, and finally decision making are done according to several goals which can be independent or correlated. We will describe now the related approaches that deal with multi-goal optimization.

B. Combinatorial optimization

1) *General approaches*: A problem of combinatorial optimization can be defined as follow: considering a set of combinations S and a fitness function $f : S \rightarrow IR$, the combinatorial optimization consists of find the combination $s \in S$ minimizing f such as: $f(s) \leq f(s_i), \forall s_i \in S$.

According the related works exposed in [9], two main types of optimization algorithms exists: complete approaches and heuristic approaches.

Complete approaches have to explore every combination contained into S . Branch and bound solutions have to explore all branches of a tree, to determine the best branch, then the best solution. Whereas dynamic programming solutions is to begin to resolve the shortest subproblems, and then to go toward problems which are increasingly complex. Complete approaches do not permit to resolve problems whose complexity class is NP. They consume too much resources because of the exploration of the entire combinations.

Heuristic approaches can be decomposed into two kinds of approaches: local search approaches and constructive approaches.

Local search approaches consist of working with current solutions, which can be chosen randomly. For a current solution, the method has to parse neighboring solutions and select the best one. The method is reiterated until no better solution was found. Then, the best returned solution is selected according to the comparison of all fixed current solutions. The main difficulty of this approach is to avoid local optimum. To do that, the Taboue algorithm [10] accepts temporarily worse solutions in its neighboring, in order to have a more deeper exploration. Simulated annealing method [11] was inspired from annealing in metallurgy. It is a statistical method which consist of virtually control the temperature of the material, i.e. the set of solutions. If the temperature is hot (or cold), particles are free (or not) and then solutions are free to move from one to another (or not). Then, the method consist of progressively decreasing the temperature, to make the system converging to the better optimums. Inspired from observations of the nature, genetic approaches [12] consist of making in competition several individuals of a population. Each individual is determined by a genetic code, itself initially determined randomly. Then according to a selection corresponding to individuals evaluations, these last are authorized to reproduce themselves. Moreover, mutations are possible during the reproduction process. Several strategies have been developed in the genetic algorithms domain, to avoid local optimum, but global optimum cannot be insured. Particle swarm optimization [13] is also an evolutionary approach, but based on a stochastic approach. The method is inspired from the observation of the movement of organisms in a bird flock or fish school. Particles adapt their speed and their directions according to the current optimal solutions, to discover new potential solutions.

Constructive approaches start with empty solutions, and construct them incrementally. Greedy algorithms aims to construct incrementally solutions until obtaining a complete solution. The choice of each element can be made randomly, or according to a heuristic, called gradient criterion. The

performance of this approach depends highly on the gradient criterion. So it is not adapted for all applications. Estimation of distribution algorithms [14] are evolutionary models based on the progressive construction of probabilities defining the quality of the choice of some elements. If the model evolves initially randomly, it gradually constructs solutions whose elements have produced good solutions in the past. Initially proposed in [15], optimizations by ant colony consist of taking advantage in the use of the potential of the intelligence emerging from a collective work of an ant colony. The ant colony optimization (ACO) consists for ants in founding the shortest path between the anthill and the nearest located food. Ant colony algorithms are well adapted for problems whose complexity class is NP. We will see how this algorithm is mathematically formalized for multi-goal optimization problems.

In most time, our application should coordinate local resources to manage a crisis. Moreover, every combination of task and every strategy do not have to use all human and material resources. But sometimes, in a severe crisis context, the system could take into account a large set of resources. For example, in a forest fire context, the system could have to call firemen which come from other regions, and sometimes from other countries. Taking account of every resource of every region could become a very complex problem. But the ant colony algorithm let show us that ants do not really explore all their environment and become near from their anthill. Moreover it is possible to limit the exploration of the ants according to the best current obtained path. For this reason, ant colony algorithm appears for us well adapted. Moreover, since this algorithm is also well adapted for routing the Internet traffic [16], it is also adapted for an embedded application encapsulated into a distributed cloud, as we are trying to develop. Then, even if a workstation does not have the entire data of the entire regions, the ants can move from a workstation to another to look for their goal. And because, they don't have to explore all their environment, the system can resolve a goal without making all workstations contributing to the problem.

III. THE PROPOSED APPROACH

A. Overview of our approach

1) *Goals*: Thanks to a rules database, the system has to generate a plan of actions and a plan of deployment of the different human and material resources to manage the crisis ongoing, its uncertainty, and its dynamical reaction to human intervention. Another challenge is to develop an ACO algorithm that do not explore all the space of solutions, when it is not necessary.

2) *Data structure*: Structurally, the input data is organized as follow. From a graphical interface, and before a crisis appears, different actors into the management center can enter into the system different kind of possible events. These events correspond to possible real events like fire, health problem, aggression, etc. Then, each of these events/problems can be specialized into sub-events. For example, fire can be specified as a house fire, a public building fire or a nuclear power station fire. Moreover, a set of events can be a superposed state. For example, during a crisis, if a user has entered that a fire has

appeared, and if he has not got time to specify if the fire concerns a public building or a house, then two events are in a superposed state with their respective probability of apparition. Each event parameter can be fuzzy. For example, the fire can be described by a parameter named 'importance', and this importance can be 'low', 'normal', or 'high'. Moreover, each event can trigger other events. For example, a fire into nuclear power station can trigger several events like 'irradiation events'. More generally, trigger functions are available for the user to trigger an event according to the evolution of the parameters of another event. For each event, the system has to solve a goal. A goal can have several fitness functions to minimize. To reach this goal, the user has to enter a graph of possible tasks to apply. For each task, a list of resources can be used. Moreover, each task can be localized or not. If it is the case, the system generates tasks of transportation of resources. Fitness functions are dependent from event parameters. The application of the main tasks alter the event parameters. Then, this has directly an effect to the results returned by fitness functions.

3) *Algorithm*: The first step of our method is to decompose according to a discrete grid, events which have superposed time position and space position, into superposed events which have unique positions.

Once we get events with unique space and time positions, the system deduces the list of goals it has to solve. For each goal, the system will have to plan the allocation of resources according to time and space, and without conflict with other goals. One solution to solve these goals is to consider them as a global multi-goal optimization problem. However, this solution will force the system to process a global optimization centralized into one server, and making the entire network dependent from this server. We propose instead of this solution a transactional model. Our solution consists of optimizing each individual goal as bubble of realities. Then, if two bubbles have conflicts of resources, a parent bubble will be generated to solve these conflicts by generating sets of constraints adapted for each possible alternative solution. Thanks to this transactional model, resources data which can be located only into some servers, workstations, or smart-phones, are explored during the goal optimization step only when no close free resource have been detected. As for the transactional memory programming [17], this method should give in most cases, better performances. When all bubbles have been optimized, and when all bubbles have any conflict with any bubbles, then the system goes to the dynamical projection step.

When a task is applied into the environment, the state of the system is likely of changing. With our data structure, the application of a task, or the application of another task will change the evolution of the events, and those of the triggered events. So deliberation of the system can alter the variables responsible of this deliberation, and then alter this deliberation. Since we do not consider global exploration (see previous paragraph), we have developed the next strategy. When all bubbles have been optimized, the system projects for each bubble their related decisions over time, i.e. for each bubble, the system deduces new events from the application of tasks into the environment. These deduced events are

merged with equivalent events which have been deduced in the previous loops. The fusion process consists of considering a same deduced event as an event which has superposed states. From this set of deduced events, is deduced a new set of bubbles, with new bubbles and/or bubbles that have changed. If changes have been detected, the system goes back to the bubble optimization step. Else, it proposes its plan of resources deployment and its plan of tasks to apply.

The figure 1 summarizes the working of our system. The sub-section III-B describes the data structure of our system. The sub-section III-C describes an ant colony optimization algorithm for multi-goal optimization problem. The sub-section III-D describes how we adapted this algorithm into the context of our application.

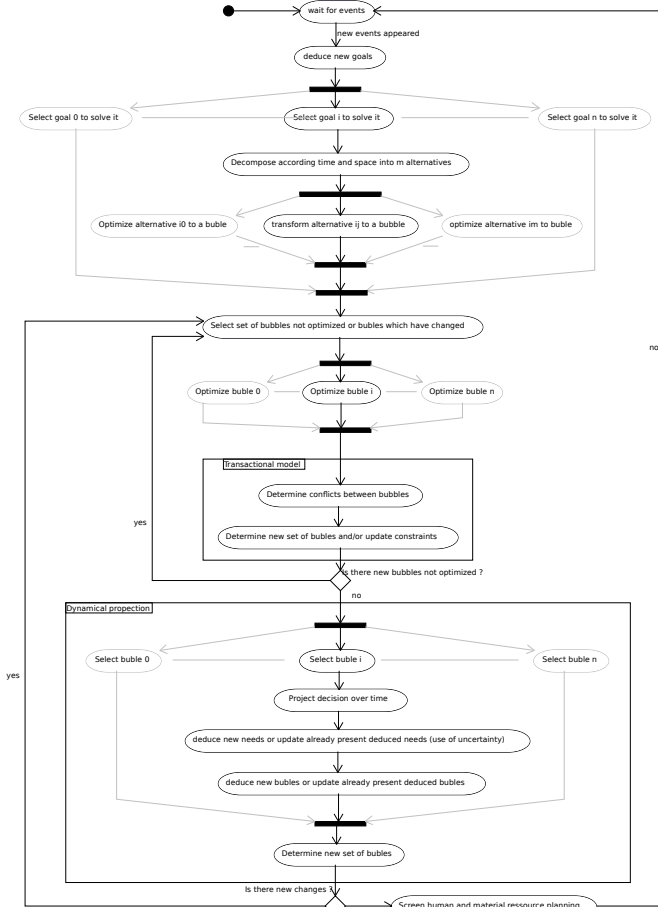


Fig. 1. Overview of our system (diagram of activities).

B. Data structure

The data structure of our model is composed of events, goals and tasks.

1) *Events*: When events are triggered, their occurrences define each property content and each probability related to each superposed state. However, an event can see some of its properties, like its position, predefined. The occurrence of an event E is noticed O_E .

There are different types of events E :

- atomic events AE : they are not reducible. Moreover, the occurrence of these events can have a location which is a superposed state:
 $|L(O_{AE})\rangle = \sum_{i=1}^{|L(O_{AE})|} \lambda_{O_{AE},i} |L(O_{AE})_i\rangle$, where $|\lambda_{O_{AE},i}|^2$ is the probability that the location $|L(O_{AE})_i\rangle$ is valid.
- list of events that must occurs at the same time which is a set SE of events, whatever their types.
- superposed events are events, whatever their types, which are exclusives. They are defined as follow
 $|SIE\rangle = \lambda_{SIE,\phi} |\phi\rangle + \sum_{i=1}^{|SIE|} \lambda_{SIE,i} |Event_i\rangle$,
 $|Event_i\rangle$ are the alternative states of the super imposed event $|SIE\rangle$, and $|\lambda_{SIE,i}|^2$ are their probability of apparition, with
 $|\lambda_{SIE,\phi}|^2 + \sum_{i=1}^{|SIE|} |\lambda_{SIE,i}|^2 = 1$ and $|\lambda_{SIE,\phi}|$ the probability that the event $|SIE\rangle$ is not an event/is nothing. The occurrence of superposed events, is a superposed occurrences of events.

Events are triggered after relative time $T(E)$, with respect to time of apparition of the parent event. The absolute time of an event is defined as follow: $TA(E) = T(E) + T(Parent(E))$, where $Parent(E) = \emptyset$ if the current event has no parent, and where $T(\emptyset) = 0$. The time $T(E)$ can be a superposed state $|T(E)\rangle$.

In addition to location and time properties, atomic events have customized properties $P(AE)$ defined by the user. Each property $|p(O_{AE}, t)_j\rangle \in P(O_{AE})$ can be a superposed state.

The function $EEP(p(O_E)_j, t)$ makes evolve event parameter $p(O_E)_j$ at relative time t . This function is defined by the user. It can be defined for example as an affine function, but it can also be defined by a simulation based model. For example, it is possible to add for a radioactive cloud, a simulation model that make evolving parameters according to time.

The function $DE(E, t)$ deduces new events from event E at time t . This deduction is dependant from the evolution of the event parameters. This function is defined by the user.

The function $|CE(E, t)\rangle$ tells if the event is closed at time t . This function depends from the evolution of the event parameters.

2) *Decomposition of atomic events*: Atomic events are decomposed into superposed events if they have superposed location in the space $|L(O_{AE})\rangle$, and/or if they have superposed time of triggering $|T(E)\rangle$. The decomposition is done through the algorithm III-B2.

3) *Goals and fitness functions*: They are defined by users according to predefined fitness functions. For each atomic event AE , a global fitness function to minimize $F_{AE}(strategy)$ is defined. This last evolves according to events parameters which are altered after applying a plan of deployment of resources. It can be composed of several local fitness functions to minimize:

$$F_{AE}(x) = (f_{AE}(x)_1, f_{AE}(x)_2, \dots, f_{AE}(x)_m) \quad m \geq 1$$

with $F = (f_1 : D \rightarrow \mathbb{R}, f_2 : D \rightarrow \mathbb{R}, \dots, f_m : D \rightarrow \mathbb{R})$

(1)

Algorithm 1 Decomposition of occurrences of events according to their superposed states of position and time.

LST_{OE} is the list of current occurrences of events.
for all $O_E \in LST_{OE}$ **do**
 if O_E is an occurrence of atomic event **then**
 if $|L(O_E)| > 1$ **or** $|T(O_E)| > 1$ **then**
 (if the event have several positions in the space or over the time)
 $|O_{SIE}\rangle \leftarrow$ new superposed event with $|L(O_E)| \times |T(O_E)|$ superposed states.
 for $i = 1$ **to** $i = |L(O_E)|$ **do**
 for $j = 1$ **to** $j = |T(O_E)|$ **do**
 $|O_{SIEi+j \times |L(O_E)|}\rangle \leftarrow O_E$ where:
 $L(|O_{SIEi+j \times |L(O_E)|}\rangle) = |L(O_E)_i\rangle$
 $T(|O_{SIEi+j \times |L(O_E)|}\rangle) = |T(O_E)_j\rangle$
 $\lambda_{O_{SIEi+j \times |L(O_E)|}} \leftarrow \lambda_{L(O_E),i} \times \lambda_{T(O_E),j}$
 end for
 end for
 Add $|O_{SIE}\rangle$ to LST_{OE} .
 Remove O_E from LST_{OE} .
 end if
 end if
end for

where m if the number of functions to optimize, and x being the vector of decision variables. In our case, x corresponds to the resulting event parameters after the application of a strategy of resources deployment.

Each of these fitness functions is classified according to an integer greater or equal than zero. The class i is optimized independently and primarily versus class $i+1$, and the class $i+1$ is optimized only when the class i was optimized, taking only remaining resources. This classification enables us to say that a life-threatening like heart attack is managed independently and primarily than a less critical problem, like an angina. F_{AE} is now defined by a set of functions, each of them defined by a class and a set of functions:

$$F_{AE}(x) = \{C_{iAE}(x)\} \text{ with } i \in C_{AE}, C_{AE} \subset \mathbb{N},$$

$$C_{iAE}(x) = (f_{C_{iAE}}(x)_1, f_{C_{iAE}}(x)_2, \dots, f_{C_{iAE}}(x)_{m_{C_i}})$$

with $m_{C_i} \geq 0$

(2)

with C_{AE} being the set of classes.

a) Superposed scores: since events parameters are superposed, fitness functions can also be a superposed state. Let $|f_{C_{iAE}}(x)_j\rangle$ being the superposed score of the effects produced by the strategy x related to the atomic event AE and the fitness function f_j (see equation 2). We say for the effects x and y that $|f_{C_{iAE}}(x)_j\rangle \prec |f_{C_{iAE}}(y)_j\rangle$, i.e. x is a better situation than y regarding the fitness function $f_{C_{iAE}}(j)$, if $\forall v \in \mathbb{R}, p(|f_{C_{iAE}}(x)_j\rangle < v) > p(|f_{C_{iAE}}(y)_j\rangle < v)$ where $p(|f_{C_{iAE}}(x)_j\rangle < v)$ is the probability that the superposed state $|f_{C_{iAE}}(x)\rangle$ gives values lower than v . This formula is directly inspired from the Pareto law, and is justified by the necessity to manage uncertainty in a stable way.

b) Domination: Domination of a strategy versus another strategy enables to know if a strategy is better from another. According our fitness functions classification, the algorithm III-B3b defines how between two strategies x and

y , we can say if x dominates y , if y dominates x , or if x and y are equivalent. If x dominates y , we notice $x \prec y$.

Algorithm 2 Relation of dominance between the effects produced by two strategies x and y .

x and y are strategies to compare
for all $i \in C_{AE}$ with C_{AE} ordered **do**
 $x_{dom} \leftarrow true$
 $y_{dom} \leftarrow true$
 for all $j \in [1, m_{C_i}]$ **do**
 if $|f_{C_{iAE}}(x)_j\rangle \prec |f_{C_{iAE}}(y)_j\rangle$ **then**
 $y_{dom} \leftarrow false$
 end if
 if $|f_{C_{iAE}}(y)_j\rangle \prec |f_{C_{iAE}}(x)_j\rangle$ **then**
 $x_{dom} \leftarrow false$
 end if
 end for
 if $x_{dom} = true$ **and** $y_{dom} = false$ **then**
 Returns “ x dominates y .”
 end if
 if $x_{dom} = false$ **and** $y_{dom} = true$ **then**
 Returns “ y dominates x .”
 end if
end for
Returns “ x and y are equivalent.”

c) Fusion: when two managed events needs the same resources, a transactional model is used to managed conflicts of resources. During this process, the fitness functions related to the two events are merged. Let F_{AE_0} being the fitness function related to the event AE_0 and F_{AE_1} being the fitness function related to event AE_1 . The merging of the two fitness functions $F_{AE_0} \cup F_{AE_1}$ is done as follow:

$$F_{AE_{0+1}} = \{C_{iAE_{0+1}}(x)\}$$

with $i \in C_{AE_{0+1}},$

$$C_{AE_{0+1}} = C_{AE_0} \cup C_{AE_1},$$

$$C_{AE_{0+1}} \subset \mathbb{N}$$

(3)

$$C_{iAE_{0+1}}(x) = (|f_{C_{iAE_0}}(x)_1\rangle, \dots, |f_{C_{iAE_0}}(x)_{m_{C_i,AE_0}}\rangle,$$

$$|f_{C_{iAE_1}}(x)_1\rangle, \dots, |f_{C_{iAE_1}}(x)_{m_{C_i,AE_1}}\rangle)$$

with $m_{C_i,AE_0} \geq 0, m_{C_i,AE_1} \geq 0$

4) Resources and tasks: A resource O_r can be a material, a human, or an autonomous material. Each resource can be organized according to types r , and each type can generalize several types. A resource must be used during the application of a task, and each resource can be used into several tasks. The percentage $U(d, r)$ defines in which proportion the task d use the resource r . A resource can be transportable or not. Moreover, a resource may need other resources to work. For example, ambulances need a driver to be used.

For every atomic event AE , a directed graph $G_{AE}(E_{AE}, V_{AE})$ of possible tasks to apply is defined by the user. This graph have a starting node SN and a ending node EN . It corresponds to a set of possible alternatives to apply during a crisis. For example, for a specific health problem, it is possible to send a doctor to treat patient, or it is possible to give instructions through

phone conversation, if no doctor is available. Each node $d_{G_{AE}} \in E_{AE}, d_{G_{AE}} \neq SN \neq EN$ represents a possible task to apply. A task need resources R to be applied. When a task is applied, the system alter event parameters according to the function $Influence(P(O_{AE}), d_{G_{AE}}, O_R)$, where $P(O_{AE})$ is the properties of the occurrence of an atomic event AE , and O_R is the set of resources used for this tasks.

The association of successive tasks, starting from the node SN and ending to the node EN , determines a plan $pl_{AE} \in PL_{AE}$ where PL_{AE} is the set of all possible plans to apply.

The task of human and material transportation need a specific implementation. The objective of our system is to optimize this task according to a specific implementation. Moreover, several alternatives can be generated. For example, a task of transportation of resources used to apply a task A can also be the same task of transportation of resources used into the next task B. The goal of our system is to automatically generate these alternatives as nodes into the graph G_{AE} . Then, for each alternative, the system looks for the best path between a location and the available resources, without exploring paths for each resource.

C. Ant colony optimization (ACO) for multi-goal optimization problems

Our system must be able to solve problems which are formalized according to several goals, i.e. several fitness functions (see section III-B3). [18] discussed several ACO approaches adapted to multi-goal problems. Experimentations have been done in [9] and it appears that the best of these approaches (m-ACO₆) is also a Pareto based approach. m-ACO₆ appears also better [9] than several evolutionary algorithms. We will present in this subsection the mathematical formalism of the m-ACO₆ approach [9], that will enable us to develop our model.

Considering that ants construct solutions through a graph $G = (V, E)$ whose definition depends on the problem to solve, pheromones are associated to each node of the graph. The algorithm 3 show how an anthill loop is repeated until optimization has found a stable state, or has reached a maximum number of loops. Pheromones are initialized to τ_{max} values. When the algorithm evolves, pheromones who evaporates along time cannot be lower than τ_{min} , with $0 < \tau_{min} < \tau_{max}$. This is done to avoid the system converging too quickly.

To construct a solution, a node v_i from G is chosen according to the probability $p(v_i)$. It is added to the solution S . Considering the constraints of the graph G , several nodes are added to S until the solution S is considered as complete. The probability to choose a node v_i is defined as follow:

$$p^c(v_i) = \frac{\tau^c(v_i)^\alpha \times \eta^c(v_i)^\beta}{\sum_{v_j \in G} \tau^c(v_j)^\alpha \times \eta^c(v_j)^\beta} \quad (4)$$

where $\tau^c(v_i)$ is the pheromone factor and $\eta^c(v_i)$ is the heuristic factor of the candidate v_i . α and β are the parameters that determine the importance of each of these factors. c corresponds to the c^{th} structure of pheromone, which corresponds the c^{th} fitness function.

Algorithm 3 Generic ACO algorithm for multi-goal optimization problems

```

initialize pheromones marks to  $\tau_{max}$ 
repeat
  processing an anthill loop...
  for all ant  $k \in$  colony do
    construct a solution
    simulating the application of this solution into the
    environment through  $Influence(P(O_{AE}), S^i, O_R)$ 
    choose randomly a fitness function to optimize
     $f_{C^i, AE_0}(x)_c$ 
    evaluating the results of this simulation thanks to
    chosen fitness function
    forgive the effects of the simulation
  end for
  for all  $c \in$  pheromones structures do
    update the  $c^{th}$  structure of pheromones
    if one mark is lower than  $\tau_{min}$ , set it to  $\tau_{min}$ 
    if one mark is higher than  $\tau_{max}$ , set it to  $\tau_{max}$ 
  end for
until A maximum number of cycles is reached or the
optimization process is stable

```

To update pheromones, when ants have constructed their solutions, each mark of pheromone is evaporated from a constant factor. Then, new pheromones are deposited on elements of G constituting the best solutions:

$$\tau^c(v_i) \leftarrow (1 - \rho) \times \tau^c(v_i) + \Delta\tau^c(v_i) \quad (5)$$

where ρ is the factor of evaporation such as $0 \leq \rho \leq 1$. $\Delta\tau^c(v_i)$ is the quantity of deposited pheromones for the component v_i . Let S^c the solution of the colony that maximizes the c^{th} goal, in our case defined by $f_{C^i, AE_0}(x)_c$ (see section III-B3), let S_{best}^c the solution that maximizes the c^{th} goal from the beginning of the optimization process, and let S_p being the set of non-dominated solutions produced during the anthill loop. The quantity of pheromones deposited for each element $v_i \in S^c$, is defined as follow:

$$\Delta\tau^c(v_i) = \begin{cases} \frac{1}{1 + f_{C^i, AE}(S^c)_c - f_{C^i, AE}(S_{best}^c)_c} & \text{if } v_i \in S^c, \\ 1 & \text{if } v_i \in S_p, \\ 0 & \text{else.} \end{cases}$$

D. Optimization

When each atomic event has been decomposed according to time and space (see section III-A and section III-B2), each atomic event is transformed into a bubble of reality where optimization are done according to the limited context of the event. When each bubble is optimized, conflicts of resources are analyzed between bubbles and if necessary, bubbles are merged according to the transactional model described into the section III-D2.

1) *Bubble optimization*: Two kinds of bubbles exists. Leaf bubbles represent atomic events, whereas node bubbles represent peer of bubbles. Peer of bubbles are generated by the transactional model when resources conflicts are detected. Node bubbles contains peer of constraints related to each sub-bubble.

Leaf bubbles are optimized according to the generic algorithm 3. The construction of a solution by an ant, which is a part of the previous enumerated algorithm, is described by the algorithm 4. The leaf bubbles have to produce a plan pl_{AE} of tasks to apply to solve the problem related to the atomic event AE . This plan results from the optimization of series of possible tasks represented by the graph $G_{AE}(E_{AE}, V_{AE})$ (see section III-B4). Some of these possible tasks consists of optimizing a topographic path by choosing the nearest resources from next chosen path. The construction of this topographic path follow the generic algorithm 3, by constructing solutions defined by the algorithm 5. During the path construction, if the path becomes too long, regarding the best already got path, then the path construction is restarted until a sufficiently short path is got. The path is considered too long if $\frac{f(Path_{best})}{f(Path)} < \mu$, where μ is an empirically determined threshold, and $f(x)$ is the value given by a path, finished or not.

Algorithm 4 Constructing a solution for one ant associated to a leaf bubble

Mark A
constructing a solution where S_k is the set of elements of this solution:
choose randomly a function $f_{AE}(x)_j$ to optimize.
 $CurrentNode \leftarrow SN$, where SN is the start node of the graph G_{AE} corresponding the atomic event AE .
repeat
 choose a task $d_{G_{AE},i}$ from $CurrentNode$ with probability $p^j(d_{G_{AE},j})$
 if $d_{G_{AE},i} \neq EN$ **then**
 if $d_{G_{AE},i}$ is a transport mission **then**
 Optimize this task (see algorithm 5).
 $EEP(p(O_E)_j, TimeEnd_{d_i})$, where $time_{d_i}$ is the time corresponding the end of the task $d_{G_{AE},i}$
 end if
 add $d_{G_{AE},i}$ to S_k
 $Influence(P(O_{AE}), d_{G_{AE}}, O_R)$ (see section III-B4).
 $CurrentNode \leftarrow d_{G_{AE},i}$
 end if
until $d_{G_{AE},i} = EN$
evaluate the simulation of the solution S_k
forgive AE event modifications until up to mark A.

The figure 2 show how node bubbles are optimized. To optimize the two sub-bubbles, fitness functions of each sub-bubble is merged according to equation 3.

2) *Transactional resolution of independent goals*: When each bubble has been optimized, the system checks if there are conflicts of resources between each bubbles. However, instead of checking all bubbles of all related events in all regions, the system checks resources numerically localized into one workstation. Then, if one resource is used by several bubbles, more than it can be used, the system generates a parent bubble destined to solve conflicts between the two bubbles. In this case, considering the conflict, it generates peer of constraints. Each set of constraints should correspond to one sub-bubble. For example, if a resource is managed by two bubbles at the same time, two peers of constraints will be generated. The first one will consider that the first bubble can't use the conflicting resource, and the second one will consider that this

Algorithm 5 Constructing a solution for one ant associated to optimization of topographic path between located resources and a located task to apply.

Let $Paths$ being the set of paths to follow

repeat
 choose randomly a transportable resource $R_j \in R$ to look for.
 remove R_j from R .
 constructing a path $Path$ according to the ACO algorithm, until a free occurrence $O(R_j)_k$ of the resource type R_j was found. Reset construction if the path is too long.
 if $O(R_j)_k$ must be transported thanks to another resource R_t **then**
 repeat
 constructing a path $PathT$ according to the ACO algorithm from the located resource $O(R_j)_k$ until an occurrence $O(R_t)_{tk}$ of the resource type R_t was found.
 $valid \leftarrow true$
 if $O(R_t)_{tk}$ is used by another resource but not located at the same place than $O(R_j)_k$ **then**
 $valid \leftarrow false$
 end if
 if $valid$ **and** $O(R_t)_{tk}$ has not sufficient space to transport $O(R_j)_k$ **then**
 $valid \leftarrow false$
 end if
 until $valid$
 Mark the resource $O(R_t)_{tk}$ as transporting the resource $O(R_j)_k$. Note that the resource of transportation $O(R_t)_{tk}$ will have finally only one path to follow.
 if $O(R_t)_{tk}$ is not already used by another resource **then**
 $PathT \leftarrow$ concatenate $Path$ with $PathT$
 add $PathT$ to $Paths$
 end if
 else
 add $Path$ to $Paths$
 end if
until R is empty
Returns the different paths to follow, with their corresponding resources.

is the second bubble that cannot use the conflicting resource. This operation is repeated for each peer of conflicting bubbles, and for each conflicting resources. After that, each bubble is optimized another time, until no new bubble is generated and until no constraints set is altered.

3) *Dynamical projection*: When all bubbles are optimized, and when no more resources conflicts are detected, the system projects its plan of resource deployment over time. Each task will have an effect to the current events, but some events will evolve by producing new events. For example, if an epidemic is not treated quickly enough, then this epidemic will be propagated over space and time. This propagation is formalized by the generation of new events, each of them being uncertain. This generation is obtained thanks to the function $DE(E, t)$ (see section III-B1), which trigger events according to user's definitions. Then, the system had to opti-

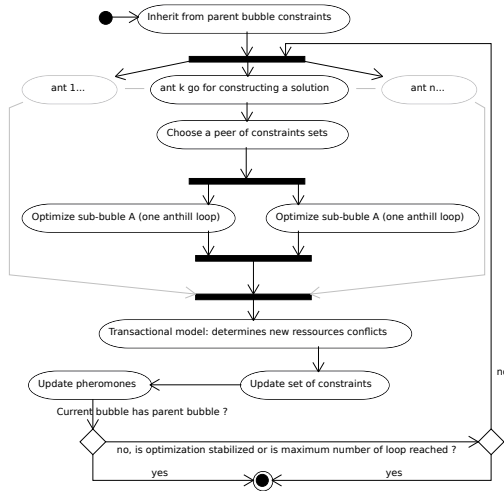


Fig. 2. Node bubble optimization process (diagram of activities).

mize a plan for the likely future events, which can generate themselves new events, then new bubbles. For each loop of optimization/projection, the system updates events uncertainty according to their occurrences.

IV. CONCLUSION AND PERSPECTIVES

We have proposed a model able to plan a set of tasks and able to deploy a set of resources according to declared events during a crisis, but also according to a set of simulated events over the future. The system processes a reflexive deliberation by applying a projection of its decisions over time, and by deducing related issues to deal with. Moreover, it manages data uncertainty, according to a formalism based on the Pareto law that produces stable results, in the context of a stochastic environment. Finally, the model produces a global optimum by exploring solutions locally in a first step, and globally if necessary. This solution is then adapted to large scale systems, that should be a huge distributed network of workstations. This work can then be introduced into a more global project, i.e. the conception of an asynchronous and distributed embedded application able to manage the deployment of human and material resources in the context of a crisis.

Future works should be centered into: the introduction of a user avatar taking into account the preferences of the user; the management of uncertainty related to avatars when workstations are disconnected from the network; the simplification of data managed by users through a participative solution and through an ergonomic interface [19], [20]; the process of experimentations and evaluations.

REFERENCES

- [1] M. Traore, M. Sayed-Mouchaweh, and P. Billaudel, "Learning Diagnoser and Supervision Pattern in Discrete Event System : Application to Crisis Management," in *Annual Conference of the Prognostics and Health Management Society*, 2013.
- [2] M. Sediri, N. Matta, S. Loriette, and A. Hugerot, "Vers une représentation de situations de crise gérées par le SAMU," in *IC 2012 Paris*, 2012.

- [3] N. El Mawas and J.-P. Cahier, "Towards a knowledge-intensive serious game for training emergency medical services," in *Proceedings of the 10th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Baden-Baden (Germany), 2013.
- [4] N. Matta, S. Loriette, M. Sediri, J. M. Nigro, Y. Barloy, J. P. Cahier, and A. Hugerot, "Representing Experience on Road Accident Management," in *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, Jun. 2012, pp. 364–366.
- [5] H. Van Dyke Parunak, S. A. Brueckner, J. A. Sauter, and R. Matthews, "Global convergence of local agent behaviors," in *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. New York, NY, USA: ACM, 2005, pp. 305–312.
- [6] H. Markowitz, "Portfolio selection*," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.
- [7] B. Mandelbrot and R. L. Hudson, *The Misbehavior of Markets: A Fractal View of Financial Turbulence*. Basic Books, 2006.
- [8] N. N. Taleb, *The Black Swan: Second Edition: The Impact of the Highly Improbable*. Random House Trade Paperbacks, 2010.
- [9] I. Alaya, "Optimisation multi-objectif par colonies de fourmis. Cas des problèmes de sac à dos." Ph.D. dissertation, Universt de la Manouba et Universit Claude Bernard Lyon 1, 2009.
- [10] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers & Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science (New York, N.Y.)*, vol. 220, no. 4598, pp. 671–80, May 1983.
- [12] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, May 1992.
- [13] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4. IEEE, 1995, pp. 1942–1948.
- [14] P. Larraaga and J. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publisher, 2001.
- [15] M. Dorigo, "Optimization, Learning and Natural Algorithms (in Italian)," 1992.
- [16] G. Di Caro and M. Dorigo, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal Of Artificial Intelligence Research*, vol. 9, pp. 317–365, May 1998.
- [17] J. R. Larus and R. Rajwar, "Transactional Memory," *Synthesis Lectures on Computer Architecture*, vol. 1, no. 1, pp. 1–226, 2006.
- [18] S. Iredi, D. Merkle, and M. Middendorf, "Bi-criterion optimization with multi colony ant algorithms," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, and D. Corne, Eds. Springer Berlin Heidelberg, 2001, vol. 1993, pp. 359–372.
- [19] X. Ma and J.-P. Cahier, "Semantically Structured VDL-Based Iconic Tags System," in *Human Interface and the Management of Information. Information and Interaction Design*, ser. Lecture Notes in Computer Science, S. Yamamoto, Ed. Springer Berlin Heidelberg, 2013, vol. 8016, pp. 465–474.
- [20] F. Rousseaux and J. Petit, "Towards an Anthropological-Based Knowledge Management," in *10th International Conference on Intellectual Capital*, K. M. . O. Learning, Ed., Washington, 2013.