A Multi-agent Approach for Space Occupation Problems

Jamila Boussaa, Mohammed Sadgal and Aziz Elfazziki

**Computer Science Department, Faculty of sciences Semlalia, University Cadi ayyad
Marrakesh, BP 2390, Morocco**

Abstract

The occupation of space is a recurring problem in many areas for constraint satisfaction and optimization. The used approaches tend to privilege the optimization or the satisfaction without leading to a general solution. In spite of the success of the few methods of space occupation problems, it can be interesting to consider new ways for resolution, in particular methods resulting from Artificial Intelligence techniques. Because the problem is NP-complex, one possibility of overcoming this complexity is to distribute it across multiple processing units and adopt an appropriate form for decision-making. To construct and evaluate possible solutions for this class of problems, we propose in this paper a general architecture that can accommodate several approaches for resolution through agglomerates of specialized solvers. On this basis, a general model of agent solver is provided. The competences and interactions of agents will be studied and classified according to space occupation problem types. One case is presented here, the resolution by coalition.

Keywords: *Space Occupation, Constraints, Satisfaction, Optimization, Coalition, Artificial Intelligence, MAS (Multi-Agent System), DSCSP (Distributed Space CSP).*

1. Introduction

The Space Occupation Problem **(SOP)** consists of placement of objects in a preset space with respect to a set of imposed constraints; some declared and others not; which appears at the installation time. Such occupation must be *optimal*. The mathematical models dealing with this problem privilege the optimization aspect (occupied space minimization, for example) without explicating or explaining the constraint satisfaction aspect. In fact, the expression of constraints is reduced to an evaluation function that is inadequate in the most cases. Because, the space is simplified in fixed zones (places) and the effort is focused on the assignment cost, the problems with non-numeric expression of constraints can't be resolved. So, the modeling cannot understand the problem as an objects assignment to a set of places.

Many of these problems, undertaken by traditional processing, are confronted with a realistic representation of constraints and objectives. Models as linear programming that require to translate the constraints in equations or statistical models that deal with the problem in term of classification or models based on physical phenomena (Annealing simulated) [1], confirm the difficulty to generalize the expression of constraints in a numeric aspect. More recently, Thierry Petit and al. [2] study the propagation of side constraints to solve problems. They provide a theoretical and experimental comparison of two main approaches for encoding over-constrained problems with side constraints. Even if their work is oriented constraint programming, the resolution is still problem-dependent.

Generally, constraints are used under an imposed numerical structure that doesn't allow several specialized procedures to cooperate. Indeed, constraints like ***"objects using water must be placed as far as possible from any electrical instrument" or "the object A must be seen by the object B"*** introduce inaccuracy and ambiguity and claim representation techniques and reasoning supported by Artificial intelligence (A.I.) approaches [3,4]. In this sense, the authors in [5] provide an expert approach system in firms.

The problem is NP-complex, thus we can't postulate that only the algorithm performance will be able to overcome the complexity, in spite of the efforts provided in [6]. Some models based on elementary behaviors of reactive agents (i.e ACO and PSO []) had succeeded for certain problems. Whereas other more complex models (cognitive agents) inspired from human behaviors like negotiation, cooperation and game theory still offer approaches to reduce the problem difficulty. These approaches are promising for several reasons such as the expression power to integrate qualitative constraints, need for cooperation between different types of knowledge and modeling by using the "Agent" paradigm [7].

Generally, the SOP is assimilated to a Constraint Satisfaction Problem (CSP). Constraints are treated on a scale of "severities" incorporating the preferences. The priority is to find solutions satisfying "severe" constraints then, order by preferences.

This article presents a multi-agent architecture allowing to express user requests (demands and preferences) easily and naturally and to greet several communities of agents. Each community uses a specific approach for resolution. The goal is double: first, we search to solve CSP with optimization and second, we develop an "infrastructure" able to integrate more than one approach for the resolution.

In the following section, we present the State of the art in the field. Section 3 exposes the detailed description of the

suggested approach. A presentation of negotiation and cooperation as a way to resolve the problem by an agents' community based on the coalition will be given in section 4. Section 5 provides an example to illustrate our method. Finally, we conclude by an optimistic note for work to come.

2. Related work

2.1 Space Occupation Problem (SOP)

Generally, the space occupation problem is expressed by using a CSP or SCSP (Space CSP) [8,10]: Objects are identified by multidimensional variables. A variable would be, for example, a vector of position, orientation and object dimensions. In the most approaches, the constraints are expressed using geometrical relations. But there exist other constraint types representing topological and/or functional nature. According to several authors [8,12], the main difficulty to solve this problem comes from some aspects like the presence of constraints and objectives together. The hardness to optimize antagonistic criteria and to obtain a discrete formulation of the problem leads to NP-Complexity or worse.

2.2 Resolution Methods

Traditional techniques: To solve this problem, some classical approaches were used. They can be classified in three main categories: The constructive approach which is a top-down approach type according to [8]. The iterative approach tries to improve an occupation of space starting from an earlier one by moving an object or by permuting two objects [10].The hybrid approach is a coupling, more or less extremely, of two preceding approaches. The basic algorithm is the chronological Backtrack [9]. But this mechanism produces a selectivity problem. Indeed, in the failure case, the algorithm reconsiders the last choice carried out, without worrying to know if this choice has any responsibility in the current failure.

The distribution aspect, Resolution by MAS/DSCSP:
A DSCSP uses the traditional definition of a SCSP, by adding the assumption that variables (or constraints) are managed by agents in order to satisfy constraints. Constraints can exist between variables of the same agent (intra-agent constraints) or variables of different agents (inter-agents constraints). Solving the problem is usually seen like carrying out the coherence or the consistency of a multi-agent system. The resolution in a distributed environment allows a parallel processing; therefore time can be saved. But this implies the use of communication mechanisms efficiently in order to ensure the system coherence during the resolution.

In its work, Yokoo [11] has adapted several algorithms: DBA, ABT... The agents are considered responsible for maintaining the environment update. Then, these approaches lead to obtain partial solutions quickly. It might be interesting for dynamic problems that require a great reactivity.
Another approach based on cooperation was imagined by [12] in APO. The agents have a priority and cooperate during mediation meetings. When an agent cannot find consistent value with the more priority agents, it launches a mediation meeting or it changes its value and transmits it to its neighbors. Method ADOPT [13] has for principal application, distributed optimization under constraints. Each constraint is associated with a cost and each agent has to minimize the function 'global objective' (total cost).
Since the purpose of the problem is a configuration (assignment of the variables) satisfying the constraints, [14] has imagined reaching this solution by emergence with agents' auto-organization. Thus, the artificial system must fulfill an adequate function. To change function, just change the organization of the system components [26]. Several other authors propose the resolution of certain particular cases by organization and coalition [15,16].

The approaches above improve conventional algorithms by introducing the distribution and often in the guise of parallelism while admitting certain assumptions such as communication by messages according to Yokoo [11]. But, the majority of these systems encounter a problem on the communication level where it is necessary to manage a great number of messages generated by the agents. The idea of the solution emergence is interesting but the research of the specific adequate functions is very difficult. So we are looking for our approach to solve the problem by a collective decision by all agents using coalition and deliberation mechanisms.

3. Suggested approach

3.1 General architecture

We present a MAS architecture allowing to adapt resolutions by using several types of agents, from a simple reactive ("reflex") agent to a cognitive one more complex. The recourse to a multi-Community architecture (figure 1) of contextual agents depending on the CSP categories is justified by several arguments: 1) the resolution is perceived as the effort of several units, each one contributes by a partial or total solution. 2) The interpretation of the problem expressed by the user belongs to the resolution. The presence of the highly cognitive agents is useful in the clarification of the user requests that are often too general. 3) Adapting the algorithm to the problem (or the reverse) influences the solution's quality. 4) The parallelism question, as reconsidered under the Distributed A.I. MAS

Paradigm, offers new possibilities to converge towards solutions adapted via: competition, cooperation, negotiation, organization…

Our long-term goal is to offer a system model able to receive several communities of agents (solvers [17]), each one is specialized in the resolution of a class of problems. The community has its own behaviors and has its own methods of resolution. The Interface agent (Supervisor) deals with the interpretation of the initial problem and the contexts for the specialized communities. The solutions suggested by a community can be retained by the supervisor according to some evaluation criteria. A definite decision will be concerted with the user (figure 1).

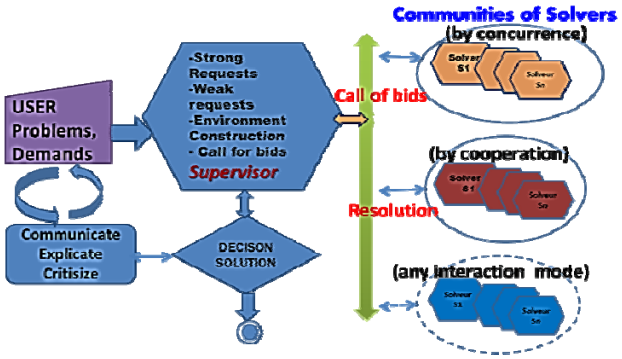


Fig. 1. General architecture of the MAS

a. Types of agents

User Agent (UA) : The User agent represents the user. He is the initiator of the problem in form of descriptions concerning space occupation, objects to be placed and the demands (constraints and objective). He communicates with the Supervisor agent through an interface.

Supervisor Agent (SA): The Supervisor agent is a "mediator" between the communities of the solvers and the User agent. It represents descriptions and demands into a Base of facts and rules in predicate logic. By using rules of transformation, the requests are converted into "severe" constraints which must be respected obligatorily and into preferences that are hopes to be carried out as well as possible. The SA has the heavy task for selecting the community that would be "able" to solve the problem by using a set of problem categories and task announcements. So this agent has a diagram even an ontology enabling it to classify the problem (placement, cutting, routing…) and to suggest the community of suitable solvers.

The Community of agents: A community consists of homogeneous agents equipped with competences and are specialized in the resolution of one or more classes of problems according to a resolution model (competition, cooperation…, (see section 4).

b. Environments

The choice of the environment and its properties is closely related to the CSP type and to the singularity of the community. In general, within the framework of a closed system, it is possible to determine neighbors for each agent. One way to formulate is to fix them initially since their creation. The choices, made during the construction of the neighbors, can orient a model in any direction.

c. Interaction of agents within a community

How agents interact and how they are organized make them to coordinate themselves, to cooperate or to negotiate. Coordination is an essential point, especially with respect to process implementation of the multi-agent models, to determine which does what and when is a non-trivial problem that can have infinity of solutions. Each one of them can appreciably modify results obtained from simulations [18]. Figure 2 recapitulates what will be integrated in the system.

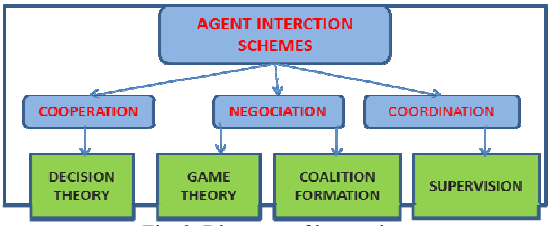


Fig. 2. Diagrams of interaction

The decision theory, where an agent tries to maximize a criterion (called utility), is rather close to the game theory. The difference between decision and game theories is that the game theory takes into account the current situation and also the future choices of agents. The coalition formation is another approach used in interactions between agents. It acts, for gents confronted with a request, to make the individual compromise in order to reach a consensus for all parts (ideal case). Then, difficulty will define the communication protocol in an adequate way. The protocol must make agents exchange their current choices and modify them until realization of consensus (see section 4).

3.2 Basic concepts common to agent communities

In order to provide a general structure for the SOP resolution, we present here a description based on the following definitions:

A placement space of two or three dimensions in which geometrical objects with possible functional and topological characteristics will be placed. The installation is governed by all constraints using characteristics of objects and space.

The goal is to occupy space by installing all objects, satisfying constraints and carrying out objectives as well as possible.

In the distributed version of a CSP, authors traditionally distribute variables or constraints on agents. Each agent is

given the responsibility to solve its problem locally while contributing to the global resolution.
We consider that each agent deals with one object to place in the space (figure 3).

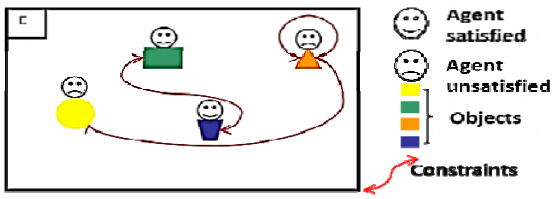


Fig. 3. Occupation of space by agents (objects)

3.2.1 Notations and definitions

In our model, the physical world is abstracted in States, Actions and State Transitions caused by actions.
A State is the assignment of all objects (agents) to places in the occupation space. A State Transition relates to the passage from one state to another. Lastly, an action is the way by which a transition will be realized.
The resolution will be seen like series of transitions to pass from an initial State to a final State (solution).
Let us consider the following definitions:
E: space in **2D or 3D** (e.g **rectangle (x0, y0, xd, yd))**
A = $\{A_1,\ldots A_n\}$ set of agents (agent represents an object)
S= $\{S_0,\ldots,S_m\}$ set of States
C = $\{c_1,\ldots,c_p\}$ set of constraints
B= $\{b_1,\ldots,b_q\}$ set of objectives
AC = $\{a_1,\ldots,a_r\}$ set of actions of agents
An action a_j is regarded as the joint action of all the agents: $a_j = (a_{j1},\ldots,a_{ji},\ldots, a_{jn})$ where a_{ji} is the i^{th} agent action.
P= $\{p_1,\ldots,p_t\}$ set of plans
A plan p_i is a set of joint actions: $p_i = \{a_0, a_1,\ldots,a_k\}$
Actions: An *individual* action "**a**" of an agent is a change of its place in space **E**. This change can be performed using combination of geometrical operators like translation and rotation (in certain cases of design, the action can also be a change of object dimensions).
For example (figure 4):
$a = (\delta t_u, \delta t_v, r_w) \in R^3$, if P_i = position occupied by A_i in **E**, $P_i = (x_i, y_i, t_i)$: location $(x_i, y_i) \in E$ and orientation t_i of the local reference (object reference).
$a(P_i) = P'_i$: Change of place and orientation of A_i by application of action $a = (t_u, t_v, r_w)$.
Then $P'_i = (x'_i, y'_i, t'_i)$ where $x'_i = x + \delta t_u$, $y'_i = y_i + \delta t_v$ and $t'_i = t_i + \delta r_w$
$a_0 = (0, 0, 0)$ is the action identity: the agent does not move.

Place: A place for an agent is defined by a geometrical position P_i (Coordinates, Orientation) and object Dimensions. We note this place $Pl_i = (P_i, Dg_i)$, where Dg_i are geometrical object dimensions (e.g. length and width for a rectangular object in **2D**).

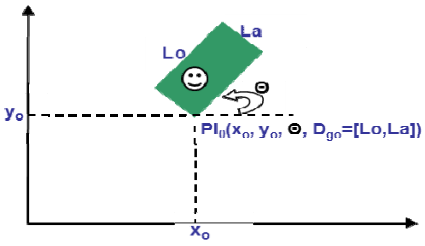


Fig. 4. An object in a referential related to space E

State: We define a State S_k like a triplet $<\mathcal{PL}_k, C_k, \mathcal{B}_k>$:
$\mathcal{PL}_k = \{Pl_{k1},\ldots,Pl_{ki},\ldots, Pl_{ki}\}$ is all places occupied by agents in **E** (Pl_{ki} is the place of agent A_i in S_k).
C_k is the subset of satisfied constraints in the State S_k
$\mathcal{B}_k$ is the subset of objectives achieved in the State S_k

States' Transition: A transition on States is defined by the (S_k, S_l) pair, and denoted $S_k \mapsto S_l$, State S_l is regarded as a consequence of the action causing the $S_k \mapsto S_l$ transition on S_k. We note $(S_k \mapsto S_l)$ /a_j to indicate the transition and its cause a_j.

3.2.2 Admissibility and optimality

Admissibility: $\mathcal{C}$ is the set of "severe" constraints in the **SOP** with cardinal $(\mathcal{C})=|\mathcal{C}|=$ **p.**
A constraint is seen like a relation (arcs on figure 3) between one or more agents.
A State $S = <\mathcal{PL}_S, C_S, \mathcal{B}_S>$ is known as **admissible** if and only if $C_S = \mathcal{C}$ (situation where all the constraints are satisfied).
Note: The relation **"same set of satisfied constraints"** is a relation of equivalence on the set of States $\mathcal{S}$. Thereby, the **admissible** States constitute a class of equivalence by: *"they have $\mathcal{C}$ like set of satisfied constraints"*. Let $\mathcal{E}_a$ this class, $\mathcal{E}_a = \{S \in \mathcal{S} / S = <\mathcal{PL}_S, C, \mathcal{B}_S>\}$.

Agent level satisfaction: If $\mathcal{C}^{Ai}$ indicates the set of constraints for agent A_i, then: $\bigcup_i \mathcal{C}^{Ai} = \mathcal{C}$ with $A_i \in A$
State $S = <\mathcal{PL}_S, C_S, \mathcal{B}_S>$ satisfying the agent A_i is said **Ai-admissible** if and only if $\mathcal{C}^{Ai} \subset C_S$.
Generally, $S = <\mathcal{PL}_S, C_S, \mathcal{B}_S>$ satisfies a group of agents G_A if $\bigcup_i \mathcal{C}^{Ai} = \mathcal{C}$ with $A_i \in G_A$
Notice (based on individual satisfactions of the agents):
For any State **S** if $\forall i$, $\mathcal{C}^{Ai} \subset C_S$ then **S** is admissible.

Optimality: The goal is to find an admissible State optimizing the objectives (preferences): These preferences can be regarded as less severe constraints, but it will pose a difficult problem: the optimization. Most of the applications propose optimization by seeking more adequate models. On our side, we suppose the existence of an evaluation function

measuring a realization degree for objectives. Thus, our problem is to define an evaluation function based on each agent's *appreciation*.

Appreciation: An agent can evaluate any State **S** according to the satisfaction of its own constraints, the remainder of constraints and objectives. Then, evaluation is defined by the function $ju: \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$.

$ju(A_i, S)$ expresses a satisfaction degree (constraints and preferences) by agent A_i for State **S.** This value is calculated by the agent A_i taking into account its position in **PLs**, subset of satisfied constraints and objectives carried out, locally and globally.

3.2.3 Elements of resolution

To solve the problem, it is primordial to have $\mathcal{Ea}$ (set of admissible solutions). In this case, procedures must determine the optimal solution in $\mathcal{Ea}$.
Given the problem complexity, we emit two assumptions:

- To prove the existence of $\mathcal{Ea}$ is not necessary to begin the resolution.
- The optimal solution search is evaluated in terms of convergence on $\mathcal{Ea}$ (objectives are carried out on acceptable States rather admissible States)

The search for an acceptable State:

An A_i**-k-admissible** State is a State where **k** constraints are not satisfied (**k**: dissatisfaction degree) for the agent A_i.
An A_i**-admissible** State is an A_i**-0-admissible** State (all A_i constraints are satisfied: k=0).
A State is called **acceptable** if it is admitted admissible even if $k \neq 0$.
We will say that A_i and A_j are **neighbors** if there is at least a constraint between A_i and A_j.
Let A_{Gi} the A_i neighbors' set.
We present a **Pseudo-algorithm** to filter the A_i**-k-admissible** States with the smallest degree **k**:

```
                /**** for each agent Ai and Sa : actual State ****/
Search_Ai_k_admissibles(Ai, Sa) {
kai          : current dissatisfaction degree in Sa for the agent Ai
k=0          : initialization of degree dissatisfaction at the begin of this search
ESi = ∅      :ESi : list to contain possible A-k-admissible States
Pi = ∅       :Pi : lists of plans to reach the A-k-admissible States.
1-find Zik(Sa):zone of possible places with k unsatisfied constraints of Ai.
   If Zik(Sa) != ∅ do:
          - For each Pli(possible place) for Ai in Zik(Sa) do:
            determine the pjk plan and Sjk such as (Sa↦Sjk)/pjk
            with Sjk = <PLi, Ci, Bi>
            ESi = ESi ∪ {Sjk}; Pi = Pi ∪ {pjk}
          - End For
          If ESi != ∅ go to 2 endif
   Else
        k = k+1
         if k ≥ kai to go to 2
         else go to 1
         endif
   Endif

2-if ESi = ∅, ESi = {Sa}, k=kai
 else
      consider Sjk, and  record:
          the degree of dissatisfaction k.
          the pjk plans to reach Sjk from Sa,
          the evaluation of Sjk by ju.
endif
}
```

Notes:

- All agents are executed in a parallel way; the algorithm provides the possible places when the domain is discrete. To avoid an intensive CPU time, the agent may save only the envelope of the $Z_{ik}(Sa)$ zone and determine places by applying associated plans in time of need.
- The Search Algorithm can also cover neighboring: Every acceptable State $S \in ES_i$ for A_i must be validated by its neighbors (A_{Gi})

Ai-Optimal State: The first task of each agent is the generation of plans leading, at least, with **Ai-k-admissible** solutions. But with objectives, the agent will seek to propose the "best plan" within the meaning of the appreciation ju.
The **Ai-optimal** State is S_{opi} such as:
$S_{opi} = \mathbf{arg}(\mathbf{Max}(ju(Ai,S_i)))$ with $S_i \in ES_i$ **(Ai-k-adm**
We also note $u_{ji} = ju(Aj,S_i)$ for very $S_i \in ES_i$: the S_i evaluation by the A_j agent

4. Resolution Techniques

Multi-criteria traditional techniques: With a utility function, the problem can be solved in a centralized way according to mathematical models and using the multi-criterion techniques [19]. These techniques establish an order on some offered possibilities and the decision maker (human) has to arbitrate. Thus, in our case, the agents will be able to replace the decision makers and the problem will be multi-criterion and multi-decision makers.

Indeed, the numerical measurement of the agent utility is already a strong assumption compared to the simple classification of the available choices. The comparison of the utility of two individuals is even difficult.
Why would a plan, appreciated with 0.8 by an agent and 0.5 by another, be preferred with that respectively appreciated 0.4 and 0.9? It is difficult to represent the importance of a decision maker by a weight in order to respect the general structure of these techniques. The disadvantage is that the aggregation procedures and criteria transformation into constraints are delicate to carry out especially when certain decision makers and interlocutors have no scientific culture.

Negotiation / Cooperation: In order to build a model of negotiation by a multi-agent system, some elements must be defined. In [20], authors identify three components as being most fundamental:

1- *Negotiation Protocol:* It is a set of rules managing the interaction.
2- *Negotiation Object:* It consists of attributes on which agents wish to find an agreement
3- *Decision Strategy or model of the agents:* It is a reasoning process used by agents, in agreement with negotiation protocol, to achieve their goals. In literature, there exist three great approaches of negotiation in the multi-agent systems [21] based on:

- *The game theory*: it studies the behavior of a "rational" agent confronted with one (many) adversary during a game, in order to find an optimal strategy maximizing its own utility. Several protocols were studied [22]. The game theory also uses strong concepts of convergence based on the Nash equilibrium or the Pareto's optimum.
- *Heuristics:* The lack of resources and time does not make it possible to elaborate a better policy by analysis with game theory. In order to mitigate these limits, the heuristic approaches try to reach acceptable approximations with the theoretical optimal results found by the game theory [23].
- *Argumentation:* The argumentation is an adequate model to represent the internal reasoning of an agent, and it is based on the construction of arguments. It considers the model interactions of multi-agent in form of dialogs [24].

4.2 A proposition: resolution by forming coalition among agents

In a community of cooperative agents, an agent uses a coalition when it is unable to satisfy all its constraints, i.e. when it cannot find $\mathbf{A_i}$**-0-admissible** States. Here's an example of resolution based on the negotiation using an analysis by the game theory, among different other models of negotiation (cited above), that will be integrated in the system. As it is difficult to incorporate utilities of agents, an agent will seek an accepted plan (or plans) by all its neighbors. This plan must produce a State which is "better" or at least equivalent of the Actual State. The solution can be achieved with a **Pareto's optimum** (in the game theory): an agent cannot increase its utility if at least another utility agent is decreased. The protocol of negotiation is based on this principle [25]. The agent initializing the negotiation seeks the plans which it prefers. It transfers them by grouping and ordering to an agent close to its choice. The agent receiving plans, filters those "better" than the actual state, reorders them according to its utility, and sends them with the same procedure to an agent of its choice. The last agent selects the most interesting plan (or plans) that will constitute the Pareto's optimum.

4.2.1 Definitions

Coalition: A coalition is a subset $\mathbf{S \subseteq A = \{A_1, \dots, A_n\}}$, where **A** is the agents' set ($\mathbf{2^n - 1}$coalitions are possible).
In DCSP and SOP contexts, the resolution by coalition was applied to several problems such as the task allocation, resource allocation… But the CSP differs from those problems that define coalition structure as a priori. Indeed in CSP there is a strong dependence, the agent choices are not fixed and change permanently by the others' choices.
Then, agents form a coalition around one or several constraints to find a solution. Several neighbor form a coalition and provide concerted action plans to satisfy the joint constraints as well as possible.
Set of coalitions: a set representing a solution to the problem of coalitions' formation. Agents form coalitions to satisfy constraints with objectives. It is about the set of plans which provides a **State solution** in our case. Then we denote by ***Group*** a set of coalitions' sets.
Context**:** the parameters taken into account in the problem (must be stable during the negotiation).
Utility function: the utility function can be ordinal or cardinal. The cardinal associates a utility with a set of coalitions and a given context. The ordinal permits to compare two sets in a given context. In this case, to measure the utility of a **State** means to compare it with a reference **State** (see section 4.4).
Reference State: The agents must know if they accept "States solutions**"**, so it is necessary that they can compare a State with what they are able to obtain during the negotiation. This minimum is the reference State.

4.2.2 Negotiation Algorithm

Each negotiation proceeds in three phases:
Phase 1: Initialization of the negotiation and transfer of constraints. The initiating agent informs all the others that it begins a new negotiation. Any agent which will want to begin another from them will have to await the end of the actual negotiation. The initiating agent calculates all the possible coalitions. It gathers them in group of solution sets

and sends it to itself and/or to the agent which must begin the negotiation.

Phase 2: Negotiation
When an agent receives a group of sets: it preferably classifies by order (with its utility) the received sets in homogeneous groups. It classifies only the sets at least equivalent, with its reference State. Then, groups are sent to the following agent by a decreasing order.
If all agents already took part in the negotiation (the agent is thus the last). **So** at least one of the sets received is acceptable, it considers the best set. This set is **Pareto's optimum**.

Phase 3: Transmission of the solution
Once the last agent identified a Pareto's optimum, it transmits this set to all agents which accept it as a solution of the negotiation.

Resolution Steps for SOP: Due to the problem complexity, each agent works initially to satisfy its constraints. For those not satisfied, it will form a coalition with the agents implied in these same constraints. The problem looks like a "repetitive game", the solution of the problem can be obtained using several negotiation rounds.

1. A starting State **So** is given:
   Several heuristic can be used here, for example:
   - **So** is the first space occupation without constraints.
   - Since there is dependence, one can proceed by a sequential occupation: an order is established and each agent will seek the plan, by regarding the placed agents.
2. With the reference State (**So** at the beginning), the algorithm (section 3.3.1) is carried out. Any agent $\mathbf{A_i}$ which is not able to propose an **Ai-0-admissible** State will seek to form a coalition with its neighbors.
3. A coalition solves the problem by negotiation according to algorithm 4.2:
   Plans provided into 2. are considered and evaluated by each member of the coalition, the initiating agent of the negotiation orders plans by groups according to its utility and sends them to an agent of its choice, this one retains only those that provide "better" States and so on, to the last agent. Plans retained by the last will constitute the solution. If the State obtained is **Ai-0-acceptable** for each $\mathbf{A_i}$, then the negotiation is finished. If not the actual State will be regarded as reference State and it begins again since 2.
4. Without improvement and at the end of a number predefined of negotiation rounds an agent decides to stop the formation of coalitions.

4.2.3 Determination of agent utility function (SOP)

We specify here how to calculate utility value $\mathbf{u_{ij}}$ by $\mathbf{A_j}$ agent for a plan $\mathbf{p_i}$ suggested by the agent $\mathbf{A_i}$.

Although it is difficult to model this evaluation using a quantitative function, it is necessary to use certain indices:(i) Satisfaction rate of constraints relating to the agent, (ii) Potential utilization ratio, (iii) Total satisfaction rate and (iv) Satisfaction neighborhood rate.

a. Definition of rates
Let:
S: actual state,
E: Occupation Space,
C: set of constraints,
$\mathbf{C^{Aj}}$: set of constraints of agent A_j,
$\mathbf{Z_S^{Aj}}$: satisfaction zone of A_j in **S**,
$\mathbf{C^j_{Si}}$: satisfied set constraints of A_j in $\mathbf{S_i}$
$\mathbf{a_i}$ is action of $\mathbf{A_i}$ with $\mathbf{a_i(S) = S_i}$; $\mathbf{(or\ (S \mapsto S_i)/a_i)}$
We call:
- Relative satisfaction rate in $\mathbf{S_i}$: $\mathbf{r_{ij} = 1 - |C^j_{Si}|/|C^{Aj}|}$
- Potential utilization ratio in $\mathbf{S_i}$: $\mathbf{z_{ij} = |Z_S^{Aj}|/|E|}$
- Total satisfaction rate in $\mathbf{S_i}$: $\mathbf{g_i = 1 - |C_{Si}|/|C|}$
- Satisfaction neighborhood rate: $\mathbf{n_i = 1 - |C^{Agj}_{Si}|/|C|}$; **Agj** is the set of $\mathbf{A_j}$ neighbors

b. Utility
We can model the utility $\mathbf{u_{ij}}$ that is an $\mathbf{A_j}$ judgment on the $\mathbf{A_i}$ action by using a linear combination as follows:
$\mathbf{u_{ij} = w_{1j}*r_{ij} + w_{2j}*z_{ij} + w_{3j}*g_i}$ (we use $\mathbf{g_i}$ when all agents are linked by constraints, otherwise $\mathbf{n_i}$)
The term: $\mathbf{w_{1j}*r_{ij} + w_{2j}*z_{ij}}$ expresses the personal interest of the $\mathbf{A_j}$ agent
The term: $\mathbf{w_{3j}*g_i}$ expresses the global interest
Let: $\mathbf{u^p_{ij} = r_{ij} + z_{ij}}$ and $\mathbf{u^g_{ij} = g_i}$
If $\mathbf{w_{1j} = w_{2j} = \alpha_j}$ and $\mathbf{w_{3j} = \beta_j}$ then
$\mathbf{u_{ij} = \alpha_j * u^p_{ij} + \beta_j * u^g_{ij}}$
With $\mathbf{\beta_j = 1 - \alpha_j}$: the more one privileges the personal interest, the more it ignores the global interest and vise-versa ($\mathbf{\alpha_j \in [0,1]}$).
Finally: $\mathbf{u_{ij} = \alpha_j * u^p_{ij} + (1 - \alpha_j) * u^g_{ij}}$
Each $\mathbf{A_j}$ agent adopts its own strategy (choice of $\mathbf{\alpha_j}$) to calculate its preference (i.e. $\mathbf{\alpha_j}$= ½ is a neutral strategy)

4.2.4 Discussion

In our open architecture, we can use any form of cooperation. So, we use a *Generic Cooperation-based Method definition* [26] that is held on: (i) Cooperation can be viewed as a generic concept manipulated by problem solvers, (ii) It transcends to all the CSP methods, (iii) Taking inspiration from biological and socio-economic notions of cooperation and (iv) An agent alone is unable to find the global solution and it has to interact locally with its neighbors in order to find its current actions and to be able to reach its individual goals and help its neighbors

Thus, it can produce some categories of Cooperation-based algorithms as the Population-based approaches inspired by evolution and the behavior of insects, birds…
Their principles are: (i) A population is a set of individuals (agents), (ii) Each agent is able to find a solution to the problem and (iii) An agent knows the whole set of variables that define the problem. Agents coordinate to find a solution.
The common problem is how to coordinate several concurrent searches to efficiently find a good solution?
Several methods are essentially used in optimization problems: Evolutionary algorithms, genetic algorithms (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) [16].
In ACO, The pheromone deposited by ants gives relevant information about the region of the search space and modifies later the behavior of the other ants.
In PSO, Particles are influenced by the velocity and position of the local and global bests: cooperative information exchange allowing efficient exploration phase.
The fitness function of GA determines at a time the better individuals which will share their genes with other members of the population to produce new relevant offspring.
The essential difference with the coalition resolution: the cooperation is based on negotiations using game theory. Agent has a pseudo-global vision (must know its neighbors) and not a local vision. It offers, accepts or rejects solutions in concert with its neighbors. The utility function takes into account local and global aspects (see 4.2.3).

5. An illustrative example

To simplify, we consider 4-queens problem. There is no optimization. Only rigid constraints to satisfy, namely: ***"neither jointed line nor diagonal with the occupied squares".*** We will restrict the definition of the utility as follows: $\mathbf{u_i^p}$= number of satisfied constraints for one agent $\mathbf{A_i}$, and $\mathbf{u_i^g}$: number of satisfied constraints for all agents.
Each $\mathbf{A_i}$ represents one queen **i**. All agents are neighbors.
With α_i**=1/2**, then, for $\mathbf{A_i}$ we have: $\mathbf{u_i = 1/2(u_i^p + u_i^g)}$.
Let us consider for example, the $\mathbf{S_0 = (1,1,1,1)}$ positions of the 4 agents (line occupied by agent $\mathbf{A_i}$ in its column).
Let us notice that for two agents, the maximum number of non-satisfied constraints is **1**. Thus, for an agent, the maximum number of non-satisfied constraints is **3**.
If $\mathbf{A_1}$cannot satisfy all its constraints (whatever its position), then it forms a coalition with its neighbors **($\mathbf{A_2}$, $\mathbf{A_3}$ and $\mathbf{A_4}$)**.
So is the reference State at the beginning.

So =

We note:
- $\mathbf{U_p=(u_1^p, u_2^p, u_3^p, u_4^p)}$,
- $\mathbf{U=(u_1, u_2, u_3, u_4)}$ **with** $\mathbf{u_i = 1/2(u_i^p + u_i^g)}$ and $\mathbf{u_i^g = u_1^p + u_2^p + u_3^p + u_4^p}$

At **So,** utilities of agents are identical:
$\mathbf{U(So) = (u_1(So), u_2(So), u_3(So), u_4(So)) = (0,0,0,0)}$
The plans (thus States obtained by these plans) suggested with the first round:

States proposed by $\mathbf{A_1}$:
S_{11}= (2,1,1,1)→ Up (S_{11}) = (2,0,1,1) from where u_1 = ½ (2 + (2+0+1+1))=3, u_2=2, u_3=u_4=5/2, thus U(S_{11})=(3,2,5/2,5/2)
S_{21}= (3,1,1,1)→ Up (S_{21}) = (2,1,0,1) → U(S_{11})=(3, 5/2,2,5/2)
S_{31}= (4,1,1,1)→ Up (S_{31}) = (2,1,1,0) → U(S_{11})=(3,5/2,5/2,2)
States proposed by $\mathbf{A_2}$:
S_{12}= (1,2,1,1)→ Up (S_{12}) = (0,1,0,1) → U(S_{12})=(1,3/2,1,3/2)
S_{22}= (1, 3,1,1)→ Up (S_{22}) = (1,2,1,0) → U(S_{22})=(5/2,3,5/2,2)
S_{32}= (1,4,1,1)→ Up (S_{32}) = (1,3,1,1)→U(S_{32})=(7/2,9/2,7/2,7/2)
States proposed by $\mathbf{A_3}$:
S_{13}= (1,1,2,1)→ Up (S_{13}) = (1,0,1,0) → U(S_{13})=(3/2,1,3/2,1)
S_{23}= (1, 1,3,1)→ Up (S_{23}) = (0,1,2,1) → U(S_{23})=(2,5/2,3,5/2)
S_{33}= (1,1,4,1)→ Up (S_{33}) = (1,1,3,1)→U(S_{33})=(7/2,7/2,9/2,7/2)
States proposed by $\mathbf{A_4}$:
S_{14}=(1,1,1,2)→ Up (S_{14}) = (1,1,0,2) → U(S_{14})=(5/2,5/2,2,3)
S_{24}=(1, 1,1,3)→ Up (S_{24}) = (1,0,1,2) → U(S_{24})=(5/2,2,5/2,3)
S_{34}=(1,1,1,4)→ Up (S_{34}) = (0,1,1,2) → U(S_{34})=(2,5/2,5/2,3)
All these $\mathbf{S_{ij}}$ solutions can be retained by any $\mathbf{A_j}$ agent because $\mathbf{u_j (S_{ij}) > u_j (So)}$
$\mathbf{A_1}$ initiates negotiation; then forms 6 groups of plans ordered (decreasing order) according to its utility $\mathbf{u_1}$:
Groups in decreasing order: $\mathbf{G_1}$**= (**$\mathbf{S_{32}}$**,** $\mathbf{S_{33}}$**);** $\mathbf{G_2}$**= (**$\mathbf{S_{11}}$**,** $\mathbf{S_{12}}$**,** $\mathbf{S_{13}}$**);** $\mathbf{G_3}$**= (**$\mathbf{S_{22}}$**,** $\mathbf{S_{14}}$**,** $\mathbf{S_{24}}$**);** $\mathbf{G_4}$**= (**$\mathbf{S_{23}}$**,** $\mathbf{S_{34}}$**);**$\mathbf{G_5}$**= (**$\mathbf{S_{13}}$**)** and $\mathbf{G_6}$**= (**$\mathbf{S_{12}}$**).** These groups are sent in this order to $\mathbf{A_2}$

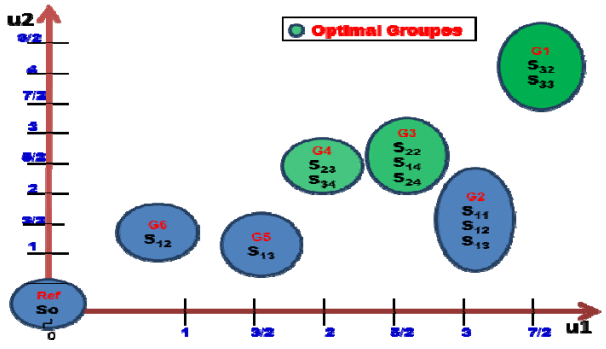


Fig. 5. Graph representing the utilities of the agents A_1 and A_2

$\mathbf{G_1}$ is sent at first to $\mathbf{A_2}$agent (chosen by $\mathbf{A_1}$, heuristics can be used), $\mathbf{A_2}$ evaluates $\mathbf{G_1}$ according to its utility here:$\mathbf{G_1}$**= (**$\mathbf{S_{32}}$**,** $\mathbf{S_{33}}$**),** $\mathbf{u_2\ (G_1)}$ **= (9/2, 7/2)**, which would be better than:$\mathbf{u_1\ (G_1)}$ **= (7/2, 7/2)**

$\mathbf{A_2}$ will then keep $\mathbf{G_1}$ in its entirety and sends it to A_3 (according to its choice), this one will keep $\mathbf{G_1}$because $\mathbf{u_3(G_1)}$ **= (7/2,9/2),** and sends it finally **to**$\mathbf{A_4}$, $\mathbf{u_4\ (G_1)}$ **= (7/2, 7/2)** and decides to keep it because it is better than $u_4(S_0)$=0.

X	X	X	X

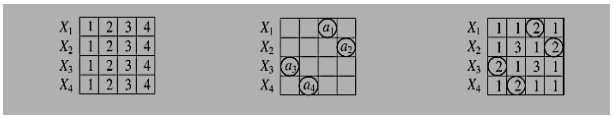
The satisfaction solution is $\mathbf{S_f}$ with $\mathbf{U(S_f)}$ = **(15/2,15/2,15/2,15/2)** and is not reached yet, the agents will decide on $\mathbf{S_{32}}$ or $\mathbf{S_{33}}$ to continue, let us suppose that S_{32}.(because there is no objectives for comparison)
The reference State then is changed: $\mathbf{S_0 \leftarrow S_{32}}$, with: $\mathbf{u_1(S_0) = 7/2}$, $\mathbf{u_2(S_0) = 9/2}$, $\mathbf{u_3(S_0) = 7/2}$ and $u_4(S_0) = 7/2$,
Agents proceed to another round (2^{nd}) of negotiation. We have two propositions:
$\mathbf{S^2_{11}=(2,4,1,1) \rightarrow Up(S^2_{11})=(3,3,2,2) \rightarrow U(S^2_{11})= (13/2,13/2,6,6)}$
$\mathbf{S^2_{24}=(1,4,1,3) \rightarrow Up(S^2_{24})=(2,3,2,3) \rightarrow U(S^2_{24})= (6,13/2,6,13/2)}$
$\mathbf{S^2_{11}}$, is chosen according to the same process, $\mathbf{S_0 \leftarrow S^2_{11}}$, with $\mathbf{U(S_0) = (13/2,13/2, 6, 6)}$
In the last round (3^{th}): the State $\mathbf{S^3_{14} = (2,4,1,3)}$ proposed by $\mathbf{A_4}$ is accepted by all other agents:
$\mathbf{U(S^3_{14})}$ = **(15/2,15/2,15/2,15/2),** thus $\mathbf{S^3_{14}}$ is a solution (satisfying all the constraints) and end of negotiation.

$\mathbf{S^3_{14}}$ =

		X	
X			
			X
	X		

This problem was used by all CSP algorithms for tests like:
-The *nogoods* (conflictual configurations) and potential solutions communicated by agents to their neighborhood in ABT or AWCS cooperatively [11].
-The heuristic min-conflict used ERA is a means to represent the fact that agents cooperatively act by minimizing the negative impact of their actions
- Population-based approaches (ACO, PSO, GA…)

In our example, we want just to show that CSP solution can be obtained under game theory as a pareto-optimal or nash-equilibrium. By comparison, we quote the model ERA (Environment, Reactive rules and Agents) [27] to have an idea of utility function that is reduced to the number of constraint violations:
In solving a CSP with ERA method, each agent represents a variable and its position corresponds to a value assignment for the variable. The environment for the whole multi-agent system contains all the possible domain values for the problem, and at the same time, it also records the *violation numbers* for all the positions. An agent can move within its row, which represents its domain. Three reactive behaviors (rules) were introduced: *better-move*, *least-move*, and *random-move*. The move of an agent will affect the violation numbers of other rows in the environment.



- (a) (b) (c)

Fig. 6. (a) The representation of domain values for a 4-queen problem. (b) Four agents dispatched into the 4-queen environment. (c) Updated violation numbers corresponding to the positions of the four agents.

(a) (b)

Fig. 7. (a) Violation numbers at the initialization step. (b) Violation numbers updated having placed a1 at (3, 1).

At the initialization step, the domain values will be recorded as e(i, j).*value* (figure. 6(a)) and the violation numbers for all positions will be set to *zero* (figure 7(a)). After that, agents will be randomly placed into different rows. For instance, if agent $\mathbf{a_i}$ is placed at position (3, 1), the violation numbers in the environment will be updated accordingly, as shown in figure 6(b).
ERA was tested in several applications like n-queen problems and coloring problems and compared with earlier algorithms. Although successful, ERA suffers from a lack of explicit communication and cooperation mechanisms.

6. Conclusion

We presented an open community SMA architecture that allows receiving several types of agent societies. The objective is to develop resolution models for constraint satisfaction problems and optimization. Two communities were studied. The first, not described here, relates to the implementation of a deliberation process. The second implements a resolution using the coalition.
The coalition approach permits all agents to participate and to treat proposals, which guarantees to select admissible solution according to the Pareto's optimum. The negotiation provides an environment to obtain solutions by coalition and to avoid the problem's complexity.
An implementation of our approach based on BDI agents made possible to check certain assumptions and to adapt certain resolutions. Agents BDI constitute a favorable environment to express constraints and objectives widely.
Future work will relate to the development of the supervisor's role in the interpretation and the backtracking of certain solutions. We will lean towards the use of other interaction modes between the agents for resolution such as the concept of emergence or argumentation.

References

[1] Sechen C. Chip planning, total placement, and routing of macro/custom concealment integrated circuits using simulated annealing. In 25th Design Automation Conference, 1988, pp. 73-80.
[2] Petit T, Poder E. Global propagation of side constraints for solving over-constrained problems, Annals of Operations Research. 184(1), 2011, pp. 295-314.
[3] Sadgal M. Contribution to the problems of placement and routing. PhD Thesis, Claude-Bernard, University, Lyon 1, 1989.

[4] Sapena O, Onaindia E, Garrido A, Arangu M. A distributed CSP approach for collaborative planning systems. Original Research Article, Engineering Applications of Artificial Intelligence, 21(5), 2008, pp. 698-709
[5] Xidonas P, Ergazakis E, Ergazakis K, Metaxiotis K, Askounis D, Mavrotas G, Psarras J. On the selection of equity securities: An expert systems methodology and an application on the *Athens Stock Exchange*. Expert Systems with Applications. 36 (9), 2009, pp. 11966-11980.
[6] Hamadi Y, Jabbour S, Sais L. ManySAT a parallel SAT solver. In Journal on Satisfiability, Boolean Modeling and Computation, JSAT, IOS Press, 6(Spec. Issue on Parallel SAT), 2009, pp. 245-262.
[7] Hsairi L, Ghedira k, Alimi AM, BenAbdellhafid A. Argumentation Based Negotiation Framework for MAIS-E2 model. Chapter VI in book : Open Information Management: Applications of Interconnectivity and Collaboration, Tampere Univ, ISBN:978-1-60566-246, 2009.
[8] Baykan C, Fox M. Constraint satisfaction techniques for space planning. In Intelligent CAD Systems III - Practical Experiment and Evaluation, 1991, pp. 187-204.
[9] Golomb S, Baumert L. Backtrack programming. J. ACM, 12, 1965, pp. 516 - 524.
[10] Shahookar K., Mazunder P. Technical VLSI concealment placement.ACM Computing Surveys, 23 (2), 1991.
[11] Yokoo M. Algorithms for distributed constraint satisfaction problems:With review. Autonomous Agents & Sys Multi-Agent. 3, 2000, pp. 198-212.
[12] Mailler R, Lesser VR. Asynchronous Partial Overlay: A New Algorithm for Solving Distributed Constraint Satisfaction Problems. 25, 2006, pp. 529-576
[13] Modi PJ, Shen W, Tambe M, Yokoo M. An Asynchronous Supplement Method for Distributed Constraint Optimization, Proc.Autonomous Agents and Systems Multi-Agent, Melbourne, Australia, 2003, pp. 161-168.
[14] Georgé JP, Edmonds B, Glize P. Making Coil-Organizing Adaptive Multi-Agent Systems Work - Towards the engineering off emergent multi-agent systems, Methodologies and Software Engineering for Systems Agent, F. Bergenti, M-P.Gleizes, and F. Zambonelli, editors, Kluwer Publishing, 2004.
[15] Guerra-Hernández A, El Fallah-Seghrouchni A, Soldano H. Distributed Learning in Intentional BDI Systems Multi-Agent, in Proc. ENC, 2004, pp. 225-232.
[16] Pour HD, Nostary M. Solving the facility and layout and hiring problem by ant-colony optimization-meta heuristic. International Newspaper of Research Production. 44(23), 2004, pp. 5187-5196.
[17] Boussaa J, Sadgal M. A cognitive Agent for solving problems of occupation of space. Proceedings of the 3rd International Conference on Communications and information technology, December 29-31, 2009, Vouliagmeni, Athens, Greece, pp. 146-152.
[18] Lawson B, Park S. Asynchronous Time Evolution in year Artificial Society. Newspaper of Artificial Societies and Social Simulation. 3(1), 2000.
[19] Roy B, Bouyssou D. Multicriterion Assistance with the Decision:Methods and Case. Edition Economica. 1993.
[20] Jennings NR, Faratin P, Lomuscio AR, Parsons S, Wooldridge M, Sierra C. Automated negotiation: Prospective customers methods and challenges. Intl Newspaper of Group Decision and Negotiation, 10 (2), 2001, pp. 199-215.
[21] Rahwan I, Ramchurn SD, Jennings NR, Macburney P, Parsons S, Sonenberg L. Argumentation-based negotiation. 18 (4), 2003, pp. 343-375.
[22] Rubinstein A. Perfect equilibrium in The Knowledge Review Engineering,has bargaining model. Econometrica. 50, 1982, pp. 97-109.
[23] Faratin P, Sierra C, Jennings NR. Using similarity criteria to make trade-offs in automated negotiations. Artificial Intelligence, 142 (2), 2002, pp. 205-237.
[24] Amgoud L, Dimopoulos Y, et Moraitis P. With unified and general framework for argumentation-based negotiation. Proc.6th Intl. Joint Conf on Autonomous Agents and Multi-Agent Systems (AAMAS' 07), 2007, Hawaii.
[25] Caillou P, Aknine, S, Pinson S. How to Form and Restructure Multi-agent Coalitions. National Conference on Artificial Intelligence (AAAI 02) Workshop on Coalition Formation, Edmonton, Canada, AAAI Press, 2002, pp. 32-37.
[26] Picard G, Glize P. Model and Analysis of Local Decision Based on Cooperative Self-Organization for Problem Solving. Multiagent and Grid Systems (MAGS), 2(3), 2006, pp. 253-265.
[27] Liu J, Jing H, Tang YY. Multi-agent Oriented Constraint Satisfaction. Artificial Intelligence, 136(1), 2002, pp. 101-144

Jamila Boussaa : Is a Ph.D student in Computational Intelligence and Constraint Satisfaction Problems at Cadi Ayyad University. She received her B.A and DESS degree in computer science from Cadi Ayyad University in 2006. From 2007 to 2009, she was a a web technologie engeneer for SQL Group, and she is currently a trainer for banking system for HPS Solution

Mohammed Sadgal : Received the Ph.D degree in computer Science from the university of Lyon in 1989. He received the Ph.D degree in computer vision in 2005. He is currently a Professor (Since 2002) at Cadi Ayyad University (Marrakesh, Morocco).
From 1985 to 1987 he was Leader engineer for Net, CAD and CAM from 1987 to 1994 at CONCEPT Society (France). His research interests include Computer Vision, Artificial Intelligence and Multi-agent Systems.

Aziz Elfazziki : Received the Ph.D degree in computer Science from the university of Nancy in 1985. He received the Ph.D degree in Multi-agent Systems from Cadi Ayyad University.in 2002, His research interests include Information Systems and Multi-agent Systems.