



Self-Organizing Agents for Mechanical Synthesis

Davy Capera, Marie-Pierre Gleizes, Pierre Glize

► To cite this version:

Davy Capera, Marie-Pierre Gleizes, Pierre Glize. Self-Organizing Agents for Mechanical Synthesis. International Workshop on Engineering Self-Organising Applications (ESOA 2003), Jul 2003, Melbourne, Australia. pp.169-185, <10.1007/978-3-540-24701-2_12>. <hal-03818125>

HAL Id: hal-03818125

<https://hal.science/hal-03818125v1>

Submitted on 19 Oct 2022

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.



HAL Authorization

Self-Organizing Agents for Mechanical Design

Davy Capera, Marie-Pierre Gleizes, and Pierre Glize

IRIT – Université Paul Sabatier
118, Route de Narbonne, 31062 Toulouse, Cedex 4, France
+33 561 558 343
{capera,gleizes,glize}@irit.fr
<http://www.irit.fr/SMAC>

Abstract. This paper describes an automated process for designing mechanical systems based on the adaptive multi-agent system theory. At the beginning of the design process, the designer adds the elements of his problem: goal, envelope, constraints, known mechanical components ... During the solving process, mechanical components (previously “agentified”) organize themselves in order to find a solution, by modifying their parameters, by creating new mechanical components fitting in with the needs, or by modifying the system topology. While this paper presents an overview of the theoretical basis of AMAS theory, it primarily focuses on the method for developing the Mechanical Synthesis Solver and the results from the first prototype.

1 Problematics

Mechanical design consists in assembling mechanical components such as links (bars, rigid bodies) and joints (hinges, cams...), in order to build a system which performs an objective function like following a precise trajectory. The kinematic structure contains the essential information about which link is connected to which other links and by what types of joint. Moreover, designer may wish to satisfy additional constraints on available types of components: maximal number of components, mechanism weight, maximal dimensions, bounding envelope...

In the SYNAMEC¹ project (SYNthesis of Aeonautical MEchanisms), mechanical design is composed of two phases: preliminary design and detailed design. The preliminary design process -which is a pure kinematics study- is composed of three sub-phases:

1. Search for the topology of the mechanism (« type synthesis »), in other words establish how many rigid bodies are used in the system and in which way they are linked. At this level, the type of joints between bodies and their dimensions are not considered. Thus, criteria are related to more “abstract” considerations such as degrees of freedom of the mechanism and its dimensional space (planar mechanism or 3D one).

¹ SYNAMEC is a Growth European project involving SAMTECH (Belgium), ALA (Italy), CRANFIELD University (England), INTEC (Argentina), PAUL SABATIER University (France), SABCA (Belgium), SNECMA (France).

2. Search for the best alternative from the topology, in other words instantiate the linkages: for instance, a planar mechanism can only contain prismatic or rotoïdal joints.
3. Dimensional synthesis. In this phase, components dimensions must be adjusted to fit in with the given goal of the system (following a trajectory for example).

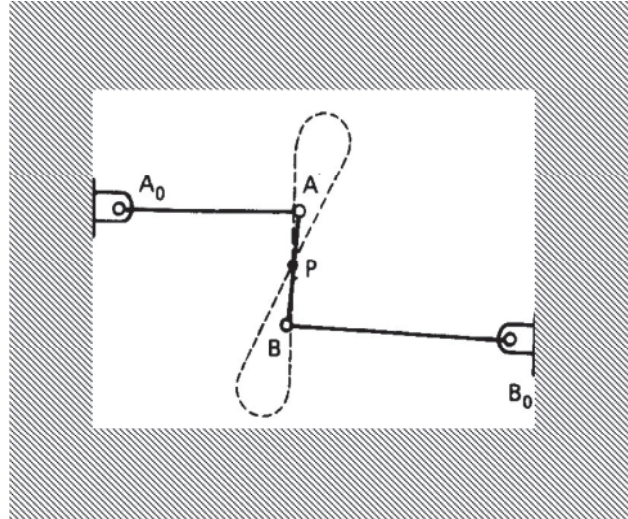


Fig. 1. Example of mechanism

Fig. 1. shows a mechanism example the designer has to provide. The mechanism, which is made up by tree bars and four hinges (A_0 , A , B , B_0), follows the dotted trajectory at the point P . Mechanism is contained in a bounding envelop on which the attachment points are (A_0 and B_0).

A mechanism is created by the engineer helped by his intuition, ingenuity and experience. He processes usually by trials and errors by using some computer aided design tools such as CatiaTM from Dassault Systemes² or Samcef³ Field MecanoTM. This process could be very time consuming and synthesis techniques are not very used in industry. After the designer completes the preliminary design, the resulting mechanism should be modified and even discarded at the sight of problems which occur during the next design phases. Thus, design is the creation of synthesized solutions in the form of products or systems that satisfy customer's requirements. Design is a continuous process of refining customer requirements into a final product. The process is iterative in nature and solutions are usually not unique. It involves a process of decision making. Recent methodological approaches based on systematic creation and classification of mechanisms to arrive at the best mechanism in shorter time are proposed by Tsai [18].

In order to solve this mechanical design problem, we propose a very different approach based on the Adaptive Multi-Agent Systems theory (AMAS theory). The

² http://www.3ds.com/en/brands/catia_ipf.asp

³ <http://www.samcef.com/>

principle is to solve the problem by a self-assembling process in which the mechanical components organize themselves from their interactions and the feedback of the environment. Mechanical components (rigid bodies and joints) are “agentified” and the mechanism builds itself by self-organization of its agentified components. The originality of this approach resides in the manner the re-organization rules work and in the fact that they are completely independent from the function the system has to achieve.

So, an automated tool for mechanical design task would be able to autonomously find mechanisms for the given goals and constraints. Nevertheless, this is a complex task because the combinatorial explosion between the basic mechanical components involves a huge space search. This complexity is increased when two other points are taken into consideration:

1. The designer cannot know the number and type of components the problem requires.
2. The topological modification of a mechanism leads to change drastically its global function: the solution space search is discontinuous and no relevant heuristic exists.

The second section of this paper presents briefly the AMAS theory which is the foundation of our way to handle adaptive systems. In the second part, we apply this theory to mechanical synthesis by explaining the overall process and the behaviour of the basic agentified components. The fourth section gives some results of this approach for XBars problems. We analyze these results and compare them with related works in the section five.

2 AMAS Theory

In the AMAS theory (6)], a multi-agent system is a system composed of autonomous entities interacting in a common environment. A Multi Agent System (MAS) has also an environment and it can reach a behavioral or a functional adequacy. For example, in a simulation, reaching a behavioral adequacy is to reproduce the behavior of the simulated entity; a functional adequacy is to perform the right task, the task for which the system had been built. We are specifically interested in Adaptive MAS. Classically such a system is defined by the fact that it is a MAS which is able to change its behavior to react to the evolution of its environment and has the following characteristics:

- the system is evolving in an environment with which it interacts,
- the system is composed of different parts: the agents,
- each agent has its own function to compute,
- the global function of the system is not implemented in the agents and there is no global control,
- the design is a bottom-up one: agents are firstly defined.

In our vision, the important notion is the collective; the AMAS theory must then lead to a coherent collective activity that realizes the right task. This property is called “functional adequacy” and the following theorem has been proved 13:

“For any functionally adequate system, there is at least a cooperative interior medium system which fulfills an equivalent function in the same environment”.

Therefore, we focused on the design of cooperative interior medium systems in which agents are in cooperative interactions.

The specificity of our AMAS theory resides in the fact that we do not code the global function of the system within an agent. Due to the agents' self-organization ability, the system is able to adapt itself and realizes a function that it is not coded in the agent, that is emerging and this is due in part to the interactions between components. If the organization between the agents changes, the function which is realized by the collective changes. Each agent possesses the ability of self-organization i.e. the capacity to locally rearrange its interactions with others depending on the individual task it has to solve. Changing the interactions between agents can indeed lead to a change at the global level and this induces the modification of the global function. This capacity of self-organization at the lowest level enables to change the global function without coding this modification at the upper level of the system and so the system adapts itself.

Self-organization is founded on the capacity an agent possesses to be locally “cooperative”, this does not mean that it is always helping the other ones or that it is altruistic but only that it is able to recognize cooperation failures called “Non Cooperative Situations” (NCS, which could be related to exceptions in classical programs) and to treat them. The local treatment of NCS is a means to build a system that does the best it can when a difficulty is encountered. Such a difficulty is primarily due to the dynamical nature of the environment of the system, as well as the dynamics of the interactions between agents. More precisely an agent can detect three kinds of NCS:

1. when a signal perceived from its environment is not understood and not read without ambiguity;
2. when the information perceived does not induce the agent to an activity process;
3. when concluding results lead to act in a useless way in the environment.

An AMAS-compliant system is emergent because its global function is not coded within the agents. Like for natural systems, the adaptation occurs when the system is functioning in a given environment. At the conception phase, designers have mainly to fill in a nutshell associated to each generic class of agents with its cooperative behavior as described above.

3 Mechanical Synthesis Solver

The development of the software to generate mechanisms has begun with the University of CRANFIELD for mechanical advices and SAMTECH for the kinematic simulation engine called MECANO. In this section, we present the global functioning of the system and the mechanical agents specification.

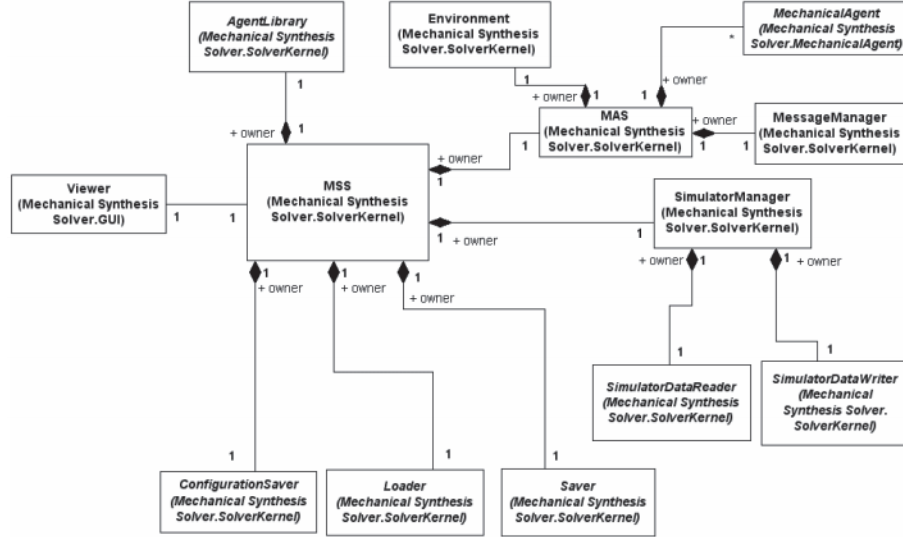


Fig. 2. General class diagram of the *Mechanical Synthesis Solver*

3.1 The Process

The mechanism moves in a specific environment, generally defined by a trajectory, an envelope and a movement control. Thus, the multi-agent technique needs a simulation engine which is able to compute mechanical movements (interactions between components, collision with envelope) and to return information about the state of each component (position, forces and loads, for example) to the multi-agent system. During the analysis phase of this problem, an agreement between partners has been reached to use MECANO as the environment engine. Thus, the software learns a relevant mechanism by using a loop composed of the cycle: MECANO simulation followed by the agents self-organization (Figure 4). This is the basic learning evolutionary process of the mechanism. Each cycle is decomposed into three phases:

1. Simulation engine computes the motion of the current mechanism.
2. Data related to the new mechanism state are sent to the mechanical agent in order to update them.
3. The AMAS tool performs “optimization” –resolving non cooperative situations detected by the mechanical agents compliantly the AMAS theory- that leads to a new mechanism.

The interface with the simulator is performed through files which describe, in input, characteristics of the mechanism, the environment and the commands related to the wished motion (the “actuation function”), and allow to recover the new mechanism characteristics after the motion performed by MECANO.

Figure 2 shows the global class diagram of the Mechanical Synthesis Solver specified with UML (Unified Modeling Language) notation.

Design of the multi-agent system was performed in three steps (see 3.2 for details):

1. Identification of agents: mechanical components are represented by agents.
2. Specification of non cooperative situations and related actions: this work was performed in collaboration with the CRANFIELD University.
3. Implementation of agents from a generic model.

These development steps are compliant with the methodology of AMAS-based systems design called ADELFE (French acronym for “Atelier pour le Developpement de Logiciels à Fonctionnalité Emergente”) which is currently under development ([3]).

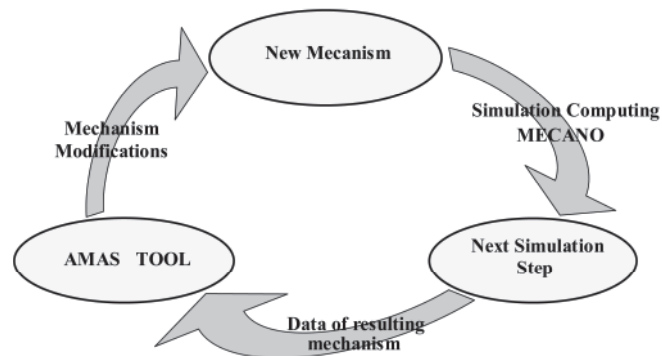


Fig. 3. Adaptation process for type synthesis of mechanisms

3.2 The Mechanical Agents

Following the ADELFE methodology, the design of *Mechanical Synthesis Solver* has focused on two main steps: the identification of agents and the specification of the non cooperative situations and related actions for each type of agent.

3.2.1 Identification of Agents

Every basic mechanical component of the component library (chosen from the MECANO element library) will be an autonomous entity which has to collectively self-organize. Each autonomous agent is able to manage its own mechanical-related characteristics (for instance, length for a bar) and has some agent-related capabilities (communication, memorization, believes management...). The prototype which is currently developed includes the following components as agents: hinges, bars, attachment points and trajectory points.

A trajectory point agent is an exception: it does not represent any mechanical component. Its goal is to fit with the trajectory. It can perceive the distance between its position and the goal trajectory and interprets any situation for which this distance is different from zero as a non cooperative situation. A trajectory point agent can fix itself on rigid bodies and define its own attachment position on the rigid body.

3.2.2 Non Cooperative Situations and Related Actions

As explained in section 2, a designer of an AMAS-based system has to exhaustively specify the non cooperative situations which may occur for every agent. A mechanical expert advice is necessary for the definition of these non cooperative situations which lead the self-organizing process. They are situations in which an agent decides to change its internal parameters (adjustments) and/or its place in the current mechanism (topology transformation). The detection of these situations has to be only performed on the local available information of an agent.

This work was performed in collaboration with mechanical experts from CRANFIELD University ([7]). All situations are derived from the cooperation definition given in section 2, and may be classified in three types: incomprehension, incompetence and uselessness ([13]).

As agents share the same application field and a common language, thus ambiguity or misunderstood situations cannot occur (see case one in the AMAS theory part). The following table (Table 1) is an example of a uselessness situation for a bar agent (case three in the AMAS theory):

Table 1. Uselessness situation for bar a agent

Name	Uselessness
Description	The agent is totally useless in the mechanism.
Condition(s)	The agent is linked with none other agent.
Action(s)	Find a link with a join agent Create a new agent join agent Suicide

Obviously, when a mechanical component is separated from the global mechanism it is totally useless. In this case, its corresponding agent tries to integrate it or, when failing, it disappears. This situation is defined in the same way for other agents but actions “links” and “creations” can only concern “rigid body agents” (such as bar agents).

Some situations are related to the goal and can be linked with incompetence situations. Here is an example of such a situation for a bar agent:

Table 2. Goal incompetence situation for a bar agent

Name	Goal incompetence
Description	Direct perception of a stiffness related to a difference between ideal and real trajectory
Condition(s)	The value of “goal stiffness” is not nil
Action(s)	Change position of the bar extremities Break a link with one of its adjacent agents

In this case, the agent perceives local information (the stiffness see §5.1) indicating that the environment is not satisfied by the behaviour of the system. At this stage, the agent must decide if this feedback is related to its own activity by taking into account his beliefs and memory about previous interactions. This feedback allows also the agent to adjust its beliefs about the relevance of its actions and the other known agents.

Observing the corpus of agent actions, one can see that self-organizing approach for mechanical design allows discovering topologies from scratch: at the beginning, the mechanism can be empty. The self-organization process is composed of four phases:

1. Component apparition. When the current topology is sub-optimal (for example there exists some residual stiffness for the trajectory point), a new component is added from the set of generic agents.
2. Component reorganization. When a given component of the current alternative is frequently in uncooperative situation, it tries to modify its location (for example a bar links with another component of the system).
3. Component adjustment. When uncooperative situation occurs in a component, it tries to self-adjust in a better way (for example a bar can modify its length). Thus, the global system finds the better response for a given alternative.
4. Component disappearance. When a component had a great amount of uncooperative situations, he “thinks” to be useless. The component deletion occurs only after trying to find a relevant location inside the mechanism.

Thus, the core process of self-organization is deciding if the current alternative is relevant to achieve the goal of the mechanism. In this paper, results of this core work from two usual examples in mechanical design, the XBars problem, are described.

4 Results

The first developed version is a prototype with reduced functionalities: it only focuses on the basic “X-bar” problems solving. These problems consist in finding mechanisms whose one point must follow a given trajectory. These mechanisms are composed of basic components such as bars, hinges and attachment points (to fix the mechanism on the envelope). So, the agents that are available in the prototype are related to these three mechanical components. They are also linked to an agent “trajectory point” which represents the point that must follow the trajectory defined by the user.

For this first prototype, we only test the dimensions adjustment problem which takes place with the third sub phase of the preliminary design: the mechanism has to adapt the length of the bars in order to the point fits with the given trajectory.

4.1 3-Bars Problem

The 3-Bar problem consists in a mechanism composed by three bars, four hinges and two attachment points. The goal trajectory is a vertical straight line.

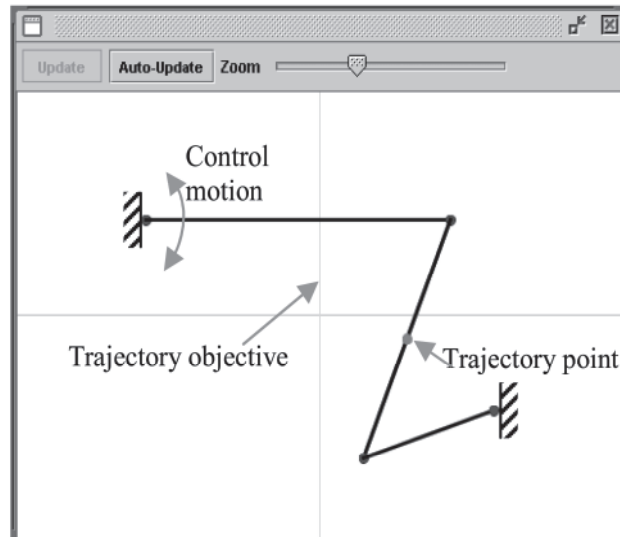


Fig. 4. Initial state of a bad-dimensioned 3-Bar mechanism

The Figure 4 shows an initial state –with bad dimensions- of a 3-Bar mechanism (the centered straight line is the trajectory, and the trajectory point is on the middle of the central bar):

The actuation is a rotation on the upper left hinge: it performs cyclically a wipe of 0.4 radians by steps of 0.05 radians. A cycle is the number of steps motion the system realizes in order to find the same position. In this case a cycle is composed of 16 steps

In figure 5, the curve measures at each step the error (abscissa) of the mechanism, i.e. the distance between the point and the trajectory (ordinate). This figure shows that the mechanism adjusts itself quickly (less than one cycle) and reaches a configuration in which the distance is lower than 0.1.

In this simulation, the process runs 48 steps that correspond to 4 cycles. After the great changes obtained during the first steps, the adaptation very slightly improves the global behavior.

4.2 5-Bars Problem

The 5-Bar problem consists of a mechanism composed of five bars, four hinges and two attachment points. The goal trajectory is a horizontal straight line.

The picture below shows an initial state –with bad dimensions- of a 5-Bar mechanism (the straight line at the bottom of the figure represents the trajectory and the trajectory point is at the bottom of the triangular structure):

The actuation is a rotation on the upper left hinge: it cyclically performs a wipe of 0.4 radians by steps of 0.05 radians. Here, there also are 16 steps for a cycle.

In the figure 7, the abscissa and ordinate have the same meaning as in the figure 5. The figure shows that the mechanism converges toward a solution and that the distance between the trajectory and the point decreases. However, solving is slower

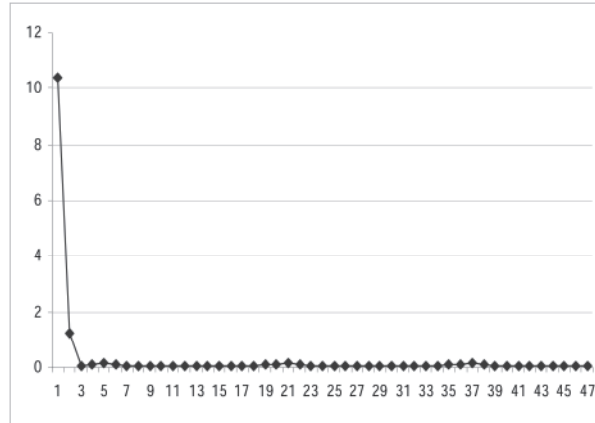


Fig. 5. Distance to the trajectory (ordinate) of a 3-Bar mechanism as the function of step number (abscissa)

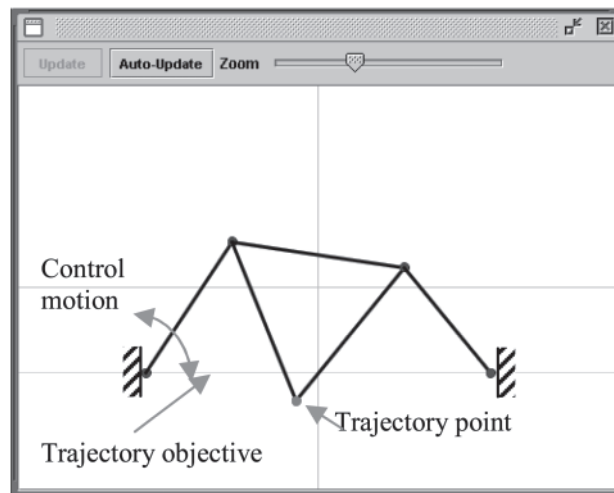


Fig. 6. Initial state of a bad-dimensioned 3-Bar mechanism

than for the 3-Bars, because the maximum distance during the 40th cycle is still about 0.12. This comes from three reasons:

1. A 5-Bars mechanism contains more components than a 3-Bars one. This gives more fuzzy information to local agents and leads to adjustments that are not always relevant.
2. Stiffness (see 5.1) comes from geometry (and not cinematic) analysis of the mechanism. Thus, the derived agent adaptation is sometimes relatively poor.
3. All cooperative behaviors are not yet implemented inside agents. This certainly reduces the adaptation efficiency for the more complex mechanisms.

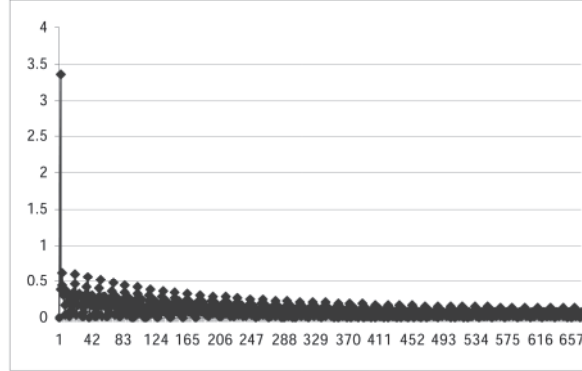


Fig. 7. Distance to the trajectory (ordinate) of a 5-Bar mechanism as the function of step number (abscissa)

4.3 Further Developments

The results are satisfying and the further objectives are foreseen as:

- Implementation of the full self-organizing process for agents: adding non cooperative situations and related actions to process topological modifications and to allow adding/deletion of agents. The specification has already been performed with experts from CRANFIELD University ([7]).
- Definition of new types of agent to complete the set of available agents: the specification has already been performed with experts from CRANFIELD University ([7]).

The adjustments-related actions will be improved in order to obtain a quicker convergence time. This improvement will be at the agent behavior level and more precisely in the action module. Indeed, the current data processing that calculates the modifications is not optimal and can be improved: for instance, by adding a memory of the previous actions an agent performed in order to take them into account.

5 Analysis and Comparison

The results given by the MSS for the 3-Bars and the 5-Bars problems are similar to those computed directly (without learning) by MECANO when we give the theoretical dimensions. This indicates the relevance and efficiency of the local adaptation process derived from the AMAS theory.

5.1 Emergence

In the section 2, we claim that systems design following AMAS techniques provides solutions that are emergent. Actually, the mechanical agents behavior does not contain any part which directly processes the “distance to trajectory” decrease. Fig. 8.

shows correlation between the local “non cooperation” measure of agents and the distance to the trajectory. The top dots represent the average distance to the trajectory during a complete cycle (16 steps) and the lower ones represent the average non cooperation degree –or stiffness- of the whole system (the sum of non cooperation degrees of all the agents).

The stiffness depends on the distance error between the trajectory point and the trajectory objective. This global information is the feedback of the environment (returned by the MECANO tool) but unknown by the agents. Nevertheless the geometric approach computed by MECANO allows giving a local stiffness associated to each agent. This local stiffness is neither exactly informative to the magnitude of adjustment to apply nor its direction. This information is derived from the cooperative learning process of the agents.

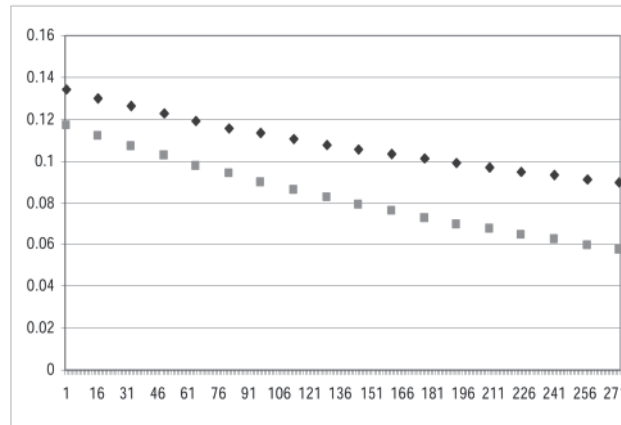


Fig. 8. Distance to the trajectory (up) and stiffness

There is a strong correlation between the local and global criteria of non cooperation (distance to the trajectory is considered as the non cooperation measure of the whole system). That experimentally shows that the local reasoning of the parts of a system leads to a global emergent behavior.

5.2 Related Works in Mechanical Field

There is very few works on preliminary design problematics in literature; besides the A-Design application which is based on a global evolutionary algorithm and a mechanism synthesis method develop by Lung-Weng Tsai, only tools which support designers exist (DAO, CAO, simulators ...).

5.2.1 A-Design

This tool was developed by Campbell, Cagan and Kotovsky ([4]) from Carnegie Mellon University. The solving process is a classical evolutionary one whose selection, generation and reproduction operations are performed by multi-expert

systems based on agent principle (one agent for each expert module) and blackboard architecture. Authors argue in 4]: *“This system differs from previous work in that 1) designs are not only generated but iteratively improved upon to meet objectives specified by a designer, 2) synthesis of designs is performed within a rich descriptive representation of components and configurations that models real-world component interactions, and 3) key design alternatives are retained to allow the system the flexibility to adjust to changes in the problem description made by the designer throughout the design process.”*

However, this method is very time and memory consuming because the system has to handle a set of mechanisms whereas In the Mechanical Synthesis Solver, we only deal with one mechanism which adapts. Moreover, modification and evaluation operations are complex to develop and are related to the problem the system has to solve while mechanical agents of the Mechanical Synthesis Solver always have the same behavior which is related to their function and to the non cooperative situations independently of the problem. As the solving process is based on the global evaluation of the mechanism, this method clashes with the need to deal with a discontinuous space search. AMAS theory provides a solution to develop systems which are able to avoid this clash because the global solution is not known by the agents and emerges.

5.2.2 Mechanism Synthesis Method

Recently a new approach based on an abstract representation of the kinematic structure has evolved, it is extensively developed in the book of Lung-Weng Tsai *“The kinematic structure contains the essential information about which link is connected to which other links by what type of joint. It can be conveniently represented by a graph and the graph can be enumerated systematically using combinatorial analysis and computer algorithms”* 18]. This methodology is composed of three main steps:

1. Enumeration of all possible solutions (according some functional requirements) using graph theory and combinatorial analysis.
2. Creation of a class of feasible mechanisms based to the remaining functional requirements.
3. Iteration on candidates in order to find the most promising for the product design.
4. This methodology has been successfully applied in the structure synthesis of many domains. In the Synamec project, Alberto Cardona ([8]) is working on a tool based this methodology. This software will use genetic algorithm to evaluate candidate mechanisms.

As in the MSS tool, this approach based on kinematic structure needs to evaluate a given mechanism at a time. Nevertheless the evaluation engine on MSS is exactly the same we use to transform the current mechanism in order to find a more relevant, and theoretically all the functional requirements could be considered for this evaluation. We think that these considerations increase the search efficiency and thus reduce the time to conception.

5.3 Works Learning on Self-Organization Using Cooperation

Hogg and Huberman in [16] showed that when agents cooperate in a distributed search problem, they can solve it faster than any agent working in isolation. A similar result was obtained by Mataric in [19] with a set of mobile robots foraging in order to bring back tiles to "home". She has observed that when the number of individualist robots increases, the global performance decreases due to the interfering activities. For her, the ideal result will be obtained with robots having altruistic behaviors.

Steels in [23] have taken an interest in self-organization in the case of two applications. The first concerns the foraging of a geographical zone by several robots. The self-organizing mechanism is similar to the one made by ants when they try to find food a depositing of crumbs simulates the pheromone depositing, which guides other robots. The second application concerns the autonomous elaboration of a vocabulary by several agents present in a given environment. Each agent possesses for that its own set of associations word/meaning. So it can experiment associations used by other agents present in the system according to the following 3 manners: either by propagating the association it possesses, or by creating a new association or by self-organization (based on retroaction mechanism and allowing the agent to bring up to date the confidence it has in its association).

Cooperation was extensively studied in computer science by Axelrod [2] and Huberman [17] for instance. From the initial studies about the condition of cooperation emergence in societies, the question was now extended to the more general problem of the emergence of norms (see Flentge, Polani and Uthmann [11] on this question). "Everybody will agree that co-operation is in general advantageous for the group of co-operators as a whole, even though it may curb some individual's freedom." (Heylighen [15]). Relevant bio-inspired approaches using cooperation are the swarm algorithms (Kennedy and Eberhart [26]) and the ants algorithms (Dorigo and Di Caro [9]) which give efficient results in many domains. We have shown in [24] that natural ants have a behaviour very closed (but not perfectly) to the cooperative one we gave in the AMAS theory.

Multi-agent learning relies on, or even requires, the presence of multiple agents and their interactions. Many authors in this domain (Goldman [14], Sekaran [21], Sen [22], Weiß [24]) have studied the role of social behavior of agents on the global performance. They found that cooperation between agents improves the results. If we consider each agent of the system as a piece of knowledge, these works mean that knowledge is well learnt when it is organized in a cooperative manner. This is a criterion independent of the meaning (the semantic needed for common knowledge), and thus could be a good approach for a general learning theory based on cooperation.

6 Conclusion

This paper presents a new approach to solve autonomously the preliminary synthesis problem. The main principle of Mechanical Synthesis Solver -based on the Adaptive Multi-Agent System theory- is to "agentify" mechanical components and gives them

a cooperative behavior: they are able to locally detect non cooperative situations and to solve them. We also present results from a prototype of this software which lead us to claim that this approach is relevant.

The AMAS theory and its application to mechanical design is a mix between two main approaches:

1. Works on problem solving based on multi-agent systems where agents have a great autonomy in order to adapt their behaviour according to their local situations. This idea is closed to ants algorithms [9] or particle swarm optimization [26] in adding explicit agent cooperation reasoning to guarantee a coherent space search process.
2. Works directed by fitness decision like genetic algorithms (see A-Design in §5.2.1) and derived algorithms based on evolution of population of “solutions” and using a fitness function (particle swarm optimization algorithms [26] and other evolutionary algorithms using implicit or explicit operators). Nevertheless our fitness criterion is not directly based on global function evaluation, but a local one for each agent and its cooperative behaviour with its neighbourhood).

Further development will focus on the improvement and completion of agent behaviors in order to perform autonomously the type synthesis phase of design.

In order to test a theory (by falsification or validation) a great amount of experiments must be done in various fields. For this reason, we have implemented our theory on cooperative self-organization on many applications:

1. The first was the tileworld game (Piquemal 20]) in which we have experimentally verified that cooperative agents have better results than selfish ones.
2. The second concerns an application of cooperative information systems with France-Telecom (Camps 5]) and with Deutsch Telekom (Athanassiou 1]). In this software, the agents representing the users and the services create a dynamic network of mutual interest based on the cooperative self-organization process.
3. The third is about a national multi-disciplinary project about natural and artificial collective intelligence. The results of a cooperative self-organized ants society application gives performances at least better than natural simulated insects (Topin [24]).
4. The fourth is a real-time application for flood forecast (Georgé 12]). This software runs in the organism in charge of crisis situations in the Midi-Pyrénées region of France and depending of the Ministry of the environment.

Usual search algorithms (stochastic or determinist), which impose the knowledge of a cost function have the same global efficiency when the corpus of examples are sufficiently large. *“In our investigation of the search problem from this match-f-to-a perspective, the first question we addressed was whether it may be that some algorithm A performs better than B, on average. Our answer to this question, given by the NFL theorem is that this is impossible”* 10]. Surprisingly, we have observed (using the same algorithm pattern derived from AMAS theory) the same performances in all the applications cited previously. An explanation of this fact (not a demonstration) could be the inexistence of global cost functions for theories

allowing the emergence of the global function: the global function is unknown and obviously we cannot have an associated cost function. This could be a reason to investigate massively the field on really emergent algorithms. This reflexion is based on experiences where:

1. We can know, as an observer, what is the global optimum. For example in the X-bar problems, there are optimal known solutions and we can judge the result given by the system.
2. Though the search space has a huge amount of local minima, the system avoids them. We have observed this in applications such as flood forecast, ant foraging, time tabling...

References

- [1] Athanassiou E., Léger A., Gleizes M.P., Glize P. - Abrose : Adaptive Brokerage Based on Self-Organisation Services and Users – Short paper on « European Conference on Modelling Autonomous Agents in Multi-Agent World » - 1999.
- [2] Axelrod R. 1984. The Evolution of Cooperation, Basic Books, New York.
- [3] Bernon C., Gleizes M.P., Peyruqueou S., Picard G. – ADELFE, a Methodology for Adaptive Multi-Agent Systems Engineering – In Third International Workshop "Engineering Societies in the Agents World" (ESAW-2002), 16-17 September 2002, Madrid.
- [4] Campbell, Cagan & Kotovsky – Agent-based Synthesis of electro-mechanical design configurations – In Proceedings of DETC98 1998 ASME Design Engineering Technical Conferences September 13-16, 1998, Atlanta, GA.
- [5] Camps V., Gleizes M.P. - Cooperative and mobile agents to find relevant information in a distributed resources network, Workshop on Artificial Intelligence-based tools to help W3 users, Fifth international conference on World Wide Web – 1996.
- [6] Capera D., Georgé J.P., Gleizes M.P., Glize P. - Emergence of organisations, emergence of functions – AISB 2003.
- [7] Capera D.: Integrated Expert Advisor tool – SYNAMEC deliverable, January 2003.
- [8] Cardona A. – Computational Methods for Synthesis of Mechanisms – 10-02-2002.
- [9] Dorigo M. and Di Caro G.. The Ant Colony Optimization Meta-Heuristic – In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization - McGraw-Hill, 1999.
- [10] David H. Wolpert and William G. Macready - No free lunch theorems for search - Tech. Rep. SFI-TR-95-02-010, Santa Fe Institute – 1995.
- [11] Flentge F., Polani D., Uthmann T. - On the Emergence of Possession Norms in Agent Societies - Journal of Artificial Societies and Social Simulation vol. 4, no. 4 <http://www.soc.surrey.ac.uk/JASSS/4/4/3.html> - 2001.
- [12] Georgé J.P., Gleizes M.P., Pierre Glize, Régis C. – Real-time Simulation for Flood Forecast: an Adaptive Multi-Agent System STAFF – In Proceedings of the AISB'03 symposium on Adaptive Agents and Multi-Agent Systems, University of Wales - 2003.
- [13] Gleizes M.P., Camps V., Glize P. – A Theory of emergent computation based on cooperative self-organization for adaptive artificial systems – In Fourth European Congress of Systems Science, Valencia, 1999.
- [14] Goldman C.V., Rosenschein J.S. 1994. Emergent Coordination through the Use of Cooperative State-Changing Rule – AAAI.
- [15] Heylighen f. (1992): Evolution, Selfishness and Cooperation; Selfish Memes and the Evolution of Cooperation, Journal of Ideas, Vol 2, # 4, pp 70-84.

- [16] Hogg T., Huberman B.A. 1992. Better than the best: The power of cooperation - Lectures notes in complex systems - Addison Wesley.
- [17] Huberman B.A. 1991. The performance of cooperative processes - In emergent computation, Edited by Stephanie Forrest - Special issue of Physica D.
- [18] Lung-Weng Tsai: Mechanism design: Enumeraion of kinematic structures according to function. CRC Press, ISBN: 0-8493-0901-8.
- [19] Mataric Maja J. 1994. Interaction and Intelligent Behavior PHD of Philosophy Massachussets Institute of Technology May 1994.
- [20] Piquemal-Baluard C., Camps V., Gleizes M.P., Glize P. - Cooperative agents to improve adaptivity of multi-agent systems, Intelligent Agent Workshop of the British Computer Society, In Specialist Interest Group on Expert Systems & Representation and Reasoning – 1995.
- [21] Sekaran M., Sen S. 1995. To help or not to help - Seventeenth Annual Cognitive Sciences Conference - Pitsburg Pennsylvannia.
- [22] Sen S., Sekaran M. 1995 Using reciprocity to adapt to others – IJCAI.
- [23] Steels L. 1996 . The spontaneous Self-organization of an Adaptive Language, Machine Intelligence 15 - Oxford University Press - Oxford - Muggleton S. (Ed.) <http://arti.vub.ac.be/www/steels/mi15.ps> - 1996.
- [24] Topin X., Fourcassié V., Gleizes M.P., Régis C., Théraulaz G., Glize P. - Theories and experiments on emergent behaviour : From natural to artificial systems and back – ECCS 1999.
- [25] Weiß G. 1993. Learning To Coordinate Actions In Multi-Agent Systems - in Proceedings of the International Joint Conference on Artificial Intelligence.
- [26] Kennedy, J. and Eberhart, R. - Particle Swarm Optimization -IEEE International Conference on Neural Networks (Perth, Australia), IEEE Service Center, 1995.