14th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 24-26, 2018

Multi-Agent Cognitive System for Optimal Solution Search

Ababii Victor, Sudacevschi Viorica, Munteanu Silvia

Informatics and Systems Engineering Department
Technical University of Moldova
Chisinau, Republic of Moldova
Victor.ababii@calc.utm.md,
Viorica.sudacevschi@calc.utm.md,
Silvia.munteanu@calc.utm.md

Bordian Dimitrie, Calugari Dmitri, Nistiriuc Ana, Dilevschi Sergiu

Computer Sciences, Electronics and Communications Doctoral Schools
Technical University of Moldova
Chisinau, Republic of Moldova
Dimitrie.bordian@scec.utm.md,
Dmitri.calugari@scec.utm.md

Abstract— **Solving optimization problems is a challenging issue in almost all engineering applications. In this paper we present the design of a multi-agent system with cognitive properties meant to find optimal solution according to the game theory. The multi-agent system consists of two subsystems that compete for maximum profit for their teams. Subsystems have no communication capabilities with each other. The status of an agent can only be identified based on its actions on the environment. In order to find an optimal solution we use a Nash Equilibrium concept of a non-cooperative game. The implementation of the system is done on a microcontroller. The evaluation of the Nash Equilibrium convergence rate was performed for 4 cases. The results are presented in the form of convergence charts.**

Keywords—multi-agent system; cognitive systems; optimal solutions, Nash Equilibrium.

I. INTRODUCTION

Multi-Agent Systems (MAS) are collaborative intelligent systems composed of a number of interacting computational entities capable of solving complex problems based on minimal or reduced data processing resources. These systems consist of a number of intelligent software or hardware agents, homogeneous or heterogeneous, which exchange information and knowledge, have the ability to co-operate, coordinate and negotiate their activities [1, 2, 3, 4].

MAS are becoming more and more popular in such areas as economy, technology, mathematics, computing, networking, artificial intelligence, robotics, collaborative decision support systems, data mining, social science, etc. A MAS is a group of autonomous, interacting agent sharing a common environment, which they perceive with sensors and upon which they act with actuators. From a structural point of view, MAS represents a subclass of distributed, parallel or concurrent processing systems [5, 6], where the agents have autonomy (their own interests and knowledge). In these systems synchronization and coordination are done at the common decision level [7].

A particular interest in the MAS development is based on the use of two types of agents: agents who are directing their activities to solve the issue positively (positive agents) and agents who are opposed to solve the problem positively (negative agents). The evolution of these systems is based on game theory, in particular, on the concept of Nash Equilibrium. [8,9]. Nash Equilibrium is considered one of the most important concepts of game theory, which attempts to determine mathematically and logically the actions that participants of a game should take to secure the best outcomes for themselves.

The ability and correctness of solving a problem for a multi-agent system is determined by its cognitive capabilities [10]. Cognitive systems have the ability to build strategies based on relevant hypotheses and contextual data, providing information and knowledge to the agents. Collective intelligence is the most important effect of sharing data and information. Over time, it has been found that cognitive systems are one of the most important steps in the long process of adapting to the evolution of information technology [11].

The article is structured as follows. The mathematical model of the optimal solution search in multi-agents systems is described in section 2. The description of the multi-agent system and cognitive agent synthesis is presented in detail in section 3. The example of an agent design on a micro-controller is shown in section 4. Convergence rate analysis for four cases is presented in section 5. A brief summary and an outlook on future work is offered in section 6.

II. MATHEMATICAL MODEL

Let's consider the process $\boldsymbol{S} \subset \boldsymbol{R}^N$ defined in an $\boldsymbol{N}$-dimensional space. The dynamics of the process $\boldsymbol{S}$ is

978-1-5386-1495-2/18/$31.00

determined by the set of functions $F = \{f_1, f_2, ..., f_K\}$, where $F : S \to R^N$. The state of the process S is determined by the state vector $X = \{x_1, x_2, ..., x_N\}$ [12, 13].

The set of functions F act on the process S according to the following system of equations (1):

$$\begin{cases} f_1(x_1, x_2, ..., x_N) = 0 \\ f_2(x_1, x_2, ..., x_N) = 0 \\ ... \\ f_K(x_1, x_2, ..., x_N) = 0 \end{cases} \quad (1)$$

Based on the model (1), can be defined the search criteria for optimal solutions $\mathbf{min/max}$ (2):

$$\begin{cases} S = \sum_{j=1}^{K_{\min}} f_j^2(x_1, x_2, ..., x_N) \to S_{\min} \\ S = \sum_{i=1}^{K_{\max}} f_i^2(x_1, x_2, ..., x_N) \to S_{\max} \end{cases} \quad (2)$$

where: $S_{\min}$ - the minimal solution for the process S; $S_{\max}$ - the maximal solution for the process S; $K_{\min}$ - the number of functions that minimize the process S; $K_{\max}$ - the number of functions that maximize the process S, $K_{\min} \cap K_{\max} = \varnothing$.

III. Cognitive Agents Sinthesis

To solve the model (2) the Muti-Agent System consisting of two subsets of agents is proposed (Figure 1): The set of ***Positive Agents*** which solve the condition $S \to S_{\min}$, and the set of ***Negative Agents*** which solve the condition $S \to S_{\max}$. Positive agents generate control signals $A_{\min} = \{y_j, \forall j = \overline{1, K_{\min}}\}$, while negative agents generate control signals $A_{\max} = \{y_i, \forall i = \overline{1, K_{\max}}\}$. The state of the process S is identified by reading state signals $ES = \{x_n, \forall n = \overline{1, N}\}$.

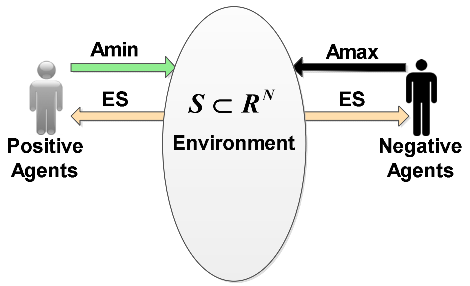

Fig. 1. Multi-Agent System

Each agent has information and knowledge that he assesses over time. The direct exchange of information between agents does not take place, the status of the positive and negative agents is determined by their actions carried out in the process. These criteria defined for agent behavior are part of the game theory and correspond to the Nash Equilibrium concept [9].

The behavioral model of the cognitive agent is presented in Figure 2, where: X_k - state vector of the process S; ADC - analog-to-digital converter; $RAM\ ES$ - state memory of the process; $\mathbf{Pr}$ - processor; $Rg\ Act$ - register that contains the binary code of the control signal applied to the process; DAC - digital-to-analog converter; y_k - analog control signal; $ROM\ Instr$ - program memory; $Cogn\ RAM$ - cognitive memory; $C[T-1]$ - cognitive model used to control the process at the time $T-1$; and $C[T]$ - the cognitive model calculated for the next step.

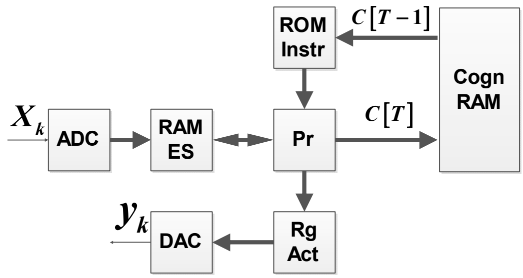

Fig. 2. Behavioral model of the cognitive agent

The agent's activity begins with the acquisition of the state vector X_k of the process S, converting it into digital code and store it in memory $RAM\ ES$. The processor $\mathbf{Pr}$ executes the code sequence from the program memory $ROM\ Instr$, which solves the model (1) with respect to the conditions (2). The code sequence executed by the processor $\mathbf{Pr}$ is determined by the content of the memory $Cogn\ RAM$, which changes its state at each iteration of the command as a result of the convergence rate analysis of the conditions (2). The binary results of the calculation necessary to control the process are written in the register $Rg\ Act$ and converted by DAC to analog control signal y_k.

Knowledge building stages are presented in Figure 3, where: X - the state of the process S; X/D - the conversion of the signal X to binary code ***Data*** acording to the knowledge model $C(X/D)$; D/I - conversion of the binary code ***Data*** to ***Information*** according to the knowledge model $C(D/I)$; I/C - conversion of the ***Information*** to ***Knowledge*** according to the knowledge model $C(I/C)$; I/D - conversion of the ***Information*** to ***Decision*** according to the knowledge model $C(I/D)$; y - control signal applied to the process S; ***Knowledge*[T − 1]** - knowledge models for conversion of data, information and knowledge; ***Knowledge*[T]** - new knowledge, generated according to the previous information and knowledge. Knowledge represent methods and models used to process data. Data processing algorithms used by agents in their strategies are based on knowledge.