

COGNITIVE MULTI-AGENT SYSTEM FOR SEARCHING OPTIMAL SOLUTIONS

VICTOR ABABII, SUDACEVSCHI VIORICA, MUNTEAN SILVIA, LAZU VICTORIA

Technical University of Moldova

Abstract: This article proposes the development of a multi-agent system with cognitive properties designed to find optimal solutions in the work environment. The Multi-Agent system consists of two agent sub-agents that contribute to maximum profits for their team. Aggregation subsystems are devoid of functions the communication capabilities. The status of an agent can only be identified in terms of his actions on the business environment. The process of finding an optimal solution is to solve the Nash equilibrium condition. The operating mode of the system is based on the structure diagram, the cognitive agent model and the knowledge and knowledge building diagram. The implementation is intended to use MCUs. Evaluation of the Nash equilibrium conversion rate was performed for 4 cases: one agent in each subunit, 5 agents in each subunit, 10 to 5 and 5 to 10 agents in each subset. The results are presented in the form of convergence graphs.

Key words: Multi-agent system; cognitive system, optimal solution, Nash equilibrium.

Résumé:

Cet article propose le développement d'un système multi-agent avec des propriétés cognitives conçues pour trouver des solutions optimales dans l'environnement de travail. Le système Multi-Agent se compose de deux sous-agents d'agents qui concourent à obtenir un maximum de profits pour leur équipe. Les sous-systèmes d'agrégation manquent de capacités de communication. Le statut d'un agent ne peut être identifié qu'en fonction de ses actions sur l'environnement d'activité. Le processus de recherche d'une solution optimale consiste à résoudre la condition d'équilibre de Nash. Le mode de fonctionnement du système est basé sur le diagramme de structure, le modèle d'agent cognitif et le diagramme de construction des connaissances et des connaissances. La mise en œuvre est destinée à utiliser des MCU. L'évaluation du taux de conversion à l'équilibre de Nash a été effectuée pour 4 cas: un agent dans chaque sous-unité, 5 agents dans chaque sous-unité, 10 à 5 et 5 à 10 agents dans chaque sous-ensemble. Les résultats sont présentés sous la forme de graphiques de convergence.

Mots-clés: Système multi-agent; système cognitif, solution optimale, équilibre de Nash.

1. INTRODUCTION

Les systèmes multi-agents (SMA) présentent une nouvelle classe de systèmes intelligents collaboratifs capables de résoudre des problèmes complexes basés sur des ressources de traitement de données minimales ou réduites. Ces systèmes sont constitués d'un certain nombre d'agents logiciels ou matériels intelligents, homogènes ou hétérogènes, qui ont: un échange intense d'informations et de connaissances; capacité de coopérer, de coordonner et de négocier des activités [1,2,3,4].

Le domaine de SMA est interdisciplinaire et intègre des connaissances provenant de diverses branches de l'économie, de la technologie, des mathématiques, de la logique, des sciences sociales, etc. Du point de vue structurel, SMA présentent une sous-classe de systèmes de traitement distribués, parallèles ou concurrents de données [5,6], où les agences ont une autonomie (leurs propres intérêts et connaissances), mais la synchronisation et la coordination se fait au niveau d'une décision commune [7].

En même temps, les SMA sont capables de résoudre des problèmes communs, où tous les agents déplacent leurs activités pour solutionner positivement le problème sans la présence d'agents qui peuvent opposés à résoudre ce problème ou la solution négative du problème. Cette propriété de SMA peut être considérée comme un désavantage lors de la résolution de problèmes basés sur la théorie des jeux, où la présence de deux types d'agence est obligatoire: positive (PA) - avec des intentions de profit pour la propre équipe; et négatif (NA) - avec des intentions de profit pour l'équipe adverse. Ces problèmes peuvent être attribués à la théorie de l'équilibre de Nash. [8,9].

La capacité et l'exactitude de la résolution d'un problème pour un SMA est déterminée par les capacités cognitives des agents individuels et collectifs [10].

Les systèmes cognitifs ont la capacité de construire des stratégies basées sur des hypothèses et des données contextuelles appropriés, fournissant des informations et des connaissances aux agents. L'intelligence collective représente l'effet le plus important du partage de données et d'informations. Au fil du temps, on a constaté que les systèmes cognitifs sont l'une des étapes les plus importantes du long processus d'adaptation de l'homme à l'évolution de la technologie de l'information [11].

Cet article propose le développement et la recherche d'un SMA avec des propriétés cognitives conçues pour trouver la solution optimale à la résolution de problèmes dans la théorie des jeux, en particulier la recherche de l'équilibre de Nash. [9].

2. LE MODÈLE MATHÉMATIQUE DE RECHERCHE DE LA SOLUTION OPTIMALE

Soit le processus $S \subset \mathbb{R}^N$ défini dans un espace N - dimensionnel. La dynamique du processus S est déterminée par l'ensemble des fonctions $F = \{f_1, f_2, \dots, f_K\}$, où $F : S \rightarrow \mathbb{R}^N$. Le statut du processus S est déterminé par le vecteur d'état $X = \{x_1, x_2, \dots, x_N\}$ [12,13].

L'ensemble des fonctions F agit sur le processus S selon le système d'équation (1):

$$\begin{cases} f_1(y_1, x_1, x_2, \dots, x_N) = 0 \\ f_2(y_2, x_1, x_2, \dots, x_N) = 0 \\ \dots \\ f_K(y_K, x_1, x_2, \dots, x_N) = 0 \end{cases} \quad (1)$$

Sur la base du modèle (1), on peut également être définis des critères de recherche pour des solutions optimales **min/ max** (2):

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

Dans le modèle (2) sont définis: S_{\min} - la solution minimale pour le processus S ; S_{\max} - la solution maximale pour le processus S ; K_{\min} - le nombre de fonctions qui minimise le processus S ; K_{\max} - le nombre de fonctions qui maximise le processus S , où $K_{\min} \cap K_{\max} = \emptyset$.

3. DÉVELOPPEMENT D'AGENTS COGNITIFS

Afin de résoudre le modèle (2) il est proposé le système Multi-Agent (**MAS**) constitué de deux sous-ensembles d'agents (Figure 1): L'ensemble d'agents positifs (**PA**) qui résout la condition $S \rightarrow S_{\min}$, et l'ensemble d'agents négatifs (**NA**) qui solutionnent la condition $S \rightarrow S_{\max}$. Pour l'action sur le processus S les agents positifs génèrent

des signaux de commande $EC_{\min} = \{y_j, \forall j = \overline{1, K_{\min}}\}$, et les agents négatifs génèrent des signaux de commande $EC_{\max} = \{y_i, \forall i = \overline{1, K_{\max}}\}$. L'état du processus S est identifié par les Agents en lisant les signaux d'état $ES = \{x_k, \forall n = \overline{1, K}\}$.

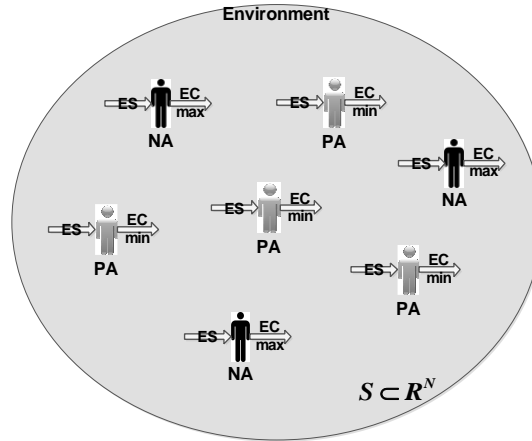


Figure 1. Le système Multi-Agent.

Chaque agent a des informations et des connaissances qu'il évalue au fil du temps. L'échange direct d'informations entre d'agents n'a pas lieu. L'état d'agents positifs (PA) ou négatifs (NA) est déterminée par leurs actions sur le processus S . Ces critères de comportement des agents font partie de la théorie des jeux et correspondent au modèle d'équilibre de Nash [9].

Le modèle fonctionnel de l'agent cognitif est représenté sur la figure 2, où: x_k - le signal d'identification de l'état du processus S ; **ADC** - convertisseur analogique-numérique; **ES Stor** - mémoire pour le stockage de l'état du processus pendant un intervalle de temps défini; **Data Pr** - une unité de traitement des données d'état du processus S ; **EC Stor** - mémoire pour le stockage les décisions de contrôle de processus S pour un intervalle de temps défini; **DAC** - convertisseur numérique-analogique; y_k - un signal d'action sur le processus; **Data Instr Stor** - mémoire pour stocker le code de programme pour le traitement de données; **Cogn Stor** - mémoire pour stocker des connaissances appliquées pour le traitement de données; **Cogn Instr Stor** - mémoire pour stocker le code de programme pour générer de nouvelles connaissances; **Cogn Pr** - unité de traitement des connaissances anciennes et de génération de nouvelles connaissances; $C_k[T-1]$ - modèle de connaissance utilisé pour la commande dans le moment de temps $T-1$; et $C_k[T] = q_k(C_k[T-1], X_k, Y_k)$ - modèle de connaissance utilisé pour suivante cas de commande T , où: $X_k = \{x_k[0], x_k[1], \dots, x_k[T-1]\}$ - la chaîne d'état du processus S jusqu'au temps $T-1$; $Y_k = \{y_k[0], y_k[1], \dots, y_k[T-1]\}$ - la séquence des signaux d'action sur le processus effectuée jusqu'au moment de temps $T-1$; q_k - le modèle de génération de nouvelles connaissances.

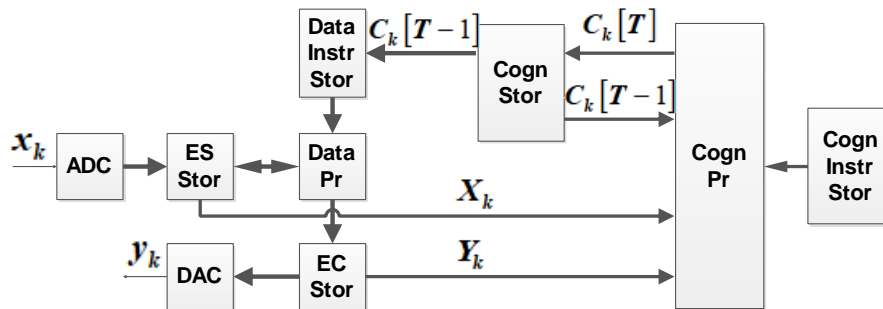
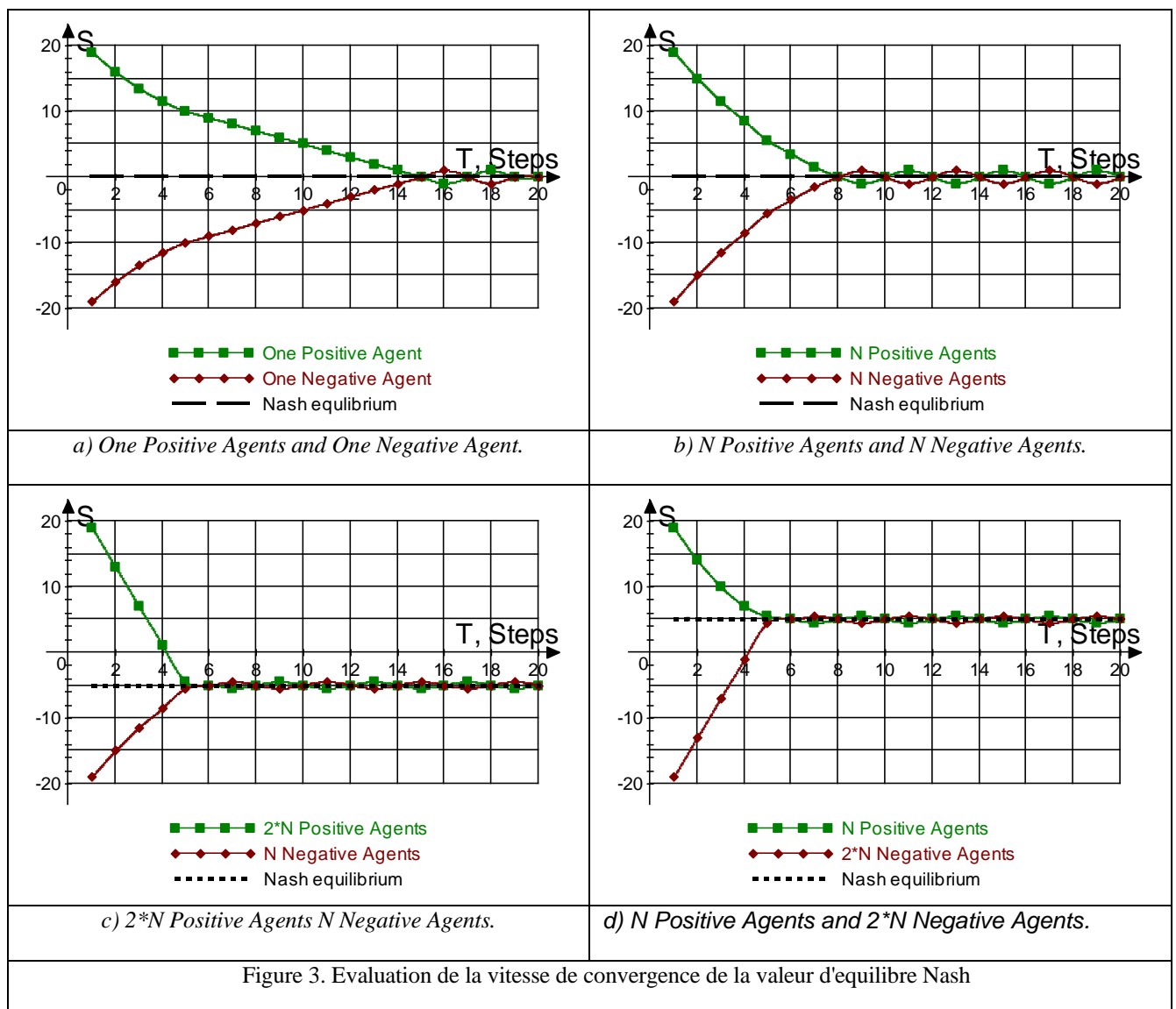


Figure 2. Le modèle fonctionnel de l'agent cognitif.

La fonctionnalité des agents commence par l'acquisition des données d'état x_k du processus S , sa conversion en code numérique et le stockage de ces données en mémoire $ES\ Stor$. Le processeur $Data\ Pr$ exécute la séquence de code à partir de la mémoire de programme $Data\ Instr\ Stor$ qui fournit le modèle (1) à résoudre et le respect des conditions le modèle (2). La séquence de code exécutée par le processeur $Data\ Pr$ est déterminée par les connaissances stockées dans la mémoire $Cogn\ Stor$ qui modifie le contenu de chaque itération de commande à la suite de l'analyse par le processeur $Cogn\ Pr$ de la vitesse de convergence du modèle (2), et du contenu $C_k[T-1]$, X_k et Y_k . Le résultat des calculs effectués par le processeur $Data\ Pr$ est enregistré dans la mémoire de stockage $EC\ Stor$ des données de commande. La dernière valeur calculée est transmise au convertisseur DAC qui convertit le code de commande en un signal d'action y_k .



4. EVALUATION DE LA VITESSE DE CONVERGENCE DE LA VALEUR D'EQUILIBRE DE NASH

Évaluation de la convergence de vitesse a été réalisée pour quatre cas de rapport du nombre d'agents positifs (PA) et le nombre d'agents négatifs (NA). Les résultats de l'évaluation sont présentés à la figure 3 où:

a) *La figure 3, a. Système composé d'un agent positif et d'un agent négatif.* Pour résoudre ce cas, le système d'équations (1), en respectant la condition (2), l'équilibre de Nash est établi au point 15. Après le point 15 il y a une oscillation de l'équilibre par le fait que chaque agent appelle pour obtenir un plus grand profit que l'adversaire mais finalement cette tentative est compensée par les actions des adversaires.

b) *La figure 3, b. Système constitué d'un nombre N d'agents positifs et d'un nombre N d'agents négatifs.* Pour ce cas l'équilibre de Nash est réglé au point 8. La réduction du temps pour atteindre l'équilibre Nash est déterminé par la présence de plusieurs agents dans le système.

c) *La figure 3, c. Système constitué de $2*N$ nombres d'agents positifs et d'un nombre N d'agents négatifs.* L'équilibre de Nash pour ce cas est déjà déterminé au point 5, mais la valeur de l'équilibre est déplacée en faveur d'agents positifs qui sont 2 fois plus.

d) *La figure 3, d. Système constitué d'un certain nombre d'agents N positifs et un nombre de $2*N$ d'agents négatifs.* Dans ce cas donné l'équilibre de Nash a été établi au point 5, mais l'équilibre est déplacé en faveur d'agents négatifs qui sont 2 ou plus.

5. CONCLUSIONS

Aujourd'hui, les systèmes multi-agents, présentent une zone d'intérêt particulier en matière de recherche et de leurs utilisations dans la pratique. Les propriétés cognitives des agents permettent le déploiement des systèmes d'adaptation en l'environnement d'affaires et d'infrastructures. Nous présentons les deux types d'agents qui permettent des modèles de conduite l'application de la théorie des jeux, en particulier l'équilibre de Nash. L'application de ces modèles permet de réduire considérablement la complexité des agents, parce qu'est exclue la nécessité d'une communication entre eux. Le statut et les activités des agents sont déterminés par leur influence sur l'environnement d'activité. Chaque agent individuel résout sa partie de problème d'optimisation globale. Les agents positifs optimisent l'état d'activité de l'environnement en réduisant au minimum ses paramètres d'état, mais les agents négatifs permettent d'optimiser l'environnement d'activité en maximisant les paramètres d'état de celui-ci.

Dans cet article, nous avons étudié quatre cas pour atteindre l'équilibre de Nash qui démontre l'importance des systèmes multi-agents et leur nombre dans le processus de commande.

Pour les recherches futures sont prévues des systèmes multi-agents basés sur un illimité nombre d'agents et modèles non linéaires d'action sur l'environnement d'activité.

MENTIONS

Les résultats obtenus font partie des recherches menées sous le thème "Modèles, méthodes et interfaces pour la gestion et l'optimisation des systèmes de fabrication intelligents" au sein du Département d'Informatique et de Systèmes.

6. RÉFÉRENCES

1. M. Wooldridge, "An Introduction to MultiAgent Systems," John Wiley & Sons. 2002, 366 p, ISBN: 0-471-49691-X.
2. L. Rodriguez, E. Insfran and L. Cernuzzi, "Requirements Modeling for Multi-Agent Systems," in Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications, Dr. Faisal Alkhateeb (Ed.), 2011, InTech, DOI: 10.5772/14776. (Available on: <https://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications/requirements-modeling-for-multi-agent-systems>).

3. M. Guijarro, R. Fuentes-Fernández and G. Pajares, "A Multi-Agent System Architecture for Sensor Networks," in *Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications*, Dr. Faisal Alkhateeb (Ed.), 2011, InTech, DOI: 10.5772/14309. (Available on: <https://www.intechopen.com/books/multi-agent-systems-modeling-control-programming-simulations-and-applications/a-multi-agent-system-architecture-for-sensor-networks>).
4. Y. Shoham, K. Leyton-Brown, "Multi-Agent systems. Algorithmic, Game-Theoretic, and Logical Foundations," Published by Cambridge University Press. Revision 1.1, 2010. 532 p. (Available on: <http://www.masfoundations.org>).
5. Hagit, J. Welch, "Distributed Computing: Fundamentals, Simulations, and Advanced Topics," Wiley-Interscience, 2004, 432 p. ISBN: 0-471-45324-2.
6. Gh. Safonov, V. Ababii, V. Sudacevschi, "Analysis of distributed computing architectures for synthesis of Multi-Agent systems," *European Applied Sciences Journal*, № 9, 2016 (September), ISSN 2195-2183, pp. 34-37.
7. G. Weiss, "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence," MIT Press, 1999, 615 p. ISBN: 0-262-23203-0.
8. E. Rasmusen, "Games and Information: An Introduction to Game Theory," 3rd ed. Blackwell, Oxford. 2001.
9. F. Nash, "Non-Cooperative Games," *Annals of Mathematics* 54, 286-295. 1951.
10. H. I. Christiansen, A. Sloman, G.-J. Krujiff, J. Wyatt, "Cognitive systems," Berlin: Springer-Verlag. 2009.
11. D. Dumitrescu, "Algoritmi genetici și strategii evolutive - Aplicații în inteligența artificială și în domenii conexe," Editura Albastră, Cluj-Napoca, 2000, 228p.
12. A.R. Parkinson, R. Balling, and J.D. Hedengren, "Optimization Methods for Engineering Design," Brigham Young University, 2013.
13. I.C. Rădulescu, "Rezolvarea unor probleme de optimizare multi-obiectiv bazată pe algoritmi evolutivi," *Revista Română de Informatică și Automatică*, vol. 25. Nr. 2, 2015, pp.39-48.