

STRATEGII DE ADAPTARE ÎN SISTEME MULTIAGENT

dr. ing. Mihaela M. Oprea

Universitatea Petrol-Gaze din Ploiești
Catedra de Informatică

Rezumat: Dezvoltarea sistemelor multiagent performante implică includerea capacității de adaptare a agenților inteligenți la mediul lor de lucru. Lucrarea prezintă principalele strategii de adaptare care pot fi implementate în sisteme multiagent. Drept aplicație este considerată problema adaptării agenților la procesul de negociere într-un sistem de comerț electronic bazat pe agenți și sunt descrise două modalități de implementare a capacității de adaptare a agenților.

Cuvinte cheie: agenți inteligenți, sisteme multiagent, strategii de adaptare.

1. Introducere

În ultimul deceniu s-a impus o nouă abordare a sistemelor distribuite prin metoda sistemelor multiagent. Un sistem multiagent este un sistem care constă din cel puțin doi agenți inteligenți capabili să interacționeze între ei în vederea realizării unor scopuri individuale sau comune (scopuri globale) și care partajează același mediu de lucru. Un agent inteligent este o entitate capabilă să acționeze autonom în mediul său de lucru. Autonomia unui agent înseamnă capacitatea acestuia de a acționa singur, fără intervenție din exterior (de exemplu, intervenție umană). Printre proprietățile unui agent inteligent menționăm: mobilitatea, raționalitatea, capacitatea de învățare/adaptare etc. Lucrarea își propune să abordeze problema adaptării unui agent inteligent la mediul său de lucru atunci când face parte dintr-un sistem multiagent. În exemplul prezentat în lucrare ne vom referi la un caz particular de agenți inteligenți, agenții software. Un agent software este un program care se execută independent și care este capabil să realizeze automat selecția acțiunilor atunci când apar diferite evenimente (așteptate sau neașteptate). Modalitatea de implementare a capacității de adaptare a unui agent software la mediul său de lucru este ilustrată în cazul procesului de negociere dintr-un sistem de comerț electronic bazat pe agenți.

2. Agenți inteligenți adaptivi

Adaptarea reprezintă o proprietate a organismelor vii care nu constă numai în capacitatea acestora de a învăța și în capacitatea de a se auto-organiza astfel încât răspunsurile pe care le dau să fie cât mai adecvate la schimbările din mediul în care trăiesc. În mod similar, agenții inteligenți se pot adapta la mediul lor de lucru. Adaptarea permite generarea unui model al procesului de selecție a acțiunilor în cadrul unui sistem multiagent, indicând astfel interacțiuni viitoare de succes.

Figura 1 prezintă arhitectura unui agent inteligent. Agentul inteligent păstrează un model al lumii exterioare, sub forma unei mulțimi de stări posibile S și mapează percepția mediului de lucru de la un anumit moment în mulțimea stărilor S determinând o anumită acțiune care va fi executată în mediul de lucru. Figura 2 prezintă un model general de agent inteligent dotat cu capacitate de adaptare la mediul său de lucru (agent inteligent adaptiv) [1]. Agentul percepe mediul de lucru prin intermediul senzorilor și în conformitate cu un standard de performanță modulul *Critică* va emite o anumită reacție către elementul de adaptare, care la rândul său va genera

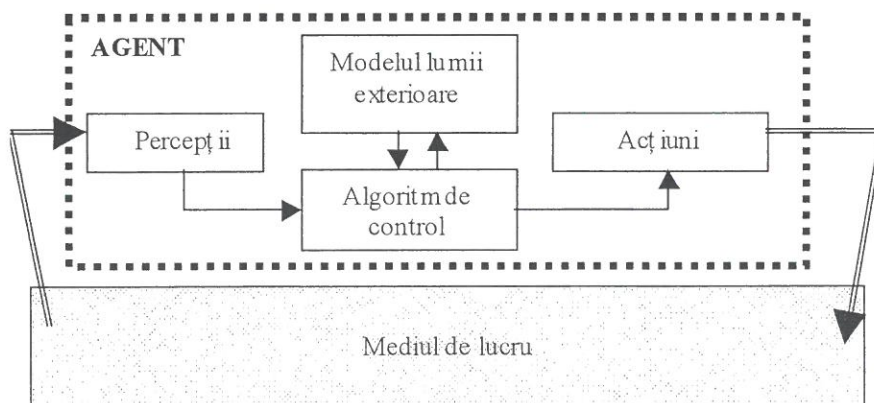


Figura 1. Arhitectura unui agent inteligent

scopuri ale adaptării ce vor fi în final transformate în acțiuni realizate de efectorii agentului.

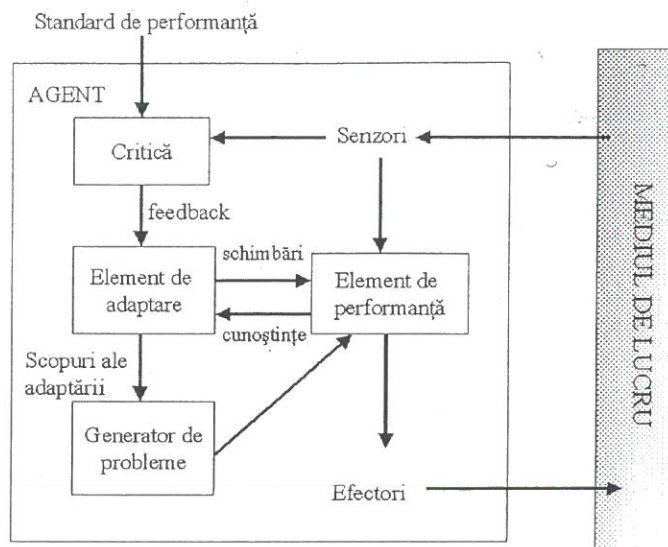


Figura 2. Model general al unui agent adaptiv

Într-un sistem multiagent, agenții sunt integrați mediului în care trăiesc și ei trebuie să interacționeze cu ceilalți agenți pentru a-și realiza scopurile individuale și globale. Un agent integrat mediului său de lucru este un agent care există în acel mediu și care lucrează ca o singură entitate. Agenții încearcă să se adapteze la mediul de lucru prin învățare sau prin auto-organizare, realizând astfel o anticipare a interacțiunilor cu ceilalți agenți. Adaptarea într-un sistem multiagent este complicată de faptul că pe măsură ce agenții se adaptează, mediul se schimbă în continuu. De exemplu, atunci când agenții învață și acționează simultan, deciziile lor afectează și limitează ceea ce urmează să învețe. În cazul comerțului electronic bazat pe agenți, agenții software trăiesc într-un mediu virtual extrem de dinamic (Internet-ul), iar capacitatea lor de adaptare la un astfel de mediu are rolul de a le îmbunătăți performanțele prin anticiparea unor evenimente viitoare.

3. Strategii de adaptare

Adaptarea agenților inteligenți la mediul lor de lucru se poate realiza prin diferite metode care se împart în două mari categorii: metode evolutive și metode de învățare. Metodele evolutive cuprind algoritmi genetici, strategiile evolutive și programarea genetică. În cazul acestor metode există două posibilități: de a avea indivizi care evoluează în mod gradual sau de a construi indivizi printr-un proces ontogenetic (adică, prin evoluția comportamentului agentului) prin așa numita metodă etogenetică. Metodele de învățare includ învățarea inductivă, învățarea bazată pe model, învățarea bazată pe scheme, învățarea bayesiană, învățarea de tip *reinforcement-learning* (incluzând cazul particular de învățare *Q-learning*) etc. Prezentăm pe scurt diferite metode din cele două categorii.

Algoritmii genetici se bazează pe simularea evoluției. Dintr-o populație de indivizi (reprezențați sub forma unui șir de biți, de exemplu) sunt selectați indivizi (numiți părinți) cărora li se aplică operatori de încrucișare (crossover) și operatori de mutație generându-se astfel indivizi mai buni care sunt determinați pe baza unei funcții de evaluare (numită funcție fitness). În continuare, este prezentată forma generală a unui algoritm genetic.

```

Function GA(populatie, FF) return individ
inputs: populatie, o mulțime de indivizi
         FF, funcția fitness, de evaluare a unui individ
repeat
    parinti ← Selectie(populatie, FF)
    populatie ← Crossover & Mutatie(parinti)
until *un individ se potrivește cel mai bine
return *cel mai bun individ din populatie, în conformitate cu FF

```

În urma repetării celor trei pași (selecție, încrucișare și mutație) se va obține la un moment dat un cel mai bun individ, în conformitate cu funcția de evaluare *FF*.

Algoritmii genetici utilizează indivizi care au aceeași lungime a reprezentării lor sub formă de șir, acesta fiind unul din dezavantajele utilizării lor [2].

Metoda etogenetică este o metodă evolutivă care furnizează principii generale pentru proiectarea comportamentelor unui agent. Un exemplu de model din această categorie este modelul ATNoSFERES [3]. Caracteristicile acestui model sunt următoarele: (1) separă structura informației genetice (reprezentarea sub formă de șir, respectiv, structura lexicală) de interpretarea ei (structura semantică, graful ATN); (2) nivelul influenței operatorilor genetici clasici (mutație și încrucișare) nu depinde de porțiunea șirului de biți în care sunt implicați (nici de locația lor și nici de dimensiunea lor); (3) modelul nu utilizează nici un parametru pentru a construi comportamente, execuția unui comportament depinde doar de condițiile mediului de lucru.

Graful ATN reprezintă un model pentru proiectarea comportamentului unui agent inteligent. Într-o astfel de schemă sunt specificate doar condițiile și acțiunile asociate unui anumit comportament. Acțiunile reprezintă primitivele comportamentale, care pot fi executate de agent. Condițiile sunt percepții sau stimuli care induc selectarea unei anumite acțiuni. Muchiile unui graf ATN sunt etichetate cu o mulțime de condiții și o secvență de acțiuni. Figura 3 prezintă un exemplu de ATN.

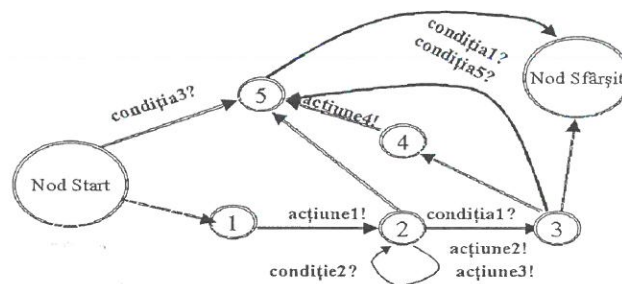


Figura 3. Exemplu de ATN

Structura de bază conține două noduri, Start și Sfârșit. La fiecare moment t , agentul alege în mod aleator o muchie din cele care fie nu au o condiție asociată, fie toate condițiile asociate sunt adevărate simultan și execută acțiunea asociată muchiei respective după care va trece în nodul destinație.

Modelul ATNoSFERES permite construirea graduală a sistemelor multiagent capabile să se reorganizeze și să se dezvolte, prin evoluția comportamentelor agenților (care pot fi parțial specificate anterior).

Programarea genetică extinde algoritmiile genetice către evoluția programelor, fiind o formă de calcul evolutiv în care indivizii din populația care evoluează sunt programe și nu șiruri de biți [4]. Programele sunt reprezentate sub forma unor arbori în care fiecare apel de funcție este reprezentat printr-un nod, iar argumentele funcției sunt date de nodurile descendente. Populația este alcătuită dintr-o mulțime de arbori. La fiecare iterație se induce o nouă generație de indivizi cu ajutorul celor trei operatori clasici, selecție, mutație și încrucișare. Funcția de evaluare, FF , se determină prin executarea programului asupra unei mulțimi de date de instruire. Un exemplu de operator de încrucișare este înlocuirea unui subarbore ales aleator al programului părinte, printr-un subarbore din celălalt program părinte. Unul din dezavantajele principale ale programării genetice este complexitatea timp și spațiu foarte mare. Utilizarea unor calculatoare paralele poate conduce la creșterea performanțelor programării genetice.

Învățarea într-un sistem multiagent este complicată de faptul că pe măsură ce alți agenți învață, mediul de lucru se schimbă. Astfel, o metodă de învățare eficientă trebuie să încerce să contruiască un model al mediului de lucru și modele pentru agenții cu care interacționează agentul care învață, pentru a anticipa unele din schimbările ulterioare. De cele mai multe ori, aceste modele se construiesc pornind de la cunoștințe legate de domeniul aplicației și de la observațiile directe pe care agentul le poate prelua din interacțiunile sale în cadrul sistemului multiagent.

Învățarea inductivă utilizează o mulțime de exemple de instruire care reprezintă un istoric al deciziilor anterioare ale agentului inteligent. O astfel de metodă are o aplicabilitate redusă în cazul învățării on-line.

Învățarea de tip *reinforcement-learning* utilizează o funcție Reacție(R) care premiază sau penalizează acțiunile selectate de agent [5]. Alegerea secvenței de acțiuni care produce valoarea maximă a premierii cumulate reprezintă scopul acestei metode de învățare. În figura 4 este prezentată schema de bază a metodei de învățare de tip *reinforcement-learning*.

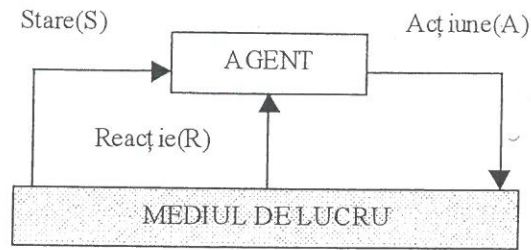


Figura 4. Metoda de tip reinforcement-learning

Dacă un agent inteligent care învață trece printr-o secvență de stări S_0, S_1, \dots, S_n , ca urmare a selectării acțiunilor corespunzătoare A_0, A_1, \dots, A_n și primește reacțiile R_0, R_1, \dots, R_n de la mediul de lucru, valoarea cumulată a premierii este dată de funcția $V^\pi(S_0)$ din ecuația (1).

$$V^\pi(S_0) = \sum_{i=0}^n \gamma^i R_i \quad (1)$$

Parametrul γ este un număr subunitar, iar π este o politică arbitrară de selectare a acțiunilor pornind din starea inițială S_0 . Agentul inteligent învață să aleagă secvența de acțiuni care va maximiza funcția $V^\pi(S_0)$.

Învățarea bazată pe model utilizează, de exemplu, teoria jocurilor pentru a modela interacțiunea dintre agenți. Procesul de învățare se desfășoară în două etape: (1) agentul inferează un model al celuilalt agent pe baza interacțiunilor anterioare; (2) agentul utilizează modelul învățat pentru a proiecta strategia de interacțiune viitoare.

O altă metodă de învățare este dată de rețelele neuronale artificiale care pot fi utilizate pentru a realiza diferite predicții ale comportamentului celorlalți agenți.

Învățarea bayesiană se bazează pe raționamentul bayesian care furnizează o metodă probabilistică de inferență și necesită cunoștințe inițiale ale probabilităților (ipoteze, date anterioare disponibile etc).

4. Aplicație

Considerăm un sistem de comerț electronic bazat pe agenți în care fiecare participant are asociat un agent inteligent care acționează în interesul proprietarului său. Într-un astfel de sistem există mai multe tipuri de agenți, agenți vânzători, agenți cumpărători, agenți mediatori, agenți de tip broker etc. Obiectivul global al sistemului este de a facilita încheierea cu succes a cât mai multor tranzacții. Avantajul principal al unui sistem de comerț electronic bazat pe agenți este dat de costurile scăzute ale tranzacțiilor atât pentru vânzător cât și pentru cumpărător. Simularea unui astfel de sistem a fost realizată în JADE [6], o platformă JAVA destinată dezvoltării sistemelor multiagent conform standardului FIPA [7]. Utilizatorul poate crea diferite tipuri de agenți care au asociate preferințele acestuia referitoare la produse și servicii. Fiecare produs tranzacționat are un număr de atribute asociate, iar unele atribute se pot negocia. O modalitate de creștere a gradului de autonomie a agenților într-un mediu dinamic de lucru, cum este comerțul electronic, o reprezintă includerea în arhitectura agenților a unei capacități de adaptare. În continuare, ne vom referi la procesul de negociere a unui sau a mai multor atribute ale unui produs, și vom ilustra două modalități de adaptare a agenților la acest proces de negociere.

Negocierea este un proces general în care se pot implica doi sau mai mulți agenți în vederea determinării unui contract acceptat de către toți agenții, în anumite condiții. Acest proces implică de fapt comunicarea continuă între agenți până în momentul acceptării condițiilor de către toți agenții sau până când termenul de încheiere a negocierii este atins. Între agenți are loc un schimb de propuneri care se referă la anumite atribute ale produselor tranzacționate, propuneri care sunt valori ale acestor atribute, generate automat de către strategia de negociere a agentului care face propunerea.

Elementele de bază ale procesului de negociere multiagent sunt protocolul negocierii și strategiile de negociere asociate agenților inteligenți implicați în negociere. Strategia de negociere a unui agent determină modul în care va acționa acesta în cadrul protocolului încercând să obțină un beneficiu cât mai mare. Ea se bazează pe un model al preferințelor economice ale utilizatorului și pe strategiile/comportamentul altor părți implicate. În general, luarea unei decizii se bazează pe considerarea mai multor atribute ale unui produs. De exemplu, evaluarea unui calculator de către cumpărător va depinde de mai multe atribute, capacitatea memoriei, tipul procesorului și preț propus de vânzător.

Funcția utilitate a unui utilizator, U , aplicată unui produs p cu n atribute este definită de ecuația (2).

$$U(p) = \sum_{i=1}^n w_i f_i(a_i) \quad (2)$$

unde $w_i \in [0,1]$ reprezintă ponderea atributului a_i în cadrul evaluării globale a produsului p .

$$f_i(a_i): Dom(a_i) \rightarrow R$$

$Dom(a_i)$ reprezintă domeniul de valori pentru atributul a_i .

În cazul vânzării de calculatoare, forma funcției f pentru preț este dată de exemplu de ecuația (3).

$$f_{pret}(x) = 1 - x \quad (3)$$

Valoarea atributului x este normalizată în intervalul $[0,1]$.

O primă modalitate de implementare a capacității de adaptare a agenților la procesul de negociere este dată de algoritmi genetici. Vom ilustra pe scurt această metodă similară cu cea prezentată în [8]. Scopul este de a dota agenții cu capacitate de învățare a funcției utilitate a agenților cu care negociază. Funcția utilitate este codificată sub forma unui individ al unei populații asupra căreia se va aplica un algoritm genetic similar celui prezentat în secțiunea 3. Considerăm că forma generală a funcției utilitate este dată de ecuația (4).

$$f_i(a_i) = a_i^{r_i} \quad (4)$$

Parametrul $r_i \in [0,5]$ codifică funcția utilitate asociată atributului a_i . Forma generală a unui individ t este dată în figura 5.

w_1	r_1	w_2	r_2	w_3	r_3	w_n	r_n
-------	-------	-------	-------	-------	-------	-----	-----	-------	-------

Figura 5. Forma generală a unui individ t

Populația inițială este generată aleator. Operatorul de selecție este de tip ruletă, iar operatorul de încrucișare este în două puncte. Probabilitatea de aplicare a unui operator de încrucișare este de 0.7, iar a unui operator de mutație este de 0.001. Cei trei pași, selecție, încrucișare și mutație sunt repetați până când funcția fitness $F_f(t)$ este maximizată. Produsele negociate sunt clasificate și evaluate. Clasificării i se asociază o funcție g , iar evaluării o funcție h . Aceste funcții vor genera forma funcției fitness care este dată de ecuația (5).

$$F_f(t) = w_h h(t) + w_g g(t) \quad (5)$$

unde $w_h + w_g = 1$. Funcția g a fost neglijată, întrucât ea generează restricții slabe. Forma funcției h este dată de ecuația (6).

$$h(t) = \frac{1}{\sum_{i=1}^m |\Delta u_i|} \quad (6)$$

unde m reprezintă numărul de produse, iar Δu_i reprezintă diferența între utilitatea estimată și cea reală, a utilizatorului, pentru produsul i .

Prin aplicarea celor trei operatori de selecție, încrucișare și mutație se generează populații cu indivizi din ce în ce mai buni ca rezultat al metodei de căutare modelată de cei trei operatori. Rolul operatorului de încrucișare este de a recombină materialul genetic existent în noi moduri, iar rolul operatorului de mutație este de a introduce material genetic nou în populație prin modificări aleatoare.

În lucrarea [9] este prezentată o metodă evolutivă, bazată pe algoritmi genetici, pentru adaptarea agenților la contextul negocierii, care permite selectarea celei mai potrivite strategii de negociere pornind de la modelul general de negociere prezentat în [10].

O altă modalitate de implementare a capacității de adaptare a unui agent inteligent la procesul de negociere este dată de utilizarea unei rețele neuronale artificiale cu reacție înainte (*feed-forward*), similară celei prezentate

în [11]. Rolul rețelei neuronale artificiale este de a modela strategia de negociere a celui alt agent, în cazul unei negocieri bilaterale. Utilizarea acestei metode de învățare permite creșterea numărului de tranzacții încheiate precum și efectuarea unor tranzacții cât mai avantajoase pentru agentul dotat cu capacitate de învățare.

Considerăm doi agenți a și b care negociază mai multe atribute ale unui produs, atribute reprezentate sub forma unui vector X . Lanțul de propuneri asociat negocierii la momentul t este $X_{a \leftrightarrow b}^t = \{X_{a \rightarrow b}^1, X_{b \rightarrow a}^2, \dots, X_{a \rightarrow b}^t\}$. $X_{a \rightarrow b}^t[j]$ reprezintă propunerea făcută de agentul a către agentul b la momentul t pentru atributul j aflat în negociere. Fiecare agent are o funcție utilitate U^i , $U^i: S^i \rightarrow R$. S^i este spațiul stărilor relevante agentului i și R este mulțimea numerelor reale. Funcția utilitate ordonează stările după preferințele utilizatorului. Fiecare agent își cunoaște funcția utilitate și poate observa acțiunile celorlalți agenți, adică, propunerile lor. De asemenea, fiecare agent are cunoștințe legate de domeniul în care lucrează și poate face estimări referitoare la stările în care s-ar putea ajunge în urma selectării anumitor acțiuni disponibile. Un agent poate învăța din observațiile interacțiunilor cu ceilalți agenți. Propunerile făcute de un agent sunt generate de strategia sa de negociere care este specifică fiecărui atribut al unui produs. Strategia de negociere a unui agent reprezintă informație privată a acestuia.

Presupunem că doi agenți, cumpărător și vânzător, negociază prețul unui produs. Fiecare agent cunoaște prețul său de rezervă PR , care reprezintă valoarea maximă permisă (pentru cumpărător, $PR^{\text{cumpărător}}$) sau minimă admisă (pentru vânzător, $PR^{\text{vânzător}}$) în cazul încheierii unei tranzacții. O tranzacție se poate încheia doar dacă există o zonă de suprapunere între cele două prețuri de rezervă ale agenților, adică dacă $PR^{\text{vânzător}} \leq PR^{\text{cumpărător}}$. Întrucât un agent își cunoaște doar prețul său de rezervă, nu poate ști de la început dacă o negociere se va încheia cu succes. Soluția propusă este de a modela seria de timp a propunerilor făcute de celălalt agent ca o rețea neuronală artificială de tip feed-forward (RNA-FF) și de a realiza predicții ale propunerilor viitoare făcute de agent. Presupunem că agentul cumpărător are implementată capacitatea de învățare sub forma unei rețele neuronale artificiale RNA-FF. La momentul t ($t > 5$), agentul cumpărător va putea estima următoarea propunere pentru atributul j făcută de agentul vânzător pe baza ultimelor trei propuneri ale acestuia. Arhitectura rețelei neuronale RNA-FF este dată în figura 6.

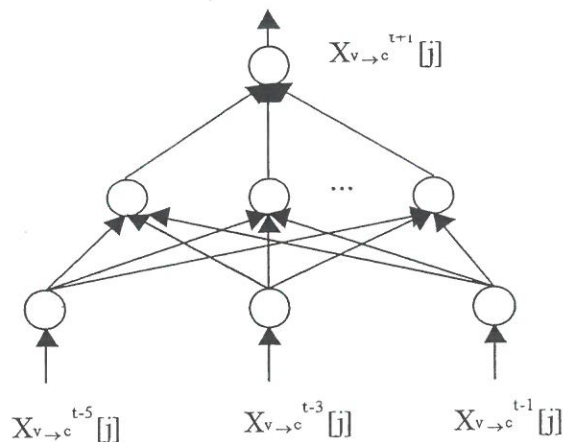


Figura 6. Rețeaua neuronală artificială RNA-FF

Pe baza valorii estimate a propunerii făcută de agentul vânzător la momentul $t+1$ pentru atributul j , agentul cumpărător poate deveni mai flexibil pentru a grăbi încheierea cu succes a negocierii. Astfel, agentul cumpărător își ajustează strategia de negociere la strategia de negociere estimată a agentului vânzător cu ajutorul rețelei neuronale artificiale RNA-FF. Similar agentului cumpărător, agentul vânzător poate utiliza rețeaua RNA-FF pentru a estima propunerile făcute de agentul cumpărător. Întrucât mulțimea de instruire a rețelei este extrasă online, există un interval de timp la începutul negocierii în care capacitatea de învățare nu este aplicată. Rezultatele experimentale obținute (prezentate în [11]) au relevat o îmbunătățire a numărului de tranzacții încheiate cu succes în procesul de negociere a unui singur atribut al unui produs (de exemplu, prețul). Experimentele realizate au inclus tranzacționarea unor produse la mâna a doua (tip second hand), calculatoare, telefoane mobile, imprimante, într-un sistem de comerț electronic bazat pe agenți simulat cu ajutorul platformei JADE.

5. Concluzii

Includerea în arhitectura unui agent inteligent a capacității de adaptare la mediul său de lucru în contextul unui sistem multiagent reprezintă un deziderat privind creșterea gradului de autonomie a agentului. Lucrarea a abordat problema implementării acestei capacități prezentând diferite soluții propuse în literatura de specialitate. Metodele evolutive au inclus algoritmi genetici, metoda etogenetică și programarea genetică. Dintre metodele de învățare au fost pe scurt descrise învățarea de tip *reinforcement-learning*, învățarea bazată pe model, rețele neuronale artificiale și învățarea bayesiană. Drept exemplu a fost considerată problema adaptării agenților la contextul negocierii în cadrul unui sistem de comerț electronic bazat pe agenți. Au fost descrise succint două modalități de rezolvare a problemei adaptării, adaptarea cu ajutorul algoritmilor genetici pentru estimarea funcției utilitate a agentului cu care se negociază atributele unui produs, respectiv, învățarea cu ajutorul unei rețele neuronale artificiale RNA-FF a strategiei de negociere a agentului cu care se negociază un atribut al produsului. Adoptarea unei anumite strategii de adaptare depinde de domeniul aplicației modelate cu ajutorul unui sistem multiagent.

Bibliografie

1. **OPREA, M.:** Adaptability and Embodiment in Agent-Based E-Commerce Negotiation. În: Proc. of Adaptability and Embodiment Using Multiagent Systems – AEMAS 2001 Workshop, 2001, pp. 257-265.
2. **HOLLAND, J.:** Adaptation in natural and artificial systems, MIT Press, Cambridge, MA, 1992.
3. **LANDAU, S., S., PICAULT:** Modeling Adaptive Multiagent Systems Inspired by Developmental Biology. În: Proc. of Adaptability and Embodiment Using Multiagent Systems – AEMAS 2001 Workshop, 2001, pp. 238-246.
4. **BANZHAF, W., P. NORDIN, R. E. KELLER, F. D. FRANCONI:** Genetic Programming ~ An Introduction, Morgan Kaufmann Publishers, Inc., San Francisco, CA, 1998.
5. **MITCHELL, T.:** Machine Learning, McGraw-Hill, 1997.
6. **JADE (Java Agent DEvelopment Framework):** <http://jade.cse.it.it>
7. **FIPA (Foundation for Intelligent Physical Agents):** <http://www.fipa.org>
8. **GUO, Y., J. P. MÜLLER, C. WEINHARDT:** Learning User Preferences for Multi-attribute Negotiation: An Evolutionary Approach. În: Multiagent Systems and Applications III, LNAI 2691, Springer, 2003, pp. 303-313.
9. **MATOS, N., C. SIERRA:** Evolutionary Computing and Negotiating Agent. În: Agent Mediated Electronic Commerce, Springer, LNAI 1571, 1999, pp. 126-150.
10. **FARATIN, P., C. SIERRA, N. R. JENNINGS:** Negotiation Decision Functions for Autonomous Agents. În: International Journal of Robotics and Autonomous Systems, Vol. 24, No. 1, 1998, pp. 159-182.
11. **OPREA, M.:** The Use of Adaptive Negotiation by a Shopping Agent in Agent-Mediated Electronic Commerce. În: Multiagent Systems and Applications III, LNAI 2691, Springer, 2003, pp. 594-605.