

# **MODELAREA TIPARELOR DE COMPORTAMENT EMERGENT ÎN MEDII COLABORATIVE ROBOTIZATE ȘI REȚELE DE SENZORI INTELIGENȚI**

Teză destinată obținerii  
titlului științific de doctor inginer  
la  
Universitatea "Politehnica" din Timișoara  
în domeniul Știința Calculatoarelor  
de către

**Răzvan-Dorel CIOARGĂ**

Conducător științific:  
Referenți științifici:

prof.univ.dr.ing Vladimir-Ioan CREȚU  
prof.univ.dr.ing. Mircea PETRESCU  
prof.univ.dr.ing. Sergiu NEDEVSCI  
conf.univ.dr.ing. Ștefan HOLBAN

Ziua susținerii tezei: 25.11.2011

Seriile Teze de doctorat ale UPT sunt:

- |                        |   |
|------------------------|---|
| 1. Automatică          | 7. Inginerie Electronică și Telecomunicații |
| 2. Chimie              | 8. Inginerie Industrială                    |
| 3. Energetică          | 9. Inginerie Mecanică                       |
| 4. Ingineria Chimică   | 10. Știința Calculatoarelor                 |
| 5. Inginerie Civilă    | 11. Știința și Ingineria Materialelor       |
| 6. Inginerie Electrică |   |

Universitatea „Politehnica” din Timișoara a inițiat seriile de mai sus în scopul diseminării expertizei, cunoștințelor și rezultatelor cercetărilor întreprinse în cadrul școlii doctorale a universității. Seriile conțin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susținute în universitate începând cu 1 octombrie 2006.

Copyright © Editura Politehnica – Timișoara, 2011

Această publicație este supusă prevederilor legii dreptului de autor. Multiplicarea acestei publicații, în mod integral sau în parte, traducerea, tipărirea, reutilizarea ilustrațiilor, expunerea, radiodifuzarea, reproducerea pe microfilme sau în orice altă formă este permisă numai cu respectarea prevederilor Legii române a dreptului de autor în vigoare și permisiunea pentru utilizare obținută în scris din partea Universității „Politehnica” din Timișoara. Toate încălcările acestor drepturi vor fi penalizate potrivit Legii române a drepturilor de autor.

România, 300159 Timișoara, Bd. Republicii 9,  
tel. 0256 403823, fax. 0256 403221  
e-mail: editura@edipol.upt.ro

## Cuvânt înainte

Mulțumesc domnului prof. dr. ing. Vladimir-Ioan CREJU, conducătorul științific de doctorat, pentru ajutorul și îndrumarea acordate atât pe întregul parcurs al activității mele de doctorat, cât și în toți anii de colaborare.

Activitatea mea de cercetare pentru doctorat a beneficiat de sprijinul colegilor din cadrul Universității „Politehnica” din Timișoara și din cadrul Departamentului de Calculatoare dar, în special, de ajutorul colectivului de cercetare din cadrul laboratorului DSPLabs. Îi mulțumesc domnului conf. dr. ing. Mihai V. MICEA pentru sprijinul continuu acordat încă de pe vremea studenției. Sfaturile sale și discuțiile purtate au constituit punctul de pornire al unor idei din teza de față. Menționez și ajutorul prețios și constant acordat de colegul ing. Dan CHICIUDEAN. Lista poate continua; mulțumesc tuturor colegilor și prietenilor pentru susținerea morală și materială acordată.

Recunoștința mea adâncă se îndreaptă către părinții mei, Floarea și Dorel, care prin sacrificiile făcute și dragostea lor m-au modelat în ceea ce sunt azi.

Nu în ultimul rând, mulțumesc bunului Dumnezeu pentru familia mea, pentru sănătatea mea, pentru succesele mele... pentru tot ceea ce mă reprezintă și mă înconjoară.

Timișoara, noiembrie 2011

Răzvan-Dorel CIOARGĂ

*"O furnică în mișcare face mai multe decât un bou care doarme"*

Lao Tze, sec V î.e.n.

*Familiei mele*

CIOARGĂ, Răzvan-Dorel

**MODELAREA TIPARELOR DE COMPORTAMENT EMERGENT  
ÎN MEDII COLABORATIVE ROBOTIZATE ȘI REȚELE DE  
SENZORI INTELIGENȚI**

Teze de doctorat ale UPT, Seria 10, Nr. 37, Editura Politehnica,  
2011, 152 pagini, 62 figuri, 5 tabele.

Cuvinte cheie:

Comportament emergent, colective robotizate, rețele de senzori  
inteligenti

Rezumat,

Utilizarea unor echipe de roboți pentru efectuarea unor activități complexe cum ar fi explorarea, monitorizarea și operarea în medii necunoscute sau periculoase pentru om prezintă o serie de avantaje esențiale: substituția intervenției umane, autonomie de operare pe termen lung, precizie foarte bună și constantă, creșterea eficienței prin paralelizare, creșterea fiabilității sistemului în ansamblu prin redundanță și distribuția resurselor. Coordonarea întregului sistem poate ajunge extrem de complex, dar studiul sistemelor naturale poate conduce la dezvoltarea unor soluții extrem de eficiente bazate pe aplicarea tiparelor de comportament emergent. Prin urmare, s-a dezvoltat un cadru teoretic pentru reprezentarea tiparelor de comportament emergent. Modelul de comportament emergent este utilizat pentru determinarea numărului minim de entități necesare pentru îndeplinirea unei sarcini, precum și pentru determinarea timpului necesar îndeplinirii sarcinii. Modelul matematic a fost validat teoretic prin implementarea în utilitarul Mathematica și prin simularea într-un simulator special creat. În continuare, modelul a fost adaptat, implementat și testat într-un colectiv robotic real, folosind roboți LEGO Mindstorm NXT. Au fost efectuate serii de experimente pentru validarea experimentală a modelului. Rezultatele obținute experimental sunt consecvente cu cele obținute din simulări dar și cu cele teoretice obținute prin aplicarea directă a modelului.

# CUPRINS

|  |    |
|--|----|
| Cuprins.....   | 5  |
| Notății, abrevieri, acronime .....   | 9  |
| Listă tabele .....   | 11 |
| Listă Figuri .....   | 13 |
| 1. Introducere .....   | 15 |
| 1.1 Scopul și obiectivele tezei .....  | 16 |
| 1.2 Scurtă descriere a tezei .....   | 17 |
| 2. Comportament emergent. Principii .....  | 19 |
| 2.1 Introducere.....   | 19 |
| 2.2 Definierea comportamentului emergent .....   | 20 |
| 2.3 Sisteme distribuite naturale și artificiale .....  | 22 |
| 2.4 Exemple de comportament emergent.....  | 23 |
| 2.4.1 Exemple de comportament emergent în natură.....  | 23 |
| 2.4.2 Exemple de comportament emergent în domeniul fizicii .....                                     | 24 |
| 2.4.3 Exemple de comportamente emergente în domeniul socio-cultural.....                             | 24 |
| 2.4.4 Inteligența roiurilor .....  | 25 |
| 2.4.5 Mișcarea bancurilor de pești .....   | 28 |
| 2.5 Concluzii.....   | 30 |
| 3. Stadiul actual al dezvoltării mediilor colaborative cu inteligență emergentă și distribuită ..... | 33 |
| 3.1 Introducere.....   | 33 |
| 3.2 Sisteme distribuite de comunicație bazate pe feromoni.....                                       | 34 |
| 3.3 Sisteme SPRAY cu auto-organizare .....   | 36 |
| 3.3.1 Dezvoltarea de aplicații distribuite pentru sisteme SPRAY .....                                | 36 |
| 3.3.1.1 Proiectarea directă a aplicațiilor pentru sistemele cu auto-organizare..                     | 37 |
| 3.3.1.2 Proiectarea inversă pentru sisteme cu auto-organizare .....                                  | 39 |
| 3.3.1.3 Controlul comportamentului emergent .....  | 41 |
| 3.4 Tehnologiile "roi" ale NASA .....  | 41 |
| 3.4.1 Misiunile ANTS.....  | 41 |
| 3.4.2 SARA : The Saturn Autonomous Ring Array .....  | 42 |
| 3.4.3 PAM: Prospective Asteroid Mission.....   | 42 |
| 3.5 Rețele de senzori inteligenți .....  | 43 |
| 3.5.1 MicaZ / Mica2 .....  | 43 |
| 3.5.2 Cricket.....   | 43 |
| 3.5.3 TelosB .....   | 44 |
| 3.6 Metaeuristici .....  | 44 |
| 3.6.1 A.C.O. – Ant Colony Optimization.....  | 45 |
| 3.7 Mișcarea în colective robotice.....  | 46 |
| 3.7.1 Modelul lui Aoki .....   | 46 |
| 3.7.2 Alte modele.....   | 48 |
| 3.8 Concluzii.....   | 49 |
| 4. Probleme specifice mediilor colaborative robotizate .....   | 51 |
| 4.1 Introducere.....   | 51 |
| 4.2 Caracteristici ale entităților unui mediu colaborativ robotizat .....                            | 52 |
| 4.2.1 Energia consumată.....   | 52 |
| 4.2.2 Robustețe .....  | 53 |

## 6 Cuprins

---

|   |     |
|---|-----|
| 4.2.3 Securitate .....  | 53  |
| 4.2.4 Putere computațională .....   | 53  |
| 4.2.5 Sincronizarea .....   | 54  |
| 4.2.6 Mărime și cost .....  | 55  |
| 4.3 Caracteristici ale rețelelor de senzori fără fir .....                                | 55  |
| 4.3.1 Durata de viață .....   | 55  |
| 4.3.2 Acoperirea .....  | 56  |
| 4.3.3 Timp de răspuns .....   | 56  |
| 4.4 Probleme ale sistemelor distribuite clasice .....                                     | 56  |
| 4.5 Probleme ale rețelelor de senzori inteligenți .....                                   | 57  |
| 4.6 Probleme legate de aplicarea tiparelor de comportament emergent .....                 | 58  |
| 4.7 Concluzii .....   | 59  |
| 5. Modelarea comportamentului emergent .....  | 61  |
| 5.1 Studiul și conceperea modelului .....   | 61  |
| 5.1.1. Conceperea modelului .....   | 61  |
| 5.1.2 Descrierea mediului modelului .....   | 63  |
| 5.2 Metrici și parametri ai modelului .....   | 63  |
| 5.3 Funcția scop .....  | 66  |
| 5.4 Modelul de comportament emergent .....  | 68  |
| 5.5 Concluzii .....   | 71  |
| 6. Implementarea modelului de comportament emergent în Mathematica .....                  | 73  |
| 6.1 Validarea modelului de comportament emergent folosind Mathematica .....               | 73  |
| 6.1.1 Studiu de caz: Exemplu din lumea reală .....  | 73  |
| 6.1.2 Studiu de caz: Situația cea mai defavorabilă .....                                  | 74  |
| 6.1.3 Gradul de activitate al sistemului .....  | 76  |
| 6.2 Concluzii .....   | 77  |
| 7. Suportul pentru testarea și implementarea modelului pentru comportament emergent ..... | 69  |
| 7.1 Introducere .....   | 69  |
| 7.2 Modelul WIT .....   | 82  |
| 7.3 Modelul BRAIN .....   | 84  |
| 7.4 Modelul de comunicație .....  | 85  |
| 7.5 The Emergent Behavioral Modeling Language – eBML .....                                | 85  |
| 7.5.1 Simulatorul de comportament emergent .....  | 88  |
| 7.6 Concluzii .....   | 90  |
| 8. Simularea modelului matematic folosind eBMS .....                                      | 91  |
| 8.1 Introducere .....   | 91  |
| 8.2 Cadrul general pentru simulările făcute .....   | 91  |
| 8.3 Ocolirea de obstacole .....   | 92  |
| 8.3.1 Cadrul de organizare a simulării .....  | 92  |
| 8.3.2 Rezultatele obținute .....  | 93  |
| 8.3.3 Concluzii .....   | 95  |
| 8.4 Adunarea de resurse .....   | 95  |
| 8.4.1 Cadrul de organizare a simulării .....  | 95  |
| 8.4.2 Rezultatele obținute .....  | 96  |
| 8.4.3 Concluzii .....   | 98  |
| 8.5 Explorare emergentă .....   | 100 |
| 8.5.1 Cadrul de organizare a simulării .....  | 100 |
| 8.5.2 Rezultatele obținute .....  | 100 |
| 8.5.2.1 Fără adunare de resurse .....   | 100 |
| 8.5.2.2 Cu adunare de resurse .....   | 101 |

---

|   |     |
|---|-----|
| 8.5.3 Concluzii .....   | 104 |
| 8.5.3.1 Fără adunare de resurse .....   | 104 |
| 8.5.3.2 Cu adunare de resurse .....   | 104 |
| 8.6 Concluzii .....   | 105 |
| 9. Implementarea și testarea mișcării în colective robotizate bazată pe studiul coloniilor de furnici ..... | 109 |
| 9.1 Introducere .....   | 109 |
| 9.2 Tiparul de comportament emergent de tip PATRULARE .....   | 109 |
| 9.3 Implementarea tiparului de comportament emergent PATRULARE .....  | 109 |
| 9.3.1 Calculul parametrilor statici ai roboților LEGO .....   | 109 |
| 9.3.2 Calculul parametrilor tiparului de comportament emergent PATRULARE .....                              | 110 |
| 9.4 Concluzii .....   | 113 |
| 10. Implementarea și testarea mișcării în colective robotice bazată pe studiul bancurilor de pești .....    | 107 |
| 10.1 Conceperea modelului .....   | 107 |
| 10.1.1 Cazul 1: Entitatea nu prezintă nici un vecin în apropiere .....                                      | 117 |
| 10.1.2 Cazul 2: Entitatea are exact un vecin în apropiere .....   | 118 |
| 10.1.3 Cazul 3: Entitatea are mai mult de un vecin în apropiere .....                                       | 119 |
| 10.2 Modelul matematic .....  | 120 |
| 10.2.1 Evitarea obstacolelor .....  | 122 |
| 10.2.2 Metrici .....  | 123 |
| 10.2.3 Simularea modelului de mișcare în eBMS .....   | 125 |
| 10.3 Concluzii .....  | 127 |
| 11. Aplicarea tiparelor de comportament emergent în domeniul colectării datelor în rețele de entități ..... | 123 |
| 11.1 Introducere .....  | 123 |
| 11.2 Specificații generale ale ERRIE (Emergent Retriever of Information) .....                              | 132 |
| 11.3 Evaluarea performanțelor .....   | 133 |
| 11.4 Concluzii .....  | 135 |
| 12. Concluzii și perspective .....  | 137 |
| 12.1 Concluzii .....  | 137 |
| 12.2 Sinteza contribuțiilor .....   | 139 |
| 12.3 Perspective .....  | 140 |
| Lucrări publicate .....   | 141 |
| Bibliografie .....  | 143 |





## **NOTAȚII, ABREVIERI, ACRONIME**

|         |  |
|---------|--|
| ACO     | Ant Colony Optimization                                      |
| ANTS    | Autonomous Nano Technology Swarm                             |
| BRAIN   | Background Robotic Activity Induction Node                   |
| CORE-TX | Collaborative Robotic Environment – The Timișoara Experiment |
| DARPA   | Defense Advanced Research Projects Agency                    |
| eBML    | Emergent Behavior Modelling Language                         |
| eBMS    | Emergent Behavior Model Simulator                            |
| FTSP    | Flooding Time Synchronization Protocol                       |
| GSM     | Global System for Mobile communications                      |
| MAC     | Medium Access Control  |
| NTP     | Network Time Protocol  |
| P2P     | Peer-to-peer   |
| PAM     | Prospective Asteroid Mission                                 |
| RBS     | Reference Broadcast Synchronization                          |
| RSSI    | Radio Signal Strength Indicator                              |
| SARA    | The Saturn Autonomous Ring Array                             |
| SI      | Swarm Intelligence   |
| TPSN    | Time Synchronization Protocol for Sensor Networks            |
| UML     | Unified Modelling Language                                   |
| WIT     | Wireless Intelligent Terminal                                |
| XML     | Extensible Markup Language                                   |



## **LISTĂ TABELE**

|  |     |
|--|-----|
| Tabelul 1. Rezultatele simulărilor pentru ocolirea de obstacole.....             | 93  |
| Tabelul 2. Rezultatele simulărilor pentru adunarea de resurse .....              | 96  |
| Tabelul 3. Ratele de succes pentru experimentele legate de adunarea de resurse . | 99  |
| Tabelul 4. Ratele de succes pentru experimentele legate de adunarea de resurse   | 102 |
| Tabelul 5. Ratele de succes pentru experimentele legate de adunarea de resurse   | 104 |



## LISTĂ FIGURI

|   |     |
|---|-----|
| Fig. 1. Principiul de bază al comportamentului emergent.....  | 21  |
| Fig. 2. Algoritm de căutare a hranei de către coloniile de furnici .....                                | 26  |
| Fig. 3. Cale neîntreruptă între mușuroi și hrană .....  | 26  |
| Fig. 4. Obstacol care întrerupe calea de feromoni între mușuroi și hrană.....                           | 27  |
| Fig. 5. Furnicile încep să ocolească obstacolul.....  | 27  |
| Fig. 6. Toate furnicile aleg calea cea mai scurtă de ocolire a obstacolului .....                       | 28  |
| Fig. 7. Deplasarea peștilor în funcție de localizarea vecinului .....                                   | 29  |
| Fig. 8. Mișcarea de bază a bancului de pești .....  | 30  |
| Fig. 9. Sistem de particule cu poziționare aleatoare.....   | 38  |
| Fig. 10. Sistem de particule în stadii incipiente de poziționare.....                                   | 38  |
| Fig. 11. Sistem de particule poziționare în formă circulară.....  | 39  |
| Fig. 12. Model nesimetric de poziționare a particulelor .....   | 40  |
| Fig. 13. Misiunea PAM ([Truszkowski 2004, Truszkowski 2006]) .....                                      | 42  |
| Fig. 14. Parametrii de tip senzorial pentru modelul lui Aoki.....                                       | 47  |
| Fig. 15. Obstacol pe calea de feromoni .....  | 62  |
| Fig. 16. Mediul modelului .....   | 64  |
| Fig. 17. Numărul de entități care se află pe cele 2 căi.....  | 74  |
| Fig. 18. Numărul de entități care se află pe cele 2 căi.....  | 74  |
| Fig. 19. Numărul de entități în situația cea mai defavorabilă.....                                      | 75  |
| Fig. 20. Numărul de entități în situația cea mai defavorabilă.....                                      | 75  |
| Fig. 21. Gradul de activitate al sistemului în cazul aglomerării masive .....                           | 76  |
| Fig. 22. Gradul de activitate al sistemului în cazul aglomerării minime.....                            | 77  |
| Fig. 23. Modelul CORE-TX ([Cioarga 2006]) .....   | 80  |
| Fig. 24. Modelul CORE-TX – zone [Cioarga 2006].....   | 81  |
| Fig. 25. Modelul WIT [Cioarga 2006] .....   | 82  |
| Fig. 26. Reprezentarea vizuală a eBML [Cioarga 2007b].....  | 86  |
| Fig. 27. Mediul simulat pentru ocolirea de obstacole.....   | 92  |
| Fig. 28. Rezultatele simulărilor pentru 1 WIT (a), 2 WIT-uri (b), 5 WIT-uri (c) și 9 WIT-uri (d) .....  | 95  |
| Fig. 29. Numărul de pași corespunzător entităților simulate care au încheiat cu succes simularea .....  | 96  |
| Fig. 30. Rezultatele simulărilor pentru 1 WIT (a), 4 WIT-uri (b), 8 WIT-uri (c) și 16 WIT-uri (d) ..... | 98  |
| Fig. 31. Numărul de pași corespunzător entităților simulate care au încheiat cu succes simularea .....  | 99  |
| Fig. 32. Rata de succes.....  | 100 |
| Fig. 33. Caz de eșec la explorare emergentă: acoperire insuficientă a mediului... ..                    | 101 |
| Fig. 34. Caz de succes la explorare emergentă: acoperire satisfăcătoare a mediului .....                | 101 |
| Fig. 35. Rezultatele simulărilor pentru 1 WIT (a), 2 WIT-uri (b), 5 WIT-uri (c) și 10 WIT-uri (d) ..... | 103 |
| Fig. 36. Rata de succes.....  | 105 |
| Fig. 37. Tiparul de comportament emergent de tip <i>PATRULARE</i> .....                                 | 108 |
| Fig. 38. Parametrii mișcării diferențiale [Cioarga 2008].....   | 109 |
| Fig. 39. Raza de vedere .....   | 111 |
| Fig. 40. Unghiul de vedere mărit.....   | 112 |
| Fig. 41. Corecția rutei de patrulare.....   | 112 |
| Fig. 42. Starea inițială a sistemului .....   | 116 |
| Fig. 43. Cele trei cazuri posibile de poziționare .....   | 117 |

## 14 Listă figuri

---

|  |     |
|--|-----|
| Fig. 44. Mișcarea spre grup a unei singure entități .....  | 117 |
| Fig. 45. Poziționarea relativă a bancului și direcțiile alese.....   | 118 |
| Fig. 46. Repulsia în sistemul de 2 pești, din perspectiva entității din dreapta .....  | 119 |
| Fig. 47. Repulsia în sistemul de 2 pești, din perspectiva entității din stânga.....  | 119 |
| Fig. 48. Zone de percepție .....   | 120 |
| Fig. 49. Determinarea direcției de mișcare pentru cele 3 regiuni: a. Evitarea, b. Orientarea paralela și c. Atracția .....                     | 121 |
| Fig. 50. Calcularea gradului de coeziune și polarizarea dintr-un grup de 6 roboți care se mișcă fără influențe din exterior (obstacole).....   | 124 |
| Fig. 51. Calcularea gradului de coeziune și polarizarea unui grup de 6 roboți în timp ce efectuează manevra de evitare "fountain effect" ..... | 125 |
| Fig. 52. Calcularea gradului de coeziune și polarizarea unui grup de 6 roboți în timp ce efectuează manevra de evitare "flash expansion".....  | 125 |
| Fig. 53. Variația polarizării pentru simularea modelului Aoki folosind 5 entități pește .....  | 126 |
| Fig. 54. Variația gradului de coeziune pentru simularea modelului Aoki folosind 5 entități pește.....  | 126 |
| Fig. 55. Variația polarizării pentru 5 entități pește.....   | 127 |
| Fig. 56. Variația gradului de coeziune folosind 5 entități pește .....   | 127 |
| Fig. 57. Traiectoriile roboților în timpul unor scenarii experimentale .....   | 128 |
| Fig. 58. Arhitectura sistemului ERRIE .....  | 132 |
| Fig. 59. Numărul de pagini descărcate de sistemul WIRE. ....   | 133 |
| Fig. 60. Numărul de pagini web descărcate prin modificarea numărului de boti ERRIE.....  | 134 |
| Fig. 61. Numărul de pagini web descărcate de 1 bot ERRIE prin modificarea zonei tampon. ....   | 134 |
| Fig. 62. Timpul necesar descărcării paginilor de 1 bot ERRIE prin modificarea dimensiunii zonei tampon. ....                                   | 135 |

# 1. INTRODUCERE

Dezvoltarea, miniaturizarea și costul scăzut al sistemelor de procesare digitală a informației a dus la o extraordinară răspândire a echipamentelor numerice de control în majoritatea domeniilor de activitate umană, de la cercetarea spațiului cosmic, până la producțiile artistice.

Sistemele de control digital încorporat (embedded systems) [Micea 2006, Popa 2006], sistemele robotizate [Dai 2006, Cioarga 2007, Cioarga 2006], senzorii și echipamentele de percepție digitală (sensor networks) [Intel 2005, Ciubotaru 2007], achiziția și prelucrarea numerică a semnalelor și imaginilor (DSP-based and digital image processing systems) [Gai 2002, Stanescu 2007] sunt domenii de interes major în preocupările actuale ale comunității academice și industriale, fapt dovedit și de numărul imens de aplicații și proiecte ce se dezvoltă în aceste toate domenii. În ultimii ani, complexitatea aplicațiilor ce implică inteligența digitală încorporată a crescut într-o astfel de măsură încât s-a impus necesitatea de abordare și combinare a tuturor domeniilor amintite mai sus într-o manieră integratoare, prin definirea și implementarea de metodologii și tehnici specifice ce definesc noi domenii de cercetare-dezvoltare: medii colaborative robotizate, percepție artificială distribuită și rețele de senzori inteligenți.

Printre tendințele existente în momentul de față în industrie se numără și miniaturizarea extremă a echipamentelor precum și introducerea noțiunii de „pervasive computing”. „Pervasive computing” (sau calcul omniprezent) reprezintă un model de interacțiune om - calculator în care prelucrarea de informații a fost complet integrată în obiecte folosite în activități uzuale. În cadrul activităților obișnuite, cineva care se folosește de calculul omniprezent accesează mai multe dispozitive și sisteme de calcul simultan, și nu trebuie în mod necesar să fie conștient de faptul că acestea execută operațiunile cerute.

În cazul numărului extrem de mare de entități care apar în acest caz, coordonarea întregului sistem poate ajunge extrem de complexă. Studiul sistemelor naturale poate conduce la dezvoltarea unor soluții extrem de eficiente bazate pe aplicarea tiparelor de comportament emergent.

Emergența este procesul prin care se formează modele complexe pornind de la reguli simple. Fenomenele emergente apar datorită modelelor de interacțiune care se desfășoară între elementele constituente ale unui sistem pe o perioadă îndelungată de timp. Fenomenele emergente sunt în mod uzual nepredictibile, reprezentând rezultate netriviale ale unor interacțiuni simple, între entități simple.

Proprietățile emergente sunt extrem de neliniare și nondeterministice ceea ce previne în mod excepțional demonstrarea acestora pe baza unor principii matematice. Aceste fenomene pot fi demonstrate doar prin experimente empirice, uneori extrem de complicat de reprodus deoarece condițiile de declanșare ale tiparelor de comportament emergent sunt extrem de complexe.

Comportamentul emergent sau proprietatea emergentă poate apărea când un număr de entități simple (agenți) operează într-un mediu formând comportamente mai complexe ca și un întreg [Anthony 2004, SN 2005, Tanenbaum 2002, Privosnik 2002]. Acest comportament complex nu este suma comportamentelor individuale și nici nu poate fi dedus sau prezis pe baza acestor comportamente individuale ale entităților de nivel scăzut.

În domeniul natural, emergența apare exclusiv în sistemele cu o populație numeroasă, în special datorită interacțiunii dintre entitățile componente ale acestora. Această caracteristică a proprietăților emergente a condus la studiul

posibilității aplicării acestora într-un domeniu în care sistemele cu populație mare de entități care interacționează între ele sunt extrem de uzuale: sistemele distribuite. Urmărind tendințele actuale de evoluție ale sistemelor distribuite s-a ajuns la concluzia că domeniul cel mai recomandat pentru aplicarea tiparelor de comportament emergent este cel al senzorilor inteligenți și al rețelelor de senzori inteligenți.

Senzorii inteligenți reprezintă o extensie a senzorilor tradiționali pentru că prezintă facilități avansate de învățare și adaptare. Senzorul inteligent trebuie să prezinte și o capacitate de calcul mărită pentru a putea executa operațiile complexe care îi sunt destinate [Intel 2005, Ciubotaru 2007, Dorigo 1997a].

Prin senzor inteligent se înțelege, în contextul științific actual, orice senzor care încorporează o mare parte din capacitatea de comandă și control care până acum era dedicată procesoarelor sau microcontrolerelor. Din punct de vedere al structurii unui astfel de sistem, un senzor inteligent nu mai conține doar partea specifică de monitorizare a unei mărimi ci și partea de control dedicată. Astfel se dorește îndepărtarea controlului unor astfel de senzori de la microprocesoarele / microcontrolerelor de uz general, care aveau asociate operații consumatoare de timp și de resurse, și încorporarea acestuia la nivelul senzorilor.

Pentru comunicarea cu astfel de senzori inteligenți se folosesc în mod curent tehnici de comunicare specifice rețelelor de senzori.

Utilizarea tiparelor de comportament emergent rezolvă o serie de probleme curente din domeniul mediilor colaborative robotizate și al rețelelor de senzori inteligenți:

- Consumul de energie, în special cel datorat transmisiei / recepției radio pentru sistemele fără fir;
- Probleme legate de identificarea drumului optim în condiții extreme de constrângere;
- Costurile ridicate ale unor dispozitive încorporate care necesită unități de procesare de putere mare.

## 1.1 Scopul și obiectivele tezei

Activitatea de doctorat abordează o arie tematică multidisciplinară ce acoperă domenii de cercetare fundamentală și aplicativă prioritare atât din perspectiva strategiei CDI naționale, cât și a celei europene:

- explorarea mediului (environment exploration),
- sisteme colaborative robotice (robotic collectives, collaborative robotic environments),
- rețele de senzori inteligenți (intelligent sensor networks),
- percepție artificială distribuită (distributed artificial perception),
- comportament emergent (emergent behavior),
- modelare matematică și formalizare (mathematical modeling and formalization).

Scopul principal al activității de doctorat este conceperea unui model matematic de reprezentare a comportamentului emergent în medii colaborative robotizate și rețele de senzori inteligenți. Aplicarea tiparelor de comportament emergent rezolvă o serie de probleme specifice sistemelor încorporate precum consumul de energie sau necesitatea unei puteri computaționale mărite. Modelul matematic formalizează aceste tipare și, pe baza acestora, calculează o serie de parametri precum numărul de entități necesare pentru declanșarea unui anumit tip



de comportament emergent sau timpul necesar pentru atingerea scopului propus folosind respectivele tipare.

Scopul și principalele obiective ale tezei își propun soluționarea totală sau parțială a problemelor importante existente în prezent în domeniu prin contribuțiile originale în domenii de cercetare fundamentală și aplicativă ce includ:

- Studiul și evaluarea comparativă a tiparelor de comportament colaborativ și emergent, a modelelor de reprezentare a acestora și a metodologiilor aplicabile mediilor asociative, precum și al soluțiilor de implementare a serviciilor de comunicație, localizare și sincronizare în rețelele de senzori;
- Dezvoltarea cadrului teoretic și metodologic în domeniu prin conceperea și realizarea modelului destinat reprezentării, studiului și dezvoltării sistemelor colaborative robotice și de senzori inteligenți;
- Definirea unor metrici pentru evaluarea performanțelor modelului de comportament emergent precum numărul de entități, afinitatea și gradul de activitate;
- Dezvoltarea și validarea limbajului formal eBML (Emergent Behavioral Modeling Language), conceput special pentru reprezentarea tiparelor comportamentului emergent;
- Integrarea limbajului eBML și a fundamentelor matematice de reprezentare a interacțiunilor, pentru dezvoltarea și validarea unui model coerent de formalizare a tiparelor de comportament colaborativ/asociativ;
- Validarea teoretică a modelului de comportament emergent folosind suita de aplicații Mathematica;
- Proiectarea și implementarea unui simulator complex pentru măsurarea și evaluarea performanțelor modelului de formalizare a tiparelor de comportament colectiv/asociativ;
- Implementarea modelului matematic studiat într-un colectiv robotic real și validarea sa experimentală;
- Studiul implementării și adaptării modelului matematic pentru un set de aplicații specifice mediilor colaborative robotizate (de ex. ocolirea de obstacole, explorarea mediilor necunoscute sau colectarea de resurse);
- Studiul implementării și adaptării modelului matematic pentru alte tipuri de aplicații decât cele specifice mișcării în mediile colaborative robotizate (colectarea de date, comunicații).

## 1.2 Scurtă descriere a tezei

Prima parte a tezei de față conține capitole în care se descriu principiile de bază al comportamentului emergent și exemple de tipare de comportament emergent. Totodată se prezintă și principiile de bază ale rețelelor de senzori și ale senzorilor inteligenți, precum și o prezentare a stadiului actual.

Capitolul 2 prezintă definiția și principiile de bază ale comportamentului emergent precum și cele mai uzuale exemple de tipare de comportament emergent existente în natură, în domeniul social sau în cel științific. Se mai prezintă studiul și evaluarea celor mai cunoscute tipare de comportament emergent (inteligenta roiurilor și mișcarea bancurilor de pești).

Capitolul 3 conține o prezentare scrută a stadiului actual al dezvoltării mediilor colaborative cu inteligență emergentă și distribuită (sisteme de comunicație bazate pe feromoni, metaeuristici etc.). Se descriu sisteme și medii colaborative prezente în domeniul comunicațiilor și al sistemelor încorporate. Se prezintă

principiile de aplicare a tiparelor de comportament emergent în matematică (metaeuristici) și la mișcarea în colective robotice (bazată pe studiul fie al coloniilor de furnici, fie al bancurilor de pești).

Următorul capitol, capitolul 4, conține o prezentare a problemelor existente în domeniile mediilor colaborative robotizate și al rețelelor de senzori inteligenți.

Partea a doua conține prezentarea modelului folosit pentru specificarea și analiza tiparelor de comportament emergent. Se mai prezintă, totodată, și validarea teoretică a modelului folosind suita de aplicații de calcul Mathematica.

Capitolul 5 introduce modelul matematic pentru reprezentarea, analiza și evaluarea comportamentului emergent. Totodată, acest capitol mai conține și modalitatea de concepere a modelului matematic, specificațiile utilizării tiparelor de comportament emergent. Se introduc aici parametri calculați de model precum numărul de entități necesare pentru declanșarea comportamentului emergent sau timpul atingerii scopului, și metricile folosite la evaluarea modelului precum afinitatea și gradul de activitate.

În capitolul 6 se prezintă validarea formală a acestui model folosind suita de aplicații pentru calcule complexe, Mathematica.

Capitolul 7 prezintă suportul folosit pentru evaluarea și validarea modelului matematic, atât experimental cât și prin simulări: sistemul integrat CORE-TX, simulatorul eBMS, limbajul de specificare a comportamentului emergent eBML.

În capitolul 8 se prezintă simulările efectuate pentru evaluarea și validarea modelului folosind un simulator de comportament emergent special conceput, eBMS. Folosind simulatorul eBMS, împreună cu un limbaj de reprezentare a tiparelor de comportament emergent, s-a realizat un set de simulări care au validat modelul matematic introdus. Simulările au fost efectuate prin varierea numărului de entități simulate, prin modificarea tiparelor de comportament emergent folosite și prin modificarea mediului simulat.

În ultima parte a tezei se prezintă implementarea și testarea tiparelor de comportament emergent în medii colaborative robotizate și rețele de senzori.

Următoarele capitole studiază mișcarea în medii colaborative robotizate din două puncte de vedere: un tip de mișcare bazat pe studiul coloniilor de furnici, într-un capitol, și un alt tip de mișcare bazat pe studiul bancurilor de pești, în capitolul următor. În acest capitol se mai introduc două metrici, polarizarea și gradul de coeziune, care descriu performanțele aplicării acestor tipare la mișcarea în colective robotice.

Tiparele de comportament emergent au o largă arie de folosire și, de aceea, capitolul care urmează studiază implementarea acestor tipare și în alte domenii decât cel al mișcării roboților, dar care sunt totuși aplicabile mediilor colaborative. Acest capitol prezintă aplicarea tiparelor de comportament emergent în domeniul web și a comunicației într-un mediu colaborativ.

Ultimul capitol, capitolul 12, prezintă concluziile și perspectivele acestei teze precum și o sinteză a contribuțiilor acestei lucrări.

## 2. COMPORTAMENT EMERGENT. PRINCIPII

Aplicarea tiparelor de comportament emergent în domeniul mediilor colaborative robotizate și al rețelelor de senzori inteligenți a condus la creșterea performanțelor acestor sisteme, fie că este vorba despre reducerea costurilor asociate prin reducerea puterii computaționale necesare, fie că este vorba de îmbunătățirea timpului de viață prin reducerea consumului de energie implicat. De asemenea, studiul coloniilor de furnici și aplicarea cunoștințelor astfel dobândite în domeniul matematic a condus la dezvoltarea unor soluții specifice, performante pentru rezolvarea problemelor de optimizare combinatorială (de exemplu: problema comis-voiajorului).

Capitolul de față prezintă o descriere succintă a principiilor comportamentului emergent. Se definește comportamentul emergent și se prezintă o serie de exemple de tipare de comportament studiate în diferite domenii: natură, mediul social, mediul economic etc. În continuare se prezintă două tipare de comportament emergent preluate din studiul coloniilor de furnici și a roiurilor de albine (care țin de domeniul inteligenței roiurilor – eng. swarm intelligence) și, respectiv, din studiul mișcării bancurilor de pești.

Tiparele de comportament emergent prezentate în acest capitol vor fi folosite în conceperea modelului de comportament emergent prezentat în capitolele care urmează. Totodată, respectivele tipare de comportament emergent vor fi utilizate la implementarea, validarea și testarea modelului definit precum și la definirea unor metrici specifice pentru evaluarea modelului respectiv.

### 2.1 Introducere

Mișcarea, și, în special, evitarea obstacolelor, a fost și este în continuare o dispută complexă în domeniul colectivelor robotizate [SN 2005] și [Tanenbaum 2002]. Sistemele mari centralizate, care folosesc echipamente cu suficientă putere de calcul, sunt capabile să execute algoritmi de coordonare complecși. Pe de altă parte, colectivele robotizate prezintă unele constrângeri legate de dimensiune, consum de energie și preț [Privosnik 2002] ceea ce le împiedică să întrebuițeze acești algoritmi performanți și eficienți. Dimensiunile reduse ai roboților ieftini, care de regulă, alcătuiesc aceste colective favorizează utilizarea modelelor de comportament emergent inspirate din studiul naturii [Dorigo 1997a, Dorigo 1997b]. Asemenea modele de comportament emergent precum deplasarea generală și manevrele de eschivare descoperite la bancurile de pești s-au dovedit a fi extrem de eficiente în ceea ce privește mișcarea / evitarea obstacolelor a unui număr mare de entități de dimensiuni mici.

Printre tiparele de comportament emergent cele mai cunoscute și folosite în literatura de specialitate se numără cele preluate din studiul coloniilor de furnici și al bancurilor de pești. În continuare se prezintă principiile care stau la baza celor două tipare de comportament emergent: inteligența roiurilor, tipar de comportament emergent preluat din studiul coloniilor de furnici și al roiurilor de albine, și mișcarea bancurilor de pești.

Fiecare din aceste tipare de comportament emergent are un domeniu de aplicabilitate bine definit. Tiparul de comportament emergent derivat din inteligența roiurilor se folosește în aplicații de explorare de medii și colectare de resurse și

permite ocolirea eficientă a obstacolelor. Tiparul de comportament emergent preluat din mișcarea bancurilor de pești este folosită pentru colectarea de resurse dar în special pentru ocolirea de obstacole.

Similar cu vietățile studiate pentru deducerea tiparelor de comportament emergent, entitățile robotice din cadrul mediilor colaborative robotizate au o viziune locală asupra mediului în care se află, ei reacționează la stimulii pe care îi primesc din mediu. Cu ajutorul senzorilor, acestea detectează obstacolele și hrana. Pe urmă, transformă informațiile din mediu în mesaje pe care le transmit celorlalți roboți. Fiecare robot are o experiență locală, o bază de cunoștințe pe care o partajează cu ceilalți roboți. Astfel, prin globalizarea bazei de cunoștințe, roboții pot să ia decizii mai bune decât dacă s-ar folosi doar de cunoștințele lor particulare.

## 2.2 Definirea comportamentului emergent

Emergența este procesul prin care se formează comportamente complexe pornind de la reguli extrem de simple. Acest proces este dinamic, entitățile prezentând emergența pe o perioadă mai mare de timp (de exemplu stadiile succesive de evoluție ale structurilor pluricelulare de la organisme simple la un creier uman capabil de conștiință [Anthony 2004, Privosnik 2002]). Totodată, emergența poate apărea între entități de dimensiuni care diferă prin multe ordine de mărime (de exemplu interacțiunea dintre un număr mare de neuroni care are ca rezultat creierul uman capabil de gândire rațională, deși neuronii înșiși nu prezintă această capacitate [Anthony 2004, Truskowski 2004]).

Comportamentul emergent sau proprietatea emergentă poate apărea când un număr de entități simple (agenți) operează într-un mediu formând comportamente mai complexe ca și un întreg [Anthony 2004, Privosnik 2002, Truskowski 2004]. Acest comportament complex nu este suma comportamentelor individuale și nici nu poate fi dedus sau prezis pe baza acestor comportamente individuale ale entităților de nivel scăzut. Ca exemple de comportament emergent se pot aminti stolurile de păsări sau bancurile de pești.

În Fig. 1 se prezintă principiul de bază al comportamentului emergent în care comportamentul simplu al componentelor individuale creează un comportament mult mai complex decât se poate estima al sistemului.

Aceste proprietăți emergente se manifestă în mod total impredictibil, fără precedentă. În multe cazuri manifestarea comportamentului emergent reprezintă o nouă fază de evoluție a sistemului de entități.

Un motiv care poate explica apariția comportamentelor emergente este acela că numărul de interacțiuni între entitățile individuale crește exponențial cu numărul acestora. Acest fapt dă naștere posibilității apariției unor noi comportamente subtile care nu pot fi deduse sau prezise pe baza unui aparat matematic riguros datorită numărului extrem de mare de posibilități [Anthony 2004, Privosnik 2002].

Totuși un număr mare de interacțiuni între entități nu explică în mod individual emergența. S-au identificat situații în care numărul mare de entități nu a condus la emergență ci, dimpotrivă, a determinat eliminarea completă a comportamentului emergent din sistem. Totodată numărul mare de entități produc un nivel de zgomot proporțional care maschează semnalul util până la eliminarea acestuia.

Astfel se demonstrează că nu doar numărul mare de entități din sistem sau numărul mare de interconexiuni realizate dintre ele contează, ci și felul în care

aceste conexiuni sunt organizate: ierarhic, centralizat, descentralizat etc. Se pare, totuși că comportamentul emergent apare în special în cazul structurilor puternic descentralizate.

În mod uzual un sistem de entități distincte trebuie să atingă anumite praguri de diversitate, conectivitate și organizare pentru a manifesta comportamente emergente.

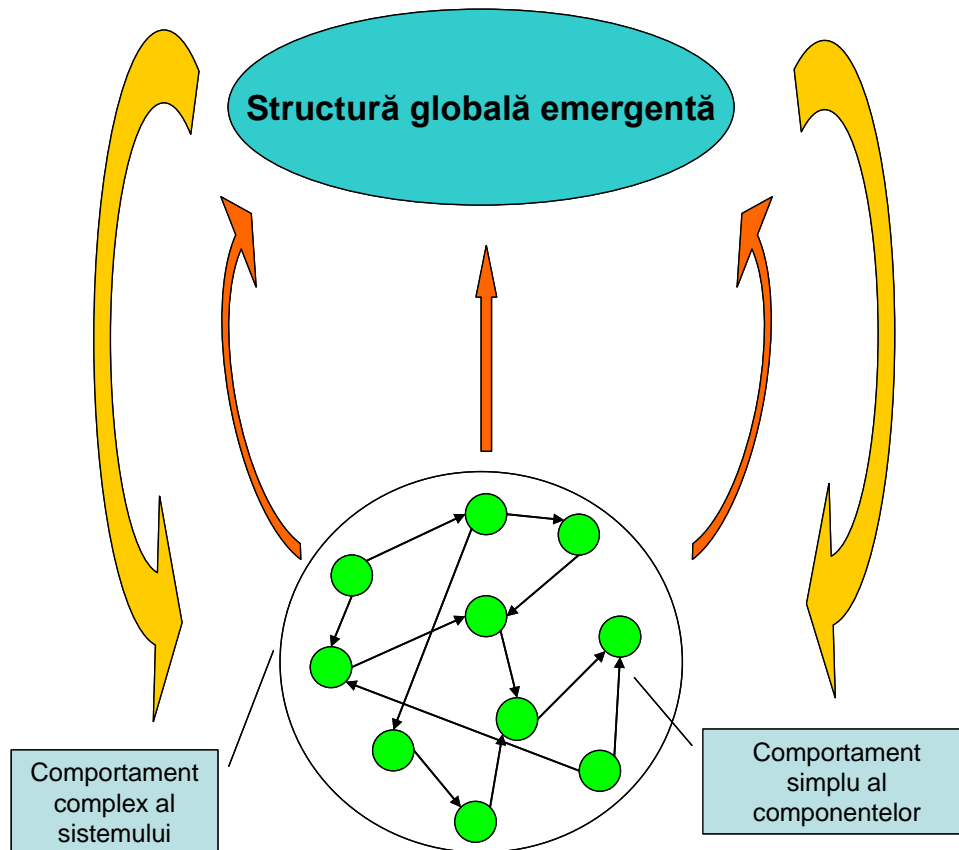


Fig. 1. Principiul de bază al comportamentului emergent

Sistemele care manifestă comportament emergent pot apărea că sfidează principiile entropiei sau legea a doua a termodinamicii deoarece se formează și își măresc dimensiunile în lipsa unor structuri de comandă și control centralizate. Acest lucru este posibil deoarece aceste sisteme deschise pot prelua informații direct din mediul lor.

Comportamentul emergent al sistemelor autonome împreună cu scara extremă a acestor sisteme conduc la imposibilitatea prezicerii tuturor scenariilor posibile pentru configurare sau pentru erori. Practic este imposibil de a crea strategii de conducere sau de recuperare pentru toate rezultatele posibile.

## 2.3 Sisteme distribuite naturale și artificiale

Datorită descentralizării accentuate a sistemelor care pot prezenta comportament emergent, acestea pot prezenta anumite atribute ale sistemelor de calcul distribuite [Tanenbaum 2002].

Comportamentul intern, tradițional al sistemelor distribuite este profund deterministic [Tanenbaum 2002]. Acesta mod de abordare introduce limitări cu privire la cerințele care nu fac referință la funcționalitate: eficiență, robustețe, scalabilitate. Intensitatea comunicațiilor sau complexitatea interacțiunilor cresc proporțional cu dimensiunea sistemului. O proiectare robustă a sistemului distribuit tinde să mărească intensitatea comunicației datorită nevoii de implementare de mecanisme de replicare sau de monitorizare a componentelor. Astfel, robustețea unui sistem distribuit împreună cu scalabilitatea acestuia sunt deseori puse în conflict cu eficiența sistemului [Tanenbaum 2002].

Sistemele distribuite naturale au cerințe non-funcționale similare cu sistemele de calcul distribuite. Astfel, pentru a supraviețui, coloniile de insecte trebuie să aibă un număr mare de indivizi care să îndeplinească diferite sarcini pentru apărare sau pentru obținerea mâncării. Datorită numărului mare de indivizi, aceste sisteme naturale trebuie să fie extrem de scalabile. Sistemele naturale trebuie să fie foarte robuste pentru a rezista cu succes amenințărilor mediului. Uzual, resursele din natură sunt limitate astfel că supraviețuirea pe termen lung implică o eficiență mărită a sistemului.

Comportamentul non-deterministic al acestor sisteme naturale este denumit emergent deoarece se formează din comportamentele multor agenți simpli care urmăresc un set limitat de reguli simple bazate doar pe viziunea locală a acestora asupra sistemului. Agenții individuali nu sunt conștienți de starea globală a sistemului sau de scopurile funcționale finale ale sistemului [Anthony 2004, Privosnik 2002, Truszkowski 2004, Truszkowski 2006].

Comunicația (interacțiunea) este imperativă în ambele tipuri de sisteme distribuite, atât cel deterministic, artificial cât și cel emergent, natural.

Comunicația în sistemele tradiționale distribuite are următoarele caracteristici [Tanenbaum 2002]:

- Comunicația este în mod uzual sincronă și bine structurată (se desfășoară pe un număr bine stabilit de runde);
- Fiecare mesaj este important iar pierderea oricărui mesaj individual duce la afectarea rezultatului. Pentru a proteja aceste mesaje este necesară introducerea de mecanisme de aprobare (acknowledgement), expirare (timeout) și retransmisie. Aceste mecanisme duc la mărirea necesității de bandă de transmisie, la mărirea latenței de comunicație și la mărirea timpului de procesare a mesajelor în cazul folosirii primitivelor cu blocare.
- Nodurile sunt identificate în mod unic. Adresele IP sunt folosite pentru a identifica în mod unic nodurile de rețea, iar în mod uzual, pentru a mări siguranța unicității nodurilor se folosesc numere unice de identificare generate de către noduri specializate pentru fiecare nod.
- Algoritmii folosiți la sistemele distribuite sunt deterministici. Predictibilitatea acestora este obținută prin modele de comunicație complexe și / sau intense.
- Sistemele bazate pe comunicație tind să folosească modele structurate de comunicație. Cele mai simple modele utilizate sunt datagramele, fiecărei

datagrame fiindu-i alocat un număr de secvență. În mod uzual, în aceste modele de comunicație se folosesc protocoale de tip cerere-răspuns.

În contrast cu aceste sisteme distribuite artificiale, sistemele naturale prezintă următoarele caracteristici [Anthony 2004, Privosnik 2002, Kashima 2004, Dorigo 1997b, Hinchey 2005]:

- Comunicația este asincronă și stabilită ad-hoc;
- Mesajele individuale sunt de mică importanță, apar doar ca urmare a întâlnirilor întâmplătoare și, ca urmare, pierderea unui mesaj are doar impact local și nu afectează performanțele globale ale sistemului.
- Nodurile din sistemele distribuite emergente sunt anonime.
- Sistemele distribuite naturale sunt non-deterministic ceea ce permite relaxarea intensității comunicației. Aceste sisteme prezintă comportamente care tind să fie predictibile deși comportamentul momentan, instantaneu nu poate fi prezis.
- Comunicarea în sistemele naturale distribuite este implicită și uni-direcțională.

Schimbul de feromoni este unul din cele mai utilizate moduri de comunicare în sistemele naturale care prezintă comportament emergent. Coloniile de furnici reprezintă un exemplu extrem de important al unui sistem descentralizat cu comunicare bazată pe feromoni. Furnicile individuale își formează o imagine localizată a întregului sistem bazată pe schimbul de feromoni dintre acestea și celelalte furnici pe care le întâlnesc. Diferențelor tipuri de feromoni le corespund diferite informații; de asemenea, furnicile extrag informații și din intensitatea feromonilor. Extrăgând informațiile din feromoni, furnicile determină comportamentul furnicilor vecine și de aici își pot deduce propriul rol [Anthony 2004, Maniezzo 2004], fără a mai stabili un dialog sau orice alt tip de negociere.

## **2.4 Exemple de comportament emergent**

### **2.4.1 Exemple de comportament emergent în natură**

Structurile emergente sunt modele care nu apar ca urmare a unei singure reguli sau a unei simple condiții. Nu există o structură de comandă care să genereze semnalele corespunzătoare pentru crearea comportamentului emergent. Acesta este determinat de interacțiunile care apar între părțile componente ale sistemului precum și între acestea și mediul înconjurător. Astfel se poate ajunge la concluzia că aceste sisteme care prezintă comportamente emergente sunt mai mult decât suma părților componente .

Unul dintre cele mai elocvente exemple de comportament emergent în domeniul biologic îl constituie coloniile de furnici. Regina nu dă comenzi directe anumitor grupuri de furnici pentru ca acestea să înceapă să caute hrană sau să apere mușuroiul. Fiecare furnică reacționează la stimuli chimici generați de larve, alte furnici, intruși etc. și, la rândul ei lasă o urmă chimică de feromoni care reprezintă un stimul pentru celelalte furnici [Anthony 2004]. Astfel, orice furnică reacționează doar la mediul local, iar regulile de comportament pentru fiecare tip de furnică sunt codate genetic.

În lipsa unui control centralizat furnicile prezintă un comportament extrem de complex care permite unei colonii de furnici să rezolve probleme de ordin geometric (în mod uzual furnicile identifică cea mai lungă distanță față de toate intrările în mușuroi pentru a depozita cadavrele ).

O serie de alte comportamente emergente în care comunicația între indivizi este într-un fel sau altul bazată pe feromoni sau se desfășoară într-un alt mod chimic sunt reprezentate de stoluri sau de roiuri: albine, termite etc.

În natură pot fi identificate o multitudine de comportamente emergente: stoluri de păsări, turme de animale, haite de lupi etc.

Galaxiile reprezintă o formă de comportament emergent având ca proprietate emergentă forma și structura spațială. Aceste proprietăți emergente apar ca urmare a distribuției materiei și energiei în Univers .

### **2.4.2 Exemple de comportament emergent în domeniul fizicii**

În fizică, comportamentul emergent descrie o proprietate care este validă la nivel microscopic, în spațiu și timp, însă nu este validă la nivel microscopic deși, la nivel macroscopic sistemul poate fi văzut ca o uniune a entităților microscopice.

În continuare se prezintă o serie de exemple de proprietăți fizice care pot fi descrise în termeni de comportament emergent:

- Culoare: Particulele elementare precum electronii sau protonii nu prezintă culori; doar în momentul în care se aranjează în atomi, aceștia prezintă proprietatea de a absorbe sau de a emite lumină cu lungimi de undă specifice.
- Frecare: Particulele elementare precum electronii sau protonii nu prezintă nici un fel de frecare, forțele care apar între aceste particule sunt conservative. Frecarea apare la structurile macroscopice care prezintă suprafețe care sunt în stare să absoarbă energia disipată la frecarea cu o altă suprafață.
- Vâscozitate, elasticitate: sunt definite asemănător cu proprietățile descrise mai sus.
- Mecanica clasică în raport cu mecanica cuantică prezintă o serie de proprietăți emergente în sensul că aceste proprietăți emergente sunt derivate la limită din mecanica cuantică dacă se ia în considerare o masă suficient de mare.
- Tranzițiile de fază la care este supusă materia: topire, înghețare, sublimare etc. În mod uzual, grupurile mici de entități (atomi) nu prezintă tranziții de fază la modificări ale temperaturii. Starea de agregare a obiectelor macroscopice depinde în mod direct de nivelul temperaturii interne și a mediului exterior.
- În mecanica cuantică, masa, spațiul și timpul prezintă proprietăți inerent emergente odată cu aprofundarea teoriei stringurilor: pentru a califica o entitate macroscopică sunt de ajuns determinarea poziției, impulsului etc., pe când din punct de vedere a mecanicii cuantice adevărata stare a materiei este descrisă printr-o multitudine de funcții de undă, care nu prezintă o poziție sau un impuls unice și / sau bine determinate (principiul incertitudinii a lui Heisenberg).

### **2.4.3 Exemple de comportamente emergente în domeniul socio-cultural**

Domeniile social, cultural, tehnic conțin o mulțime de exemple de comportamente emergente:

- Bursa de acțiuni: văzută ca un întreg bursa de acțiuni are proceduri de auto-reglare, însă nu are nici un control central. Fiecare actor din bursa de acțiuni



cunoaște un număr limitat de alte entități care activează pe piață și trebuie să urmărească un set de reguli bine stabilite.

- Jocuri: prezintă un profund caracter emergent (jocurile de cărți, deși se supun unor reguli stricte sunt extrem de aleatoare; chiar și în aceleași condiții ale mediului extern, folosind același set de reguli, jocul de cărți curent poate diferi fundamental de cel urmărit).

#### 2.4.4 Inteligența roiurilor

Societățile de insecte sociale-furnici, albine, termite și viespi-sunt sisteme distribuite în care comportamentul general al coloniei reiese din interacțiunile dintre indivizii acesteia [McLurkin 2004, Rothermich 2004].

Considerând exemplul termitelor, la nivel individual, ele au un nivel scăzut de inteligență. Lucrează fără supraveghere și totuși la nivel de colonie au rezultate spectaculoase. Construiesc mușuroaie care sunt adevărate minuni ale naturii, capabile să păstreze în interior un nivel optim de oxigen și dioxid de carbon, căldură și umiditate chiar în condițiile în care mușuroiul se modifică. Oamenii de știință au descoperit că la nivelul unei colonii cooperarea este auto-organizată: în majoritatea situațiilor se realizează prin interacțiunile dintre indivizi. Deși aceste interacțiuni pot părea simple (o furnică urmând calea altei furnici), ele pot rezolva probleme complexe (găsirea celui mai scurt drum spre o sursă de hrană). Cu referire la insectele sociale, acest comportament colectiv emergent este numit "inteligenta roiurilor" (swarm intelligence).

Swarm Intelligence este o formă de inteligență artificială bazată pe comportamentul colectiv al sistemelor distribuite, auto-organizate. Această expresie a fost introdusă pentru prima dată de către Gerardo Beni și Jing Wang în 1989 în contextul sistemelor celulare de roboți. O altă definiție, dată de către Bonabeau, Dorigo, Theraulaz este: "orice încercare de a proiecta algoritmi sau echipamente distribuite inspirate din comportamentul colectiv al coloniilor de insecte sociale sau alte societăți de animale" [Bonabeau 1997].

Sistemele bazate pe SI sunt formate din agenți ce interacționează local unii cu alții și cu mediul înconjurător. Agenții urmează reguli foarte simple și chiar dacă nu există o structură de control care să dicteze comportamentul fiecărui agent, interacțiunile locale conduc la emergența comportamentului global, complex. Exemplele naturale de SI includ coloniile de furnici, stolurile de păsări, bancurile de pești, roiurile de albine, turmele de animale.

În [Wee 2004, Bonabeau 1997] s-a demonstrat că șirurile de furnici se formează datorită feromonilor lăsați de către acestea pe traseul pe care se deplasează. Feromonii sunt substanțe chimice care atrag alte furnici. Deneubourg a mai demonstrat că procesul de marcare cu feromoni este o strategie foarte bună de a găsi cel mai scurt drum dintre mușuroi și sursa de hrană. Prin intermediul unor experimente cu furnici, Deneubourg a construit un pod cu două ramificații, una dintre ele fiind de două ori mai lungă decât cealaltă. El și-a dat seama că furnicile se împrăștie pe căi diferite atunci când caută mâncarea. Fiecare furnică lasă feromoni care atrag alte furnici. Prima furnică ce se întoarce acasă cu hrana este cea care a găsit cel mai scurt drum în ambele sensuri. Deoarece această rută este prima care a fost marcată cu feromoni în ambele sensuri, celelalte furnici vor fi atrase spre ea.

Comportamentul eficient al coloniei este datorat respectării a 2 reguli simple de către fiecare: depozitarea feromonilor și urmărirea celorlalți indivizi.

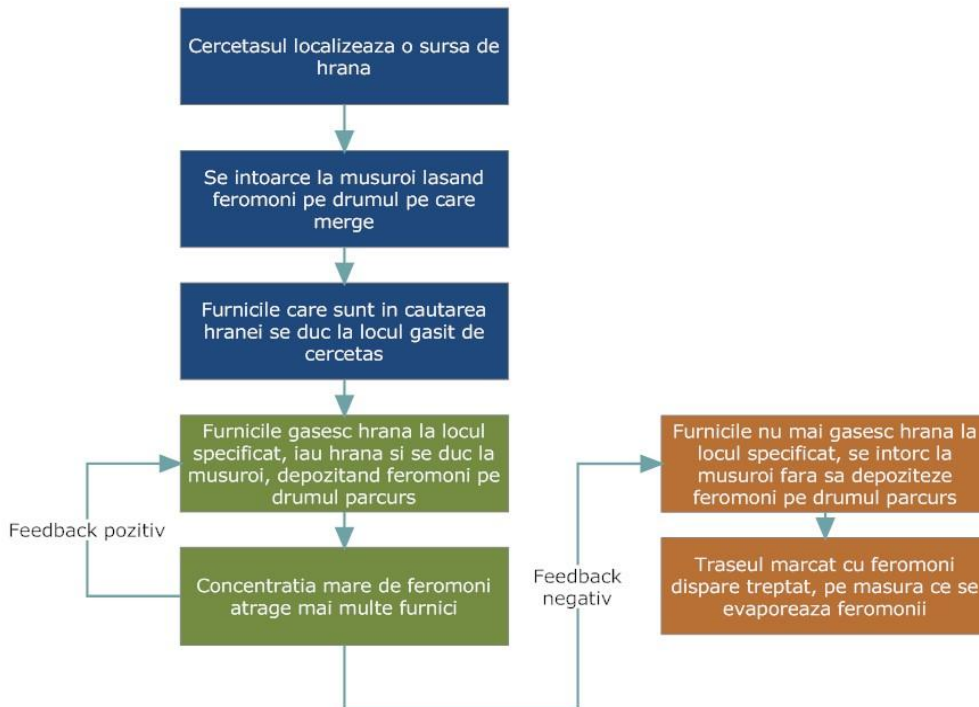


Fig. 2. Algoritm de căutare a hranei de către coloniile de furnici

Studiul coloniilor de furnici aflate atât în lumea reală, cât și în laborator au determinat o serie de fapte extrem de importante pentru aplicarea acestui studiu într-un mediu computațional. Furnicile reale sunt capabile de a identifica drumul cel mai scurt aflat între mușuroi și sursa de hrană, fără a se folosi de dovezi vizuale. Aceste furnici sunt totodată capabile de a se adapta la schimbările de mediu, precum invalidarea unui drum cunoscut între mușuroi și o sursă de hrană prin apariția unui obstacol.

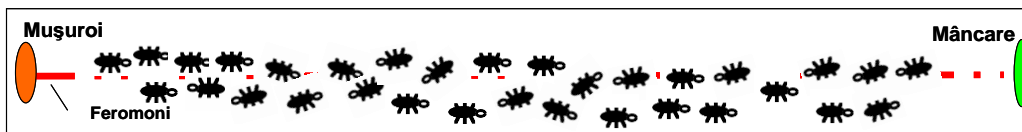


Fig. 3. Cale neîntreruptă între mușuroi și hrană

În figura de mai sus, furnicile se mișcă în linie dreaptă între sursa de hrană și mușuroi, nu există nici un obstacol în calea lor.

Principalul mod de comunicație la furnici constă în folosirea feromonilor. Furnicile depun o anumită cantitate de feromoni pe obiectele de interes sau chiar în timpul mersului, marcând astfel calea pe care o urmează. Ulterior, furnicile preferă să urmeze direcțiile bogate în urme de feromoni, alegând în majoritatea cazurilor direcțiile cele mai bogate și cele mai recente în feromoni. Feromonii secretați de furnici au capacitatea de a expira în timp ceea ce permite faptul ca o direcție

nefolosită să nu mai fie luată în considerare în cele din urmă. Acest comportament elementar al furnicilor este folosit pentru a explica metoda prin care furnicile pot identifica în continuare drumul cel mai scurt care conectează două puncte întrerupte ale căii de feromoni prin plasarea unui obstacol, după cum se vede în figura următoare.

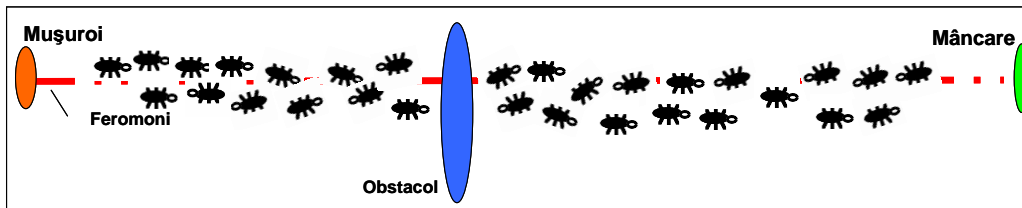


Fig. 4. Obstacol care întrerupe calea de feromoni între mușuroi și hrană

Odată ce obstacolul apare, acele furnici care sunt chiar înaintea acestuia nu mai pot să urmărească calea de feromoni, și de aceea trebuie să ia o decizie legată de direcția în care ar trebui să se miște: stânga sau dreapta. În mod probabilistic jumătate din furnici vor alege să ocolească obstacolul prin dreapta, iar jumătate vor alege să ocolească obstacolul prin stânga. O situație similară se poate găsi de cealaltă parte a obstacolului, după cum este reprezentat în figura următoare.

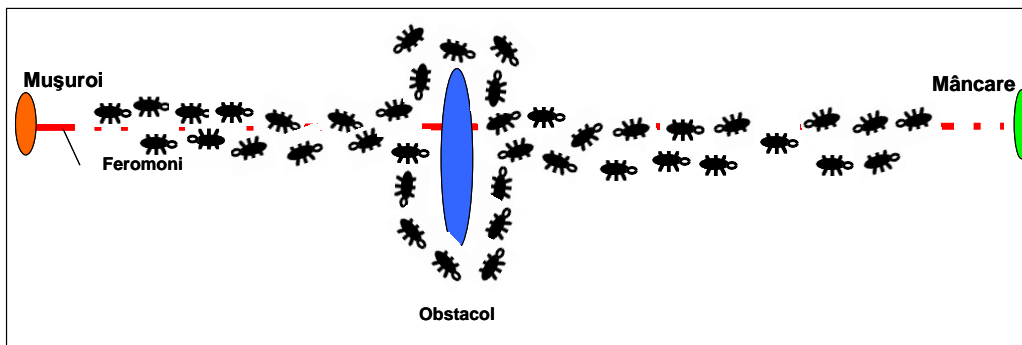


Fig. 5. Furnicile încep să ocolească obstacolul

Acele furnici care au decis, datorită șansei, calea mai scurtă de ocolire a obstacolului vor reconstitui mai rapid calea de feromoni întreruptă decât furnicile care au ales calea mai lungă. Astfel, pe calea mai scurtă se va depune o cantitate mai mare de feromoni în unitatea de timp decât pe calea mai lungă ceea ce va determina un număr mai mare de furnici să aleagă calea respectivă, fapt care la rândul său va conduce la creșterea cantității de feromoni depusă. Datorită acestui proces autocatalitic (de răspuns pozitiv, "positive feedback"), toate furnicile vor alege calea mai scurtă foarte repede, după cum se vede în figura următoare.

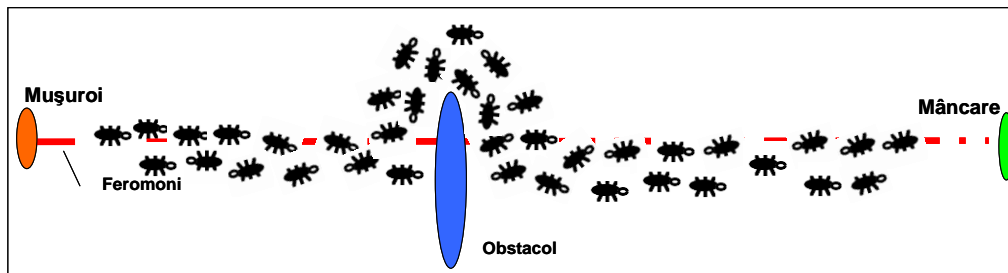


Fig. 6. Toate furnicile aleg calea cea mai scurtă de ocolire a obstacolului

Un aspect interesant al acestui proces autocatalitic este faptul că identificarea drumului cel mai scurt care ocolește obstacolul este o proprietate emergentă a interacțiunii dintre forma obstacolului, modul în care așezat obstacolul și comportamentul distribuit al furnicilor. Deși toate furnicile se mișcă aproximativ cu aceeași viteză, și depun aproximativ aceeași cantitate de feromoni în unitatea de timp, faptul că un drum este mai scurt, conduce la acumularea mai rapidă a cantității de feromoni. Acest lucru combinat cu preferința naturală a furnicilor pentru urmarea direcțiilor bogate în feromoni și cu proprietatea de expirare a feromonilor conduce la faptul că într-un interval scurt de timp calea mai lungă este abandonată în întregime.

#### 2.4.5 Mișcarea bancurilor de pești

Bancurile de pești se caracterizează printr-o puternică coeziune și deplasare paralelă fără să aibă vreun lider. Cercetările au arătat că aceștia se deplasează folosind în special vederea și anumite organe sensibile la mișcarea și vibrația apei din jur. Acestea sunt folosite pentru a afla distanța și viteza vecinilor apropiați. Peștii sunt capabili să stea împreună păstrând o orientare paralelă care dictează direcția comună de deplasare pentru întregul grup. Grupul poate astfel să acopere o distanță mare într-un timp scurt.

Pentru a explica mișcarea acestora s-au creat numeroase modele. Una dintre teorii propune folosirea interacțiunii dintre forțele de atracție și cele de respingere cu ajutorul cărora se dirijează mișcarea indivizilor. [Aoki 1982]

O altă teorie a fost propusă de Aoki și mai târziu revizuită de Huth și Wissel. Acest model pleacă de la mai multe premise [Huth 1992, Bhoosan 2001, Parrish 2002]:

- fiecare membru al grupului folosește aceleași reguli pentru deplasare, astfel că nu este nevoie de nici un lider;
- bancul de pești se deplasează independent de stimuli externi;
- mișcarea fiecărui pește este influențată doar de vecinii săi cei mai apropiați;

Aoki a considerat că în jurul fiecărui pește există patru zone cu ajutorul cărora se măsoară apropierea sau depărtarea de cel mai apropiat vecin. În funcție de posibila localizare a vecinului, acesta are patru alternative: respingere, orientare paralelă, atracție și căutare. Alternativa aleasă oferă unghiul de influență și modalitatea de comportare față de acest vecin. [Bhoosan 2001].

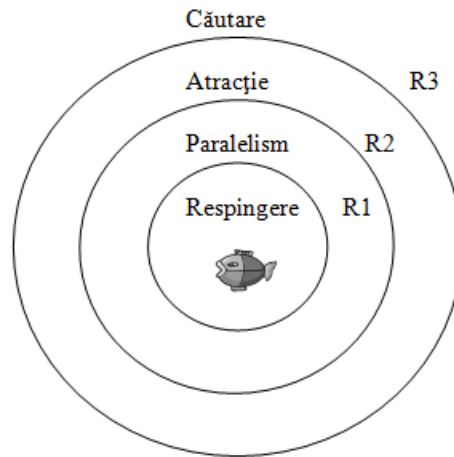


Fig. 7. Deplasarea peștilor în funcție de localizarea vecinului

Dacă vecinul este la o distanță mai mică decât R1 în raport cu peștele, atunci peștele va arăta un comportament de respingere și va înota perpendicular pe direcția de deplasare a vecinului pentru a evita o coliziune. Astfel, unghiul de influență este  $\pm 90^\circ$ . Dacă vecinul este între R1 și R2 atunci peștele va înota în aceeași direcție cu vecinul, unghiul de influență fiind cel al vecinului. Dacă este localizat între R2 și R3, peștele va încerca să se apropie de vecin, arătând un comportament de atracție. Peștele va înota în direcția vecinului.

Huth și Wissel au adăugat ultimul tipar modelului. Dacă vecinul este la o distanță mai mare ca R3, peștele nu poate percepe vecinul deoarece simțurile sale sunt limitate. Astfel că peștele va manifesta un comportament de căutare, mergând aleator.

Atât Aoki, cât și Huth și Wissel au folosit distribuțiile de probabilitate pentru a determina unghiul de întoarcere pe care peștele îl va lua în considerare în funcție de gradul de incertitudine legat de detectarea poziției și orientării vecinului [62-64].

Ultimul element al modelului este modul în care peștii răspund influențelor exercitate de ceilalți vecini. Aoki a folosit un concept decizional cu ajutorul căruia peștele alege care este vecinul pe care îl va urma. Fiecare vecin primește un anumit factor în funcție de importanța pentru peștele respectiv, iar unghiul de întoarcere se calculează din două sau mai multe distribuții ponderate. Dacă vecinul 1 este mai în față decât vecinul 2, importanța distribuției sale va fi de două ori mai mare decât a vecinului 1, astfel că peștele va urma vecinul numărul 1. Huth și Wissel au propus un concept în care peștele face o combinație între influențele vecinilor săi luând media unghiului de influență al fiecărui vecin, iar probabilitatea de distribuție este doar una și constă în distribuția normală față de unghiul mediu.

Conform modelului prezentat în [Aoki 1982] și revizuit mai târziu în [Huth 1992], bancurile de pești nu au un lider. Fiecare membru al grupului se mișcă după aceleași reguli ca și ceilalți.

De asemenea, mișcările peștelui sunt influențate numai de cei mai apropiați vecini. Fiecare pește poate aproxima vizual distanța dintre el și vecinii săi, iar înaintarea vecinilor și viteza printr-un senzor natural pe care îl are un pește, numit linia laterală care poate sesiza mișcarea curenților de apă din apropierea peștelui.

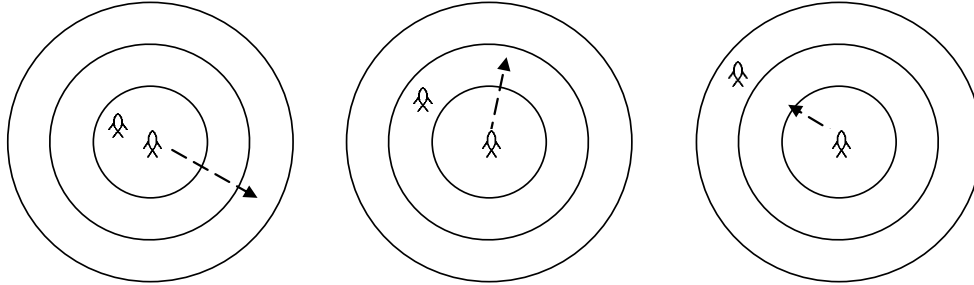


Fig. 8. Mișcarea de bază a bancului de pești

Conform cu [Aoki 1982], fiecare pește este înconjurat de patru zone distincte, așa cum se arată în Fig. 7. În funcție de poziția relativă a peștilor vecini față de peștele principal, există patru cazuri, după cum urmează (Fig. 8):

- Dacă un pește vecin este situat în cercul cel mai apropiat din jurul peștelui principal, următoarea lor mișcare va fi de repulsie deoarece sunt prea aproape și există posibilitatea unei coliziuni.
- Dacă vecinul peștelui este situat în cel de-al doilea cerc, atunci se consideră că distanța dintre el și peștele principal este optimă; peștele principal va urma direcția vecinului descriind o direcție paralelă.
- În cel de-al treilea caz, peștele vecin este în cel mai îndepărtat cerc; peștele principal, care calculează următoarea mișcare, va încerca să se deplaseze în funcție de poziția vecinului său.
- Când peștele vecin este situat în afara cercului cel mai îndepărtat, atunci nu va fi luat în considerare (ca în viața reală, nu poate fi văzut).

Când nu este nici un pește în apropiere, peștele principal va adopta o strategie de căutare pentru a găsi grupul pierdut. Când un pește are mai mulți vecini, el poate să i-a în considerare doar unul dintre ei (pe baza factorului distanță și greutate primiți de la început de fiecare pește), sau i-a în considerare toți vecinii săi.

## 2.5 Concluzii

În acest capitol s-au prezentat principiile de bază ale comportamentului emergent precum și o serie de exemple semnificative de tipare de comportament emergent.

S-a definit comportamentul emergent și s-au prezentat exemplele cele mai semnificative de tipare de comportament emergent din domenii variate: natură, domeniul socio-economic etc. S-au mai prezentat două tipuri de tipare de comportament emergent folosite la conceperea, dezvoltarea și implementarea modelului de comportament emergent descris în restul lucrării: inteligența roiurilor, în special, referitor la mișcarea în grup, ocolirea de obstacole, explorare emergentă a mediului, și mișcarea specifică bancurilor de pești.

Capitolul următor conține o prezentare scrută a stadiului actual al dezvoltării mediilor colaborative cu inteligență emergentă și distribuită (sisteme de comunicație bazate pe feromoni, colective robotizate, medii robotice, metaeuristici etc.). Acesta conține descrieri succinte ale sistemelor prezente în literatura de specialitate care utilizează tiparele de comportament emergent descrise în acest capitol și prezintă

---

atât sisteme care folosesc tiparele de comportament emergent la modelarea mișcării entităților componente cât și sisteme matematice care folosesc comportamentul emergent la rezolvarea unor probleme de optimizare combinatorială.





### **3. STADIUL ACTUAL AL DEZVOLTĂRII MEDIILOR COLABORATIVE CU INTELIGENȚĂ EMERGENTĂ ȘI DISTRIBUITĂ**

Sistemele distribuite artificiale sunt întâlnite în toate domeniile de activitate. Datorită tendinței generale de miniaturizare a acestor sisteme, este necesară rezolvarea anumitor probleme precum durata de viață, asigurarea comunicațiilor fără fir sau reducerea costurilor generate de reducerea dimensiunilor. Aplicarea tiparelor de comportament emergent prezentate în capitolul precedent reprezintă o soluție posibilă la aceste probleme.

Capitolul de față prezintă o serie de sisteme din literatura de specialitate care aplică tipare de comportament emergent pentru rezolvarea anumitor probleme.

Într-o secțiune a acestui capitol se prezintă sisteme de comunicație care folosesc feromonul coloniilor de furnici pentru implementarea mesajelor semi-persistente. În continuare se prezintă sisteme care folosesc comportamentul emergent pentru poziționare sau pentru explorarea mediilor necunoscute. Se mai prezintă, totodată, și aplicarea tiparelor de comportament emergent în domeniul matematicii, la rezolvarea unor probleme de optimizare combinatorială (problema comis-voiajorului). Capitolul mai conține și prezentarea celor mai importante aplicări ale tiparelor de comportament emergent în domeniul mișcării în colective robotice în special mișcarea bazată pe studiul coloniilor de furnici și a bancurilor de pești.

Principiile evidențiate de metaeuristicile prezentate în acest capitol au fost utilizate de autor la conceperea modelului de comportament emergent. Totodată, ideile care au stat la baza sistemelor care utilizează tiparele de comportament emergent specifice bancurilor de pești au fost îmbunătățite prin adaptarea modelului de comportament emergent. Aceste îmbunătățiri împreună cu evaluarea performanțelor lor sunt prezentate în capitolele care urmează.

#### **3. 1 Introducere**

Sistemele distribuite, deși au ajuns într-o fază de maturitate, prezintă o serie de probleme importante atunci când se încearcă aplicarea principiilor acestora la rețele de senzori, în principal din cauza numărului foarte mare de entități implicate care conduce la o creștere exponențială a cerințelor computaționale precum și a complexității algoritmilor folosiți. Totodată, rețelele de senzori sunt formate din entități care prezintă capacități reduse de comunicare: lățime de bandă mică, raza de transmisie mică (în cazul rețelelor de senzori fără fir).

În prezent există mai multe abordări ale problemei coordonării colectivelor robotice. Abordarea tradițională este cea a coordonării centralizate care are avantajul unui control foarte precis al activității fiecărui robot cât și a unei planificări precise a operațiilor efectuate. De asemenea pentru o mai bună coordonare, un sistem robotizat poate fi combinat cu o rețea de senzori. Dezavantajul principal îl reprezintă scalabilitatea redusă și necesitatea comunicării unei cantități foarte mari de date. La polul opus se află soluțiile distribuite, în general de tip comportamental, care asigură o scalabilitate bună a sistemului dar nu asigură o strategie bine determinată de operare și nici posibilitatea de învățare automată a sistemului. În

[Singh 2002] este prezentat un sistem robotic de explorare care se bazează pe o coordonare inspirată din sistemul imunitar uman.

Sistemele distribuite prezintă o serie de algoritmi extremi de performanți care rezolvă pentru sistemele distribuite artificiale uzuale o serie de probleme legate de: consistență și replicare, sincronizare, servicii de nume, comunicație, pierdere de mesaje, fiabilitate și retransmisie. Algoritmii care rezolvă aceste probleme sunt extrem de complecși, necesită o putere computațională mare (de ordinul sistemelor de calcul PC) și o lățime de bandă mare, folosită mai ales în cazul retransmisiei de mesaje.

În general, sistemele de rețele de senzori prezintă următoarele caracteristici:

- Număr mare de noduri;
- Puterea computațională redusă;
- Lățime de bandă redusă;

Datorită acestor caracteristici, folosirea tiparelor de comportament emergent este recomandată în cazul rețelelor de senzori și al senzorilor inteligenți. Pentru a identifica avantajele și dezavantajele folosirii de tipare de comportamente emergente se prezintă un studiu comparativ între soluțiile de comportament emergent existente. Printre soluțiile prezentate se numără:

- Comunicația bazată pe feromoni în sistemele distribuite: prin studiul coloniilor de furnici și identificarea modului de comunicație între furnicile din același mușuroi s-a dezvoltat un model de comunicație destinat sistemelor distribuite [Adjie 1999].
- Computere SPRAY: reprezintă un concept de rețele de sisteme de calcul, de dimensiuni suficient de mici pentru a fi incluse într-o suspensie lichidă care, apoi, să fie efectiv „vopsită” pe o suprafață a unui obiect pentru a-i mări acestuia funcționalitatea [Kashima 2004, Zambonelli 2004].
- Smart Dust: proiectul Smart Dust inițiat la Berkeley și sponsorizat de agenția DARPA încearcă aducerea rețelelor de senzori la un nou nivel tehnologic [Smart 2005a, Smart 2005b, Darpa 2005].

După prezentarea comparativă a soluțiilor de comportament emergent se prezintă posibilitățile de utilizare a acestor soluții pentru a surmonta problemele întâlnite la rețelele de senzori inteligenți.

### **3.2 Sisteme distribuite de comunicație bazate pe feromoni**

Comportamentul emergent, caracterizat drept non-deterministic al sistemelor naturale poartă această denumire deoarece se formează din comportamentele multor agenți simpli care urmăresc un set limitat de reguli simple bazate doar pe viziunea locală a acestora asupra sistemului. Agenții individuali nu sunt conștienți de starea globală a sistemului sau de scopurile funcționale finale ale sistemului [Anthony 2004, Adjie 1999, Camazine 2001, Kube 2000, Couzin 2002].

Comunicația (interacțiunea) este imperativă în ambele tipuri de sisteme distribuite, atât cel deterministic, artificial cât și cel emergent, natural. În domeniul natural, studiul coloniilor de furnici a relevat faptul că schimbul de feromoni joacă un rol foarte important în comunicația în interiorul mușuroiului.

În sistemele naturale, schimbul de feromoni este un element important care generează tipare de comportament emergent pentru întregul sistem. Furnicile individuale își formează o imagine localizată a întregului sistem bazată pe schimbul

de feromoni dintre acestea și celelalte furnici pe care le întâlnesc. Diferențelor tipuri de feromoni le corespund diferite informații; de asemenea, furnicile extrag informații și din intensitatea feromonilor. Extrăgând informațiile din feromoni, furnicile determină comportamentul furnicilor vecine și de aici își pot deduce propriul rol [Adjie 1999], fără a mai stabili un dialog sau orice alt tip de negociere.

Schimbul de feromoni prezintă o modalitate simplă de comunicație prin care informația este transmisă folosind o serie de mesaje chimice. Entitatea care recepționează mesajul poate extrage informație chiar din însăși mesajul chimic (tipul feromonului folosit indică tipul mesajului). Din mesajul chimic se pot extrage însă informații suplimentare (meta-data) cum ar fi importanța mesajului (interpretată din frecvența de transmisiei feromonului) sau cât de recent este mesajul (dedus din puterea chimică a feromonului).

În contextul comunicării în sisteme distribuite, comunicația pe baza feromonilor poate fi văzută ca având o serie de parametri similari cu cei asociați comunicației digitale:

- Tipul feromonului este analog cu tipul mesajului;
- Puterea chimică a feromonului este asociată cu vârsta mesajului;
- Rata de transmisie a feromonului este asociată importanței mesajului, priorității mesajului;
- Lățimea de bandă este irelevantă în cazul comunicației bazate pe feromoni deoarece aceștia sunt într-o mare măsură persistent;

Comunicația bazată pe feromoni este în mare măsură semi-fiabilă deoarece mesajele chimice continuă să fie transmise până când cauza care le-a determinat încetează să existe (în cazul furnicilor această situație ar putea fi eliminarea inamicilor, terminarea unei surse de hrană). Datorită acestui mod de prezentare a mesajelor pierderea unor mesaje individuale este fără consecință.

Acest tip de comunicație este simplex (unidirecțional) urmele de feromoni fiind create de furnici pentru ca alte furnici să le urmeze. Comunicația de tip duplex nu există, apare doar o pereche de mesaje simplex care însă nu sunt corelate ca parte a unei conversații.

Comunicarea naturală pe bază de feromoni este non-deterministică și totuși extrem de fiabilă, folosind un număr mic de mesaje simple, este stabilă, robustă și scalabilă.

Se încearcă astfel implementarea unui tip de comunicație destinat sistemelor distribuite în care mesajele să fie implementate ca feromoni; se elimină necesitatea stabilirii unei conexiuni între partenerii de comunicație. Dezavantajele folosirii unui sistem non-deterministic este determinarea empirică a condițiilor optime de funcționare: folosirea protocolului emergent de comunicație bazat pe feromoni nu este eficientă în cazul existenței a unui număr mic de entități, după cum nici un mușuroi de furnici nu supraviețuiește cu un număr limitat de indivizi. Totodată un număr exagerat de indivizi care comunică prin feromoni duce invariabil la gâtuirea rețelelor de interconexiune (în cazul coloniilor de furnici, când apare un număr mai mare de indivizi, colonia se împarte în două colonii, fiecare cu regina sa; colonia care aparține de regina nouă părăsește mușuroiul și construiește altul pe un alt teritoriu pentru a nu sufoca colonia originală).

### 3.3 Sisteme SPRAY cu auto-organizare

Ideea de computere SPRAY [Kashima 2004, Zambonelli 2004] a apărut mai întâi ca un concept de rețele futuristice în care acestea ar putea fi puse la dispoziția utilizatorilor sub formă de cutii de aerosoli pentru a fi aplicate pe diferite obiecte pentru a le mări funcționalitatea: haine inteligente, materiale auto-asamblante, auto-reparante.

Deoarece stadiul tehnologic actual face ca implementarea unei astfel de rețele futuristice să fie prea îndepărtată, principiile de miniaturizare, control descentralizat și auto-administrare ale sistemelor SPRAY sunt aplicate cu succes în domeniul rețelelor de senzori cu auto-organizare, formate ad-hoc: rețele de dispozitive mobile, rețele de micro-computere care se întind pe o suprafață de câțiva metri pătrați, rețele de dispozitive conectate la Internet care acoperă suprafețe mult mai mari (rețele metropolitane, backbone național etc.).

Principiile computerelor SPRAY sunt aplicate în toate domeniile rețelelor cu auto-organizare:

- Scară microscopică: computere SPRAY adevărate (proiectul Smart Dust – Praful Inteligent dezvoltat la Berkeley);
- Scară medie: computere handheld și haine inteligente (wearable computers);
- Scară globală: calcul global (întregul Internet este privit ca un sistem eterogen distribuit);

Astfel de sisteme trebuie instalate cu efort nul de pre-configurare datorită numărului mare de entități prezente. Astfel de sisteme descentralizate, autonome, care operează în medii dinamice și iau decizii pe baza informațiilor disponibile din aceste medii, prezintă în mod constant comportamente non-deterministice, neașteptate care sunt clasificate drept comportamente emergente.

#### 3.3.1 Dezvoltarea de aplicații distribuite pentru sisteme SPRAY

Dezvoltarea de aplicații distribuite pentru computere SPRAY presupune proiectarea unui comportament care derivă din cooperarea unui număr mare de componente, nefiabile, interconectate în modele necunoscute, dinamice. Acest fapt necesită dezvoltarea de algoritmi specifici care să permită rețelelor de computere SPRAY să se auto-organizeze pentru a atinge un scop comun fără o supraveghere a-priori sau un control centralizat.

Mecanismul de nivel inferior folosit pentru a crea proprietățile de auto-organizare este practic același, indiferent de scara folosită ( sisteme SPRAY sau rețele P2P):

- descoperirea dinamică de resurse (parteneri, servicii, informații) prin broadcast (dacă rețeaua pune la dispoziție servicii de broadcast, în caz contrar prin multicast sau unicast, în funcție de serviciile oferite de rețea [Kteam 2005, Monada 2005]);
- localizare și navigare [Ducatelle 2005, Di Caro 2005] (fie într-un spațiu fizic pentru sistemele SPRAY sau rețele de senzori, fie într-un spațiu digital pentru rețelele P2P).

Proiectarea aplicațiilor pentru sistemele SPRAY presupune folosirea unor tehnici speciale care țin cont de caracterul profund emergent al acestor sisteme.

Este extrem de important ca mecanismele implicate să fie cât mai simple, pentru ca sistemul să prezinte comportament emergent eficient, fiabil și robust.

Astfel de mecanisme simple, printre care se numără mecanismele de broadcast, localizare, replicare, navigare, împreună cu interacțiunile locale, necesare pentru a determina generarea tiparelor de comportament emergent, pot conduce la faptul ca anumite părți din sistem sau chiar întreg sistemul de tip SPRAY să ajungă la o stare globală coerentă.

Mecanismele simple amintite mai sus (broadcast, localizare, replicare etc.) conduc la apariția unor algoritmi distribuiți, simpli, eficienți, care prezintă tipare de comportament emergent dar care sunt diferiți în mod esențial de algoritmi distribuiți tradiționali. Algoritmii emergenți elimină necesitatea urmării principiilor de micro-management aplicate la algoritmi distribuiți tradiționali: ordonarea evenimentelor, sincronizarea proceselor, replicare, structura rețelelor. Algoritmii emergenți se bazează pe faptul că deși sistemele SPRAY sunt non-deterministice, totuși se poate demonstra o convergență către o stare generală stabilă în ciuda discrepanțelor de la nivel de micro-management.

În funcție de modalitatea de analiză și specificare a aplicațiilor pentru sistemele SPRAY (top-down sau bottom-up), se disting două tehnici de proiectare a acestor aplicații: proiectarea directă și proiectarea inversă.

### **3.3.1.1 Proiectarea directă a aplicațiilor pentru sistemele cu auto-organizare**

Un exemplu simplu de proiectare directă a aplicațiilor distribuite pentru sisteme de tip SPRAY este auto-localizarea distribuită. Un număr aleator de particule distribuite într-un mediu pot să își determine poziția geografică pornind de la un set de particule baliză (determinate prin algoritmi de selecție distribuiți) și apoi în mod recursiv să își determine poziția prin triangulație referitor la poziția particulelor vecine până se ajunge la o stare globală stabilă în care se cunosc pozițiile tuturor particulelor.

Exemplul următor se referă la poziționarea spațială a particulelor mobile în forme bine stabilite. Având un număr de particule distribuite într-un mediu ([Zambonelli 2004]), este posibilă proiectarea unui algoritm distribuit care, prin deplasarea locală a particulelor, să conducă la organizarea sistemului în forme dorite. Un algoritm simplu de alegere a unui lider care se poate organiza într-un centru de greutate a particulelor. Acest centru de greutate poate propaga apoi în toată rețeaua un „câmp gravitațional” care să atragă particulele către acest centru, organizând întreg sistemul într-o structură circulară.

Proiectarea directă a aplicațiilor distribuite pentru sistemele de tip SPRAY este strâns legată de existența a-priori a acestor sisteme SPRAY, pentru a identifica în mod corect tiparele de comportament emergent. Acest mod de lucru nu se poate aplica decât unui număr limitat de aplicații. Datorită complexității comportamentelor emergente și datorită caracteristicilor non-deterministice ale acestor sisteme, numărul de tipare de comportament emergent care pot fi puse în aplicare este extrem de mic, iar aceste comportamente emergente sunt simple. Totodată necesită cunoștințe a-priori despre configurațiile posibile ale sistemului care îi limitează capacitatea de adaptabilitate. Astfel, dacă pentru sistemul din [Smart 2005a, Smart 2005b] se propune o formă de poziționare nesimetrică, atunci pentru funcționarea corectă a sistemului SPRAY ar fi nevoie de un algoritm extrem de complex care să fie executat de toate particulele, sau ar fi nevoie ca anumite particule să dețină informații a-priori despre poziționarea acestora.

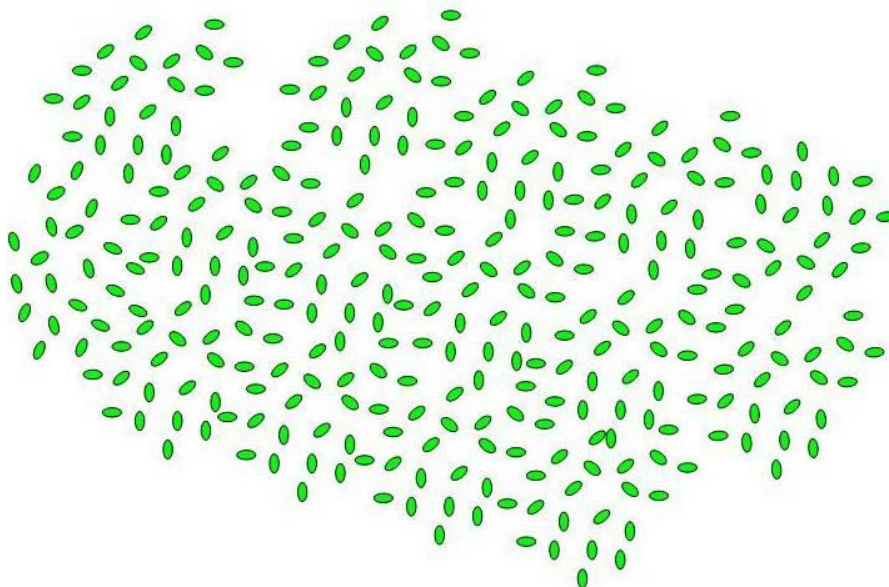


Fig. 9. Sistem de particule cu poziționare aleatoare

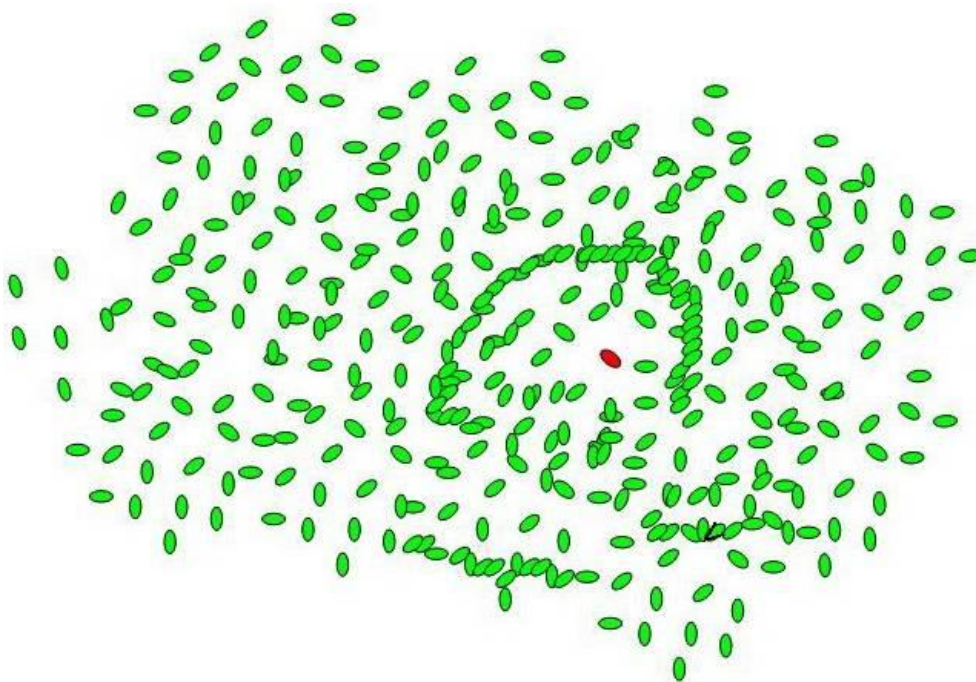


Fig. 10. Sistem de particule în stadii incipiente de poziționare

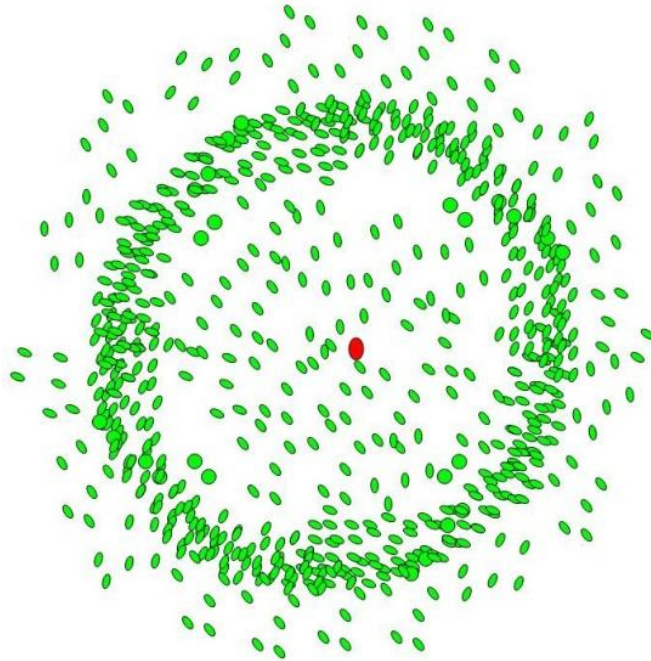


Fig. 11. Sistem de particule poziționare în formă circulară

### 3.3.1.2 Proiectarea inversă pentru sisteme cu auto-organizare

Proiectarea inversă pentru sistemele cu auto-organizare propune crearea de comportamente emergente prin recrearea condițiilor necesare dezvoltării unor astfel de comportamente. În aceste cazuri, proiectanții nu au nici un fel de control asupra dezvoltării ulterioare a sistemului, asupra evoluției sale viitoare datorită complexității comportamentelor emergente precum și datorită non-liniarității și non-determinismului pe care le implică. Proiectații sistemelor SPRAY se bazează în aceste condiții doar pe presupuneri de ordin probabilistic asupra tendințelor de evoluție ale sistemelor respective.

Simulările sunt elementul de bază al unor astfel de concepte de proiectare. Datorită comportamentului non-deterministic al sistemelor SPRAY, care este lipsit de suportul matematic pentru determinarea cu acuratețe a tendințelor de evoluție, simularea este singura posibilitate prin care se poate urmări într-o oarecare măsură evoluția sistemului. Astfel, faza de modelare în sistemele proiectate prin această metodă este înlocuită cu cicluri consecutive de simulare și rafinare a sistemului până când acesta îndeplinește condițiile cerute.

Pentru sistemele distribuite artificiale au fost determinați o serie de algoritmi creați pe baza studiului intens al unor fenomene naturale emergente (algoritmi de gossip – bârfă [Adjie 1999] folosiți cu succes în sistemele distribuite au fost preluați din studiul coloniilor de furnici).

Totodată acest tip de modelare a sistemelor SPRAY este util pentru determinarea poziționării în structuri nesimetrice. Poziționarea în structuri

nesimetrice a fost realizată prin studiul poziționării celulelor în decursul procesului de morfogeneză. Astfel, spre deosebire de procesul descris în paragraful anterior, în procesul de morfogeneză apar anumite forțe de atracție / repulsie doar dacă entitățile implicate depășesc un număr dat (entitățile se reorganizează folosind rezultanta forțelor implicate; în funcție de aplicația aleasă, valoarea rezultantei trebuie să depășească un anumit prag pentru a fi luată în considerare; valoarea pragului determină numărul minim de entități). Conform acestui principiu nu se mai alege un centru de greutate în jurul căruia se pot grupa entitățile, se aleg anumite centre de poziționare bazate pe densitatea entităților. În acest mod se pot crea forme nesimetrice.

Tipul de proiectare prezentat are următoarele avantaje:

- Se poate baza pe cercetările din alte domenii pentru a explora o serie de comportamente emergente.
- Odată ce au fost stabilite condițiile pentru determinarea comportamentului emergent, programarea sistemelor de tip SPRAY este simplă, fiind redusă la enunțul unor reguli locale după care este obligat sistemul să funcționeze.

Acest tip de proiectare prezintă și o serie de dezavantaje:

- Principiile non-deterministice care fac să funcționeze comportamentele emergente sunt extrem de greu de controlat.
- Sistemul poate ajunge la o serie de stări finite, globale extrem de complicat de dedus.

Uneori este necesar adoptarea unei forme nesimetrice specifice și nu a unei forme nesimetrice care a apărut ca rezultat al comportamentului emergent creat pentru sistemul respectiv.



Fig. 12. Model nesimetric de poziționare a particulelor



### 3.3.1.3 Controlul comportamentului emergent

Pentru proiectarea unui sistem SPRAY cu adevărat eficient, robust și fiabil este necesară găsirea unui model de proiectare care să îmbine ambele modele prezentate în paragrafele anterioare. Astfel se poate aplica proiectarea de tip invers până la un punct de unde se folosește proiectarea directă. Introducerea unui control rezervat proiectării directe trebuie să fie făcută într-o măsură corespunzătoare astfel încât să nu interfereze cu condițiile prestabilite pentru declanșarea comportamentului emergent.

Pentru diferitele aplicații care se pot crea pe baza comportamentelor emergente este imposibil de a se proiecta un model general, o rețetă universală care să permită cunoașterea a-priori a condițiilor de declanșare a comportamentului emergent pentru o aplicație de uz general.

Principiile de proiectare prezentate în paragrafele anterioare pot fi aplicate cu succes la număr mare de aplicații. Aceste aplicații trebuie să fie însă extrem de specifice pentru a se putea identifica cu exactitate condițiile necesare declanșării comportamentului emergent. De asemenea, pentru a simplifica proiectarea unui astfel de sistem este necesară identificarea domeniului din care se poate prelua descrierea comportamentului emergent, posibilitățile de adaptare la aplicația curentă.

## 3.4 Tehnologiile “roi” ale NASA

Viitoarele misiuni spațiale ale NASA vor implica noi tehnologii de explorare care se vor baza pe aplicarea tiparelor de comportament emergent specifice de tip “roi” [Truszkowski 2004, Truszkowski 2006, Hinchey 2005]. Misiunile tradiționale bazate pe nave spațiale de mari dimensiuni vor fi completate cu concepte care implică nave de dimensiuni reduse care operează și colaborează în moduri specifice roiurilor de insecte din natură.

Este extrem de important din punctul de vedere al NASA ca misiunile spațiale viitoare să fie pe deplin autonome, să dețină un grad mărit de inteligență. Aplicarea conceptelor de comportament emergent în aceste misiuni permite lansarea de echipamente de foarte mici dimensiuni, specifice explorării spațiului (spre deosebire de marile nave spațiale a căror folosire pentru explorare se poate dovedi nepractică), de cost redus, cu puțină inteligență încorporată dar care pot realiza sarcina cerută cu o mai mare eficiență. Aplicarea conceptelor de comportament emergent este necesară în acest caz pentru mărirea gradului de autonomie al echipamentelor în condițiile în care timpul de călătorie al semnalului radio până la Pământ este de peste 30 minute. Odată cu mărirea gradului de autonomie al echipamentelor de explorare, luarea deciziilor se poate face mai curând *in situ* decât la sol.

### 3.4.1 Misiunile ANTS

“Autonomous Nano Technology Swarm” (ANTS) reprezintă o colaborare dintre Goddard Space Flight Center și Langley Research Center destinată dezvoltării unei arhitecturi emergente de misiuni spațiale.

### 3.4.2 SARA : The Saturn Autonomous Ring Array

Misiunea SARA va lansa 1000 de nave spațiale de clasă *pico* organizate sub forma a 10 roiuri, fiecare conținând instrumente specializate care vor executa *in situ* explorarea inelelor lui Saturn pentru a determina compoziția acestora precum și modul lor de formare. Această misiune va necesita structuri auto-configurabile pentru propulsie nucleară și control, pentru navigarea prin inelele lui Saturn și detectarea / evitarea coliziunilor.

### 3.4.3 PAM: Prospective Asteroid Mission

Misiunea PAM va lansa un roi de nave spațiale de clasă *pico* (cu o greutate de aproximativ de 1 Kg) care vor explora centura de asteroizi căutând asteroizi cu anumite caracteristici.

În [Truszkowski 2004, Truszkowski 2006] se prezintă o privire generală a misiunii PAM. În această misiune, o navă spațială de transport va fi lansată de pe Pământ spre un punct din spațiu unde forțele gravitaționale care acționează asupra obiectelor spațiale de clasă *pico* sunt neglijabile (acesta este denumit punct Lagrange). Din acest punct se vor lansa cele 1000 de *pico*-nave stelare asamblate în cursul zborului. Se estimează ca peste 70% din aceste nave să fie pierdute în timpul misiunii ca urmare a coliziunii cu asteroizii sau între ele. Navele spațiale vor avea la dispoziție doar vele solare care le vor reduce considerabil mobilitatea, ceea ce implică faptul că folosirea unor algoritmi complicați de detectare / evitare de obstacole este nepractică de vreme ce nava nu va fi în stare ca să le ocolească în timp util.

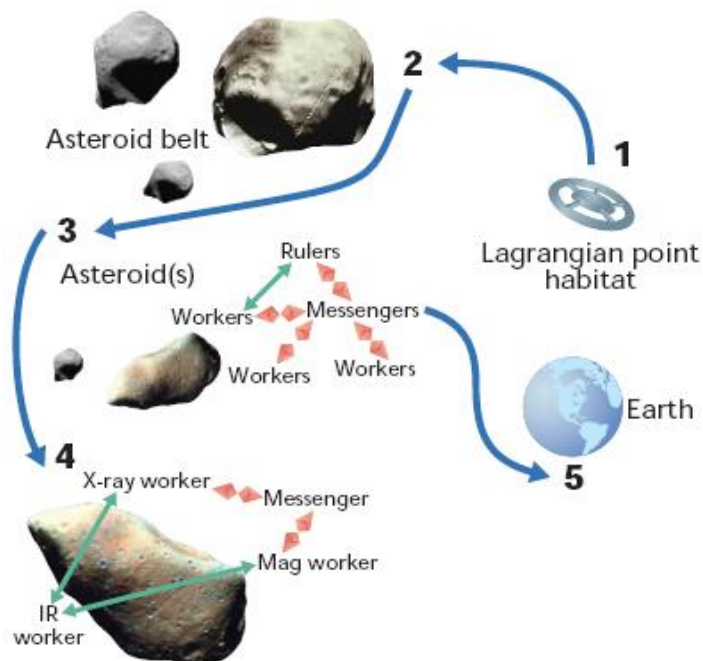


Fig. 13. Misiunea PAM ([Truszkowski 2004, Truszkowski 2006])

Din cauza dimensiunilor reduse, fiecare navă spațială va fi dotată doar cu un singur instrument specializat de măsură (magnetometru, spectrometru etc.). Aproximativ 80% din toate navele vor fi configurate să lucreze ca și simpli colectori de date. O parte din navele rămase vor fi folosite ca și lideri care vor ști reguli de alegere a asteroizilor precum și datele de care misiunea are nevoie. Al treilea tip de nave vor fi mesagerii care vor coordona comunicația dintre lideri și muncitori precum și comunicația cu stațiile terestre.

Cele 1000 de nave se pot împărți apoi în sub-roieri conduse de un lider care va coordona muncitorii care, la rândul lor se vor folosi de instrumentele specializate din dotare pentru a investiga tipurile de asteroizi. Pe baza datelor conținute, liderul va determina tipul de asteroid necesar spre a fi investigat, muncitorii vor încerca să găsească un asteroid de tipul cerut prin investigații aleatoare ale asteroizilor din zona în care au fost lansați.

## **3.5 Rețele de senzori inteligenți**

### **3.5.1 MicaZ / Mica2**

MicaZ este un modul wireless care funcționează pe o frecvență de 2.4 GHz și care este utilizat pentru implementarea rețelelor de senzori inteligenți. Acesta permite dezvoltarea de aplicații personalizate pentru senzori inteligenți și este în mod special optimizat pentru rețele de mică putere care funcționează pe bază de baterii. Sensorii inteligenți bazați pe MicaZ / Mica2 folosesc sistemul de operare open-source TinyOS, care oferă capacități de încredere pentru formarea de rețele ad-hoc, programare over-the-air, instrumente de dezvoltare, servere middleware pentru integrarea în rețea, interfața cu utilizatorul de tip client pentru analiză offline și configurație.

Rețelele de senzori inteligenți implementate cu MicaZ / Mica2 pot fi folosite cu succes în medii colaborative robotizate care utilizează tipare de comportament emergent dar sistemul de operare TinyOS nu implementează structurile de comandă și control specifice utilizării tiparelor de comportament emergent în vederea reducerii consumului de energie și a puterii computaționale necesare. Totodată, implementarea unor tipare de comportament emergent specifice comunicației bazată de schimbul de feromoni conduce la simplificarea protocoalelor de comunicație existente.

### **3.5.2 Cricket**

Cricket este un sistem de localizare pentru interior utilizat în rețele de senzori inteligenți. Cricket folosește o combinație de tehnologii de radio-frecvență îmbinate cu ultrasunete pentru a oferi informații referitoare la localizarea dispozitivelor la care acestea sunt atașate. Balize montate pe pereții și pe tavanul unei clădiri, publică informații pe un canal de radio-frecvență. La fiecare transmisie de radio-frecvență baliza transmite și un impuls concomitent de ultrasunete. Alte dispozitive Cricket atașate la dispozitive fixe sau mobile ascultă semnalele radio, și la primirea primilor biți, ascultă și un puls ultrasonic corespunzător. Atunci când sosește acest puls, dispozitivul obține o distanță estimată pentru baliza corespunzătoare, calculând diferența de viteze de propagare între impulsul de radio-frecvență (viteza luminii) și impulsul de ultrasunete (viteza sunetului).

Sistemul Cricket nu dispune de un control centralizat. Folosirea tiparelor de comportament emergent conduce la complexitatea protocoalelor de comunicație implicate, și, totodată și al consumul de energie; folosirea acestora permite și scalarea eficientă suplimentară a sistemului împreună cu creșterea densității dispozitivelor implicate în rețeaua Cricket.

### 3.5.3 TelosB

Derivat din MicaZ / Mica2, TelosB a fost proiectat conform a trei principii majore pentru a facilita integrarea acestora în rețele de senzori inteligenți: consum de energie redus, ușurința în folosire și robustețe hardware și software. Similar cu MicaZ / Mica2, și TelosB se folosește de serviciile oferite de sistemul de operare TinyOS: programare over-the-air, formarea de rețele ad-hoc și interfața cu utilizatorul de tip client pentru analiză offline și configurația rețelei.

Aplicarea tiparelor de comportament emergent în cadrul rețelelor de senzori inteligenți implementate cu dispozitive TelosB conduce la reducerea suplimentară a consumului, simplificarea protocoalelor de comunicație precum și la creșterea scalabilității sistemului și la descentralizarea acestuia.

## 3.6 Metaeuristici

Aplicații ale tiparelor de comportament emergent se găsesc nu numai în cazuri practice, reale a unor entități fizice. Tiparele de comportament emergent inspirate din studiul coloniilor de furnici se folosesc în mod frecvent la rezolvarea unor probleme complexe a căror soluție este una euristică.

Fenomenele emergente rezultă din interacțiunile simple care apar între entitățile componente ale unui sistem. Există, în consecință, o serie de probleme pentru care găsirea unei soluții optime este dificilă și uneori nepractică datorită lipsei unui aparat matematic. Pentru rezolvarea acestor probleme se preferă identificarea unor tehnici euristice pentru generarea unor soluții corespunzătoare dar care nu sunt întotdeauna și optime [Dorigo 1997]. Printre metaeuristicile existente în momentul actual se numără optimizarea bazată pe studiul coloniilor de furnici (Ant Colony Optimization [Maniezzo 2004, Gambardella 2000, Montemanni 2005a, Montemanni 2005b]), precum și convergența globală (Global Convergence [Parunak 2005]). Deși aceste metaeuristici se bazează pe un suport matematic semnificativ, nu există o formalizare pentru descrierea tiparelor de comportament emergent folosite, ci doar reprezentări parțiale urmărind în special efectul global pe care acestea îl produc. De asemenea, comportamentul local, predictibil, poate fi descris în termeni formali, însă interacțiunile dintre aceste comportamente locale care generează în final emergența nu pot fi cuantificate din cauza numărului mare al acestora.

Există, în mod normal, o serie de probleme de complexitate combinatorică pentru soluționarea cărora găsirea unei soluții optime este dificilă și uneori nepractică datorită spațiului prea mare de soluții sau datorită unor cerințe computaționale prea mari generate de constrângeri existente. Pentru rezolvarea acestor probleme se preferă identificarea unor tehnici euristice pentru generarea unor soluții corespunzătoare dar care nu sunt întotdeauna și optime. Acești algoritmi se folosesc de cele mai multe ori de cunoștințe apriori asupra aplicației ceea ce restrânge extrem de mult aria lor de aplicabilitate [Maniezzo 2004, Gambardella 2000].

Metaeuristicile reprezintă un set mai general de tehnici și algoritmi euristici, ca și cei descriși mai sus, dar care pot fi aplicați la probleme diverse cu mici modificări. Exemple de metaeuristici sunt căutarea tabu (eng. – “tabu search”) sau calculul evolutiv (eng. – “evolutionary computation”).

Metaeuristicile sunt reprezentate de algoritmi care sunt conduși de o euristică simplă, de bază, pentru a evada din capcana optimelor locale și o parte metaeuristică. Aceștia fie se folosesc de o euristică constructivă care pornește de la o soluție goală la care se adaugă mai apoi elemente necesare pentru a ajunge la o soluție acceptabilă, fie o euristică bazată pe căutări locale care pornesc de la o soluție completă căreia îi modifică elementele în mod iterativ pentru a ajunge la o soluție mai bună. Partea metaeuristică permite ca folosirea împreună cu euristica de bază să se obțină rezultate mai bune decât în cazul absenței acesteia, chiar și prin iterarea euristicii de bază. Mecanismul de îmbunătățire se folosește fie de introducerea constrângerilor, fie de alegerea în mod aleator a soluțiilor luate în vedere pentru căutarea locală, fie de combinarea elementelor extrase din soluții precedente diferite.

### 3.6.1 A.C.O. – Ant Colony Optimization

În mod uzual, metaeuristicile sunt inspirate din procese naturale precum procesul de evoluție sau administrarea de către creier a structurilor de memorie. Una dintre cele mai recente metaeuristici inspirate din mediul natural este optimizarea bazată pe studiul coloniilor de furnici (eng. – “Ant Colony Optimization”, ACO) [Maniezzo 2004, Gambardella 2000, Montemanni 2005a, Montemanni 2005b]. Una din trăsăturile principale ale acestei metaeuristici este combinarea pentru identificarea unei soluții semi-optimale a unor informații apriori despre structura posibilă a unei soluții cu informații suplimentare despre structura unei soluții precedente acceptate.

Un algoritm ACO funcționează pe baza următoarelor principii [Gambardella 2000]:

- Un set de agenți asincroni și concurenți, a căror reprezentare formală se bazează pe descrierea unei colonii de furnici, se mișcă prin stările problemei care corespund soluțiilor parțiale ale acesteia. Agenții se mișcă folosind un set de decizii locale, stohastice care sunt luate pe baza valorilor a doi parametri: căi de feromoni și atractivitate.
- “furnica” construiește incremental soluția problemei, alegând următoare mișcare pe baza tiparului de comportament emergent folosit.
- La identificarea unei soluții acceptabile și complete, “furnica” evaluează soluția găsită și modifică calea de feromoni existentă care va fi folosită în căutare de viitoare “furnici”.

Algoritmii ACO prezintă două mecanisme de control importante [Gambardella 2000]:

- Evaporarea căii de feromoni decrementează toate valorile căii de feromoni pentru a evita acumularea nelimitată de informație în cazul unor componente ale soluției.
- Acțiuni centralizate folosite pentru a înnoi informații globale ale sistemului.

O “furnică” este un agent computațional care construiește iterativ o soluție pentru o problemă dată. Soluțiile parțiale sunt reprezentate ca stări intermediare. Nucleul fiecărui algoritm ACO este reprezentat de o buclă în care, la fiecare iterație, fiecare “furnică” se mișcă (execută un pas al algoritmului) de la o stare  $\delta$  la o stare  $\theta$  care corespunde unei soluții parțiale mai complete. Astfel, la fiecare pas  $\sigma$ , o

“furnică”  $k$  calculează  $A_k^\sigma(\theta)$  posibile stări viitoare și alege una dintre ele cu o anumită probabilitate. Pentru o “furnică”  $k$ , probabilitatea  $p_{\delta\theta}^k$  de a trece de la starea  $\delta$  la starea  $\theta$  depinde de combinația a două valori:

- Atractivitatea  $\eta_{\delta\theta}$  care este calculată pe baza unei euristici din informații apriori reprezentând probabilitatea generală de a se muta în noua stare;
- Nivelul  $\tau_{\delta\theta}$  al căii de feromoni care indică cât de eficientă a fost o anumită mișcare în trecut, fiind o indicație curentă a probabilității de mișcare în noua stare;

Căile de feromoni sunt reînnoite doar atunci când toate furnicile au determinat o soluție completă prin incrementarea sau decrementarea nivelului căii care corespunde mișcărilor care au generat soluții corespunzătoare, respectiv necorespunzătoare.

### 3.7 Mișcarea în colective robotice

Tiparele de comportament emergent au fost utilizate în colectivele robotice în special pentru modelarea mișcării unui grup omogen de entități aplicând principii specifice coloniilor de furnici, bancurilor de pești etc. pentru rezolvarea problemelor specifice: ocolire de obstacole, colectarea de resurse.

În continuare se prezintă unul dintre cele mai reprezentative modele de mișcare preluat din studiul bancurilor de pești precum și alte lucrări reprezentative din acest domeniu.

#### 3.7.1 Modelul lui Aoki

Autorul lui [Aoki 1982] a fost unul dintre primii care a prezentat un model matematic pentru mișcarea bancurilor de pești precum și o simulare pe calculator pentru modelul lui. Scopul lucrării sale a fost să determine modul cum interacționează peștii între ei în vederea declanșării comportamentului de banc a întregului grup.

Deoarece lucrările lui Aoki au fost punctul de plecare multe alte lucrări în domeniul comportamentului emergent inspirat din natură, l-am ales ca bază de comparare pentru simularea modelului meu. Ca atare, o scurtă descriere a modelului lui Aoki este prezentată în continuare.

Sunt făcute următoarele ipoteze primare:

- Timpul este cuantizat în pași, deciziile sunt luate înaintea fiecărui pas
- Organismele simulate se mișcă într-un spațiu bidimensional și nu sunt influențate de mediu (nu există obstacole)
- Cele două componente ale mișcării, viteza și direcția, sunt independente și, mai mult, pentru a simplifica problema, interacțiunile dintre indivizi sunt reduse la direcție; aceste două componente sunt determinate în urma unui set de reguli exacte (descrise mai jos) pentru fiecare individ.

Viteza fiecărui individ este determinată folosind o distribuție gama cu doi parametri care sunt constanți în timpul simulării și care sunt determinate prin observație.

Direcția de deplasare este determinată utilizând o distribuție normală. Variația densității de probabilitate este constantă în timp ce media este calculată

înaintea fiecărui pas și este definită în relație cu poziția și direcția vecinilor dintr-o arie definită de o sferă  $RC$  (Fig. 14).

Primul pas în determinarea este alegerea unui individ ca referință dintre cel mult 4 indivizi în cadrul sferei  $RC$ . Acești indivizi sunt cei cu cea mai mică magnitudine a unghiului  $RDR$ , unghiul dintre direcția membrului curent și poziția vecinului luat în considerare, cum se arată în [Huth 1992]. Probabilitatea de alegere al  $j$ -lea individ ( $j \leq 4$ ) este invers proporțional cu unghiul sau  $RDR$ . Cu alte cuvinte cu cât este mai aproape un individ  $j$  de direcția membrului  $i$  al grupului, cu atât este mai mare probabilitatea ca  $j$  să fie ales ca referință când se determină media pentru  $i$ .

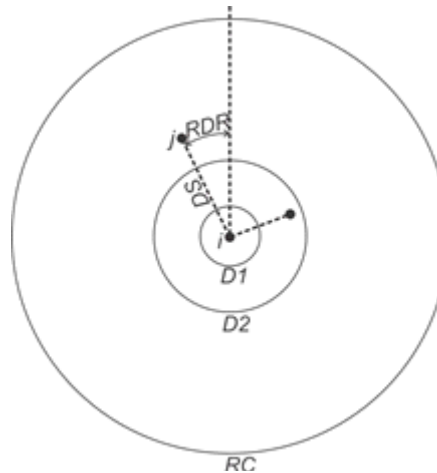


Fig. 14. Parametrii de tip senzorial pentru modelul lui Aoki

Considerând că a fost selectată referința  $j$ ,  $DS$  este distanța dintre  $i$  și  $j$ ,  $DR_j$  direcția către  $j$ ,  $D1$  distanța de evitare și  $D2$  distanța de apropiere, așa cum se arată în [Aoki 1987]., media este calculată potrivit uneia din următoarele cazuri:

- $DS > D2$ , individul de referință este considerat prea departe, media este definită ca  $DR_j$  direcția înspre el, pentru a permite apropierea;
- $D1 \leq DS \leq D2$ , individul de referință este considerat la distanța potrivită, media este definită ca direcția lui, pentru a permite orientarea paralelă;
- $DS < D1$ , individul de referință este considerat prea apropiat, există pericolul unei coliziuni, media este definită ca  $DR_j + 90$  sau  $DR_j - 90$ , depinzând care dintre ele formează un unghi mai mic cu direcția curentă a individului  $i$ ;
- Nici un vecin nu a fost găsit în interiorul sferei  $RC$ , nici un individ de referință nu poate fi găsit, media este determinată cu o probabilitate uniformă între 0 și 360 de grade.

Așa cum s-a menționat mai sus, acest model are meritul să fie unul dintre primele care au simulat cu succes comportamentul de banc al peștilor, dar are de asemenea multe neajunsuri. Faptul că mișcările componentelor sunt determinate de distribuție și probabilitate înseamnă că acțiunile fiecărui individ sunt imprevizibile (aceasta poate fi o bună simulare a comportamentului peștilor, dar nu este dorită într-un colectiv robotizat). Aceasta necesită de asemenea generarea de numere aleatoare la fiecare pas putând fi un calcul intensiv în special luând în considerare constrângerile de resurse a colectivelor robotizate.

### 3.7.2 Alte modele

În [Inada 2004] autorii prezintă un model similar modelului lui Aoki: are aceleași trei comportamente (apropierea, evitarea și orientarea paralelă), interacțiunile cu mediul sunt ignorate și viteza este de asemenea o variabilă stohastică neinfluențată de nici o interacțiune. Principala diferență este în alegerea direcției: Huth & Wissel au propus o medie între influențele mai multor vecini și aplicând o distribuție normală în jurul acestui unghi mediu pentru a alege direcția de mișcare. Ei au definit de asemenea două metrici, NND (distanța de vecinul cel mai apropiat) și polarizarea, folosită pentru evaluarea preciziei simulărilor lor.

Autorii din [Wu 2001] prezintă un model de auto-organizare a formării grupului în spațiul tridimensional, și îl folosesc pentru a investiga spațiul dinamic a grupurilor de animale precum bancurile de pești și stolurile de pasări. Folosește aceleași trei comportamente ca cele descrise anterior: apropierea, evitarea și orientarea paralelă, dar dă evitării cea mai mare prioritate, ignorând alte influențe dacă cel puțin un vecin este în interiorul "zonei de repulsie". În acest caz, membrul se îndepărtează de centrul masei tuturor vecinilor din interiorul zonei de repulsie. Dacă nu este nici un vecin în zona de repulsie, celălalt comportament se manifestă: se calculează media pozițiilor tuturor vecinilor din "zona de atracție". În final se setează direcția membrului ca media între aceste două valori.

Autorii susțin de asemenea că lucrarea lor este prima dovada a memoriei colective în aceste tipuri de grupuri de animale, în timpul tranziției unui grup de la un tip de comportament colectiv la altul.

Autorii din [Hwang 2009] încearcă să ofere un cadru comun pentru caracterizarea obiectivă și cantitativă a relațiilor dintre comportamentele la nivel individual și modelele la nivel de grup în modelele de bancuri. Ei studiază pe scurt similaritățile și diferențele algoritmilor de comportament și statisticile agregate printre alte metode de predare existente și propun un schelet de modelare care sintetizează mult din această muncă anterioară.

Ei modelează deciziile de mișcare ale fiecărui pește ca și o sumă de forțe locomotorii, sociale și arbitrare. Forța locomotoare constă din frânare, numită coeficient de frânare. Forța socială are trei aspecte: funcția de atracție / respingere, regula de cântărire a vecinului, și normele privind dimensiunea. Atracția / respingerea a fost modelată ca lineară, porțiuni lineare, sau convexă. Regula de cântărire a vecinului ponderează forța de atracție / respingere prin distanța față de fiecare vecin influent. Ponderarea a fost constantă ( de ex. pondere egală), lineară negativ, sau sigmoid negativ. Forța aleatoare a fost aleasă pentru fiecare pas din distribuția normală și echivalentul deviației standard de 1/3, 1/6 și 1/8 din forța socială maximă pe care un pește individual o poate experimenta de la alt pește. Cantitatea de forță aleatoare al fiecărui individ a fost o funcție a distribuției vecinilor din apropiere. Mișcarea a fost calculată ca accelerația rezultată din suma tuturor acestor forțe.

În [Codling 2007] autorii studiază efectul pe care ei îl numesc "principiul multor greșeli" în mișcarea animalelor grupate: numărul mare de erori de navigare ale indivizilor de-a lungul grupului sunt suprimate de interacțiuni și coeziunea grupului. Ei simulează principiul multor greșeli utilizând un model simplu bazat pe indivizi, al mișcării bazat pe o mișcare aleatoare părtinitoare care include interacțiunile de grup. Ei măsoară și analizează abilitatea grupului ca un întreg pentru a atinge ținta, date fiind diferitele nivele de erori de navigare ale indivizilor, dimensiunea grupului, raza de interacționare, și turbulența mediului.



Modelul propus de ei i-a în considerare faptul că fiecare individ are propria direcție preferată, direcția în care se va mișca dacă nu ar fi prezenți alți vecini. Aceasta direcție preferată este apoi influențată de interacțiunile grupului (atracția, respingerea, orientarea paralelă) și de o variabilă aleatoare reprezentând eroarea dată de faptul că fiecare individ încearcă să se orienteze către direcția preferată. Scenariile de simulare realizate au arătat faptul că atunci când sunt prezente nivele scăzute de turbulențe ale mediului, există un beneficiu navigațional din grupul de membrii, particular pentru grupuri de dimensiuni mici. Cu toate acestea, când are loc o mișcare într-un mediu cu turbulențe mari, rezultatele simulării sugerează faptul că cea mai bună strategie este să navigheze ca indivizi și nu ca un grup.

Autorii [Koenig 1998] au început lucrul prin studierea mișcărilor de învățare ale două specii de pești, concentrat pe două tipuri de mișcări: mișcarea de apropiere, și mișcarea de orientare paralelă, pe care ei o consideră esențială pentru formarea învățării. Corelarea parametrilor de mișcare măsurați (poziția, direcția corpului, și direcția de mișcare) printre indivizi a fost utilizată pentru a determina care mișcare o realizează fiecare individ. Similaritățile evidente și diferențele dintre două mișcări au fost găsite în numărul de vecini cu care interacționează și în poziția relativă și întârzierea de timp dintre indivizii care interacționează.

Bazat pe aceste studii, ei au dezvoltat un model analitic pentru a simula mișcarea de învățare. Mișcarea de învățare simulată reproduce bine mișcarea unui banc de pești naturali și a fost astfel utilizată să clarifice relația dintre interacțiunea locală dintre indivizi și proprietățile globale ale mișcării de învățare.

### 3.8 Concluzii

În acest capitol s-a prezentat stadiul actual al dezvoltării mediilor colaborative cu inteligență emergentă și distribuită. S-au prezentat atât sisteme de comunicație care folosesc mesaje semi-persistente similare feromonilor furnicilor cât și sisteme robotice care utilizează comportamentul emergent pentru poziționare sau mișcare.

Aceste medii colaborative prezintă un grad ridicat de emergență determinată de o modalitate de control puternic descentralizat și de un număr ridicat de entități componente. Tiparele de comportamente emergente au o probabilitate de apariție mai mare în cazul sistemelor descentralizate. Un astfel de sistem trebuie să atingă anumite praguri de organizare, diversitate și conectivitate pentru a putea afișa comportamente tipic emergente. Sistemele care prezintă comportamente emergente pot părea că încalcă legi ale entropiei precum și legea a doua a termodinamicii deoarece se formează și se ordonează în absența comenzilor directe și a unui control centralizat. Acest lucru este posibil deoarece sistemele deschise pot extrage informații direct din mediu.

Mediile colaborative robotizate care au fost proiectate prin aplicarea tiparelor de comportament emergent preluate din studiul naturii (colonii de furnici, bancuri de pești, roiuri de albine etc.) prezintă unele caracteristici importante:

- flexibilitate (sistemul se poate adapta singur la orice schimbare de mediu);
- robustețe (scopul întregului sistem este finalizat chiar dacă un număr de entități componente sunt nefuncționale);
- auto-organizare (toate activitățile grupului nu sunt nici controlate, nici supravegheate de nici o entitate individuală).

Metoda de căutare a hranei în cazul furnicilor a condus la o nouă tehnică de rutare a traficului în rețelele din cadrul sistemelor de telecomunicații. Interacțiunea

colaborativă dintre furnici pentru a transporta o cantitate mare de hrană, conduce la algoritmi mai eficienți pentru roboți în aplicații de colectare de resurse, explorare și ocolire de obstacole. Modul de divizare al muncii în cazul unei colonii de albine poate fi de folos pentru stabilirea fluxului liniilor de asamblare din fabrici. Teoria roiurilor are aplicații și în management.

Mișcarea inspirată din studiul bancurilor de pești este aplicată cu succes în aplicații de mobilitate pentru colectivități mari de entități robotice. Algoritmii relativi simpli aplicați la ocolirea de obstacole conduc la eficientizarea colectivelor robotice prin scăderea necesarului de putere computațională și a dimensiunilor fizice ale entităților componente.

Inteligența roiurilor precum și tiparele de comportament emergent preluate de la bancurile de pești sunt folositoare într-o gamă largă de domenii:

- robotică
- nanotehnologie
- biologie moleculară
- medicină
- controlul traficului
- tactici militare
- artă interactivă
- proiectare hardware și software
- management și business

În capitolul următor se prezintă o serie de metrice ale mediilor colaborative robotizate și ale rețelelor de senzori prezentate în acest capitol precum și problemele întâlnite.

## 4. PROBLEME SPECIFICE MEDIILOR COLABORATIVE ROBOTIZATE

O tendință de cercetare relativ recentă și de mare interes în prezent este căutarea soluțiilor la problemele interacțiunii și coordonării colectivelor de entități inteligente prin studiul și utilizarea tiparelor de comportament emergent.

În capitolul de față se prezintă problemele sistemelor distribuite și colaborative prezentate în capitolul anterior. Capitolul descrie caracteristicile entităților mediilor colaborative robotizate precum și problemele asociate fiecărei astfel de caracteristici, problemă care se dorește a fi rezolvată prin aplicarea tiparelor de comportament emergent. Similar, se mai prezintă caracteristicile și problemele rețelelor de senzori fără fir. În continuare, se prezintă problemele specifice comunicațiilor în sistemele distribuite clasice și o comparație cu sistemele emergente care folosesc mesaje bazate pe feromoni.

Problemele prezentate în capitolul de față pot fi rezolvate prin aplicarea tiparelor de comportament emergent. În capitolele următoare se prezintă evaluarea performanțelor aplicării acestor tipare de comportament emergent în contextul rezolvării problemelor descrise.

### 4.1 Introducere

În prezent exista mai multe abordări ale problemei coordonării colectivelor robotice. Abordarea tradițională este cea a coordonării centralizate care are avantajul unui control foarte precis al activității fiecărui robot cât și a unei planificări precise a operațiilor efectuate. De asemenea pentru o mai bună coordonare, un sistem robotizat poate fi combinat cu o rețea de senzori [Ciubotaru 2007, Cioarga 2005]. Dezavantajul principal îl reprezintă scalabilitatea redusă și necesitatea comunicării unei cantități foarte mari de date. La polul opus se află soluțiile distribuite, în general de tip comportamental, care asigură o scalabilitate bună a sistemului dar nu asigură o strategie bine determinată de operare și nici posibilitatea de învățare automată a sistemului.

Tiparele de comportament emergent apar la interacțiunea dintre entitățile unui sistem care funcționează pe baza unor algoritmi simpli. Există, în consecință, o serie de probleme de complexitate combinatorică pentru care găsirea unei soluții optime este dificilă și uneori nepractică datorită spațiului prea mare de căutare sau datorită unor cerințe computaționale prea mari generate de constrângeri existente. Pentru rezolvarea acestor probleme se preferă identificarea unor tehnici euristice pentru generarea unor soluții corespunzătoare dar care nu sunt întotdeauna și optime [Maniezzo 2004, Gambardella 2000, Montemanni 2005a, Montemanni 2005b, Parunak 2005]. Printre metaeuristicile existente în momentul actual se numără optimizarea bazată pe studiul coloniilor de furnici precum și convergența globală.

Deși aceste metaeuristici se bazează pe un suport matematic semnificativ, nu există o formalizare pentru descrierea tiparelor de comportament emergent folosite, ci doar reprezentări parțiale urmărind în special efectul global pe care acestea îl produc. De asemenea, comportamentul local, predictibil, poate fi descris în termeni formali, însă interacțiunile dintre aceste comportamente locale care

generează în final urgența nu pot fi cuantificate din cauza numărului mare al acestora.

Mediile colaborative robotizate prezintă o serie de caracteristici precum: energia consumată, robustețea, securitate etc. O parte din aceste caracteristici sunt comune și rețelelor de senzori inteligenți.

Deși, din punct de vedere al sistemelor distribuite clasice, majoritatea problemelor apărute la medii robotizate sunt deja rezolvate, din cauza caracteristicilor prezentate în continuare, vechile paradigme nu mai pot fi aplicate cu succes în cazul mediilor colaborative robotizate.

Caracteristicile identificate în domeniul mediilor colaborative robotizate, al senzorilor inteligenți și al rețelelor de senzori inteligenți generează principalele probleme ale sistemelor descrise mai sus. Pentru a putea funcționa în condiții optime, mediile robotizate sunt proiectate cu constrângeri extreme asupra metricilor menționate, ceea ce conduce la probleme importante în acest domeniu.

## **4.2 Caracteristici ale entităților unui mediu colaborativ robotizat**

În continuare se prezintă o serie de caracteristici importante precum și problemele specifice unui senzor inteligent ca nod într-o rețea de senzori fără fir, care de fapt, stă la baza mediilor colaborative robotizate.

### **4.2.1 Energia consumată**

Energia consumată este extrem de importantă în cazul unui nod individual al unei rețele de senzori. Pentru a avea o durată de viață considerabilă, de câteva luni până la nivelul anilor, nodurile trebuie să îndeplinească condițiile de funcționare ale unor dispozitive cu consum extrem de mic de energie (ultra low power). Astfel, dispozitivele hardware care constituie un nod de rețea, precum și rutinele software trebuie proiectate astfel încât să impună cerințele de funcționare cu consum redus întregului sistem.

Echipamentele radio de emisie – recepție care fac parte din consumatoarele importante de energie, trebuie păstrate în modul de lucru cu consum redus un timp cât mai mare. Astfel, este necesar ca algoritmi aplicației care rulează pe nodul de rețea să facă o planificare corespunzătoare a cerințelor de comunicație ale sistemului, pentru a eficientiza consumul. În mod curent, un nod de rețea de senzori realizat în tehnologia actuală consumă 200  $\mu$ A ceea ce îi permite să funcționeze un an folosind o pereche de baterii de uzual de 1.5V. În rețelele fără fir comunicația este definită prin anumiți parametri ca: raza de transmisie, rata de transmisie și consumul de energie. Parametrul raza globală de acoperire a rețelei de senzori nu este conectat cu raza de transmisie a unui nod, deoarece datele sunt direcționate de la nod la nod, prin rețea, ceea ce presupune că o acoperire mai mare a rețelei înseamnă un număr mai mare de noduri de rețea.

Parametrul simplu de rază de acoperire a unui nod se referă la granularitatea minimă a rețelei. În cazul în care aplicația necesită un număr mic de noduri active care să acopere o suprafață mare este necesară introducerea de noi noduri care să direcționeze informațiile între nodurile active.

Comunicația este strâns legată de consumul de energie, modulele de comunicație generând procentul majoritar din energia consumată la nivelul întregului sistem.

### 4.2.2 Robustețe

Robustețea unui mediu colaborativ robotizat se definește ca gradul de distrugere al componentelor sale interne care poate fi suportat de acesta pentru ca sistemul să își păstreze funcționalitatea deplină.

Pentru a atinge durata de viață cerută este necesar ca nodul de rețea să fie extrem de robust. O modalitate de atingere a gradului de robustețe cerut de aplicație este folosirea încă din stadiul de proiectare a rețelei a conceptelor de modularitate. Un astfel de sistem modular este mult mai redus decât un sistem nemodular prin faptul că modulele pot fi proiectate, create, testate independent.

Modularitatea este importantă și în contextul procesului de mentenanță, fiecare modul putând fi testat și, după necesitate, înlocuit în mod independent.

Modulele unui sistem trebuie să fie puternic independente iar interfețele dintre ele trebuie să fie înguste pentru a împiedica orice interacțiuni nedorite.

Se dorește implementarea unei interfețe robuste între nodul de rețea și exterior deoarece mediul în care este instalată rețeaua fără fir poate fi folosit în comun cu alte rețele care folosesc același spectru de frecvențe și se dorește o interferență minimă între acestea. În prezența interferențelor se pot crea situații în care comunicația se desfășoară extrem de încet sau chiar apar erori de comunicație. Pentru rezolvarea acestei probleme se folosesc canale multiple de comunicație și spectre largi de frecvențe radio. Este foarte important pentru fluidizarea comunicațiilor evitarea congestionării frecvențelor de comunicație [Minami 2004, Reilly 1997].

În cazul rețelelor de senzori, caracteristica de robustețe prezintă probleme importante când se dorește desfășurarea unei rețele de senzori într-un mediu extern. Cerințele de robustețe conduc în mod invariabil la creșteri de costuri, la creșterea mărimii sensorului.

### 4.2.3 Securitate

Este foarte importantă păstrarea intimității mesajelor transmise în rețeaua de senzori și prevenirea oricărei intruziuni din exterior folosind concepte de criptografie și autentificare criptografică [Deng 203].

Folosirea acestor algoritmi criptografici nu este obligatorie pentru fiecare aplicație. Nivelul acesta ridicat de securitate este recomandat pentru anumite tipuri de aplicație. În cazul folosirii acestor algoritmi criptografici, mai ales în cazul folosirii metodelor de securizare cu cheie publică, este necesar păstrarea sigură a cheii private la nivelul nodurilor individuale.

Folosirea unor algoritmi avansați la nivelul nodului din rețeaua de senzori conduce la cerințe computaționale importante care, la rândul lor, conduc la mărirea consumului de energie.

### 4.2.4 Putere computațională

Majoritatea puterii computaționale a unui nod de rețea este dedicată în mod exclusiv comunicațiilor și procesării datelor din rețea. Protocoalele de rețea fără fir de nivel jos necesită procesare în timp real din partea unității de procesare. Totodată, această unitate de procesare trebuie să eșantioneze datele care sosesc de la unitatea de senzori, să combine aceste date cu datele care sosesc de la celelalte

noduri și să trimită întreg pachetul de date prin rețea către centrul de comunicație [Chrysan. 2004].

Toate operațiile descrise mai sus trebuie să fie executate de o singură unitate centrală de procesare (posibil mai multe dacă arhitectura aplicației o permite). Doar pentru partea de procesare în timp real a comunicației este nevoie de un nivel de procesare de 2 până la 4 MIPS.

Mărirea puterii de procesare a unui senzor prin folosirea unor procesoare mai performante poate conduce la îmbunătățiri deoarece calculele se desfășoară mai rapid, însă acele calcule intensive pot consuma mai mult în acel timp scurt decât ar consuma un procesor mai lent.

#### 4.2.5 Sincronizarea

Sincronizarea de înaltă precizie este extrem de importantă în rețelele de senzori fără fir. Datele colectate de la senzori trebuie să fie corelate în timp.

Sincronizarea unui mediu colaborativ robotizat se poate măsura ca diferența de timp care apare între seturi consecutive de date transmise. Aceasta este direct afectată de distanța dintre transmițător și receptor, dar și de ordinea în care ajung seturile de date.

Sincronizarea în timp este importantă și pentru mărirea eficienței transmițătorului radio. Acesta trebuie să lucreze în moduri de lucru cu consum redus un timp cât mai îndelungat și scoase din acest mod doar atunci când un set de date este gata pentru a fi transmis. Nodurile de rețea trebuie să fie scoase din modurile de lucru cu consum redus în același timp pentru a efectua transmisiile dorite de date și apoi introduse înapoi în modurile de lucru cu consum redus până la următorul transfer.

Orice problemă de funcționare a modulelor de sincronizare conduce de utilizarea ineficientă a întregului sistem sau chiar la incapacitarea unor elemente ale rețelei.

Sincronizarea în timp reprezintă un serviciu foarte important care trebuie oferit în cadrul oricărui mediu distribuit. Există o mulțime de modalități pentru a realiza acest obiectiv. NTP (Network Time Protocol) propus în [Mills 1991, Mills 1994] este folosit de decade în domeniul rețelelor de calculatoare. Este posibil ca nodurile unei rețele să fie dotate cu receptoare GPS cu ajutorul cărora s-ar putea realiza o sincronizare globală [Hofmann 1997]. Folosirea unor astfel de modalități tradiționale de sincronizare în domeniul rețelelor de senzori inteligenți este foarte complexă, în special datorită cerințelor inerente din domeniu. Costul este principalul element prohibitiv din domeniul rețelelor de senzori. Din această cauză un senzor inteligent are în mod uzual putere computațională redusă, o durată de viață care depinde în mod direct de calitatea acumulatorilor cu care este dotat precum și resurse de memorie scăzute. Multe din protocoalele de sincronizare sunt dedicate rețelelor cablate de calculatoare și vehiculează un număr foarte mare de mesaje, sincronizarea realizându-se prin calcule probabilistice, complexe. Un alt element prohibitiv din domeniul rețelelor de senzori inteligenți este rapiditatea. Fiind dispuse în medii variate, viteza de răspuns a rețelelor de senzori este extrem de importantă. Algoritmii uzuali de sincronizare pot permite întârzieri de ordinul milisecundelor care în domeniul rețelelor cablate distribuite sunt acceptabile. În cadrul rețelelor de senzori inteligenți este necesar ca sincronizarea să nu permită întârzieri mai mari de ordinul microsecundelor.

În [Elsan 2001] se propune o sincronizare *post facto* care are loc după primirea unui mesaj special de sincronizare de la un nod, de asemenea special, cu

rol de conducător al procesului de sincronizare. Algoritmul se bazează pe detecția diferenței dintre marca de timp a mesajului primit de la sincronizator și valoarea propriului ceas intern, urmată de ajustarea corespunzătoare a celui din urmă.

În [Elson 2002] se propune un algoritm asemănător pentru sincronizarea în sisteme distribuite denumit RBS (Reference Broadcast Synchronization). O caracteristică interesantă a celor doi algoritmi este că sincronizarea are loc doar la nivelul receptorilor. Acest lucru presupune ca nodul sincronizator să fie un nod special, cu o abilitate mărită de a păstra constant devierea ceasului intern (dispozitive mult mai complexe și mai scumpe care nu fac obiectul folosirii lor într-o rețea de senzori inteligenți). Individualitatea nodurilor de sincronizare scade gradul de omogenitate a rețelei de senzori.

Lucrarea [Romer 2001] propune un sistem de sincronizare pentru rețelele ad-hoc de rețele de senzori care presupune folosirea unor oscilatoare cu cuarț care să prezinte un set de întârzieri maxime cunoscut. Acest protocol se bazează anumite relații temporale între evenimente de tipul "evenimentul X a avut loc înaintea evenimentului Y" sau "evenimentele X și Y au avut loc într-un anumit interval de timp".

Mecanismele TPSN (Time synchronization Protocol for Sensor Networks) propus în [Ganerwal 2003, Simon 2004] precum și FTSP (Flooding Time Synchronization Protocol) propus în [Yoon 2007] încearcă să minimizeze anumite valori variabile ale marcajelor de timp prin folosirea unor marcaje de timp generate la nivelul MAC (Medium Access Control). Deși aceste mecanisme pot prezenta precizii impresionante (2 $\mu$ s) sunt puternic constrânse de accesul la nivelul MAC.

#### 4.2.6 Mărime și cost

Mărimea și costul fiecărui nod al rețelei are un impact asupra costului și posibilităților de instalare a întregii rețele.

Cele mai importante părți ale costului total al folosirii unei rețele de senzori sunt costul de instalare și costul de mentenanță. Este extrem de important ca rețeaua de senzori să colecteze cât mai multe date posibil pentru același cost de mentenanță. În cazul scăderii costurilor nodurilor individuale mai multe noduri pot fi adăugate în rețea crescându-se astfel granularitatea rețelei ceea ce conduce la mărirea cantității de date colecționate.

Mărimea fizică a nodurilor individuale este importantă pentru instalarea rețelelor de senzori. Nodurile de mici dimensiuni pot fi instalate în regiuni mai puțin accesibile ale mediului.

### 4.3 Caracteristici ale rețelelor de senzori fără fir

Rețelele de senzori fără fir prezintă o serie de metrici diferite față de rețele de senzori tradiționale.

#### 4.3.1 Durata de viață

Fiecare nod al rețelei are nevoie de o sursă de energie pentru a funcționa corect. În anumite condiții, nodurile rețelelor de senzori pot fi conectate direct la rețeaua de alimentare locală, dar acest lucru limitează mobilitatea și flexibilitatea rețelei. Pentru a obține o flexibilitate maximă, este necesară folosirea unei surse de energie proprii fiecărui nod de rețea.

Pentru folosirea corespunzătoare a bateriei, nodurile trebuie să prezinte algoritmi de consum extrem de eficienți, astfel prelungindu-se durata de viață a nodului de rețea la nivelul lunilor sau chiar a anilor. Se pot folosi surse de energie naturală precum energia solară. Durata de viață a unei rețele de senzori nu se calculează ca o valoare medie a duratelor de viață ale nodurilor individuale, ci ca valoarea minimă a acestor durate de viață.

Introducerea de moduri de lucru cu consum redus implică o creștere a complexității aplicațiilor care rulează într-o rețea de senzori fără fir.

#### **4.3.2 Acoperirea**

Deoarece în rețelele de senzori inteligenți, transmisia datelor este făcută prin rutarea prin nodurile vecine, noțiunea de acoperire nu se referă la raza de transmisie a unui nod individual. Dacă rețeaua trebuie extinsă pentru a acoperi o suprafață mai mare, este necesar introducerea unor noduri suplimentare.

Comunicația între noduri este făcută pe baza undelor radio, ceea ce consumă o mare cantitate de energie. Pentru a micșora consumul de energie este necesar ca distanța dintre nodurile rețelei să scadă suficient de mult.

Scăderea distanței dintre noduri, în condițiile păstrării aceleiași suprafețe de acoperit, conduce la creșterea numărului de senzori care formează rețeaua și implicit la cantitatea de mesaje implicate în comunicație.

#### **4.3.3 Timp de răspuns**

În anumite aplicații specifice, cum este detectarea intruziunilor într-o locație, este necesar ca timpul de răspuns al alarmei să fie extrem de mic. Pentru ca să se micșoreze consumul, echipamentul radio de emisie – recepție este trecut în moduri de lucru de consum redus. Comunicația radio este esențială în rețelele de senzori fără fir, iar scoaterea din modurile de lucru cu consum redus presupune un interval de timp semnificativ, ceea ce necesită un compromis între durata de viață a nodului individual și timpul de răspuns. Acest compromis poate conduce la scăderea drastică a duratei de viață a nodurilor rețelei și implicit la duratei de viață a rețelei propriuzise.

### **4.4 Probleme ale sistemelor distribuite clasice**

Problemele întâlnite la rețelele de senzori sunt similare cu problemele întâlnite în rețelele de calculatoare, în sistemele distribuite [Tanenbaum 2002, Gardner 2001, Ko 2004].

Printre aceste probleme se numără:

- Folosirea unui sistem centralizat pentru rețelele de senzori conduce invariabil la existența unui punct unic de incapacitate (single-point-of-failure).
- Folosirea unor rețele distribuite de senzori mărește complexitatea operațiilor pentru păstrarea consistenței datelor (în cazul existenței replicării pentru micșorarea timpului de acces sau pentru redundanță complexitatea operațiilor crește exponențial cu numărul de replici existente);
- Sistemele distribuite sunt în mare majoritate construite pe principii de client-server ceea ce introduce necesitatea proiectării diferențiate a entităților care fac parte din sistemul respectiv. Sisteme distribuite artificiale



omogene sunt extrem de complicate, entitățile sunt complexe de mari dimensiuni și prezintă o cantitate mare de redundanță.

- Sincronizarea entităților în sisteme distribuite introduce probleme. În sistemele distribuite în care comunicația este bazată pe mesaje este extrem de importantă ordinea în care au sosit mesajele, și, mai mult, ordinea în care au apărut evenimentele care au generat respectivele mesaje. Au fost introduse o serie de îmbunătățiri la nivelul sincronizării sistemelor distribuite, a fost creat un întreg set de algoritmi specifici destinați sincronizării entităților din sistem. Dezavantajul major al unor astfel de algoritmi este creșterea majoră a cantității de mesaje vehiculate pe rețea.
- Comunicația este în mod uzual sincronă și bine structurată (se desfășoară pe un număr bine stabilit de runde) ceea ce mărește complexitatea entităților;
- Fiecare mesaj este important, iar pierderea oricărui mesaj individual duce la afectarea rezultatului. Pentru a proteja aceste mesaje este necesară introducerea de mecanisme de aprobare (acknowledgement), expirare (timeout) și retransmisie. Aceste mecanisme duc la mărirea necesității de bandă de transmisie, la mărirea latenței de comunicație și la mărirea timpului de procesare a mesajelor în cazul folosirii primitivelor cu blocare.
- Nodurile sunt identificate în mod unic. Adresele IP sunt folosite pentru a identifica în mod unic nodurile de rețea, iar în mod uzual, pentru a mări siguranța unicității nodurilor se folosesc numere unice de identificare generate de către noduri specializate pentru fiecare nod.
- Sistemele bazate pe comunicație tind să folosească modele structurate de comunicație. Cele mai simple modele utilizate sunt datagramele, fiecărei datagramă fiindu-i alocat un număr de secvență. În mod uzual, în aceste modele de comunicație se folosesc protocoale de tip cerere-răspuns care măresc complexitatea sistemelor distribuite.
- Pentru îmbunătățirea eficienței în sistemele distribuite se obișnuiește folosirea replicării, datelor sau serverelor, care conduce la situații de inconsistență pentru rezolvarea cărora se introduc noi concepte complexe de păstrare a consistenței care, de asemenea, complică sistemele distribuite.
- Introducerea de sisteme distribuite tolerante la erori conduce la mărirea complexității în aceste sisteme.
- Securizarea în sistemele distribuite este extrem de importantă, atât pentru protejarea datelor private cât și pentru protejarea sistemului împotriva atacurilor. Această proprietate a sistemelor distribuite crește însă cantitatea de mesaje vehiculate pe rețea, iar în cazul codificării acestor mesaje, crește necesarul de putere de calcul.

#### 4.5 Probleme ale rețelelor de senzori inteligenți

Proiectanții unor rețele de senzori trebuie să țină cont de trei aspecte importante ale funcționalității rețelei: procesarea datelor, comunicația dintre noduri și întreținerea senzorilor. Dacă se încearcă folosirea unui suport de rețea fără fir, mai apar probleme suplimentare legate de dinamicitatea mediului, lățimea de bandă necesară, consumul de putere, descoperirea resurselor de rețea, controlul rețelei, rutarea, procesarea datelor în mod cooperativ.

Printre probleme de importanță crucială ale rețelelor de senzori se numără:

- Formarea de rețele ad-hoc și descoperirea de resurse în rețelele ad-hoc: fiecare nod trebuie să fie conștient de poziția sa în cadrul rețelei pentru a interacționa cu vecinii săi. În cazul rețelelor fixe, topologia rețelei este cunoscută în cazul cel mai defavorabil imediat după instalarea acesteia, dar în cazul rețelelor ad-hoc noduri de rețea pot fi îndepărtate sau adăugate în mod dinamic. Este necesară cunoașterea poziției absolute în cadrul rețelei, sau dacă aceasta nu este disponibilă, este necesară cunoașterea poziției relative față de vecinii a sensorului în cadrul rețelei [Seah 2006].
- Controlul rețelei și rutarea: un sistem distribuit de nivelul unei rețele de senzori trebuie să fie în stare să folosească și să gestioneze resurse precum lățimea de bandă, puterea computațională, și consumul de energie care se schimbă în mod dinamic și pentru care rețeaua trebuie să se adapteze în mod dinamic și autonom. În cazul rețelelor wireless, nodurile trebuie să formeze noi căi de rutare folosind algoritmi distribuți. Numărul de noduri dintr-o rețea fără fir trebuie să fie ales astfel încât să se păstreze viabile anumite concepte de redundanță necesare [Boppana 2001, Broxton 2005].
- Procesarea datelor în mod cooperativ: această problemă este legată de partea de achiziție de date corespunzătoare rețelei de senzori. Datele de la senzori parcurg întreaga cale de la senzor la partea computațională. Nodurile ar trebui să unifice datele primite de la senzorii vecini cu propriile sale date și să trimită setul de date obținut către nodul următor [Heinzelman 2002, Intanag. 2000].
- Securitate: presupune luarea de măsuri pentru împiedicarea identificării rețelei de senzori în mediul în care a fost instalată.

#### **4.6 Probleme legate de aplicarea tiparelor de comportament emergent**

În momentul de față, aplicarea tiparelor de comportament emergent în medii robotizate a rămas la nivelul studiului sau al aplicării fără suport teoretic puternic: echipe de roboți care imită comportamentul de roi al insectelor, echipe de roboți care joacă fotbal. Din cauza lipsei unui astfel de suport teoretic, aplicarea tiparelor de comportament emergent în aplicații reale a fost evitată.

Un alt motiv pentru care s-a evitat aplicarea acestor tipare emergente în aplicații reale a fost lipsa de predictibilitate a acestora. Comportamentul emergent este vizibil doar unui observator extern sistemului și conține o caracteristică de predictibilitate care crește odată cu numărul entităților care prezintă acest comportament emergent. Creșterea necontrolată a numărului entităților prezente în sistem pentru a conduce la creșterea predictibilității conduce la pierderea controlului elementelor individuale, ceea ce în aplicațiile stricte este nepermis.

Aplicarea tiparelor de comportament emergent în aplicații reale duce la lipsă de validare, de stabilire a unor metrici corespunzătoare care să stabilească printre altele nivelul creșterii eficienței sistemului în cazul implementării tiparelor emergente, nivelul scăderii consumului de energie, a puterii computaționale în sistemele embedded.

## 4.7 Concluzii

Domeniile senzorilor inteligenți și a mediilor colaborative sunt extrem de actuale beneficiind de un interes deosebit în mediile academice și industriale internaționale, care alocă o cantitate importantă de resurse materiale și umane de înaltă calificare pentru cercetările teoretice și aplicative, deschizându-se noi direcții de studiu și dezvoltare.

În capitolul de față s-au prezentat o serie de probleme, subliniate în literatura de specialitate:

- Dezvoltarea unor soluții scalabile, eficiente și precise de coordonare a sistemelor colaborative robotice și de senzori inteligenți;
- Definirea unor politici eficiente de luare a deciziilor;
- Distribuirea eficientă a sarcinilor exploatând și optimizând paralelismul acestor sisteme;
- Deși s-a dovedit potențialul și avantajele utilizării tiparelor de comportament emergent pentru o multitudine de aplicații ale sistemelor colaborative, cum ar fi cele de explorare și supraveghere de mediu, acestea nu sunt încă implementate și utilizate în mod real la nivelul decizional și de coordonare;
- În prezent, implementarea comportamentului emergent în diferite aplicații se bazează pe un set de rețele empirice, fără a se defini un suport matematic riguros;
- Nu există o formalizare a descrierii tiparelor de comportament emergent;
- Asigurarea serviciilor suport necesare operării eficiente a sistemelor colaborative: platforma de comunicație (infrastructura + stiva de protocoale), mecanisme cu cerințe minimale de resurse, pentru localizarea nodurilor sistem, asigurarea unei baze de timp comune în cadrul sistemului, cu o precizie suficientă, prin tehnici performante de sincronizare;
- Lipsa unui model unitar de reprezentare a sistemelor colaborative, care să integreze nivelul infrastructurii și nivelul coordonării împreună cu serviciile suport necesare, astfel încât să permită proiectarea, analiza și evaluarea eficienței unui anumit sistem pentru aplicația dorită.

În capitolul următor se prezintă modelul de comportament emergent care face obiectul tezei de față, precum și conceperea modelului, metricile și parametrii implicați.



## 5. MODELAREA COMPORTAMENTULUI EMERGENT

În acest capitol se prezintă conceperea unui model pentru comportament emergent bazat pe metaeuristicile descrise în capitolul 3 și care rezolvă o serie din problemele specifice mediilor colaborative distribuite prezentate în capitolul precedent.

Tiparele de comportament emergent descrise în capitolul 2 sunt folosite și ele la conceperea modelului de comportament emergent. Acestea mai sunt utilizate și la descrierea parametrilor modelului care vor fi prezentați în continuare.

Modelul pentru comportament emergent se bazează pe anumite tipare de comportament emergent preluate din studiul coloniilor de furnici. Acesta reprezintă un colectiv robotic, colaborativ, emergent ca un sistem de entități independente care au doar o cunoaștere locală a mediului și care interacționează una cu alta. Tiparele de comportament emergent de control bazate pe coloniile de furnici pot fi aplicate la o varietate de aplicații printre care se numără ocolirea de obstacole, explorarea mediilor necunoscute și colectarea de resurse.

Modelul de comportament emergent calculează următorii parametri cheie:

- Numărul minim de entități care este necesar pentru declanșarea comportamentului emergent în sistem,  $N_{TE}$ ;
- Timpul necesar entităților pentru atingerea emergenței,  $T_{TE}$ ;
- Timpul necesar pentru atingerea scopului,  $T_{TG}$ ;

Modelul se bazează pe o serie de parametri specifici aplicației precum rata de intrare, afinitatea căii, scop etc., care vor fi descriși în paragrafele care urmează.

Modelul pentru comportament emergent este validat teoretic dar și testat într-un mediu colaborativ real folosit ca și suport de implementare pentru acesta dar și pentru adaptarea acestuia pentru mișcarea bazată pe studiul coloniilor de furnici sau pe cel al bancurilor de pești; acestea vor fi prezentate în capitolele care urmează.

### 5.1 Studiul și conceperea modelului

Următoarea secțiune conține pașii care au fost urmați de la studiul coloniilor de furnici până la dezvoltarea modelului pentru comportament emergent.

#### 5.1.1. Conceperea modelului

Studiul tiparelor de comportament emergent existente în natură dezvăluie faptul că cunoștințele locale ale mediului sunt în mod uzualacompaniate de anumite tipuri de comunicație localizată care folosește mesaje semi-persistente, relevante doar în contextul locației respective. Unul dintre cele mai folosite moduri de comunicație localizată în natură este schimbul de feromoni. Studiul coloniilor de furnici demonstrează faptul că insectele sociale, precum furnicile, sunt capabile să identifice căile dintre resursele de hrană și propriile colonii fără a folosi indicatori vizuali. Furnicile depun cantități infime de feromoni pe obiectele de interes sau pe căile pe care le folosesc, marcându-le astfel pentru o utilizare ulterioară. În mod normal, furnicile urmăresc calea care are cea mai mare intensitate a feromonilor. După cum se vede în Fig. 15, dacă un obstacol obstrucționează calea de feromoni, prima furnică va ocoli obstacolul alegând în mod aleator una dintre căile posibile. În

timpul mișcării, furnica depune noi feromoni în urma sa, și, la reîntâlnirea vechii căi de feromoni de cealaltă parte a obstacolului, își continuă sarcina inițială. Alte furnici vor urma și ele calea subțire de feromoni sau vor alege în mod aleator o altă cale posibilă. Considerând că furnicile se mișcă cu aceeași viteză, cele care, datorită, șansei au ales calea mai scurtă de ocolire a obstacolului vor ajunge mai repede de cealaltă parte și, respectiv, la țel mai repede. Prin urmărirea respectivei căi, furnicile depun o cantitate de feromoni mai mare pe calea cea mai scurtă, forțând restul furnicilor să urmeze această cale.

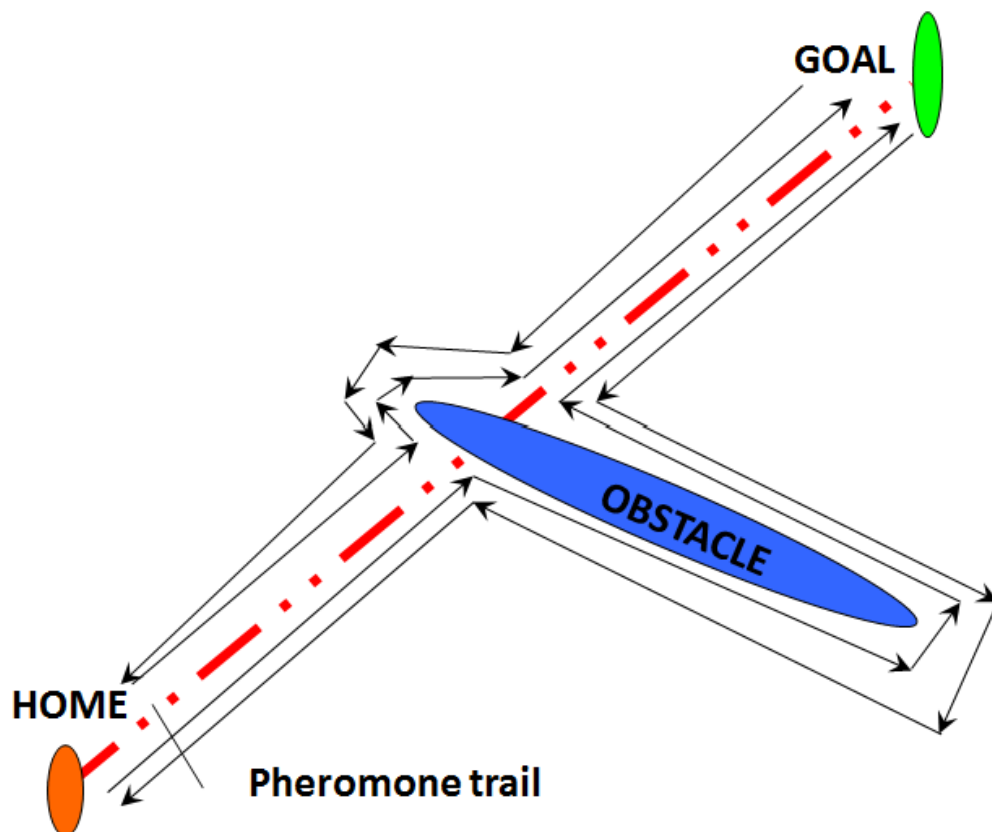


Fig. 15. Obstacol pe calea de feromoni

Modelul pentru comportament emergent a fost conceput folosind descrierea traiectoriilor tuturor căilor și a faptului că furnicile se mișcă cu o viteză constantă. O altă condiție este faptul că numărul de furnici din sistem este constant: numărul de furnici care intră în sistem trebuie să fie egal cu numărul de furnici care părăsesc sistemul, fie prin completarea țelului propus, fie printr-un eșec datorat unor cauze externe.

### 5.1.2 Descrierea mediului modelului

Mediul în care își desfășoară activitatea entitățile modelate este un sistem deschis în care acestea pot intra și ieși la momente aleatoare de timp (Fig. 16).

Pe măsura ce entitățile intră în sistem, acestora le este atribuită o sarcină specifică precum ocolirea de obstacole, explorarea mediului sau colectarea de resurse. Entitatea părăsește mediul fie prin îndeplinirea sarcinii atribuite, fie printr-o defecțiune.

Datorită faptului că există mai mult de un singur mod pentru îndeplinirea sarcinii, entitățile sunt împărțite în indivizi care au ales soluția optimă (de exemplu calea cea mai scurtă în cazul ocolirii de obstacole) și alți indivizi care au ales soluții mai puțin optime (de exemplu calea cea mai lungă în cazul ocolirii de obstacole). Dimensiunea celor două mulțimi de indivizi variază în timp, dar începând de la un anumit moment de timp, dacă comportamentul emergent a fost declanșat, numărul de entități care au ales soluția optimă ar trebui să depășească numărul de entități care au ales soluțiile mai puțin optime, marcând astfel momentul declanșării comportamentului emergent. De asemenea, numărul de entități care au trecut prin sistem până în acest moment reprezintă numărul de entități necesare pentru declanșarea comportamentului emergent.

Pentru fiecare tip de soluție (cale) se definește un interval de timp folosit de fiecare entitate pentru a-și îndeplini sarcina specifică (în cazul ocolirii de obstacole, acesta este intervalul de timp necesar unei entități să ocolească obstacolul).

## 5.2 Metrici și parametri ai modelului

Se definește  $N_E$  ca fiind numărul de entități care au intrat în sistem până la un moment de timp (momentul curent):

$$N_E = N + \zeta \quad (1)$$

unde  $N$  este numărul de entități funcționale (format din numărul de entități active care sunt în proces de îndeplinire a sarcinii și din numărul de entități care și-au îndeplinit deja sarcina) și  $\zeta$  simbolizează numărul de entități pierdute.

Numărul de entități care au ales soluția optimă sau care și-au îndeplinit sarcina prin alegerea soluției optime se notează cu  $N_S$ . Soluția optimă este reprezentată prin „calea cea mai scurtă” în aplicația de colectare de resurse cu ocolire de obstacole folosită ca exemplu în Fig. 15. Numărul de entități care au ales sau care și-au îndeplinit sarcina prin alegerea unei soluții neoptime („calea mai lungă” din exemplu folosit) se notează cu  $N_L$ .

$$N = N_S + N_L \quad (2)$$

$T_S$  este timpul necesar unei entități ca să își atingă sarcina folosind calea mai scurtă (soluția optimă) iar  $T_L$  este timpul necesar unei entități să își atingă sarcina prin alegerea unei căi mai lungi (soluția neoptimă).  $T_S$  și  $T_L$  sunt calculate de la momentul de timp la care entitatea respectivă a intrat în sistem.

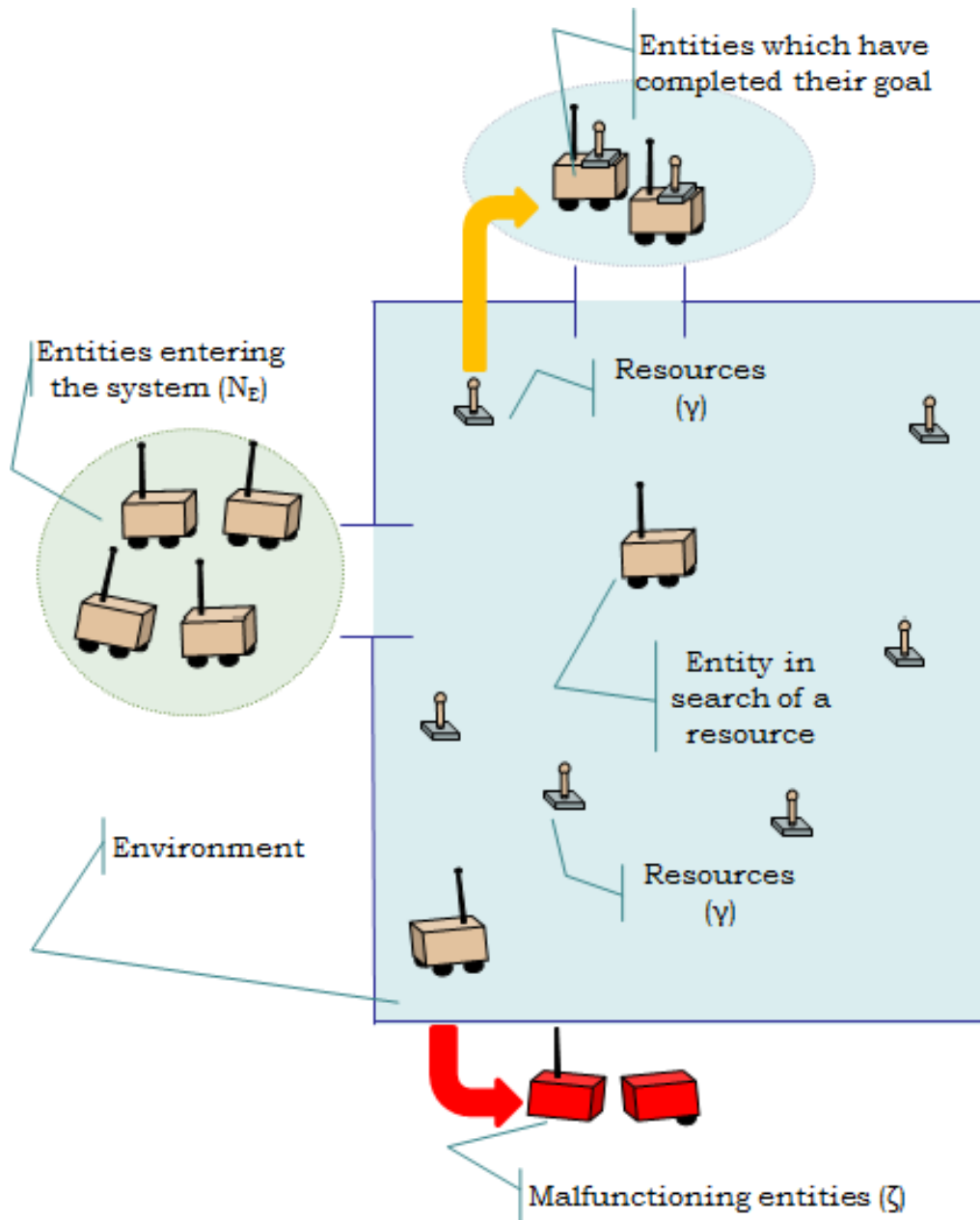


Fig. 16. Mediului modelului

Fiecare tip de soluție (sau cale) are atașat un set de metrice. Într-un sistem ne-emergent, o entitate poate alege o cale/soluție fără a avea la dispoziție informații anterioare. Într-un sistem emergent modelat în teza de față similar cu coloniile de



furnici, o entitate poate alege o cale bazându-se pe gradul de „apreciere” sau „atractivitate” ce denotă cât de mult preferă o entitate o soluție. În cazul unei colonii de furnici, decizia este bazată pe intensitatea feromonilor fiecărei căi; furnicile individuale aleg întotdeauna calea de feromoni cu intensitatea cea mai mare.

Fiecare cale are asociată o valoare,  $A$ , reprezentând gradul de preferință pe care o are entitatea pentru alegerea căii respective. Această valoare se numește *afinitatea căii*. Afinitatea căii este incrementată în cazul unei colonii de furnici reale în momentul în care furnica alege calea respectivă, prin depunerea feromonilor proprii în timpul mișcării pe cale. Pe de altă parte, în cazul mediilor colaborative robotizate afinitatea căii trebuie incrementată doar la sfârșitul sarcinii, după ce obstacolul a fost evitat, pentru a semnaliza faptul că calea este validă (obstacolul este ocolit într-un timp finit).

Toate entitățile care urmează vor alege dintr-un set posibil de soluții pe cea care are afinitatea cea mai mare (dacă există mai multe soluții care au aceeași afinitate, soluția finală va fi aleasă aleator). Depinzând de cât de similare sunt soluțiile (sau, în cazul ocolirii de obstacole, cât de lungi sunt căile), afinitățile corespunzătoare pot avea valori similare, forțând astfel alegerea unei soluții neoptime. Totuși, comportamentul emergent garantează alegerea soluției optime de toate entitățile rămase, într-un interval finit de timp.

Se definește  $A_S$  ca fiind afinitatea soluției optime („calea cea mai scurtă”) și  $A_L$  ca fiind afinitatea soluției neoptime („calea cea mai lungă”). Amândouă afinitățile,  $A_S$  și  $A_L$ , vor fi afectate doar după  $T_S$  și, respectiv,  $T_L$  secunde deoarece  $T_S$  și  $T_L$  reprezintă intervalele de timp minime necesare unei entități să ocolească obstacolul și să incrementeze afinitatea căii alese.

Inițial,

$$A_S(\tau) = 0, \forall \tau < T_S \quad (3)$$

unde  $\tau$  reprezintă timpul.

Similar,

$$A_L(\tau) = 0, \forall \tau < T_L. \quad (4)$$

În cazul în care afinitățile soluțiilor sunt egale, o entitate poate să aleagă o soluție în mod aleator cu o probabilitate  $P$ . Acest parametru este asociat fiecărei tip de soluție:  $P_S$  reprezintă probabilitatea asociată soluției optime, iar  $P_L$  reprezintă probabilitatea asociată soluțiilor neoptime. Ecuațiile (3) și (4) reprezintă condițiile inițiale ale modelului.

O altă metrică care va fi folosită pentru evaluarea modelului propus este *gradul de activitate* (eng. liveness), care este reprezentat de raportul dintre numărul de entități care își îndeplinesc sarcina propusă și numărul de entități care îndeplinesc și sarcini suplimentare (de exemplu ocolirea de obstacole dacă sarcina principală este colectarea de resurse).

$$L = \frac{N_{task}}{N_{sup}} \quad (5)$$

unde  $N_{task}$  reprezintă numărul de entități care își îndeplinesc sarcina propusă și  $N_{sup}$  reprezintă numărul de entități care îndeplinesc și sarcini suplimentare.

*Gradul de activitate* al sistemului descrie eficiența sistemului: pe măsură ce crește *gradul de activitate*, crește și eficiența sistemului deoarece mai multe entități sunt dedicate îndeplinirii sarcinii primite fără a mai executa alte sarcini suplimentare.

### 5.3 Funcția scop

Modelul de comportament emergent este un model general, capabil de a formaliza tipare de comportament emergent în cazul a numeroase aplicații. Din acest motiv, modelul are o formă generală fixă dar conține o serie de parametri variabili a căror descrieri formale se pot modifica de la o aplicație la alta cum ar fi descrierea mediului.

Fiecare aplicație modelată poate avea asociate o mulțime de tipare de comportament emergent folosite pentru a rezolva anumite probleme specifice (printre probleme amintite în capitolul 4 se numără: consumul de energie datorat unei necesități mărite de putere de calcul folosită pentru executarea unor algoritmi complecși precum și costul ridicat care rezultă din aceasta). Tiparele de comportament emergent sunt descrise prin intermediul funcției *scop*,  $\gamma$ ; aceasta formalizare depinde de aplicația modelată; de exemplu, tiparul de comportament emergent „ocolire de obstacole” prezentat în paragrafele anterioare are interpretări diferite pentru aplicații de colectare de resurse și pentru aplicații de explorare a mediilor.

Totodată, este necesară existența unui parametru care să descrie măsura în care sarcinile specifice aplicației modelate au fost îndeplinite. Acesta este formalizat tot prin intermediul funcției *scop*,  $\gamma$ .

Funcția *scop*,  $\gamma$ , reprezintă gradul în care fiecare entitate și-a îndeplinit sarcina propusă; funcția are valori mai mari pe măsură ce entitatea se apropie de îndeplinirea scopului.

Forma funcției  $\gamma$  este determinată pe baza unor proprietăți și constrângeri în faza inițială, de proiectare a aplicației modelate, urmând ca valorile sale să fie calculate în timpul aplicării algoritmului.

Funcția *scop*

$$\gamma : Z \rightarrow \{0, \dots, \Gamma\}, \gamma(\tau) \in \mathbb{N} \quad (6)$$

are următoarele proprietăți:

- funcția *scop* este monoton crescătoare:  $\gamma(\tau) \leq \gamma(\tau + 1), \forall \tau \in Z$ ;
- $\Gamma$  este valoarea maximă a funcției *scop* reprezentând faptul că sarcina a fost îndeplinită. Valoarea  $\Gamma$  poate fi orice întreg mai mare sau egal cu 1 a cărui valoare se potrivește cu aplicația pentru care se folosește modelul. Pe măsură ce valoarea lui  $\Gamma$  crește, crește și complexitatea calculului dar se mărește și precizia modelului. Folosirea unei valori de 1 pentru  $\Gamma$  reduce semnificativ complexitatea calculului, în acest caz existând doar două valori posibile: 0 reprezentând neîndeplinirea sarcinii, 1 reprezentând îndeplinirea sarcinii.
- Prima derivată a funcției  $\gamma$  trebuie să aibă cel puțin o rădăcină în punctul reprezentat de coordonatele scopului (precum coordonatele resurselor în cazul aplicațiilor de colectare de resurse). Acest punct trebuie să reprezinte valoarea maximă a funcției *scop* (semnul derivatei trebuie să se schimbe de o parte și de alta a acestui punct).
- Derivata a doua a funcției  $\gamma$  trebuie să aibă cel puțin o rădăcină în punctul reprezentat de coordonatele scopului. Valorile derivatei a doua trebuie să fie negative pentru intrări mai mici decât scopul propus, asigurându-se astfel forma concavă a funcției *scop*.

În continuare se prezintă interpretarea funcției *scop* pentru o serie de aplicații specifice mediilor robotizate: ocolire de obstacole, explorarea emergentă a unui mediu și colectarea de resurse.

Depinzând de aplicația care este modelată, funcția *scop*,  $\gamma$ , poate avea diferite interpretări. În cazul aplicațiilor de ocolire de obstacole, *scop* înseamnă că toate entitățile trebuie să ajungă la poziția *țintă*. Prin introducerea funcției  $\delta_{oa}(\tau)$  (reprezentând inversul pătratului distanței dintre o entitate anume și ținta acesteia) ca o funcție ale cărei valori cresc în timp pe măsură ce entitatea se apropie de țintă,

$$\delta_{oa}(\tau) = ((x_c[i][\tau] - x_g)^2 + (y_c[i][\tau] - y_g)^2)^{-1} \quad (7)$$

funcția *scop*,  $\gamma$  devine:

$$\gamma(\tau) = \sum_{i=1}^{N_E} \left[ \frac{\Gamma^2 \delta_{oa}(\tau)}{1 - \Gamma \delta_{oa}(\tau)} \right] \quad (8)$$

unde  $\lceil x \rceil$  reprezintă cel mai mic întreg, mai mare sau egal cu  $x$ ,  $(x_c[i][\tau], y_c[i][\tau])$  sunt coordonatele entității  $i$  la momentul de timp  $\tau$ ,  $(x_g, y_g)$  reprezintă coordonatele țintei,  $N_E$  numărul de entități și  $\Gamma$  este egală cu cea mai mare distanță din mediu. De exemplu, dacă mediul este reprezentat ca un pătrat cu latura  $a$  atunci  $\Gamma > a\sqrt{2}$ ; dacă mediul este reprezentat ca un dreptunghi cu laturile  $a$  și  $b$  atunci  $\Gamma > \sqrt{a^2 + b^2}$ .

În aplicațiile de explorare a mediului *scopul* este ca întregul mediu să fie explorat cu un număr minim de entități. Dacă harta mediului este cunoscută datorită unor investigații anterioare, aceasta este divizată într-o grilă și, în acest caz, *scopul* este reprezentat prin navigarea întregului set de coordonate *țintă* reprezentat de celulele grilei. Dacă harta mediului nu este cunoscută, atunci entitățile vor explora mediul într-o manieră recursivă, folosind poziția curentă ca poziție de start și, apoi, calculând poziția următoare dintr-un set de mișcări posibile.

Astfel,

$$\delta_{es}(\tau) = ((x_c[i][\tau] - x_g[j])^2 + (y_c[i][\tau] - y_g[j])^2)^{-1} \quad (9)$$

$$\gamma(\tau) = \sum_{i=1}^{N_E} \sum_{j=1}^m \left[ \frac{\Gamma^2 \delta_{es}(\tau)}{1 - \Gamma \delta_{es}(\tau)} \right] \quad (10)$$

unde  $\delta_{es}(\tau)$  este o funcție similară cu (7),  $(x_c[i][\tau], y_c[i][\tau])$  sunt coordonatele entității  $i$  la momentul de timp  $\tau$ ,  $(x_g[j], y_g[j])$  reprezintă coordonatele celulei  $j$  a grilei de explorare,  $N_E$  numărul de entități și  $m$  este dimensiunea grilei. Originea sistemului de axe de coordonate se suprapune peste poziția de start  $(x_s, y_s)$ .  $\Gamma$  are aceeași semnificație ca în paragrafele precedente.

Aplicațiile de colectare de resurse, care sunt o combinație a celor două tipuri de aplicație prezentate în paragrafele precedente, au o funcție *scop* care prezintă în mod adițional clauze pentru întoarcerea la poziția *acasă*.

$$\delta_{rg}^{rf}(\tau) = ((x_c[i] - x_g[j])^2 + (y_c[i] - y_g[j])^2)^{-1} \quad (11)$$

$$\delta_{rg}^{rf}(\tau) = ((x_c[i] - x_h)^2 + (y_c[i] - y_h)^2)^{-1} \quad (12)$$

$$\gamma(\tau) = \begin{cases} \sum_{i=1}^{N_E} \sum_{j=1}^m \left[ \frac{\Gamma^2 \delta_{rg}^{rnf}(\tau)}{1 - \Gamma \delta_{rg}^{rnf}(\tau)} \right], & \tau < T_r \\ \sum_{i=1}^{N_E} \left[ \frac{\Gamma \delta_{rg}^{rf}(\tau)}{1 - \Gamma^2 \delta_{rg}^{rf}(\tau)} \right], & \tau \geq T_r \end{cases} \quad (13)$$

unde  $\delta_{rg}^{rnf}(\tau)$  și  $\delta_{rg}^{rf}(\tau)$  sunt funcții similare cu (7),  $(x_c[i][\tau], y_c[i][\tau])$  sunt coordonatele entității  $i$  la momentul de timp  $\tau$ ,  $(x_g[j], y_g[j])$  reprezintă coordonatele resursei  $j$ ,  $T_r$  este timpul necesar unei entități să parcurgă distanța de la intrarea în sistem până la prima resursă,  $(x_h, y_h)$  reprezintă coordonatele poziției *acasă*,  $N_E$  numărul de entități și  $m$  este dimensiunea grilei. Originea sistemului de axe de coordonate se suprapune peste poziția de start  $(x_s, y_s)$ .  $\Gamma$  are aceeași semnificație ca în paragrafele precedente. Condiția  $\tau < T_r$  din (13) descrie faptul că resursa curentă nu a fost colectată încă, iar  $\tau \geq T_r$  descrie faptul că entitatea este în drum spre casă, după colectarea unei resurse.

#### 5.4 Modelul de comportament emergent

Modelul descrie  $N_S$ , numărul de entități care au ales soluția optimă, ca o funcție recursivă pe două nivele. Valorile  $A_S$  și  $A_L$  sunt calculate la rândul lor pe baza unor valori anterioare ale  $N_S$  și, respectiv,  $N_L$ .

$$N_S(\tau) = \begin{cases} \gamma(\tau)(N_S(\tau-1) + \lambda), & A_S(\tau-1) > A_L(\tau-1) \\ \gamma(\tau)N_S(\tau-1), & A_S(\tau-1) < A_L(\tau-1) \\ \gamma(\tau)(N_S(\tau-1) + \lambda), & A_S(\tau-1) = A_L(\tau-1), P_S > P_L \\ \gamma(\tau)N_S(\tau-1), & A_S(\tau-1) = A_L(\tau-1), P_S \leq P_L \end{cases} \quad (14)$$

unde

$$\lambda = \frac{dN_E}{dt} r(1, P_S, 0, P_L) \quad (15)$$

reprezintă numărul de entități care aleg soluția optimă în unitatea de timp,  $r(a, P_a, b, P_b)$  este o funcție care alege în mod aleator una din opțiunile  $a$  și  $b$  cu posibilitățile de apariție  $P_a$  și, respectiv,  $P_b$  și  $N_E$  a fost definit în (1).

Condițiile inițiale sunt:

$$N_S(0) = 0, \quad (16)$$

$$N_S(1) = \begin{cases} 1, & P_S > P_L \\ 0, & P_S \leq P_L \end{cases}, \quad (17)$$

$$N_S(2) = \begin{cases} 0, & P_S > P_L \\ 1, & P_S \leq P_L \end{cases} \quad (18)$$

O declarație de gardă a fost introdusă în condițiile inițiale din (17) și (18) pentru a mări eficiența sistemului și pentru a-i păstra stabilitatea: după ce prima entitate a ales o soluție în mod aleator, următoarea entitate va alege soluția opusă

pentru a rezolva problema cazului cel mai defavorabil. Cazul cel mai defavorabil pentru această situație este atunci când entitatea a ales soluția neoptimă în mod aleatoriu, iar următoarea entitate intră în sistem la mult timp după ce prima entitate și-a îndeplinit țelul. În acest caz, afinitatea soluției optime o fost incrementată înainte ca oricare altă entitate să aibă ocazia să incrementeze afinitatea soluției optime, rezultând în forțarea alegerii soluției neoptime de toate celelalte entități.

Similar cu (14):

$$N_L(\tau) = \begin{cases} \gamma(\tau)N_L(\tau-1), & A_S(\tau-1) > A_L(\tau-1) \\ \gamma(\tau)(N_L(\tau-1) + \rho), & A_S(\tau-1) < A_L(\tau-1) \\ \gamma(\tau)N_L(\tau-1), & A_S(\tau-1) = A_L(\tau-1), P_S > P_L \\ \gamma(\tau)(N_L(\tau-1) + \rho), & A_S(\tau-1) = A_L(\tau-1), P_S \leq P_L \end{cases} \quad (19)$$

Unde

$$\rho = \frac{dNE}{dt} r(0, P_S, 1, P_L), \quad (20)$$

reprezintă numărul de entități care aleg soluția neoptimă în unitatea de timp.

Condițiile inițiale sunt

$$N_L(0) = 0, \quad (21)$$

$$N_L(1) = \begin{cases} 0, & P_S > P_L \\ 1, & P_S \leq P_L \end{cases} \quad (22)$$

$$N_L(2) = \begin{cases} 1, & P_S > P_L \\ 0, & P_S \leq P_L \end{cases} \quad (23)$$

Declarația de protecție este prezentă, de asemenea, în (22) și (23) folosind valorile opuse celor din (17) și (18), asigurându-se astfel faptul că a doua entitate va alege soluția alternativă.

Afinitățile căilor sunt calculate pornind de la valorile anterioare ale lui  $N_S$ , respectiv  $N_L$ . Similar unei colonii de furnici în care furnicile urmează calea care are cea mai mare valoare a intensității feromonului, afinitatea căii crește de fiecare dată când o entitate a ales calea respectivă. Valoarea prin care afinitatea căii crește depinde de  $\lambda$ , respectiv  $\rho$ . Astfel:

$$A_S(\tau) = \begin{cases} A_S(\tau-1) + \lambda, & N_S(\tau-1) > N_L(\tau-1) \\ A_S(\tau-1), & N_S(\tau-1) \leq N_L(\tau-1) \end{cases} \quad (24)$$

cu condițiile inițiale:

$$A_S(0) = 0, A_S(\tau) = 0, \forall \tau \leq T_{0a} \quad (25)$$

unde  $T_{0a}$  reprezintă timpul discret în care prima entitate reușește să își îndeplinească scopul alegând oricare din soluții.

Similar cu (24):

$$A_L(\tau) = \begin{cases} A_L(\tau-1), & N_S(\tau-1) > N_L(\tau-1) \\ A_L(\tau-1) + \rho, & N_S(\tau-1) \leq N_L(\tau-1) \end{cases} \quad (26)$$

cu condițiile inițiale:

$$A_L(0) = 0, A_L(\tau) = 0, \forall \tau \leq T_{0a} \quad (27)$$

Folosind un set de funcții ajutătoare, (14) devine

$$N_S(\tau) = \gamma(\tau) * [N_S(\tau - 1) + I(P_S - P_L)eq(A_S(\tau - 1) - A_L(\tau - 1)) + I(A_L(\tau - 1) - A_S(\tau - 1))] \quad (28)$$

unde

$$I : \mathbb{R} \rightarrow \{0, 1\}, I(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leq 0 \end{cases} \quad (29)$$

și

$$eq : \mathbb{R} \rightarrow \{0, 1\}, eq(x) = \begin{cases} 1, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (30)$$

Similar, (19) devine

$$N_L(\tau) = \gamma(\tau) * [N_L(\tau - 1) + \rho(P_L - P_S)eq(A_S(\tau - 1) - A_L(\tau - 1)) + \rho(A_S(\tau - 1) - A_L(\tau - 1))] \quad (31)$$

De asemenea, folosind același set de funcții ajutătoare, formulele pentru calcularea afinității căii devin:

$$A_S(\tau) = A_S(\tau - 1) + I(N_S(\tau - 1) - N_L(\tau - 1)) \quad (32)$$

și

$$A_L(\tau) = A_S(\tau - 1) + I(N_L(\tau - 1) - N_S(\tau - 1)) \quad (33)$$

În situațiile din viața reală, unele entități se pot pierde în mediu datorită unui număr de eșecuri posibile, așadar numărul de entități care este necesar pentru declanșarea comportamentului emergent se calculează prin minimizarea funcției

$$N_E(\tau) = N_S(\tau) + N_L(\tau) + \zeta(\tau) \quad (34)$$

unde, conform cu (1),  $\zeta$  reprezintă numărul de entități care sunt pierdute în mediu.

Modelul de comportament emergent este:

$$\mathcal{M} = \langle NE, A, N, P, \{(28), (31), (32), (33), (34)\} \rangle \quad (35)$$

Ecuția (35) prezintă modelul de comportament emergent ca fiind compus din parametrii  $N_E, A, N, P$  fiecare cu valorile corespunzătoare împreună cu sistemul descris de ecuațiile (28), (31), (32), (33) și (34).

Eficiența sistemului este descrisă de procentul din numărul total al entităților care și-au efectuat sarcina într-un anumit interval de timp.

Afinitățile sunt definite de (24) și (26). Această metrică influențează performanța întregului, cuantificând cât de repede o entitate își poate atinge obiectivul.

Numărul de entități este un parametru important pentru evaluarea modelului și este specificat de (14) și (19).

Din modelul de comportament emergent din (35) se pot calcula următorii parametri cheie:

- Timpul necesar entităților pentru atingerea emergenței,  $T_{TE}$ : acesta se calculează prin integrarea relației (28) în raport cu variabila de timp,  $\tau$ .

$$T_{TE} = \int_{-\infty}^{\infty} N_S(\tau) d\tau \quad (36)$$

- Numărul minim de entităţi care este necesar pentru declanşarea comportamentului emergent în sistem,  $N_{TE}$ : se calculează prin introducerea în ecuaţia (28) a valorii calculate anterior,  $T_{TE}$ ;

$$N_{TE} = N_S(T_{TE}) \quad (37)$$

- Timpul necesar pentru atingerea scopului,  $T_{TG}$ : deoarece acesta depinde de numărul de entităţi din sistem ( $T_{TG}$  fiind timpul în care toate entităţile intrate în sistem şi-au îndeplinit sarcina),  $T_{TG}$  se calculează prin integrarea relaţiei (34) în raport cu variabila de timp,  $\tau$ .

$$T_{TE} = \int_{-\infty}^{\infty} N_E(\tau) d\tau = \int_{-\infty}^{\infty} (N_S(\tau) + N_L(\tau) + \zeta(\tau)) d\tau \quad (38)$$

## 5.5 Concluzii

Pornind de la studiul comportamentului coloniilor de furnici în cazul explorării mediilor necunoscute, colectării de resurse precum şi al ocolirii de obstacole, s-a definit un model matematic folosit pentru determinarea numărului minim de entităţi necesar pentru declanşarea comportamentului emergent în situaţia unui colectiv robotic.

Pe baza acestui model se mai pot calcula timpul necesar declanşării comportamentului emergent precum şi timpul necesar îndeplinirii ţelului în aplicaţiile implementate.

Modelul introdus propune un nou suport matematic pentru analizarea, modelarea şi evaluarea interacţiunilor şi a gradului de cooperare în sisteme robotizate şi reţele de senzori inteligenţi prin prisma comportamentului emergent. Modelul propus poate fi de asemenea folosit ca bază pentru tehnicile de control avansate pe vehiculele autonome.

Au fost definite trei metrici: *afinitatea*, *gradul de activitate* şi *numărul de entităţi*; acestea sunt folosite pentru a evalua performanţele modelului simulat.

Evaluarea şi validarea formală a modelului este prezentată în capitolul următor. Acestea au fost realizate folosind un simulator de comportament emergent special dezvoltat şi a suitei de aplicaţii de calcul complex Mathematica.





## 6. IMPLEMENTAREA MODELULUI DE COMPORTAMENT EMERGENT ÎN MATHEMATICA

În acest capitol se prezintă validarea teoretică a modelului de comportament emergent folosind suita de aplicații de calcul complex Mathematica.

Se studiază valorile parametrilor modelului precum și valorile metricilor propuse în capitolul precedent pentru două cazuri specifice: un exemplu de ocolire de obstacole pentru care parametrii modelului au valori similare cu cele din lumea reală, și un exemplu similar de ocolire de obstacole pentru care valorile parametrilor modelului sunt deformați spre extrem pentru modelarea cazului cel mai defavorabil.

Pentru fiecare studiu de caz se determină valorile metricilor: *afinitate*, *număr de entități* și *grad de activitate*.

Primul studiu de caz se referă la situația în care în calea entităților este un obstacol depus astfel încât secțiunea sa de mijloc este situată mai departe de calea originală a entităților. Al doilea studiu de caz se referă la situația în care în calea entităților este un obstacol depus astfel încât secțiunea sa de mijloc este situată foarte aproape de calea originală a entităților determinând valori similare ale lungimii căilor de ocolire a obstacolului.

### 6.1 Validarea modelului de comportament emergent folosind Mathematica

#### 6.1.1 Studiu de caz: Exemplu din lumea reală

În primul caz parametrii  $T_S$  și  $T_L$  au valori foarte diferite (vechea cale de feromon este la mai mult de jumătate depărtare de secțiunea de mijloc a obstacolului). De asemenea, posibilitatea de alegere a fiecărei căi este identică (așa cum ar fi în situațiile reale). Valorile parametrilor modelului este  $T_S = 4$ ,  $T_L = 10$ ,  $P_S = 0.5$ ,  $P_L = 0.5$ . Așa cum se arată în figura următoare (Fig. 17) toate entitățile decid rapid să urmeze calea cea mai scurtă. Un număr de 9 entități este necesar declanșării comportamentului emergent.

În Fig. 17 graficele de culoare albastră se referă la entitățile care au ales calea mai scurtă (cea corectă), iar cele de culoare roșie se referă la entitățile care au ales calea mai lungă.

Pentru un set de valori ale parametrilor modelului de  $t_S = 5$ ,  $t_L = 10$ ,  $P_S = 0.4$ ,  $P_L = 0.6$ , toate entitățile decid rapid să urmeze calea cea mai scurtă (Fig. 18). Un număr de 10 entități este necesar pentru a declanșa comportamentul emergent în acest caz. Micile diferențe dintre aceste valori și cele anterioare se explică prin posibilele coliziuni dintre entități.

În Fig. 18 graficele de culoare albastră se referă la entitățile care au ales calea mai scurtă (cea corectă), iar cele de culoare roșie se referă la entitățile care au ales calea mai lungă.

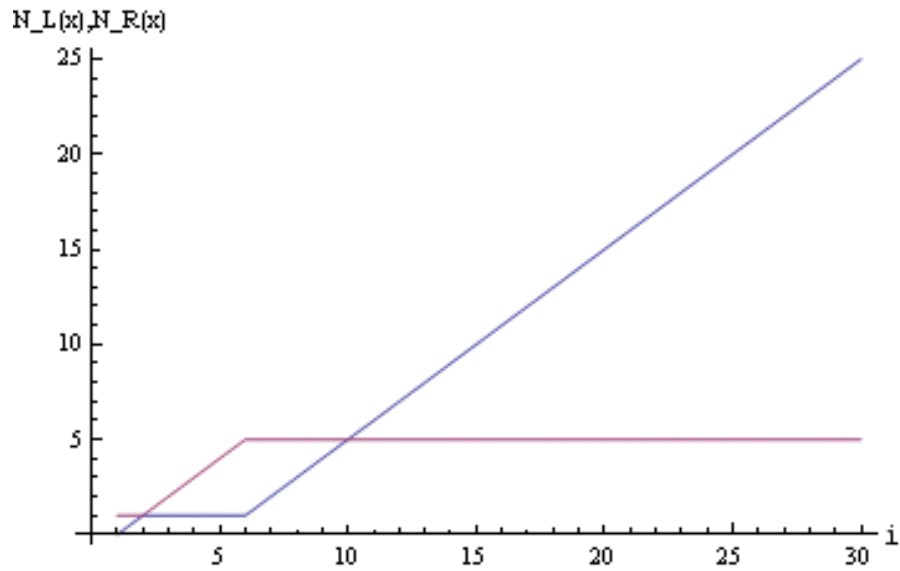


Fig. 17. Numărul de entități care se află pe cele 2 căi

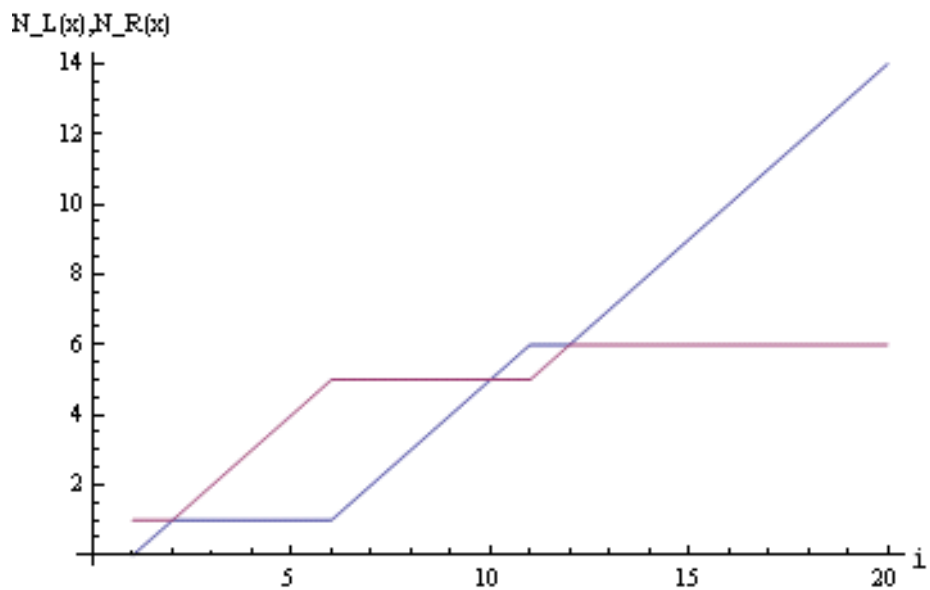


Fig. 18. Numărul de entități care se află pe cele 2 căi

### 6.1.2 Studiu de caz: Situația cea mai defavorabilă

Al doilea caz studiat corespunde situației celei mai defavorabile, în care parametrii  $T_S$  și  $T_L$  au valori similare (vechea cale de feromoni traversează aproape prin mijlocul obstacolului). De asemenea, probabilitatea de a alege calea „greșită” este mult mai mare decât probabilitatea de a alege calea „bună”. Toate aceste

condiții înseamnă că, din cauză că prima entitate care a ales calea cea mai scurtă va ajunge în cealaltă parte aproape în același timp cu prima entitate care a ales calea cea mai lungă, în acest interval de timp un număr mai mare de entități ar fi ales calea cea lungă, crescându-i astfel mai mult afinitatea. Valorile parametrilor modelului sunt  $T_S = 9$ ,  $T_L = 10$ ,  $P_S = 0.1$ ,  $P_L = 0.9$ .

Așa cum se observă în figura următoare (Fig. 19), pe tot parcursul fazei inițiale instabile a sistemului, entitățile sunt indecise alegând fie calea cea scurtă, fie calea cea lungă (din cauza probabilității mai mari de alegere a căii mai lungi); după ce 11 entități au pătruns în sistem, acestea se stabilizează și toate entitățile următoare care pătrund în sistem aleg calea cea scurtă.

În Fig. 19 graficele de culoare albastră se referă la entitățile care au ales calea mai scurtă (cea corectă), iar cele de culoare roșie se referă la entitățile care au ales calea mai lungă.

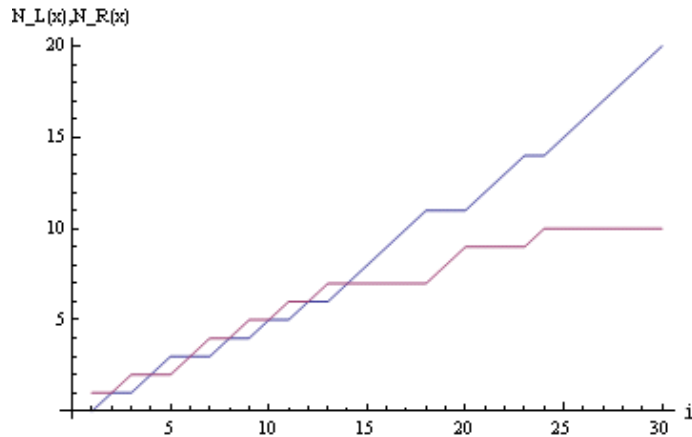


Fig. 19. Numărul de entități în situația cea mai defavorabilă

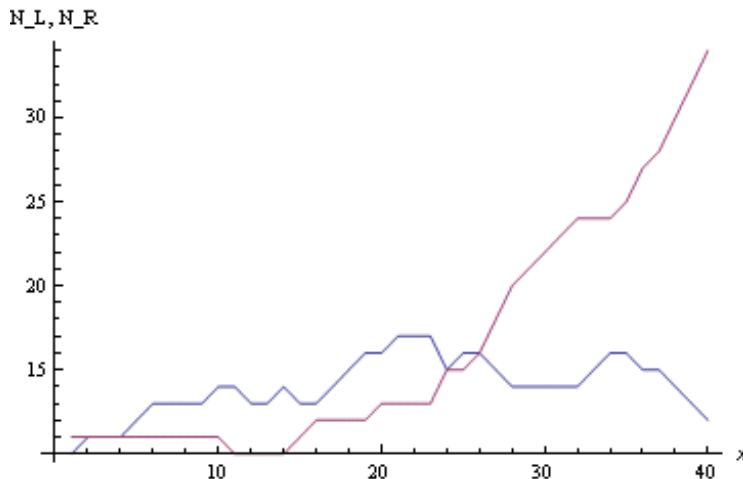


Fig. 20. Numărul de entități în situația cea mai defavorabilă

Pentru a declanșa comportamentul emergent în cazul folosirii valorilor următoare ale parametrilor modelului:  $T_S = 11$ ,  $T_L = 12$ ,  $P_S = 0.05$ ,  $P_L = 0.95$ , sunt necesare un număr de 12 entități (Fig. 20). În Fig. 20, graficele de culoare roșie se referă la entitățile care au ales calea mai scurtă, iar cele de culoare albastră se referă la entitățile care au ales calea mai lungă.

### 6.1.3 Gradul de activitate al sistemului

Validarea modelului în simulator a prezentat o limită superioară a numărului de entități care pot fi utilizate pentru a declanșa comportamentul emergent. O concentrație ridicată de entități afectează serios comportamentul emergent a sistemului, și astfel, performanța lui, din cauza suprainteracțiunilor și posibilelor coliziuni din sistem.

Această limită superioară pentru numărul de indivizi a fost calculată, pe baza unui set de parametri a simulatorului. După cum se vede în figura următoare, Fig. 21, gradul de activitate al sistemului scade când numărul entităților crește peste 30.

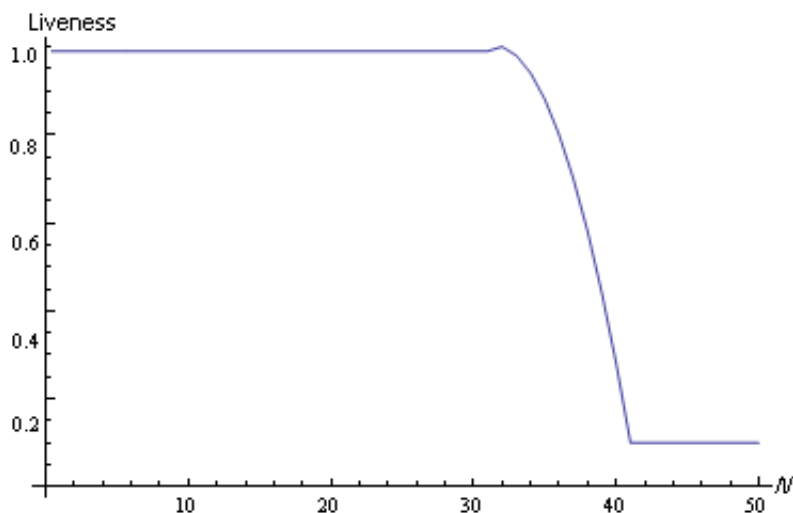


Fig. 21. Gradul de activitate al sistemului în cazul aglomerării masive

În Fig. 22, în interiorul intervalului calculat de 9 până la 30 de entități posibile, creșterea numărului de entități duce la creșterea performanței sistemului. Când dublăm numărul entităților de la 9 la 18 crește performanța către un maxim de 19,3%.

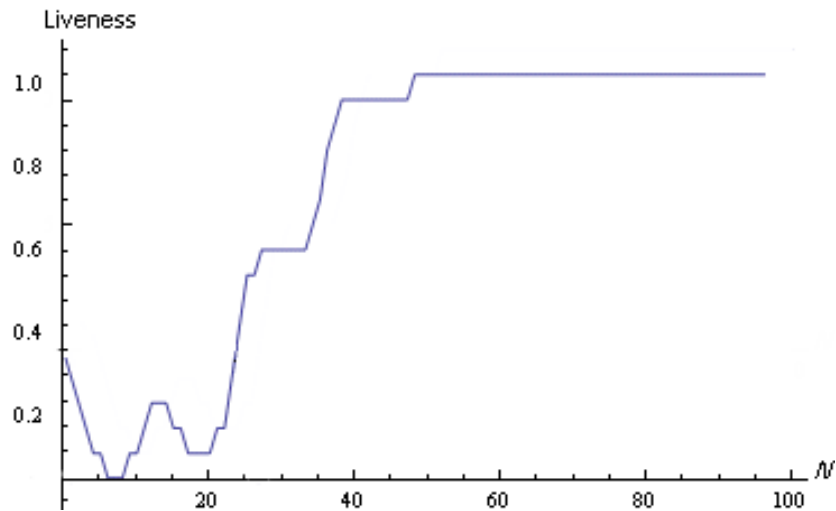


Fig. 22. Gradul de activitate al sistemului în cazul aglomerării minime

## 6.2 Concluzii

Modelul de comportament emergent introduce noi formalizări care oferă o bază matematică pentru modelarea căilor de control a comportamentului emergent și își demonstrează utilitatea în modelarea comportamentului emergent prin prezentarea unui algoritm de evitare a obstacolului.

Aceste formalisme reprezintă fundamentul matematic pentru analiza și evaluarea interacțiunilor și gradul de cooperare în sistemele robotizate și rețelele senzoriale inteligente. Modelul rezultat poate fi de asemenea utilizat ca bază pentru dezvoltarea tehnicilor avansate de control pe vehicule autonome.

Rezultatele experimentale ale modelului aplicat algoritmului de evitare a obstacolului au prezentat un interval în care se declanșează comportamentul emergent, cu o limită inferioară de 9 entități și o limită superioară de 30 de entități.

Experimentele de validare a modelelor propuse de comportament emergent au demonstrat viabilitatea acestora și le recomandă pentru analiza și dezvoltarea unor aplicații complexe ce necesită inteligențe distribuite și colaborative, percepție și funcționare.

Capitolul următor prezintă suportul folosit pentru evaluarea și validarea modelului matematic, atât experimental cât și prin simulări: sistemul integrat CORE-TX, simulatorul eBMS, limbajul de specificare a comportamentului emergent eBML.



## **7. SUPORTUL PENTRU TESTAREA ȘI IMPLEMENTAREA MODELULUI PENTRU COMPORTAMENT EMERGENT**

În acest capitol se prezintă o platformă integrată folosită la validarea, implementarea și testarea modelului pentru comportament emergent prezentat în capitolul 5.

Platforma integrată prezentată este folosită pentru implementarea modelului și pentru testarea acestuia în două cazuri care vor fi prezentate în capitolele care urmează:

- folosind tipare de comportament emergent preluate din studiul mișcării în coloniile de furnici (capitolul 9);
- folosind tipare de comportament emergent preluate din studiul mișcării în bancurile de pești (capitolul 10).

Simulatorul care face parte din platforma integrată este folosit pentru evaluarea teoretică a modelului de comportament emergent. Rezultatele simulărilor sunt prezentate în capitolul următor.

### **7.1 Introducere**

Validarea și analiza performanțelor modelului dezvoltat în cadrul programului de doctorat se face prin utilizarea unui simulator special creat precum și prin utilizarea prototipului platformei integrate destinată studiului acestor sisteme, CORE-TX [Cioarga 2006], dezvoltată de către doctorand împreună cu echipa de proiect în cadrul unui grant de cercetare de excelență, numărul CEEX-ET-07/2006-2008, MEdC – UEFISCSU.

În scopul evaluării și testării modelelor definite a fost proiectat și implementat un simulator de comportament emergent numit eBMS (Emergent Behavior Model Simulator). Comportamentul emergent este descris cu ajutorul limbajului eBML. Fiecare entitate dispune de o instanță a interpretorului de limbaj eBML [Cioarga 2007a, Cioarga 2007b]. Acesta folosește ca intrare un fișier de descriere a comportamentului pe care, mai apoi, îl transformă într-un cod executabil specific procesorului țintă. Interpretorul este structurat într-o manieră modulară folosind fișiere de XML pentru configurarea subsetului limbajului eBML folosit precum și a regulilor de transformare în limbajele necesare.

Simulatorul este o aplicație orientată pe obiecte, scrisă în C++, folosind mediul de dezvoltare Microsoft Visual Studio 2008. Acesta este compus din două module principale: un modul de configurare și un modul de simulare.

Modulul de configurare este reprezentat, în principal, de o interfață grafică destinată configurării unei colecții de obiecte de tip Entitate fiecare având o serie de parametri: poziția de start, țelul, setul de senzori (senzori de atingere, senzori de poziționare), dimensiuni (înălțime, lungime, lățime) precum și a comportamentului pe care aceste entități trebuie să îl manifeste.

Modulul de simulare primește de la modulul de configurare lista de obiecte entitate pe care apoi le simulează corespunzător într-un mediu a cărui parametri (dimensiuni, obstacole, poziții inițiale etc) sunt descriși printr-o hartă.

Sistemul CORE-TX (COllaborative Robotic Environment – the Timișoara eXperiment, a se citi "cortex") este conceput ca o platformă complexă, compusă la nivelul arhitecturii dintr-un set eterogen de micro sisteme autonome cu inteligență încorporată, un mediu de comunicații de tip colaborativ și o entitate centrală cu rol de configurare, control și supervizare a întregului sistem.

Modelul sistemului CORE-TX prezentat în Fig. 23 se bazează pe aplicarea principiilor avansate de arhitectură a sistemelor digitale complexe: modularitate, flexibilitate, fiabilitate, reconfigurabilitate, abstractizare funcțională etc. precum și pe aplicarea principiilor avansate de inginerie software împreună cu modalități de încorporarea a tiparelor de comportament emergent.

Nivelul superior de abstractizare a modelului CORE-TX este reprezentat de entitatea BRAIN (Background Robotic Activity Induction Node), implementată cu ajutorul unui calculator gazdă (PC) cu conectivitate sporită pentru comunicații (se intenționează utilizarea sistemelor tip radio/wireless), pe care rulează un sistem de pachete software originale, concepute special pentru asigurarea următoarelor funcții principale.

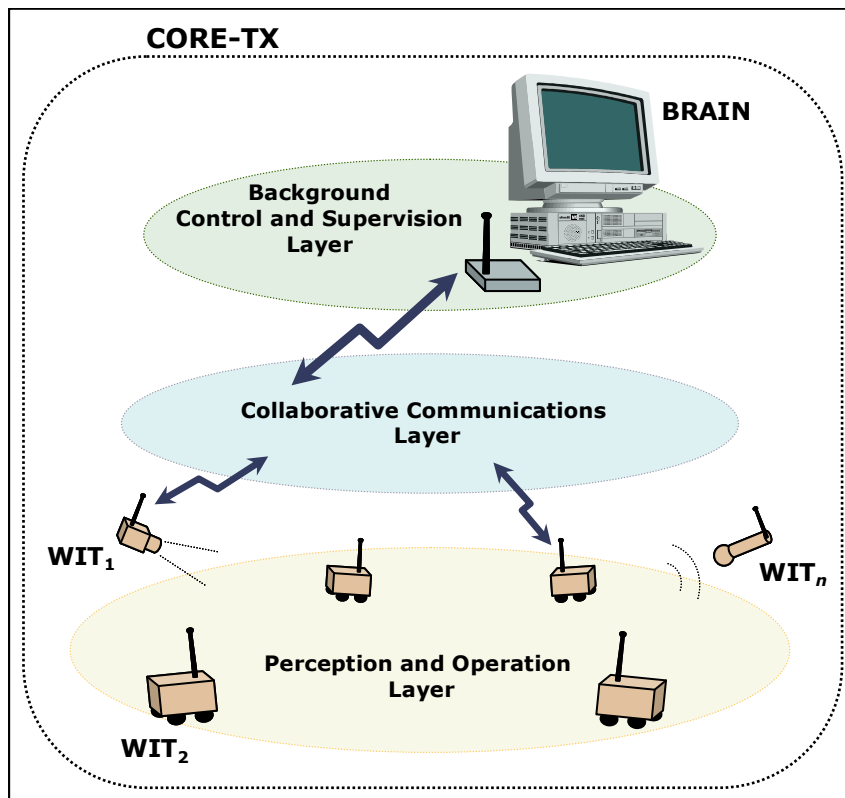


Fig. 23. Modelul CORE-TX ([Cioarga 2006])

La nivelul percepției și operării, sistemul CORE-TX interacționează cu mediul înconjurător prin intermediul micro sistemelor autonome cu inteligență încorporată, denumite WIT (Wireless Intelligent Terminal). Elementele WIT pot avea funcții de



percepție (senzori inteligenți) sau de operare / acționare (mini-roboti autonomi), sau combinate.

Atunci când entitatea WIT intră în mediul CORE-TX, aceasta își prezintă proprietățile entității BRAIN. Unele din entitățile WIT pot fi mobile; în aceste condiții, apare problema localizării. Sistemul CORE-TX este proiectat într-o astfel de măsură încât deși entitățile WIT pot fi adresate individual în scopuri specifice aplicației care este rulată pe sistemul CORE-TX, la un nivel superior de abstractizare, entitățile WIT pot fi identificate prin locația acestora. Este foarte important ca interconexiunile de comunicație să nu fie inundate cu informații de fundal referitoare la adresarea individuală a nodurilor. Problema localizării este lăsată unui nivel inferior de abstractizare care este responsabil cu administrarea rutelor.

Restul dispozitivelor WIT sunt împrăștiate în mediul dorit (Fig. 24). Acestea sunt împărțite conceptual în zone. Acest principiu asigură localizarea entităților WIT bazată pe zona de care aparține. Totuși, și în acest context apare problema localizării, deoarece, chiar dacă unele dispozitive WIT sunt imobile (camere digitale, senzori de temperatură etc), unele dispozitive WIT sunt mobile (roboti, camere mobile etc); pe măsură ce dispozitivele WIT se mișcă în mediul de desfășurare, acestea își schimbă zona și de aceea sistemul CORE-TX are proceduri specifice similare cu procedura de handover de la sistemele GSM [Cioarga 2006].

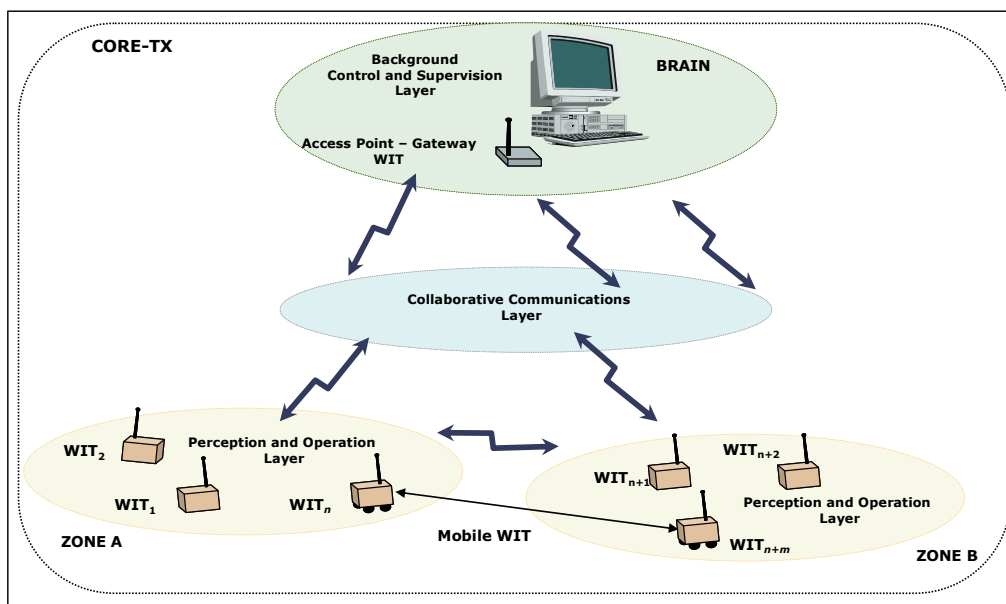


Fig. 24. Modelul CORE-TX – zone [Cioarga 2006]

Divizarea în zone este foarte eficientă când se pune problema fiabilității comunicației pentru că ascunde detalii care nu sunt necesare. De exemplu, dacă cineva are nevoie de temperatura unei încăperi, senzorii care sunt asignați acelei zone căreia îi este alocată încăperea vor transmite o temperatură medie, și nu temperatura instantanee a unui senzor oarecare. Principiul prezentat mai sus conduce la o imagine a rețelei de dispozitive WIT ca și o bază de date distribuită. Regăsirea informației dorite conduce la interogări bazate pe zone spre deosebire de interogări bazate pe ID.

## 7.2 Modelul WIT

La baza proiectării și realizării elementelor WIT stau principiile de modularitate, autonomie și inteligență digitală încorporată [Cioarga 2006]. Din punct de vedere formal dispozitivele WIT sunt identice. Deși entitățile WIT pot fi folosite pentru scopuri diverse precum punctul de acces la entitatea BRAIN, module senzoriale, toate dispozitivele WIT prezintă aceeași arhitectură. Entitatea WIT prezintă un design modular, fiind compusă din diferite plăci applicative interconectate printr-o interfață serială SPI de mare viteză.

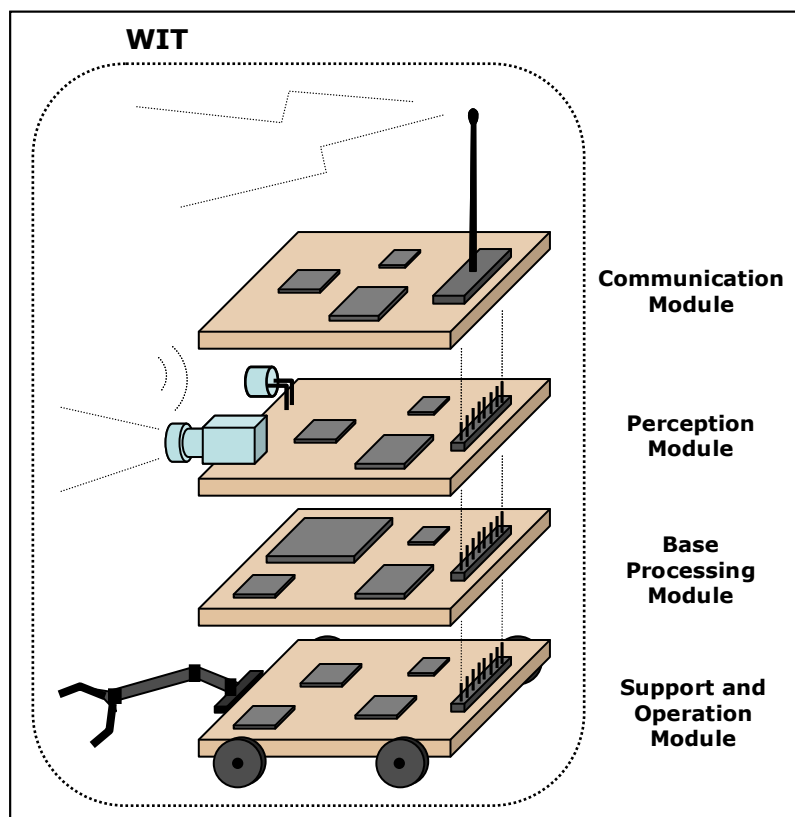


Fig. 25. Modelul WIT [Cioarga 2006]

În continuare se prezintă o configurație minimă a unui prototip WIT:

- Placa de bază sau modulul de procesare: este elementul de bază al oricărui WIT; este bazat pe microprocesorul Philips LPC2294. Modulul de procesare al unui dispozitiv WIT reprezintă elementul central de inteligență digitală al microsistemului, servind ca și o platformă pentru kernelul de operare timp-real HARETICK. Acest modul va servi ca bază pentru execuția aplicației; acest modul va fi responsabil pentru întregul comportament al dispozitivului WIT. Tot aici are loc procesarea locală a datelor achiziționate din mediu.

Toate operațiile sunt strict supravegheate și controlate pentru a minimiza consumul de energie.

- Modulul de alimentare: acest modul nu are inteligență proprie, este atașat plăcii de bază. Toate dispozitivele WIT sunt presupuse a rula pe acumulatori, fiecare WIT dispunând de resurse limitate iar modelul de comunicație trebuie să prezinte protocoale de comunicație care sunt de mică dimensiune și conștiente de nivelul de încărcare a bateriei.
- Modulul de comunicație: din principiu sistemul CORE-TX folosește comunicația fără fir. Acest tip de comunicație este implementat prin folosirea unor module X-Bee PRO. Aceste module folosesc comunicare radio bidirecțională precum și standardul Zigbee (IEEE 802.15.4). Modulele de comunicație garantează interconexiunea dintre modulele WIT la un nivel mai înalt – cel al comunicației colaborative, conținând interfețele de transmisie și de recepție a datelor precum și procesarea conform cu protocoalele care au fost special proiectate pentru acesta.

În plus față de configurația minimă descrisă mai sus, mai pot fi adăugate plăcii de bază numeroase plăci de extensie prin folosirea unei interfețe seriale SPI de mare viteză. Aceste plăci de extensie se vor baza pe microprocesorul Philips LPC2103.

- Modulul de operare și suport al dispozitivelor WIT garantează suportul (static sau mobil – cu ajutorul unui set de roți active și motoare) pentru microsistem. Ca și o opțiune suplimentară, un braț robotic poate fi atașat unui astfel de modul. Pentru un astfel de modul este necesar un modul suplimentar de alimentare pentru actuatori.
- Modulul de percepție: conține toți senzorii microsistemului WIT precum și partea de preprocesare a datelor senzoriale. Acest modul a fost proiectat folosind tehnici avansate de achiziție digitală și procesare de semnal.

Toate modulele microsistemului WIT au pinii de intrare și de ieșire disponibili pentru testare, toate modulele au conectori pentru magistrala de extensie (interfața serială SPI de mare viteză) care va avea și rolul de a distribui energia de la modulul de alimentare localizat în proximitatea plăcii de bază spre toate modulele de extensie (cu excepția modulului de actuatori care va avea propria alimentare din motive de eficientizare a consumului). Adăugarea / eliminarea anumitor module din microsistemul WIT va fi realizată după principii "plug-and-play", configurația curentă a elementelor WIT fiind recunoscută și executată de modulul de bază.

Cel puțin un microsistem WIT va avea un modul de extensie care îi va permite o conexiune la un calculator PC – BRAIN. Acest WIT "special" va fi punctul de acces al sistemului CORE-TX. Deși în sistemul CORE-TX este folosită comunicația fără fir, sunt posibile și alte modalități de comunicare:

- comunicare acustică (prin folosirea unui sonar pentru scopuri multiple, atât comunicare cât și localizarea vecinilor);
- comunicare vizuală: prin folosirea unor diode infraroșu;

Toate dispozitivele WIT pot fi programate prin interfața wireless folosind protocoale IAP (In-Application Programming).

### 7.3 Modelul BRAIN

Entitatea BRAIN (Background Robotic Activity Induction Node) este un sistem software de supraveghere care coordonează toate activitățile care au loc în mediul CORE-TX. Este proiectat pentru a asigura funcțiile descrise în paragrafele următoare.

Printre cele mai importante funcții se regăsește specificarea, analiza și dezvoltarea unor tipare de comportament particulare pentru întregul sistem colaborativ. Module speciale software au fost proiectate pentru a formaliza descrierea și dezvoltarea tiparelor de comportament din mediul CORE-TX. Aceste module software sunt bazate pe un interpretor de limbaj comportamental [Cioarga 2007a]. Acest interpretor, la rândul său, este bazat pe un limbaj formal folosit pentru a descrie tiparele de comportament ale dispozitivelor WIT și ale sistemului CORE-TX în general.

O altă funcție importantă al nivelului BRAIN este reconfigurarea / autoconfigurarea întregului mediu colaborativ prin tehnici de auto-descoperire și plug-and-play. Nu este necesară configurare pentru majoritatea schimbărilor din sistem. Fiecare dispozitiv WIT care intră în mediul CORE-TX este imediat recunoscut de sistemul BRAIN care execută schimbările necesare pentru întregul sistem iar noile tipare de comportament sunt încărcate în dispozitivul respectiv. Tehnica folosită pentru acest principiu de auto-descoperire este similară cu tehnica de "port-knocking": un punct terminal fix de comunicație este folosit pentru a asculta după noi conexiuni; această adresă este cunoscută de toate dispozitivele WIT. Un nou microsystem WIT încearcă să comunice cu entitatea BRAIN prin intermediul acestei adrese. Atunci când s-a stabilit un canal de comunicație, microsystemului WIT ii este atribuită de către BRAIN o nouă adresă prin care vor urma cele două entități să comunice, după care canalul de comunicație se închide. Toate încercările de comunicație ulterioare vor avea loc prin o nouă adresă stabilită.

Următoarea funcție a sistemului BRAIN este coordonarea generală și monitorizarea comportamentului întregului sistem prin tehnici de inducere de directive (inducere de fundal – "background induction") și prin interogarea stării tuturor elementelor componente. Sistemul BRAIN este direct responsabil de monitorizarea și supervizarea întregului mediu colaborativ; toate microsystemele WIT au sarcina de a raporta activitatea proprie sistemului BRAIN.

O altă funcție importantă a nivelului BRAIN este interfațarea cu operatorul sistemului, extragerea, corelarea și agregarea datelor primite din rețea, afișarea stării întregului sistem precum și tendințele de comportament ale acestuia. Sistemul BRAIN este folosit ca punct de intrare în sistemul CORE-TX. Orice comandă care este generată de un utilizator al sistemului CORE-TX trebuie să treacă prin entitatea BRAIN care va avea și un rol de dispecer la nivelul întregului sistem.

Sistemul BRAIN prezintă un modul specializat pentru interfața cu utilizatorul. Prin intermediul acestei interfețe un utilizator poate interacționa cu sistemul CORE-TX. Aceasta are o interfață grafică care transformă comenzile utilizatorului în directive BRAIN.

Entitatea BRAIN este implicată în cele mai mici detalii în funcționarea întregului sistem CORE-TX:

- Inducerea tiparelor de comportament emergent;
- Configurarea mediului CORE-TX;

- Monitorizarea mediului CORE-TX, atât a funcționării acestuia în ansamblu cât și a echipamentelor WIT individuale.
- Interfața cu utilizatorul;

Deși implicarea acestuia este extensivă, entitatea BRAIN nu interacționează cu „executarea” tiparelor de comportament emergent, limitându-se la inducerea acestor tipare în mediul CORE-TX și la supravegherea funcționării acestuia.

Existența entității BRAIN este necesară în contextul mediului CORE-TX ca fiind singurul punct de acces în sistemul CORE-TX atât pentru modificarea elementelor acestuia cât și pentru monitorizarea lor.

## 7.4 Modelul de comunicație

Nivelul de comunicație colaborativă asigură suportul fizic și protocoalele necesare pentru a interschimba mesaje și date între elementele sistemului CORE-TX. Pentru a implementa și dezvolta mediul fizic de comunicație precum și protocoalele de comunicație sunt luate în considerare caracteristicile impuse de modelul sistemului CORE-TX referitoare la mobilitate și autonomie. Astfel, este proiectată folosirea interfețelor radio / fără fir, concepte de interconectare, identificare, rutare „multihop”, rețele ad-hoc precum și dezvoltarea de protocoale simple de comunicație, care sunt modulare și flexibile și care permit interschimbarea de mesaje în mod colaborativ. Modelul de comunicație al sistemului CORE-TX se bazează pe principiile descrise în paragrafele următoare.

Se folosește comunicația fără fir sub forma standardului de comunicație radio Zigbee / IEEE 802.15.4 cu module X-bee PRO. Nu există necesitatea de contact direct între micro sistemele WIT; se folosește comunicație multi-hop, iar rutele sunt stabilite în mod dinamic și stocate pentru o posibilă folosire ulterioară. Deoarece micro sistemele WIT pot fi mobile, pot intra sau părăsi mediul CORE-TX în mod aleator, aceste rute sunt actualizate permanent și au o durată de viață limitată. Se poate folosi orice protocol de comunicație prin simpla actualizare a firmware-ului modulului de comunicație al micro sistemului WIT și prin instalarea modului corespunzător la nivelul entității BRAIN. Se preferă comunicația timp-real, deoarece întregul sistem CORE-TX interacționează direct cu mediul exterior, ceea ce necesită în majoritatea cazurilor reacții deosebit de rapide.

Pentru un sistem CORE-TX desfășurat într-un mediu de mari dimensiuni este posibil ca între anumite micro sisteme WIT să existe imposibilitatea stabilirii unui canal de comunicație. Atunci când nu sunt conexiuni, este posibil ca un micro sistem WIT să fie folosit pe post de releu de retransmisie. Astfel entitatea BRAIN poate rămâne conectată la întregul sistem de dispozitive WIT printr-un singur micro sistem suplimentar (cel mai apropiat, cel pentru care indicatorul RSSI, Radio Signal Strength Indicator, prezintă un maxim). Mesajele sunt apoi diseminate prin rețea până când ajung la destinație.

## 7.5 The Emergent Behavioral Modeling Language – eBML

Limbajul eBML [Cioarga 2007a, Cioarga 2007b] a fost conceput ca un limbaj interpretat care are la bază o colecție de macrocomenzi avansate și complexe scrise în diferite limbaje de programare [Mateas 2004].

eBML este bazată ușor pe UML (Unified Modeling Language). eBML este un limbaj formal, permițând o diversitate de concepte de la simple condiții booleene la executarea condiționată, bucle etc. Acesta are o reprezentare vizuală prin utilizarea principiilor de diagrame de stare și o reprezentare textuală. Reprezentările sunt interschimbabile; reprezentarea textuală este folosită, în principal, pentru modelarea comportamentală, în timp ce reprezentarea vizuală este utilizată ca un nivel de prezentare pentru reprezentarea textuală.

Pentru a fi mai bine înțelese principiile de funcționare ale eBML, se prezintă în continuare un simplu algoritm emergent destinat ocolirii obstacolelor. Reprezentarea vizuală (Fig. 26) conține o diagramă de stare simplă: entitatea rămâne în starea inițială până la întâlnirea unui obstacol, moment în care entitatea evită obstacolul și revine la starea inițială.

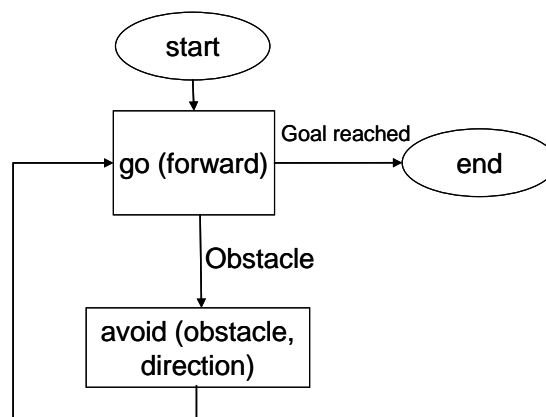


Fig. 26. Reprezentarea vizuală a eBML [Cioarga 2007b]

Secvența de cod 1. Reprezentarea textuală a eBML [Cioarga 2007b]

```

behavior is
go (forward)
  if obstacle
avoid (obstacle, random(left,right))
end if
end behavior
  
```

Un interpretor a fost special conceput pentru eBML. Acesta este capabil de a interpreta un fișier text care conține reprezentarea textuală (Secvența de cod 1) și de a genera codul pentru o țintă hardware specifică. Acest lucru îi dă posibilitatea unui utilizator să poată introduce algoritmi de comportament emergenți într-un fișier text utilizând eBML și apoi compila fișierul text într-un cod executabil care să poată fi încărcat pe un anumită platformă cu microcontroler / microprocesor.

Limbajul de modelare eBML poate fi aplicat pentru a dezvolta aplicații în viața reală pentru mediile colaborative. În dezvoltarea unui sistem colaborativ sunt implicate diferite etape de proiectare a sistemului colaborativ, de la simulare la fabricarea prototipurilor. Toate aceste etape sunt dificile, implică costuri ridicate și procese consumatoare de timp. eBML a fost conceput pentru a aborda unele dintre problemele care apar în procesul de dezvoltare. Prin urmare, eBML este folosit ca:

- un instrument de programare pentru sistemul colaborativ,

- un limbaj de modelare, care este în măsură să descrie complet comportamentul și funcționalitatea unui sistem colaborativ (un simplu interpretor / compilator poate simula comportamentul pentru orice entitate dată pe un calculator PC).

eBML este similar cu UML și, prin urmare, prezintă constructe similare. Fiecare script eBML trebuie să înceapă cu cuvintele cheie "*behavior is*" și trebuie să se termine cu "*behavior end*". De asemenea, diagrama vizuală eBML trebuie să înceapă și să se încheie cu entitățile *START* și *END*.

Construcția de bază eBML este reprezentată de o "*activitate*". O activitate reprezintă un apel de funcție, un pas în logica algoritmului, un algoritm complet sau o operațiune.

Activitatea de tip "*gaură neagră*" ("*black hole*") reprezintă un tip special de activitate. Există o mulțime de tranziții care conduc către aceasta, dar nici o tranziție care să pornească de la aceasta. De asemenea, activitatea de tip "*miracol*" este o activitate care prezintă numai tranziții care pornesc de la aceasta și nici o tranziție care să se finalizeze în aceasta.

O altă construcție eBML este punctul de decizie; aceasta reprezintă un punct de diviziune în fluxul normal de comportament. Aceste puncte de decizie nu au nici o etichetă atașată și, prin urmare, ar trebui să reflecte activitatea anterioară sau unele condiții precum au deciziile în diagramele normale. Multiple tranziții cu etichete asociate pot porni din fiecare punct de decizie.

Tranzițiile apar între activități sau între activități și puncte de decizie. Fiecare tranziție este reprezentată printr-o săgeată care indică sensul de curgere dintr-o construcție la alta, în reprezentarea vizuală a eBML. Pentru reprezentarea textuală eBML, tranzițiile nu sunt reprezentate, dar sunt prezente implicit prin înlănțuirea normală a comenzilor și instrucțiunilor.

Construcția de tip "*element de gardă*" reprezintă o condiție asociată unei tranziții. Tranziția, prin urmare, poate fi trecută numai atunci când această condiție este adevărată.

În ceea ce privește construcțiile de tip "*element de gardă*", se aplică următoarele constrângeri:

- Fiecare tranziție care conduce departe de la un punct de decizie trebuie să aibă un "*element de gardă*" asociată;
- Toate construcțiile asociate cu tranzițiile care duc de la punctele de decizie nu trebuie să se suprapună (de exemplu, dacă o tranziție are asociat "*elementul de gardă*"  $T \leq 1$  și  $T \geq 1$  atunci când  $T = 1$  există o confuzie legată de tranziția ce ar trebui să fie traversată);
- Toate construcțiile asociate cu tranzițiile care duc de la punctele de decizie trebuie să conțină seturi complete de date (de exemplu, dacă o tranziție are asociat "*elementul de gardă*"  $T < 1$  și  $T > 1$  atunci când  $T = 1$  există o confuzie legată de tranziția ce ar trebui să fie traversată);
- Anumite tranziții nu au asociată nici un "*element de gardă*" ceea ce presupune traversarea ei întotdeauna, absența acesteia fiind similară cu prezența unui "*element de gardă*" de tip *true* (adevărat).

eBML este un limbaj de modelare de nivel înalt, care oferă utilizatorului posibilități de proiectare, simulare și programare a unui colectiv de roboți. eBML este capabil să descrie comportamente simple, care sunt extrem de eficiente în mediile colaborative de mari dimensiuni.

Limbajul eBML este un limbaj formal care are următoarea specificație:

## Secvența de cod 2. Specificația limbajului eBML

```

Identifier: IdentifChrs
IdentifChrs: Letter | IdentifChrs LetterOrDigit
Letter: [a-z, A-Z]
LetterOrDigit: [a-z, A-Z, 0-9]
Keyword: [behavior, is, end, if, while, for]
Literal: IntegerLiteral | FloatingPointLiteral | BooleanLiteral |
CharacterLiteral | StringLiteral | NullLiteral
IntegerLiteral: DecimalLiteral | HexLiteral
DecimalLiteral: [1-9[0-9]*
HexLiteral: 0[xX[0-9a-fA-F]*
BooleanLiteral: true | false
NullLiteral: null
eBMLInterpretationUnit: behavior is InstructionList end behavior
InstructionList: Instruction | InstructionList Instruction
Instruction: SimpleInstruction | IfInstruction | WhileInstruction |
ForInstruction
SimpleInstruction: Identifier(ParameterList)
ParameterList: Parameter | ParameterList, Parameter
IfInstruction: if (Expression) InstructionList
WhileInstruction: while (Expression) InstructionList
ForInstruction: for(ForInit, ForExpression, ForUpdate) InstructionList
ForInit: "" | InstructionList
ForExpression: "" | Expression
ForUpdate: "" | InstructionList

```

**7.5.1 Simulatorul de comportament emergent**

În scopul evaluării și testării modelelor definite a fost proiectat și implementat un simulator de comportament emergent numit eBMS (Emergent Behavior Model Simulator).

Simulatorul este o aplicație orientată pe obiecte, scrisă în C++, folosind mediul de dezvoltare Microsoft Visual Studio 2008. Acesta este compus din două module principale: un modul de configurare și un modul de simulare.

Modulul de configurare este reprezentat, în principal, de o interfață grafică destinată configurării unei colecții de obiecte de tip Entitate fiecare având o serie de parametri: poziția de start, țelul, setul de senzori (senzori de atingere, senzori de poziționare), dimensiuni (înălțime, lungime, lățime) precum și a comportamentului pe care aceste entități trebuie să îl manifeste.

Comportamentul este descris cu ajutorul limbajului eBML. Fiecare entitate dispune de o instanță a interpretorului de limbaj eBML. Acesta folosește ca intrare un fișier de descriere a comportamentului pe care, mai apoi, îl transformă într-un cod executabil specific procesorului țintă. Interpretorul este structurat într-o manieră modulară folosind fișiere de XML pentru configurarea subsetului limbajului eBML folosit precum și a regulilor de transformare în limbajele necesare.

Modulul de interpretare și traducere a fișierelor de intrare eBML este format dintr-un sub-modul sintactic și un sub-modul de translație.

Modulul sintactic conține un analizor lexical dedicat care determină arborele lexical specific fișierului de intrare încărcat. Pentru fiecare item, se caută apoi descrierea corespunzătoare în fișierul sintactic XML. Acesta are următoarea structură:

- Descrierea comezilor:



## Secvența de cod 3. Formatul unei comenzi

```
<comanda name="nume_comanda">
  <com_param name="nume_parametru_1"/>
  <com_param name="nume_parametru_2"/>
  <com_param name="nume_parametru_3"/>
  ...
  <com_param name="nume_parametru_n"/>
</comanda>
```

- Pentru fiecare parametru se definește lista de valori posibile:

## Secvența de cod 4. Formatul unui parametru

```
<opt name = "nume_parametru_i">
  <val value = "valoare_posibila_1"/>
  <val value = "valoare_posibila_2"/>
  <val value = "valoare_posibila_3"/>
  ...
  <val value = "valoare_posibila_m"/>
</opt>
```

- Instrucțiunile posibile:

## Secvența de cod 5. Formatul unei instrucțiuni

```
<instrucțiune name="nume_instrucțiune">
  <!-- lista parametri posibili -->
</instrucțiune>
```

În continuare se prezintă un exemplu de fișier XML sintactic.

## Secvența de cod 6. Fișier XML sintactic

```
<configuration>
  <comanda name = "mergi">
    <com_param name = "directie"/>
  </comanda>

  <opt name = "directie">
    <val value = "inainte"/>
    <val value = "inapoi"/>
    <val value = "stanga"/>
    <val value = "dreapta"/>
  </opt>

  <instrucțiune name = "if">
    <instr_param name = "conditie"/>
    <instr_comanda name = "mergi"/>
  </instrucțiune>

  <instr_opt name = "conditie">
    <instr_opt_val value = "obstacol"/>
  </instr_opt>
</configuration>
```

Fișierul XML de translație trebuie să conțină câte o intrare pentru fiecare atom lexical detectat de analizorul lexical al modului sintactic. Fiecare intrare din acest fișier are următoarea formă:

Secvența de cod 7. Formatul fișierului XML de translație

```
<replace atom="atom_detectat" type="text" with="text_de_inlocuit" />
<replace atom="atom" type="hex" with="text_de_inlocuit_hex" />
```

Secvența de cod 8. Exemplu de fișier XML de translație

```
<configuration>
  <replace atom="mergi" with="abs" />
  <replace atom="inainte" with="i++" />
  <replace atom="inapoi" with="i--" />
  <replace atom="stanga" with="i<<2" />
  <replace atom="dreapta" with="i>>2" />
  <replace atom="if" with="if" />
  <replace atom="obstacol" with="!i" />
</configuration>
```

Modulul de simulare primește de la modulul de configurare lista de obiecte entitate pe care apoi le simulează corespunzător într-un mediu a cărui parametri (dimensiuni, obstacole, poziții inițiale etc) sunt descriși printr-o hartă.

## 7.6 Concluzii

Utilizarea unor echipe de roboți pentru efectuarea unor activități complexe cum ar fi explorarea, monitorizarea și operarea în medii necunoscute sau periculoase pentru om prezintă o serie de avantaje esențiale: substituirea intervenției umane, autonomie de operare pe termen lung, precizie foarte bună și constantă, creșterea eficienței prin paralelizare, creșterea fiabilității sistemului în ansamblu prin redundanță și distribuirea resurselor. Pe de alta parte, utilizarea colectivelor robotice implică și o serie de probleme, în special legate de controlul roboților, comunicația între roboți, precum și de sincronizarea și distribuirea eficientă a sarcinilor în cadrul echipei de roboți. Aplicarea comportamentului emergent rezolvă problemele sistemelor asociative, în special în cazul în care aceste sisteme au un număr ridicat de entități componente.

Pentru studiul modelelor existente pentru formalizarea tiparelor de comportament colaborativ s-a elaborat un mediu colaborativ robotizat, o platformă integrată, care conține un simulator de comportament emergent. Comportamentul simulat este descris cu ajutorul unui limbaj comportamental special, dezvoltându-se un interpretor pentru acesta.

Pentru această platformă integrată:

- S-a dezvoltat și validat limbajului formal eBML (Emergent Behavioral Modeling Language), conceput special pentru reprezentarea comportamentului emergent;
- S-a realizat integrarea limbajului eBML și a fundamentelor matematice de reprezentare a interacțiunilor, pentru dezvoltarea și validarea unui model coerent de formalizare a tiparelor de comportament colaborativ/asociativ într-un simulator de comportament emergent;
- S-a implementat un sistem de control integrat pentru validarea și analiza performanțelor modelului dezvoltat în cadrul programului de doctorat.

În capitolul următor se prezintă rezultatele validării modelului de comportament emergent prin simularea acestuia în eBMS.

## 8. SIMULAREA MODELULUI MATEMATIC FOLOSIND EBMS

### 8.1 Introducere

Acest capitol descrie evaluarea și validarea setului integrat de modele pentru reprezentarea tiparelor de control ale comportamentelor emergente în medii robotizate prin simularea acestuia în eBMS.

Modelele au fost validate cu ajutorul simulatorului de comportament emergent eBMS și testate folosind platforma prototip CORE-TX. Simulările create folosesc ca și date de intrare seturi cu număr variabil de WIT-uri, fiecare WIT putând urma câte un tipar de comportament emergent specific.

Pentru fiecare simulare s-au generat traseele WIT-urilor simulate și s-a calculat rata de succes și numărul de pași necesari pentru atingerea țelului.

### 8.2 Cadrul general pentru simulările făcute

Pornind de la premisa ocularii de obstacole care stă la baza modelului matematic pentru comportament emergent prezentat în capitolul 6, s-au proiectat o serie de simulări pentru evaluarea și validarea modelului.

Există trei seturi majore de simulări folosite la validarea modelului, fiecare set înglobând posibilitățile setului anterior:

- simulări ale tiparelor de comportament emergent de tipul „evitare de obstacol”;
- simulări ale tiparelor de comportament emergent de tipul „explorare emergentă”;
- simulări ale tiparelor de comportament emergent de tipul „adunare de resurse”.

Fiecare set de simulări:

- implică tipare de comportament emergent similare dar ușor diferite pentru a valida modelul în condiții cât mai apropiate de cele reale;
- folosesc funcții țel,  $\Gamma$ , cu parametri diferiți;
- lucrează cu număr variabil de entități simulate.

Pentru setul de simulări care folosesc tipare de comportament emergent pentru evitarea obstacolelor, fiecare WIT are asociată o funcție  $\Gamma$  de tipul *goto*( $x, y$ ), va încerca să ajungă la destinație, prin mișcarea spre această direcție bună. În cazul în care se întâlnește un obstacol, entitatea simulată va încerca să-l evite și, imediat, să continue pe direcția obiectivului. Pentru acest set de simulări, un WIT poate să eșueze doar în cazul în care depășește numărul maxim permis de pași (setat la 1000 de pași) înainte de a ajunge la destinație.

Pentru setul de simulări corespunzătoare explorării emergente, harta nu conține nici un fel de resursă, iar fiecare WIT are asociată o funcție  $\Gamma$  de tipul *caută*. Entitatea va începe să exploreze medul în mod aleatoriu. Aceasta se va opri în cazul în care s-a parcurs întregul mediu sau s-a depășit numărul maxim de pași. În cazul în care s-a depășit numărul de pași, WIT-ul se află în situația de eșuare doar dacă procentul de mediu acoperit scade sub o anumită valoare.

Pe hartă obstacolele sunt reprezentate cu culoarea neagră, iar punctul de plecare al WIT-ului este reprezentat cu culoarea albastră. Prezența WIT-ului pe hartă precum și traseul acestuia este reprezentat pe harta prin culoarea verde.

Pentru setul de simulări corespunzătoare căutării de resurse, fiecare WIT are asociată o funcție  $\Gamma$  combinată din cele două prezentate în paragrafele precedente. Entitatea va începe să exploreze mediul în mod aleatoriu. Aceasta se va opri în cazul în care s-a parcurs întregul mediu sau s-a adunat numărul setat de resurse sau s-a depășit numărul maxim de pași. În cazul în care s-a depășit numărul de pași, WIT-ul se află în situația de eșuare doar dacă procentul de mediu acoperit și numărul de resurse adunate scad sub anumite valori.

Similar cu setul precedent de simulări, obstacolele sunt reprezentate pe hartă cu culoarea neagră, iar punctul de plecare al WIT-ului este reprezentat cu culoarea albastră. Prezența WIT-ului pe hartă precum și traseul acestuia este reprezentat pe harta prin culoarea verde. Resursele sunt reprezentate prin culoarea roșie.

## 8.3 Ocolirea de obstacole

### 8.3.1 Cadrul de organizare a simulării

În Fig. 27 se prezintă mediul simulat pentru ocolirea de obstacole.



Fig. 27. Mediul simulat pentru ocolirea de obstacole

Simularea a folosit un set variabil de WIT-uri, între 1 și 9 folosind următoarele perechi de coordonate pentru puncte de start: (10, 10), (10, 20), (10, 30), (10, 40), (10, 50), (10, 60), (10, 70), (10, 80), (10, 90).

### 8.3.2 Rezultatele obținute

Tabelul 1 prezintă rezultatele simulării ocolirii de obstacole.

Tabelul 1. Rezultatele simulărilor pentru ocolirea de obstacole

| Nr. WIT-uri | Pași | Rezultat |
|-------------|------|----------|
| 1           | 263  | succes   |
| 2           | 273  | succes   |
| 3           | 263  | succes   |
| 4           | 264  | succes   |
| 5           | 263  | succes   |
| 6           | 263  | succes   |
| 7           | 263  | succes   |
| 8           | 264  | succes   |
| 9           | 252  | succes   |

În continuare se prezintă o serie de rezultate vizuale ale setului de simulări descris în acest paragraf. Fig. 28 conține harta finală după parcurgerea algoritmului de către fiecare WIT pentru 1, 2, 5 și, respectiv, 9 entități simulate.



(a)



(b)



(c)



(d)

Fig. 28. Rezultatele simulărilor pentru 1 WIT (a), 2 WIT-uri (b), 5 WIT-uri (c) și 9 WIT-uri (d)

### 8.3.3 Concluzii

Pe parcursul celor 9 simulări rulate toate WIT-urile au ajuns la final.

În cazul simulării cu 2 WIT-uri numărul de pași a crescut deoarece cele 2 entități ajung la primul obstacol în același timp, și, datorită condițiilor de declanșare a comportamentului emergent menționate pentru modelul matematic general, fiecare îl ocolesc folosind direcții diferite ceea ce conduce la cazul defavorabil prezentat în Fig. 29.

Pe de altă parte, cazul folosirii a 5 WIT-uri este mai avantajos din punct de vedere al vitezei de atingere a destinației deoarece numărul mai ridicat de WIT-uri conduce la o evitare eficientă și mai rapidă a obstacolelor întâlnite.

Din motive similare, simularea tuturor celor 9 WIT-uri este cea mai eficientă.

## 8.4 Adunarea de resurse

### 8.4.1 Cadru de organizare a simulării

Mediul folosit pentru desfășurarea simulării este identic cu cel din Fig. 27. Resursele sunt reprezentate pe hartă cu culoarea verde.

Simularea a folosit un set variabil de WIT-uri, între 1 și 16 folosind următoarele perechi de coordonate pentru puncte de start: (10, 10), (10, 20), (10, 30), (10, 40), (10, 50), (10, 60), (10, 70), (10, 80), (10, 90), (10, 100), (10, 110), (10, 120), (10, 130), (10, 140), (10, 150), (10, 160).

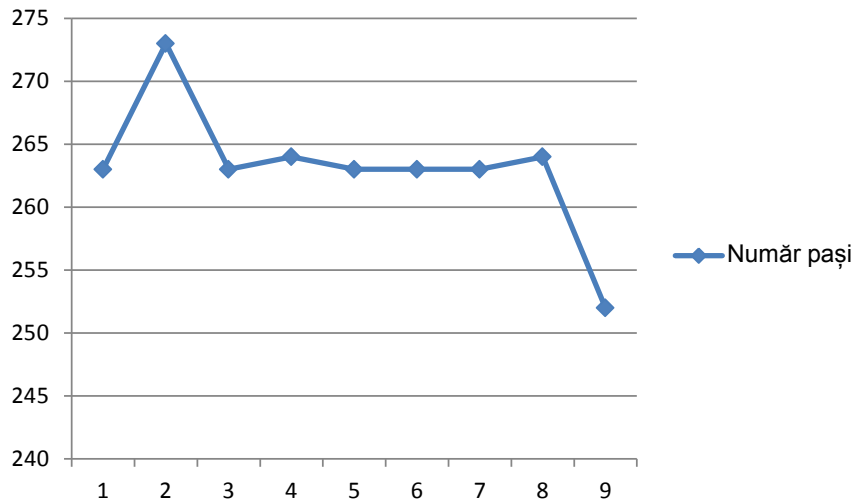


Fig. 29. Numărul de pași corespunzător entităților simulate care au încheiat cu succes simularea

#### 8.4.2 Rezultatele obținute

Rezultatele simulării adunării de resurse sunt prezentate în Tabelul 1

Tabelul 2. Rezultatele simulărilor pentru adunarea de resurse

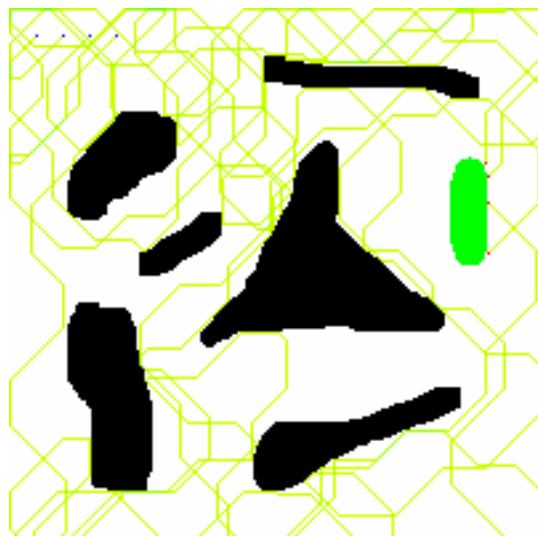
| Nr.Wit-uri | Poziția primei resurse | Pași | Rezultat |
|------------|------------------------|------|----------|
| 1          | (62, 179)              | 749  | succes   |
| 2          | (72, 179)              | 479  | succes   |
| 3          | (59, 178)              | 642  | succes   |
| 4          | (91, 179)              | 2259 | succes   |
| 5          | (75, 166)              | 849  | succes   |
| 6          | (78, 179)              | 890  | succes   |
| 7          | (58, 176)              | 301  | succes   |
| 8          | (57, 176)              | 363  | succes   |
| 9          | -                      | 3000 | eșec     |
| 10         | (86, 179)              | 1790 | succes   |
| 11         | (96, 173)              | 1677 | succes   |
| 12         | (59, 177)              | 769  | succes   |
| 13         | (83, 166)              | 886  | succes   |
| 14         | -                      | 3000 | eșec     |
| 15         | (91, 167)              | 1207 | succes   |
| 16         | (88, 179)              | 1863 | succes   |

În continuare se prezintă o serie de rezultate vizuale ale setului de simulări descris în acest paragraf. Fig. 30 conține harta finală după parcurgerea algoritmului de către fiecare WIT pentru 1, 2, 5 și, respectiv, 9 entități simulate.

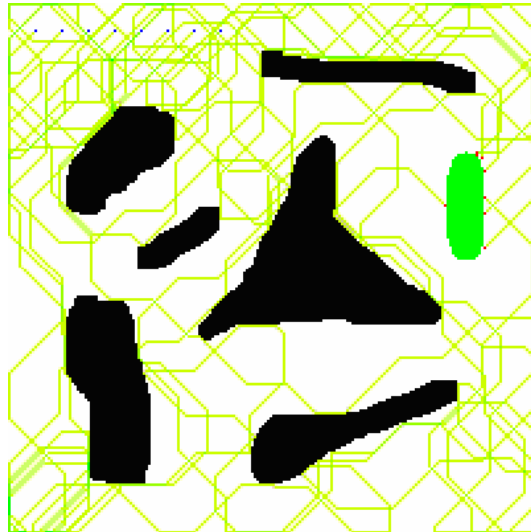




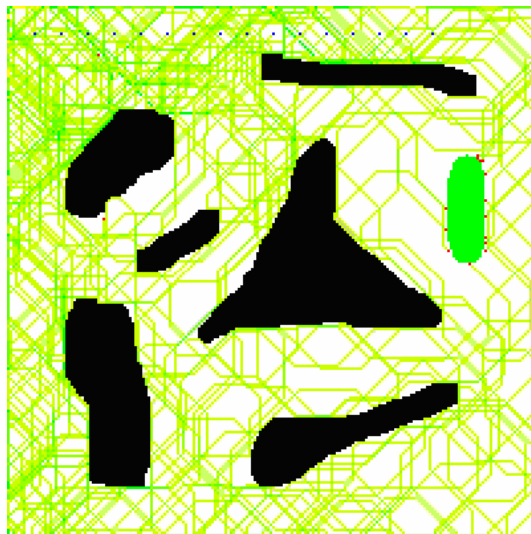
(a)



(b)



(c)



(d)

Fig. 30. Rezultatele simulărilor pentru 1 WIT (a), 4 WIT-uri (b), 8 WIT-uri (c) și 16 WIT-uri (d)

### 8.4.3 Concluzii

Pe parcursul celor 16 simulări rulate o serie de WIT-uri nu au ajuns la final datorită gradului de aglomerare.

Tabelul 3. Ratele de succes pentru experimentele legate de adunarea de resurse

| Nr. WIT-uri | Rata de succes | Rata de eșec |
|-------------|----------------|--------------|
| 1           | 100%           | 0%           |
| 2           | 100%           | 0%           |
| 3           | 100%           | 0%           |
| 4           | 100%           | 0%           |
| 5           | 100%           | 0%           |
| 6           | 100%           | 0%           |
| 7           | 100%           | 0%           |
| 8           | 100%           | 0%           |
| 9           | 88.9%          | 11.1%        |
| 10          | 90%            | 10%          |
| 11          | 90.9%          | 9.1%         |
| 12          | 91.7%          | 8.3%         |
| 13          | 92.3%          | 7.7%         |
| 14          | 85.7%          | 14.3%        |
| 15          | 86.7%          | 13.3%        |
| 16          | 87.5%          | 12.5%        |

Tabelul 3 prezintă ratele de succes corespunzătoare celor 16 simulări rulate. Acestea sunt ideale în cazul folosirii unui număr de entități mai mic decât 10, similar cu rezultatele obținute în setul precedent de simulări. Pentru numere mai mari decât 10, declanșarea comportamentului emergent este strangulată de creșterea numărului de interacțiuni posibile între WIT-uri.

În cazul simulării cu 4 WIT-uri numărul de pași a crescut deoarece entitățile ajung la obstacole în același timp, și, datorită condițiilor de declanșare a comportamentului emergent menționate pentru modelul matematic general, fiecare îl ocolesc folosind direcții diferite ceea ce conduce la cazul defavorabil prezentat în Fig. 31.

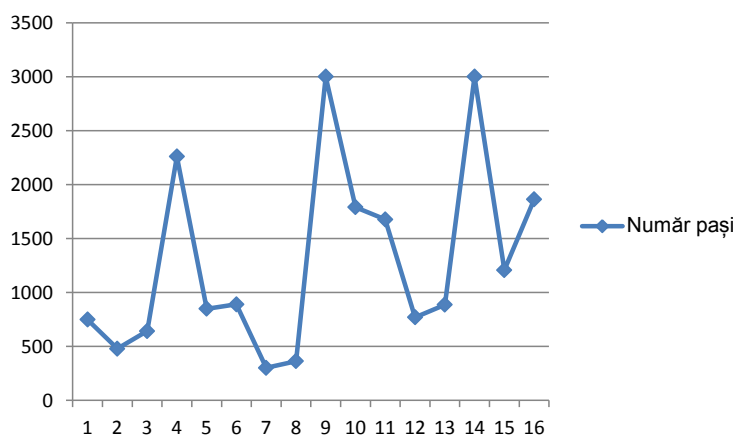


Fig. 31. Numărul de pași corespunzător entităților simulate care au încheiat cu succes simularea

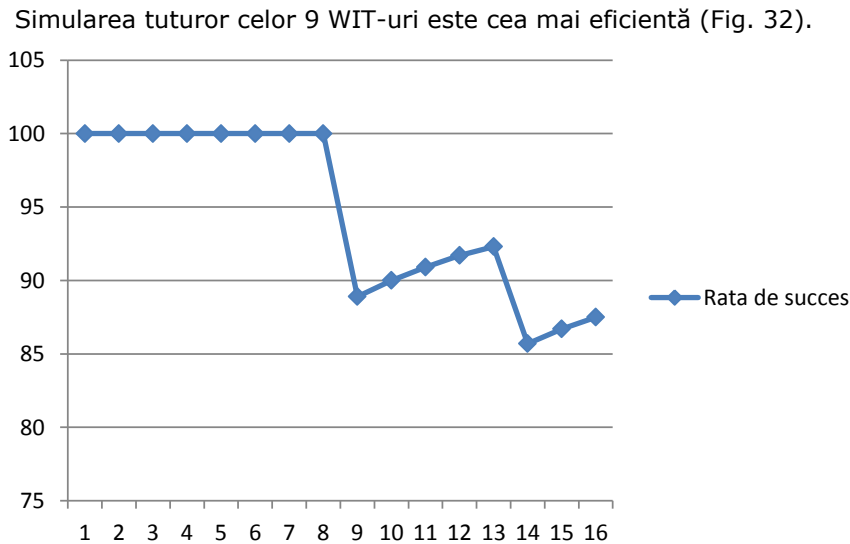


Fig. 32. Rata de succes

## 8.5 Explorare emergentă

### 8.5.1 Cadrul de organizare a simulării

Simularea explorării emergente a fost executată în 2 pași:

- Simularea explorării unui mediu necunoscut, fără adunare de resurse și obstacole;
- Simularea explorării unui mediu necunoscut, cu ocolire de obstacole.

Explorarea unui mediu necunoscut pornește de la o hartă inițial goală care este, mai apoi, completată cu obstacolele întâlnite precum și cu resursele identificate.

În cazul explorării fără adunare de resurse singura metrică de succes care este folosită este gradul de acoperire al explorării (cât din mediul necunoscut a fost explorat). Pentru simulările efectuate, aceasta metrică este strâns corelată cu numărul maxim de pași configurați pentru simulare.

În cazul explorării cu adunare de resurse, simularea a folosit un set variabil de WIT-uri, între 1 și 10 folosind următoarele perechi de coordonate pentru puncte de start: (10, 10), (10, 20), (10, 30), (10, 40), (10, 50), (10, 60), (10, 70), (10, 80), (10, 90), (10, 100).

### 8.5.2 Rezultatele obținute

#### 8.5.2.1 Fără adunare de resurse

Pentru 1 singură entitate și 3000 de pași, simularea se încheie cu eșec datorită gradului insuficient de acoperire a mediului (Fig. 33).

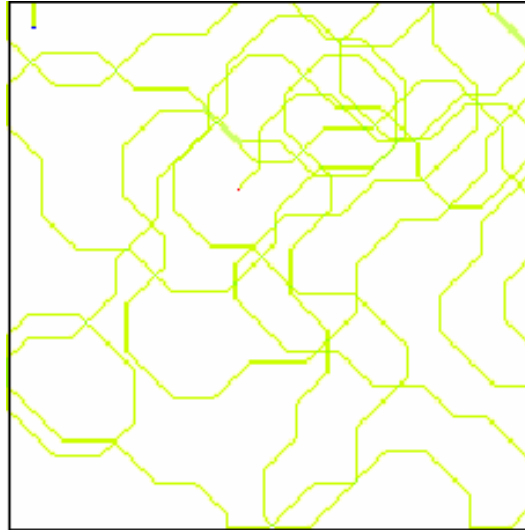


Fig. 33. Caz de eșec la explorare emergentă: acoperire insuficientă a mediului

Pentru un număr suficient de mare de pași entitățile reușesc să acopere întregul mediu cu un grad satisfăcător (peste 95 %), caz prezentat în Fig. 34.

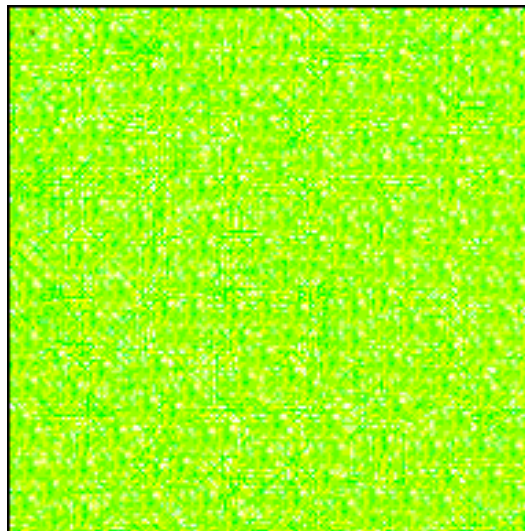


Fig. 34. Caz de succes la explorare emergentă: acoperire satisfăcătoare a mediului

#### **8.5.2.2 Cu adunare de resurse**

În cazul adunării de resurse, entitățile implicate în explorarea emergentă a mediului necunoscut construiesc o hartă similară cu cea din Fig. 27. Rezultatele simulării adunării de resurse sunt prezentate în Tabelul 4.

Tabelul 4. Ratele de succes pentru experimentele legate de adunarea de resurse

| Nr.Wit-uri | Poziția primei resurse | Pași | Rezultat |
|------------|------------------------|------|----------|
| 1          | (93, 178)              | 1647 | succes   |
| 2          | (91, 179)              | 372  | succes   |
| 3          | (84, 179)              | 1770 | succes   |
| 4          | (70, 179)              | 273  | succes   |
| 5          | (91, 171)              | 149  | succes   |
| 6          | (76, 179)              | 157  | succes   |
| 7          | (74, 179)              | 554  | succes   |
| 8          | -                      | 3000 | eșec     |
| 9          | -                      | 3000 | eșec     |
| 10         | -                      | 3000 | eșec     |

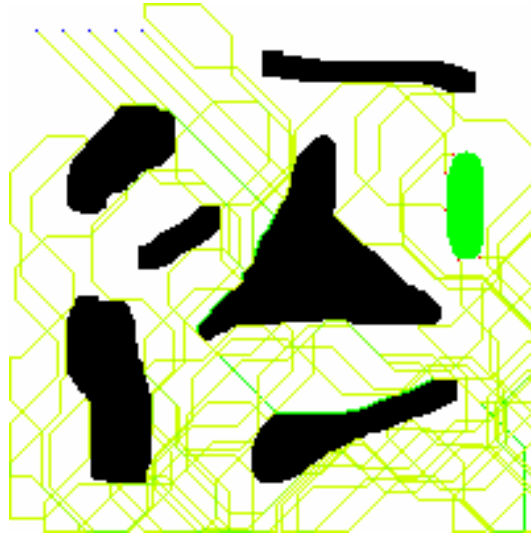
În continuare se prezintă o serie de rezultate vizuale ale setului de simulări descris în acest paragraf. Fig. 35 conține harta finală după parcurgerea algoritmului de către fiecare WIT pentru 1, 2, 5 și, respectiv, 10 entități simulate.



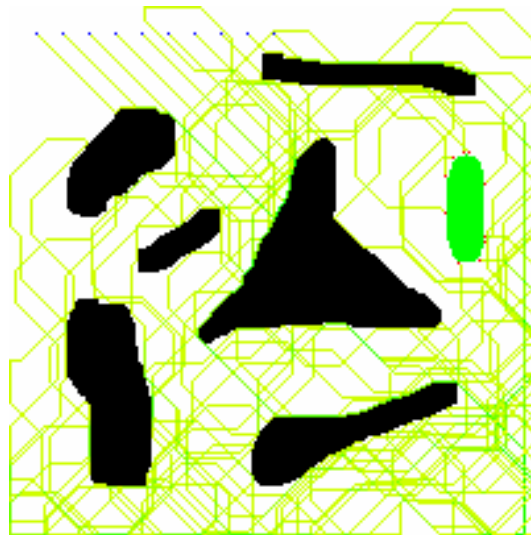
(a)



(b)



(c)



(d)

Fig. 35. Rezultatele simulărilor pentru 1 WIT (a), 2 WIT-uri (b), 5 WIT-uri (c) și 10 WIT-uri (d)

### 8.5.3 Concluzii

#### 8.5.3.1 Fără adunare de resurse

În cazul explorării emergente fără adunare de resurse, creșterea numărului de entități conduce la o reducere semnificativă a timpului de acoperire a mediului necunoscut, însă datorită algoritmului prezentat în capitolul 6 și a funcției  $f$  folosite declanșarea comportamentului emergent determină o parcurgere neuniformă a mediului.

#### 8.5.3.2 Cu adunare de resurse

Pe parcursul celor 10 simulări rulate o serie de WIT-uri nu au ajuns la final datorită gradului de aglomerare.

Tabelul 5. Ratele de succes pentru experimentele legate de adunarea de resurse

| Nr. WIT-uri | Rata de succes | Rata de eșec |
|-------------|----------------|--------------|
| 1           | 100%           | 0%           |
| 2           | 100%           | 0%           |
| 3           | 100%           | 0%           |
| 4           | 100%           | 0%           |
| 5           | 100%           | 0%           |
| 6           | 100%           | 0%           |
| 7           | 100%           | 0%           |
| 8           | 87.5%          | 12.5%        |
| 9           | 88.8%          | 11.2%        |
| 10          | 89%            | 11%          |

Tabelul 5 prezintă ratele de succes corespunzătoare celor 10 simulări rulate. Acestea sunt ideale în cazul folosirii unui număr de entități mai mic decât 8. Pentru entități în număr mai mare, numere mai mari decât 7, declanșarea comportamentului emergent este strangulată de creșterea numărului de interacțiuni posibile între WIT-uri.

Fig. 36 conține rata de succes în cazul explorării emergente cu adunare de resurse, succesul fiind reprezentat prin explorarea completă a mediului și colectarea simultană a tuturor resurselor.



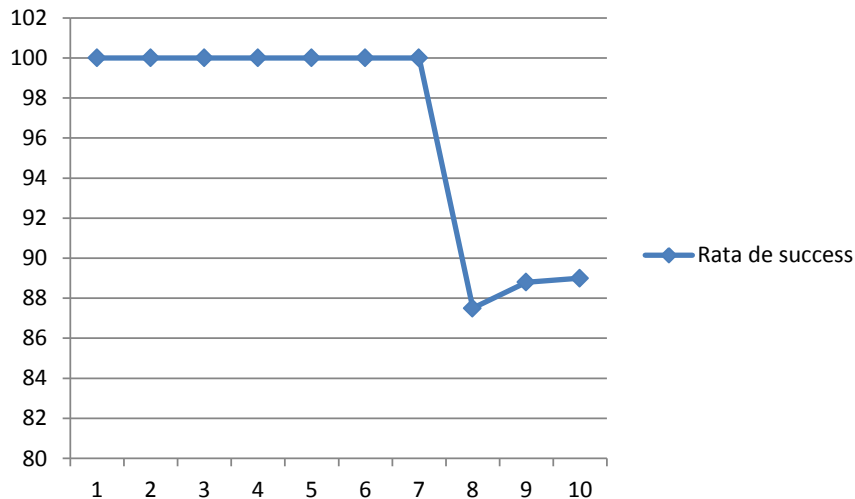


Fig. 36. Rata de succes

## 8.6 Concluzii

Modelul de comportament emergent a fost simulat cu succes folosind eBMS, tiparele de comportament emergent ale fiecărei simulări fiind descrise prin eBML.

Au fost rulate seturi complete de simulări folosind tipare de comportament emergent specifice ocolirii de obstacole, colectării de resurse și explorării emergente; fiecare set de simulări este variat prin modificare mediului în care WIT-ul se mișcă, prin modificarea incrementală a numărului de WIT-uri, prin modificarea poziției și a numărului de resurse.

Similar cu rezultatele validării formale a modelului de comportament emergent prezentate în capitolul precedent, analiza atentă a rezultatelor simulărilor a condus la determinarea unui interval probabilistic de declanșare a comportamentului emergent și, respectiv, de atingere completă a țelului propus. Acest interval de dimensiuni ale colectivului robotizat folosit în experiment este determinat pentru fiecare tip de tipar de comportament emergent în parte.

În cazul ocolirii de obstacole, un colectiv robotic își poate îndeplini sarcina dacă conține între 1 și 9 roboți în condițiile descrise în primele paragrafe ale acestui capitol. Folosirea unui număr între 3 și 8 roboți conduce la rezultate similare pentru viteza de atingere a țelului ceea ce conduce la concluzia folosirii doar a 3 roboți pentru a reduce costurile. În cazul în care se dorește atingerea țelului într-un interval de timp minim, se folosesc 9 roboți.

În cazul colectării de resurse într-un mediu cunoscut, este recomandată folosirea unui număr maxim de 8 WIT-uri pentru îndeplinirea completă a sarcinii. Un număr mai mare de WIT-uri conduce la strangularea comportamentului emergent sau chiar la imposibilitatea declanșării acestuia din cauza numărului mărit de interacțiuni posibile între roboți.

Similar, în cazul explorării emergente, un număr maxim de 6 roboți este recomandat pentru îndeplinirea sarcinii, mai mulți roboți determinând strangularea comportamentului emergent.

Rezultatele simulării sunt consecvente cu rezultatele validării formale prezentate în capitolul anterior precum și cu valorile teoretice determinate prin aplicarea directă a modelului matematic, ceea ce conduce la o validare suplimentară a modelului de comportament emergent.

Capitolul următor va prezenta aplicarea modelului de comportament emergent într-o aplicație reală, folosind tipare de comportament emergent preluate din studiul coloniilor de furnici folosind un colectiv robotizat format din roboți LEGO Mindstorms NXT.

## 9. IMPLEMENTAREA ȘI TESTAREA MIȘCĂRII ÎN COLECTIVE ROBOTIZATE BAZATĂ PE STUDIUL COLONIILOR DE FURNICI

### 9.1 Introducere

Modelul de comportament emergent prezentat în capitolul 5 a fost validat formal precum și printr-o serie de simulări în capitolele precedente. În capitolele următoare se prezintă aplicarea modelului pe colective robotice reale, formate din roboți LEGO Mindstorms NXT modificați în mod special pentru implementarea și testarea modelului.

În capitolul de față se prezintă implementarea și testarea modelului de comportament emergent folosind tipare de comportament emergent preluate din studiul mișcării în coloniile de furnici.

În capitol următor se prezintă implementarea și testarea modelului de comportament emergent folosind tipare de comportament emergent preluate din studiul mișcării în bancurile de pești.

În acest capitol se introduce un nou tipar de comportament emergent, *PATRULARE*, împreună cu descrierea modelului matematic implicat. Totodată se prezintă modalitatea de control a mișcării roboților LEGO și se analizează o modalitate de explorare emergentă a mediului prin prisma detecției obstacolelor și identificării resurselor prin folosirea doar a senzorilor cu care este echipat un robot LEGO Mindstorms NXT (senzor de culoare pentru discriminarea între obstacole și resurse și senzor ultrasonic pentru identificarea obstacolelor și a resurselor).

### 9.2 Tiparul de comportament emergent de tip PATRULARE

Tiparul de comportament emergent care rezultă din proprietatea de a avea o destinație la care se revine periodic după explorarea parțială a unui mediu necunoscut poartă denumirea de "*patrulare*". Comportamentul de "*patrulare*" pornește de la situația de a parcurge un mediu necunoscut prin modificarea aleatoare a poziției și a direcției; fiecare entitate este supusă unor stimuli specifici care o atrag către casă și o resping în direcții opuse față de entitățile foarte apropiate; acest regim de mișcare se stabilizează repede într-un model previzibil, modelat în continuare.

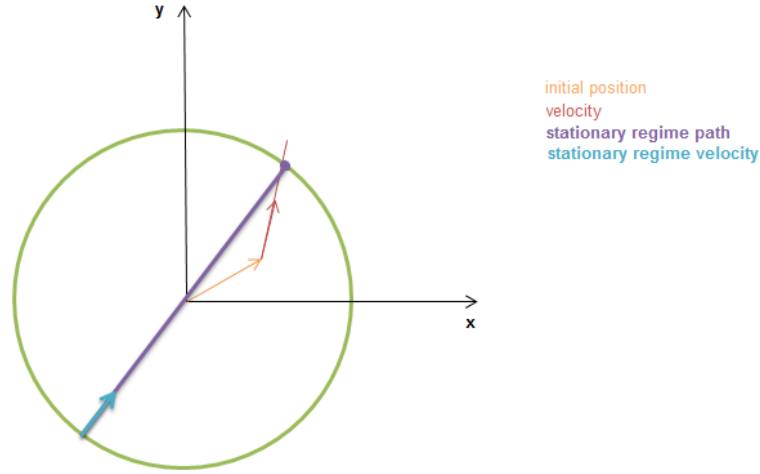
Fig. 37. Tiparul de comportament emergent de tip *PATRULARE*

Fig. 37 prezintă cele două regimuri de funcționare a entității: în primul rând, regimul tranzitoriu în care entitatea se deplasează de la poziția inițială la o distanță maximă de la poziția *HOME* și regimul staționar, în care aceasta se mișcă pe diametrul cercului descris, alternativ, în direcții opuse.

Poziția entității poate fi determinată la orice moment de timp prin formula următoare:

$$\vec{P}(t) = \begin{cases} \vec{P}(0) + \vec{v}_i(t), & t < t_0 \\ \vec{P}(t_0) + \vec{v}_d(t), & t \in \left[ t_0 + kT; t_0 + \frac{2k+1}{2}T \right) \\ -\vec{P}(t_0) - \vec{v}_d(t), & t \in \left[ t_0 + \frac{2k+1}{2}T; t_0 + (k+1)T \right) \end{cases} \quad (39)$$

unde

$$\vec{P}(t) = (x, y), \vec{P}(t_0) = (x_0, y_0), P(0) = (x_i, y_i) \quad (40)$$

$$\vec{v}_i(t) = (v_x t, v_y t) \quad (41)$$

și  $t_0$  este momentul de timp la care entitatea atinge distanța maximă față de poziția *HOME* iar  $(x_0, y_0)$  reprezintă perechea corespunzătoare de coordonate.

Astfel,

$$(x_i + t_0 v_x)^2 + (y_i + t_0 v_y)^2 = r^2 \quad (42)$$

Ecuția (42) conduce la calculul lui  $t_0$ .

$$t_0 = \frac{-(v_x x + v_y y) \pm \sqrt{(v_x x + v_y y)^2 - (v_x^2 + v_y^2)(x_i^2 + y_i^2 - r^2)}}{v_x^2 + v_y^2}, \quad t_0 \geq 0$$

$$t_0 = \frac{-(v_x x + v_y y) \pm \sqrt{(v_x x + v_y y)^2 - (v_x^2 + v_y^2)(x_i^2 + y_i^2 - r^2)}}{v_x^2 + v_y^2}, \quad t_0 \geq 0 \quad (43)$$

Viteza în regimul staționar este identică, în modul, cu viteza inițială ( $|\vec{v}_d| = |\vec{v}_i|$ ) și poate fi reprezentată prin următoarea relație:

$$\vec{v}_d(t) = \begin{cases} 0, & t \in [t_0 + k \frac{T}{2}; t_0 + T_t + k \frac{T}{2}] \\ \left(-\frac{2x_0}{T/2 - T_t}, -\frac{2y_0}{T/2 - T_t}\right), & t \in [t_0 + T_t + k \frac{T}{2}; t_0 + (k+1) \frac{T}{2}] \end{cases} \quad (44)$$

unde  $t_t$  este timpul necesar pentru a întoarce entitatea. Aceasta este dedus în paragrafele care urmează.

### 9.3 Implementarea tiparului de comportament emergent PATRULARE

Entitățile descrise în acest capitol au fost implementate cu ajutorul unor roboți LEGO Mindstorms NXT. Aceștia se deplasează folosind mișcarea diferențială. Aceasta constă în folosirea a două roți motoare dispuse central pe robot și o roată inactivă care permite rotirea robotului în jurul axei sale centrale. Cele 2 roți motoare pot fi acționate independent: dacă asupra motoarelor care le acționează se aplică curenți identici atunci cele 2 roți de vor învârti în același sens deplasând robotul înainte sau înapoi; dacă asupra motoarelor se aplică curenți de polarități opuse, roțile se vor învârti în direcții opuse, cauzând rotirea robotului în jurul axei sale.

#### 9.3.1 Calculul parametrilor statici ai roboților LEGO

Mișcarea diferențială depinde de un set de parametri printre care se numără: distanța dintre roți ( $DBW$ ) și diametrul unei roți ( $WD$ ) (în Fig. 38).

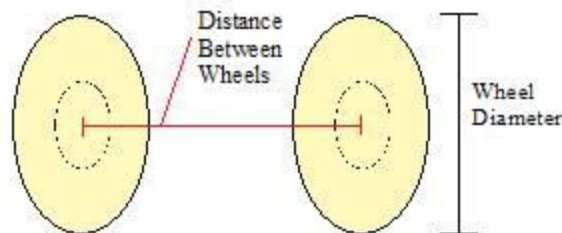


Fig. 38. Parametrii mișcării diferențiale [Cioarga 2008]

Pentru experimentul de față, roboții LEGO prezintă următorii parametri:  $WD = 5$  cm și  $DBW = 14$  cm. Atât pentru măsurarea gradului de rotație al robotului cât și pentru măsurarea distanței parcurse trebuie identificat cu precizie numărul de

grade necesar rotirii fiecărei roți pentru a produce efectul dorit (translație sau rotație). Astfel:

$$WDR = DPD * d \quad (45)$$

semnifică faptul că o roată se rotește cu  $d$  grade pentru a roti robotul cu  $WDR$  grade;  $DPD$  reprezintă o constantă care ține de fiecare robot și reprezintă numărul de grade cu care se rotește roata respectivă pentru a roti întregul robot cu 1 grad; cealaltă roată se rotește cu același număr de grade, dar în sens invers.

Totodată

$$WDM = DPC * c \quad (46)$$

semnifică faptul că o roată se rotește cu  $d$  grade pentru a cauza înaintarea robotului cu  $WDM$  centimetri;  $DPC$  reprezintă o constantă care ține de fiecare robot și reprezintă numărul de grade cu care se rotește roata respectivă pentru a mișca întregul robot cu 1 centimetru; cealaltă roată se rotește cu același număr de grade, și în același sens.

Atunci,

$$CC = DBW * \pi (= 43.96) \quad (47)$$

$$DiPD = CC / 360^\circ (= 0.122) \quad (48)$$

$$WC = WD * \pi (= 15.7) \quad (49)$$

$$DPC = 360^\circ / WC (= 22.93) \quad (50)$$

$$DPD = DiPD * DPC (= 2.8) \quad (50)$$

unde  $CC$  reprezintă circumferința cercului pe care îl parcurge robotul dacă o roată este staționară,  $DiPD$  este unghiul sub care se rotește fiecare roată pentru a roti întregul robot cu 1 grad iar  $WC$  este circumferința unei roți.

Înlocuind (47) - (50) în (45) și (46) se obțin:

$$WDR = 2.8 * d \quad (51)$$

$$WDM = 22.93 * c \quad (52)$$

### 9.3.2 Calculul parametrilor tiparului de comportament emergent PATRULARE

Viteză unghiulară  $AV$  este dedusă din (45) și (46) astfel:

$$AV = \frac{WDR}{T} = DPD \frac{d}{T} \quad (53)$$

$$AV = \frac{WDM}{T} = DPC \frac{c}{T} = DPC \cdot v \quad (54)$$

Pentru o întoarcere completă  $d = 180$ , astfel încât:

$$T_t = \frac{DPD \cdot d}{AV} = \frac{180 \cdot DPD}{DPC \cdot v} \quad (55)$$

unde  $T_t$  reprezintă timpul de întoarcere (de rotire completă).

Perioada de patrulare (timpul necesar parcurgerii cercului de 2 ori la care se adună timpul necesar întoarcerii pe loc a robotului),  $T$ , se calculează astfel:

$$T = \frac{4r}{v} + 2T_t \quad (56)$$

Comportamentul de patrulare obținut poate fi eficientizat prin considerarea faptului că robotul deține și atribuțiile de supraveghere; acesta patrulează cu scopul de a găsi ceva de interes în cadrul cercului reprezentat de distanța maximă de poziția *HOME*. Având în vedere că acesta are un câmp de vedere de 90 de grade, în continuare se calculează distanța minimă de vedere a robotului pentru ca să exploreze întregul mediu (discul cu raza egală cu distanța maximă de la poziția *HOME*) într-o singură perioadă  $T$ .

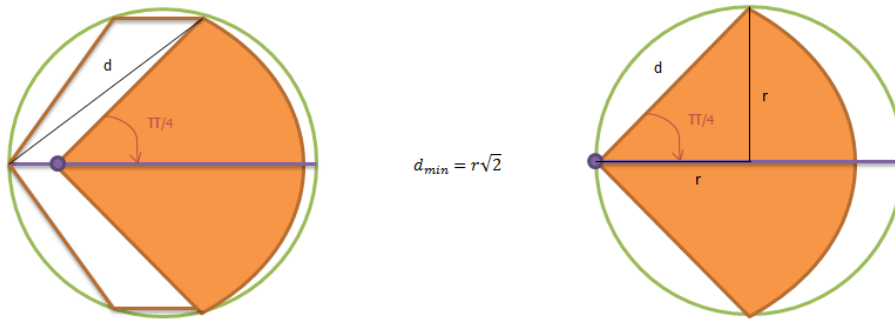


Fig. 39. Raza de vedere

Fig. 39 prezintă situația razei de vedere pentru robot. Aceasta este maximă dacă este poate fi circumscrisă unui diametru a cercului în cauză (la un unghi de 45° distanța de vedere este  $r\sqrt{2}$ ).

Pentru un unghi de vedere de 120°,

$$\left(\frac{d}{2}\right)^2 + r^2 = 4d^2 \quad (57)$$

determină o distanță minimă de  $\frac{2r\sqrt{3}}{3}$  care este mai mică decât cazul precedent.

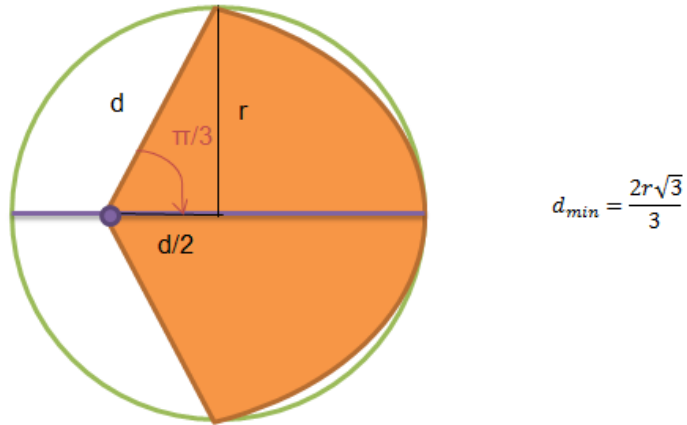


Fig. 40. Unghiul de vedere mărit

Astfel, pentru cazul general în care  $fov = 2 * \theta$ , se obține

$$d^2 = d^2 \cos^2 \theta + r^2 \quad (58)$$

$$d^2 = \frac{r^2}{\sqrt{1 - \cos^2 \theta}} \quad (59)$$

$$d_{min} = \frac{r}{\sin \theta} \quad (60)$$

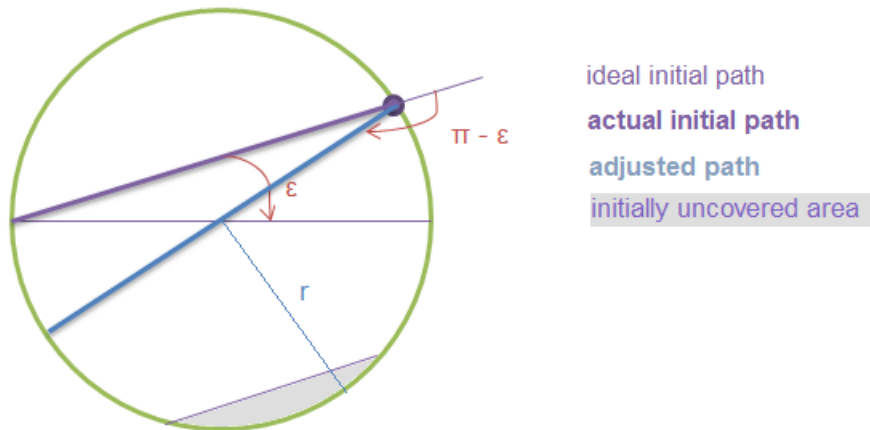


Fig. 41. Corecția rutei de patrulare



În cazul real, este posibil ca rotirea robotului să genereze o anumită derivă (Fig. 41).

Acest lucru determină lăsarea unei arii neacoperite de robot. Totuși algoritmul prezentat mai sus determină anumite rotații corectoare pentru unghiuri mai mici de  $180^\circ$  ceea ce conduce la acoperirea perfectă a întregii suprafețe.

Tiparul de comportament emergent de patrulare descris mai sus oferă o serie de posibilități de reglare automată a sistemului în vederea acoperirii întregii suprafețe care trebuie explorată.

## 9.4 Concluzii

Modelul de comportament emergent prezentat în acest capitol introduce noi formalizări care oferă o bază matematică pentru modelarea căilor de control a comportamentului emergent și își demonstrează utilitatea în modelarea comportamentului emergent prin prezentarea unui nou algoritm de explorare emergentă a unui mediu necunoscut și a unui nou tipar de comportament emergent.

Folosirea senzorilor cu care sunt echipați roboții LEGO pentru detecția și discriminarea obstacolelor și a resurselor determină o îmbunătățire a performanțelor modelului implementat pe acești roboți.

Rezultatele experimentale ale implementării modelului validează modelul matematic implementat pe roboți LEGO Mindstorms NXT.

În capitolul următor se prezintă implementarea și testarea mișcării în colective robotizate bazată pe studiul bancurilor de pești precum și două metrici, polarizarea și coeziunea, importante pentru descrierea performanțelor mișcării unui grup de entități.



## **10. IMPLEMENTAREA ȘI TESTAREA MIȘCĂRII ÎN COLECTIVE ROBOTICE BAZATĂ PE STUDIUL BANCURILOR DE PEȘTI**

În acest capitol se prezintă implementarea modelului de comportament emergent la mișcarea colectivelor robotice bazată pe studiul bancurilor de pești. Tiparele de comportament emergent prelevate din studiul bancurilor de pești, în special "fountain effect" și "flash expansion", sunt folosite la evitarea obstacolelor, respectiv la evitarea prădătorilor cu o minimă degradare a performanțelor grupului (păstrarea unei distanțe constante între membri grupului pentru a evita aglomerarea și păstrarea direcției generale de mișcare a grupului pentru a atinge țelul în timp minim).

În prima secțiune se prezintă modalitatea de concepere a abordării mișcării în colective robotice bazată pe studiul bancurilor de pești. Se prezintă, totodată, și principalele îmbunătățiri aduse modelelor de mișcare prezentate în capitolele anterioare.

Două metrici, coeziunea și polarizarea, se introduc în acest capitol pentru validarea modelului de comportament emergent descris în capitolul 6. Ele sunt derivate din metricile similare prezentate în [Huth 1992], adaptate și modificate semnificativ pentru a se potrivi unei game mai largi de mișcări. S-a folosit o simulare pe computer pentru a dovedi validitatea modelului matematic propus. De asemenea, utilizând aceleași metrici, o serie de experimente au fost realizate cu un grup de roboți LEGO Mindstorms NXT. Experimentele au fost efectuate variind numărul roboților de la 3 la 6, folosind sau nu obstacole și realizând diferite manevre de eschivare. Rezultatele experimentale arată că colectivele robotizate alcătuite de 6 roboți manifestă o performanță mai bună decât una alcătuită din 3 roboți. Pe de altă parte, un grup de 5 roboți depășesc ca performanță un grup de 6 roboți. Aceasta se explică prin faptul că datorită coeziunii sistemului, când au de a face cu un obstacol, mișcarea entităților este mult mai dezordonată. De asemenea aceasta dovedește proprietățile intrinseci ale comportamentului emergent a sistemului. Similar cu oricare alt sistem emergent, creșterea numărului de entități componente în modelul propus, poate îmbunătăți performanța până în momentul în care numărul maxim de interacțiuni posibile împiedică sistemul să declanșeze orice proprietate emergentă, scăzând de fapt performanța.

### **10.1 Conceperea modelului**

Tipurile de mișcare din [Aoki 1982, Huth 1992, Bhooshan 2001, Parrish 2002] prezintă doar trei influențe posibile, care sunt induse de entitatea vecină (atracție, repulsie și mișcarea paralelă). Acestea generează un anumit tip de mișcări limitate (paralelă cu direcția mișcării entității vecine în cazul influenței mișcării paralele sau o întoarcere completă în caz de repulsie). Pentru a oferi entităților din colectivele robotice o gamă mai largă de mișcări, algoritmul prezentat în [Cioarga 2008a, Cioarga 2008b, Cioarga 2009] introduce o gamă infinită de influențe în locul a doar celor trei numite anterior. De la atracție la repulsie, robotul se poate mișca în orice unghi orientat, care poate fi indus de un vecin, depinzând de distanța dintre ei. Dacă distanța este mai mică decât valoarea minimă dată, ceea ce înseamnă că o

coliziune este pe cale să aibă loc, atunci între entități se formează un sistem de repulsie totală, îndepărtându-se unul de celălalt în direcție opusă; dacă distanța dintre entități este egală cu valoarea maximă, atunci cele două entități vor fi complet atrase una de cealaltă. Dacă distanța dintre ele este între valorile de constrângere, unghiul dintre noua direcție de mișcare a peștelui se va modifica corespunzător. Dacă distanța este mai mare decât valoarea maximă atunci cele două entități nu se vor influența deloc.

O altă îmbunătățire este că atunci când entitatea principală este influențată de mai mult de o entitate, se calculează o medie de distribuție a greutateii tuturor influențelor. Cu scopul de a stabili noua poziție pentru un pește, algoritmul calculează un unghi de mișcare și o distanță. Unghiul arată noua orientare a peștelui și distanța arată lungimea mișcării. Pentru a calcula noua poziție vom folosi un vector de mișcare. Acest vector se calculează ca sumă a vectorilor care descriu mișcarea fiecărui pește vecin. Pentru fiecare pește, vectorul este reprezentat printr-un unghi și o distanță relativă față de starea inițială. Când un pește calculează lungimea mișcării ia în considerare distanța dintre el și vecini; cu cât vecinii sunt mai îndepărtați, cu atât este mai scurtă lungimea mișcării.

În cazul opus, când peștele este mai aproape de vecin, distanța este mai mare. Această analiză a fost făcută în așa fel încât, în orice moment, repulsia este mai importantă decât atracția, în scopul de a evita coliziunile.

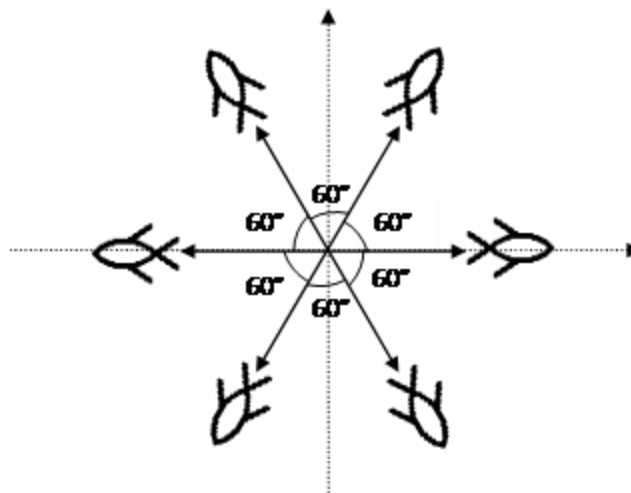


Fig. 42. Starea inițială a sistemului

Pentru ca algoritmul să funcționeze corect, este necesară cunoașterea stării inițiale (Fig. 42) pentru a poziționa întregul sistem.

Entitățile sunt plasate într-o formă circulară. Ele sunt echidistante și se îndreaptă în direcție opusă față de centru, făcând astfel un unghi egal între fiecare doi roboți consecutivi (în Fig. 42 sunt 6 roboți, astfel încât unghiul este exact  $60^\circ$ ). După ce roboții sunt poziționați, fiecare dintre ei înaintează cu un număr aleator de centimetri, după care se opresc și își calculează coordonatele. Acestea vor fi considerate coordonate inițiale.

Există de asemenea o stare finală la care ar trebui să ajungă peștele. În viața reală, acest obiectiv poate fi reprezentat de alimente sau un loc sigur pentru supraviețuire.

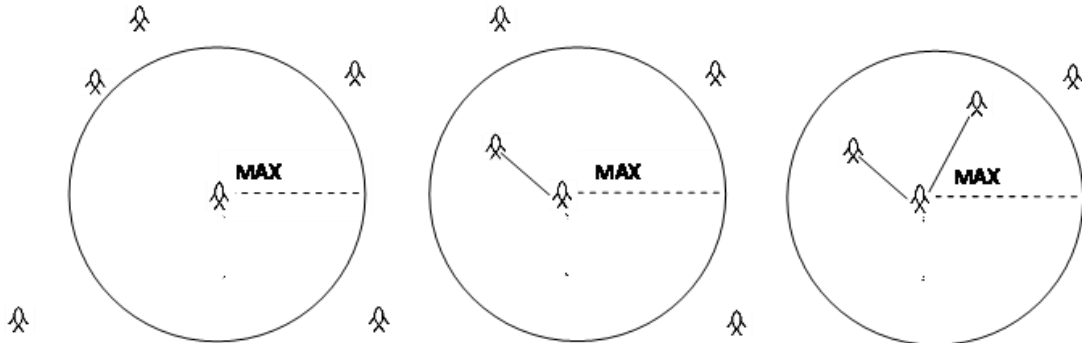


Fig. 43. Cele trei cazuri posibile de poziționare

La fiecare pas robotul va calcula distanța dintre el și vecinii săi. În funcție de legătura dintre distanța calculată și raza de vizibilitate a robotului, există trei cazuri posibile așa cum se vede în Fig. 43.

### 10.1.1 Cazul 1: Entitatea nu prezintă nici un vecin în apropiere

În acest caz, strategia necesară este de a căuta grupul; robotul cunoaște poziția relativă a vecinilor, calculează coordonatele centrului grupului și apoi se va îndrepta spre acest centru. Explicația extrasă din lumea naturală se caracterizează prin faptul că chiar dacă un pește nu poate distinge toți ceilalți pești vecini, este capabil să simtă întregul banc de pești și se poate îndrepta spre el.

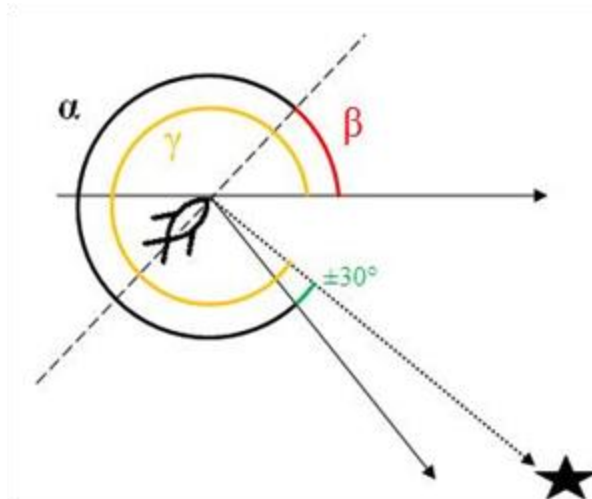


Fig. 44. Mișcarea spre grup a unei singure entități

Fig. 44 prezintă modul în care robotul își schimbă orientarea și începe să se deplaseze spre grup, unde:

- $\alpha$  este unghiul dintre orientarea curentă și cea dorită;
- $\beta$  este unghiul dintre axa orizontală ( $Ox$ ) și direcția curentă a peștelui (cunoscută la fiecare pas);
- $\gamma$  este unghiul dintre axa orizontală și direcția țintă a peștelui.

Pentru a da un factor de realitate algoritmului, am introdus un unghi aleator de  $\pm 30^\circ$ . Acest unghi se adaugă la unghiul calculat pentru direcția țintă a peștelui.

### 10.1.2 Cazul 2: Entitatea are exact un vecin în apropiere

În acest caz, influența care apare între cele două entități variază de la repulsie la atracție, în funcție de distanța dintre ele. Dacă această distanță este maximă, peștii se vor atrage reciproc, și dacă distanța este minimă, teoretic 0, peștii se vor respinge. Cu scopul de a crea un algoritm general, s-a introdus o funcție care primește ca valori de intrare distanțele posibile dintre doi pești, și generează un unghi final pe care peștele principal ar trebui să-l urmeze ( $\theta$ ). Acest unghi este relativ la direcția perechii peștilor vecini și evaluarea lui poate fi pozitivă sau negativă, ținând seama de poziția bancului. Direcția finală de deplasare îl va aduce pe pește mai aproape de țintă.

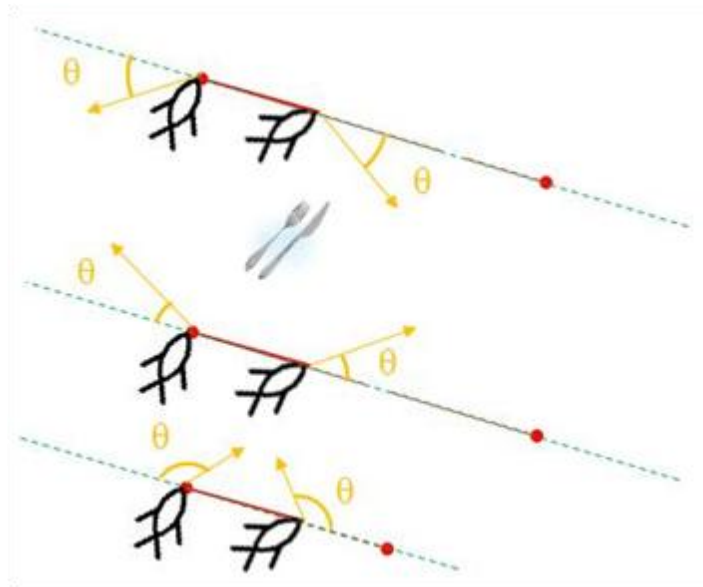


Fig. 45. Poziționarea relativă a bancului și direcțiile alese

Fig. 45 prezintă trei situații posibile care pot apărea în acest caz: repulsia în cazul în care cei 2 pești trebuie să se întoarcă sau să depășească punctul țintă al hranei și atracția către direcția hranei.

Pentru a calcula noul unghi de mișcare, încă mai avem nevoie de unghiul dintre axa orizontală și direcția peștelui vecin. Acesta este reprezentat de unghiul  $\delta$ , și valoarea sa este determinată la fel ca  $\theta$  din prima secțiune a capitolului, care înlocuiește centrul grupului cu cel al vecinului.

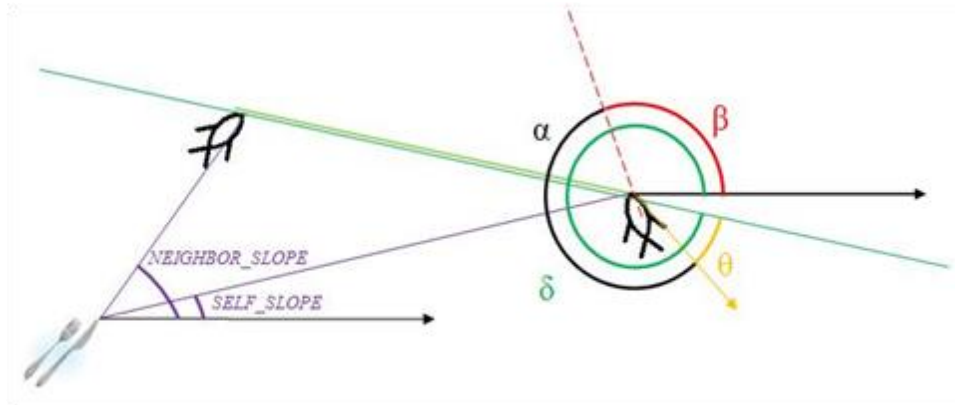


Fig. 46. Repulsia în sistemul de 2 pești, din perspectiva entității din dreapta

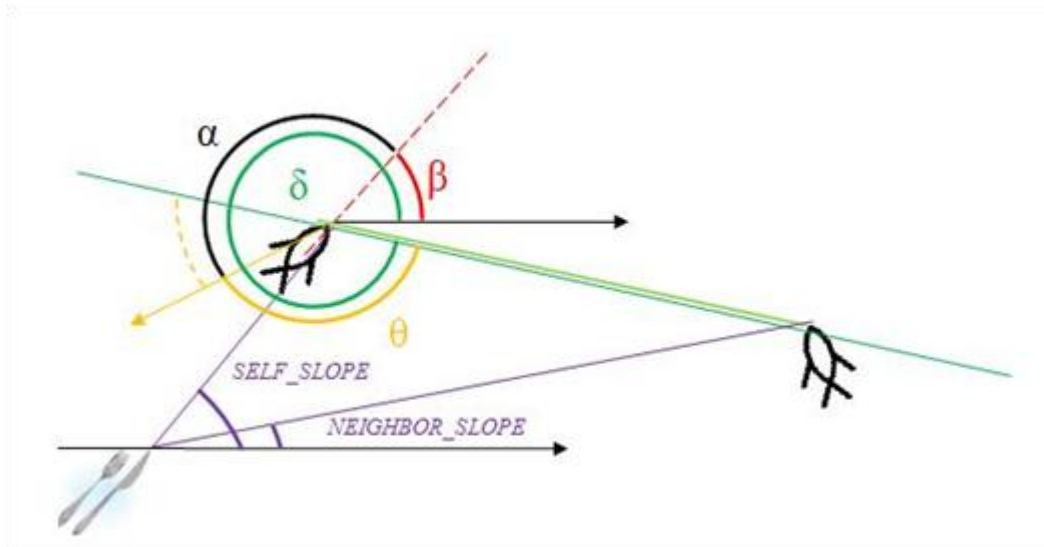


Fig. 47. Repulsia în sistemul de 2 pești, din perspectiva entității din stânga

Fig. 46 și Fig. 47 prezintă cazul de repulsie din perspectiva celor 2 pești din sistem.

### 10.1.3 Cazul 3: Entitatea are mai mult de un vecin în apropiere

În această situație generală, entitatea principală va calcula un vector rezultat pe baza tuturor vectorilor calculați pentru fiecare din vecinii entității.

## 10.2 Modelul matematic

Scopul principal al modelului este simularea comportamentului de învățare a peștilor și aplicarea acestuia la colectivele robotizate. Înainte de a descrie modelul, trebuie realizate câteva ipoteze:

- Mișcarea are loc într-un spațiu bi-dimensional și are două componente: direcția și viteza;
- Mișcarea este măsurată în pași; înainte de fiecare pas sunt calculate direcția și viteza și acestea nu se modifică până la următorul pas;
- Fiecare membru al grupului își știe starea inițială ca pe o pereche de coordonate într-un sistem cartezian absolut și un unghi care reprezintă direcția;
- Fiecare membru al grupului respectă același set de reguli descrise anterior și de aceea nu există lideri.

Comportamentul de învățare al peștilor are două aspecte principale: gruparea și orientarea paralelă. Fiecare membru al grupului, când determină direcția pentru următorul pas, poate fi una din cele trei regiuni conform poziției celui mai apropiat vecin în una din cele trei zone posibile: zona de confort, o zona optimală și o zonă de atracție. Aceste zone sunt arii circulare în jurul fiecărui membru și sunt delimitate de trei limite RC, RO și RA ca în Fig. 48.

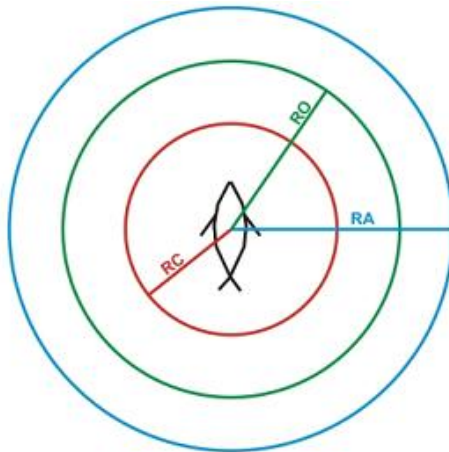


Fig. 48. Zone de percepție

Cele trei regiuni sunt ilustrate în Fig. 49:

- Evitarea: cei mai apropiați vecini sunt în zona de confort astfel încât există pericolul de coliziune (Fig. 49 a.);
- Orientarea paralelă: cei mai apropiați vecini se află în zona optimală. Membrul face parte dintr-un grup și va încerca să rămână în această stare (Fig. 49 b.);
- Atracție: cei mai apropiați vecini sunt în zona de atracție, însemnând că pot fi simțiți dar sunt considerați ca fiind prea departe astfel încât membrul nu face încă parte dintr-un grup (Fig. 49 c.).



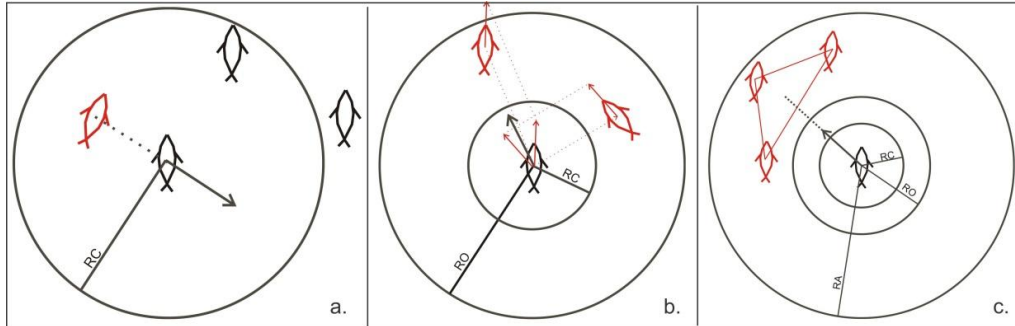


Fig. 49. Determinarea direcției de mișcare pentru cele 3 regiuni: a. Evitarea, b. Orientarea paralela și c. Atracția

Conform acestor regiuni, înainte de fiecare pas, este calculată o direcție utilizând (61),

$$\alpha_i = \begin{cases} -pa(x_m - x_i, y_m - y_i) & , \delta_m \leq RC \\ \frac{\sum_{j=1}^n \frac{\alpha_j}{\delta_{ij}}}{\sum_{j=1}^n \frac{1}{\delta_{ij}}} & , RC < \delta_{ij} \leq RO \\ pa(x_c - x_i, y_c - y_i) & , RO < \delta_c \leq RA \end{cases} \quad (61)$$

unde  $\alpha_i$  reprezintă direcția membrului  $i$ ,  $(x_i, y_i)$  sunt coordonatele membrului  $i$ ,  $pa(x, y)$  este funcția care calculează unghiul polar pentru setul de coordonate carteziene dat, definite în (2),  $\delta_m$  este distanța de la membrul  $i$  la cel mai apropiat vecin în timp ce  $(x_m, y_m)$  sunt coordonatele celui mai apropiat vecin,  $\alpha_j$  este direcția membrului  $j$  care este în regiunea optimă RO a membrului  $i$ ,  $\delta_{ij}$  este distanța dintre membrul  $i$  și  $j$ ,  $\delta_c$  este distanța de la membrul  $i$  la centrul masei al tuturor membrilor din zona de atracție RA, în timp ce  $(x_c, y_c)$  sunt coordonatele pentru centrul masei calculat în (3), unde  $n$  este numărul total de entități din regiunea RA.

$$pa(x, y) = \begin{cases} \arctan \frac{y}{x} & , x \geq 0 \\ \arctan \frac{x}{y} + \pi & x < 0, y \geq 0 \\ \arctan \frac{x}{y} - \pi & x < 0, y < 0 \\ \frac{\pi}{2} & x = 0, y > 0 \\ -\frac{\pi}{2} & x = 0, y < 0 \\ 0 & x = 0, y = 0 \end{cases} \quad (62)$$

$$x_C = \frac{\sum_{j=1}^n x_j}{n}, y_C = \frac{\sum_{j=1}^n y_j}{n} \quad (63)$$

Dacă membrul  $i$  nu detectează nici un vecin în una din aceste regiuni, va alege ori o direcție aleatoare între  $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$  radiani sau poate fi oferită o țintă și direcția de mișcare va fi spre aceasta.

După ce a fost stabilită o direcție  $\alpha_j$ , modelul calculează următorul set de coordonate pentru membrul  $i$ , conform (64), unde  $t$  reprezintă pasul de timp curent și  $S$  este distanța acoperită de robot la fiecare pas.

$$\begin{aligned} x_i(t+1) &= x_i(t) + S \cdot \cos(\alpha_j) \\ y_i(t+1) &= y_i(t) + S \cdot \sin(\alpha_j) \end{aligned} \quad (64)$$

### 10.2.1 Evitarea obstacolelor

Toate regulile explicate mai sus se aplică pentru fiecare membru al grupului dacă sunt neglijate toate interferențele externe. Cu toate acestea, modelul ia în considerare obstacolele / pericolele posibile din mediu. Într-o astfel de situație membrul grupului care a detectat un obstacol intra într-un așa numit regim de panică și se poate utiliza una dintre cele două tehnici de evitare :

- *Efectul fântână*: membrul  $i$  detectează un obstacol și își setează direcția conform (65), unde  $n_L$  și  $n_R$  este numărul de vecini din zona optimală situată la stânga și, respectiv, la dreapta, relativ la direcția anterioară a membrului  $i$ .

$$\alpha_i(t+1) = \begin{cases} \alpha_i(t) - \frac{\pi}{2} & , n_L < n_R \\ \alpha_i(t) + \frac{\pi}{2} & , n_L \geq n_R \end{cases} \quad (65)$$

Dependent de dimensiunea și poziția unui obstacol, se poate ca alți membrii ai grupului să nu îl detecteze. În acest caz, ei vor calcula direcția lor conform (61) și vor încerca fie să prevină coliziunea cu membrii evitând obstacolul fie vor încerca să mențină o orientare paralelă cu aceștia. Acesta va crea un așa numit "efect de fântână" văzut în învățarea peștilor la evitarea prădătorilor.

- *Flash expansion*: când membrul  $i$  detectează un obstacol / pericol, anunță toți membrii grupului, intră într-un regim de panică și își alege direcția conform (66).

$$\alpha_i(t+1) = \begin{cases} \alpha_i(t) - \alpha_R & , n_L < n_R \\ \alpha_i(t) + \alpha_R & , n_L \geq n_R \end{cases} \quad (66)$$

La recepționarea acestui mesaj, ceilalți membrii ai grupului își calculează poziția relativ la poziția transmițătorului și, pentru oricare vecin al transmițătorului,  $j$ , pentru care condiția (67) este adevărată (este situată în regimul optimal  $RO$ ), și de asemenea intră într-un regim de panică și își alege direcția conform (6), unde  $\alpha_i$  este direcția membrului  $i$  și  $\alpha_R$  este un număr aleator în virgulă mobilă în intervalul  $\left[\frac{\pi}{2}, \pi\right]$ .

$$\delta_{ij} \leq RO \quad (67)$$

Toți vecinii care nu satisfac condiția (67) și care nu intră în regimul de panică vor urma un set de reguli normal, creând efectul de "bliț de expansiune" datorită evitării coliziunii și comportamentelor de orientare paralelă. În timpul unui regim de panică pentru ambele tehnici de evitare a obstacolelor se pot regla viteza de mișcare și pasul  $S$  conform câtorva parametrii ca dimensiunea obstacolului și forma.

### 10.2.2 Metrici

Validitatea modelului propus a fost dovedit utilizând o simulare pe calculator precum și mai multe experimente efectuate cu până la 6 roboți. Două metrici au fost folosite pentru analizarea rezultatelor setărilor diferitelor experimente: gradul de coeziune și polarizarea. Aceste metrici sunt derivate din cele din [Bhoosan 2001]. Lucrarea citată ia în considerare doar o gamă limitată de mișcări (atracția, repulsia și mișcarea paralelă) și stabilește în consecință parametrii corespunzători. Matricele introduse, polarizarea și coeziunea, au fost intens modificate pentru a include orice unghi posibil între atracție și repulsie.

Gradul de coeziune,  $C$ , se definește ca fiind media aritmetică a distanțelor dintre fiecare membru a unui grup și cel mai apropiat vecin la un anumit moment (68).

$$C(t) = \frac{\sum_{i=1}^n \delta m_j(t)}{n} \quad (68)$$

Polarizarea,  $P$ , este definită ca fiind deviația (în procente) a direcției unui membru, raportat la direcția medie a grupului la un anumit moment (69).

$$P(t) = 100 - \frac{|\alpha_m(t) - \alpha_j(t)|}{180} \cdot 100 \quad (69)$$

unde  $\alpha_m$ , este direcția medie a grupului calculat folosind (70).

$$\alpha_m(t) = \operatorname{atan2} \left( \frac{\sum_{j=1}^n \sin \alpha_j(t)}{n}, \frac{\sum_{j=1}^n \cos \alpha_j(t)}{n} \right) \quad (70)$$

Fig. 50, Fig. 51 și Fig. 52 arată cum se calculează gradul de coeziune la un anumit moment într-un scenariu cu 6 roboți care se mișcă într-un mediu lipsit de obstacole, ocolind un obstacol utilizând tehnica "flash expansion", și, respectiv ocolind un obstacol folosind tehnica "fountain effect".

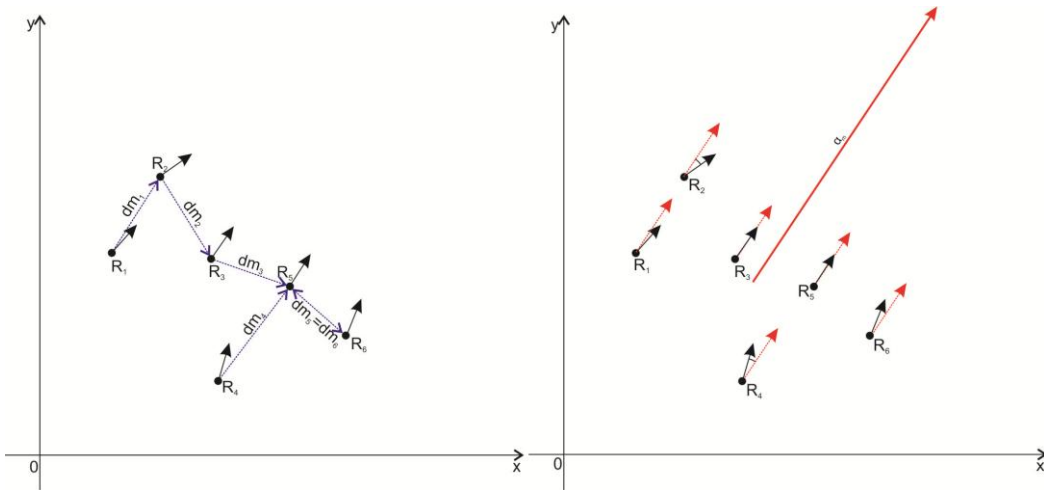


Fig. 50. Calcularea gradului de coeziune și polarizarea dintr-un grup de 6 roboți care se mișcă fără influențe din exterior (obstacole)

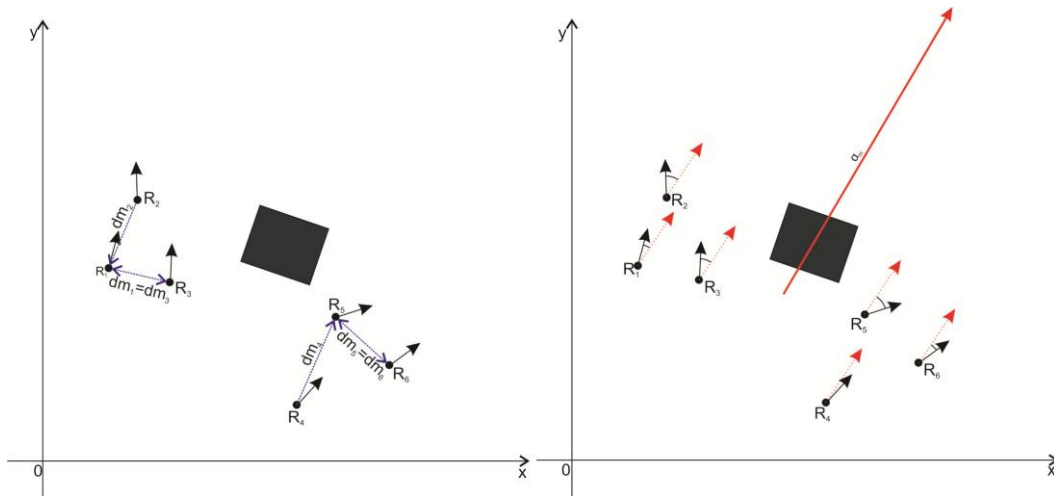


Fig. 51. Calcularea gradului de coeziune și polarizarea unui grup de 6 roboți în timp ce efectuează manevra de evitare "fountain effect"

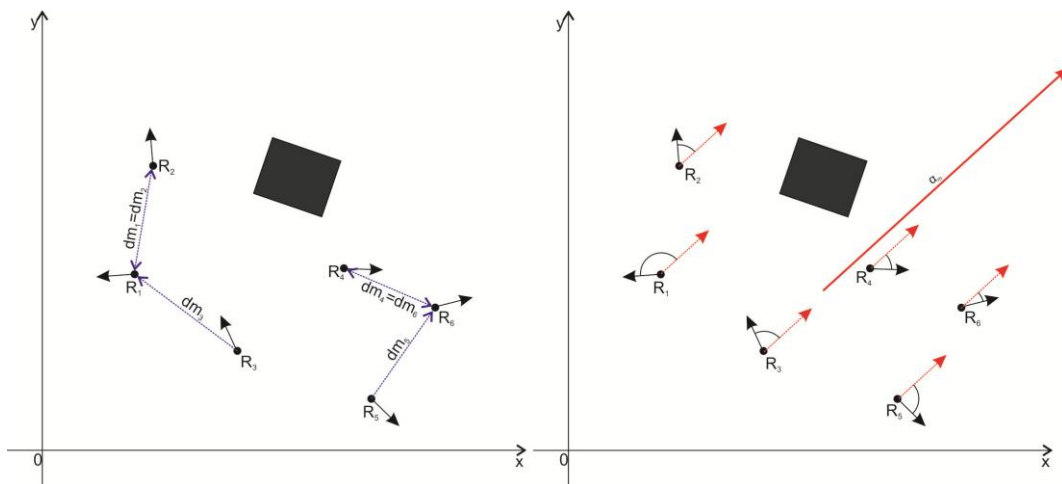


Fig. 52. Calcularea gradului de coeziune și polarizarea unui grup de 6 roboți în timp ce efectuează manevra de evitare "flash expansion"

### 10.2.3 Simularea modelului de mișcare în eBMS

Modelul matematic prezentat în această lucrare a fost simulat în eBMS și rezultatele sale au fost analizate prin prisma metricilor de polarizare și grad de coeziune. Modelul lui Aoki a fost folosit ca bază pentru comparațiile ulterioare.

Fig. 53 și Fig. 54 prezintă variația polarizării și respectiv a gradului de coeziune, când am simulat modelul Aoki de-a lungul a peste 100 de pași, folosind 5 entități de pești și următoarele valori pentru parametrii de tip senzorial:  $D1=30\text{cm}$ ,  $D2=60\text{cm}$ ,  $RC=400\text{cm}$ . Se poate observa că, în ciuda condițiilor ideale de simulare, ambele metrici prezintă variații semnificative și sistemul nu atinge o stare stabilă. Aceasta este de așteptat luând în considerare că modelul Aoki a simulat de

asemenea efectele stohastice ale erorilor senzoriale și de mișcare în procesul de luare a deciziilor, prin utilizarea unei distribuții normale pentru selecția direcției de deplasare.

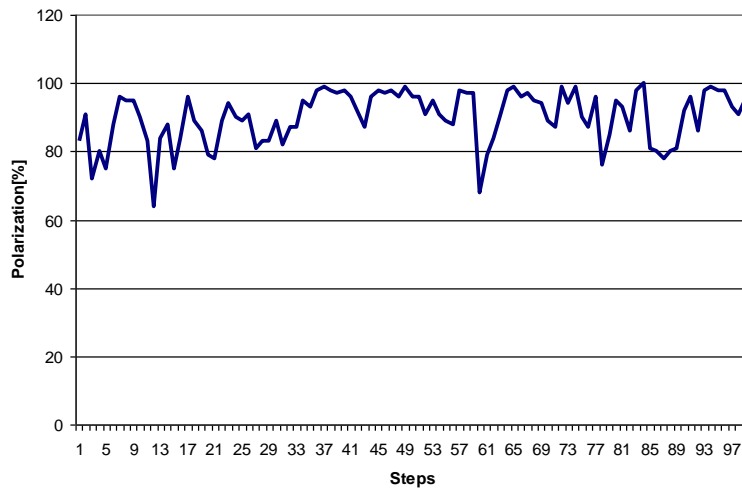


Fig. 53. Variația polarizării pentru simularea modelului Aoki folosind 5 entități pește

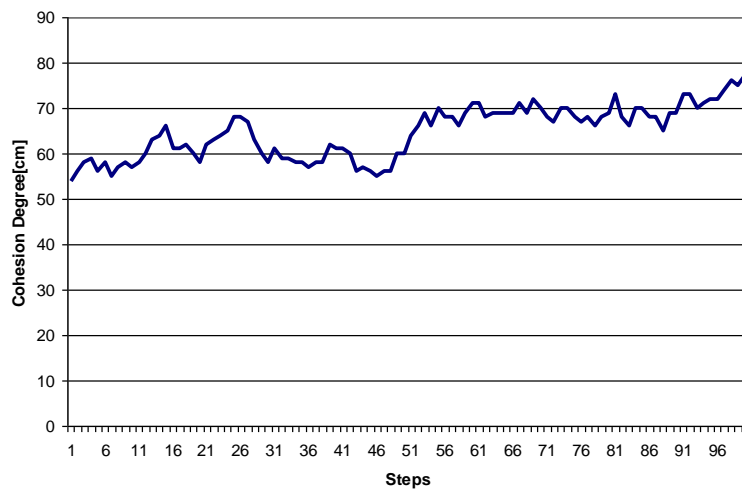


Fig. 54. Variația gradului de coeziune pentru simularea modelului Aoki folosind 5 entități pește

Fig. 55 și Fig. 56 arată variația polarizării și respectiv a gradului de polarizare, când a fost simulat modelul matematic propus, în aceleași condiții (de-a lungul a 100 de pași, 5 entități, aceași parametri senzoriali). Sistemul simulat introduce o stare stabilă după un anumit număr de pași (aproximativ 30) depinzând de poziția de start și orientarea entităților pește. În această stare stabilă, sistemul are o polarizare mai mare cu foarte mici variații, 89% - 93%, însemnând că întregul

grup se deplasează într-o direcție dreaptă. De asemenea gradul de coeziune are o valoare aproape constantă de 55 cm.

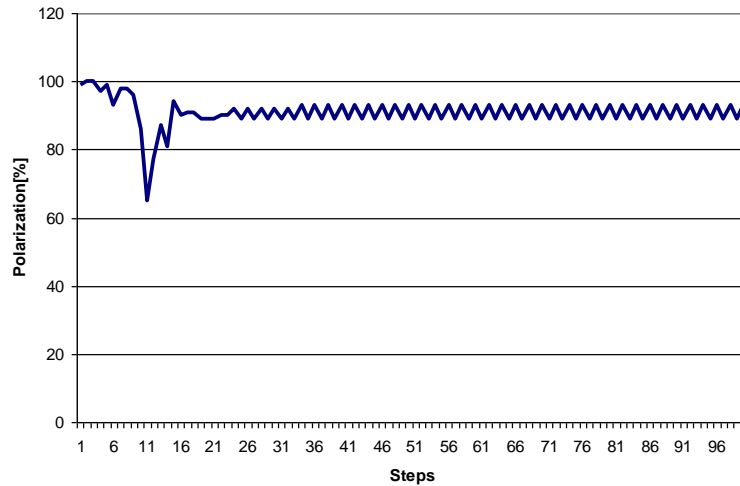


Fig. 55. Variația polarizării pentru 5 entități pește

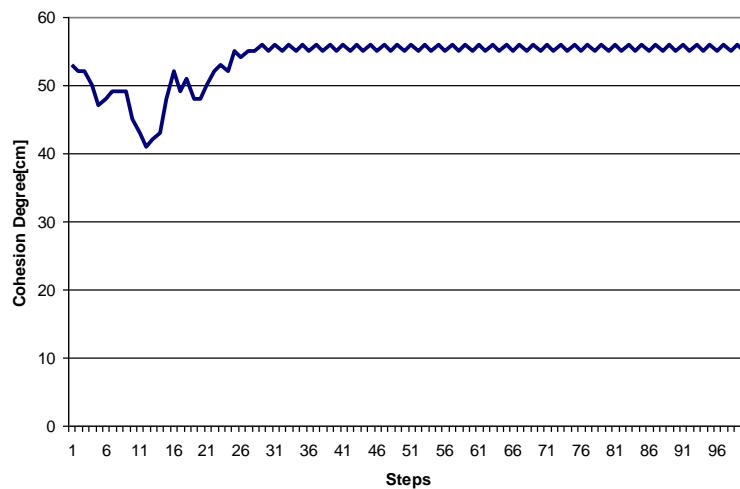


Fig. 56. Variația gradului de coeziune folosind 5 entități pește

### 10.3 Concluzii

În toate experimentele care au fost realizate, toți roboții și-au atins scopul, evitând obstacolele când a fost necesar. De asemenea, nu au avut loc coliziuni între ei.

Fig. 57 arată traiectoriile roboților din diferite scenarii experimentale, după cum urmează: a. 5 roboți într-un mediu fără obstacole (scenariul 4), b. 5 roboți ocolesc un obstacol utilizând manevra "flash expansion" (scenariul 6), c. 3 roboți evită un obstacol utilizând manevra "fountain effect" (scenariul 2) și d. 5 roboți ocolesc un obstacol utilizând manevra "fountain effect" (scenariul 5).

Experimentul inițial utilizând 3 roboți și nici un obstacol a fost utilizat ca punct de pornire pentru acest studiu. În acest caz, după gruparea inițială a roboților care a necesitat 10s, toți roboții au început să miște ca un grup în aceeași direcție generală (informație care este oferită de o valoare de polarizare aproape constantă de 100%). De asemenea, nu s-au înregistrat coliziuni între roboți, coliziunea având o valoare constantă, non-zero, pe durata experimentului. Acest experiment a validat miezul modelului, deoarece roboții s-au comportat conform așteptărilor și modelați, prin atingerea scopului acestora și prin lipsa coliziunilor între aceștia.

La utilizarea a 6 roboți într-un scenariu fără obstacole, măsurătorile coliziunii și polarizării au avut valori mai bune decât în cazul anterior. Întregul grup se mișcă mult mai grupat (gradul de coliziune scade la valorile de 40-45). Acest lucru validează de asemenea modelul deoarece creșterea numărului de roboți crește numărul posibil de vecini, și astfel gradul de respingere / atracție ale roboților din vecinătate.

Alte experimente au utilizat același număr de roboți, 5, dar au schimbat condițiile mediului prin adăugarea unui număr aleator de obstacole și prin realizarea unui experiment individual pentru fiecare banc de pești inspirat de manevrele de ocolire: "Flash Expansion" și "Fountain Effect". Toate aceste experimente au validat modelul, roboții s-au comportat conform prezicerilor:

- În cazul "fountain effect" coliziunea sistemului prezintă valori aproape constante deoarece întregul grup s-a împărțit în două grupuri mai mici, dar cu subgrupuri compacte; utilizând această tehnică de ocolire a avut un impact major asupra polarizării grupului în momentul ocolirii obstacolului;
- În cazul "Flash Expansion", coliziunea și polarizarea grupului au fost afectate în momentul evitării obstacolului.

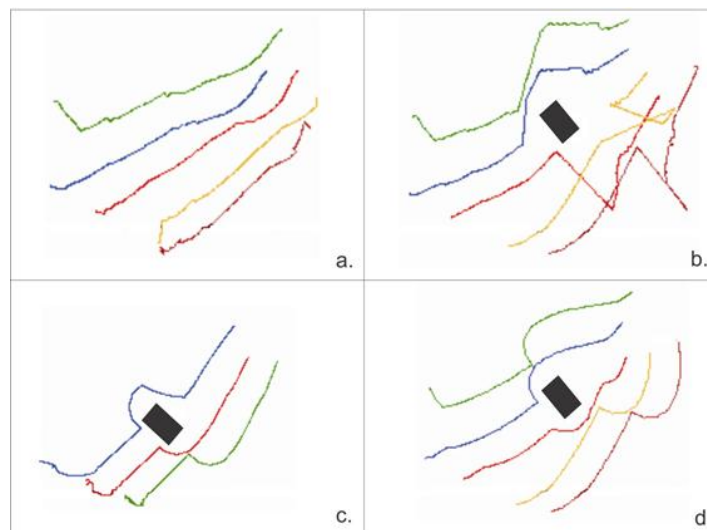


Fig. 57. Traiectoriile roboților în timpul unor scenarii experimentale



---

Rezultatele experimentului indicat în model prezintă o proprietate de comportament emergent puternic, conform așteptărilor, fiind inspirat de studiul învățării naturale a peștilor. Sistemul afișează o performanță mai mare în cazul utilizării a 6 roboți decât în cazul utilizării a 3 roboți, aparent dovedind ideea generală că cu cât crește numărul de roboți, cu atât trebuie să crească și coliziunea grupului. În ciuda acestui fapt, sistemul afișează o performanță mai mare la utilizarea a 5 roboți în loc de 6, dovedind comportamentul emergent al sistemului (utilizând mai mulți roboți înseamnă că interacțiunile posibile dintre aceștia cresc și polarizarea sistemului indică valorile reprezentând mișcarea neregulată în grupul de roboți).

Capitolul următor prezintă aplicarea modelului de comportament emergent într-un alt domeniu decât cel al mișcării colectivelor robotizate, și anume, colectarea datelor în rețele de calculatoare, senzori inteligenți sau alte entități.



# **11. APLICAREA TIPARELOR DE COMPORTAMENT EMERGENT ÎN DOMENIUL COLECTĂRII DATELOR ÎN REȚELE DE ENTITĂȚI**

Capitolul de față prezintă aplicarea tiparelor de comportament emergent într-un alt tip de aplicații decât cele specifice mișcării în colective robotizate prezentate în capitolele precedente.

Tiparele de comportament emergent preluate din studiul coloniilor de furnici sunt folosite pentru a modela comportamentul unei rețele de aplicații destinate colectării de date. Aceste tipare sunt identice cu cele descrise în capitolul 2 și folosite la implementarea și testarea modelului pentru comportament în capitolul 10.

## **11.1 Introducere**

Evoluția sistemelor de informații și a World Wide Web, combinată cu schimbarea de perspectivă, de la conținut static la conținut dinamic a paradigmei WEB 2.0, generează o nevoie extraordinară de stocare, regăsire, indexare a datelor pe Internet. În acest context, motoare de căutare extrem de puternice precum Google, AltaVista sau Yahoo utilizează aplicații software specializate numite crawlerele web.

Crawlerul web este o aplicație software automată care navighează pe World Wide Web într-un mod metodic. Motoarele de căutare pe internet utilizează web crawling ca un mijloc de a oferi informații actualizate cu privire la interogările care au fost postate de utilizatorii lor. Crawlerele web sunt de obicei folosite pentru a face copii locale a paginilor web care au fost vizitate; acestea pot fi prelucrate ulterior de către motoarele de căutare. Uneori, crawlerele pot fi utilizate pentru regăsirea de informații specifice de pe internet, cum ar fi adrese de e-mail, care sunt apoi folosite pentru spam.

Fiecare din crawlerele web comerciale, cum ar fi Google Crawler [Brin 1998], PolyBot [Shkapenyuk 2002] sau WebFountain [Edwards 2001], sau crawlerele web open-source cum ar fi GNU Wget, HTTrack sau WIRE [Castillo 2005], fac uz de algoritmi specifici sistemelor distribuite care necesită o cantitate mare de putere de calcul. Există unele crawlerele ca Ubicrawler care au fost special concepute să folosească tehnici colaborative.

Crawlerele web de sine stătătoare precum și cele distribuite folosesc algoritmi sofisticăți, de înaltă performanță, care necesită un grad ridicat de putere de calcul și folosesc tehnici complexe de inter-comunicare (multithreading, memorie partajată, etc.). Pe de altă parte, sistemul de crawlere web ERRIE [Cioarga 2007c] prezentat în continuare dezvoltă tipare de comportament emergent prin folosirea unor algoritmi extrem de simpli, care sunt foarte eficienți atunci când sunt folosiți de un număr mare de entități.

## 11.2 Specificații generale ale ERRIE (Emergent Retriever of Information)

ERRIE (Emergent Retriever of Information) folosește un algoritm bazat pe conceptul de feromoni preluat din studiul coloniilor de furnici. Fiecare furnică folosește aceste substanțe chimice cunoscute ca feromoni pentru comunicare. Schimbul de feromoni este una dintre modalitățile cele mai utilizate în sistemele de comunicare emergente naturale. Coloniile de furnici sunt un exemplu foarte important al unui sistem descentralizat, care folosesc modalități de comunicare bazate pe feromoni [Adjie 1999]. Furnicile individuale au o imagine locală a întregului sistem, creată pe baza schimbului feromoni între ele și alte furnici pe care le întâlnesc. Diferite tipuri de informații corespund la diferite tipuri de feromoni; furnicile extrag, de asemenea, informații din intensitatea feromonului. Prin extragerea informațiilor din feromoni, furnicile determină comportamentul furnicilor vecine și, astfel, își stabilesc propriul rol fără a determina un dialoguri sau orice alt tip de negociere. Comunicarea pe bază de feromoni este semi-fiabilă, deoarece mesajele chimice continuă să fie transmise chiar și după ce sursa mesajului încetează să mai existe. Pierderea de mesaje individuale nu are nici o importanță. Acest tip de comunicare este unidirecțional; traseele de feromoni sunt create de furnici ca pentru alte furnici să le urmeze. Nu există nici o comunicare duplex; se folosește în loc o pereche de mesaje simplex, dar acestea nu sunt corelate ca parte dintr-o conversație.

Algoritmul emergent care stă la baza aplicației ERRIE folosește o bază de date MySQL (Fig. 58) ca un suport pentru comunicare. Baza de date MySQL oferă suport pentru indexare și căutări full-text, care sunt necesare atunci când sunt preluate link-uri și sunt utile pentru a preveni parcurgerea unor link-uri deja vizitate.

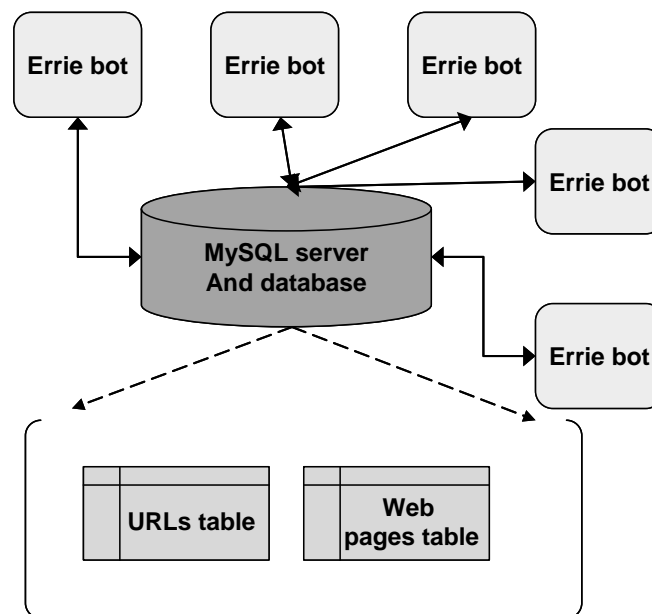


Fig. 58. Arhitectura sistemului ERRIE

Fiecare acces la baza de date este tranzacțională (tranzacțiile sunt pe deplin susținute de serverul MySQL), permițând astfel accesul concurrent la datele stocate.

### 11.3 Evaluarea performanțelor

Au fost efectuate o serie de experimente în scopul de a compara performanțele crawler-ului ERRIE cu cele ale versiunii sale non-emergente, WIRE.

Crawlerul web WIRE este extrem de predictibil pentru că acesta se execută în cicluri, care sunt foarte bine definite. Singurul criteriu de incertitudine este introdus de extragerea de link-uri de la pagini web, ca și de numărul de link-uri găsite în fiecare pagina web, care este extrem de imprevizibil. Pentru a avea un set intrări similare pentru toate experimentele, s-a folosit același set pagini inițiale.

Experimentul realizat pentru sistemul WIRE este folosit ca o bază pentru celelalte experimente. Numărul de link-uri recuperate a fost înregistrat după fiecare ciclu și este prezentat în Fig. 59.

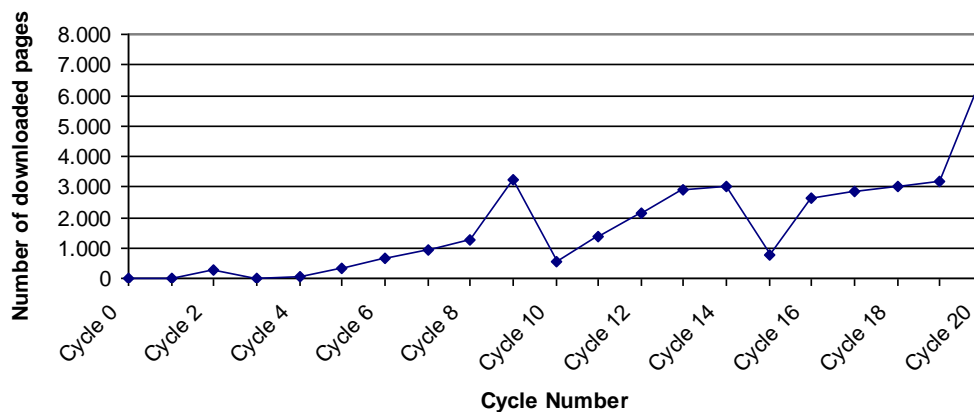


Fig. 59. Numărul de pagini descărcate de sistemul WIRE.

Următorul experiment a fost efectuat pentru a determina numărul de pagini web descărcate de un număr variabil de crawlerele web ERRIE. Baza de date MySQL a fost populată cu aceleași surse inițiale ca și în experimentul precedent. Un număr variabil de boți ERRIE a fost stabilit pentru a rula în același timp: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 35, 40, 45, 50 și 100 de boți ERRIE. Pentru acest experiment s-au înregistrat atât numărul de link-uri descărcate și timpul necesar pentru a descărca toate paginile web folosind lățime de bandă standard de 100 kbps. Similar cu experimentul WIRE, sistemul ERRIE a fost configurat pentru a rula timp de 24 de cicluri complete. Rezultatele acestui experiment sunt prezentate în Fig. 60.

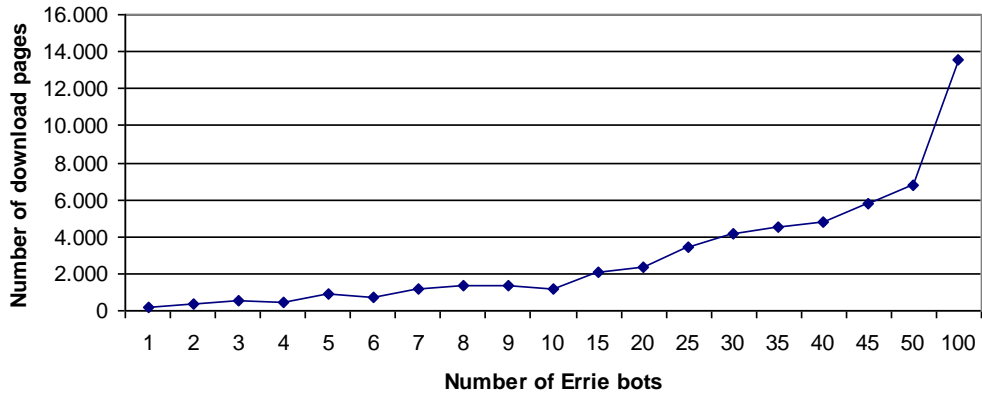


Fig. 60. Numărul de pagini web descărcate prin modificarea numărului de boti ERRIE.

Fig. 61 prezintă numărul de pagini descărcate de un singur crawler ERRIE modificând de data aceasta dimensiunea zonei tampon din baza de date. In Fig. 62 se prezintă timpul necesar descărcării acestor pagini.

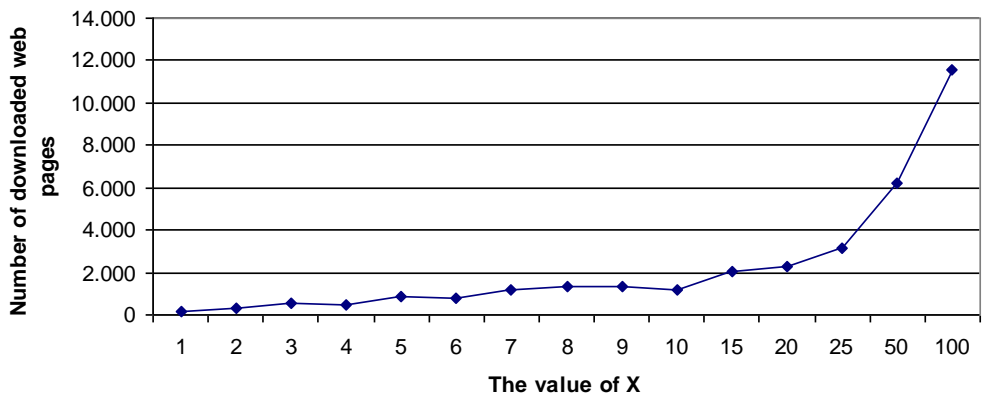


Fig. 61. Numărul de pagini web descărcate de 1 bot ERRIE prin modificarea zonei tampon.

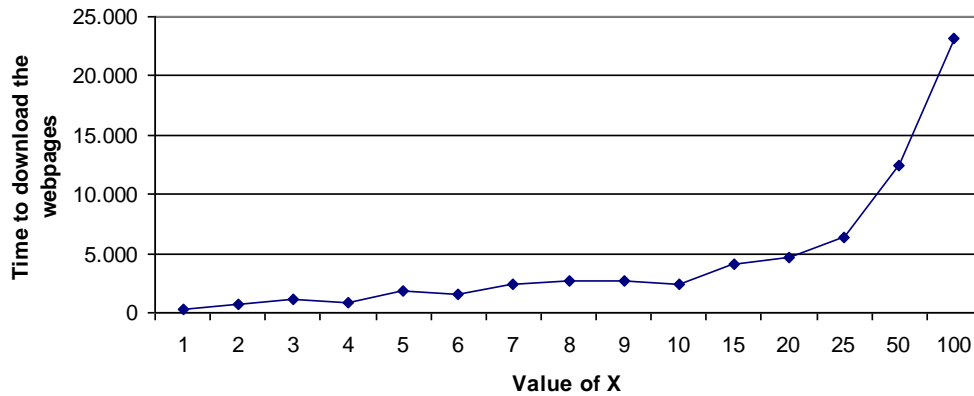


Fig. 62. Timpul necesar descărcării paginilor de 1 bot ERRIE prin modificarea dimensiunii zonei tampon.

## 11.4 Concluzii

În acest capitol s-a prezentat un tipar de comportament emergent preluat din studiul coloniilor de furnici și aplicat în domeniul web, în cadrul comunicației între anumite entități. Acest capitol introduce un nou tip de crawler web care face uz de căutarea full-text și posibilitățile de indexare ale serverului de baze de date MySQL. Crawler-ul web emergent prezentat aici folosește principiile preluate din studiul comunicării în coloniile de furnici, folosind serverului de baze de date MySQL ca un strat de persistență pentru comunicarea inter-crawler.

Totodată, s-a realizat și o comparație între crawler-ul emergent și unul comun, open-source. S-au realizat o serie de experimente pentru a demonstra valabilitatea conceptului. Conform principiilor emergente, s-a găsit un compromis între numărul de crawlere și numărul de pagini descărcate pentru a mări eficiența sistemului.

Capitolul următor conține concluziile lucrării de doctorat precum și perspectivele de dezvoltare ulterioară.





## 12. CONCLUZII ȘI PERSPECTIVE

### 12.1 Concluzii

Teza de față dezbate problema modelării tiparelor de comportament emergent în medii colaborative robotizate și rețele de senzori inteligenți. Lucrarea se bazează pe următoarele idei:

- Utilizarea unor echipe de roboți pentru efectuarea unor activități complexe cum ar fi explorarea, monitorizarea și operarea în medii necunoscute sau periculoase pentru om prezintă o serie de avantaje esențiale: substituirea intervenției umane, autonomie de operare pe termen lung, precizie foarte bună și constantă, creșterea eficienței prin paralelizare, creșterea fiabilității sistemului în ansamblu prin redundanță și distribuirea resurselor;
- Utilizarea colectivelor robotice implică o serie de probleme, în special legate de controlul roboților, comunicația între roboți, precum și de distribuirea eficientă a sarcinilor în cadrul echipei de roboți. Aplicarea comportamentului emergent rezolvă problemele sistemelor asociative, în special în cazul în care aceste sisteme au un număr ridicat de entități componente;
- Prin studiul modelelor existente pentru formalizarea tiparelor de comportament colaborativ s-au putut elabora un set coerent de modele pentru formalizarea tiparelor de comportament colaborativ (folosind în special tiparele de comportament emergent preluate de la coloniile de furnici și bancurile de pești);
- Modelul de comportament emergent a fost validat teoretic cu ajutorul suitei de aplicații de calcul complex Mathematica; în plus, acesta a mai fost validat printr-un set de simulări și experimente.

În continuare este prezentată o discuție detaliată despre modul în care au fost atinse obiectivele propuse.

- 1) Dezvoltarea cadrului teoretic și metodologic în domeniu prin conceperea și realizarea modelului destinat reprezentării tiparelor de comportament emergent

S-a definit un model matematic folosit pentru determinarea numărului minim de entități necesar pentru declanșarea comportamentului emergent în situația unui colectiv robotic. S-au prezentat pașii următori pentru conceperea modelului respectiv pornind de la studiul comportamentului coloniilor de furnici în cazul explorării mediilor necunoscute, colectării de resurse precum și al ocolirii de obstacole.

Pentru o aplicație anume, modelul introdus în teza de față poate să calculeze următorii parametrii cheie:

- Numărul minim de entități care este necesar pentru declanșarea comportamentului emergent în sistem,  $N_{TE}$ ;
- Timpul necesar entităților pentru atingerea emergenței,  $T_{TE}$ ;
- Timpul necesar pentru atingerea scopului,  $T_{TG}$ ;

Pentru validarea modelului s-au introdus o serie de metrici precum *afinitatea*, *numărul de entități* și *gradul de activitate*. Acestea au fost folosite pentru a calcula performanțele modelului.

S-a introdus funcția *scop* pentru a modela diferitele tipare de comportament emergent, s-au prezentat proprietățile acestei funcții care trebuie respectate pentru aplicarea sa cu succes în cadrul modelului și s-au prezentat formele acesteia pentru un set de aplicații de tipul ocolirii de obstacole, explorării emergente și colectării de resurse ([Cioa07, Cioa08, Cioa08-2, Cioa09, Cioa10]).

## 2) Validarea teoretică a modelului de comportament emergent

S-a prezentat validarea teoretică a modelului de comportament emergent folosind suita de aplicații de calcul complex Mathematica. S-au studiat valorile parametrilor modelului precum și valorile metricilor propuse pentru două cazuri specifice: un exemplu de ocolire de obstacole pentru care parametrii modelului au valori similare cu cele din lumea reală, și un exemplu similar de ocolire de obstacole pentru care valorile parametrilor modelului sunt deformați spre extrem pentru modelarea cazului cel mai defavorabil.

Validarea și analiza performanțelor modelului dezvoltat în cadrul programului de doctorat s-a realizat prin utilizarea unui simulatorului special creat precum și prin utilizarea prototipului platformei integrate destinată studiului acestor sisteme, CORE-TX [Cioarga 2006], dezvoltată de către doctorand împreună cu echipa de proiect în cadrul unui grant de cercetare de excelență, numărul CEEEX-ET-07/2006-2008, MEEdC – UEFISCSU.

Modelul de comportament emergent a fost simulat cu succes folosind eBMS, tiparele de comportament emergent ale fiecărei simulări fiind descrise prin eBML.

S-au rulat seturi de simulări folosind tipare de comportament emergent specifice ocolirii de obstacole, colectării de resurse și explorării emergente; fiecare set de simulări a fost variat prin modificare mediului în care WIT-ul se mișcă, prin modificarea incrementală a numărului de WIT-uri, prin modificarea poziției și a numărului de resurse.

Similar cu rezultatele validării teoretice a modelului de comportament emergent, analiza rezultatelor simulărilor a condus la determinarea unui interval probabilistic de declanșare a comportamentului emergent și, respectiv, de atingere completă a țelului propus. Acest interval de dimensiuni ale colectivului robotizat folosit în experiment este determinat pentru fiecare tip de tipar de comportament emergent în parte și este consecvent cu valorile teoretice determinate prin aplicarea directă a modelului ([Cioa07-2, Cioa09]).

## 3) Adaptarea modelului de comportament emergent la aplicații reale și validarea experimentală a acestuia

Platforma integrată CORE-TX a fost folosită pentru implementarea modelului și pentru testarea acestuia în două cazuri:

- folosind tipare de comportament emergent preluate din studiul mișcării în coloniile de furnici;
- folosind tipare de comportament emergent preluate din studiul mișcării în bancurile de pești.

Amândouă cazurile au fost descrise prin prisma implementării și adaptării modelului pentru tipare de comportament emergent pe un colectiv robotic format din roboți LEGO Mindstorms NXT.

S-a introdus un nou tipar de comportament emergent, *PATRULARE*; a fost prezentată adaptarea modelului matematic pentru descrierea noului tipar. Totodată, s-a prezentat modalitatea de control a mișcării roboților LEGO și s-a analizat o

modalitate de explorare emergentă a mediului prin prisma detecției obstacolelor și identificării resurselor prin folosirea doar a senzorilor cu care este echipat un robot LEGO Mindstorms NXT (senzor de culoare pentru discriminarea între obstacole și resurse și senzor ultrasonic pentru identificarea obstacolelor și a resurselor).

S-a prezentat modalitatea de concepere a abordării mișcării în colective robotice bazată pe studiul bancurilor de pești și principalele îmbunătățiri aduse modelelor de mișcare existente în literatură și prezentate în capitolele anterioare. Tiparele de comportament emergent prelevate din studiul bancurilor de pești, în special "fountain effect" și "flash expansion", au fost folosite la aplicații de tipul ocolirii de obstacolelor, respectiv de tipul evitării prădătorilor cu o minimă degradare a performanțelor grupului (păstrarea unei distanțe constante între membri grupului pentru a evita aglomerarea și păstrarea direcției generale de mișcare a grupului pentru a atinge țelul în timp minim).

Au fost introduse două metrici, *gradul de coeziune* și *polarizarea*, pentru validarea adaptării descrisă a modelului de comportament emergent.

S-a prezentat și aplicarea tiparelor de comportament emergent într-un alt tip de aplicații decât cele specifice mișcării în colective robotizate prezentate deja ([Cioa08, Cioa09, Cioa10, Cioa11]).

## 12.2 Sinteza contribuțiilor

Un rezumat al contribuțiilor prezentate în teza de față este prezentat în continuare.

- Dezvoltarea unui cadru teoretic și metodologic destinat reprezentării tiparelor de comportament emergent.
- Definierea unui model matematic pentru determinarea numărului minim de entități necesar pentru declanșarea comportamentului emergent, a timpului necesar pentru atingerea emergenței, respectiv a timpului necesar pentru atingerea scopului. Pentru validarea modelului s-au introdus o serie de metrici specifice precum afinitatea, numărul de entități și gradul de activitate.
- Introducerea funcției scop pentru modelarea diferitelor tipare de comportament emergent, stabilindu-se formele acestora pentru diverse seturi de aplicații cum ar fi ocolirea de obstacole, explorarea emergentă și colectarea de resurse.
- Validarea teoretică a modelului de comportament emergent folosind utilitarul Mathematica.
- Concepția și dezvoltarea unui limbaj de descriere a tiparelor de comportament emergent numit eBML.
- Concepția și dezvoltarea unui simulator de comportament emergent numit eBMS.
- Validarea și analiza performanțelor modelului propus prin simulare utilizând simulatorul eBMS și limbajul eBML pe prototipului platformei integrate CORE-TX destinată studiului acestui tip de sisteme, platformă dezvoltată în cadrul laboratorului DSPLabs.
- Implementarea, adaptarea și validarea experimentală a modelului de comportament emergent la aplicații de mișcare folosind tipare de comportament emergent bazate pe studiul coloniilor de furnici și a bancurilor de pești.
- Implementarea, adaptarea și validarea modelului de comportament emergent în domeniul colectărilor de date.

### **12.3 Perspective**

Se dorește adaptarea, implementarea și validarea modelului pentru tipare de comportament emergent la domenii diverse, diferite de mobilitatea în sisteme robotice care a stat la baza conceperii acestuia:

- Crearea unui protocol de comunicație emergent care să rezolve problema consumului de energie al unui nod al rețelei de senzori datorat comunicației radio, fără fir;
- Crearea unui protocol de localizare a entităților și de procesare a imaginilor pentru compunerea unei imagini complexe din imagini simple capturate de entitățile colectivului robotic.

## LUCRĂRI PUBLICATE

- [Cioa05] Razvan Cioarga, Mihai V. Micea, "Applications of Emergent Behavior Patterns on Intelligent Sensor Networks", *Scientific Bulletin of "Politehnica" University of Timisoara, Transactions on Automatic Control and Computer Science*, Vol. 50 (64), No. 2, Timisoara, Romania, 2005, pp. (15-24), ISSN 1224-600X;
- [Cioa06] Razvan D. Cioarga, Mihai V. Micea, Bogdan Ciubotaru, Dan Chiciudean, Daniela Stanescu, "CORE-TX: Collective Robotic Environment - the Timisoara Experiment", in *Proceedings of the 3-rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence*, SACI'2006, Timisoara, Romania, May 2006, pp. (495-506), ISBN 963-7154-46-9;
- [Stan07] Stanescu Daniela, Stratulat Mircea, Ciubotaru Bogdan, Chiciudean Dan, Cioarga Razvan-Dorel, Micea Mihai Victor, "Embedding Data in Video Stream Using Steganography", *Proceedings of the 4-th International Symposium on Applied Computational Intelligence and Informatics*, SACI 2007, Timisoara, Romania, Mai 2007, pp. 241-244, ISBN 1-4244-1234-X;
- [Cioa07] Cioarga Razvan-Dorel, Ciubotaru Bogdan, Chiciudean Dan, Micea Mihai Victor, Cretu Vladimir-Ioan, Groza Voicu, "Emergent Behavioral Modeling Language in Obstacle Avoidance", *Proceedings of the 24-th IEEE Instrumentation and Measurement Technology Conference*, IMTC 2007, Warsaw, Poland, Mai 2007, pp.1614-1619, ISBN 1-4244-0589-0;
- [Ciub07] Ciubotaru Bogdan, Cioarga Razvan-Dorel, Chiciudean Dan, Micea Mihai Victor, Stratulat Mircea, "Connectivity Improvement in Wireless Sensor Networks Based on Mobile Nodes", *Proceedings of the 24-th IEEE Instrumentation and Measurement Technology Conference*, IMTC 2007, Warsaw, Poland, Mai 2007, pp. 1640-1645, ISBN 1-4244-0589-0;
- [Stan07-2] Stanescu Daniela, Stratulat Mircea, Ciubotaru Bogdan, Chiciudean Dan, Cioarga Razvan-Dorel, Borca Daniel, "Digital watermarking using Karhunen-Loeve transform", *Proceedings of the 4-th International Symposium on Applied Computational Intelligence and Informatics*, SACI 2007, Timisoara, Romania, Mai 2007, pp 187-190, ISBN 1-4244-1234-X;
- [Cioa07-2] Cioarga Razvan-Dorel, Micea Mihai Victor, Ciubotaru Bogdan, Cretu Vladimir-Ioan, Chiciudean Dan, "Emergent System for Information Retrieval", *Springer Series: IFIP International Foundation for Information Processing, Artificial Intelligence and Innovations 2007: From Theory to Applications*, Ed.: C. Boukis, A. Pnevmatikakis, L. Polymenakos, I. Maglogiannis, Springer Science and Business Media, Boston, 2007, Vol. 247, pp 409-417;
- [Cioa07-3] Cioarga Razvan-Dorel, Micea Mihai Victor, Ciubotaru Bogdan, Chiciudean Dan, Cretu Vladimir-Ioan, Groza Voicu, "eBML: A Formal Language for Behavior Modeling and Application Development in Robotic Collectives", *Proceedings of the International Workshop on Robotic and Sensors Environments (ROSE 2007)*, Ottawa, Canada, Oct 2007, pp. 80-85, ISBN 1-4244-1527-6;
- [Cioa08] Cioarga Razvan-Dorel, Nalatan Iuliana, Tura-Bob Sergiu, Micea Mihai Victor, Cretu Vladimir-Ioan, Groza Voicu, "Emergent Exploration and Resource Gathering in Collaborative Robotic Environments", *Proceedings of the IEEE International Workshop on Robotic and Sensors*

- Environments (ROSE 2008)*, Ottawa, Canada, Oct 2008, pp. 13-18, ISBN 978-1-4244-2594-5;
- [Cioa08-2] Cioarga Razvan-Dorel, Panus Bogdan, Oancea Claudia, Micea Mihai Victor, Cretu Vladimir-Ioan, Petriu Emil, "Fish Shoal Inspired Movement in Robotic Collectives", *Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE 2008)*, Ottawa, Canada, Oct 2008, pp. 7-12, ISBN 978-1-4244-2594-5;
- [Mice08] Micea Mihai Victor, Cioarga Razvan-Dorel, Cretu Vladimir-Ioan, Certejan Cristina, Stangaciu Valentin, Petriu Emil, "Inter-Task Communication and Synchronization in the Hard Real-Time Compact Kernel HARETICK", *Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE 2008)*, Ottawa, Canada, Oct 2008, pp. 19-24;
- [Cioa09] Razvan Cioarga, Mihai V. Micea, Vladimir Cretu, Emil M. Petriu, "Movement in Collaborative Robotic Environments Based on the Fish Shoal Emergent Patterns", *Sensors & Transducers Journal*, Vol. 5 (Spec. Issue), No. 3, International Frequency Sensor Association (IFSA) Publishing, Mar. 2009, ISSN 1726-5479;
- [Cioa10] Razvan D. Cioarga, Mihai V. Micea, Vladimir Cretu, Voicu Groza, "Modeling and Evaluation of Fish Shoal Inspired Movement in Collaborative Robotic Environments", *International Instrumentation and Measurement Technology Conference*, Pages: 1539 - 1544, Mai 2010, ISBN 978-1-4244-2832-8, ISSN 1091-5281;
- [Cioa11] Razvan-Dorel Cioarga, Mihai V. Micea, Vladimir Cretu, Andrei Banc, Voicu Groza, "Foodball: Emergent resource gathering in collective robotic environments", *Proc. Of IEEE International Workshop on Robotic and Sensors Environments, ROSE 2011*, (accepted for publication).

## BIBLIOGRAFIE

- [Adjie 1999] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, Jeremy Lilley, "The Design and Implementation of an Intentional Naming System", *Proc. of the 17th ACM Symposium on Operating Systems Principles*, Dec. 1999;
- [Anthony 2004] Richard John Anthony, Natural Inspiration for Self-Adaptive Systems; Database and Expert Systems Applications, 2004. Proceedings. 15th International Workshop on 30 Aug.-3 Sept. 2004 Page(s):732 – 736;
- [Aoki 1982] Aoki, I., "A simulation study on the schooling mechanism in fish", *Bulletin of the Japan Society of Scientific Fisheries*, Vol. 48(8), 1081-1088, 1982.
- [Ashokaraj 2009] I.A.R. Ashokaraj, P.M.G. Silson, A. Tsourdos, and B.A.White, "Robust sensor-based navigation for mobile robots," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 3, pp. 551 –556, march 2009;
- [Bhoosan 2001] N. Bhooshan, "The Simulation of the Movement of Fish Schools", *Technical Report for The Institute for Systems Research, ISR, UG-2001-4*, University of Maryland.
- [Bonabeau 1997] Eric Bonabeau, Andrej Sobkowski, Guy Theraulaz, Jean-Louis Deneubourg, "Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects". *Proc. of Biocomputing and emergent computation*, BCEC 1997, pp 36-45.
- [Boppana 2001] R. Boppana, S. Konduru, "An Adaptive Distance Vector Routing Algorithm for Mobile, Ad- Hoc Networks", *Proc. IEEE INFOCOM*, Anchorage, USA, Apr. 2001.
- [Brin 1998] S. Brin, L. Page, "The anatomy of a large-scale hypertextual Web search engine", *Proceedings of the seventh international conference on World Wide Web*, Pages: 107 – 117, 1998.
- [Broxton 2005] M. Broxton, J. Lifton, J. Paradiso, "Localizing a Sensor Network via Collaborative Processing of Global Stimuli", *Proc. EWSN Conf.*, 2005.
- [Burgard 2000] Burgard W., Fox D., Moors M., Simmons R., Thrun S., "Collaborative Multi-Robot Exploration." In *Proceedings of the International Conference on Robotics and Automation*, San Francisco, May 2000;
- [Camazine 2001] Scott Camazine, Nigel R. Franks, James Sneyd, Eric Bonabeau , Jean-Louis Deneubourg, Guy Theraulaz, "Self-Organization in Biological Systems", *Princeton University Press*, Princeton, NJ, 2001.
- [Castillo 2005] C. Castillo, R. Baeza-Yates, "WIRE: an Open Source Web Information Retrieval Environment", *Workshop on Open Source Web Information Retrieval*, 2005;
- [Chrysan. 2004] G. Chrysanthakopoulos, W. L. J. Fox, R. T. Miyamoto, R. J. Marks II, et al., "A Fuzzy-Logic Autonomous Agent Applied as

- a Supervisory Controller in a Simulated Environment", *IEEE Trans. Fuzzy Syst.*, Vol. 12, No. 1, Feb. 2004.
- [Cioarga 2005] R. Cioarga, M. V. Micea, *Applications of Emergent Behavior Patterns on Intelligent Sensor Networks*, Sci. Bulletin of UPT, Trans. Autom. Control Computer Sci., Vol. 50 (64), No. 2, Timisoara, 2005, pp. (15-24).
- [Cioarga 2006] R. D. Cioarga, M. V. Micea, B. Ciubotaru, D. Chiciudean, D. Stanescu, *CORE-TX: Collective Robotic Environment - the Timisoara Experiment*, in Proc. 3-rd Romanian-Hungarian Joint Symp. on Applied Computational Intelig., SACI 2006, Timisoara, Romania, May 2006, pp. (495-506), ISBN 963-7154-46-9, IEEE Cat. No. 07EX1788.
- [Cioarga 2006] Razvan D. Cioarga, Mihai V. Micea, Bogdan Ciubotaru, Dan Chiciudean, Daniela Stanescu, "CORE-TX: Collective Robotic Environment - the Timisoara Experiment", in *Proceedings of the 3-rd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence, SACI'2006*, Timisoara, Romania, May 2006, pp. (495-506), ISBN 963-7154-46-9;
- [Cioarga 2007] R. Cioarga, B. Ciubotaru, D. Chiciudean, M. V. Micea, V. Cretu, V. Groza, *Emergent Behavioral Modeling Language in Obstacle Avoidance*, in Proc. 24-th IEEE Instrum. and Meas. Technol. Conf., IMTC 2007, Warsaw, Poland, May 2007, [CD support], ISBN 1-4244-0589-0.
- [Cioarga 2007a] Cioarga Razvan-Dorel, Ciubotaru Bogdan, Chiciudean Dan, Micea Mihai Victor, Cretu Vladimir-Ioan, Groza Voicu, "Emergent Behavioral Modeling Language in Obstacle Avoidance", *Proceedings of the 24-th IEEE Instrumentation and Measurement Technology Conference, IMTC 2007*, Warsaw, Poland, Mai 2007, pp.1614-1619, ISBN 1-4244-0589-0;
- [Cioarga 2007b] Cioarga Razvan-Dorel, Micea Mihai Victor, Ciubotaru Bogdan, Chiciudean Dan, Cretu Vladimir-Ioan, Groza Voicu, "eBML: A Formal Language for Behavior Modeling and Application Development in Robotic Collectives", *Proceedings of the International Workshop on Robotic and Sensors Environments (ROSE 2007)*, Ottawa, Canada, Oct 2007, pp. 80-85, ISBN 1-4244-1527-6;
- [Cioarga 2007c] Cioarga Razvan-Dorel, Micea Mihai Victor, Ciubotaru Bogdan, Cretu Vladimir-Ioan, Chiciudean Dan, "Emergent System for Information Retrieval", *Springer Series: IFIP International Foundation for Information Processing, Artificial Intelligence and Innovations 2007: From Theory to Applications*, Ed.: C. Boukis, A. Pnevmatikakis, L. Polymenakos, I. Maglogiannis, Springer Science and Business Media, Boston, 2007, Vol. 247, pp 409-417;
- [Cioarga 2008a] Cioarga Razvan-Dorel, Nalatan Iuliana, Tura-Bob Sergiu, Micea Mihai Victor, Cretu Vladimir-Ioan, Groza Voicu, "Emergent Exploration and Resource Gathering in Collaborative Robotic Environments", *Proceedings of the IEEE*



- International Workshop on Robotic and Sensors Environments (ROSE 2008)*, Ottawa, Canada, Oct 2008, pp. 13-18, ISBN 978-1-4244-2594-5;
- [Cioarga 2008b] Cioarga Razvan-Dorel, Panus Bogdan, Oancea Claudia, Micea Mihai Victor, Cretu Vladimir-Ioan, Petriu Emil, "Fish Shoal Inspired Movement in Robotic Collectives", *Proceedings of the IEEE International Workshop on Robotic and Sensors Environments (ROSE 2008)*, Ottawa, Canada, Oct 2008, pp. 7-12, ISBN 978-1-4244-2594-5;
- [Cioarga 2009] Razvan Cioarga, Mihai V. Micea, Vladimir Cretu, Emil M. Petriu, "Movement in Collaborative Robotic Environments Based on the Fish Shoal Emergent Patterns", *Sensors & Transducers Journal*, Vol. 5 (Spec. Issue), No. 3, International Frequency Sensor Association (IFSA) Publishing, Mar. 2009, ISSN 1726-5479;
- [Ciubotaru 2007] B. Ciubotaru, R. Cioarga, D. Chiciudean, M. V. Micea, M. Stratulat, *Connectivity Improvement in Wireless Sensor Networks Based on Mobile Nodes*, in Proc. 24-th IEEE Instrum. and Meas. Technol. Conf., IMTC 2007, Warsaw, Poland, May 2007, [CD support], ISBN 1-4244-0589-0.
- [Codling 2007] E.A.Codling, J.W. Pitchford, and S.D. Simpson, "Group navigation and the many-wrongs principle in models of animal movement," *Ecology*, vol. 88, no. 7, pp. 1864–1870, 2007;
- [Couzin 2002] Iain D. Couzin, Jens Krause, Richard James, Graeme D. Ruxton, and Nigel R. Franks, "Collective memory and spatial sorting in animal groups", *Journal of Theoretical Biology*, vol. 218, no. 1, pp. 1 – 11, 2002;
- [Dai 2006] Y.-S. Dai, M. Hinchey, M. Madhusoodan, J. L. Rash, X. Zou, *A Prototype Model for Self-Healing and Self-Reproduction In Swarm Robotics System*, 2-nd IEEE Symp. Depend., Auton. Secure Comput., Sept. 2006, pp. (3-10).
- [Darpa 2005] DARPA / MTO / MEMS, DARPA Agency, online: <http://www.darpa.mil/mto/mems/>
- [Deng 2003] Jing Deng, Richard Han, Shivakant Mishra; „Sensor networks: Security support for in-network processing in Wireless Sensor Networks"; *Proceedings of the 1st ACM workshop on Security ducaof ad hoc and sensor networks*; October 2003;
- [Di Caro 2005] Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella; *AntHocNet: An Adaptive Nature-inspired Algorithm for Routing In Mobile Adhoc Networks*; *European Transactions on Telecommunications*, Euro. Trans. Telecomms. 2005; 16:443–455;
- [Dorigo 1997a] Marco Dorigo, Luca Maria Gambardella; *Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem*, *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, 1997;
- [Dorigo 1997b] Marco Dorigo, Luca Maria Gambardella; *Ant colonies for the traveling salesman problem*, *BioSystems*, Elsevier, 1997.;

- [Ducatelle 2005] Frederick Ducatelle, Gianni Di Caro, Luca Maria Gambardella; *Using Ant Agents to Combine Reactive and Proactive Strategies for Routing in Mobile Ad Hoc Networks*; International Journal on Computational Intelligence and Applications (IJCIA), Special Issue on Nature-Inspired Approaches to Networks and Telecommunications, Volume 5, Number 2, Pages 169-184, June 2005;
- [Edwards 2001] J. Edwards, K. McCurley, and J. Tomlin, "An adaptive model of optimizing performance of an incremental Web crawler", *Proceedings of the Tenth International World Wide Web Conference*, pages 106-113, May 2001.
- [Elson 2001] Elson, J. and Estrin, D. 2001. "Time synchronization for wireless sensor networks", In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (San Francisco, CA);
- [Elson 2002] Elson, J., Girod, L., and Estrin, D., "Fine-Grained network time synchronization using reference broadcasts", *Proceedings of the 5th symposium on Operating systems design and implementation*, Volume 36 Issue SI, Winter 2002;
- [Gai 2002] P. Gai, L. Abeni, G. Buttazzo, *Multiprocessor DSP Scheduling in System-on-a-Chip Architectures*, Proc. 14-th Euromicro Conf. Real-Time Syst. ECRTS'02, Vienna, Austria, 2002, pp. (231-240).
- [Gambardella 2000] Luca Maria Gambardella, Marco Dorigo; *An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem*; INFORMS Journal on Computing Vol.12, No.3, Summer 2000;
- [Ganeriwala 2003] Ganeriwala, S., Kumar, R., And Srivastava, M. B., "Timing-Sync protocol for Sensor Networks", In *Proceedings of the 1st ACM Conference on Embedded Networked Sensor Systems (SenSys)*. 138-149, 2003;
- [Gardner 2001] J.W. Gardner, V. Varadan, O. Awadelkarim, *Microsensors, MEMS and Smart Devices*, Wiley, New York, 2001.
- [Heinzelman 2002] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", *IEEE Trans. Wireless Commun.*, Vol.1, No. 4, Oct. 2002, pp. (660- 670).
- [Hinchee 2005] M. Hinchee, C. Rouff, J. Rash, *Requirements of an Integrated Formal Method for Intelligent Swarms*, FMICS'05, September 5-6, 2005, Lisbon, Portugal, Page(s) 125 - 133;
- [Hofmann 1997] Hofmann-Wellenhof, B., Lichtenegger, H., and Collins, J. 1997. "Global Positioning System: Theory and Practice", 4th ed. Springer Verlag.
- [Huth 1992] Huth, A., & Wissel, C, "The simulation of the movement of fish schools". *Journal of Theoretical Biology*, Volume 156, Issue 3, 7 June 1992, Pages 365-385;

- [Hwang 2009] Kao-Shing Hwang, Chia-Yue Lo, and Wen-Long Liu, "A modular agent architecture for an autonomous robot," *Instrumentation and Measurement, IEEE Transactions on*, vol. 58, no. 8, pp. 2797–2806, aug. 2009;
- [Inada 2004] Y. Inada, K. Kawachi, and H. Liu, "Simulation Study of Schooling Motion of Fish based on Two Observed Motions: Approaching Motion and Parallel Orientating Motion", in *Proc. Of Modelling and Simulation (MS 2004)*, California, USA, ISBN 0-88986-391-1;
- [Intanag. 2000] C. Intanagonwiwat, R. Govindan, D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Proc. 6th Intl. Conf. Mob. Comput. and Netwk.*, Boston, USA, Aug 2000, pp.(56-67).
- [Intel 2005] Intel Research, *Instrumenting the World: An Introduction to Wireless Sensor Networks*, Intel Corp., 2005, online:  
[http://www.intel.com/research/exploratory/instrument\\_world.htm](http://www.intel.com/research/exploratory/instrument_world.htm).
- [Kashima 2004] Takuya Kashima, Shunsuke Saruwatari, Hiroyuki Morikawa, Tomonori Aoyama, "A Bind Control Model For Real-space Programming in Ubiquitous Computing Environment", The Sixth International Conference on Ubiquitous Computing, Ubicomp 2004, 7-10 September, 2004;
- [Ko 2004] Bong-Jun Ko, Dan Rubenstein; "Distributed Server Replication in Large Scale Networks"; in *NOSSDAV'04*, June 16–18, 2004, Cork, Ireland.
- [Koenig 1998] Sven Koenig, Reid G. Simmons, "A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models", *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, MIT Press, 1998, pp. 91 – 122.
- [Kteam 2005] K-TEAM K-Robots: Khepera I, II, III, Koala, I, II, online:  
<http://www.k-team.com/kteam/home.php?rub=0&site=1&version=EN>;
- [Kube 2000] Kube R., Bonabeau E., "Cooperative transport by ants and robots", *Robotics and Autonomous Systems*, Issue 30, 2000, pp. 85-101;
- [Mainwaring 2002] A. Mainwaring, J. Polastre, R. Szewczyk, D. E. Culler, J. Anderson, *Wireless Sensor Networks for Habitat Monitoring*, Proc. ACM Wshop. Sensor Networks and Applic., WSNA, Sept. 2002.
- [Maniezzo 2004] Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luigi; *Ant Colony Optimization, New Optimization Techniques in Engineering*, by Onwubolu, G. C., and B. V. Babu, Springer-Verlag Berlin Heidelberg, 101-117, 2004;
- [Martinez 2007] Martinez, S., Bullo, F., Cortes, J., and Frazzoli, E., "On synchronous robotic networks - Part I: Models, tasks and

- complexity" *IEEE Transactions on Automatic Control*, Vol. 52, No. 12, December 2007;
- [Mateas 2004] Michael Mateas and Andrew Stern; "A Behavior Language: Joint Action and Behavioral Idioms", In *H. Prendinger and M. Ishizuka (Eds), Life-like Characters. Tools, Affective Functions and Applications*, Springer, 2004;
- [McLurkin 2004] McLurkin, J., and Smith, J. "Distributed Algorithms for Dispersion in Indoor Environments using a Swarm of Autonomous Mobile Robots.", 7th International Symposium on Distributed Autonomous Robotic Systems (DARS) (2004);
- [Micea 2006] M. V. Micea, V. Cretu, V. Groza, *Maximum Predictability in Signal Interactions With HARETICK Kernel*, IEEE Trans. on Instrum. and Measurement, Vol. 55, Nr. 4, Aug. 2006, pp. (1317-1330), ISSN 0018-9456.
- [Mills 1991] Mills, D. L. 1991. "Internet time synchronization: The network time protocol.", in *IEEE Trans. Commun.*39, 10 (Oct.), 1482-1493.
- [Mills 1994] Mills, D. L. 1994. "Improved algorithms for synchronizing computer network clocks", In *Proceedings of the ACM Conference on Communication Architectures (ACM SIGCOMM)* (London, UK).
- [Minami 2004] M. Minami, S. Saruwatari, T. Kashima, et al., "Implementation-based Approach for Designing Practical Sensor Network Systems", *Proc. 11-th Asia-Pacific Software Eng. Conf., APSEC 2004*, pp. (703-710).
- [Monada 2005] F. Mondada, L. M. Gambardella, D. Floreano, S. Nolfi, et al., *The Cooperation of Swarm-bots: Physical Interactions in Collective Robotics*, IEEE Robotics & Autom. Mag., Vol. 12, No 2, Jun. 2005, pp. (21-28).
- [Montemanni 2005a] R.Montemanni , L.M. Gambardella, A.E.Rizzoli,A.V.Donati; *Ant Colony System for a Dynamic Vehicle Routing Problem*, Journal of Combinatorial Optimization, 10, 327-343, December 2005;
- [Montemanni 2005b] R.Montemanni , L.M. Gambardella; *Exact Algorithms For The Minimum Power Symmetric Connectivity Problem in Wireless Networks*; Computers and Operations Research, 32(11), 2891-2904, November 2005;
- [Nouyan 2006] Nouyan, S., Groß, R., Bonani, M., Mondada, F., Dorigo, M., "Group transport along a robot chain in a self-organized robot colony", In *Proceedings of the 9th International Conference on Intelligent Autonomous Systems*, Amsterdam 2006, pp. 433-442;
- [Parrish 2002] Julia K. Parrish, Steven V. Viscido and Daniel Grünbaum, "Self-Organized Fish Schools: An Examination of Emergent Properties", *Biological Bulletin*, Vol. 202, No. 3 (Jun., 2002), pp. 296-305.

- [Parunak 2005] H. Van Dyke Parunak, S. Brueckner, J. Sauter, R. Matthews, *Global Convergence of Local Agent Behaviors*, AAMAS'05, July 25-29, 2005, Utrecht, Netherlands, Page(s) 305 – 312;
- [Popa 2006] M. Popa, A. S. Popa, V. Cretu, M. V. Micea, *Monitoring Serial Communications in Microcontroller Based Embedded Systems*, in Proc. IEEE Intl. Conf. on Computer Eng. and Syst., ICCES06, Cairo, Egypt, Nov. 2006, pp. (56-61), ISBN 1-4244-0272-7, IEEE Cat. No. 06EX1338.
- [Privosnik 2002] Privosnik, M.; Marolt, M.; Kavcic, A.; Divjak, S.; Evolutionary construction of emergent properties in multi-agent systems; Electrotechnical Conference, 2002. MELECON 2002. 11th Mediterranean 7-9 May 2002;
- [Reilly 1997] W. Scott Neal Reilly; „A methodology for building believable social agents”; *Proceedings of the first international conference on Autonomous agents*; February 1997;
- [Romer 2001] Romer, K. “Time synchronization in ad hoc networks”, In *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (Long Beach, CA), 173–182, 2001;
- [Rothermich 2004] Rothermich, J., Ecemis, M., and Gaudiano, P. “Distributed localization and mapping with a robotic swarm”, *Swarm Robotics*, Springer-Verlag , 2004, pp. 59–71;
- [Roumeliotis 2002] Roumeliotis S., Bekey G., “Distributed Multirobot Localization”, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, October 2002;
- [Seah 2006] W. K. G. Seah, K. Z. Liu, et al.: *Mobility-enhanced Wireless Sensor-Actuator Networks Sensor Networks*, 2006 IEEE Intl. Conf. Ubiquitous and Trustworthy Computing,, Vol.1, Jun. 2006, pp. (548- 551).
- [Shkapenyuk 2002] V. Shkapenyuk, T. Suel, “Design and Implementation of a High-Performance Distributed Web Crawler”, *Proceedings of the 18th International Conference on Data Engineering*, Page 357, 2002.
- [Simon 2004] Simon, G., Maroti, M., Ledeczki, A., Balogh, G., Kusy, B., Nadas, A., Pap, G., Sallai, J., And Frampton, K., “Sensor Network-Based Countersniper System”, In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems. (Sen Sys)*. ACM Press, New York, 2004;
- [Singh 2002] S. P. N. Singh, S. M. Thayer, W. P. Thayer, *A foundation for kilorobotic exploration*, Proc. 2002 Congress on Evolutionary Computation, CEC 2002, Vol.2, 2002, pp. (1033-1038).
- [Smart 2005b] Smart Dust Project, online: <http://www-bsac.eecs.berkeley.edu/archive/users/warneke-brett/SmartDust/>;
- [Smart 2005a] Smart Dust, Autonomous sensing and communication in a cubic millimeter; on-line: <http://robotics.eecs.berkeley.edu/~pister/SmartDust/>;

- [SN 2005] "Intelligent Sensors"; Intelligent Sensors , Sensor Networks and Information Processing ;online: <http://www.sensornetworks.net.au/intsens.html>;
- [Stanescu 2007] D. Stanescu, M. Stratulat, B. Ciubotaru, D. Chiciudean, R. Cioarga, M. V. Micea, *Embedding Data in Video Stream Using Steganography*, in Proc. 4-th Intl. Symp. Applied Computational Intellig. and Informatics, SACI 2007, Timisoara, Romania, May 2007, pp. (241-244), ISBN 1-4244-1234-X, IEEE Cat. No. 07EX1788;
- [Tanenbaum 2002] Andrew S. Tanenbaum & Maarten van Steen. *Distributed Systems: Principles and Paradigms*; Prentice Hall 2002;
- [Truszkowski 2004] W. Truszkowski, M. Hinchey, J. Rash, and C. Rouff; *NASA's swarm missions: The challenge of building autonomous software*. IEEE IT Professional, 6(5):47-52, September/October 2004;
- [Truszkowski 2006] W. F. Truszkowski, M. G. Hinchey, J. L. Rash, and C. A. Rouff. *Autonomous and autonomic systems: A paradigm for future space exploration missions*. IEEE Transactions on Systems, Man and Cybernetics, Part C, 2006.
- [Vaughan 2002] Vaughan R., Stoy K., Sukhatme G., Mataric M., "LOST: Localization-space trails for robot teams.", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, 2002;
- [Wee 2004] Wee Kiat Ng; Leng, G.S.B.; Yee Leong Low, "Coordinated movement of multiple robots for searching a cluttered environment", *Proc. of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004)*.
- [Wu 2001] Jason Wu, W.K. Chan, and G. Thomas, "Rapid and accurate inter-robot position determination in robot teams," *Instrumentation and Measurement, IEEE Transactions on*, vol. 50, no. 1, pp. 163 -168, feb 2001;
- [Yoon 2007] Yoon, S., Veerarittiphan, C., and Sichitiu, M. L., "Tiny-Sync: Tight Time Synchronization for wireless sensor networks", in *ACM Trans. Sens. Netw.* 3, 2, Article 8 (June 2007), 34 pages;
- [Zambonelli 2004] Franco Zambonelli, Marie-Pierre Gleizes, Marco Mamei, Robert Tolksdorf; *Spray Computers: Frontiers of Self-Organization for Pervasive Computing*; Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'04);