INTELIGENȚĂ ARTIFICIALĂ

Sisteme inteligente

Învățare prin întărire

Laura Dioşan

Sumar

A. Scurtă introducere în Inteligenţa Artificială (IA)

B. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

C. Sisteme inteligente
- Sisteme care învaţă singure
  - Arbori de decizie
  - Reţele neuronale artificiale
  - Maşini cu suport vectorial
  - Algoritmi evolutivi
  - *Q-learning*
- Sisteme bazate pe reguli
- Sisteme hibride

Materiale de citit şi legături utile

- capitolul IV.20 din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- capitolul 6 din *H.F. Pop, G. Şerban, Inteligenţă artificială, Cluj Napoca, 2004*
- capitolul 13 din *T. M. Mitchell, Machine Learning, McGraw-Hill Science, 1997*
- capitolul 17 din *C. Groşan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*
- capitolul 9 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*

Învățare automată

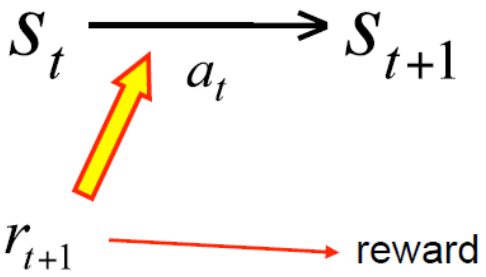
- Învăţarea automată, în funcţie de experienţa pe care se bazează, poate fi:
  - Învăţare supervizată
  - Învăţare nesupervizată
  - Învăţare cu întărire

Învățare prin întărire (reinforcement learning)

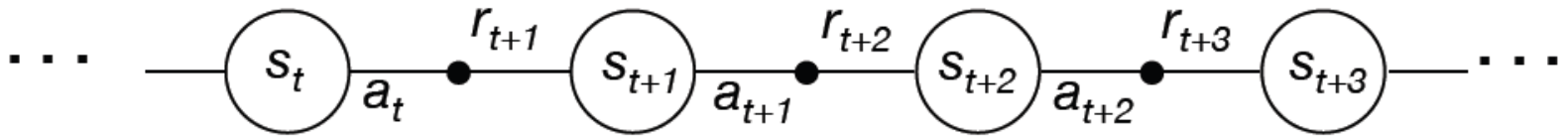
- Ideea de bază
  - Învăţarea din interacţiuni
  - Învăţarea unui mod de acţiune pentru a maximiza o recompensă (numerică)
    - Ex. *Ce recompensa primesc daca fac acest lucru?*
  - Unul sau mai mulţi agenţi care
    - Învaţă permanent conform principiului "încercare şi eroare"
    - Planifică permanent
    - Afectează mediul înconjurător
    - Execută o mulţime de sarcini (acţiuni)

Învățare prin întărire (reinforcement learning)

Definirea problemei



- Un mediu definit print-un set de stări posibile
  - S = {s_1, s_2, ...}
  - Stări temporale
- Un set de acţiuni posibil de efectuat
  - într-o anumită stare A(s_t) = {a_1, a_2, ...}
  - în toate stările A = $u_{st \in S}$ A(s_t)
- O recompensă r_{t+1} obţinută prin deplasarea dintr-0 stare s_t într-o stare s_{t+1} executând o acţiune a_t
- Agentul şi mediul interacţionează la diferite momente de timp t, t =0,1,2,...
  - Agentul observă starea la momentul t: s_t
  - Efectuează o acţiune la momentul t: a_t
  - Primeşte o recompensă: r_{t+1}



Învățare prin întărire (reinforcement learning)

- Aplicaţii
  - Robotică
  - Planificare
  - Jocuri
  - Sisteme de control

Învățare prin întărire (reinforcement learning)

- Învăţare supervizată vs. Învăţare cu întărire
  - Învăţarea supervizată
    - Informaţiile de antrenare → date brute + scopul urmărit
      - Caracteristici ale datelor şi clasele din care fac parte
    - Modul de învăţare
      - Minimizarea erorii
        - Diferenţa între ieşirea corectă şi ieşirea calculată de algoritm
      - Fără interacţiune cu mediul
  - Învăţarea cu întărire
    - Informaţiile de antrenare → date brute + evaluări (recompense/penalizări)
      - Acţiuni posibile şi recompensele corespunzătoare lor
    - Modul de învăţare
      - Optimizarea evaluărilor primite
      - Interacţiune cu mediul

Învățare prin întărire (reinforcement learning)

- Învăţare nesupervizată vs. Învăţare cu întărire
  - Învăţarea nesupervizată
    - Informaţiile de antrenare → date brute, fără rezultate (scop sau recompensă/pedeapsă)
      - Caracteristici ale datelor (fără clasele din care fac parte)
    - Modul de învăţare
      - Minimizarea diferenţelor dintre elementele aceleiaşi clase
      - Maximizarea diferenţelor dintre elementele claselor diferite
      - Fără interacţiune cu mediul
  - Învăţarea cu întărire
    - Informaţiile de antrenare → date brute + evaluări (recompense/penalizări)
      - Acţiuni posibile şi recompensele corespunzătoare lor
    - Modul de învăţare
      - Optimizarea evaluărilor primite
      - Interacţiune cu mediul

Învățare prin întărire (reinforcement learning)

- Scopul unui algoritm de învăţare prin întărire
  - Maparea unor stări în alte stări prin intermediul execuţiei unor acţiuni de interacţiune cu mediul cu scopul maximizării unei recompense

- Elemente
  - Mediul de interacţiune
  - Funcţia de întărire (de recompensare)
  - Funcţia de evaluare (a unei stări)

Învățare prin întărire (reinforcement learning)

- Elemente → Mediul
  - Dinamic
    - Acţiuni de nivel primar
      - Ex. Voltajul motoarelor
    - Acţiuni de nivel înalt
      - Ex. Acceptarea unui job
  - Observabil
    - Citiri de senzori
    - Descrieri simbolice

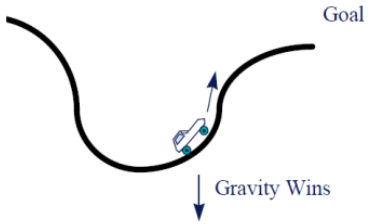
Învățare prin întărire (reinforcement learning)

- Elemente → Funcţia de întărire
  - Are scopul evaluării unei stări în care a ajuns agentul ca urmare a unei acţiuni
  - Măsoară recompensa primită ca urmare a efectuării unei acţiuni
  - Agentul va învăţa să execute acele acţiuni care îi
    - maximizează recompensa sau
    - minimizează penalizarea

Învățare prin întărire (reinforcement learning)

- Elemente → Funcţia de întărire
  - Tipologie
    - Funcţii de tip recompensă întârziată pură (“probleme de evitat”)
      - Orice stare este evaluată cu 0, mai puţin
      - starea finală care este evaluată cu
        - +1, dacă este o stare obiectiv
        - –1, dacă este o stare ne-obiectiv (stare “de evitat”)
      - Ex.
        - jocul de table

    - Funcţii care minimizează timpul necesar atingerii scopului
      - Efectuarea acelor acţiuni care conduc la starea finală pe “drumul cel mai scurt”
      - Evaluare stări
        - 0, pentru starea finală
        - -1, pentru toate celelalte stări
      - Ex
        - Maşina care urcă un deal



    - Funcţii specifice jocurilor
      - Doi sau mai mulţi jucători cu obiective opuse
      - Întărire de tip maximin, minimax
      - Ex
        - Rachete şi avioane ţintă

Învățare prin întărire (reinforcement learning)

- Elemente → Funcţia valoare
  - Utilitate
    - Cum se aleg cele mai bune acţiuni?
    - Cum se poate măsura utilitatea unei acţiuni?
  - Concepte de bază
    - Politică $\pi(s)$ → ce acţiuni trebuie executate din fiecare stare *s*
    - Valoarea unei *stări V(s)* = suma recompenselor primite ca urmare a execuţiei unei politici (set de acţiuni) din acea stare *s* până la o stare finală
  - Definiţie
    - o mapare de la valoarea unei stări la valoarea unei alte stări
    - poate fi aproximată

Învățare prin întărire (reinforcement learning)

- Elemente → Funcţia valoare
  - Valoarea aproximativă a unei stări $V(s_t)$
  - Valoarea optimă a unei stări $V^*(s_t)$

$$V(s_t) = V^*(s_t) + e(s_t)$$

$$V(s_t) = r(s_t) + \gamma V(s_{t+1})$$

    - unde:
      - $e(s_t)$ – eroarea de aproximare
      - $r(s_t)$ – recompensa primită în starea s_t

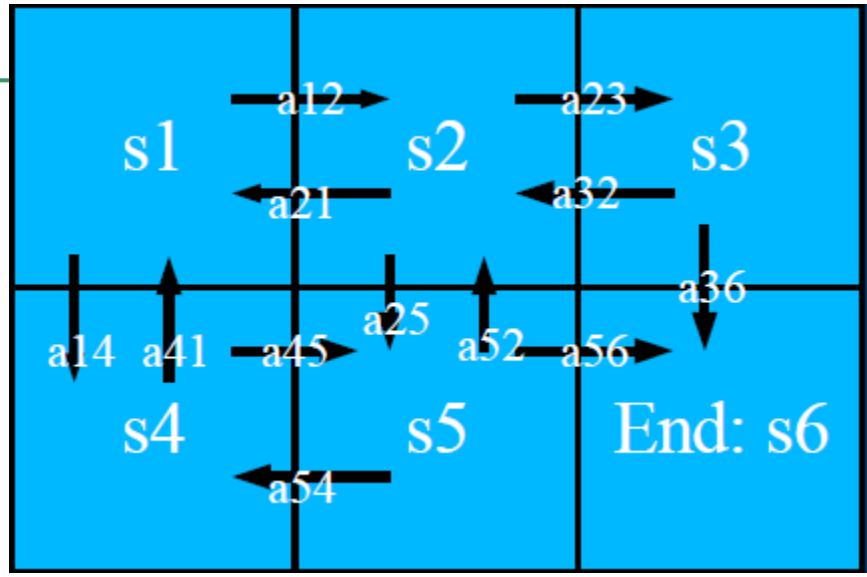
  - Cum se aproximează valoarea unei stări (algoritmi)?
    - Q-learning
    - SARSA (State-Action-Reward-State-Action)

Învățare prin întărire (reinforcement learning)

- Elemente → Funcţia valoare → Q-learning
  - Ideea de bază
    - $Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a')$
    - dintre toate acţiunile posibile a_i se alege acea acţiune care produce cea mai mare valoare $Q(s, a_i)$
  - Algoritm
    - Pentru fiecare pereche (s, a) se reţine valaorea 0 (într-un tablou Q)
    - Fie starea curentă s
    - Repetă
      - Se alege o acţiune *a* şi se execută
      - Se înregistrează recompensa imediată *r*
      - Se observă noua stare s'
      - Se modifică valoarea $Q(s, a) = r + \gamma \max_{a'} Q(s', a')$
      - s = s'
    - Până când se ajunge la un număr de repetări prefixat

Învățare prin întărire (reinforcement learning)
Q-learning (exemplu)

s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0



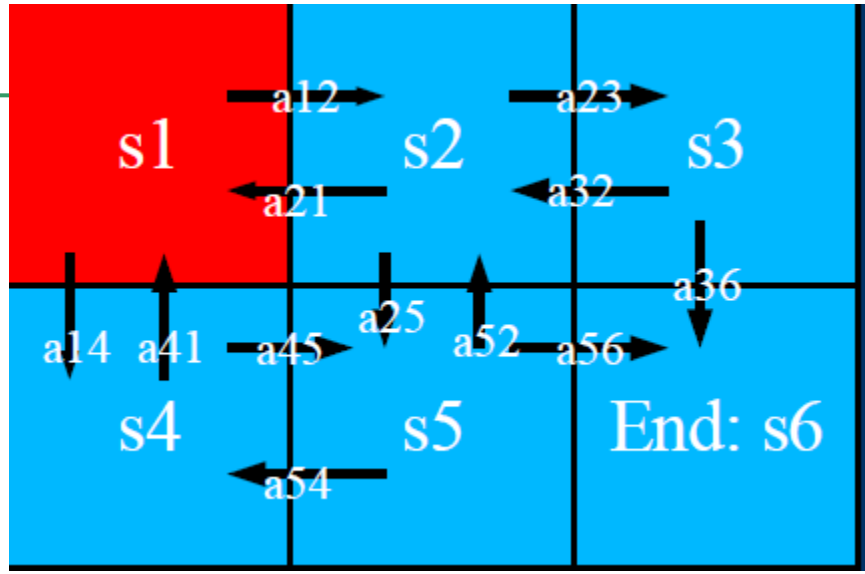
Pp:

- r = 100, dacă se ajunge în s_6
  0, altfel
- γ = 0.5

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)

s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0



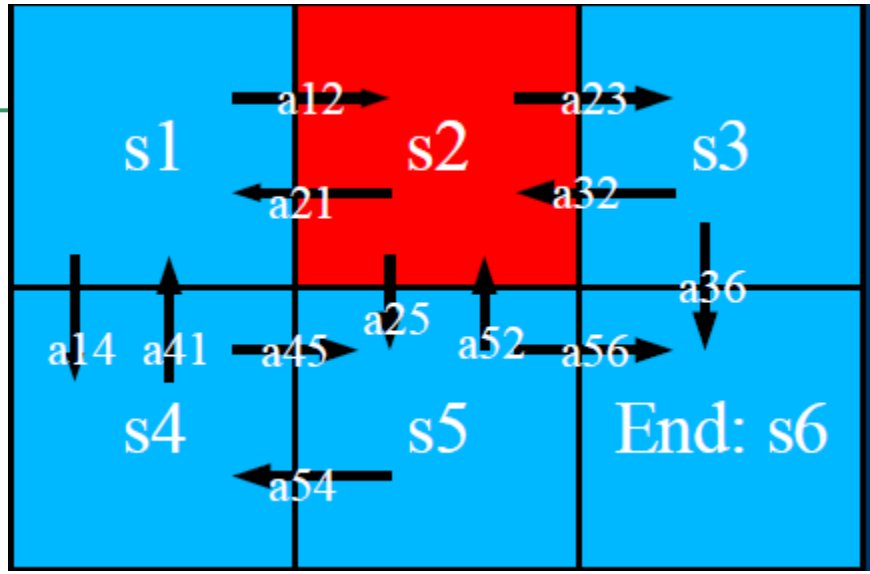
poziţie curentă: s_1

acţiuni posibile: a_{12}, a_{14}

acţiune selectată: a_{12}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

acţiuni posibile din s_2: a_{21},a_{23},a_{25}

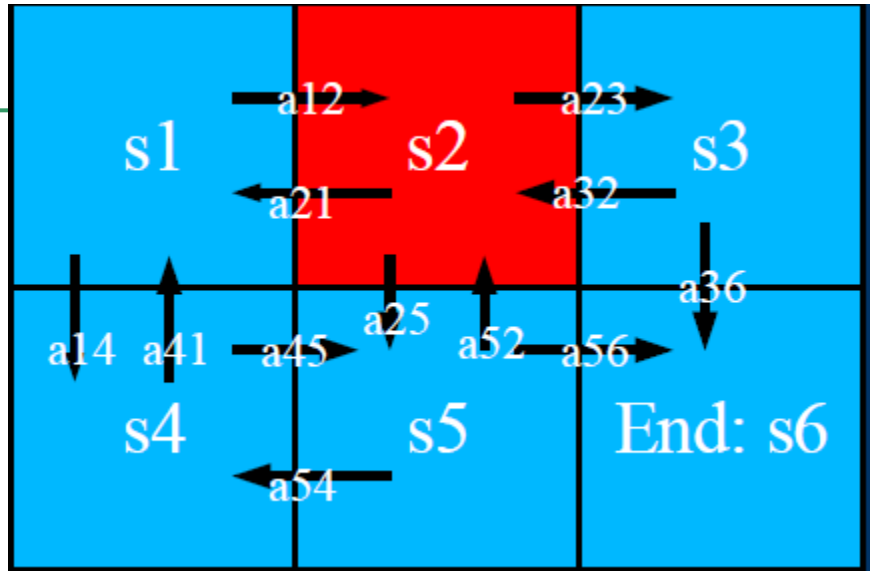
$Q(s_1,a_{12})=r + \gamma * \max\{Q(s_2,a_{21}), Q(s_2,a_{23}), Q(s_2,a_{25})\}$

$Q(s_1,a_{12}) = 0$

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)

s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0



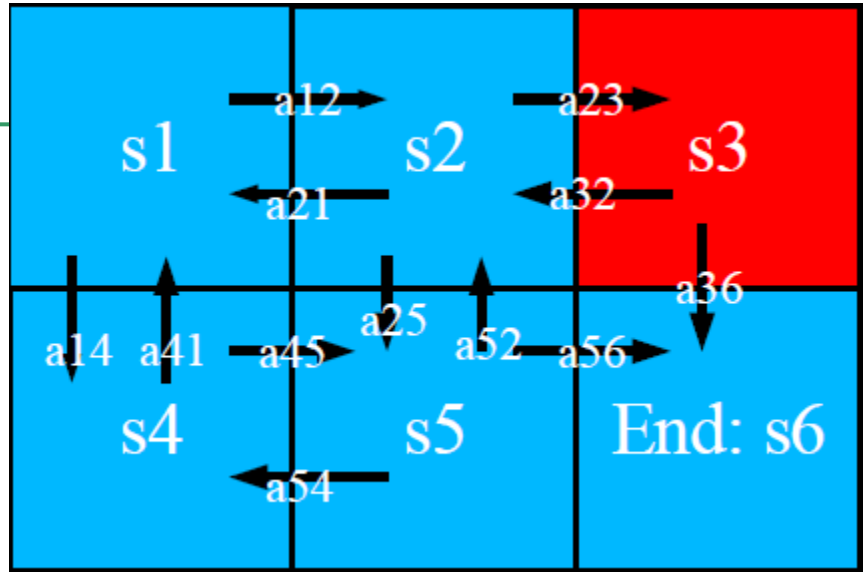
Poziţie curentă: s_2

Acţiuni posibile: a_{21},a_{23},a_{25}

Acţiune selectată: a_{23}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

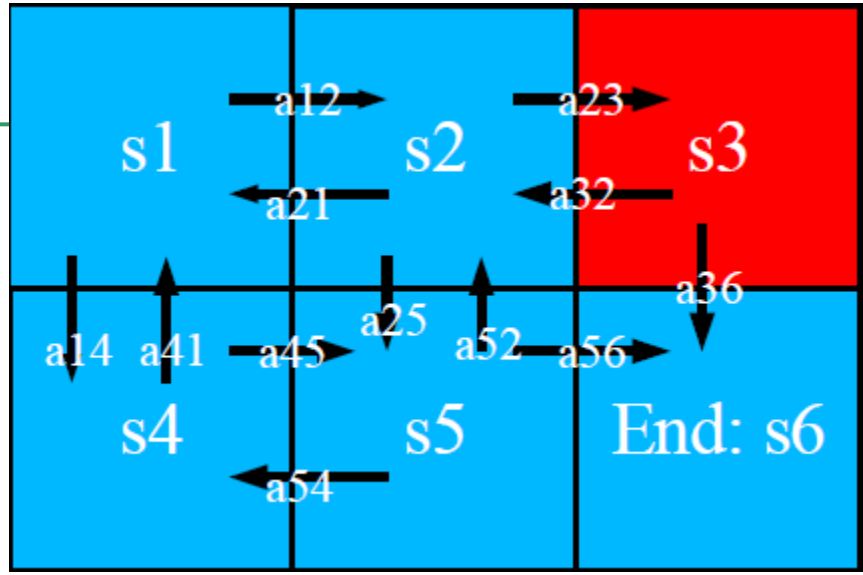
acţiuni posibile din s_3: a_{32},a_{36}

$Q(s_2,a_{23})=r + \gamma * \max\{Q(s_3,a_{32}), Q(s_3,a_{36})\}$

$Q(s_2,a_{23}) = 0$

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	0
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

Poziţie curentă: s_3

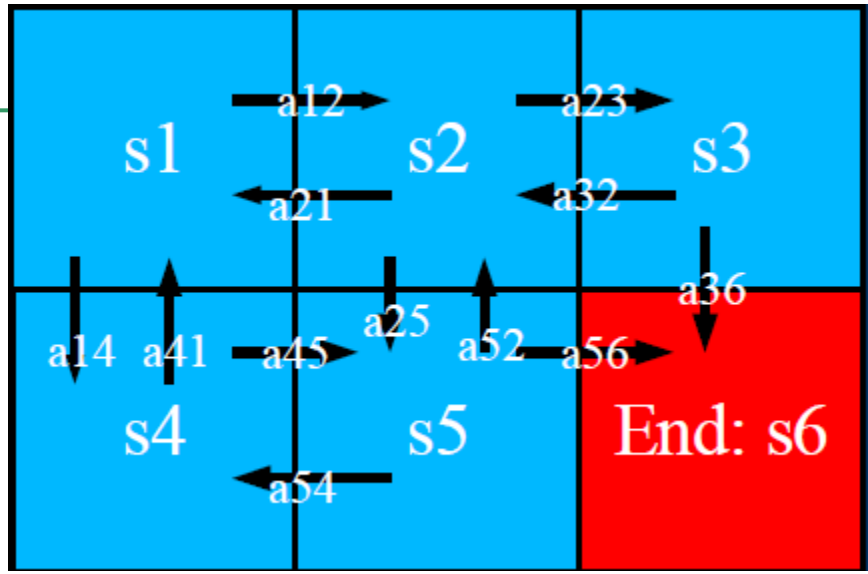
Acţiuni posibile: a_{32}, a_{36}

Acţiune selectată: a_{36}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)

s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0



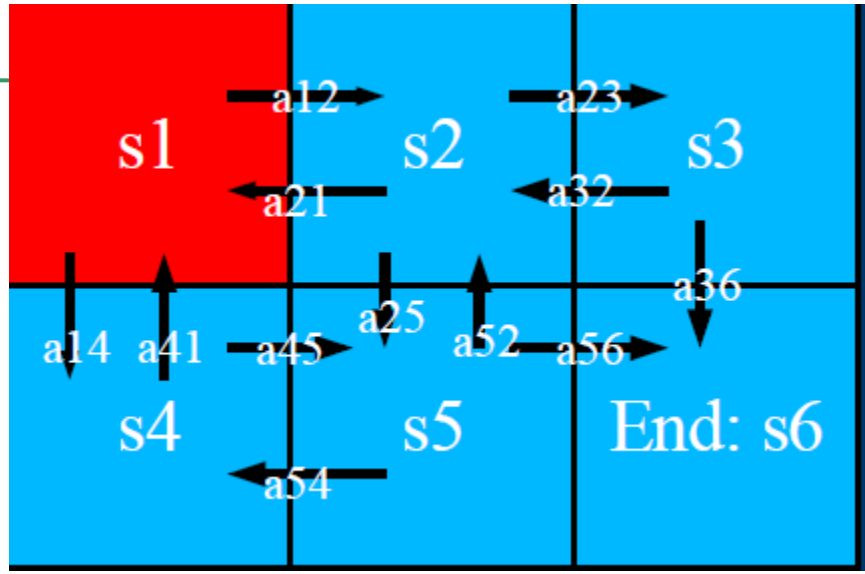
S_6 – stare finală

$Q(s_3,a_{36}) = r$

$Q(s_3,a_{36}) = 100$

Învățare prin întărire (reinforcement learning)
Q-learning (exemplu)

s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0



JOC NOU

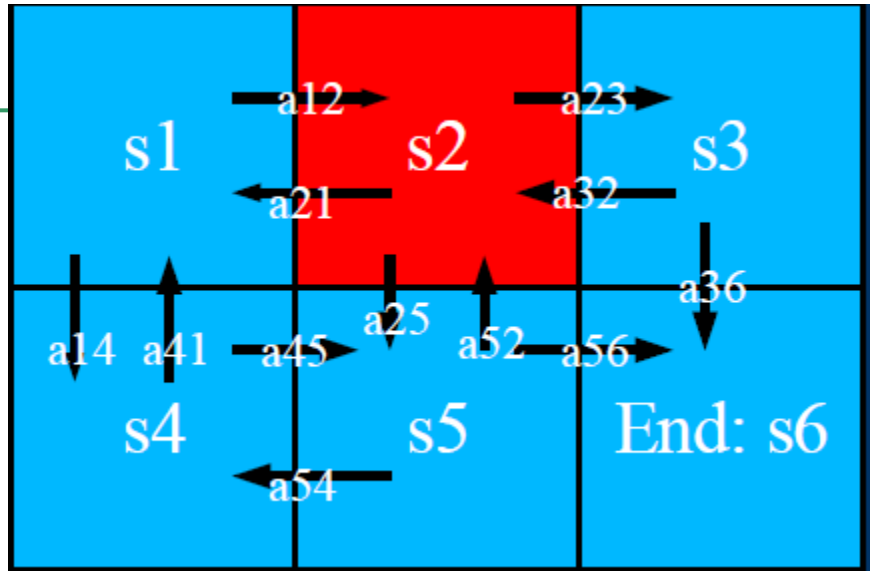
poziţie curentă: s_1

acţiuni posibile: a_{12}, a_{14}

acţiune selectată: a_{12}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

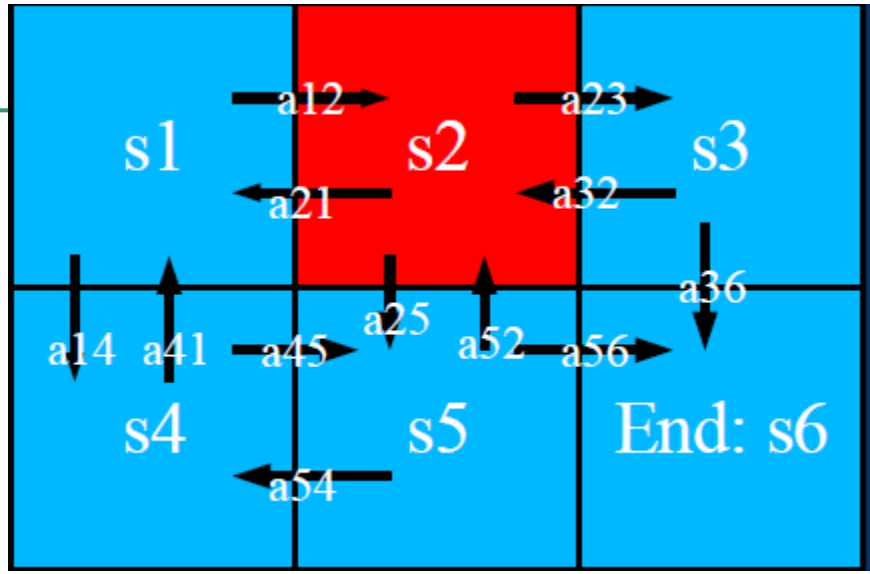
acţiuni posibile din s_2: a_{21},a_{23},a_{25}

$Q(s_1,a_{12})=r + \gamma * \max\{Q(s_2,a_{21}), Q(s_2,a_{23}), Q(s_2,a_{25})\}$

$Q(s_1,a_{12}) = 0$

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

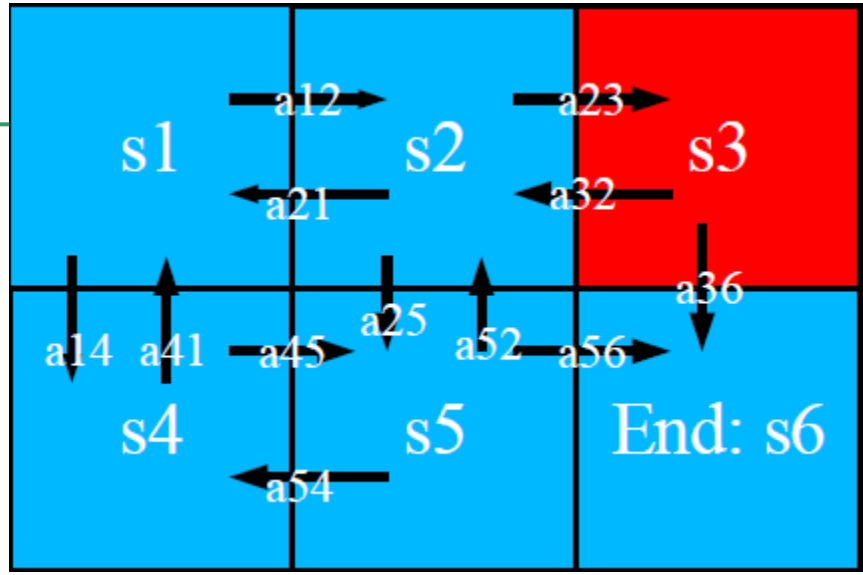
Poziţie curentă: s_2

Acţiuni posibile: a_{21},a_{23},a_{25}

Acţiune selectată: a_{23}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	50
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

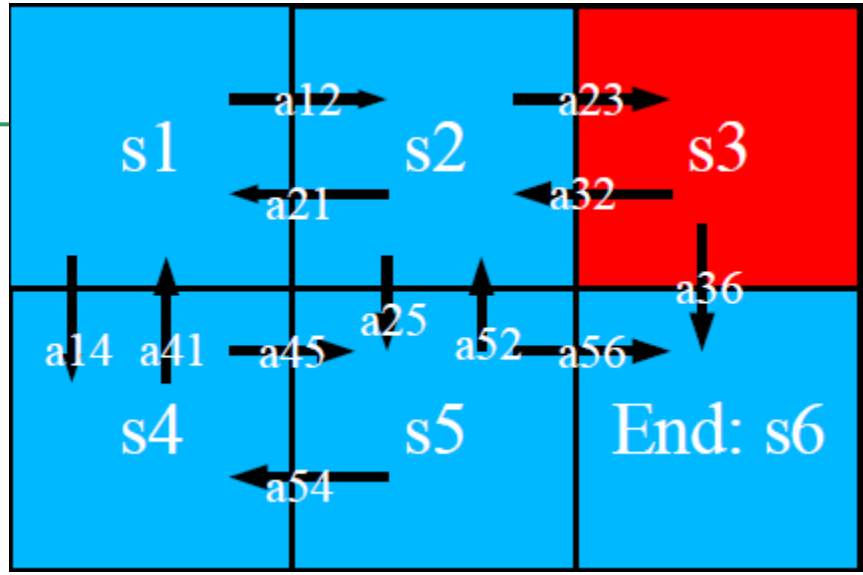
acţiuni posibile din s_3: a_{32},a_{36}

$Q(s_2,a_{23})=r + \gamma * \max\{Q(s_3,a_{32}), Q(s_3,a_{36})\}$

$Q(s_2,a_{23}) = 50$

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	50
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

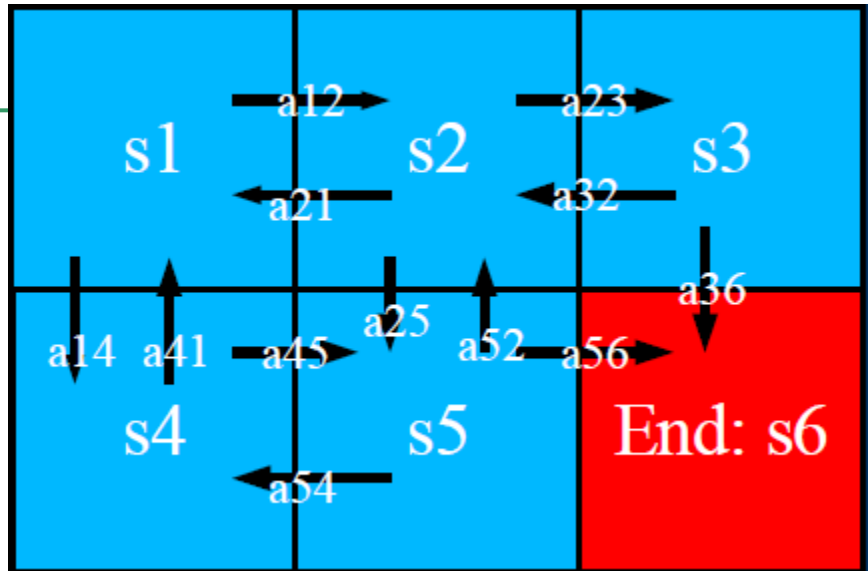
Poziţie curentă: s_3

Acţiuni posibile: a_{32}, a_{36}

Acţiune selectată: a_{36}

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	0
s_1, a_{14}	0
s_2, a_{21}	0
s_2, a_{23}	0
s_2, a_{25}	0
s_3, a_{32}	0
s_3, a_{36}	100
s_4, a_{41}	0
s_4, a_{45}	0
s_5, a_{54}	0
s_5, a_{52}	0
s_5, a_{56}	0

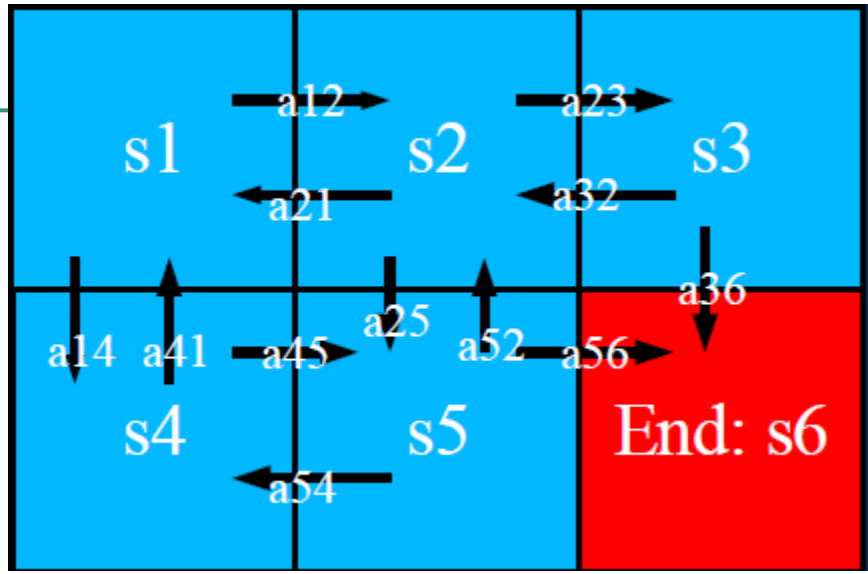
S_6 – stare finală

$Q(s_3,a_{36}) = r$

$Q(s_3,a_{36}) = 100$

Învățare prin întărire (reinforcement learning)

Q-learning (exemplu)



s,a	Q(s,a)
s_1, a_{12}	25
s_1, a_{14}	25
s_2, a_{21}	12.5
s_2, a_{23}	50
s_2, a_{25}	25
s_3, a_{32}	25
s_3, a_{36}	100
s_4, a_{41}	12.5
s_4, a_{45}	50
s_5, a_{54}	25
s_5, a_{52}	25
s_5, a_{56}	100

....

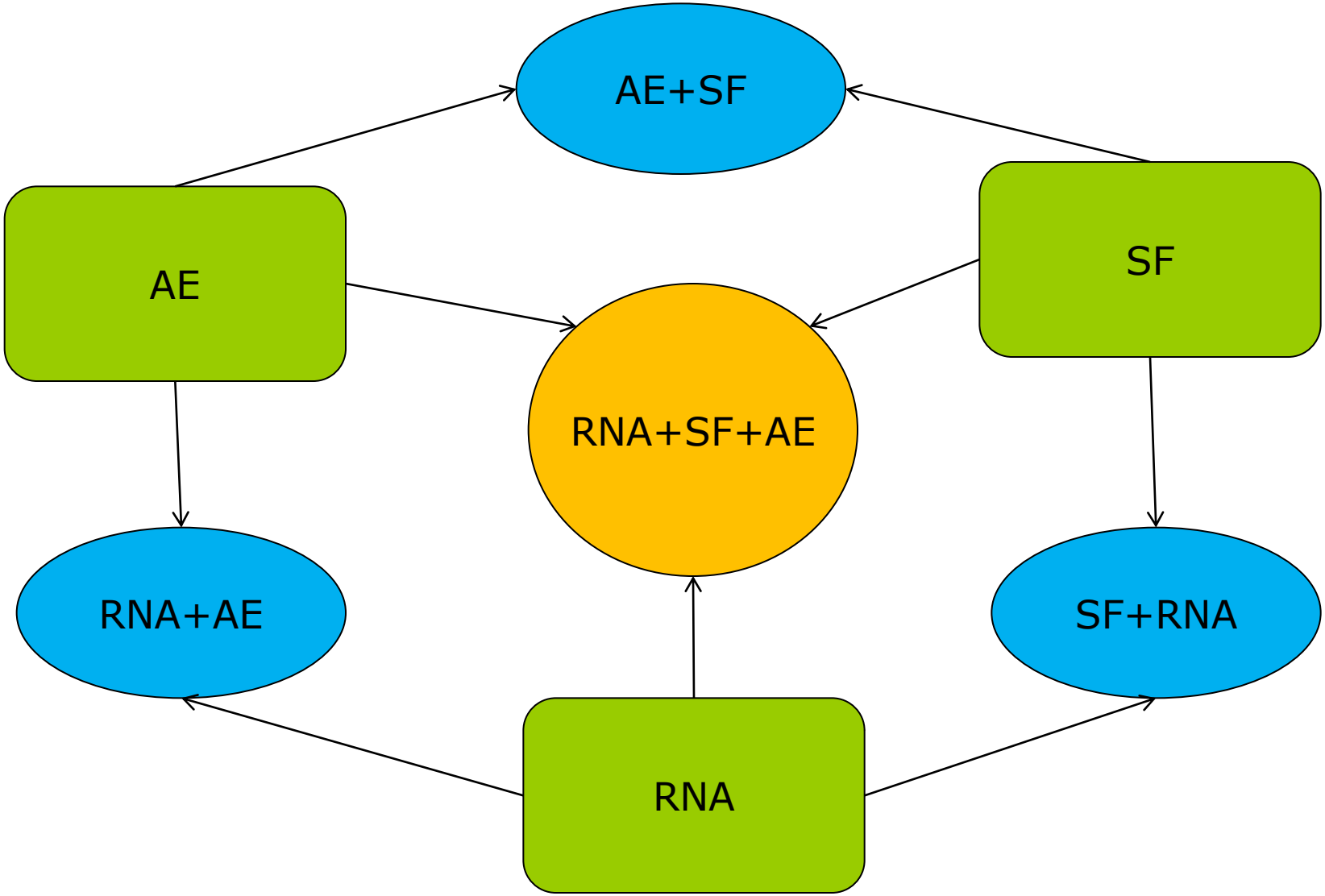
Învățare prin întărire (reinforcement learning)

- Caracteristici
  - Învăţare din recompense
  - Interacţiune cu sarcinile
    - Secvenţe de stări, acţiuni şi recompense
  - Lumi incerte şi nedeterministe
  - Consecinţe întârziate
  - Învăţare direcţionată către ţintă
  - Echilibru între explorare şi exploatare

Sisteme inteligente – sisteme hibride

- O combinaţie între 2 sau mai multe sisteme inteligente
  - Sisteme bazate pe reguli
    - Sisteme fuzzy (SF)
  - Sisteme bazate pe învăţare automată
    - Reţele neuroname artificiale (RNA)
    - Algoritmi evolutivi (AE)

Sisteme inteligente – sisteme hibride



Sisteme inteligente – sisteme hibride

- Sisteme neuronale fuzzy (SF+RNA)
  - Modele cooperative
    - RNA determină parametrii SF
      - funcţiile de apartenenţă → aproximare
    - şi/sau
      - regulile fuzzy → clustering (SOM – Self Organised Map)
    - pe baza datelor de antrenament
  - Modele concurente
    - RNA asistă SF în determinarea parametrilor

Sisteme inteligente – sisteme hibride

- Sisteme evolutive fuzzy (AE+SF)
  - Adaptarea evolutivă a funcţiilor de apartenenţă
    - Cromozomii codează parametrii diferitelor funcţii de apartenenţă trapezoidale, triunghiulare, logistice, Laplace, Gaussian, etc)
    - Necesită existenţa unei baze de reguli
    - Se optimizează performanţa unui motor de inferenţă deja existent
  - Învăţarea evolutivă a regulilor *if-then*
    - Cromozomii codează una sau toate regulile din baza de cunoştinţe → se construieşte un nou motor de inferenţă
    - Complexitate mărită

Sisteme inteligente – sisteme hibride

- Reţele neuronale evolutive (RNA+AE)
  - Adaptarea evolutivă a design-ului RNA
    - Adaptarea evolutivă a ponderilor RNA
    - Adaptarea evolutivă a structurii RNA şi a funcţiilor de activare a nodurilor RNA
    - Adaptarea evolutivă a regulilor de învăţare

Recapitulare

- Sisteme care învaţă singure (SIS)
  - Instruire (învăţare) automata (Machine Learning - ML)
    - Învăţare supervizată → datele de antrenament (stări ale problemei) sunt deja etichetate cu elemente din E, iar datele de test trebuie etichetate cu una dintre etichetele din E pe baza unui model (învăţat pe datele de antrenament) care face corespondenţa date-etichete. Nu există interacţiune cu mediul.
    - Învăţare nesupervizată → datele de antrenament (stări ale problemei) NU sunt etichetate, trebuie învăţat un model de etichetare, iar apoi datele de test trebuie etichetate cu una dintre etichetele identificate de model. Nu există interacţiune cu mediul.
    - Învăţare cu întărire → datele de antrenament (stări ale problemei şi acţiuni posibile) sunt etichetate cu măsuri de recompensă astfel încât să se poată optimiza recompensa primită prin efectuarea anumitor acţiuni. În plus, există interacţiune cu mediul prin intermediul acestor acţiuni.
  - Sisteme hibride
    - O combinaţie de 2 sau mai multe sisteme inteligente
    - Care se "ajută" între ele
      - În optimizarea diferitelor elemente sau algoritmi

Cursul următor

A. Scurtă introducere în Inteligenţa Artificială (IA)

B. Rezolvarea problemelor prin căutare
- Definirea problemelor de căutare
- Strategii de căutare
  - Strategii de căutare neinformate
  - Strategii de căutare informate
  - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
  - Strategii de căutare adversială

C. Sisteme inteligente
- Sisteme care învaţă singure
  - Arbori de decizie
  - Reţele neuronale artificiale
  - Maşini cu suport vectorial
  - Algoritmi evolutivi
  - Q-learning
- Sisteme hibride
- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)

- Informaţiile prezentate au fost colectate din diferite surse de pe internet, precum şi din cursurile de inteligenţă artificială ţinute în anii anteriori de către:
  - Conf. Dr. Mihai Oltean – www.cs.ubbcluj.ro/~moltean
  - Lect. Dr. Crina Groşan - www.cs.ubbcluj.ro/~cgrosan
  - Prof. Dr. Horia F. Pop - www.cs.ubbcluj.ro/~hfpop