

Alte metaeuristici bazate pe populații

- Modelul sistemului imunitar (AIS – Artificial Immune systems)
- Algoritmi evolutivi bazati pe diferențe (DE – Differential Evolution)
- Algoritmi bazați pe estimarea unei distribuții de probabilitate (PMB - Probabilistic Model Building Algorithms)
- Algoritmi memetici (MA – Memetic Algorithms)

Sisteme imunitare naturale/artificiale

Sistem imunitar natural: un sistem complex de componente celulare și moleculare având rolul de a identifica ceea ce este propriu organismului și de a-l apăra împotriva organismelor și substanțelor străine (agenții patogeni de tipul microbilor și virușilor).

Sistem imunitar artificial (AIS – Artificial Immune Systems): sistem adaptiv inspirat de imunologie și aplicat în rezolvarea unor probleme complexe [De Castro and Timmis, 2002]

Sisteme imunitare artificiale

Scurt istoric: domeniu inițiat la mijlocul anilor 1980

- 1990 – Ishida prima utilizare a algoritmilor inspirați de imunologie în rezolvarea problemelor
- Mijlocul anilor 1990:
 - Forrest et al: aplicații în securitatea calculatoarelor
 - Hunt et al: aplicații în analiza datelor
- Inceputul anilor 2000
 - deCastro, Timmis: optimizare multimodală
- Tendința actuală: modelarea caracteristicilor sistemului biologic

Caracteristici

Sistem imunitar natural

- Specific fiecărui individ
- Distribuit
- Detecție a anomaliilor
- Invățare/adaptare
- Memorie
- Extragere caracteristici



Sistem imunitar artificial

- Robustețe
- Scalabilitate
- Flexibilitate

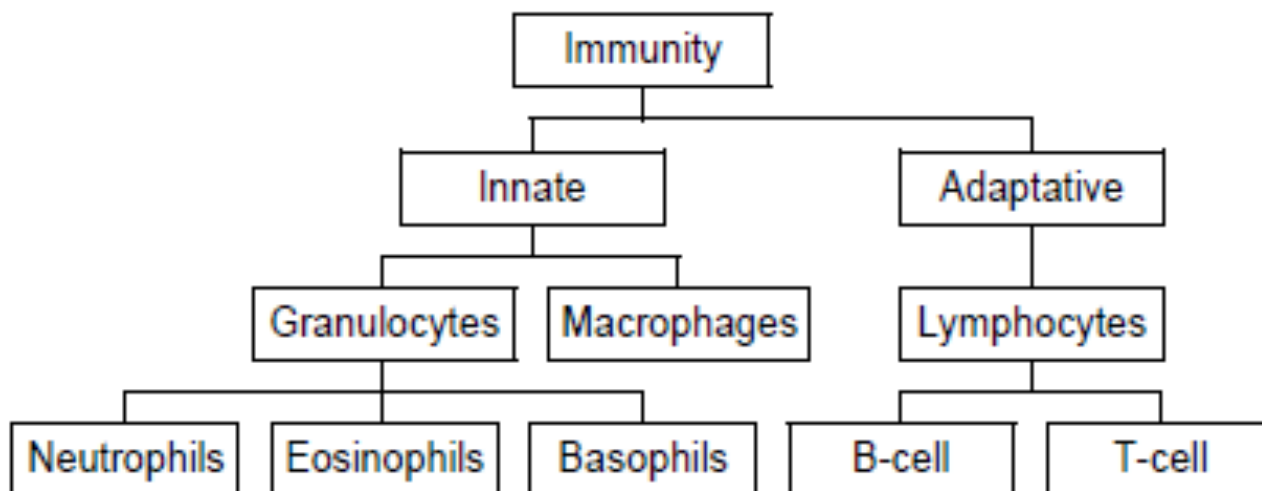
Aplicații

1. Detecție anomalii și securitatea sistemelor informaționale;
2. Analiza datelor (clasificare, recunoaștere forme, clustering etc)
3. Planificare;
4. Căutare și optimizare;
5. Auto-organizare și control autonom;

Sistem imunitar natural

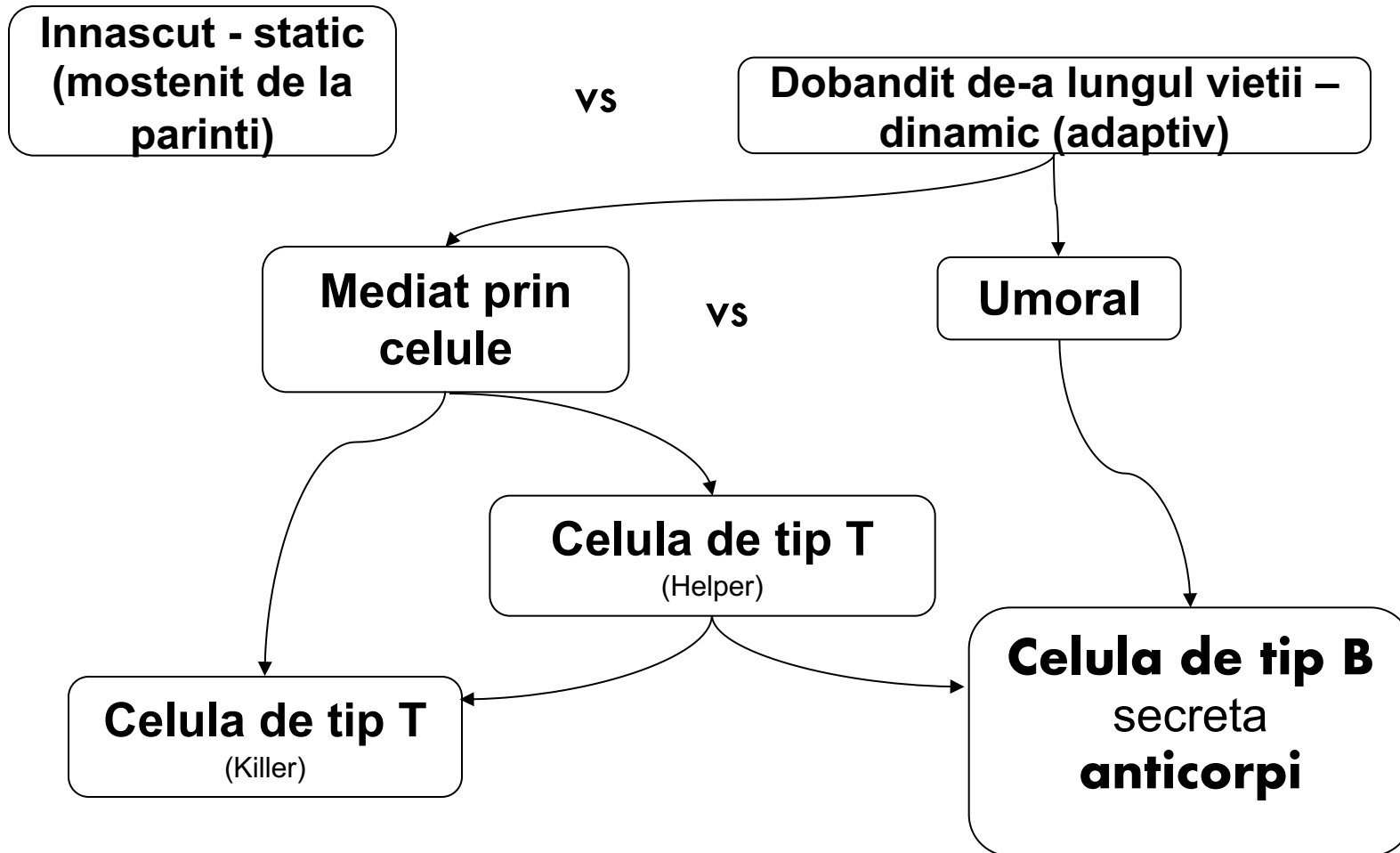
Specificul sistemului imunitar natural: două componente:

- Innăscut (moștenit de la părinți) – se bazează pe granulocite (neutrofile, eozinofile și bazofile) și celule macrofage
- Dobândit pe parcursul vieții – se bazează pe limfocite (celule B și celule T)



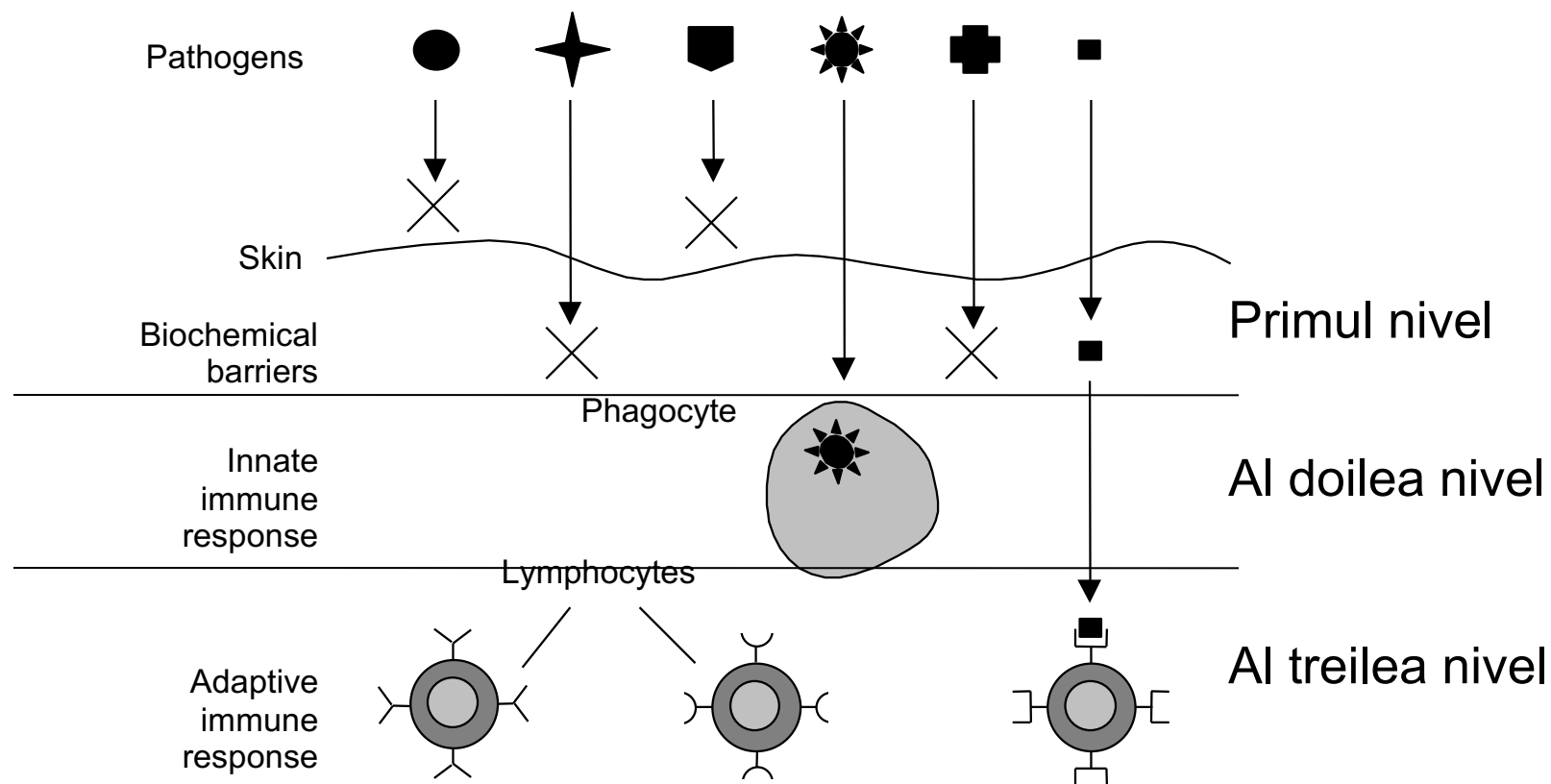
Sistem imunitar natural

Specificul sistemului imunitar natural



Sistem imunitar natural

Specificul sistemului imunitar natural – diferite nivele de acțiune



Sistem imunitar natural

Componenta adaptivă a sistemului imunitar (acționează la al treilea nivel) se caracterizează prin abilități de:

- **Memorare** (capacitatea de a-și “reaminti” de contactul anterior cu agenți patogeni și de a reacționa mai rapid la un nou contact)
- **Invățare** (capacitatea de a recunoaște agenți patogeni neîntâlniți anterior)
- **Elementele active: limfocite**
 - Conțin receptori specifici pentru recunoașterea antigenilor (organismele conțin un repertoriu de milioane de receptori)
 - Sunt de două tipuri:
 - **Celule de tip B**
 - Sintetizate în măduva oaselor (bone marrow)
 - Conțin receptori numiți anticorpi – recunoașterea se bazează pe complementaritatea între regiunea de legare (binding region sau paratop) a anticorpului și o regiune specifică a antigenului (epitop)
 - **Celule de tip T**: sintetizate de către glanda numită timus

Sistem imunitar natural

Mecanisme principale:

Selecție negativă: cenzurarea celulelor de tip T al căror rol este să identifice ce este propriu organismului (definesc comportamentul normal)

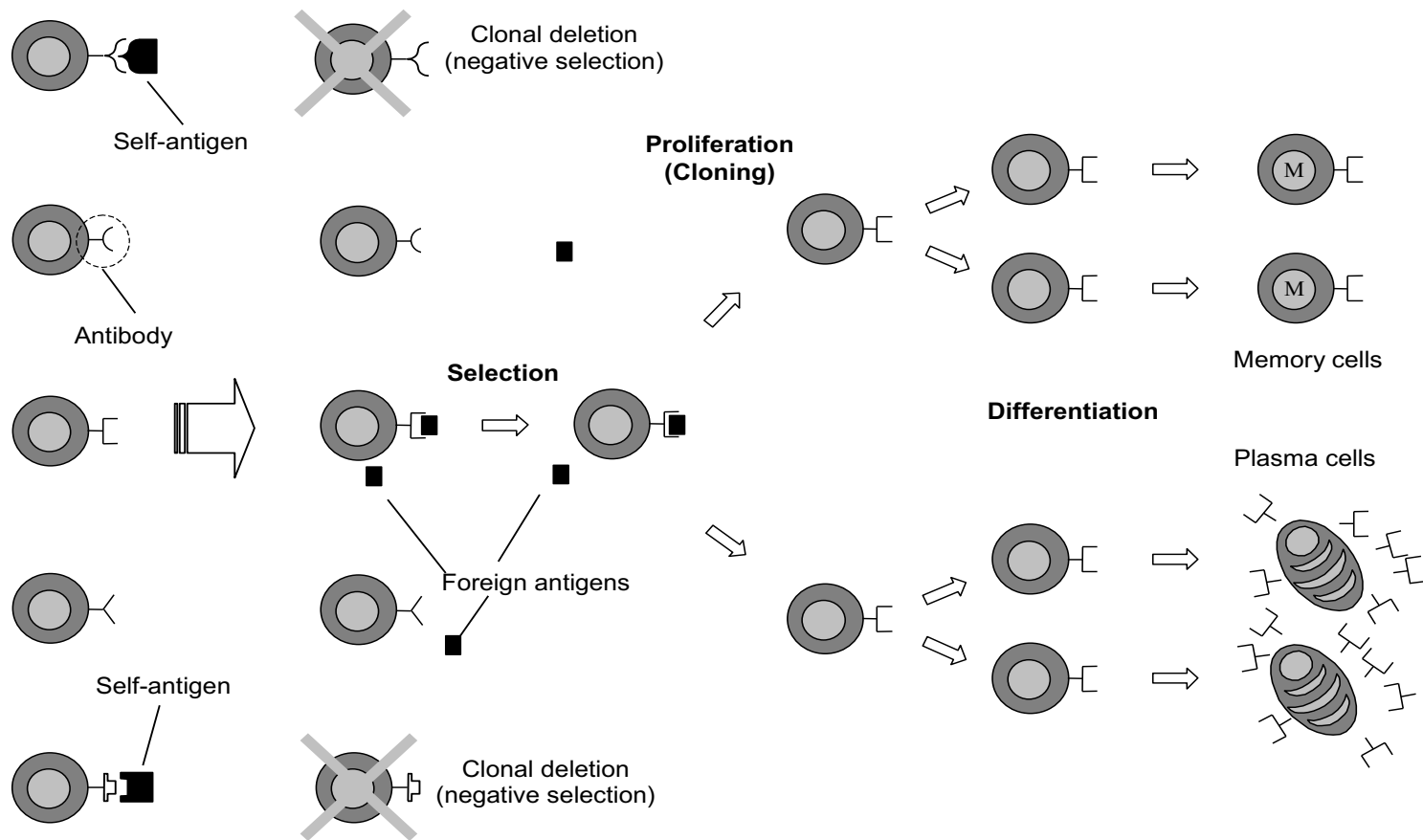
Selecție clonală: proliferarea și diferențierea celulelor care au recunoscut un antigen (învățare și generalizare)

Maturizarea afinității: afinitatea celulelor (de tip B) care au recunoscut un antigen este “întărită” prin

- Mutații asupra receptorilor (probabilitatea de mutație este invers proporțională cu afinitatea)
- Stocarea celulelor cu afinitate crescută într-un “bazin” de celule cu memorie (celule B cu rol de memorie)
- Eliminarea celulelor cu comportament incorect

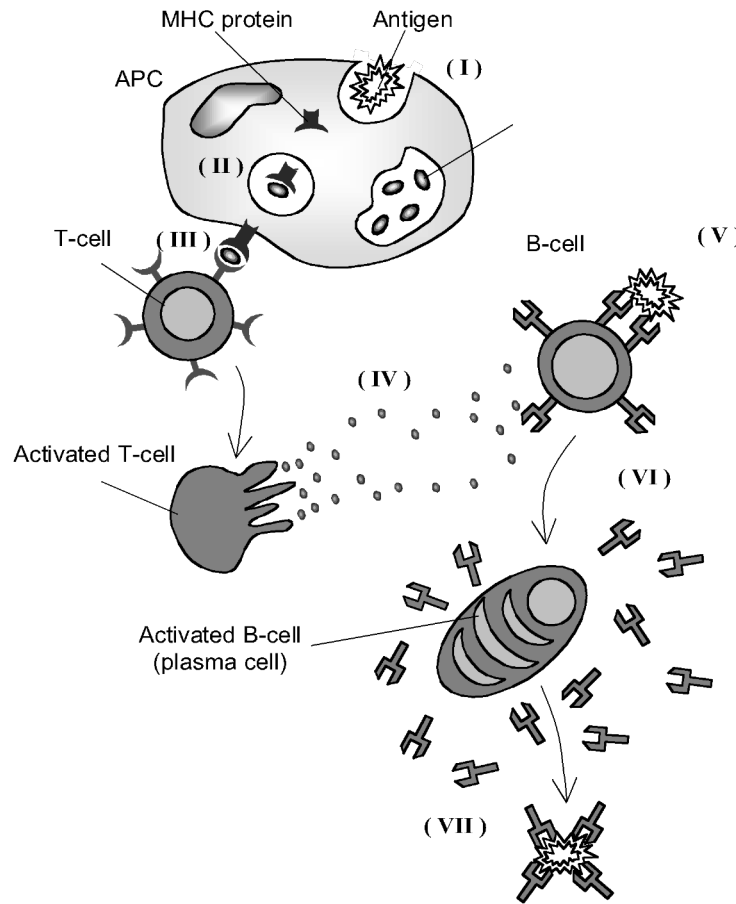
Sistem imunitar natural

Mecanisme principale:



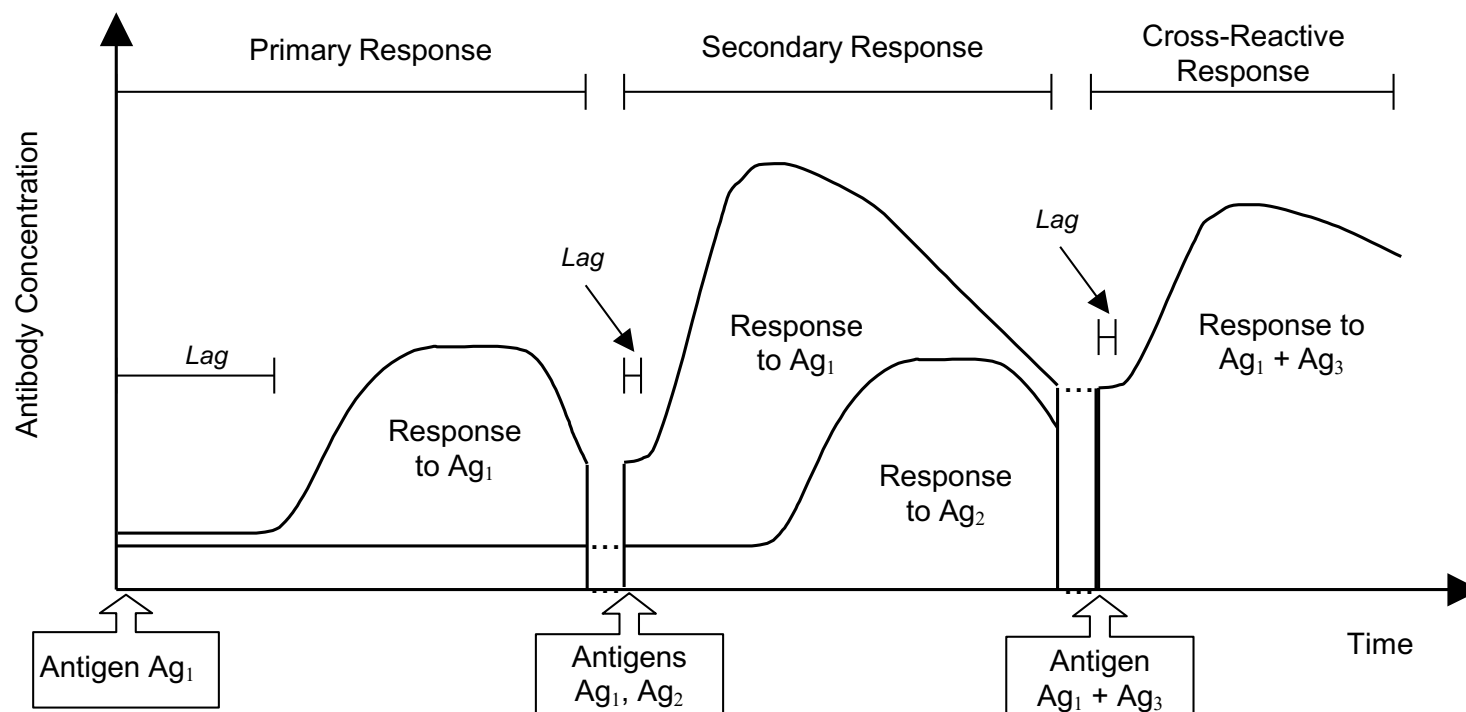
Sistem imunitar natural

Modul de acțiune al sistemului imunitar natural



Sistem imunitar natural

Modul de acțiune al sistemului imunitar natural



Reacție primară: primul răspuns la atacul unui antigen

Reacție secundară: reacție mai rapidă bazată pe rememorarea atacurilor anterioare

Sistem imunitar artificial

Principiul rezolvării problemelor cu AIS:

Problema de rezolvat = mediul in care este plasat organismul

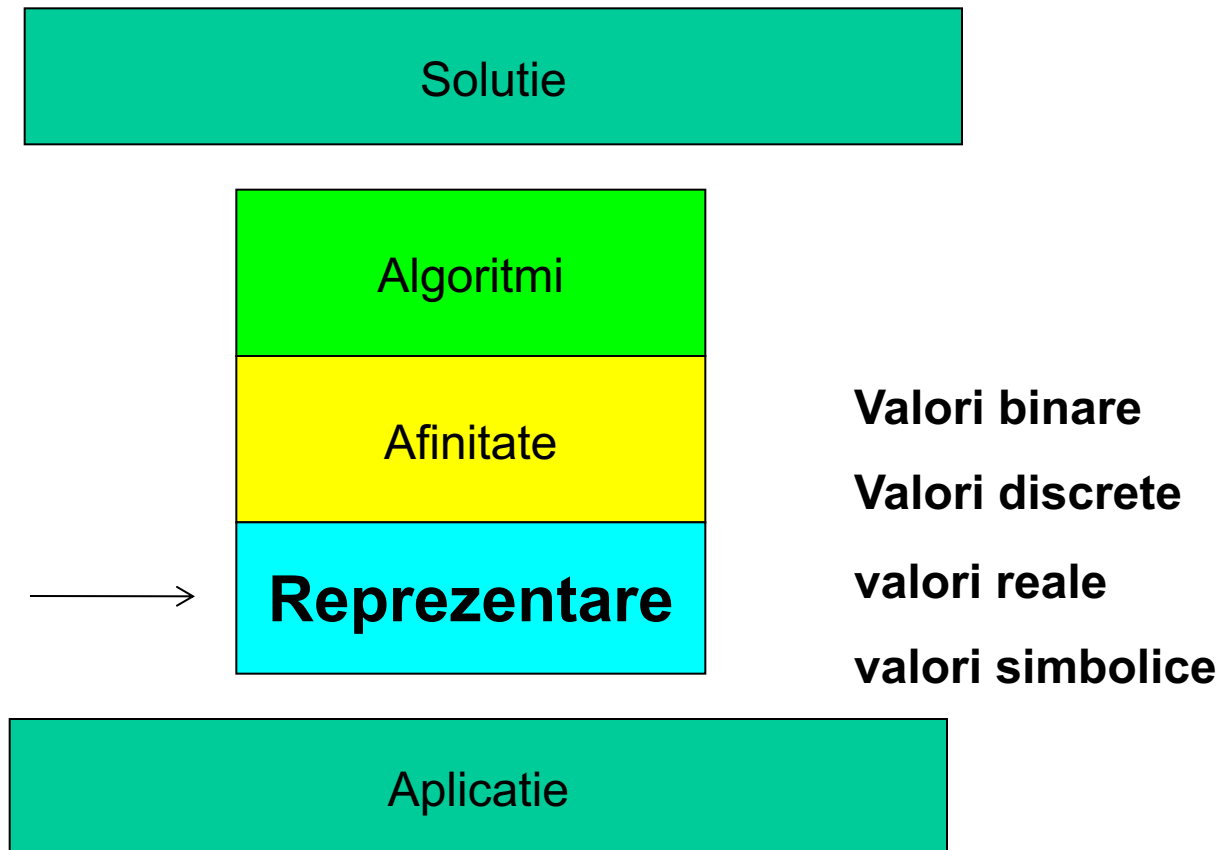
Soluția problemei = antigen

Estimator al soluției (element al populației) = anticorp

Măsură a calității unui estimator = afinitate

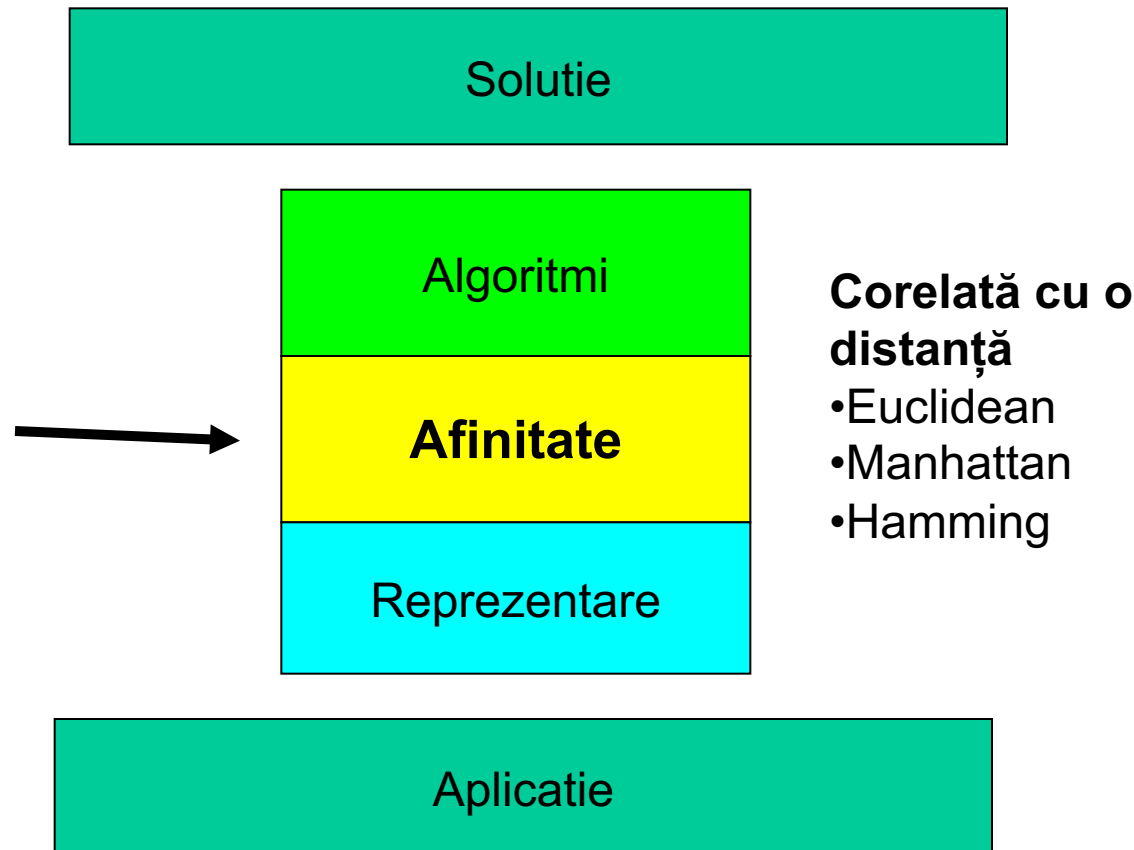
Sistem imunitar artificial

Principiul rezolvării problemelor cu AIS[DeCastro, Timmis, 2002]



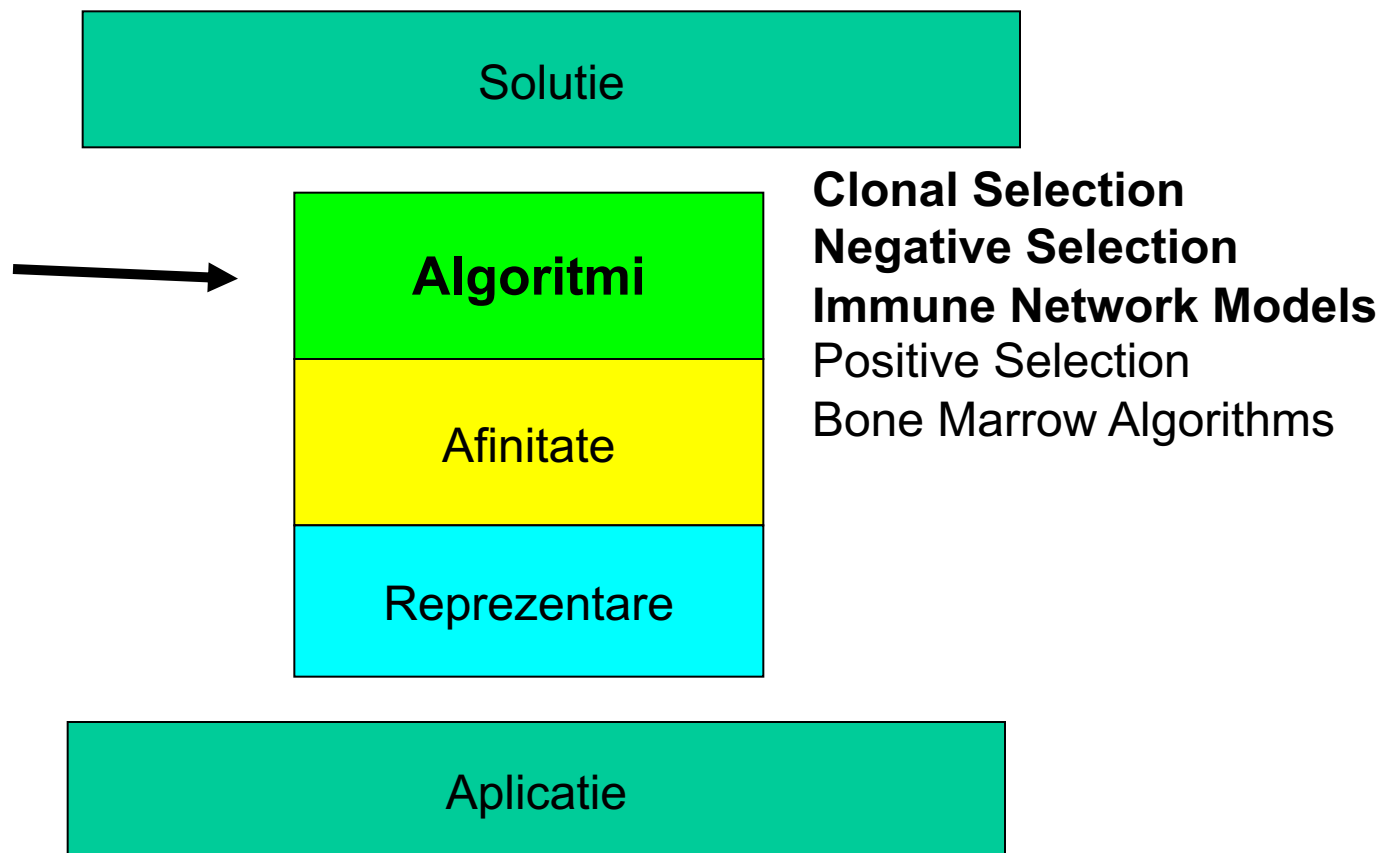
Sistem imunitar artificial

Principiul rezolvarii problemelor cu AIS[DeCastro, Timmis, 2002]



Sistem imunitar artificial

Principiul rezolvarii problemelor cu AIS [DeCastro, Timmis, 2002]



Sistem imunitar artificial

Algoritmul CLONALG (Selecție clonală)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinității)
- b. Clonal selection and expansion (selecție clonală și multiplicare)
- c. Affinity maturation (maturizarea afinității)
- d. Metadynamics (modificare prin mutație aleatoare)

UNTIL “conditie de oprire”

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

- Creează o populație de indivizi (anticorpi)

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinității)
- b. Clonal selection and expansion (selecție clonală și multiplicare)
- c. Affinity maturation (maturizarea afinității)
- d. Metadynamics (modificare prin mutație aleatoare)

UNTIL “conditie de oprire”

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinității)
- b. Clonal selection and expansion (selecție clonala si multiplicare)
- c. Affinity maturation (maturizarea afinitatii)
- d. Metadynamics (modificare prin mutatie aleatoare)

UNTIL “conditie de oprire”

Pentru fiecare șablon antigenic (dată din setul de intrare sau element al populației) se efectuează prelucrările a-d

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinității)
- b. Clonal selection and expansion (selecție clonală și multiplicare)
- c. Affinity maturation (maturizarea afinității)
- d. Metadynamics (modificare prin mutație aleatoare)

UNTIL “conditie de oprire”

Calculează afinitatea

- a) Pb de analiză a datelor: afinitatea e cu atât mai mare cu cât similaritatea dintre data de intrare (antigen) și elementul populației (anticorp) este mai mare
- b) Pb de optimizare: afinitatea e cu atât mai mare cu cat valoarea fitness-ului (corelată cu funcția obiectiv) este mai mare

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

a. Affinity evaluation (evaluarea afinitatii)

b. Clonal selection and expansion (selectie clonala si multiplicare)

c. Affinity maturation (maturizarea afinitatii)

d. Metadynamics (modificare prin mutatie aleatoare)

UNTIL "conditie de oprire"

- Selectează n elemente din P în ordinea descrescătoare a afinității
- Generează pt. fiecare element selectat din P un număr de clone direct proporțional cu afinitatea.

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinitatii)
- b. Clonal selection and expansion (selectie clonala si multiplicare)
- c. Affinity maturation (maturizarea afinității)
- d. Metadynamics (modificare prin mutație aleatoare)

UNTIL “conditie de oprire”

- Aplică mutație fiecărei clone
- Rata de mutație e invers proporțională cu afinitatea
- Se adaugă indivizii obținuți prin mutație la populație
- Se evaluează afinitatea pentru indivizii adăugați și cel cu afinitatea maximă este memorat

Sistem imunitar artificial

Algoritmul CLONALG (Selectie clonala)

Initializare

REPEAT

Antigenic presentation (contact cu antigenul)

- a. Affinity evaluation (evaluarea afinitatii)
- b. Clonal selection and expansion (selectie clonala si multiplicare)
- c. Affinity maturation (maturizarea afinitatii)
- d. Metadynamics (modificare prin mutatie aleatoare)

UNTIL "conditie de

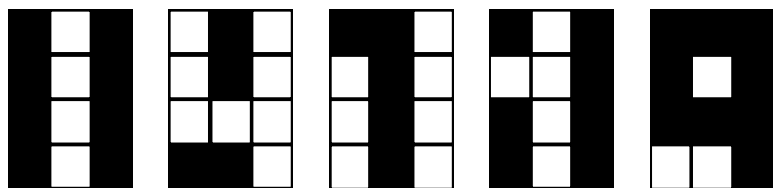
- O parte dintre indivizii cu afinitate mică sunt înlocuiți cu elemente generate aleator

Sistem imunitar artificial

Aplicații ale algoritmului CLONALG (Selecție clonală)

- Recunoaștere forme = generare “detectori” pentru recunoașterea unor simboluri reprezentate prin bitmap-uri

Obs: afinitatea se măsoară folosind distanța Hamming

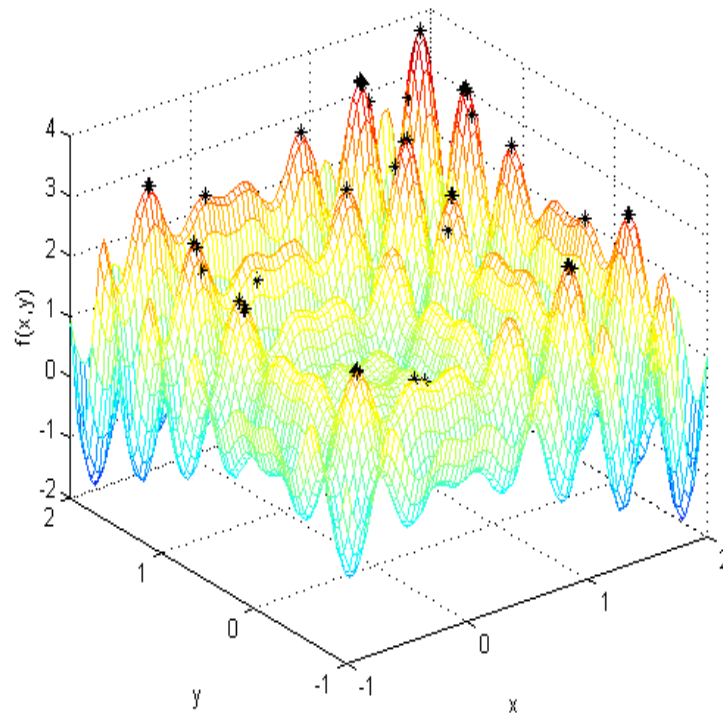


$$\mathbf{P} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Sistem imunitar artificial

Aplicatii ale algoritmului CLONALG (Selectie clonala)

- Optimizare multi-modală = identificarea tuturor optimelor (globale și locale) ale unei funcții



Sistem imunitar artificial

Proprietăți ale algoritmului CLONALG (Selecție clonală)

- Structura generală este similară cu cea a unui algoritm evolutiv (în ipoteza că rolul fitness-ului este transferat măsurii de afinitate)
- Elementele specifice se referă la:
 - Procesul de clonare controlat de valoarea afinității
 - Probabilitatea de mutație este invers proporțională cu valoarea afinității
 - Elementele cu afinitate mică sunt înlocuite cu elemente aleatoare

Sistem imunitar artificial

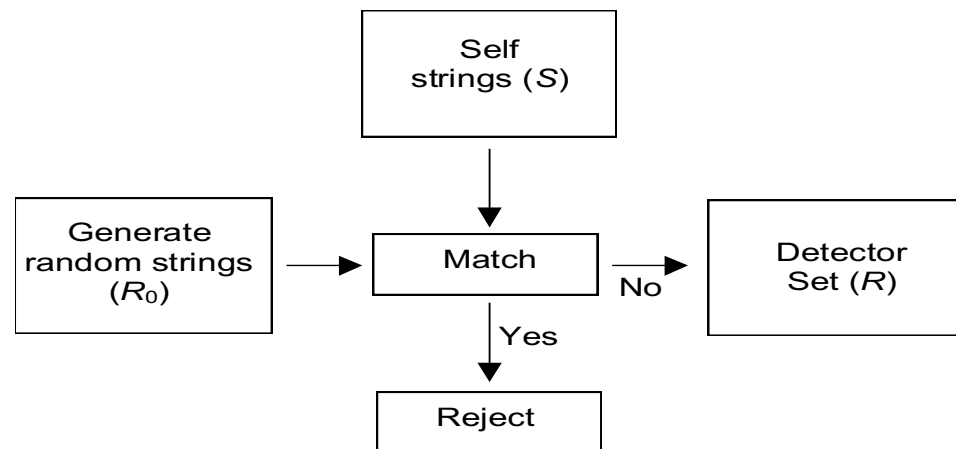
Algoritm de selecție negativă

- Se bazează pe principiul discriminării dintre propriu (self) și străin (non-self)
- Elementele de tip “propriu” se consideră ca fiind reprezentări ale comportamentului normal al unui sistem – aceste reprezentări formează un set S
- Scopul algoritmului este să genereze un set de detectori D care nu se potrivesc cu elementele din S (detectori ai elementelor străine – corespund unor comportamente anormale)
- Algoritmul monitorizează comportarea sistemului pentru a detecta apariția în S a unor elemente care se potrivesc cu detectorii din D - reprezintă semnalul unui comportament anormal al sistemului

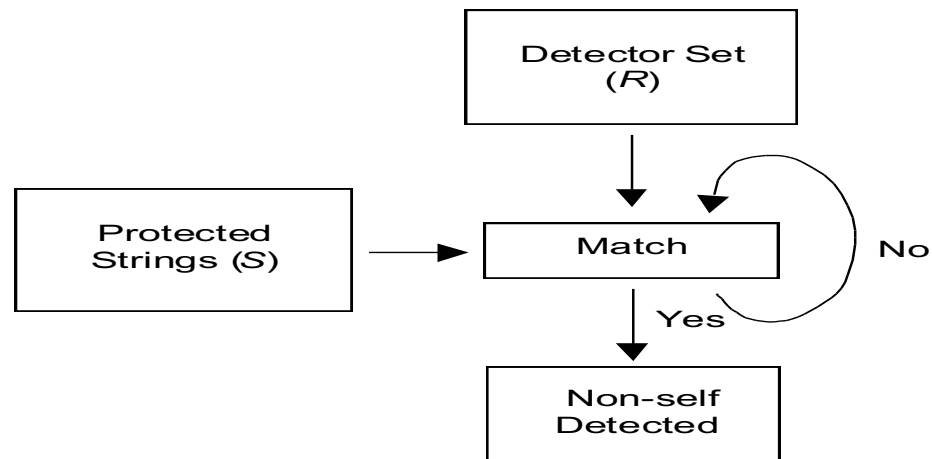
Sistem imunitar artificial

Algoritm de selecție negativă

1. Generare set detectori



2. Monitorizare sistem



Aplicații: securitatea calculatoarelor (detcție intruși) – aplicabilitatea este totusi limitată

Sistem imunitar artificial

Algoritm de selecție negativă

```
input  :  $S_{seen}$  = set of seen known self elements
output:  $D$  = set of generated detectors
begin
  repeat
    Randomly generate potential detectors and place them in a set  $P$ 
    Determine the affinity of each member of  $P$  with each member of
    the self set  $S_{seen}$ 
    If at least one element in  $S$  recognises a detector in  $P$  according to a
    recognition threshold, then the detector is rejected, otherwise it is
    added to the set of available detectors  $D$ 
  until Until stopping criteria has been met
end
```

J. Timmis, P. Andrews, N. Owens, E. Clark – An Interdisciplinary Perspective of Artificial Immune Systems, Evolutionary Intelligence, Volume 1, Number 1, 5-26, 2008

Sistem imunitar artificial

Algoritmul aiNET

Initializare

REPEAT

- Antigenic presentation (contact cu antigenul)
 - a. Affinity evaluation (evaluarea afinității)
 - b. Clonal selection and expansion (selecție clonală și multiplicare)
 - c. Affinity maturation (maturizarea afinității)
 - d. Metadynamics (modificare prin mutație aleatoare)
 - e. Clonal suppression (eliminarea clonelor cu afinitate mica)
- Network interactions (analiza interacțiunilor dintre anticorpii rețelei = calcul afinitate între perechi de anticorpi)
- Network suppression (eliminarea anticorpilor similari cu alți anticorpi)
- Diversity (introducerea unor anticorpi aleatori)

UNTIL “conditie de oprire”

Sistem imunitar artificial

Proprietati ale algoritmului aiNET:

- aiNET este in mare parte similar cu algoritmul CLONALG cu deosebirea ca utilizează un mecanism de supresie bazat pe afinitatea dintre elementele populatiei
- aiNET a fost initial utilizat pentru rezolvarea problemelor de grupare a datelor (ulterior s-a arătat ca are dificultati in gruparea datelor cu distribuție neuniformă)
- aiNET a fost aplicat cu succes in rezolvarea problemelor de optimizare multimodală

Sistem imunitar artificial

aiNET (analogia cu sistemul imunitar natural)

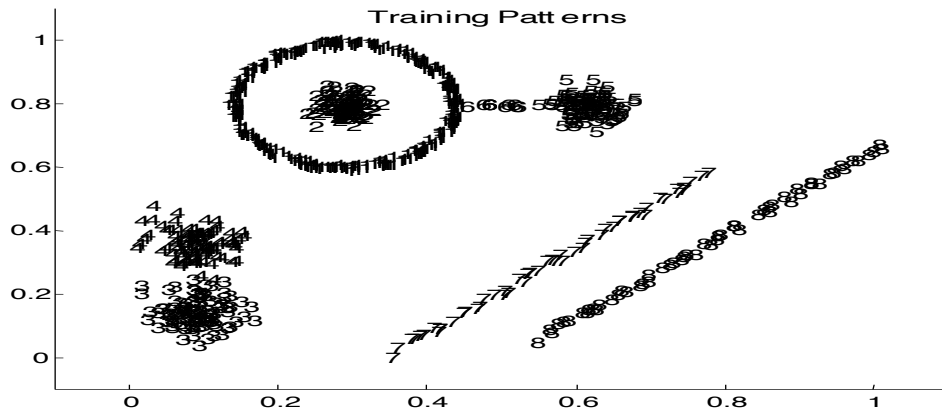
Sistem imunitar	aiNET
Anticorp	Data internă (soluție potențială a problemei, model, element al populației)
Antigen	Data de antrenare
Afinitate	Măsură a calității unui element
Clonarea unei celule	Duplicarea datelor interne
Hipermutatie somatica	Mutație invers proporțională cu afinitatea
Rețea imuna	Rețea de date interne
Metadinamica	Eliminarea/crearea unor date interne

<http://www.aickelin.com>

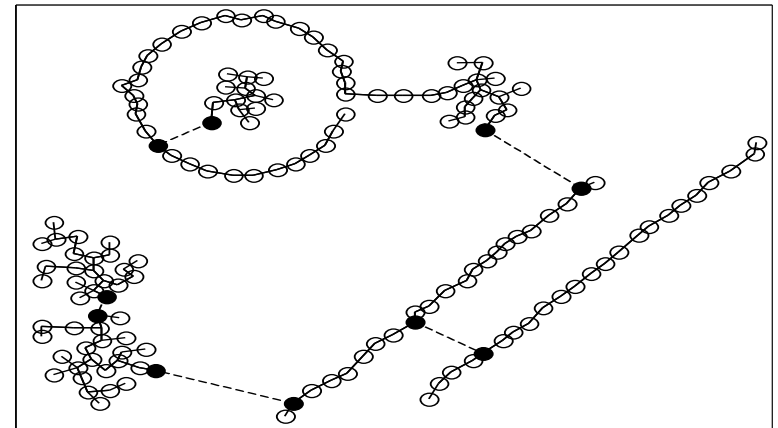
Sistem imunitar artificial

aiNET - aplicatii in clustering

Training Pattern



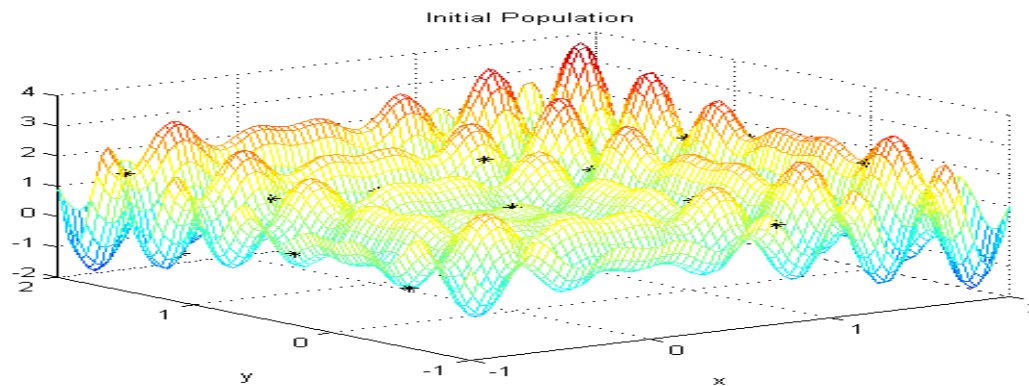
Result immune network



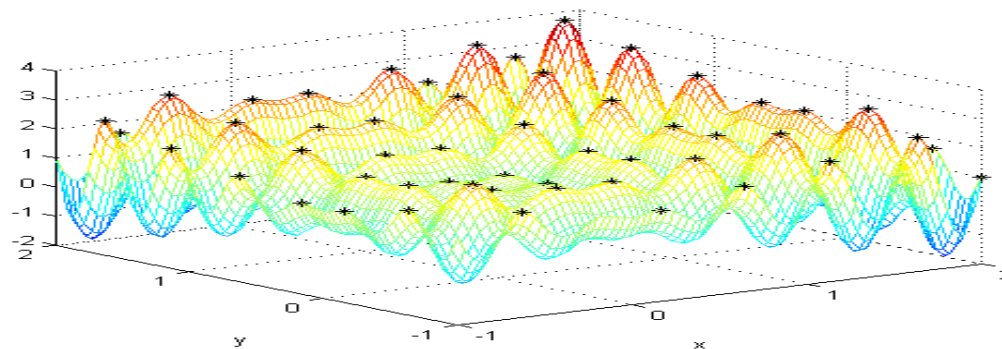
Sistem imunitar artificial

aiNET - optimizare multimodala

Populatie initiala



Populatie finala



Differential Evolution (DE)

Creatori: Rainer Storn & Kenneth Price (1995)

Scop: optimizare în domenii continue

Idee: pentru fiecare element al populației curente:

- se selectează aleator **3 elemente** din populație
- Mutația se bazează pe **calculul diferenței** dintre două elemente alese aleator din populație și pe adăugarea diferenței înmulțită cu un factor de scalare la un alt element aleator din populație (această operație stă la originea denumirii metodei)
- elementul construit la etapa anterioară se încrucișează cu elementul curent
- dacă noul element obținut prin încrucișare este mai bun decât elementul curent atunci îl înlocuiește

Structura generală: identică cu cea a strategiilor evolutive

Differential Evolution (DE)

Problema: maximizare $f: D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$

$X = \{x_1, \dots, x_m\}$ – populatie curenta

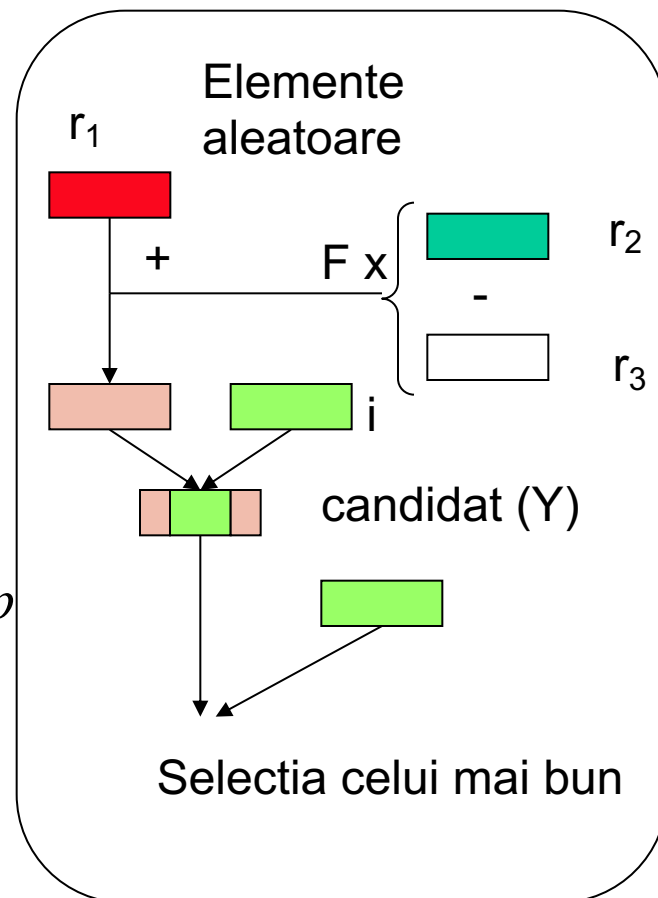
$Y = \{y_1, \dots, y_m\}$ – populatie de candidati

$Z = \{z_1, \dots, z_m\}$ – populatie noua

$$y_i^j = \begin{cases} x_{r_1}^j + F \cdot (x_{r_2}^j - x_{r_3}^j), & \text{cu probabilitatea } p \\ x_i^j, & \text{cu probabilitatea } 1 - p \end{cases}$$

$r_1, r_2, r_3 = \text{indici aleatori din } \{1, \dots, m\}$

$$F \in (0, 2], \quad p \in (0, 1] \quad z_i = \begin{cases} x_i, & f(x_i) > f(y_i) \\ y_i, & f(x_i) \leq f(y_i) \end{cases}$$



Differential Evolution (DE)

Variante

$$y_i^j = \begin{cases} x_{r_1}^j + F \cdot (x_{r_2}^j - x_{r_3}^j) \cdot N(0,1), & \text{cu probabilitatea } p \\ x_i^j, & \text{cu probabilitatea } 1 - p \end{cases}$$

$$y_i^j = \begin{cases} x_{r_1}^j + F_1 \cdot (x_{r_2}^j - x_{r_3}^j) + F_2 \cdot (x_{r_4}^j - x_{r_5}^j), & \text{cu probabilitatea } p \\ x_i^j, & \text{cu probabilitatea } 1 - p \end{cases}$$

$$y_i^j = \begin{cases} \lambda x_*^j + (1 - \lambda)x_{r_1}^j + F \cdot (x_{r_2}^j - x_{r_3}^j), & \text{cu probabilitatea } p \\ x_i^j, & \text{cu probabilitatea } 1 - p \end{cases}$$

x_* = cel mai bun element al populatiei

Taxonomie: DE/element de bază/număr de diferențe/tip de încrucișare
(e.g. DE/rand/1/bin, DE/rand/2/bin, DE/best/1/bin etc.)

Differential Evolution (DE)

Parametri de control:

Factor de scalare (F):

- domeniu de valori: (0,2)
- valori mici: efect de căutare locală;
pot conduce la situații de convergență prematură
- valori mari: efect de căutare globală

Probabilitate de încrucișare:

- valori mici (<0.5): adecvate pt probleme separabile (optimizarea se poate realiza separat pe componente)
- valori mari (>0.5): adecvate pentru probleme neliniar separabile

Differential Evolution (DE)

- Autoadaptare DE
 - Parametrii F și p sunt modificați în timpul procesului de optimizare:
 - Pe baza informațiilor anterioare colectate despre elemente: fiecare element din populație are propria valoare pentru F și p
 - Perturbări aleatoare
 - Selecția dintr-o mulțime posibilă de valori aleatoare
 - Exemple
 - jDE (J. Brest, 2006)
 - JADE (Zhang & Sanderson, 2009)
 - SHADE (Tanabe&Fukunaga,2013)

Differential Evolution (DE)

Auto-adaptare [Brest, 2006]

- Se extinde fiecare element al populației cu două componente, una corespunzătoare factorului de scalare, F_i , iar cealaltă corespunzătoare probabilității de încrucișare, p_i .
- La fiecare generație parametrii dacă elementul rezultat este mai bun decât părintele se păstrează parametrii altfel se înlocuiesc cu valori aleatoare

$$F_i^{new} = \begin{cases} rand(0.1,1) & \text{daca } rand(0,1) < \tau_F \\ F_i^{old} & \text{altfel} \end{cases}$$

$$p_i^{new} = \begin{cases} rand(0,1) & \text{daca } rand(0,1) < \tau_p \\ p_i^{old} & \text{altfel} \end{cases}$$

Valori initiale: $F=0.5$, $p=0.9$

Noi parametrii: τ_F , τ_p

Differential Evolution (DE)

JaDE

- Varianta de mutație: DE/p-best –vectorul de bază este selectat aleator din p% cele mai bune elemente ale populației
- Elementele din populație care sunt înlocuite de copii sunt stocate într-o arhivă și selectarea termenului x_{r3} se realizează din această arhivă
- La fiecare generație parametrii care corespund fiecarui element sunt înlocuiți folosind următoarele probabilități de distribuție
 - F_i este generat folosind o distribuție Cauchy(mF_i , 0.1)
 - p_i este generat folosind Normal(mp_i , 0.1)
 - mF si mp sunt inițializați cu 0.5 (pentru toate elementele) și sunt calculați pe baza mediei elementelor acceptate in populatie

Differential Evolution (DE)

SADE:

- Similar cu JADE
- Diferență:
 - În loc de înlocuirea aleatoare a parametrilor F_i și p_i se selectează valori din arhivă

Differential Evolution (DE)

Aplicații:

- optimizare globală, multicriterială, multimodală
- Analiza datelor (clustering, reguli de clasificare)
- Planificare activități (grid scheduling)
- Prelucrarea imaginilor

Harmony Search (HS)

Sursa de inspirație: modul de ajustare a tonalităților în compoziția muzicală (Geem, 2001)

Structura generală: similară cu structura de la DE

Element specific: modul de construire al unui nou element

$$y_i^j = \begin{cases} x_r^j + bw(j) \cdot U_3 & \text{daca } U_1 < p_1 \text{ si } U_2 < p_2 \\ x_r^j & \text{daca } U_1 < p_1 \text{ si } U_2 \geq p_2 \\ U(a_j, b_j) & \text{altfel} \end{cases}$$

Semnificație notații:

- r = index aleator din $\{1, 2, \dots, m\}$ (m = dim populație)
- U_1, U_2 : variabile aleatoare uniform repartizate în $[0, 1]$
- U_3 : variabilă aleatoare uniform repartizată în $[-1, 1]$
- $bw(j) = \text{stdev}(X(j))$ (abatere standard a valorilor componente j)
- p_1, p_2 : parametri control (ex: $p_1=0.9, p_2=0.75$)

Probabilistic Model Building Algorithms

Specific: reprezintă o clasă de algoritmi care realizează căutarea soluției prin simularea unor distribuții de probabilitate

Alte denumiri/variante:

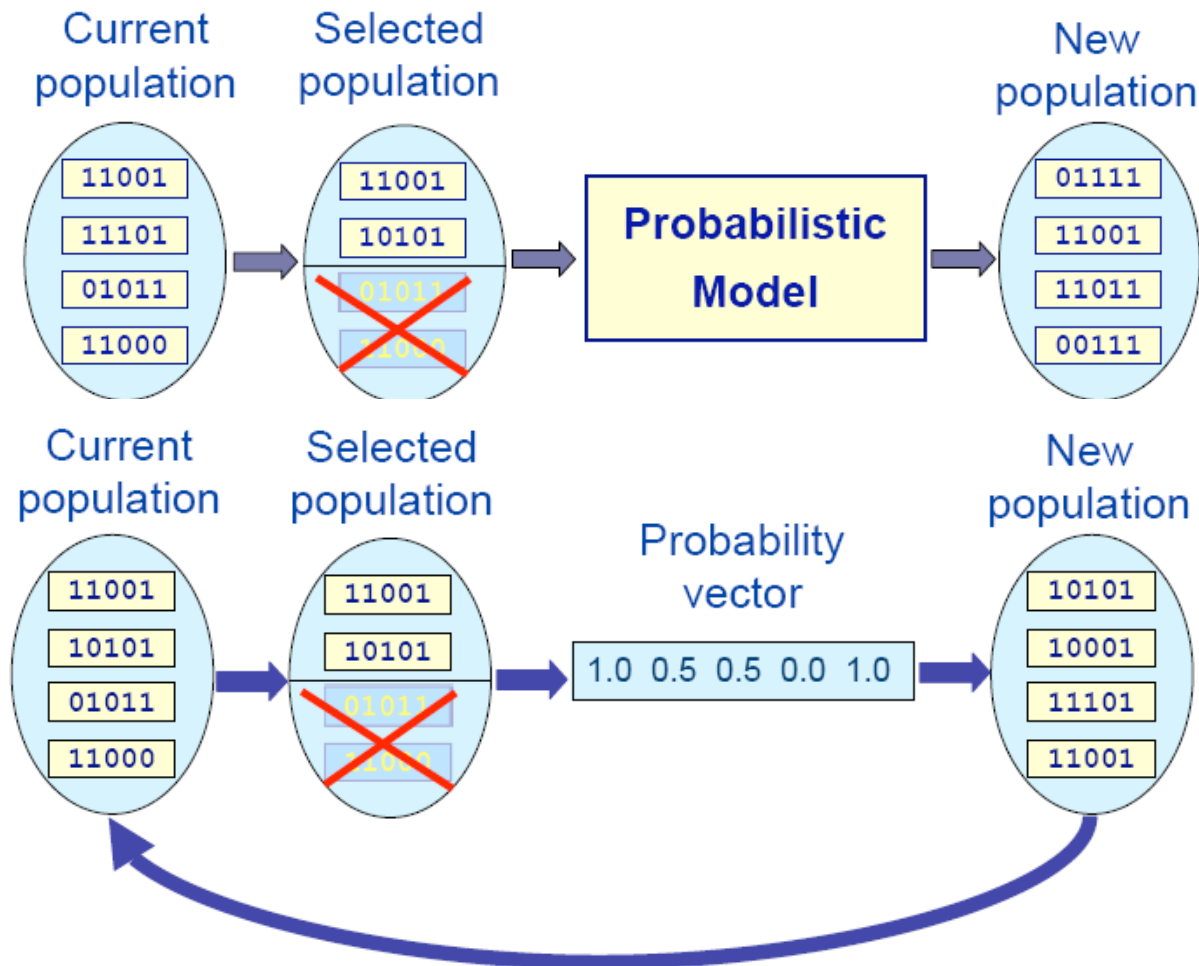
- Estimation of Distribution Algorithms (EDA) [Mühlenbein & Paass, 1996]
- Iterated Density Estimation Algorithms (IDEA) [Bosman & Thierens, 2000]
- Bayesian Optimization Algorithms (BOA) [Pelikan, Goldberg, & Cantu-paz, 1998]

Idee: se înlocuiește mutația și încrucișarea cu un proces de estimare a unei distribuții de probabilitate a elementelor selectate iar noile elemente sunt generate prin simulare în conformitate cu acea distribuție de probabilitate

Observație: în felul acesta se exploatează distribuția elementelor promițătoare din populație

Probabilistic Model Building Algorithms

Ilustrarea ideii [M.Pelikan – Probabilistic Model Building GA Tutorial]



Probabilistic Model Building Algorithms

Structura generală.

Pas 1: Inițializarea populației (m elemente)

Pas 2: REPEAT

- selectează $m' < m$ elemente din populația curentă (în funcție de calitatea lor)
- estimează o distribuție de probabilitate folosind elementele selectate
- generează m elemente în conformitate cu distribuția de probabilitate estimată

UNTIL <conditie de oprire>

Probabilistic Model Building Algorithms

Observații:

- Dificultatea principală constă în estimarea distribuției, în special în cazul în care componentele elementelor (variabilele funcției obiectiv) sunt corelate
- Pentru simplificare se poate presupune ca variabilele sunt independente; în acest caz probabilitățile corespunzătoare lor pot fi estimate independent

Variante de algoritmi bazate pe ipoteza independenței:

- UMDA (Univariate Marginal Distribution Algorithm)
- PBIL (Probabilistic Based Incremental Learning)

Probabilistic Model Building Algorithms

UMDA (Mühlenbein, Paass, 1996)

$$P^t(x_i) = \frac{\sum_{j=1}^{m'} \delta_j(X_i = x_i | S(t-1))}{m'} \quad \text{probabilitatea corespunzatoare componentei } i$$

$S(t-1)$ este populatia selectata la iteratia $(t-1)$

$\delta_j(X_i = x_i | S(t-1)) = 1$ daca al j -lea element selectat contine valoarea x_i pe componenta i

PBIL (Baluja, 1995)

$$P^t(x_i) = (1 - \alpha)P^{(t-1)}(x_i) + \alpha \frac{\sum_{j=1}^{m'} \delta_j(X_i = x_i | S(t-1))}{m'}$$

$$\alpha \in (0,1]$$

Memetic Algorithms

Creator: Pablo Moscato (1989)

Specific: hibridizarea algoritmilor evolutivi cu tehnici de cautare locala cu scopul de a introduce in metoda cunostinte specifice problemei de rezolvat

Denumire: “memetic” provine de la “meme” un termen introdus de Richard Dawkins pentru a desemna “unitatea de transmitere a diferitelor entitati (biologice, culturale, materiale) între generații”

Variante: Hybrid Evolutionary Algorithms, Baldwinian Evolutionary Algorithms, Lamarckian Evolutionary Algorithms, Cultural Algorithms or Genetic Local Search

Memetic Algorithms

Structura generală:

Pas 1: Inițializarea populației

Pas 2: WHILE <condiție de continuare>

- evaluează elementele populației
- generează noi elemente aplicând operatorii de evoluție (de exemplu încrucișare și mutație)
- selectează o subpopulație asupra căreia se aplică operatori specifici de căutare locală (de exemplu Simulated Annealing sau Tabu Search)

Observatii:

1. Căutarea locală se poate baza pe o colecție de algoritmi dintre care se alege la fiecare etapă câte un algoritm (în manieră aleatoare)
2. Elementele ce definesc operatorii de căutare locală pot face parte din componentele populației și pot fi transformate în procesul de evoluție