

PROCESAREA CUNOȘTINȚELOR ȘI CALCUL INTELIGENT - PCCI

Calcul evolutiv

Algoritme genetice. Programarea genetică

Cuprins:

- Calculul evolutiv. Inspirația naturală
 - ❖ Principii și terminologie
- Algoritme genetice
 - ❖ Algoritmul de principiu
 - ❖ Operatorii genetici: selecție, încrucișare, mutație
- Programarea genetică
 - ❖ Algoritmul de principiu
 - ❖ Detalii de implementare, asemănări și deosebiri față de RNA și AG

Calcul evolutiv

- Optimizare: găsirea soluției celei mai bune a unei probleme, pe baza unor criterii stabilite inițial, dintr-un set de opțiuni disponibile.
 - ❖ Exemplu: descoperirea mărului celui mai copt în funcție de mărime (min. 0, max. 1), culoare (min. 0, max. 1) și consistența pulpei (min. 0, max. 1). Un măr „optim” ar avea mărimea 1, culoarea 1 și consistența 0.7.
- Calculul evolutiv este un subdomeniu al calculului inteligent care cuprinde metode **metaeuristice** de rezolvare evolutivă a unor probleme de optimizare.

Metode euristice și metaeuristice (1/2)

- EURÍSTIC, -Ă, euristici, -ce, adj. 1. (Despre procedee metodologice) Care servește la descoperirea unor cunoștințe noi.
- Metodă **euristică**: metodă sau algoritm care determină o soluție aproximativă a unei probleme
- Nu este o metodă exactă, scopul ei este de a descoperi o soluție rezonabil de precisă mai repede decât o metodă clasică
- Pleacă de la o populație sau un set de soluții inițiale, pe care le rafinează într-un proces **iterativ**, folosind pe parcursul rezolvării metode de căutare care implică un grad mai mare sau mai mic de **aleatoriu (stochastic)**

Metode euristice și metaeuristice (2/2)

- META- Element de compunere însemnând „după” sau exprimând ideea de transformare, de schimbare, folosit la formarea unor substantive și a unor adjective.
- EURÍSTIC, -Ă, euristici, -ce, adj. 1. (Despre procedee metodologice) Care servește la descoperirea unor cunoștințe noi.
- Metodă **metaeuristică**: metodă generală care determină o soluție aproximativă a unei categorii de probleme
- Metodele euristice sunt adaptate unei probleme anume; metodele metaeuristice aplică același principiu general de rezolvare unu set mai larg de probleme, din domenii diferite: sunt metode-șablon care pot fi adaptate în funcție de problema rezolvată

Calcul evolutiv și inspirația sa naturală

- Specifică calculului evolutiv este utilizarea unei populații inițiale de soluții valide ale problemei studiate (de obicei generate aleatoriu), care este rafinată (evoluează) pe baza principiilor evoluției naturale descoperite de Darwin:
 - ❖ Reproducerea
 - ❖ Crearea de indivizi noi prin combinații de caracteristici moștenite de la părinți
 - ❖ Mutația: schimbări punctuale, accidente de evoluție
 - ❖ Supraviețuirea prioritară a celor mai bune adaptați indivizi din cadrul speciei (selecția naturală)
 - ❖ Cooperarea între indivizii unei populații în vederea atingerii unui scop comun
- Algoritmii evolutive modelează matematic aceste principii.

Calcul evolutiv

- Sunt algoritme de **căutare paralelă**: fiecare individ din populație explorează spațiul de căutare disponibil, având șansa să descopere o soluție globală mai bună.
- Gradul de optimalitate sau de adaptare al unei soluții (calitatea, precizia soluției) este evaluată prin intermediul unei **funcții de adaptare**. Soluțiile pot fi ierarhizate în funcție de valoarea acestei funcții de adaptare.
- Spațiul de căutare este limitat; căutarea trebuie să țină cont de aceste limite (de exemplu, pentru evaluarea gradului de coacere a unui măr, dacă s-au stabilit pentru cei parametri mărime, culoare și consistență a pulpei valori între 0 și 1, orice soluție care presupune valori situate în afara acestui interval va fi **invalidă**).

Algoritme de calcul evolutiv

- Algoritmemele genetice și programarea genetică (genetic algorithms, genetic programming)
- Algoritmul furnicii (the ant algorithm, ant colony optimization)
- Optimizarea deplasării roiurilor de particule (Particle Swarm Optimization, PSO)
- Algoritmul imunitar (The Artificial Immune Algorithm)
- Evoluția diferențială (Differential Evolution)

Algoritme genetice

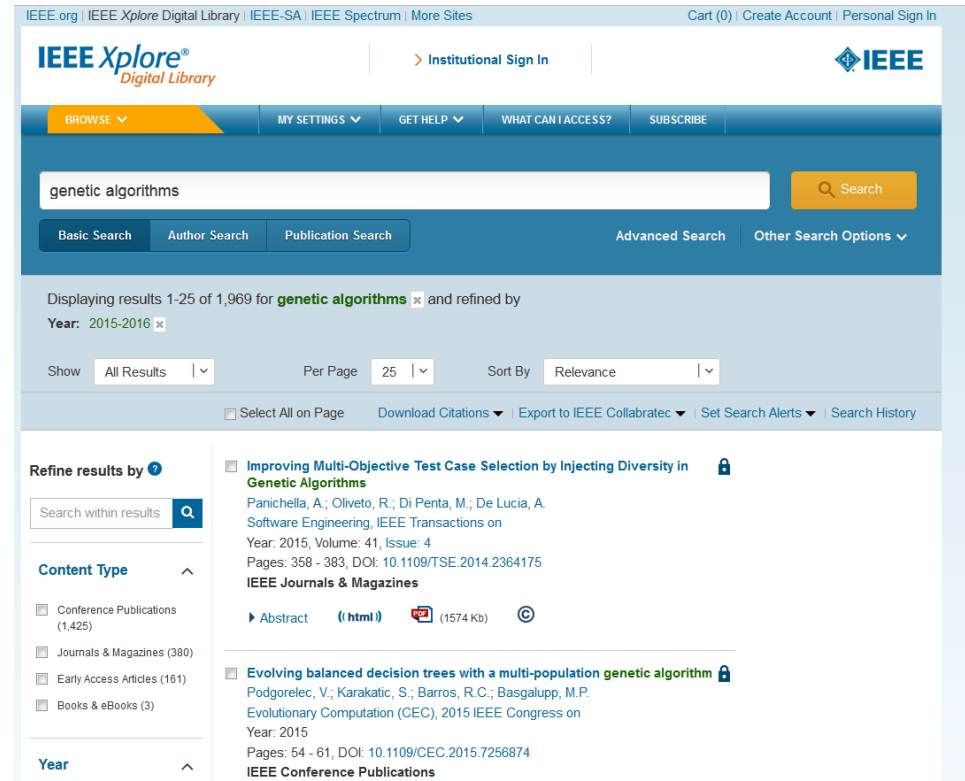


Istoric

- Au fost inventați de profesorul American John Henry Holland (1929 - 2015)
 - ❖ Cartea *Adaptation in Natural and Artificial Systems* (1975)



Sunt folosiți și în prezent, în forma originală sau cu implementări îmbunătățite, pentru rezolvarea problemelor de optimizare.



The screenshot shows the IEEE Xplore Digital Library interface. At the top, there are navigation links for IEEE.org, IEEE Xplore Digital Library, IEEE-SA, IEEE Spectrum, and More Sites. On the right, there are links for Cart (0), Create Account, and Personal Sign In. The main header features the IEEE Xplore Digital Library logo and an Institutional Sign In button. Below the header is a navigation bar with links for BROWSE, MY SETTINGS, GET HELP, WHAT CAN I ACCESS?, and SUBSCRIBE. A search bar contains the text 'genetic algorithms' and a Search button. Below the search bar are tabs for Basic Search, Author Search, and Publication Search, along with links for Advanced Search and Other Search Options. The results section displays 'Displaying results 1-25 of 1,969 for genetic algorithms' and 'Year: 2015-2016'. It includes filters for Show (All Results), Per Page (25), and Sort By (Relevance). Below the filters are links for Select All on Page, Download Citations, Export to IEEE Collaborate, Set Search Alerts, and Search History. The results list shows two entries: 'Improving Multi-Objective Test Case Selection by Injecting Diversity in Genetic Algorithms' by Panichella, A.; Oliveto, R.; Di Penta, M.; De Lucia, A. (2015, Volume: 41, Issue: 4, Pages: 358 - 383, DOI: 10.1109/TSE.2014.2364175) and 'Evolving balanced decision trees with a multi-population genetic algorithm' by Podgorelec, V.; Karakatic, S.; Barros, R.C.; Basgalupp, M.P. (2015, Pages: 54 - 61, DOI: 10.1109/CEC.2015.7256874).

Algoritme genetice - inspirația naturală

- Trăsăturile fizice (înălțime, culori, rezistență la efort sau la boli) sunt codate în lanțuri de gene stocate în ADN (în cromozomi).
- Copiii moștenesc de la părinți o combinație de trăsături care îi poate face mai adaptați sau mai puțin adaptați la mediu
- Pot apărea accidente genetice, prin mutație
- Populația tinde să evolueze din generație în generație către o adaptare mai bună la mediul ambiant
- În general, supraviețuiesc indivizii cei mai bine adaptați (selecție naturală)
- Populațiile sunt limitate numeric



Forma generală a unui algoritm genetic

- Formarea populației inițiale și inițializarea contorului de generații.
- Evaluarea funcțiilor de adaptare pentru cromozomii din populația inițială.
- Procesul iterativ, aplicarea operatorilor genetici:
 - ❖ Selectarea celor mai bine adaptați cromozomi din populația curentă
 - ❖ Încrucișarea cromozomilor-părinți din populația curentă și obținerea cromozomilor-urmași
 - ❖ Aplicarea mutațiilor cromozomilor-urmași din generația curentă
 - ❖ Formarea populației pentru noua generație și evaluarea funcțiilor de adaptare ale cromozomilor acesteia
 - ❖ Dacă nu s-a îndeplinit criteriul de oprire, trecerea la generația următoare
- Soluția optimă determinată de către algoritm corespunde cromozomului cu funcția de adaptare cea mai bună din ultima generație.

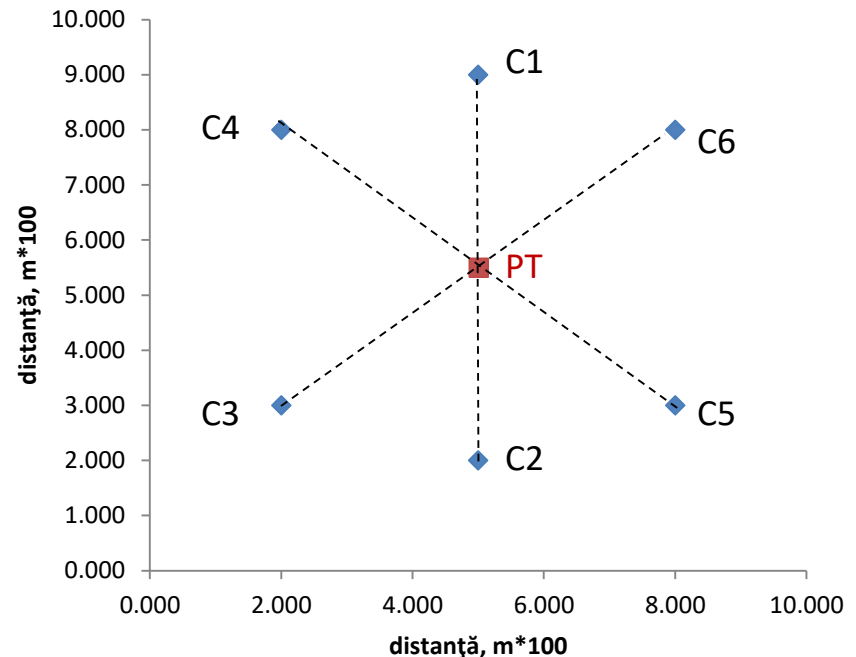
Problema distanței minime:

- Se dau punctele fixe (**albastre**) indicate de coordonatele carteziane, care indică amplasarea unor consumatori:

	x	y
p_1	2.000	8.000
p_2	5.000	2.000
p_3	8.000	8.000
p_4	5.000	9.000
p_5	8.000	3.000
p_6	2.000	3.000

- Să se identifice punctul optim (**roșu**) de amplasare al unui post de transformare astfel încât rețeaua de joasă tensiune să rezulte de lungime minimă.

- Simplificare: se ignoră restricțiile de teren (clădiri, zone interzise)



Formarea populației inițiale

- Algoritmile genetice (AG) urmăresc determinarea soluției optime a unei probleme prin evoluția unei populații inițiale.
- Spre deosebire de selecția naturală, care acționează asupra unui număr mare de indivizi într-o perioadă lungă de timp, AG dispun de o populație redusă de cromozomi, care va evolua într-un număr mic de generații (de exemplu, 100 de indivizi și 100 de generații).
- În cazul selecției naturale, populația variază ca număr (crește sau se stinge). AG au de obicei o populație fixă.

Formarea populației inițiale

- În cadrul AG, cromozomii reprezintă codificări ale soluțiilor problemei studiate, prezentate sub forma unor șiruri sau lanțuri de caractere.
- Populația inițială trebuie să îndeplinească o serie de condiții:
 - cromozomii să fie valizi, adică să poată fi decodificați în soluții ale problemei, iar aplicarea asupra lor a operatorilor genetici să producă soluții viabile
 - populația inițială trebuie să acopere un domeniu suficient de larg din spațiul soluțiilor posibile
 - reprezentarea aleasă pentru soluții trebuiesă încurajeze deplasarea cromozomilor către soluția optimă a problemei.
- Elementele unui cromozom sau genele conțin codificarea valorilor parametrilor de intrare ai problemei care trebuie optimizați. Se poate folosi o reprezentare binară (cu valori 0 și 1), dar și reprezentări cu numere reale.

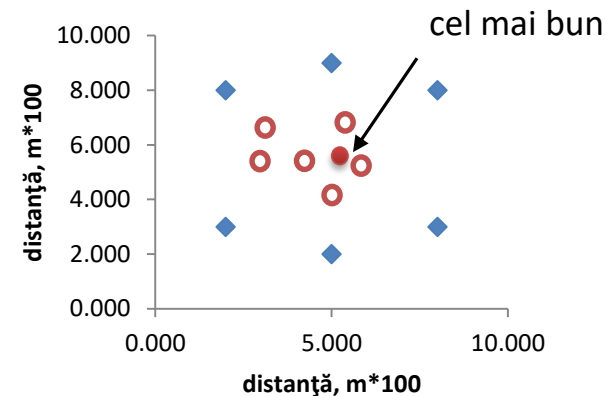
Calculul funcției de adaptare

- Pentru a se determina cât de bine rezolvă cromozomii din populația curentă problema dată, li se calculează fiecăruia funcția de adaptare:
 - ❖ Cromozomii sunt decodificați
 - ❖ Problema este rezolvată cu parametrii decodificați din cromozom
 - ❖ Dacă rezultatul obținut este bun (cromozom adaptat), funcția de adaptare trebuie să aibă o valoare mare
 - ❖ Dacă rezultatul obținut este slab (cromozom neadaptat), funcția de adaptare trebuie să aibă o valoare mică.
- Funcția de adaptare se calculează diferit, în funcție de problema care trebuie rezolvată.

populația inițială și funcția de adaptare – problema distanței minime

- Distanța între PT și consumatori trebuie să fie minimă:

$$FA_{cromozom} = \sum_{i=1}^6 dist(PT, C_i) = \min$$



- Exemplu de populație inițială:

	coordonate x, binare	coordonate y, binare	X , zecimal	Y, zecimal	distanță (1/FA)
	10001101100010	00000101100101	9058	357	42429.04
	10001110101110	10010001111100	9134	9340	37570.96
	00001111001111	01110110011001	975	7577	32368.32
cel mai bun	01010101011101	00111101010010	5469	3922	23514.62
	10010110110001	00011010110000	9649	1712	39553.99
	10010111101010	00000100111110	9706	318	44961.08
	01001011110110	00000111001110	4854	462	34173.90
	00010110001011	10000000101011	1419	8235	31740.26
	10001111000101	00110001100011	9157	3171	33406.29
	10010101111011	00000101011000	9595	344	44433.01

populația inițială și funcția de adaptare – problema distanței minime

- S-a ales o populație inițială de 10 indivizi, cu valori ale coordonatelor (x, y) reprezentate binar, codificând numere zecimale care reprezintă coordonate în plan.
- O soluție a problemei este reprezentată de o pereche de cromozomi, unul pentru coordonata x și unul pentru coordonata y .
- Soluția problemei este unică - punctul central (x_{optim}, y_{optim}) pentru care suma tuturor distanțelor până la punctele din plan care reprezintă consumatorii este minimă
- Codificarea binară este, în acest caz, cea mai comodă
 - ❖ Șirul de caractere (cromozomul) este suficient de lung
 - ❖ Folosește doar două valori, 0 și 1
- AG trebuie să descopere prin selecție, încrucișare și mutație o combinație de coordonate (x, y)

Selecția

- Pornind de la o populație de lanțuri cromozomiale existentă, se creează o populație nouă, pe baza gradului de adaptare a fiecărui cromozom.
- Noua populație este generată folosind duplicate exacte ale cromozomilor existenți, după regula generală: cromozomul cu cea mai mare valoare a funcției de adaptare produce cele mai multe duplicate, iar cromozomul cu funcția de adaptare cea mai mică se reproduce cel mai puțin (cromozomii, adică indivizii slab adaptați sunt eliminați din populație).
- Cromozomii mai puțin performanți vor fi selectați și ei, deși în număr mai mic, deoarece, chiar dacă în ansamblu structura lor nu codifică soluția optimă, pot conține secvențe de gene care să conducă la aceasta prin încrucișare cu un cromozom mai evoluat.

Tipuri de selecție

- **Selecția uniformă**: fiecare cromozom-părinte are șanse egale de a fi selectat, indiferent de valoarea funcției de adaptare asociate:

$$p_i = \frac{1}{N} \quad i = 1, \dots, N$$

- N - numărul total de cromozomi din populația curentă,
 - p_i este probabilitatea de selecție a cromozomului i .
- cea mai simplă regulă de selecție, dar și cea mai puțin eficientă, deoarece nu ia în considerare gradul de adaptare a soluțiilor la problema studiată.

Tipuri de selecție

- **Selecția proporțională:** probabilitatea de selectare a cromozomului i se calculează luând în considerare aportul funcției sale de adaptare în ansamblul populației:

$$p_i = \frac{FA_i}{\sum_{k=1}^N FA_k}$$

- N - numărul total de cromozomi din populația curentă,
- p_i - probabilitatea de selecție a cromozomului i ,
- FA_i - funcția de adaptare a cromozomului i .

Tipuri de selecție

- **Selecția prin competiție (tournament selection):** se aleg la întâmplare m cromozomi din populație și se rețin dintre ei w cromozomi, cei cu funcția de adaptare cea mai mare ($w < m$).
- Procesul se repetă până la completarea numărului de cromozomi necesar pentru formarea noii populații.

Tipuri de selecție

- Selecția după regula ruletei (roulette rule selection)
- Suma valorilor funcțiilor de adaptare pentru toți cromozomii din populația curentă se asociază întregii lungimi a ruletei, considerată de valoare 1.
- Ruleta se împarte apoi în sectoare de lungimi egale cu proporția funcției de adaptare a fiecărui cromozom.
- Pentru selectarea unui cromozom, se generează un număr aleatoriu în intervalul $(0, 1)$ și se identifică pe ruletă sectorul în care se încadrează acest număr. Cromozomul corespunzător sectorului respectiv va fi selectat.

Regula ruletei – problema distanței minime

10001101100010	00000101100101	00010110001011	10000000101011
10001110101110	10010001111100	01010101011101	00111101010010
00001111001111	01110110011001	10001111000101	00110001100011
01010101011101	00111101010010	10001110101110	10010001111100
10010110110001	00011010110000	10001111000101	00110001100011
10010111101010	00000100111110	01001011110110	00000111001110
01001011110110	00000111001110	10001110101110	10010001111100
00010110001011	10000000101011	00001111001111	01110110011001
10001111000101	00110001100011	10010111101010	00000100111110
10010101111011	00000101011000	10010110110001	00011010110000



				C1		C2		C3		C4				C5		C6		C7		C8		C9		C10							
	distanța	FA=1/distanța	selecție:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
C1	42429.04	0.02	rand*28=22,8 = C8																												
C2	37570.96	0.03	rand*28=11,7 = C4																												
C3	32368.32	0.03	rand*28=25,4 = C9																												
C4	23514.62	0.04	rand*28=3,5 = C2																												
C5	39553.99	0.03	rand*28=25,6 = C9																												
C6	44961.08	0.02	rand*28=17,7 = C7																												
C7	34173.9	0.03	rand*28=2,73 = C2																												
C8	31740.26	0.03	rand*28=7,8 = C3																												
C9	33406.29	0.03	rand*28=15,3 = C6																												
C10	44433.01	0.02	rand*28=14,8 = C5																												
suma	364151.47	0.28																													

C9 și C2 sunt selectați de două ori, în vreme ce C1 și C10 sunt eliminați din populație

Încrucișarea

- Se aleg doi părinți, care schimbă între ei părți din cromozom, rezultând doi urmași.
- Încrucișarea se face cu o probabilitate numită rată de încrucișare
 - ❖ Valoare recomandată: mare (0.9)



Încrucișarea

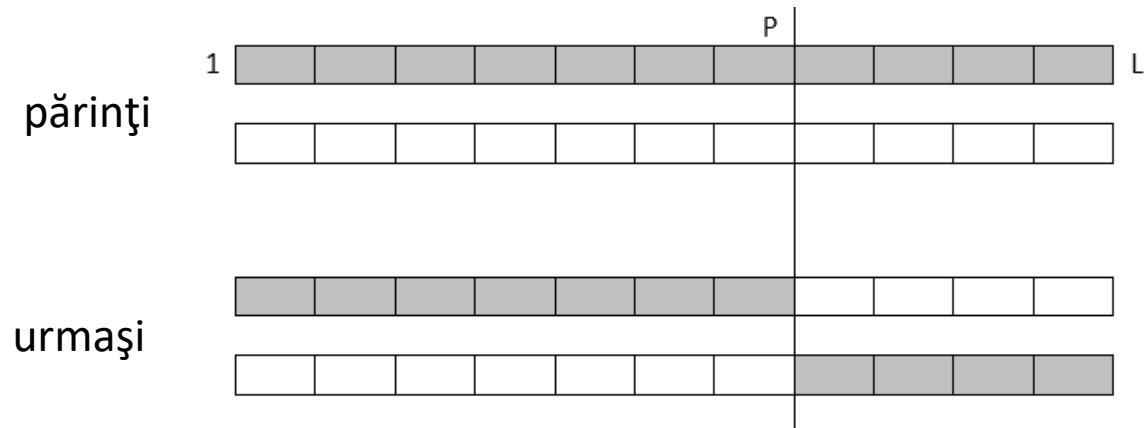
- Asigură schimbul de informație sub formă de gene între perechi de cromozomi părinți, rezultând cromozomi noi, care pot reprezenta soluții mai bune ale problemei studiate.
- Procesul de căutare a soluției prin încrucișare este unul cu caracter pronunțat paralel.
- Cromozomii mai slab adaptați trecuți de procedura de selecție își pot aduce contribuția la evoluția populației cedând segmente de informație care pot face parte din soluția optimă.
- Cromozomii mai bine adaptați își pot spori calitățile, evoluând și îmbunătățindu-și structura prin recombinare. Încrucișarea este considerată cel mai important pas al procesului evolutiv, prin ea realizându-se primenirea populației.

Tipuri de încrucișare

- Pentru toate tipurile de cromozomi:
 - ❖ Încrucișare într-un punct
 - ❖ Încrucișare în mai multe puncte (multipunct)
 - ❖ Încrucișarea uniformă
- Pentru cromozomi cu numere reale:
 - ❖ Încrucișarea aritmetică
 - ❖ Încrucișarea euristică

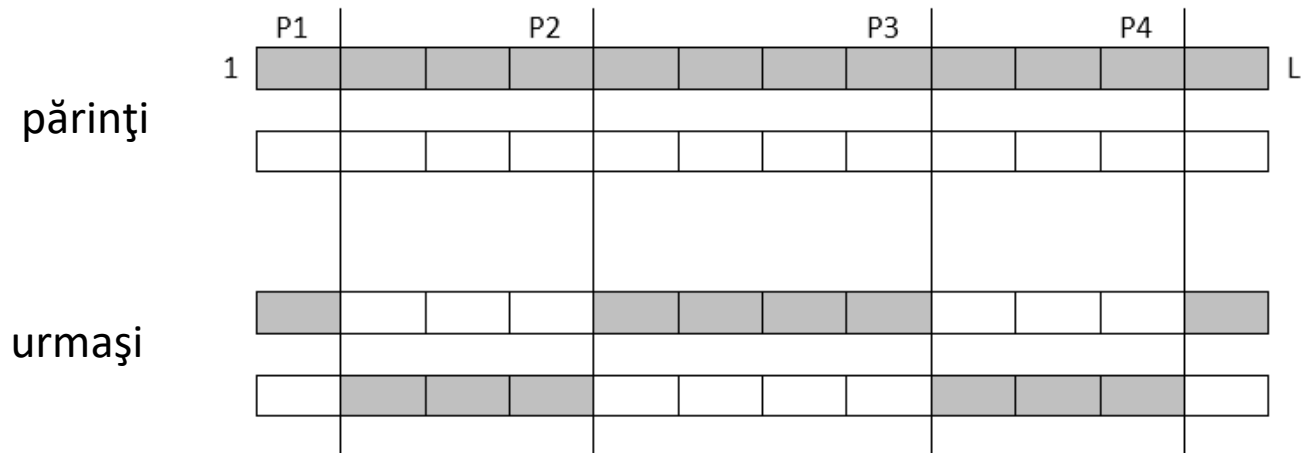
Încrucișarea într-un punct

- Este cel mai utilizat tip de încrucișare.
- Se generează o poziție aleatorie P cu valori între 2 și $L-1$ (L - lungimea cromozomului)
- De la acea poziție încolo, părinții schimbă genele între ei



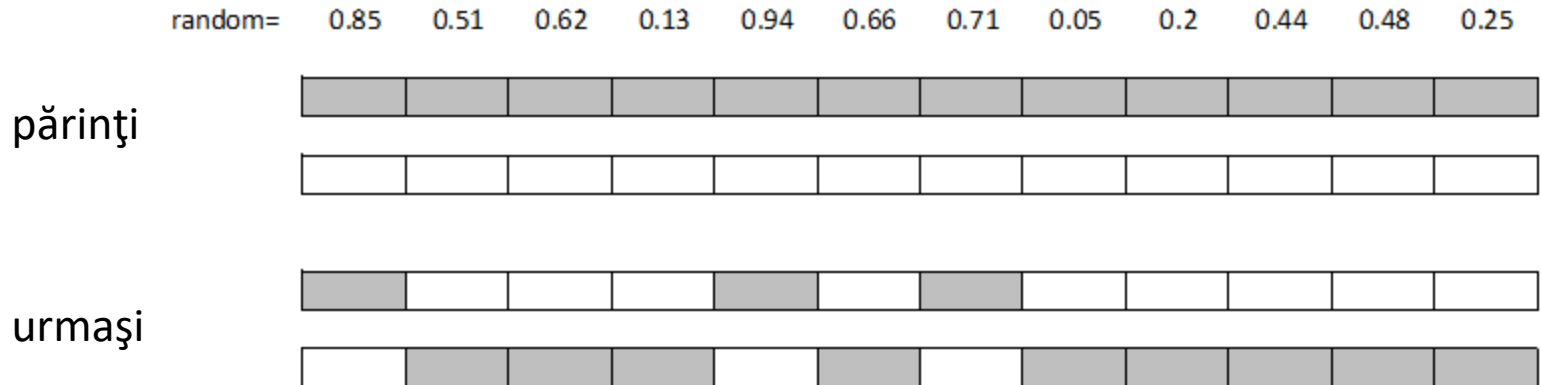
Încrucișarea multipunct

- $P > 1$, $P < L$ (exemplu, $P=4$)



Încrucișarea uniformă

- $P=L$
- Încrucișarea se face cu o probabilitate definită de utilizator
- exemplu: $P=0.7$



Încrucișarea aritmetică și euristică

- Încrucișarea aritmetică, plecând de la doi cromozomi părinți $P1$ și $P2$, generează un număr aleatoriu m între 0 și 1 și calculează cromozomii urmași $U1$ și $U2$ cu relațiile [Gavrilaș]:

$$U1 = m \cdot P1 + (1 - m) \cdot P2$$

$$U2 = (1 - m) \cdot P1 + r \cdot P2$$

- Încrucișarea euristică păstrează nemodificat cromozomul părinte cu funcția de adaptare cea mai mare, fie acesta $P1$, și îl recalculează pe cel de-al doilea cu o relație de forma:

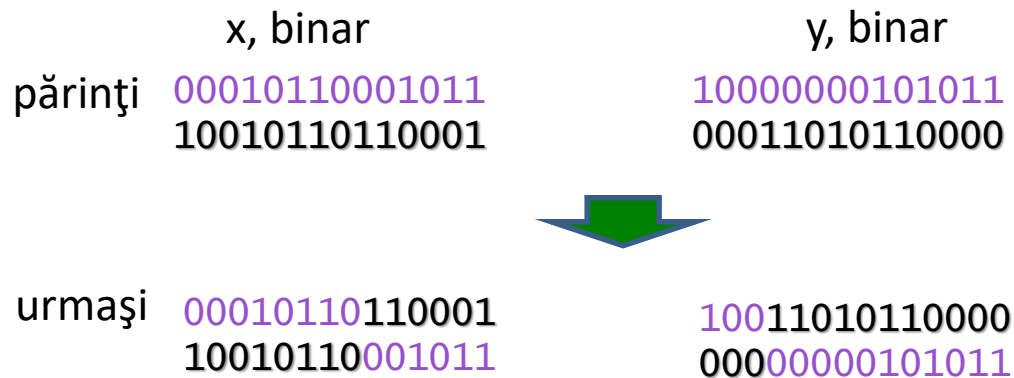
$$P2 = P2 + m \cdot (P2 - P1)$$


Încrucișarea aritmetică și euristică

- Pentru evitarea generării de cromozomi invalizi, se verifică de fiecare dată dacă valorile genelor nou obținute se încadrează între două valori limită stabilite apriori. În cazul încălcării restricției, urmașii sunt eliminați și încrucișarea se reface până la obținerea unor cromozomi valizi.
- Pentru evitarea blocării în acest pas, refacerea încrucișării se va face într-un număr limitat de încercări, după care cei trecuți în generația următoare vor fi cromozomii părinți.

Problema distanței minime: Încrucișarea

- Încrucișarea într-un punct



	x ,zecimal	y, zecimal	distanța
părinți	1419 9649	8235 1712	31740.26 39553.99
			
urmași	1457 9611	9904 0043	37549.80 45745.31

Problema distanței minime: Încrucișarea

- Populația după selecție

X, binar	y, binar	x, zec.	y, zec	distanța
00010110001011	10000000101011	1419	8235	31740.25
01010101011101	00111101010010	5469	3922	23514.61 cel mai bun
10001111000101	00110001100011	9157	3171	33406.29
10001110101110	10010001111100	9134	9340	37570.95
10001111000101	00110001100011	9157	3171	33406.29
01001011110110	00000111001110	4854	462	34173.89
10001110101110	10010001111100	9134	9340	37570.95
00001111001111	01110110011001	975	7577	32368.32
10010111101010	00000100111110	9706	318	44961.07
10010110110001	00011010110000	9649	1712	39553.99
devine populația după încrucișare				
01010100001011	00111100101011	5387	3883	23519.73 mai slab
00010111011101	10000001010010	1501	8274	31498.57
10001111000110	00000111001110	9158	462	42335.85
01001011110101	00110001100011	4853	3171	24393.55
00001111001110	00010001111100	974	1148	39064.72
10001110101111	11110110011001	9135	15769	68154.84
10001111000101	00011010110011	9157	1715	37470.44
10010110110001	00110001100000	9649	3168	35713.12
10010110101110	00000101111100	9646	380	44479.69
10001111101010	10010000111110	9194	9278	37597.42

Se pierde cromozomul cel mai bun din generația anterioară

Deși schimbările în populație sunt majore, nu se descoperă un cromozom mai bun, cu distanță mai mică decât 23514.61

Mutația

- Dacă populația are dimensiuni mici sau după parcurgerea unui număr mare de generații, structura populației tinde să se uniformizeze, predominând un număr mic de cromozomi, de obicei dintre cei bine adaptați dintre cei descoperiți până în acel moment.
- De cele mai multe ori, acești cromozomi codifică un optim local al soluției problemei studiate. Pentru a da posibilitatea algoritmului să continue procesul evolutiv, se apelează la operatorul mutație.

Mutația

- La fel ca în procesul evoluției naturale, mutația definită la nivelul AG înseamnă **modificarea accidentală a unei singure gene din interiorul unui cromozom**, prin bascularea valorii ei din 0 în 1 sau viceversa, ori modificarea valorii unei gene număr real.
- Rezultatul mutației este un cromozom cu structură nouă care, el însuși sau ulterior, prin recombinare, poate conduce la îmbunătățirea soluției optime identificate de AG.
- Mutația se face de obicei cu o probabilitate mică, numită **rată de mutație** (valori recomandate: 0.001 - 0.1)

Problema distanței minime: mutația

- Se schimbă aleatoriu o genă 0 în 1

coordonata y, vechi

00000000101011



coordonata y, nou

0**1**000000101011

	x, binar	y, binar	X , zecimal	Y, zecimal	distanța
(x,y) vechi	10011010110000	00000000101011	9611	43	45745.31
(x,y) nou	10011010110000	0 1 000000101011	9611	4139	34169.69

Problema distanței minime: mutația

- Populația după încrucișare

X, binar	y, binar	x, zec.	y, zec	distanța
01010100001011	00111100101011	5387	3883	23519.73
00010111011101	10000001010010	1501	8274	31498.57
10001111000110	00000111001110	9158	462	42335.85
01001011110101	00110001100011	4853	3171	24393.55
00001111001110	00010001111100	974	1148	39064.72
10001110101111	11110110011001	9135	15769	68154.84
10001111000101	00011010110011	9157	1715	37470.44
10010110110001	00110001100000	9649	3168	35713.12
10010110101110	00000101111100	9646	380	44479.69
10001111101010	10010000111110	9194	9278	37597.42

devine populația după mutație

01010100001011	00111100101011	5387	3883	23519.73
00010111011101	10000001010010	1501	8274	31498.57
10001111000110	00000111001110	9158	462	42335.85
01001011110101	00110001100011	4853	3171	24393.55
00001111001110	00010001111100	974	1148	39064.72
10001110101111	11110110011001	9135	15769	68154.84
10001111000101	00011010110011	9157	1715	37470.44
10010110110001	00110001100000	9649	3168	35713.12
10010110101110	00000101111100	9646	380	44479.69
10001111110101	10010000111110	9210	9278	37663.17

Din cauza ratei de mutație mici, se face o singură schimbare în populație (CZ 10, coord. x)
Nu se descoperă un cromozom mai bun, cu distanță mai mică decât 23514.61

Formarea populației pentru generația următoare

- Cel mai des utilizată metodă pentru reînnoirea populației este **înlocuirea completă**, trecerea mai departe a tuturor cromozomilor rezultați în urma selecției, încrucișării și mutației, renunțându-se total la cromozomii părinți din generația anterioară.
- **Înlocuirea selectivă** păstrează pentru generația următoare doar o parte dintre cromozomii nou-descoperiți (de exemplu, pe cei cu funcția de adaptare mai bună decât părinții lor), trecând mai departe o parte dintre părinți.
- Deoarece cromozomul optim identificat în iterația anterioară poate să sde piardă în urma încrucișării și a mutației, adesea se utilizează tehnica **elitismului**, care memorează de la o generație la alta acest cromozom optim, asigurându-i perpetuarea și posibilitatea de a se recombină până la eventuala identificare a unui cromozom mai bun.

Problema distanței minime: elitismul

- Se va perpetua în generația următoare populația obținută după mutație, care, în exemplul dat, nu mai conține cromozomul cel mai bun generat în populația inițială.
- În generația a doua, această populație va fi supusă selecției, încrucișării și mutației.
- Cromozomul cel mai bun, cu distanța minimă calculată de 23514,61 se pierde la încrucișare. Folosirea elitismului i-ar permite să se perpetueze în generația a doua, cu posibilitatea de a forma un cromozom încă și mai bun.

Criterii de oprire pentru AG

- Cel mai des utilizat criteriu de oprire pentru AG este atingerea unui număr maxim de generații.
- Alte criterii de oprire pot fi atingerea unui anumit prag limită de uniformizare a populației sau neactualizarea soluției optime de-a lungul unui număr de generații.

Aplicații ale AG în energetică

- probleme de optimizare:
 - ❖ Determinarea structurii și configurației optime a rețelelor electrice de transport și distribuție
 - ❖ Compensarea sarcinilor reactive și optimizarea amplasării surselor de putere reactivă în rețelele electrice
 - ❖ Stabilirea plotului optim de funcționare pentru transformatoarele din posturile de transformare
 - ❖ Determinarea curbelor tip pentru micii consumatori, în vederea efectuării tranzacțiilor pe piața de energie

Programare genetică



Programarea genetică

- Tehnică de calcul inteligent care face parte din categoria algoritmilor evolutive, la fel ca algoritmele genetice, cu care împărtășește o serie de trăsături comune.
- Este un algoritm de optimizare dezvoltat în principal pentru rezolvarea automată a unor probleme pentru care utilizatorul nu cunoaște forma sau structura soluției.
- Din acest punct de vedere, PG se aseamănă cu aproximarea funcțiilor cu ajutorul RNA perceptron multistrat, care pot învăța expresia analitică a unei funcții doar pe baza unor perechi $x-f(x)$ definite tabelar și apoi oferă soluții aproximative pentru întreg domeniul de antrenare al variabilelor de intrare.

Programarea genetică

- Spre deosebire de RNA, PG descoperă și ilustrează corespondența dintre intrările și ieșirile cunoscute ale unei probleme sub forma unor **expresii analitice** alcătuite din variabilele de intrare ale problemei, constante, operatori și funcții matematice (înmulțire, adunare, sinus, logaritm etc.) reprezentate sub forma unor **arbori**.
- Acești arbori pot forma expresii matematice sau secvențe de programe de calculator.

Programarea genetică

- În mod asemănător AG, PG pornește de la o populație de arbori generați aleator din setul disponibil de variabile, constante, operatori și funcții, pe care o rafinează folosind operatorii de selecție, încrucișare și mutație, pentru a descoperi combinații noi și potențial mai bune.
- Performanțele soluțiilor obținute se măsoară cu ajutorul funcției lor de adaptare, ce exprimă diferența dintre rezultatul obținut rezolvând expresia matematică sau secvența de program codificată în arborele care reprezintă soluția și valoarea dorită corespunzătoare din setul de date de învățare.

Forma de principiu a unui algoritm de PG

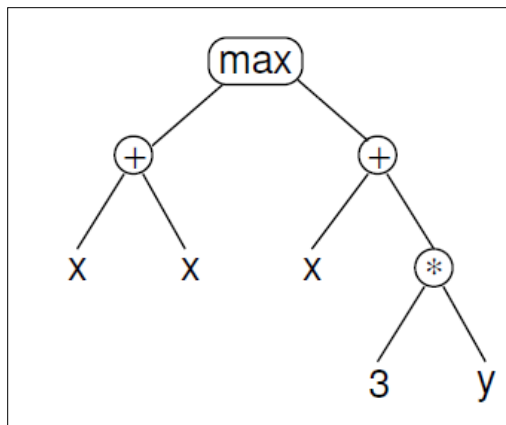
- Construiește setul de date de învățare, sub forma unor perechi de intrare-ieșire
- Generează aleatoriu, folosind variabilele de intrare și un set predefinit de constante, operatori matematici, funcții matematice și/sau cuvinte cheie o populație de soluții posibile ale problemei (expresii matematice sau programe complexe de calculator)
- Repetă până la îndeplinirea unui criteriu de oprire (de exemplu, nr max. de iterații)
 - ❖ evaluează fiecare soluție, calculându-i funcția de adaptare
 - ❖ selectează soluții pe baza funcției de adaptare
 - ❖ aplică operatorii de încrucișare și mutație pentru a obține soluții noi

Setul de date de învățare

- Se construiește ca la RNA, în perechi de intrări-ieșiri dorite
- Datele de intrare reprezintă combinații de valori discrete ale variabilelor de intrare care ilustrează scenarii posibile de rezolvare a problemei studiate.
- Următoarele etape ale algoritmului urmăresc să determine forma explicită a expresiei analitice sau a codului informatic prin care, pornind de la variabilele de intrare, se ajunge la variabilele de ieșire, formă care este inițial necunoscută.

Populația inițială

- Indivizi reprezentați sub formă de arbori cu
 - ❖ noduri - funcții și operatori matematici (+, -, *, cos, exp, ȘI, SAU)
 - ❖ terminale variabile și constante,
- toate alese dintr-un set disponibil.



Reprezentarea sub formă de arbore a funcției $\max(x+x, x+3*y)$ [1]

Populația inițială

- Ca și la algoritmele genetice, populația inițială este generată aleatoriu, sub formă de arbori pentru care se impun restricții de **adâncime** și **dimensiune**.
 - ❖ Adâncimea unui nod este numărul de laturi parcurs de la nodul cel mai de sus (numit rădăcină) până la nodul respectiv.
 - ❖ Dimensiunea unui arbore este numărul maxim de noduri care poate intra în componența sa.
- Uneori, reprezentarea tip arbore este înlocuită cu o reprezentare cu prefixe, care permite vizualizarea mai ușoară a subexpresiilor și a subarborilor. Folosind acest timp de reprezentare, de ex, notația $\max(x+x, x+3*y)$ se transformă în $(\max (+ x x) (+ x (* 3 y)))$.

Generarea populației inițiale

- Sunt folosite pe scară largă trei metode.
 - ❖ Metoda **full** generează întotdeauna arbori cu toate ramurile de adâncime egală cu adâncimea maximă admisă.
 - ❖ Metoda **grow** permite crearea unor arbori cu ramuri de adâncime mai mică sau egală cu cea maxim admisă.
 - ❖ Metoda **ramped half-and-half** generează jumătate din populație cu metoda full și cealaltă jumătate cu metoda grow și crește treptat adâncimea arborelui până la cea maxim admisă, permițând astfel folosirea unor arbori de diverse dimensiuni și forme în populația inițială.

Calculul funcției de adaptare

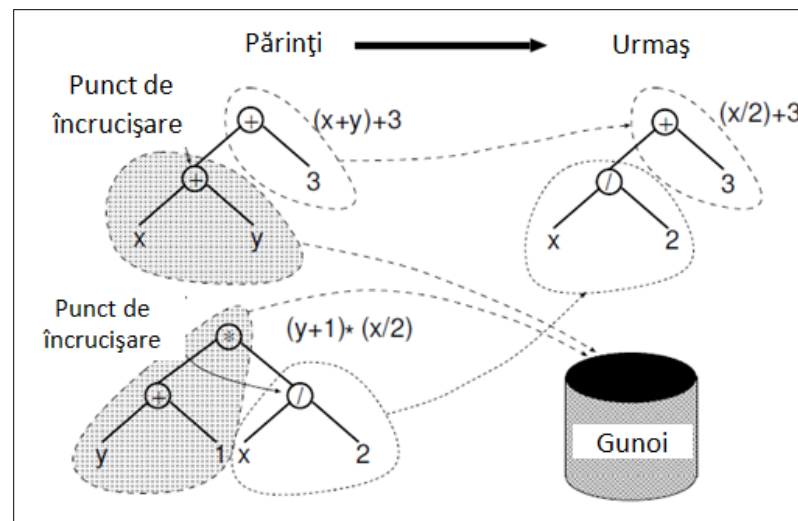
- **Calculul FA** are două etape:
 - ❖ evaluarea expresiilor matematice codificate în arbori
 - ❖ compararea rezultatului obținut cu cel dorit pentru toate modelele din setul de date de învățare. Indivizii cu funcțiile de adaptare cele mai mici sunt considerați cei mai bine adaptați.

Selecția

- alege din populația curentă indivizi pentru realizarea încrucișării.
- metoda preferată este selecția prin competiție, dar se pot folosi și altele
- Se realizează în paralel cu încrucișarea și mutația, adică se selectează perechile de indivizi, după care se realizează încrucișare și mutație, după care se selectează alți doi indivizi ș.a.m.d.
- De obicei, operatorii genetici se aplică până la generarea unui număr de urmași egal cu dimensiunea populației curente, însă există și implementări PG care permit creșterea progresivă a populației de la o iterație la alta.

Încrucișarea

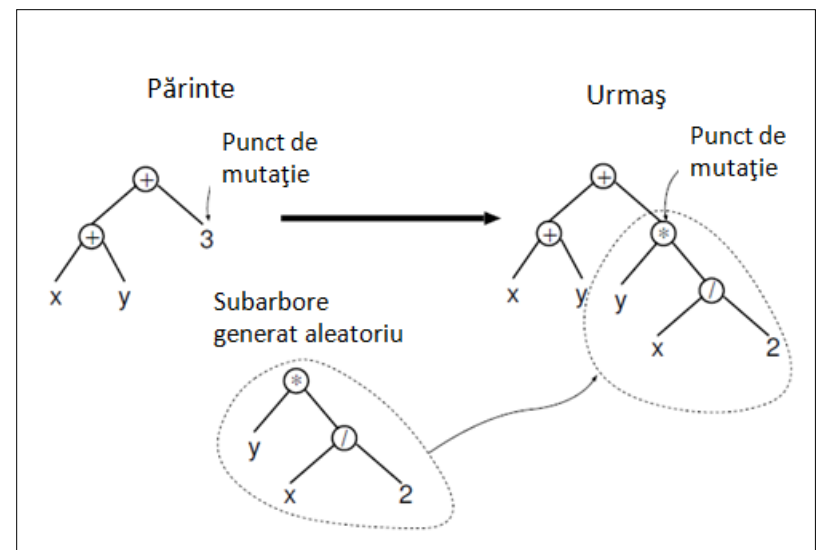
- Se utilizează cel mai des încrucișarea pe subarbori
 - ❖ se aleg un părinte donator și un părinte primitor.
 - ❖ se generează aleatoriu și independent pentru fiecare părinte câte un punct de încrucișare într-un nod,
 - ❖ părintelui primitor i se înlocuiește subarboarele de după punctul de încrucișare cu subarboarele primit de la părintele donator, rezultând un urmaș
 - ❖ părintele donator nu este distrus, ci încrucișarea se face pe o copie, pentru a se permite participarea aceluiași părinte la mai multe încrucișări.



Sursa: [1]

Mutația

- Mutația pe subarbore: care alege un nod la întâmplare și regenerează subarboarele de dedesubtul său cu unul nou, generat aleatoriu folosind setul de funcții și terminale avut la dispoziție
- O alt variantă de mutație este alegerea aleatoare a unei funcții sau a unui terminal din arbore și înlocuirea cu o funcție cu același număr de intrări sau un terminal ales la întâmplare din setul disponibil.



Mutația pe subarbore, sursa [1]

Încrucișarea și mutația

- La fel ca în cazul algoritmilor genetice, încrucișarea și mutația se efectuează cu o anumită probabilitate aleasă de utilizator, definită prin intermediul unor **rate de încrucișare și de mutație**.

Aplicații ale PG

- Aplicațiile teoretice programării genetice includ domenii ca [1]:
 - ❖ Identificarea relațiilor existente între variabile
 - ❖ Identificarea formei și dimensiunii soluțiilor unor probleme
 - ❖ Probleme dificil de rezolvat prin matematica convențională
 - ❖ Probleme în care identificarea unor corecții mici poate avea beneficii mari
 - ❖ Probleme în care soluția trebuie căutată folosind un volum mare de date
 - ❖ Probleme în care o soluție aproximativă este satisfăcătoare.
- În practică. PG a fost utilizată în domenii precum: procesarea semnalelor și a imaginilor, controlul proceselor industriale, modelarea tranzacțiilor economice și financiare, medicină și biotehnologii.

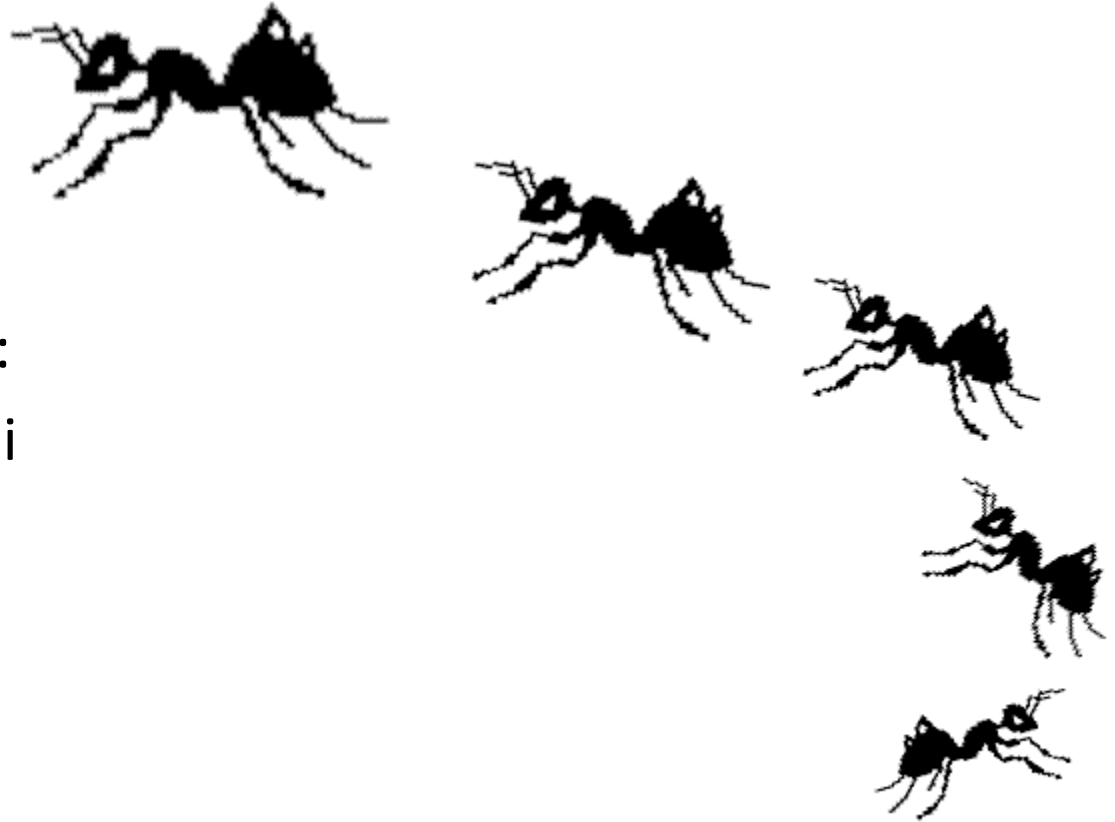
Bibliografie

[1] Riccardo Poli, William B. Langdon, and Nicholas F. McPhee - A Field Guide to Genetic Programming, 2008, <http://www.gp-field-guide.org.uk>

- Pentru aplicații:

[2] GPLAB - A Genetic Programming Toolbox for MATLAB, Sara Silva, ECOS - Evolutionary and Complex Systems Group, University of Coimbra, Portugal, Version 3, April 2007, <http://gplab.sourceforge.net/>

va urma...



Agenți inteligenți:
Algoritmul furnicii

Vă mulțumesc pentru atenție !