

## Experiente cu un algoritm evolutiv pentru rezolvarea problemelor de ordonantare pe masini paralele

Lect.dr. Adina UTA  
Catedra de Informatica Economica, A.S.E. Bucuresti

*În articol sunt prezentate rezultatele experimentelor efectuate cu un algoritm evolutiv pentru rezolvarea problemelor de ordonantare pe masini paralele. Cercetarile au fost efectuate în cadrul unui contract de cercetare cu Agentia Nationala pentru Stiinta, Tehnologie si Cercetare, cu tema "Modelarea si simularea procesului de productie folosind retele neuronale si algoritmi genetici" [1]. În cadrul cercetarii au fost atacate mai multe tipuri de probleme de ordonantare si a fost elaborata o colectie de metode si modele bazate pe retele neuronale si genetice. Ulterior ele au fost implementate si au fost efectuate experiente pe baza de date din literatura de specialitate [7].*

**Cuvinte cheie:** experiente, open shop, ordonantare pe masini paralele, algoritmi evolutivi

**P**entru problema ordonantarii pe masini paralele a fost propus un algoritm genetic, care, pentru un caz concret, decurge în felul urmator:

- se realizeaza o initializare a populatiei prin intermediul functiei "InitializePopulatie"
- se determina fitness-ul fiecarui cromozom din populatie prin intermediul functiei "EvaluezFitness(cromozom i)";
- atâta timp cât numarul de generatii nu depaseste un anumit numar dat ca parametru :
  - din populatia veche se pastreaza cei mai buni cromozomi pentru a face parte din noua populatie, ordonând crescator cromozomii în functie de fitness;
  - din toata populatia veche se selecteaza apoi parintii, prin intermediul algoritmului Wheel Roulette Selection;
- cromozomii selectati doi câte doi se combina prin intermediul unui operator de crossover. Recombinarea prin crossover se face cu o probabilitate,  $p_c$ , pentru fiecare generatie. Se aplica Single Point Crossover.
- punctul de crossover este determinat tot aleator, generând un întreg pozitiv în intervalul  $[1, m-1]$ . Copiilor formati prin crossover sau parintilor, în cazul în care nu s-a aplicat crossover-ul, li se aplica apoi ope-

ratorul de mutatie cu o probabilitate,  $p_m$ , astfel ca fiecare gena din toti cromozomii populatiei are sanse egale de a suferi o mutatie.

### Probleme de test

Problemele de test utilizate au fost obtinute cu ajutorul unui generator preluat de pe Internet, adresa de la **Imperial College, Management School** anonymous ftp la [mscmga.ms.ic.ac.uk](http://mscmga.ms.ic.ac.uk), pagina web: <http://mscmga.ms.ic.ac.uk>

Numarul de job-uri variaza între 5 si 20, numarul masinilor între 5 si 20, numarul de tipuri de probleme este 3, iar daca din fiecare tip se considera 5 instante, avem 15 probleme distincte de considerat.

### Notatii si parametrii de testare

Parametrii de testare utilizati sunt urmatoorii: numarul de masini, numarul de job-uri, timpii pe operatii, marimea populatiei, numarul maxim de generatii, facilitatea de elitism (de pastrare a celor/celui mai buni cromozomi din vechea populatie în noua populatie) prin intermediul unui singur cromozom sau a unui numar (setat ca procent) de cromozomi.

Facilitatea de crossover permite setarea a trei parametri: rata initiala de crossover,

modificarea ratei la fiecare iteratie, rata minima de crossover.

Rezultatele obtinute sunt ilustrate prin intermediul urmatoarelor informatii: numarul curent de generatii atins, numarul încrucișarilor, rata la care a ajuns crossover-ul, numarul mutatiilor efectuate precum si rata mutatiilor care este fixa, timpul minim de executie a job-urilor pe masini, timpul mediu, timpul maxim.

		Var.1	Var.2	Var.3	Var.4
Variaza mutatia	VM	0,05 %	0.1 %	0.2 %	0.3 %
Variaza crossover	VX	0.6 %	0.5 %	0.4 %	0.3 %
Variaza elitismul	VE	0.5 %	0.1 %	0.15 %	0.20 %

**Rezultatele testelor**

Calitatea solutiilor se reprezinta în procent de abatere a valorilor functiilor scop fata de valorile cele mai bine cunoscute din literatura.

În timpul testarilor s-a studiat dependenta calitatii solutiilor de caracteristicile problemelor de test. Problemele de test fiind caracterizate de  $n$  – numarul job-urilor,  $m$  – numarul masinilor, media duratelor de prelucrare si dispersia duratelor de prelucrare fata de mesia lor.

Tabel 1

Dimensiune Nr. Instanta	5x5	10x10	20x20
1	3%	18%	35%
2	8%	17%	
3	26%	25%	
4	12%	7%	
5	22%	19%	
Calitate medie	14%	17%	35%

Se constata faptul ca, odata cu cresterea dimensiunii problemelor se pierde din calitatea solutiei, adica la probleme de dimensiuni mari, procentul de abatere a valorii functiei de eficienta fata de cele mai bune valori cunoscute creste.

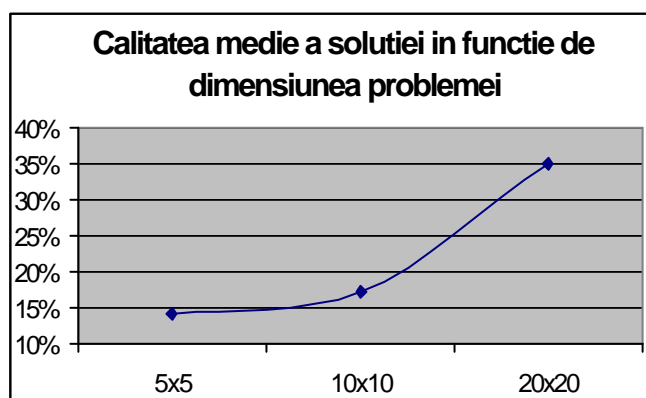
**Plan de testare**

- Parametrii constanti:
  - dimensiunea populatiei – 200 indivizi
  - numar de generatii: 300 (pentru probleme de 5 x 5), 500 (pentru probleme de 10 x 10), 1000 (pentru probleme de 20 x 20)
- Fiecare problema:

În continuare s-a studiat dependenta calitatii solutiilor fata de parametrii algoritmului evolutiv folosit.

Dintre parametrii algoritmului evolutiv s-au considerat procentul de elitism, probabilitate de mutatie si probabilitate de încrucișari.

Calitatea medie a solutiei în functie de dimensiunea problemelor este reprezentata în tabelul 1, respectiv figura 1.



**Fig.1.** Calitatea medie a solutiei

Dependenta calitatii solutiei fata de dispersia duratelor de prelucrare a reperelor pe masini este reprezentata în tabelul 2 si figura 2 pentru problemele de dimensiune 5 x 5, si în figura 2a pentru problema de dimensiune 10 x 10.

Tabel 2

	Instanta				
<b>5x5</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Dispersia duratelor	29.50813	28.17074	24.94455	24.32197	24.75693
Calitatea solutiei	3%	8%	26%	12%	22%
	Instanta				
<b>10x10</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Dispersia duratelor	27.44815	29.82197	27.36594	28.08207	29.13311
Calitatea solutiei	18%	17%	25%	7%	19%
	Instanta				
<b>20x10</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
Dispersia duratelor	28.42795				
Calitatea solutiei	35%				

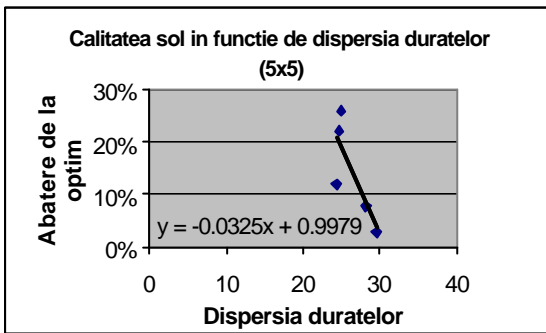


Fig.2. Calitatea solutiei în functie de dispersia duratelor

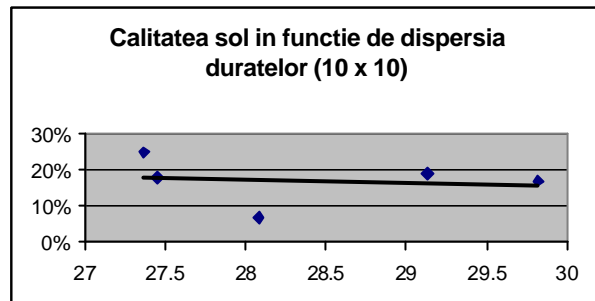


Fig.2a. Calitatea solutiei în functie de dispersia duratelor

În figurile 2 si 2a se poate constata ca dispersia duratelor de prelucrare si calitatea solutiilor sunt invers proportionale, adica se obtin solutii calitativ mai bune la dispersii mai mari ale duratelor de prelucrare.

În ceea ce priveste parametrii algoritmului evolutiv, studiul procentului de elitism asupra valorii functiei scop, are o tendinta de scadere, indiferent de dimensiunea problemei (vezi tabelul 3 si corespunzator figurile 3, 3a, 3b si 3c).

Tabelul 3

5x5 Instanta 1				
	<b>ve1</b>	<b>ve2</b>	<b>ve3</b>	<b>ve4</b>
<b>Elitism</b>	5%	10%	15%	20%
Solutie medie	337.8	330.2	333.2	330.4
	<b>vm1</b>	<b>vm2</b>	<b>vm3</b>	<b>vm4</b>
<b>Mutatie</b>	5%	10%	20%	30%
Solutie medie	340.4	332.4	324.2	317.6
	<b>vx1</b>	<b>vx2</b>	<b>vx3</b>	<b>vx4</b>
<b>Crossover</b>	60%	50%	40%	30%
Solutie medie	328.4	333.8	341.6	337.2

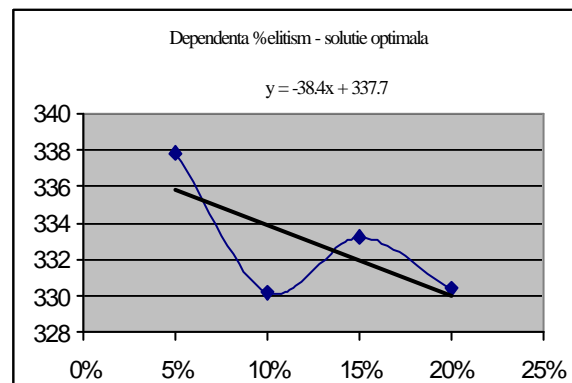
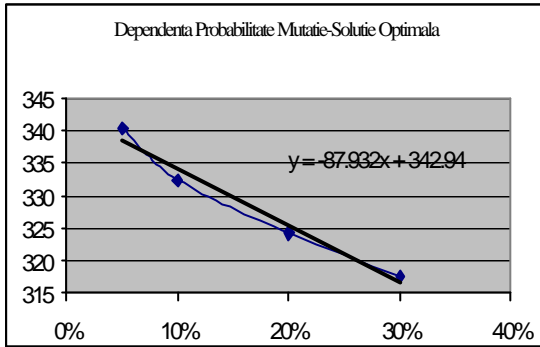
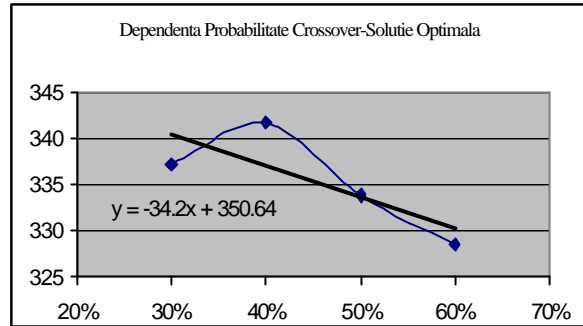


Fig. 3. Dependenta procent de elitism-solutie optima



**Fig.3a.** Dependenta mutatie-Solutie optimala



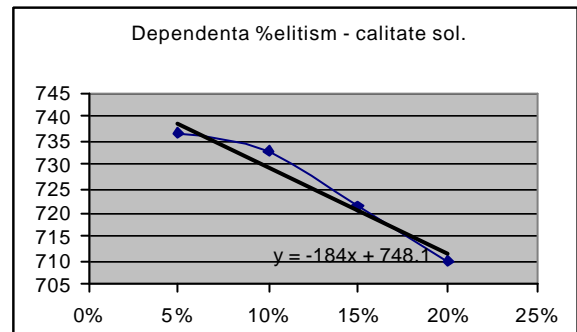
**Fig.3b.** Dependenta Probabilitate Crossover-Solutie optimala

Procentul de mutatii nu a furnizat rezultate concludente, adica la dimensiuni diferite ale problemelor, în functie de procentul de mutatii, s-a obtinut valoarea functiei de eficienta nesemnificativa. În acest caz se

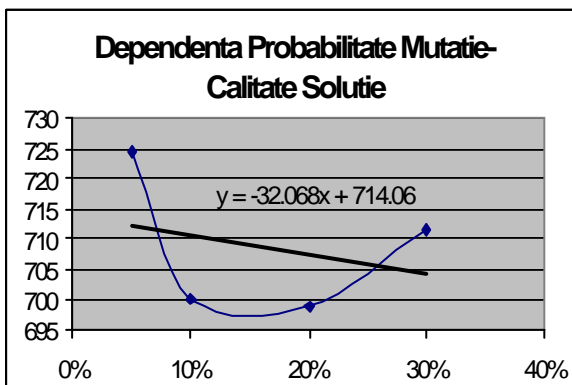
recomanda studiul pe intervale de procente de mutatii, cum ar fi între 0 si 10%. Rezultatele sunt interpretabile, cum se poate observa din figurile 4 si 4a.

Tabelul 4

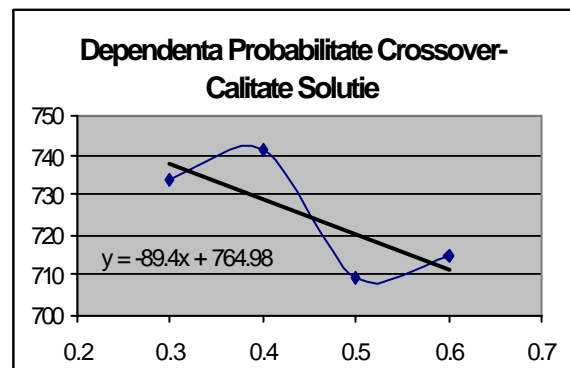
10x10 Instanta 4				
	ve1	ve2	ve3	ve4
<b>Elitism</b>	5%	10%	15%	20%
Solutie medie	736.6	732.8	721.2	709.8
	vm1	vm2	vm3	vm4
<b>Mutatie</b>	5%	10%	20%	30%
Solutie medie	724.6	700	699	711.8
	vx1	vx2	vx3	vx4
<b>Crossover</b>	0.6	0.5	0.4	0.3
Solutie medie	714.8	709.2	741	734



**Fig.4.** Dependenta procent elitism-calitate solutie



**Fig.4a.** Dependenta probabilitate mutatie-calitate solutie



**Fig.4b.** Dependenta probabilitate crossover-calitate solutie

Probabilitatea de încrucisari la problemele de dimensiune 5 x 5 si 10 x 10, prezinta tendinte asemanatoare cu procentul de

Tabelul 5

20x20 Instanta 1				
	ve1	ve2	ve3	ve4
<b>Elitism</b>	5%	10%	15%	20%
Solutie medie	1638	1623.6		
	vm1	vm2	vm3	vm4
<b>Mutatie</b>	5%	10%	20%	30%
Solutie medie	1591.4	1605	1590.4	1598.8
	vx1	vx2	vx3	vx4
<b>Crossover</b>	0.6	0.5	0.4	0.3
Solutie medie	1624.6	1650.8	1611.8	1622

elitism si procente de mutatii (mai mici de 10%). Rezultatele acestor influente se pot vedea în figurile 5 si 5a.

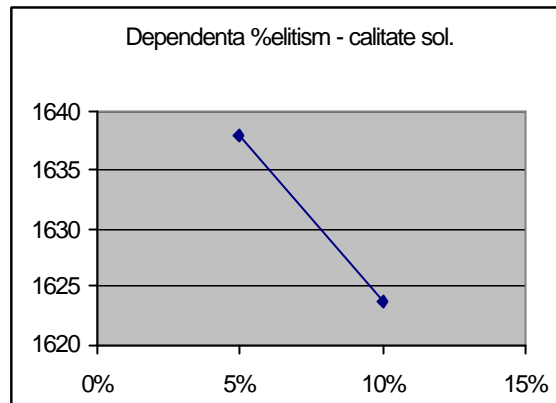


Fig.5. Dependenta procent elitism-calitate solutie

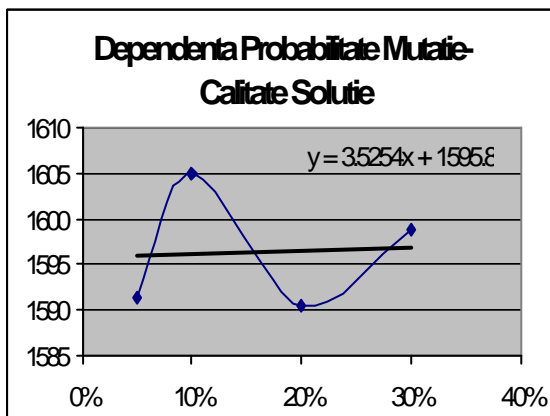


Fig.5a. Dependenta probabilitate mutatie-calitate solutie

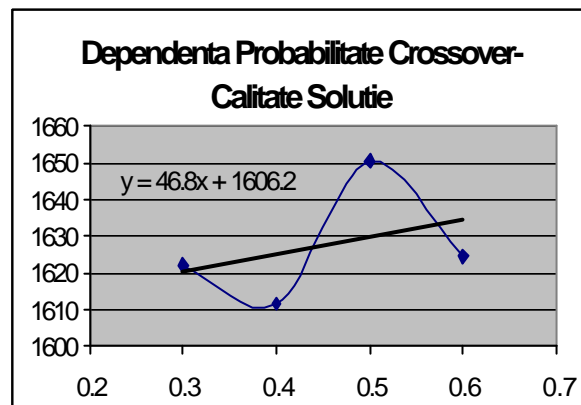


Fig.5b. Dependenta probabilitate crossover-calitate solutie

Graficele au fost obtinute pe valori medii de 5 rulari pe 5 probleme generate pe aceeasi dimensiune.

Precizam ca numarul generarilor a fost variat, în functie de dimensiunea problemelor. Astfel, la probleme de 5x5 au fost 300 de generari, la probleme de 10x10 au fost 500 generari, iar la probleme de 20x20 au fost 1000 de generari, populatia fiind mentinuta constant la 200 de indivizi. Numarul problemelor folosite pentru aceste experimentari au fost 300 pentru dimensiuni 5x5 si 10 x 10 si 100 pentru dimensiunea 20x20.

### Concluzii

Se recomanda utilizarea cu precautie a problemelor de dimensiuni mari (eventual ele sa fie descompuse în probleme de dimensiuni mai mici, aplicând un principiu de divide et impera). De asemenea, dispersia duratelor de prelucrare trebuie sa fie mai mare, ceea ce se poate obtine prin multiplicarea cu factor a duratelor, respectiv reprezentarea într-o unitate de masura mai mica. Se recomanda folosirea algoritmului evolutiv în problemele de OpenShop cu procentul de elitism mai mare sau egal cu 20%, procentul de mutatie simpla sa fie aproxi-

mativ de 10%, iar procentul de încrucisari sa fie cât mai mare, respectiv mai mare sau egal cu 60%.

Este de asteptat ca timpul de calcul sa creasca cu încrucisari procentuale mai mari si cu dimensiune mai mare a problemelor.

### **Bibliografie**

[1] Fabian, Cs., Mihalca R., Nica V., Uta A. – *Modelarea si simularea a procesului de productie folosind retele neuronele si algoritmi genetici*, contract cu MCT, Bucuresti, 1996-1999

[2] S. Jung – *Genetische Algorithmen in der simulations-untertuetzte Produktionsplanung, Op.Res/97*

[3] E. Mayrand – *A genetic search algorithm to optimize job sequencing under a technological constraints in a rolling-mill facility*, OR-Spectrum, 17/1995

[4] \*\*\* – *Evolutionary Computation, MIT Journal (1993-1999)*

[5] \*\*\* – *Genetic and Evolutionary Computation Conference, July 1999, Orlando, Florida, USA*

[6] \*\*\* – *GAlib- A C++ Library of Genetic Algorithm Components*

[7] \*\*\* – *Imperial College, Management School: <http://mscmga.ms.ic.ac.uk>*