

# Tehnici de optimizare in programare

## Raport tema de curs

Chisalescu Bogdan si Chivu Gheorghe-Iulian  
Grupa 432A

### 1 Introducere

Algoritmii genetici sunt o subclasa a algoritmilor evolutivi ce se pot aplica in orice problema de optimizare. Acestia sunt algoritmi matematici inspirati din teoria evolutiei speciilor ce se bazeaza pe un set de caracteristici. Acest set include existenta unei populatii (indivizii reprezentand posibile solutii), o modalitate de a genera noi indivizi in populatie (prin recombinarea a doi indivizi deja existenti si/sau mutatia unui individ) si o modalitate ce simuleaza selectia naturala (un algoritm prin care se decide ce indivizi raman in populatie si modul in care se modifica dimensiunea populatiei). Prezenta lucrare are ca scop implementarea unui algoritm genetic specific apartinand unei lucrari de studiu de cercetare. Lucrarea de cercetare se intituleaza "An Effective Real-Parameter Genetic Algorithm with Parent Centric Normal Crossover for Multimodal Optimisation" si este realizata de Pedro J. Ballester si Jonathan N. Carter in anul 2004.

Urmatoarele sectiuni ce vor fi descrise sunt o imagine a modului in care am reusit sa intelegem notiunile teoretice din spatele acestui studiu, a modului in care am reusit sa implementam algoritmul si in care am interpretat rezultatele obtinute; si nu o reprezentare exacta a studiului, astfel am incercat pe cat posibil sa reproducem algoritmul si performantele sale.

## 2 Descrierea Algoritmului

Algoritmul descris de studiul de cercetare se intituleaza "Scaled Probabilistic Crowding Genetic Algorithm with Parent Centric Normal crossover" la care ne vom referi pe tot parcursul lucrarii ca SPC-PNX.

Acest model de algoritm genetic foloseste un numar constant de indivizi care fac parte din populatie. In fiecare generatie doi parinti sunt selectati din aceasta populatie pentru a crea  $\lambda$  copii prin metoda crossover. Apoi se elimina din populatie sau din copii creati astfel incat populatia sa ramana constanta.

### 2.1 Selectia

In general pentru algoritmi genetici, selectia nu depinde de valoare functiei fiecarui individ. De aceea folosim o selectie aleatoare dupa o distributie uniforma, pentru a selecta doi parinti din populatia curenta.

### 2.2 Crossover

Pentru fiecare  $\lambda$  copil creat prin aceasta metoda procedam astfel pentru a-i calcula componenta  $j$  ( $y_j$ ) : Prima data tragem un numar aleator  $\omega$  care apartine intervalului  $[0,1]$ , daca  $\omega < 0.5$  atunci folosim forma  $y_j^{(1)}$ , daca  $\omega \geq 0.5$  atunci folosim forma  $y_j^{(2)}$ . Odata ce una din forme a fost aleasa, aceasta va fi folosita pentru toate cele  $j$  componente ale copilului. Aceste forme se calculeaza astfel:

$$y_j^{(1)} = N(x_j^{(1)}, |x_j^{(2)} - x_j^{(1)}|/\eta) \quad y_j^{(2)} = N(x_j^{(2)}, |x_j^{(2)} - x_j^{(1)}|/\eta)$$

Unde:

- $N(\mu, \sigma)$  reprezinta un numar aleator dintr-o distributie normala cu media  $\mu$  si dispersia  $\sigma$
- $x_j^{(i)}$  reprezinta componenta numarul  $j$  a parintelui  $i$
- $\eta$  este un paramentru variabil. Cu cat  $\eta$  este mai mare, copilul se afla mai aproape de zona parintilor.

### 2.3 Selectia (Scaled Probabilistic Crowding Replacement)

Se selecteaza aleator  $\kappa$  indivizi din populatia curenta, pentru a concura cu cei  $\lambda$  copii pentru un loc in populatie.

Pentru fiecare copil ( $x^{ofp}$ ) se determina care individ din cei  $\kappa$  selectati se afla mai aproape de el ( $x^{cst}$ ). Apoi se calculeaza pentru fiecare probabilitatea de eliminare din populatie astfel:

$$p(x^{ofp}) = \frac{f(x^{ofp}) - f_{best}}{f(x^{ofp}) + f(x^{cst}) - 2f_{best}} \quad p(x^{cst}) = \frac{f(x^{cst}) - f_{best}}{f(x^{ofp}) + f(x^{cst}) - 2f_{best}}$$

Unde  $f_{best}$  este valoarea functiei celui mai bun individ din cei  $\lambda$  copii si cei  $\kappa$  indivizi selectati.