

Algoritmi metaeuristici.

Lab 6: Optimizare multicriterială

1. Optimizare multicriterială.

Optimizarea multicriterială are ca scop optimizarea simultană a mai multor criterii. Funcția obiectiv este de forma $F: \mathbb{R}^n \rightarrow \mathbb{R}^r$, iar componentele sunt $F=(f_1, f_2, \dots, f_r)$.

Criteriile de optimizat sunt de regulă conflictuale, astfel că nu există o soluție unică a problemei. În aceste condiții se caută soluții în sens Pareto, care se caracterizează prin faptul că nu există alte configurații care să fie mai bune în raport cu toate criteriile de optimizat (orice îmbunătățire în raport cu unul dintre criterii conduce la o înrăutățire în raport cu un alt criteriu). Astfel de soluții se numesc nedominate (în sensul că nu există alte soluții care să le domine în raport cu fiecare dintre criteriile de optimizat).

Există mai multe abordări ale unei astfel de probleme. Principalele categorii de metode sunt:

- *Metode bazate pe tehnica agregării:* se transformă problema inițială de optimizare multicriterială într-una de optimizare uni-criterială prin combinarea criteriilor. Noua funcție de optimizat este $f(x)=w_1f_1(x)+w_2f_2(x)+\dots+w_rf_r(x)$ unde w_1, w_2, \dots, w_r sunt ponderi asociate criteriilor. Pentru fiecare set de ponderi se poate obține o altă soluție.
- *Aproximarea directă a mulțimii Pareto optimale:* se utilizează o populație de elemente care vor aproxima mulțimea Pareto optimală (mulțimea tuturor soluțiilor nedominate).

Exemple de funcții test utilizate pentru a analiza performanțele algoritmilor de optimizare multicriterială sunt disponibile la: http://en.wikipedia.org/wiki/Test_functions_for_optimization sau la <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/>

Aplicație 1. Se consideră funcția $F:[0,4] \rightarrow \mathbb{R} \times \mathbb{R}$, $F(x)=((x-1)^2, (x-2)^2)$. Să se aproximeze mulțimea Pareto optimală și frontul Pareto corespunzător (frontul Pareto reprezintă mulțimea valorilor funcțiilor obiectiv corespunzătoare elementelor din mulțimea Pareto optimală).

Varianta 1. Se aplică tehnica agregării

- a) se construiește funcția obiectiv:

```
function y=fw(x)
    w=0.1;
    y1=(x-1)*(x-1);
    y2=(x-2)*(x-2);
    y=w*y1+(1-w)*y2;
endfunction
```

- b) se aplică succesiv o strategie evolutivă (de exemplu cea descrisă în [SE.sci](#) – vezi lab3) pentru optimizarea funcției obiectiv pentru următoarele valori ale ponderii w : (0.1, 0.2, 0.3, ..., 0.9) și se stochează valoarea într-un vector x
- c) comparați rezultatele obținute folosind alta strategie evolutivă de exemplu DE
- d) se vizualizează valorile funcțiilor obiectiv pentru valorile stocate la pasul anterior (va reprezenta o aproximare a frontului Pareto):

```
function pareto(x)
    f1=(x-1).^2;
    f2=(x-2).^2;
```

```
plot(f1, f2, '*');  
endfunction
```

E) afisati doar elementele nedominante ale frontului parento

Varianta 2. Se folosește implementarea algoritmului NSGA-II (funcția `optim_nsga2` în Scilab) sau a algoritmului MOGA (funcția `optim_moga` în Scilab).

Indicație: vezi `exNSGA.sci`

Exercitiu: Să se analizeze comportamentul algoritmilor MOGA și NSGA2 pentru funcțiile de test ZDT1, ZDT3 (de la <http://people.ee.ethz.ch/~sop/download/supplementary/testproblems/>)