

Lucrarea 5.a

Algoritmi genetici

1. Baza teoretica

Introducere. Algoritmii genetici sunt tehnici adaptive de căutare euristică, bazate pe principiile geneticii și ale selecției naturale, enunțate de Darwin (supraviețuiește cel mai bine adaptat). Mecanismul este similar procesului biologic al evoluției. Acest proces posedă o trăsătură prin care numai speciile care se adaptează mai bine la mediu sunt capabile să supraviețuiască și să evolueze peste generații, în timp ce acelea mai puțin adaptate nu reușesc să supraviețuiască și cu timpul dispar, ca urmare a selecției naturale. Probabilitatea ca specia să supraviețuiască și să evolueze peste generații devine cu atât mai mare cu cât gradul de adaptare crește, ceea ce în termeni de optimizare înseamnă că soluția se apropie de optim.

Ca și aplicații practice, algoritmii genetici sunt cel mai adesea utilizați în rezolvarea problemelor de optimizare, planificare ori căutare. Condiția esențială pentru succesul unei aplicații cu agenți inteligenți este ca problema de rezolvat să nu ceară obținerea soluției optime, ci să fie suficientă și o soluție apropiată de optim.

Terminologie. Algoritmii evolutivi utilizează un vocabular împrumutat din genetica:

- evoluția este simulată printr-o succesiune de **generații** ale unei **populații** de soluții candidat;
- o soluție candidat poartă numele de **cromozom** și este reprezentată ca un sir de gene;
- populația evoluează prin aplicarea **operatorilor genetici**: **mutația** și **incrucisarea**;
- cromozomul asupra căruia se aplică un operator genetic se numește **parinte** iar cromozomul rezultat se numește **descendent**;
- **selectia** este procedura prin care sunt aleși cromozomii ce vor supraviețui în generația următoare; indivizilor mai bine adaptați li se vor da șanse mai mari;
- gradul de adaptare la mediu este măsurat de **funcția fitness**;
- **soluția** returnată de un algoritm genetic este cel mai bun individ din ultima generație.

A. Functia fitness. Functia fitness este utilizata pentru a masura calitatea cromozomilor. Ea este formulata plecand de la functia numerica de optimizat.

B. Selectia. Stabileste cei mai buni cromozomi din populatie. Cea mai utilizata metoda de selectie este selectia proportionala sau **principiul ruletei** :

1. Se stabileste functia de evaluare $fitness(x_i)$, pentru fiecare cromozom x_i din populatie
2. Se sumeaza toate functiile de evaluare $fitness = \sum(fitness(x_i))$
3. Cromozomilor li se atribuie aleator numerele naturale i
4. Se alege cromozomul x_i , (unde i este cel mai mic numar care satisface relatia $\sum(fitness(x_j)) > n$) pana la generarea pseudopopulatiei de N cromozomi

Exemplu de folosire a ruletei pentru aflarea maximului functiei $f(x) = -\frac{1}{4}x^2 + 2x + 5$, pentru 5 cromozomi reprezentati pe 10 biti.

Nr.	Cromozomi	Valoare ₁₀	X	Fitness f(x)	% din Total
1	0001101011	107	1.05	6.82	31
2	1111011000	984	9.62	1.11	5
3	0100000101	261	2.55	8.48	38
4	1110100000	928	9.07	2.57	12
5	1110001011	907	8.87	3.08	14
TOTAL				22.05	100

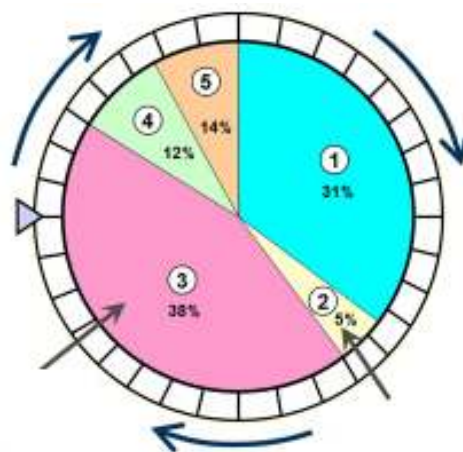


Figura 1.1 Exemplu de folosire a ruletei pentru selectie

Selectia bazata pe rang previne convergenta prematura ce apare la utilizarea schemelor de selectie proportionala cu fitnessul. Indivizii sunt ordonati in functie de valoarea fitnessului iar probabilitatea de selectie este proportionala cu rangul ocupat. Presiunea de selectie este in acest mod scazuta in cazul in care varianta fitnessului este mare si este crescuta daca varianta fitnessului este mica.

Selectia turneu alege in mod aleatoriu k indivizi iar dintre acestia doar cei mai buni j sunt selectati pentru supravietuire. Procedura se repeta pana se obtine numarul dorit de indivizi. Este cea mai eficienta din punct de vedere al complexitatii timp. Din punct de vedere al presiunii de selectie se aseamana selectiei bazata pe rang.

Elitism. Trece cel mai bun cromozom in generatia urmatoare fara a fi alterat. Acest lucru garanteaza faptul ca cel mai puternic membru al populatiei din generatia $t + 1$ nu va fi mai slab decat cel din generatia t .

Operatori genetici

C. Incrucisarea. Stabileste modul de schimbare a materialului genetic intre cromozomii parinti.

Poate fi :

- incrucisare cu un punct de taietura

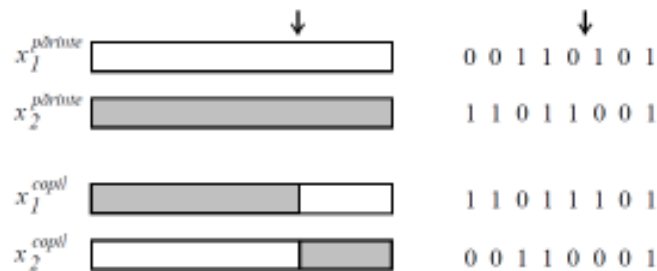


Figura 1.2. Exemplu de incrucisare cu un punct de taietura

- incrucisare cu mai multe puncte de taietura

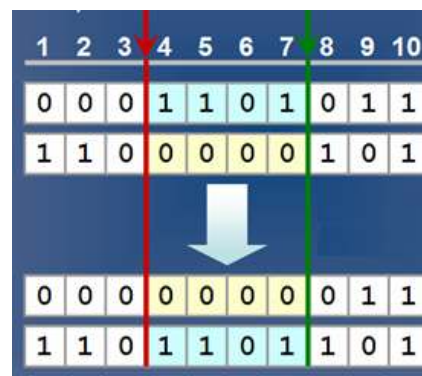
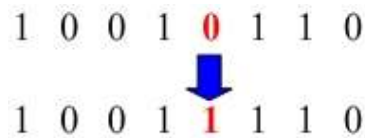


Figura 1.3. Exemplu de incrucisare cu 2 puncte de taietura

Incrucisarea se face cu o probabilitate P_c , între 60% și 100%. Adică din cei N cromozomi ai pseudopopulației, $P_c\%$ din ei vor fi încrucișați, restul trecând direct în generația următoare. Să fie număr par.

D. Mutatia. Permite găsirea unor soluții noi.

Pentru fiecare bit al fiecărui cromozom din generația următoare, se generează un număr random r și se modifică bitul respectiv dacă $r < P_m$ unde P_m este de obicei o probabilitate foarte mică.



Structura unui algoritm genetic standard :

1. Se initializează populația de cromozomi (aleator), numărul de generații, P_c și P_m
2. Se evaluează fiecare cromozom din populație utilizând funcția *fitness*
3. Se creează o nouă generație de cromozomi folosind : selecție, încrucișare, mutație
4. Se șterge o parte din membrii populației actuale, pentru a fi înlocuiți cu cei din nouă generație
5. Se evaluează noii cromozomi și se înserează în nouă populație
6. Dacă timpul de căutare nu s-a terminat, se merge la pasul 3. În caz contrar, se oprește execuția algoritmului.

2. Problema

Fie funcția $f(x) = -\frac{1}{4}x^2 + 2x + 5$. În vederea aflării valorii lui x în intervalul $0..10$ care maximizează funcția, se aplică algoritmi genetici. În cadrul acestei metode, pasul de cuantizare al spațiului de soluții este 0.2 și se pornește cu următoarea populație de cromozomi :

C1 : 001000

C2 : 101111

C3 : 010110

C4 : 101001

C5 : 011000

Să se construiască ruleta de selecție.

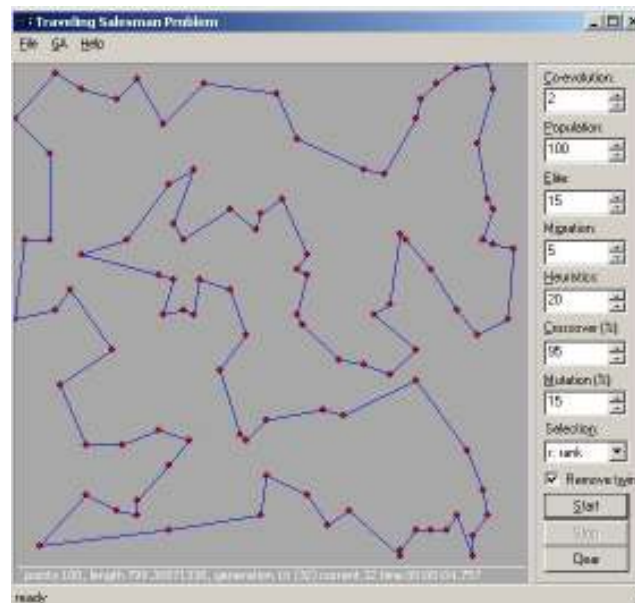
3. Aplicatie de laborator

Problema Comis Voiajorului folosind algoritmi genetici

Problema *Comis Voiajorului* este intalnita in literatura engleza sub denumirea de *Travelling Salesman Problem* (TSP) si se enunta astfel : dandu-se o multime de orase si costul calatoriei dintre oricare doua orase, comis voiajorul trebuie sa gaseasca drumul de cost minim astfel incat sa viziteze toate orasele si sa se intoarca in orasul din care a plecat, dar sa nu treaca de 2 ori prin acelasi oras.

Desfasurarea lucrarii :

- Se va porni aplicatia TSPApp (program implementat de Konstantin Boukreev) ce rezolva problema comis voiajorului folosind algoritmi genetici
- Se vor initializa random pozitiile oraselor
- Se vor varia urmatorii parametri si se vor compara rezultatele (drumul minim gasit si timpul de calcul):
 - o Co-evolution: intre 1 si 16
 - o Population: numar de cromozomi intre 10 si 1000
 - o Elite: de la 0 la dimensiunea populatiei.
 - o Migration: de la 0 la dimensiunea populatiei.
 - o Crossover: probabilitatea de crossover, de la 0 la 100
 - o Mutation: probabilitatea de mutatie, de la 0 la 100
 - o Selection: una dintre metodele de selectie: ruleta, turnir, rang



Bibliografie

- [1] *The basic algorithm for a GA*, Newcastle University,
<http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g01.php>
- [2] *Genetic Algorithm and Traveling Salesman Problem*, Konstantin Boukreev
<http://www.generation5.org/content/2001/tspapp.asp>