Problema Domino

**Proiectare problema:**

M-am gandit sa rezolv problema privind paritatea pieselor de domino folosind permutari.

**Genele individului** sunt reprezentate prin permutari care ne arata ordinea de verificare a conditiei de paritate a pieselor de domino.

Lungimea invidului( **numarul de gene**) este egala cu lungimea listei de liste. ( Cate piese de domino avem)

*Exemplu:*

ld=[[9,1],[4,4],[3,1],[3,5],[7,8],[2,4]]  
" 0 1 2 3 4 5"

*Secventa de mai sus are o lungime de 6, asta inseamna ca avem 6 piese de domino. Cele 2 valori din fiecare sublista reprezinta valorile de pe piesa de domino.*

**Spațiul soluțiilor** este reprezentat de mulțimea tuturor permutărilor posibile ale punctelor din lista de piese de domino. Adica**, elementele permutarilor va lua valori intre 0 si lungimea listei de piese de domino.**

**Functia fitness:**

Initializam un contor cu valoarea 0 pe care il incrementam de fiecare data cand gasim 2 piese alaturate au aceeasi paritate.

O piesa de domino este formata din 2 valori. Pentru prima piesa, verificam daca a doua valoare inscrisa pe piesa are acceasi paritate cu prima valoare de pe piesa urmatoare. Daca este adevarat, incrementam **contorul pe care il returnam** la finalul functiei fitness.

În cazul de față, **funcția "fitness" este o funcție de maximizare**, deoarece se dorește găsirea unei soluții care să maximizeze numărul de piese de domino care respectă condiția dată. **Cu cât mai mare este valoarea returnată de funcția "fitness", cu atât soluția este considerată mai bună.**

**Populatia:**

Alegem ca dimensiunea poulatiei sa o dam ca parametru in functia de generare. Pentru cazul meu, o sa aleg o dimensiune de 20 de indivizi.

**Selectia parintilor**:

Selectam parinti din populatia actuala prin mecanismul ruleta, pentru ca acordam o probabilitate de selectie mai mare indivizilor cu scor fitness mai bun (fps). Vom forma perechile de parinti luand la rand parintii selectati doi cate doi.

**Recombinare: (Probabilitate 0.7)**

Folosesc operatorul de recombinare prin permutari **OCX.** Acesta presupune alegerea aleatoare a doua puncte p1 si p2. Copiem in primul copil valorile dintre p1 si p2 din primul parinte. Dupa asta, copiem valorile neutilizate in primul copil, din cel de-al doilea parinte incepand de la p2, in ordinea aparitiei lor.

**Mutatie:(Probabilitate: 0.1)**

Alegem **interschimbarea** ca operator de mutatie specific permutarilor. Aceasta consta in alegerea a doua pozitii din permutare ale caror valori se interschimba.

**Selectia indivizilor care raman / sunt transmisi mai departe in urmatoarea generatie:**

Selectia indivizilor din toti cromozomii rezultati se face pe baza mecanismului de **selectie elitism**, pentru a transmite in generatia urmatoare doar cei mai buni indivizi.

Elitismul presupune compararea populatiei de copii cu populatia initiala prin prisma celui mai bun individ din fiecare dintre cele doua populatii.

**Conditii de terminare ale algortimului genetic:**

1. (NMAX) Daca este atins numarul maxim de generatii (Eu am ales 20)
2. Daca maximul din generatia actuala nu se imbunatateste pe parcursul NMAX/4
3. Daca cel mai bun individ din generatia actuala are scorul fitness egal cu cel mai slab individ din generatia actuala (toti indivizii din populatia mea au acelasi scor fitness)
4. Conditie de terminare cand se gaseste un individ cu fitness maxim, adica cu valoarea 5..

