**Algoritmi Genetici**

* algoritmii genetici sunt folosiți în cazul problemelor de optim pentru care nu avem niciun algoritm fezabil de a le rezolva.

Cum se procedeaza?

Avem un spațiu al soluțiilor, iar algoritmul va căuta în spațiul tuturor soluțiilor posibile, dar mai ales se va concentra asupra unor puncte de maxim local.

Abordarea este inspirată din modelul evolutionist din biologie:  
succesiune de populații (mai multe generații) fiecare dintre ele fiind mai aptă (fit).

**Problema rucsacului:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | G | H | Capacitate  Rucsac |
| Val: | 3 | 5 | 10 | 4 | 8 | 6 | 8 | 7 | 30 |
| G: | 2 | 4 | 12 | 6 | 10 | 7 | 6 | 4 |

Putem codifica o soluție candidat pentru problema rucsacului ca un vector binar de elemente.

EX: 1 1 0 0 1 0 0 1 - adică selectez A, B, E și H, cu o valoare totală de 23 și greutate totala 20

O asemenea codificare se numește **individ/cromozom**

**Funcția de fitness** al unui cromozom arata “cât de buna” este soluția codificată de el

Ex: fitness(1 1 0 0 1 0 0 1)=23 (valoarea totala a elementelor selectate cand acestea incap in rucsac)

fitness(0 0 1 1 1 1 0 0)=0 (deoarece greutatea totală a elementelor selectate > capacitatea rucsacului).

Este important sa alegem o funcție de fitness care este cât mai bine croită pentru problema noastră.

În cazul rucsacului, o funcție de fitness “bună” ar fi următoarea: f(cromozom) = valoarea totala a obiectelor selectate codificate în cromozom, dacă acestea încap în rucsac, respectiv 0 altfel.

**Populatie:**

O mulțime de indivizi de la un moment dat

Acești indivizi pot fi comparati între ei cu ajutorul funcției de fitness.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | **23**/*20* |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | **0**/*31* |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | **19**/20 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | **22**/20 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | **25**/22 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | **24**/24 |
| **0** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **27/27** |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | **22**/23 |

Populația inițială este generata aleator (ex tabelul de mai sus)

Tot algoritmul se bazează pe o succesiune de generații:

Din populatia curentă selectez niște indivizi pentru a pune bazele populației viitoare.

Cum creăm populația viitoare:

* criteriul elitist: primul (un mic procent: aprox 5% dintre primii cei mai fit indivizi) individ va fi implicit trimis și în populația următoare (cel mai fit individ dintr-o generatie viitoare nu va fi niciodată mai prost decat cel mai fit individ dintr-o generatie precedentă)
* Criterii de selecție pentru indivizii care **vor contribui la** *(nu vor alcătui)* următoarea generatie:
  + criteriul ruletei: indivizii sunt “așezați” pe o ruletă, fiecare avand un slice din ruletă direct proporțional cu funcția sa de fitness.
  + criteriul turneului: sunt selectați aleator cate 2 indivizi din populatie, sunt puși să “concureze” unul cu celălalt. Cel care câștigă este selectat.
* Odată ce am selectat un număr de indivizi (se poate intampla ca un individ sa fie selectat de mai multe ori!) se aplica principiile de recombinare și mutație:
  + Recombinare, sau crossing over reprezinta generara unei noi perechi de indivizi pe baza unei perechi de indivizi selectați. Se combină prima jumătate a primului cromozom cu a doua jumătate a celui de-al doilea. Respectiv prima jumătate a celui de-al doilea cu a doua jumătate a primului

Ex: pt indivizii **11001001** și **10110001** va rezulta perechea **11000001** și **10111001** care se va regăsi în nouă generație de indivizi (noua populație)

* criteriul de point mutation: Odată obținut un individ nou, înainte de a se adăuga populației noi, se parcurg toate “genele” și cu o probabilitate foarte mica (<3%) se poate schimba o genă din ‘0’ în ‘1’ sau invers
* Ex: **10111101**

Avem apoi o noua populatie. Succesiunile de generatii se vor repeta ori pana cand atingem un număr maxim apriori stabilit (ex: 1000 de generații) sau cand cel mai bun element din populatie ramana le fel pt un anumit număr de generații (ex dacă elementul cel mai bun nu se schimbă timp de 10 generatii).

Aleg ca soluție pentru problema cel mai bun element din ultima generație de indivizi.