Algoritmi Genetici: 1/0 knapsack problem

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID Obj | A | B | C | D | E | F | G | H | Capacitate  Rucsac |
| Val | 10 | 12 | 5 | 14 | 9 | 7 | 15 | 10 |
| Weight | 7 | 9 | 2 | 10 | 10 | 6 | 12 | 12 | 30 |

Cum as putea reprezenta o solutie pentru problema?

Vector caracteristic X cu proprietatea ca X[i]=1⇔ obiectul *i* este selectat, si 0 altfel

Ex:

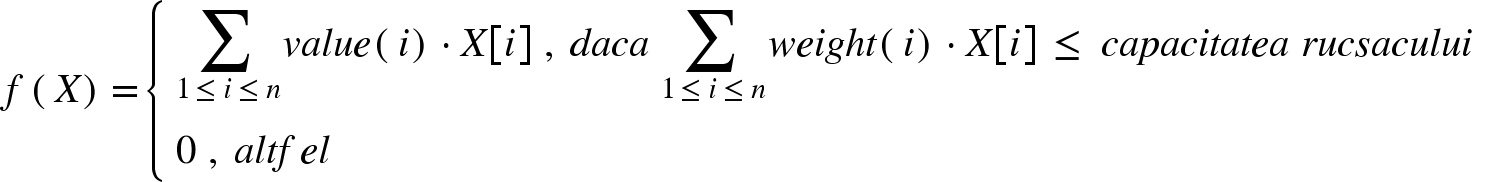
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

In cazul unui algoritm genetic o astfel de reprezentare poate fi numita cromozom/individ.

* reprezentarea are o lungime fixa
* acopera toate solutiile posibile
* nu apare problema selectarii unui obiect de mai multe ori.

Trebuie sa definim o functie de fitness relevanta pentru problema noastra

f:2n->R+



f(10100110)=val(A)+val(C)+val(F)+val(G)=37 (greutate 27<30)

f(10010110)=0 (greutatea >30)

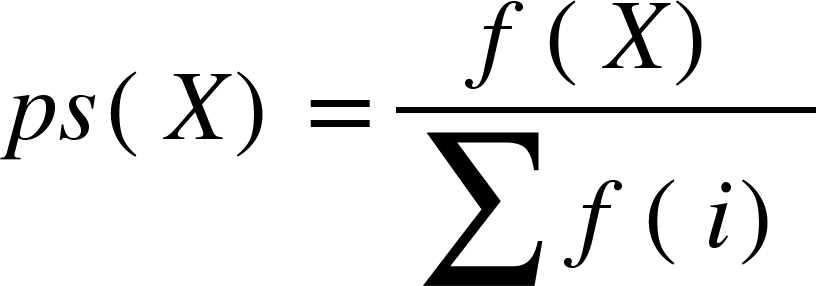
Generam aleator o populatie initiala.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Populatie: | | | | | | | | Fitness |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 31 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| **0** | **0** | **1** | **1** | **0** | **1** | **1** | **0** | **41** |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 19 |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 29 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 32 |

Trebuie sa construim o noua generatie pe baza celei curente:

* criteriul elitist: elementul (sau alementele, dupa caz) cel mai bun va trece direct (si nemodificat) in generatia urmatoare. Garantează ca individul cel mai bun de la o anumita generatie este mai bun de cat oricare element din oricare generatie precedenta.
* raman n-1=5 locuri disponibile.
* criteriul ruletei:

Calculăm probabilitatea de selecție (ps) pentru fiecare individ



trebuie sa calculăm intervalele de selecție

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 1 | 2 | 3 | 4 | 5 | 6 |
| ps(x) | 0.205 | 0 | 0.270 | 0.125 | 0.190 | 0.210 |
| interval de selecție | [0-0.205) | - | [0.205-0.475) | [0.475-0.6) | [0.6-0.7.9) | [0.79-1) |

trebuie sa selectam n-1 elemente, adica sa generam n-i numere aleatoare intre 0 si 1

0.590 0.897 0.733 0.725 0.244

Indivizii selectati sunt: 4, 6, 5, 5, 3

Populatia intermediara:  
**01000100**

**10000110**

**00110001**

**00110001**

**00110110**

Pe aceasta populatie intermediara aplicam operatorul genetic de crossing over (încrucișare)

Avem o probabilitate de crossing over (data ca parametru de intrare) ex: pc=0.25

0.759 0.209 0.010 0.305 0.193

s-au selectat indivizii

**10000110**

**00110001**

**00110110**

**Odata ce avem multimea de indivizi selectati pt incrucisare, eventual dam un shuffle si ii luam perechi de 2 cate 2 (eventual in caz de numar impar ii luam pe ultimii 3 la un loc)**

**Generam aleator un i - punct de rupere: ex: i=5**

**10000|110**

**00110|001**

**00110|110**

**Are loc incrucisarea:  
10000001**

**00110110**

**00110110**

**Aceste 3 elemente se vor alătura celor care nu au fost selectati pentru incrucisare  
10000001**

**00110110**

**00110110**

**01000100**

**00110001**

**Aplicăm operatorul de mutație.**

* **mutatie rara: fiecare cromozom are o probabilitate de selectie. Daca un cromozom este selectat, i se va schimba valoarea unei gene aleasa aleator cu complementul ei**
* **mutatie “normala”: iterez prin multimea de gene - fiecare are o probabilitate de a-si schimba valoarea**

**Dupa mutatie obtinem**

**11000001**

**00110110**

**00110110**

**01001100**

**00110001**

**Acesti cromozomi se vor alatura elementului elitist in noua generatie formata:  
00110110**

**11000001**

**00110110**

**00110110**

**01001100**

**00110001**

**Repetam procesul (calculam fitnessul, calculam intervalele de selectie, selectie elitista, metoda ruletei, aplicam operatorii genetici).**

**Pana cand repetăm întregul proces?**

**Pana cand se indeplineste unul dintre urmatoarele criterii”**

* **S-au exacutat deja un numar suficient de iteratii/a trecut destul de mult timp (ex: am ajuns la 1000 de generatii/ a rulat deja timp de 10 secunde)**
* **Daca solutia obtinuta la un anumit pas trece de un threshold prestabilit (ex: am gasit o solutie de profit 52, ma multumeam cu orice solutie cu profit >50)**
* **Cazul in care elementul elitist ramane acelasi timp de mai multe generatii (ex: timp de 10 generatii a ramas mereu acelasi element care sa fie cel mai bun).**

**Odata ce am oprit algoritmul, solutia noastra este elementul cel mai bun din ultima generatie.**