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 la aceasta problema?

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

Ex:

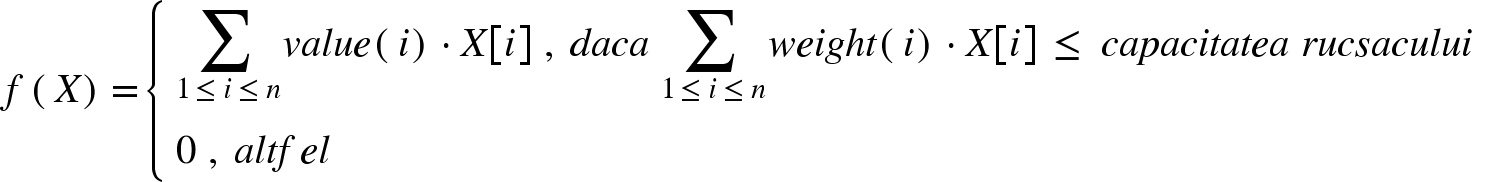
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 |

In cazul unui algoritm genetic codificarea unei posibile solutii se numeste 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 (greutate 35>30)

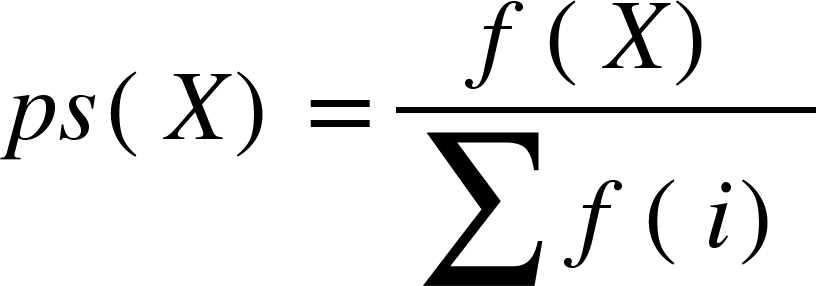
Se genereaza aleator un numar de indivizi repezentand generatia 0

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 generam o noua populatie pe baza celei curente:

* criteriul elitist: elementul (sau, dupa caz, elementele) cel mai bun va fi trecut automat in generatia urmatoare fara a fi supus vreunui operator genetic. Ne asigura ca la generatia curenta, elementul elitist este cel putin la fel de bun ca orice element dintr-o generatie anterioara.
* trebuie sa selectam indivizi pt locurile ramase disponibile in generatia urmatoare.
  + criteriul ruletei (alternativa ar putea fi turneu, etc)

Trebuie sa calculam probabilitatea de selectie pentru fiecare element.



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

trebuie sa generam un numar de 5 variabilie aleatoare intre 0 si 1:  
0.034 0.341 0.116 0.480 0.713

Au fost selectati indivizii: 1, 3, 1, 4, 5

Populatia intermediara:  
**11001000**

**00110110**

**11001000**

**01000100**

**00110001**

Pe aceasta populatie intermediara aplica operatorul genetic de crossing over (incrucisare). Fiecare cromozom are o probabilitate de incrucisare (pc) data ca parametru (ex: 0.35)

0.530 0.208 0.089 0.101 0.884

**00110110**

**11001000**

**01000100**

**Odata ce am multimea de indivizi selectati pentru incrucisare, se da shuffle la multime, si se iau perechi de indivizi peste care se face icrucisare. In cazul in care numarul de indivizi selectati este impar, ultimul set va fi un triplet in loc de pereche.**

**Cum arata incrucisarea?**

**trebuie sa generam un punct de rupere: ex 5**

**indivizii dupa incrucisare arata astfel:**

**00110000**

**11001100**

**01000110**

**Indivizii rezultati isi vor inlocui parintii in populatia intermediară**

**11001000**

**00110000**

**11001100**

**01000110**

**00110001**

**Aplicam operatorul de mutatie genetica:**

* **mutatie rara: Fiecare cromozom are o probabilitate de mutatie (ex 0.02). Odata un cromozom selectat (ca in cazul precedent) i se modifica valoarea unei singure gene.**
* **mutatie “regular”: iteram pe fiecare gena, si cu o probabilitate foarte mica (ex: 0.01) ii modificam valoarea.**

**Obtinem :**

**11001000**

**00110000**

**11011100**

**01000110**

**00010001**

**Aceasta populatie intermediara i se alatura elementului elitist in noua generatie:**

**00110110**

**11001000**

**00110000**

**11011100**

**01000110**

**00010001**

**Cat timp se repeta procesul?**

**cand se indeplineste unul dintre criteriile urmatoare:**

* **“a rulat destul” (ex: a rulat 1000 de generatii; a rulat 100 de secunde, “s-a incins procesorul”)**
* **valoaea maxima nu se mai schimba (ex: timp de 10 generatii am avut acelasi element elitist)**
* **cand fitnessul unui individ a depasit un anumt prag (ex: am gasit o configuratie de obiecte care incap in rucsac si au un profit total >45)**

**Odata oprit algoritmul, solutia furnizata va fi elementul elitist din ultima generatie.**