

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

Sub ce forma codificam o solutie pentru aceasta problema?

Sub form unui vector caracteristic X, cu proprietatea ca $X[i]=1 \Leftrightarrow$ obiectul i este selectat.

Ex:

1	0	1	0	0	1	1	0
---	---	---	---	---	---	---	---

O astfel de codificare se numeste cromozom/individ. Ar fi bine ca individul sa aiba proprietati precum: codificare unica, simpla, lungime fixa, in acest caz este usor de vazut ca fiecare obiect este selectat cel mult o singura data, etc.

Odata ce am stabilit codificarea, ce alt element trebuie stabilit inainte de a incepe algoritmul? Trebuie sa definim o functie de fitness cat mai buna pentru problema noastra

$f: 2^n \rightarrow \mathbb{R}_+$

$$f(X) = \begin{cases} \sum val(i) \cdot X[i] & \text{daca } \sum weight(i) \cdot X[i] \leq \text{capacitatea rucsacului} \\ 0, & \text{altfel} \end{cases}$$

$f(10100110)=37$ (greutatea totala este $27 < 30$)

$f(10010110)= 0$ (greutatea totala este >30)

trebuie sa generam populatia initiala:

populatia initiala va fi formata din indivizi generati aleator (distribuite uniforma)

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

aplicam criteriu elitist - elementul [00110110] va trece direct in generatia urmatoare.
 Elementul elitist de la o generatie este intotdeauna cel putin la fel de bun ca oricare element obtinut intr-o generatie anterioara.

Ramane sa selectam n-1=5 indivizi din populatia curenta:

aplicam criteriul ruletei:

Trebuie sa asociem o probabilitate de selectie (ps) pentru fiecare individ:

$$ps(x) = \frac{f(x)}{\sum_{1 \leq i \leq n} f(i)}$$

x	1	2	3	4	5	6
ps(x)	0.205	0	0.270	0.125	0.190	0.210
interval de selectie	[0-0.205)	-	[0.205-0.475)	[0.475-0.6)	[0.6-0.790)	[0.790-1.00)

Generez n-1 variabile aleatoare pe intervalul [0,1)

0.050 0.263 0.490 0.218 0.771

Elementele selectate sunt: 1, 3, 4, 3, 5

11001000

00110110

01000100

00110110

00110001

pentru fiecare element, exista sansa de 0.35 (parametru dat la input) sa participe la crossing over:

0.807 **0.253** **0.189** 0.657 **0.174**

indivizii selectati pentru crossing over sunt:

00110110

01000100

00110001

Pantru fiecare pereche de cromozomi (in cazul acesta avem triplet) generez aleator un punct de rupere ex: i=3

001|10110

010|00100

001|10001

Dupa recombinare avem:

00100100

01010001

00110110

Acesti indivizi isi vor inlocui parintii si se vor alatura celor care nu au fost selectati pentru incrucisare

00100100
01010001
00110110
11001000
00110110

Se aplica operatorul de mutatie:

- mutatie rara: se itereaza prin multimea de indivizi - fiecare individ are o mica probabilitate (0.02) de a fi selectat pentru mutatie. Odata selectat individul, acestuia i se schimba valoarea unei gene cu complementul ei de pe o pozitie aleatoare
- mutatie "regular": iterez prin toate genele fiecarui individ. Exista o foarte mica probabilitate (ex: <0.01) ca o gena sa isi schimbe valoarea.

Dupa mutatie, populatia intermediara devine

00100100
00010001
00110110
11101000
00110110

Aceasta populatie se va alatura individului elitist din popuatie initiala, formand noua generatie:

00110110
00100100
00010001
00110110
11101000
00110110

Cat timp rulez algoritmul?

Cand macar unul dintre urmatoarele 3 criterii este satisfacut:

- s-a realizat un numar de generatii (ex: opresc algoritmul dupa 1000 de generatii) sau dupa un numar de secunde.
- s-a gasit o solutie care depaseste un anumit prag de acceptare (ex: am gasit o solutie cu profit 41, iar eu ma multumesc cu orice solutie cu profit >40)
- nu se mai schimba elementul maxim (sau media elementelor) timp de un numar de generatii (ex: timp de 10 generatii mereu a fost aceeași solutie cea mai buna, opresc algoritmul, si ma multumesc cu ea)