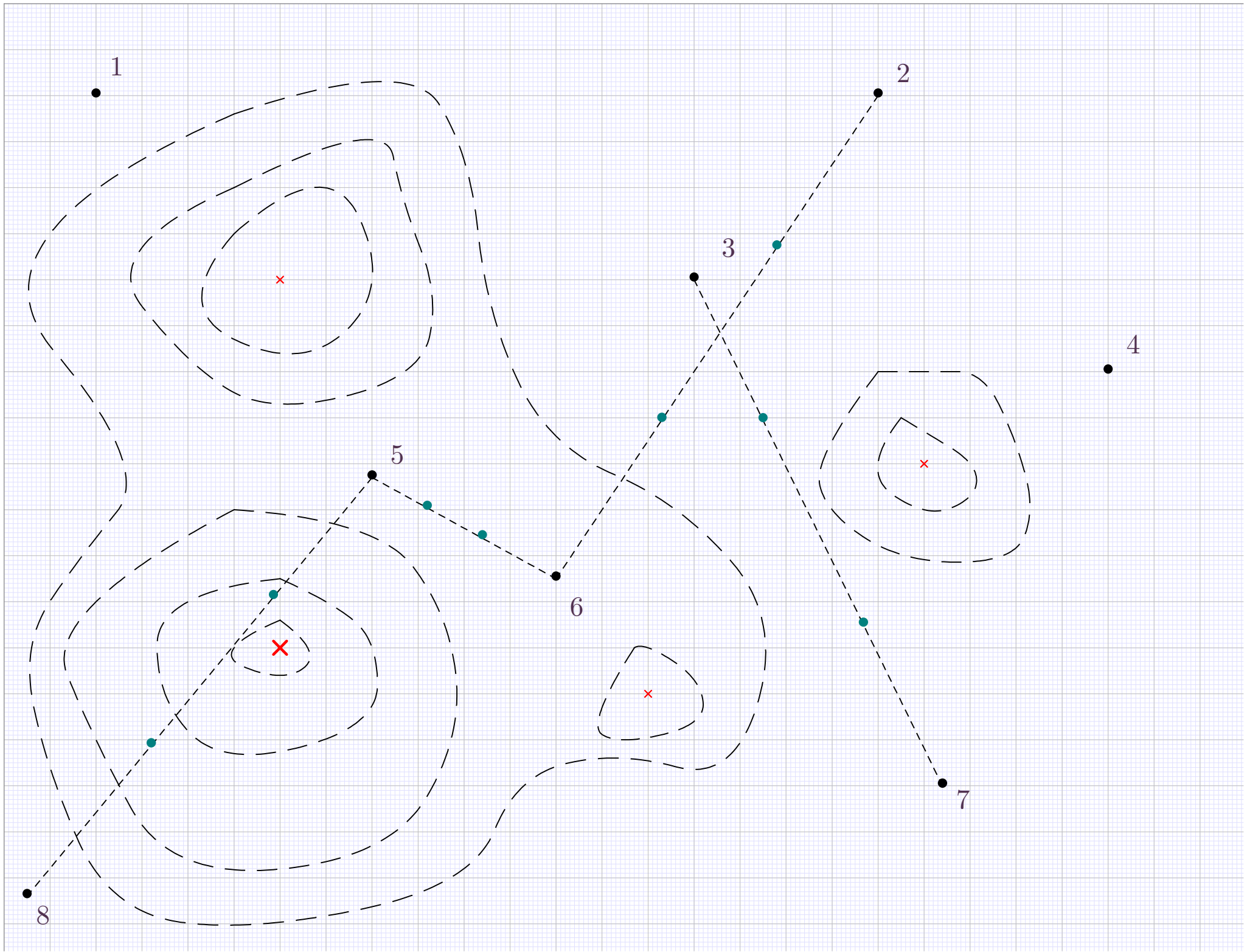
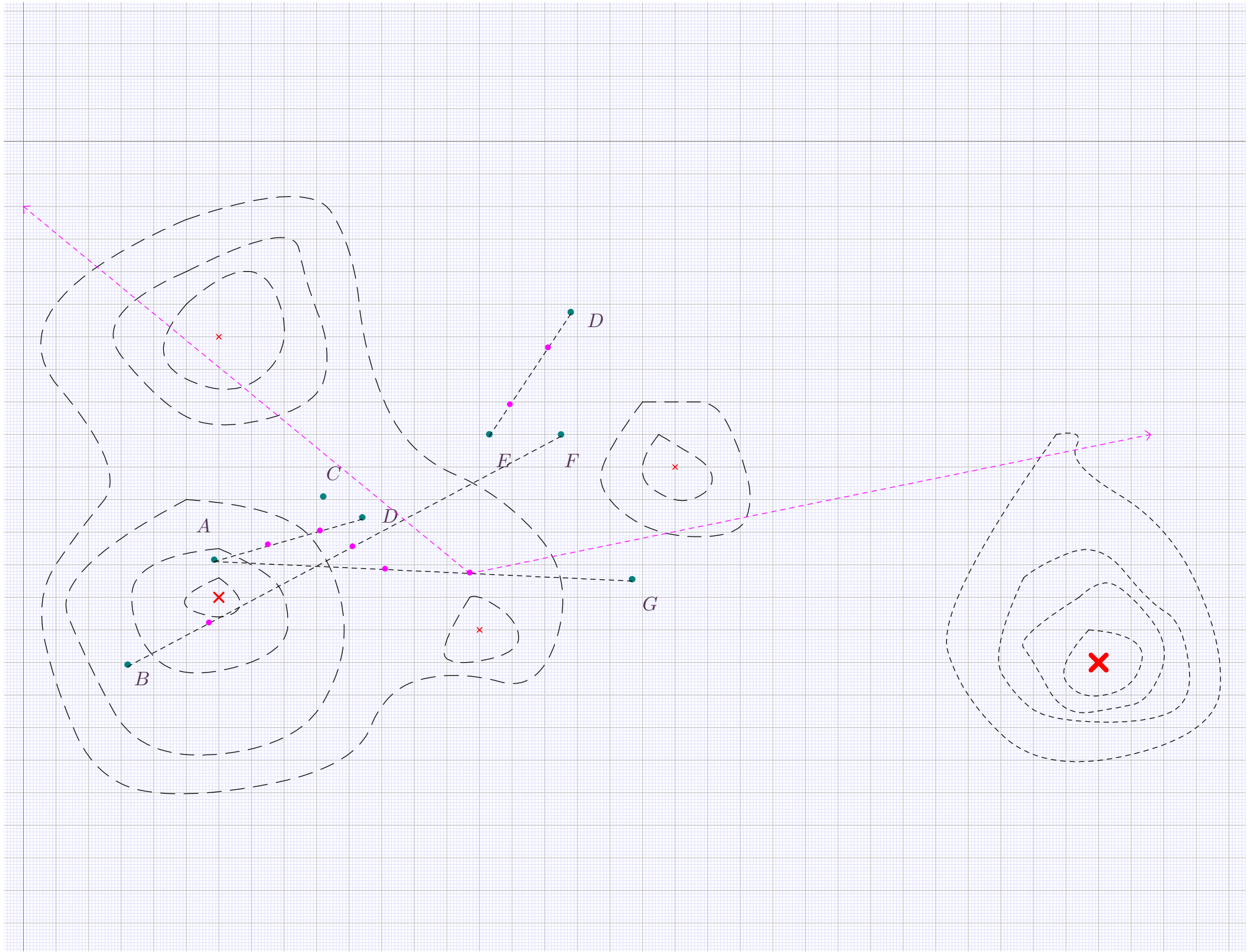


Principi optimizacije genetskim algoritom

MILAN R. RAPAIĆ

Katedra za automatsko upravljanje
Departman za računarstvo i automatiku
Fakultet tehničkih nauka
Univerzitet u Novom Sadu





$$f(x) \rightarrow \max$$

- f - kriterijum optimalnosti - kriterijum prilagođenosti, **prilagođenost**
 - Sama potencijalna rešenja nazivaju se **jedinkama**
 - Ako imamo dve jedinke, x_1 i x_2 , i ako je $f(x_1) > f(x_2)$ reći ćemo da je x_1 **bolja** jedinka od x_2 (**prilagođenija**)
 - Skup jedinki koje “žive” u isto vreme, odnosno koje posmatramo u okviru iste iteracije, nazivamo **populacijom**
 - Umesto termina iteracija, obično se u ovom kontekstu koristi termin **generacija**
1. Izaberemo početnu populaciju: odaberemo veličinu populacije (broj jedinki u populaciji, N) i na neki slučajan način “razbacamo” te jedinke po prostoru pretrage. Označimo populaciju sa $\mathcal{X} = \{x_1, x_2, \dots, x_N\}$
 2. Za svaku jedinku iz tekuće populacije, sračunamo njenju prilagođenost: $f(x)$
 3. Biraju se parovi roditelja iz populacije, i to tako što jedinke koje imaju veliku prilagođenost imaju i veliku verovatnoću da bud birane:
 - U odnosu na biološki ekvivalent, *pojam pola se ukida*. Dakle, svaka jedinka se može upariti sa bilo kojom drugom.
 - Biranje se vrši u $N/2$ iteracija. U svakoj iteraciji biramo par jedinki, i u svakoj iteraciji možemo izabrati bilo koju (jedna jedinka može biti birana kao roditelj puta).
 - Za razliku od biološke evolucije, svaki par roditelja uvek generiše tačno jedan par potomaka (2 roditelja \rightarrow 2 potomka). To znači da nakon $N/2$ iteracija selekcije i reprodukcije imamo N novih jedinki (isti broj kao u polaznoj roditeljskoj populaciji).
 - U ovom trenutku imamo dve populacije: roditeljsku i potomačku. Od ovih ukupno $2N$ jedinki, moramo izabrati N koje će nastaviti da “žive”. Preostalih N ćemo ukloniti iz razmatranja.
 - Vratimo se na korak 2. Ovo ponavljamo unapred definisati broj generacija (M).

Ruletska selekcija. Cilj je postići da je verovatnoća izbora svake pojedinačne jedinke proporcionalna njenoj prilagođenosti.

x	$f(x)$	rand	score
A	2	0.2	0.4
B	3	0.4	1.2
C	12	0.2	2.4
D	8	0.9	7.2
E	1	0.6	0.6
F	1	0.4	0.4

A A B B B C C C C C C C C C C D D D D D D D E F

Ključna stvar: Iako najveću verovatnoću izbora moraju da imaju najprilagođenije jedinke, sve jedinke imaju nenultu verovatnoću izbora. Čak i najmanje prilagođena jedinka mora imati šansu da bude birana kao roditelj.

x	$f(x)$	rank	rand	score
A	1	3	0.2	0.6
B	1200	6	0.4	2.4
C	240	5	0.2	1.0
D	5	4	0.9	3.6
E	0.1	2	0.6	1.2
F	0.01	1	0.4	0.4

Uvođenjem rangiranja, umanjuje se stepen favorizacije jako dobrih jedinki i sprečavaju negativni efekti pojave tzv. **superjedinki** (rešenja koja su mnogo bolja od drugih rešenja unutar iste populacije).

Kodiranje jedinki predstavlja mehanizam predstave jedinki u računarskom kodu.

Realno kodiranje podrazumeva da se jedinka prikaže nizom (vektorom) realnih brojeva. Ovo kodiranje je isto kao kodiranje rešenja koje koristimo kod drugih numeričkih optimizacionih metoda.

$$x = (1, 2.6, 8, 15.4, 3)$$

Binarno kodiranje podrazumeva da kompletnu jedinku zapišemo kao niz binarnih cifara

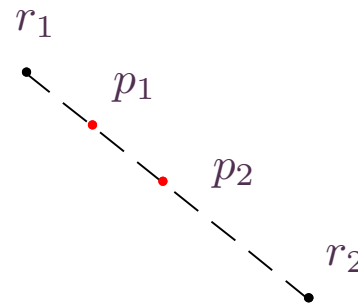
0 0 1 0 1 1 1 0 0 1 0 1

Razlika između ova dva tipa kodiranja leži u tome kako na njih deluju operatori ukrštanja i mutacije (koje ćemo definisati na sledećem slajdu). Element kodiranja (zapisa) koji ne možemo deliti (u gornjem slučaju to su brojevi, a u donjem to su biti) nazivaju se **hromozomima**.

Ukrštanje i mutacija su operacije pomoću kojih se, polazeći od roditelja, generišu potomci. Uvek se od dva roditelja dobijaju dva potomka. Populacija u genetskom algoritmu stalno ima isti broj jedinki.

r_1, r_2 – roditelji

p_1, p_2 – potomci



$$p_1 = a r_1 + (1 - a) r_2$$

$$p_2 = b r_1 + (1 - b) r_2$$

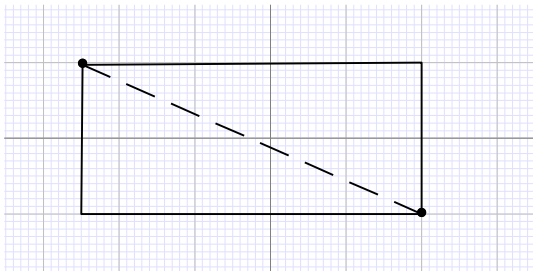
a, b – slučajni brojevi koji pripadaju intervalu $(0, 1)$

simetrično ukrštanje: izaberem samo jedan slučajan broj a

$$p_1 = a r_1 + (1 - a) r_2$$

$$p_2 = (1 - a) r_1 + a r_2$$

- Možemo uzeti jednu koordinatu od jednog, drugu od drugog, treću od prvog, ... i uopšte mogli bi smo za svaku koordinatu da generišemo slučajan broj r , pa ako je $r < 0,5$ onda prvom potomku dodelimo tu koordinatu od prvog roditelja, a drugom potomku od drugog roditelja, a u suprotnom dodeljujemo na obrnut način.



- Kako se pravi konveksna kombinacija dva vektora? $r_1 = (r_{x,1}, r_{y,1})$ $r_2 = (r_{x,2}, r_{y,2})$

$$p_{x,1} = ar_{x,1} + (1 - a)r_{x,2}$$

$$p_{y,1} = ar_{y,1} + (1 - a)r_{y,2}$$

Ako hoću da dozvolim da mi se potomci mogu naći bilo gde u okviru opisanog pravougaonika, onda treba da biram različite sličajne brojeve kada vršim ukrštanje pojedinih koordinata rešenja!

Ukrštanje binarno kodiranih jedinki.

0 0 1 0 1 1 1 0 0 1 0 1 - prvi roditelj

0 1 0 1 0 0 1 0 1 0 0 1 - drugi roditelj

0 0 1 0 0 0 1 0 1 0 0 1 - prvi potomak

0 1 0 1 1 1 1 0 0 1 0 1 - drugi potomak

Ukrštanje u jednoj tački. Postoje i drugi načini, recimo ukrštanje u dve tačke ili u više tačaka. Najveći stepen slučajnosti se dobija tzv. nasumično ukrštanje kada se ukrštanje vrši na svakoj bit poziciji.

0 0 1 0 1 1 1 0 0 1 0 1 - prvi roditelj

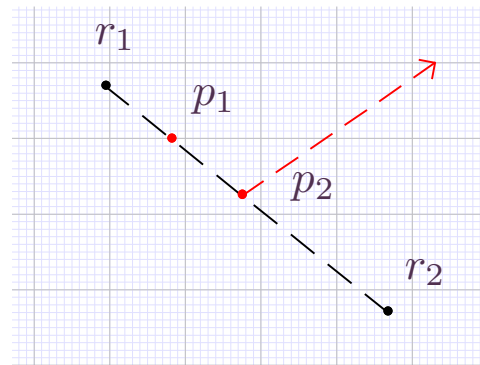
0 1 0 1 0 0 1 0 1 0 0 1 - drugi roditelj

0 1 0 - prvi potomak (početni deo, itd...)

0 0 1 - drugi potomak (početni deo, itd...)

Mutacija je operacija koja se obavlja nad jednom jedinkom. Najčešće se sa nekom malom verovatnoćom svaki od potomaka mutira: promeni na slučajan način, koji nije vezan za to kako su mu izgledali roditelji.

Mutacija kod realno kodiranih jedinki. Najjednostavniji način jeste da se nakon ukrštanja, sa *nekom malom verovatnoćom*, potomcima prosto doda neki slučajan vektor.



Mutacija kod binarno kodiranih jedinki. Najjednostavniji način jeste da se, sa *nekom malom verovatnoćom*, potomcima invertuju neke bitne pozicije.

0 1 0 1 0 0 1 0 1 0 0 1 - pre mutacije

0 1 0 0 0 0 1 1 1 0 0 1 - posle mutacije

Smisao mutacije jeste da unese u populaciji osobine koje nije imala roditeljska populacija! To znači da mutacija poboljšava eksplorativna svojstva algoritma. Mutacija unosi novu informaciju u samu populaciju.

Imamo na raspolaganju više opcija:

- Pobjedimo sve roditelje i pustimo decu da žive.
- Rangiramo svih $2N$ jedinki po prilagođenosti i pustimo N najboljih da prežive, a preostalih N ubijemo.

Ukoliko uklonimo sve roditelje, desiće nam se (relativno često) da ćemo time ukloniti i najbolju jedinku, odnosno da ni jedan potomak neće biti bolji od najboljeg roditelja. Praktično posmatrano, to će značiti da ćemo jako usporiti postupak dolaska do rešenja.

Ukoliko zadržavamo samo najbolje jedinke, jako brzo će nam kompletna populacija konvergirati, tako da će sve jedinke jako ličiti jedna na drugu. Međutim, eksplorativna svojstva algoritma će biti narušena, zato što će nakon malog broja iteracija sve jedinke početi jako da liče jedna na drugu.

Elitizam. Podrazumeva da se određeni mali broj jedinki pusti da prežive i pređu u narednu generaciju, bez obzira da li ove jedinke potiču iz roditeljske ili potomačke populacije. Sve ostale jedinke koje preživljavaju moraju poticati iz potomačke populacije. Svi drugi roditelji se uklanjaju.