Populacijske metaheuristike

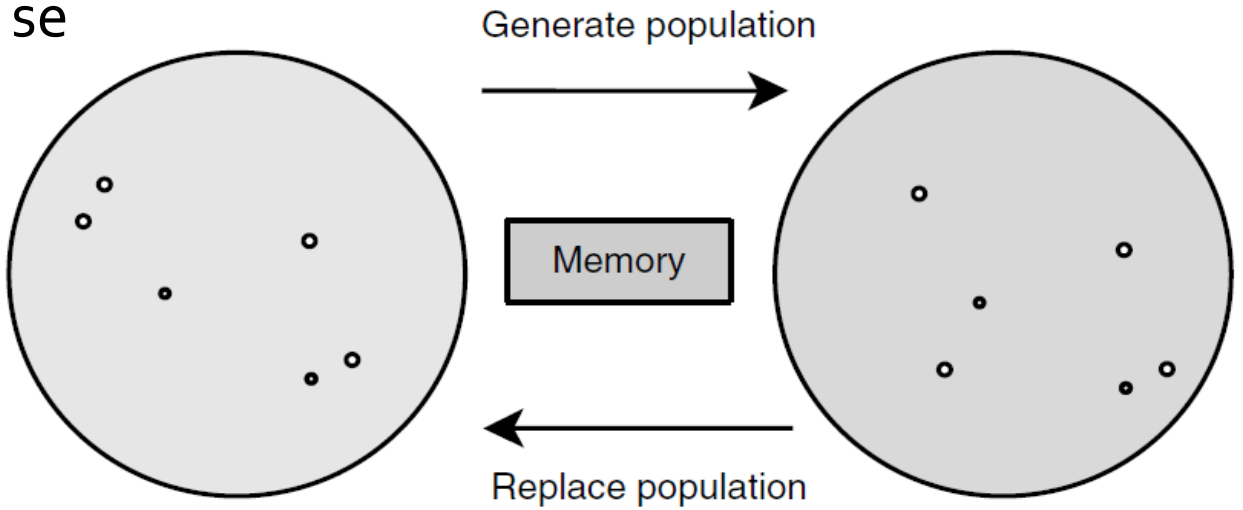
P-metaheuristike

Uvod

- Rad sa kolekcijom kandidata za rešenja.
- Kandidati međusobno utiču jedni na druge.
- Lošiji kandidati za rešenja se odbacuju u korist boljih ili se poboljšavaju po ugledu na bolja rešenja.
- Inspiracija u prirodnim procesima.
- Algoritmi:
  - Evolucioni algoritmi (Evolutionary algorithms EAs)
  - Optimizacija kolonijom mrava (Ant colony optimization – ACO)
  - Optimizacija rojem čestica (Particle swarm optimization – PSO)
  - Optimizacija kolonijom pčela (Bee colony optimization – BCO)
  - ...

Opšti način rada populacijskih metaheuristika

- Iterativno poboljšavanje populacije rešenja.
  - Korak 1. Generisanje početne populacije
  - Korak 2. Generisanje nove populacije.
  - Korak 3. Zamena trenutne populacije rešenjima iz nove i trenutne populacije bazirano na selekciji.
  - Koraci 2 i 3 se smenjuju sve dok ne bude zadovoljen uslov za zaustavljanje algoritma.
- Faza generisanja i faza zamene mogu teći bez pohranjivanja podataka u memoriju, a može se čuvati i istorija pretrage.



Opšti princip rada algoritama P-metaheuristika

$P = P_0$; /* Generation of the initial population */
$t = 0$;
Repeat
  Generate(P'_t); /* Generation a new population */
  P_{t+1} = Select-Population($P_t \cup P'_t$); /* Select new population */
  $t = t + 1$;
Until Stopping criteria satisfied
Output: Best solution(s) found.

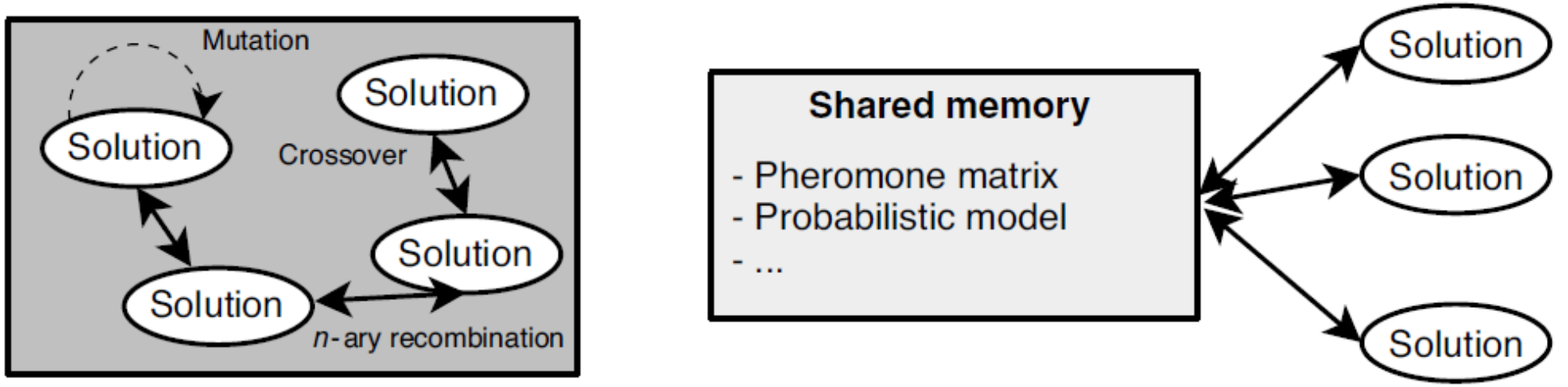
P-metaheuristike se razlikuju po

- memorisanju pretrage
- načinu na koji generišu i selektuju rešenja za novu populaciju

Memorisanje pretrage

P-metaheuristic	Search Memory
Evolutionary algorithms	Population of individuals
Scatter search	Population of solutions, reference set
Ant colonies	Pheromone matrix
Estimation of distribution algorithms	Probabilistic learning model
Particle swarm optimization	Population of particles, best global and local solutions
Bee colonies	Population of bees
Artificial immune systems: clonal selection	Population of antibodies

Generisanje populacije i selekcija



(a) Evolutionary-based P-metaheuristics: evolutionary algorithms, scatter search, ...

(b) Blackboard-based P-metaheuristics: ant colonies, estimation distribution algorithms, ..

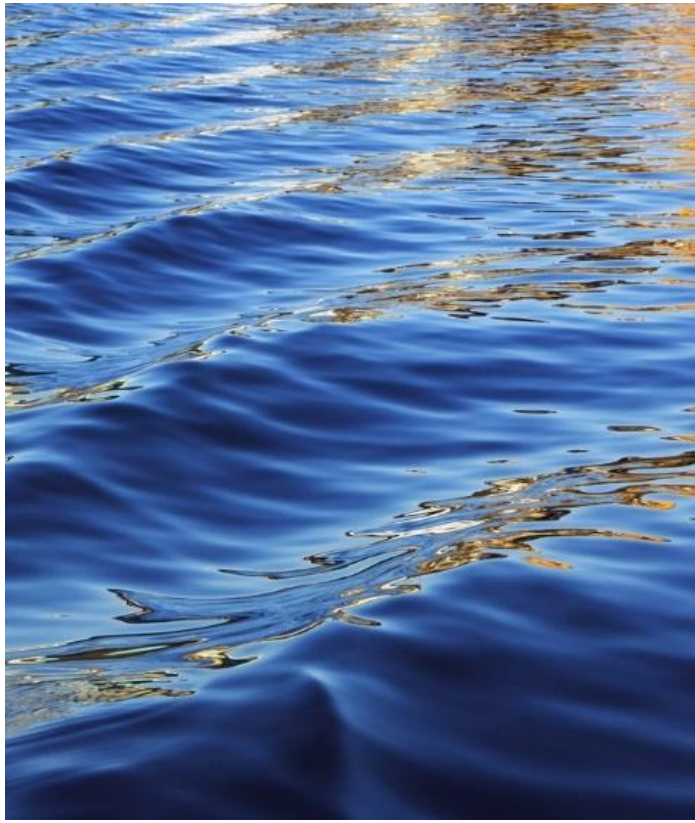
Početna populacija

- Odabir početne populacije je često zanemareno pitanje.
- Ključna za efektivnost i efikasnost algoritma.
- Raznolikost sprečava preuranjenu konvergenciju.
- Slučajni odabir
- Odabir rešenja metodom koja obezbeđuje raznolikost.
- Odabir upotrebom heuristike.

Kriterijum za zaustavljanje algoritma

- Statička procedura
  - Predefinisani broj evaluacija rešenja, broj generacija,...
- Adaptivna procedura
  - Predefinisani broj generacija bez poboljšanja, dostizanje optimalnog ili zadovoljavajućeg rešenja

Evolucioni algoritmi



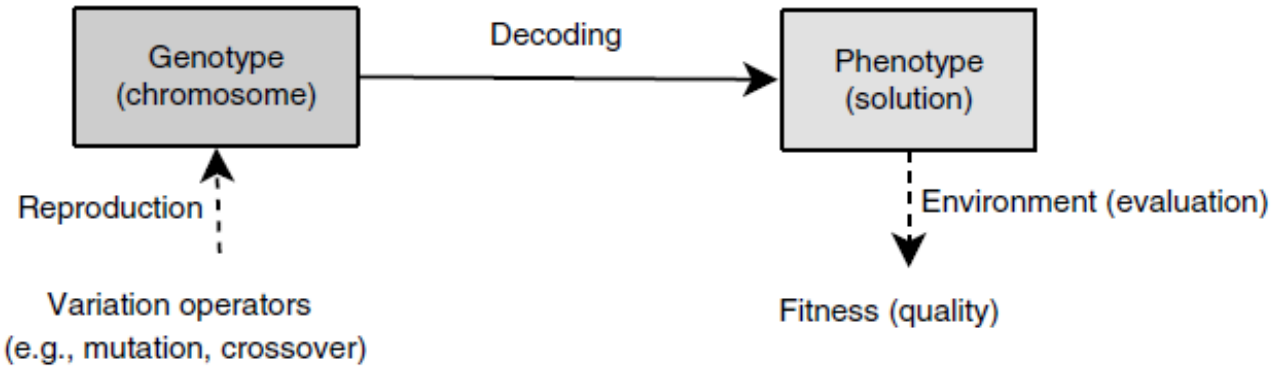
Uvod

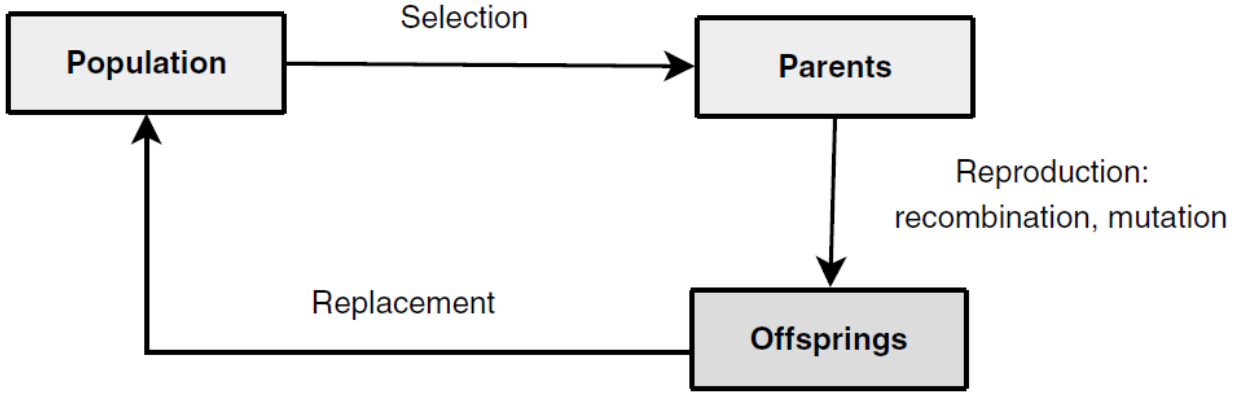
- Stohastičke optimizacione metode.
- Simuliraju prirodan proces evolucije – opstanak najsposobnijih.
- Najizučavanije P-metaheuristike.
- Razvoj počinje polovinom XX veka, paralelno u nekoliko naučnih centara:

  - Evolucione strategije - Rechenberg i Schwefel
  - Evoluciono programiranje – Fogel
  - Genetski algoritmi – Holland
  - Genetsko programiranje – Cramer i Koza

Opšti postupak rada i terminologija EA

Evolucioni algoritam	Optimizacija
Populacija	Skup kandidata za rešenje
Jedinka, individua	Kandidat za rešenje
Fitness, prilagođenost, podobnost jedinke	Vrednost funkcije cilja za rešenje kodirano jedinkom





Opšti princip rada EA

Generate($P(0)$) ; /* Initial population */
$t = 0$;
While not Termination_Criterion($P(t)$) **Do**
    Evaluate($P(t)$) ;
    $P'(t)$ = Selection($P(t)$) ;
    $P'(t)$ = Reproduction($P'(t)$); Evaluate($P'(t)$) ;
    $P(t+1)$ = Replace($P(t)$, $P'(t)$) ;
    $t = t + 1$;
End While
Output Best individual or best population found.

Evolucione strategije

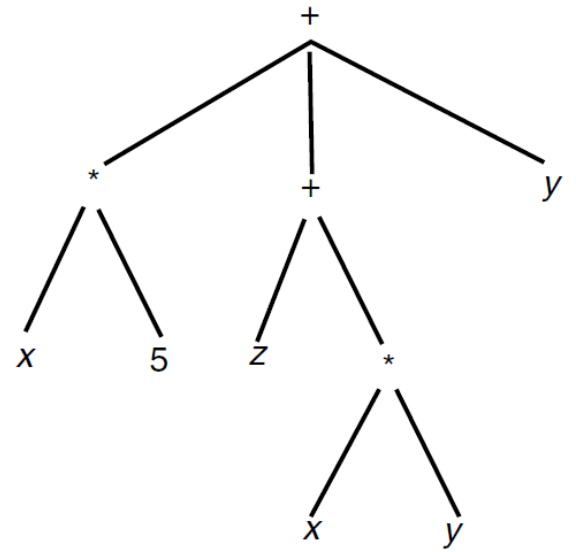
- Početak: eksperimentalno određivanje optimalnih parametara sistema.
- Mutacija i selekcija vektora parametara.
- Jedan roditelj i jedan potomak: $(1+1)-\mathrm{ES}$
- Opšti oblik $\left(\mu / \rho \overset{+}{,} \lambda\right)-\mathrm{ES}$
  - μ broj roditelja
  - λ broj potomaka
  - ρ broj roditelja koji učestvuju u stvaranju jednog potomka

---

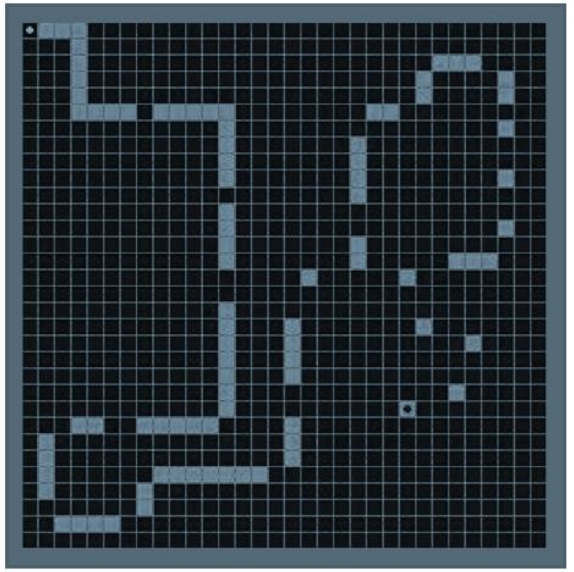
Initialize a population of μ individuals ;
Evaluate the μ individuals ;
Repeat
  Generate λ offsprings from μ parents ;
  Evaluate the λ offsprings ;
  Replace the population with μ individuals from parents and offsprings ;
Until Stopping criteria
Output Best individual or population found.

Genetsko programiranje

- Evoluiranje populacije računarskih programa.
- Promenljive i konstante programa – terminalni simboli (listovi stabla)
- Operacije – čvorovi.
- Evaluacija fitnesa – rezultat izvršavanja programa.
- Ukrštanje – ukrštanje podstabala
- Mutacija – zamena podstabla slučajno generisanim podstablom.



$(x*5)+(z+(x*y))+y$



The IF statements are sensing functions that detect if there is a food around. Leaves represent move operators in several directions (left, right, forward).

Selekcija

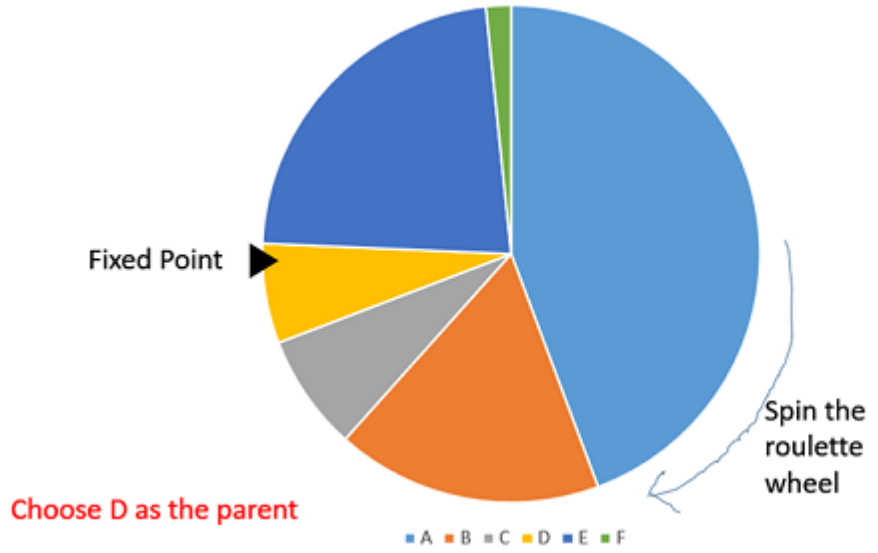
- Selekcija - proces odabira roditelja koji će ući u proces ukrštanja i dati potomstvo.
- Mehanizam selekcije favorizuje naprosečno prilagođene jedinke tako da one dobijaju veću šansu da stvore potomstvo.
- Dva osnovna načina dodele fitnesa jedinkama:
  - Proporcionalna dodela fitnesa (Proportional fitness assignment)
  - Dodela fitnesa bazirana na rangu (Rank-based fitness assignment)

Roulette Wheel Selection

- Proporcionalnost između stepena prilagođenosti jedinke i verovatnoće da ona bude odabrana za ukrštanje.
- Jedinke sa većim stepenom prilagođenosti imaju veće šanse da ostvare potomstvo.
- Algoritam:

1. Calculate S = the sum of a finesses.
2. Generate a random number between 0 and S.
3. Starting from the top of the population, keep adding the finesses to the partial sum P, till P>S.
4. The individual for which P exceeds S is the chosen individual.

Let f_i be the fitness of the individual p_i in the population P. Its probability to be selected is $p_i = f_i \Big/ \left(\sum_{j=1}^{n} f_j\right)$.

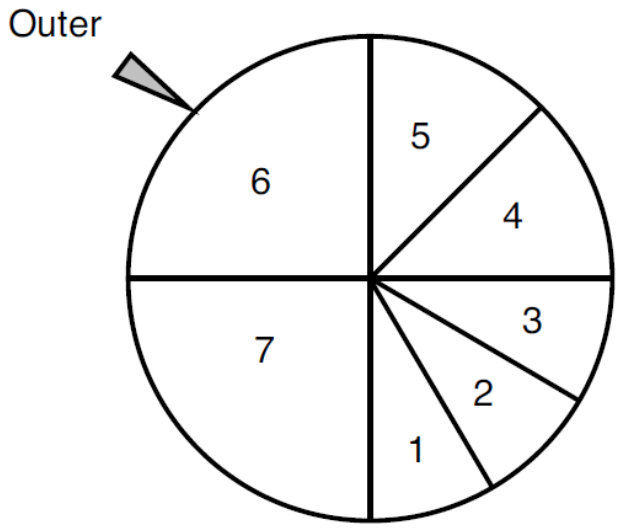


Chromosome	Fitness Value
A	8.2
B	3.2
C	1.4
D	1.2
E	4.2
F	0.3

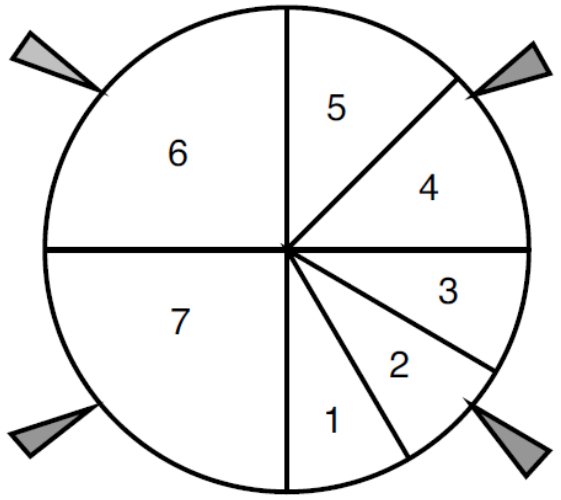
Poboljšanje turnirske selekcije korišćenjem SUS

- Stohastičko univerzalno uzrokovanje (Stochastic Universal Sampling)

Individuals:	1	2	3	4	5	6	7
Fitness:	1	1	1	1.5	1.5	3	3



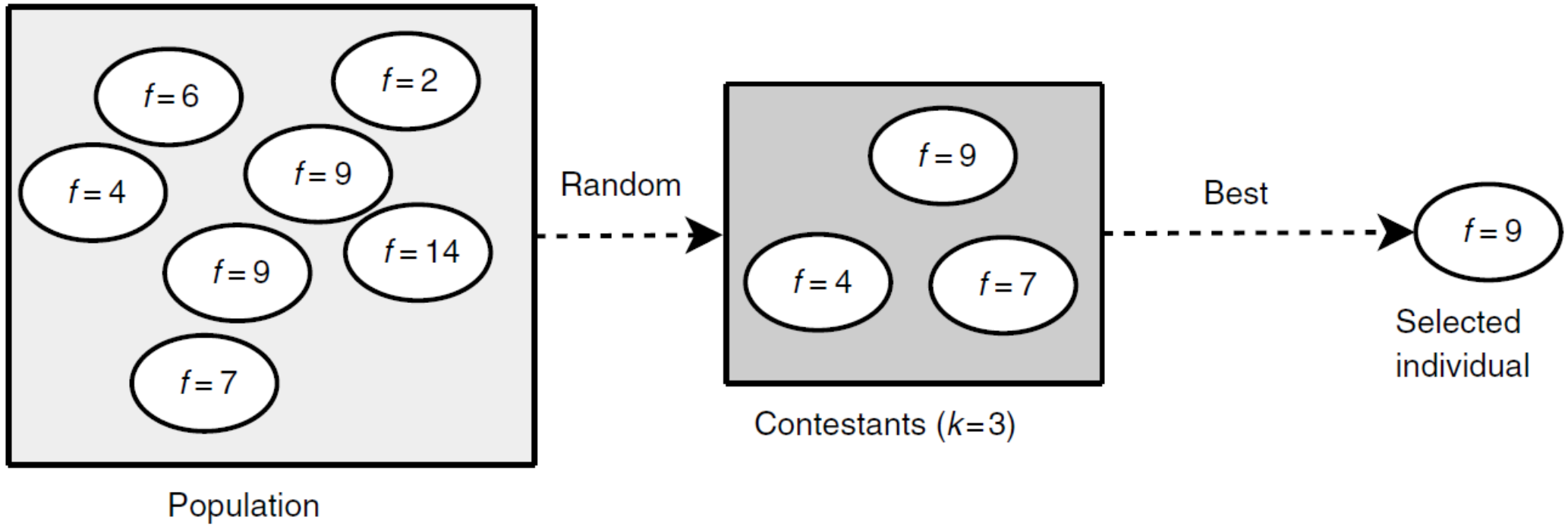
Roulette selection



Stochastic universal sampling

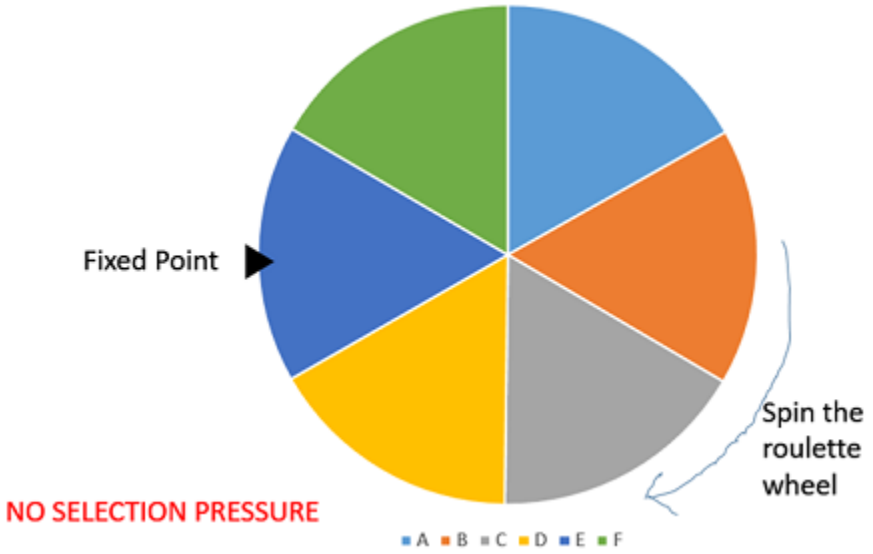
Turnirska selekcija

- K rešenja iz populacije "učestvuju u turniru".
- Pobednik je najbolje prilagođeno rešenje.
- Lošija rešenja ostaju bez mogućnoosti za reprodukciju.



Rank Selection

- Funkcioniše i sa negativnim vrijednostima fitnessa.
- Koristi se kada jedinke u populaciji imaju vrlo bliske vrednosti fitnesa, usled čega svaka od njih ima gotovo jednak udeo točka, pa samim tim i svaka individua ima približno istu vjerovatnoću da bude roditelj.
- U ovom slučaju ne koristi se koncept vrednosti fitnesa pri izboru roditelja.
- Svaki pojedinac u populaciji se rangira prema svom fitnesu.
- Izbor roditelja dalje zavisi od ranga svakog pojedinca a ne od fitnesa.
- Više rangirane individue su poželjnije od niže rangiranih.



Chromosome	Fitness Value
A	8.1
B	8.0
C	8.05
D	7.95
E	8.02
F	7.99

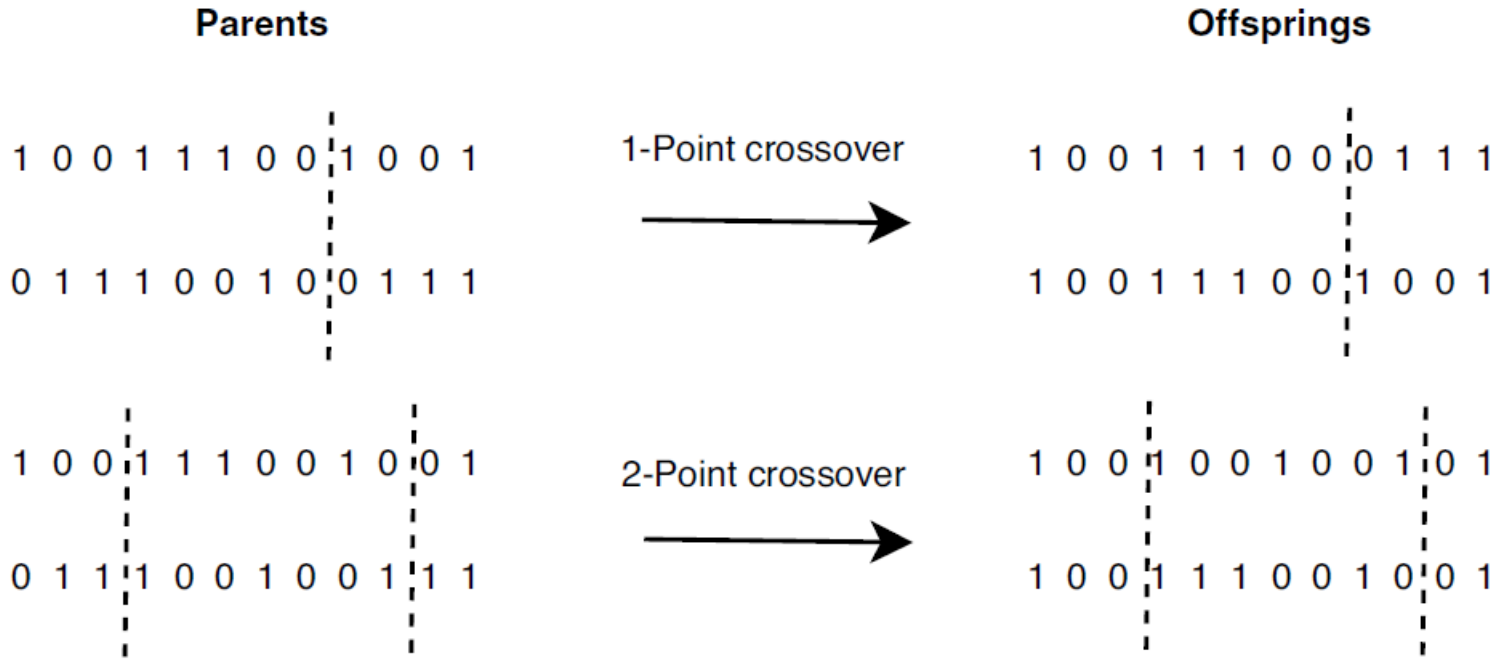
Chromosome	Fitness Value	Rank
A	8.1	1
B	8.0	4
C	8.05	2
D	7.95	6
E	8.02	3
F	7.99	5

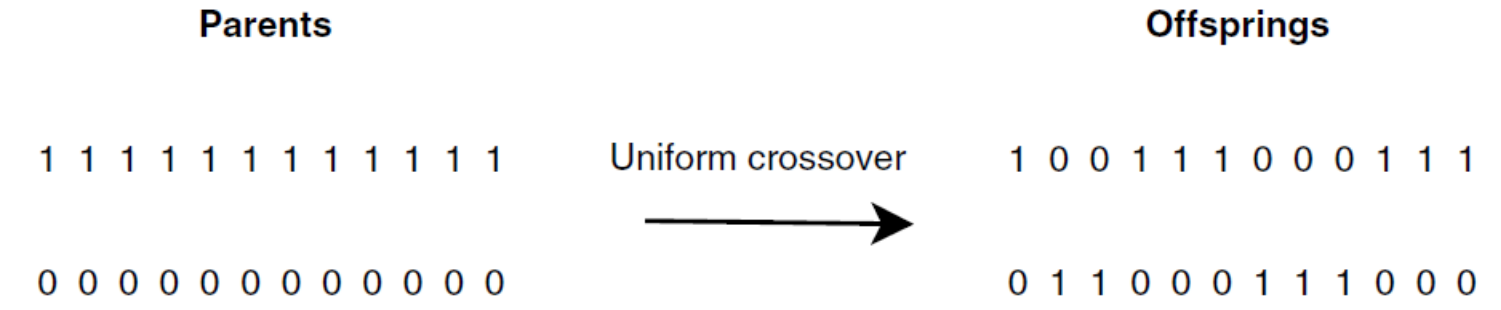
Mutacija

- Unarni operatori
- Unose male promene.
- Verovatnoća mutacije.
- Različite vrste mutacije za različite tipove rešenja.

Ukrštanje

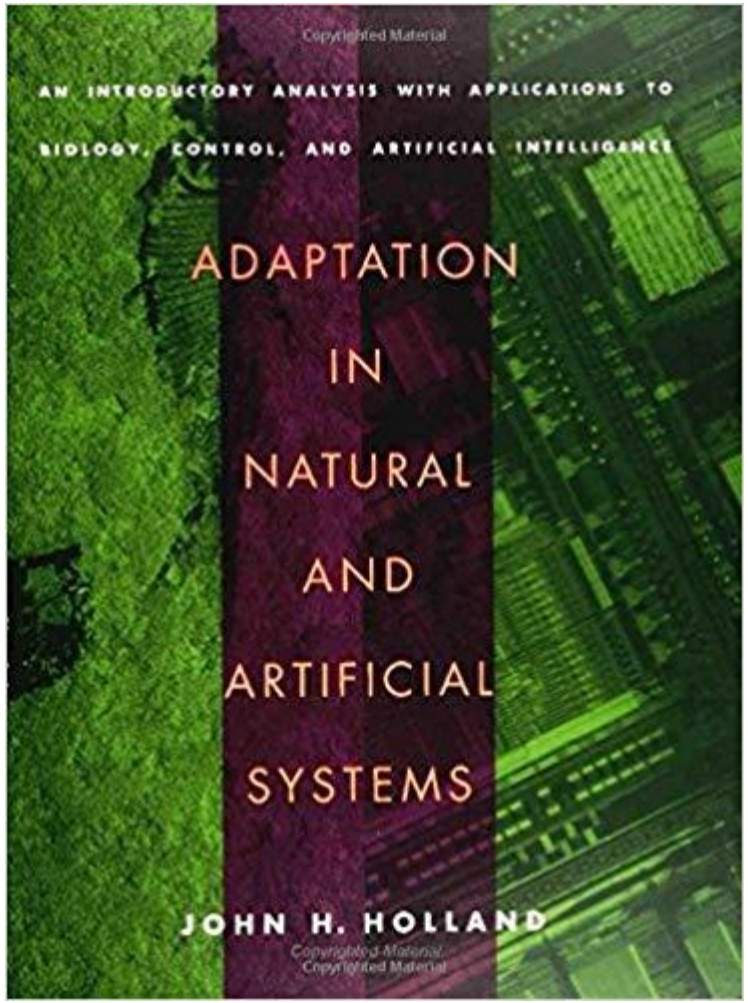
- Binarni ili n-arni operator.
- Prenošenje roditeljskih osobina na potomstvo.
- Dizajn operatora zavisi od tipa kdiranja rešenja.
- Verovatnoća ukrštanja.





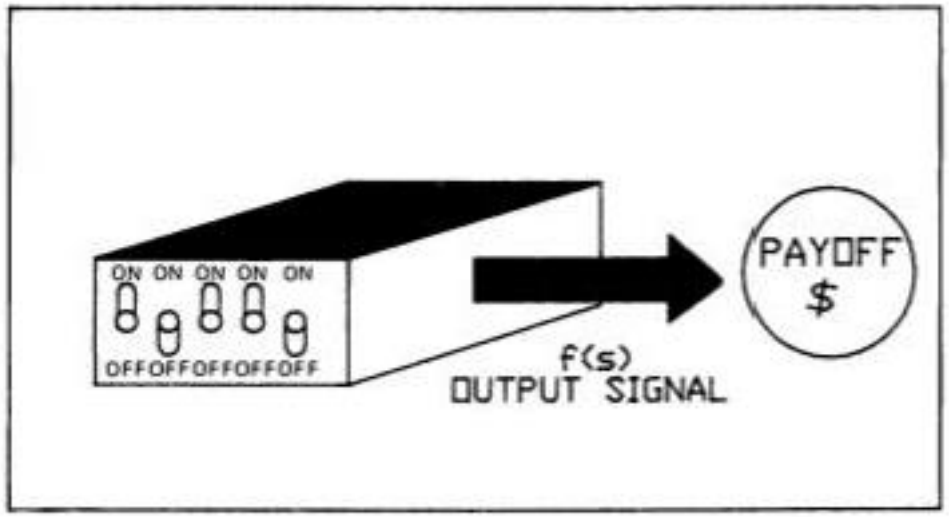
Genetski algoritmi (GA)

- Vrsta evolucionih algoritama.
- Tvorac: John Holland i njegova istraživačka grupa na Univerzitetu Mičigen, SAD



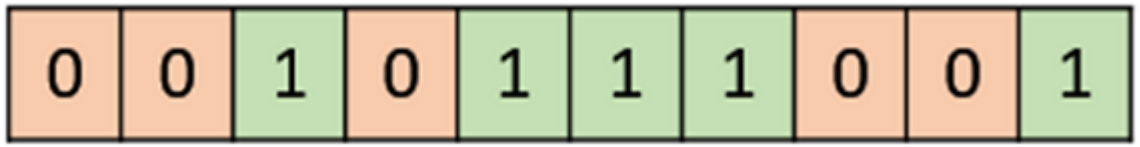
Kodiranje varijabli odluke

- Preslikavanje prostora pretrage u hromozome se naziva kodiranje.
- GA zahteva kodiranje prirodnih vrednosti varijabli u stringove konačnih dužina nad nekim konačnim alfabetom.
- Ako je alfabet sačinjen samo od nula i jedinica, onda imamo binarno kodiranje.
- Dekodiranje – tumačenje rešenja u kontekstu problema koji se rešava



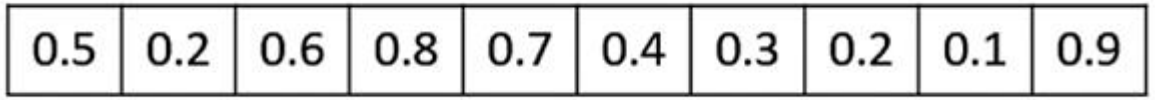
Vrste kodiranja

- binarno (problem ranca)

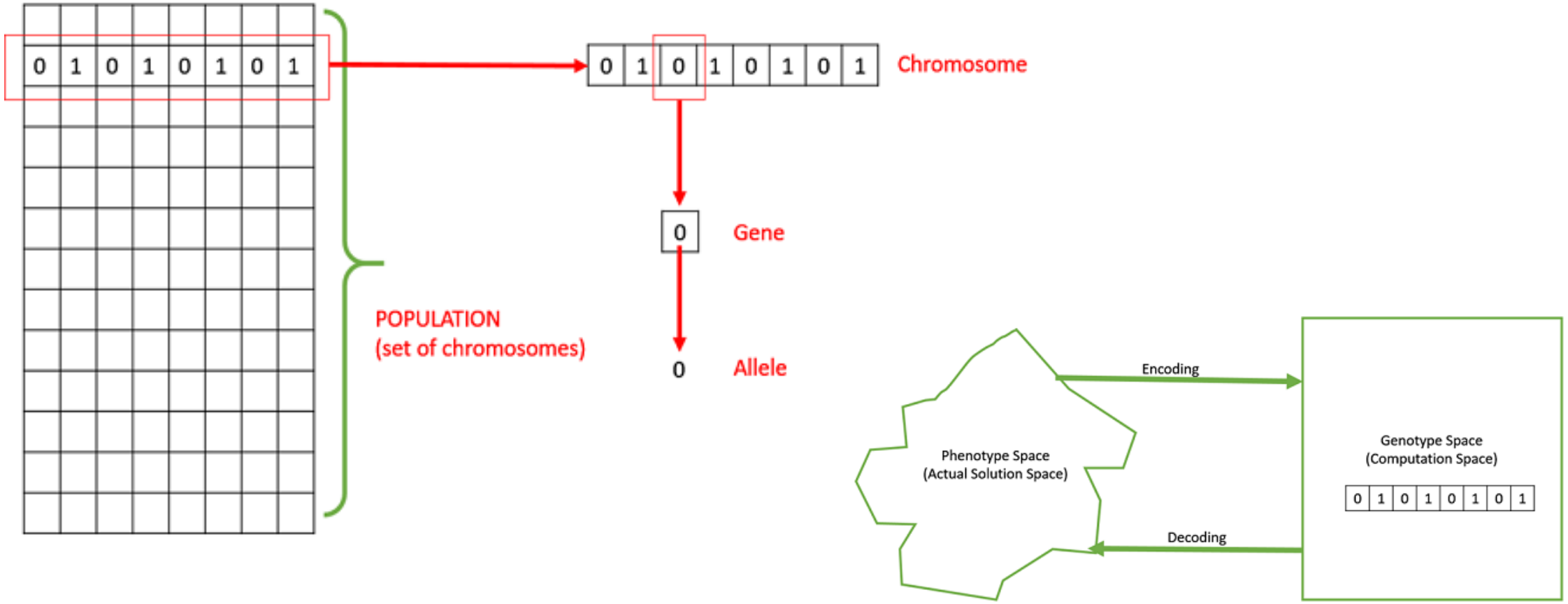


- Alfabet čini konačno mnogo celih brojeva - permutacijsko (problem trgovačkog putnika)

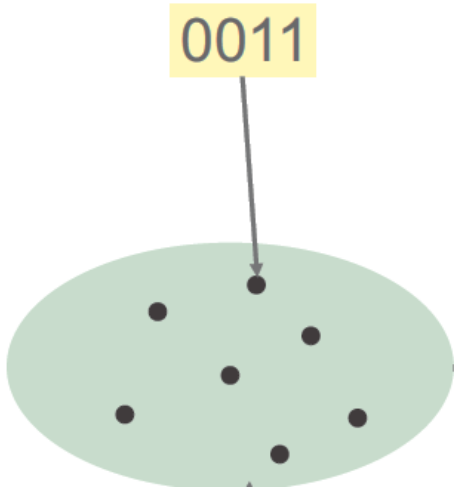
- Alfabet čine realni brojevi - realno



Primer – binarno kodiranje

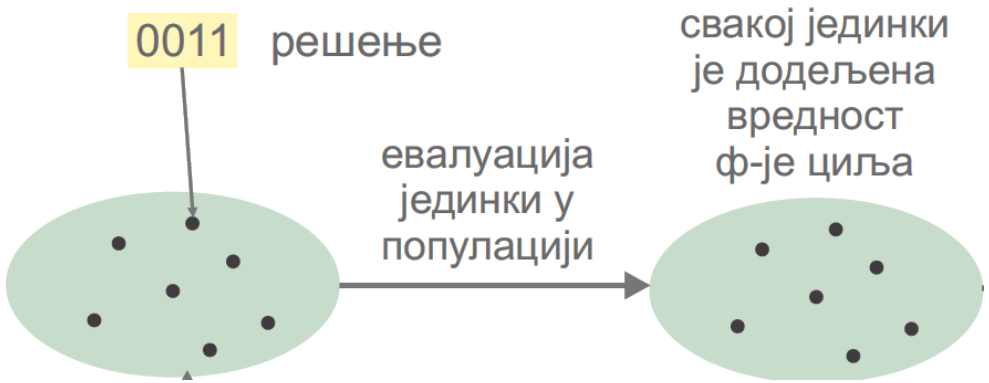


Kako radi GA?



1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.

Kako radi GA?



1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.

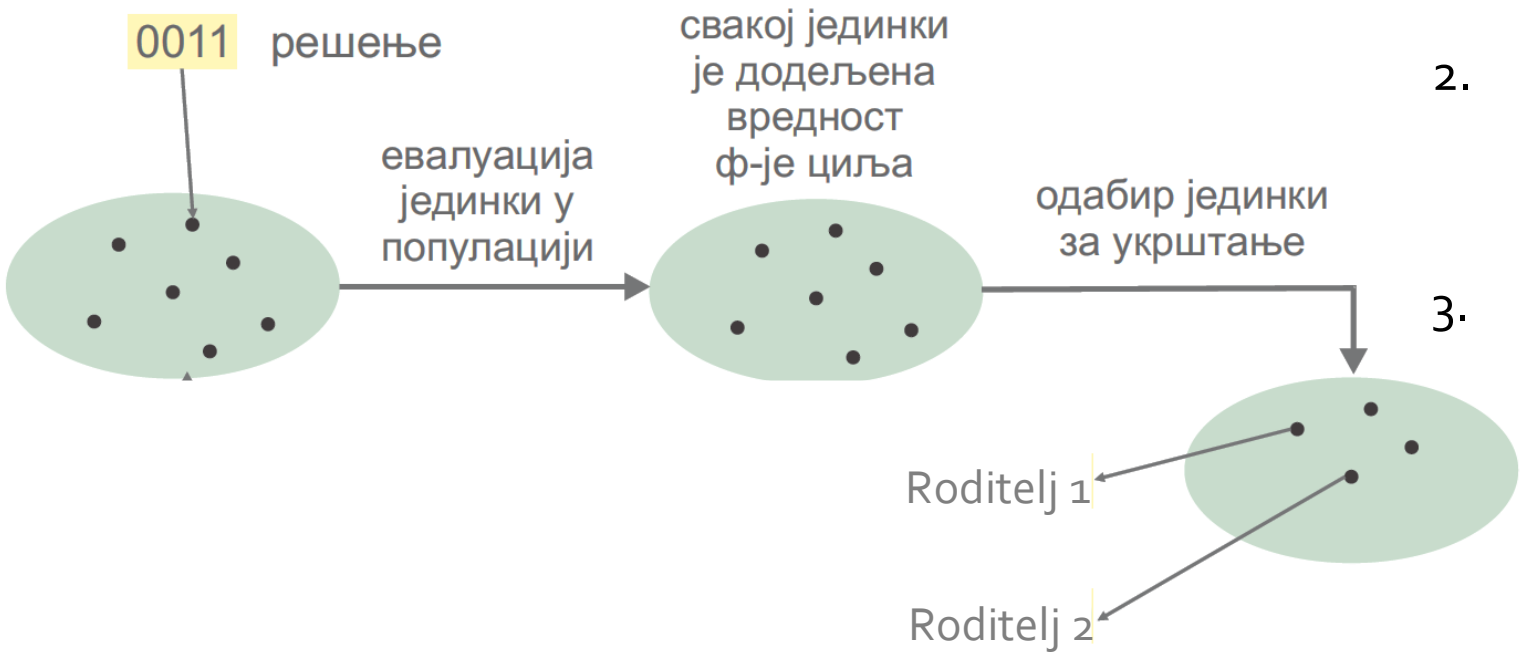
2. Svaka jedinka u populaciji se potom evaluira kako bi se odredio njen *fitness*.

Fitness

- Stepen prilagođenosti
- Bliskost rešenju problema
- Mera korisnosti
- Mera profita koji ostvarujemo odabirom neke jedinke
- Dobrota jedinke

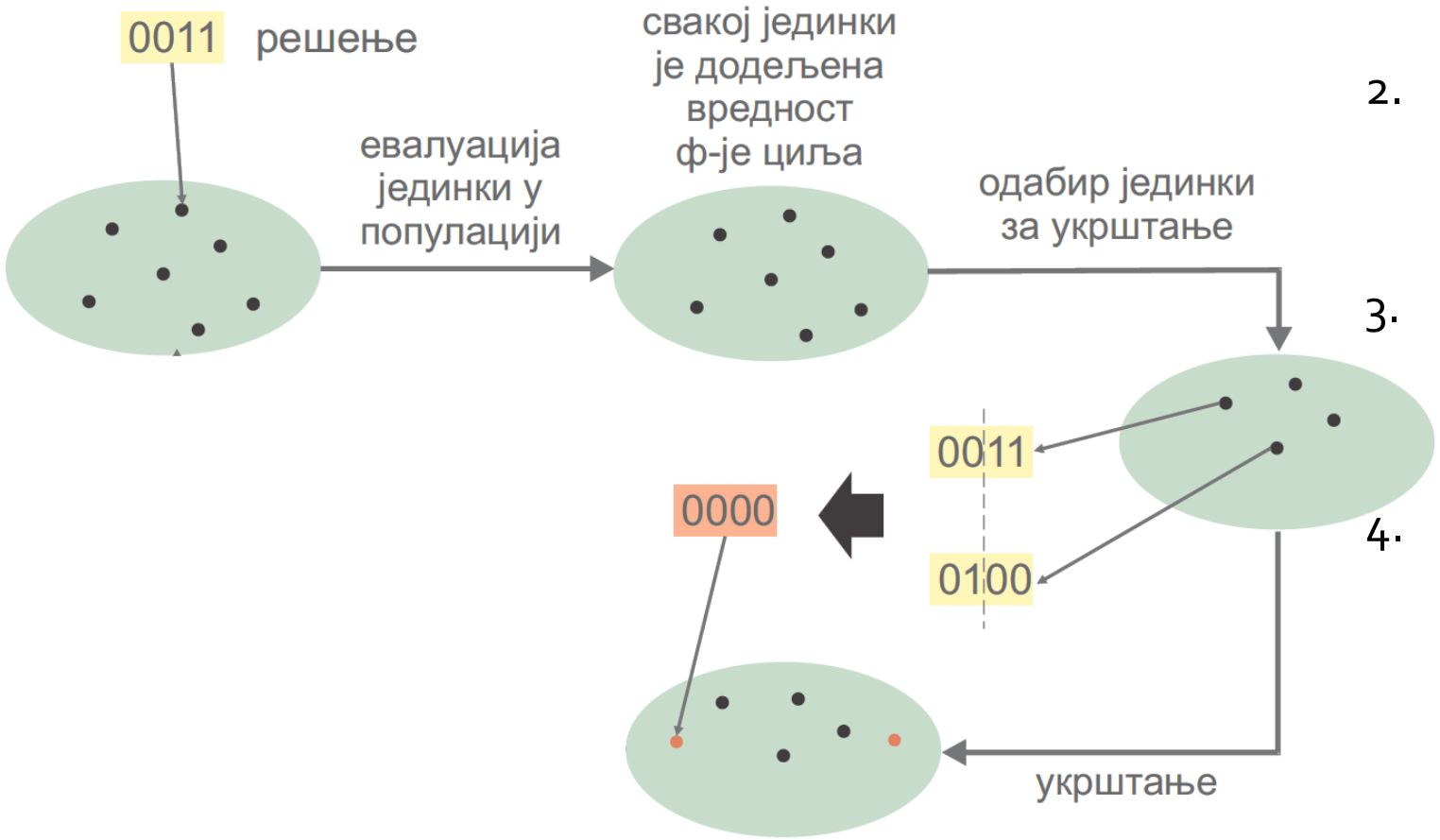
- Evaluacija (ocenjivanje) jedinki predstavlja izračunavanje vrednosti ciljne funkcije optimizacionog problema za datu jedinku.

Kako radi GA?



1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.
2. Svaka jedinka u populaciji se potom evaluira kako bi se odredio njen *fitness*.
3. Vrši se odabir roditelja čijim ukrštanjem će biti stvoreno potomstvo.

Kako radi GA?

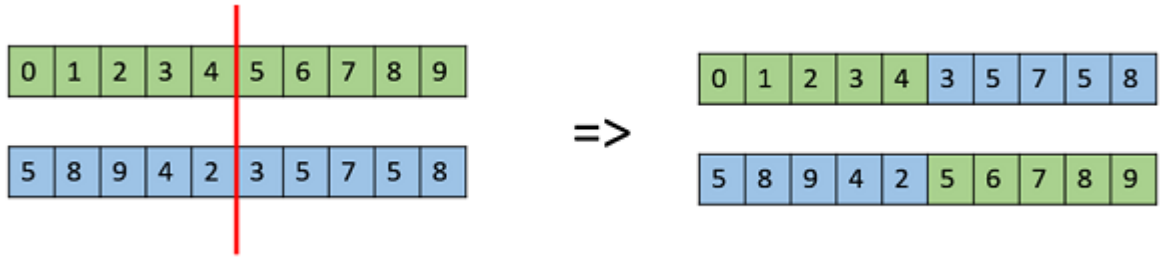


1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.
2. Svaka jedinka u populaciji se potom evaluira kako bi se odredio njen *fitness*.
3. Vrši se odabir roditelja čijim ukrštanjem će biti stvoreno potomstvo.
4. Vrši se ukrštanje odabranih parova roditelja.

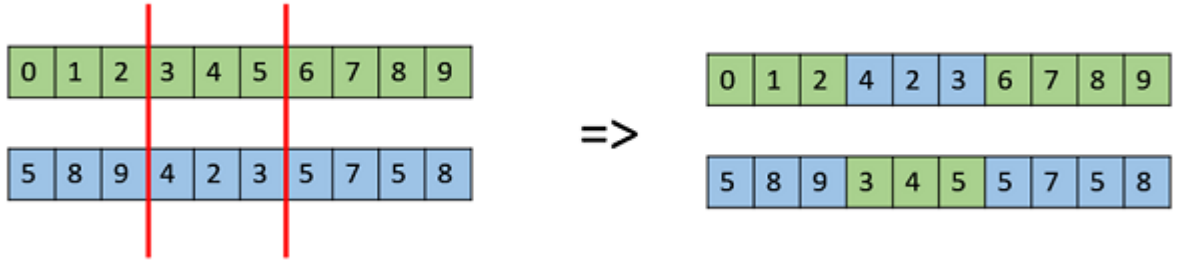
Operatori ukrštanja

- One Point Crossover – Ukrštanje u jednoj tački

Odabira se pozicija ukrštanja nakon čega se bitovi iza odabrane pozicije razmene između roditelja i tako proizvedu dve nove jedinke.



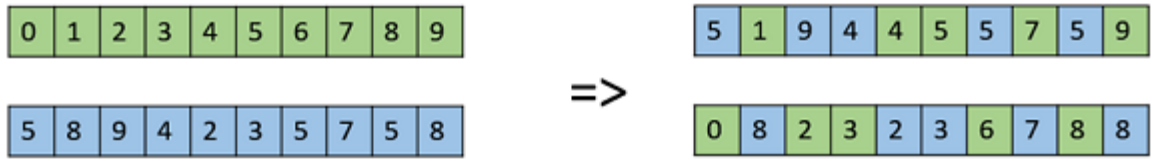
- k Point Crossover – Ukrštanje u k tačaka
- Odabira se k pozicija duž oba roditeljska hromozoma deleći ih tako na $k + 1$ segmenata. Zatim, počev od drugog segmenta, roditelji međusobno razmenjuju svaki drugi podniz bitova



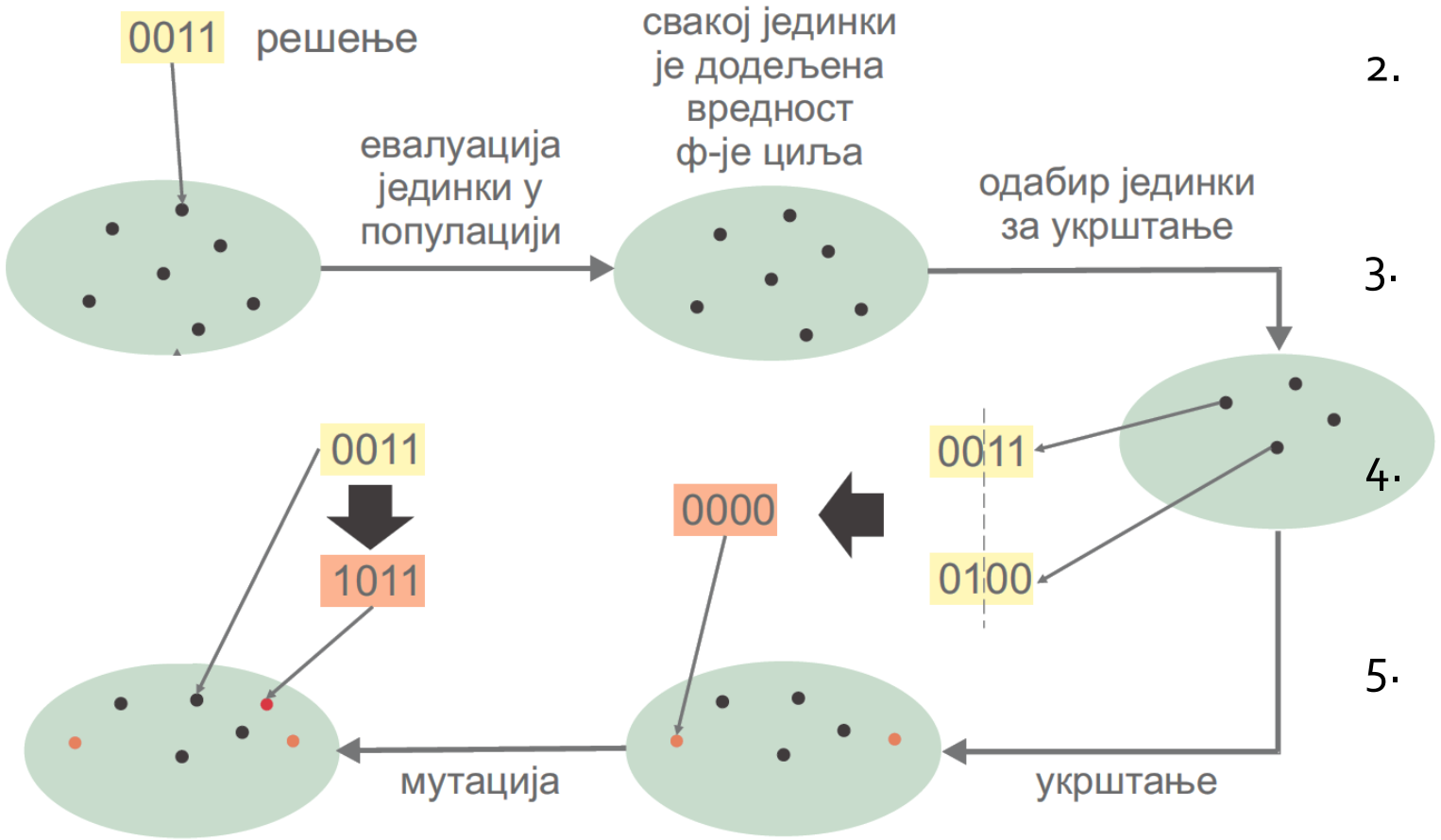
Operatori ukrštanja

- Uniform crossover – uniformno ukrštanje.

Svaki gen se tretira odvojeno. Za svaki par gena se sa određenom verovatnoćom odlučuje od koga roditelja će biti uzet gen na datoj poziciji. Pri tome, ili oba roditelja imaju jednaku šansu da njihov gen završi u potomstvu, ili jedan roditelj dobija veću šansu od drugog.



Kako radi GA?



1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.
2. Svaka jedinka u populaciji se potom evaluira kako bi se odredio njen *fitness*.
3. Vrši se odabir roditelja čijim ukrštanjem će biti stvoreno potomstvo.
4. Vrši se ukrštanje odabranih parova roditelja.
5. Mutacija novonastalih rešenja

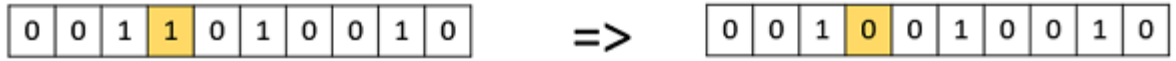
Mutacija

- Operator mutacije uvodi nasumične proneme u karakteristike hromozoma.
- U opštem slučaju, mutacija se primenjuje na nivou gena.
- Stopa mutacije je obično jako mala i zavisi od dužine hromozoma.
- Dakle, novi hromozom neće biti mnogo drugačiji od onog koji je mutirao.
- Mutacija igra ključnu ulogu u GA.
- Ukrštanjem, populacija konvergira ka sve sličnijim hromozomima. Mutacija vraća gensku raznovrsnost i pomaže izlazak iz lokalnih optimuma.

Operatori mutacije

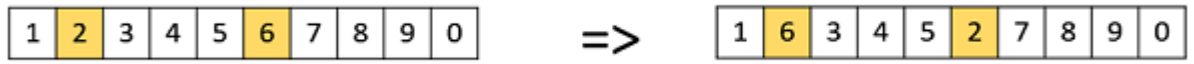
- Bit Flip Mutation

Vrši se promena jednog bita sa određenom verovatnoćom. (Za binarno kodirane hromozome)



- Swap Mutation

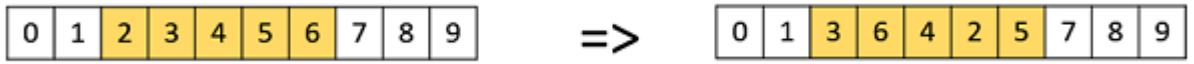
Slučajno se biraju se dve pozicije u hromozomu i razmenjuju se vrednosti gena koji se nalaze na tim pozicijama. (Za permutacijske hromozome)



Mutation operators

- Scramble Mutation

Slučajnim izborom određuje se podniz gena i vrednosti gena u podnizu se randomno izmešaju.



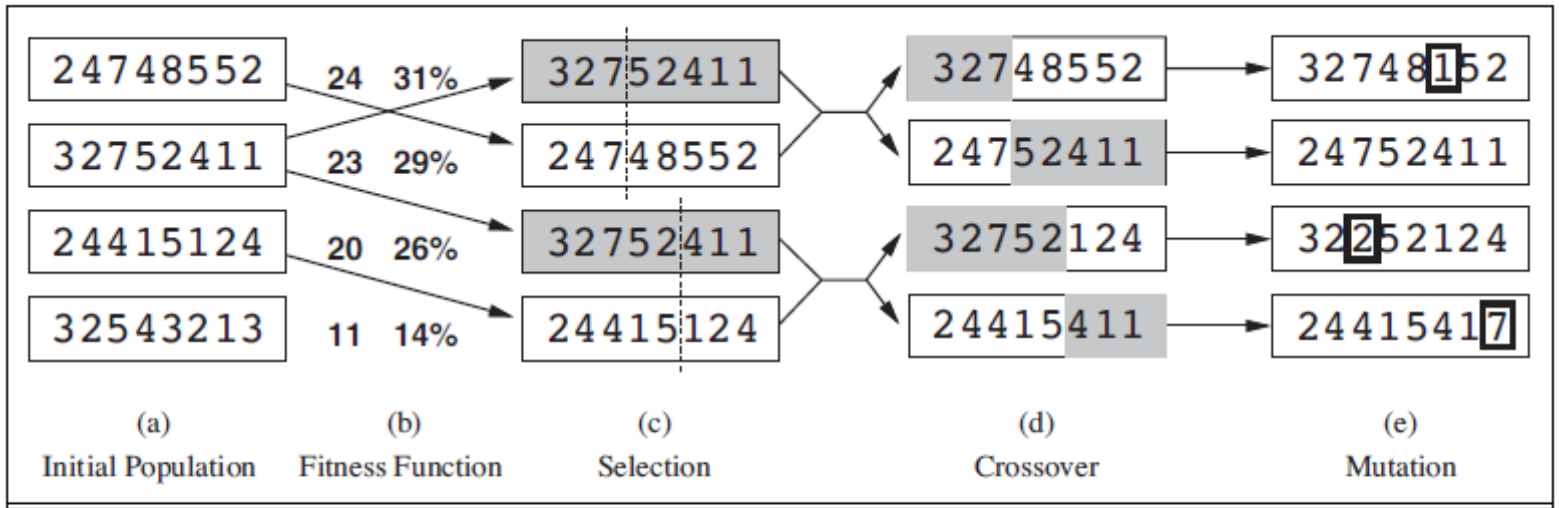
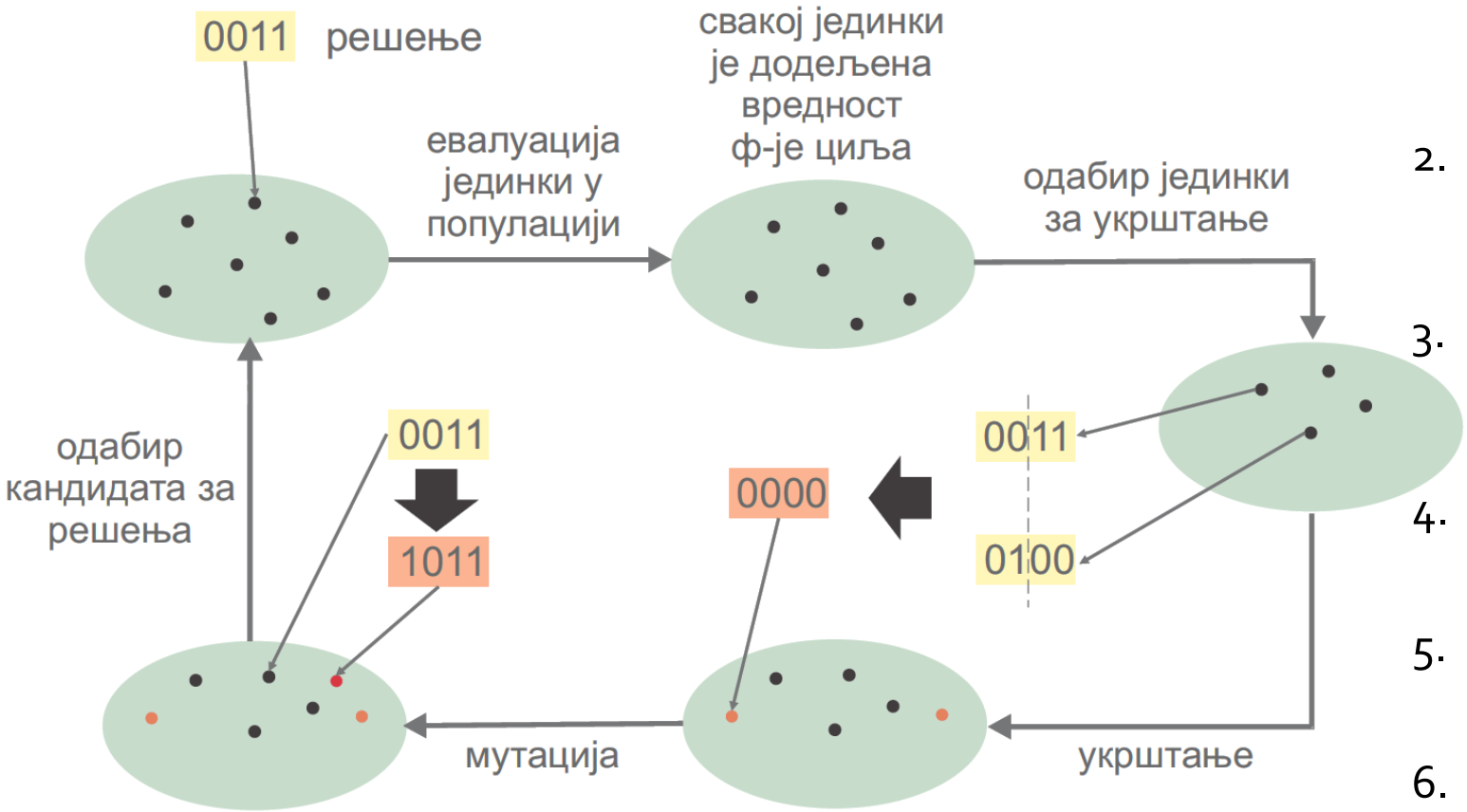


Figure 4.6 FILES: figures/genetic.eps (Tue Nov 3 16:22:53 2009). The genetic algorithm, illustrated for digit strings representing 8-queens states. The initial population in (a) is ranked by the fitness function in (b), resulting in pairs for mating in (c). They produce offspring in (d), which are subject to mutation in (e).

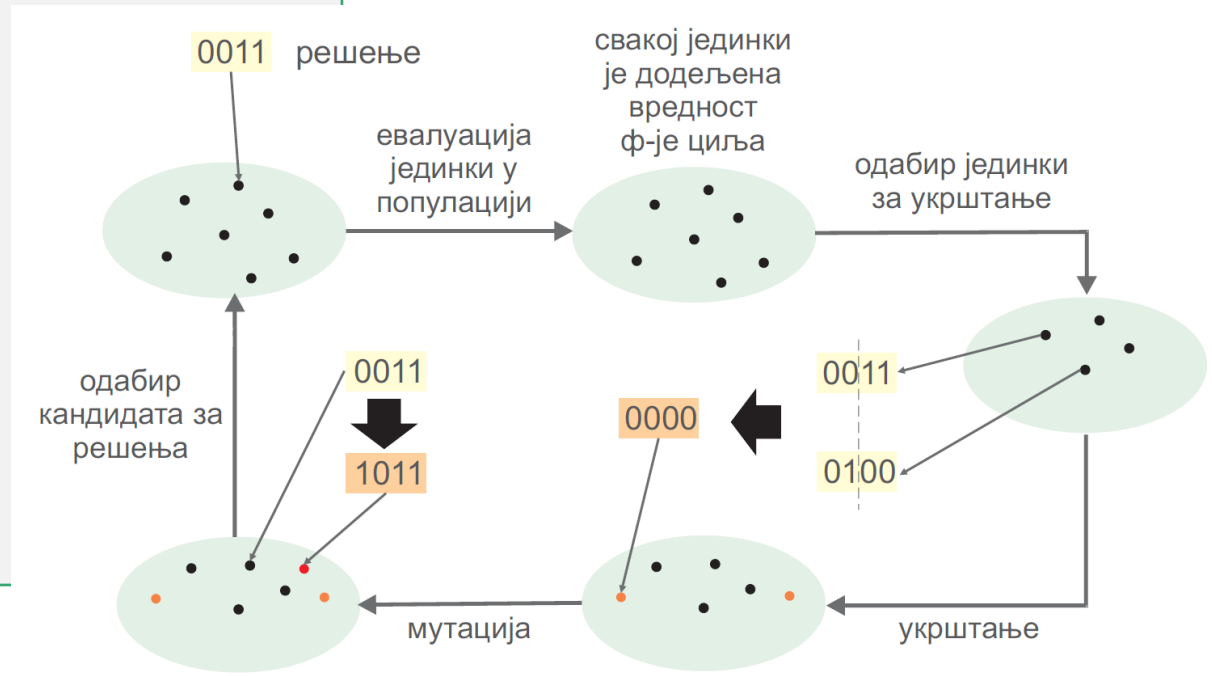
Kako radi GA?



1. GA polazi od generacije slučajno odabranih jedinki u prostoru pretrage.
2. Svaka jedinka u populaciji se potom evaluira kako bi se odredio njen *fitness*.
3. Vrši se odabir roditelja čijim ukrštanjem će biti stvoreno potomstvo.
4. Vrši se ukrštanje odabranih parova roditelja.
5. Mutacija novonastalih rešenja
6. Dodavanje potomaka u populaciju potomaka

Pseudokod algoritma

```
generisi pocetnu populaciju jedinki na slucajan nacin
za svaku jedinku u populaciji izvrsi
{
     odredi vrednost funkcije cilja
}
dok nije zadovoljen uslov_za_zavrsetak_algoritma
{
     dok nije velicina_populacije_potomaka == velicina_populacije
     {
         selekcija
         ukrstanje
         mutacija
         dodavanje potomaka u populaciju potomaka
      }
     populacija potomaka postaje nova populacija
     za svaku jedinku u populaciji izvrsi
     {
         odredi vrednost funkcije cilja
     }
}
```



Nova populacija

- Hollandov originalni GA imao je generacijski pristup: selekcija, ukrštanje i mutacija primenjivane su na populaciju M hromozoma sve dok nije generisan novi skup M jedinki. Ovaj skup je tada postao nova populacija.

- Sa gledišta optimizacije ovo se nije dobro rešenje - možda smo uložili značajan napor da bismo dobili dobro rešenje, zašto bismo rizikovali da ga odbacimo i na taj način sprečimo da učestvuje u daljoj reprodukciji.

- Iz tog razloga, De Jong je uveo koncepte elitizma i preklapanja populacija.

Nova populacija

- Elitizam - kopiranje najboljeg primerka iz jedne generacije u narednu generaciju.

- Preklapanje populacija podrazumeva da u novoj populaciji potomci imaju samo udeo G (*generation gap*)

Problem ranca

Item	Value	Weight
1	4	12
2	2	1
3	10	4
4	1	1
5	2	2