

Prirodom inspirirani optimizacijski algoritmi

Dr.sc. Marko Čupić
Zagreb, 24. svibnja 2012.



Uvod

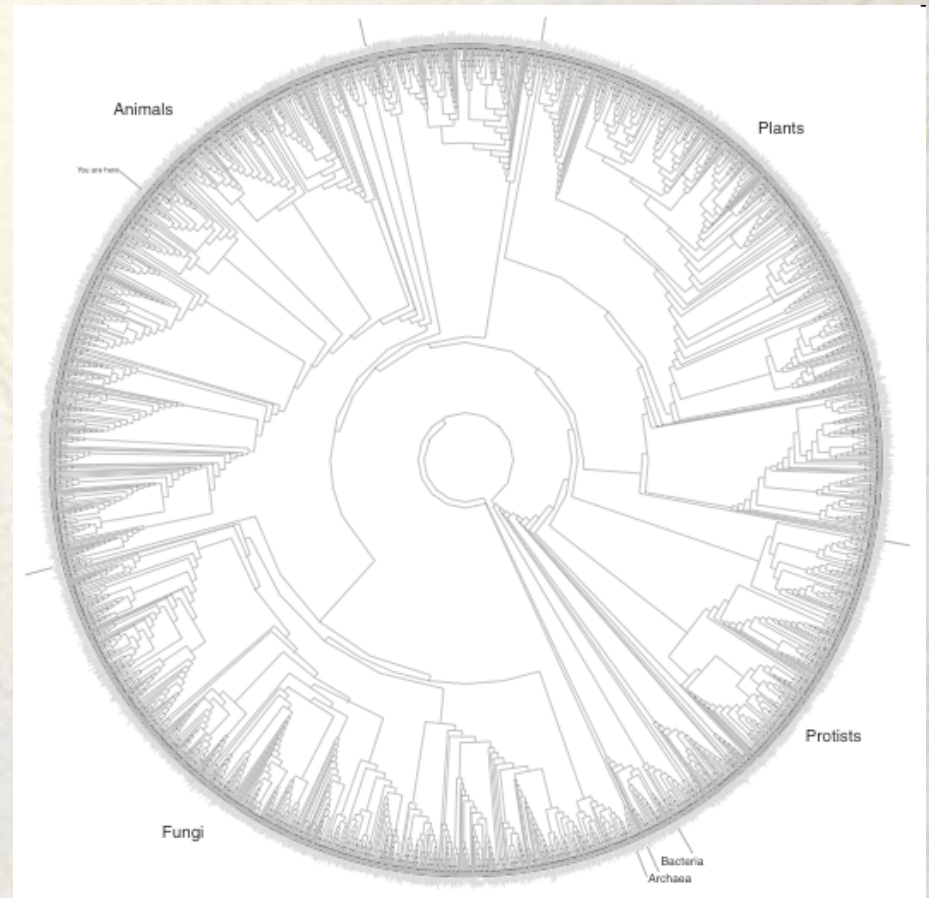
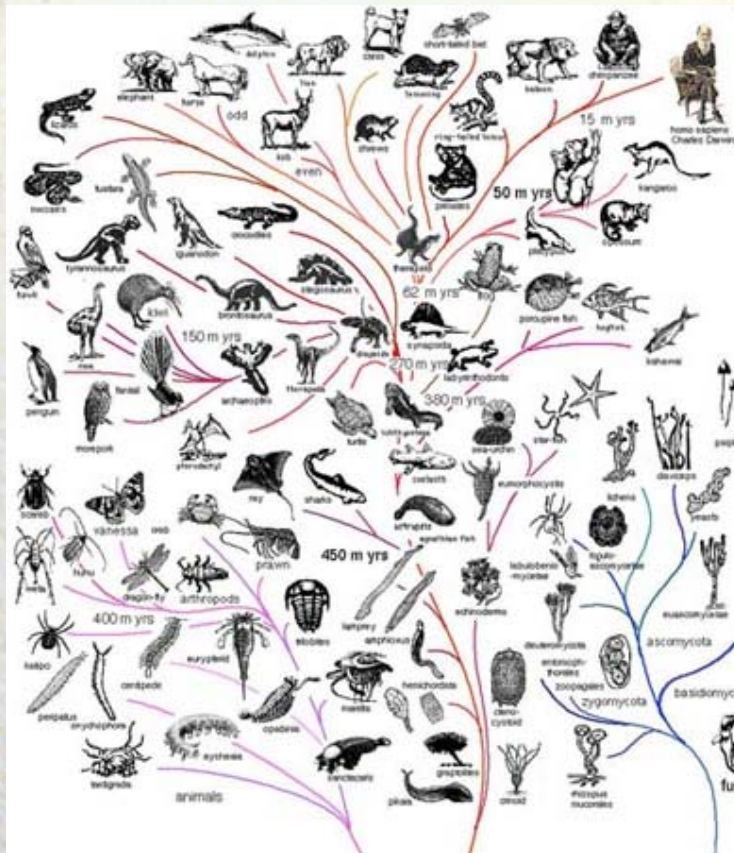
- Izranjajuća inteligencija
- Optimizacijski problemi
- Genetski algoritam
- Algoritam kolonije mrava

Izranjajuća inteligencija

- **Izranjajuća inteligencija**
- Optimizacijski problemi
- Genetski algoritam
- Algoritam kolonije mrava

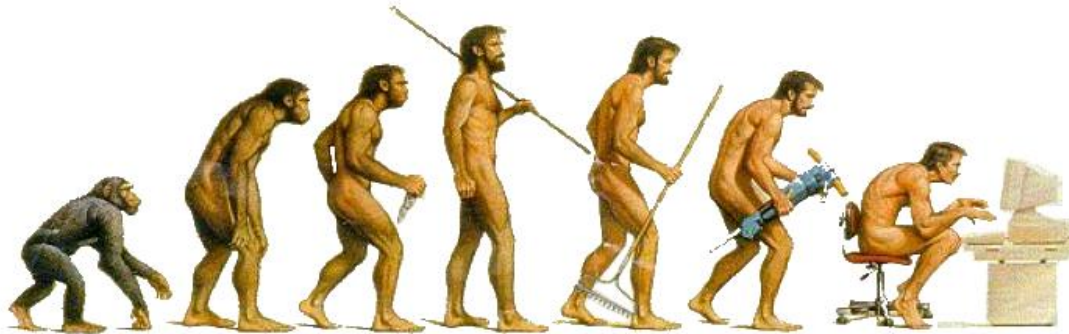


Uvod: Evolucija



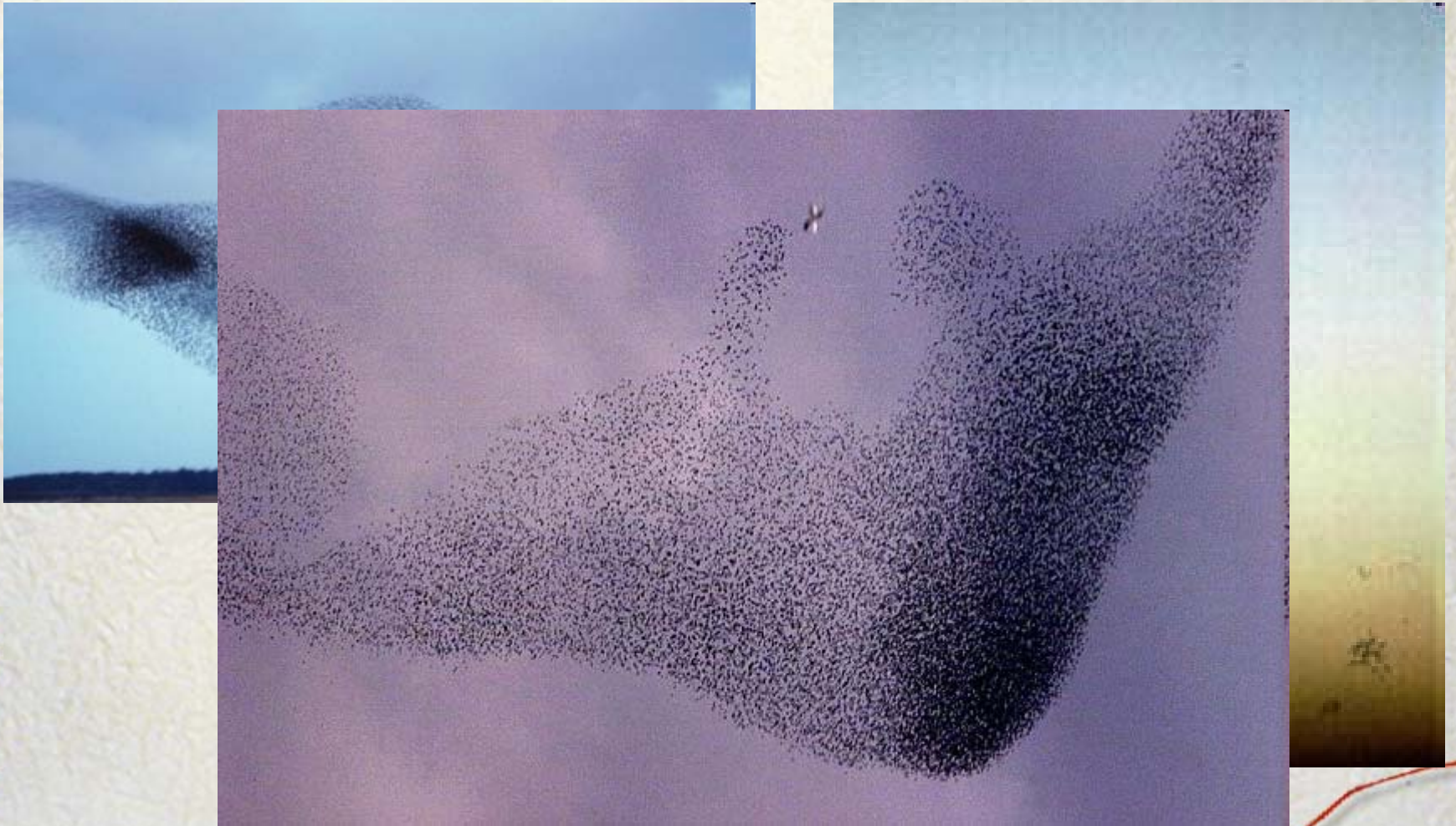
Uvod: Evolucija

Evolution



(OR is it?)

Uvod: Jato ptica



Uvod: Rojenje pčela



Uvod: Mravlje kolonije



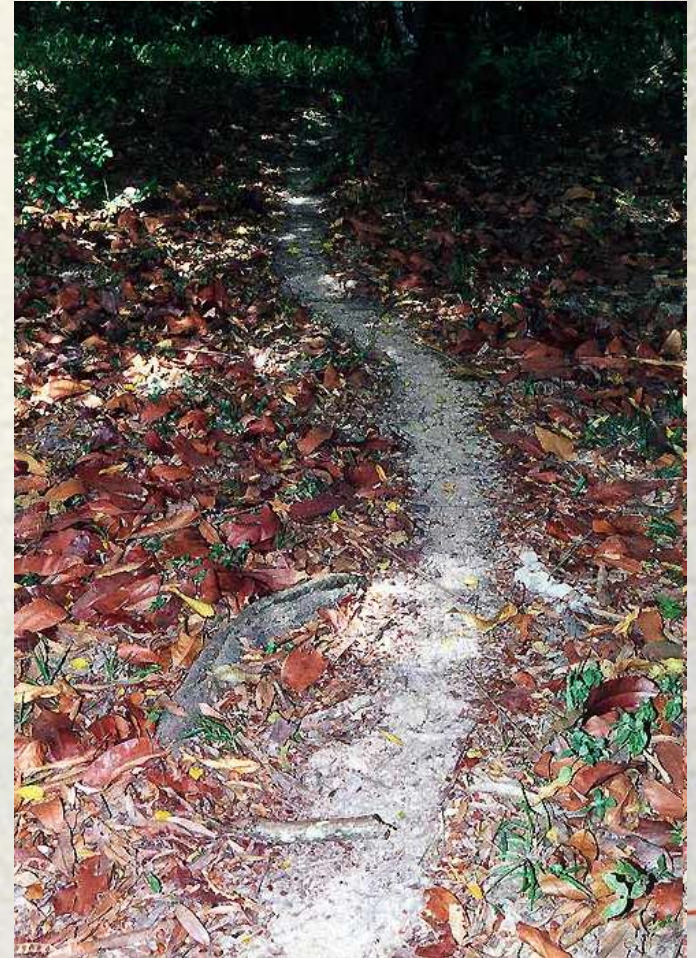
- Površina 50 m², dubina 8 metara

Uvod: Mravlje kolonije



- Mravinjak, put...

Uvod: Mravlje kolonije



Uvod: Mravlje kolonije



Uvod: Jato riba



Uvod: zaključci

Dumb parts, properly
connected
into a swarm, yield smart
results.

Kevin Kelly

New Rules for the New Economy

Sep 1997

Uvod: zaključci

The **whole** is **greater**
than
the sum of the **parts.**

Optimizacijski problemi

- Izranjajuća inteligencija
- **Optimizacijski problemi**
- Genetski algoritam
- Algoritam kolonije mrava



Optimizacijski problemi

- Optimizacija: postupak pronalaženja najboljeg rješenja problema, rješenja s najmanjom cijenom
- Tipično:
 - Kontinuirano područje
 - Kombinatorički problemi

Optimizacijski problemi

- Pretraživanje prostora stanja:
 - Nađi put od S_0 do S_F
 - Rješenje je **put** (npr. slagalica 3x3!)
- CSP: *Constraint Satisfaction Problem*
 - Vrsta pretraživanja prostora stanja kod kojeg put od početnog do konačnog stanja nije bitan
 - Rješenje je isključivo konačno stanje

Optimizacijski problemi

- CSP: *Constraint Satisfaction Problem*
 - Definira se niz ograničenja koja moraju biti zadovoljena
 - Definira se kriterijska funkcija koja se optimira

Optimizacijski problemi

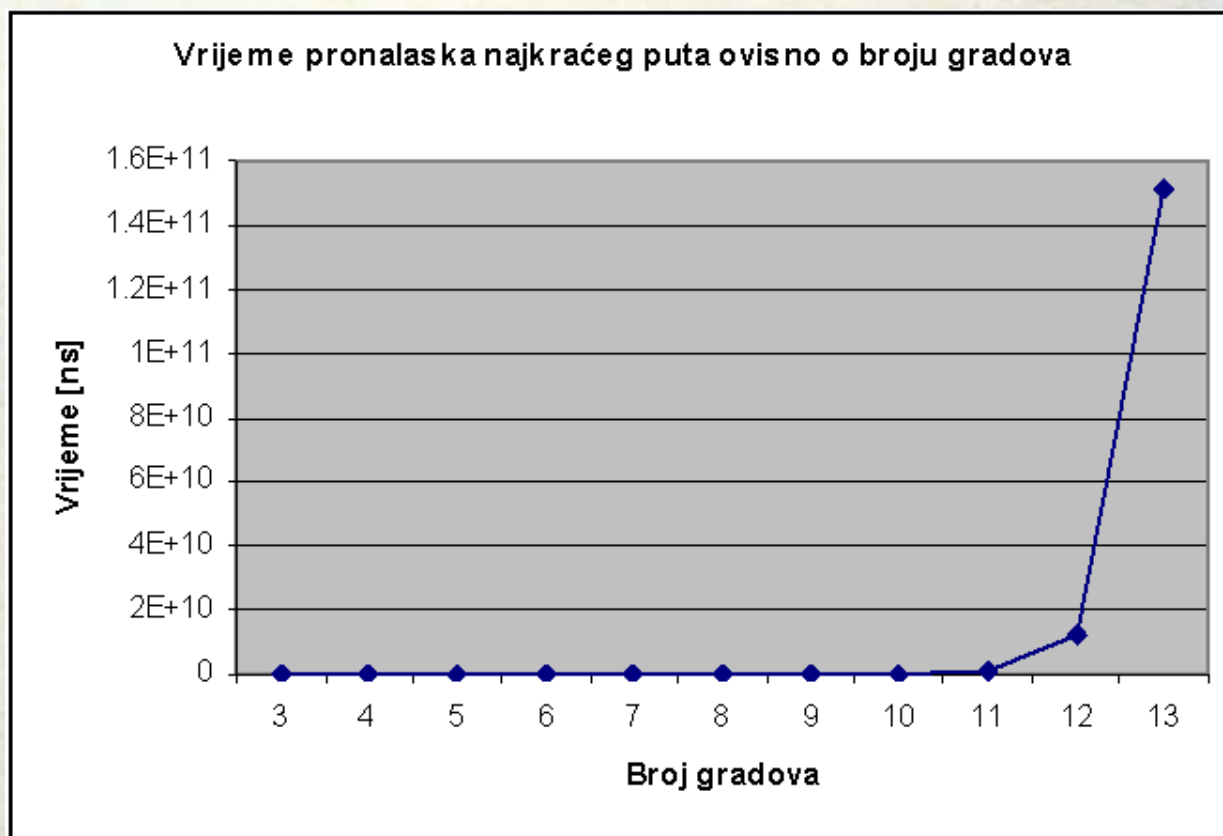
- Način rješavanja kombinatoričkih problema već smo upoznali
 - Algoritmi pretraživanja prostora stanja
 - Pretraživanje u širinu
 - Pretraživanje u dubinu
 - A^*
 - ...
- Nažalost, u praksi često neiskoristivi

Optimizacijski problemi

- Problem trgovačkog putnika
 - Poznate koordinate n gradova na karti
 - Pronaći najkraću turu kroz sve gradove
 - Matematičkim žargonom: pronaći Hamiltonov ciklus u grafu
 - **NP težak problem**
(faktorijska složenost)

Optimizacijski problemi

- Problem trgovačkog putnika



Optimizacijski problemi

- Problem trgovačkog putnika
 - 12 gradova, 12 sekundi
 - 13 gradova, 2,5 minute
 - 14 gradova, pola sata
 - 15 gradova, 7,6 sati
 - 16 gradova, 4,7 dana
 - ...
 - 500 gradova, ?????

Optimizacijski problemi

- Drugi problemi
 - Raspoređivanje neraspoređenih studenata u grupe za predavanja
 - Izrada rasporeda međuispita
 - Izrada rasporeda laboratorijskih vježbi
- Iscrpnom pretragom
 - Puno, puno više vremena no što je svemir star!

Optimizacijski problemi

- Heuristike
 - Algoritmi koji pronalaze *dovoljno* dobra rješenja, tipično ne nude garanciju optimalnosti, te imaju nisku računsku složenost (polinomijalnu)
 - Dijelimo ih na
 - Konstrukcijske
 - Algoritmi lokalne pretrage

Optimizacijski problemi

- Heuristike
 - Konstrukcijske
 - Grade rješenje segment po segment
 - Algoritmi lokalne pretrage
 - Kreću od nekog početnog rješenja i pokušavaju ga inkrementalno poboljšavati

Optimizacijski problemi

- Metaheuristike
 - Skup algoritamskih koncepata koji koristimo za definiranje heurističkih metoda primjenjivih na širok skup problema
 - heuristika koja vodi problemski specifične heuristike

Optimizacijski problemi

- Metaheuristike
 - Simulirano kaljenje
 - Tabu pretraživanje
 - Evolucijski algoritmi
 - Mravlji algoritmi
 - Algoritmi rojeva
 - Algoritmi umjetnih imunoloških sustava
 - ...

Sitan problemčić

- Teorem "No free lunch", Wolpert & Macready, 1995, 1997:
 - *Svi algoritmi koji traže ekstrem funkcije cilja ponašaju se upravo jednako s obzirom na bilo koju mjeru performansi, kada se pogleda njihovo prosječno ponašanje nad svim mogućim funkcijama cilja.*

...

Sitan problemčić

- Teorem "No free lunch", Wolpert & Macready, 1995, 1997:
 - *Konkretno, ako algoritam A nadmašuje algoritam B na nekim funkcijama cilja, tada, grubo govoreći, mora postojati upravo toliko drugih funkcija cilja nad kojima B nadmašuje A.*

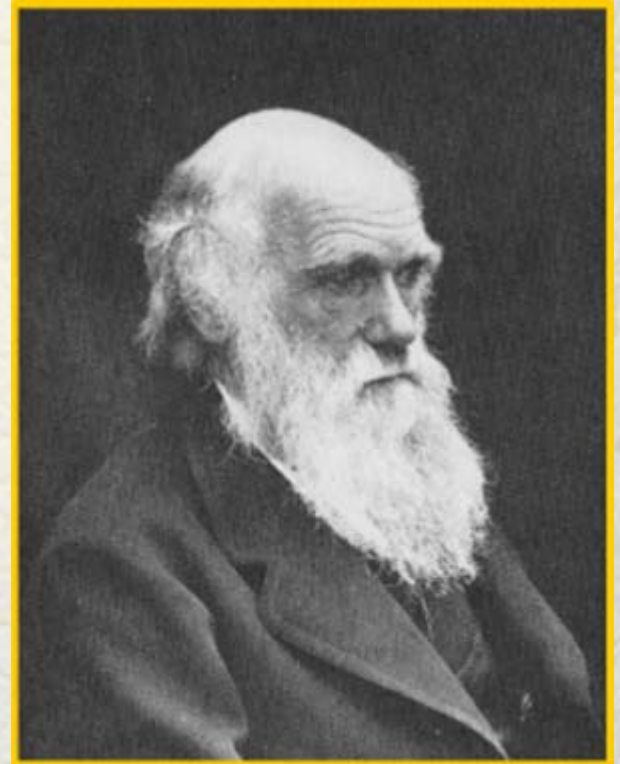
Genetski algoritam

- Izranjajuća inteligencija
- Optimizacijski problemi
- **Genetski algoritam**
- Algoritam kolonije mrava



Genetski algoritam

- Evolucija kao inspiracija
- Populacijski algoritmi
- Darwinova teorija o postanku vrsta

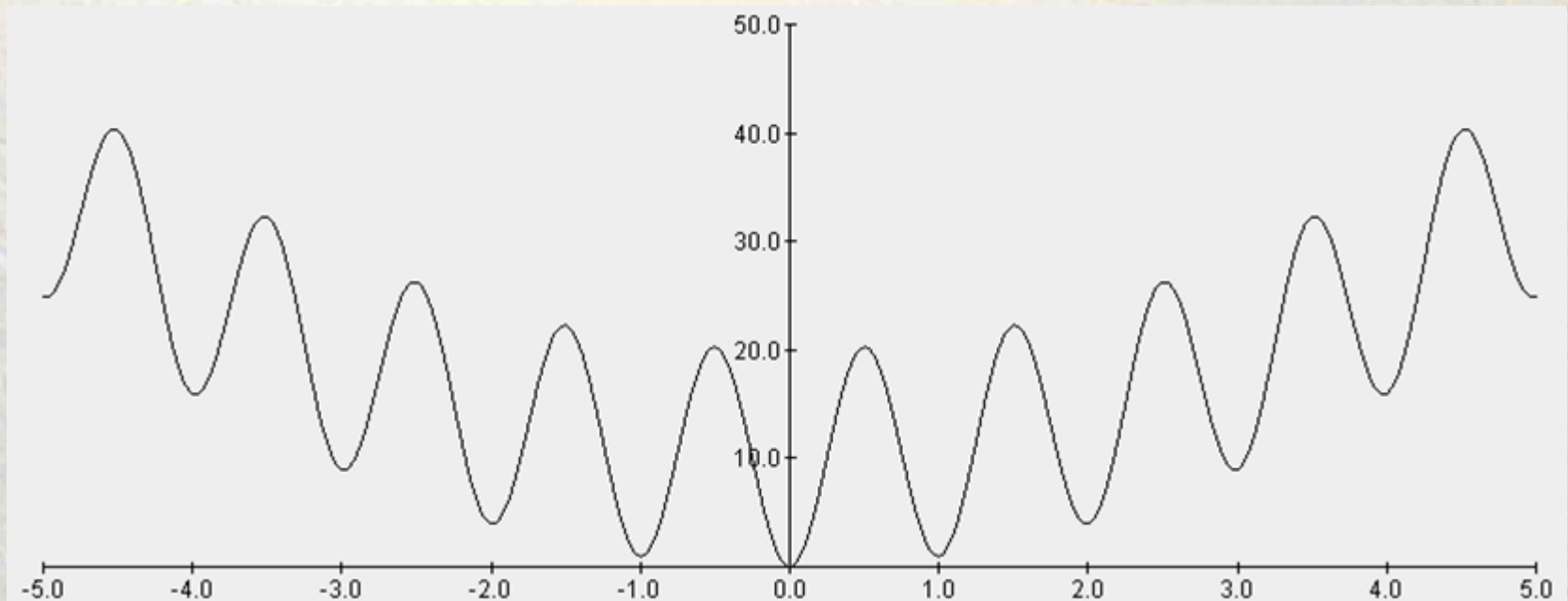


Genetski algoritam

- Temeljne postavke: Darwin
 - plodnost vrsta – potomaka uvijek ima više no što je potrebno
 - veličina populacije je približno stalna
 - količina hrane je ograničena
 - kod vrsta koje se seksualno razmnožavaju, nema identičnih jedinki već postoje varijacije, te
 - najveći dio varijacija prenosi se nasljeđem

Genetski algoritam

- Primjer problema $f(x) = 10 + x^2 - 10 \cdot \cos(2 \cdot \pi \cdot x)$
 - Naći x za koji je $f(x)$ minimalna



Genetski algoritam

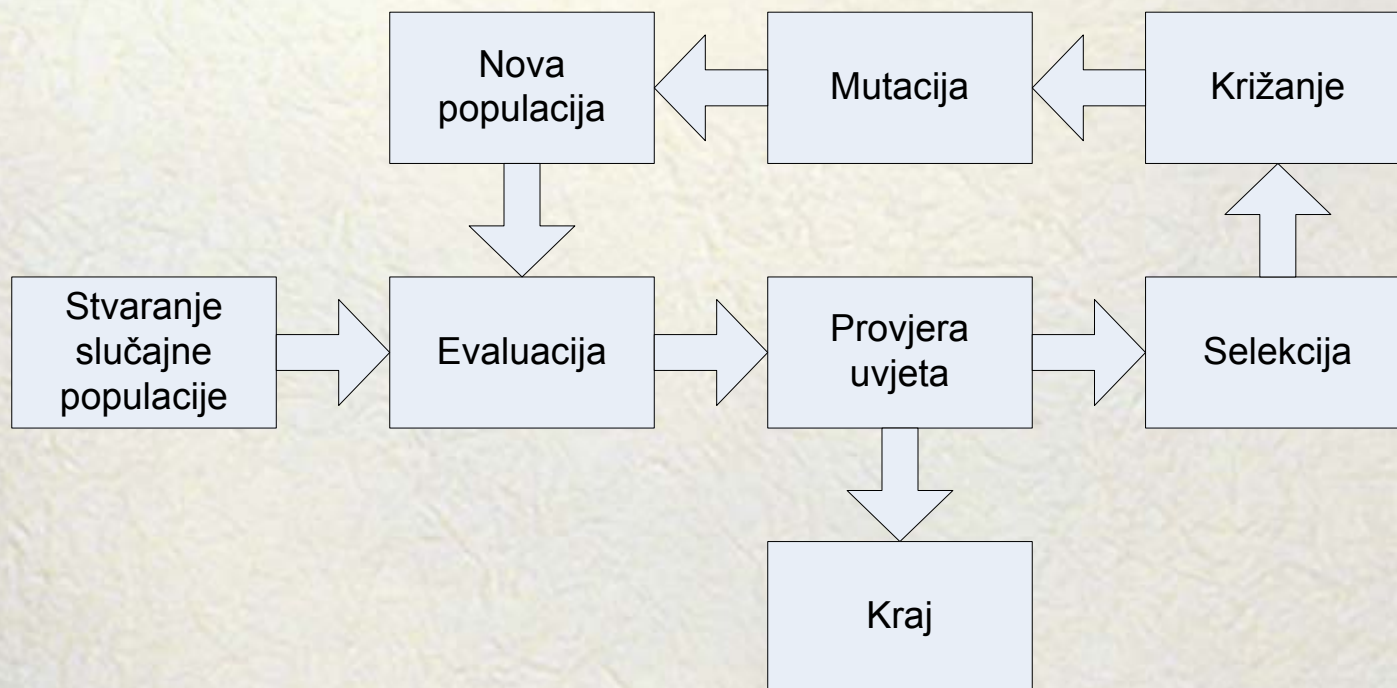
- Kako radi GA?
 - Radimo s *populacijom kromosoma*
 - Svaki *kromosom* je jedno *rješenje* problema
 - Svako rješenje ima svoju *dobrotu* (engl. fitness) ili kaznu
 - U našem primjeru dobrota i $f(x)$ su suprotni
 - ➔ veći $f(x)$, manja dobrota
 - ➔ $f(x)$ odgovara kazni

Genetski algoritam

- Implementacija
 - Iterativno iz trenutne generacije stvaramo sljedeću
 - Potomke imaju veću šansu stvoriti bolja rješenja
 - Rješenja se kombiniraju operatorom križanja
 - Operator mutacije

Genetski algoritam

- Dijagram toka



Genetski algoritam

- Uloga
 - **Selekcija** → selekcijski pritisak → brzina konvergencije
 - **Križanje** → pretraživanje okoline roditelja
 - **Mutacija** → bijeg iz lokalnih ekstrema, veliki skokovi u prostoru pretraživanja

Genetski algoritam

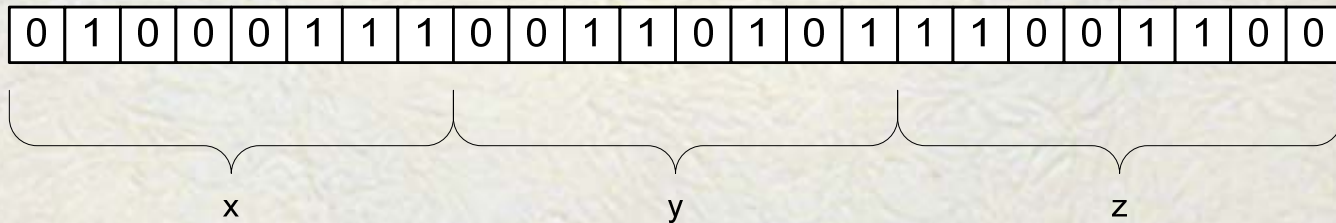
- Binarni kromosom

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

 - Niz nula i jedinica koji se interpretira kao rješenje problema (vrijednost varijable)
 - Npr. Trobitni kromosom: 000, 001, ..., 111
 - Ako predstavlja realnu varijablu iz područja $[-2, 2]$, tada:
 $000 \equiv -2$, $001 \equiv -1.43$, ..., $111 \equiv 2$
 - Broj bitova za određenu preciznost?

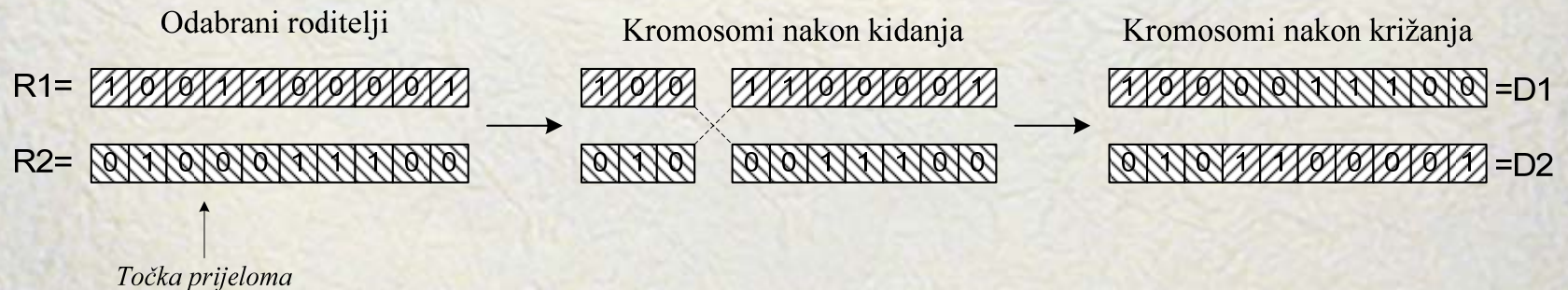
Genetski algoritam

- Binarni kromosom
 - Složeniji primjer
 - rješenje za funkciju od tri varijable x, y, z



Genetski algoritam

- Križanje s jednom točkom prijeloma
 - Odabiru se dva roditelja
 - Slučajno se odabire točka prijeloma
 - Obavlja se križanje

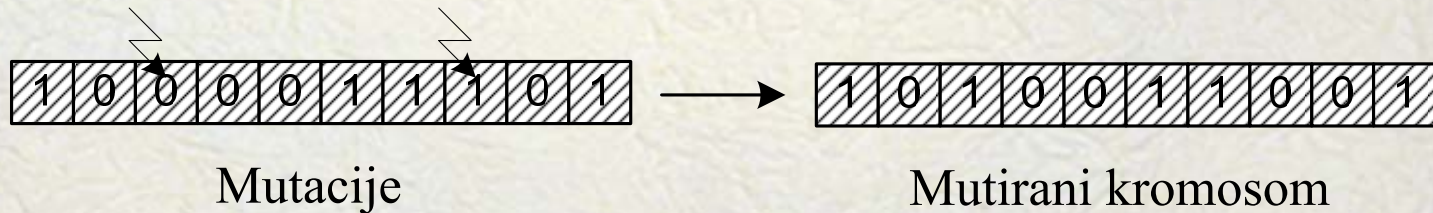


Genetski algoritam

- Više vrsta križanja
 - Križanje s jednom točkom prijeloma
 - Križanje s n točaka prijeloma
 - Uniformno križanje
 - ...

Genetski algoritam

- Operator mutacije
 - Zadana vjerojatnost mutacije bita
 - Operator okreće vrijednost bita



- Može generirati veliku promjenu!

Genetski algoritam

- Umjesto binarno, za prikaz rješenja možemo koristiti i polje decimalnih brojeva
 - Križanje:
 - $d(i) = (r1(i) + r2(i))/2$
(d ... dijete; r1, r2 ... roditelji)
 - Mutacija:
 - $d(i) += \text{gauss}(0, K)$ ili $d(i) += \text{uniformno}(-K, K)$
 - Za svaki i ili samo za neke slučajno odabrane

Genetski algoritam

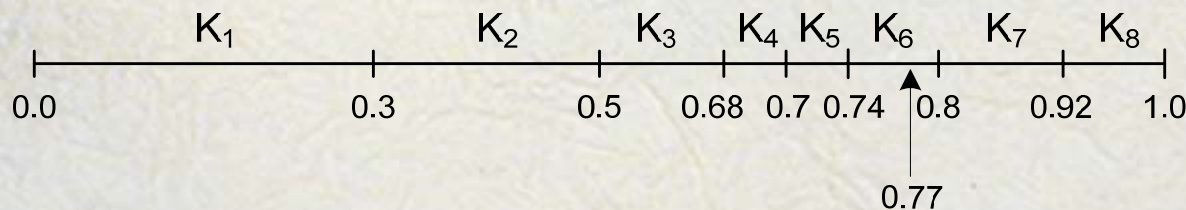
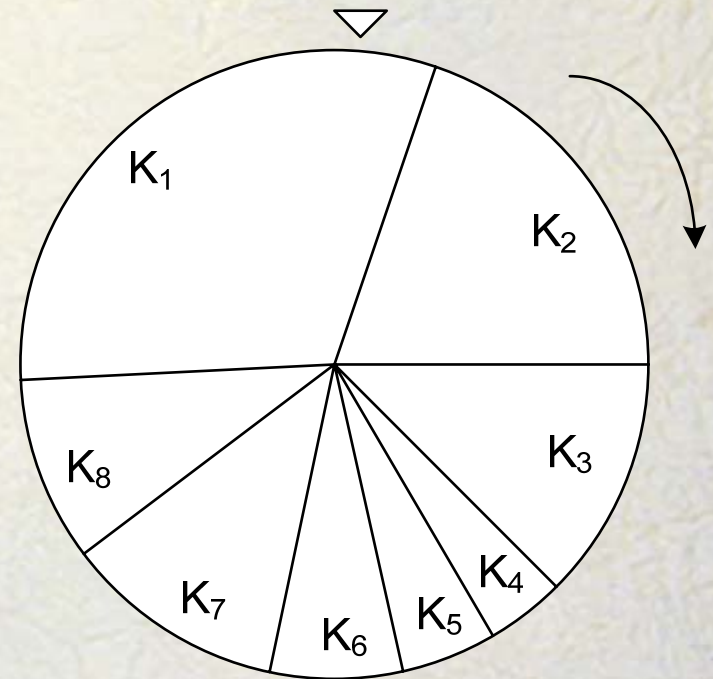
- Izbor roditelja – više tehnika
 - Proporcionalna selekcija
(engl. Roulette-wheel selection)
 - Što je jedinka bolja, to ima veću šansu biti izabrana

$$probSel(i) = \frac{fit(i)}{\sum_{j=1}^n fit(j)}$$

Genetski algoritam

- Izbor roditelja – proporcionalna selekcija

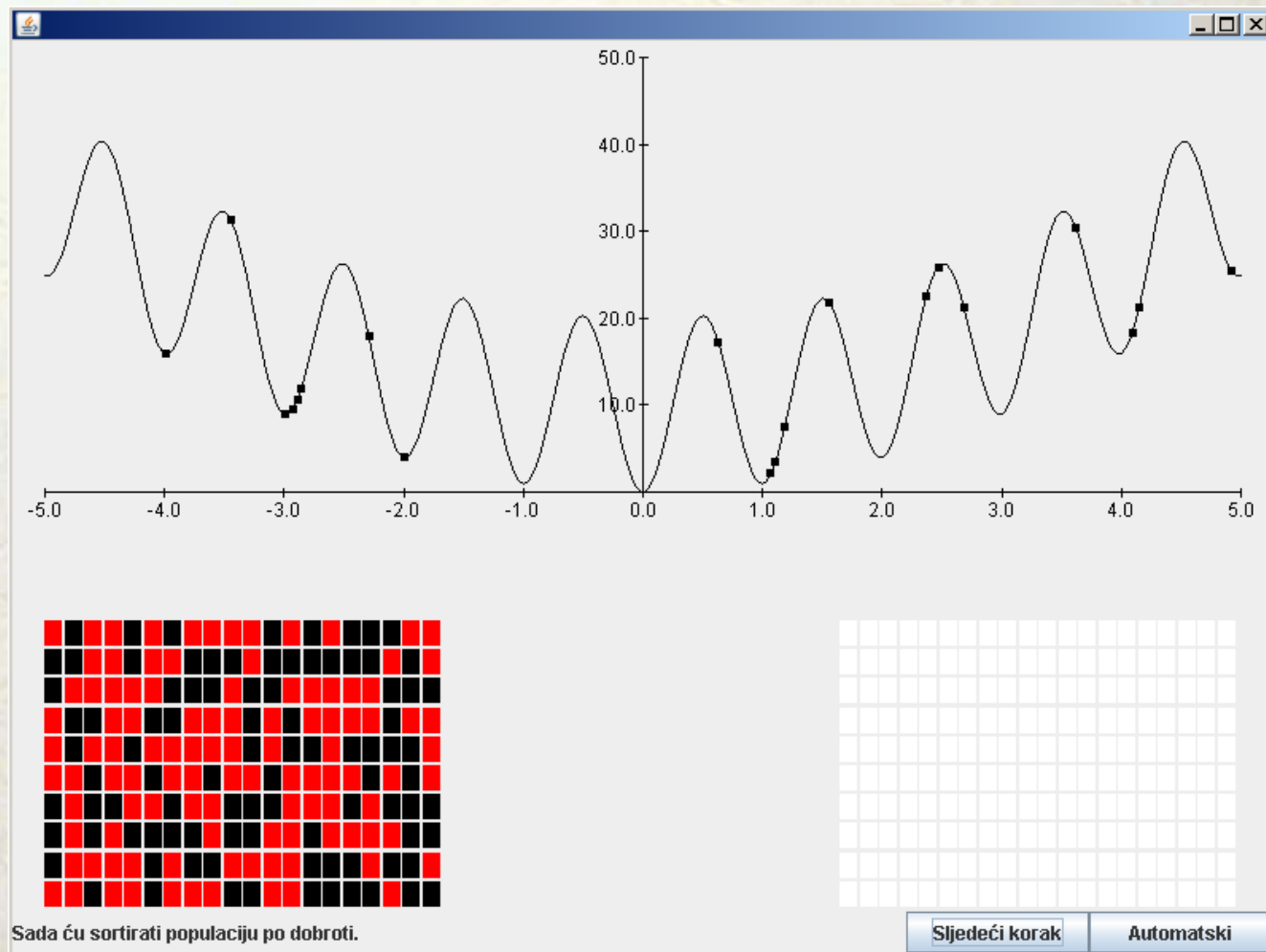
$$len(i) = \frac{fit(i)}{\sum_{j=1}^n fit(j)}$$



Genetski algoritam

```
P = stvori_početnu_populaciju(VEL_POP)
evaluiraj(P)
ponavljaj_dok_nije_kraj
    nova_populacija P' = ∅
    ponavljaj_dok_je_veličina(P') < VEL_POP
        odaberi R1 i R2 iz P
        {D1, D2} = krizaj(R1, R2)
        mutiraj D1, mutiraj D2
        dodaj D1 i D2 u P'
    kraj
    P = P'
    evaluiraj(P)
Kraj
```


Genetski algoritam



Algoritam kolonije mrava

- Izranjajuća inteligencija
- Optimizacijski problemi
- Genetski algoritam
- **Algoritam kolonije mrava**



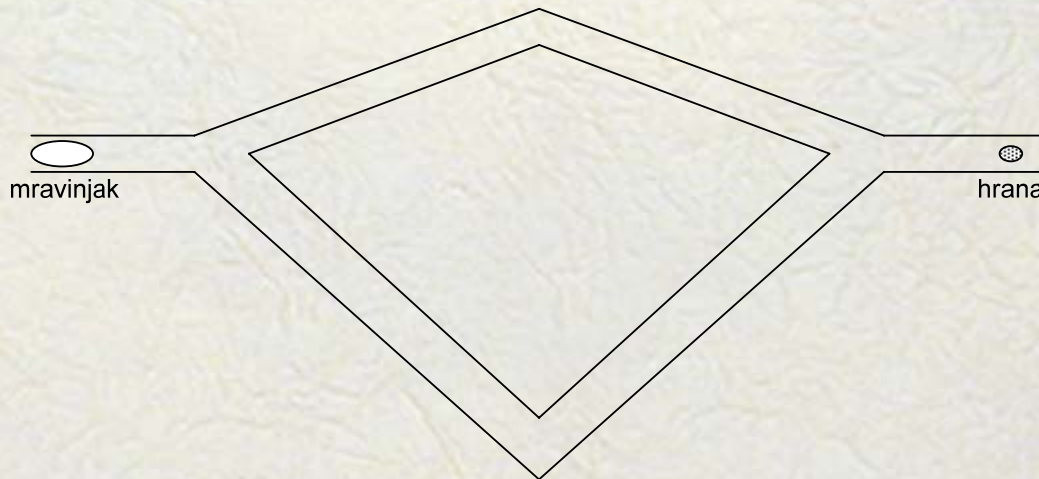
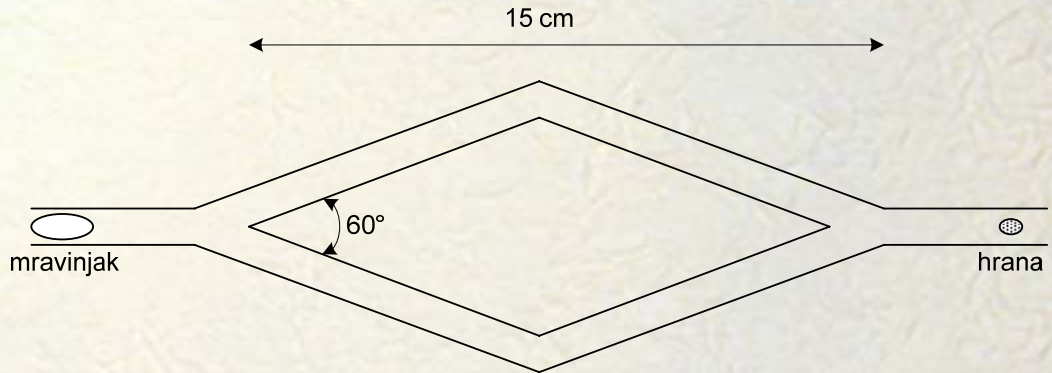
Algoritam kolonije mrava

- Mravi iskazuju zanimljivo ponašanje
 - Uspješno pronalaze najkraći put do izvora hrane



Algoritam kolonije mrava

- Eksperimenti



Algoritam kolonije mrava

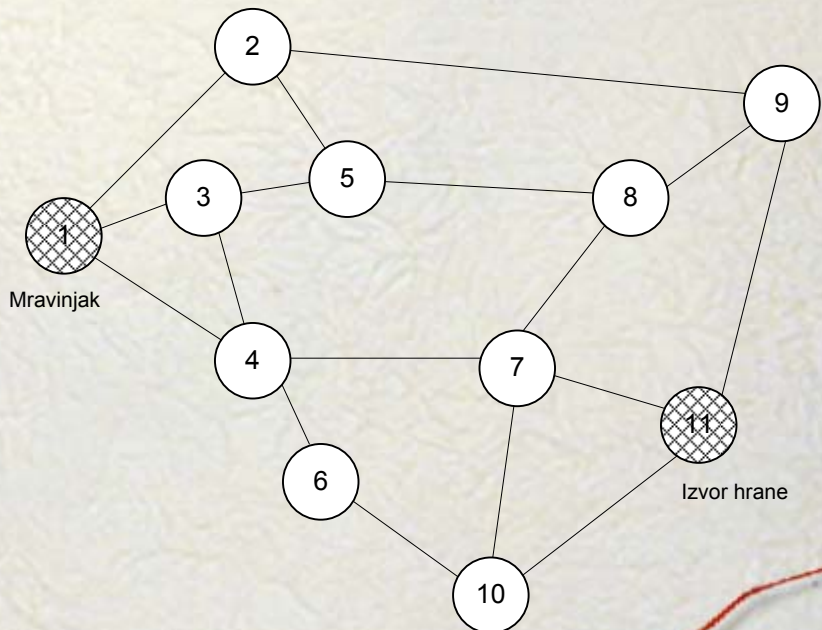
- Objašnjenje
 - Mravi prilikom kretanja za sobom ostavljaju feromonski trag
 - Mrav se kreće slučajno, ali s većom vjerojatnošću u smjeru u kojem osjeti jači feromonski trag

Algoritam kolonije mrava

- Direktna primjena na probleme prikazive grafovima
- Npr. Iz 1 može u 2, 3 i 4

$$\tau_0 = konst$$

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha}{\sum_{l \in N_i^k} \tau_{il}^\alpha}, & \text{ako } j \in N_i^k \\ 0, & \text{ako } j \notin N_i^k \end{cases}$$

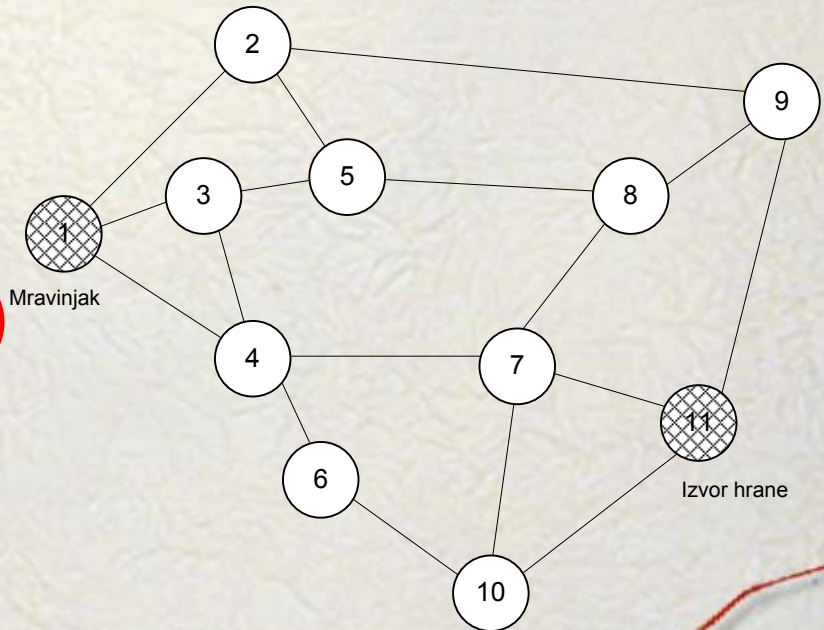


Algoritam kolonije mrava

- Algoritam Ant System
 - Uporaba heurističke informacije dodatno poboljšava ponašanje

$$\tau_0 = \frac{m}{C^{nn}}$$

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in N_i^k} (\tau_{il}^\alpha \cdot \eta_{il}^\beta)}, & \text{ako } j \in N_i^k \\ 0, & \text{ako } j \notin N_i^k \end{cases}$$



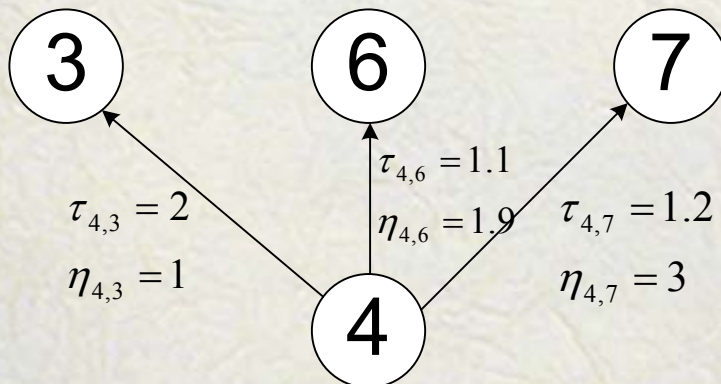
Algoritam kolonije mrava

- Algoritam Ant System

```
ponavljaj dok nije kraj
  ponovi za svakog mravca
    stvori rješenje
    vrednuj rješenje
  kraj ponovi
  ispari feromonske tragove
  ponovi za sve_ili_neke mrave
    azuriraj feromonske tragove
  kraj ponovi
kraj ponavljanja
```


Algoritam kolonije mrava

- Procedura: Stvori rješenje
 - Mrav kreće iz nekog čvora
 - Temeljem vjerojatnosti bira sljedeći čvor, pa sljedeći, sve dok ne dođe do zadnjeg čvora



Uz $\alpha=1$, $\beta=2$:

$$p(4 \rightarrow 3) = 11,9\%$$

$$p(4 \rightarrow 6) = 23,7\%$$

$$p(4 \rightarrow 7) = 64,4\%$$

Algoritam kolonije mrava

- Procedura: Vrednuj rješenje
 - Funkcija računa ukupnu duljinu puta
 - Prelazak iz jednog čvora u drugi tipično je povezan određenom cijenom (gradovi → udaljenost)

Algoritam kolonije mrava

- Procedura: Ispari tragove
 - Funkcija feromonske tragove na svim bridovima umanjuje za određeni iznos

$$\tau_{ij} \leftarrow \tau_{ij} \cdot (1 - \rho)$$

- Geometrijska progresija!
- Izuzetno skupo (graf ima puno bridova)

Algoritam kolonije mrava

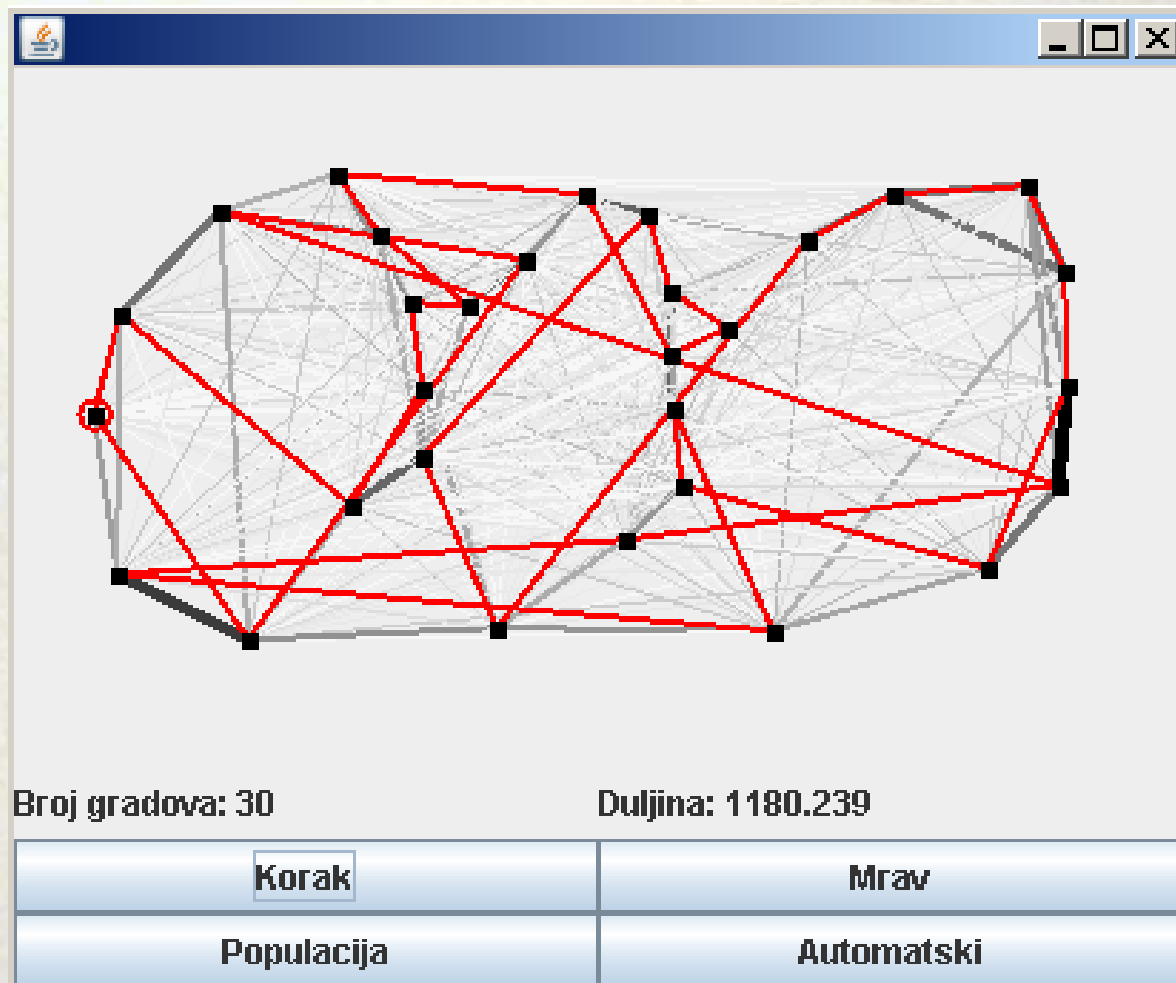
- Procedura: Ažuriraj tragove
 - Funkcija za odabranog mrava dodaje nove feromonske tragove iznosa:

$$\Delta \tau_{ij}^k = \begin{cases} 1/C^k, & \text{ako je brid } i - j \text{ na stazi } k\text{-tog mrava} \\ 0, & \text{inačn} \end{cases}$$

- Novo stanje je tada:

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^m \Delta \tau_{ij}^k$$

Algoritam kolonije mrava



Zaključak

- Prirodom inspirirani algoritmi – danas vruće područje istraživanja!
- Po prvi puta pronađen način kako se uhvatiti u koštac s prethodno nerješivim problemima
- Svaki nekoliko godina novi algoritam (primjerice Bee Colony Optimization, Intelligent Water Drops, ...)

Linkovi

- Video isječak o koloniji mrava
<http://www.inquisitr.com/14238/holy-crap-billions-of-ants-in-one-colony/>

Linkovi

- Skripta
<http://java.zemris.fer.hr/nastava/ui/>
- Implementacija svih algoritama
[http://java.zemris.fer.hr/nastava/ui/ev
oAlg.zip](http://java.zemris.fer.hr/nastava/ui/ev
oAlg.zip)

