


# Odabrane teme iz evolucijskog računarstva



# Motivacija

- Sada znate dobro osnovne metode iz evolucijskog računarstva
- Koji su trenutno smjerovi istraživanja?
- Koji su novi i interesantni algoritmi?

# Teme

- Hipeheuristike
- Transfer learning
- Evolutionary multitasking
- Surogatni modeli
- Fitness landscape analysis

# Hiperheuristike

- Metaheuristika koja izgrađuje heuristiku
- Ideja: automatski izraditi novu heuristiku za konkretnu vrstu problema, koja može riješavati više problema
- Vrste:
  - Seleksijske
  - Generativne

# Transfer learning

- Ideja iz strojnog učenja
- Motivacija:
  - Skupo je svaki problem rješavati iz nule
  - Možemo li rješenja za jedne probleme iskoristiti za rješavanje drugih problema
  - Ideja, iskoristiti znanje stečeno tijekom rješavanja jednog problema nad nekim drugim, sličnim problemom

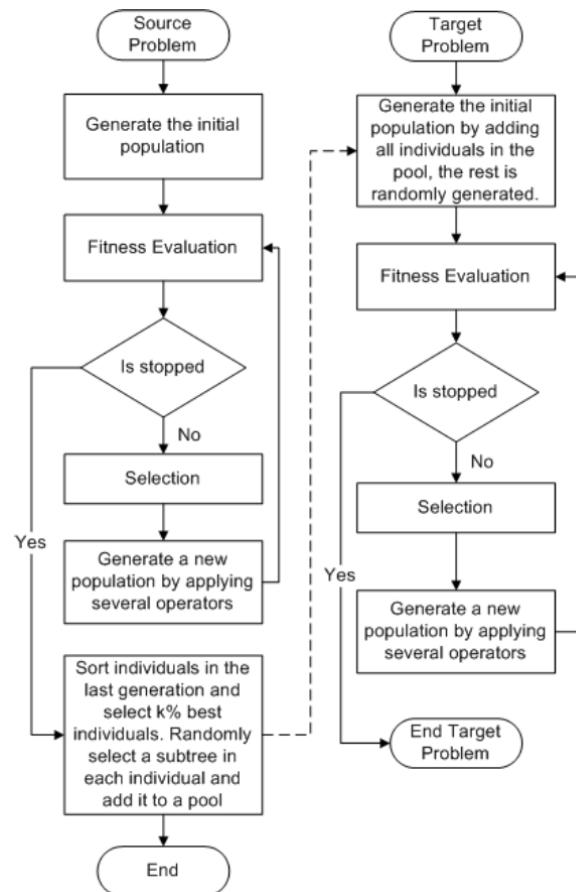
# Kako te principe iskoristiti u evolucijskim algoritmima?

- Transfer learning u genetskom programiranju
- Multifactorial evolutionary learning
- Multitasking genetic algorithm

# Transfer learning

- Popularno u strojnom učenju
- Često se koristi u problemima računalnog vida i obrade prirodnog jezika
- Npr. Model treniran za prepoznavanje automobila se može iskoristiti kao polazišna točka za treniranje klasifikatora za prepoznavanje kamiona

# Transfer learning u GP-u





# Multitask learning

- Za razliku od transfer learninga ideja je više problema rješavati u paraleli
- Iskoristiti implicitni paralelizam koji imamo u metaheuristikama u obliku populacija
- Ideja: kroz evoluciju neka rješenja će se specijalizirati za rješavanje jednog ili drugog problema

# Multitask learning

- Zašto bi uopće rješavali dva problema u paraleli?
- Intuicija: znanja o jednom problemu mogu nam možda pomoći u rješavanju drugog problema
- Cilj je da se onda kroz podjelu znanja ubrza konvergencija ili poboljšaju rješenja

# Multifactorial evolutionary algorithm

- Prvi i najpopularniji algoritam za multitasking optimizaciju
- Tražimo minimume za  $K$  problema istovremeno

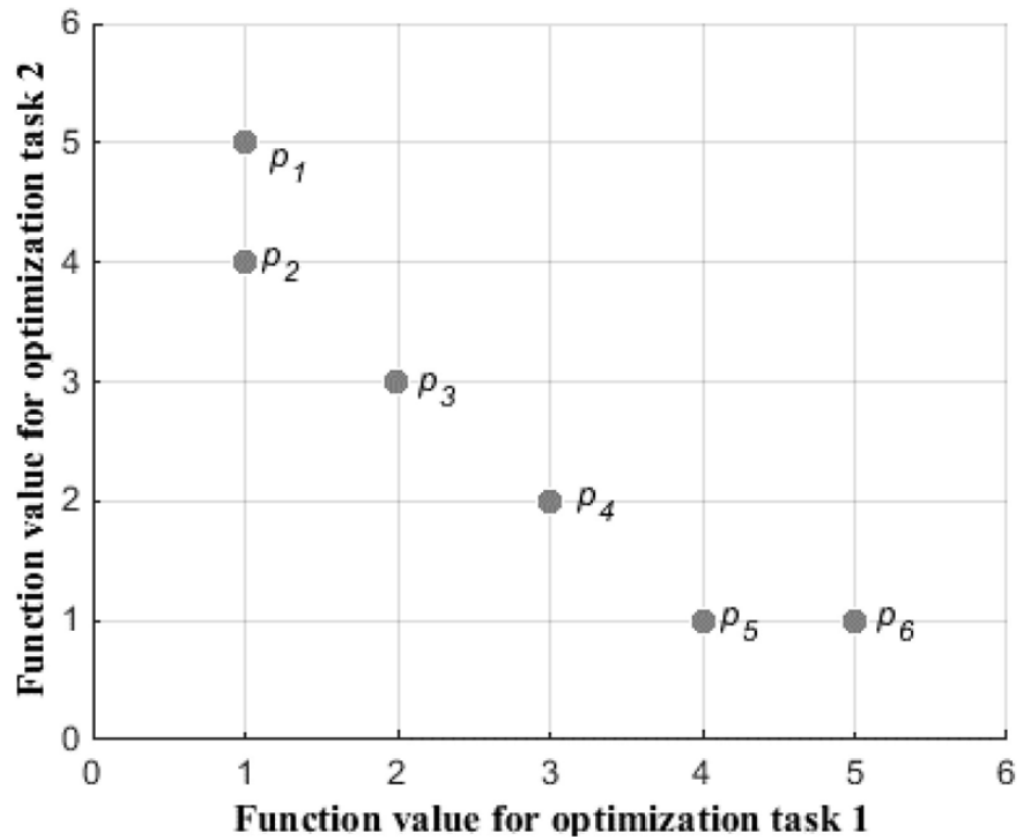
# MFEA - prikaz

- Svaka jedinka predstavlja rješenje za sve probleme
- Koristi se jedinstveni prikaz koji se dekodira za pojedino rješenje
- Koristi se prikaz realnim brojevima iz intervala  $[0,1]$
- Broj varijabli je jednak maksimalnom broju varijabli za bilo koji od problema koji se rješavaju

# MFEA određivanje dobrote

- Za svaki problem može se izračunati dobrotu jedinke
- Svakoj jedinki dodjeljuje se rang za pojedini problem, koji označava koliko je dobra u trenutnoj populaciji
- Skalarna dobrotu -> najbolji rang jedinke na svim problemima
- Faktor vještine -> označava indeks problema na kojem je jedinka najbolja

# Posjeća li na višekriterijsku optimizaciju?



# MFEA - križanje

- Određeni preduvjeti moraju biti ispunjeni
- Princip iz kulturološke evolucije: jedinke su sklonije odabiru partnera iz svoje kulturološke okoline
- Kako to iskoristiti u MFEA

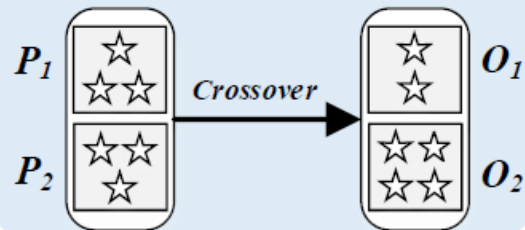
# MFEA - križanje

- Križamo jedinke koje imaju isti faktor vještine (najbolje su za jedan konkretan problem)
- Jedinke se uvijek križaju ako imaju isti faktor vještine ili uz neku vjerojatnost križanja
- U suprotnom provodit se mutacija nad jednom od jedinki roditelja

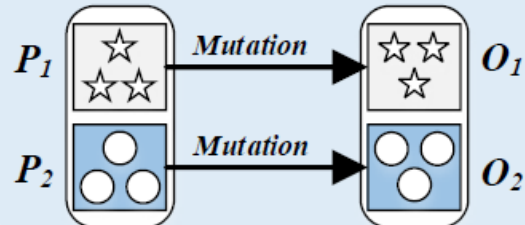


# Genetski operatori

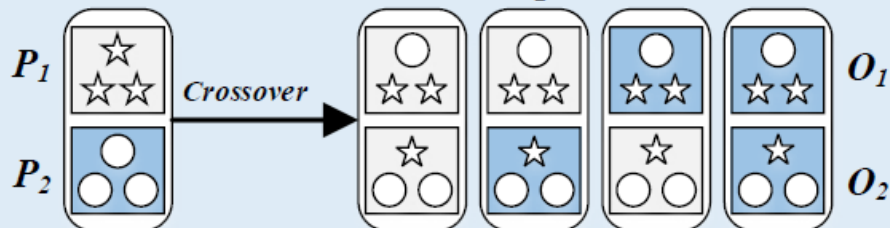
Case 1:  $\tau_1 = \tau_2$



Case 2:  $\tau_1 \neq \tau_2$  and  $rand \geq rmp$



Case 3:  $\tau_1 \neq \tau_2$  and  $rand < rmp$



## MFEA – selektivna evaluacija

- Evaluacija jedinki za svaki problem bilo bi preskupo
- Intuicija: malo je vjerojatno da će jedinka biti dobra za sve probleme, cilj ju je evaluirati samo nad onim problemima za koje bi trebala raditi dobro

# MFEA – selektivna evaluacija

- Ako je nov ajedinka nastala križanjem:
  - 50% vjerojatnosti da se evaluira na problemu prvog roditelja
  - 50% vjerojatnosti da se evaluira na problemu drugog roditelja
- Ako je nastala mutacijom, evaluira se na problemu od jedinke od koje je nastala
- Za sve ostale dobrota se postavlja na beskonačno

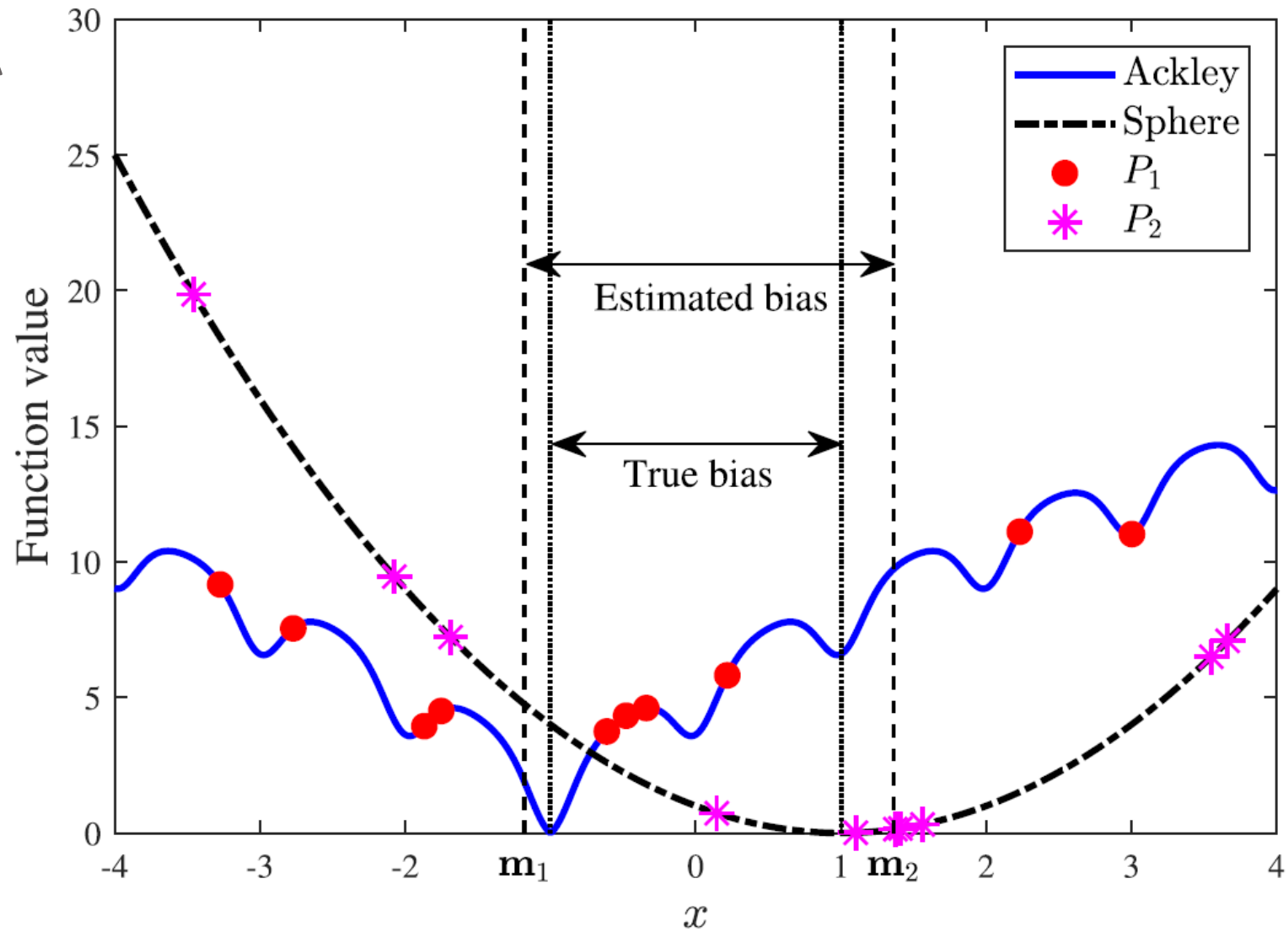
# Multitasking genetic algorithm

- Problemi s prethodnim algoritmom:
  - Ako su funkcije cilja dosta različite, onda algoritam neće imati veliku prednost nad time da svaki problem optimiramo zasebno
  - Kako u takvim slučajevima prenositi korisno znanje među jedinkama

# Multitasking genetic algorithm

- Ideja: maknuti pristranost iz rješenja tako da su optimumi blizu

## MTEA



# Multitasking genetic algorithm

- Koristimo zasebnu populaciju za svaki problem
- Optimiramo jednu populaciju, nakon toga prenesemo dio jedinki u drugu populaciju, pa optimiramo
- Ponavljamo do kriterija zaustavljanja

# Kako eliminirati pristranost?

- Izračunati srednju vrijednost na temelju najboljih jedinki

$$\mathbf{m}_1 = \frac{1}{n_t} \sum_{n=1}^{n_t} \mathbf{x}_{1,n} \quad | \quad \mathbf{m}_2 = \frac{1}{n_t} \sum_{n=1}^{n_t} \mathbf{x}_{2,n}.$$

- Konstruiramo pomoćnu populaciju

$$P_t = \{\mathbf{x}'_{1,1}, \dots, \mathbf{x}'_{1,n_t}, \mathbf{x}_{1,1}, \dots, \mathbf{x}_{1,N-n_t}\}$$

- Eliminiramo pristranost pomoću izraza

$$\mathbf{x}'_{1,n}(i) = \mathbf{x}_{2,n}(I(i)) - \mathbf{m}_2(I(i)) + \mathbf{m}_1(i)$$

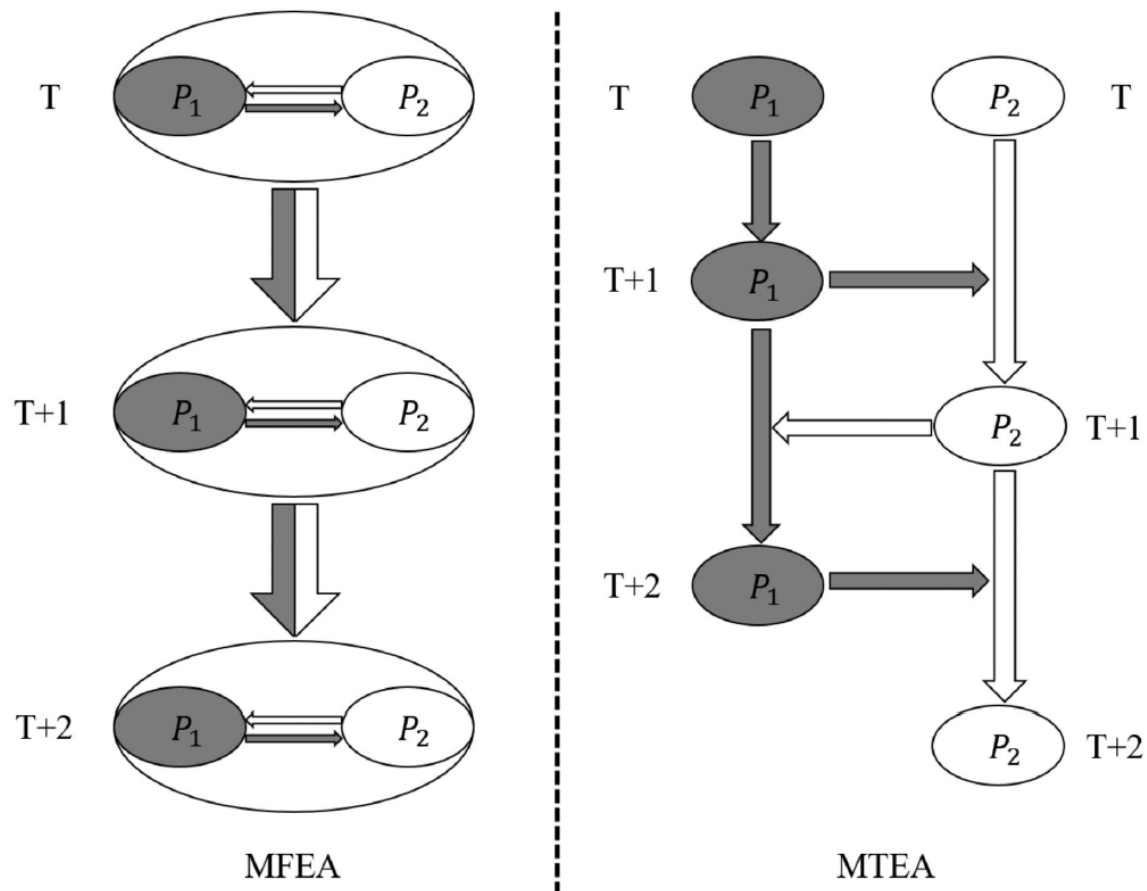
- $I(i)$  – specificira koji gen iz druge jedinke uzimamo



# Genetski operatori

- Križanje i mutacija primjenjuju se samo na jedinkama unutar iste populacije
- Forsira se da sve prenesene jedinke sudjeluju u križanju
- Na kraju se odabire najboljih  $N$  jedinki i one se prebacuju u iduću populaciju

# Razlika između MFEA i MTGA



# Surogatni modeli

- Neki problemi su skupi za evaluaciju
- Može značajno usporiti evolucijski algoritam
- Kako riješiti takav problem?

# Surogatni modeli

- Ideja: jedinke evaluirati nad jednostavnijim problemom koji može dati dobru procjenu pravog problema
- Kako:
  - Aproksimacija funkcije nekim modelom kojeg je jednostavnije evaluirati:
    - Polinomijalna regresija
    - Neuronske mreže
    - SVM

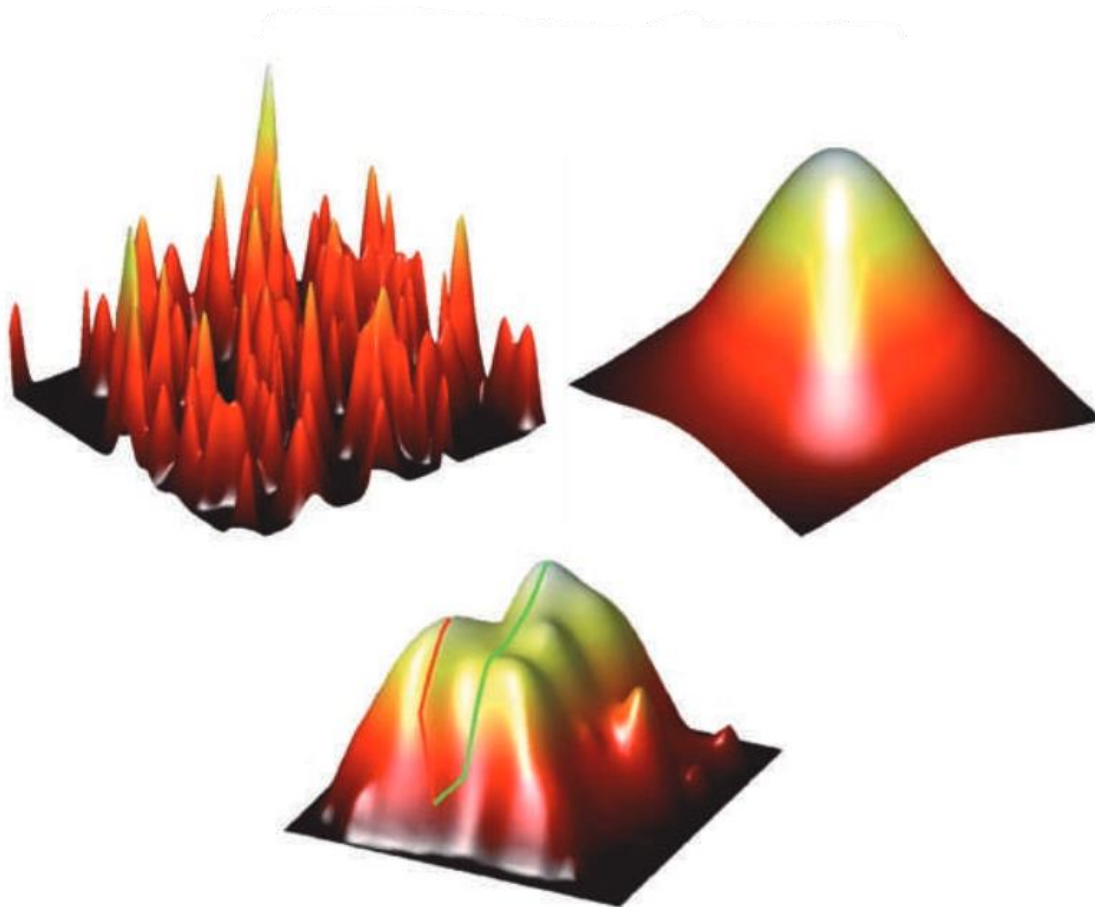
# Surogatni modeli

- Često korišteni u GP-u pri izradi hiperheuristika
- Koristiti manje probleme kako bi se procijenilo koliko dobro pronađena heuristika radi

# Fitness landscape analysis

- Problem: ne znamo kako nam izgleda funkcija cilja
- Ne znamo gdje je njezin minimum
- Ne znamo njena svojstva:
  - Glatka funkcija?
  - Mnogo minimuma?
  - Udaljenost između lokalnih minimuma?
  - ...

# Fitness landscape



# Fitness landscape analysis

- Ideja – izvući određene karakteristike o problemu koji se rješava
- Npr. Kreiramo neka rješenja i radimo slučajne šetnje
  - Primjena lokalnog operatora koji generira susjedno rješenje
  - Na temelju dobivenih rješenja računamo karakteristike o problemu
- Problem definicije susjedstva



# Fitness landscape analysis

- Karakteristike:
  - Modalitet – broj ili učestalost lokalnih optimuma
  - Hrapavost – učestalost promjene nagiba funkcije
  - Korelacija udaljenosti i dobrote
  - Neutralnost
  - ...