GA - Napredni koncepti

-> Problem prijevremene konvergencije

-> Rješenja

-> Ponovna inicijalizacija

↳ pokrećemo algoritam s reinicijaliziranom:
- cijelom populacijom
- nasumičnim dijelom populacije
- brojem najboljih jedinki

↳ dobro: uvodimo raznolikost

↳ loše: krećemo praktički ispočetka

-> Micanje duplikata

↳ pri stvaranju novih jedinki pazimo da ne uvodimo duplikate

-> Selekcija i križanje

↳ problem odabira samo na temelju dobrote

↳ strategije

- Nitching -> ideja da odabiremo jedinke koje se međusobno sličnije
  -> pronalaženje više optimuma
  -> dobro za multimodalne probleme
- Species based -> ideja da odabiramo jedinke udaljene i različite u prostoru
  -> bolje pretraživanje

-> Nitching

-> Dijeljenje dobrote

-> jedinke se dijele u niše prema sličnosti/udaljenosti

$$\rightarrow f_i' = \frac{f_i}{m_i} \rightarrow m_i = \sum_{j=1}^{N} s(d_{ij})$$

$$\downarrow$$

$$s(d_{ij}) = \begin{cases} \left(\frac{d_{ij}}{\sigma_{share}}\right)^{\alpha}, & d_{ij} < \sigma_{sh} \\ 0, & \text{inače} \end{cases}$$

$$\Downarrow$$

$$f_i' = \frac{f_i}{\sum_{j=1}^{N} \left(\frac{d_{ij}}{\sigma_{sh}}\right)^{\alpha} \{d_{ij} < \sigma_{sh}\}}$$

σ_{sh} -> radijus koji definira niše

α -> parametar

-> Čišćenje

-> umjesto raspodijele dobrote, u niši se dobrota

x najgorih jedinki postavlja na $-\infty$ čime se one efektivno odbacuju

→ grupiranje

→ način odabira koje jedinke će djecu zamijeniti u populaciji na temelju sličnosti

→ STANDARDNO

→ M djece se uspoređuje s CF jedinki roditelja

→ svako dijete mijenja najsličniju jedinku

→ DETERMINISTIČKI

→ turniri roditelja i djece → dijete zamijeni roditelja kojem je najsličnije, ako je bolje od njega

→ PROBABILISTIČKI

→ turniri roditelja i djece → dijete zamijeni roditelja kojem je najsličnije, s nekom vjerojatnošću

↓

$$P_i = \frac{f_i}{f_i + f_j}, \quad i \to \text{dijete}, \; j \to \text{roditelj}$$

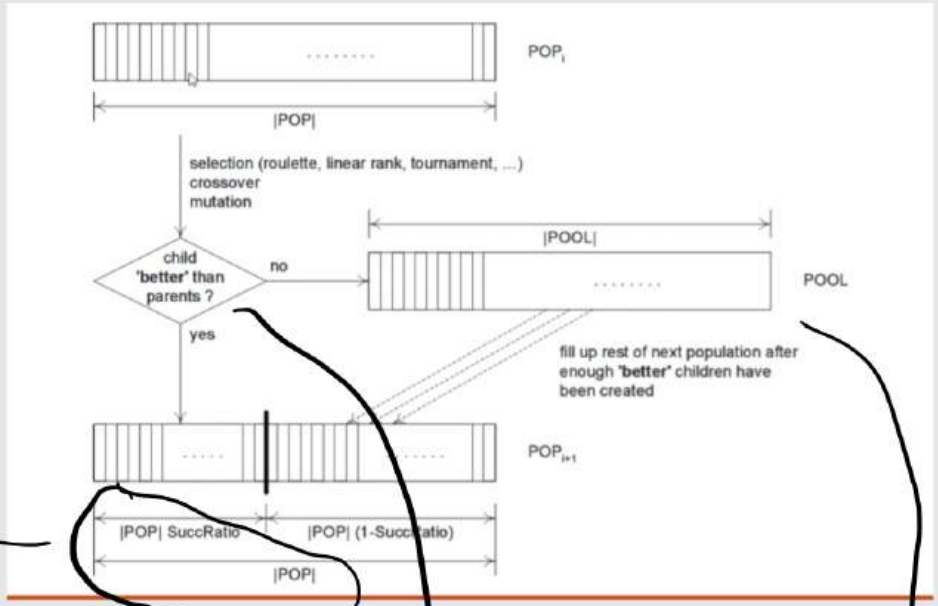
→ OGRANIČENA TURNIRSKA SELEKCIJA

→ M roditelja, CF jedinki

→ dijete mijenja najsličniju jedinku ako je bolje

-> OFFSPRING SELECTION

-> osigurava uspješne potomke u novoj populaciji



faktor uspješnosti:

$$f_u = \frac{|POP|_{succ}}{|POP|}$$

dijete je uspješno ako je u nekom intervalu između dobrota roditelja

adaptivan

detekciju prerane konvergencije ako se prerano nađemo

Otočni genetski algoritam

→ inicijalizacija nasumično ili prema pokrivenosti

→ otoka i općenito kopno

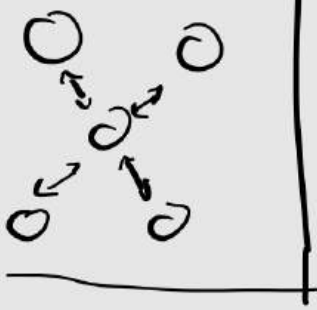
→ migracije između otoka ili s kopnom (samo u jednom smjeru?)
(oba smjera)

→ dobar zamijeni lošijeg

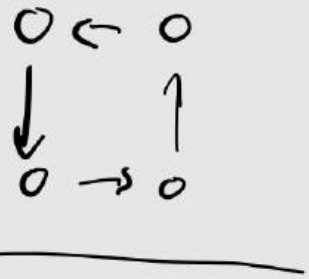
→ dobar zamijeni slučajnog

→ razne topologije

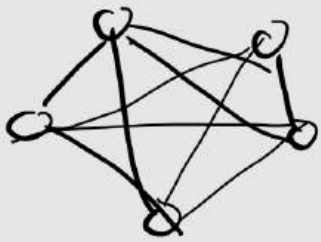
Mainland-island



topologija prstena



+	−
→ raznolikost	→ dodatni parametri
→ paralelizaciju (svaki otok zasebno)	→ trošak komunikacije
→ robusnost	→ sinkronizacije



potpuna topologija

→ adaptivne veličine podpopulacija

→ kada dođe do stagnacije spajamo više populacija u jednu

Kooperativna koevolucija

→ razdvajanje problema na komponente (vrste) koje se nezavisno optimiziraju

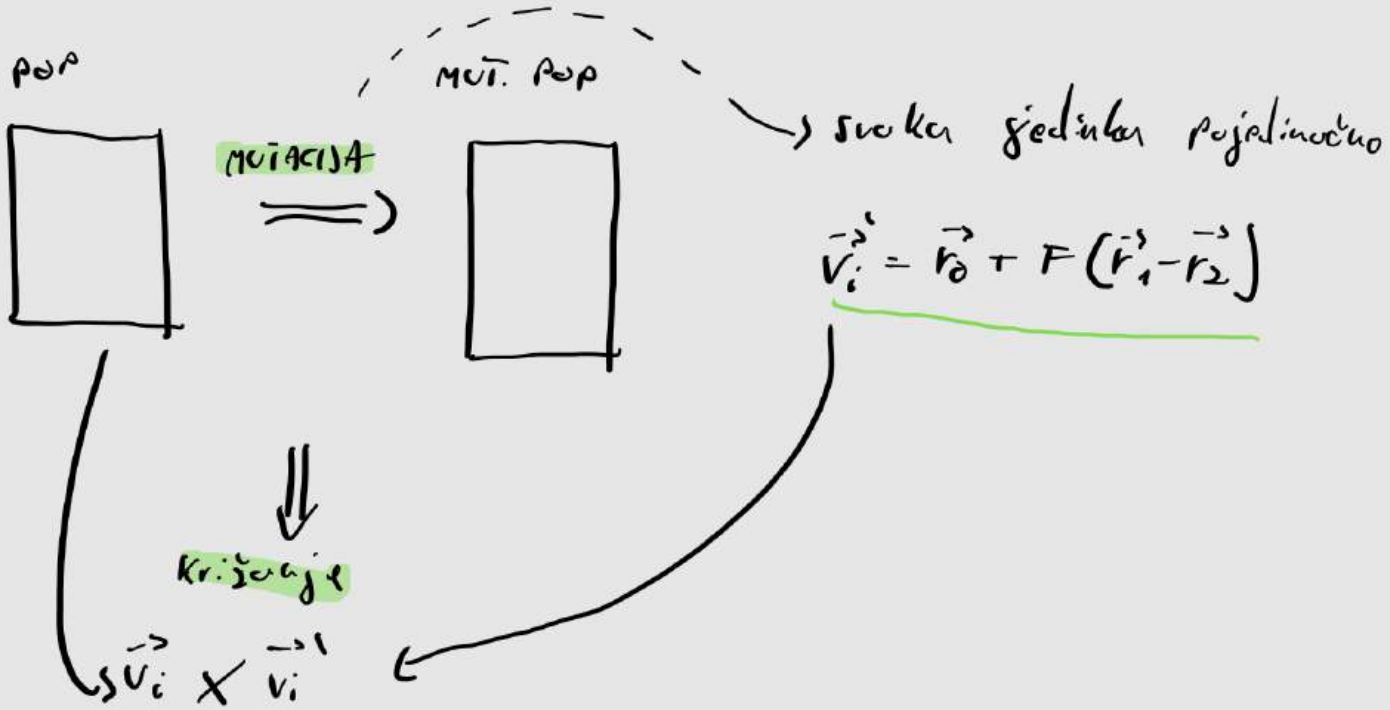
Diferencijska evolucija

DE

-> radi na principu kombiniranja vektora

=> komp. vekt. iz odredenih intervala

$$x_i \in [b_{L,i}, b_{U,i}] \Rightarrow x_i^j = \text{rand}() * (b_{U,i} - b_{L,i})$$



↳ eksponencijalno -> blokovi -> iz $\vec{v}'$ sve dok je $p < C_r$

uniformno -> svaka komponenta zasebno s vjer. C_r

⇓

rezultat: probni vektor $\vec{v}_{ip}$

nova pop

=>

$\vec{v}_{ip}$ ako je bolji il' jednak od $\vec{v}_i$,

$\vec{v}_i$ inače

Odabir $\vec{r_0}$ => random -> slučajno iz populacije

best -> najbolji iz populacije

target-to-best -> vektor translatiran prema najboljem

-> mogućе više lin. komb. u mutaciji

=> zapis: DE/bazni/broj_lin_komb/križanje

-> u/r. DE/rand/1/exp

DE/best/2/bin

Ls uniformno

...

Roj čestica

PSO

-> N čestica koje se kreću nekom brzinom -> čestica: $(\vec{x}, \vec{v})$

položaj, brzina

-> na smjer djeći faktori:

-> individualni -> najbolje pronađeno rj. te čestice

-> socijalni -> globalno najb. pronađeno rj.

kognitivna, socijalna

$$\vec{v}_i(t-1) = w\,\vec{v}_i(t) + \varphi_c C_i \cdot \left(\vec{P}_i - \vec{x}_i(t-1)\right) + \varphi_g C_g \left(\vec{P}_g - \vec{x}_i(t-1)\right)$$

faktor inercije

↑ istraživanje

↓ eksploatacija

sl. brojevi iz $[0,1]$

parametri - stope učenja

⇓

ovisnost o vremenu

-> dva parametra: w_{min}, w_{max}

$$w(t) = \begin{cases} \frac{t}{T}(w_{min} - w_{max}) + w_{max}, & t \leq T \\ w_{min}, & t > T \end{cases}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t)$$

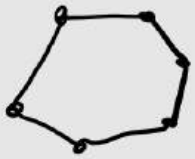
Mogoče definirati sosjedstvo za socijalno komponento

-> GLOBALNO

-> LOKALNO

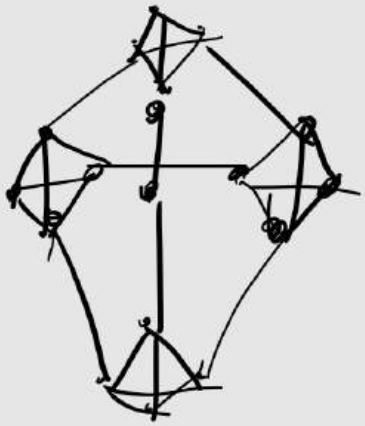
-> VON NEUMAN

-> PRSTEN

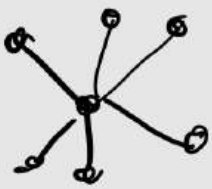


-> SLUČAJNO

-> GRUPIRANJE



-> ZVIJEZDA



Mravlji algoritam - ACO

-> mravi -> stohastički agenti koji grade rješenje
-> tragovi feromona -> težine u grafu

-> dodatna heuristika -> npr. $\frac{1}{d_{ij}}$ -> duljina brida

-> Kretanje mrava

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum_{i}\tau_{ij}^{\alpha}\,\eta_{ij}^{\beta}} & j \in N_i^k \\ 0 & j \notin N_i^k \end{cases}$$

(τ: trag feromona; α, β: parametri; η: heuristička informacija)

-> isparavanje feromona $\quad \tau_{ij} = \tau \cdot (1-\rho), \quad \rho \in [0,1]$

-> ažuriranje feromona

$$\tau_{ij} = \tau_{ij} + \Delta\tau^{k}, \quad \Delta\tau^{k} = \frac{1}{L_k}$$

(L_k: ukupna prometena duljina)

-> Online -> dok mrav prolazi grafom

Odgođeno online -> kada mrav pronađe rješenje

Offline -> kada svi mravi pronađu rješenje

-> temeljeno na: - kvaliteti -> najb. rj. ažurira
- rangu -> najboljih k ažurira

-> elitističko -> glob. najb. ažurira
-> ažuriranje najboljim

⇓ OFFLINE

→ STANDARDNO AŽURIRANJE

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta \tau_{ij}^{k}$$ / m ... broj mrava

→ ELITISTIČKO AŽURIRANJE

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{m} \Delta \tau_{ij}^{k} + e\, \tau_{ij}^{best}$$

→ TEMELJENO NA RANGU → mravi sortirani po dobroti

$$\tau_{ij} = \tau_{ij} + \sum_{k=1}^{w-1} (w-k)\, \Delta \tau_{ij}^{k} + w\, \Delta \tau_{ij}^{best}$$

⇓

što je bolji imaju veći utjecaj

Inicijalno postavljanje

→ $\tau_0 = \frac{m}{C}$ → npr. najbolje dat određenu unaprijed jednu, algoritmom

Imunološki algoritmi

Antitijelo -> rješenje

Antigen -> problem

Afinitet -> dobrota

-> Algoritmi: -> SIA
-> CLONALG

-> SIA - jednostavni imunološki algoritam
-> početna populacija veličine N
-> -> evaluacija populacije
-> klonirange -> svako antitijelo dup puta $|P_{clo}| = n \cdot dup$
$\Downarrow$ hipermutacija
selekcija
$|P_{hyp}| = |P_{clo}| \Leftarrow$ odabir N $\leftarrow P \cup P_{hyp}$
rj. po afinitetu
$\Downarrow$
elitizam osiguran

-> CLONALG

-> klonirange -> selekcija n jedinki za klonirange
↳ broj jedinki svake proporcionalan afinitetu

$$N_c = \sum_{i=1}^{n} \left\lfloor \frac{\beta \cdot n}{i} \right\rfloor$$ → parametar
$\Downarrow$
veličina klonirane populacije

-> hipermutacija

-> intenzitet ovisi o afinitetu

izvorno: $p = e^{-\beta f}$ alternativno: $p = \frac{1}{\beta} e^{-f}$

(-> afinitet)

(-> parametar)

-> nova populacija

-> N najboljih iz klonirane populacije

-> briše se d najgorih i miješaju se slučajnima

-> nema elitizma

-> operator starenja -> briše stare jedinke iz populacije

STATIČKO -> cutoff vrijeme života (prag)

STOHASTIČKO -> slučajna smrt

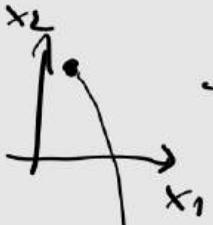
-> veća vjerojatnost za starije jedinke

Višekriterijska optimizacija

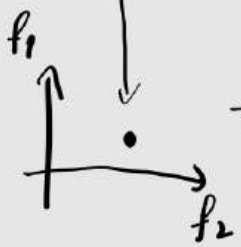
→ optimizacija više kriterija s određenim ograničenjima

- → feasable solution
- → unfeasable solution

2 PROSTORA

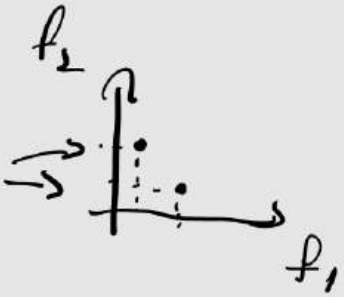


→ Decision space - prostor rješenja



→ Objective space → prostor kriterija

→ problem više jednako dobrih rješenja



METODA 1 → težinska kombinacija

↳ $f(x) = \lambda_1 f_1(x) + \lambda_2 f_2(x) + \dots + \lambda_n f_n(x)$

↓

težine

+	−
višekrit. problem sada jednokrit.	određivanje težina samo 1 rješenje na izlazu

} generalno ne radi dobro

Dominacija (minimizacija)

x_1 dominira x_2 ako $f_i(x_1) \leq f_i(x_2)$, $\forall i$

$f_j(x_1) < f_j(x_2)$ za barem jedan j

→ PARETO OPTIMALAN SKUP

→ skup rješenja koja ne dominiraju jedno nad drugim

→ PARETO FRONTA

→ pareto. opt. skup u prostoru kriterija

→ cilj: pronaći što bolju pareto frontu

→ Rang → koliko rj. dominira neko rj.

→ Nedominirano sortiranje → podjela rj. u fronte

↓

→ samo rj. iz ranijih fronti dominiraju one u trenutnoj

algoritam: - dodijeli svim rj. rang

→ - sva rj. s rangom 0 u fronti "i"

- svim rj. koja dominiraju rj. iz fronte i smanji rang za 1

$\mathcal{O}(MN^2)$

→ 2 CILJA → KONVERGENCIJA → bliska opt. fronti
→ RAZNOLIKOST → pokrivenost fronte

METODA 2 → NSGA → nema elitizam — broj jed. ↑ N

→ najveća dobrota jedinki u 1. fronti (npr. N)
→ svaka sljedeća fronta manja dobrota

→ dobrota unutar fronte

$$L_2\ d_{ij} = \sqrt{\sum_{k=1}^{M} \left(\frac{x_k^i - x_k^j}{x_k^{max} - x_k^{min}} \right)^2}$$

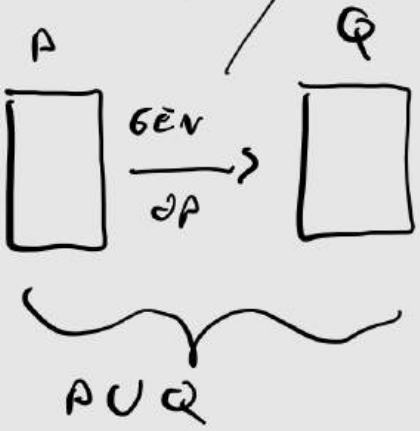
→ u prostoru rješenja → luže

$$f_i' = \frac{f_i}{nc_i} \rightarrow nc_i = \sum_{j=1}^{N} Sh(d_{ij})$$

$$Sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}} \right)^{\alpha}, & \text{ako } d \le \sigma_{share} \\ 0, & \text{inače} \end{cases}$$

$$f_i' = \frac{f_i}{\sum_{j=1}^{N} 1 - \left(\frac{d_{ij}}{\sigma_{sh}} \right)^{\alpha} \left\{ d_{ij} < \sigma_{sh} \right\}}$$

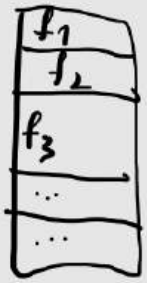
NSGA II → turnirska selekcija



P ∪ Q

NEDOMINIRANO SORTIRANJE

P ∪ Q sort



P1

→ fronte se prenose cijele dok stane

→ iz prve koja ne stane uzimaju se po crowding distance

→ crowding distance

$$\rightarrow d_j = d_j + \frac{f_m^{(j+1)} - f_m^{(j-1)}}{f_m^{max} - f_m^{min}}$$

→ za rubne $d = \infty$

→ inače dijelom

NSGA III

- postavlja referentne točke u prostoru, povlači kroz njih pravce te računa udaljenost od pravaca

PSEUDOKOD

DE/rand/1/bin

```
Inicijaliziraj početnu populaciju
Ponavljaj do kriterija zaustavljanja
    Za svaku jedinku i
        Nasumično odaberi tri različite jedinke r0, r1, r2
        mutant = r0 + F (r1-r2)
        probni=binarno_krizaj(mutant,i)
        Odaberi(probni, i)
Vrati najbolje rješenje u skupu pronađenih rješenja
```

Diferencijska Evolucija

Pseudokod algortma optimizacijom čestica

```
Inicijalizacija čitavog jata (pozicije i brzine)
Ponavljaj do kriterija zaustavljanja
    Evaluiraj sve jedinke
    Za svaku jedinku i
        v_i(t) = w × v_i(t − 1) + ρ_1 C_1 × (p_i − x_i(t − 1)) + ρ_2 C_2 × (p_g − x_i(t − 1))
        x_i(t) = x_i(t + 1) + v_i(t)
    Ažuriraj novo najbolje individualno rješenje
    Ažuriraj novo najbolje globalno rješenje
Vrati najbolje rješenje u skupu pronađenih rješenja
```

Algoritam optimizacije kolonijom mrava

```
Inicijalizacij feromonskog traga
Ponavljaj do kriterija zaustavljanja
    Izgradi rješenja korištenjem feromonskog traga
    Ažuriraj feromonske tragove
        Isparavanje
        Pojačavanje
Vrati najbolje rješenje u skupu pronađenih rješenja
```

AIS

```
Inicijaliziraj početnu populaciju P
Evaluiraj početnu populaciju P
Ponavljaj do kriterija zaustavljanja
    P_clo = kloniraj(P)
    P_hyp = hipermutacija(Pclo)
    Evaluiraj hipermutiranu populaciju P_hyp
    P=odaberi n jedinki iz unije P i P_hyp
Vrati najbolje antitijelo
```

CLONALG

```
Inicijaliziraj početnu populaciju P
Evaluiraj početnu populaciju P
Ponavljaj do kriterija zaustavljanja
        P_n = odaberi(P)
        P_clo = kloniraj(P)
        P_hyp = hipermutacija(Pclo)
        Evaluiraj hipermutiranu populaciju P_hyp
        P=odaberi n jedinki iz P_hyp
        P_new = stvori d nasumičnih jedinki
        Zamijeni najlošija antitijela iz P s onima iz P_new
Vrati najbolje antitijelo
```

NSGA

```
Inicijaliziraj populaciju P
dok(kriterij zaustavljanja nije zadovoljen)
      T=P
      dok(|T|>0)
            B = nedominirana rješenja iz T
            postavi dobrotu jedinki u B
            odstrani sve jedinke u B iz T
      D - sastoji od N djece dobivene križanjem jedinki iz P
      P=D
```

NSGA-II

```
Inicijaliziraj populaciju P
dok(kriterij zaustavljanja nije zadovoljen)
      Q = stvori novu populaciju provođenjem operatora nad P
      R= Q ∪ P
      F = Nedominirano sortiraj R
      C - nova populacija
      i = 0
      dok (|C|+|Fi|<|P|)
            izračunaj crowding distance za Fi
            C = C ∪ Fi
            i++
      Izračunaj crowding distance za Fi
      Sortiraj Fi po crowded distanceu
      C = C ∪ Fi[1:N-|C|]
      P = C
```