

Genetički algoritmi

Željka Ujević Andrijić



OPTIMIRANJE

***Optimiranje** je matematički postupak kojim se pri projektiranju ili vođenju promatranog sustava ostvaruje najbolji mogući izbor ekonomskih i/ili tehničkih veličina na temelju odabranih kriterija - Tehnička enciklopedija.*

Uspješno optimirati je moguće ako je:

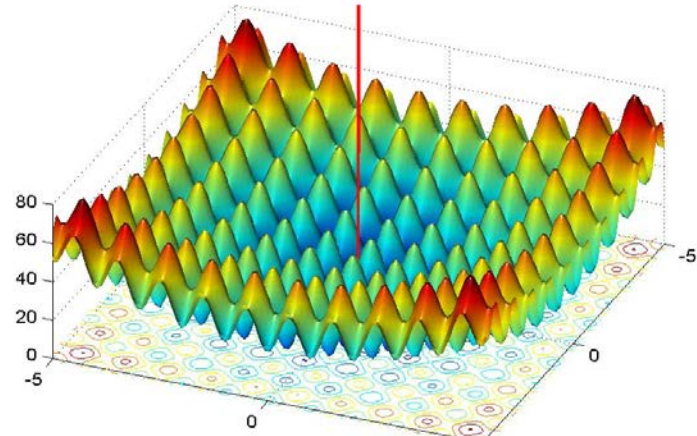
- poznat kriterij optimiranja,
- sukladno kriteriju moguće je procijeniti kvalitetu rješenja,
- postoje varijable (parametri) na koje se može utjecati (mijenjati ih),
- poznata su ograničenja.

• **Optimiranje u kemijskom inženjerstvu** - pri modeliranju, projektiranju i radu industrijskih reaktora, kod planiranja cjelokupne proizvodnje i upravljanja lancima nabave; kod modeliranja fizikalnih procesa za procjenu parametara, kod identifikacije i vođenja procesa.

• Danas je imperativ na **kontinuiranom optimiranju** postrojenja i procesa zbog stalnog napretka u tehnologiji, tržišnim i ekološkim zahtjevima te općenito boljem poslovanju.

OPTIMIRANJE

- Pri optimiranju obično se nastoji pronaći **globalni optimum** - točka gdje funkcija ima najmanju (ako se traži min) ili najveću (ako se traži max) vrijednost. **Lokalni optimum** je točka gdje je vrijednost funkcije manja ili veća nego u okolnim točkama, ali nije najmanja ili najveća u cijelom prostoru pretraživanja.
- Primjer funkcije s puno lokalnih minimuma je tzv. **Rastrigin** funkcija - koristi se za testiranje metoda optimiranja.
 - Navedena funkcija ima jedan globalni min u točki $[0\ 0]$ u x - y ravnini, gdje je vrijednost funkcije 0.
 - Kod ove funkcije prostor pretraživanja minimuma je dvodimenzionalni, inače može biti puno veći te standardni postupci optimiranja (npr. gradijentni postupci) pronalaze samo lokalni optimum.



Podjela algoritama globalnog optimiranja

Deterministički algoritmi

- Zahtijevaju određene pretpostavke za traženje optimuma kao što je npr. neprekinutost funkcije.
- U svakom koraku ovakvog algoritma, postoji najviše jedan način za nastavak algoritma, inače se on završava.
- Ne rabe slučajne brojeve za rad s podacima.
- Prostor pretraživanja mogućih rješenja može se djelotvorno pretražiti koristeći strategiju „podijeli pa vladaj“.

Deterministički algoritmi

Pretraživanje prostora stanja
(engl. *State Space Search*)

Metoda grananja i granica
(engl. *Branch and Bound*)

Metode algebarske geometrije

Algoritmi na temelju vjerojatnosti

Algoritmi slučajne pretrage

Metoda slučajne pretrage
(engl. *Random Search*)

Simulirano kaljenje
(engl. *Simulated Annealing, SA*)

Tabu pretraživanje
(engl. *Tabu Search, TS*)

Stohastično tuneliranje
(engl. *Stochastic Tunneling*)

Izravno Monte Carlo uzorkovanje
(engl. *Direct Monte Carlo sampling*)

Evolucijski algoritmi

Genetički algoritmi
(GA)

Klasifikatorski sustavi s učenjem
(LCS)

Evolucijsko programiranje, EP

Evolucijske strategije, ES

Genetičko programiranje, GP

Inteligencija roja
(engl. *Swarm Intelligence, SI*)

Kolonija mrava
(engl. *Ant Colony Optimisation, ACO*)

Optimizacija rojem čestica
(engl. *Particle Swarm Optimization, PSO*)

Ostale metode

Memetički algoritmi

Harmonijsko pretraživanje
(HS)

Raspršeno pretraživanje
(engl. *Scatter search, SS*)

Algoritmi na temelju vjerojatnosti

- Ako se relacija između mogućih rješenja i njihove **primjerenosti** (engl. “*fitness*”) ne može jasno razumjeti i definirati, ili je dimenzija prostora pretraživanja vrlo velika, obližnja moguća rješenja se mogu bitno razlikovati → najbolje koristiti **algoritme na temelju vjerojatnosti**:
 - generiraju slučajne brojeve u potrazi za globalnim optimumom.
 - relativno ih je jednostavno razumjeti i programirati.
 - ne zahtijevaju preduvjete, primjerice neprekinutost funkcije cilja te se mogu koristiti za rješavanje bilo kojeg zadatka.
 - pokazali su se robusnim pri rješavanju izrazito nelinearnih problema s velikim brojem varijabli.
 - najčešće brzo konvergiraju do globalnog optimuma.
- Mnogi algoritmi na temelju vjerojatnosti (SA, TS, GA, PSO, ACO, HS, memetički algoritmi) često se nazivaju i **meta-heurističke** metode.

Evolucijski algoritmi

Evolucijski algoritmi su postupci optimiranja, učenja i modeliranja koji se temelje na **načelima prirodne evolucije**. Oni su stvoreni na temelju ideje da se biološka načela primijene u rješavanju tehničkih problema koristeći računala.

- Tri grupe evolucijskih algoritama nezavisno su se razvile 1960-tih: **evolucijsko programiranje** (Fogel, Owens i Walsh), **evolucijske strategije** (Rechenberg i Schwefel) te **genetički algoritmi** koje je razvio J. H. Holland.
- Evolucijski algoritmi (EA) najviše se razvijaju tijekom 1990-tih.
- Evolucijski algoritmi su robusni, relativno otporni na šum, sposobni pretraživati velike i višedimenzionalne prostore i relativno se jednostavno primjenjuju.

Evolucijski algoritmi

Dodatne **karakteristike evolucijskih algoritama:**

- **Implicitni paralelizam** - Evolucijski algoritmi pretražuju prostor za sve parametre odjednom. Posebno je povoljno primjenjivati te algoritme na paralelnim računalnim arhitekturama.
- **Stohastička priroda** važna je karakteristika koja ih jasno razlikuje od determinističkih postupaka optimizacije.
- **Određivanje globalnog ekstrema** jedna je od najvažnijih karakteristika evolucijskih algoritama.
- **Jednostavno povezivanje i mogućnost proširivanja** - Evolucijski algoritmi lako se povezuju s postojećim matematičkim modelima, simulacijama i realnim procesima.

Evolucijski algoritmi

- **Mogućnost hibridizacije** – EA se mogu jednostavno povezati s drugim optimizacijskim postupcima u hibridne sustave. Oni mogu ubrzati nalaženje ekstrema, kombinirajući sposobnost nalaženja globalnog ekstrema i djelotvornost lokalnog pretraživanja za preciznije vrijednosti (determinističke metode).
- **Optimiranje originalne funkcije** - radi se s originalnom funkcijom, a ne s njezinom izvedenicom. Mnoge determinističke metode koriste neke od svojstava funkcije, npr. njezinu derivaciju.
- **Algoritam se bez velikih promjena lako primjenjuje na različite probleme** - Mnogi autori smatraju to glavnim razlogom što su EA postali tako popularni. Mogu se koristiti u rješavanju bilo kojeg problema koji se može predstaviti u obliku optimizacije. Zahtjevi poznavanja problema manji su nego kod tradicionalnih postupaka. Npr. funkcija može biti diskontinuirana, optimum se može tijekom vremena mijenjati itd.

Evolucijski algoritmi

- **Ne postoji garancija pronalazanja apsolutno najboljeg rješenja** – EA pronalaze približna rješenja te ne postoji garancija da će biti pronađeno najbolje rješenje. To nije stvarni nedostatak jer ako ostali postupci koji se primjenjuju ne mogu pronaći rješenje, tada je i približno rješenje koje se nađe pomoću EA zadovoljavajuće.
- **Sposobnost algoritma za samoprilagođavanje** – Moguće je primijeniti promjenjive parametre EA i tako stvoriti samoprilagodljiv algoritam.
- **Problemi koji se ne bi trebali rješavati pomoću evolucijskih algoritama:**
 - problemi s malo parametara, koji se mogu riješiti na zadovoljavajući način pomoću determinističkih postupaka,
 - problemi koji se rješavaju pomoću specijalnih postupaka koji daju dobre rezultate i koji se ne mogu jednostavno predstaviti u obliku pogodnom za EA,
 - problemi u kojima podjela na funkcionalne cjeline nije jasna.

Jedinka

- Zadaću koja se rješava evolucijskim algoritmom treba predočiti u obliku prikladnom za rješavanje. To znači da prostor zadaće treba preslikati u formalni prostor EA. Svaku točku mogućih rješenja zadaće treba preslikati u točku prostora rješenja formalnog sustava EA. Svaka tako preslikana točka bit će **jedinka** evolucijskog algoritma.
- Kod evolucijskih algoritama **jedinka populacije** (engl. *individual*) predočena je neovisnim matematičkim objektom, primjerice: realnim brojem, višedimenzionalnim vektorom ili uređenim nizom.
- Predodžba jedinice bitno utječe na djelotvornost algoritma ili čak na mogućnost pronalaženja zadovoljavajućeg rješenja. Budući da je izbor predodžbe djelomično rješenje zadatka, treba mu posvetiti posebnu pozornost.

Populacija i veličina populacije

- U biologiji **populacija** je obično broj jedinki iste vrste živih bića unutar određenog područja u određenom trenutku. Ista vrsta živih bića obično označava mogućnost međusobnog stvaranja potomaka.
- Kod evolucijskih algoritama **populacija** je konačni skup jedinki (matematičkih objekata). Populacija je skup jedinki (roditelja, engl. *parent*) iz kojih mogu nastati nove jedinke (potomci, engl. *offspring*).
- Nova jedinka može nastati iz samo jedne jedinke ili iz para ili više jedinki (engl. *multi-parent recombination*).

Veličina populacije

- Bitno svojstvo populacije je **veličina populacije** jer izravno utječe na trajanje provedbe algoritma. Izabere li se mala populacija, tada je potrebno vrijeme obrade relativno maleno, ali postoji opasnost da algoritam završi na lokalnom ekstremu.
- Populacija s malo jedinki ima malu **genetičku raznolikost** (engl. *diversity*), pa je sužena mogućnost izbora. Što je populacija veća, veći je prostor pretraživanja, te je veća vjerojatnost pronalaženja globalnog ekstrema, ali je algoritam zahtjevniji s obzirom na vrijeme obrade.
- Veličina populacije označava se sa μ . Može biti nepromjenljiva tijekom algoritma ili se može mijenjati. Kod većine EA veličina populacije je stalna i u rasponu $2 \leq \mu \leq 10000$. Iako postoje upute glede izbora veličine populacije npr. od 20 ili 50 do 100 jedinki, ne postoji jednoznačna uputa za određivanje veličine populacije.

Generacija

- U slučaju evolucijskih algoritama, **generacija** predstavlja period između stvaranja roditelja i njihovih potomaka. To znači period između početka jedne populacije i početka populacije koja je stvorena od nje. U EA generacija je jednaka jednoj iteraciji algoritma.
- Prva (nulta) generacija ili **inicijalna populacija** može se stvoriti na dva načina:
 - stvaranjem slučajnih jedinki populacije ili
 - stvaranjem jedinki na osnovi poznavanja problema koji se rješava.

Primjerenost i skaliranje primjerenosti

- U biologiji **primjerenost** je karakteristika jedinke koja opisuje stupanj njezine **prilagođenosti** okolini. Primjerenost živih bića ovisi o okolini te o drugim jedinkama koje se nalaze u toj okolini. Svi ti utjecaji doživljavaju promjene, tako da se primjerenost organizma mijenja tijekom vremena.
- **Primjerenost jedinke** u evolucijskim algoritmima (*fitness function*) je karakteristika jedinke koja opisuje kvalitetu rješenja problema pomoću te jedinke. Primjerenost jedinke obično se izražava pomoću broja i računa se pomoću funkcije primjerenosti.
- Ako se jedinka najpovoljnije primjerenosti smatra optimalnom, a postupak traženja takve jedinke optimiranje, tada je funkcija primjerenosti **kriterij optimiranja**.

Skaliranje primjerenosti

Skaliranje (preslikavanje) primjerenosti pretvara vrijednost funkcije primjerenosti u područje vrijednosti koje je prikladnije za postupak izbora. Može se provesti na više načina:

- **Skaliranje primjerenosti na temelju položaja svrstane jedinke** (engl. *Rank*) svrstava jedinke sukladno vrijednosti funkcije primjerenosti. Rang najbolje jedinke je 1, druge po redu 2 itd.

- **Proporcionalno preslikavanje** - zbroj svih preslikanih vrijednosti primjerenosti jednak jedan, a međusobni omjeri primjerenosti između jedinki su isti kao i prije preslikavanja.

- **Preslikavanje na temelju skupa najboljih jedinki** (engl. *Top*) - svim jedinkama iz skupa najboljih jedinki se pripisuje ista primjerenost. Broj najboljih jedinki treba unaprijed odrediti, a može biti cijeli broj od 1 do veličine populacije ili realni broj od 0 do 1 koji određuje postotak veličine populacije. Svaka od tih jedinki ima istu vjerojatnost stvaranja potomaka. Sve ostale jedinke ne mogu stvarati potomke.

- **Preslikavanje s linearnim pomakom** (engl. *Shift linear*) - primjerenost najbolje jedinke jednaka umnošku konstante pomnožene srednjom vrijednošću primjerenosti populacije → očekivanje koliko će potomaka nastati od roditelja s najpovoljnijom primjerenosti. Očekivanje broja potomaka ostalih jedinki određuje se tako da se zadrži međusoban odnos primjerenosti kakav je bio prije preslikavanja.

Genotip i fenotip. Mutacija

- U EA **genotip** je matematički objekt koji predstavlja **rješenje** problema. Kod biološke evolucije to je kromosom. **Lokus** je položaj gena u kromosomu, dok je **alela** oblik gena i kod EA je to obično numerička vrijednost.
- **Fenotip** je vrijednost koja se dodjeljuje kvaliteti rješenja za određeni genotip, tj. **primjerenost**.
- U biologiji **mutacija** je prirodno ili umjetno izazvana promjena nasljednih karakteristika organizma. Prirodna mutacija je uvijek slučajna. U biološkoj evoluciji mutacija ne mijenja značajno genotip jedinke ili populacije.
- U EA to znači da **mutacija** mora djelovati na **relativno malom broju** jedinki unutar populacije. Postoji mnogo naputaka s obzirom na iznos vjerojatnosti mutacije p_m , npr. $0,001 < p_m < 0,01$.
- U evolucijskim strategijama mutacija je glavni operator algoritma, dok je kod genetičkih algoritama manje važan operator od rekombinacije te se primjenjuje na relativno malenom broju jedinki.

Mutacija

Postupak mutacije - slučajno se mutira jedan gen (gen na lokusu 2)

roditelj	2,3	4,5	6,8	5,4
potomak	2,3	5,8	6,8	5,4

- **Parametri mutacije** određuju način nastanka slučajne promjene komponenti ulaznog argumenta funkcije primjerenosti (slučajne promjene gena jedinke) koje se primjenjuju na roditelje i tako stvaraju potomci. Svaki je potomak, dakle, mutirani roditelj. Mutacija omogućava genetičku raznovrsnost i doprinosi da GA pretraži veće područje rješenja.
- Moguće je određivati **iznos mutacije** u svakoj generaciji kroz parametre:
 - Parametar koji određuje varijancu mutacije prve generacije pomnožen s **rasponom početne populacije** (engl. *Initial range*).
 - Parametar koji određuje iznos za koji se mutacija smanjuje tijekom generacija.

Mutacija

GA primjenjuje mutaciju pomoću **funkcije mutacije**:

a) **Gaussova funkcija** mutacije svakoj komponenti roditeljskog vektora dodaje slučajni broj iz Gaussove razdiobe sa srednjom vrijednosti 0. Obično se taj iznos koji se dodaje smanjuje tijekom svake sljedeće generacije, tj. mutacija je sve manja. Pretpostavlja se da se tijekom provedbe algoritma jedinke sve više približavaju rješenju pa bi preveliki iznos mutacije mogao algoritam “odvući” od područja rješenja. S druge strane, mutacija je potrebna da se algoritam ne “zaglavi” na lokalnom ekstremu.

b) **Jednolika** (engl. *Uniform*) funkcija mutacije - U prvom koraku algoritam izabire elemente vektora jedinke koji će se promijeniti slučajnim izborom tako da svaki element ima određenu vjerojatnost izbora. U drugom koraku svaki od odabranih elemenata se zamjenjuje slučajnim brojem iz jednolikog raspona određenog *rasponom početne populacije*.

Rekombinacija

- **Rekombinacija** je u biološkom smislu razmnožavanje ili reprodukcija. Kod EA **rekombinacija** je postupak stvaranja novih jedinki populacije na temelju jedne ili više postojećih jedinki.
- Matematička funkcija kojim nastaju nove jedinke na temelju postojećih zove se **operator rekombinacije**.
- Operator rekombinacije može se primijeniti na sve jedinke populacije ili samo na stohastički izabrane jedinke. U potonjem slučaju vjerojatnost primjene rekombinacije p_r je u rasponu $0,7 < p_r < 0,95$. Primjerice, neka je vjerojatnost rekombinacije $p_r = 0,9$. To znači da će približno 90% jedinka biti rekombinirano (približno jer je postupak stohastičan, tj. postoji vjerojatnost da će jedinka biti rekombinirana).

Rekombinacija – vrste križanja

- **Križanje u jednoj točki** (engl. *single-point crossover*). Kromosom se podijeli na: glavni dio (engl. *head, fragment*) i ostatak (engl. *tail, reminder*). Nakon toga razmjenjuju se dijelovi kromosoma jedinki između te točke.

Postupak križanja jedinki u jednoj točki

roditelj 1	2,5	3,2	4,8	5,5	6,2	7,3
roditelj 2	1,5	4,6	5,3	5,8	6,7	7,1
potomak 1	2,5	3,2	5,3	5,8	6,7	7,1
potomak 2	1,5	4,6	4,8	5,5	6,2	7,3

- **Križanje u više točaka** (engl. *multi-point crossover*). Bira se više točaka križanja te se zatim razmjenjuju dijelovi kromosoma jedinki između tih točaka.

Rekombinacija - križanje

- **Križanje u jednoj točki s permutacijom** (*shuffle crossover*). Koraci:
 - permutirati gene u jedinki uz pamćenje na kojem su lokusu bili; takva jedinka ima istu primjerenost kao i ona nepermutirana jer je riječ o istoj jedinki čiji su geni privremeno permutirani
 - križati u jednoj točki uobičajenim postupkom,
 - vratiti gene na lokus na kojem su bili prije permutacije.
- **Jednoliko križanje** (engl. *uniform crossover*) - izdvojeno se promatra svaki lokus roditelja i neovisno o drugim lokusima zamjenjuje alela roditelja. Obično se provodi tako da se za svaki lokus neovisno generira slučajni broj u rasponu od 0 do 1 i uspoređuje s vjerojatnošću zamjene koja je jednaka za sve lokuse i iznosi p_x . Ako je slučajno izabrani broj veći od p_x , zamijene se alele između roditelja, a ako je isti/manji, ne zamijene se.
- **Parametarsko jednoliko križanje** - Svaki lokus ima unaprijed određenu vlastitu vjerojatnost razmjene p_{xi} koja je neovisna o vjerojatnosti razmjene alela na ostalim lokusima.

Izbor

- U prirodi preživljavaju one jedinke koje su bolje prilagođene okolišu. Ako se mjera prilagodbe nazove **primjerenost**, tada će veću vjerojatnost preživljavanja imati jedinke bolje primjerenosti te će one imati relativno više potomaka u sljedećoj generaciji.
- **Izbor** je postupak kojim se određuje koje će jedinke preživjeti, tj. imati potomstvo u sljedećoj generaciji.
- Kod EA **izbor** (engl. *selection*) je postupak biranja jedinki za sljedeću iteraciju algoritma sukladno njihovoj primjerenosti.
- Ako se primjerenost izrazi brojem, izbor se temelji na numeričkoj usporedbi primjerenosti jedinka. Za razliku od ostalih operatora EA koji djeluju na genotip, izbor se vrši uvijek na temelju fenotipa jedinke. Hoće li jedinka preživjeti u sljedećoj generaciji, ovisi samo o njezinoj primjerenosti, tj. o tome koliko je rješenje zadatka dobro uz uporabu te jedinke. Izbor jedinki može biti **stohastički** ili **deterministički**.

Izbor

- Kod **stohastičkog izbora operator izbora** je stohastički (engl. *probabilistic selection operator*). Jedinkama se pripisuje stanovita vjerojatnost izbora u sljedeću generaciju. Nema jamstva da će bilo koja jedinka biti izabrana, već postoji samo veća ili manja vjerojatnost da određena jedinka bude izabrana.
- Primjeri stohastičkog izbora su: **proporcionalni izbor**, izbor **svrstavanjem jedinka** i **turnirski izbor**.
- **Proporcionalni izbor** - vjerojatnost izbora jedinke proporcionalna je primjerenosti jedinke podijeljenoj sa zbrojem primjerenosti svih jedinki:

$$p(x_i) = f_p(x_i) / \sum_{i=1}^n f_p(x_i)$$

Izbor

- **Izbor svrstavanjem jedinki** (engl. *rank-based selection*) - jedinke se poredaju od jedinke s najvećom primjerenosti do jedinke s najmanjom primjerenosti. Sukladno redoslijedu, jedinkama se pripisuje vjerojatnost izbora.
- **Turnirski izbor** (engl. *tournament selection*). Pri turnirskom izboru bira se nasumce $n > 1$ jedinki populacije i zatim se najbolja od njih bira za sljedeću populaciju. Postupak se ponavlja sve dok se ne izaberu sve jedinke nove populacije.
- Kod **determinističkog izbora** uvijek se bira n najboljih jedinki za sljedeću generaciju.

Postupak izbora

- Postupkom izbora stvara se **međupopulacija** (engl. *intermediate population*) - skup potencijalnih roditelja, tj. skup jedinka koje mogu sudjelovati u rekombinaciji pri nastanku jedinki nove generacije. Međupopulacija nastaje kopiranjem jedinki populacije, pri čemu neke jedinke mogu proizvesti jednu ili više kopija, a neke ni jednu. Zbog toga je genotipski sastav međupopulacije različit od genotipa populacije iz koje je nastao.
- U jednoj iteraciji postoji: **početna populacija**, **međupopulacija** (nastala postupkom izbora) i **konačna populacija** (nastala postupkom rekombinacije i mutacije).
- Izborom se nastoji međupopulaciju učiniti pogodnijom s gledišta primjerenosti jedinki u odnosu na prethodnu populaciju. Temeljna zadaća postupka izbora je osigurati primjerenijim jedinkama veću vjerojatnost preživljavanja, odnosno omogućiti im stvaranje više potomaka u međupopulaciji.
- Postupak izbora je jedini korak koji uzima u obzir primjerenost jedinke, pa je on odgovoran za usmjeravanje GA prema rješenju zadatka odnosno za konvergenciju GA.

Postupak izbora - koraci

- **Pridjeljivanje vjerojatnosti izbora jedinkama** - Svakoj se jedinki pridjeljuje vjerojatnost izbora (engl. *selection probability*). To je realni broj u rasponu [0,1]. Jedinka s većom vjerojatnosti izbora imat će više potomaka (kopija) u međupopulaciji. Ako je x_i i-ta jedinka populacije, tada je **vjerojatnost izbora te jedinke**:

$$p_e(x_i) = p_{ei} = \frac{f_{pi}}{\sum_{i=1}^{\mu} f_{pi}}$$

Vjerojatnost izbora jedinke jednaka je omjeru primjerenosti te jedinke sa zbrojem primjerenosti svih jedinki.

Ovim korakom pridjeljuje se vjerojatnost izbora, a ne i jamstvo da će se kopija jedinke naći u novoj populaciji.

Postupak izbora - koraci

- **Uzorkovanje** je postupak kojim se na temelju **vjerojatnosti izbora jedinke** u međupopulaciju određuje stvaran broj kopija jedinki koje će biti zastupljene u međupopulaciji. To je postupak kojim se realni broj pretvara u cijeli broj (broj kopija jedinke).
- **Izbor jedinki za međupopulaciju** (engl. *selection scheme*) ne treba brkati s postupkom izbora jedinki koje se doista rekombiniraju. **Izborom jedinki za međupopulaciju** stvara se međupopulacija koja sadrži **potencijalne** roditelje. Najprije se mora postupkom izbora stvoriti cjelokupna međupopulacija, a tek zatim može započeti izbor jedinki koje se doista rekombiniraju. Izborom jedinki za međupopulaciju važno je stvoriti povoljnu međupopulaciju koja sadrži dobar genetički materijal.
- Roditelji koji će doista stvoriti jedinke nove generacije, biraju se iz jedinki međupopulacije tijekom postupka **rekombinacije**.

Postupak izbora - koraci

- **Očekivani broj jedinki u međupopulaciji** (potomaka) umnožak je vjerojatnosti izbora i -te jedinke i ukupnog broja jedinki međupopulacije.

$$N_i = \mu_m \cdot p_{ei}; \quad i = 1, 2, \dots, \mu_m$$

Očekivani broj potomaka ne mora biti stvarni broj potomaka koji će se pojaviti u međupopulaciji. Prvi razlog je što je N_i realni broj, a zbog postupka uzorkovanja stvarni broj potomaka je cijeli broj. Drugi je razlog stohastičnost uzorkovanja koja u slučaju da je N_i cijeli broj može uzrokovati različit broj kopija u međupopulaciji od N_i .

Srednja vrijednost
primjerenosti populacije:

$$\overline{f_p} = \frac{1}{\mu} \sum_{i=1}^{\mu} f_{pi}$$

f_{pi} - primjerenost svake
jedinke u populaciji

**Vjerojatnost izbora
svake jedinke:**

$$p_{ei} = \frac{f_{pi}}{\sum_{i=1}^{\mu} f_{pi}}$$

Postupak izbora

- **Izborni pritisak** (engl. *selective pressure*) je omjer vjerojatnosti izbora najbolje jedinke i srednje vjerojatnosti izbora svih jedinki (populacije).
- Kad ne bi bilo izbornog pritiska, sve bi jedinke imale jednak broj kopija u međupopulaciji, pa postupak izbora ne bi imao smisla (međupopulacija bi bila ista kao i populacija iz koje je nastala).
- Izborni pritisak i raznolikost populacije su **ključni** razlog uspješnosti genetičkih algoritama.
- S jedne strane poželjan je što veći izborni pritisak, jer će se tada u međupopulaciji naći mnogo kopija jedinke s velikom primjerenošću. Srednja vrijednost primjerenosti međupopulacije u tom će slučaju biti veća od srednje vrijednosti primjerenosti populacije iz koje je nastala.

Postupak izbora

- S druge strane pojava mnogo kopija iste jedinke u međupopulaciji može dovesti do preuranjene konvergencije. Povećanjem izbornog pritiska raste i brzina konvergencije algoritma. To je u načelu povoljna značajka, pod uvjetom da ne dođe do preuranjene konvergencije.
- Ako se u populaciji pojavi jedinka čija je primjerenost mnogo veća od srednje primjerenosti populacije, a istodobno je primjerenost te jedinke mnogo manja od primjerenosti rješenja zadatka, dogodit će se da ta jedinka ima veliku vjerojatnost izbora. Takva jedinka, koja se naziva **super jedinka** ima mnogo više kopija u međupopulaciji od ostalih jedinki. Nakon nekoliko generacija ona će biti više zastupljena od ostalih jedinki, ili će sve jedinke populacije biti jednake toj jedinki.
- Rekombinacijom ne mogu nastati nove jedinke, pa je daljnji napredak moguć samo mutacijom.

Konvergencija evolucijskog algoritma

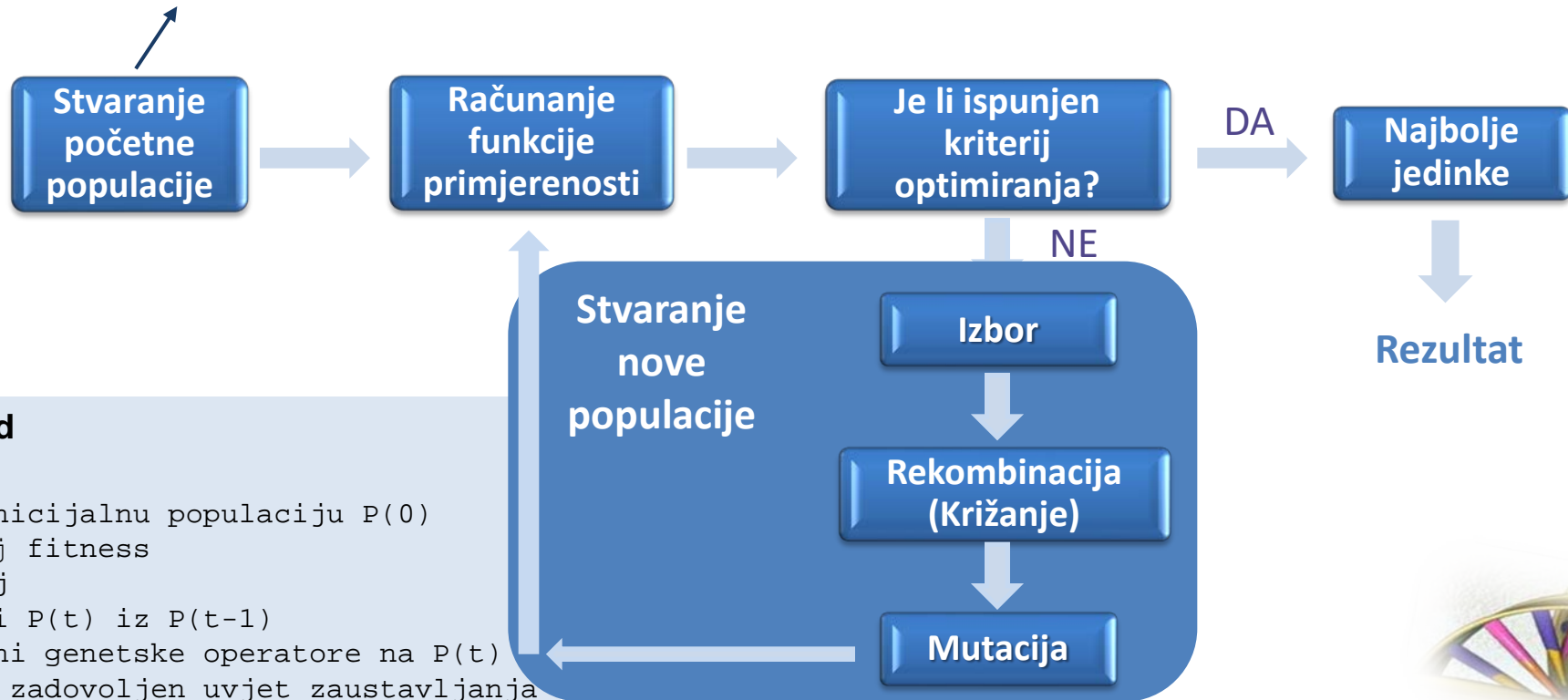
- Za EA se kaže da je konvergirao kad se bitno smanji genetička raznolikost populacije, tj. kad sve jedinke populacije imaju sličan genotip.
- Kod većine EA stanje konvergencije znači da je pronađeno rješenje zadatka i da treba zaustaviti algoritam (ne može se proizvesti jedinka povoljnije primjerenosti od najbolje jedinke postojeće populacije).
- **Preuranjena konvergencija** je pojava bitnog smanjenja genetičke raznolikosti prije nego što je pronađen globalni ekstrem. Ona može dovesti do neoptimalnog rješenja.

Konvergencija evolucijskog algoritma

- U načelu nije moguće znati je li riječ o preuranjenoj ili stvarnoj konvergenciji, jer bi to značilo da nam je poznato gdje se nalazi globalni ekstrem, te u algoritam treba ugraditi **mehanizme za sprečavanje preuranjene konvergencije**. Oni se mogu svrstati u dvije skupine, ovisno o tome održava li se genetička raznolikost pri rekombinaciji ili pri izboru.
- **Održati genetičku raznolikost pri rekombinaciji** moguće je pažljivim biranjem načina izbora jedinki za rekombinaciju. Primjerice, može se uvjetovati da nije moguće križanje jedinki čiji su genotipi slični; mogu se pri izboru jedinki za rekombinaciju spriječiti velike razlike u vjerojatnosti izbora među jedinkama i dr.
- **Održavanje genetičke raznolikosti pri izboru** - pri izboru jedinki koje će prijeći u sljedeću generaciju može se onemogućiti biranje jedinki istog genotipa ili višestruko biranje iste jedinke i sl.

Postupak optimiranja genetičkim algoritmom

Svaka jedinka u populaciji predstavlja skup mogućih rješenja – onoga što se traži minimumom funkcije cilja.



pseudokod

```

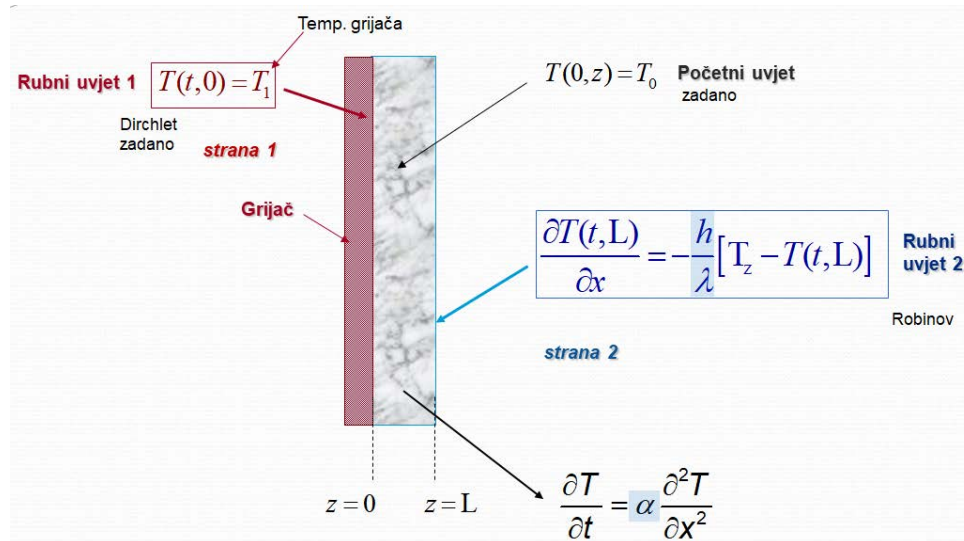
pocetak
stvari inicijalnu populaciju P(0)
izračunaj fitness
ponavljaj
  odaberi P(t) iz P(t-1)
  primjeni genetske operatore na P(t)
dok nije zadovoljen uvjet zaustavljanja
kraj
  
```



Primjer primjene GA – Prijenos topline kroz ploču

Analizira se prijenos topline na laboratorijskoj aparaturi za istraživanje prijenosa topline kroz različite materijale. Ploča je sa jedne strane naslonjena na grijač. Druga strana ploče je u kontaktu sa zrakom iz okoline na sobnoj temperaturi. Temperatura se mjeri na tri mjesta s obzirom na debljinu ploče (na točki granice grijač-ploča, polovici debljine ploče te na vanjskoj strani ploče).

Potrebno je razviti matematički model procesa prijenosa topline te izračunati profile temperatura po položaju u zadanom vremenu i profile temperatura za određeni položaj tijekom vremena i pritom **procijeniti** potrebne **parametre**.



Rubni uvjet na strani 2 → toplina koja se odvede prirodnom konvekcijom zrakom jednaka je toplini koja se dovede do strane 2 ploče vođenjem.

Primjer primjene GA – Prijenos topline kroz ploču

```
...  
expData1 = kamen(:,3)';  
expData2 = kamen(:,4)';
```

```
h = 10*rand(10,1) + 1.0;  
lambda = 1*rand(10,1) + 0.1;
```

→ poč .vr.parametara

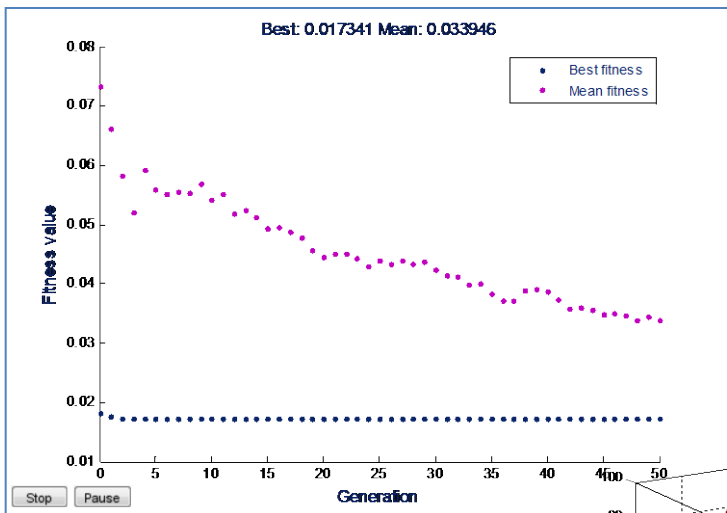
↙ Dio koda s primjenom GA u Matlabu

```
options = gaoptimset;  
options = gaoptimset(options,'PopulationSize',100);  
options = gaoptimset(options,'EliteCount',2);  
options = gaoptimset(options,'CrossoverFraction',0.5);  
options = gaoptimset(options,'Generations',50);  
options = gaoptimset(options,'StallGenLimit',50);  
options = gaoptimset(options,'InitialPopulation',[h lambda]);  
options = gaoptimset(options,'FitnessScalingFcn',@fitscalingprop);  
options = gaoptimset(options,'SelectionFcn',@selectionuniform);  
options = gaoptimset(options,'CrossoverFcn',@crossoverintermediate);  
options = gaoptimset(options,'MutationFcn',@mutationadaptfeasible);  
options = gaoptimset(options,'PlotFcns',@gaplotbestf);  
  
[p, OD] = ga(@(x) func01(x,expData1,expData2),2,[],[],[],[],[1; 0.1],[11; 1.1],[],options)
```

```
theta = parcijalna(p);
```

```
...
```

Primjer primjene GA – Prijenos topline kroz ploču

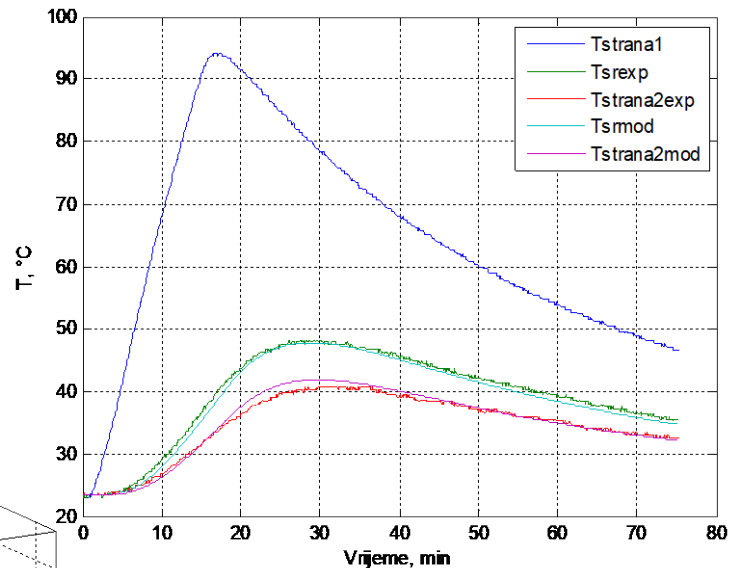
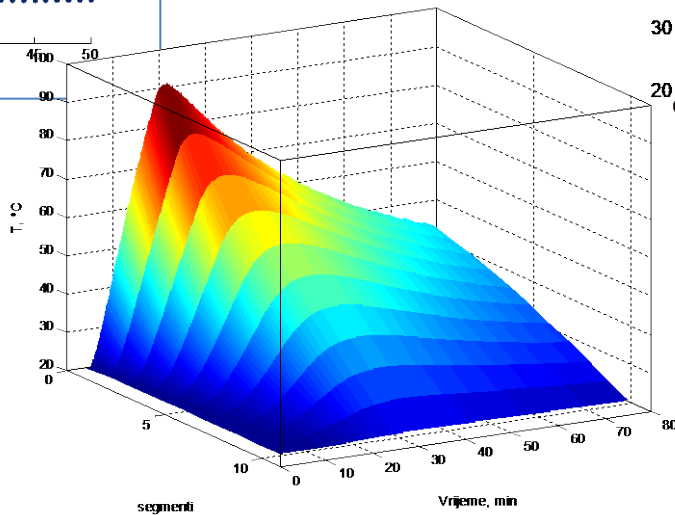


Fitness funkcija

Optimirani parametri:

$$\lambda = 0,105857 \text{ Jmin}^{-1}\text{cm}^{-1}\text{K}^{-1},$$

$$h = 2,33294 \text{ Jmin}^{-1}\text{cm}^{-2}\text{K}^{-1}$$



Profili temperature po vremenu na zadanim debljinama ploče.

Primjer primjene GA - Soft senzor za procjenu sadržaja benzena u reformatu

Primjer podešavanja parametara GA

Veličina populacije (broj jedinki)	15 – 50	50 za OE;
Broj generacija	50 - 80	60 za sve modele
Izbor	Jednoliko slučajno	Jednoliko slučajno
Križanje	<ul style="list-style-type: none">- Križanje u jednoj točki- Križanje u dvije točke- Križanje u svim točkama	Križanje u svim točkama
Mutacija	Jednolika	Jednolika
Skaliranje funkcije cilja	<ul style="list-style-type: none">-Po položaju svrstane jedinke- Proporcionalno	Po položaju svrstane jedinke
Broj elitnih jedinki	1 ili 2	1
Udio križanja	0,5 ili 0,7	0,7

Primjer primjene GA - Soft senzor

15 parametara OE modela
koji se trebaju procijeniti →

```
nb = [x01 x02 x03 x04 x05 ]  
nf = [x06 x07 x08 x09 x10 ]  
nk = [x11 x12 x13 x14 x15 ]
```

```
PopSize = 50;
```

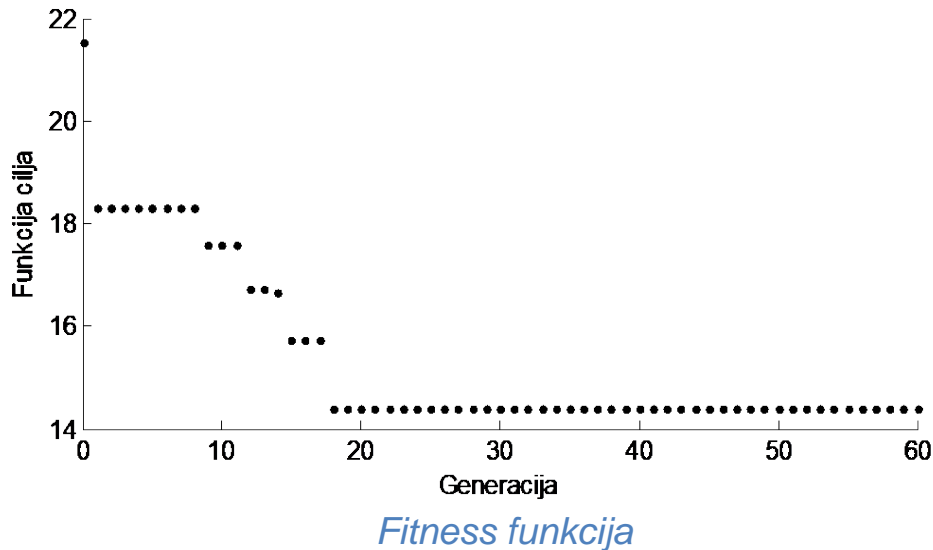
```
InitPop = [x01 x02 x03 x04 x05 x06 x07 x08 x09 x10 x11 x12 x13 x14 x15];
```

```
xmin = [x01min x02min x03min x04min x05min x06min x07min x08min x09min x10min x11min x12min  
x13min x14min x15min];
```

```
xmax = [x01max x02max x03max x04max x05max x06max x07max x08max x09max x10max x11max x12max  
x13max x14max x15max];
```

```
options = gaoptimset('PopulationType','doubleVector');  
options = gaoptimset(options,'PopInitRange',[xmin; xmax]);  
options = gaoptimset(options,'PopulationSize',PopSize);  
options = gaoptimset(options,'EliteCount',1);  
options = gaoptimset(options,'CrossoverFraction',0.8);  
options = gaoptimset(options,'Generations',60);  
options = gaoptimset(options,'StallGenLimit',60);  
options = gaoptimset(options,'StallTimeLimit',Inf);  
options = gaoptimset(options,'InitialPopulation',InitPop);  
options = gaoptimset(options,'FitnessScalingFcn',@fitscalingrank);  
options = gaoptimset(options,'SelectionFcn',@selectionstochunif);  
options = gaoptimset(options,'CrossoverFcn',@crossoverscattered);  
options = gaoptimset(options,'MutationFcn',{@mutationuniform,0.1});  
options = gaoptimset(options,'PlotFcns',{@gaplotbestf,@gaplotrange});  
options = gaoptimset(options,'Vectorized','off');
```

Primjer primjene GA - Soft senzor za procjenu sadržaja benzena u reformatu

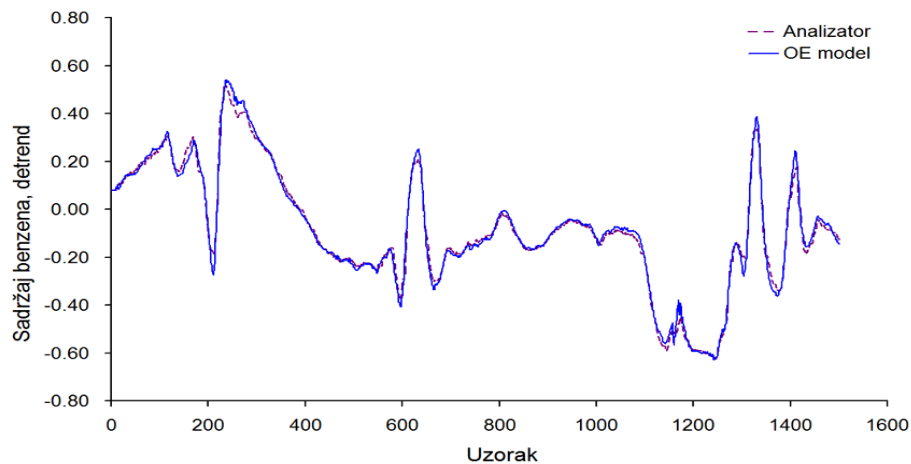


Optimirani parametri OE modela:

$$nb = [6 \ 1 \ 2 \ 3 \ 3]$$

$$nf = [1 \ 1 \ 2 \ 2 \ 1]$$

$$nk = [1 \ 8 \ 7 \ 9 \ 13]$$



Usporedba sadržaja benzena određenog analizatorom i OE modelom

Još 1 primjer u Pythonu: <https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6>