Računarski fakultet

Paralelni genetski algoritmi

Profesor: Student:

dr Jelena S. Vasiljević Marko Nenadović RN 41/14

Sadržaj

[Genetski algoritmi 3](#_Toc483828912)

[Paralelni genetski algortmi 4](#_Toc483828913)

[Podela po strategiji paralelizacije 4](#_Toc483828914)

[Standardni pristup 5](#_Toc483828915)

[Dekompozicijski pristup 5](#_Toc483828916)

[Vrste paralelnih genetskih algoritama 6](#_Toc483828917)

[Distribuirani genetski algoritam 6](#_Toc483828918)

[Masovno paralelni genetski algoritam 7](#_Toc483828919)

[Globalno paralelni genetski algoritam 8](#_Toc483828920)

[Trivijalni paralelni genetski algoritam 8](#_Toc483828921)

[Hijerarhijski paralelni genetski algoritam 9](#_Toc483828922)

[Hibridni paralelni genetski algoritam 9](#_Toc483828923)

[Zaključak 10](#_Toc483828924)

[Literatura 10](#_Toc483828925)

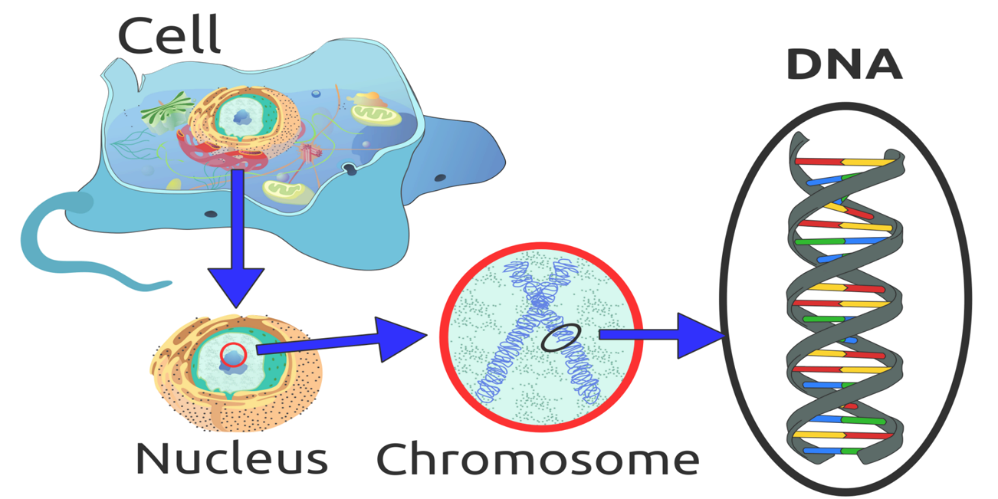
# Genetski algoritmi

Genetski algoritam je heuristički optimizacijski algoritam koji oponaša prirodni evolucijski proces. Po načinu delovanja ubraja se u metode usmerenog pretraživanja prostora rešenja (guided random search techniques) u cilju traženja globalnog optimuma. Ova heuristika (takođe ponekad nazivana metaheuristika) se rutinski koristi da generiše korisna rešenja za optimizaciju i probleme pretrage.

Genetski algoritmi pripadaju većoj klasi evolucionih algoritama (EA) koji generišu rešenja za optimizaciju problema korišćenjem tehnika insprisanih prirodnom evolucijom, kao što su nasleđivanje, mutacija, selekcija i ukrštanje (krosing-over).

Rad genetskih algoritama možemo opisati u osam koraka:

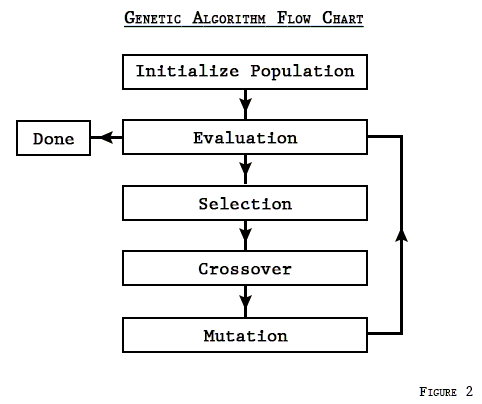
1. Definisanje svih potrebnih parametara problema i GA
2. Formiranje početne populacije
3. Dekodovanje hromozoma — ovaj korak se javlja samo kod binarnih GA
4. Određivanje cena hromozoma (fitnes funkcijom)
5. Odabir selekcije hromozoma koji će opstati za parenje
6. Parenje — obično se iz boljeg dela populacije odabiru roditelji koji će na neki način ukrstiti svoj genetski materijal i dati jednog ili više potomaka koji obično zamene nekog od lošijih hromozoma
7. Mutcije, pri kojima se menja genetski sadržaj hromozoma
8. Ispitivanje konvergencije, da bi se utvrdilo da li ima osnova da se tok algoritma prekine. Ukoliko nije ispunjen uslov konvergencije, vratiti se na korak 3 odnosno 4.



# Paralelni genetski algortmi

Funckije troška, u kompleksnim simulacijama za koje se koriste genetski algoritmi, su sve komplikovanije. Zbog velikog broja izračunavanja, genetski algoritmi su jako procesorski zahtevni i mogu da budu vrlo spori, te kako bi se postiglo ubrzanje oni se paralelizuju koršćenjem procesorskih niti. Za komplikovane funkcije troška ova paralelizacija je dobra, ali za one koje ne zahtevaju toliko vremena, međuprocesorska komunikacija može oduzeti većinu vremena koja bi se inače dobila parelelizacijom. Iz tog razloga se mora pažljivo odabrati šta mora da se odradi na celom skupu hromozoma, a šta je dovoljno odraditi samo na nekoj potpopulaciji.

Prednost paralelizacije, pored ubrzanja, može biti i potreba da potpopulacije evoluiraju nezavisno jedna od druge, kao u prirodi, a da se komunikacija između tih „ostrva“ pojavljuje samo povremeno. Na taj način se sprečava prerana konvergencija, jer se dozvoljava da svako ostrvo pretražuje unutar svoje oblasti sa svojim kombinacijama hromozoma, pa jedna jedinka ne dominira celom populacijom. Nekada se na taj način može naći više rešenja.

U poslednjih dvadesetak godina nastao je veliki broj strategija paralelizacije i oprobanih metoda za ovaj problem. Konkretnu metodu koju primenjujemo za paralelizaciju našeg genetskog algoritma biramo u zavisnosti od arhitekture i prirode problema sa kojim se suočavamo.

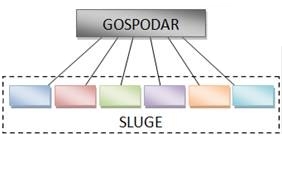
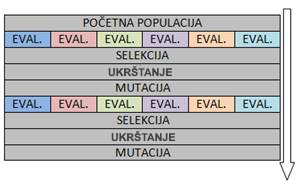
## Podela po strategiji paralelizacije

Paralelni genetski algoritmi se po strategiji paralelizacije dele na:

* *standardni pristup* – paralelizujemo genetske operatore i izračunavanje vrednosti funkcije troška ( cilja )
* *dekompozicijski pristup* – podelimo populaciju na manje delove - potpopulacije i obavljamo celi genetski algoritam nad potpopulacijama

### Standardni pristup

Kod standardnog pristupa paralelizacije genetskih algoritama paralelizuje se samo posao evaluacije. Evaluacija ili vrednovanje je postupak izračunavanja vrednosti funkcije troška (takođe može da se nazove funkcijom cilja). Genetski operatori deluju samo nad jednom, zajedničkom populacijom pa se ovaj model naziva i jednopopulacijski model.

Najlakša varijanta ovog pristupa je master-slave organizacija koji predstavlja paralelnu implementaciju standardnog genetskog algoritma. Kod ovog pristupa jedan procesor ima ulogu „mastera“ (gospodara) i kontroliše komunikaciju, sortiranje i uparivanje, dok funkciju troška šalje “slave“ procesorima (slugama) za paralelno izračunavanje. Mana ovog pristupa je što master često čeka na rezultate i što je algoritam spor koliko i komunikacija sa najsporijim slave čvorom. Neke ekstremne verzije menjaju celokupnu populaciju u svakoj generaciji, dok neki drugi menjaju po jedan ili dva hromozoma. Master-slave organizacije su uglavnom sinhrone, tj. pojavljuje se bottleneck ako neki processor mnogo kasni. U asinhronom slučaju, master kreće u selekciju dok slave čvorovi računaju i vraćaju vrednost prilagođenja. Ovakav pristup je zgodan za turnirsku selekciju, jer ne mora da se čeka odgovor od svih čvorova.

### Dekompozicijski pristup

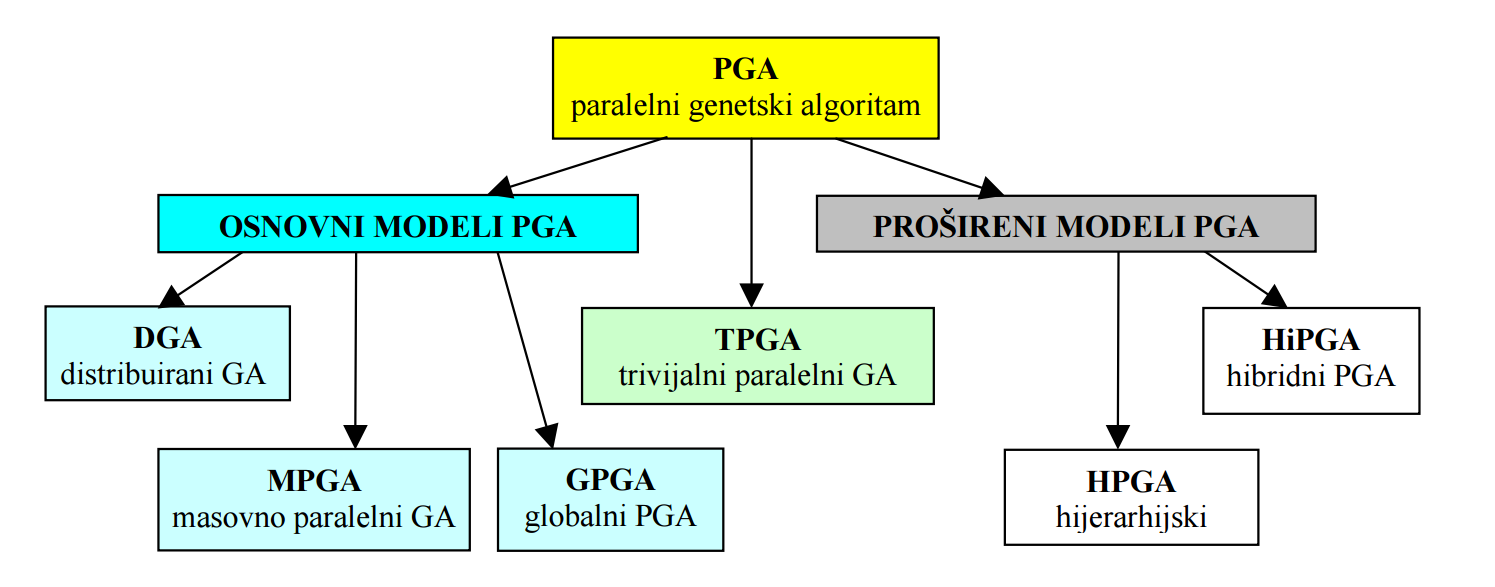
Kod ovog pristupa populacija se deli na ostrva, tj. potpopulacije pa se genetski algoritmi koji koriste ovaj pristup nazivaju multipopulacijskim paralelnim genetskim algoritmima. Svaki procesor ima odvojenu populaciju hromozoma koji se reprodukuju i izračunava se prilagođenje odvojeno od svih ostalih populacija. U slučaju da nema komunikacije između ostrva, ovaj pristup je ekvivalentan obavljanju više poziva genetskog algoritma nad nekoliko manjih populacija. Uobičajeno je da postoji periodična komunikacija između ostrva i na taj način se promena unutar populacije odvija većom brzinom nego bez spoljašnjeg uticaja. To se čini tako što neke jedinke migriraju sa jednog na drugo ostrvo. Veze između ostrva određuje topologija, a koliko jedinki i koje od njih migriraju zavisi od implementacije. Pri migraciji članovi se mogu kopirati ili izbacivati iz populacije. Ovakve implementacije su tipično brže od master-slave implementacija. Ovaj model je uglavnom pogodan za odvijanje na MIMD (multiple instruction, multiple data) računarima.

Podgrupa ovog pristupa je takozvani ćelijski ili sitnozrnasti algoritmi. Potpopulacije  
su veoma male, možda i samo jedna jedinka. Ova jedinka može biti u komunikaciji sa svojim komšijama. Karakteristike najbolje jedinke polako se razlivaju po rešetki, ne dozvoljavajući jednoj jedinki da rano dominara populacijom. Na SIMD (Single instruction, multiple data) računarima kada je funkcija troška dovoljno teška, ovo je prirodna realizacija.

## Vrste paralelnih genetskih algoritama

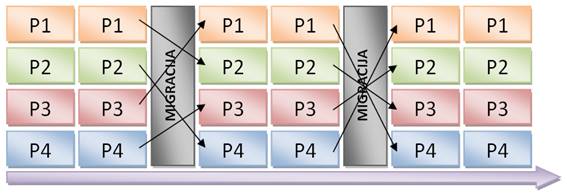
Paralelni genetski algoritmi prema tipu paralelizacije se dele na :

* Distribuirane genetske algoritme
* Masovno paralelne genetske algoritme
* Globalne paralelne genetske algoritme
* Trivijalne paralelne genetske algoritme
* Hijerarhijske paralelne genetske algoritme
* Hibridne paralelne genetske algoritme



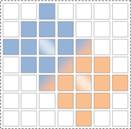
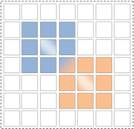
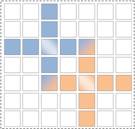
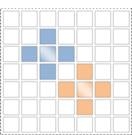
### Distribuirani genetski algoritam

Distribuirani genetski algoritam (skraćeno DPGA) ili raspodeljeni genetski algoritam sastoji se prema krupnozrnatoj podeli od nekoliko potpopulacija pa se naziva još i višepopulacijski genetski algoritam. To je najpopularniji model paralelnih genetskih algoritama.

Čvorovi su najčešće umreženi računari, ali mogu biti i procesori u višeprocesorskom sistemu koji međusobno komuniciraju preko zajedničkog radnog okvira. Genetski algoritmi u čvorovima mogu se međusobno razlikovati. Zajednički im je problem optimizacije koji svi paralelno rešavaju. Čvorovi međusobno razmenjuju jedinke u nadi da će novopristigla jedinka u novoj okolini podstaknuti pretraživanje još neistraženog područja prostora rešenja i da ćemo na taj način dobiti još bolje rešenje.

*Distribuirani sistem sa četiri populacije*

### Masovno paralelni genetski algoritam

Masovno paralelni genetski algoritam (skraćeno MPGA) je veoma sličan DPGA, jedna od razlika je to što se migracije obavljaju samo između susednih jedinki. Za razliku od ostalih modela MPGA zahteva multiprocesorski računar koji se sastoji od mnogo (nekoliko stotina ili nekoliko hiljada) procesora. On se prema sitnozrnastoj podeli sastoji od Np procesora koji predstavljaju Np jedinki. To znači da je veličina populacije jednaka broju procesora, ali ukoliko veličina populacije mora biti veća od broja procesora (jer u suprotnom genetski algoritam ne daje zadovoljavajuće rezultate), postoji mogućnost da neki ili svi procesorski elementi predstavljaju i po više od jedne jedinke. Svaki procesor obavlja genetske operacije nad svojim ali i nad susednim jedinkama.

*Različite topologije susedstva u MPGA, procesori obojeni sa dve boje predstavljaju procesore na kojima se događa preklapanje susedstva istaknutih procesora*

### Globalno paralelni genetski algoritam

Globalni paralelni genetski algoritam (skraćeno GPGA) je predstavnik master-slave organizacije. GPGA uvek koristi standardni pristup, što znači da postoji samo jedna populacija. Kod tradicionalnog GPGA, paralelni deo posla obavljaju sluge, dok sekvencijalni deo obavlja gospodar. Sluge su zadužene samo za evaluaciju jedinki, a gospodar obavlja sve ostale delove genetskog algoritma. Komunikacija između gospodara i slugu odvija se kada gospodar dodeli jedinke slugama i kada sluge vraćaju gospodaru izračunate vrednosti funkcije troška. Međutim ako sluge pored funkcije troška rade i selekciju, ukrštanje i mutaciju onda se taj GPGA ne naziva tradicionalnim.

GPGA je jednostavan za implementaciju i nema potrebe za uvođenjem dodatnih parametara genetskog algoritma. Veća ubrzanja GPGA-a postižu se što je funkcija evaluacije vremenski zahtevnija. U tom slučaju se paralelizacijom postižu veće uštede nego u slučajevima kada je trajanje evaluacije zanemarljivo u odnosu na trajanje ostalih operacija.

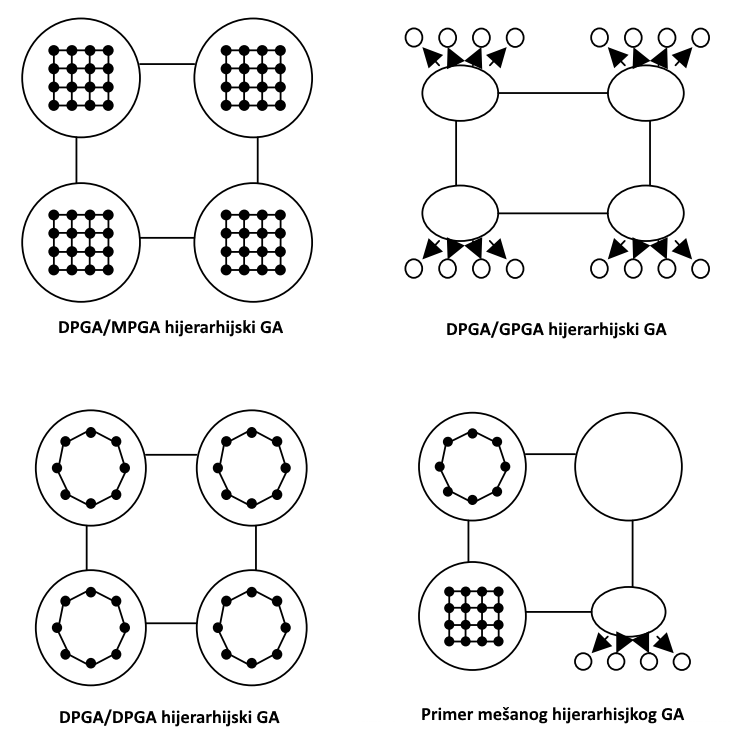
### Trivijalni paralelni genetski algoritam

Trivijalni paralelni genetski algoritam (skraćeno TPGA) zapravo predstavlja više genetskih algoritama koji se paralelno obavljaju na nekoliko potpuno nezavisnih računara (računari ne moraju biti povezani u mrežu) kako bi se, na primer, statistički obradili eksperimentalno dobijeni rezultati ili kako bi se odredio optimalan skup parametara. Ovaj ekstremno jednostavan paralelni metod je, u stvari, izuzetno koristan upravo za statističku analizu algoritma. Na primer, isti algoritam se pokrene na nekoliko odvojenih računara s različitim početnim parametrima i posmatra se kvalitet dobijenog rešenja. Potom se rezultati statistički obrađuju. S obzirom da je genetski algoritam stohastički proces, prikupljanje statističkih podataka je od izuzetne važnosti.

|  |  |
| --- | --- |
| Prednosti | Mane |
| * Dobar za statističku analizu * Jednostavan | * Zahteva statističku obradu rezultata * U većini slučajeva nije primenljiv |

### Hijerarhijski paralelni genetski algoritam

Hijerarhijski paralelni genetski algoritam (skraćeni HPGA) je kombinacija distribuiranog, masovno paralelnog i globalno paralenog genetskog algoritma. Na primer, to može biti distribuirani genetski algoritam na nekoliko međusobno povezanih računara, a na svakom od tih računara obavlja se globalni paralelni genetski algoritam nad potpopulacijama. Hijerarhijski model uglavnom na višem nivou ima DPGA.



### Hibridni paralelni genetski algoritam

|  |  |
| --- | --- |
| Prednosti | Mane |
| * Brže i bolje fino podešavanje rešenja * Brža konvergencija | * veća verovatnoća zaostajanja u lokalnom optimumu zbog manje raspršenosti rešenja |

Hibridni paralelni genetski algoritam (skraćeno HyPGA) je kombinacija jednog DPGA, MPGA, GPGA ili HPGA modela sa nekim drugim algoritmom za lokalno pretraživanje. Najčešće se radi o nekoj od gradijentnih metoda koje se primjenjuju nakon određenog broja iteracija i to samo nad nekim jedinkama. Hibridni genetski algoritam ima sva dobra svojstva sekvencijalnog genetskog algoritma, s tim da je još poboljšano fino podešavanje rešenja.

# Zaključak

Napretkom tehnologije izrade hardvera pojavila su se i hardverska rešenja tzv. masovnih paralelnih računara koja su se primenila i za rešavanje problema optimizacije koristeći paralelne genetske algoritme. Kao posledica toga ubrzano počinju da se javljaju i novi modeli paralelizacije ovih genetskih algoritama, što utiče na razvoj mnogih industrija.

Kombinacije hibridnih genetskih algoritama i neuralnih mreža će u budućnosti imati sve veći značaj. Glavni faktori budućeg razvoja će biti dostupnost još većih hardverskih kapaciteta i bolje infrastrukture za povezivanje u bolje distribuirane sisteme. Evoluciono računarstvo u ovom trenutku sve više pomera fokus ka adaptivnim veštačkim neuronskim mrežama, vrlo složene strukture, koje su sposobne da se same rekonfgurišu i razvijaju.

# Literatura

* Milan Tomić, Genetski algoritmi, 2017.
* <http://www.zemris.fer.hr/~golub/ga/ga_skripta2.pdf>
* <http://www.zemris.fer.hr/~golub/ga/studenti/2010_bradvica/index.html>
* <https://sr.wikipedia.org/wiki/Genetski_algoritam>
* <https://bib.irb.hr/datoteka/443082.ZRMajaLegac.pdf>
* <https://www.mi.sanu.ac.rs/~jkratica/papers/phd.pdf>