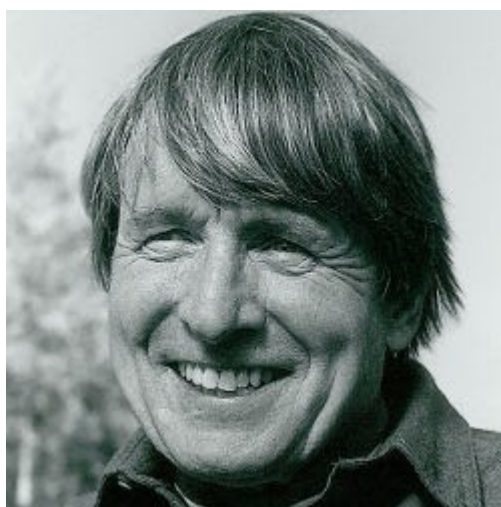


Evolucija programa upotrebom Genetičkog programiranja

Uvod

Sredinom 60-tih godina profesor Holland je prvi razvio metodu rešavanja problema optimizacije genetskim algoritmom. Njegov student Goldber razvio je i usavršio metodu za rešavanje kompleksnih problema optimizacije. 1989 godine Goldber objavljuje knjigu, "Genetski algoritmi u pretrazi, optimizaciji i mašinskom učenju", koja se poslije javlja kao inspiracija i startna tačka skoro svih metoda razvijenih na bazi genetskog programiranja.



Slika 1. John H. **Holland**

Genetsko programiranje je jedan od pokušaja odgovora na pitanje kako navesti računar da reši određeni zadatak bez davanja direktnih instrukcija o postupku rešavanja, tj kako potići da računar uradi nešto korisno, a da mu nismo rekli kako ti treba da učini. Genetsko programiranje pripada paradigmi racunarske inteligencije, evolutivnim izracunavanjima, zajedno sa genetskim algoritmom, evolutivnim programiranjem, evolutivnim strategijama.

Kao sto smo vec rekli, u evolutivnim izracunavanjima se ne definiše kako računar treba da reši neki problem, ali definišemo način na koji može naučiti rešavanje nekog problema. U našem slučaju postupak „učenja“ se odnosi na oponašanje procesa evolucije u prirodi upotrebom određenih operatora na skup rešenja (populaciju). Na kraju postupka najbolje rešenje (ili više njih) u skupu predstavlja rešenje izvornog problema. Jedinственost genetskog programiranja je u tome što je svako rešenje predstavljeno strukturom promenljive veličine i oblika, što je idealna postavka za predstavljanje računarskih programa.

Genetsko Programiranje

Genetsko programiranje je tehnika koja omogućava rešavanje nekog problema bez potrebe za stvaranjem programa koji rešava zadati problem. Način rada genetskog programiranja je uporaba evolutivnog procesa u stvaranju računarskih programa. Evolucija se u prirodi javlja radi preživljavanja i reprodukcije sposobnih jedinki koje predaju svoj genetski kod sledećim generacijama. Kao posledica ovog procesa, genetski kod koji čini sposobnije jedinke ima mogućnosti da poveća svoj udeo u celokupnoj populaciji, što uzrokuje promene na nivou vrste.

Ono što čini genetsko programiranje posebnim u odnosu na ostale tehnike evolutivnih izračunavanja, je u činjenici da jedinke u populaciji genetskog programa predstavljaju računarske programe, odnosno strukture koje se mogu jednoznačno preslikati u oblik pogodan za izvođenje na računaru. Kako bi se olakšao proces stvaranja novih programa iz jednoga ili više roditeljskih, programi su u većini primera genetskog programiranja napisani u obliku stabla. Novi programi se dobijaju jednostavnim manipulacijama nad već postojećim stablima, uklanjanjem grana iz jednoga stabla, dodavanjem tih grana na određeno mesto u drugom stablu i slično. Ovaj proces kao rezultat takođe daje stablo i osigurava sintaksičku ispravnost dobijenih rešenja.

Elementi genetskog programiranja

Pri rešavanju problema genetskim programiranjem potrebno je definisati određene elemente sistema:

- podacijski i funkcijski elementi rešenja
- funkcija dobrote (fitnes funkcija)
- parametri genetskog programa
- uslov zaustavljanja
- građa rešenja

U slučaju rešenja predstavljenog stablom, podatkovni elementi su završni čvorovi stabla tj. listovi, koje nazivamo podatkovnim čvorovima, dok funkcijske elemente nazivamo funkcijskim čvorovima. Funkcijski čvorovi su u većini slučajeva unutrašnji čvorovi stabla, ali u nekim primerima funkcijski element može imati nula argumenata, pa samim tim mora biti i

završni čvor stabla. Odabir podatkovnih i funkcijskih elemenata određuje mogući prostor pretraživanja, kao i mogućnosti rešavanja problema.

Parametri algoritma uključuju veličinu populacije, maksimalna dubina i širina stabla (opciono), verovatnoće primene operatora i slično. Određivanje građe rešenja podrazumeva definisanje struktura u obliku potprograma koje glavni program može pozivati, one se nazivaju automatski definisanim funkcijama (engl. automatically defined functions, ADF).

Odabir podatkovnih i funkcijskih elemenata

Najvažniji zahtev koji treba zadovoljiti prilikom odabira podatkovnih i funkcijskih elemenata je taj da je uz pomoć njih moguće izraziti rešenje problema kojeg pokušavamo da rešimo. Ovo svojstvo se naziva potpunost (engl. sufficiency), ono zavisi od samog problema i nije ga moguće definisati u opštem slučaju. U najvećem broju slučajeva u skup podatkovnih i funkcijskih elemenata stavljaju se svi elementi za koje se smatra da bi mogli biti od koristi za trenutni problem.

Drugo svojstvo koje skup elemenata rešenja treba zadovoljiti je zatvorenost (engl. closure). Zatvorenost zadatog skupa elemenata definiše se, za bilo koji funkcijski element, kao sposobnost prihvatanja rezultata od bilo kojeg drugog funkcijskog ili podatkovnog elemenata. Zadovoljavanje ovog svojstva zavisi o vrsti podataka svojstvenoj za određeni problem. U slučajevima mešanja različitih vrsta podataka, potrebno je za svaku moguću kombinaciju definisati transformaciju ili način interpretacije ulaznih vrednosti. Čak i u slučaju jedinstvene vrste podataka, zatvorenost može biti narušena ponašanjem nekih funkcija, npr. rezultat deljenja nulom... Neka od rešenja ovakvih problema jesu menjanje definicija funkcija za kritične vrednosti, ili definisanje neadekvatnog tipa podataka.

Funkcija dobrote

Funkcija dobrote ili fitnes funkcija je najzaslužniji element u rukovođenju evolucijom i upravljanjem kretanja celokupne populacije. Ona treba da nagrađuje ne samo bolja rešenja, već i sva poboljšanja pronađena tokom evolucije. U većini primera genetskog programiranja određivanje fitnes funkcije oduzima najviše procesorskog vremena. Određivanje fitnes funkcije zavisi od samog primera, često se ispituje ponašanje rešenja na nizu ispitanih primera.

Parametri genetskog programa

Izbor odgovarajućih vrednosti zavisi od konkretne primene i ne postoji skup vrednosti parametara koji su najbolji za svaki problem. Postavljanje parametara postaje složenije sa povećanjem zavisnosti između njih, pa se u većini slučajeva koriste one vrednosti koje ispitivač smatra najprikladnijima za konkretan problem.

Jedan od najvažnijih parametara u genetskom programiranju je veličina populacije, za netrivialne probleme preporučuje se da ona bude veća od 1000 jedinki, a u nekim primerima korišćene su populacije i sa više od 100000 jedinki.

Uslov zaustavljanja

Najčešći oblik uslova zaustavljanja je dostizanje predefinisano broja generacija. J. Koza u svom radu za sve eksperimente koristi broj od 50 generacija kao uslov zaustavljanja ako prethodno nije pronađeno tačno rešenje. Argument iza ove vrednosti je zapažanje da nakon tog broja generacija genetski program gubi sposobnost pronalazanja bitno različitih rešenja, tj. da su eventualna naknadna poboljšanja vrlo mala. U najvećem broju slučajeva radi se dvostruka provera, algoritam se zaustavlja ili nakon zadatog maksimalnog broja generacija ili nakon što u zadatom broju uzastopnih generacija nije postignuto poboljšanje fitnes funkcije najbolje jedinice.

Građa rešenja

Ako posmatramo uobičajeni prikaz rešenja genetskog programiranja u obliku stabla, građa rešenja može podrazumevati definisanje potprograma odnosno, automatski definisanih funkcija (ADF). ADF je potprograma koji može biti pozvan od strane glavnog programa, a svi eventualni potprogrami, kao i glavni program, se izgrađuju dinamički tokom evolucije. U svom radu Koza daje niz primera u kojima genetski program sa automatskim funkcijama brže pronalazi rešenje ili uspeva rešiti teže probleme nego klasičan genetski program. Osim upotrebe automatskih funkcija, građa rešenja može uključivati i višestabalnu strukturu, u kojoj je svako pojedinačno stablo zaduženo za različit deo problema.

Prednosti i nedostaci genetskog programiranja

Prednost genetskog programiranja je što se njime može rešavati proizvoljni optimizacioni problem, pri čemu postoji veliki broj mogućih nadogradnji i povećanja učinkovitosti.

Jednom kada se pronađe rešenje, ono ne predstavlja rešenje odabranog problema samo za konkretne vrednosti, već predstavlja algoritam rešavanja koji se može primeniti nad proizvoljnim podacima. Genetskim programiranjem dobijamo čitavu populaciju rešenja, čime se omogućuje odabir više najboljih rešenja ukoliko je to potrebno.

Još jedna prednost genetskog programiranja je jednostavnost izvođenja, jednom kada se definišu osnovni elementi izgradnja rešenja se u potpunosti prepušta genetskom programu.

Jedan od najčešćih problema genetskog programiranja je preuranjena konvergencija. Ova pojava se dešava ukoliko jedna ili nekoliko relativno dobrih jedinki, ali ne i optimalnih, postepeno preovlada u populaciji i proces konvergira u lokalnom ekstremu. U tom slučaju mogućnosti za poboljšanje trenutnog rešenja su male, iz razloga što selekcija i ukrštanje u populaciji sa istim jedinkama nema efekta. Jedino mutacija može da doprinese izlasku iz date situacije, međutim u praksi je često bez efekta jer su nivoi mutacije, u većini slučajeva, veoma mali i prave neznatne razlike na genetskim materijalima, i dominantne jedinke ponovo eliminišu ostale jedinke iz populacije. Preuranjena konvergencija najčešće nastaje kao posledica primene proste ruletske selekcije.

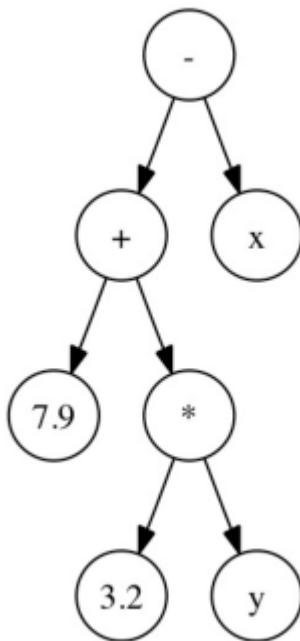
Ukoliko u populaciji postoji jedinka sa relativno velikom funkcijom prilagođenosti, ona će najverovatnije istisnuti sve ostale jedinke iz populacije.

Drugi čest problem jeste spora konvergencija, koja se dešava u kasnijoj fazi izvršavanja. Dešava se kada populacija postane dovoljno slična, ali nije postigla optimalno rešenje. Razlike između najbolje jedinke i ostalih jedinki u populaciji su male, zbog čega postoji nedovoljni gradijent u fitnes funkciji koji bi pomogao algoritmu da dostigne optimalnu vrednost.

Operatori i parametri genetskog programiranja

Reprezentacija podataka

Podaci se u memoriji predstavljaju preko strukture stabala, svaki čvor drveta sadrži funkciju operatora a svaki terminalni čvor sadrži operand. Drvo se lako može obilaziti na rekurzivan način, što takođe predstavlja jedan od razloga upotrebe stabla kao strukture.



Slika 2. Reprezentacija izraza $7.9 + (3.2 * y) - x$ upotrebom stabla

Genetički operatori

Genetički algoritmi prirodno opisuju proces evolucije i varijacija gena je nužna da bi se taj proces uspešno simulirao. Genetski operatori koji se koriste u genetičkim algoritmima su analogni onima u prirodnom svetu.

- Selekcija koja predstavlja opstanak naprikladnijih gena
- Ukrštanje koje predstavlja reprodukciju, rekombinaciju gena
- Mutacija koja predstavlja izmene na jednom genu

Selekcija

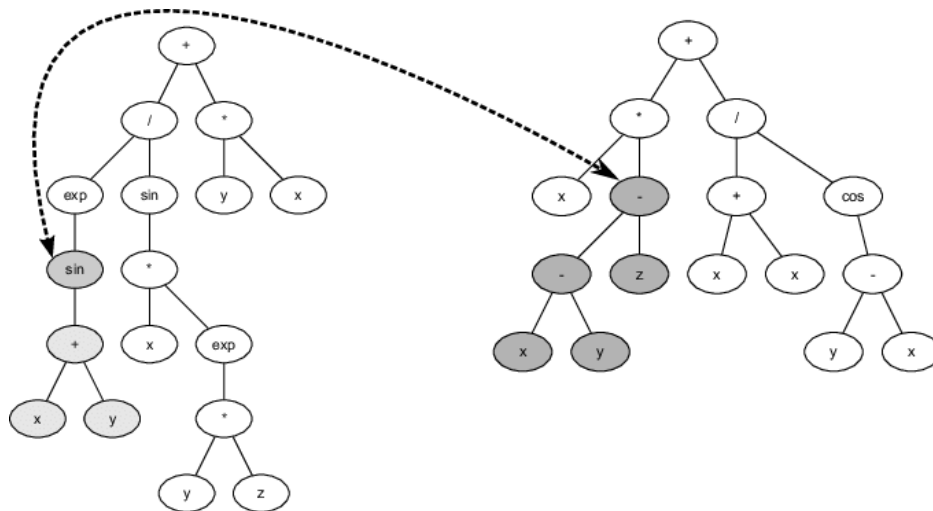
Selekcija predstavlja proces kojim se određeni pojedinci biraju iz trenutne generacije za roditelje naredne generacije. Pojedinci se biraju tako da bolje jedinke imaju veće šanse da budu odabrane. Najčešća metoda selekcije je selekcija turnira mada je dokazano da i druge metode kao što su proporcionalna selekcija, ruletska selekcija imaju dobar učinak.

Cilj selekcije je čuvanje i prenošenje dobrih osobina na sledeću generaciju jedinki. Selekcijom se biraju dobre jedinke koje će učestvovati u sledećem koraku, u ukrštanju. Na taj način se dobri geni čuvaju i prenose na sledeće populacije, a loši odumiru.

Ukrštanje

Ukrštanje (crossover) je osnovni operator kojim omogućava stvaranje novih jedinki. Ukrštanje predstavlja razmenu genetskog materijala između dve jedinke analogno istoimenom procesu nad živim jedinkama.

Najčešći operator ukrštanja: zamena dva slučajno odabrana prosta stabla između roditelja.

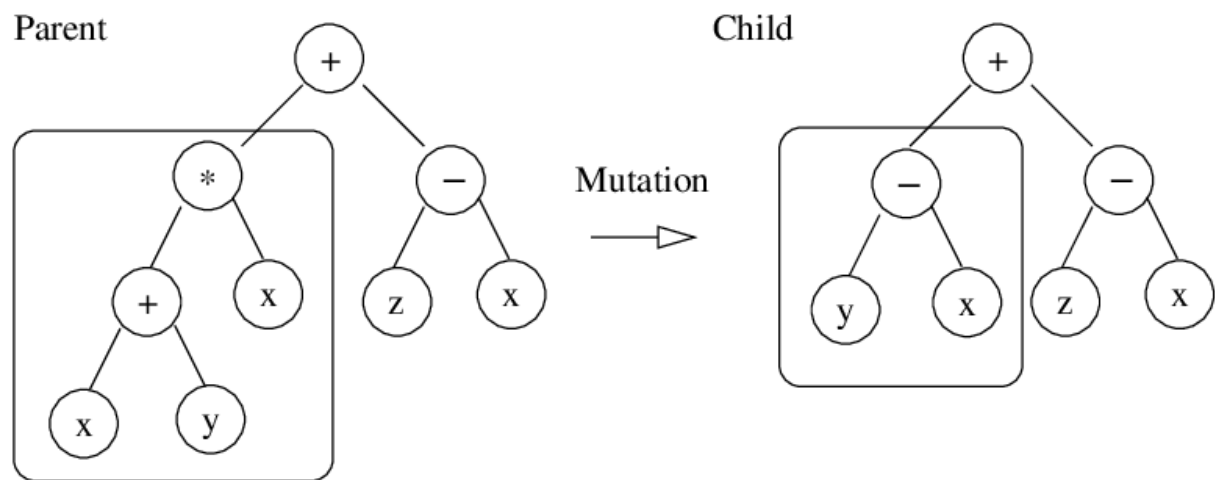


Slika 3. Zamena dva podstabla

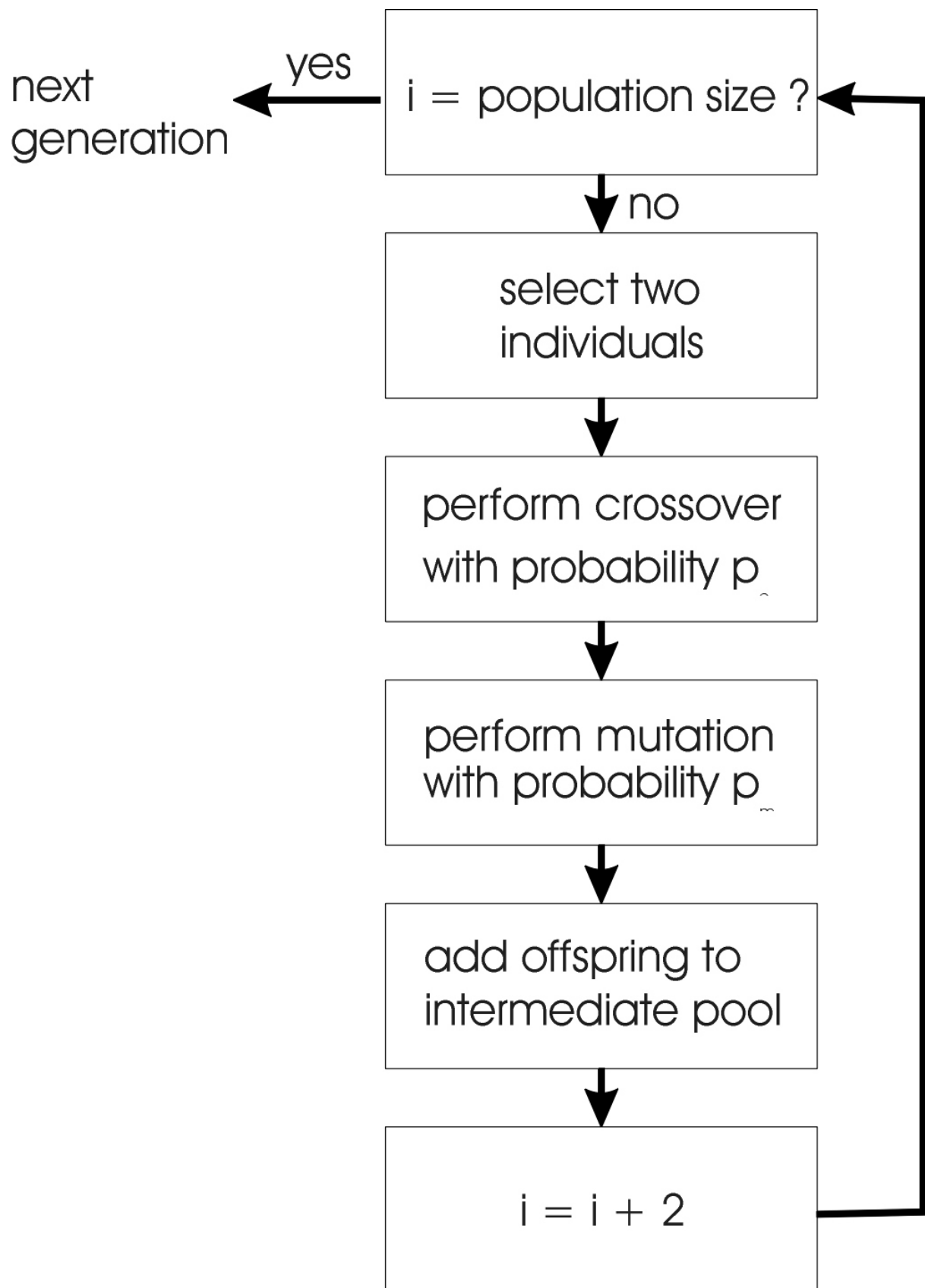
Mutacija

Mutacija je najvažniji operator čijim delovanjem se vrši izmena slučajno izabranih gena jedinke. Kako deluje nad samo jednom jedinkom, mutacija je unarni operator koji kao rezultat daje izmenjenu jedinku.

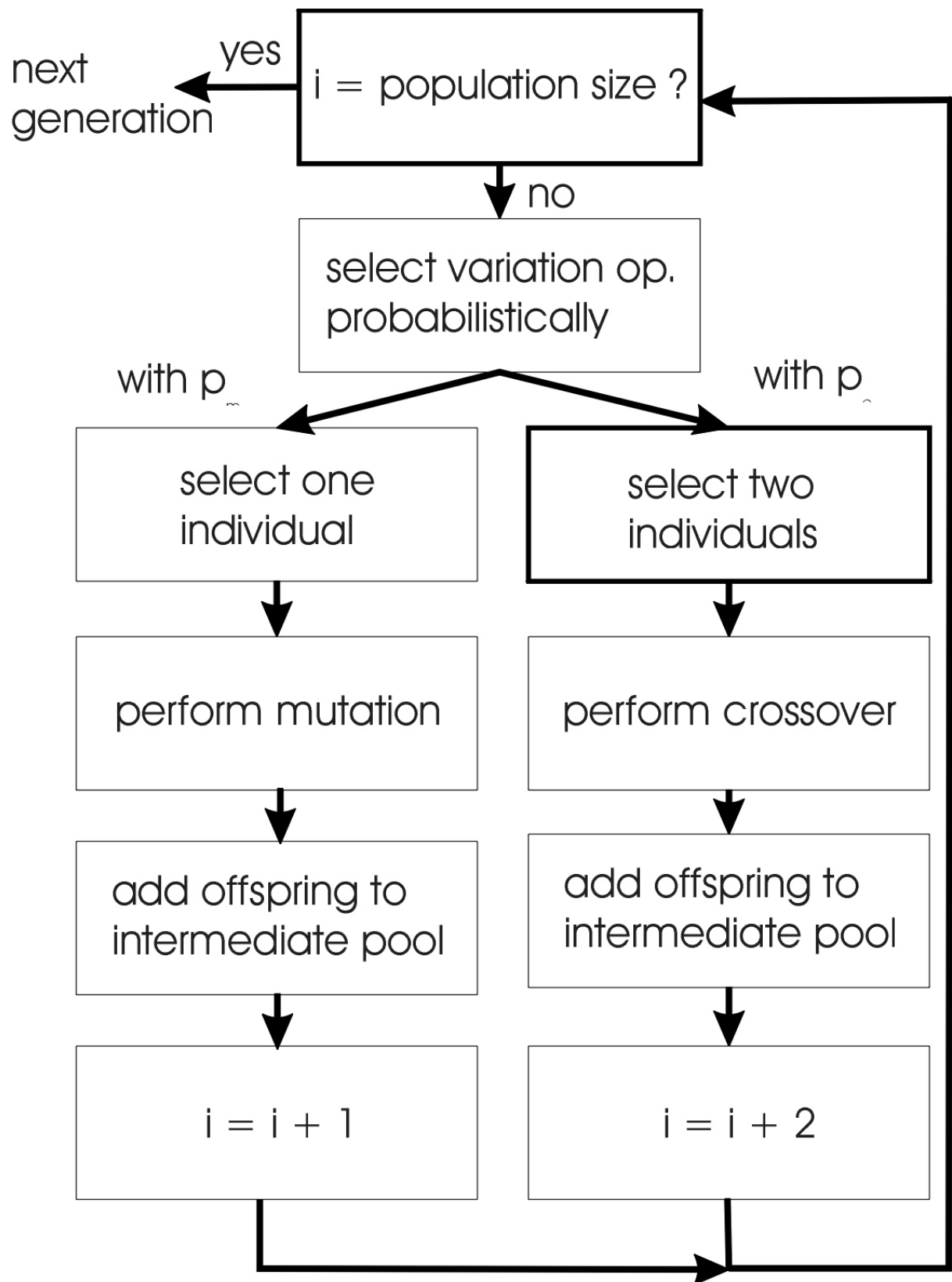
Najčešći operator mutacije: Zameni slučajno odabrano podstablo novim slučajno generisanim stablom



Slika 4. Zamena podstabla slučajno generisanim podstablom



Slika 5. Genetski algoritam



Slika 6. Genetsko programiranje

Opis praktičnog problema

Minimalistički program koji implementira primenu genetičko programiranja za rešavanja simboličke regresije.

Populacija se sastoji od 250 članova.

Ideja je da se pronađe izraz sa jednom nezavisnom varijablom x čiji ispis predstavlja vrednost funkcije $x^4 + x^3 + x^2 + x + 1$.

Funkcije koje se koriste su sabiranje, oduzimanje i množenje.

Terminali su x , -2 , -1 , 0 , 1 , 2 .

Funkcija dobrote je predstavljen inverznom srednje apsolutnom greškom u skupu podataka od 101 ciljne vrednosti.

Literatura

https://en.wikipedia.org/wiki/Genetic_programming#Program_representation

https://www.mi.sanu.ac.rs/~jkratica/papers/phd.pdf?fbclid=IwAR1Kipglw9NLRNQw-9O2bEZVXeKKKXnS4D_txvDX3Ji-nrsyr0HRI5ZuAuY

https://bib.irb.hr/datoteka/408164.Ognjen_Dragoljevic_-_Semanticko_genetsko_programiranje.pdf?fbclid=IwAR2u_GUf6EG4kRLcMp_hgsNUALcmVLB3dJHgKN5euTanMjlbDi7kiVqlgS8
<http://poincare.matf.bg.ac.rs/~kartelj/nastava/RI2019/07.Genetsko.programiranje.pdf>