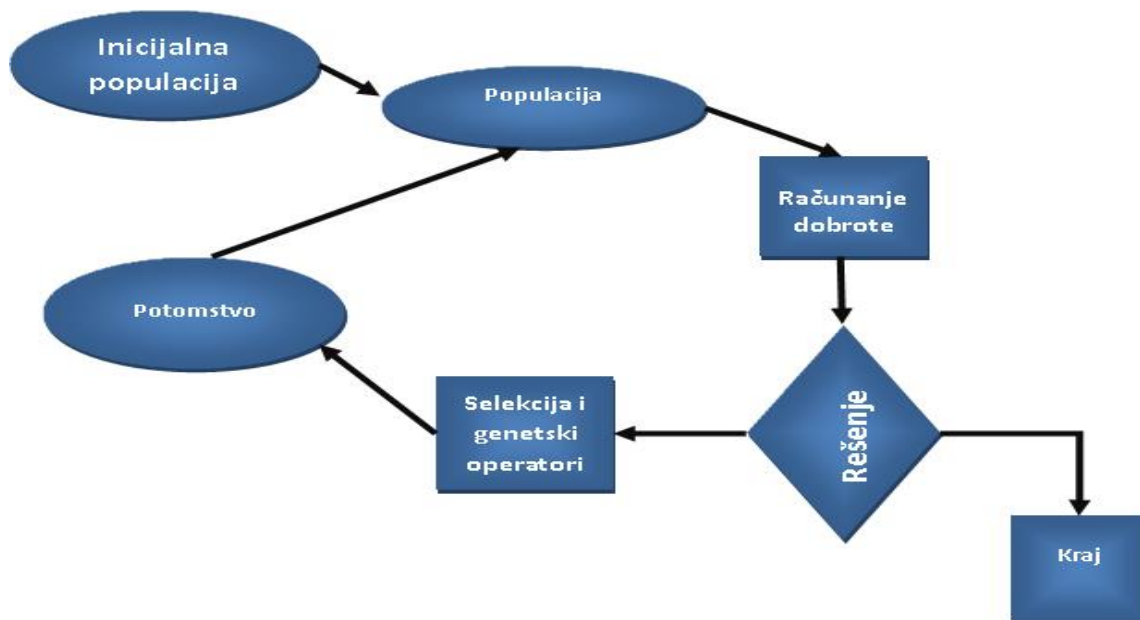


Genetski algoritmi

Genetski algoritmi (slika 3.5) jesu algoritmi pretrage op-te namene. GA koriste principe inspirisane prirodnom genetikom. Osnovna ideja je da se odrflava populacija struktura znanja, koja evoluira tokom vremena kroz proces konkurencije i kontrolisanih varijacija.



Slika 3.5 Genetski algoritam [12]

Svaka struktura u populaciji predstavlja kandidata za re-enje specifi nog problema i ima pridruženu ocenu prilago enosti, koja odre uje koje strukture se koriste za kreiranje novih u procesu konkurencije i takmi enja.

Na primer, posmatra se funkcija $y=f(x)$. Kad se postavi problem odre ivanja optimuma ove funkcije, prvo -to bi trebalo da se uradi jeste da se izra una derivacija funkcije, te se izjedna i s nulom i re-i se matemati ka jedna ina. Me utim, -ta ako funkcija nije derivabilna, odnosno neprekidna u nekom intervalu (-to je est slu aj realnih procesa), -ta ako ona ima vi-e od 3 nezavisno promenljive. U tom slu aju, koristi e se iterativne metode postepenog pribliflavanja datom cilju (jer su jednostavnije i lak-e primenjive pomo u ra unara). Me utim, za vi-e nezavisnih veli ina (10, 20, pa ak i 100) ovaj problem postaje vrlo slofen. Ista je situacija ako je funkcija vrlo kompleksna matemati ka jedna ina. ak se ide do toga da je funkcija $y=f(x)$, skup diskretnih brojeva, odnosno skup eksperimentalnih podataka. U tom slu aju koristi se regresijska analiza i Gausova metoda izra unavanja najmanjih kvadrata, prevo enjem eksperimentalnih rezultata u polinomni oblik, te se opet traffi derivacija polinoma u odre enom intervalu itd.

Svi navedeni problemi (kod re-avanja optimizacije) upravo su **prednosti genetskog algoritma**. Sumiranjem iznesenog, mogu se navesti prednosti kori-enja genetskog algoritma (GA):

- optimizacija kako kontinualnih tako diskretnih promenljivih;
- neprekidnost funkcije nije potreban uslov;
- simultano trafljenje re-enja iz -irokog skupa mogu ih re-enja, inteligentno pribliflavanje ta nom re-enju;
- uspe-no se primenjuje kod velikog broja nezavisno promenljivih veli ina;
- nema tendenciju padanja u lokalne optimume (veliki problem klasi nih metoda);
- uspe-no se primenjuje nad generisanih podacima - eksperimentalnim ili analiti kim funkcijama.

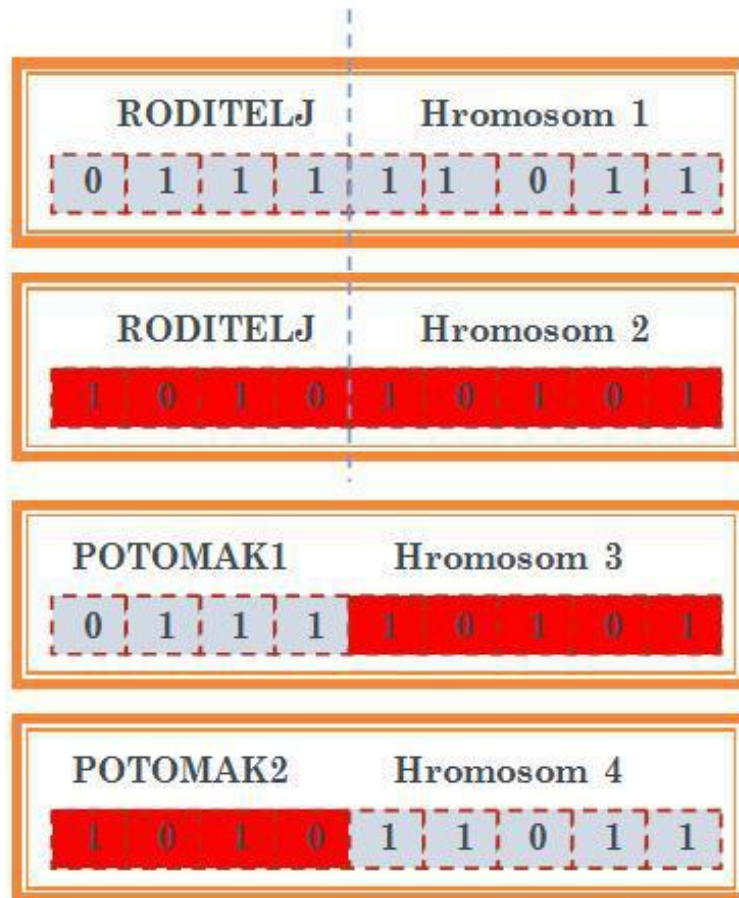
Veli ine u GA, odnosno hromosome prikazujemo binarno (binarnim brojevima), celim ili realnim brojevima. Zavisno od prirode procesa kojeg optimiziramo, primenjuje se i adekvatan tip hromozoma.

Selekcija je proces kojim se osigurava preno-enje boljeg genetskog materijala iz generacije u generaciju. Postupci selekcije me usobno se razlikuju po na inu odabira hromozoma koji e se preneti u slede u generaciju. Prema na inu preno-enja genetskog materijala selekcije se dele na:

- **generacijske selekcije** ó proces izabere najbolje jedinke i od njih kreira novu generaciju;
- **eliminacijske selekcije** ó proces selekcije elimini-e najgore jedinke iz te generacije.

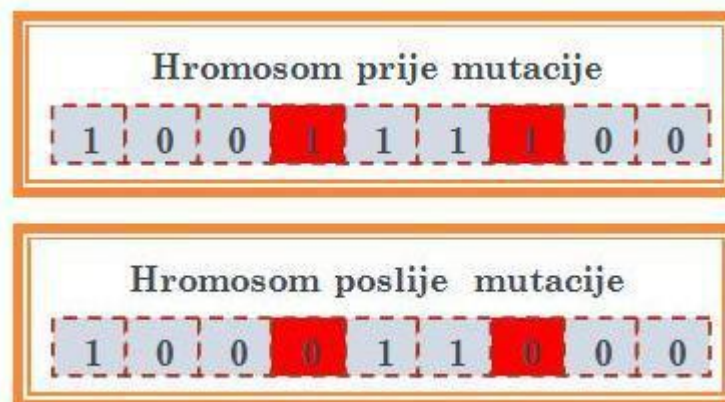
Ukrštanje je proces u kojem se od dva roditelja, parenjem njihovih gena, dobiju jedan ili dva hromozoma koji predstavljaju njihovo potomstvo. Ukr-tanje u GA pomo u ra unara uveliko zavisi od tipa hromozoma. Kada je hromozom prikazan u obliku vektora bitova (binarni hromozom), postoji nekoliko na ina ukr-tanja:

1. Ukr-tanje s jednom ta kom prekida (slika 3.6);
2. Ukr-tanje s dve ta ke prekida;
3. Jednoliko ukr-tanje.



Slika 3.6 Ukr-tanje hromosoma s jednom ta kom prekida [13]

Mutacija kao i sama pojava u prirodi dovodi do toga da se u hromozomu ili jedinki unosi potpuno novi genetski materijal. Ona uglavnom pomaže da se izbegne opadanje u lokalni optimum funkcije cilja. Primenom mutacija postiže se raznolikost genetskog materijala i omogućava pretraživanje novih potencijalno najboljih rešenja (slika 3.7).



Slika 3.7 Mutacija hromozoma predstavljena binarnim brojevima [13]

Kad svi geni u jednom hromozomu postanu jednaki, vrednost hromozoma se ne može promeniti putem ukrštanja, nego mutacijom, gde se postiže da upravo takvi delovi u hromozomu budu promenjeni.

Parametri GA određuju na in evolucije hromozoma u populaciji, te njihovim pode-avanjem može se znatno smanjiti vreme izvo- enja algoritma i agresivnost s kojom e algoritam konvergirati u optimum.

Parametri GA:

1. veli ina populacije M ;
2. broj iteracija I ;
3. verovatno a mutacije v_m ;
4. verovatno a ukrštanja v_u ;
5. metoda selekcije.

Zavisno o tipu hromozoma i vrsti selekcije mogu se pojaviti i neki parametri svojstveni npr. binarnom hromozomu kao to je dužina hromozoma, selekcijski pritisak kod turnirske selekcije i sl.

Verovatno a mutacije je jedan od najvažnijih parametara optimizacionog genetskog algoritma i direktno uslovljava izbegavanje padanja u lokalni optimum funkcije cilja.

Pored verovatno e mutacije, veli ina populacije (M) je parametar koji direktno uti e na kvalitet dobijenih re- enja. Me utim, to je i parametar koji najvi- e uti e na brzinu kompjuterske obrade GA. Ovaj parametar je u direktnoj vezi sa brojem iteracija. To je veli ina populacije manja, potrebnije je smanjiti i broj iteracija, i obrnuto.

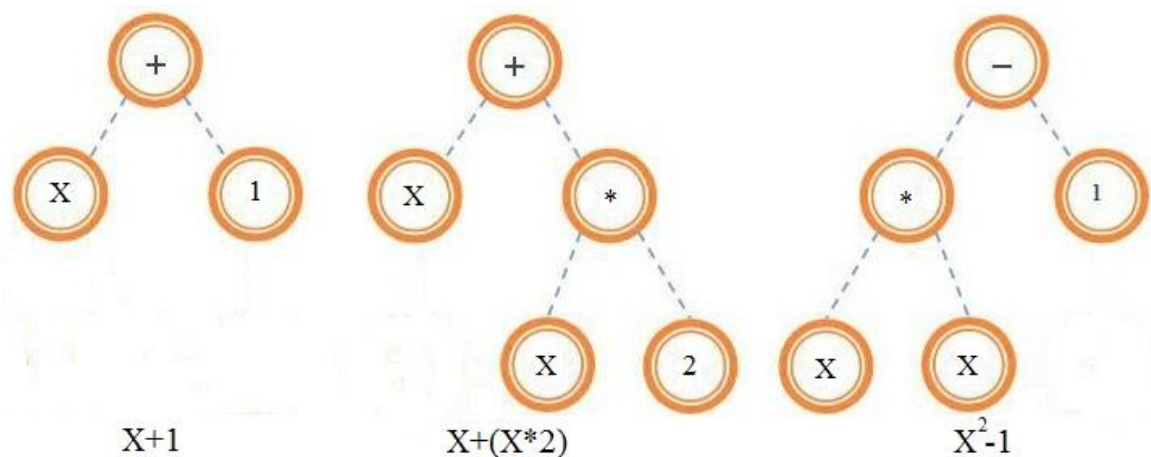
Broj iteracija tako e uti e na vreme obrade algoritma i na kvalitet dobijenih re- enja. To je ve i broj iteracija, to su i re- enja kvalitetnija. S obzirom na to da je cilj da se za to kra e vreme dobiju to kvalitetniji rezultati, cilj svakog istraživanja pomo u GA je da se minimalnim brojem iteracija i veli ine populacije dobiju zadovoljavaju a re- enja.

Poku- aji da se parametri GA poveflu i formira njihova me uzavisnost uvek su rezultirali neuspehom, te se do- lo do injenice da re- enje jednog problema nema uticaja za re- avanje nekog drugog primenom istih parametara.

S obzirom na to da se GA izvodi pomo u ra unara, mogu e je da se parametri GA dinami ki menjaju u toku izvo- enja algoritma tako da se npr. posle 500 iteracija menjaju parametri GA ako algoritam nije u tom razmaku ponudio bolje re- enje.

Genetsko programiranje

Genetsko programiranje (slika 3.8) je napredna grana ra unarske tehnike. To je prvenstveno influjerska metoda koja re- ava odre ene probleme iz oblasti statistike modeliranja influjerskih problema. Genetski algoritmi su prete a genetskog programiranja.



Slika 3.8 Genetsko programiranje [13]

Ideja genetskog programiranja (GP) došla je kao generalizacija GA. S obzirom na to da u GA vršimo manipulaciju sa hromozomima koji predstavljaju binarne prirodne ili realne brojeve, moguće je formirati hromosome koji predstavljaju kompjuterske programe nad kojim bi se vršile operacije ukrštanja i mutacija i tako bi se dolazilo do kompjuterskog programa koji bi rešavao određeni problem. Ovu metodu prezentovao je Džon Koza (John Koza) 1990. godine. Slično kao kod GA, u GP razvijena je metoda reprezentacije hromozoma.

U GP hromosomi u populaciji su u obliku hijerarhijske strukture, sastavljeni od primitivnih funkcija i terminala za pojedina problemska područja.

Skup primitivnih funkcija od kojih su hromozomi sastavljeni čine aritmetičke operacije, matematičke funkcije, Bulove logičke operatore i posebne funkcije specifične za pojedine probleme.

Skup terminala koji tako čine strukturu hromozoma obično su sastavljeni od ulaznih parametara procesa i različitih numeričkih konstanti.

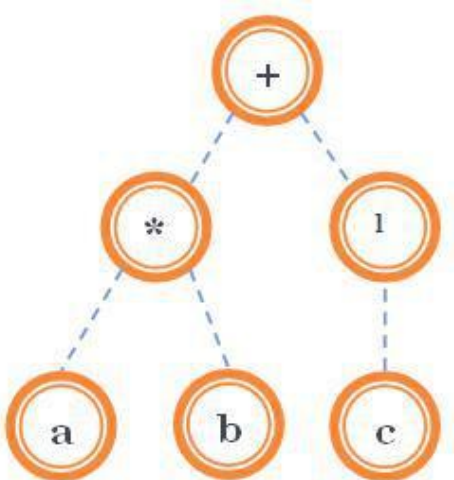
Kompozicija primitivnih funkcija i terminala kao ideja pronadjena je u programskom jeziku LISP, gdje se takve kompozicije nazivaju simbolički izrazi ili **Simbolički izrazi**. Simbolički izrazi predstavljaju se kao binarno uređeno drvo, gdje su koren i ostali unutrašnji čvorovi binarnog drveta označeni funkcijama, dok su spoljašnji čvorovi označeni terminalima. Tako definisan algoritam pretražuje rešenje problema u prostoru svih mogućih kompozicija funkcija, koje se rekurzivno generišu od dostupnih primitivnih funkcija i terminala.

U GP, populacija sastavljena od hromozoma se razmnožava po Darwinovom principu opstanka i reprodukciji najprilagođenijih; preko genetskog reprodukovanja (ukrštanja) operacija prilagođenih parenju kompjuterskih programa.

GP kao i GA počinje inicijalizacijom početne populacije slučajno odabranim kompjuterskim programima, sastavljenim od funkcija i terminala. Svaki kompjuterski program (hromozom) u populaciji evaluira se u smislu kako dobro rešava problem. Analogno GA, ovo merenje zove se merenje funkcije cilja hromozoma.

Hromosomi u GP predstavljaju se u obliku binarnog drveća. Koren i unutra-nji vorovi sastavljeni su od **funkcija**, dok su spolja-nji vorovi sastavljeni od **terminala**.

Kompjuterski programi se predstavljaju u obliku hromozoma: $(+, (*, a, b), (\ln, c))$, njegova reprezentacija (slika 3.9).



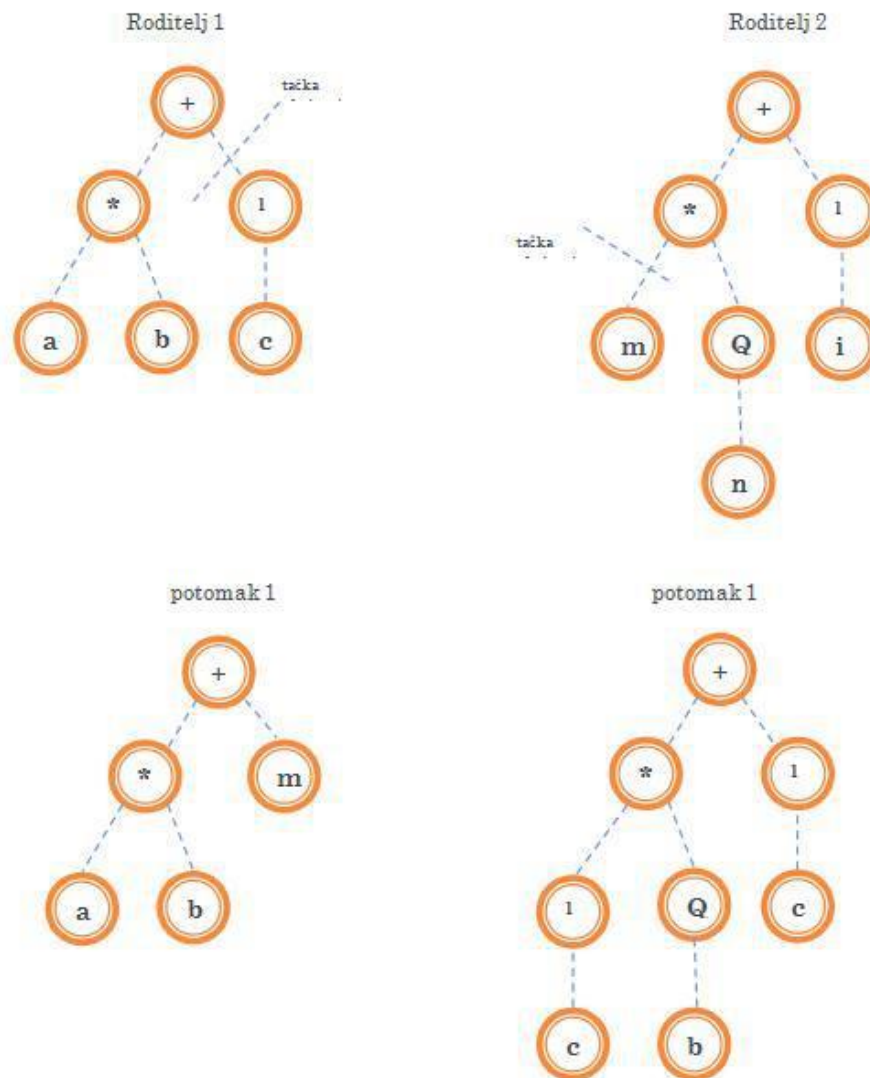
Slika 3.9 Reprezentacija hromosoma u GP ($a*b+\ln(c)$) [13]

Program uzima 3 terminala a, b, c i 3 algebarske funkcije sabiranja, množenja i prirodnog logaritma. Kompjuterski program u prirodnom obliku može se prikazati kao: $a*b+\ln(c)$, gde su a, b, c ulazni parametri problema ili slučajno generisane konstante.

- **Ukrštanje**

U GP kompjuterski programi ukrštaju se shodno Darwinovom principu reprodukcije i opstanka najprilagođenijih jedinki. Slučajnim odabirom dva kompjuterska programa (hromozoma) iz populacije, ukršta se na način da se slučajno odabere tačka ukrštanja, te se genetski materijal razmeni između roditelja, stvarajući na taj način potomke.

Ukrštanje dva kompjuterska programa (slika 3.10).

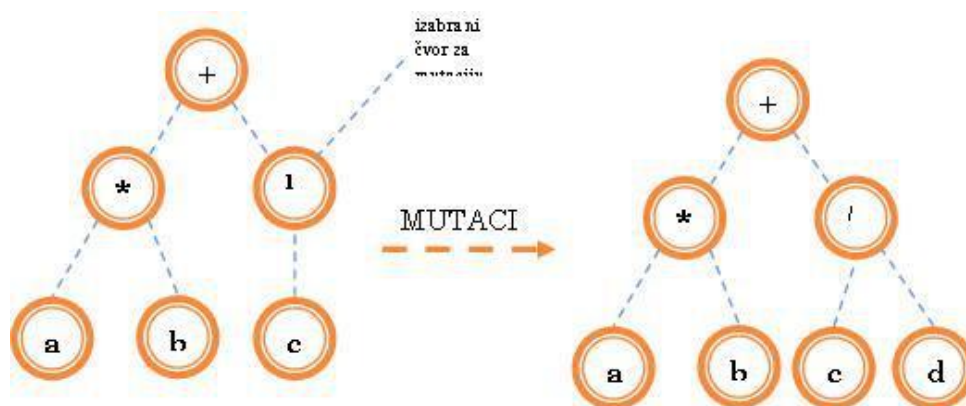


Slika 3.10 Ukr-tanje u GP [13]

Slučajno izabrane tačke ukr-tanja u oba kompjuterska programa označavaju koje delove programa razmeniti i tako generisati svoje potomke.

- **Mutacija**

Kao i kod GA, mutacija predstavlja unošenje novog genetskog materijala u kompjuterski program (slika 3.11). Mutacija se izvodi tako što se slučajno izabere deo programa koji će mutirati, te se slučajno generiše simbol koji predstavlja uniju skupa funkcija i terminala. Donja slika predstavlja primer mutacije kompjuterskog programa, gde je deo programa sa oznakom funkcije *ln* mutirao u funkciju *l*. S obzirom na to da funkcija *l* zahteva dva argumenta, slučajno se generiše drugi argument iz skupa terminala ili funkcija. S obzirom na to da kod mutiranja ovakav proces se može neprekidno generisati, ograničenja u pogledu dubine mutiranja nisu sastavni deo parametara GP.



Slika 3.11 Mutacija u GP [13]

Početna struktura u GP sadrži jedinice u početnoj populaciji sastavljene od slučajno generisanih S-izraza. Početak generisanja počinje slučajnim odabirom jedne od funkcija iz skupa F. Ovo ograničenje korenskog čvora samo na skup funkcija je iz razloga da se struktura ne degenerira u samo jedan čvor.

Nakon generisanja korenskog čvora postoji nekoliko metoda generisanja strukture.

Puna (Full) metoda generisanja

Kod pune metode generisanja, struktura drveta je određena konstantnom veličinom dubine strukture. Punom metodom generiramo binarno drvo iz skupa funkcija sve dok se ne dostigne maksimalni nivo (dubina) strukture. Kada se to desi, generiraju se simboli iz skupa terminala T.

Rastuća (GROW) metoda generisanja

Rastuća metoda sastoji se u slučajnom generisanju strukture. Ovakvo generisanje proizvodi različite oblike strukture. Metoda kreće sa slučajnim generisanjem čvora koji poprima vrednosti iz skupa $C = F \cup T$, odnosno skupa funkcija i terminala. Ovakva metoda ograničena je uslovom da dubina ovako generisane strukture može biti u intervalu 0 do maksimalne dubine drveta. Ako struktura dostigne maksimalnu dubinu drveta, u tom nivou se prisiljava algoritam da poprimi vrednost terminala, a samim tim i završava generisanje.

Mešana ("ramped half-and-half") metoda generisanja

Ovu metodu autor GP.dotneta (ime programa za GP) preporučuje za mnoge probleme koji se rešavaju sa GP. Suština ove metode je da se iskoriste prednosti obe pomenute metode. Iz tog razloga, ova metoda se zove još i mešana metoda generisanja. Metoda se jednostavno objašnjava na primeru:

Ako je 6 maksimalna dubina drveta, tada je 20% populacije sadrži binarne strukture od 2. nivoa dubine, 20% 3. nivoa, 20% 4. nivoa itd., sve do 6. nivoa. Tada populacija u ovom stanju sadrži 50% binarnih struktura koje su generisane rastuom metodom, a 50 % punom metodom.