

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

**Primjena genetskog programiranja u financijskom
sektoru**

Mihael Miklec

Voditelj: *Marko Đurasević*

Zagreb, svibanj, 2018.

Sadržaj

1. UVOD	1
2. OSNOVNI POJMOVI I TEMELJI DIONIČARSTVA	1
2.1 VRSTE ANALIZA	2
3. OSNOVE GENETSKOG PROGRAMIRANJA	3
3.1 GENETSKE OPERACIJE	6
3.2 PRIPREMNI KORACI ZA GENETSKO PROGRAMIRANJE	7
4. PRIMJENA GENETSKOG PROGRAMIRANJA U TRGOVANJU DIONICA	8
5. ZAKLJUČAK	12
6. LITERATURA	13
7. SAŽETAK	14

1. UVOD

Uz sve strelovitiji razvoj računala i raznih drugih tehnologija, ljudi su počeli nalaziti sve više načina da tehnologiju uklope u svoje svakodnevne živote i poslove te da uporabom iste rješavaju brojne šarolike probleme. Tako će tema ovog rada upravo biti primjena jedne relativno nove tehnologije u jednoj od svijetu najvažnijih poslovnih grana. Iako isprva genetsko programiranje i algoritmi zvuče komplicirano, oni su u srži zapravo dosta jednostavni, što ih čini moćnim i općenitim alatom za rješavanje čitavog niza problema iz inženjerske domene. Ukratko, oslanjajući se na računala koja samostalno i automatski rješavaju probleme, ljudi u današnje vrijeme pokušavaju navigirati burnim trendovima u financijskoj industriji, poglavito tržištem dionica. Dok su se nekoć ekonomisti oslanjali na teoriju igre kako bi razumjeli tržišta, danas se sve više ljudski i računalno rađene simulacije koriste kao dopuna, a genetsko programiranje je jedan od najčešćih alata koji se koriste za izradu takvih simulacija. Tako je cilj ovoga rada objasniti osnove genetskog programiranja i zatim opisati kako bi se ono moglo primijeniti na trgovanje dionicama.

2. OSNOVNI POJMOVI I TEMELJI DIONIČARSTVA

Financijska industrija pokriva mnoge grane, ali za primjenu novih tehnologija i automatizacija je najinteresantnije tržište kapitala (burza). Postoji mnoštvo različitih burza i svaka prodaje nešto drugo: od temeljne imovine kao što su valute, kriptovalute, roba, dionice, obveznice pa sve do posebnih ugovora poput budoćnosnica (engl. *futures contracts*). Ovaj rad će se ograničiti na dionice i malo pobliže objasniti kako funkcioniraju, iako je dosta toga primjenjivo i na ostale oblike imovina koje se mogu lagano ponuditi na tržištu kapitala.

Dionice su vrijednosni papiri koji predstavljaju određen udio vlasništva u kompaniji kao legalnom entitetu i potraživanje dijela njezine imovine, dobara i zarade. Koliki udio jedna dionica predstavlja ovisi o broju izdanih dionica.

Postoje dva glavna tipa dionica: redovne i povlaštene. Redovne dionice obično daju vlasnicima pravo da prisustvuju sastancima dioničara i da sudjeluju u donošenju odluka koje će oblikovati budućnost kompanije. Također polažu pravo na isplaćivanje dividendi, dio ostvarene dobiti dioničkog društva (kompanije) koji ide dioničarima. Dividende nisu obavezne i njihovo postojanje je najčešće uvjetovano rezultatima poslovanja i politikom kompanije te ako je taj kapital možda potreban u druge svrhe, poput daljnjeg ulaganja u samu kompaniju kako bi se proširio njezin doseg, ušlo u nova tržišta/područja ili pojačalo istraživanje (npr. novih tehnologija koje bi poboljšale usluge te kompanije). Upravo zato što isplata dividenda nije garantirana, vlasnici povlaštenih dionica imaju jače pravo na imovinu i zaradu kompanije, te imaju prioritet pri likvidaciji kompanije ako bude proglašen bankrot. Dividende će biti prije isplaćene vlasnicima povlaštenih dionica nego onim koji posjeduju redovne. Ali zato povlašteni dioničari nemaju pravo glasa pri donošenju odluka u kompaniji.

Sve dionice koje možemo kupiti/prodati na tržištu kapitala su u jednom trenutku morali proći kroz prvotnu javnu ponudu (engl. Initial public offering – IPO), prva prodaja dionica neke kompanije široj javnosti i institucijskim investitorima. Onda kažemo da su kompanije **javne** i dužne su javnost obavještavati o svojem financijskom poslovanju, najčešće kroz sastanke dioničara i službena financijska izvješća koja najčešće objavljuju na tromjesečnoj bazi. Prije toga su **privatne** i dioničari su najčešće oni s puno kapitala (*venture capitalists* i biznis anđeli) ili osnivatelji tvrtke, i oni njima najbliži (prijatelji, obitelj, radnici). Ali nema načina na koji bi šira javnost mogla postati dioničar u tom stadiju.

Jednom kada su dionice kompanije izlistane na tržištu kapitala, njihovu cijenu će odrediti samo tržište i osnovni zakon ponude i potražnje. Ugrubo pojednostavljeno, ako se prodaje manje dionice nego što su ljudi voljni kupiti, cijena će rasti, a ako je ponuđeno više dionica nego što su ljudi spremni kupiti, cijena će pasti.

Naravno, koliko će neka dionica biti tražena i kako će se cijena kretati ovisi o mnoštvu faktora, više nego što ih se može pobrojati.

2.1 VRSTE ANALIZA

Ugrubo kategorizirano, danas postoji dva načina pomoću kojih dioničari pokušavaju predvidjeti kretanje cijena dionica. Te vrste analiza poznate su kao:

1. **FUNDAMENTALNA ANALIZA** – Fundamentalna analiza se fokusira na „temelje“ kompanije, one podatke koji se mogu izvući iz financijskih izvješća, pregleda samog tržišta i njegove perspektivnosti, imovine koju posjeduje kompanija i udjela tržišta koja drži kompanije. Promatra se poslovanje tijekom nekoliko posljednjih kvartala (ne čita se samo najnovije izvješće) kako bi se stvorila kompletnija slika o mogućem trendu u poslovanju te kompanije. Naravno, analitičari se ne mogu ograničiti samo na podatke vezane za kompaniju, nego moraju i gledati puno širu sliku ako žele napraviti informiranu i valjanu analizu. Tako u obzir moraju uzeti i makroekonomske faktore poput poslovanja cijelog tržišta ili sektora kojem dionica pripada. Kao rezultat fundamentalne analiza dobiva se neka vrijednost (cijena) koju investitori mogu usporediti s trenutnom cijenom dionice. Ovisno o tome kakav je odnos između vrijednosti dobivene analizom i realne, tržišne vrijednosti, dionica može biti precijenjena ili podcijenjena odnosno možda je vrijeme da se proda ili kupi. Zbog truda i opsega posla koji ulazi u izvođenje fundamentalne analize, ona predstavlja dugoročni plan, a ljude koji se bave dionicama na taj način zovemo **investitorima**.
2. **TEHNIČKA ANALIZA** – Za razliku od fundamentalne analize, tehnička analiza predviđa buduće kretanje cijena dionica proučavajući obrasce prijašnjih kretanja, volumena trgovanja, otvorenog interesa i raznih drugih statističkih pokazatelja. Za razliku od fundamentalne analize koja se fokusira na same temelje poslovanja kompanije ili tržišta, tehnička se oslanja na pretpostavku da se cijene kreću u trendovima i na psihologiju tržišta (*povijest se voli ponavljati*), a sve fundamentalne faktore već uzima

kao uračunate u cijenu. Ključ je u prepoznavanju trenutka u kojem se trend zaokreće i zauzeti poziciju u skladu s očekivanim trendom. Grafovi su osnovni alat svakoga tko koristi ovu vrstu analize. Na primjer, korištenjem grafa tehnički analitičar može odrediti razinu potpore, najniže cijene, ili otpora, najviše razine cijene nekog proizvoda unutar promatranog vremenskog razdoblja. Tehnička analiza nije iznimno pouzdana jer u obzir ne uzima vanjske faktore koji mogu utjecati na cijenu dionice, poput skandala, pravnih sporova, političke klime, promjena u vodstvu, promjena u monetarnoj politici, spajanja poduzeća i mnogih drugih vanjskih faktora koji mogu dramatično utjecati na cijenu dionice. Upravo zbog te inherentne nepouzdanosti, tehnička analiza se koristi kratkoročno u donošenju odluka oko prodaje/kupnje dionica kako bi se izbjegao kumulativan učinak mnogih vanjskih faktora. Trgovci dionicama koji se isključivo oslanjaju na tehničku analizu zovemo **špekulantima**.

ANALIZA		
TEHNIČKA	FUNDAMENTALNA	
grafovi	Financijska i druga izvješća	Osnovni alat
kratkoročna	dugoročna	Vremenski interval
špekuliranje	investiranje	Karakter
Univerzalna (jer se ne gledaju fundamenti, samo trendovi)	Specijalizacija (svako tržište nosi svoje specifičnosti koje treba uzeti u obzir)	Širina primjenjivosti
DA	NE	Psihološki faktor

Tablica 1 - Osnovne razlike između tehničke i fundamentalne analize

Trgovac dionicama može odabrati jednu od tih dviju vrsta analiza i držati se isključivo iste, ali može i kombinirati obje u različitim omjerima. Kombiniranje fundamentalne i tehničke analize je najbolja opcija za svakoga tko se bavi trgovinom dionicama na srednje staze - ne trguje dionicama na dnevnoj bazi (tzv. *day trading*), a niti ide tako daleko da kupuje dionice s namjerom da ih drži godinama (pa možda čak i desetljećima) u svojem portfelju. Potonji pristup se može smatrati investiranjem.

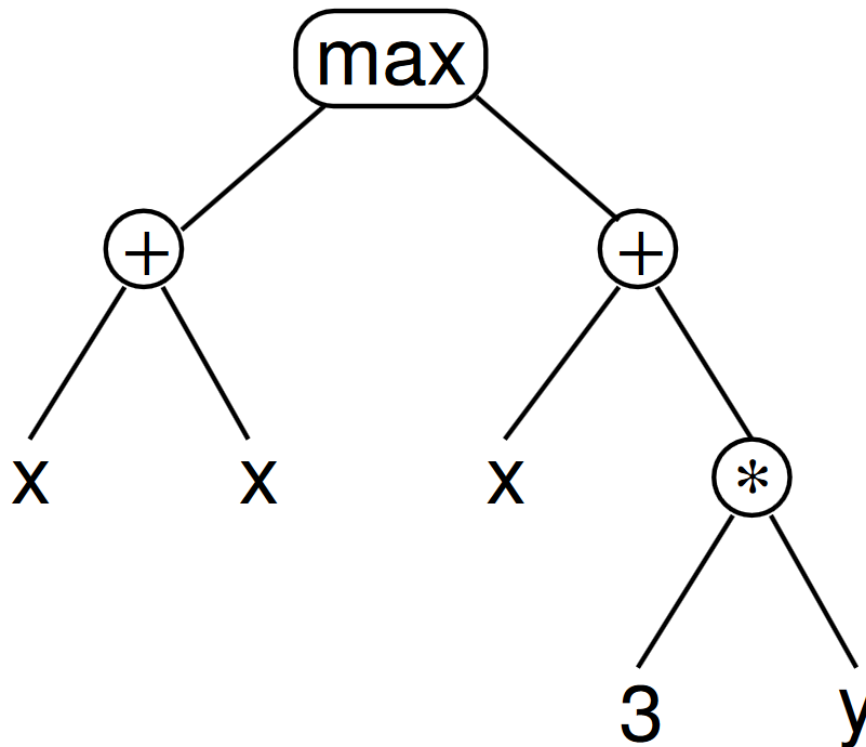
Genetsko programiranje se najbolje može iskoristiti za obavljanje tehničke analize i taj dio posla se može automatizirati, te bi to zajedno s botovima moglo kompletno automatizirati trgovanje dionicama u iznimno kratkim intervalima (toliko kratkim da fundamentalna analiza uopće nije bitna) na dnevnoj bazi.

3. OSNOVE GENETSKOG PROGRAMIRANJA

Najjednostavnije i najsažetije definirano, genetsko programiranje je vrsta evolucijskog računanja koja automatski rješava probleme bez da korisnik mora unaprijed znati ili specificirati formu ili strukturu željenog rješenja. Na najapstraktnijoj razini, genetsko programiranje možemo shvatiti kao sistematsku metodu neovisnu o domeni, koja probleme rješava automatski tako da kreće od high-level tvrdnje koja opisuje što se treba napraviti.

Genetsko programiranje je specijalizacija genetskih algoritama, te se na neke načine razlikuje od ostalih tipova genetskih algoritama. Najprije se razlikuje po načinu na koji su pojedinačni genotipi (jedinke populacije) predstavljeni. Kod drugih genetskih algoritama, genotipi su predstavljeni kao *stringovi* ili *vektori*, dok su u genetskom programiranju predstavljeni kao posebna struktura podataka – **sintaksna stabla**. [3]

Na *Slici 1* je prikazano jedno takvo stablo koje predstavlja program. Slika je preuzeta iz *A Field Guide to Genetic Programming* [2]:

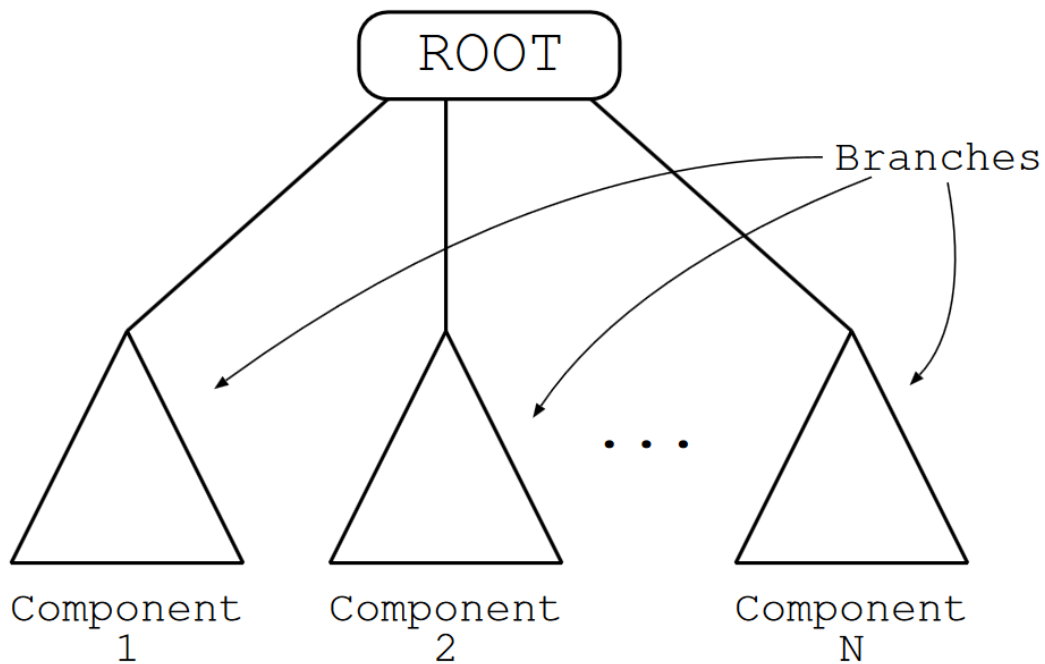


*Slika 1 - Sintaksno stablo koje predstavlja $\max(x+x, x+3*y)$*

Konstante (3 na *Slici 1*) i varijable (x, y) u programu su listovi toga stabla. Njih nazivamo **terminalima** u genetskom programiranju, dok aritmetičke (+, *, max) i druge dozvoljene operacije su interni čvorovi/atomi koje zovemo **funkcijama**. Setovi dozvoljenih terminala i funkcija zajedno formiraju primitivni set jednog GP sistema. [2]

Obično se u literaturi koja se bavi genetskim programiranjem programi zapisuju u prefix notaciji jer je tako lakše vidjeti odnos između (pod)programa i njihovih (pod)stabala. Na primjer, program sa *Slike 1* bi se zapisao kao (max (+ x x) (+ x (* 3 y))). [2]

Nije rijetkost da su programi u naprednijem genetskom programiranju sastavljeni od više stabala (podrutina) povezanih jednim zajedničkim čvorom – korijenom, on se ponaša poput ljepila. Sljedeća ilustracija je također preuzeta iz *A Field Guide to Genetic Programming*.



Slika 2 - Program složen od više stabala (podrutina)

U genetskom programiranju evoluiramo populaciju računalnih programa. Generaciju po generaciju, genetsko programiranje transformira početnu populaciju u populaciju novijih, boljih programa. Da će programi biti bolji, to se mi tek možemo nadati. Kako je genetsko programiranje nasumičan proces, povoljni rezultati ne mogu biti garantirani. Unatoč tome, iako možda malo neintuitivno genetsko programiranje je baš kao i priroda iznimno uspješno u pronalaženju novih i inovativnih načina rješavanja problema. Upravo ta nasumičnost nam omogućava da izbjegnemo zamke u koje bi determinističkim metodama mogli upasti.

Genetsko programiranje možemo opisati algoritmom od par koraka, koji je sažeto prikazan Slikom 1 u obliku osnovnog dijagrama toka.



Slika 3 - Dijagram toka genetskog programiranja

Algoritam genetskog programiranja razbijen u korake:

1. Nasumičnu stvori početnu populaciju koristeći se dostupnim primitivima.

Ponavljaj:

2. Pokreni svaki program i evaluiraju njegovu podobnost/kvalitetu
3. Na osnovi vjerojatnosti podobnosti, izaberi jedan ili više programa iz populacije nad kojima će biti izvršene **genetske operacije**
4. Stvori nove programe koristeći genetske operacije sa specificiranim vjerojatnostima nad odabranim programima

Stani kad:

5. Pronađeno je prihvatljivo rješenje ili je neki drugi uvjet zadovoljen koji signalizira da je vrijeme za prekinuti.
6. Vрати program koji se pokazao kao najbolje rješenje

3.1 GENETSKE OPERACIJE

Spomenute su genetske operacije. Preko znanja iz biologije, već se intuitivno nameće što one predstavljaju. To su operacije koje navode algoritam prema rješenju danog algoritma i kreiraju te održavaju genetsku raznolikost. Postoje tri glavna tipa:

1. **Selekcija/Reprodukcija** – Ova operacija daje prednost boljim rješenjima u populaciji i jednostavno ih kopira iz prošle generacije u novu.
2. **Križanje** – Binarna operacija u kojoj se od nasumično odabranih dijelova (*kromosoma*) programa roditelja stvara novi program dijete koji će biti dio sljedeće generacije. Kombiniranjem dijelova dobrih rješenja, veća je šansa da će genetsko programiranje stvoriti još bolje rješenje. Postoji više različitih vrsta križanja, kao što su, *uniformno*, *rekombinacija brida (ERO)* i *cut & splice*.
U genetskom programiranju, obično se koristi *subtree* križanje. Prvo se u programu nasumično odabire čvor koji će biti točki križanja, zatim se stvara nova jedinka zamjenom podstabla u točki križanja kopije prve jedinice podstablom u točki križanja kopije druge jedinice. Kopije se koriste kako originalni programi roditelji ne bi bili izmijenjeni.
3. **Mutacija** – Unarna operacija (jer djeluje nad jednom jedinkom) koja kreira novu, izmijenjenu jedinku tako da nasumično promijeni odabrane dijelove (*gene*) odabranog programa roditelja. Mutacijom se pretražuje prostor rješenja i kao takva je mehanizam pomoću kojeg izbjegavamo lokalne minimume i slučaj u kojem se programi previše približi jedni drugima po sličnosti. Kada cijela populacija završi u jednom od lokalnih minimuma, jedino slučajnim pretraživanjem prostora rješenja možemo naći bolje rješenje. Dovoljno je da jedna jedinka nastala mutacijom bude bolja od ostalih, pa će se ostale jedinice preseliti u prostor gdje se nalazi bolje rješenje kroz sljedećih nekoliko generacija. [1]

Još jedna uloga mutacije je i obnavljanje izgubljenog genetskog materijala jer ako se slučajno dogodi da svi programi populacije imaju isti gen na određenom mjestu u kromosomu, tada se taj gen nikada ne bi mogao promijeniti samo križanjem. [1] Moguće je doći do rješenja samo mutirajući inicijalnu populaciju, no to će mnogo duži proces nego kada kombiniramo više genetskih operatora.

Najčešće korištena vrsta mutacije u genetskom programiranju je *mutacija podstabla*. Odabire se čvor koji će poslužiti kao točka mutacije i zatim se podstablo u točki mutacije zamijeni nasumično generiranim podstablom.

Osim ova 3 genetska operatora, računalni znanstvenik John Koza je identificirao još jedan – **inverziju** ili **permutaciju**, ali tu genetsku operaciju zanemaruje jer njezina efektivnost nikad nije bila uvjerljivo, zaključno dokazana.¹

Generalno, genetske operacije se u genetskom programiranju međusobno isključuju, za razliku od ostalih evolucijskih algoritama gdje se pri kreiranju novih programa ponekad koristi kompozicija više genetskih operacija. U genetskom programiranju, uobičajeno je da križanje ima vjerojatnost primjene od 90%, mutacije tek 1%, a reprodukcija 9%.

3.2 PRIPREMNI KORACI ZA GENETSKO PROGRAMIRANJE

U ovom radu je u kratkim crtama izloženo koji su koraci u genetskom programiranju i neki osnovni pojmovi, ali još nije objašnjeno ono najbitnije – kako uopće započeti s genetskim programiranjem? Na što se sve treba obratiti pažnja kako bi se mogla pokrenuti simulacija koja će rješavati neki problem pomoću genetskog programiranja?

Kako bi se uopće moglo krenuti s genetskim programiranjem, prvo se trebaju odrediti odgovori na par pitanja koja se mogu smatrati *pripremim koracima* [2]:

- Koji će se terminali koristiti za stvaranje programa?
- Koje će se funkcije koristiti u stvaranju programa?
- Što će biti *mjera dobrote* (mjera kojom ćemo utvrditi koji programi imaju poželjne karakteristike)?
- Koji će parametri biti korišteni za kontroliranje simulacije?
- Što će biti kriterij zaustavljanja i što će biti uzeto kao rješenje simuliranog problema?

Definicija terminala i funkcija koje će se koristiti se mogu definirati kao jedan set – **set primitiva**. Taj set definira sastojke koji su dostupni algoritmu za stvaranje programa. Osim već prije spomenutih varijabli i konstanti, kao terminal se može uzeti i funkcija koja ne prima nikakve argumente. Primjer takve funkcije bi bila `rand()` funkcija. Ona ne prima nikakav dodatan argument, nego samo vraća neki nasumično odabran broj.

Setom primitiva su definirani svi mogući programi koje genetsko programiranje može istraživati, ali i dalje nije jasno kako odvojiti dobre od loših programa kako bi mogli pronaći zadovoljavajuće rješenje ili bar aproksimirati. To evaluiranje se izražava kroz mjeru dobrote, a

¹ Citirano iz: ftp://ftp.cis.upenn.edu/pub/hollick/public_html/genetic/node7.html

ono je rezultat funkcije dobrote. Generalno, prikladnost programa ovisi o rezultatima koji su dobiveni kroz višestruko pokretanje programe uz različite ulazne varijable ili uvjete. [2]

Neki od kontrolnih parametara koji su korišteni u genetskom programiranju su veličina populacije, vjerojatnost provedbe određenih genetskih operacija, maksimalna veličina programa, broja generacija, itd. Uobičajena praksa je da se inicijalna populacija nasumično složi koristeći romped half-and-half, dubine stabla u intervalu od 2 do 6. Veličina populacije bi trebala biti 500, makar se znaju koristiti i mnogo veće populacije. Broj generacije je uglavnom između 10 i 50, jer je najproduktivnija potraga za rješenjem u onim ranijim, pa nema smisla trošiti previše vremena ako točno rješenje nije pronađeno unutar desetak generacija.⁸

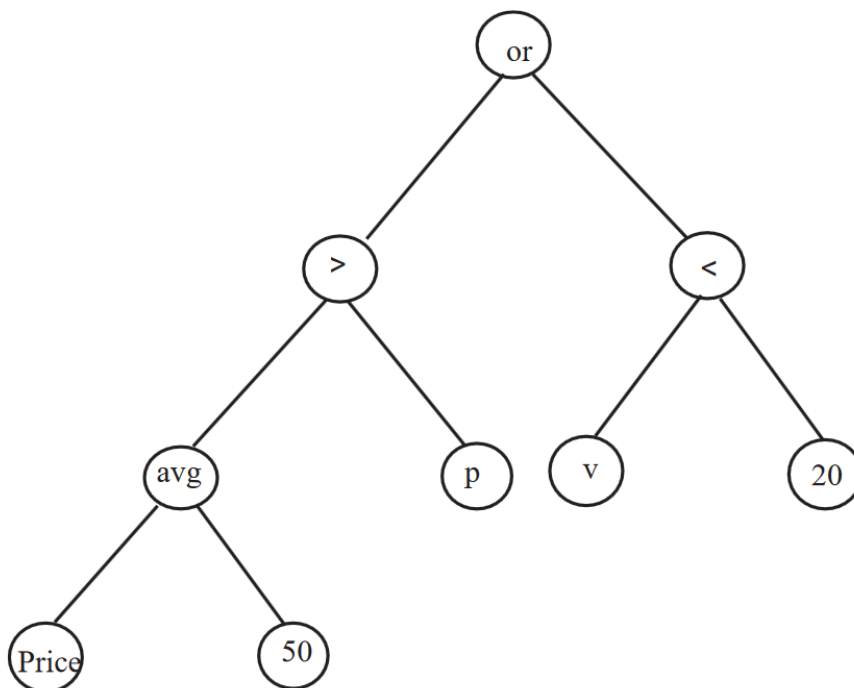
Zadnji pripremni korak je određivanje kriterija zaustavljanja i odlučivanje koji program uzeti za konačan rezultat. Simulacija se može zaustaviti kada naiđe na program koji odgovara očekivanom rješenju nekog problema, ili nakon određenog, maksimalnog broja generacija. Osim određivanja maksimalnog broja generacija koje se mogu izvršiti, možemo ostaviti da se simulacija vrti u „beskonačnost“, sve dok se ne primjeti da u zadnjih nekoliko (npr. 15) generacija nije došlo do bitnijih promjena. Tada se kao rješenje uzima onaj program koji je najprikladniji, iako bi ovisno o problemu ponekad možda htjeli uzeti više jedinki iz zadnje generacije.

4. PRIMJENA GENETSKOG PROGRAMIRANJA U TRGOVANJU DIONICA

Genetsko programiranje bi zapravo trebalo razvijati jedinke programe koje će predstavljati strategije trgovanja. One su u stvari jednostavne booleanske funkcije koji će nam u određenom trenutku reći ili da kupimo ili da prodamo dionice neke kompanije.

Kao set terminala se koriste numeričke i booleanske konstante te druge varijable poput cijene i volumena.

U set funkcija ulaze aritmetičke funkcije, booleanski operatori (and, or, not) i funkcije (if-then-else), relacijski operatori (<, >) i „realne“ funkcije. [4] Slika 4 pokazuje kako bi jedna takva strategija trgovanja mogla izgledati:



Slika 4 - Strategija trgovanja (preuzeta iz [4])

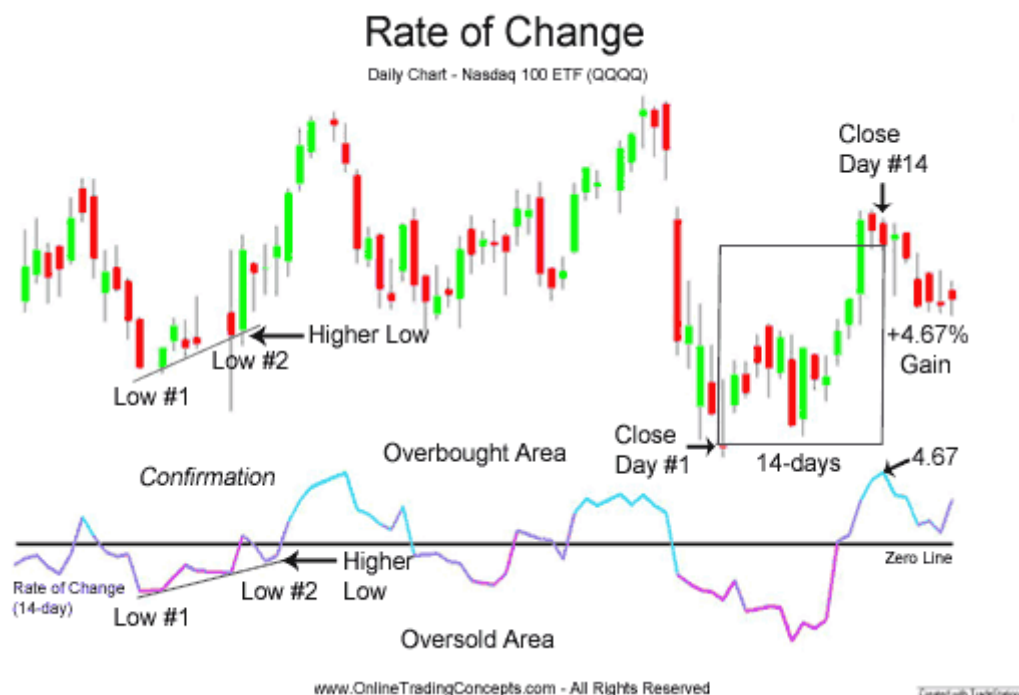
Spomenute su „realne“ funkcije. Zapravo su to složene funkcije koje mogu uzimati viša vanjskih varijabla, npr.:

- Max(n) – uzima maksimalnu vrijednost cijene ili volumena u periodu od n dana
- Min(n) – uzima minimalnu vrijednost cijene ili volumena u periodu od n dana
- Avg(n) – uzima srednju vrijednost cijene ili volumena u periodu od n dana
- Volatilnost(n) – statistička mjera rasapa dobiti od neke dionice kroz period od n dana

Također se mogu koristiti tehnički indikatori koji se inače koriste u tehničkoj analizi. U jednom od primjera korištenja genetskog programiranja pri trgovini s dionicama, korišteni su ROC (Rate of Change) i RSI (Relative Strength Index). [4]

ROC (Rate of Change) – najjednostavniji pomoćni indikator iz grupe oscilatora koji mjeri postotnu promjenu cijene i na taj način odražava relativne pomake krivulje zaključnih cijena. [5] Crta se na posebnom grafu čije je glavno obilježje središnja, nulta linija. Ako je graf iznad nulte linije, kažemo da je stopa promjene pozitivna, a negativna je ako je graf ispod nulte linije promjene. Zapravo se kao signali uzimaju sjecišta grafa i nulte linije. Presijecanje linije prema gore (prelazak iz *oversold* u *overbought* područje na Slici 5) sugerira postojanje kratkoračnog uzlaznog trenda u cijeni, odnosno da bi bilo dobro vrijeme za kupiti dionicu. S druge strane, presijecanje linije prema dolje sugerira silazni trend u cijeni, što bi značilo da je pravo vrijeme za prodati dionicu. ROC se izračunava prema sljedećoj formuli:

$$ROC(n) = \left(\frac{\text{zaključna cijena trenutnog dana}}{\text{zaključna cijena prije } n \text{ dana}} - 1 \right) * 100$$



Slika 5 - Rate of Change tehnički indikator

RSI (Relative Strength Index) – jedan od pomoćnih indikatora koji uspoređuje snagu porasta cijene dionica i snagu njenih padova u određenom razdoblju, kao i svi oscilatori koristi se za prepoznavanje trenutane prenapuhanosti potražnje/ponude (*overbought/oversold*). RSI za razliku od ROC-a ne koristi nultu liniju, nego se graf crta isključivo na pozitivnim dijelovima osi, a vrijednost RSI-a može biti između 0 i 100. RSI manji od 30 tumači se kao signal za kupnju, dok se RSI veći od 70 tumači kao signal za prodaju. Računa se prema sljedećoj formuli:

$$RSI(n) = 100 - \left(\frac{100}{1 + RS(n)} \right)$$

gdje $RS(n)$ označava omjer prosječnih dnevnih porasta i prosječnih dnevnih padova cijene u nekom razdoblju od n dana. [5] Najčešće se gleda kroz period od 14 dana.



Slika 6 - Relative Strength Index tehnički indikator

Tamo su za terminale i funkcije dobiveni sljedeći rezultati: *Slika 7* za kratki period treniranja od 256 dana i *Slika 8* za duži period treniranja od 1498 dana. Period testiranja je u oba slučaja bio isti – 256 dana. Veličina populacije je bila postavljena na 500, broj generacija bio je ograničen na 50, a inicijalna populacija je stvorena *ramped half-and-half metodom* (pola populacije je generirano *full metodom*, druga polovica *grow*). Vjerojatnost da se na jedinku primjeni selekcija bila je 35%, križanje 60%, a mutacija 0.5%. Maksimalna dubina programa u inicijalnoj populaciji bila je 6, a u sljedećim generacijama 17 kako bi se izbjeglo nadimanje koje bi bespotrebno koštalo resursa. [4] Važno je napomenuti da podaci koji se koriste za testiranje moraju biti različiti od onih koji su korišteni za trening.

Numerical results for the short training period

Symbol	Training	Testing
<i>A</i>	144.16%	15.29%
<i>ABX</i>	202.16%	38.12%
<i>AL</i>	135.95%	15.79%
<i>BBD</i>	123.04%	-34.92%
<i>CBG</i>	174.01%	-75.09%
<i>CP</i>	120.70%	0.86%
<i>CU</i>	56.69%	7.88%
<i>CXY</i>	183.40%	-102.16%
<i>MOL</i>	99.23%	17.38%
<i>RY</i>	125.68%	3.25%
<i>SCC</i>	110.80%	-21.84%
<i>TMA</i>	211.93%	42.27%
<i>TRP</i>	85.78%	36.04%
<i>VO</i>	146.78%	-10.83%
Average	131.17%	-4.85%
Number of positive returns	14	9

Slika 7 - Rezultati dobiveni kratkim testiranjem

Numerical results for the long training period

Symbol	Training	Testing
<i>A</i>	31.00%	18.55%
<i>ABX</i>	34.27%	7.69%
<i>AL</i>	21.99%	19.83%
<i>BBD</i>	18.30%	-79.46%
<i>CBG</i>	33.15%	-35.58%
<i>CP</i>	19.67%	12.11%
<i>CU</i>	9.77%	6.45%
<i>CXY</i>	30.39%	-44.85%
<i>MOL</i>	19.36%	13.74%
<i>RY</i>	19.59%	8.13%
<i>SCC</i>	16.53%	-7.33%
<i>TMA</i>	30.57%	54.80%
<i>TRP</i>	16.57%	21.09%
<i>VO</i>	19.88%	-45.41%
Average	22.93%	-3.59%
Number of positive returns	14	9

Slika 8 - Rezultati dobiveni dužim testiranjem

Kao što je očekivano, rezultati dobiveni u trening periodu su mnogo bolji nego za testiranja. Štoviše, testiranje je dalo negativnu dobit (iako minimalnu), što bi značilo da bi u ovom slučaju bolji pristup bio jednostavno kupiti dionicu i držati je duže vrijeme (*buy-and-hold* pristup) nego tempiranje tržišta i pokušaja zarade na volatilnosti dionice. Kratki period treninga perioda je dao bolje rezultate nego duži, što se može objasniti manjim obujmom podataka. Kako u kratkom periodu nije bilo toliko podataka kao u dužem, genetsko programiranje je razvilo pravila trgovanja koja su previše vezana za nereprezentativni obim podataka.

5. ZAKLJUČAK

Iako genetsko programiranje možda neće odmah polučiti željene rezultate pri signaliziranju u trgovini dionicama, ono je ipak iznimno moćan alat koji daljnim optimiziranjem, razvijanjem i poboljšavanjem itekako može ostaviti traga i pospješiti učinkovitost trgovaca dionicama, što su mnoga istraživanja već potvrdila. Genetsko programiranje, kao i druge tehnologije bazirane na genetskim algoritmima poput neuronskih mreža, definitivno obećavaju i treba ih se dalje istraživati.

Pri istraživanju primjene genetskog programiranju u trgovini dionicama, treba obratiti pozornost na problem *overfitting-a*, dizajniranja sistema trgovanja ravnajući se prema povijesnim podacima umjesto da se identificiraju ponavljajuća ponašanja. Odabir parametara je bitan proces i treba uzimati one koji najbolje opisuju ponašanje neke dionice, tako da je važno istestirati simulaciju uzimajući nekoliko različitih indikatora.

6. LITERATURA

[1] Golub, M. Genetski algoritam: Prvi dio, skripta. 2004. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva. URL:

http://www.zemris.fer.hr/~golub/ga/ga_skripta1.pdf

[2] Poli R., Langdon W., McPhee N. A Field Guide to Genetic Programming. 2008. University of Essex, University of Minnesota. URL:

<https://pdfs.semanticscholar.org/755d/1c9d1f63d448c05ca96a871356c05f8b1526.pdf>

[3] Reid S. Using Genetic Programming to evolve Trading Strategies. 5. lipnja

2013. URL: <http://www.turingfinance.com/using-genetic-programming-to-evolve-security-analysis-decision-trees/>

[4] Potvin J.Y., Soriano P., Vallee M. Computers & Operations Research 31: Generating trading rules on the stock markets with genetic programming. 2004. str 1033-1047

[5] Poslovni.hr. Tehnička analiza. Zagreb. URL:

http://www.poslovni.hr/media/PostAttachments/650406/Tehnicka_analiza%5B1%5D.pdf

7. SAŽETAK

Genetsko programiranje je specijalizacija genetskih algoritama, odnosno vrsta evolucijskog računanja koje automatski rješava probleme bez da korisnik mora unaprijed znati ili specificirati formu ili strukturu željenog rješenja. Ono je ustvari algoritam koji u nekoliko koraka (inicijalizacija inicijalne populacije, određivanje dobrote jedinki u sljedećim generacijama, primjena genetskih operacija nad jedinkama) dolazi do idealnog rješenja. U ovom radu za problem kojemu treba naći rješenje uzeto trgovanje dionicama, odnosno dobivanje signala kada neku dionicu treba kupiti ili prodati. To se postiže razvojem trgovinskih strategija, a one su predstavljene programima (jedinkama) u genetskom programiranju. Osim aritmetičkih i booleanskih funkcija, u stvaranju tih programa su još korišteni i određeni indikatori koji se inače koriste u tehničkoj analizi tržišta.