

# Primjena genetskog programiranja u održavanju vodoopskrbnih sustava

---

**Kuzmić, Jurica**

**Master's thesis / Diplomski rad**

**2018**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:113651>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-24**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# **DIPLOMSKI RAD**

**Jurica Kuzmić**

Zagreb, 2018.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# DIPLOMSKI RAD

Mentori:

Prof. dr. sc. Dragutin Lisjak, dipl. ing.

Dr. sc. Anita Rakić, dipl. ing.

Student:

Jurica Kuzmić

Zagreb, 2018.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru prof.dr.sc. Dragutinu Lisjaku, na pruženom povjerenju i savjetima kao i posvećenom vremenu pri izradi ovog rada.

Također, zahvaljujem se dr.sc. Aniti Rakić, za pružanje informacija i podataka potrebnih za izradu modela na kraju ovog rada.

Naposljetku, zahvaljujem se svojoj obitelji čija mi je nepokolebljiva podrška i razumijevanje tijekom svih godina studiranja, dala snage za uspješni dovršetak studija.

Jurica Kuzmić



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za diplomske radove studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment,  
inženjerstvo materijala te mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	
Ur. broj:	

### DIPLOMSKI ZADATAK

Student: **JURICA KUZMIĆ** Mat. br.: 0035191236

Naslov rada na hrvatskom jeziku: **Primjena genetskog programiranja u održavanju vodoopskrbnih sustava**

Naslov rada na engleskom jeziku: **Application of genetic programming in maintenance of water supply systems**

Opis zadatka:

Ispitivanje zdravstvene ispravnosti vode za ljudsku potrošnju na izvorištu te iz vodoopskrbnih objekata i/ili iz vodoopskrbne mreže, kao i nadzor nad dezinfekcijom sirovih voda, osnovne su mjere za sprječavanje i suzbijanje širenja zaraznih bolesti. Dezinfekcija vode u cjevovodnom sustavu predstavlja niz postupaka, kojima je cilj smanjenje broja patogenih mikroorganizama koji samostalno i/ili u zajednici s fizikalno-kemijskim čimbenicima predstavljaju opasnost za zdravlje ljudi. Stoga je potrebno uklanjanje i onemogućavanje kolonizacije različitih vrsta patogenih mikroorganizama (*Legionella* spp) u vodoopskrbnom sustavu. Najčešći način održavanja cjevovodne mreže u cilju dezinfekcije vode kojim se uklanjaju patogeni organizmi je postupak kloriranja. Nakon kloriranja, u vodi ostane određena količina tzv. rezidualnog klora koja mora biti o određenim granicama kako previsoka količina ne bi negativno utjecala na ljudsko zdravlje. Hipoteza rada je da je moguće iznaći matematički model za procjenu potrebnog i rezidualnog klora na temelju podataka analize vode u cilju optimalnog sadržaja klora, a time ujedno i troškova održavanja cjevovoda. U skladu s navedenim, u radu je potrebno:

1. Opisati način održavanja vodoopskrbnih sustava.
2. Analizirati dostupne podatke analize vode te statistički dokazati korelaciju između količine mikroorganizama i utrošenog klora za dezinfekciju.
3. Primjenom genetskog programiranja kreirati matematički model za predviđanje utroška potrebnog klora za dezinfekciju kao i optimalne količine rezidualnog klora u cilju zaštite ljudskog zdravlja te smanjenja troškova održavanja cjevovoda.
4. Analizom dobivenih rezultata izvesti zaključke.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:  
27. rujna 2018.

Rok predaje rada:  
29. studenog 2018.

Predviđeni datum obrane:  
05. prosinca 2018.  
06. prosinca 2018.  
07. prosinca 2018.

Zadatak zadao:  
prof. dr. sc. Dragutin Lisjak

Komentor:  
dr. sc. Anita Rakić

Predsjednica Povjerenstva:  
prof. dr. sc. Biserka Runje

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	IV
POPIS TABLICA.....	V
POPIS OZNAKA .....	VI
SAŽETAK.....	VIII
SUMMARY .....	IX
1. UVOD.....	1
2. ODRŽAVANJE VODOOPSKRBNIH SUSTAVA .....	2
2.1. Izrada strategije upravljanja i održavanja .....	2
2.1.1. Plan upravljanja imovinom vodovoda .....	3
2.1.2. Dokument bilance vode u vodoopskrbnom sustavu .....	8
2.1.3. Plan očuvanja zdravstvene ispravnosti vode.....	9
2.2. Strategija održavanja.....	10
2.2.1. Proaktivno održavanje .....	14
2.2.1.1. Analiza temeljnih uzroka kvarova .....	14
2.2.1.2. Analiza načina i posljedica kvarova .....	15
2.2.1.3. Održavanje usmjereno ka pouzdanosti .....	17
2.3. Cjelovitost vodoopskrbnog sustava .....	17
2.3.1. Osnovni elementi sustava.....	17
2.3.2. Zahtjev fizičke cjelovitosti.....	19
2.3.2.1. Osnovni principi održavanja fizičke cjelovitosti .....	21
2.3.3. Zahtjev hidrauličke cjelovitosti.....	22
2.3.3.1. Osnovni principi održavanja hidrauličke cjelovitosti .....	24
2.3.4. Zahtjev kvalitativne cjelovitosti.....	24
2.3.4.1. Osnovni principi održavanja kvalitativne cjelovitosti .....	28
2.4. Kondicioniranje vode .....	29
2.4.1. Koagulacija i pahuljičenje.....	31
2.4.2. Taloženje.....	31
2.4.3. Flotacija.....	31
2.4.4. Filtriranje.....	31
2.4.5. Kloriranje .....	31
2.5. Najčešće operacije u održavanju po komponentama sustava .....	33
2.5.1. Cijevi.....	33
2.5.1.1. Lociranje cijevi i ostalih komponenti .....	33
2.5.1.2. Lociranje pukotina u mreži .....	34
2.5.1.3. Popravljanje ili zamjena napuknutih cijevi.....	35
2.5.2. Ventili i ostali elementi sustava .....	36
2.5.2.1. Održavanje izolacijskih ventila.....	37
2.5.2.2. Održavanje zračnih ventila .....	37
2.5.2.3. Održavanje ispusnih ventila.....	38
2.5.2.4. Održavanje nepovratnih ventila .....	38
2.5.2.5. Održavanje kontrolnih ventila.....	38

2.5.2.6.	Održavanje hidranta .....	38
2.5.2.7.	Održavanje vodomjera .....	38
2.5.2.8.	Održavanje oznaka .....	39
2.5.3.	Vodospremnici .....	39
2.5.3.1.	Održavanje vodospremnika .....	39
2.5.4.	Pumpe .....	40
2.5.4.1.	Održavanje pumpi .....	40
2.5.5.	Uređaji za dezinfekciju vode .....	41
2.5.5.1.	Održavanje uređaja za dezinfekciju klorom .....	41
2.5.5.2.	Održavanje uređaja za UV zračenje .....	41
2.6.	Mogućnosti primjene strojnog učenja u održavanju .....	42
2.6.1.	Primjena genetskog programiranja u održavanju vodoopskrbnih sustava .....	43
3.	OSNOVNO O GENETSKOM PROGRAMIRANJU .....	44
3.1.	Začetak .....	44
3.2.	Definicija GP-a .....	45
3.3.	Prikaz u obliku stabla .....	45
3.4.	Princip rada GP-a .....	46
4.	GPTIPS – PLATFORMA ZA GENETSKO PROGRAMIRANJE I SIMBOLIČKO RUDARENJE PODATAKA U MATLAB-U .....	48
4.1.	Uvodne napomene o GPTIPS-u .....	48
4.2.	Simboličko rudarenje podataka u GPTIPS-u .....	48
4.2.1.	Simbolička regresija .....	49
4.2.2.	Simbolička klasifikacija .....	50
4.2.3.	Simbolička optimizacija .....	51
4.2.4.	Multi-genska simbolička regresija .....	52
4.2.5.	Procjena koeficijenata kod multi-genskog GP-a – Linearna metoda najmanjih kvadrata .....	52
4.3.	Multi-gensko genetsko programiranje (MGGP) .....	55
5.	UPOTREBA GPTIPS-A U MATLABU .....	56
5.1.	Učitavanje skupova podataka i definiranje seta operanada .....	57
5.2.	Definiranje seta operatora .....	59
5.3.	Temeljni preduvjeti za definiranje primitivnog seta .....	60
5.3.1.	Potpunost .....	60
5.3.2.	Dostatnost .....	61
5.4.	Rad s multi-genskim pojedincima u GPTIPS-u .....	62
5.5.	Definiranje funkcije pogodnosti .....	62
5.6.	Upravljački parametri za izvršavanje GP algoritma u GPTIPS-u .....	65
5.6.1.	Veličina populacije .....	65
5.6.2.	Maksimalni broj generacija .....	66
5.6.3.	Maksimalna dopuštena dubina stabla .....	67
5.6.4.	Metode stvaranja inicijalne populacije .....	67
5.6.4.1.	Full metoda .....	68
5.6.4.2.	Grow metoda .....	69
5.6.4.3.	Ramped Half-and-Half metoda .....	71
5.6.5.	Metode selekcije pojedinaca .....	71
5.6.5.1.	Standardna turnirska selekcija .....	72
5.6.5.2.	Pareto turnirska selekcija .....	72
5.6.5.3.	Leksikografska turnirska selekcija .....	73

---

5.6.5.4. Elitizam .....	73
5.6.6. Genetske operacije .....	73
5.6.6.1. Križanje .....	73
5.6.6.2. Mutacija .....	76
5.6.6.3. Reprodukcijska .....	79
6. KREIRANJE MATEMATIČKOG MODELA ZA PREDVIĐANJE REZIDUALNOG KLORA PRIMJENOM GP-a .....	80
6.1. Opis problema .....	80
6.2. Eksperimentalni podaci .....	81
6.3. Dobivanje inicijalnog modela .....	82
6.4. Povećanje skupa podataka primjenom Monte Carlo simulacije .....	86
6.5. Dobivanje konačnog modela .....	87
7. ZAKLJUČAK .....	95
LITERATURA .....	97
PRILOZI .....	100



## POPIS SLIKA

Slika 1.	Primjer hijerarhijske podjele imovine na 5 razina [3].....	5
Slika 2.	Životni ciklus upravljanja imovinom [3].....	6
Slika 3.	Primjer tjednog rasporeda održavanja u obliku izvještaja [2].....	12
Slika 4.	Dijagram toka kod otklanjanja kvarova u korektivnom održavanju [4] .....	13
Slika 5.	Pitanja u svakom koraku provođenja analize temeljnih uzroka kvarova [6] .....	14
Slika 6.	Pojednostavljena ilustracija izgleda vodoopskrbnog sustava [9] .....	18
Slika 7.	Prikaz nadležnosti i važećih zakona u provođenju nadzora izvorišta i vode za piće u RH [12].....	27
Slika 8.	Postotak kemijski i mikrobiološki neispravnih uzoraka vode po županijama u RH [13] .....	28
Slika 9.	Shema uobičajenog postupka kondicioniranja vode [14].....	30
Slika 10.	Sustav za kemijsku dezinfekciju vode, izvor rijeke Rude [7] .....	32
Slika 11.	WEDECO UV uređaji, Vodoopskrbni sustav Pakrac – Lipik, Vodozahvat Šumetlica [7] .....	33
Slika 12.	Dijagram karakteristike pumpe [2].....	40
Slika 13.	Način prikaza u obliku stabla .....	46
Slika 14.	Princip rada standardnog GP algoritma.....	47
Slika 15.	Skup podataka nazvanih <code>training_set</code> .....	57
Slika 16.	Skup podataka nazvanih <code>testni_set</code> .....	58
Slika 17.	Princip izgradnje stabla kod Full metode inicijalizacije .....	68
Slika 18.	Princip izgradnje stabla kod Grow metode inicijalizacije.....	70
Slika 19.	Postupak kod operacije križanja pod-stabala .....	74
Slika 20.	Princip rada križanja više razine uz $G_{max} = 5$ .....	76
Slika 21.	Princip rada operacije mutacije pod-stabala.....	78
Slika 22.	Kontrolna ploča u postrojenju za kemijsku dezinfekciju vode s vrijednošću SRK u vodoopskrbnoj mreži [7] .....	80
Slika 23.	Usporedni prikaz svih predviđanja i traženih vrijednosti inicijalnog modela na trening skupu podataka.....	85
Slika 24.	Usporedni prikaz predviđanja i traženih vrijednosti inicijalnog modela na testnom skupu podataka .....	85
Slika 25.	Osnovni statistički pokazatelji osnovnog skupa.....	86
Slika 26.	Dijagram tijeka Monte Carlo simulacije izrađene za povećanje skupa podataka .....	87
Slika 27.	Geni od kojih se sastoji konačni model.....	88
Slika 28.	Raspršeni dijagram pogađanja konačnog modela na trening skupu podataka .....	89
Slika 29.	Raspršeni dijagram pogađanja konačnog modela na testnom skupu podataka....	90
Slika 30.	Sve predikcije i tražene vrijednosti za konačni model na trening skupu podataka .....	90
Slika 31.	Sve predikcije i tražene vrijednosti za konačni model na testnom skupu podataka .....	91
Slika 32.	Usporedba stvarnih i predviđenih vrijednosti – trening skup.....	92
Slika 33.	Usporedba stvarnih i predviđenih vrijednosti – testni skup .....	93

## POPIS TABLICA

Tablica 1. Faze i koraci razvoja plana upravljanja imovinom [2].....	3
Tablica 2. Matrica rizika održavanja [3] .....	7
Tablica 3. Kategorije bilance vode prema IWA metodologiji [2].....	8
Tablica 4. Koraci u izradi plana očuvanja zdravstvene ispravnosti vode [2].....	9
Tablica 5. Preporučena učestalost pojedinih operacija održavanja [2] .....	10
Tablica 6. Primjer analize temeljnog uzroka kvara pumpe u crpnoj stanici .....	15
Tablica 7. Primjer provođenja metode analize načina i posljedica kvara centrifugalne pumpe [4] .....	16
Tablica 8. Utjecajni faktori na fizičku cjelovitost vodoopskrbnog sustava [2].....	19
Tablica 9. Statistički podaci o količinama prodane i izgubljene vode u RH [10].....	21
Tablica 10. Utjecajni faktori na hidrauličku cjelovitost kod vodoopskrbnih sustava [2].....	23
Tablica 11. Utjecajni faktori na kvalitativnu cjelovitost vodoopskrbnog sustava [2].....	24
Tablica 12. Preporuka za korištenje tehnika lociranja pukotina u ovisnosti o promjeru cijevi [2] .....	35
Tablica 13. Dio prikupljenih podataka za modeliranje .....	82
Tablica 14. Parametri za rad GP algoritma pri dobivanju inicijalnog modela .....	83
Tablica 15. Performanse inicijalnog modela na trening skupu podataka.....	84
Tablica 16. Performanse inicijalnog modela na testnom skupu podataka .....	84
Tablica 17. Performanse konačnog modela na trening skupu podataka .....	88
Tablica 18. Performanse konačnog modela na testnom skupu podataka .....	88
Tablica 19. Usporedba predviđanja konačnim modelom i stvarnih mjerenja sadržanih u osnovnom skupu podataka .....	92
Tablica 20. Performanse konačnog modela na osnovnom trening skupu podataka.....	93
Tablica 21. Performanse konačnog modela na osnovnom testnom skupu podataka .....	94

## POPIS OZNAKA

Oznaka	Jedinica	Opis
$b_0$	/	Koeficijent pristranosti (engl. <i>bias term</i> )
$b_1$	/	Koeficijent težine (engl. <i>weight term</i> )
$\beta$	/	Vektor nepoznatih parametara
$C(j)$	/	Ciljna izlazna vrijednost za slučaj pogodnosti $j$
$C_j$	s	Vrijeme dovršetka $j$ -tog posla
D	/	Matrica modelom predviđenih odziva
d	/	Trenutna dubina stabla
$d_j$	s	Predviđeno vrijeme dovršetka $j$ -tog posla
$d_{max}$	/	Maksimalna dopuštena dubina stabla
$e$	/	Vektor ostataka tj. pogreške
err	/	Matrica iznosa pogreški predviđanja
$\varepsilon$	/	Vektor neprimijećenih smetnji
F	s	Prosječno vremensko kašnjenje
$F_{max}$	s	Maksimalni vremenski protok
$f_j$	s	Vrijeme protoka $j$ -tog posla koji se računa pomoću: $C_j - r_j$
$f_R$	/	Sirova pogodnost
$f_R^{max}$	/	Maksimalni mogući iznos sirove pogodnosti (pretpostavlja se da je taj iznos poznat)
$f_s$	/	Standardizirana pogodnost
G	/	Broj članova koji tvore MGGP model
L	/	Broj poslova koji kasne (skup u kojem vrijedi $C_j - d_j > 0$ )
MSE	/	Prosječna kvadrirana pogreška (engl. <i>Means Squared Error</i> )
N	/	Broj opservacija odzivne (ciljne) varijable
P	/	Veličina populacije
PK	/	Parametar križanja
R	/	Broj vrtnji GP algoritma
$R^2$	/	Koeficijent determinacije
RMSE	/	Korijen prosječne kvadrirane pogreške (engl. <i>Root Mean Squared Error</i> )
RPN	/	Brojevi prioriteta rizika (engl. <i>Risk Priority Numbers</i> )

$r_j$	s	Vrijeme koje je potrebno proteći kako bi se j-ti posao mogao započeti
S	/	Prosječna veličina modela
$S(i, j)$	/	Vrijednost pogodnosti dobivena evaluacijom pojedinca $i$ na slučaju pogodnosti $j$
SSE	/	Suma kvadriranih pogreški (engl. <i>Sum of Squared Errors</i> )
T	s	Prosječno kašnjenje
TWT	s	Ukupno težinsko kašnjenje
$T_{max}$	s	Maksimalno kašnjenje
t	s	Vrijeme potrebno za rad GP algoritma
X	/	Matrica odzivnih vrijednosti svih stabala od kojih se multi-genski pojedinac sastoji
$x_1$	mg/L	Dodani klor
$x_2$	CFU x 10 <sup>-2</sup> / 1 mL	Ostale bakterije
$x_3$	broj/100 mL	Ukupni koliformi
$x_4$	broj/100 mL	Enterokoki
$x_5$	broj/100 mL	Escherichia coli
y	/	Traženi odziv/izlaz
$y_1$	mg/L	Rezidualni klor
$\hat{y}$	/	Odziv/izlaz kojeg je model predvidio
$\omega_j$	/	Težina j-tog posla

## SAŽETAK

U prvom dijelu ovog rada, opisan je pristup razvoju strategija održavanja vodoopskrbnih sustava sa aspekta upravljanja imovinom. Dan je pregled najčešćih operacija održavanja (općenito i po komponentama) potrebnih za ispunjavanje tri osnovna zahtjeva svih vodoopskrbnih sustava, a to su: fizička, hidraulička i kvalitativna cjelovitost. Kao najčešći novi pristup u održavanju modernih vodoopskrbnih mreža koristi se proaktivno održavanje tj. pripadajuće metode koje podrazumijevaju analizu temeljnih uzroka nastanka kvarova. Kod opisa tih metoda izneseni su konkretni primjeri njihove primjene. Nadalje, drugi dio rada bavi se mogućim primjenama algoritma genetskog programiranja na održavanje vodoopskrbnih sustava s naglaskom na održavanje kvalitativne cjelovitosti. Najprije je razrađen princip rada GP algoritma te je objašnjen način upotrebe GPTIPS skupine alata za MATLAB. Temeljem podatak prikupljenih na lokalitetu rijeke Jadro, pomoću GPTIPS-a, izrađen je model za predviđanje rezidualnog klora u vodoopskrbnoj mreži, kao ključnog pokazatelja zdravstvene ispravnosti vode za piće. Takav model omogućuje djelatnicima vodovoda da optimiziraju doziranje klora, čime se štedi novac i poboljšava razina usluge.

Ključne riječi: *plan upravljanja imovinom, strategije održavanja vodoopskrbnih sustava, proaktivno održavanje, analiza temeljnih uzroka kvarova, analiza načina i posljedica kvarova, održavanje usmjereno ka pouzdanosti, najčešće operacije održavanja vodoopskrbnih sustava, genetsko programiranje (GP), multi-gensko genetsko programiranje, simbolička regresija, GPTIPS, MATLAB, model predviđanja rezidualnog klora, Monte Carlo simulacija.*

## SUMMARY

First part of this thesis deals with development of maintenance strategies for water distribution systems. An overview of common maintenance procedures (in general and by certain components) needed to fulfill three basic requirements of all water distribution systems: physical, hydraulic and quality integrity, is given. As the most commonly used new approach to maintaining modern water distribution systems, proactive maintenance implies methods for analysis of the root cause of failure. In description of these methods, concrete examples of their applications are also presented. Furthermore, the second part of this thesis deals with possible applications of genetic programming (GP) algorithm for maintaining water supply systems with emphasis on maintaining a qualitative integrity. Firstly, core principles of GP were elaborated and then usage of GPTISP toolbox for MATLAB was explained. Based on the data collected at the site of the Jadro river, a model for prediction of residual chlorine, as a key indicator for water health safety, was developed using GPTIPS. Such model allows water supply workers to optimize chlorine dosing, saving money and improving service levels.

Key words: *asset management plan, water distribution systems maintenance strategies, proactive maintenance, root cause analysis, failure modes analysis, reliability centered maintenance, common maintenance procedures in water distribution systems, genetic programming (GP), multi-gene genetic programming, symbolic regression, GPTIPS, MATLAB, residual chlorine prediction model, Monte Carlo simulation.*

## 1. UVOD

Voda je oduvijek imala ključnu ulogu u razvoju života na zemlji, pa tako i čovječanstva kakvog danas poznajemo. Još od najranijih dana, kada je čovjek usvojio sjedilački način života, bila je najvažniji resurs te preduvjet napretka. S postepenim razvojem poljoprivrede, razvijaju se i naselja koja sve više počinju ovisiti o izvorima vode i njenoj kvaliteti. S obzirom na tada niska raspoloživa znanja kada je u pitanju liječenje bolesti, štetni organizmi ili patogeni u vodi, postaju ozbiljna prijetnja dobrobiti svih ljudi. S porastom broja ljudi u naseljima, uviđa se potreba za pomnijim planiranjem sustava vodoopskrbe i odvodnje. Prvi gradovi radili su se u blizini prirodnih izvora, te su ljudi tamo gradili brojne bunare kako bi snabdijevali svoje stanovništvo. Kasnije, s razvojem tehnologije, u drevnom Rimu se koriste akvadukti, koji su sposobni prenositi pitku vodu na mnogo udaljenija mjesto od njenog izvora. Također u to doba, javljaju se prvi naputci kada je u pitanju kvaliteta vode, pa se ona tako počinje razmatrati prema nekim osnovnim parametrima. Voda koja nije imala okus, miris, zamućenost te je bila hladna, smatrala se najkvalitetnijom vodom za piće. Danas uvjeti kvalitete i zdravstvene ispravnosti vode nisu tako jednostavni. S prolaskom godina i dolaskom u današnje moderno doba, razvoj tehnologija vodoopskrbe i odvodnje nije posustao nego je, moglo bi se reći, postao još veći i važniji nego ikada prije. Trenutno dostupne tehnologije, materijali i znanja omogućuju nam izgradnju znatno naprednijih, a samim time i kompleksnijih sustava, koje koristi veliki broj ljudi. Ti sustavi moraju ispunjavati sva očekivanja u vidu kvantitete i kvalitete vode koju distribuiraju i to u okruženju rastućih problema koji ih opterećuju u današnje moderno doba. Da bi se očekivanja ispunila, potrebne su kvalitetne strategije održavanja i upravljanja, koje se rabe najboljim dostupnim znanjima i iskustvima. Ovaj rad napisan je upravo u tu svrhu, pa će se u njemu navesti moderne prakse u održavanju vodoopskrbnih sustava, koje se danas smatraju integralnim dijelom strategije upravljanja. Danas je jasno da je održavanje vodoopskrbne mreže izravno povezano sa očuvanjem zdravstvene ispravnosti vode, pa djelatnici održavanja moraju surađivati i sa ostalim strukama kako bi se osiguralo da je uvjet kvalitete vode ispunjen. U tu svrhu moguća je primjena raznih metoda strojnog učenja poput tehnike genetskog programiranja (GP-a), što će biti demonstrirano ovim radom, u kojem će primjenom softverskog paketa MATLAB biti kreiran matematički model za predviđanje rezidualnog klora u distribucijskom sustavu.

## 2. ODRŽAVANJE VODOOPSKRBNIH SUSTAVA

### 2.1. Izrada strategije upravljanja i održavanja

S ekonomskim razvojem i urbanizacijom, povećava se opterećenje nad sve manje dostupnim izvorima pitke vode i sustavima njene distribucije. Rastući utjecaj industrije, divlja gradnja, sve raširenija upotreba pesticida i herbicida u poljoprivredi, klimatske promjene, loše strategije gospodarenja otpadom te slaba osviještenost i educiranost ljudi kada je u pitanju zaštita okoliša, samo su neki od problema s kojima se vodoopskrbni sustavi danas bore [1]. Na spomenute probleme, u razvijenim zemljama izravno se nadovezuje problem starenja postojećih mreža distribucije zbog kojeg se javljaju značajni vodni i novčani gubici. S druge strane, zemlje u razvoju se bore sa nemogućnošću zadovoljavanja vlastitih potreba pa je kod njih opskrba pitkom vodom isprekidana ili, najčešće, u potpunosti nedostupna [1, 2]. Kada se sve navedeno uzme u obzir, važnost razvoja kvalitetnih i efikasnih strategija upravljanja i održavanja je očigledna i to ne samo kod postojećih sustava već i kod onih u nastanku. Vrlo ih je važno početi razvijati već u najranijim fazama planiranja izgradnje ili proširenja, kako bi se kasnije osigurala dobra podloga za dopunjavanje i izradu konačnih strategija u tijeku eksploatacije. Njihovo prekasno razvijanje, nedostatak ili neadekvatnost može dovesti do potpunog kolapsa te potrebe za velikim novim kapitalnim investicijama kako bi se postojeći sustav zamijenio novim [2]. Općenito, dobra strategija upravljanja trebala bi sadržavati sve informacije potrebne za praćenje, rad, održavanje, nadogradnju i raspolaganje sustavom vodoopskrbe na ekonomičan način uz ispunjenje svih procedura i propisa [3]. Unutar te definicije, strategija održavanja jedna je od najvažnijih točki. Ona podrazumijeva sve aktivnosti nužne za utvrđivanje, očuvanje i ponovno uspostavljanje nazivnog stanja sustava [4]. Navedene strategije uspješno se izrađuju pomoću slijedećih planova i dokumenata koje literatura preporuča u tu svrhu, a to su [2]:

- **plan upravljanja imovinom** (engl. *asset management plan*)
- **bilanca vode** (engl. *water balance*)
- **plan očuvanja zdravstvene ispravnosti vode** (engl. *water safety plan*)

O svakom od njih više će biti rečeno u narednim poglavljima, gdje će biti opisan način njihove izrade po koracima, kao i njihova važnost u kontekstu sustava vodoopskrbe u Hrvatskoj.



### 2.1.1. Plan upravljanja imovinom vodovoda

Plan upravljanja imovinom vodovoda definira skup koraka koje donositelj odluka može slijediti pri kupovini, izradi, održavanju, zamjeni ili uklanjanju neke imovine ili sredstava kako bi to proveo na troškovno učinkovit način koji osigurava kontinuitet i dugotrajnost pružanja tražene usluge [2]. Kod javnih komunalnih poduzeća kao što je to vodovod, koristi se poseban pristup upravljanja imovinom koji nosi naziv upravljanje infrastrukturnom imovinom (engl. *infrastructure asset management*) [2,3]. Takav pristup temelji se na planiranim intervencijama u ključnim periodima u životnom ciklusu pojedinog imovinskog sredstva, kako bi mu se produljio očekivani životni vijek, a samim time i održala njegova očekivana djelotvornost [3]. Razvoj ovog plana moguće je podijeliti u nekoliko ključnih faza i pripadajućih koraka, kao što je to prikazano slijedećom tablicom.

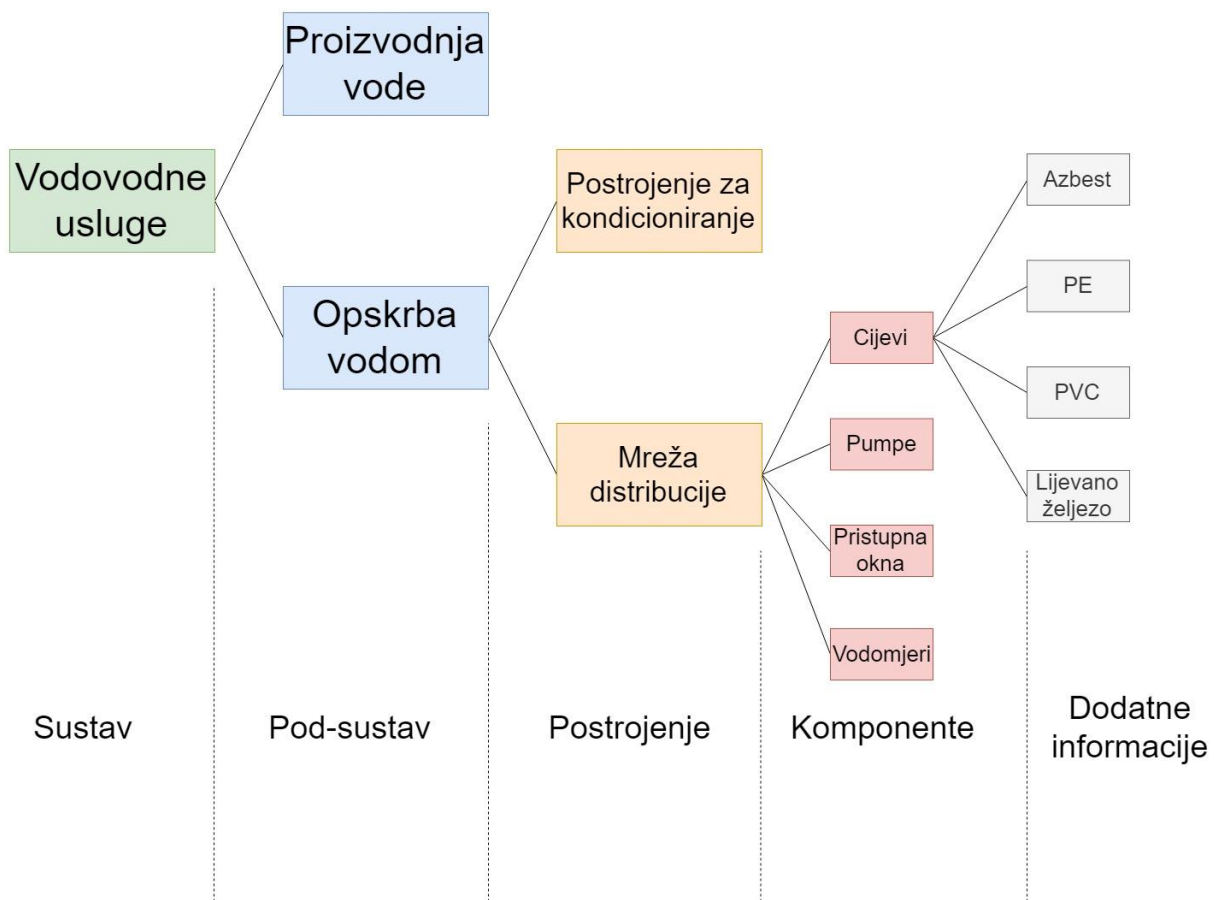
**Tablica 1. Faze i koraci razvoja plana upravljanja imovinom [2]**

Faza	Korak	Pitanja na koja treba odgovoriti	Što podrazumijeva?
<b>Tehnička procjena</b>	1.Registar imovine	<ul style="list-style-type: none"> <li>• Kojom imovinom se raspolaže?</li> <li>• Gdje se ta imovina nalazi?</li> <li>• Koliko vrijedi?</li> </ul>	<ul style="list-style-type: none"> <li>• Stvaranje i popunjavanje registra imovine</li> </ul>
	2.Procjena stanja	<ul style="list-style-type: none"> <li>• Koje je stanje imovine?</li> <li>• Koja imovina je kritična za rad?</li> <li>• Kakva je vijek trajanja imovine?</li> </ul>	<ul style="list-style-type: none"> <li>• Procjena stanja sustava i određivanje preostalog životnog vijeka</li> </ul>
<b>Financijska procjena</b>	3.Trenutne i buduće potrebe	<ul style="list-style-type: none"> <li>• Potrebe u životnom vijeku imovine?</li> <li>• Investicijske potrebe?</li> <li>• Novčane potrebe za rad?</li> </ul>	<ul style="list-style-type: none"> <li>• Procjena razine usluge i potražnje koju je potrebno zadovoljiti</li> </ul>
	4.Analiza troškova i rizika	<ul style="list-style-type: none"> <li>• Gubitak vrijednosti imovine s vremenom?</li> <li>• Trošak životnog ciklusa imovine?</li> <li>• Analiza rizika?</li> </ul>	<ul style="list-style-type: none"> <li>• Procjena očekivanih troškova u životnom vijeku sustava te identifikacija poslovnih rizika</li> </ul>
<b>Procedure upravljanja imovinom</b>	5.Operativni plan	<ul style="list-style-type: none"> <li>• Organizacijski razvojni plan?</li> <li>• Obuka i razvoj?</li> <li>• Postupci u hitnim slučajevima?</li> </ul>	<ul style="list-style-type: none"> <li>• Izrada detaljnih planova upravljanja i održavanja, procjene kapitalnih investicija, modela budućih troškova i financiranja te plana upravljanja imovinom koji u obzir uzima raspoloživa financijska sredstva</li> </ul>
	6.Plan održavanja	<ul style="list-style-type: none"> <li>• Analiza vremenskih razmaka između operacija održavanja?</li> <li>• Rutinsko i dugoročno održavanje?</li> </ul>	

Postupak iznesen u tablici, definiralo je Ministarstvo vodnih poslova Južnoafričke Republike u suradnji s njihovim Institutom regionalnog razvoja te on može poslužiti bilo gdje kao pomagalo za izradu planova upravljanja imovinom. U razvoj naputaka kako što bolje implementirati strategije upravljanja imovinom u vodnom sektoru, Južnoafrička Republika krenula je kako bi se uhvatila u koštac sa rastućim problemom lošeg stanja vodnih usluga, zbog čega su čak i zakonom propisali obvezu njihovog razvoja i implementacije [2]. Oni međutim, nisu jedinstveni po tom pitanju, već samo slijedi trendove proizašle iz država s visokom razinom provedbe upravljanja imovinom u vodnom sektoru poput Australije, Novog Zelanda, Ujedinjenog Kraljevstva i Portugala [3]. Republika Hrvatska također je propisala obvezu donošenja tih planova Zakonom o upravljanju i raspolaganju imovinom u vlasništvu Republike Hrvatske [5]. S obzirom da se zakoni o vlasništvu Republike Hrvatske istovjetno odnose i na vlasništva jedinica lokalne samouprave pod čiju ingerenciju spadaju javni vodovodi i vodoopskrbni sustavi, proizlazi da i vodovodi imaju obvezu razvitka planova upravljanja imovinom. U prvoj fazi tehničke procjene, identificiraju se imovinska sredstva za koja je poduzeće (vodovod) odgovorno te se prikupljaju svi pripadajući podaci vezni uz tu imovinu [2]. Njih se prikuplja na jedan od slijedećih načina [3]: inspekcijom na terenu, fotografijama, videozapisima, temeljem projekata i studija (izvedbeni nacrti, projektni nacrti), priručnika dobavljača/proizvođača, podacima prikupljenim tijekom redovnog održavanja te podacima prikupljenim od osoblja održavanja (iskustvo). Nakon toga, stvaraju se kategorije kojima se dodjeljuju raspoloživa imovinska sredstva po hijerarhijskom principu [3]. Hijerarhija bi se trebala sastojati od koliko god je moguće razina, ali se općenito može definirati pet osnovnih kao slijedi [2]:

1. sustav (npr. vodovodne usluge, cestovna infrastruktura, opskrba energijom)
2. pod-sustav (npr. proizvodnja vode, opskrba vodom)
3. postrojenje (npr. postrojenje za kondicioniranje vode, mreža distribucije, pumpna stanica)
4. komponenta (npr. pumpa, cijev, ventil, vodomjeri, pristupna okna itd.)
5. informacija o komponenti (npr. vrsta materijala, promjer i sl.)

Pojednostavljeni primjer spomenute hijerarhijske podjele imovinskih sredstava vodovoda u 5 razina, prikazan je na sljedećoj slici.



Slika 1. Primjer hijerarhijske podjele imovine na 5 razina [3]

Svakom imovinskom sredstvu dodjeljuje se neka oznaka koja sadrži osnovne informacije o njegovoj poziciji u hijerarhiji, fizičkoj lokaciji i veličini. Nakon toga se sve te informacije skupljaju na jednu lokaciju u tzv. registar imovine. U idućem koraku slijedi upisivanje starosti te ocjenjivanje stanja svakog imovinskog sredstva pomoću unaprijed definirane skale. Ocjenjuje se opće stanje sredstva, razina oštećenja te stupanj utjecaja nekog kvara na mogućnost da imovinsko sredstvo ispuni svoju zadaću (nazivnu i sigurnosnu) [2]. U svrhu te procjene najčešće se provode vizualni pregledi, pa osoblje na temelju njih i vlastitog iskustva procjenjuje koju ocjenu na definiranoj skali upotrijebiti, međutim, također postoji i niz analognih i digitalnih metoda testiranja koje mogu dati bolje rezultate. U vodovodnoj distribucijskoj mreži moguće je pomoću raznih instrumenata i senzora mjeriti npr. tlak i curenje u cijevima, kod pumpa se može mjeriti tlak i protok kako bi se procijenila radna karakteristika itd. [3]. Tako prikupljeni podaci posebno su korisni za analize u održavanju pomoću računala, gdje je rezultate mjerenja moguće primijeniti kako bi se stvorile znatno

kompleksnije skale za ocjenjivanje koje posjeduju faktore težine i važnosti, što rezultira vrlo preciznim rezultatima procjene ali zahtjeva ulaganje većih novčanih sredstava. Završetkom faze tehničke procjene ulazi se u fazu one financijske u kojoj se imovinska sredstva procjenjuju po raznim parametrima kao što su: vrijednost, koristan životni vijek, troškovi rada, investicijske potrebe te razina rizika koji je vezan uz njih. Vrijednost imovinskog sredstva može biti izražena kao trenutna (plaćena) vrijednost ili kao vrijednost jednaka novčanom iznosu potrebnom da bi se to sredstvo zamijenilo novim. Pri procjeni je također važno u obzir uzeti činjenicu da imovina linearno s vremenom gubi vrijednost, pa se procjenjuje i njena očekivana vrijednost na kraju životnog vijeka. Unutar životnog vijeka postoji jedan vremenski period koji se zove korisni životni vijek imovine. To je vremenski period, u kojem ona može ispunjavati svoju svrhu uzevši u obzir postepeno prirodno opadanje njene funkcije u tijeku eksploatacije [2]. On se definira temeljem procjena, proračuna ili projektnog životnog vijeka od strane izvođača radova ili proizvođača. Kod određivanja troškova, u obzir se uzimaju procijenjene troškova nad imovinom u njezinom čitavom životnom ciklusu. Oni obuhvaćaju sve očekivane troškove izgradnje ili nabave, stavljanja imovine u funkciju, procjenu njenih performansi i stanja, upravljanja njome te njenog održavanja [3]. Opisani životni ciklus upravljanja imovinom prikazan je slijedećom slikom.



Slika 2. Životni ciklus upravljanja imovinom [3]

Prikazani ciklus podrazumijeva da se plan upravljanja imovinom konstantno revidira i poboljšava temeljem uočenog stanja u tijeku eksploatacije imovine. Na kraju faze financijske procjene, vrši se identifikacija rizika od kvara svakog imovinskog sredstva. Takva analiza vrši se kako bi se odredili prioriteta u planu održavanja, što će omogućiti planiranje koje bolje odgovara potrebama ali i financijskim mogućnostima vlasnicima imovine. Najkritičnijom imovinom smatra se ona sa najvećim posljedicama ako se pokvari u kombinaciji sa najvećom vjerojatnošću da do kvara dođe. Koja je to točno imovina, određuje se pomoću matrice rizika imovine koja je prikazana na slici. Prikazana matrica rizika dobije se množenjem ocjene vjerojatnosti kvara nekog imovinskog sredstva sa ocjenom posljedica. Za obje mjere koje se množe obično se koristi neka jednostavna skala, npr. od 1 do 5 kao što je to slučaj u danom primjeru. U njoj bijela polja obilježavaju nizak rizik, zelena su prihvatljiv, a crvena neprihvatljiv rizik.

**Tablica 2. Matrica rizika održavanja [3]**

Pomnoženo		Posljedica (trošak) kvara				
		1	2	3	4	5
Vjerojatnost kvara	1	1	2	3	4	5
	2	2	4	6	8	10
	3	3	6	9	12	15
	4	4	8	12	16	20
	5	5	10	15	20	25

Pri određivanju ocjene vjerojatnosti kvara u obzir se uzimaju [3]: starost imovinskog sredstva, trenutno stanje imovine, povijest kvarova te opća iskustva. Što se tiče određivanja ocjene posljedice kvara u obzir se uzimaju slijedeći parametri [3]: trošak popravka, socijalni troškovi vezani uz gubitak funkcije imovine (npr. nezadovoljstvo korisnika), troškovi kolateralne štete, ekološki trošak te trošak smanjene razine usluge. Nakon što se uspješno izvedu sve navedene faze i njihovi pripadajući koraci, ulazi se u posljednju fazu uspostavljanja samih procedura upravljanja imovinom. Ta faza podrazumijeva razvoj planova upravljanja i održavanja koji se temelje na svim prethodno prikupljenim podacima i iskustvima. O samom planu održavanja biti će više rečeno u nastavku rada.

### 2.1.2. Dokument bilance vode u vodoopskrbnom sustavu

Bilanca vode je dokument koji služi kako bi pružio detaljni uvid u način na koji se voda u distribucijskom sustavu troši. Za potrebe izrade, procjenjuju se godišnje količine vode koje pripadaju nekoj od kategorija. Kategorije su unaprijed propisane od strane Međunarodnog društva za zaštitu voda (engl. *International Water Association*), na način prikazan u tablici [2].

**Tablica 3. Kategorije bilance vode prema IWA metodologiji [2]**

Ukupna ulazna količina vode	Dozvoljena potrošnja	Naplaćena dopuštena potrošnja	naplaćena izmjerena potrošnja (uključujući izvezenu vodu)	Prihodovna voda
			naplaćena neizmjerena potrošnja	
		Nenaplaćena dopuštena potrošnja	nenaplaćena izmjerena potrošnja	Ne-prihodovna voda
			nenaplaćena neizmjerena potrošnja	
	Gubici vode	Prividni gubici	nedozvoljena potrošnja	
			pogreške u mjerenju	
		Stvarni gubici	curenje u magistralnoj i/ili distributivnoj mreži	
			curenje ili prelijevanje u vodospremnici curenje u mreži do točke mjerenja kod kupaca	

Dozvoljena potrošnja obuhvaća količinu vode koja je dostavljena potrošačima koji su ovlaštenim za njenu upotrebu (kućanstva, komercijala ili industrija). Ta voda se može podijeliti na onu koja prolazi i onu koja ne prolazi kroz vodomjer. U kategoriju dozvoljene potrošnje također spada voda koja se izvozi u drugi sustav, voda za vatrogasce, ispiranje itd.. Kategoriju gubitka vode moguće je izračunati na način da se od ukupne ulazne količine vode oduzme dozvoljena potrošnja. U gubitke vode spadaju stvarni i prividni gubici. Stvarni gubici uključuju vodu koja je izgubljena kroz sve vrste curenja, puknuća i prelijevanja. Prividni gubici sastoje se od količine vode za koju se drži da je gubitak, a ustvari se radi o vodi koja se koristi. Taj oblik gubitaka se većinom sastoji od nedozvoljene potrošnje u vidu ilegalnih priključaka, a nešto manje od pogrešaka u mjerenju ili samoj administraciji. Na kraju, iznos vode od koje sustav vodoopskrbe ne uprihođuje novčana sredstva, može se izračunati tako da se od ukupne ulazne količine vode oduzme naplaćena dopuštena potrošnja [1]. Bilanca vode

izrađena na ovaj način, usmjerava razvoj strategija održavanja i upravljanja sustavom prema smanjivanju najutjecajnije kategoriju gubitka vode.

### 2.1.3. Plan očuvanja zdravstvene ispravnosti vode

Ovaj se plan ustvari sastoji od nekoliko koraka, čijim se ispunjenjem osigurava zdravstvena ispravnost vode za piće. Njime se razvijaju postupci osiguranja ispravnosti u svim fazama distribucije vode, od faze zahvaćanja, pročišćavanja, dostave pa do upotrebe vode kod samog krajnjeg korisnika. On sadrži načine zaštite vodnih izvora, potrebne metode pročišćavanja vode te načine distribucije i upotrebe vode. Izradom plana očuvanja zdravstvene ispravnosti vode, minimizira se rizik od naglog neočekivanog gubitka njene kvalitete te su njime također definirani postupci za iznimne slučajeve kad unatoč svemu ipak do toga dođe [2]. U nastavku su prikazana tri temeljna koraka u izradi ovog plana.

**Tablica 4. Koraci u izradi plana očuvanja zdravstvene ispravnosti vode [2]**

1.korak	2.korak	3.korak
Procjena stanja u sustavu vodoopskrbe	Procjena rizika	Upravljanje rizicima
<ul style="list-style-type: none"> <li>• Formiranje tima ljudi za izradu i provedbu plana</li> <li>• Definiranje namijenjenog načina upotrebe vode</li> </ul>	<ul style="list-style-type: none"> <li>• Identifikacija opasnosti (bioloških, kemijskih, fizikalnih ili radioloških) od izvora do potrošača</li> <li>• Identifikacija potencijalno opasnih događaja</li> <li>• Vjerojatnost da će opasnost uzrokovati štetu korisnicima</li> </ul>	<ul style="list-style-type: none"> <li>• Primjena mjera nadzora</li> <li>• Nadzor rada sustava</li> <li>• Postupci menadžmenta (korektivne radnje i pravovremena reakcija na probleme)</li> <li>• Programi podrške</li> <li>• Dokumentiranje svih mjera</li> <li>• Validacija i verifikacija poduzetih mjera</li> </ul>

Iz prethodne tablice, moguće je uočiti da je dobar dio faze procjene i upravljanja rizicima sličan postupcima kod izrade plana upravljanja imovinom. Iako pojedini planovi upravljanja imovinom vodoopskrbnih sustava, kod analize rizika u obzir mogu uzeti i rizik vezan uz zdravstvenu ispravnost vode, to često nije slučaj, pa se preporuča izrada posebnog plana koji se bavi tim problemom.

## 2.2. Strategija održavanja

Strategije održavanja nastaju u sklopu plana upravljanja imovinom sustava. Njeno stvaranje polazi od identifikacije vrste i učestalosti operacija održavanja koje je potrebno provesti nad svakim pojedinim imovinskim sredstvom koje se nalazi u registru imovine. Temeljem te informacije, stvara se raspored operacija održavanja koji će poslužiti za uspostavljanje sustava i postupaka podrške planiranim, ali i neplaniranim, zastojsima. To podrazumijeva stvaranje sustava nabave koji osigurava posjedovanje odgovarajućih komponenti na skladištu, u potrebnoj količini i trenutku kada one zatrebaju (planirano ili neplanirano). Primjeri preporučenih operacija održavanja mreže distribucije vode u vodoopskrbnom sustavu te njihova učestalost, dani su u slijedećoj tablici.

**Tablica 5. Preporučena učestalost pojedinih operacija održavanja [2]**

Br. komp.	Operacija održavanja	Dnevno	Tjedno	Mjesečno	Godišnje	Ostalo
<b>1</b>	<b>Cjevovod</b>					
1.1	Provjeriti postojanje curenja (segment po segment)			+		
1.2	Provjeriti i očistiti pristupno okno. Provjera lokota.			+		
1.3	Prekontrolirati i dopuniti zalihe rezervnih dijelova			+		
1.4	Isprati cijeli sustav					+
1.5	Provjeriti oznake				+	
1.6	Obnoviti oznake					+
<b>2</b>	<b>Cijevi za gašenje požara (u javnim zgradama)</b>					
2.1	Provjeriti postojanje curenja	+				
2.2	Očistiti otvor i okruženje cijevi		+			
2.3	Očistiti vlagu uslijed kondenzacije		+			
2.4	Održavati opremu i dodatke cijevi za gašenje		+			
2.5	Očitati vodomjere i očistiti pristupna okna			+		
2.6	Zamijeniti ventile za puštanje vode				+	
<b>3</b>	<b>Područni vodomjeri</b>					
3.1	Očitavanje područnih vodomjera			+		
3.2	Čišćenje pristupnih okna i kućišta vodomjera			+		
3.3	Provjeriti postojanja curenja			+		
3.4	Kalibracija vodomjera				+	
3.5	Zamjena mjerača unutar vodomjera					+
3.6	Zamjena cijelog vodomjera					+
3.7	Provjeriti i očistiti filtere područnih vodomjera			+		
<b>4</b>	<b>Vodomjeri u kućanstvima</b>					
4.1	Očitavanje vodomjera			+		
4.2	Čišćenje kućišta vodomjera			+		
4.3	Provjeriti postojanje curenja			+		
4.4	Provjeriti postoji li kakva zloupotreba					+



4.5	Provjeriti točnost vodomjera					+
4.6	Zamjena cijelog vodomjera					+
<b>5</b>	<b>Ventili</b>					
5.1	Otvoriti i zatvoriti izolacijske ventile			+		
5.2	Provjeriti ispravnost ispusnih ventila			+		
5.3	Provjeriti ispravnost zračnih ventila			+		
5.4	Provjeriti ispravnost ventila za regulaciju tlaka			+		
5.5	Servisiranje ventila za regulaciju tlaka svake 2 god.					+
5.6	Održavati opremu i dodatke za ventile				+	
5.7	Zamjena ventila					+
5.8	Čišćenje pristupnih okna. Provjera lokota.			+		
<b>6</b>	<b>Hidranti</b>					
6.1	Provjeriti postojanje curenja			+		
6.2	Provjeriti postojanje oznake ili popraviti vidljivost				+	
6.3	Provjeriti ispravnost hidranata			+	+	
6.4	Testirati protok hidranata				+	
6.5	Održavati opremu i dodatke za hidrante				+	
6.6	Zamjena hidranata					+
<b>7</b>	<b>Dokumentiranje i izvještavanje</b>	+	+	+	+	

Osim toga, razvijaju se postupci koji se slijede u slučaju iznenadne potrebe za provođenjem operacija održavanja (korektivno održavanje). Bilo koja aktivnost održavanja obuhvaćena rasporedom, unosi se u osobne rasporede osoblja koje provodi održavanje te mora biti odobrena s obzirom na potrebu i raspoloživa sredstva. Naposljetku, potrebno je bilo kakav zahvat detaljno i kvalitetno dokumentirati kako bi se omogućio kasniji uvid i analiza potrebna kod preventivnog pristupa održavanju. Primjer osobnog rasporeda održavanja crpne stanice na koji djelatnik održavanja detaljno dokumentira sve potrebne informacije, dan je na slijedećoj slici.

### Tjedne obveze održavanja crpne stanice

Lokalitet: \_\_\_\_\_ Ime crpne stanice/broj: \_\_\_\_\_

Tjedni raspored broj: \_\_\_\_\_ Ime zadužene osobe: \_\_\_\_\_

	Provjereno		Napomene
	Da	Ne	
<b>1. Potopne pumpe</b>			
1.1 Provjeriti prohodnost usisne strane ili stranih stvari u <u>impeleru</u> te ih ukloniti po potrebi			
1.2 Pokrenuti pumpu i provjeriti jakost struje pri punom opterećenju			Struja pri punom opterećenju: [A]
1.3 Očitati tlakove sa mjerača			Dostavni tlak: [kPa]
1.4 Podignuti pumpu i provjeriti brzo spajanje i <u>odspajanje</u> (ako postoji)			
1.5 Ukloniti pumpu i očistiti naslage unutar nje (svakih 6 mjeseci)			
<b>2. Centrifugalne pumpe</b>			
2.1 Provjeriti razinu ulja			
2.2 Provjeriti stanje kućišta i prilagoditi			
2.3 Provjeriti jesu li vijci koji pumpu drže u temeljnoj ploči dovoljno čvrsto pritegnuti			
2.4 Provjeriti postojanje abnormalnih vibracija			
2.5 Provjeriti postojanje abnormalnih temperatura			
2.6 Osluškiivati postoje li abnormalni zvukovi			
2.7 Pokrenuti pumpu i provjeriti jakost struje pri punom opterećenju			Struja pri punom opterećenju: [A]
2.8 Očitati tlakove sa mjerača			Dostavni tlak: [kPa]
2.9 Provjeriti poravnatost vratila motora i pumpe			
2.10 Provjeriti stanje spojeva na pumpi			
2.11 Provjeriti ispravnost nepovratnog ventila			
2.12 Provjeriti ispravnost ventilatora za hlađenje motora			
<b>3. Električno upravljanje</b>			
3.1 Ispitati ispravnost elektropokretača			
3.2 Ispitati ispravnost zaštite od prekomjernog/nedovoljnog napona			
3.3 Provjeriti ispravnost mehanizma preopterećenja i sklopki			
3.4 Pokrenuti pumpu i provjeriti ispravnost ampermetra i voltmetra			
3.5 Provjeriti ispravnost svih indikacijskih svjetala			
3.6 Provjeriti ispravnost rasvjete			
3.7 Provjeriti ispravnost mjerača razine vode			
3.8 Provjeriti ispravnost telemetrijskog sustava (ako postoji)			
3.9 Provjeriti stanje ležajeva elektromotora			

Ime izvođača: \_\_\_\_\_

Potpis: \_\_\_\_\_

**Slika 3. Primjer tjednog rasporeda održavanja u obliku izvještaja [2]**

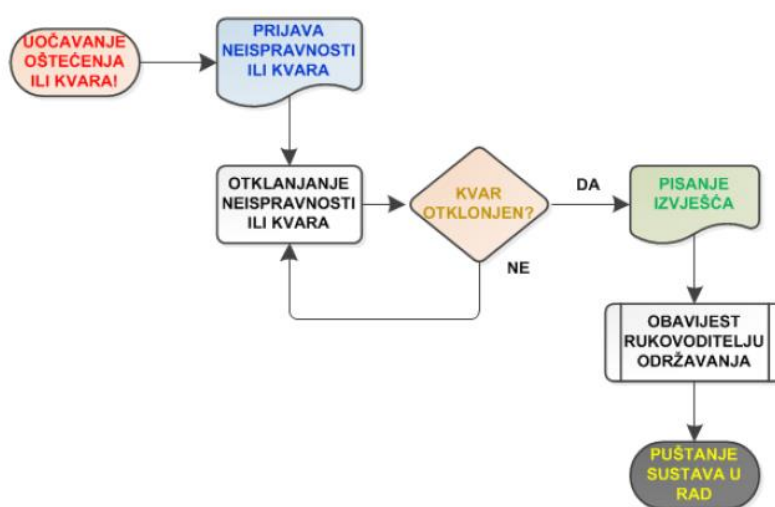
Ovakve strategije počinju se razvijati za narednu godinu te se zatim proširuju na razdoblje od pet godina sa godišnjim evaluacijama njihove kvalitete. Ako se utvrdi da je neki plan za

promatranu godinu bio neadekvatan, rade se dopune i izmjene budućih planova po potrebi [2].

Današnje strategije održavanja u grubo se mogu podijeliti na slijedeće dvije osnovne skupine [4]:

- **korektivne:** kod kojih se operacije održavanja provode nakon što se kvar dogodi
- **preventivne:** koje se temelje na provođenju operacija održavanja po unaprijed utvrđenom rasporedu, prije nego se kvar dogodi

Iako je težište stavljeno na provođenje preventivnih operacija održavanja, potrebu za korektivnim operacijama nikada nije moguće u potpunosti iskorijeniti. Općeniti dijagram toka otklanjanja uočenog kvara kod korektivnog održavanja prikazan je na slijedećoj slici.



Slika 4. Dijagram toka kod otklanjanja kvarova u korektivnom održavanju [4]

Korektivni pristup održavanju ima niz nedostataka zbog kojih se ne koristi kao primarna strategija održavanja, a neki od ključnih su [4]: veći troškovi zbog dužih neplaniranih zastoja u radu, dulje vrijeme popravka, potreba za većim zalihama rezervnih dijelova, potreba za većim brojem ljudi koji su između kvarova neiskorišteni te mogućnost kolateralne štete. Preventivni pristup održavanju nema takve nedostatke te omogućuje racionalniju upotrebu novčanih sredstava i znatno veće produženje životnog vijeka imovine. Kod strategija održavanja vodoopskrbnih sustava, danas se najčešće primjenjuje jedna nova metodologija nastala unutar područja preventivnog održavanja, a to je proaktivno održavanje o kojem će više biti rečeno u nastavku.

### 2.2.1. Proaktivno održavanje

Kod proaktivnog održavanja, glavni je cilj identifikacija i eliminacija uzroka kvarova. Strategija održavanja koja slijedi tu ideju, podrazumijeva stalno praćenje i kontrolu osnovnih uzroka kvarova te implementira načine kako eliminirati ili ukloniti njihov utjecaj. Temeljna razlika u odnosu na preventivno održavanje je usmjerenost ka otkrivanju glavnih uzroka kvarova nasuprot otkrivanju i praćenju njegovih ranih simptoma. Za provođenje ove strategije održavanja koristi se metode poput: analize temeljnih uzroka kvarova (engl. *Root Cause Failure Analysis*), analize načina i posljedica kvarenja (engl. *Failure Mode and Effects Analysis*), održavanje usmjereno ka pouzdanosti (engl. *Reliability Centered Maintenance*) [4].

#### 2.2.1.1. Analiza temeljnih uzroka kvarova

Ova se metoda koristi u situacijama kada se neki kvar na imovinskom sredstvu često ponavlja. Uzroci ponavljanja kvara mogu biti [4]: greška u konstrukciji, greška u proizvodnji te nepravilno rukovanje ili održavanje. Utvrđivanje uzroka vrši se pomoću vođenja detaljnih izvještaja o događajima koji su prethodili nastanku kvara. U tu svrhu koriste se izvještaji ali i usmeni intervjui sa osobljem kojeg se kvar tiče. Nadalje, vrši se prikupljanje svih materijalnih dokaza o kvaru kako bi se nad njima provela analiza i u konačnici rekonstruirao niz događaja koji je prethodio kvaru [4]. Rezultat ove analize trebao bi dati odgovor na pitanja prikazana na slici.



Slika 5. Pitanja u svakom koraku provođenja analize temeljnih uzroka kvarova [6]

Pojednostavljeni primjer analize problema propadanja radne funkcije pumpe u crpnoj stanici, koja često u kratkom vremenskom razdoblju gubi traženi protok vode dan je u nastavku.

**Tablica 6. Primjer analize temeljnog uzroka kvara pumpe u crpnoj stanici**

Opis problema	Uzrok problema	Potencijalna rješenja problema	Najbolje rješenje
Dostavni tlak pumpe pada ispod proizvođačkih specifikacija i projektnih potreba sustava vodoopskrbe	Pojava kavitacije na impeleru pumpe	<ul style="list-style-type: none"> <li>• povišenje razine vode u vodospremniku</li> <li>• smanjenje hidrauličkih gubitaka u usisnoj cijevi povećanjem njenog promjera</li> </ul>	<ul style="list-style-type: none"> <li>• smanjenje hidrauličkih gubitaka u usisnoj cijevi povećanjem njenog promjera</li> </ul>

Rješenje za koje se utvrdi da je najbolje, radnici održavanja odmah primjenjuju, čime se minimizira vjerojatnost potrebe korektivnih radnji. Nadalje, radnje koje bi se inače provodile u sklopu strategije preventivnog održavanja na primjeru iz prethodne tablice, uključivale bi procjenu ranih simptoma pada dostavnog tlaka pumpe, kao što su vibracije ili procjenu vremena potrebnog da dođe do propadanja tražene funkcije (dostavnog tlaka). Nakon što bi se rani simptomi i korisni životni vijek kvantificirali, krenulo bi se u planirano mijenjanje impelera, što je jeftinije od korektivnih zahvata ali i dalje skuplje od proaktivnog održavanja, jer se korisni životni vijek komponente ne produljuje za onoliko, za koliko bi se produljio ako se ukloni temeljni uzrok kvara. Upravo u tome leži temeljena prednost u proaktivnom pristupu održavanju, u odnosu na klasični preventivni pristup.

#### 2.2.1.2. Analiza načina i posljedica kvarova

Ova metoda bazira se na analizi potencijalnih uzroka i vrsta kvarova te određivanju posljedica (rizika) ukoliko do kvara dođe. Za uspješnu upotreba potrebno je posjedovati tehničku dokumentaciju imovine koja se održava kao i dobro dokumentiranu povijest održavanja. Ukoliko je implementiran kvalitetan sustav upravljanja imovinom, sve potrebne informacije moguće je pronaći u tom dokumentu ili informacijskom sustavu. Analiza se provodi na način da se za svaku komponentu sustava najprije unese njena funkcija. Nakon toga se za svaku funkciju unose potencijalni načini zatajenja, odnosno potencijalni kvarovi. Kod svakog potencijalnog kvara navode se njihove potencijalne posljedice koje se ocjenjuju po razini ozbiljnosti (od 1 do 10, gdje ocjena 10 znači vrlo ozbiljno). Zatim slijedi unošene potencijalnih uzroka tih kvarova kojima se dodjeljuje ocjena učestalosti. (ponovno od 1 do 10,

gdje je 10 vrlo često). Za svaki potencijalni problem predlaže se metoda detekcije te se ocjenjuje njegova primjetljivost (ocjenom od 1 do 10 gdje je 10 nizak stupanj primjenjivosti što znači da postoji velika vjerojatnost da se problem previdi). Naposljetku slijedi izračun vrijednosti prioriteta rizika RPN (engl. *Risk Priority Numbers*) kao umnožak ozbiljnosti S (engl. *Severity*), pojavljivosti O (engl. *Occurence*) i primjetljivosti D (engl. *Detectability*) kvara [4]:

$$RPN = S \cdot O \cdot D \quad (1)$$

Pojednostavljeni primjer provođenja ove metode analize nad pumpom u crpnoj stanici vodovoda prikazan je u nastavku.

**Tablica 7. Primjer provođenja metode analize načina i posljedica kvara centrifugalne pumpe [4]**

Komponenta	Centrifugalna pumpa			
Funkcija	Pumpanje vode			
Potencijalni kvar	Kvar elektromotora			
Posljedice	Kvar separatora		Zaustavljanje pumpe	
Ozbiljnost	9		7	
Potencijalni uzroci	Pregorijevanje elektromotora	Pregorijevanje osigurača	Pregorijevanje elektromotora	Pregorijevanje osigurača
Pojavljivost	1	3	1	3
Metoda detekcije	Vizualna inspekcija izolacije	Vizualna inspekcija izolacije	Vizualna inspekcija izolacije	Vizualna inspekcija izolacije
Primjetljivost	7	9	7	9
RPN	63	243	49	189

Iz ovog primjera vidljivo je da najveći rizik kod kvara elektromotora na centrifugalnoj pumpi, predstavlja kvar separatora kao posljedica pregorijevanja osigurača. Za tu kombinaciju je također najniža primjetljivost (veća ocjena znači nižu primjetljivost). Na temelju tog rezultata, djelatnici održavanja vodovoda trebaju odrediti troškove rizika u slučaju pojave najrizičnijih

kvarova te troškove potrebne za njihovu eliminaciju. Usporedbom tih troškova pomaže im u donošenju odluke o poduzimanju radnje održavanja [4].

### 2.2.1.3. Održavanje usmjereno ka pouzdanosti

Ova metoda proaktivnog principa održavanja, koristi se u svrhu osiguranja dostupnosti i pouzdanosti opreme uz minimalne troškove. U njenom provođenju mogu pomoći prethodno objašnjene metode jer se sadrži od četiri slična elementa: identifikacija vrsta kvarova koji se mogu pojaviti i naštetiti sustavu, određivanja prioriternih kvarova te definiranja aktivnosti za njihovu kontrolu. Naglasak se ovdje stavlja na uspostavljanje procedura za produljenje korisnog životnog vijeka sustava kao cjeline [4].

## 2.3. Cjelovitost vodoopskrbnog sustava

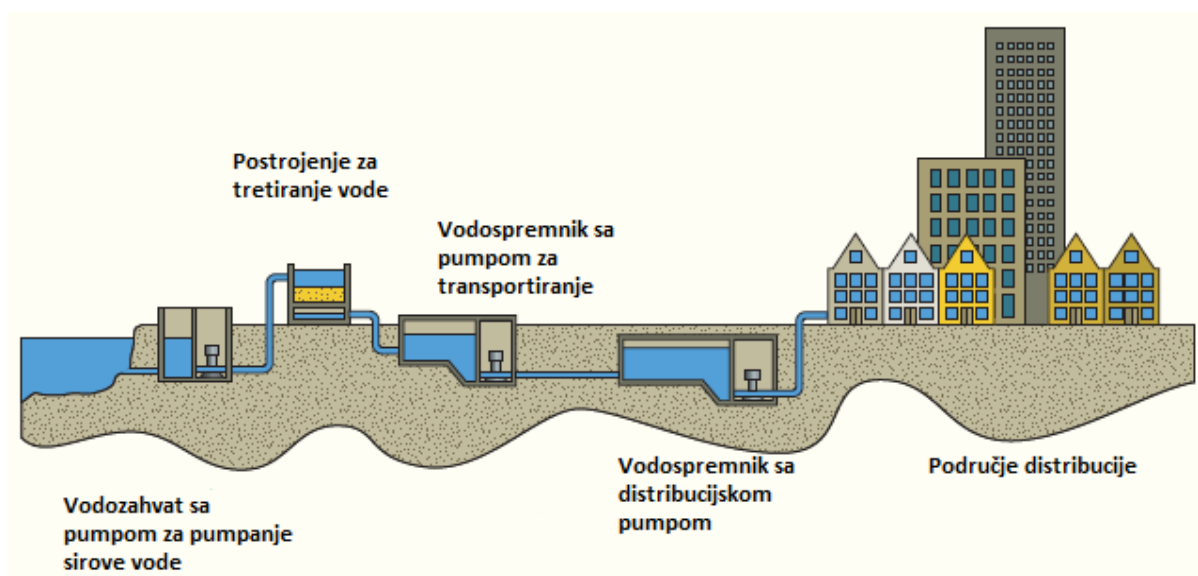
### 2.3.1. Osnovni elementi sustava

Vodoopskrbni sustavi primarno su osmišljeni kako bi distribuirali pitku vodu u traženoj količini i kvaliteti, uz zadovoljavajući tlak, svakom pojedinom korisniku koji to zahtjeva [2]. Kako bi se usluga isporučila kupcima, sustav mora posjedovati neke elemente koji će mu to omogućiti. Osnovni elementi koji tvore vodoopskrbni sustav su [7]:

- **vodozahvati**
- **crpne stanice**
- **vodospremnici**
- **uređaji za kondicioniranje vode**
- **mreža cjevovoda** – postoji glavna (magistralna) i razdjelna (distributivna) mreža
- **pumpe**
- **ventili** (izolacijski, zračni, ispusni, nepovratni, kontrolni)
- **ostali elementi** (područni vodomjeri, kućni vodomjeri, hidranti, pristupna okna, oznake, potisni blokovi)

Kako bi se osiguralo pružanje kvalitetne i održive usluge distribucije vode, neophodno je osigurati da se upravljanje i održavanje navedenim elementima vodoopskrbnog sustava provodi na ispravan način [2]. Voda koja se iz prirode zahvaća u vodozahvatima naziva se sirova voda. Postoje tri osnovne kategorije sirove vode s obzirom na to od kuda ona dolazi, a to su [8]: površinska, podzemna voda te voda prikupljena kišnicom. Voda prikupljena isključivo kišnicom karakteristična je za siromašne zemlje u kojima sustavi distribucije vode

nisu razvijeni. Takva metoda prikupljanja ne predstavlja održiv izvor napajanja vodom kod modernih vodoopskrbnih sustava razvijenih zemalja te stoga neće biti obrađena ovim radom. Voda, prije nego li se podvrgne bilo kojem postupku kondicioniranja u postrojenju za obradu, sadrži velike količine raznih konstituenata zbog čega se za nju upotrebljava prethodno upotrijebljen naziv sirova voda. Dok su neki od tih konstituenata neophodni za ljudsko zdravlje, postoje i oni koji se smatraju opasnim a čiji je utjecaj potrebno što je više moguće umanjiti. Njihova se koncentracija smanjuje obradom u uređajima za kondicioniranje. Takvi uređaji primjenom raznih metoda nastoje utjecati na smanjenje koncentracije štetnih tvari, kako bi se postigli odgovarajući zakonski propisani, dopušteni iznosi. Niti jedan konstituent ne može se ukloniti u potpunosti, pa tako niti onaj štetni, međutim, moguće je sniziti njegovu koncentraciju na dovoljno nisku razinu koja osigurava da je voda zdravstveno ispravna i spreman za upotrebu u kućanstvu [2]. Ilustracija tipičnog vodoopskrbnog sustava sa pripadajućim elementima prikazana je sljedećom slikom.



Slika 6. Pojednostavljena ilustracija izgleda vodoopskrbnog sustava [9]

Na slici je prikazano zahvaćanje vode iz prirodnih izvorišta pomoću crpki. Zatim slijedi potiskivanje pumpama dalje do mjesta kondicioniranja, spremanja te u konačnici mjesta potrošnje obrađene, zdravstveno ispravne vode. Kako bi se osiguralo da sustav vodoopskrbe ispunjava svoju temeljnu svrhu, potrebno je definirati tri tipa cjelovitosti sustava kojima se to postiže. Prema literaturi, vrste cjelovitosti sustava koje moraju biti zadovoljene su sljedeće [2]:



- **Fizička cjelovitost:** komponente sustava moraju pružiti fizičku barijeru između vode i vanjskih opasnosti te funkcionirati kako je to zamišljeno
- **Hidraulička cjelovitost:** sustav mora biti u stanju pružiti protočnost i pritisak koji su potrebni za pružanje zadovoljavajuće usluge
- **Kvalitativna cjelovitost:** sustav mora pružati vodu prihvatljive kvalitete svim svojim korisnicima

Navedene vrste cjelovitosti međusobno su ovisne jedna o drugoj te narušavanje samo jedne od njih uzrokuje lančanu reakciju narušavanja preostalih [2]. U nastavku će biti izneseno nešto više o svakom od spomenutih zahtjeva kao i pripadajući osnovni utjecajni faktori koje je važno imati na umu kod razvoja i provođenja strategija održavanja.

### 2.3.2. Zahtjev fizičke cjelovitosti

Pojam fizičke cjelovitosti označava sposobnost distribucijskog sustava da se nosi sa svim unutarnjim i vanjskim opterećenjima, a da fizička barijera između vode i vanjskih opasnosti ostane cjelovita. Unutarnja opterećenja proizlaze iz variranja u radnom tlaku tijekom dana, tlačnih udara te korozije s unutarnje strane. Što se tiče opterećenja s vanjske strane, ona najčešće nastaju zbog naprezanja u zemlji, opterećenja uslijed težine objekata iznad sustava te vanjske korozije. Faktori koji uzrokuju gubitak fizičke cjelovitosti mogu se svrstati u tri kategorije: projektantski, fizički i kemijski [2]. Slijedeća tablica prikazuje sve faktore koji pripadaju navedenim kategorijama.

**Tablica 8. Utjecajni faktori na fizičku cjelovitost vodoopskrbnog sustava [2]**

	Utjecaj na fizičku cjelovitost
<b>PROJEKTANTSKI FAKTORI</b>	
<b>neadekvatno projektiranje</b>	Neadekvatno projektiranje može dovesti do zatajenja komponente. Primjer toga je odabir pogrešne potrebne debljine stjenke cijevi u odnosu na vršni tlak kojeg treba podnijeti ili odabir materijala sklonog koroziji u slučaju primjene u nekom agresivnom mediju.
<b>defektivan materijal</b>	Moguće je da se upotrebljavaju komponente koje u slučaju čelika sadrže mikroskopske napukline ili u slučaju plastike, zarobljene nečistoće. Takvi nedostaci mogu ostati neidentificirani u proizvodnom postupku.
<b>nedostatak ili neispravnost dijelova</b>	Neke komponente koje se odaberu za upotrebu možda nisu dobre u obavljanju svoje zadaće na ispravan način ili nisu uopće odabrane a potrebne su. Primjer toga je nedostatak ili

	odabir lošeg pokrivača rezervoara ili barijere za insekte.
<b>nepravilna ugradnja ili popravak</b>	Kod izvođenja operacija ugradnje ili održavanja, dio sustava otvoren je prema svojem okruženju zbog čega je u tom trenutku njegova fizička cjelovitost narušena.
<b>nepravilno rukovanje ili upotreba komponenti i materijala izvan njihovih radnih parametra</b>	Važno je osigurati da se pri održavanju, komponente sustava ugrađuju i upotrebljavaju prema uputama proizvođača. Primjer toga je ugradnja mjerača potrošnje vode kod kućnog priključka koji mora biti ugrađen tako da voda ne curi te da se poštuje maksimalni propisani protok vode kojeg može podnijeti kako se ne bi oštetio.
<b>slučajne unakrsne konekcije</b>	Treba osigurati da u područjima gdje postoje paralelne mreže za dostavu vode različite kvalitete kućanstvima ili industriji, ne dođe do slučajnog spajanja tih sustava uslijed provođenja operacija održavanja.
<b>FIZIČKI FAKTORI</b>	
<b>prekomjerna opterećenja</b>	Opterećenja uzrokovana vanjskim silama koja je potrebno uzeti u obzir su: točkasto opterećenje zbog npr. oštrog kamena u podlozi na koju se polažu cijev, težine zemlje, objekata ili prometa iznad cijevi i kretanja zemlje. S druge strane, unutarnja opterećenja mogu nastati zbog: prevelikog tlaka vode koji opterećuje stjenke cjevovoda te tlačnih udara koji se javljaju zbog naglih promjena u kretanju vode unutar sustava.
<b>oštećenje</b>	Oštećenja mogu nastati tijekom izgradnje i održavanja sustava ili kao posljedica vandalizma na izloženim dijelovima.
<b>erozija</b>	Erozija predstavlja postepeni gubitak materijala kroz pojave brušenja od strane čestica u pokretu. To se događa u slučaju oštećenja cijevi kada voda počne prodirati u njenu okolinu čime se uzburkaju okolne čestice. Te čestice mogu dovesti do brušenja materijala cijevi koja se onda dodatno oštećuje. Osim toga, moguće je da dođe do erozije podloge na koju je cijev položena, pa se može dogoditi da se ona uslijed pritiska zemlje iznad nje uvine i pukne.
<b>izloženost sunčevoj svjetlosti</b>	S obzirom da se u novije doba sve češće koriste PVC cijevi, važno je imati na umu da se one mogu oštetiti ukoliko su izložene suncu. Zbog toga tijekom ugradnje ili zamjene ne smiju biti predugo izložene vanjskim uvjetima na gradilištu.
<b>kavitacija</b>	Kavitacija je proces u kojem se tlak u vodi smanji ispod tlaka zasićenja što izaziva pojavu malih mjehurića pare u vodi. Kada ti mjehurići odjednom dospiju u područje visokog tlaka, npr. unutar pumpe, postanu nestabilni i implodiraju. Te implozije izazivaju valove visokog intenziteta i tlaka koji oštećuju okolni materijal. Kavitacija može izazvati ozbiljna i skupa oštećenja pumpi koje su nepravilno ugrađene i upotrebljavane.
<b>KEMIJSKI FAKTORI</b>	
<b>korozija</b>	Korozija je pojava degradacije materijala uslijed kemijske reakcije sa okolinom. Postoje tri vrste korozije koje se

	najčešće javljaju u sustavima distribucije vode: galvanska, korozija uz lutajuće struje te mikrobiološka korozija. Potrebno je poduzeti mjere za sprječavanje ove štetne pojave, a koje mogu uključivati katodnu zaštitu, sprječavanje lutajućih struja te upotrebu inhibitora korozije.
<b>permeacija (prodiranje)</b>	Ovaj pojam odnosi se na mehanizam propadanja cijevi gdje zagađivači koji se nalaze izvan cijevi kompromitiraju strukturalni integritet te cijevi na takav način da dolazi do prolaženja kroz njihove stjenke i ulaska u vodu unutra. Permeacija se javlja kada plastične cijevi dolaze u kontakt sa kemijskim otapalima povezanih sa naftom i benzinom. Prisustvo takvih zagađivača može se detektirati kroz kemijsku plinsku kromatografsku analizu.

Prema podacima našeg Državnog zavoda za statistiku, postoje značajni gubici vode u domaćim distribucijskim sustavima. Podaci o gubicima pitke vode usporedno s podacima o prodanim količinama u sustavima distribucije u RH prikazani su u nastavku. Iskazane brojke ukazuju na to da zahtjev fizičke cjelovitosti nije u potpunosti zadovoljen s obzirom da su gubici po godinama u nekim slučajevima čak vrlo bliski prodanim, naplaćenim količinama vode [10].

**Tablica 9. Statistički podaci o količinama prodane i izgubljene vode u RH [10]**

tis. m <sup>3</sup> / godina	2012.	2013.	2014.	2015.	2016.	2017.
Prodana voda	282.392	278.591	251.892	270.701	251.430	255.182
Ukupni gubici vode	211.125	211.498	193.219	193.635	216.110	212.121

### 2.3.2.1. Osnovni principi održavanja fizičke cjelovitosti

Neki osnovni principi koje stručnjaci održavanja primjenjuju u svrhu očuvanja fizičke cjelovitosti vodoopskrbnog sustava su [2]:

- **aktivan nadzor stanja komponenti**

Nadzor je moguće provoditi kroz uspostavu rutinskih kontrola, primjenom područnih vodomjera, senzora te ručnog ili bežičnog bilježenja podataka koji će omogućiti stvaranje povijesnih performansi sustava. Takvi povijesni podaci kasnije mogu poslužiti za analize trendova kvarova, promjena tlakova i gubitaka vode što je posebno korisno kod primjene metoda proaktivnog održavanja.

- **primjena metodologije proaktivnog održavanja**

Ova metodologija oslanja se na identifikaciji uzroka kvarova te pravovremenog uočavanja potrebe za provođenjem operacija popravljivanja, dorade ili zamjene komponenti. Ako se

uspješno implementira, moguće je minimizirati vrijeme prekida isporuke usluge a samim time i troškove.

- **zaštita sustava od okoline**

Sustav se štiti od vandalizma, slučajnog ulaska ljudi, ulaska vode iz okoline te ulaska ptica ili insekata.

- **zaštita od korozije**

Vrši se analiza sustava kako bi se odredila područja gdje su cijevi i ostale komponente najranjivije po pitanju korozije, nakon čega se na tim mjestima implementiraju mjere za njeno sprječavanje. Kod metalnih cijevi, faktori koji pogoduju koroziji su električni otpor zemlje, njezin pH, redoks potencijal, prisutnost sulfida te stupanj drenaže. Za betonske cijevi temeljni faktori su kloridi u zemlji te električni otpor. Metode zaštite od vanjske korozije podrazumijevaju odabir prikladnog materijala, premazivanje, prevlačenje, ugradnju katodne zaštite te smanjivanje zalutalih struja u zemlji. Što se tiče mjera za zaštitu od unutarnje korozije, one uključuju dodavanje inhibitora korozije kao što su fosfati koji se dodaju vodi ili tvari kojima je cilj prilagoditi pH i alkalnost zemlje na željenu razinu.

- **sprječavanje unakrsnih konekcija paralelnih sustava**

Unakrsne konekcije sprječavaju se kvalitetnom dokumentacijom te obukom zaposlenika koji trebaju slijediti sve potrebne, propisane korake pri izvođenju operacija održavanja.

- **sprječavanje povratnog toka u sustavu**

Sustav je osmišljen na takav način da je povratni tok vode natrag u njega nemoguć. To se postiže ugradnjom kontrolnih ventila ili bilo kojih drugih mjera koje djelatnici održavanja mogu naknadno ugraditi ako za to postoji potreba.

- **dokumentiranje svih postupaka**

Sve operacije održavanja koje se izvode potrebno je detaljno i savjesno dokumentirati za potrebe kasnijih analiza i uvida.

### **2.3.3. Zahtjev hidrauličke cjelovitosti**

Hidraulička cjelovitost distribucijskog sustava, ispunjena je ako on zadovoljava sve korisničke potrebe, a da su pritom osigurani poželjni tlakovi, brzine i starost vode u njemu. Najvažniji indikator hidrauličke cjelovitosti upravo je tlak u sustavu. On nije konstantan već se mijenja tijekom dana, pa je tako najveći tijekom noćnih sati kada su tokovi i energetske gubici najniži, a najveći tijekom dana u doba najveće potrošnje vode kada vrijedi obrnuto. Neke opće smjernice po pitanju tlaka u sustavu, izraženog visinom stupca vode ( $h = p/\rho g$ ),

nalažu da se on mora kretati u rasponu od 24 do 90 m. Uz gustoću vode od  $1000 \text{ kg/m}^3$ , te visine stupca odgovaraju rasponu tlakova od 2.35 do 8.83 bara [2]. U slijedećoj tablici prikazani su osnovni uzroci gubitka hidrauličke cjelovitosti s objašnjenjima.

**Tablica 10. Utjecajni faktori na hidrauličku cjelovitost kod vodoopskrbnih sustava [2]**

	<b>Utjecaj na hidrauličku cjelovitost</b>
<b>prekomjerna potreba</b>	Prosječna godišnja dnevna potražnja vode s vremenom raste zbog ekonomskog rasta, povećanja gustoće naseljenosti i curenja vode iz sustava koji stari. Ukoliko sustav distribucije vode nije konstruiran za takvu potražnju ili traženi životni vijek dolazi do gubitka potrebnog pritiska te nemogućnosti isporuke vode potrošačima.
<b>smanjenje kapaciteta sustava</b>	Hidraulički se kapacitet cijevi s vremenom smanjuje zbog pojave raznih naslaga na stjenkama koje su posljedica raznih soli metala kao što su kalcijev karbonat i aluminijski silikat te nusprodukta unutarnje korozije. Takve cijevi lakše pucaju i izazivaju gubitak tlaka.
<b>negativni tlakovi</b>	U nekim ekstremnim slučajevima potražnje vode, tlakovi u sustavu mogu pasti ispod atmosferskog. Takav će negativan tlak usisati zagađenu vodu iz okoline u slučaju postojanja pukotina ili oštećenja cijevi. Osim onečišćene vode, može doći do usisavanja zraka koji će promiješati potencijalne naslage u cijevi i zamutiti vodu ili oštetiti mjerač potrošnje kod krajnjeg kupca. U slučaju ozbiljnijeg negativnog tlaka može doći do oštećenja cijevi ili urušavanja u slučaju cijevi velikog promjera.
<b>pumpanje izravno iz mreže</b>	Neki kupci u uvjetima niskog pritiska u sustavu ugrađuju pumpe u svojoj kući kako bi si sami poboljšali kvalitetu usluge, međutim, to uzrokuje dodatno sniženje tlaka ostalim kupcima u mreži. Pored navedenog, pumpanje izravno iz mreže može izazvati pojavu negativnog tlaka, pa je upotreba privatnih pumpi u tu svrhu obično zabranjena.
<b>tlačni udari</b>	Tlačni udari javljaju se pri naglim promjenama u protoku vode unutar sustava. Primjeri kod kojih se to obično dešava su naglo zatvaranje ventila ili gašenje pumpe. Nagla promjena u protoku izaziva rastući ili padajući tlačni val koji putuje sustavom i kada dođe do njegovog završetka, biva reflektiran natrag (ako je bio rastući onda postaje padajući i obrnuto). Ukoliko nisu odabrane odgovarajuće cijevi ili one nisu zadovoljavajuće kvalitete, tlačni udari mogu uzrokovati oštećenja i curenja vode.
<b>prekomjerni tlakovi</b>	Prekomjerni tlakovi u sustavu povećavaju vjerojatnost pojave novih oštećenja kao i količinu vode koja se gubi curenje kroz postojeća. Osim što smanjuju životni vijek cijevi, također mogu uzrokovati oštećenja u kućanskim aparatima poput perilice rublja kod krajnjih korisnika.
<b>niske brzine</b>	Moraju se osigurati određene minimalne brzine kretanja vode u sustavu kako bi se spriječilo nakupljanje raznih naslaga koje služe kao hrana bakterijama. Nadalje, voda koja je previše statična u sustavu može potrošiti klor te izazvati nakupljanje nusprodukata dezinfekcije.
<b>loši radni parametri</b>	Može se dogoditi da su neki izolacijski ventili nepotrebno otvoreni ili zatvoreni ili da se koriste pogrešne postavke na kontrolnim ventilima. Takve stvari, ako se ne poprave mogu dovesti do gubitka hidrauličke cjelovitosti sustava. Kako bi se to izbjeglo važno je educirati osoblje koje upravlja tim komponentama, kako bi se znali ispravni radni parametri te

	dokumentirati sva poduzimanja kao što su otvaranje ili zatvaranje ventila.
<b>zrak u sustavu</b>	Zrak može u sustav ući kroz vodu, kada je voda ispuštena iz sustava zbog održavanja ili zbog pojave negativnog tlaka. U toku rada distribucijskog sustava, zrak se sažima zbog postojanja pritiska u sustavu te naglo expandira pri izlasku iz njega kod krajnjeg kupca što može oštetiti kućanske aparate ili prije kupca, neke elemente distribucijskog sustava.

### 2.3.3.1. Osnovni principi održavanja hidrauličke cjelovitosti

Najčešće mjere održavanja kako bi se osigurala hidraulička cjelovitost sustava su [2]:

- **ispiranje i čišćenje cijevi**

Ove operacije trebalo bi provoditi redovito, kako bi se osiguralo da se nakupljeni sedimenti uklanjaju sa cijevnih stjenki.

- **aktivno traženje mjesta curenja**

O tehnikama aktivnog traženja mjesta curenja cijevi više će riječi biti u jednom od narednih poglavlja.

- **ažuran popravak prijavljenih kvarova ili curenja**
- **mjere kontrole kvalitete popravaka**

Mjere kontrole kvalitete su ključne kako bi se osiguralo da su popravci uspješno provedeni i da ne stvore neke nove probleme.

### 2.3.4. Zahtjev kvalitativne cjelovitosti

Kvalitativna cjelovitost sustava distribucije vode podrazumijeva sposobnost pružanja vode čiji su kvalitativni parametri unutar onih propisanih nekim zakonom o zdravstvenoj ispravnosti vode za piće. Danas je poznato da su gotovo svi konstituenti koje se može pronaći u vodi bezopasni, kada se njihova koncentracija snizi na dovoljno nisku razinu. To je naravno tako pod uvjetom da je smanjenje koncentracije pojedinog konstituenta moguće postići na tehnički i ekonomski opravdan način. Neka opća najvažnija svojstva i konstituenti vode, na koje se gleda pri ispitivanju i osiguravanju kvalitete u svakom vodoopskrbnom sustavu, prikazani su idućom tablicom, gdje su smješteni u opće kategorije [2].

**Tablica 11. Utjecajni faktori na kvalitativnu cjelovitost vodoopskrbnog sustava [2]**

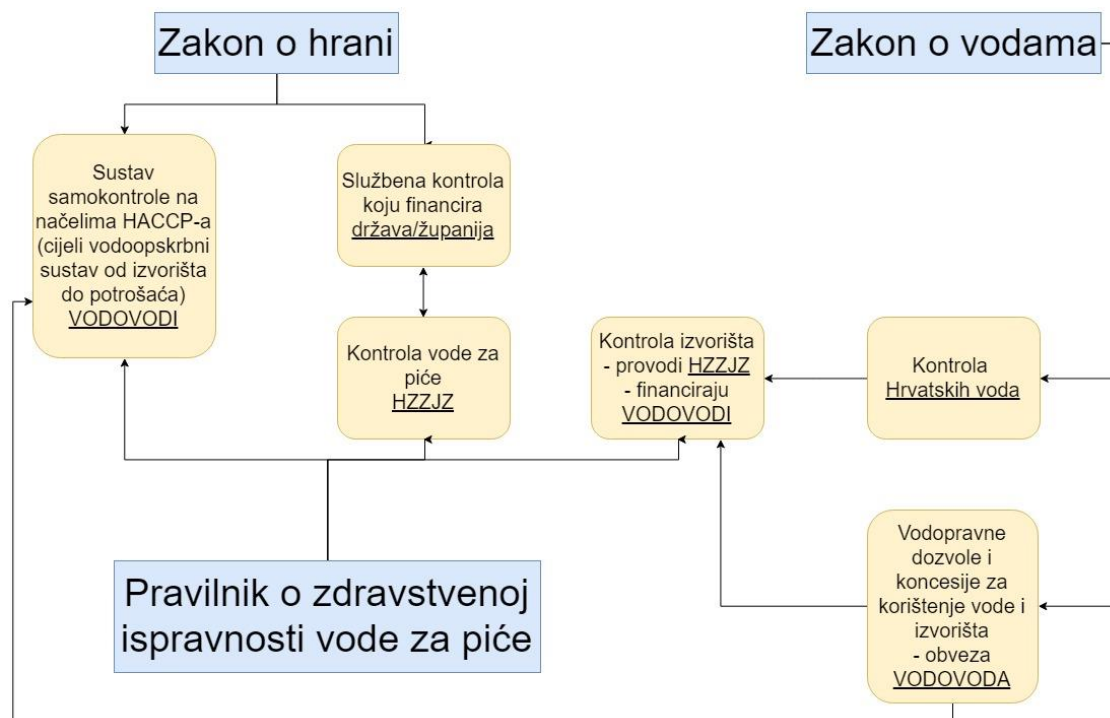
	<b>Utjecaj na kvalitativnu cjelovitost</b>
<b>FIZIKALNA SVOJSTVA</b>	
<b>dezinfekcijski ostaci</b>	U postupku pročišćavanja vode upotrebljavaju se razne tvari za dezinfekciju. Kako se tako pročišćena voda kreće kroz sustav, ona sadrži određene ostatke tih otopljenih tvari. Dezinfekcijski ostaci sprječavaju rast mikroorganizama u vodi koja je izašla iz

	postrojenja za tretiranje jer ulaze u reakciju s njima u kojoj se troše. S obzirom na to, ti ostaci ujedno predstavljaju parametar za ispitivanje zdravstvene ispravnosti vode, jer preniske koncentracije mogu upućivati na veliki broj mikroorganizama, dok prevelike uzrokuju neugodan okus i unošenje nepotrebno velikih koncentracija kemikalija u ljudski organizam.
<b>boja</b>	Boja u vodi ne mora nužno predstavljati opasnost, već može biti posljedica postojanja određene koncentracije organskih tvari i kemikalija kao što su željezo, mangan i bakar, koji se nalaze unutar dopuštenih razina. Bez obzira na to, najčešće posjedovanje neke boje ukazuje na određeno zagađenje vode te osim toga nije estetski prihvatljivo krajnjim korisnicima u vodoopskrbnom sustavu.
<b>električna vodljivost</b>	Čista voda je loš vodič elektriciteta međutim, ukoliko su u njoj otopljene soli, ta vodljivost se povećava. Stoga je vodljivost vode ključan parametar za određivanje ukupne količine otopljenih soli. Električna vodljivost izražava se u mjernoj jedinici mili Siemens po metru (mS/m).
<b>okus i miris</b>	Čista voda je bez okusa i mirisa. Određene kemikalije i tvari koje se nalaze u vodi prije pročišćavanja mogu izazvati postojanje okusa i mirisa, ali to nije prihvatljivo za krajnjeg korisnika.
<b>ukupna koncentracija otopljenih krutina</b>	Predstavlja mjeru prisutnosti organskih i anorganskih tvari u vodi.
<b>mutnost</b>	U vodi se mogu nalaziti čestice gline, mulja ili organskih tvari što će ju učiniti manje propusnom za svjetlost. Visoka razina mutnosti je estetski neprihvatljiva te može ukazivati na loš postupak tretiranja vode u pogonu. Mutnost se mjeri instrumentom imena nefelometar kod kojeg se mjeri intenzitet svjetlosti koja prolazi kroz uzorak vode. Taj intenzitet iskazuje se u jedinicama nefelometarske mutnoće (engl. <i>Nephelometric Turbidity Units – NTU</i> ).
<b>pH</b>	Mjerna jedinica pH u vodi upućuje na to je li ona kisela ili alkalna. pH skala kreće se od 0 do 14, a sve tekućine čiji je iznos manji od 7 (iznos 7 ukazuje na neutralnost) su kisele, dok su sve one s iznosom većim od 7, alkalne. Iznos pH ima važnu ulogu u vidu kontrole mnogih kemijskih reakcija i bioloških aktivnosti u vodi. Obično se teži da iznos bude između 6.5 i 8.5.
<b>KEMIJSKA SVOJSTVA</b>	
<b>anorganske tvari</b>	Prije nego se izuzme iz prirode, voda je u kontaktu sa velikim rasponom elemenata i tvari prisutnih u zemlji, kamenju i zraku. Prema tome, lako može doći u kontakt s ljudskim aktivnostima, industrijskim procesima i zagađenjem. Kao posljedica tog kontakta, u vodi se može pojaviti niz anorganskih tvari koje treba kontrolirati kako bi bili u dopuštenim rasponima.
<b>organske tvari</b>	Glavne grupe organskih tvari koje utječu na kvalitetu vode su sljedeće: otopljene organske tvari (koje mogu poslužiti mikroorganizmima kao hrana za rast), trihalometani (nusprodukti dezinfekcije tj. reakcije klora sa organskim tvarima u vodi – neki trihalometani su kancerogeni zbog čega se stavlja ograničenje na

	njihovu prisutnost) te fenoli (vrsta kemijskih spojeva koji većinom dolaze od industrijskog zagađenja te utječu na okus vode)
<b>MIKROBIOLOŠKA SVOJSTVA</b>	
<b>koliformi</b>	Koliformi su indikatori fekalnog zagađenja, a samim time i prisutnosti mikroorganizma koji mogu izazvati gastrointestinalne bolesti. Escherichia Coli jedna je od bakterija koje spadaju u tu skupinu, te je vrlo česta, zbog čega se, pored općenitog ispitivanja koncentracije ukupnih koliforma, može vršiti specifično testiranje na nju.
<b>heterotrofne bakterije</b>	Ove bakterije koriste organski ugljik kao izvor hrane te njihova koncentracija pokazuje opće mikrobiološko stanje vode.
<b>citopatogeni virusi</b>	Ovo je grupa virusa koji uzrokuju bolesti poput gastroenteritisa, encefalitisa, dječje paralize, meningitisa, hepatitisa, respiratornih bolesti i dijareje.
<b>protozojski paraziti</b>	Dva najvažnija protozojska parazita u sustavima distribucije vode su Giardia i Cryptosporidium. Ti paraziti obično imaju ljusku koja ih štiti od vanjskih prijetnji kao što je klor. Njihovu prisutnost potrebno je ispitivati jer se u protivnom mogu sustavom širiti potpuno neometano i izazivati dijareju, povraćanja i ostale zdravstvene probleme kod korisnika.
<b>somatski kolifagi</b>	Somatski kolifagi su virusi koji mogu zaraziti bakterije (poput E. coli.)

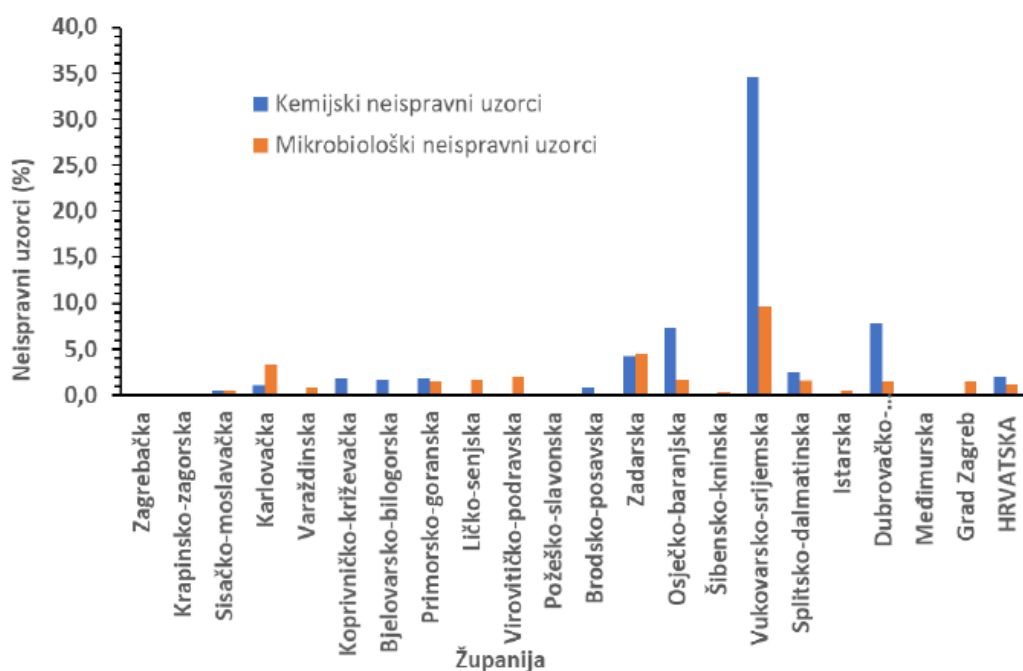
Svaka država izrađuje vlastite pravilnike o zdravstvenoj ispravnosti vode u vodoopskrbnim sustavima, pa dopušteni rasponi mogu varirati. U Republici Hrvatskoj, dopušteni iznosi definirani su Pravilnikom o parametrima sukladnosti, metodama analize, monitoringu i planovima sigurnosti vode za ljudsku potrošnju [11]. Za nadzor izvora i vode koja se koristi za piće, nadležno je više institucija i ministarstava u RH, što je prikazano na slici dolje.





**Slika 7. Prikaz nadležnosti i važećih zakona u provođenju nadzora izvorišta i vode za piće u RH [12]**

Prema Izvještaju o zdravstvenoj ispravnosti vode za ljudsku potrošnju u Republici Hrvatskoj za 2017. godinu [13], ispitana voda u javnoj vodoopskrbi vrlo je dobra izuzev Vukovarsko-srijemske županije. U tom razdoblju uzeto je i ispitano ukupno 7081 uzorak iz svih županija zajedno. Podaci o postotku neispravnih uzoraka po županijama kao i prosjek za cijelu državu dani su na slijedećoj slici.



Slika 8. Postotak kemijski i mikrobiološki neispravnih uzoraka vode po županijama u RH [13]

#### 2.3.4.1. Osnovni principi održavanja kvalitativne cjelovitosti

U nastavku su opisani općeniti najčešći postupci kod održavanja koji osiguravaju očuvanje kvalitete vode u sustavu [2]:

- **čišćenje i ispiranje cijevi i vodospremnika** kako bi se uklonile nakupljene naslage
- **provođenje procedura dezinfekcije** i ispiranja pri svakoj ugradnji, zamijeni ili popravku bilo koje komponente sustava
- **redovito čišćenje filtera** nizvodno od mjerača protoka, kontrolnih ventila i sličnih komponenti

Pored navedenih najčešćih postupaka ulaze i postupci održavanja nad uređajima za kondicioniranje vode. O samom postupku kondicioniranja vode kao i o postupcima potrebnim za održavanje tih procesa više će biti rečeno u narednim, zasebnim poglavljima.

## 2.4. Kondicioniranje vode

Najvažnija operacija za postizanje kvalitativne cjelovitosti vodoopskrbnog sustava, svakako je postupak kondicioniranja kojim se osigurava da je voda sigurna za piće te kemijski stabilna. Voda je sigurna za piće kada zadovoljava sve parametre propisane pravilnikom o parametrima sigurnosti vode za ljudsku potrošnju, dok je njena kemijska stabilnost postignuta onda kada ne izaziva stvaranje korozije u cijevima. Još u samoj fazi planiranja nekog vodoopskrbnog sustava, uzimaju se uzorci sirove vode iz izvora koji se planira koristiti. Uzete uzorke ispituju ovlaštene laboratoriji kako bi se odredili svi konstituenti čije koncentracije prelaze one dopuštene kod ljudske potrošnje [14]. Temeljem tih podataka odabire se tehnološki proces kondicioniranja kojim će se svi nedopušteni konstituenti umanjiti ili ukloniti na tehnički i ekonomski najisplativiji način [7]. Ukoliko niti jedan od konstituenata ne prelazi maksimalno dopuštene koncentracije, voda se može u pravilu koristiti bez izmjene, ali pod uvjetom da su poduzete odgovarajuće mjere za zaštitu vodnog izvora od naknadnog zagađenja. Međutim, bez obzira na to jesu li koncentracije pojedinih konstituenata previsoke ili nisu, voda mora proći kroz neki korak kondicioniranja, pa makar to bilo samo kloriranje kojim se sprječava naknadna kontaminacija u distribucijskoj mreži. Općenito, neke osnovne kategorije tvari koje se u postupku kondicioniranja nastoje ukloniti su [14]:

- **neotopljene (suspendirane) čestice**

To su čestice poput mulja, algi, mikroorganizama, trulog biljnog materijala te drugih organskih i anorganskih tvari. Ukoliko se radi o velikim česticama, onda se one uklanjaju postupkom taloženja, a ako se radi o malim tj. koloidnim česticama, njih je najprije potrebno kemijski destabilizirati prije nego li ih se može ukloniti.

- **mikroorganizmi**

U mikroorganizme se ubrajaju bakterije, virusi i ostali organizmi koji također mogu biti prisutni u vodi u obliku koloidnih čestica. Iako neki od njih nisu opasni po zdravlje, postoje i oni koji jesu (*Salmonella*, *Vibrio cholera*, *Legionella pneumophila*, *Mycobacterium tuberculosis*, *E. Coli* itd.), a koje se uništava postupkom dezinfekcije vode.

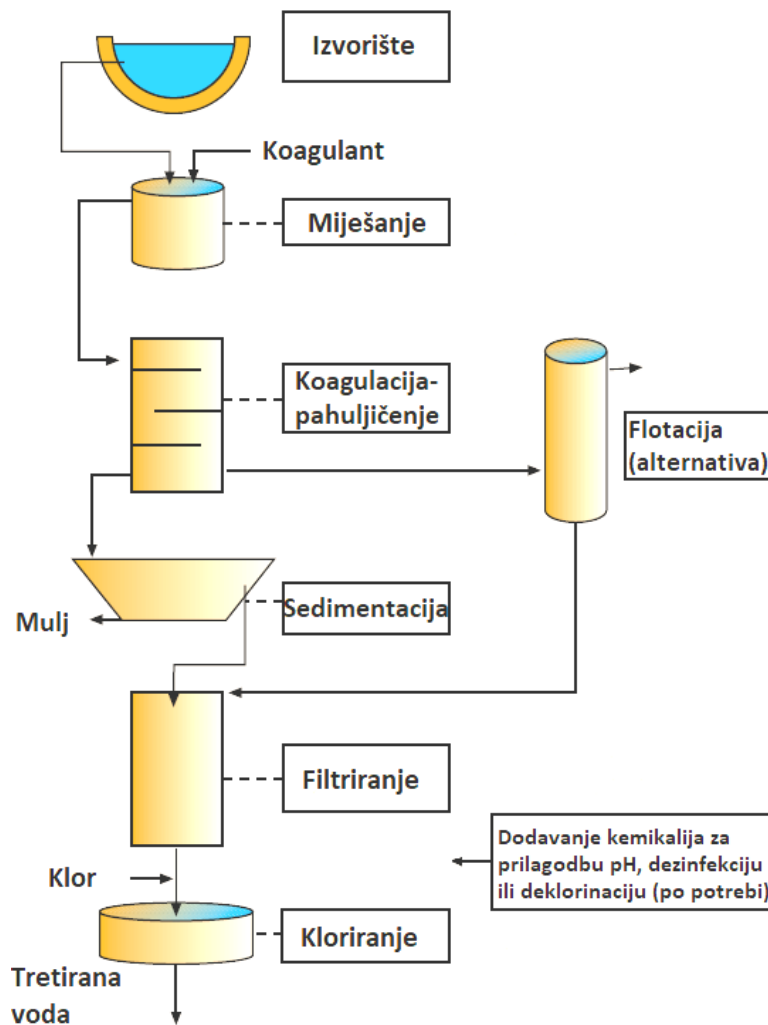
- **otopljene anorganske tvari**

One se mogu javiti u obliku teških metala poput žive ili arsena. Osim teških metala, mogu se pojaviti i neke druge anorganske tvari poput fluorida čija previsoka koncentracija može biti također štetna po zdravlje.

- **otopljene organske tvari**

Organske su tvari gotovo uvijek prisutne u vodi te većinom nisu štetne za zdravlje, međutim, ako su u obliku pesticida ili herbicida, onda jesu. Osim njih, u ovu kategoriju također spadaju i nusprodukti dezinficiranja. Obično se radi o tvarima koje se zovu trihalometani, koji nastaju tijekom reakcije klora sa zagađivačima u vodi.

Uobičajeni postupak kondicioniranja vode podrazumijeva sljedeće korake [7, 14]: uklanjanje neotopljenih i koloidnih tvari, dezinfekciju te kemijsku stabilizaciju. Ti koraci prikazani su zajedno sa pripadajućim postupcima koji se koriste za njihovo postizanje na slijedećoj slici.



Slika 9. Shema uobičajenog postupka kondicioniranja vode [14]

### **2.4.1. Koagulacija i pahuljičenje**

Dodavanjem tzv. koagulanata, destabiliziraju se male koloidne čestice u vodi na takav način da dolazi do njihova pahuljičenja, odnosno skupljanja u veće skupine. Ukoliko u sirovoj vodi prije ovog postupka ima algi ili nekih drugih biljnih ostataka, potrebno joj je prethodno dodati malu količinu klora (prethodno klorirati), kako bi ti ostaci odumrli prije nego se uđe u miješanje koagulanata [14].

### **2.4.2. Taloženje**

Taloženje slijedi nakon što se provede koagulacija i pahuljičenje čestica u vodi. Voda se ostavi neko vrijeme da miruje uslijed čega pahuljice koloidnih i krupnijih čestica padaju na dno tvoreći tamo naslage. Nakon toga, voda se propušta na filtriranje, a ono što preostane na dnu se uklanja [14].

### **2.4.3. Flotacija**

Kao alternativna metoda za taloženje, može se koristiti metoda flotacije. Ona podrazumijeva stvaranje zračnih mjehurića koji sa sobom pobiru relativno lagane čestice koje se onda skupljaju na površini vode. Takve nakupine zaposlenici mogu lako pokupiti prije puštanja vode na filtraciju [14].

### **2.4.4. Filtriranje**

Voda prolazi kroz sloj pijeska u posebno konstruiranom spremniku. Ovim postupkom uklanjaju se i one najmanje čestice. Općenito se u filtriranju koriste dvije vrste pješčanih filtriranja: brzo gravitacijski i sporo. Brzo filtriranje funkcionira kao što mu ime upućuje, pri velikim brzinama protoka, te je donji dio tog filtera potrebno povremeno ispirati. S druge strane, spora filtracija ne zahtjeva takva ispiranja, međutim, filter može biti ispran na način da se zamijeni gornji dio pijeska ali to je potrebno raditi u znatno većim intervalima nego što brzi filter treba ispirati [14].

### **2.4.5. Kloriranje**

Kloriranjem se provodi posljednji korak u sklopu kondicioniranja vode, a to je dezinfekcija. Ukoliko je vodni izvor kvalitetan i zaštićen od vanjskih zagađivača, ovo je ujedno i jedini korak potreban za puštanje vode u distribucijsku mrežu. Dezinfekcija podrazumijeva uništavanje mikroorganizama kroz neko vremensko razdoblje, čime se voda u bakteriološkom smislu dovodi u stanje zdravstveno ispravne vode za piće. Primarni cilj ovog postupka je uništavanje infektivnih mikroorganizama (intestinalne i fekalne bakterije, neke vrste parazita i virusa), zatim oksidacija anorganskih i organskih tvari te uklanjanje okusa i mirisa vode koji

su neprihvatljivi iz estetskih razloga. Kao najčešći postupak dezinfekcije upotrebljava se onaj uz pomoć plinovitog klora u sustavu za doziranje klora [7]. Spomenuti sustav prikazan je na idućoj slici.



**Slika 10.** Sustav za kemijsku dezinfekciju vode, izvor rijeke Rude [7]

Općenito, sustav za doziranje plinovitog klora sastoji se od više komponenti [15]: spremnika plinovitog klora, raznih izolacijskih i regulacijskih ventila, sustava automatske izmjene dotoka plinovitog klora, mjerača pritiska, klorinatora, ubrizgavača te mreže cijevi. U spremnicima se nalazi plinoviti klor koji se cijevima šalje u klorinator. Između spremnika i klorinatora nalazi se više izolacijskih ventila koji služe za zatvaranje dotoka klora iz spremnika ili kako bi se izolirao određeni dio cijevi u svrhu održavanja. Na tom dijelu sustava također se upotrebljavaju regulacijski ventili koji reguliraju pritisak u sustavu doziranja, kako se ne bi pojavili prekomjerni pozitivni i negativni tlakovi. Sustav automatske izmjene dotoka klora služi za promjenu izvora opskrbe, kada se trenutni spremnik potroši, kako bi se osigurao kontinuitet u doziranju. Kada plinoviti klor prođe kroz opisani dio sustava, dolazi do klorinatora koji djelatnicima održavanja služi za određivanje dezinfekcijske doze klora koja će se u ubrizgavaču unositi u vodu. Završni dio tog sustava je spomenuti ubrizgavač koji može biti izveden kao cijev čiji je otvor uronjen u vodu i kroz koju se plinoviti klor pušta u nju [15]. Kao najčešće ostale metode koje se koriste za dezinfekciju ili kao dopuna metodi

kloriranja su: toplinske (prokuhavanje vode), fizikalne (UV zračenje, ultrafiltracija) te kemijske (jod, ozon, klorov dioksid) [7].



Slika 11. WEDECO UV uređaji, Vodoopskrbni sustav Pakrac – Lipik, Vodozahvat Šumetlica [7]

## 2.5. Najčešće operacije u održavanju po komponentama sustava

### 2.5.1. Cijevi

#### 2.5.1.1. Lociranje cijevi i ostalih komponenti

Prije nego li se krene s provođenjem bilo koje operacije održavanja, djelatnici moraju točno znati gdje se cijevi nalaze i kakvog su materijala. Osim tih podataka, također je potrebno poznavati i točne lokacije svih čvorova, ventila te pristupnih okna. Sve te informacije trebale bi biti negdje ucrtane i dokumentirane, što u papirnatom obliku, što u digitalnom pomoću GIS-a (Geografskog informacijskog sustava). U nekim slučajevima, hidrant je dovoljna indikacija o tome gdje se neka cijev nalazi međutim, za bilo kakve ozbiljnije radove ili situaciju u kojoj podaci nisu dobro dokumentirani ili dostupni, potrebno je koristiti neku od sljedećih tehnika traženja cijevi [2]:

- **detektor metala**
- **detekcija signala poslanog u cijev preko nekog obližnjeg hidranta**

- **penetrirajući radar**
- **odašiljanja zvučnog vala**
- **infracrvena termografija**
- **iskapanje zemlje**

#### 2.5.1.2. *Lociranje pukotina u mreži*

Traženje pukotina, odnosno mjesta curenja vode u mreži, provodi se nekom od slijedećih tehnika [2]:

##### **a) Ubrizgavanjem plina**

Kod ove metode, iz cijevi je potrebno ispustiti svu vodu te ih popuniti plinom, najčešće vodikom, u rjeđim slučajevima helijem. Detektorom plina prolazi se putem koji slijedi lokaciju cijevi koje se ispituju te se mjeri curi li igdje plin ili ne.

##### **b) Osluškivanjem**

Za osluškivanje postoji li neka pukotina kroz koju curi voda ili ne, koristi se stetoskop kojeg djelatnik održavanja prislanja na cijev. Ovom metodom moguće je locirati samo pukotine koje su u neposrednoj blizini mjesta osluškivanja.

##### **c) Korelacijom podataka**

Ovom metodom se lociranje pukotine provodi pomoću mjerenja na dvije različite lokacije u cjevovodu. Senzori kojima se mjeri neka fizikalna veličina poput jačine zvuka ili vibracije, također mjere vrijeme potrebno da ti signali prođu između njih. Ta informacija služi za izračun udaljenosti mjesta gdje se potencijalna pukotina nalazi međutim, zbog mogućnosti da senzori detektiraju čitavi niz drugih zvukova i vibracija koji ne moraju nužno predstavljati pukotinu, njeno postojanje potrebno je dodatno potvrditi osluškivanjem pomoću mikrofona na dobivenoj lokaciji. Za provođenje korelacije, fizikalne veličine potrebno je detektirati jednim od sljedeća dva uređaja:

1. **Akcelerometrima** – mjere ubrzanje dinamičnih sila unutar medija s kojim su u kontaktu (u ovom slučaju metalne cijevi) odnosno vibracije. Usporedbom vibracijskih signala zabilježenih u prostoru između dva akcelerometra moguće je odrediti lokaciju poremećaja koji može upućivati na pukotinu. Ovi senzori nisu pogodni za upotrebu na plastičnim cijevima koje zbog svoje elastičnosti prigušuju vibracije, kao niti na cijevima velikih promjera.
2. **Hidrofonomima** – radi se o sensorima koji se mogu ugraditi u hidrante i koji su u izravnom dodiru sa vodom u sustavu gdje mjere zvukove za korelaciju.



**d) Unutarnje tehnike detekcije**

Kod ovih metoda, razni senzori i sonde smještaju se unutar cijevi gdje mjere potrebne stvari u kontaktu s vodom i po potrebi ih bežično odašilju. Ove tehnike prikladne su za upotrebu u cijevima velikih promjera.

**e) Senzori za bilježenje i analizu zvukova curenja**

Ovdje se radi o sensorima koje je moguće trajno ugraditi na elemente distribucijskog sustava. Ti senzori bilježe razne podatke, analiziraju ih te u slučaju da detektiraju neki neželjeni događaj, na primjer curenje, javljaju bežično djelatnicima održavanja sve potrebno za lociranje tog problema.

Tablica u nastavku prikazuje koje tehnike lociranja pukotina su pogodne za koje veličine cijevi i materijale.

**Tablica 12. Preporuka za korištenje tehnika lociranja pukotina u ovisnosti o promjeru cijevi [2]**

<b>Materijal cijevi</b>	<b>Mali promjeri (75 – 400 mm)</b>	<b>Veliki promjeri (veći od 300 mm)</b>
<b>Metalne</b>	a, b, c1, c2, e	c1, c2, e
<b>Azbestne</b>	a, c1, c2	d
<b>PVC</b>	a, c2	d
<b>Polietilenske</b>	a, c2	d

**2.5.1.3. Popravljanje ili zamjena napuknutih cijevi**

Koraci kod popravljanja ili zamijene cijevi su slijedeći [2]:

1. Locirati puknuće.
2. Zatvoriti izolacijske ventile i pribilježiti da su zatvoreni u dokumentaciji te ljepljivom trakom.
3. Osigurati radno područje potrebnim oznakama, regulacijom prometa i sl..
4. Doći do cijevi uklanjanjem svih prepreka i iskapanjem zemlje.
5. Isušiti svu vodu iz iskopane lokacije pomoću pumpe koja može pumpati i blato.
6. Popraviti ili zamijeniti cijev.
  - a. Kod manjih pukotina moguće je upotrijebiti spojnicu koja se fiksira preko nje kako bi ju zatvorila ili ako je moguće, pristupiti odmah zavarivanju te pukotine u slučaju da je ona vrlo mala.
  - b. Kod većih pukotina, oštećeni dio cijevi se izreže te se na to mjesto najčešće ugrađuje PVC cijev pomoću spojnih elemenata (postoje elementi za spajanje PVC

- cijevi sa cijevima od drugih materijala). Ukoliko je oštećena cijev od azbestnog cementa, nju se nikako ne smije rezati već se pristupa potpunom zamijeni cijevi.
- c. Kod vrlo velikih oštećenja ili bilo kakvog oštećenja azbestnih cijevi, pristupa se zamijeni kompletne cijevi.
7. Osigurati da je nova cijev kao i svaka izložena tijekom postupka popravka ili zamijene, pažljivo očišćena i dezinficirana. To znači da treba paziti da kod skladištenja, transporta i ugradnje cijevi one budu očuvane u što čistijem stanju. Nakon ugradnje ili popravka potrebno je cijevi dezinficirati puštanjem vode koja sadrži barem 25 mg/L klora i to u trajanju od 24 h. Nakon tog perioda, sadržaj rezidualnog klora trebao bi biti barem 10 mg/L. Ako se to želi, moguće je cijev popuniti vodom koja sadrži 100 mg/L klora, pa dezinfekcijski postupak traje svega 3 sata, nakon čega mora biti barem 50 mg/L rezidualnog klora. Nakon što se provede opisani postupak dezinfekcije, cijev se ispire sa uobičajenom pitkom vodom iz sustava, nakon čega je sve spremno za puštanje u pogon.
  8. Polako otvaranje jednog ili više prethodno zatvorenih izolacijskih ventila uz ozračivanje kroz hidrante.
  9. Provjeriti jesu li svi izolacijski i ispusni ventili vraćeni u svoje izvorno, traženo stanje.
  10. Provesti dodatna ispiranja sustava kroz hidrante ako je to potrebno, kako bi se osiguralo uklanjanje bilo kakvih krutih tvari.
  11. Kada se ponovno postigne traženi radni tlak u sustavu, vrše se provjere kvalitete izvršenih operacija zamijene ili popravka.
  12. Vraćanje prostora oko cijevi u prvotno stanje (popravak podloge, zakapanje itd.).
  13. Čišćenje okolnog radnog područja, uklanjanje svih oznaka i povlačenje radnika sa terena.
  14. Popunjavanje izvješća o provedenoj korektivnoj radnji, koji mora sadržavati analizu zašto je do potrebe za popravkom došlo i kako se ta operacija obavila. Taj izvještaj potrebno je pohraniti kako bi bio dostupan na uvid kada kasnije zatreba.

### **2.5.2. Ventili i ostali elementi sustava**

Glavne vrste ventila koji se koriste u distribucijskom sustavu su: izolacijski, zračni, ispusni, nepovratni te kontrolni. Izolacijski služe za zatvaranje nekog dijela sustava kako bi se pomoću ispusnog mogao isprazniti njegov sadržaj prije nego li se krene s izvođenjem nekog zahvata. Danas se umjesto ispusnih ventila obično koriste hidranti kojima je također moguće ispustiti vodu iz sustava kada to treba. Što se tiče zračnih ventila, oni se automatski otvaraju kada dođe do nakupljanja zraka u sustavu koji je tamo dospio zbog radova ili zbog zračnih mjehura u

vodi na izvoru. Prije zračnog ventila, obavezno je potrebno ugraditi jedan izolacijski kako bi se omogućila zamjena zračnog ventila kada je to potrebno. Hidranti se tu ponovno javljaju kao moguća zamjena za ventile pri ozračivanju. Nadalje, u sustavu je važno osigurati da se voda kreće samo u jednom smjeru, za to služe nepovratni ventili. Ti ventili trebali bi posjedovati neki mehanizam koji osigurava zatvaranje i u uvjetima kada voda u sustavu miruje, kako se ne bi vraćala i izazivala naglo zatvaranje koje može oštetiti komponente. Na kraju ostaju kontrolni ventili koji služe za upravljanje protokom i tlakovima. Ostale elementi koji se koriste u vodoopskrbnoj mreži su vodomjeri, pristupna okna, oznake te potisni blokovi. Potisni blokovi su elementi koji se ugrađuju na mjesta u sustavu gdje se javljaju neke unutarnje sile uzrokovane promjenama toka, slijepim završecima, promjenama u promjeru cijevi ili protokom kroz neke ventile. Takve sile na tim mjestima mogu uzrokovati vibracije i pomicanje komponenti, zbog čega se ugrađuju neki potisni elementi koji ih učvršćuju i osiguravaju da do toga ne dođe [2].

#### 2.5.2.1. Održavanje izolacijskih ventila

Najčešći postupci održavanja izolacijskih ventila uključuju [2]:

- redovne preglede (jednom godišnje) koje uključuju provjeru da li ventil curi, zatezanje matica i vijaka uz oprez da ne dođe do prekomjernog zatezanja, uklanjanje nečistoća, po potrebi primjenu zaštite od korozije, podmazivanje te zamjenu brtvi (svakih 5 godina)
- otvaranje i zatvaranje u redovitim intervalima, čime se provjerava rade li ispravno
- otvaranje i zatvaranje kutije sa zupčanicima, ako se koristi ventil s njima, barem jednom godišnje, njihovo podmazivanje i zamjena te kutije po potrebi

#### 2.5.2.2. Održavanje zračnih ventila

Kod zračnih ventila glavni problem je nakupljanje naslaga koje ga mogu začepiti ili ulazak insekata zbog neadekvatne mrežice. Postupci održavanja zračnih ventila su [2]:

- redovni pregledi kojima se provjerava da bilo koji zračni ventil nije potopljen u vodu, što zbog kiše, što zbog puknuća cijevi te da zaštita od insekata nije oštećena
- kada se utvrdi potreba za popravkom, ventil se odmah mijenja na licu mjesta sa novim, a stari se odnosi u radionicu gdje ga je moguće dodatno pregledati, popraviti i podmazati te će on dalje poslužiti za neku novu zamjenu

### 2.5.2.3. Održavanje ispusnih ventila

Ispusne ventile potrebno je dobro označiti kako bi bili jasno uočljivi, kako bi ih zaposlenici na održavanju bez većih problema mogli pronaći kada je to potrebno. Njihovo održavanje sastoji se od [2]:

- primjene korozijske zaštite kako do nje ne bi došlo, jer je ventil izložen kišnici
- redovnih pregleda, čišćenja od nečistoća iz okoline te testiranja rade li ispravno

### 2.5.2.4. Održavanje nepovratnih ventila

Ovi ventili podliježu redovnim pregledima kao i ostali, ali obično ne zahtijevaju previše operacija održavanja. Pregledima se utvrđuje rade li ispravno ili ne, ako ne rade, moraju se zamijeniti u potpunosti sa novim [2].

### 2.5.2.5. Održavanje kontrolnih ventila

Održavanje kontrolnih ventila sastoji se od redovitih pregleda (barem 2 puta godišnje), čišćenja nečistoća i servisiranja ili zamijene (svakih 5 godina) [2].

### 2.5.2.6. Održavanje hidranta

Operacije održavanja hidranata podrazumijevaju: redoviti godišnji pregled koji uključuje uklanjanje i čišćenje bilo kakvih prepreka koje bi onemogućile pristup hidrantu, otvaranje i zatvaranje ventila na hidrantu kako bi se provjerilo radi li ispravno, po potrebi zatezanje vijaka i matica kao i primjenjivanje zaštite od korozije [2].

### 2.5.2.7. Održavanje vodomjera

Prioritet u održavanju vodomjera imaju oni koji se koriste na ulazu u distribucijsku mrežu te područni vodomjeri. Operacije održavanja koje se nad njima provode uključuju:

- čišćenje ili zamjenu filtera i kućišta
- sprječavanje svih uočenih curenja
- zamjenu oštećenih brojača

Praksa je da se popravci obično rade na vodomjerima manjim od 50 mm, dok se kod većih radi kompletna promjena novim. Veći vodomjeri moraju biti pregledani jednom godišnje na način da se rastave kako bi se dobio uvid u unutrašnji mehanizam i stanje njegove istrošenosti. Svi vodomjeri obuhvaćeni planom održavanja moraju se nalaziti u jedinstvenoj bazi podataka (digitalnoj ili papirnatoj) koja se popunjava pri svakoj operaciji održavanja sa informacijama o stanju i poduzetim radnjama. Na kraju, moguće je testirati razinu preciznosti

mjerenja vodomjera međutim, to se obično u praksi ne radi jer su za to potrebni certificirani laboratoriji. Što se tiče kućnih vodomjera, oni se obično pregledavaju prilikom zaprimanja pritužbe krajnjeg korisnika ili pri periodičnom popisivanju potrošnje, prilikom čega djelatnik vodovoda može prijaviti neko uočeno oštećenje ili kvar, temeljem čega osoblje održavanja onda dalje postupa [2].

#### 2.5.2.8. Održavanje oznaka

Oznake se upotrebljavaju u svrhu ukazivanja na neku poziciju od interesa za osoblje vodovoda. Označena može biti lokacija ispusnog ventila, nekog važnog spoja, točka prelaska cijevi ispod ceste itd.. Oznaka može biti u obliku boje, metalne oznake ili stupa. Sve navedene oznake moraju se periodično obnavljati kako bi i dalje služile svojoj svrsi [2].

### 2.5.3. Vodospremnici

Vodospremnici služe za balansiranje između ponude i potražnje te predstavljaju nužnu komponentu sustava zbog varijabilnosti u potrebama potrošača. Obično se grade od betona u obliku bazena u koji se pumpa tretirana i ispitana voda spremna za daljnju distribuciju. Oko bazena izgrađena je čitavi objekt koji štiti tretiranu vodu od vanjskih elemenata.

#### 2.5.3.1. Održavanje vodospremnika

Vodospremnike je potrebno periodično čistiti silaskom ljudi u njih. Za vrijeme čišćenja, opskrba se nastavlja pomoću sekundarnog vodospremnika ili iz drugog dijela vodospremnika ako se sastoji od više njih. Operacija čišćenja se sastoji od sljedećih koraka [14]:

1. Pražnjenje vodospremnika do najniže moguće razine pomoću prirodnog odlijeva uslijed normalne potrošnje u sustavu nakon čega se otvara ispusni ventil kako bi se ispraznila preostala voda.
2. Očistiti nakupljene naslage na dnu vodospremnika pometanjem ili ispiranjem s visokotlačnim peraćem. Voda sa ispranim sedimentima ispušta se ponovno pomoću ispusnog ventila. Prilikom pranja treba paziti da se ne ošteti vodootporna obloga vodospremnika.
3. Ispiranje zidova spremnika koristeći deterdžent i visokotlačni perać. Osigurati da su svi spojevi i kutovi dobro oprani.
4. Dobro isprati pod i zidove kako ne bi ostali neki ostaci korištenog deterdženta.
5. Detaljno pregledati sve spojeve i kutove te provesti popravke po potrebi.

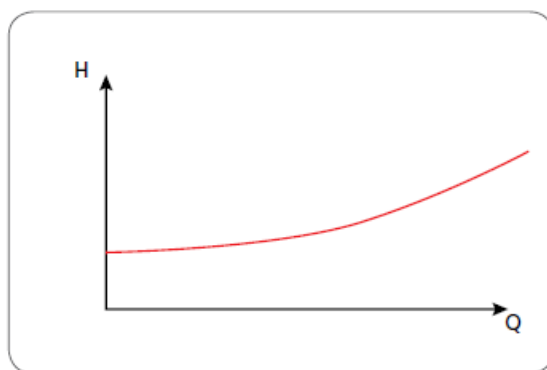
6. Dezinficirati spremnik vodenom otopinom čija je koncentracija klora 50 mg/L i pH 7.5 (što je optimalno za dezinfekcijsko djelovanje klora). Nakon što se sve dobro ispere i osoblje izađe iz spremnika pomoću ljestava, potrebno je isprati ljestve sa istom otopinom. Klorirana voda se ispušta kroz ispusni ventil te je iz nje potrebno ukloniti klor kako se okoliš ne bi zagadio.
7. Nakon što prođe barem 6 sati, vodospremnik se ponovno puni pitkom vodom i vraća u pogon.

#### 2.5.4. Pumpe

U vodoopskrbnim sustavima najčešće se koriste centrifugalne pumpe. To su pumpe koje koriste impeler kako bi generirale centrifugalne sile. Njih se koristi za crpljenje vode te stvaranje pritiska u sustavu [2].

##### 2.5.4.1. Održavanje pumpi

Operacije održavanja pumpi temelje se na mjerenju protoka i tlakova na ulaznoj i izlaznoj strani uz kontrolu da su mjerni uređaji u dobrom stanju i precizni. Pomoću prirasta tlaka (razlike izmjerenog tlaka na ulazu i izlazu pumpe) moguće je izračunati visinu dobave pumpe pri nekom protoku. Kada se ta dobivena točka ucrtta na dijagram karakteristike pumpe (h-Q dijagram) u specifikacijama njenog proizvođača, on bi se trebao otprilike nalaziti na toj krivulji ako je sve u redu. Kod navedenog postupka, nastoji se ucrtati nekoliko točaka na toj krivulji kako bi se napravila što bolja kontrola. Više točaka dobiju se variranjem protoka pomoću ventila na dobavnoj strani pumpe. U slučaju da točke previše odstupaju, potrebno je prekontrolirati mjerne uređaje. Ako je sve u redu s tim uređajima, potrebno je izvršiti daljnja ispitivanja pumpe kod ovlaštenih institucija, a postojeću pumpu zamijeniti novom. Ako je moguće, pumpu treba servisirati kako bi se mogla ponovno koristiti [2].



Slika 12. Dijagram karakteristike pumpe [2]

### **2.5.5. Uređaji za dezinfekciju vode**

U ovom poglavlju biti će spomenuta samo dva najvažnija uređaja za dezinfekciju u postupku kondicioniranja vode, a to je uređaj za dezinfekciju plinovitim klorom i uređaj za UV zračenje.

#### *2.5.5.1. Održavanje uređaja za dezinfekciju klorom*

Održavanje ovog uređaja sastoji se od uobičajenog održavanja ventila, provjere i sprječavanja curenja plinovitog klora koji je u većim količinama opasan za ljude, nadzora nad spremnicima klora kako se ne bi dogodilo da ga nema dovoljno za željenu dezinfekciju. Ukoliko se na vrijeme ne uoči nedostatak klora, to može rezultirati narušavanjem kvalitativne cjelovitosti vodoopskrbnog sustava. Također je potrebno provjeravati radi li vodena crpka koja dovodi vodu na dezinfekciju, jesu li svi tlakovi u sustavu doziranja unutar dozvoljenih vrijednosti te radi li sustav grijanja i ventilacije ispravno, u prostoriji u kojoj je sustav dezinfekcije smješten. Ta prostorija mora imati dobru cirkulaciju zraka, pogotovo kada se provode neke operacije održavanja, te zaposlenici trebaju poduzeti sve potrebne mjere vlastite zaštite, kako ne bi došlo do neočekivanog curenja plinovitog klora.

#### *2.5.5.2. Održavanje uređaja za UV zračenje*

Potrebno je mijenjati UV lampu u pravilnim vremenskim intervalima, ovisno o proizvođačkim specifikacijama. Uređaji za UV zračenje obično imaju mjerač intenziteta koji može oglasiti alarm kada je potrebno zamijeniti lampu kako ne bi došlo do narušavanja kvalitete vode u distribucijskoj mreži [7]. Ti mjerači također mogu poslužiti i za utvrđivanja kada se upotrebljava previsoka razina radijacije za dezinfekciju, pa se ona regulira ili djelomičnim gašenjem (ako je to moguće) ili preko regulacije jačine struje u uređaju. UV lampama se umeće u utor napravljen od kvarca, te je njega potrebno pri svakoj zamijeni lampe očistiti sa kiselom otopinom. Osim navedenog, potrebno je redovito provoditi operacije održavanja nad napajanjem ovog uređaja, kako bi se umanjili potrebni prekidi u radu te osiguralo ispravno doziranje radijacije u svakom trenutku [15]. Od djelatnika održavanja se očekuje da predvide trenutke u kojima će se zamijene provesti te da osiguraju posjedovanje adekvatnih komponenti i opreme onda kada to treba [14].

## 2.6. Mogućnosti primjene strojnog učenja u održavanju

Danas je sve više fizičkih sustava pametno, zahvaljujući upotrebi senzora i naprednih tehnologija, koje su postale cjenovno prihvatljivije i dostupnije nego ikada prije. Slijedeći taj trend, javljaju se i pametne vodoopskrbne mreže koje upotrebljavaju najnovije tehnologije i sustave poput: daljinske detekcije curenja i gubitaka vode, rane detekcije zagađenja, daljinskog prikupljanja raznih podataka u stvarnom vremenu, analize tih podataka, predviđanja potražnje te optimizacije potrošnje energenata. S obzirom da takvi sustavi generiraju veliku količinu podataka, javlja se potreba za pronalaskom tehnika analize i modeliranja, koje će pružiti pouzdan, efikasan i precizan prikaz kompleksnih, nelinearnih pojava koje se javljaju unutar njih [1]. U tu svrhu razvijaju se i koriste razni statistički modeli, ali osim njih, moguće je upotrebljavati i čitavi niz drugih tehnika i algoritama. Među algoritmima koji su najpogodniji za opisivanje nelinearnih pojava u realnim sustavima, su oni iz domene strojnog učenja. Ono što je zajedničko tim algoritmima jest princip rada koji podrazumijeva samostalan odabir i podešavanje svih veličina nekog traženog modela, bez potrebe korisničkog uplitanja, izuzev podešavanja parametara potrebnih za izvođenje samog algoritma. Kod primjene algoritma strojnog učenja razlikuju se dva pristupa koja treba imati na umu: nadzirano učenje i nenadzirano učenje. Nadzirano učenje podrazumijeva posjedovanje podataka u kojima su svim ulaznim podacima dodijeljeni odgovarajući izlazni na temelju nekog mjerenja ili iskustva, pa je zadatak algoritma učiti pomoću njih. Ovim pristupom mogu se rješavati dvije osnovne vrste problema: klasifikacija i regresija. Kod regresije, algoritmi temeljem dostupnih ulaznih i izlaznih podataka nauče kako su oni međusobno povezani, što rezultira modelom koji je sposoban vršiti predviđanja. Što se klasifikacije tiče, ona podrazumijeva da algoritam nauči neke podatke svrstavati u klase kojima pripadaju, također na temelju naučenog iz podataka u kojima su klase unaprijed poznate. S druge strane, nenadzirano učenje je pristup koji za razliku od nadziranog, ne koristi neke podatke za učenje. To znači da nema neku ciljnu varijablu koju pokušava naučiti, nego nastoji pronaći neke pravilnosti u ulaznim podacima, koje ljudi inače ne uočavaju. U nenadzirano strojno učenje spada grupiranje, procjena gustoće i smanjenje dimenzionalnosti podataka [16].



### **2.6.1. Primjena genetskog programiranja u održavanju vodoopskrbnih sustava**

Ovaj rad bavi se tehnikom genetskog programiranja (GP) koja spada u kategoriju nadziranog učenja. Mogućnosti njegove primjene u održavanju vodoopskrbnih sustava su mnoge, počevši od hidrološkog i hidrauličkog modeliranja, traženja pukotina u sustavu, razvijanja optimalnog plana izmjene komponenti sustava, vibrodijagnostike, modeliranja kvalitete vode u sustavu i sl.. Hidrološko modeliranje podrazumijeva razvijanje matematičkih modela koji predviđaju kretanja razina podzemnih i površinskih voda te utjecaj vegetacije i oblika vodotoka na brzinu nailaženja vode u vodozahvatima. Pomoću tako dobivenih modela, mogu se razviti bolje strategije održavanja vodnih tokova, a mogu poslužiti i za bolje projektiranje vodozahvata i crpnih stanica, koje će umanjiti naknadnu potrebu za održavanjem. GP algoritam također može poslužiti za predviđanje zagađenja uzrokovanog okolinom vodnog izvora. Pomoću mjerenja ključnih pokazatelja onečišćenja na mjestu njihova nastanka (industrijska zona, gradsko područje i sl.), primjenom GP algoritma moguće je razviti model vremenske serije, koji će predvidjeti mogućnost zagađenja vodnih izvora u budućnosti. Ukoliko inženjeri održavanja uspješno razviju takve modele, mogu stvoriti velike uštede dobrim planiranjem oko ugradnje novih uređaja potrebnih za pročišćavanje u budućnosti. Nadalje, postavljanjem senzora za mjerenje tlakova i brzina vode u distribucijskoj mreži, moguće je razviti hidraulički model mreže. Taj model može poslužiti za pronalazak kritičnih mjesta u opskrbi (previsoki pozitivni ili niski tlak), eliminaciju tlačnih udara, eliminaciju povratnih tokova, optimizaciju rada pumpi temeljem potreba sustava itd.. Stavljanjem senzora mjerenja vibracija na komponente kao što su pumpe, moguće pomoću GP algoritma temeljem povijesnih podataka o kvarovima, razviti model predviđanja kvara. Pomoću takvih sličnih modela, značajno se poboljšavaju strategije izmjene komponenti kao što su pumpe i cijevi u distribucijskom sustavu. Naposljetku, također je moguća primjena kod procjene kvalitete vode, pa je tako teza ovog rada da je GP moguće upotrebljavati u predviđanju rezidualnog klora u vodoopskrbnim sustavima, kao temeljnog pokazatelja mikrobiološke kvalitete vode u sustavu [1, 17, 18] . Više o GP tehnici i načinu na koji će se rezidualni klor modelirati, biti će izneseno u nastavku.

### 3. OSNOVNO O GENETSKOM PROGRAMIRANJU

#### 3.1. Začetak

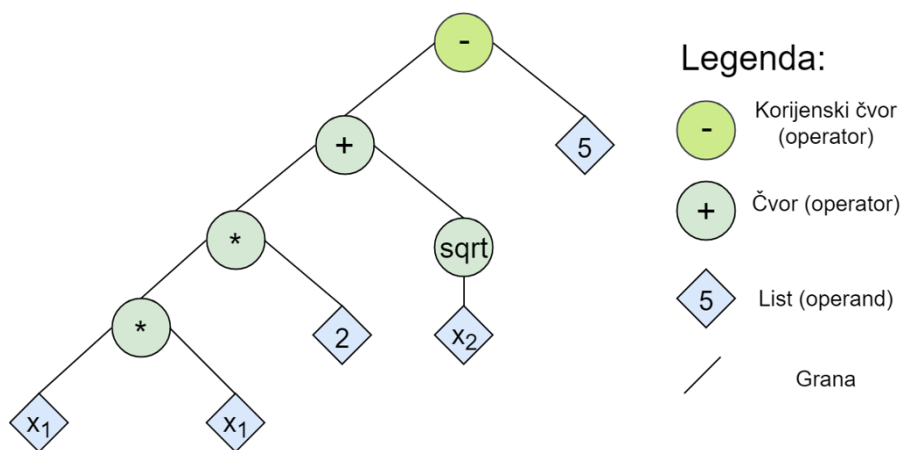
Pojam genetskog programiranja osmišljen je neovisno od strane dva istaknuta znanstvenika: Johna R. Koze i Huga de Garisa 1990. godine, koji su ga upotrijebili u svojim radovima kako bi nadjenuli ime tehnikama evolucijskih algoritama koje u to vrijeme razvijaju. Tehnika koju de Garis razvija, temeljila se na upotrebi genetskih algoritama (GA) za razvijanje umjetnih neuronskih mreža, dok su Kozini radovi predlagali upotrebu hijerarhijske strukture u obliku stabla, kojima bi se unaprijedile mogućnosti tehnike genetskih algoritama. Tek 1992. godine, Koza postaje osoba kojoj se pripisuje najviše zasluga za nastanak tehnike genetskog programiranja (GP) zahvaljujući svojoj knjizi naslova *Genetsko programiranje – O programiranju računala pomoću prirodne selekcije*, koja je predstavljala značajno djelo u području evolucijskog računarstva. Nakon te knjige, pojam genetskog programiranja veže se isključivo uz njegovu tehniku evolucije programa u strukturama nalik stablu, dok de Garis naziv napušta te za svoju tehniku počinje koristiti naziv evolucijsko inženjerstvo [19]. Još 1988. godine jedan od pionira evolucijskog računarstva, Kenneth De Jong tijekom svojeg istraživanja u području tada već dobro poznatih genetskih algoritama (GA), došao je do zaključka da postoje određeni nedostaci ove metode koji onemogućuju njenu primjenu u svrhu rješavanja praktičnih problema. Temeljni nedostatak genetskih algoritama leži u načinu prikazivanja rješenja u obliku binarnih nizova fiksne duljine koji nisu pogodni za prikazivanje npr. modela (ili računalnih programa) te kompliciraju implementaciju iterativnog postupka, svojstvenog za sve evolucijske algoritme. Takva reprezentacija rješenja ne podržava hijerarhijsku organizaciju zadataka i pod-zadataka koja je potrebna za prikazivanje računalnih programa. Mogućnost jednostavnog prikaza programa omogućila bi rješavanje složenije formuliranih problema, a jednostavnija implementacija iterativnog postupka, olakšala bi traženje što boljeg konačnog rješenja [20]. Tehnika genetskog programiranja iznesena u Kozinoj knjizi, nastala je kao reakcija na spomenute nedostatke genetskih algoritama, te ih rješava na elegantan način koji će detaljnije biti opisan ovim radom u poglavljima koja slijede.

### 3.2. Definicija GP-a

Tehnika genetskog programiranja (GP-a) pripada kategoriji evolucijskih algoritama. Ona omogućuje istraživanje prostora rješenja u obliku modela ili računalnih programa, predstavljajući tako proširenje metode genetskih algoritama. To je tehnika koja je sposobna automatski riješiti problem bez potrebe da korisnik unaprijed zna oblik rješenja. Kod GP-a moguće je formulacijom onoga što se traži na višoj razini (npr. pronalazak upravljačkog algoritma za igranje robo-nogometa [21]) riješiti problem (tj. za dani primjer, osposobiti minijaturne robote za igranje nogometa). Takvo nešto postiže se stohastičkim generiranjem programa koji se zatim iterativno evoluiraju u nove programe s ciljem traženja poboljšanja u svakom koraku sve dok se ne dođe do konačnog rješenja koje zadovoljava osnovnu formulaciju problema. Upravo je zahvaljujući elementu nasumičnosti, genetsko programiranje uspješno u evoluiranju novih i neočekivanih rješenja, što je karakteristika preuzeta od prirodnih procesa koje GP imitira [22].

### 3.3. Prikaz u obliku stabla

Strukture koje GP evoluiraju, kao što je to već bilo spomenuto, imaju oblik stabla sagrađenog od čvorova i listova. Listovi u stablu mogu sadržavati: varijable koje predstavljaju vanjske ulaze, funkcije bez argumenata koje izbacuju neku vrijednost (npr. funkcija *dist\_to\_ball* koja izbacuje udaljenost mini robota od lopte u primjeru traženja upravljačkog algoritma za igranje robotskog nogometa [21]) te konstantu koja može biti unaprijed definirana, nasumično generirana kao dio procesa stvaranja stabla ili stvorena mutacijom. Varijable, funkcije i konstante u GP-u nazivaju se operandi te se obično nalaze na završetku svake od grana. Čvorovi u stablu pak, sadrže operatore čiji je oblik vođen prirodom domene kojoj rješavani problem pripada. Oni mogu biti: aritmetički, logički, transcendentni, uvjetni operatori ali i operatori petlje te subrutinski operatori [19]. Svaki operator u operatorskom setu radi s fiksnim brojem ulaznih argumenata (eng. *arity*) koji direktno utječe na razgranatost stabla u postupku njegova nastanka. Početni, odnosno najviši čvor u nekom stablu nosi naziv korijenski čvor te se pod njim grupiraju svi ostali čvorovi i listovi [20, 22]. Slika koja prikazuje izgled strukture u obliku stabla sa pripadajućim granama sastavljenim od čvorova i listova prikazana je u nastavku.



$$y = 2x_1^2 + \sqrt{x_2} - 5$$

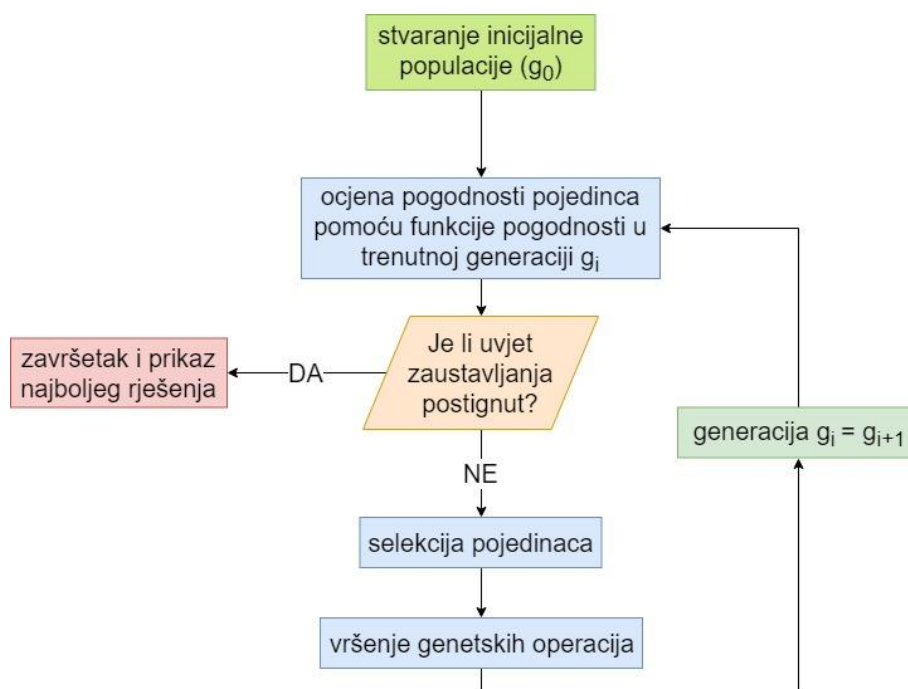
Slika 13. Način prikaza u obliku stabla

Veličina prethodno prikazanog stabla određena je brojem čvorova i listova od kojih se ono sastoji te je taj pojam potrebno razlikovati od pojma dubine stabla. Dubina stabla je za razliku od veličine, jednaka dubini najnižeg pripadajućeg lista, a koja se definira kao minimalni broj rubova kroz koje je potrebno proći da bi se od njega došlo do početnog, korijenskog čvora za kojeg se uzima da je na dubini nula. Pomoću tih mjera definirana je kompleksnost stabala koja može biti vezana isključivo uz veličinu ili kao ukupna kompleksnost izraza koja u obzir uzima i veličinu i dubinu. To svojstvo stabla nastoji se minimizirati a da se pritom zadrži što je veća moguća točnost modela kojeg reprezentira [22, 23].

### 3.4. Princip rada GP-a

Sam postupak traženja rješenja GP-om započinje nasumičnim stvaranjem inicijalne populacije. Za dobivanje inicijalne populacije, najčešće se koriste metode naziva: *Full*, *Grow* te *Ramped Half-and-Half*. Te metode iz takozvanog primitivnog seta, koji se sastoji od operatora i operanda, nasumično biraju argumente kojima tvore čvorove i listove stabla svakog pojedinca u populaciji. Postupak inicijalizacije, kao što to samo ime kaže, provodi se samo jednom, na početku izvršenja genetskog programiranja. Nakon što je inicijalizacija uspješno provedena, slijedi korak ocjenjivanja dobivene inicijalne populacije prema uspješnosti u rješavanju postavljenog problema. Ocjenjivanje se provodi pomoću funkcije pogodnosti koja ovisi o problemu koji GP rješava. Nakon što je svim pojedincima inicijalne populacije dodijeljen iznos pogodnosti, slijedi stvaranje nove populacije (prve generacije)

rješenja probabilističkom selekcijom onih najpogodnijih iz inicijalne generacije. Nad odabranim pojedincima primjenjuju se konzervatorske i inovacijske operacije po uzoru na genetske operacije u prirodi. U inovacijske operacije spada mutacija koja probabilistički vrši izmjenu male količine informacija u pojedincu kako bi se osiguralo razmatranje novih, inovativnih potencijalnih rješenja u populaciji. S druge strane, u konzervatorske operacije spadaju križanje i reprodukcija koji se koriste u svrhu osiguravanja da se u budućim generacijama već naučene korisne informacije sačuvaju i dodatno poboljšaju. Križanje to postiže miješanjem određene količine nasumično odabranih dijelova rješenja između pojedinaca u populaciji, dok reprodukcija uzima neke pojedince (obično one najpogodnije) iz populacije te ih izravno bez izmjena prenosi u iduću generaciju. Stvaranjem nove populacije, dobiva se nova generacija nad kojom se ponovno izvršava selekcija i primjenjuju spomenute operacije. Formiranje novih generacija vrši se sve dok neki pojedinac u nekoj generaciji ne ispuni postavljeni kriterij uspješnosti rješavanja problema tj. dok se ne ispuni kriterij zaustavljanja [19, 20, 22]. Prethodno opisan princip rada GP-a, grafički je prikazan na sljedećoj slici kako bi se olakšalo njegovo shvaćanje.



Slika 14. Princip rada standardnog GP algoritma

## 4. GPTIPS – PLATFORMA ZA GENETSKO PROGRAMIRANJE I SIMBOLIČKO RUDARENJE PODATAKA U MATLAB-U

### 4.1. Uvodne napomene o GPTIPS-u

GPTIPS je kratica punog engleskog naziva *Genetic Programming Toolbox for the Identification of Physical Systems* koji preveden na hrvatski znači: Skup alata za identifikaciju fizičkih sustava pomoću genetskog programiranja. Radi se o skupini alata (eng. *toolbox*) za primjenu u softverskom paketu MATLAB. Zahvaljujući tim alatima, moguća je jednostavnija primjena tehnike GP-a za tzv. simboličko rudarenje podataka odnosno traženje empiričkog matematičkog modela temeljem podataka prikupljenih u nekom sustavu ili procesu. Alat je besplatno dostupan za preuzimanje, korištenje, pa i modifikaciju istraživačima diljem svijeta putem web stranice na kojoj se osim alata nalaze i neke osnovne autorske upute za korištenje [24]. Autor ovog alata Dominic Searson, podatkovni je stručnjak koji se bavi područjima strojnog učenja i umjetne inteligencije. Doktorirao je na Sveučilištu Newcastle u Velikoj Britaniji radom koji se bavi modeliranjem nelinearnih sustava i multivarijatnom statističkom analizom. U sklopu svojeg istraživačkog rada napisao je GPTIPS skupinu alata za MATLAB u obliku otvoreno dostupnog koda (eng. *open source*), koja je citirana u preko 300 znanstvenih radova različitih autora i znanstvenih područja. GPTIPS olakšava provođenje genetskog programiranja nad učitanim setom podataka, istraživanje populacije evoluiranih modela, istraživanje ponašanja tih modela te izvoz rezultata rada u nizu pogodnih formata (npr. grafički, LaTeX, samostalna *.m* datoteka) [23, 25]. U najnovijoj verziji 2.0 GPTIPS-a, autor je veliki naglasak stavio na upotrebu tog alata u razvijanju tzv. multi-genskih pojedinaca odnosno na nov način multi-genskog genetskog programiranja, o čemu će više biti izneseno u narednim poglavljima.

### 4.2. Simboličko rudarenje podataka u GPTIPS-u

Simboličko rudarenje podataka je pojam u koji se ubraja nekoliko tehnika za pronalaženje rješenja koje se mogu iskoristiti GPTIPS skupinom alata, a to su simbolička: regresija, klasifikacija te optimizacija. Simbolička optimizacija podrazumijeva generiranje jednadžbi koje su optimizirane po nekom kriteriju, simbolička klasifikacija omogućuje predviđanje neke diskretne kategorije na temelju ulaznih varijabli, dok simbolička regresija označava generiranje neke simboličke jednadžbe koja predviđa kontinuiranu odzivnu varijablu pomoću

ulaznih tj. prediktorskih varijabli. Upravo je simbolička regresija ona tehnika koja se najčešće primjenjuje kod GP-a te je GPTIPS alat najprikladniji za nju [25]. Termin rudarenja podataka podrazumijeva pristup nekoj velikoj količini mjerenih podataka, koji se pretražuju u svrhu pronalaska nekog implicitnog, skrivenog znanja. Kada se u tu svrhu primjenjuju spomenute tehnike regresije, klasifikacije i optimizacije pomoću nekih pogodnih algoritama strojnog učenja, kao što je GP, uobičajenoj da se podaci dijele na trening i testni skup. Trening skup podataka u tom slučaju služi kao podloga algoritmu kako bi naučio ono traženo znanje, a testni skup kako bi se provjerila kvaliteta tog naučenog znanja. Osim trening i testnog skupa podataka, prilikom primjene takvih algoritama, često se koristi i tzv. validacijski skup. U nastavku će biti rečeno više o skupovima podataka koji se mogu rabiti kao i o tehnikama simboličkog rudarenja podataka za koje GPTIPS skupina alata može poslužiti.

#### 4.2.1. Simbolička regresija

Simbolička regresija usko je povezana sa klasičnom tehnikom regresijske analize. Temeljna razlika koja simboličku regresiju, zahvaljujući GP-u, čini daleko pogodnijom za korištenje leži u tome da klasična tehnika od korisnika unaprijed zahtjeva definiciju izgleda traženog modela (linearni, polinomni ili nelinearni), dok kod simboličke regresije, strukturu nije potrebno unaprijed poznavati već se ona automatski generira u procesu rada GP-a. Osim strukture, automatski se razvijaju i svi ostali traženi parametri u modelu te se provodi samostalni izbor najutjecajnijih svojstava tj. ulaznih varijabli koje će formirati model. Na taj način, GP omogućuje lakši prikaz ne-linearnih pojava i ponašanja [23]. Navedena svojstava karakteristična su za neki realni sustav ili proces, stoga je GP, za razliku od ostalih metoda, najpogodniji za upotrebu kod rješavanja praktičnih problema s kojima se inženjeri susreću u industriji. Svaki model simboličke regresije može se zapisati slijedećim izrazom [23, 25]:

$$\hat{y} = f(x_1, \dots, x_M) \quad (2)$$

U prethodnom izrazu,  $\hat{y}$  predstavlja neki odziv tj. izlaz kojeg je model predvidio, dok varijable  $x_M$  predstavljaju ulazne veličine odnosno svojstva koja su zabilježena u nekom promatranom sustavu. Osim poznatih ulaznih veličina nekog sustava, potrebno je bilježiti i odgovarajuće izlaze  $y$ , koji će poslužiti kao referenca za pokušaj predviđanja modela. Sam model obično je neka simbolička nelinearna funkcija  $f$  koja je prikazana u obliku stabla. Algoritam GP-a problemu regresije pristupa na način da generira neku inicijalnu populaciju rješenja koju zatim evoluiraju sve dok se u jednoj narednoj generaciji ne pojavi pojedinac,

odnosno model koji najbolje zadovoljava tražene kriterije. Na to koji je model najbolji, ukazuju pokazatelji kao što su: suma kvadriranih pogreški (engl. *sum of squared errors* – *SSE*), srednja kvadrirana pogreška (engl. *mean squared error* - *MSE*) te korijen srednje kvadratne pogreške (engl. *root mean squared error* – *RMSE*). U svrhu poboljšanja rješenja, odnosno efikasnosti dobivanja dobrih modela, GP algoritam kod simboličke regresije uvodi dva koeficijenta: pristranosti (engl. *bias term*)  $b_0$  te težine (engl. *weight*)  $b_1$ . Izraz dobiven regresijskom analizom u GP-u tada se može zapisati u obliku [23, 25]:

$$\hat{y} = b_0 + b_1 \cdot \text{stablo} \quad (3)$$

Koeficijenti  $b_0$  i  $b_1$  zajedno pomažu u modifikaciji predikcije modela  $\hat{y}$  kako bi ona što bolje odgovarala traženom iznosu  $y$ , tj. minimiziraju jednu od spomenutih mjera greški. Navedeni koeficijenti zasebno se dobivaju u svakoj generaciji za svako valjano stablo odnosno potencijalno rješenje, linearnom metodom najmanjih kvadrata. Prethodni izraz može se zapisati u matričnom obliku pogodnom za upotrebu u MATLAB-u kako slijedi [23, 25]:

$$\hat{y} = D \cdot b \quad (4)$$

Ovdje  $D$  označava matricu dimenzija  $N \times 2$  gdje  $N$  predstavlja broj opservacija odzivne varijable  $y$  koju se nastoji predvidjeti. Ova matrica ima dva stupca, prvi stupac sadrži samo jedinice dok drugi stupac sadrži sve izlazne veličine stabla. Oznaka  $b$  predstavlja vektor dimenzije  $2 \times 1$  koji sadrži koeficijente  $b_0$  i  $b_1$  u svojim recima [25].

#### 4.2.2. Simbolička klasifikacija

Simbolička klasifikacija podrazumijeva dobivanje tzv. klasifikacijskih izraza, odnosno modela koji temeljem mjerenih ulaznih parametara uspješno smještaju podatke u neku od unaprijed definiranih kategorija. Kod GPTIPS skupine alata, klasifikacija se provodi po istom principu kao i regresijska analiza. U slučaju klasifikacije, GP pokušava pronaći matematički model koji temeljem dostupnih ulaza pokušava što bolje predvidjeti izlaze, u ovom slučaju neke klase [26].



### 4.2.3. Simbolička optimizacija

S obzirom da GP algoritam samostalno generira kompletnu strukturu traženog modela minimiziranjem pogreške predviđanja, konačno dobiveni matematički izraz ujedno je optimiziran po tom kriteriju pomoću funkcije pogodnosti. Prema tome, koriste li se podaci o npr. terminiranju proizvodnje, kao ulazni trening podaci, pomoću GP-a moguće je evoluirati izraze optimirane prema nekom kriteriju poput: prosječnog vremena protoka, maksimalnog vremena protoka, prosječnog kašnjenja, maksimalnog kašnjenja ili ukupnog težinskog kašnjenja. U nastavku su prikazani spomenuti izraze koji mogu poslužiti za optimizaciju [27].

Prosječno vrijeme protoka:

Prosječno vrijeme protoka:

$$F = \frac{\sum_{j \in N} f_j}{N} \quad (5)$$

Maksimalni vremenski protok:

$$F_{max} = \max_{j \in N} \{f_j\} \quad (6)$$

Prosječno kašnjenje:

$$T = \frac{\sum_{j \in T} (C_j - d_j)}{L} \quad (7)$$

Maksimalno kašnjenje:

$$T_{max} = \max_{j \in L} \{C_j - d_j\} \quad (8)$$

Ukupno težinsko kašnjenje:

$$TWT = \max_{j \in L} \{\omega_j \cdot (C_j - d_j)\} \quad (9)$$

$f_j$  – vrijeme protoka j-tog posla koji se računa pomoću:  $C_j - r_j$

$C_j$  – vrijeme dovršetka j-tog posla

$r_j$  – vrijeme koje je potrebno proteći kako bi se j-ti posao mogao započeti

$d_j$  – predviđeno vrijeme dovršetka j-tog posla

$\omega_j$  – težina j-tog posla

$N$  – broj poslova za koje su bilježeni podaci (broj instanci u setu podataka)

$L$  – broj poslova koji kasne (skup u kojem vrijedi  $C_j - d_j > 0$ )

#### 4.2.4. Multi-genska simbolička regresija

Kod multi-genske simboličke regresije, vrijede svi principi izneseni u poglavlju o simboličkoj regresiji s jednom temeljnom razlikom. Razlika leži u tome da se ovdje pri konstrukciji predviđanja odzivne varijable  $\hat{y}$ , tj. modela, koristi više stabala. Svako stablo, kojih može biti  $G$ , ima svoj težinski koeficijent  $b_G$ . Množenje s pripadajućim koeficijentima, rezultira linearnom kombinacijom stabala, gdje je svako stablo zasebno skalirano s ciljem što većeg smanjenja pogreške konačnog rješenja, tj. razlike predikcije  $\hat{y}$  tog modela i ciljne varijable  $y$ . Struktura modela multi-genske simboličke regresije izgleda ovako [23, 25]:

$$\hat{y} = b_0 + b_1 \cdot \text{stablo}(1) + b_2 \cdot \text{stablo}(2) + \dots + b_G \cdot \text{stablo}(G) \quad (10)$$

Prethodni izraz, moguće je zapisati u matričnom obliku kako slijedi [23, 25]:

$$\hat{y} = X \cdot b \quad (11)$$

Matrica  $X$  je ovdje matrica odzivnih vrijednosti svih stabala od kojih se multi-genski pojedinac sastoji te je njezina dimenzija  $N \times (G + 1)$ , gdje je  $N$  broj opservacija odzivne varijable. U matrici  $X$ , prvi je stupac popunjen jedinicama, dok su ostali popunjeni odzivima pripadajućeg stabla. Sa  $b$  je označen vektor koeficijenata pristranosti (u prvom retku) i težina (u ostalim recima) dimenzije  $(G + 1) \times 1$  [25].

#### 4.2.5. Procjena koeficijenata kod multi-genskog GP-a – Linearna metoda najmanjih kvadrata

Opći linearni regresijski model u matričnom obliku zapisuje se kako slijedi [28]:

$$y = X \cdot \beta + \varepsilon \quad (12)$$

$X$  – matrica odzivnih vrijednosti regresijskog modela (dimenzije  $N \times (G + 1)$ )

$\beta$  – vektor nepoznatih parametara (dimenzije  $(G + 1) \times 1$ )

$\varepsilon$  – vektor neprimijećenih smetnji (dimenzije  $N \times 1$ )

$N$  – broj opservacija odzivne (ciljne) varijable

$G$  – broj članova koji tvore model (članova koji nisu konstante – kod MGGP-a to je jednako broju gena koji se koriste za formiranje multi-genskog pojedinca)

S obzirom da se traži model koji procjenjuje ciljano rješenje, on se zapisuje u obliku koji sadrži procjene nepoznatih parametara  $\beta$  te neke ostatke koji nastaju kao razlika predviđene i ciljane vrijednosti:

$$y = X \cdot b + e \quad (13)$$

$X$  – matrica odzivnih vrijednosti regresijskog modela (dimenzije  $N \times (G + 1)$ )

$b$  – vektor stupac procjene nepoznatih parametara  $\beta$  (dimenzije  $(G + 1) \times 1$ ) – nepoznati parametri kod MGGP-a su koeficijenti pristranosti ( $b_0$ ) i težina ( $b_G$ )

$e$  – vektor ostataka tj. pogreške (dimenzije  $N \times 1$ )

Vektor ostataka iz prethodnog zapisa regresijskog modela može se dobiti preokretanjem tog izraza:

$$e = y - X \cdot b \quad (14)$$

Sada se može tražiti suma kvadriranih pogreški, odnosno ostataka koja je u matricnom obliku jednaka:

$$\sum e_i^2 = e^T \cdot e \quad (15)$$

$e^T$  – transponirana matrica ostataka (pogreški)

Nakon uvrštavanja, dobiva se suma kvadriranih greški kao funkcija procijenjenih nepoznatih parametara  $b$  [28]:

$$S(b) = (y - X \cdot b)^T \cdot (y - X \cdot b) \quad (16)$$

Nakon množenja, dobije se:

$$S(b) = y^T y - y^T X b - b^T X^T y + b^T X^T X b \quad (17)$$

Zatim slijedi daljnje sređivanje izraza. Svi članovi prethodnog izraza daju matricu 1x1 odnosno skalar, ali drugi i treći član se, osim što su zahvaljujući toj dimenziji simetrični, sastoje od istih konstituenata pa su ustvari jednaki. Prema tome, drugi član se može zapisati u istom zapisu kao i treći te se njih dva svode na  $-2b^T X^T y$  [28]:

$$S(b) = y^T y - 2b^T X^T y + b^T X^T X b \quad (18)$$

Sada se traži minimum te funkcije, s obzirom da se greška pokušava minimizirati. Kako bi se to pronašlo, potrebno je funkciju derivirati i izjednačiti s nulom. Da bi se derivacija pojednostavila, uvode se slijedeće matrice:  $X^T y = p$  te  $X^T X = Q$ . Ovdje je matrica  $p$  dimenzije  $(G + 1) \times 1$ , dok matrica  $Q$  ima dimenziju  $(G + 1) \times (G + 1)$ . Izraz sada poprima oblik [28]:

$$S(b) = y^T y - 2b^T p + b^T Q b \quad (19)$$

Prva dva člana se mogu jednostavno derivirati po nepoznatim parametrima matrice  $b$ , međutim posljednji član je nešto kompliciraniji. S obzirom da je  $b$  vektor nepoznatih koeficijenata (dimenzije  $(G + 1) \times 1$ ) umnožak transponirane verzije tog istog vektora sa ne transponiranom, daje skalar koji je suma kvadrata vrijednosti koje se nalaze u tom vektoru. Prema tome taj umnožak poprima oblik  $b^T \cdot b = \sum b^2$ . Derivira li se ta suma, dobije se  $2b_i$ , odnosno rezultat je ekvivalentan množenju matrice (vektor stupca)  $b$  sa 2. Prema tome derivacija posljednjeg člana iz funkcije  $S(b)$  poprima izgled  $2Qb$ , a cijela derivacija, nakon što se natrag uvrste  $Q$  i  $p$ , piše se [28]:

$$\frac{\partial S}{\partial b} = -2X^T y + 2X^T X b \quad (20)$$

Izjednačavanjem s nulom dobiva se tzv. normalna jednadžba pomoću koje se traži minimum:

$$-2X^T y + 2X^T X b = 0 \quad (21)$$

$$2X^T X b = 2X^T y \quad (22)$$

Konačno je moguće izraziti traženi vektor stupac procjene nepoznatih parametara u obliku kojeg i GPTIPS skupina alata koristi kod traženje njihovih iznosa, kako glasi [28, 23, 25]:

$$b = (X^T X)^{-1} X^T y \quad (23)$$

$X$  – matrica odzivnih vrijednosti regresijskog modela (dimenzije  $N \times (G + 1)$ )

$X^T$  – transponirana matrica  $X$  (dimenzije  $(G + 1) \times N$ )

$y$  – ciljan varijabla u trening setu podataka

### 4.3. Multi-gensko genetsko programiranje (MGGP)

Multi-gensko genetsko programiranje (MGGP) ne razlikuje se previše od standardnog GP algoritma. Prvi korak MGGP-a je generiranje inicijalne populacije u kojoj algoritam brine o tome da se pojedinci unutar nje ne ponavljaju kako bi se osigurala što bolja početna raznolikost. Prilikom toga, potrebno je napomenuti da algoritam MGGP-a u sklopu GPTIPS skupine alata za MATLAB, kod svake generacije koja uslijedi nakon inicijalne, ne vodi računa o neponavljanju pojedinaca kako bi se smanjilo računalno opterećenje. Na početku MGGP-a, korisnik definira maksimalni broj gena  $G_{max}$ , od kojih se mogu graditi svaki pojedinci u populaciji. Od koliko gena, tj. različitih stabala, će koji pojedinac biti izgrađen, algoritam odlučuje nasumičnim biranjem između broja 1 i  $G_{max}$ . Od tuda i proizlazi multi-genski naziv odnosno temeljna razlika u odnosu na standardni GP. Kod MGGP-a, pojedinac je dakle izgrađen od više gena koji su svaki reprezentirani vlastitim stablom. Svako stablo izgrađeno je pomoću definiranog seta operatora i operanda te se nalazi unutar maksimalne definirane dopuštene dubine  $d_{max}$ . MGGP-om se dakle provodi multi-gensko simboličko rudarenja podataka te je kod njega svaki pojedinac, kao i konačno rješenje, odnosno traženi model, zapisan u obliku koji je prikazan u tom poglavlju. Korištenje multi-genskih pojedinaca omogućuje dobivanje nelinearnih modela niže kompleksnosti i više točnosti nego što je to slučaj kod standardnog GP-a međutim, potrebno je na umu imati da se ne smije koristiti preveliki broj gena  $G_{max}$ . U slučaju prevelikog broja, osim što je vrijeme provedbe GP algoritma produženo, može doći do pretjerane naučenosti (engl. *overfitting*) algoritma na trening podacima ili do pojave tzv. horizontalne nadutosti (engl. *horizontal bloat*). Horizontalna nadutost je pojam kod GP-a koji obilježava stvaranje stabala sa kompleksnim izrazima koji uopće ne doprinose ili doprinose vrlo malo predviđajućim sposobnostima modela. Osim razlike u vidu upotrebe korištenja više gena od kojih se sastoji neki pojedinac, za potrebe MGGP-a koriste se i modificirane metode križanja koje se ovdje nazivaju križanje više razine (engl. *high level crossover*) te križanje niže razine (engl. *low level crossover*). Kod GPTIPS skupine alata se, ukoliko korisnik to ne definira drugačije, križanje visoke razine koristi u 20% svih slučajeva križanja kod MGGP-a, a kako točno to križanje, kao i ono standardno, funkcioniraju, biti će objašnjeno u narednim poglavljima [25].

## 5. UPOTREBA GPTIPS-A U MATLABU

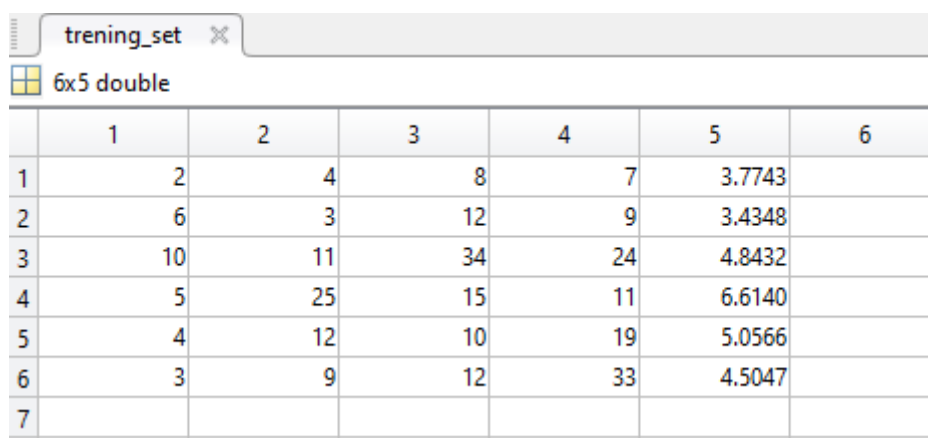
Za početak korištenja GPTIPS skupine alata, potrebno je preuzeti odgovarajuću datoteku koja je besplatno dostupna na internetu [24]. Ona se zatim smješta u datoteku unutar MATLAB direktorija koja se zove toolbox gdje joj se pristupa pri upotrebi. Nakon što je sve spremno, GPTIPS se koristi na željenim podacima na način da se napišu dvije nužne skripte. Prva skripta je tzv. konfiguracijska skripta. Ona učitava željene podatke te definira sve parametre koje korisnik želi primijeniti pri radu GP-a. Druga skripta je skripta za izvođenje samog GP-a. Ona sadrži poveznicu na konfiguracijsku datoteku te služi za ispisivanje koraka koji će GPTIPS provesti kao i za definiranje tih koraka pomoću unaprijed definiranih naredbi. Sve naredbe i ključne riječi potrebne za pisanje ovih dviju skripti ispisane su u prilogu na kraju ovog rada, gdje su podijeljene u kategorije radi lakšeg snalaženja, a mnoge od njih biti će spomenute uz pripadajuću teoriju u nastavku. Prije nego li se počne raditi s bilo kakvim alatima koji koriste GP algoritam, pa tako i s ovim, potrebno je posjedovati odgovore na sljedeća ključna pitanja [22]:

- **Što čini set operanada?**
- **Što čini set operatora?**
- **Što je mjera pogodnosti?**
- **Kojim parametrima će se upravljati radom GP-a?**
- **Što će biti uvjet zaustavljanja GP-a?**
- **Što će biti definirano kao rezultat rada GP-a?**

U nastavku će kroz odgovor na ova pitanja biti demonstrirano pisanje potrebnih skripti kao i znanja koja su nužna pri odabiru pojedinih radnih parametara. Na kraju će se prikazati mogućnosti upotrebe naredba za analizu rezultata GPTIPS-a.

### 5.1. Učitavanje skupova podataka i definiranje seta operanada

Set operanada može se sastojati od varijabli, funkcija i konstanti. Varijablama se označavaju neki ulazi ili izlazi sustava koji se razmatraju za modeliranje. Ulazne varijable definiraju se na način da se učitaju željeni stupci iz seta podataka pohranjenog u *.mat* formatu. Za primjer učitavanja uzima se jednostavni set podataka nazvan *primjer\_podataka*, koji sadrži odvojene trening i test podatke (nazvane *trening\_set* i *test\_set*). Na slici dolje prikazani su trening podaci nazvani *trening\_set* koji se nalaze u primjernom setu te je na njima moguće učitati da sadrže pet stupaca, međutim samo prva četiri su ulazni podaci, dok su u posljednjem stupcu izlazni.



	1	2	3	4	5	6
1	2	4	8	7	3.7743	
2	6	3	12	9	3.4348	
3	10	11	34	24	4.8432	
4	5	25	15	11	6.6140	
5	4	12	10	19	5.0566	
6	3	9	12	33	4.5047	
7						

Slika 15. Skup podataka nazvanih *trening\_set*

Prije nego li se krene s učitavanje ulaznih i izlaznih podataka, potrebno je učitati cijeli set, u ovom slučaju to se postiže linijom koda [23]:

```
load primjer_podataka
```

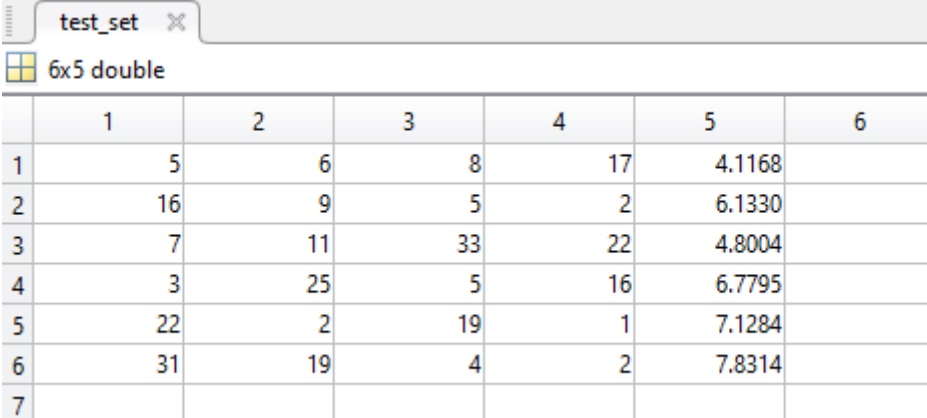
Trening podaci iz prethodno pokazanih setova mogu se učitati upotrebom sljedećih naredbi u MATLAB-u [23]:

```
gp.userdata.xtrain = trening_set(1:6, 1:4)
gp.userdata.ytrain = trening_set(1:6, 5)
```

Ekvivalentno tome, testni podaci učitavaju se naredbama [23]:

```
gp.userdata.xtest = testni_set(1:6, 1:4)
gp.userdata.ytest = testni_set(1:6, 5)
```

Spomenuti testni set podataka prikazan je na sljedećoj slici.



	1	2	3	4	5	6
1	5	6	8	17	4.1168	
2	16	9	5	2	6.1330	
3	7	11	33	22	4.8004	
4	3	25	5	16	6.7795	
5	22	2	19	1	7.1284	
6	31	19	4	2	7.8314	
7						

Slika 16. Skup podataka nazvanih `testni_set`

Ovim učitavanjem definiran je točan broj ulazni i izlaznih varijabli, pa nema potrebe za nekim dodatnim uputama GPTIPS-u kako bi znao s čime se treba koristiti. Osim trening i test seta podataka, moguće je koristiti i validacijski set. Taj skup podataka sastoji se od nekih nasumično izuzetih podataka iz osnovnog seta koji se na taj način uskraćuju za korištenje trening i testnom skupu. Tako izuzeti podaci ne koriste se za treniranje modela već za ocjenjivanje najboljih dobivenih pojedinaca koje GPTIPS dobiva u svakoj generaciji. Na kraju rada algoritma, kada se pronađe najbolje konačno rješenje, GPTIPS označava i onog pojedinca koji je imao najbolje performanse na validacijskom skupu. Taj pojedinac često nije isti kao i onaj ukupno najbolji dobiven na trening skupu te od njega obično bolje predviđa još neviđene podatke. Osim toga, takvo rješenje ima veću vjerojatnost za viši iznos pogodnosti prilikom provjere na testnom skupu. Upravo zbog navedenog, validacijski skup podataka najčešće se upotrebljava kako bi se riješio problem tzv. prekomjerne naučenosti modela (engl. *overfitting*) na način da se upotrijebi onaj model, koji bolje generalizira na neviđenim podacima. Pretpostavi li se da je taj skup nazvan *validacijski\_set* te da je jednakih dimenzija kao i primjerni trening i testni set, učitati ga je moguće pomoću sljedeće naredbe [23]:

```
gp.userdata.xval = validacijski_set(1:6, 1:4)
gp.userdata.yval = validacijski_set(1:6, 5)
```

Kao što je bilo spomenuto, set operanda može sadržavati i funkciju međutim, GPTIPS skupina alata tu ne omogućuje mnogo opcija. Moguće je generirati ulazne podatke pomoću neke funkcije no nije moguće upotrebljavati same funkcije kao operande, pa to valja imati na umu. Naposljetku se dolazi do konstanti, koje također spadaju u set operanda, a koje su važne



pri simboličkoj regresiji. Kod GP-a koristi se tzv. privremena nasumična konstanta (engl. *ephemeral random constant*) kao operand. Ona se nasumično generira iz zadanog dopuštenog raspona kao dio postupka inicijalizacije populacije. Za svako stablo, odnosno svakog pojedinca neke generacije, jednom kad se generira ova konstanta ona postaje nepromjenjiva i vezana isključivo uz njega. Te konstante mijenjaju se daljnjim postupkom evolucije rješenja kod GP-a, gdje može doći do mutacije, čime list koji sadrži konstantu mijenja svoj iznos te ona postaje vezana za nekog novog pojedinca u idućoj generaciji [22]. Raspon iz kojeg će se konstante generirati po normalnoj raspodjeli pri izvođenju GP algoritma kao i vjerojatnost upotrebe lista koji je konstanta, a ne neki drugi operand, u procesu inicijalizacije, korisnik u GPTIPS-u definira sljedećim naredbama [23]:

```
gp.nodes.const.range = [0,50]
gp.nodes.const.p_ERC = 0.25
```

Važno je spomenuti da ukoliko bilo koje od ovdje navedenih naredbi korisnik ne definira u konfiguracijskoj skripti, GPTIPS uzima neke zadane vrijednosti, gdje god one postoje, a koje su to zadane vrijednosti moguće je vidjeti u prilogu na kraju ovog rada za svaku pojedinu naredbu. Također, generirane konstante nisu u formatu cijelog broja, pa ako korisnik želi da se osim decimalnih brojeva, generiraju konstante takvog formata, potrebno je definirati vjerojatnost upotrebe cijelog broja. Definirani iznos je ustvari vjerojatnost da za neki list za koji je odabran operand koji je konstanta, bude upotrijebljena konstanta koja je cijeli broj. Navedeno se definira na sljedeći način [23]:

```
gp.nodes.const.p_int = 0.1
```

## 5.2. Definiranje seta operatora

Kao što je ranije bilo spomenuto, operatori mogu biti: aritmetički, logički, transcendentni, uvjetni, operatori petlje te subrutinski operatori. Set operatora obično je vođen prirodom domene u kojoj se nalazi rješavani problem. Kod simboličke regresije u GPTIPS skupini alata koriste se isključivo aritmetički i transcendentni (trigonometrijski, logaritamski, eksponencijalni, hiperbolni) operatori. Njih se definira pozivanjem njihova naziva pod kojim se vode u MATLAB-u, a neki od tih naziva koji se mogu upotrijebiti izneseni su u nastavku: plus, minus, times, rdivide, ldivide, power, square, sqrt, cube, exp, negexp, sin, cos, tan, cot, sinh, cosh, tanh, coth, neg, abs, log. Primjer definiranja seta operatora koji se sastoji od +, -, /,\* je sljedeći [23]:

```
gp.nodes.functions.name = {'plus', 'minus', 'rdivide', 'times'}
```

Osim njih, moguće je da korisnik definiira neki vlastiti operator na način da napiše potrebne stvari u posebnu *.m* datoteku naslovljenu po želji. Ta datoteka mora biti pohranjena unutar GPTIPS datoteke. Primjer tako dodanog operatora je *add3* operator. Kada se otvori njegova *.m* datoteka ona izgleda ovako [23]:

```
function y = add3(x1, x2, x3)
    y=x1 + x2 + x3;
```

Ovaj operator prihvaća tri ulazne varijable te ih zbraja. Nadjenuto mu je ime *add3* pa se to ime upotrebljava unutar `gp.nodes.functions.name` naredbe kako bi se upotrijebio.

### 5.3. Temeljni preduvjeti za definiranje primitivnog seta

Primitivni set sačinjen je od seta operatora i operanda, čije je definiranje u GPTIPS-u opisano u prethodna dva pod-poglavlja. Prilikom tog definiranja važno je da su ispunjeni preduvjeti [20,22]:

- **potpunosti** (enlg. *closure*)
- **dostatnosti** (engl. *sufficiency*)

#### 5.3.1. Potpunost

Svojstvo potpunosti podrazumijeva da svaki pripadnik operatorskog seta mora biti u mogućnosti prihvatiti bilo koju vrijednost ili vrstu podataka, koji mogu nastati kao izlaz bilo kojeg operatora i operanda [20]. Drugim riječima, mora biti osigurana mogućnost evaluacije rješenja te konzistentnost vrste podataka unutar primitivnog seta, kako bi se osigurala potpunost u funkcionalnosti pri radu s GP algoritmom [22]. Evaluacija rješenja nije moguća ukoliko se na primjer, pojavljuje stablo u kojem se dijeli s nulom. Neki drugi primjeri su logaritmiranje s nulom ili korištenje negativnog broja pod korijenom. Taj problem rješava se upotrebom tzv. zaštićenih funkcija u MATLAB-u, koje najprije provjeravaju postoje li neki potencijalni problemi s ulazima prije nego što se izvršavaju. Na primjeru dijeljenja s nulom, zaštićeno dijeljenje provjerava jeli mu drugi argument tj. onaj s kojim se dijeli, jednak nuli. Ako je to slučaj, prekida operaciju ili, što je najčešće, koristi jedinicu kao zadani rezultat

takve operacije. Što se tiče konzistentnosti vrste podataka, ona je nužna kako bi se pod-stabla prilikom križanja mogla izmjenjivati bez posljedica. U tom slučaju, važno je da svi operandi (ulazne varijable i konstante) budu iste vrste te da operatori prihvaćaju jednak broj ulaznih argumenata (engl. *arity*). Na primjer, u setu operatora koji se sastoji od  $+$ ,  $-$ ,  $*$ ,  $/$ ; svaki pojedini operator prima dva ulazna argumenta te za rezultat može biti definirano da mora biti u obliku cijelog broja. Prema tome se set operanda u tom slučaju, također mora sastojati od cijelih brojeva te se ne smije upotrijebiti više od dva operanda kod svakog operatora čime je postignuto ispunjenje uvjeta potpunosti [20, 22].

### 5.3.2. Dostatnost

Dostatnost je još jedno svojstvo koje primitivni set mora zadovoljiti kako bi GP algoritam davao ispravna, zadovoljavajuća rješenja. Ovaj preduvjet smatra se ispunjenim kada je primitivni set (dakle set operatora i operanda) dostatan za izražavanje rješenja promatranog problema. To znači da se set operatora, ukoliko se traži nelinearno rješenje, ne može sastojati samo od operacija zbrajanja i oduzimanja, već mora sadržavati barem kvadrat ili eksponent [20]. Osim toga, morala bi postojati neka korelacija između članova seta operanda (ulaznih i izlaznih, ciljnih varijabli). Kada primitivni set nije dostatan, GP može razviti programe koji aproksimiraju rješenje sa nezadovoljavajućom razinom pogreške tj. s onom razinom pogreške koja nije dovoljna za neku praktičnu upotrebu. Nažalost, ispunjenje tog svojstva sa sigurnošću je moguće osigurati jedino kod problema kod kojih korisniku teorija, iskustvo ili neka druga metoda, ukazuju na potrebnu kombinaciju elemenata primitivnog seta, što je vrlo rijedak slučaj u praksi [22]. Prema tome, za probleme iz mnogih domena, ne postoji način kako to svojstvo osigurati, pa postupak definiranja dostatnih setova ovisi isključivo o posjedovanju prethodnog znanja i iskustva korisnika GP-a [20]. Pod pretpostavkom da neki realni mjereni procesi i sustavi implicitno zadovoljavaju uvjet korelacije u setu operanda, za set operatora nije odveć upotrebljavanje i članova za koje korisnik možda smatra da nisu potrebni. Upravo takvi članovi možda mogu osigurati dostatnost te značajno utjecati na kvalitetu rješenja, a da pritom na današnjim modernim računalima, ne utječu previše na performanse rada [22].

#### 5.4. Rad s multi-genskim pojedincima u GPTIPS-u

S novom verzijom 2.0 GPTIPS skupine alata, koja se koristi u ovom radu, način rada s multi-genskim pojedincima je unaprijed omogućen, no ukoliko ga korisnik želi isključiti, dovoljno je da u konfiguracijskoj datoteci upiše sljedeću liniju [23]:

```
gp.genes.multigene = 'False'
```

U slučaju da želi raditi u modu multi-genskog genetskog programiranja, nije potrebno ništa posebno dodati osim željenog broja gena (stabla) od kojih se multi-genski pojedinac smije maksimalno sastojati. U GPTIPS-u je, ukoliko korisnik ne definira, zadano da se upotrebljavaju četiri gena, međutim taj broj korisnik može proizvoljno zadati u konfiguracijskoj datoteci [23]:

```
gp.genes.max_genes = 3
```

U praksi nije lako dati neke savjete za definiranje maksimalnog broja gena za MGGP model, no poznato je da korištenje prevelikog broja izaziva prekomjernu pojavu stabla koja malo doprinose konačnom rješenju, a pored toga dolazi i do značajnog povećanja vremena potrebnog da se GP algoritam izvrši [25].

#### 5.5. Definiranje funkcije pogodnosti

Nakon prva dva koraka u kojima se definira prostor rješenja kojeg će GP pretraživati, u ovom je potrebno definirati mjeru pogodnosti koja će uputiti algoritam na onu regiju prostora, u kojoj se nalaze dobra rješenja, pa možda i ono konačno, najbolje rješenje. Mjere pogodnosti koje služe u tu svrhu mogu biti: iznos pogreške između dobivenog i traženog izlaza, iznos ciljnog atributa (vremena, goriva, novca i sl.) koji je potreban kako bi se sustav doveo u ciljano stanje, točnost u prepoznavanju uzoraka ili klasificiranju objekata te usklađenost strukture sa nekim korisničkim kriterijem (npr. kriterijem jednostavnosti) [22]. Kako bi se provela procjena pogodnosti, potrebno je sva potencijalna rješenja dobivena u pojedinoj fazi rada GP-a, provjeriti nizom tzv. slučajeva pogodnosti (engl. *fitness cases*). Ti slučajevi pogodnosti ustvari predstavljaju niz kombinacija ulaznih i izlaznih podataka koji su korisniku poznati, odnosno radi se o trening setu podataka kod simboličkog rudarenja. Standardne mjere pogodnosti koje se koriste za procjenu dobrote rješenja kod simboličke regresije su: sirova pogodnost (engl. *raw fitness*) i standardizirana pogodnost (engl. *standardized fitness*). Sirova

pogodnost izračunava se kao suma apsolutne udaljenosti između izlaza trenutnog rješenja (matematičkog modela) i tražene vrijednosti koja se očekuje i to na svim slučajevima pogodnosti, odnosno na cijelom trening setu. To se može zapisati sljedećim izrazom gdje  $f_R$  označava sirovu pogodnost [20]:

$$f_R(i) = \sum_{j=1}^N |S(i,j) - C(j)|^k \quad (24)$$

$S(i,j)$  – vrijednost pogodnosti dobivena evaluacijom pojedinca  $i$  na slučaju pogodnosti  $j$

$C(j)$  – ciljna izlazna vrijednost za slučaj pogodnosti  $j$

$N$  – broj slučajeva pogodnosti

$k$  – neki prirodni broj (obično 1 ili 2)

U slučaju simboličke regresije, teži se sirovoj pogodnosti što bližoj nuli jer onda model najbolje opisuje dobivene podatke međutim, ne teži se uvijek minimiziranju pogodnosti. U slučajevima u kojima korisnik ručno zadaje mjeru odnosno funkciju pogodnosti, ona može biti definirana na način da veći iznosi znače bolje rješenje, pa je u tom slučaju tu funkciju potrebno maksimizirati. U GPTIPS-u, ako korisnik ne definira drugačije, funkcija pogodnosti se minimizira međutim, ukoliko se želi maksimizirati, potrebno je u konfiguracijsku datoteku upisati sljedeće [23]:

```
gp.fitness.minimisation = 'False'
```

Neku proizvoljnu mjeru odnosno funkciju pogodnosti, korisnik u GPTIPS-u definira na način da napiše *.m* datoteku u kojoj je sadržana željena mjera, te u konfiguracijskoj datoteci za izvršavanje GP algoritma upiše [23]:

```
gp.fitness.fitfun = @ime_funkcije
```

S druge strane, standardizirana pogodnost predstavlja mjeru koja se koristi u slučaju maksimiziranja funkcije cilja kada je bolja ona pogodnost koja je veća. U tom slučaju standardizirana pogodnost preformulira visoke iznose sirove pogodnost u standardiziranu pogodnost koja je bliska nuli. Prema tome, ponovno se dobiva isti uvjet kao i kod minimiziranja funkcije kod sirove pogodnosti. Standardizirana pogodnost računa se prema izrazu [20]:

$$f_S = f_R^{max} - f_R \quad (25)$$

$f_R$  – sirova pogodnost

$f_R^{max}$  – maksimalni mogući iznos sirove pogodnosti (pretpostavlja se da je taj iznos poznat)

U GPTIPS-u kod multi-genske simboličke regresije koristi se funkcija pogodnosti pohranjena pod .m datotekom naslovljenom sa *regressmulti\_fitfun*. Glavni kriterij mjerenja pogodnosti definiran unutar te datoteke je tzv. korijen srednje kvadratne pogreške (engl. *Root Mean Squared Error* – *RMSE*). Ta mjera računa se pomoću sljedećeg izraza [25]:

$$RMSE = \sqrt{\frac{err^T * err}{N}} \quad (26)$$

*err* – matrica (dimenzije  $N \times 1$ ) iznosa pogreški predviđanja koje se u GPTIPS-u dobiju iz razlike: *gp.userdata.ytrain* – *ypredtrain*

*err<sup>T</sup>* – transponirana matrica *err*

*ypredtrain* – varijabla u obliku matrice unutar GPTIPS-a koja obilježava iznos predviđanja trenutno evaluiranog rješenja za sve slučajeve pogodnosti

*N* – broj opservacija odzivne varijable tj. broj unosa (redaka) u setu podataka

Spomenuta funkcija pogodnosti se u slučaju multi-genskog genetskog programiranja poziva u konfiguracijskoj datoteci pomoću [23]:

```
gp.fitness.fitfun = @regressmulti_fitfun
```

Ukoliko se pored trening i testnog skupa podataka upotrebljava i validacijski skup, uz prethodnu liniju, potrebno je dodati i [23]:

```
gp.userdata.user_fcn = @regressmulti_fitfun_validate
```

Ostale mjere pogodnosti koje GPTIPS prikazuje su: suma kvadriranih pogreški (engl. *summ of squared errors* – *SSE*) te srednja kvadrirana pogreška (engl. *mean squared error* – *MSE*).

One se računaju pomoću izraza koji su očiti kod *RMSE*-a:

$$SSE = err^T * err \quad (27)$$

$$MSE = \frac{err^T * err}{N} \quad (28)$$

*err<sup>T</sup> \* err* – rezultat umnoška ovih matrica pogreški nije ništa drugo nego suma svih kvadriranih pogreški za sve varijable – rezultat ustvari nije matrica nego skalar koji predstavlja konačnu vrijednost *SSE*-a

Ukoliko se SSE ili MSE žele koristiti kao glavne mjere dobrote rješenja umjesto zadane mjere RMSE, to je moguće postići jednostavnom izmjenom odgovarajućeg retka unutar *.m* datoteke *regressmulti\_fitfun* gdje se trenutna mjera zamijeni željenom formulom [25].

## 5.6. Upravljački parametri za izvršavanje GP algoritma u GPTIPS-u

Osim što korisnik treba definirati set operatora i operanda te funkciju pogodnosti, potrebno je GPTIPS-u definirati i neke druge parametre važne za funkcioniranje GP algoritma. U prvom redu, potrebno je definirati veličinu populacije a zatim odabrati broj generacija koje će se vrtjeti GP algoritmom, maksimalnu dopuštenu dubinu stabla, željenu metodu stvaranja inicijalne populacije (nulte generacije), metodu selekcije u narednim populacijama (generacijama), vrstu i učestalost križanja, vrstu i učestalost mutacije, upotrebu elitizma kao i niz drugih korisničkih parametara koji se tiču prikazivanja rezultata, a o kojima će više riječi biti u drugom poglavlju [20]. GPTIPS skupina alata za većinu tih parametara posjeduje neke zadane vrijednosti, pa ga je korisnicima koji nemaju iskustva u radu sa evolucijskim algoritmima, jednostavno koristiti [23, 25]. Bez obzira na to definira li korisnik detaljno svaki od parametara koji su mu dostupni, GP algoritma je općenito dovoljno robustan da daje zadovoljavajuća rješenja za čitavi niz kombinacija tih parametara. Zahvaljujući tome, nije potrebno trošiti mnogo vremena na podešavanje parametara, međutim oni napredniji korisnici mogu upotrijebiti neko svoje stečeno znanje kako bi se algoritam na najbolji mogući način dotjerao, čime je moguće postići iznad prosječno kvalitetne rezultate. Nemoguće je iznijeti neke osnovne savjete oko optimalnih parametara zbog toga što oni ovise o okolnostima primjene GP algoritma, no postoje određene smjernice koje mogu usmjeriti korisnika u dobrom smjeru, a koje će biti spomenute u nastavku [22].

### 5.6.1. Veličina populacije

Veličina populacije trebala bi biti što je moguće veća, a zbog činjenice da su današnja moderna računala prilično moćna, jedino ograničenje u tom pogledu je vrijeme koje je potrebno uložiti da bi se evaluirala pogodnost, za koje se teži da ostane u razumnim razmjerima. Vrijeme rada GP algoritma može se u grubo procijeniti slijedećim umnoškom [22]:

$$t = R \cdot G \cdot P \cdot S \cdot F [s] \quad (29)$$

$R$  – broj vrtnji GP algoritma (engl. *number of GP runs*)

$G$  – broj generacija (engl. *generation number*)

$P$  – veličina populacije (engl. *population size*)

$S$  – prosječna veličina modela (engl. *mean model size*)

$F$  – broj slučajeva pogodnosti odnosno mjerenih instanci u setu podataka (engl. *number of fitness cases*) =  $N$

U GPTIPS-u je, ako korisnik to ne promijeni, zadana veličina populacije od 100 pojedinaca dok literatura nalaže korištenje veličine populacije od barem 500 pojedinaca ili više [22]. Razlog težnje za što većom populacijom leži u tome što ona omogućuje veći prostor pretraživanja kod svake generacije čime se povećava vjerojatnost da će postupak evolucije kod GP-a doći do zadovoljavajućeg rješenja. Prema tome, neko općenito pravilo je, što je problem kompleksniji, to je bolje koristiti se velikom populacijom kako bi se uspješno došlo do željenog rješenja [21]. Veličina populacije u konfiguracijskoj datoteci kod GPTIPS-a definira se linijom [23]:

```
gp.runcontrol.pop_size = 500
```

### 5.6.2. *Maksimalni broj generacija*

Maksimalnim brojem generacija zadaje se neki vremenski okvir unutar kojeg GP algoritam mora izbaciti rezultat te je prema tome to ujedno i jedan od mogućih uvjeta zaustavljanja algoritma. Što je maksimalni broj generacija veći, to je više vremena na raspolaganju kako bi algoritam pronašao rješenje, međutim zadavanje prevelikog broja nema nikakvog korisnog učinka pa samim time niti smisla [21]. U GPTIPS-u je zadani broj od 150 generacija, no literatura preporuča upotrebu između 10 i 50 generacija. Razlog zbog kojeg se preporuča takav raspon leži u činjenici da se najproduktivnije pretraživanje prostora rješenja od strane GP algoritma odvija upravo do 50-te generacije, nakon koje obično slijedi stagnacija. Bez obzira na to, kao neka narodna mudrost ostaje generalni savjet da se koristi više generacija od toga. Osim toga, također je moguće upotrebljavati manje generacije ali da se GP algoritam iznova koristi više puta prilikom čega se povećava vjerojatnost pronalaska dobrih rješenja, jer se pri svakoj vrtnji algoritma stvara nova inicijalna populacija koja ulazi u evoluciju [3]. Funkcionalnost višestrukih vrtnji GP algoritma ispočetka je integrirana u GPTIPS skupinu alata te je broj tih vrtnji moguće definirati u konfiguracijskoj datoteci slijedećom linijom [23]:



```
gp.runcontrol.runs = 4
```

Broj generacija definira se linijom [23]:

```
gp.runcontrol.num_gen = 150
```

### 5.6.3. *Maksimalna dopuštena dubina stabla*

Dubina stabla se, kao što je to već u jednom prethodnom poglavlju bilo napomenuto, definira kao minimalni broj rubova kroz koje je potrebno proći da bi se od najdonjeg došlo do početnog, korijenskog čvora za kojeg se uzima da je na dubini nula [22, 23]. Parametrom maksimalne dubine, označenog sa  $d_{max}$ , ujedno se zadaju i maksimalne dopuštene veličine stabla. Veličinu stabla, koja se definira kao ukupni zbroj svih čvorova i listova u njemu, može se dobiti pomoću  $2^{d_{max}}$ , pod uvjetom da se pri izgradnji stabla koriste isključivo operatori koji za ulaz prihvaćaju dva ulazna argumenta (engl. *arity*), što je slučaj kod operatora u simboličkoj regresiji [19]. U konfiguracijskoj datoteci kod primjene GPTIPS-a, ovaj broj definira se [23]:

```
gp.treedef.max_depth = 4
```

Literatura upućuje korisnike da upotrebljavaju *Ramped Half-and-Half* metodu za inicijalizaciju (o njoj će više biti rečeno u narednom poglavlju) sa maksimalnim dopuštenim dubinama stabla od 2 do 6 [22]. U slučaju da se umjesto maksimalne dubine stabla želi definirati maksimalna veličina (suma čvorova i listova), onda se umjesto prethodne linije koristi slijedeća [23]:

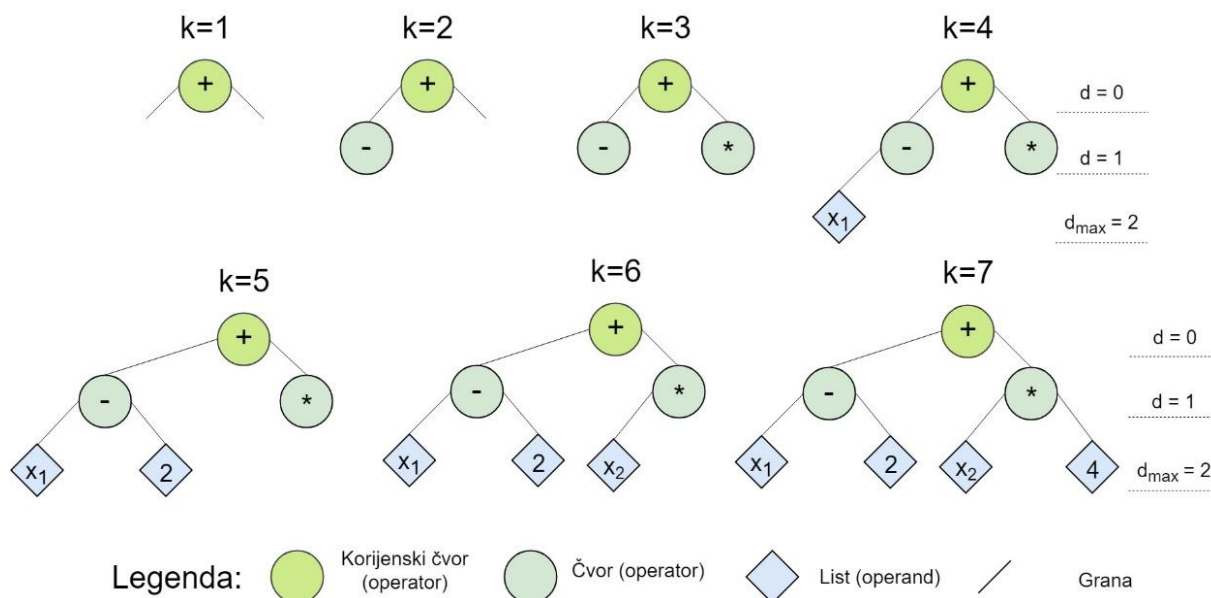
```
gp.treedef.max_nodes = 10
```

### 5.6.4. *Metode stvaranja inicijalne populacije*

Nakon što se definira veličina populacije, broj generacija koje će se evoluirati GP algoritmom te maksimalna dopuštena dubina stabla, slijedi odabir metode stvaranja početne populacije, od koje kreće evolucija. Postoji nekoliko različitih pristupa tom problemu, od kojih će biti opisana tri koja su ujedno i dostupna unutar GPTIPS-a, a to su: *Full*, *Grow* i *Ramped Half-and-Half* metoda [19, 22].

## 5.6.4.1. Full metoda

Pojedinci u inicijalnoj populaciji kod ove metode, nastaju tako da se najprije nasumično uzimaju pripadnici iz operatorskog seta (+, -, \*, / itd.) sve dok se ne dođe do čvora koji se nalazi na maksimalnoj mogućoj dubini koja u obzir uzima definiranu maksimalnu dopuštenu dubinu stabla  $d_{max}$  od strane korisnika. Ukoliko korisnik umjesto maksimalne dopuštene dubine stabla  $d_{max}$  koristi maksimalnu dopuštenu veličinu stabla (jednaka sumi broja čvorova i listova)  $s_{max}$ , onda se članovi operatorskog seta prestaju dodavati stablu nakon što se ispuni to ograničenje. Ispunjenjem zadanog ograničenja, preostale grane popunjavaju se listovima tj. nasumično odabranim pripadnicima seta operanda [19, 22]. Na taj način se dobivaju puna stabla uniformnog izgleda (od tud naziv – engl. *full* – pun) čiji su svi listovi na jednakim maksimalnim dubinama. Uz navedeno, ukupni broj čvorova i listova jednak je za sva stabla u slučaju korištenja operatorskog seta čiji svi članovi primaju samo dva ulazna argumenta kao što je to slučaj kod simboličke regresije, pa od tuda i dolazi taj ujednačen izgled čitave populacije. U slučaju da navedeno ne bi vrijedilo, dobila bi se malo nepravilnija stabla, ali bi ona i dalje bila ograničena rasponom i veličinom [22]. Na slijedećoj je slici u koracima prikazan nastanak jednog stabla pomoću *Full* metode s ograničenjem dubine od 2.



Slika 17. Princip izgradnje stabla kod Full metode inicijalizacije

U prvom koraku ( $k = 1$ ), prikazanom na prethodnoj slici, iz seta operatora ova metoda nasumično bira operaciju zbrajanja (+) koja predstavlja korijenski čvor nulte dubine. U drugom koraku ( $k = 2$ ) odabrana je operacija oduzimanja (-) te se taj čvor nalazi na dubini 1.

S obzirom da je maksimalna dopuštena dubina stabla  $d_{max} = 2$ , to znači da se na toj dubini kasnije ne smije birati čvor iz seta operatora, nego to trebao biti argument iz seta operanda tj. list. U koraku  $k = 3$  stablo se nastavlja popunjavati sa operatorima, pa se tako na desnu granu korijenskog čvora dodaje operacija množenja (\*) nasumičnim izvlačenjem iz seta operatora. U preostalim koracima ( $k = 4,5,6,7$ ) preostaje samo dodati listove iz seta operanda (dodani su  $x_1, 2, x_2$  te 4). čime je izgradnja stabla gotova [22, 29]. Jednostavna formulacija pravila izgradnje stabla *Full* metodom glasi [29, 30]:

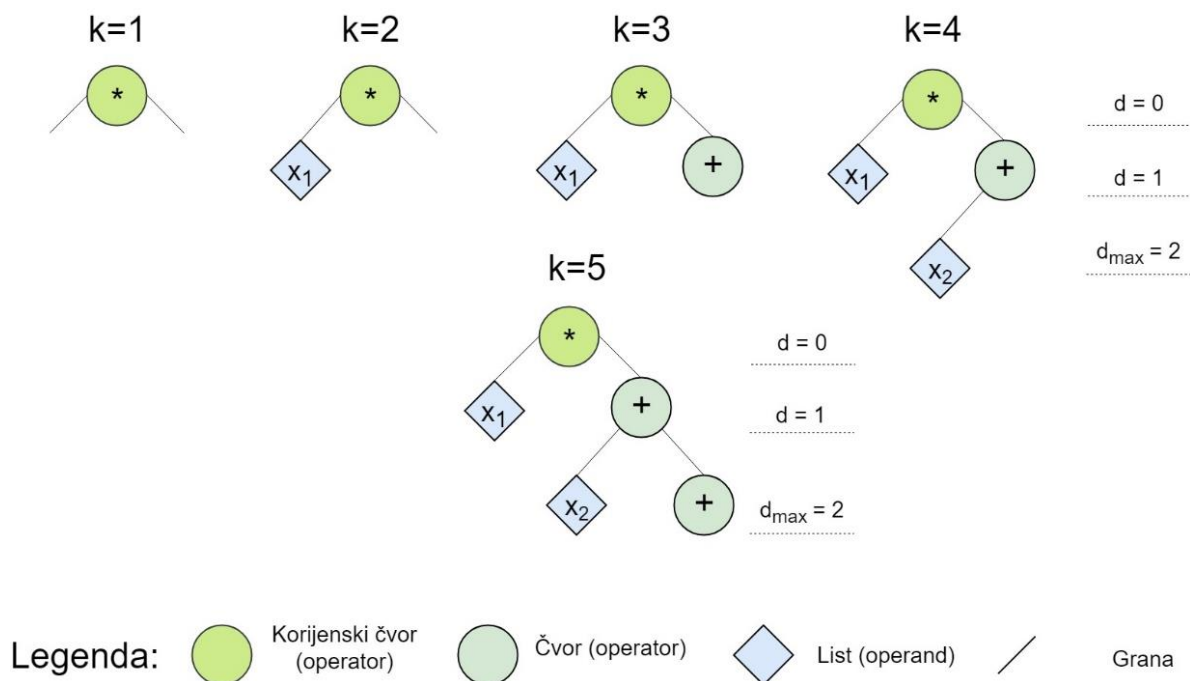
- dodavati čvorove nasumično odabrane iz seta operatora do dubine  $d < d_{max} - 1$ , sve dok je to moguće
- dodati listove na preostala mjesta koja su na dubini  $d = d_{max}$ , čime završava postupka izgradnje stabla

U konfiguracijskoj datoteci kod GPTIPS-a, korištenje ove metode inicijalizacije određuje se pisanjem broja 1 u liniji kako slijedi [23]:

```
gp.treedef.build_method = 1
```

#### 5.6.4.2. *Grow metoda*

Pojedinci inicijalne populacije, kod ove metode nastaju nasumičnim odabirom čvorova (operatora) ili listova (operanda) iz cijelog dostupnog primitivnog seta (seta operatora i operanda), sve dok se ne dosegne maksimalna dopuštena dubina stabla  $d_{max}$  ili, ako je tako zadano, maksimalni dopušteni broj čvorova i listova  $s_{max}$ . Iznimka je jedino korijenski čvor, koji se bira prvi i to nasumičnim izvlačenjem isključivo iz operatorskog seta, jer on ne može biti operand (varijabla ili konstanta). Jednom kada se u jednoj grani odabere pripadnik seta operanda, njezina izgradnja je završila bez obzira na to jeli postignuta maksimalna dopuštena dubina. Ova metoda stvara stabla koja su nepravilnijeg izgleda za razliku od *Full* metode [19]. Veličine i oblici stabla generiranih *Grow* metodom, jako su osjetljivi na veličine seta operatora i operanda. Ako je na primjer set operanda (varijable i konstante) značajno veći od seta operatora (matematičke operacije), ova metode će gotovo sigurno generirati jako kratka stabla bez obzira na definirani  $d_{max}$ . U obrnutom slučaju, metoda će se ponašati slično kao *Full* te će generirati slična stabla. Osim toga, na oblik stabla, baš kao i kod *Full* metode, utječe broj ulaznih argumenata (engl. *arity*) kojeg pojedini pripadnici seta operatora prihvaćaju [22]. U nastavku je po koracima dan primjer generiranja stabla maksimalne dubine 2, pomoću ove metode.



Slika 18. Princip izgradnje stabla kod Grow metode inicijalizacije

U prvom koraku ( $k = 1$ ) moguće je za korijenski čvor birati samo iz seta operatora, pa je odabrana operacija množenja (\*). Za sve ostale pozicije stabla, nasumično se odabire pripadnik iz cjelokupnog primitivnog seta. U drugom koraku ( $k = 2$ ) u jednoj grani odabran je operand ( $x_1$ ) čime ona više ne može dalje rasti. U preostalim koracima stablo se popunjava sve do maksimalne dubine na kojoj se biraju listovi tj. operandi, kao i kod *Full* metode [22]. Jednostavna formulacija pravila izgradnje stabla *Grow* metodom glasi [29, 30]:

- za prvi (korijenski) čvor, odabrati nasumično nekog pripadnika operatorskog seta
- dodavati proizvoljno pripadnike seta operatora ili operanda sve dok je dubina  $d < d_{max} - 1$ , prekinuti širenje grane ukoliko je odabran operand bez obzira na dubinu
- za pozicije na dubini  $d = d_{max}$ , nasumično birati samo iz seta operanda (varijable i konstante)

U konfiguracijskoj datoteci kod GPTIPS-a, korištenje ove metode inicijalizacije određuje se pisanjem broja 2 u liniji kako slijedi [23]:

```
gp.treedef.build_method = 2
```

#### 5.6.4.3. *Ramped Half-and-Half metoda*

John R. Koza je 1992. godine predložio novu metodu inicijalizacije kod koje se pola populacije generira *Full*, a druga polovica *Grow* metodom. Razlog zbog kojeg je metoda nastala, leži u činjenici da niti *Full* niti *Grow* metoda, svaka zasebno, nisu pružale pretjerano široki raspon veličina ili oblika pri formiranju pojedinaca. Uvođenjem ove metode, postiže se poboljšanje raznolikosti struktura početne populacije što rezultira boljom inicijalizacijom GP algoritma jer je prostor pretraživanja od kojeg polazi raznolikiji [22]. Metoda funkcionira na način da korisnik, kao i kod prethodnih, definira maksimalnu dopuštenu dubinu stabla  $d_{max}$ , ali uz temeljnu razliku u vidu toga da se te dubine dijele na kategorije s inkrementom jedan (ako je npr.  $d_{max} = 6$ , onda se dijeli na kategorije 2, 3, 4, 5 i 6), gdje se za svaku dubinu, polovica stabala generira *Full*, a polovica *Grow* metodom [19]. U GPTIPS skupini alata, ova metoda inicijalizacije unaprijed je zadana, no ako korisnik nije siguran ili želi to posebno naglasiti, to se može postići pisanjem broja 3 u liniji konfiguracijske datoteke kako slijedi [23]:

```
gp.treedef.build_method = 3
```

#### 5.6.5. *Metode selekcije pojedinaca*

U toku rada GP algoritma, najvažniji korak je postupak selekcije pojedinaca iz neke populacije kako bi se nad njima mogle izvršiti genetske operacije. Te operacije se vrše u svrhu dobivanja nove generacije koja je po mogućnosti bolja od one prethodne [22]. Selekcija se u svojoj osnovi temelji na odabiru pojedinaca za razmnožavanje, po uzoru na prirodnu selekciju. Kod prirodne selekcije preživljavaju samo najbolji, u ovom su slučaju to kod GP-a, rješenja koja su najbolja nakon procjene funkcijom pogodnosti. O načinu dodjele iznosa pogodnosti pojedincima pomoću funkcije pogodnosti, više je bilo rečeno u jednom od prethodnih poglavlja, stoga se ovdje više neće ulaziti u to. Selekcija se provodi u svakoj generaciji osim u posljednjoj, ako je u njoj postignut neki od kriterija zaustavljanja. Najčešće metode koje se u tu svrhu koriste su metode turnirske selekcije i elitizma, a koje su ujedno i dostupne za korištenje u GPTIPS skupini alata. Turnirske selekcije u GPTIPS-u mogu biti [25]: standardna turnirska selekcija, Pareto turnirska selekcija te leksikografska turnirska selekcija. Prije nego li se krene s definiranjem tih metoda, potrebno je u konfiguracijskoj datoteci definirati veličinu turnira s kojom korisnik želi da algoritam radi, a to se postiže linijom [23]:

```
gp.selection.tournament.size = 10
```

#### 5.6.5.1. Standardna turnirska selekcija

Kod standardne turnirske selekcije, iz trenutne populacije (generacije) nasumično se bira prethodno definirani broj pojedinaca za turnir. Turnir bira ono rješenje koje najbolje rješava postavljene problem temeljem iznosa pogodnosti. Pobjednik ulazi u skupinu potencijalnih rješenja, namijenjenih da se nad njima provode genetske operacije (mutacije, križanja ili reprodukcije). Ukoliko se na tom pojedincu primjenjuje operacija mutacije ili reprodukcije, onda je za jednu operaciju dovoljno pružiti jednog roditelja, međutim u slučaju primjene križanja potrebna su dva roditelja. Ova metoda selekcije temelji se isključivo na usporedbi članova turnira i traženja najboljeg rješenja bez da se pruži informacija o tome za koji iznos je to rješenje bolje od nekog drugog. Zahvaljujući takvom pristupu, kao i nasumičnim uzorkovanjem pojedinaca za turnir, omogućena je evolucija i onim rješenjima koja nisu tako dobra u trenutnoj generaciji. Time je osigurano da jedno vrlo pogodno rješenje, ne može odmah preplaviti iduću generaciju i izazvati gubitak raznolikost prostora rješenja koji se pretražuje, a samim time i onemogućiti pronalaženje možda još boljeg [22]. Veličinom turnira, izravno se utječe na vjerojatnost izbora takvih pojedinaca s najvećim iznosom pogodnosti odnosno utječe se na veličinu tzv. selekcijskog pritiska. Veliki turniri izazivaju veliki selekcijski pritisak, odnosno selekciju koja jako pogoduje izboru pojedinaca sa najvećim iznosom pogodnosti, dok mali imaju obrnuti efekt. Kod manjih turnira važno je osigurati da nisu prekomjerno mali, jer tada raste vjerojatnost da se neka dobra rješenja nikad ne odaberu za turnir te ostanu zanemarena [19, 22]. Ovakva turnirska selekcija provodi se u GPTIPS-u, te ju nije potrebno posebno definirati već je potrebno definirati jedino njenu varijantu ili neke opcije, o čemu će više uslijediti u nastavku.

#### 5.6.5.2. Pareto turnirska selekcija

Ova vrsta turnirske selekcije u obzir uzima najbolje modele po pitanju njihovih performansi i kompleksnosti. Performansa modela na trening setu izražena je preko koeficijenta determinacije  $R^2$ . Ovaj turnir traži najpovoljniju kombinaciju tog koeficijenta (što je veći mogući) i kompleksnosti izraza (koja treba biti što manja) kako bi odabrao najbolje pojedince [26]. U konfiguracijskoj datoteci, udio ove selekcije definira se slijedećom linijom [23]:

```
gp.selection.tournament.p_pareto = 0.2
```

### 5.6.5.3. Leksikografska turnirska selekcija

Leksikografska turnirska selekcija ustvari nije posebna vrsta turnirske selekcije već samo dodatna opcija na standardnu. Kada je omogućena, u slučaju pojave dva pojedinca jednake pogodnosti u nekom turniru, bira onog koji ima nižu kompleksnost izraza. Ova opcija u GPTIPS-u se omogućuje linijom u konfiguracijskoj datoteci [23]:

```
gp.selection.tournament.lex_pressure = 'True'
```

### 5.6.5.4. Elitizam

Osim mogućnosti selekcije metodom turnira, moguće je provesti selekciju najboljih pojedinaca koji se izravno kopiraju u slijedeću generaciju bez da se nad njima provode genetske operacije [25]. Važno je naglasiti da se elitizam koristi kao dio selekcijskog mehanizma u paru sa nekom metodom selekcije, u ovom slučaju turnirske, te da se razlikuje od genetske operacije reprodukcije s kojom ju se često miješa. U GPTIPS-u je zadano da se 15% populacije kopira izravno u iduću, no ako korisnik želi definirati drugačiji udio to može učiniti u konfiguracijskoj datoteci kako slijedi [23]:

```
gp.selection.elite_fraction = 0.1
```

## 5.6.6. Genetske operacije

Bilo koja generacija u tijeku rada GP algoritma, modificira se pomoću genetskih operacija u nastojanju da se raspoloživa rješenja poboljšaju. Te operacije u kontekstu područja strojnog učenja, kojem GP pripada, predstavljaju tehnike kojima se prostor rješenja pretražuje. Genetske operacije koje GP algoritam provodi su: križanje (engl. *crossover*), mutacija (engl. *mutation*) i reprodukcija (engl. *reproduction*) [19]. U nastavku opisan je princip njihova djelovanja kao i upute vezane uz korištenje ovih operacija u GPTIPS-u.

### 5.6.6.1. Križanje

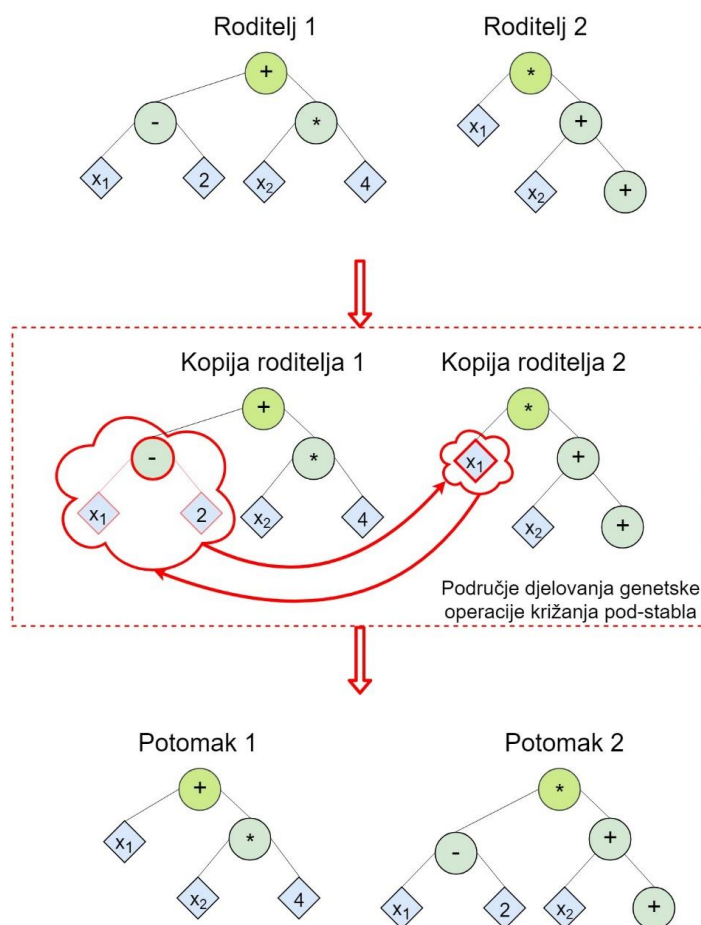
Križanje podrazumijeva izmjenu određene količine genetičkog materijala između roditelja odabranih nekom od metoda selekcije. Kako bi se u GPTIPS-u definirala učestalost korištenja genetske operacije križanja, potrebno je u konfiguracijskoj skripti upisati [23]:

```
gp.operators.crossover.p_cross = 0.7
```

Ukoliko se prethodna linija ne upotrijebi, zadana vrijednost koja se upotrebljava je 84%. Kod GPTIPS-a, koriste se dvije vrste križanja: standardno križanje pod-stabala i križanje stabla između multi-genskih pojedinaca. Križanje pod-stabala jednostavno se naziva standardno križanje, dok se križanja koja podrazumijevaju upotrebu multi-genskog načina rada nazivaju križanje na nižoj razini (engl. *low level crossover*) te križanje na visokoj razini (engl. *high level crossover*) [25].

Standardno križanje pod-stabala

Za dva pojedinca odabrana nekom od metoda selekcije, standardno križanje pod-stabala proizvoljno bira točku križanja u oba roditelja. Nakon toga, zamjenom pod-stabala ukorijenjenih u odabranim točkama križanja, stvara dva nova pojedinca koji pripadaju novoj generaciji. Roditelji, od kojih novi pojedinci nastaju, ostaju nepromijenjeni jer se ustvari koriste njihove kopije kako bi se izbjegao poremećaj izvornih pojedinaca. Na taj način isti pojedinci mogu biti odabrani za roditelje više puta [19]. Ako nije omogućen multi-genski način rada u GPTIPS-u, sva križanja koja se vrše nad nekom populacijom ovog su tipa.



**Slika 19. Postupak kod operacije križanja pod-stabala**



Prva vrsta križanja koju GPTIPS koristi kod križanja multi-genskih pojedinaca ukoliko je omogućen takav način rada. Koristi se umjesto standardnog križanja pod-stabala koje nije pogodno za izmjenu čitavih stabala kod MGGP načina rada u kojem se pojedinci mogu sastojati od više njih. Vjerojatnost upotrebe definira se na isti način kao i za standardno križanje, odnosno GPTIPS povlači taj iznos iz varijable `gp.operators.crossover.p_cross`. Ova metoda ustvari predstavlja proširenje standardne metode križanja jer na početku svog rada bira po jedan gen (stablo) iz dva multi-genska roditelja, odabrana nekom od metoda selekcije, nakon čega ta stabla ulaze u prethodno opisanu standardnu metodu križanja pod-stabala kao roditelji. Takvo standardno križanje stvara dva nova potomka kojima se u potpunosti mijenjaju roditelji (geni) u multi-genskim modelima iz kojih potječu [25].

#### Križanje više razine

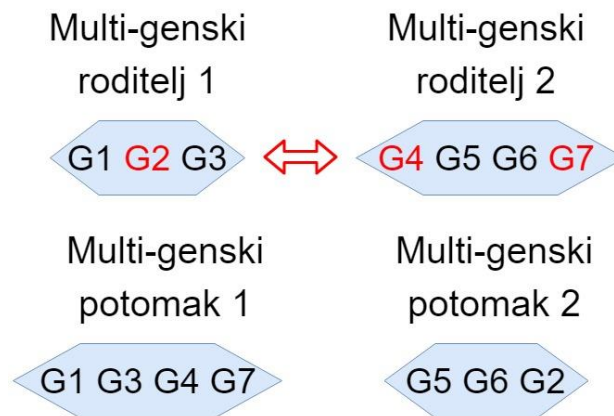
Kod ove vrste križanja, vrši se izmjena jednog ili više čitavih gena (stabla) između multi-genskih pojedinaca. Učestalost primjene ove vrste križanja definira se u konfiguracijskoj skripti kako slijedi [23]:

```
gp.genes.operators.p_cross_hi = 0.3
```

Unutar GPTIPS-a je zadano da se ova vrsta upotrebljava u 20% svih slučajevima u kojima se koristi operacija križanja, ali prethodnom linijom se to može promijeniti. Kako bi metoda odlučila koji geni će se u odabranim pojedincima izmjenjivati, koristit se tzv. parametar križanja - **PK** (za koji vrijedi  $0 < PK < 1$ ). U GPTIPS-u je za taj parametar unaprijed zadan iznos 0.5, ali ako to korisnik želi promijeniti, može to učiniti u konfiguracijskoj skripti sa [25]:

```
gp.genes.operators.p_cross_rate = 0.4
```

Pri definiranju tog parametra, važno je da on bude u spomenutom rasponu između 0 i 1. Za svaki se gen u odabranim pojedincima, zasebno generira nasumični broj unutar tog intervala. Ukoliko je generirani broj za neki od gena manji ili jednaki od iznosa **PK**, onda se taj gen prebacuje u drugog multi-genskog pojedinca. Na taj način ispituju se svi geni od kojih se dva multi-genska pojedinca sastoje sve dok se ne naprave sve (ili niti jedna) izmjene. Primjer rada ove vrste križanja dan je na slici u nastavku.



**Slika 20.** Princip rada križanja više razine uz  $G_{max} = 5$

U slučaju da se nakon završetka svih izmjena pojavi jedan multi-genski pojedinac koji je prekoračio maksimalni dopušteni broj gena  $G_{max}$  od kojih se smije sastojati, višak gena se uklanja nasumičnim odabirom i njihovim brisanjem dok se ne zadovolji traženi uvjet [25].

#### 5.6.6.2. Mutacija

Operacija mutacije koristi samo jednog po jednog pojedinca kojeg mehanizam selekcije odabere. Ovdje se radi o promjeni manje količine podataka u odabranim pojedincima gdje mutacija proizvoljno odabire jednu točku u stablu gdje će se promjena dogoditi [19]. Zadana vjerojatnost primjene mutacije nad nekim pojedincem u GPTIPS-u je 14%, a ukoliko korisnik to želi promijeniti, može to učiniti u konfiguracijskoj skripti upisivanjem [23]:

```
gp.operators.mutation.p_mutate = 0.2
```

GPTIPS skupina alata raspolaže sa 6 operacija mutacije [23]: standardna mutacija pod-stabla, zamjena ulaznog operanda sa nasumično odabranim ulaznim operandom, Gaussova perturbacija nasumično odabrane konstante, nasumičan odabir konstante i njezine zamjene s nulom, zamjena nasumično odabrane konstante sa nasumično generiranom te zamjena nasumično odabrane konstante sa jedinicom. Učestalost primjene pojedine vrste mutacije, u slučaju da se kao genetska operacija uopće odabere mutacija, može se podesiti ručno u konfiguracijskoj skripti kako slijedi [23]:

```
gp.operators.mutation.mutate_par = [0.8 0.1 0.5 0.5 0 0]
```

U prethodnom izrazu zadane su učestalosti korištenja pojedine vrste mutacija i to prema redoslijedu koji je korišten i pri prethodnom nabranjanju vrsta. U GPTIPS-u, zadane vrijednosti primjena tih mutacija su: [0.9 0.05 0.05 0 0 0]. Važno je napomenuti da kada korisnik definira te vrijednosti, mora paziti da one daju zbroj 1 odnosno 100% kako bi GP algoritma ispravno funkcionirao u slučaju mutacije [23]. Kod gotovo svih prethodno navedenih mutacija, jasno je iz samog njihovog naziva što rade, pa o njima ne treba previše govoriti. U nastavku će biti rečeno nešto više samo o mutaciji pod-stabla kao i o Gaussovoj perturbaciji nasumično odabranih konstanti jer se uz njih vežu još neki dodatni upravljački parametri.

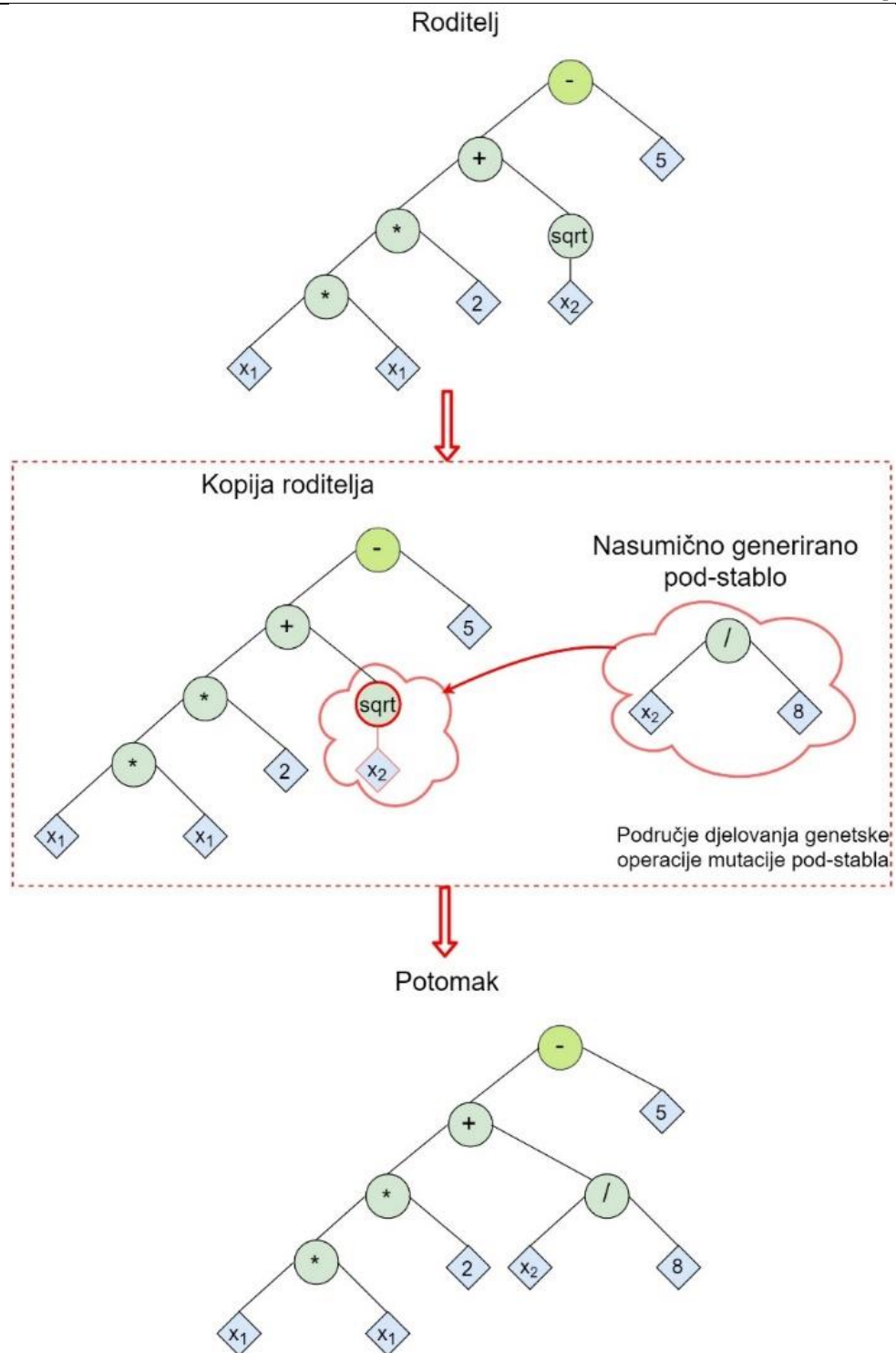
#### Gaussova perturbacija nasumično odabranih konstanti

U slučaju u kojem se primjenjuje Gaussova perturbacija nasumično odabrane konstante, moguće je definirati standardnu devijaciju nasumične konstante koja će se dodati konstanti na nasumično odabranoj točki. Standardna devijacija je mjera raspršenosti podataka koja govori o tome koliko je prosječno kvadratno odstupanje nekog podatka od prosjeka. Perturbacija se provodi na način da se staroj konstanti, koja se nalazi na odabranoj točki mutacije u stablu, doda neki nasumično generiran broj pomnožen sa zadanom standardnom devijacijom Gaussove (normalne) raspodjele. Ta standardna devijacija može biti broj između 0 i 1, a u GPTIPS-u je zadani iznos 0.1. Ako korisnik želi taj iznos promijeniti, potrebno je u konfiguracijskoj skripti napisati [23]:

```
gp.operators.mutation.gaussian.std_dev = 0.2
```

#### Mutacija pod-stabala

Mutacija pod-stabla najčešće je upotrebljavana vrsta mutacije kod GP-a. Ona nasumično odabire neku mutacijsku točku u stablu te mijenja pod-stablo ukorijenjeno u toj točki sa nasumično generiranim pod-stablom [22]. Princip rada prikazan je na slijedećoj slici.



Slika 21. Princip rada operacije mutacije pod-stabala

Tijekom nasumičnog generiranja pod-stabla, metoda mora voditi računa o definiranoj maksimalnoj dopuštenoj dubini stabala  $d_{max}$ , koja je definirana još na samom početku rada s GP-om. Postoji jedan korisnički parametar koji se može podesiti, a koji će definirati koliko blisko korijenskom čvoru metoda mutacije pod-stabla može birati točke. Taj parametar se zove maksimalna dopuštena dubina pod-stabala stvorenih mutacijom. Tu dubinu moguće je proizvoljno zadati u konfiguracijskoj skripti kako slijedi [23]:

```
gp.treedef.max_mutate_depth = 3
```

U GPTIPS skupini alata zadana je vrijednost maksimalne dubine pod-stabala kreiranih mutacijom od 4, zbog toga što je unaprijed zadana maksimalna dopuštena dubina svih stabala  $d_{max}$  isto jednaka 4, ukoliko ju korisnik nije promijenio. To znači da je unaprijed zadano da ova operacija mutacije može mutirati čak cijela odabrana stabla. Prema tome, u slučaju da je korisnik promijenio iznos  $d_{max}$ , potrebno je promijeniti i ovu dubinu kako bi se sve poklapalo.

#### 5.6.6.3. Reprodukција

Operacija reprodukcije je posljednja genetska operacija koju je moguće koristiti. Princip kod nje je jednostavan. Određeni udio pojedinaca najviše pogodnosti izravno se, bez bilo kakvih promjena, kopira u iduću generaciju. Kod GPTIPS-a je zadano da se samo 2% populacije kopira na taj način, ali taj iznos korisnik može u konfiguracijskoj skripti promijeniti upisivanjem [23]:

```
gp.operators.directrepro.p_direct = 0.05
```

## 6. KREIRANJE MATEMATIČKOG MODELA ZA PREDVIĐANJE REZIDUALNOG KLORA PRIMJENOM GP-a

### 6.1. Opis problema

Kvalitativna cjelovitost vodoopskrbnih sustava prvenstveno se održava procesom kondicioniranja vode koji je opisan u prethodnim poglavljima ovog rada. U tom je procesu jedan od najvažnijih koraka dezinfekcija vode. Dezinfekcija predstavlja skup postupaka koji se provode s ciljem uklanjanja i onemogućavanja kolonizacije različitih vrsta patogenih mikroorganizama u sustavu [7]. S obzirom da se u Hrvatskoj u postupku dezinfekcije najčešće upotrebljava klor, primjena modela za predikciju rezidualnog klora u mreži bila bi korisna u ispitivanju doziranja [13]. Prema tome, takav model bi se upotrebljavao za kontrolu doziranja klora na izvoru kako bi se iznosi slobodnog (rezidualnog) klora u mreži držali unutar propisanih granica (za RH maksimalna gornja granica je 0,5 mg/L). Kloriranje se provodi na izvoru gdje se nalazi kontrolna ploča za doziranje klora i mjerenje koncentracija rezidualnog klora u distribucijskoj mreži. Izgled takve kontrolne ploče prikazan je na sljedećoj slici.



**Slika 22.** Kontrolna ploča u postrojenju za kemijsku dezinfekciju vode s vrijednošću SRK u vodoopskrbnoj mreži [7]

U mreži se rezidualni klor javlja kao ostatak klora nakon inaktivacije patogena te oksidacije organskih i anorganskih tvari [7]. Preniska koncentracija može uzrokovati niz zdravstvenih problema kod krajnjih korisnika, jer se vjerojatnost naknadne kontaminacije na pojedinim lokacijama u distribucijskoj mreži znatno povećava. S druge strane, previsoka koncentracija također nije dobra jer može biti štetna po zdravlje te uzrokovati pojavu neugodnog okusa, mirisa i boje. Najpogodnija koncentracija nalazi se u rasponu između 0.2 i 0.5 mg/L [31]. Gornja granica tog raspona ujedno je i maksimalna dopuštena pravilnikom, dok je donja granica i dalje dovoljna za efikasnu dezinfekciju, a nalazi se daleko ispod maksimalno dopuštenog iznosa. Mogućnost da se vodoopskrbni sustav drži unutar tog raspona te što je bliže donjoj granici, nije prednost samo zbog kvalitetnije usluge korisnicima, nego i zbog manjeg utroška novčanih sredstava za nabavku i doziranje klora. U tu svrhu će se u ovom radu pomoću algoritma genetskog programiranja (GP) pokušati pronaći matematički model za predviđanje rezidualnog klora u mreži distribucije pitke vode. GP algoritam, kao jedna od tehnika strojnog učenja, odabran je zbog mogućnosti da nauči i prikaže složene nelinearne procese s kojima se najčešće susrećemo u prirodi, pa tako i procese koji se odvijaju uslijed dezinfekcije unutar vodovodne mreže. Dobiveni model, trebao bi biti u stanju na temelju ulaznih podataka o doziranom kloru i podataka o koncentracijama bakterija, predvidjeti kretanja iznosa rezidualnog klora te time doprinijeti u planiranju održavanja kvalitete vode.

## 6.2. Eksperimentalni podaci

Nad kondicioniranom vodom iz rijeke Jadro, kojom se opskrbljuju Split, Solin, Kaštel i Trogir, vršila se kemijska i mikrobiološka analiza u pravilnim vremenskim razmacima (jednom mjesečno) kroz 6 godina. U tom periodu iz slavina potrošača uzeto je 70 uzoraka nad kojima se vršilo ispitivanje po ISO standardima. Veličine koje su se iz tih ispitivanja izdvojile su: dodani klor, ukupni koliformi, enterokoki, *Escherichia coli*, rezidualni klor, ukupni broj kolonija bakterija pri 37 °C te ostale bakterije. Dio tih podataka koji su upotrijebljeni za modeliranje u ovom radu, kao i njihovi osnovni statistički pokazatelji, prikazani su slijedećim tablicama.

Tablica 13. Dio prikupljenih podataka za modeliranje

Dodani Cl	Ostale bakterije	Ukupni koliformi	Enterokoki	Escherichia coli	Rezidualni klor
mg/L	CFU x 10 <sup>-2</sup> / 1 mL	broj/100 mL	broj/100 mL	broj/100 mL	mg/L
x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	y <sub>1</sub>
0.5	199650	190	80	80	0.23
0.5	49800	200	0	0	0.25
0.5	9977	18	5	0	0.18
0.35	3074	9	5	12	0.18
0.5	12990	108	0	2	0.2
0.35	1292	1	2	5	0.25
0.5	34960	7	0	33	0.2
...	...	...	...	...	...

### 6.3. Dobivanje inicijalnog modela

Da bi se nad osnovnim skupom podataka od 70 unosa mogao primijeniti GP algoritam pomoću GPTIPS skupine alata za MATLAB, najprije ga je potrebno nasumično podijeliti na trening i test skup. To je napravljeno pisanjem skripte pohranjene pod nazivom *podjela\_podataka\_vodovod\_tren\_test.m* gdje je osnovni skup od 70 podataka nazvan *osnovni\_podaci*. Ta skripta dana je u nastavku.

```
%Podjela podataka na TREN i TEST set te pohrana

load ('osnovni_podaci')

%definiranje postotka podataka za trening set
p_tren = 0.7;

%podjela seta podataka na trening i testni set u definiranom omjeru
[m,n] = size (osnovni_podaci);
index = randperm(m);
vodovod_trening = osnovni_podaci (index(1:round(p_tren*m)),:);
vodovod_test = osnovni_podaci (index(round(p_tren*m)+1:end),:);

save ('vodovod_tren_test_osnovno');
```



Podjela je napravljena u omjeru 70:30% u korist trening skupa te su podaci za daljnju upotrebu pohranjeni pod nazivom *vodovod\_tren\_test\_osnovno*. Sa tako podijeljenim skupom podataka sada se kreće u modeliranje GP algoritmom. Parametri koji su korišteni u radu algoritma dani su slijedećom tablicom.

**Tablica 14. Parametri za rad GP algoritma pri dobivanju inicijalnog modela**

Veličina populacije	2000
Maksimalni broj generacija	200
Ulaznih varijabli	5
Trening podataka	49
Veličina turnira	50
Udio elitizma	0.1
Leksikografski turnirski pritisak	DA
Vjerojatnost pareto turnira	0.3
Broj gena multi-genskog pojedinca	4
Maksimalna dubina stabala	5
Vjerojatnost nasumične konstante	0.1 (vjerojatnost cijelog broja 0.2)
Vjerojatnost križanja	0.84 (vjerojatnost križanja visoke razine 0.2, vjerojatnost križanja niske razine 0.8)
Vjerojatnost mutacije	0.14 (pod-stabala 0.9, izmjena operanda 0.05, perturbacija po Gaussu 0.05)
Operatorski skup	times, rdivide, plus, minus, power, exp, negexp, square, cube, sqrt, sin, cos, tan, cot, asin, acos, atan, acot, abs, log, log10

Konfiguracijska skripta, kao i skripta za pokretanje, dane su u prilogu na kraju rada pod nazivima: *Vodovod\_konfig.m* i *Vodovod\_pokretanje\_osnovno.m*.

Nakon dovršetka rada algoritma, dobiven je inicijalni model slijedećeg oblika:

$$\begin{aligned}
y_1 = & 5.39^{-5} \cdot \tan(\ln(x_2))^2 - 0.0144 \cdot \sin(\tan(|\cos^{-1}(x_3)|)) - 0.0179 \cdot \cos(x_3^3) - 0.0269 \\
& \cdot \sin(0.0311 \cdot x_4 + (\ln(x_2))^3) - 0.0179 \cdot \sin\left(\frac{9.97}{x_1}\right) - 5.39^{-5} \cdot \sin(x_3) \\
& - 32.1) \cdot (x_2 + x_3)^{1/2} - 0.0179 \cdot \cos(2.87^{-5} \cdot x_3^3) + 0.24
\end{aligned} \quad (30)$$

Performanse ovog modela nad trening i test skupom podataka dane su u nastavku.

**Tablica 15. Performanse inicijalnog modela na trening skupu podataka**

TRENING SKUP	
Metrika uspješnosti	Vrijednost
R <sup>2</sup>	0.73945
RMSE	0.017865
MAE	0.013609
SSE	0.015638
Maks. aps. greška	0.059929
MSE	0.00031914

**Tablica 16. Performanse inicijalnog modela na testnom skupu podataka**

TEST SKUP	
Metrika uspješnosti	Vrijednost
R <sup>2</sup>	-0.94298
RMSE	0.041622
MAE	0.031597
SSE	0.03638
Maks. aps. greška	0.10121
MSE	0.0017324

Legenda:

R<sup>2</sup> – koeficijent determinacije

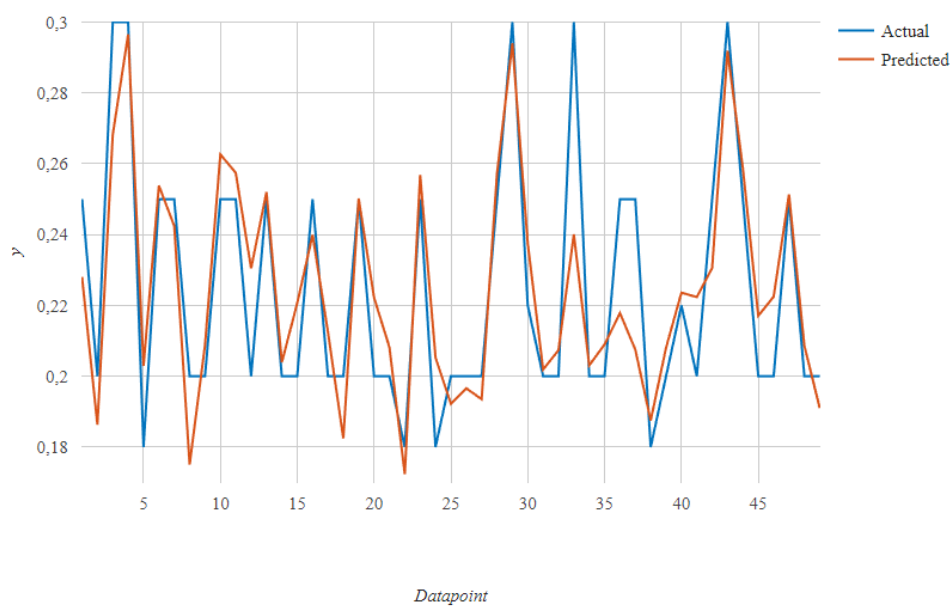
RMSE – korijen prosječne kvadrirane greške (engl. *Root mean squared error*)

MAE – prosječna apsolutna greška (engl. *Mean absolute error*)

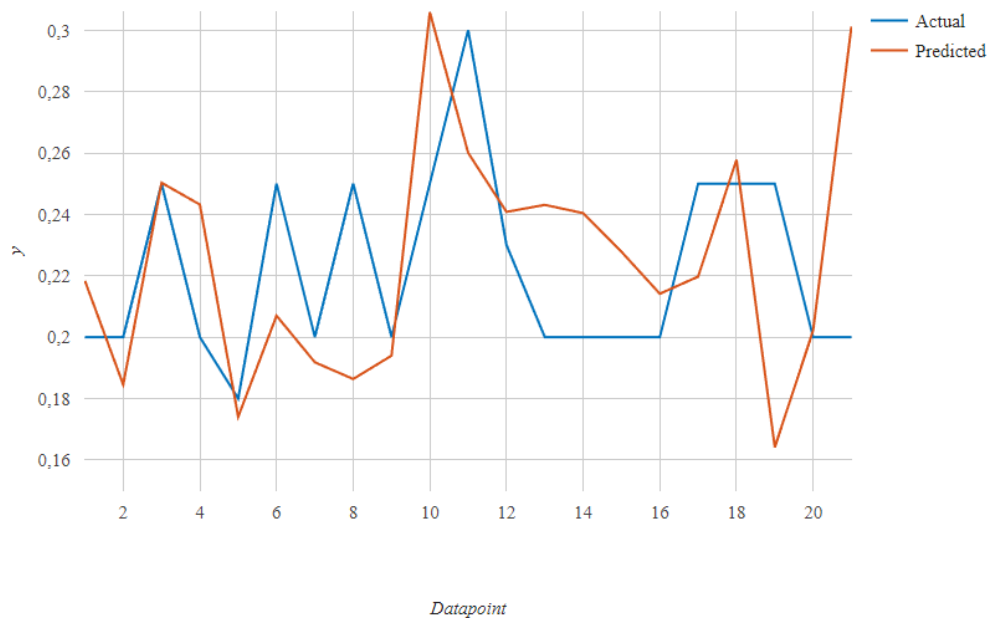
SSE – suma kvadriranih greški (engl. *Summ of squared erros*)

MSE – prosječna kvadrirana greška (engl. *Mean squared error*)

Grafički prikaz predikcija usporedno sa traženim stvarnim vrijednostima za svaki skup dani su na slijedećim slikama.



**Slika 23.** Usporedni prikaz svih predviđanja i traženih vrijednosti inicijalnog modela na trening skupu podataka



**Slika 24.** Usporedni prikaz predviđanja i traženih vrijednosti inicijalnog modela na testnom skupu podataka

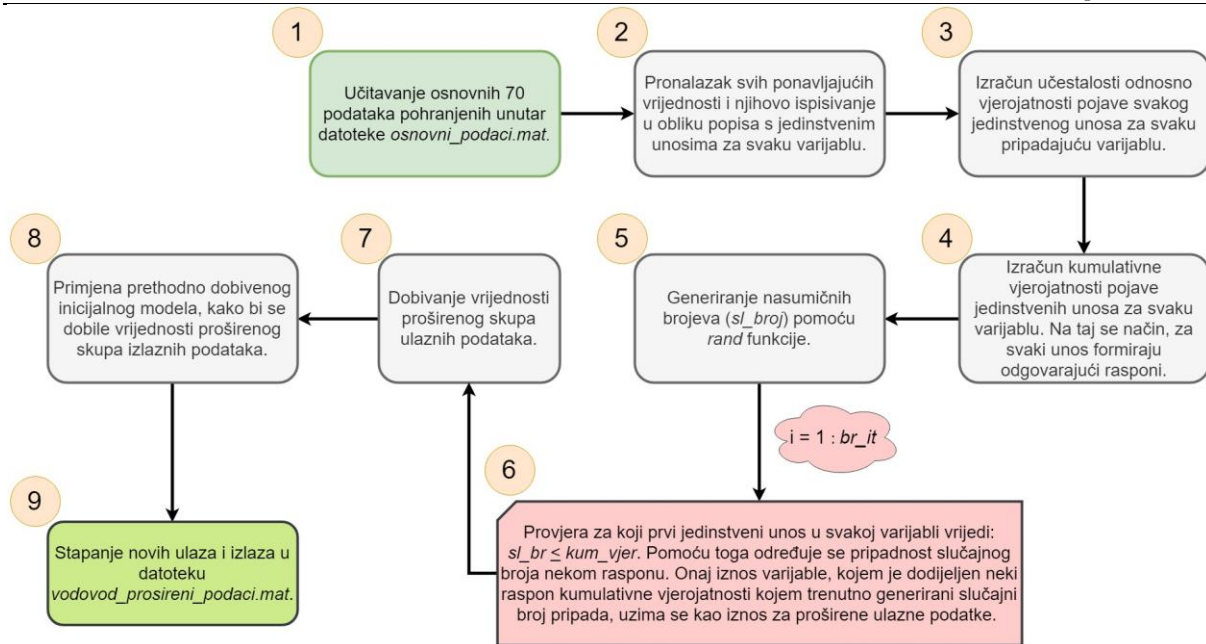
#### 6.4. Povećanje skupa podataka primjenom Monte Carlo simulacije

Prethodno dobiveni inicijalni model, sada će se koristiti u sklopu Monte Carlo simulacije kako bi se povećao osnovni skup podataka za daljnje modeliranje. Izradi ove simulacije pristupilo se zbog posjedovanja relativno male količine podataka (70) te njihove raspršenosti. Dokaz raspršenosti podataka dan je u slijedećoj tablici u obliku standardnih devijacija za svaku varijablu iz osnovnog skupa.

Varijabla	Prosječna vrijednost	Standardna devijacija	Minimum	Maksimum
Dodani Cl - $x_1$	0.433571	0.074509	0.35	0.5
Ostale bakterije - $x_2$	47651.59	84937.88	482	399951
Ukupni koliformi - $x_3$	323.5286	553.2934	0	2500
Enterokoki - $x_4$	62.55714	304.5863	0	2500
Escherichia coli - $x_5$	33.75714	81.99346	0	420
Rezidualni klor - $y_1$	0.223857	0.033564	0.18	0.3

Slika 25. Osnovni statistički pokazatelji osnovnog skupa

U tu svrhu izrađena je skripta pod nazivom *MC\_simulacija\_vodovod* koja se nalazi u prilogu ovog rada. Monte Carlo simulacija je tehnika koja pomoću generiranja nasumičnih brojeva i velikog broja ponavljanja, nastoji predvidjeti ponašanje složenih realnih sustava. U ovom slučaju, simulacija se temelji na generiranju nasumičnih brojeva i formiranju raspona temeljem kumulativnih vjerojatnosti za svaki podatak u osnovnom skupu. Za svaki generirani nasumični broj, provjerava se kojem rasponu on pripada, te se iz tog raspona izvlači pripadajuća vrijednost. Tako se radi za svaku varijablu uz ponavljanje definirano varijablom *br\_iter*. Kod ove simulacije korišteno je 1000 iteracija, prilikom čega je formiran novi, prošireni skup podataka temeljem vjerojatnosti pojave podataka u osnovnom skupu. Na kraju, da bi se dobile izlazne vrijednosti, na dobivenom proširenom skupu podataka upotrijebljen je inicijalni model. Novi dobiveni podaci, pohranjeni su pod nazivom *vodovod\_prošireni\_podaci*. Radi boljeg razumijevanja postupaka primijenjenih u ovom koraku, odnosno postupaka Monte Carlo simulacije, izrađen je dijagram tijekom koji se nalazi na slijedećoj slici.



Slika 26. Dijagram tijeka Monte Carlo simulacije izrađene za povećanje skupa podataka

## 6.5. Dobivanje konačnog modela

Prethodno prošireni podaci sada se koriste kao ulazni podaci za ponovno modeliranje. Parametri za rad GP algoritma identični su onima prikazanim kod dobivanja inicijalnog modela, pa ovdje neće biti posebno istaknuti. Podaci su ponovno podijeljeni na 70% trening podataka i 30% testnih podataka, pomoću skripte koja je ekvivalentna onoj prikazanoj kod podjele podataka za inicijalni model (ova se zove *podjela\_podataka\_vodovod\_tren\_test.m*). Konfiguracijska skripta, kao i skripta za pokretanje, dane su u prilogu na kraju rada pod nazivima: *Vodovod\_konfig\_prosireno.m* i *Vodovod\_pokretanje\_prosireno.m*.

Nakon dovršetka rada algoritma, dobiven je konačni model slijedećeg oblika:

$$\begin{aligned}
 y_1 = & 0.0169 \cdot \sin(x_3 - x_4) - 0.0161 \cdot \cos(x_4 - x_5) - 0.00827 \cdot \sin(x_1 + 2 \cdot x_2) - 9.46^{-5} \\
 & \cdot \sin(x_2 - x_3) - 1.89^{-4} \cdot \sin(x_2 - x_4) - 0.00827 \cdot \sin(4.91 \cdot x_3 \cdot x_4) \\
 & - 0.0252 \cdot \sin(x_3 \cdot x_4) + 0.0161 \cdot \sin(x_3 \cdot x_5) + 0.0161 \cdot \sin(x_4 \cdot x_5) \\
 & - 0.0169 \cdot \sin(\sin(\sin(x_2))) - 0.00827 \cdot \sin(x_3) - 0.0169 \cdot \sin(5 \cdot x_4^2) \\
 & + 9.46^{-5} \cdot x_4 \cdot \cos(x_5) + 0.231
 \end{aligned} \tag{31}$$

Pojedinačni geni od kojih se sastoji prikazani su idućom slikom.

Bias	0.231
Gene 1	$9.46e-5 x_4 \cos(x_5) - 1.89e-4 \sin(x_2 - 1.0 x_4) - 9.46e-5 \sin(x_2 - 1.0 x_3)$
Gene 2	$0.0169 \sin(x_3 - 1.0 x_4) - 0.0169 \sin(x_3 x_4) - 0.0169 \sin(\sin(\sin(x_5))) - 0.0169 \sin(5.0 x_4^2)$
Gene 3	$-0.00827 \sin(4.91 x_3 x_4) - 0.00827 \sin(x_1 + 2.0 x_2) - 0.00827 \sin(x_3 x_4) - 0.00827 \sin(x_3)$
Gene 4	$0.0161 \sin(x_3 x_5) - 0.0161 \cos(x_4 - 1.0 x_5) + 0.0161 \sin(x_4 x_5)$

**Slika 27. Geni od kojih se sastoji konačni model**

Performanse ovog modela nad trening i test skupom podataka dane su u nastavku.

**Tablica 17. Performanse konačnog modela na trening skupu podataka**

TRENING SKUP	
Metrika uspješnosti	Vrijednost
$R^2$	0.8467
RMSE	0.012172
MAE	0.0091273
SSE	0.1037
Maks. aps. greška	0.039755
MSE	0.00014815

**Tablica 18. Performanse konačnog modela na testnom skupu podataka**

TEST SKUP	
Metrika uspješnosti	Vrijednost
$R^2$	0.82558
RMSE	0.014995
MAE	0.010442
SSE	0.067456
Maks. aps. greška	0.092014
MSE	0.00022485

Legenda:

$R^2$  – koeficijent determinacije

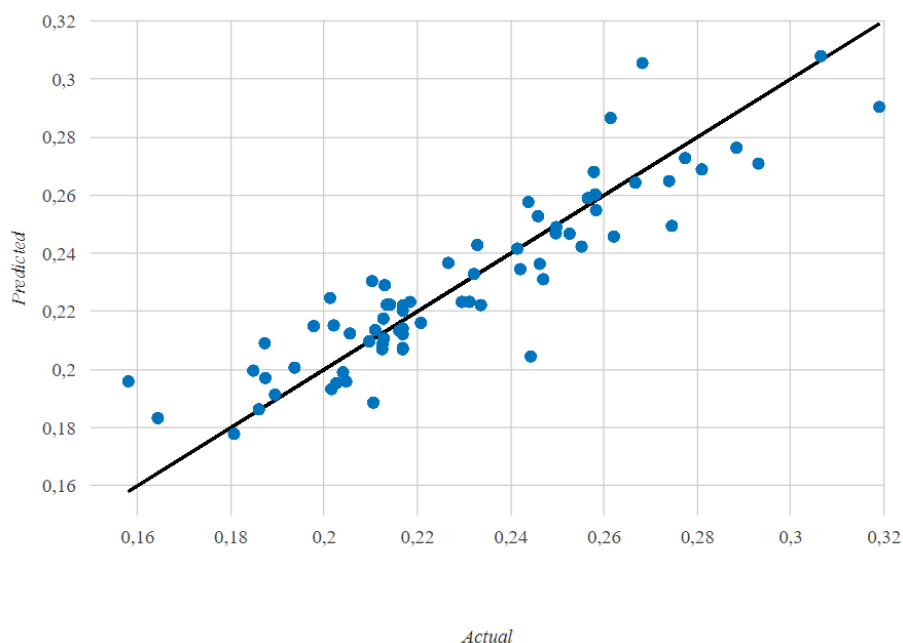
RMSE – korijen prosječne kvadrirane greške (engl. *Root mean squared error*)

MAE – prosječna apsolutna greška (engl. *Mean absolute error*)

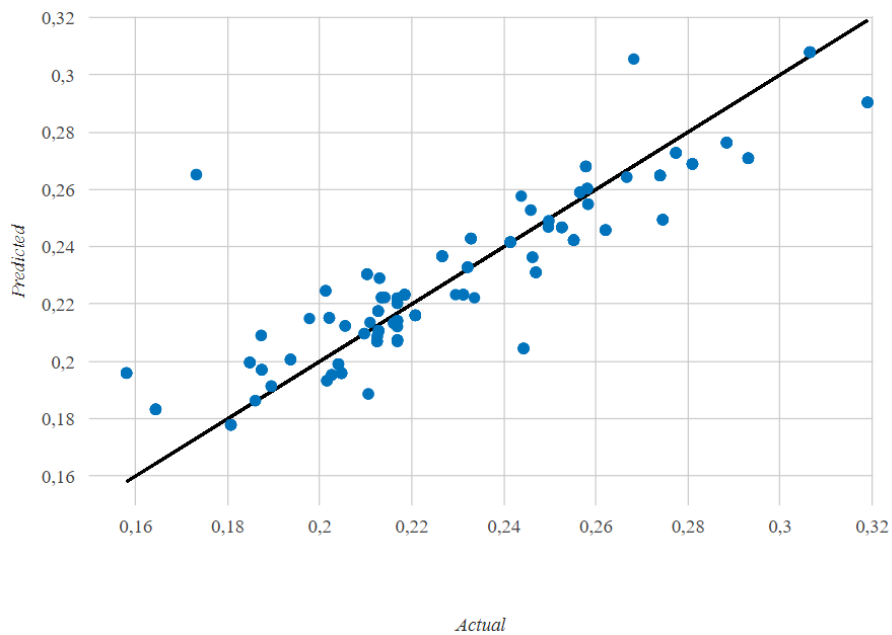
SSE – suma kvadriranih greški (engl. *Summ of squared erros*)

MSE – prosječna kvadrirana greška (engl. *Mean squared error*)

Moguće je uočiti da je dobiven veći koeficijent determinacije na trening skupu, nego što je to bio slučaj kod inicijalnog modela, međutim taj isti iznos na testnom skupu je nešto lošiji. Bez obzira na to, zbog ulaznih podataka dobivenih Monte Carlo metodom, ovaj model trebao bi bolje generalizirati na podacima, a u kombinaciji sa znatno nižim korijenom prosječne kvadrirane greške (standardnom devijacijom greški), trebao bi biti i pouzdaniji u predviđanju. Slike u nastavku prikazuju koliko dobro model predviđa na trening i testnom setu u obliku raspršenog dijagrama s prikazom stvarnih i predviđenih vrijednosti. Pravac na toj slici simbolizira savršeno predviđanje.

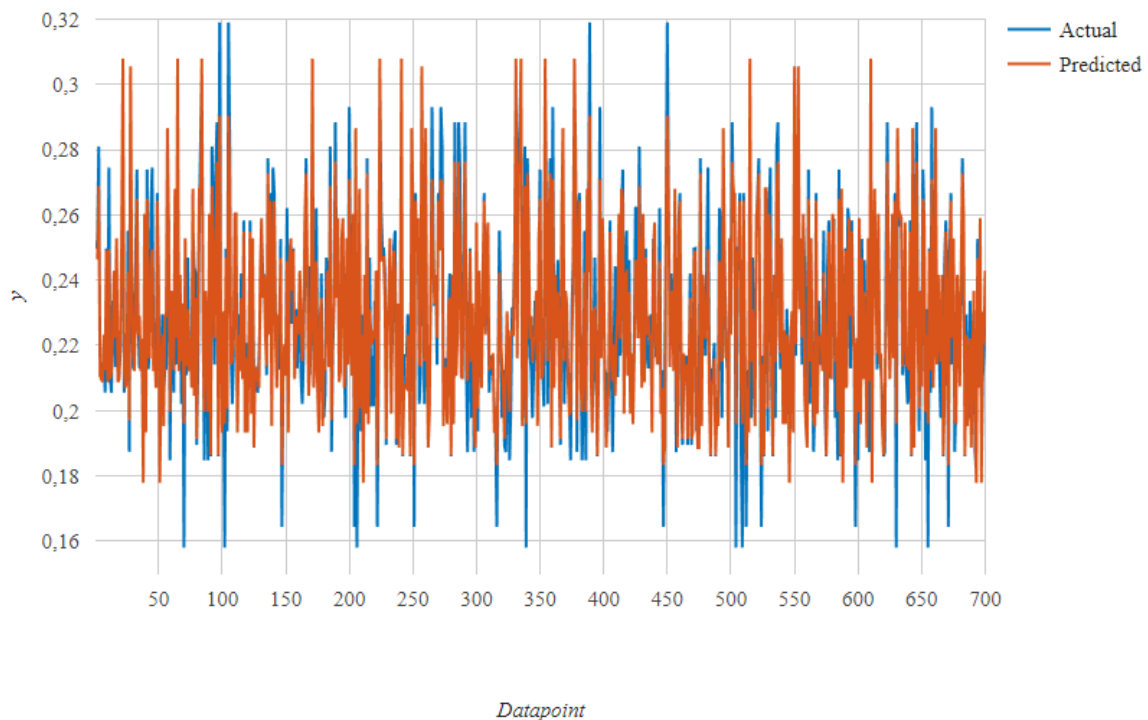


**Slika 28.** Raspršeni dijagram pogađanja konačnog modela na trening skupu podataka



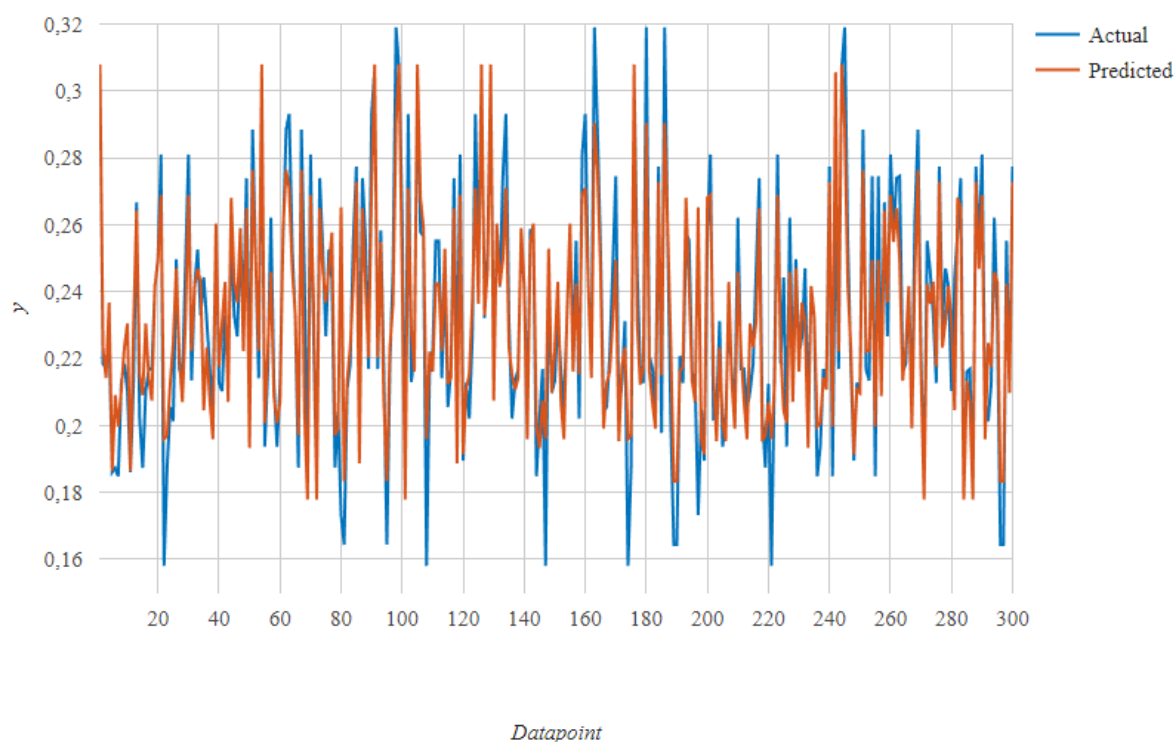
**Slika 29.** Raspršeni dijagram pogađanja konačnog modela na testnom skupu podataka

Prikaz svih predikcija i traženih vrijednosti na trening i test setu prikazan je u nastavku.



**Slika 30.** Sve predikcije i tražene vrijednosti za konačni model na trening skupu podataka





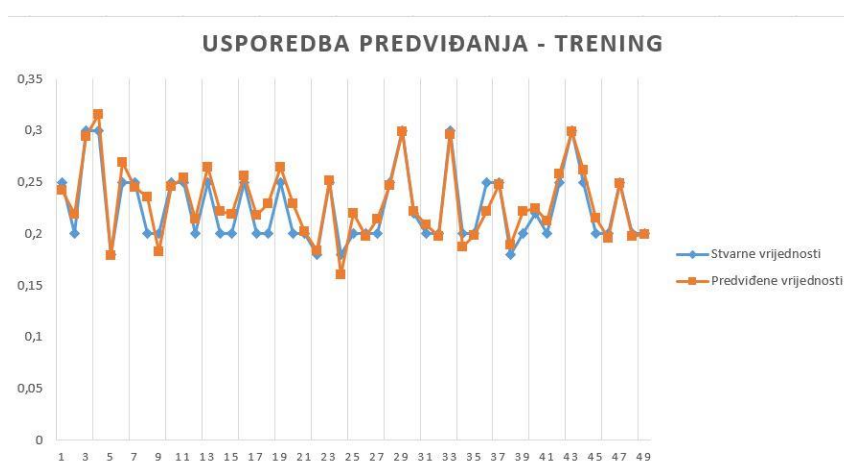
**Slika 31.** Sve predikcije i tražene vrijednosti za konačni model na testnom skupu podataka

U nastavku slijedi prikaz kako konačni model, dakle onaj koji je proizašao iz proširenih podataka dobivenih MC simulacijom, predviđa na osnovnom skupu od 70 podataka. Da bi se to ilustriralo, upotrijebit će se tablica sa usporednim podacima o rezidualnom kloru iz osnovnog skupa od 70 podataka i podacima dobivenim temeljem predviđanja ovim modelom. Na kraju tablice stavljeni su osnovni statistički pokazatelji.

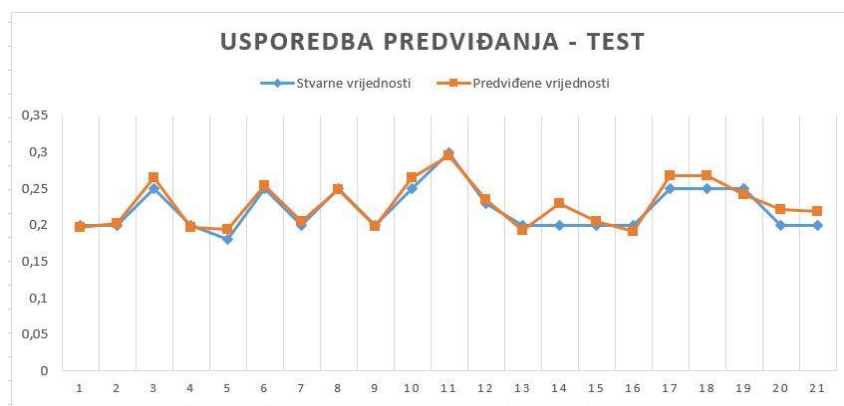
**Tablica 19.** Usporedba predviđanja konačnim modelom i stvarnih mjerenja sadržanih u osnovnom skupu podataka

Iznosi rezidualnog CI predviđeni konačnim modelom [mg/L]	Vrijednosti rezidualnog CI temeljem stvarnih mjerenja [mg/L]	Razlika [%]
0.294131	0.3	1.995
0.235489	0.2	15.070
0,160435	0.18	12.194
0.296179	0.3	1.289
0.197466	0.2	1.283
0.248736	0.25	0.508
0.205290	0.2	2.576
...	...	...
$\bar{x} = 0.229665$ $\sigma = 0.033581$ MIN = 0.160436 MAX = 0.315284	$\bar{x} = 0.223857$ $\sigma = 0.033564$ MIN = 0.18 MAX = 0.3	$\bar{x} = 4.647684$ $\sigma = 3.882825$ MIN = 0.064041 MAX = 15.07034

Kada se prethodne vrijednosti prikažu na raspršenom dijagramu, posebno za trening set, posebno za testni set, dobivaju se sljedeće slike.



**Slika 32.** Usporedba stvarnih i predviđenih vrijednosti – trening skup



Slika 33. Usporedba stvarnih i predviđenih vrijednosti – testni skup

Izračunate performanse ovog modela nad trening i test skupom osnovnih podataka, dane su u nastavku.

Tablica 20. Performanse konačnog modela na osnovnom trening skupom podataka

TRENING SKUP	
Metrika uspješnosti	Vrijednost
$R^2$	0.868511
RMSE	0.013981
MAE	0.010689
SSE	0.009579
Maks. aps. greška	0.035489
MSE	0.000195

**Tablica 21. Performanse konačnog modela na osnovnom testnom skupu podataka**

<b>TEST SKUP</b>	
Metrika uspješnosti	Vrijednost
$R^2$	0.8769923
RMSE	0.012683
MAE	0.010072
SSE	0.003378
Maks. aps. greška	0.029470
MSE	0.000161

## 7. ZAKLJUČAK

O vodi koja je čista i zdravstveno ispravna ovisi vrlo veliki broj stanovništva priključenog na vodoopskrbnu mrežu. Na zaposlenicima vodovoda je stoga velika odgovornost kako bi se ispunila osnovna svrha pružanja vode onda kada ona treba, u potrebnim količinama i potrebnoj kakvoći. Ta svrha, može se sročiti u tri osnovna zahtjeva cjelovitosti sustava o kojima vodovod mora brinuti, a to su: fizička, hidraulička i kvalitativna cjelovitost. Navedeni zahtjevi lakše se ispunjavaju razvojem kvalitetnih strategija upravljanja i održavanja. Prilikom razvoja takvih strategija, najbolji je pristup koji započinje razvojem plana upravljanja imovinom. Plan upravljanja imovinom predstavlja holistički pristup kojim se evidentira sva imovina, vrši se procjena njenog stanja, identificiraju se i procjenjuju rizici kao i buduće potrebe sustava te se temeljem svih prikupljenih informacija razvija strategija održavanja. Iako takav pristup zahtjeva više vremena i novaca, on ipak omogućuje dobivanje najkvalitetnijih strategija održavanja pa je između ostaloga i zbog toga zakonski propisan kao obveza svim poduzećima pod vlasništvo lokalnih samouprava. Kao dopuna planu upravljanja imovinom, mogu poslužiti dokumenti kao što su bilanca vode, koja ukazuje na gubitke u sustavu te plan očuvanja zdravstvene ispravnosti vode za piće, kojim se propisuju procedure kako bi se izbjegle bilo kakve havarije po pitanju kakvoće vode u sustavu. Razvijene strategije održavanja temelje se na metodama prediktivnog održavanja, a u novije doba kod vodoopskrbnih sustava koristi se jedna nova metoda unutar te grane koja se zove proaktivno održavanje. Strategija održavanja koja slijedi proaktivni pristup, podrazumijeva stalno praćenje i kontrolu osnovnih uzroka kvarova te implementira načine kako eliminirati ili ukloniti njihov utjecaj, po čemu se razlikuje od čistog prediktivnog pristupa koji se temelji na identificiranju ranih simptoma kvarova, bez identifikacije čimbenika koji ih uzrokuju. U sklopu strategije, izrađuju se tjedni i mjesečni planovi za djelatnike održavanja koji ujedno služe i za dokumentiranje obavljenih radnji. S obzirom da su zahtjevi fizičke, hidrauličke i kvalitativne cjelovitosti sustava međusobno ovisni, djelatnici održavanja koji se brinu o prva dva zahtjeva izravno se brinu i za posljednji. Kroz interdisciplinarnu suradnju između odjela održavanja vodovoda i odjela ili institucija koji brinu o zdravstvenoj ispravnosti vode, moguće je razviti tehnike koje pomažu u održavanju vodoopskrbnih sustava. Jedan od takvih primjera suradnje može biti pronalazak modela za predviđanje rezidualnog klora u mreži. Tijekom postupka dezinfekcije vode, u Hrvatskoj se najčešće koristi klor, koji u vodi reagira

sa raznim patogenim mikroorganizmima. U toj reakciji se ne troši sva koncentracija klora, već ostaje neki određeni višak koji služi za naknadno dezinficiranje, kada voda putuje mrežom. Za tu koncentraciju je propisano da ona ne smije biti veća od 0.5 mg/L, a razna literatura nalaže da bi već 0,2 mg/L bilo dovoljno za efikasnu dezinfekciju. Prema tome je u ovom radu razvijen matematički model kojim je pomoću podataka o doziranim količinama klora, broju bakterija, ukupnih koliforma, enterokoka te *Escherichia coli*, moguće odrediti iznos rezidualnog klora. U tu svrhu korišten je algoritam genetskog programiranja preko skupine alata GPTIPS za MATLAB. Genetsko programiranje je metoda unutar grane strojnog učenja koja stohastičkim generiranjem programa i njihovom iterativnom evolucijom u nove programe, traži rješenja koja će što bolje zadovoljavati osnovnu formulaciju problema. Zahvaljujući elementu nasumičnosti, genetsko programiranje uspješno je u evoluiranju novih, neočekivanih i nelinearnih rješenja, pa tako može dobro poslužiti i za generiranje modela rezidualnog klora. U svrhu modeliranja korišteni su podaci koji se sastoje od 70 unosa prethodno spomenutih veličina, koje su se mjerile na rijeci Jadro. Već inicijalno dobiven model, može dosta dobro poslužiti za predviđanje rezidualnog klora, ali je u svrhu poboljšanja generalizacije, inicijalni skup podataka dodatno povećan Monte Carlo simulacijom. Temeljem 1000 ulaznih podataka generiranih tom simulacijom, dobiven je konačni model koji je dovoljno kvalitetan da se upotrebljava u praksi. Model je dosta kompleksan, zbog čega nije najpogodniji za brzu ručnu upotrebu, nego je namijenjen za računalnu primjenu u nekom softveru poput MATLAB-a, gdje je moguće vrlo brzo dobiti rezultate nad nekoj velikoj količini ulaznih podataka. Upotrebom tog modela moguće je ostvariti uštede, jer on omogućuje racionalizaciju upotrebe klora, kao i poboljšanje kvalitete usluge krajnjem korisniku zbog toga što je osigurano da je voda zdravstveno ispravna uz umanjene vjerojatnosti pojave boje ili okusa. Na kraju, može se zaključiti da je u održavanju vodoopskrbnih sustava primjena tehnike kao što je genetsko programiranje, itekako dobrodošla, što je i demonstrirano ovim radom. Algoritmi poput GP-a, konstantno se razvijaju paralelno s razvojem mogućnosti računala te ih je zbog njihovih naprednih mogućnosti koje u prošlosti nisu bile moguće, potrebno objeručke prihvatiti i implementirati, jer omogućuju napredne uvide kada su u pitanju dostupni podaci, ne samo u održavanju, već i u svim segmentima poslovanja i gospodarstva.

## LITERATURA

- [1] Mohan Kumar M.S., Manohar U., Pallavi M.R.M., Anjana G.R.: Urban Water Supply Management, Journal of the Indian Institute of Science, 2013, Vol 93:2, str. 295 do 317
- [2] van Zyl J.: Introduction to Operation and Maintenance of Water Distribution Systems, Edition 1, Water Research Commission, Private Bag X03, Gezina, 0031, Južnoafrička Republika, 2014., dostupna na stranici [http://www.wrc.org.za/Pages/Drought/4.2/Water%20loss/Water%20distribution%20systems\\_TT600-14.pdf](http://www.wrc.org.za/Pages/Drought/4.2/Water%20loss/Water%20distribution%20systems_TT600-14.pdf) , pristup 16.11.2018.
- [3] Skup alata za upravljanje imovinom u lokalnoj samoupravi, Hidro-inženjerski institut Sarejevo (HEIS), Njemačka organizacija za tehničku suradnju (GIZ), Otvoreni regionalni fond za modernizaciju lokalnih usluga (ORF), Mreža udruga lokalnih vlasti jugoistočne Europe (NALAS), 2014., dostupno na stranici [http://www.nalas.eu/Home/Download/Croatian-Asset\\_Management\\_Toolkit](http://www.nalas.eu/Home/Download/Croatian-Asset_Management_Toolkit) , pristup 16.11.2018.
- [4] Dragutin L., predavanja iz kolegija Održavanje, FSB, ak.god. 2015./2016.
- [5] [https://narodne-novine.nn.hr/clanci/sluzbeni/2013\\_07\\_94\\_2121.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2013_07_94_2121.html) , pristup 16.11.2018.
- [6] [https://www.asminternational.org/documents/10192/22833166/05278G\\_Sample\\_BuyNow.pdf/ddcdb82c-2c59-4ea3-9bc5-670a42f70d30](https://www.asminternational.org/documents/10192/22833166/05278G_Sample_BuyNow.pdf/ddcdb82c-2c59-4ea3-9bc5-670a42f70d30) , pristup 16.11.2018.
- [7] Rakić A., Štambuk-Giljanović N., Kuzmić J., Neke od mjera za suzbijanje patogenih mikroorganizama (*Legionella* spp.) u vodoopskrbnom sustavu, XXII. Znanstveno – stručni skup voda i javna vodoopskrba / Ujević Bošnjak, Magdalena – Zagreb: Hrvatski zavod za javno zdravstvo, 2018., str. 163 do 173
- [8] [http://www.who.int/water\\_sanitation\\_health/hygiene/om/linkingchap3.pdf](http://www.who.int/water_sanitation_health/hygiene/om/linkingchap3.pdf) , pristup 16.11.2018.
- [9] <https://ocw.tudelft.nl/wp-content/uploads/Water-treatment-schemes-1.pdf> , pristup 16.11.2018.
- [10] <https://www.dzs.hr/Hrv/system/starth.htm> , pristup 16.11.2018.
- [11] [https://narodne-novine.nn.hr/clanci/sluzbeni/2015\\_11\\_128\\_2429.html](https://narodne-novine.nn.hr/clanci/sluzbeni/2015_11_128_2429.html) , pristup 16.11.2018.
- [12] [http://www.husi.hr/download/danijela\\_lenac\\_wwd\\_2011.pdf](http://www.husi.hr/download/danijela_lenac_wwd_2011.pdf) , pristup 16.11.2018.

- [13] Ujević-Bošnjak M., Gajšak F.: Izvještaj o zdravstvenoj ispravnosti vode za ljudsku potrošnju Republici Hrvatskoj za 2017. godinu, Hrvatski zavod za javno zdravstvo, Služba za zdravstvenu ekologiju, Odjel za kontrolu zdravstvene ispravnosti voda i vodoopskrbu, Hrvatska, 2018., dostupno na stranici [https://www.hzjz.hr/wp-content/uploads/2018/06/Izvjestaj-RH-voda-za-ljudsku-potrosnju-2017\\_v1.pdf](https://www.hzjz.hr/wp-content/uploads/2018/06/Izvjestaj-RH-voda-za-ljudsku-potrosnju-2017_v1.pdf) , pristup 16.11.2018.
- [14] Schutte C.F.: Quality of Domestic Water Supplies, Volume 4: Treatment Guide, The Department of Water Affairs and Forestry, The Department of Health, Water Research Commission, Južnoafrička Republika, 2002., dostupno na stranici <http://www.dwaf.gov.za/IWQS/AssessmentGuides/TreatmentGuide/TreatmentGuide.pdf> , pristup 16.11.2018.
- [15] [http://files.dep.state.pa.us/Water/BSDW/OperatorCertification/TrainingModules/ww05\\_disinfection\\_chlorination\\_wb.pdf](http://files.dep.state.pa.us/Water/BSDW/OperatorCertification/TrainingModules/ww05_disinfection_chlorination_wb.pdf) , pristup 16.11.2018.
- [16] Dalebo Bašić B., Šnajder J.: Predavanja iz kolegija Strojno učenje, FER, ak.god. 2015./2016., dostupno na stranici <https://www.fer.unizg.hr/download/repository/SU-1-Uvod%5B1%5D.pdf> , pristup 16.11.2018.
- [17] Sreekanth J., Datta B.: Genetic programming: Efficient Modeling Tool in Hydrology and Groundwater Management, Genetic Programming, Sebastian Ventura, IntechOpen, DOI: 10.5772/52102, dostupno na stranici [http://cdn.intechopen.com/pdfs/40278/InTech-Genetic\\_programming\\_efficient\\_modeling\\_tool\\_in\\_hydrology\\_and\\_groundwater\\_management.pdf](http://cdn.intechopen.com/pdfs/40278/InTech-Genetic_programming_efficient_modeling_tool_in_hydrology_and_groundwater_management.pdf) , pristup 16.11.2018.
- [18] Tayfur G.: Modern Optimization Methods in Water Resources Planning, Engineering and Management, Water Resources Management 2017., Vol 31, Issue 10, str. 3205 do 3233, dostupno na stranici <http://openaccess.iyte.edu.tr/bitstream/handle/11147/6467/6467.pdf?sequence=1> , pristup 16.11.2018.
- [19] Banzhaf W., Nordin P., Keller E. R., Francone D. F.: Genetic Programming – An Introduction – On the Automatic Evolution of Computer Programs and Its Applications, Dpunkt.verlag i Morgan Kauffmann Publishers, Inc., San Francisco, SAD, 1998.
- [20] Rozenberg G., Bäck T., Kok J.N., Handbook of Natural Computing, Springer-Verlag Berlin Heidelberg, Njemačka, 2012.
- [21] Walker M.: Introduction to Genetic Programming, 2001., dostupno na stranici [https://www.cs.montana.edu/~bwall/cs580/introduction\\_to\\_gp.pdf](https://www.cs.montana.edu/~bwall/cs580/introduction_to_gp.pdf) , pristup 16.11.2018.



- [22] Poli R., Langdon B.W., McPhee N.F.: A Field Guide to Genetic Programming, 2008., dostupno na stranici [http://www0.cs.ucl.ac.uk/staff/ucacbbl/ftp/papers/poli08\\_fieldguide.pdf](http://www0.cs.ucl.ac.uk/staff/ucacbbl/ftp/papers/poli08_fieldguide.pdf) , pristup 16.11.2018.
- [23] Searson D.P.: Genetic Programming & Symbolic Regression for MATLAB – GPTIPS 1.0 – User Guide, 2009., dostupno na stranici <https://sites.google.com/site/gptips4matlab/file-cabinet/GPTIPSGuide1.0.pdf?attredirects=0&d=1> , pristup 16.11.2018.
- [24] <https://sites.google.com/site/gptips4matlab/file-cabinet> , pristup 16.11.2018.
- [25] Gandomi A. H., Alavi A. H., Ryan C.: Handbook of Genetic Programming Applications, Springer International Publishing, Švicarska, 2015.
- [26] Morrison G. A., Searson D. P., Willis M. J.: Using Genetic Programming to Evolve a Team of Data Classifiers, World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, 2010., Vol:4, No:12, str. 1815 do 1818, dostupno na stranici <https://waset.org/publications/2448/using-genetic-programming-to-evolve-a-team-of-data-classifiers> , pristup 16.11.2018.
- [27] Nguyen S., Mei Y., Zhang M.: Genetic Programming for Production Scheduling: A Survey with a Unified Framework, Complex & Intelligent Systems, 2017., Vol:3, izdanje 1, str. 41 do 66, dostupno na stranici <https://link.springer.com/article/10.1007/s40747-017-0036-x> , pristup 16.11.2018.
- [28] Heij C., de Boer P., Franses P. H., Kloek T., van Dijk H. K.: Econometric Methods with Applications in Business and Economics, Oxford University Press Inc., New York, SAD, 2004., str. 118 do 133
- [29] Weise T.: Metaheuristic Optimization, Hefei University, Faculty of Computer Science and Technology, Kina, dostupno na stranici [http://iao.hfuu.edu.cn/images/teaching/lectures/metaheuristic\\_optimization/14\\_genetic\\_programming.pdf](http://iao.hfuu.edu.cn/images/teaching/lectures/metaheuristic_optimization/14_genetic_programming.pdf) , pristup 16.11.2018.
- [30] Bäck T.: Genetic Programming – Evolutionary Algorithms, Universiteit Leiden, Nizozemska, dostupno na stranici <http://liacs.leidenuniv.nl/~csnaco/EA/slides/11.1.pdf?v=22-11-2017> , pristup 16.11.2018.
- [31] [http://www.who.int/water\\_sanitation\\_health/emergencies/WHO\\_TN\\_11\\_Measuring\\_chlorine\\_levels\\_in\\_water\\_supplies.pdf?ua=1](http://www.who.int/water_sanitation_health/emergencies/WHO_TN_11_Measuring_chlorine_levels_in_water_supplies.pdf?ua=1) , pristup 16.11.2018.

## PRILOZI

## I. CD-R

## II. SINTAKSA KONFIGURACIJSKE DATOTEKE

Kontrolni parametri	Vrsta vrijednosti	Opis	Zadana vrijednost
<code>gp.runcontrol.pop_size</code>	Cijeli broj $> 1$	Veličina populacije	100
<code>gp.runcontrol.num_gen</code>	Cijeli broj $> 0$	Broj generacija za vrtnju (uključujući generaciju nula)	150
<code>gp.runcontrol.verbose</code>	Cijeli broj $\geq 0$	Postavljanje da za svakih $n$ generacija pokazuje informacije	10
<code>gp.runcontrol.quiet</code>	Boolean	Ako se postavi na True, onda je onemogućeno prikazivanje izlaznih informacija	False
<code>gp.runcontrol.savefreq</code>	Cijeli broj $\geq 0$	Sprema svakih $n$ generacija informacije o stanju u datoteku <code>gptips_tmp.mat</code> .	0
<code>gp.runcontrol.timeout</code>	Cijeli broj $\geq 1$	Definira koliko sekundi će se tražiti čitavi model, bez obzira na definirani broj generacija koje treba izvrtjeti	$\infty$
<code>gp.runcontrol.runs</code>	Cijeli broj $\geq 1$	Koristi se uz <code>gp.runcontrol.timeout</code> jer omogućuje da se definira koliko puta da se provodi traženje modela ispočetka uz vremensko ograničenje preko <code>gp.runcontrol.timeout</code> naredbe – korištenjem ove naredbe model se traži više puta te se rezultati na kraju stapaju u jedan jedinstveni model – s obzirom da kod GP-a rezultati mogu varirati svaki puta kada se krene modelirati ispočetka, ova naredba je korisna za istraživanje	1

Korisnički parametri	Vrsta vrijednosti	Opis	Zadana vrijednost
<code>gp.userdata.showgraphs</code>	Boolean	Crta grafove kod korištenja <code>runtree</code> naredbe	True
<code>gp.userdata.name</code>	Tekst	Omogućuje da korisnik unese proizvoljno ime za pohranu trenutno korištenog seta podataka	/

Parametri pogodnosti	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.fitness.fitfun</b>	MATLAB funkcijska poveznica	Funkcijska poveznica na funkciju pogodnosti (cilja) koja se može proizvoljno definirati ili se upotrijebi neka od već gotovih unutar GPTIPS toolboxa pohranjena u obliku <i>m</i> daoteke	@regressmulti_fitfun
<b>gp.fitness.minimisation</b>	Boolean	Ako je True onda minimizira, ako je False onda maksimizira	True
<b>gp.fitness.terminate</b>	Boolean	Ako je True, onda prekida rad GPTIPS-a ako je postignut traženi iznos pogodnosti ili ako je prekoračena neka granična veličina	False
<b>gp.fitness.terminate_value</b>	Double	Granična vrijednost za prekid rada – nužno definirati ukoliko se koristi gp.fitness.terminate = True	$-\infty$

Selekcijski parametri	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.selection.method</b>	String	U GPTIPS1.0 je implementirana samo turnirska selekcija	'tour'
<b>gp.selection.tournament.size</b>	Cijeli broj > 0	Veličina turnira	10
<b>gp.selection.tournament.lex_pressure</b>	Boolean	Ako je True, onda se koristi jednostavna leksikografska turnirska selekcija	True
<b>gp.selection.elite_fraction</b>	$0 \leq \text{Double} < 1$	Elitizam, npr. onaj udio populacije koji se kopira izravno u iduću generaciju bez modifikacije	0.05
<b>gp.selection.tournament.p_pareto</b>	Cijeli broj $\geq 0$	Vjerojatnost da će se upotrijebiti pareto turnir za bilo koju promatrane selekciju koja se provodi u radu GP-a	0

Operandi	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.nodes.inputs.num_inp</b>	Integer > 0	Broj parametara (bez nasumičnih konstanti)	/
<b>gp.nodes.const.range</b>	(1x2) Double vektor	Raspon iz kojeg se generiraju konstante po normalnoj raspodjeli	[-10 10]
<b>gp.nodes.const.p_ERC</b>	$0 \leq \text{Double} \leq 1$	Vjerojatnost da će se upotrijebiti čvor sa konstantnom, a ne neki operand, pri dodavanju čvora stablu. ERC - Ephemeral Random Constants	0.1
<b>gp.nodes.const.p_int</b>	$0 \leq \text{Double} \leq 1$	Vjerojatnost da će nasumično generirana ERC konstanta biti cijeli broj	0
<b>gp.nodes.const.num_dec_places</b>	Cijeli broj $\geq 1$	Definira na koliko decimala se mogu generirati nasumični brojevi	4
Operatori	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.nodes.functions.name</b>	(1xF) cell array imena funkcija	Cell array korisničkih F funkcija. Svaki unos mora biti string koji sadrži ime odgovarajuće <i>m</i> datoteke	/
<b>gp.nodes.const.active</b>	(1xF) Boolean vektor	Boolean niz dužine F. određuje jesu li funkcijski čvorovi u gp.nodes.functions.name sadržani u trenutnom radu algoritma. True = sadržani, False = ne sadržani	(1xF) array koji sadrži True unose

Genetski operatori (njihova suma mora biti jednaka 1)	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.operators.mutation.p_mutate</b>	$0 \leq \text{Double} \leq 1$	Vjerojatnost mutacije GP stabla	0.14
<b>gp.operators.crossover.p_cross</b>	$0 \leq \text{Double} \leq 0$	Vjerojatnost križanja (crossover) GP stabla	0.84
<b>gp.operators.directrepro.p_direct</b>	$0 \leq \text{Double} \leq 1$	Vjerojatnost izravnog kopiranja GP stabla	0.02
<b>gp.operators.mutation.mutate_par</b>	1x6 double vektor	Definiranje učestalosti (udjela) korištenja dostupnih 6 mutacijskih operatora. Udjeli se definiraju redom za:	[0.9 0.05 0.05 0 0 0]

		(1) običnu mutaciju pod-stabala, (2) zamjenu jednog ulaznog operanda s nekim nasumičnim, (3) Gaussovu perturbaciju nasumično odabranih konstanti, (4) zamjenu nasumično odabrane konstante sa vrijednošću nula, (5) zamjenu nasumično odabrane konstante sa nasumično generiranom, (6) zamijeni nasumično odabranu konstantu sa vrijednošću 1. Ukupan zbroj unesenih udjela mora biti jednak jedinici.	
<b>gp.operators.mutation.gaussian.std_dev</b>	Cijeli broj $> 0$	Za tip mutacije pod rednim brojem (3) definira koja će se standardna devijacija koristiti	0,1

Parametri gradnje stabla	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.treedef.build_method</b>	$1 \leq$ Cijeli broj $\leq 3$	Odabir algoritma izgradnje stabla: 1=full 2=grow 3=ramped $\frac{1}{2}$ and $\frac{1}{2}$	3
<b>gp.treedef.max_depth</b>	Cijeli broj $> 1$	Maksimalna dubina stabla	4
<b>gp.treedef.max_nodes</b>	Cijeli broj $\geq 1$	Maksimalni broj čvorova po stablu	$\infty$
<b>gp.treedef.max_mutate_depth</b>	Cijeli broj $> 1$	Maksimalna dubina pod-stabala stvorenih mutacijom	4

Multi-genski parametri	Vrsta vrijednosti	Opis	Zadana vrijednost
<b>gp.genes.multigene</b>	Boolean	Ako je True, onda su omogućeni multi-genski pojedinci	True
<b>gp.genes.max_genes</b>	Cijeli broj $\geq 1$	Maksimalni broj gena pojedinca	4

<b>gp.genes.operators.p_cross_hi</b>	$0 \leq \text{Double} \leq 1$	Ako je omogućena multi-genska opcija, ovo je udio križanja na visokoj razini	0.2
<b>gp.genes.operators.hi_cross_rate</b>	$0 \leq \text{Double} \leq 1$	Vjerojatnost selekcije bilo kojeg gena tijekom križanja na visokoj razini	0,5

### III. KONFIGURACIJSKA SKRIPTA INICIJALNOG MODELA: *Vodovod\_konfig.m*

%kofiguracijska datoteka za modeliranje nad osnovnim setom podataka vodovod

```
function gp = Vodovod_konfig(gp);

gp.runcontrol.pop_size = 2000;
gp.runcontrol.num_gen = 200;
gp.fitness.fitfun = @regressmulti_fitfun;
gp.fitness.minimisation = true;
gp.fitness.terminate = false;
gp.fitness.terminate_value = 0.1;
gp.selection.tournament.size = 50;
gp.selection.tournament.lex_pressure = true;
gp.selection.elite_fraction = 0.1;
gp.selection.tournament.p_pareto = 0.3;
gp.runcontrol.runs = 1;
gp.nodes.const.range = [0,50];
gp.nodes.const.p_int = 0.2;

load ('vodovod_tren_test_osnovno');

%ucitavanje trening seta
gp.userdata.xtrain = vodovod_trening(:,1:5);

%ciljna varijabla - trening
gp.userdata.ytrain = vodovod_trening(:,6);

%ucitavanje test seta - 30 posto podataka
gp.userdata.xtest = vodovod_test(:,1:5);

%ciljna varijabla test
gp.userdata.ytest = vodovod_test(:,6);

%multigenski parametri
gp.genes.multigene = true;
gp.genes.max_genes = 4;
gp.treedef.max_depth = 5;

%definiranje operatora
gp.nodes.functions.name =
{'times', 'rdivide', 'plus', 'minus', 'power', 'exp', ...
'negexp', 'square', 'cube', 'sqrt', 'sin', 'cos', 'tan', 'cot', 'asin', 'acos', ...
```

```
'atan', 'acot', 'abs', 'log', 'log10'};
```

#### IV. SKRIPTA ZA POKRETANJE GP ALGORITMA ZA DOBIVANJE INICIJALNOG

MODEL: *Vodovod\_pokretanje\_osnovno.m*

```
clc;
disp('Pokrece Vodovod_konfig skriptu');
disp('>>gp=rungp(@Vodovod_konfig);');
disp('Pritisnuti bilo koju tipku za nastavak');
disp(' ');
pause;
gp = rungp(@Vodovod_konfig);

%prikaz pregleda kretanja RMSE-a
disp('Slijedi prikaz kretanja RMSE-a:');
disp('>>summary(gp)');
disp('Pritisnite bilo koju tipku');
disp(' ');
pause;
summary(gp, false);

%prikazati ispitivanje najboljeg dobivenog pojedinca na funkc. pog.
disp(' ');
disp('Evaluacija najboljeg pojedinca:');
disp('>>runtree(gp, 'best');');

disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
runtree(gp, 'best');

%prikaz najvažnijih pokazatelja vrtnje GP algoritma i rezultata
disp(' ');
disp('U nastavku æ se prikazati performansa i kompleksnost populacije.');
```

```
disp('models.');
```

```
disp('>>popbrowser(gp);');
```

```
disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
popbrowser(gp);

%ako je symbolic toolbox prisutan
if gp.info.toolbox.symbolic

    %pareto izvjestaj
    disp(' ');
    disp('Slijedi prikaz pareto izvjestaja');
    disp('>>paretoreport(gp);');
    disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
    paretoreport(gp);

    %gppretty
    disp(' ');
    disp('Prikaz dobivenog modela u command prozoru.');
```

```
disp('>>gppretty(gp, 'best');');
```

```
disp('Pritisnuti bilo koju tipku za nastavak');
disp(' ');
pause;
gppretty(gp, 'best');
end
disp(' ');
disp('Iscrtavanje dobivenog modela. ');
disp('>>drawtrees(gp, 'best');');
disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
drawtrees(gp, 'best');

%reports
if gp.info.toolbox.symbolic

    disp(' ');
    disp('Na kraju, slijedi prikazivanje kompletnog sveobuhvatnog
izvjestaja. ');
    disp('>>gpmodelreport(gp, 'best');');
    disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
    gpmodelreport(gp, 'best');
end
```



## V. SKRIPTA ZA PROVOĐENJE MONTE CARLO SIMULACIJE:

*MC\_simulacija\_vodovod.m*

```

%Monte Carlo simulacija za provodenje nad vodovodnim setom podataka
clear
clc

load ('osnovni_podaci');    %ucitavanje kompletnih osnovnih podataka

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%RADNI PARAMETRI

%Broj iteracija za simulaciju
br_it = 1000;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%pronazak jedinstvenih podataka u stupcu AddedCl, sortiranje
added_cl = podaci_vodovod_komplet (:,1);    %treba ucitati samo prvi stupac
added_cl_unikat = unique(added_cl);        %izvlaci unikatne unose i
pohranjuje ih
added_cl = sort (added_cl);                %sortira rastuce kompletni
ucitani stupac                                %da se poklapa s unikatnom
varijablom koja je isto po redu

%formiranje matrice nula koja je dimenzija matrice unikata
added_cl_frekv = zeros (size(added_cl_unikat));

for i = 1:length (added_cl_unikat)
    added_cl_frekv (i) = sum (added_cl == added_cl_unikat(i));
end

%ova for petlja ima brojac od 1 do najvece dimenzije matrice
%added_cl_unikat, u ovom slucaju to je broj redaka, znaci od 1 do 2. U
%matricu added_cl_frekv za svaku i-tu petlju umece sum dogadaja u kojima je
%vrijednost matrice added_cl jednaka vrijednosti matrice added_cl_unikat,
%odnosno broji koliko puta se koristi neka unikatna vrijednost u cijelom
%skupu.

added_cl_frekv = added_cl_frekv./ length(added_cl);

%sad je moguće podijeliti te učestalosti pojave unikatnih vrijednosti sa
%najvećom dimenzijom matrice added_cl, u ovom slučaju to je broj redaka.
%Time se dobiva vjerojatnost da se izvuče upravo ta unikatna vrijednost.

added_cl_kum_frekv = 100*cumsum(added_cl_frekv,1);
%potrebom cumsum dobije se kumulativna vjerojatnost temeljem matrice
%added_cl_frekv koja se sprema u added_cl_kum_frekv. Ta suma se radi u
%smjeru stupca (broj 2 oznacava smjer stupca, da je 1 onda bi oznacavao
%smjer retka).

```

```
%sada se zaokružuje na najbliži cijeli broj prethodno pomnožen sa 100
added_cl_kum_frekv = round(added_cl_kum_frekv);

%pronalažak jedinstvenih podataka u stupcu OBC, izračun frekvencija unikatata
OBC = podaci_vodovod_komplet (:,2);
OBC_unikat = unique (OBC);
OBC = sort (OBC);

OBC_frekv = zeros (size(OBC_unikat));

for i = 1:length (OBC_unikat)
    OBC_frekv (i) = sum (OBC == OBC_unikat(i));
end

OBC_frekv = OBC_frekv./ length(OBC);
OBC_kum_frekv = 100*cumsum(OBC_frekv,1);

%pronalažak jedinstvenih podataka u stupcu Cfm, sortiranje
Cfm = podaci_vodovod_komplet (:,3);
Cfm_unikat = unique (Cfm);
Cfm = sort (Cfm);

Cfm_frekv = zeros (size(Cfm_unikat));

for i = 1:length (Cfm_unikat)
    Cfm_frekv (i) = sum (Cfm == Cfm_unikat(i));
end

Cfm_frekv = Cfm_frekv./ length(Cfm);
Cfm_kum_frekv = 100*cumsum(Cfm_frekv,1);

%pronalažak jedinstvenih podataka u stupcu Entc, sortiranje
Entc = podaci_vodovod_komplet (:,4);
Entc_unikat = unique (Entc);
Entc = sort (Entc);

Entc_frekv = zeros (size(Entc_unikat));

for i = 1:length (Entc_unikat)
    Entc_frekv (i) = sum (Entc == Entc_unikat(i));
end

Entc_frekv = Entc_frekv./ length(Entc);
Entc_kum_frekv = 100*cumsum(Entc_frekv,1);

%pronalažak jedinstvenih podataka u stupcu Ec, sortiranje
Ec = podaci_vodovod_komplet (:,5);
Ec_unikat = unique (Ec);
Ec = sort (Ec);
```

```

Ec_frekv = zeros (size(Ec_unikat));

for i = 1:length (Ec_unikat)
    Ec_frekv (i) = sum (Ec == Ec_unikat(i));
end

Ec_frekv = Ec_frekv./ length(Ec);
Ec_kum_frekv = 100*cumsum(Ec_frekv,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%GENERIRANJE NASUMICNIH BROJEVA
%Napomena: rand - generira uniformno distribuirane brojeve; randn -
%generira normalno distribuirane brojeve koji mogu biti pozitivni i
%negativni; randi - generira uniformno distribuirane cijele brojeve;
%randperm - generira unikatne brojeve; s obzirom da ne trebaju unikatni i
cijeli brojevi te da nisu pozeljni negativni, koristi se rand

%treba generirati matricu nasumicnih brojeva dimenzije br_it x 1, pomnozenu
%sa 100
sl_broj = 100*rand (br_it,1);

%FORMIRANJE MATRICE PODATAKA OD SIMULACIJE I PROVJERA U KOJE RASPONE
%NASUMICNI BROJ SPADA
%formiranje matrice nula dimenzija broja iteracija x 1, kako bi se tu
spremali
%podaci o broju retka raspona iz kumulativne raspodjele, kojem nasumicno
%generirani broj pripada.

added_cl_kum_indx = zeros (br_it,1);
%provjera kojem rasponu pripada, za svaki nasumicni broj kojih ima = br_it.
%Trazi se prvi redak u kojem je sl_broj manji od added_cl_kum_frekv - tj.
%dobiva se matrica added_cl_kum_sim koja ce sadrzavati sve indekse redaka
%raspona kojima pripada broj

for n = 1:br_it
    added_cl_kum_indx (n) = find (sl_broj (n) <= added_cl_kum_frekv,1);
end

%isto za OBC
OBC_kum_indx = zeros (br_it, 1);

for n = 1:br_it
    OBC_kum_indx (n) = find (sl_broj (n) <= OBC_kum_frekv,1);
end

%za Cfm
Cfm_kum_indx = zeros (br_it, 1);

for n = 1:br_it
    Cfm_kum_indx (n) = find (sl_broj (n) <= Cfm_kum_frekv,1);
end

```

```

%za Entc
Entc_kum_indx = zeros (br_it, 1);

for n = 1:br_it
    Entc_kum_indx (n) = find (sl_broj (n) <= Entc_kum_frekv,1);
end
%za Ec
Ec_kum_indx = zeros (br_it, 1);

for n = 1:br_it
    Ec_kum_indx (n) = find (sl_broj (n) <= Ec_kum_frekv,1);
end

%SLIJEDI IZVLACENJE ODGOVARAJUCIH VRIJEDNOSTI IZ MATRICA UNIKATA - temeljem
%poznatih indeksa redaka

%za varijablu added_cl koja je oznacena sa x1
x1 = added_cl_unikat (added_cl_kum_indx,1);

%za varijablu OBC koja je oznacena sa x2
x2 = OBC_unikat (OBC_kum_indx,1);

%%za varijablu Cfm koja je oznacena sa x3
x3 = Cfm_unikat (Cfm_kum_indx,1);

%za varijablu Entc koja je oznacena sa x4
x4 = Entc_unikat (Entc_kum_indx,1);

%za varijablu Ec koja je oznacena sa x5
x5 = Ec_unikat (Ec_kum_indx,1);

%STAPANJE U SKUP PODATAKA ZA MODELIRANJE GP-om
x = [x1(:), x2(:), x3(:), x4(:), x5(:)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%UPISIVANJE MODELA
y1=0.000053920792051806830792563818288698.*tan(log(x(:,2))).^2 -
0.014397514068738528902891005145648.*sin(tan(abs(acos(x(:,3))))) -
0.017879905619042037873134276537712.*cos(x(:,3).^3) -
0.026862900082402722118368743053907.*sin(0.03108698760872673916152177021742
2.*x(:,4) + log(x(:,2)).^3) -
0.017879905619042037873134276537712.*sin(9.9678./x(:,1)) -
0.000053920792051806830792563818288698.*sin(x(:,3) - 32.1271).*(x(:,2) +
x(:,3)).^(1./2) -
0.017879905619042037873134276537712.*cos(0.00002865824181786410021695484519
4298.*x(:,3).^3) + 0.23967084198441249154321042169613;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

vodovod_prosireni_podaci = [x1(:), x2(:), x3(:), x4(:), x5(:), y1(:)];

```

```
%POHRANA
save('vodovod_prosireni_podaci');
```

## VI. KONFIGURACIJSKA SKRIPTA KONAČNOG MODELA:

### *Vodovod\_konfig\_prosireno.m*

```
%kofiguracijska datoteka za modeliranje nad kompletnim setom podataka
vodovod

function gp = vodovod_konfig_prosireno(gp);

gp.runcontrol.pop_size = 2000;
gp.runcontrol.num_gen = 200;
gp.fitness.fitfun = @regressmulti_fitfun;
gp.fitness.minimisation = true;
gp.fitness.terminate = false;
gp.fitness.terminate_value = 0.1;
gp.selection.tournament.size = 50;
gp.selection.tournament.lex_pressure = true;
gp.selection.elite_fraction = 0.1;
gp.selection.tournament.p_pareto = 0.3;
gp.runcontrol.runs = 1;
gp.nodes.const.range = [0,50];
gp.nodes.const.p_int = 0.2;

load (' vodovod_tren_test_prosireno'); %podaci dobiveni podjelom na trening
i test set iz skupa vodovod_prosireni_podaci

%ucitavanje trening seta
gp.userdata.xtrain = vodovod_trening_prosireno(:,1:5);

%ciljna varijabla - trening
gp.userdata.ytrain = vodovod_trening_prosireno(:,6);

%ucitavanje test seta - 30 posto podataka
gp.userdata.xtest = vodovod_test_prosireno(:,1:5);

%ciljna varijabla test
gp.userdata.ytest = vodovod_test_prosireno(:,6);

%multigenski parametri
gp.genes.multigene = true;
gp.genes.max_genes = 4;
gp.treedef.max_depth = 5;

%definiranje operatora
gp.nodes.functions.name =
{'times', 'rdivide', 'plus', 'minus', 'power', 'exp', ...
```

```
'negexp', 'square', 'cube', 'sqrt', 'sin', 'cos', 'tan', 'cot', 'asin', 'acos', ...
    'atan', 'acot', 'abs', 'log', 'log10'};
```

## VII. SKRIPTA ZA POKRETANJE GP ALGORITMA ZA DOBIVANJE KONAČNOG

MODEL: *Vodovod\_pokretanje\_prosireno.m*

```
clc;
disp('Pokrece vodovod_konfig_prosireno skriptu');
disp('>>gp=rungp(@Vodovod_konfig_komplet);');
disp('Pritisnuti bilo koju tipku za nastavak');
disp(' ');
pause;
gp = rungp(@vodovod_konfig_prosireno);

%prikaz pregleda kretanja RMSE-a
disp('Slijedi prikaz kretanja RMSE-a:');
disp('>>summary(gp)');
disp('Pritisnite bilo koju tipku');
disp(' ');
pause;
summary(gp, false);

%prikazati ispitivanje najboljeg dobivenog pojedinca na funkc. pog.
disp(' ');
disp('Evaluacija najboljeg pojedinca:');
disp('>>runtree(gp, 'best');');

disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
runtree(gp, 'best');

%prikaz najvažnijih pokazatelja vrtnje GP algoritma i rezultata
disp(' ');
disp('U nastavku æ se prikazati performansa i kompleksnost populacije.');
```

```
disp('models.');
```

```
disp('>>popbrowser(gp);');
```

```
disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
popbrowser(gp);

%ako je symbolic toolbox prisutan
if gp.info.toolbox.symbolic

    %pareto izvjestaj
    disp(' ');
    disp('Slijedi prikaz pareto izvjestaja');
    disp('>>paretoreport(gp);');
    disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
    paretoreport(gp);
```

```
%gppretty
disp(' ');
disp('Prikaz dobivenog modela u command prozoru. ');
disp('>>gppretty(gp, 'best')');
disp('Pritisnuti bilo koju tipku za nastavak');
disp(' ');
pause;
gppretty(gp, 'best');
end
disp(' ');
disp('Iscrtavanje dobivenog modela. ');
disp('>>drawtrees(gp, 'best');');
disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
drawtrees(gp, 'best');

%reports
if gp.info.toolbox.symbolic

    disp(' ');
    disp('Na kraju, slijedi prikazivanje kompletnog sveobuhvatnog
izvjestaja. ');
    disp('>>gpmodelreport(gp, 'best');');
    disp('Pritisnuti bilo koju tipku za nastavak');disp(' ');pause;
    gpmodelreport(gp, 'best');
end
```