

Optimiranje strukture neuronske mreže s radijalnim baznim funkcijama primjenom algoritma roja čestica

Vitasović, Matej

Master's thesis / Diplomski rad

2016

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:235:983795>

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

Download date / Datum preuzimanja: **2024-11-30**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Matej Vitasović

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Dr. sc. Danko Brezak, izv. prof.

Student:

Matej Vitasović

Zagreb, 2016.

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

Zahvaljujem se dr. sc. Danku Brezaku na kvalitetnim, stručnim savjetima i ukazanom povjerenju prilikom izrade ovog rada.

Zahvaljujem se svojoj obitelji na bezuvjetnoj podršci i pomoći.

Matej Vitasović



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



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

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

DIPLOMSKI ZADATAK

Student: **Matej Vitasović**

Mat. br.: 0035183664

Naslov rada na hrvatskom jeziku: **Optimiranje strukture neuronske mreže s radijalnim baznim funkcijama primjenom algoritma roja čestica**

Naslov rada na engleskom jeziku: **Particle swarm optimisation of radial basis function neural networks**

Opis zadatka:

Neuronska mreža s radijalnim baznim funkcijama (RBF) spada u skupinu troslojnih, statičkih neuronskih mreža, čiji se težinski faktori u fazi učenja adaptiraju u jednom koraku. Postupak učenja se, osim u određivanja težinskih faktora, sastoji i u definiranju strukture skrivenog sloja mreže, tj. u određivanju broja neurona skrivenog sloja i parametara njihovih aktivacijskih (radijalnih baznih) funkcija. Za aktivacijske je funkcije u fazi učenja potrebno odrediti njihovu poziciju (centar) i širinu. Klasični postupak određivanja broja neurona skrivenog sloja te navedenih parametara aktivacijskih funkcija zasniva se na primjeni uzoraka za učenje i neke od metoda njihova grupiranja. Tim se postupkom nastoji odrediti struktura mreže koja zadovoljava tražene generalizacijske karakteristike, pri čemu nije poznato je li dobivena struktura ujedno i najbolja za učeni problem.

Stoga je u ovom je radu primijenjen pristup koji podrazumijeva primjenu algoritma rojeva čestica u određivanju parametara aktivacijskih funkcija neurona skrivenog sloja, a koji osim uzoraka za učenje uzima u obzir i odzive mreže za vrijeme učenja, te osigurava optimalno rješenje za dane početne uvjete.

U radu je potrebno:

1. Izraditi programsku podršku za RBF neuronsku mrežu učenu klasičnom metodom učenja u jednom koraku.
2. Proširiti model RBF mreže s osnovnim PSO algoritmom učenja i odabranim modifikacijama osnovnog algoritma.
3. Generirati odzive mreže za sve varijante algoritama učenja primjenom nekoliko usporednih testova.
4. Usporediti dobivene rezultate i izvesti zaključke rada.

Zadatak zadan:

5. svibnja 2016.


Rok predaje rada:

7. srpnja 2016.

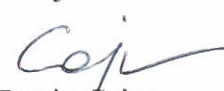
Predviđeni datum obrane:

13., 14. i 15. srpnja 2016.

Zadatak zadao:


Doc. dr. sc. Danko Brezak

Predsjednik Povjerenstva:


Prof. dr. sc. Franjo Cajner

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA.....	IV
POPIS OZNAKA	V
SAŽETAK.....	VII
SUMMARY	VIII
1. UVOD.....	1
1.1. Umjetne neuronske mreže.....	1
1.2. Optimizacija rojem čestica.....	2
1.3. Testiranje mreže.....	2
2. RBF MREŽA.....	3
2.1. Općenito o neuronskim mrežama	3
2.2. Struktura neuronske mreže.....	3
2.3. Model RBF mreže.....	7
2.3.1. Struktura RBF mreže	7
2.3.2. Matematički model RBF mreže	9
2.3.3. Učenje i testiranje mreže.....	11
2.3.4. Određivanje broja i centara neurona skrivenog sloja.....	13
3. OPTIMIZACIJA ROJEM ČESTICA	14
3.1. Uvod u optimizaciju rojem čestica.....	14
3.2. Osnovni algoritam optimizacije rojem čestica.....	15
3.3. Unificirani algoritam optimizacije rojem čestica.....	17
3.4. Algoritam optimizacije rojem čestica s osiguranom konvergencijom.....	18
3.5. Metoda suprotnih vrijednosti uz korištenje Cauchyevog mutacijskog operatora	20
4. PROBLEMI ZA UČENJE I TESTIRANJE	22
4.1. Problem 1 – Krug u kvadratu	22
4.2. Problem 2 – <i>Iris</i> set podataka.....	24
4.3. Problem 3 – Određivanje stupnja istrošenosti alata kod bušenja kompozita	26
4.3.1. Opis problema 3	26
4.3.2. Detalji mjerenja.....	27
5. REZULTATI	32
5.1. Postavke	32
5.2. Problem 1	34
5.2.1. Vrijednosti parametara algoritama za problem 1	34
5.2.2. Tablični prikaz rezultata za problem 1	35
5.2.3. Grafički prikaz učenja algoritama uz $\beta=0.07$	36
5.2.4. Grafički prikaz učenja algoritama uz $\beta=0.2$	38
5.3. Problem 2	40
5.3.1. Vrijednosti parametara algoritama za problem 2.....	40
5.3.2. Tablični prikaz rezultata za problem 2.....	41

5.3.3. Grafički prikaz učenja algoritama uz $\beta=0.2$	42
5.3.4. Grafički prikaz učenja algoritama uz $\beta=0.5$	44
5.4. Problem 3	46
5.4.1. Vrijednosti parametara algoritama za problem 3	46
5.4.2. Tablični prikaz rezultata za problem 3.....	47
5.4.3. Grafički prikaz učenja algoritama uz $\beta=0.05$	48
5.4.4. Grafički prikaz učenja algoritama uz $\beta=0.1$	50
5.5. Analiza rezultata	52
6. ZAKLJUČAK.....	54
LITERATURA.....	55

POPIS SLIKA

Slika 1.	Skica biološkog neurona [3].....	5
Slika 2.	Model umjetnog statičkog neurona [3]	6
Slika 3.	Model RBF neuronske mreže.....	8
Slika 4.	Gaussova aktivacijska funkcija	9
Slika 5.	Uzorci za učenje problema 1	22
Slika 6.	Uzorci za testiranje problema 1	23
Slika 7.	Uzorci za učenje problema 2	24
Slika 8.	Uzorci za testiranje problema 2.....	25
Slika 9.	Sinkroni servo motor [11]	27
Slika 10.	Senzor sila Kistler 9257B [11]	28
Slika 11.	Radni prostor stroja; 1. Posmični prigon X osi, 2. Posmični prigon Y osi, 3. Posmični prigon Z osi, 4. Glavno vreteno, 5. Stezna naprava, 6. Senzor sila, 7. Senzor vibracija, 8. Industrijska kamera s rasvjetom, 9. Ispitni uzorak, 10. Ispitno svrdlo [11]	29
Slika 12.	Kamera DMG41AF02 i telecentrični objektiv TC2309	30
Slika 13.	Niskopropusni Butterworthov filter	31
Slika 14.	Grafički prikaz učenja PSO algoritma za problem 1 i $\beta = 0.07$	36
Slika 15.	Grafički prikaz učenja UPSO algoritma za problem 1 i $\beta = 0.07$	36
Slika 16.	Grafički prikaz učenja GCPSO algoritma za problem 1 i $\beta = 0.07$	37
Slika 17.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 1 i $\beta = 0.07$	37
Slika 18.	Grafički prikaz učenja PSO algoritma za problem 1 i $\beta = 0.2$	38
Slika 19.	Grafički prikaz učenja UPSO algoritma za problem 1 i $\beta = 0.2$	38
Slika 20.	Grafički prikaz učenja GCPSO algoritma za problem 1 i $\beta = 0.2$	39
Slika 21.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 1 i $\beta = 0.2$	39
Slika 22.	Grafički prikaz učenja PSO algoritma za problem 2 i $\beta = 0.2$	42
Slika 23.	Grafički prikaz učenja UPSO algoritma za problem 2 i $\beta = 0.2$	42
Slika 24.	Grafički prikaz učenja GCPSO algoritma za problem 2 i $\beta = 0.2$	43
Slika 25.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 2 i $\beta = 0.2$	43
Slika 26.	Grafički prikaz učenja PSO algoritma za problem 2 i $\beta = 0.5$	44
Slika 27.	Grafički prikaz učenja UPSO algoritma za problem 2 i $\beta = 0.5$	44
Slika 28.	Grafički prikaz učenja GCPSO algoritma za problem 2 i $\beta = 0.5$	45
Slika 29.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 2 i $\beta = 0.5$	45
Slika 30.	Grafički prikaz učenja PSO algoritma za problem 3 i $\beta = 0.05$	48
Slika 31.	Grafički prikaz učenja UPSO algoritma za problem 3 i $\beta = 0.05$	48
Slika 32.	Grafički prikaz učenja GCPSO algoritma za problem 3 i $\beta = 0.05$	49
Slika 33.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 3 i $\beta = 0.05$	49
Slika 34.	Grafički prikaz učenja PSO algoritma za problem 3 i $\beta = 0.1$	50
Slika 35.	Grafički prikaz učenja UPSO algoritma za problem 3 i $\beta = 0.1$	50
Slika 36.	Grafički prikaz učenja GCPSO algoritma za problem 3 i $\beta = 0.1$	51
Slika 37.	Grafički prikaz učenja OPSO Cauchy algoritma za problem 3 i $\beta = 0.1$	51

POPIS TABLICA

Tablica 1. Tehničke karakteristike motora [11]	27
Tablica 2. Tehničke karakteristike mjernog sustava sila [11]	28
Tablica 3. Tehničke karakteristike vizijskog sustava [11]	30
Tablica 4. Vrijednosti parametara algoritama za problem 1	34
Tablica 5. Rezultati mreža za problem 1	35
Tablica 6. Vrijednosti parametara algoritama za problem 2	40
Tablica 7. Rezultati mreža za problem 2	41
Tablica 8. Vrijednosti parametara algoritama za problem 3	46
Tablica 9. Rezultati mreža za problem 3	47

POPIS OZNAKA

Oznaka	Jedinica	Opis
a	--	Koeficijent inercije brzine čestice
\mathbf{a}^P	--	Vektor minimalnih vrijednosti pozicija
\mathbf{b}^P	--	Vektor maksimalnih vrijednosti pozicija
\mathbf{C}	--	Matrica težinskih faktora skrivenog sloja
\mathbf{c}_j	--	Vektor težinskih faktora j -tog neurona skrivenog sloja
c_1, c_2	--	Koeficijenti utjecaja individualne i društvene komponente čestice
e	--	Eulerov broj, baza prirodnog algoritma
$F_t(\cdot)$	--	Kumulativna funkcija distribucije vjerojatnosti
$f(\cdot)$	--	Cauchyeva funkcija gustoće vjerojatnosti
f_c	--	Prag neuspjeha pronalaska boljeg rješenja
\mathbf{H}	--	Matrica izlaza neurona skrivenog sloja
\mathbf{H}^+	--	Moore-Penroseov pseudoinverz matrice \mathbf{H}
H_{ij}	--	Element matrice izlaza – izlaz iz j -tog neurona skrivenog sloja vezan za i -ti ulazni uzorak u mrežu
$h(\cdot)$	--	Aktivacijska funkcija neurona skrivenog sloja
K	--	Broj neurona skrivenog sloja
k	--	k -ti korak iteracije PSO algoritma
L	--	Broj neurona ulaznog sloja
M	--	Broj neurona izlaznog sloja
N	--	Broj čestica
\mathbf{O}	--	Matrica izlaza mreže
O_w	--	w -ti izlaz neurona izlaznog sloja
$\mathbf{p}_{glob.}$	--	Najbolja pozicija roja
$\mathbf{p}_{glob.,mut.}$	--	Mutirana najbolja pozicija roja
$\mathbf{p}_{indiv.}$	--	Najbolja pozicija koju je postigla čestica
$\mathbf{p}_{lok.}$	--	Najbolja pozicija susjedstva
p_0	--	Suprotna vjerojatnost
R^L	--	L -dimenzionalni prostor ulaza
R^M	--	M -dimenzionalni prostor izlaza
r_{ij}	--	Mahalanobisova udaljenost i -tog uzorka od j -tog neurona skrivenog sloja
r_1, r_2	--	Slučajne vrijednosti u intervalu (0,1)
s_c	--	Prag uspjeha pronalaska boljeg rješenja
\mathbf{t}_j	--	Vektor centra j -tog neurona skrivenog sloja
u	--	Unifikacijski faktor
\mathbf{v}_j	--	Brzina j -te čestice

$W(k)$	--	Srednja vrijednost brzina čestica u k -toj iteraciji
\mathbf{x}_j	--	Pozicija j -te čestice
$\mathbf{x}_{supr.,j}$	--	Suprotna pozicija j -te čestice
β	--	Parametar grupiranja
ρ	--	Parametar širine područja pretrage
Σ_j	--	Kovarijantna matrica vezana uz j -ti neuron skrivenog sloja
σ_g^2	--	g -ti element na glavnoj dijagonali kovarijantne matrice

SAŽETAK

U ovom radu je izrađena programska podrška za neuronsku mrežu s radijalnim baznim funkcijama (**RBF mrežu**) učenu klasičnom metodom učenja u jednom koraku. Model je proširen osnovnim algoritmom optimizacije rojem čestica (**PSO**), te s tri njegove modifikacije: unifikacijskim algoritmom optimizacije rojem čestica (**UPSO**), algoritmom optimizacije rojem čestica s osiguranom konvergencijom (**GCPSO**), te metodom suprotnih vrijednosti uz korištenje Cauchyevog mutacijskog operatora (**OPSO Cauchy**). Algoritmi su primijenjeni u optimizaciji vrijednosti elemenata glavne dijagonale kovarijantnih matrica vezanih uz neurone skrivenog sloja RBF mreže. Usporedba svih varijanti algoritama izvršena je testiranjem mreže sa tri klasifikacijska problema: „Krug u kvadratu“, *Iris* setu podataka, te na određivanju stupnja istrošenosti alata kod bušenja kompozita. Funkcija cilja korištena kod optimizacije bilo je postotak uspješno klasificiranih uzoraka iz seta za testiranje.

Ključne riječi: strojno učenje, neuronska mreža, radijalna bazna funkcija, optimizacija rojem čestica, klasifikacija

SUMMARY

In this study, a programming support was built for a radial basis function neural network (**RBF network**) using a classical one-step learning method. The model was expanded with the basic particle swarm optimisation algorithm (**PSO**), and with three of its modifications: Unified Particle Swarm Optimization (**UPSO**), Guaranteed Convergence Particle Swarm Optimization (**GCPSO**), and Opposition-Based Particle Swarm Optimisation with Cauchy Mutation (**OPSO Cauchy**). The algorithms were used to optimise the values of the elements on the main diagonal of covariance matrices associated with the hidden-layer neurons of the RBF network. A comparison of all algorithms was made by testing the network on three classification problems: „Circle-in-the-Square“, *Iris* dataset, and determining the tool wear in drilling composite materials. The objective function used in optimisation methods was the percentage of successfully classified test data samples.

Key words: machine learning, neural networks, radial basis function, particle swarm optimisation, classification

1. UVOD

1.1. Umjetne neuronske mreže

Umjetne neuronske mreže su od svojih začetaka krajem prve polovice 20. stoljeća, pa sve do danas, dobile status jedne od najpoznatijih metoda strojnog učenja, te su kao takve primjenjivane u rješavanju raznolikih problema. Od klasifikacije uzoraka, prepoznavanja rukom pisanih znakova i govora, aproksimacije funkcija i igranje igara poput šaha ili dame, pa sve do vođenja robota, predikcije unosa teksta i evolucije kaotičnih sustava poput burze ili klime, neuronske mreže su brzo širile svoje područje primjene, pritom ostvarujući zavidne rezultate koji su služili kao inspiracija za daljnji napredak [3].

Prve mreže bile su jednoslojne i sastavljene od takozvanih perceptrona – binarnih neurona čiji izlaz može biti 1 ili 0. Unatoč prvim uspjesima u rješavanju klasifikacijskih problema, takve mreže nisu bile sposobne riješiti linearno neseparabilne probleme, poput ekskluzivno ILI logičkog problema. Već se tada znalo da se uvođenjem dodatnih skrivenih slojeva s nelinearnom aktivacijskom funkcijom može riješiti linearno neseparabilan problem, no metoda učenja parametara takve mreže nije bila poznata. Otkriće danas najpoznatijeg i najrasprostranjenijeg algoritma učenja povratnim rasprostiranjem pogreške predstavljala je velik skok u razvoju neuronskih mreža: njime se po prvi put moglo učiti neuronske mreže sa skrivenim slojevima koje su bile sposobne riješiti linearno neseparabilne probleme [3].

Krajem 80-ih godina, prvi put su spomenute neuronske mreže s radijalnim baznim funkcijama u radovima Broomheada i Lowea [1] [2]. Takve mreže ne koriste iterativnu gradijentnu metodu poput algoritma učenja povratnim rasprostiranjem pogreške, već učenje parametara obavljaju u jednom koraku – pomoću inverza matrice izlaza neurona skrivenog sloja računaju se težinski koeficijenti koji povezuju skriveni sloj sa izlaznim. Pokazano je da se radijalne bazne funkcije mogu uspješno primijeniti kod interpolacije viševarijabilnih sustava [3]. U ovom se radu koristila upravo takva neuronska mreža na problemima klasifikacije uzoraka.

1.2. Optimizacija rojem čestica

Optimizacija rojem čestica (*eng.* Particle Swarm Optimization - PSO), je metoda učenja optimalnih vrijednosti parametara funkcije inspirirana zajedničkim djelovanjem nekih životinja poput riba ili ptica, te nekih kukaca poput pčela u potrazi za hranom. Metodu su prvi put spomenuli Kennedy i Eberhart u svom radu iz 1995. [6]. PSO je pokazao svoju primjenjivost u mnogim problemima, poglavito zato što zahtijeva malo, ili nimalo informacija o funkciji koja se optimizira. Ukratko, metoda započinje generiranjem željenog broja čestica, gdje svaka čestica predstavlja potencijalno rješenje u prostoru učenja. Svako se rješenje ocijeni zadanom funkcijom cilja, nakon čega se pozicije čestica ažuriraju na temelju dva glavna čimbenika. Prvi predstavlja istraživačku komponentu, i omogućava česticama da posjete nova područja prostora učenja. Drugi je društvena komponenta, kojom se propagira informacija o prethodno pronađenom najboljem rješenju kroz cijeli roj, čime se pospješuje mogućnost konvergencije rješenja i suradnje. Ažuriranja pozicija čestica se ponavljaju po spomenutom principu dok se ne zadovolji uvjet prekida, koji je često kombinacija maksimalnog broja iteracija i zadovoljavajuće vrijednosti funkcije cilja.

1.3. Testiranje mreže

U ovom je radu klasična metoda učenja parametara neuronske mreže s radijalnim baznim funkcijama u jednom koraku proširena sa nekoliko različitih verzija PSO algoritma. Spomenuta velika razina neovisnosti PSO algoritma o problemu koji se optimizira inspiracija je za primjenu PSO-a u učenju neuronskih mreža. U ovom radu se usporedbom s klasičnom metodom učenja pokušalo zaključiti može li PSO algoritam konkurirati uobičajenim metodama učenja mreže.

Performanse algoritma se testiralo na tri klasifikacijska problema. Prvi problem je „Krug u kvadratu“ u kojem mreža mora naučiti ispravno odrediti pripada li uzorak unutar kružnice ili van nje na temelju koordinata uzoraka. Drugi problem uključuje ispravno klasificiranje tri vrste cvijeta irisa na temelju četiri mjerene značajke: duljine i širine latice i čašičnog listića. Treći je problem tehničke prirode i svodi se na ispravno određivanje stupnja istrošenosti alata za bušenje kompozitnog materijala na temelju posmaka, brzine vrtnje i mjerenih signala struje.

2. RBF MREŽA

2.1. Općenito o neuronskim mrežama

Izvorna ideja za umjetnu neuronsku mrežu produkt je težnji da se modelira biofiziologija mozga čovjeka, s ciljem razumijevanja i objašnjenja kako isti funkcionira. Model bi trebao biti sposoban imitirati funkciju ljudskog mozga, odnosno trebao bi biti sposoban obrađivati informacije na isti način na koji to ljudski mozak radi. Danas se neuronske mreže koriste u rješavanju široke lepeze problema prepoznavanja uzoraka, te u problemima predikcije, aproksimacije i interpolacije [3]. Njihova je struktura ovisna o vrsti problema i namjeni, pa ih tako možemo kategorizirati na više načina – time govorimo o vrsti neurona koji čini mrežu, aktivacijskoj funkciji neurona, broju slojeva mreže, načinu učenja mreže, te ovisnosti parametara mreže o vremenu.

2.2. Struktura neuronske mreže

Umjetna neuronska mreža obrađena u ovom radu spada u kategoriju statičkih mreža, što znači da niti jedan parametar mreže ne ovisi o vremenu. Dakle, ne uzimaju se u obzir efekti kašnjenja signala, kao i to da se na ulaze neurona ne dovode izlazi tog istog neurona, ili izlazi bilo kojeg drugog neurona. U posebnom se slučaju prošli izlazi mreže mogu spojiti na ulaze, što se koristi u problemima predikcije, te kod regulacije kada konstantna vrijednost skokovite funkcije nije adekvatan ulazni podatak (zbog nepromjenjivosti signala), pa se mreži na ulaze daju i prošli izlazi.

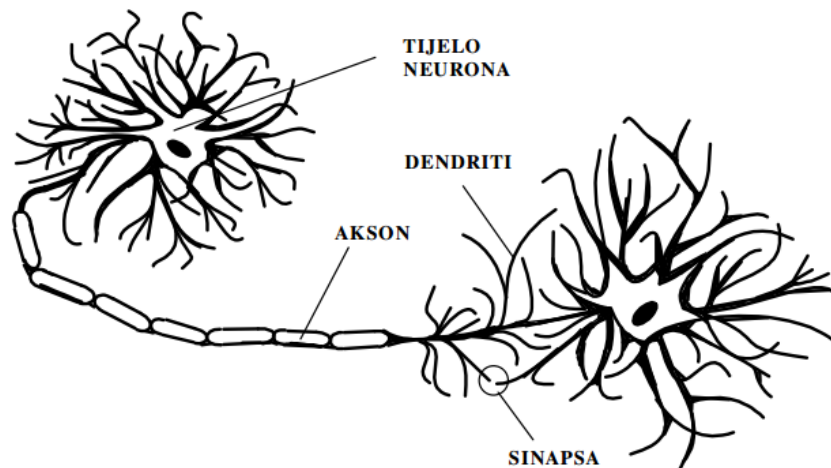
Nadalje, ova mreža je i unaprijedna, sa svojstvom da su izlazni signali neurona n -tog sloja spojeni isključivo na ulaze $(n+1)$ -tog sloja, te mogu utjecati samo na taj sloj.

Mreža se, konačno, može opisati i kao višeslojna, s obzirom da posjeduje tri distinktivna sloja neurona – ulazni, skriveni, te izlazni sloj. Važno je napomenuti da se ulazni sloj često ne smatra slojem, te se niti ne broji u definiranju slojevitosti mreže, zbog činjenice da neuroni ulaznog sloja ne posjeduju aktivacijsku funkciju i druge pripadajuće funkcije (kod perceptrona funkcija sume, kod RBF neuronskih mreža funkcija udaljenosti), što su temeljne karakteristike umjetnih neurona. Time, iako mreža posjeduje spomenuta tri sloja, ona se po definiciji naziva *dvoslojnom*.

Zadatak neuronske mreže je da promjenom svojih parametara što bolje aproksimira traženi izlaz sa svojim izlazom. Postupak promjene parametara mreže naziva se učenjem mreže i može se obaviti na više različitih načina. Perceptronske mreže se najčešće uče iterativnim metodama, od kojih je najpoznatiji algoritam povratnog prostiranja pogreške (eng. *Error Backpropagation Algorithm*), koji koristi gradijent pogreške, na način da se parametri koji su podvrgnuti učenju mreže mijenjaju prema derivaciji greške izlaza u ovisnosti o tom parametru. Određenim brojem iteracija izlaz mreže se može približiti traženom izlazu s dovoljnom, odnosno traženom točnošću. S obzirom da se algoritmom pokušava aproksimirati traženi izlaz, posljedica je da stvarni izlaz mreže nikada ne može dati traženi sa stopostotnom točnošću [3].

S druge strane, RBF mreže koriste drugačiji pristup rješavanju problema. Ovdje se učenje parametara neuronske mreže može promatrati kao problem aproksimacije ili interpolacije hiperravnine kroz zadani skup točaka za učenje. Nakon što je učenje završeno, računanje odziva neuronske mreže u stvari je računanje vrijednosti aproksimirane hiperravnine za zadani ulazni vektor. Ovakav pristup učenju neuronskih mreža blisko je vezan s klasičnim matematičkim metodama interpolacije i aproksimacije kontinuiranih viševarijabilnih funkcija. U ovom se radu koristio pristup interpolacije, i to na problemima klasifikacije uzoraka. Uzorci za učenje se dijele, prije svega u klasifikacijske skupine, i nakon toga u manje grupe. Tim se grupama računaju centri, koji se postavljaju kao centri radijalnih baznih funkcija neurona skrivenog sloja. S obzirom da izlaz radijalne bazne funkcije ovisi o udaljenosti njenog centra, može se reći da mreža hiperravninu provlači kroz točke koji predstavljaju centre. Matematički pristup izgradnji takve mreže detaljnije je razrađen niže u radu.

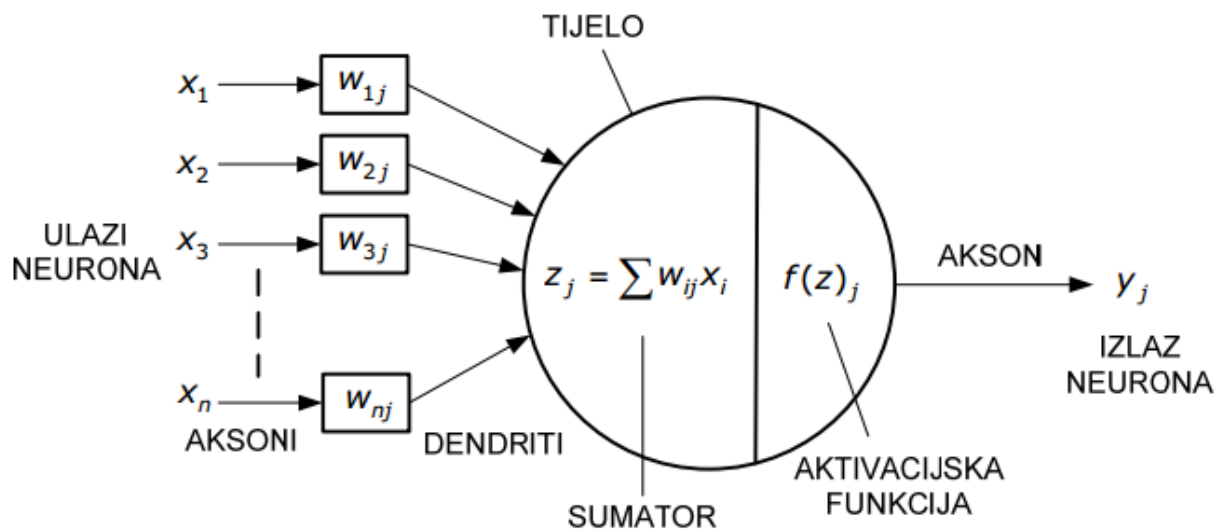
Osnovna građevna jedinica svake neuronske mreže je umjetni neuron. Kod perceptronskih mreža on je pojednostavljeni matematički model biološkog neurona [Slika 1].



Slika 1. Skica biološkog neurona [3]

Pojednostavljena struktura i funkcija biološkog neurona su sljedeći: dendriti su izdanci koji predstavljaju ulaze u neuron – njih može biti proizvoljno mnogo, a na njima su spojeni izlazi drugih neurona u okolini čije elektrokemijske signale dendriti prenose do tijela neurona. Svaki neuron ima prag osjetljivosti, te ukoliko je suma intenziteta signala veća od praga, tijelo neurona ispali svoj elektrokemijski signal. On se dalje prenosi aksonom neurona, koji predstavlja izlaz (uglavnom postoji samo jedan, no sinaptičkim vezama može biti povezan sa više dendrita daljnjih neurona).

Umjetnim statičkim neuronom se modelira pojednostavljena struktura biološkog neurona [Slika 2]. Dendriti su tada otežani ulazi na koje se spajaju izlazi neurona prošloga sloja, akson je izlaz koji signal prenosi do sljedećeg sloja, dok je tijelo neurona sastavljeno od dvije funkcije: prva je funkcija sume, koja zbraja sve otežane ulaze, a druga je aktivacijska funkcija kojom se simulira prag osjetljivosti neurona. Aktivacijska funkcija se odabire između *signum* funkcija, sigmoidalnih funkcija, sinusoidalnih funkcija, radijalnih baznih funkcija i funkcija sličnima njima.



Slika 2. Model umjetnog statičkog neurona [3]

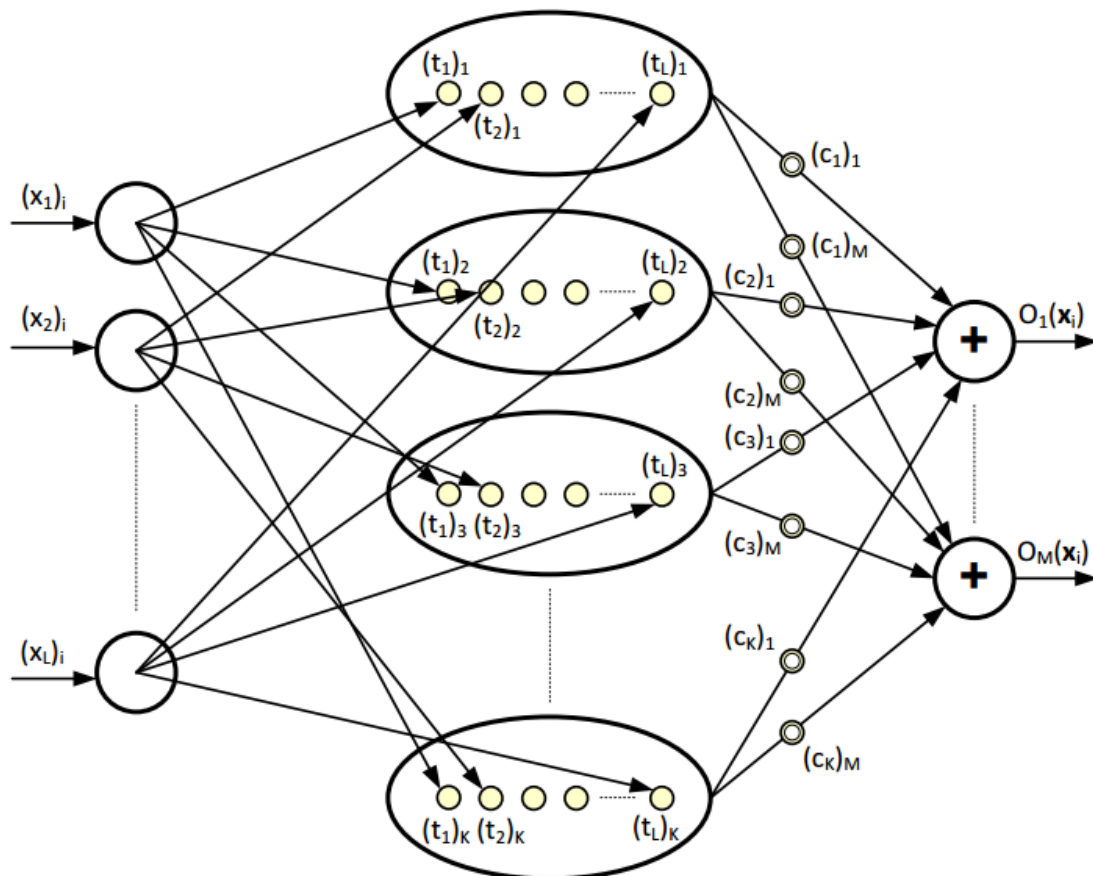
RBF neuronske mreže se, s druge strane, razlikuju u pristupu modeliranja umjetnog neurona, iako je struktura vrlo slična. Aktivacijska funkcija se odabire iz klase radijalnih baznih funkcija, po čemu je RBF mreža i dobila ime. Takve funkcije su monotonno rastuće, odnosno padajuće s obzirom na njihovo središte, a njihov izlaz ovisi o udaljenosti ulaza od središta. Ulazi u neuron se tada smatraju ulaznim vektorima, te se prvo računa udaljenost vektora ulaza od vektora centra i nakon toga izlaz aktivacijske funkcije u ovisnosti o udaljenosti. O specifičnosti odabira prikladne aktivacijske funkcije će biti nešto govora dalje u tekstu.

2.3. Model RBF mreže

2.3.1. Struktura RBF mreže

Prvobitna arhitektura neuronske mreže s radijalnim baznim funkcijama (RBF NM) predložena je 1988. godine u radu Broomheada i Lowea [1][2][3], te je od tada postala predmet istraživanja i unaprijeđivanja na polju strojnog učenja. RBF NM imaju široku upotrebu, pa se tako koriste kod aproksimacija funkcija, u predikciji, klasifikaciji i upravljanju sustavima. Dok se višeslojne perceptronske mreže uče uvriježenom iterativnom metodom povratnog prostiranja pogreške, čiji je temelj algoritam pretraživanja usponom (eng. *hill-climbing algorithm*), RBF NM se ističu osnovnim prednostima učenja u jednom koraku, te jednostavnoj adaptaciji strukture za potrebe ostvarivanja željenih generalizacijskih svojstava mreže.

Korištena varijanta RBF NM [Slika 3] prilagođena je rješavanju klasifikacijskih problema, a njezina je struktura i matematički model učenja prikazan u nastavku. Neuronske mreže s radijalnim baznim funkcijama spadaju u skupinu unaprijednih ili statičkih (eng. *feed-forward*) neuronskih mreža, a sastoje se od tri sloja. Broj neurona ulaznog sloja L određen je problemom: kod klasifikacije on će biti jednak broju značajki na temelju kojih se klasificiraju uzorci. Broj izlaznih neurona M također je određen problemom; on je jednak broju klasifikacijskih skupina u koje se svrstavaju uzorci. Broj neurona skrivenog sloja K je podložan modifikaciji i određuje ga učitelj prilikom učenja mreže. Dok neuroni ulaznog sloja služe za distribuciju ulaza na skriveni sloj, a izlazni neuroni sumiraju otežane izlaze neurona skrivenog sloja, neuroni skrivenog sloja su nešto složeniji. Svaki je neuron skrivenog sloja opisan sa L -dimenzionalnim vektorom centra (\mathbf{t}_j) i parametrima aktivacijske funkcije. Naposljetku, izlaz svakog neurona skrivenog sloja povezan je sa ulazom u svaki neuron izlaznog sloja i dodatno otežan pripadajućim težinskim faktorom \mathbf{c}_j . Time je ostvareno preslikavanje L -dimenzionalnog ulaznog vektora na M -dimenzionalni izlazni vektor ($R^L \rightarrow R^M$).



Slika 3. Model RBF neuronske mreže

Iz prethodno spomenutog opisa mreže, očito je da struktura skrivenog sloja i težinski faktori određuju kvalitetu odziva RBF NM, s obzirom da je ostala struktura prethodno određena problemom.

Aktivacijska funkcija neurona skrivenog sloja odabire se iz skupine radijalnih baznih funkcija, čiji odaziv ovisi samo o udaljenosti ulaza od njihovog središta. Prema tome, one su monotono rastuće ili padajuće s obzirom na svoje središte. Odabir takve funkcije uvjetovan je Michellijevim teoremom [3], prema kojem se funkcijom $h(\cdot)$, kontinuiranom u intervalu $[0, \infty)$ i pozitivnom u intervalu $(0, \infty)$, sa u potpunosti monotonom prvom derivacijom na području definicije funkcije, ali ne i konstantnom na intervalu $(0, \infty)$ može osigurati pozitivna determinanta matrice \mathbf{H} . Time se osigurava i invertibilnost matrice, što je nužan uvjet za provođenje postupka učenja s obzirom da se uz pomoć inverza računaju težinski faktori skrivenog sloja. U ovom radu će se koristiti zvonolika Gaussova funkcija koja zadovoljava spomenute uvjete [Slika 4].

2.3.2. Matematički model RBF mreže

Veza između neurona izlaznog i skrivenog sloja definirana je sljedećim izrazom:

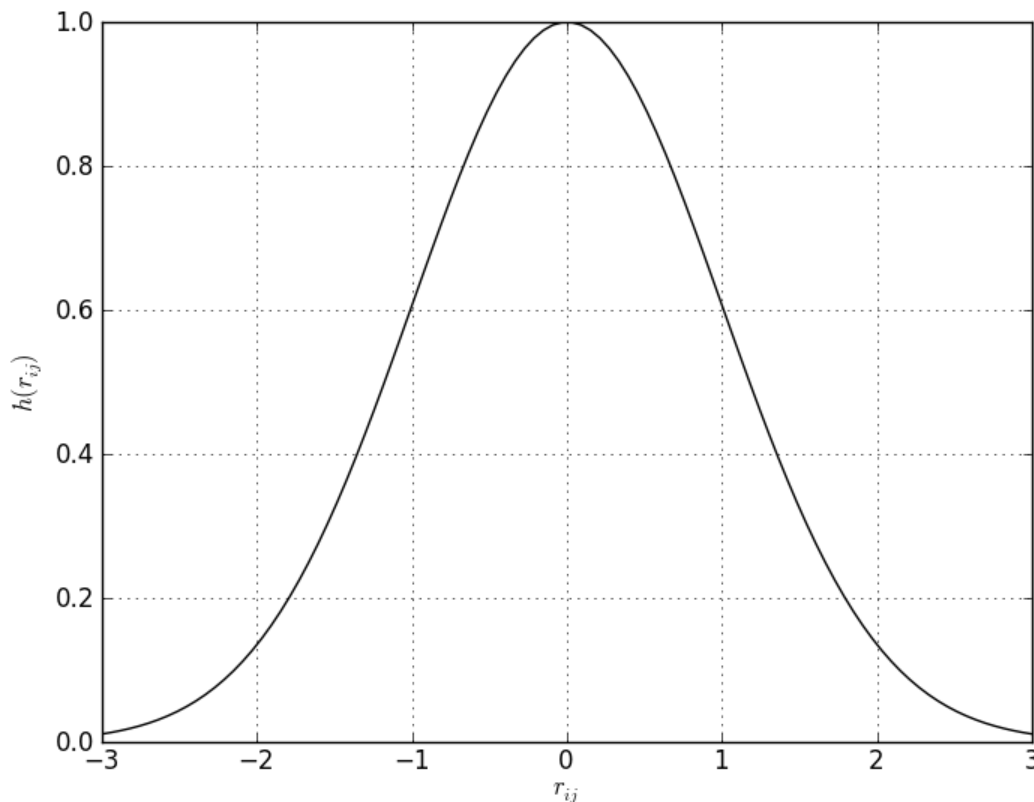
$$O_w(\mathbf{x}_i) = \sum_{j=1}^K h(r_{ij})c_{jw}, \quad w = 1, \dots, M, i = 1, \dots, L, \quad (1)$$

gdje je O_w element w -tog izlaznog neurona koji ovisi o elementima i -tog ulaznog vektora \mathbf{x}_i , $h(\cdot)$ je aktivacijska funkcija, r_{ij} Mahalanobisova udaljenost ulaznog vektora \mathbf{x}_i od vektora \mathbf{t}_j , c_{jw} je težinski faktor koji povezuje j -ti neuron skrivenog sloja sa w -tim neuronom izlaznog sloja, K broj neurona sakrivenog sloja, M broj izlaznih, a L broj ulaznih neurona.

Kao što je ranije spomenuto, korištena aktivacijska funkcija je Gaussova funkcija oblika:

$$h(r_{ij}) = H_{ij} = e^{-\frac{1}{2}r_{ij}^2}, \quad (2)$$

gdje je H_{ij} element matrice vrijednosti radijalnih baznih funkcija \mathbf{H} .



Slika 4. Gaussova aktivacijska funkcija

Udaljenost ulaznih vektora od centara neurona skrivenog sloja određena je Mahalanobisovom udaljenošću. U ovom slučaju to je udaljenost ulaznog vektora od vektora centra \mathbf{t}_j . Kvadratna vrijednost Mahalanobisove udaljenosti računa se prema izrazu:

$$r_{ij}^2 = (\mathbf{x}_i - \mathbf{t}_j)^T \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_i - \mathbf{t}_j), \quad (3)$$

gdje je $\boldsymbol{\Sigma}_j$ kovarijantna matrica definirana na temelju grupe uzoraka korištenih u izračunu centra \mathbf{t}_j . To znači da je svaki neuron skrivenog sloja, uz to što je opisan centrom, opisan i kovarijantnom matricom. Ona je dijagonalna matrica dimenzije $L \times L$ oblika:

$$\boldsymbol{\Sigma}_j = \begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \sigma_L^2 \end{bmatrix}, \quad (4)$$

gdje su $\sigma_g (g = 1, \dots, L)$ vrijednosti fiktivnog ili stvarnog maksimalno udaljenog uzorka razmatrane grupe od centra te grupe po svim dimenzijama. One se određuju izrazom:

$$\sigma_g | _j = \max\{\|\mathbf{z}_{pg} - \mathbf{t}_g\|, p = 1, \dots, n\} | _j, g = 1, \dots, L, j = 1, \dots, K, \quad (5)$$

gdje je \mathbf{z}_{pg} g -ta komponenta p -tog uzorka j -te grupe u koju je svrstano n uzoraka, a \mathbf{t}_g je g -ta komponenta vektora centra j -te grupe, odnosno j -tog neurona skrivenog sloja. U slučaju da se ovakvim izračunom dobije nula, potrebno je tu nulu zamijeniti nekakvom malenom, proizvoljnom vrijednošću, kako bi se osigurala regularnost kovarijantne matrice (i time invertibilnost).

Izraz (1) može se zapisati u matricnoj formi

$$\mathbf{O} = \mathbf{H}\mathbf{C}, \quad (6)$$

gdje je \mathbf{O} matrica izlaza dimenzije $(L \times M)$, \mathbf{H} je matrica vrijednosti radijalnih baznih funkcija ili matrica učenja dimenzije $(L \times K)$, a \mathbf{C} je matrica težinskih faktora dimenzije $(K \times M)$. Kako bi se izračunala matrica \mathbf{C} , izraz (6) se može preurediti na sljedeći način:

$$\mathbf{C} = \mathbf{H}^{-1}\mathbf{O}, \quad (7)$$

S obzirom da je u praksi češći slučaj taj da je matrica \mathbf{H} pravokutnog oblika (da ulaznih vektora za učenje mreže ima više nego neurona skrivenog sloja), inverz se mora zamijeniti pseudoinverzom.

$$\mathbf{C} = \mathbf{H}^+ \mathbf{O}, \quad (8)$$

gdje je \mathbf{H}^+ Moore-Penroseov pseudoinverz koji se računa prema izrazu

$$\mathbf{H}^+ = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T, \quad (9)$$

Kod računalnih implementacija predlaže se korištenje numerički stabilnijih algoritama za izračun pseudoinverza, npr. gotovih i optimiziranih rutina iz LAPACK –a [4].

2.3.3. Učenje i testiranje mreže

Zadatak svake neuronske mreže je kvalitetno preslikavanje ulaznih na željene izlazne vrijednosti. Kod problema klasifikacije, to znači da se što više uzoraka mora ispravno svrstati u pripadajuće klase, kako u fazi učenja, tako i u fazi testiranja i eventualne implementacije mreže. Kako bi se taj cilj ostvario, potrebno je osigurati adekvatan broj uzoraka koji kvalitetno opisuju prostor učenja u svrhu postizanja dobrih generalizacijskih svojstava mreže. Premalen broj uzoraka uglavnom uzrokuje loš odaziv mreže, kao što ni malen broj aktivacijskih funkcija u neuronima skrivenog sloja nije dovoljan da aproksimira funkciju preslikavanja. Prevelik broj uzoraka lako može dovesti do pojave pretreniranosti, na primjer mreža može početi učiti eventualni šum u podacima za učenje. Stoga je prilikom učenja mreže potrebno obratiti posebnu pozornost na kvalitetu ulaznih podataka, te iskustveno, ili metodom pokušaja i pogreške odrediti prikladan broj neurona skrivenog sloja.

Postupak učenja RBF mreže zasniva se na određivanju sljedećih parametara:

- K L -dimenzionalnih vektora centara \mathbf{t}_j neurona skrivenog sloja
- K $L \times L$ -dimenzionalnih kovarijantnih matrica $\mathbf{\Sigma}_j$ neurona skrivenog sloja
- $K \times M$ -dimenzionalne matrice težinskih faktora \mathbf{C} izlaznog sloja.

Postupak učenja definiran je sljedećim koracima:

1. Odrede se klasifikacijske značajke i skupine, uzorci za učenje se razvrstaju po skupinama.
2. Uzorci za učenje se normiraju u intervalu $[0, 1]$ (normiranje je neobavezno).
3. Svaka se klasifikacijska skupina dodatno razdjeli na niz manjih grupa, koristeći parametar grupiranja β , detaljnije opisano niže.
4. Za svaku grupu se računa centar, kao srednja vrijednost grupe.

5. Uz svaki centar izračunaju se i pripadajuće kovarijantne matrice prema izrazu (5).
6. Na temelju dobivenih centara i kovarijantnih matrica računaju se kvadratne Mahalanobisove udaljenosti prema izrazu (3). Pritom se eventualne nule na glavnoj dijagonali zamijene sa dovoljno malenim iznosom, čime se osigurava regularnost matrice.
7. Prema izrazu (2) računaju se izlazi aktivacijskih funkcija, odnosno formira se matrica H .
8. Prema izrazu (7) ili prema izrazima (8-9), računa se matrica težinskih faktora C . Pritom je matrica O matrica željenih izlaza.

Centri i kovarijantne matrice neurona skrivenog sloja, kao i težine izlaznog sloja predstavljaju bazu znanja neuronske mreže o učenom problemu. Jednom definirani u fazi učenja, oni se više ne mijenjaju.

Testiranje ili naknadno korištenje mreže provodi se u četiri sljedeća koraka:

1. Izračunaju se kvadratne Mahalanobisove udaljenosti svih uzoraka od svih centara neurona skrivenog sloja.
2. Izračuna se nova matrica vrijednosti radijalnih baznih funkcija H .
3. Korištenjem izraza (6) odrede se pripadnosti uzoraka definiranim klasifikacijskim skupinama.
4. Konačna pripadnost uzorka određenoj skupini ocjenjuje se na temelju izlaznog neurona s najvećom vrijednošću.

2.3.4. *Određivanje broja i centara neurona skrivenog sloja*

Proces razdvajanja klasifikacijskih skupina na manje grupe vrlo je važan korak u učenju mreže. Broj grupa je izravno povezan s brojem neurona skrivenog sloja, što utječe na brzinu izračuna i uspješnost klasifikacije. Bitno je spomenuti i važnost kvalitetne raspodjele centara u prostoru učenja, pogotovo za višedimenzionalne probleme. Dodatna separacija klasifikacijskih skupina se radi upravo kako bi se što bolje pokrilo područje učenja, iz čega se da pretpostaviti da nasumično postavljanje centara u većini slučajeva neće rezultirati dobrim odazivom mreže. Stoga se u ovom radu grupiranje obavlja uz pomoć takozvanog parametra grupiranja β . Povećanjem iznosa parametra povećava se i broj članova grupa, čime se broj grupa smanjuje, i obratno. Algoritam određivanja centara prema [5] opisan je u sljedećih šest koraka:

1. Prvi razmatrani uzorak postavi se za centar prve grupe.
2. Izračunaju se Euklidske udaljenosti između centra i svih uzoraka klasifikacijske skupine.
3. Ako je udaljenost uzorka od centra manja od odabranog iznosa parametra β , tada se uzorak svrstava u razmatranu grupu. U suprotnom uzorak ostaje van grupe.
4. Nakon analize svih udaljenosti ponovo se pristupa računanju centra grupe na način da se izračunaju aritmetičke sredine svih uzoraka grupe po dimenzijama.
5. Nova vrijednost centra ponovno prolazi račun opisan koracima 2-4 čime se dobiva konačan raspored uzoraka, odnosno centar promatrane grupe.
6. Grupirani uzorci izuzimaju se iz proračuna ostalih centara. Prvi sljedeći neraspoređeni uzorak uzima se za inicijalnu vrijednost centra druge grupe i postupak se ponavlja od 2. koraka.

Ovim se algoritmom jednostavno može upravljati strukturom sakrivenog sloja mreže mijenjanjem samo jednog parametra. Do željene se vrijednosti dolazi metodom pokušaja i pogreške, najčešće uz vrlo malo iteracija.

3. OPTIMIZACIJA ROJEM ČESTICA

3.1. Uvod u optimizaciju rojem čestica

Optimizacija rojem čestica (*eng.* PSO – Particle Swarm Optimization) je iterativna metoda pronalaska optimalnog rješenja inspirirana zajedničkim djelovanjem životinja, poput jata riba ili ptica, te nekih kukaca poput pčela u cilju zaštite i pronalaženja područja s najboljom hranom. Prvi su je razvili Eberhart i Kennedy sredinom devedesetih [6]. Metoda je donekle slična genetskim algoritmima: dok kod genetskih algoritama postoje fenotipi, kod PSO-a se oni nazivaju česticama, te su oboje sastavljeni od parametara koje se želi optimizirati. U PSO-u se pak ne koriste operatori poput selekcije, rekombinacije i mutacije, već čestice samostalno istražuju prostor rješenja i surađuju komunikacijom s ostalim česticama prenoseći im informacije o svojim prošlim najboljim rješenjima.

Metoda ima široko područje primjene jer zahtjeva malo ili nimalo pretpostavki o suštini problema koji se optimizira, te jer može pretražiti vrlo veliko područje potencijalnih rješenja u prihvatljivom vremenu. Dodatno, PSO algoritam ne koristi gradijent (poput algoritma povratnog prostiranja pogreške), pa ne postoji ograničenje da problem mora biti diferencijabilan. Kao i druge metaheurističke metode, PSO nema garanciju pronalaska optimalnog rješenja – moguća su zapinjanja u lokalnom maksimumu/minimumu i divergencije rješenja.

Od prvog spominjanja algoritma, pa sve do danas, razvile su se mnogobrojne modifikacije i unaprjeđenja originalnog PSO algoritma. U ovom će se radu usporediti performanse klasične RBF neuronske mreže opisane u prošlom poglavlju i performanse RBF mreže učene primjenom originalnog PSO algoritma i tri njegove modifikacije.

3.2. Osnovni algoritam optimizacije rojem čestica

Prvi korak u svakom postupku optimiziranja je određivanje parametara koji će biti podešavani algoritmom. Kod PSO-a, na potencijalna rješenja se gleda kao na čestice u višedimenzionalnom prostoru – n parametara tvori n -dimenzionalan prostor u kojem su postavljene čestice. Njihove su pozicije opisane s n koordinata, a pomak u prostoru znači i promjenu vrijednosti parametara. Čest je slučaj u praksi da se prostor ograniči po dimenzijama kako bi se onemogućilo česticama da pretražuju besmislena ili nezanimljiva rješenja, no ta su ograničenja vezana uz specifičnost problema.

Inicijalizacija roja se izvršava tako da se pozicije čestica generiraju nasumično. Sama veličina roja je vezana uz dimenzionalnost problema, no unatoč postojanju niza izraza koji daju okvirnu informaciju o potrebnom broju čestica, veličina se često određuje metodom pokušaja i pogreške. Tu se preferiraju što manji rojevi zbog brzine izračuna.

Uz pozicije čestica, generiraju se i pripadajuće brzine čestica, po jedna za svaku dimenziju. Već je ranije spomenuto kako je PSO iterativna metoda – u svakoj iteraciji algoritma kvaliteta učenja se ocjenjuje uz pomoć funkcije cilja. To može biti normalizirani korijen srednje kvadratne pogreške, postotak točno klasificiranih uzoraka ili neka druga funkcija koju korisnik želi minimizirati, odnosno maksimizirati. Brzina čestica se tada ažurira uzevši u obzir inerciju čestice, njenu poziciju, to jest iznos kod aktualnog najboljeg rezultata učenja, te te poziciju susjedne čestice koja je među susjednim česticama postigla najbolji rezultat, prema sljedećem izrazu:

$$v_{ij}(k+1) = av_{ij}(k) + r_1c_1(p_{\text{indiv},ij} - x_i(k)) + r_2c_2(p_{\text{lok},ij} - x_i(k)), \quad (10)$$

gdje su $v_{ij}(k+1)$ i $v_{ij}(k)$ j -te komponente brzine i -te čestice u sljedećoj i trenutnoj iteraciji, a je koeficijent koji određuje inerciju čestice, r_1 i r_2 su slučajno odabrane vrijednosti u intervalu $(0,1)$, c_1 i c_2 su koeficijenti koji određuju utjecaj privlačenja j -te komponente prošle najbolje pozicije čestice $p_{\text{indiv},ij}$, odnosno j -te komponente pozicije najbolje susjedne čestice $p_{\text{lok},ij}$.

Sljedeća pozicija čestice se onda ažurira prema izrazu

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \quad (11)$$

Iz izraza (10) se vidi da na brzinu utječu tri faktora. Prvi je istraživačka komponenta brzine, gdje skalirani iznos prethodne vrijednosti brzine predstavlja inerciju i povlači česticu prema

novim područjima. Druga komponenta predstavlja povlačenje u smjeru individualno postignutog najboljeg rezultata u prošlim iteracijama, a treća je takozvana „društvena komponenta“ koja česticu povlači u smjeru najboljeg rezultata postignutog u susjedstvu čestice. Posljednje dvije komponente otežane su koeficijentima c_1 i c_2 koji određuju omjer utjecaja individualne i društvene najbolje vrijednosti, te slučajnim iznosima r_1 i r_2 koji bi trebali unijeti određenu dozu stohastičnosti u pretraživanje, te su po tome slični operaciji mutacije kod genetskih algoritama. Te se dvije komponente nazivaju eksploatacijskim komponentama, jer iskorištavaju prethodno pronađene najbolje rezultate u navođenju čestica.

Susjedstvo čestice je definirano topološkom strukturom koju korisnik odredi prije početka učenja. U ovom se radu koristila prstenasta struktura: ako se i -toj čestici odredi susjedstvo veličine dva, njeni će susjedi tada biti čestice $(i-1)$ i $(i+1)$.

Iznosi brzina i pozicija se mogu prema potrebi ograničiti u željenom intervalu kako bi se spriječila divergencija putanja čestica.

Osnovni PSO algoritam opisan je u sljedećih pet koraka:

1. Inicijalizira se roj od N čestica, sa pripadajućim slučajno odabranim pozicijama u zadanom intervalu i brzinama jednakim nuli.
2. Svaka se čestica ocijeni funkcijom cilja. Ažuriraju se individualno najbolje pozicije i lokalne najbolje pozicije.
3. Izračunaju se nove brzine prema izrazu (10), i nakon toga pozicije prema izrazu (11).
4. Algoritam se ponavlja od 2. koraka, dok nije ostvaren uvjet prekida (maksimalan broj iteracija ili vrijednost funkcije cilja).
5. Iz roja se uzima najbolja čestica, te se ona smatra najboljim rješenjem.

3.3. Unificirani algoritam optimizacije rojem čestica

U ekstremnom slučaju, susjedstvo čestica se može proširiti na cijeli roj, čime se dobiva globalna varijanta PSO algoritma, za razliku od lokalne varijante opisane u prethodnom poglavlju. Sami tvorci PSO-a kritiziraju globalni pristup zato što je sklon preranoj konvergenciji koja je posljedica dominacije eksploatacije nad istraživanjem. U suštini, kada se jedna čestica zaglavi u lokalnom optimumu, ona vrlo brzo povuče i ostale čestice sa sobom, čime se prekida optimizacija i koči napredak.

Unatoč tome, bez eksploatacijskih karakteristika i komunikacije s rojem, PSO teško može doći do zadovoljavajućeg rješenja. Rad Parsopoulos i Vrahatisa iz 2004. [7] pokazuje da ujedinjavanje lokalne i globalne verzije algoritma može dovesti do zadovoljavajućih, a u nekim slučajevima i superiornijih rezultata u odnosu na njihovu individualnu primjenu. To se postiže balansiranjem lokalne varijante, sa slabijim oblikom komunikacije (zbog manjeg susjedstva, informacija o postignutim rezultatima se sporije propagira kroz roj), sa globalnom varijantom koja u istom trenutku šalje informaciju o najboljem rezultatu cijelom roju.

Takva se verzija naziva unificirani algoritam optimizacije rojem čestica, a izrazi (10) i (11) se preuređuju na sljedeći način:

$$v_{ij}(k+1) = av_{ij}(k) + r_1c_1(p_{indiv,ij} - x_i(k)) + (1-u)r_2c_2(p_{lok,ij} - x_i(k)) + ur_2c_2(p_{glob,j} - x_i(k)), \quad (12)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \quad (13)$$

gdje je u unifikacijski koeficijent kojim se određuje omjer utjecaja lokalne i globalne varijante u intervalu $[0,1]$, a $p_{glob,j}$ je j -ta komponenta pozicije čestice koja je u cijelom roju ostvarila najbolji rezultat učenja.

Algoritam se može opisati u sljedećih pet koraka:

1. Inicijalizira se roj od N čestica, sa pripadajućim slučajnim pozicijama u zadanom intervalu i brzinama jednakim nuli, te se odredi iznos unifikacijskog koeficijenta u .
2. Svaka se čestica ocijeni funkcijom cilja. Ažuriraju se individualno najbolje pozicije, lokalne najbolje pozicije, te globalno najbolja pozicija.
3. Izračunaju se nove brzine prema izrazu (12), i nakon toga pozicije prema izrazu (13).

4. Algoritam se ponavlja od 2. koraka, dok nije ostvaren uvjet prekida (maksimalan broj iteracija ili vrijednost funkcije cilja).
5. Iz roja se uzima najbolja čestica, te se ona smatra najboljim rješenjem.

3.4. Algoritam optimizacije rojem čestica s osiguranom konvergencijom

Ranije je već spomenuto da globalna varijanta PSO-a ima problema s preranom konvergencijom. Ako čestica postigne najbolju poziciju u određenom trenutku u iteraciji, ta pozicija u istom trenutku postaje njena i globalna najbolja vrijednost, što za posljedicu ostavlja samo istraživački član u izrazu za brzinu (ostali postaju jednaki nuli). Dodatno, ako je prošla brzina bila malenog iznosa, čestica lako može stati s gibanjem i ostati zatočena u toj poziciji. Druge čestice tada vrlo brzo budu privučene globalnom najboljem rezultatu, te algoritam konvergira prerano u lokalnom optimumu.

U radu Carvalha i Ludermir [8] opisan je algoritam optimizacije rojem čestica s osiguranom konvergencijom (*eng.* GCPSO, Guaranteed Convergence PSO) kojim se pokušava doskočiti preranoj konvergenciji modifikacijom izraza za brzinu onih čestica koje su postigle najbolju vrijednost.

Izrazi za promjene brzina najboljih čestica se tada pišu na sljedeći način:

$$v_{ij}(k+1) = -x_{ij}(k) + p_{glob.,j} + av_{ij}(t) + \rho(k)(1 - 2r(k)), \quad (14)$$

Iz izraza se da primijetiti da prva dva člana resetiraju česticu u globalno najbolju poziciju. Prvi od preostala dva člana predstavlja već poznatu inerciju čestice, dok se s posljednjim vrši slučajna pretraga prostora u okolišu globalno najbolje pozicije. Koeficijent $r(k)$ predstavlja slučajnu varijablu u intervalu $(0, 1)$, dok $\rho(k)$ određuje širinu područja pretrage.

Iznos koeficijenta $\rho(k)$ je prilagodljiv, a njegova vrijednost u sljedećoj iteraciji ovisi o uspješnosti algoritma u pronalasku nove, bolje globalne vrijednosti, prema izrazu (15).

$$\rho(k+1) = \begin{cases} 2\rho(k), & \text{ako broj uspjeha} > s_c \\ 0.5\rho(k), & \text{ako broj neuspjeha} > f_c \\ \rho(k) & \end{cases} \quad (15)$$

gdje su s_c i f_c pragovi za uspjehe, odnosno neuspjehe koje korisnik postavi na početku učenja. Dakle, svaki put kada broj uspjeha algoritma da pronade bolje pozicije prijeđe prag s_c , područje za istraživanje se povećava udvostručavanjem koeficijenta $\rho(k)$. Slično, svaki put kada broj neuspjeha algoritma da pronade bolje pozicije prijeđe prag f_c , područje za istraživanje se suzuje prepolovljivanjem koeficijenta $\rho(k)$. U suprotnom, ako niti jedan prag nije prijeđen, koeficijent ostaje isti kao i u prošlom koraku.

Dodatno, svaki put kada algoritam uspije pronaći bolje rješenje, brojač uspjeha se poveća za jedan, a brojač neuspjeha se resetira na nulu. Svaki put kada algoritam doživi neuspjeh, brojač neuspjeha se poveća za jedan, a brojač uspjeha se resetira na nulu.

Kako bi se smanjio efekt eksploracije u kasnijim fazama algoritma, svaki put kada se prekorači bilo koji od dva praga, on se poveća za jedan kako bi sljedeći put bio teže postignut.

Početna vrijednost koeficijenta $\rho(0)$ se obično postavlja jednakoj jedan, dok se početne vrijednosti pragova postavljaju jednakima pet.

Sve ostale brzine čestica koje ne postižu globalno najbolje vrijednosti u određenoj iteraciji se računaju standardnim izrazom (10), gdje susjedstvo predstavlja cijeli roj (globalna varijanta).

Algoritam se može opisati u sljedećih pet koraka:

1. Inicijalizira se roj od N čestica, sa pripadajućim slučajnim pozicijama u zadanom intervalu i brzinama jednakim nuli. Također se odrede početne vrijednosti pragova s_c i f_c , te koeficijenta ρ .
2. Svaka se čestica ocijeni funkcijom cilja. Ažuriraju se individualno najbolje pozicije, te globalno najbolja pozicija.
3. Brzine se računaju na dva različita načina:
 - I. Najbolje čestice se ažuriraju prema izrazu (14), a koeficijent ρ se za trenutni korak iteracije ažurira prema izrazu (15). Pozicije se ažuriraju prema izrazu (11).
 - II. Za ostale čestice nove brzine se izračunaju prema izrazu (10), uzevši globalnu verziju PSO-a, te nakon toga pozicije prema izrazu (11).
4. Algoritam se ponavlja od 2. koraka, dok nije ostvaren uvjet prekida (maksimalan broj iteracija ili vrijednost funkcije cilja).
5. Iz roja se uzima najbolja čestica, te se ona smatra najboljim rješenjem.

3.5. Metoda suprotnih vrijednosti uz korištenje Cauchyevog mutacijskog operatora

Još jedna od metoda kojom se pokušava izbjeći prerana konvergencija globalne verzije PSO algoritma je metoda suprotnih vrijednosti, opisana u radu Wanga et al. [9]. Suština postupka leži u generiranju takozvanih suprotnih pozicija u pokušaju da se time česticu prije svega izbacila iz lokalnog optimuma, no i da se stvori prilika za istraživanje i pronalazak boljih rješenja na suprotnom dijelu prostora pretraživanja.

Suprotna pozicija se generira simetričnim zrcaljenjem originalne pozicije u prostoru učenja definiranim intervalom $[a_j^P(k), b_j^P(k)]$, gdje je $a_j^P(k)$ minimalna vrijednost i $b_j^P(k)$ maksimalna vrijednost koju čestice postižu u j -toj dimenziji u k -toj iteraciji algoritma. Smisao dinamičkog osvježavanja intervala je da se njime pruži dovoljno prostora za istraživanje, no da se ne koči previše prilika za konvergencijom. Kada bi interval stalno obuhvaćao cijeli mogući prostor za učenje, postojala bi opasnost da bi suprotne pozicije često bile postavljene u nepovoljnim i predalekim područjima od optimuma, upravo zato jer bi udaljenosti bile prevelike.

Suprotna pozicija se onda definira prema sljedećem izrazu:

$$x_{supr.,ij}(k) = a_j^P(k) + b_j^P(k) - x_{ij}(k), \quad (16)$$

gdje je $x_{supr.,ij}$ j -ta komponenta suprotne pozicije čestice i .

Nakon generiranja suprotnih vrijednosti, sve se pozicije čestica ocjene funkcijom cilja, te se originalna pozicija zamijeni suprotnom ukoliko je suprotna bolja.

U metodu suprotnih pozicija uvodi se i takozvana suprotna vjerojatnost p_0 . U svakoj iteraciji algoritma generira se slučajni broj u intervalu $(0, 1)$. Ukoliko je taj broj manji od suprotne vjerojatnosti, izračun pozicija se obavlja prema izrazu (16), a u suprotnom on se odvija prema standardnoj globalnoj verziji PSO algoritma, izrazi (10) i (11). U literaturi se ta vjerojatnost drži oko 0.3 [9].

Algoritmu se još dodala i mutacija globalno najbolje čestice uvođenjem člana koji implementira Cauchyevu distribuciju, s namjerom poboljšanja šanse za izlaskom iz lokalnog optimuma. Cauchyeva funkcija gustoće vjerojatnosti za jednodimenzionalan problem s centrom u ishodištu definirana je sljedećim izrazom:

$$f(x) = \frac{1}{\pi} \frac{t}{t^2 + x^2}, \quad -\infty < x < \infty, \quad (17)$$

gdje je t parametar skaliranja funkcije. Kumulativna funkcija distribucije vjerojatnosti glasi:

$$F_t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x}{t}\right), \quad (18)$$

pri čemu $F_t(x)$ opisuje kolika je vjerojatnost da slučajna varijabla poprimi vrijednost manju ili jednaku ulaznoj vrijednosti x .

Globalno najbolja čestica se mutira uz pomoć Cauchyevog operatora:

$$W(k) = \frac{(\sum_{i=1}^N v_i(k))}{N}, \quad (19)$$

gdje je N broj čestica roja, a $W(k)$ je srednja vrijednost brzina svih čestica u k -toj iteraciji. Taj se član dodaje izrazu za izračun mutirane pozicije globalno najbolje čestice na sljedeći način:

$$p_{glob.mut.} = p_{glob.} + W(k) * F_t(x_{min}, x_{max}), \quad (20)$$

gdje je $p_{glob.mut.}$ mutirana najbolja pozicija čestice, dok se Cauchyev operator $W(k)$ množi s kumulativnom funkcijom distribucije $F_t(x)$, $t = 0$, čiji je ulaz slučajni iznos odabran između granica područja pretraživanja. Ukoliko je mutirana pozicija bolja od originalne, ona se postavi za najbolju, dok se u suprotnome odbacuje.

Algoritam se može opisati u sljedećih pet koraka:

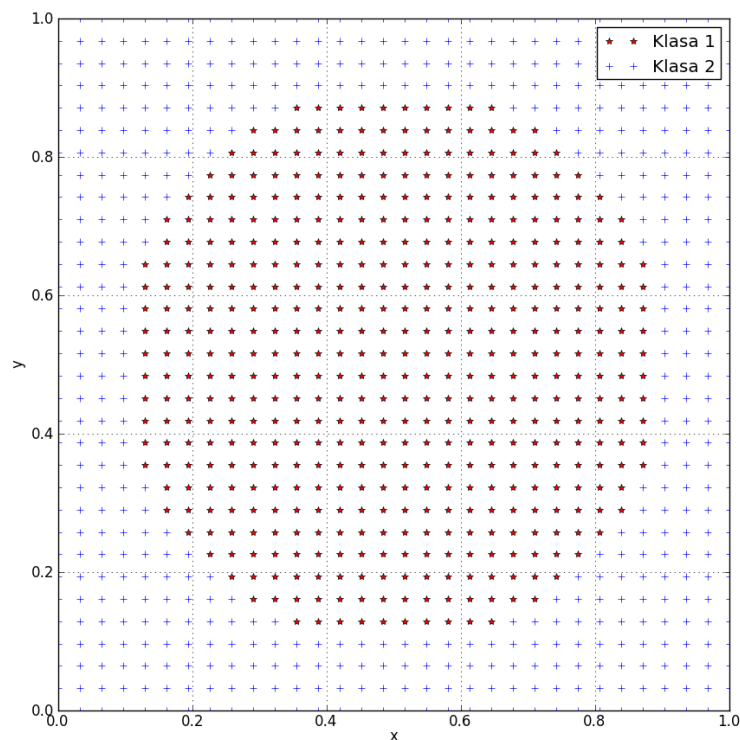
1. Inicijalizira se roj od N čestica, sa pripadajućim slučajnim pozicijama u zadanom intervalu i brzinama jednakim nuli. Također se odredi i iznos suprotne vjerojatnosti p_0 .
2. U svakom koraku algoritma generira se slučajan broj r u intervalu $(0, 1)$.
 - I. Ako je $r < p_0$, izračunaju se trenutne minimalne i maksimalne vrijednosti pozicija čestica po dimenzijama, te se generiraju suprotne čestice prema izrazu (16). Funkcijom cilja se ocjene sve originalne i suprotne čestice, te se uzme N najboljih za sljedeću iteraciju.
 - II. Ako je $r \geq p_0$, izračun brzina se odvija prema izrazu (10) za globalnu verziju PSO algoritma, a pozicije se ažuriraju prema izrazu (11).
3. Ažuriraju se individualno najbolje pozicije, te globalno najbolja pozicija.
4. Algoritam se ponavlja od 2. koraka, dok nije ostvaren uvjet prekida (maksimalan broj iteracija ili vrijednost funkcije cilja).
5. Iz roja se uzima najbolja čestica, te se ona smatra najboljim rješenjem.

4. PROBLEMI ZA UČENJE I TESTIRANJE

4.1. Problem 1 – Krug u kvadratu

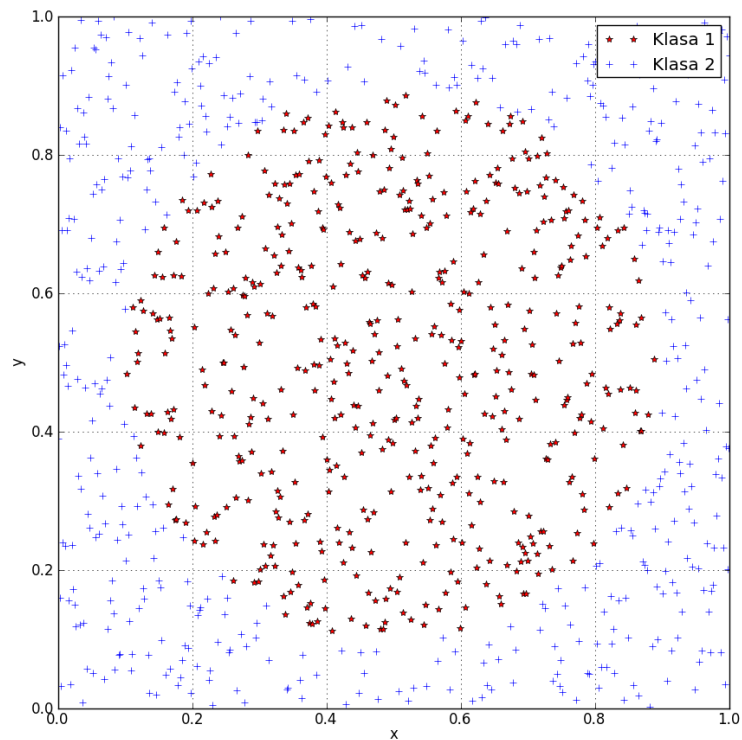
Prvi problem s kojim će se analizirati mreža je takozvani „Krug u kvadratu“, a zadatak neuronske mreže je ispravno odrediti nalazi li se uzorak u kružnici ili van nje.

Uzorci za učenje su koordinate točaka na ravnini smještene u jediničnom kvadratu, stoga će neuronska mreža imati dva ulazna neurona, po jedan za pripadajuću koordinatu. Uzorci se svrstavaju u dvije klase, gdje prva broji 484 uzoraka koji se nalaze unutar kružnice opisane unutar kvadrata, dok se ostalih 540 nalazi s vanjske strane, tako da će neuronska mreža imati dva izlazna neurona, po jedan za svaku klasu.



Slika 5. Uzorci za učenje problema 1

Uzorci za testiranje slijede istu logiku, s time da uzoraka koji pripadaju prvoj klasi (unutar kružnice) ima 521, dok uzoraka koji pripadaju drugoj klasi (van kružnice) ima 479. Kako bi se ispitala generalizacijska svojstva mreže, uzorci za testiranje se ne preklapaju s uzorcima za učenje.



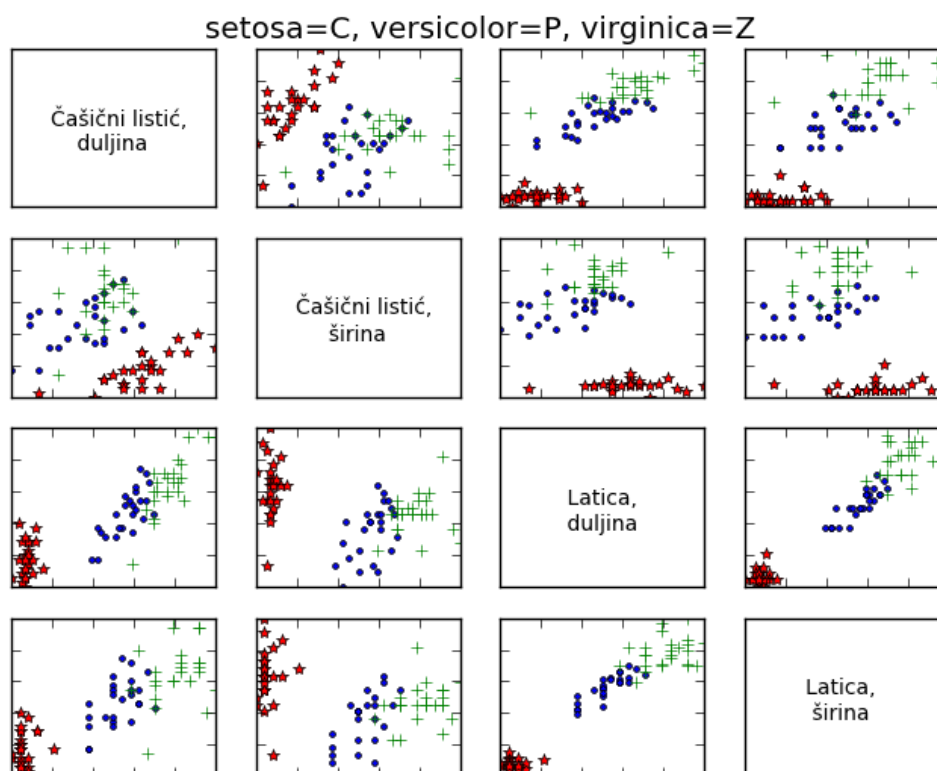
Slika 6. Uzorci za testiranje problema 1

Na grafovima uzoraka za učenje [Slika 3] i testiranje [Slika 4] jasno je da se radi o linearno neseparabilnom problemu – ne postoji pravac koji se može povući na takav način da jasno odvaja dvije klase. To znači da će neuronska mreža morati pokazati sposobnost nelinearnog odvajanja uzoraka, što će vjerojatno biti aproksimacija jednadžbe kružnice. Uzorci različitih klasa se ne preklapaju, što olakšava učenje.

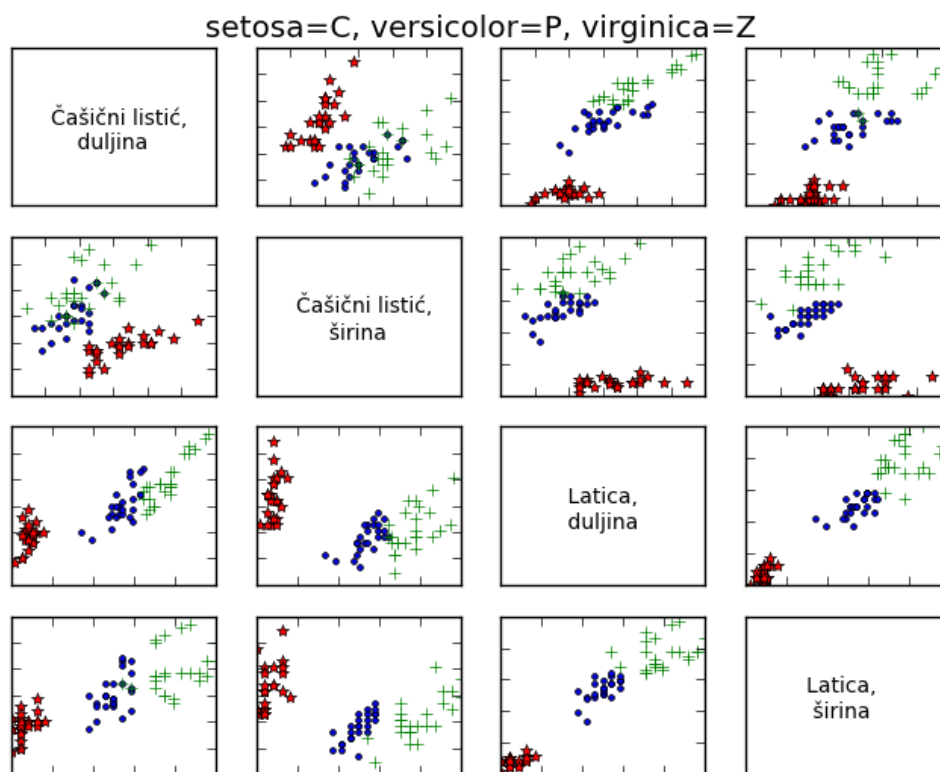
4.2. Problem 2 – Iris set podataka

Drugi problem na kojem će se testirati mreža je *Iris* set podataka, prvi put spomenut 1936. godine u radu Ronalda Fishera [10].

Problem se svodi na klasifikaciju tri vrste cvijeta irisa (*Iris setosa*, *Iris virginica* i *Iris versicolor*) na temelju četiri mjerene značajke – duljinama i širinama čašičnog listića i latice cvijeta. Mreža će onda imati četiri ulazna neurona i tri izlazna neurona. Za svaku od tri vrste cvijeta dano je 50 uzoraka, te će se u ovom radu polovica koristiti za učenje mreže, dok će se na drugoj polovici vršiti testiranje. Iako su se izvorne značajke mjerile u centimetrima, ovdje su one normirane između nule i jedinice. Uzorci za učenje [Slika 5] i testiranje [Slika 6] prikazani su dvodimenzionalnim grafovima, gdje svaki red prikazuje odnos jedne značajke sa svim ostalima. Na njima je vidljivo da su podaci za cvijet *I. setosa* (prikazani crvenim zvijezdama) linearno separabilni od podataka za ostala dva cvijeta, no podaci za *I. versicolor* (plavi kružići) i *I. virginica* (zeleni križići) pokazuju preklapanja. Tu će neuronska mreža moći pokazati kvalitetu nelinearnog odvajanja uzoraka.



Slika 7. Uzorci za učenje problema 2



Slika 8. Uzorci za testiranje problema 2

4.3. Problem 3 – Određivanje stupnja istrošenosti alata kod bušenja kompozita

4.3.1. Opis problema 3

Posljednji problem na kojem će se testirati neuronska mreža je određivanje stupnja istrošenosti alata kod bušenja kompozita. M. Zoričić je u svom diplomskom radu [11] obradio i analizirao signale sila rezanja i struja pogonskih motora alatnog stroja snimljenih tijekom bušenja polimernog kompozita ojačanog staklenim vlaknima. Pomoću značajki izdvojenih iz navedenih vrsta signala pokušalo se utvrditi može li se umjetnom neuronskom mrežom uspješno procijeniti stupanj istrošenosti svrdla.

U ovom će se radu uzeti set podataka koje je Zoričić koristio za učenje i testiranje neuronske mreže. Značajke koje su se pokazale najboljima za utvrđivanje stupnja istrošenosti alata su posmak f , brzina rezanja svrdla v_c , maksimalna struja glavnog vretena glodalice $Max(I_{gv})$, maksimalna struja motora osi z $Max(I_z)$, te površina ispod krivulje struje motora osi z $Pov(I_z)$. Klasifikacijske skupine koje su se koristile za ocjenu istrošenosti alata su $S1$ (oštro), do $S4$ (potpuno tupo). Dakle, neuronska mreža će imati pet ulaznih neurona i četiri izlazna neurona.

Na temelju rezultata koje je dobio Zoričić, napraviti će se usporedba s algoritmima opisanim u ovom radu.

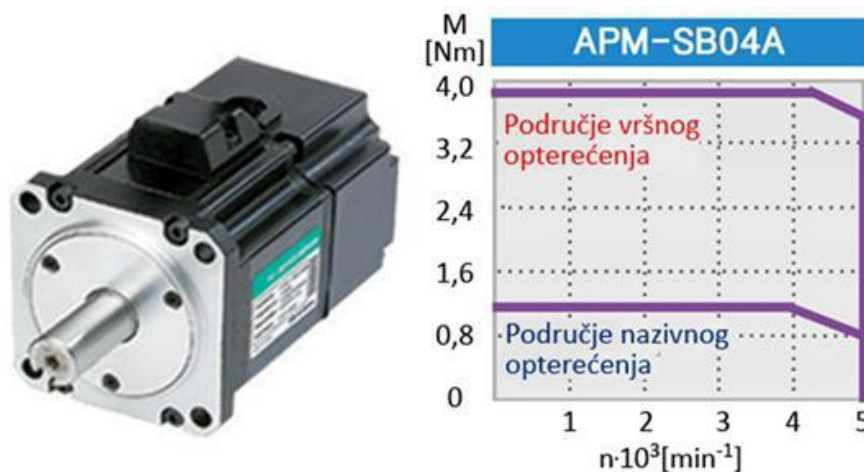
Postavke korištene u izradi njegovog rada ukratko će se objasniti niže.

4.3.2. Detalji mjerenja

Bušenje kompozita je izvedeno na troosnoj glodalici s tri translacijske osi i vertikalnim glavnim vretenom. Glavni prigon je izveden direktno, to jest prihvat alata je ugrađen na motor. Posmični prigon izveden je kombinacijom kliznih vodilica i kugličnog navojnog vretena. Za sva četiri pogonska motora koristili su se sinkroni servo motori proizvođača Mecapion Ltd., tip APM-SB04ADK.

Tablica 1. Tehničke karakteristike motora [11]

	Vrijednost	Jedinica
Nazivna snaga	400	W
Nazivni moment	1,274	N·m
Maksimalni moment	3,822	N·m
Moment inercije	0,321	$\text{kg}\cdot\text{m}^2\times 10^4$
Nazivna učestalost vrtnje	3000	min^{-1}
Nazivna jakost struje	2,89	A_{RMS}
Maksimalna učestalost vrtnje	5000	min^{-1}
Rezolucija enkodera	3000	impulsa/okretaju
Konstanta momenta vrtnje motora (K_T)	0,46	$\text{N}\cdot\text{m}/A_{\text{RMS}}$
Naponska konstanta vrtnje motora (K_V)	29,5	$V_{\text{RMS}}/(\text{min}^{-1}\times 10^3)$



Slika 9. Sinkroni servo motor [11]

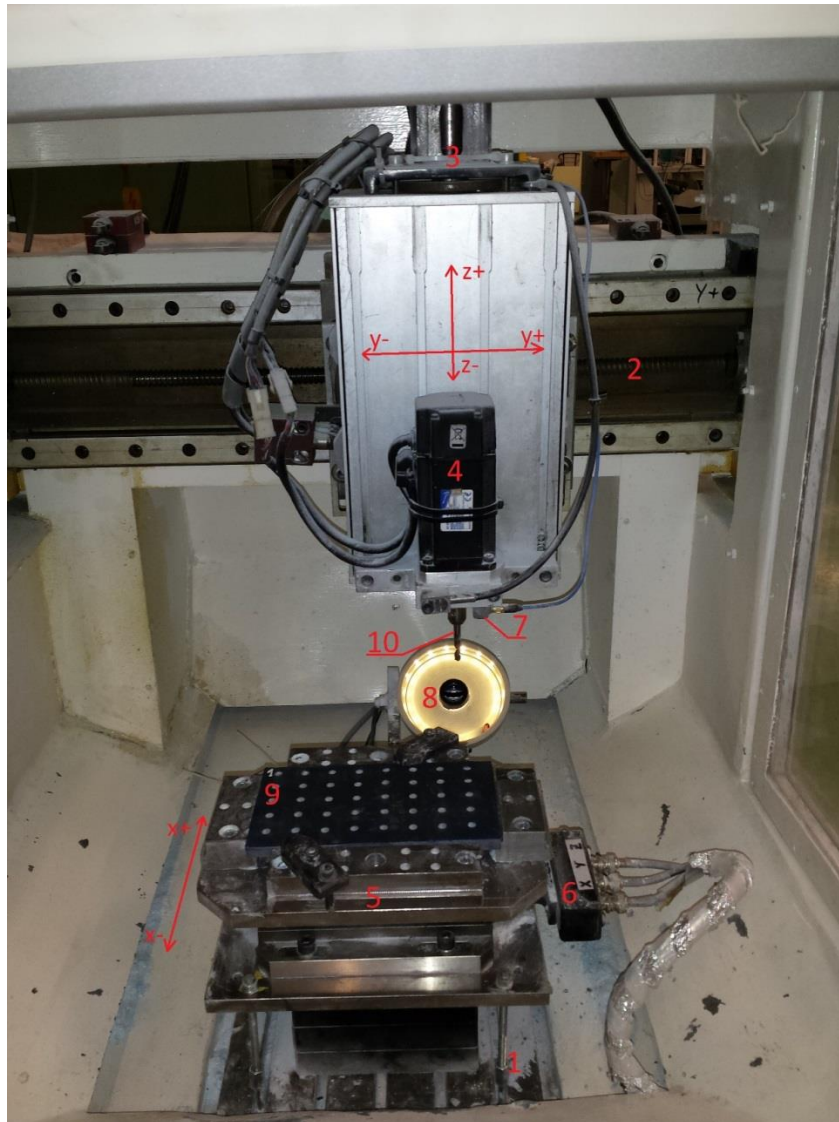
Za mjerenje signala sila korišten je mjerni sustav sačinjen od tro-komponentnog piezo-električnog senzora sila proizvođača Kistler Holding AG, model 9257B u kombinaciji s nabojnim pojačalom istog proizvođača, model 5017B. Senzor je ugrađen ispod stezne naprave na posmičnom prigonu x osi.

Tablica 2. Tehničke karakteristike mjernog sustava sila [11]

		Područje	Jedinica	
Senzor sila (tip 9257B)	Mjerno područje	F_x, F_y	-5 ... 5	
		F_z	-5 ... 10	
	Maksimalne dozvoljene sile	F_x, F_y	-7,5/7,5	
		F_z	-7,5/15	
	Osjetljivost	F_x, F_y	-7,5	
		F_z	-3,7	
	Prag osjetljivosti		<0,01	N
	Linearnost u mjernog područja		<±1	%
Prag histereze mjernog područja		<0,5	%	
Nabojno pojačalo (tip 5017B)	Broj kanala	3 (do 8)		
	Mjerno područje (podesivo po kanalu)	±10 ... 10 ⁶	pC	
	Frekvencijsko područje	0 ... 200	kHz	
	Osjetljivost (podesiva po kanalu)	0,01 ... 9990	pC/N	
	Skaliranje mjerne veličine (podesivo po kanalu)	0,001 ... 10 ⁷	N/V	
	Izlazni signal (po kanalu)	±10	V	



Slika 10. Senzor sila Kistler 9257B [11]



Slika 11. Radni prostor stroja; 1. Posmični prigon X osi, 2. Posmični prigon Y osi, 3. Posmični prigon Z osi, 4. Glavno vreteno, 5. Stezna naprava, 6. Senzor sila, 7. Senzor vibracija, 8. Industrijska kamera s rasvjetom, 9. Ispitni uzorak, 10. Ispitno svrdlo [11]

Za direktno mjerenje rezne oštrice alata korišten je vizijski sustav, koji se sastoji od industrijske kamere, telecentričnog objektiva i prstenaste LED rasvjete. Obradom fotografija snimljenih vizijskim sustavom kvantificirana je istrošenost alata mjerenjama širine područja trošenja na stražnjoj površini alata.

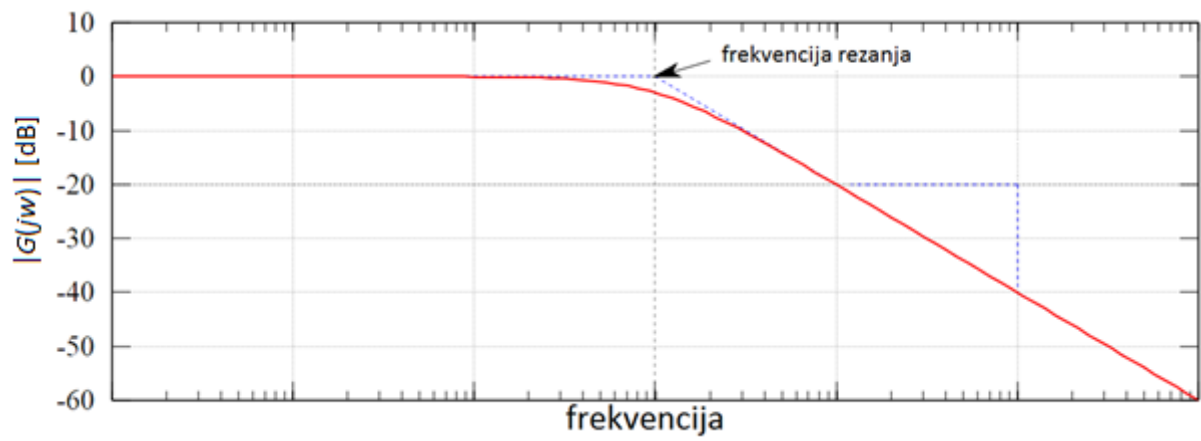
Tablica 3. Tehničke karakteristike vizijskog sustava [11]

		Područje	Jedinica	
Kamera (DMK41AF02)	Rezolucija	1280x960	piksela	
	Osjetljivost	0,05	lx	
	Frekv. osvježavanja slike	3,75 ... 15	Hz	
	Format osjetilnog senzora	1/2	in	
	Izvedba osjetilnog senzora	CCD		
	Prihvat objektiva	C/CS		
Objektiv (TC2309)	Faktor povećanja	1,00 ±3 %		
	Vidno polje (za format senzora 1/2")	6,4x4,8	mm	
	Radna udaljenost	63,3±2	mm	
	Dubinska oštrina	0,9	mm	
	Distorzija	tipična	<0,04	%
		maksimalna	0,08	
	Telecentričnost	tipična	<0,08	°
		maksimalna	0,10	
Izvedba prihvaća		C		
Rezolucija mjernog sustava u opisanoj konfiguraciji		5	µm	



Slika 12. Kamera DMG41AF02 i telecentrični objektiv TC2309 [11]

Zbog prisutnosti visokofrekventnog šuma u signalima mjerenja, za sve mjerene signale primjenjen je Butterworthov niskopropusni filter trećeg reda frekvencije rezanja 2 Hz.



Slika 13. Niskopropusni Butterworthov filter [11]

5. REZULTATI

5.1. Postavke

Klasična RBF neuronska mreža opisana u ovom radu implementirana je u *Python* [12] programskom jeziku, koji je odabran radi povoljnog odnosa jednostavnosti kodiranja i brzine izvršavanja koda, fleksibilnosti, te zbog činjenice što je *Python* besplatan i *open-source*.

Implementacija se obavila prema matematičkom modelu i algoritmu spomenutom u drugom poglavlju. Struktura ulaznog i izlaznog sloja je ovisila o problemu, dok se mreži varirao broj neurona skrivenog sloja modifikacijom parametra β , kako bi se ustanovila optimalna konfiguracija. Ovdje će biti prikazana uspješnost klasifikacije za dvije različite vrijednosti parametra β .

PSO algoritmi opisani u ovom radu su osnovna verzija (**PSO**), te tri njene modifikacije, redom: unificirani PSO algoritam (**UPSO**), PSO algoritam s osiguranom konvergencijom (**GCPSO**), te PSO algoritam koji koristi metodu suprotnih rješenja i Cauchyev mutacijski operator (**OPSO Cauchy**). Kratice navedene u zagradama koristiti će se dalje u tekstu kada se govori o pojedinom algoritmu.

Algoritmi optimizacije će biti primijenjeni u učenju mreže na način da će čestice PSO algoritma pokušati pronaći optimalne vrijednosti elemenata glavne dijagonale kovarijantnih matrica Σ skrivenog sloja. U tom slučaju će broj dimenzija svake čestice ovisiti o broju neurona skrivenog sloja, no i o broju ulaza, odnosno klasifikacijskih značajki. Točnije, dimenzionalnost će biti definirana umnoškom ta dva čimbenika. Modifikacijom parametra β će se tada moći utjecati na veličinu čestica, te će se ispitati koliko dobro PSO algoritam podnosi povećanje broja parametara. Funkcija cilja prema kojoj se optimizira roj je postotak uspješno klasificiranih uzoraka za testiranje

U iterativnim metodama učenja praksa je postaviti sljedeći uvjet prekida: učenje se prekida ako se postigne rješenje sa definiranom dovoljno malenom greškom, u suprotnom se prekida nakon definiranog maksimalnog broja koraka. S obzirom da je suradnja, odnosno interakcija čestica roja ovdje od posebnog interesa, kao i sposobnost pretrage prostora učenja, za svaki će se problem algoritam pustiti da optimizira do nekog fiksnog broja koraka, definiranim posebno za svaki problem učenja.

Klasična RBF mreža uči u jednom koraku, pa će njena uspješnost klasifikacije biti prikazana samo tablično. PSO algoritam uči iteracijski, tako da će se kod svakog algoritma uspješnost

klasifikacije, odnosno vrijednost funkcije cilja prikazati dodatno i grafički, za svaku česticu u svakom koraku iteracije. Čestica koja je u posljednjem koraku postigla najbolje iznose parametara označit će se debljom crvenom linijom, dok će ostale čestice biti označene tanjom crnom isprekidanom linijom. Takav prikaz je odabran u namjeri da se preko vrijednosti funkcije cilja pokaže suradnja roja čestica. Naravno, prikaz putanja čestica tokom učenja bi dao bolju sliku o suradnji, te bi se na takvom prikazu moglo vidjeti grupiranje čestica oko određene točke u prostoru učenja u slučaju konvergencije u lokalnom ili globalnom optimumu, no kod višedimenzionalnih problema to je vrlo teško, a često i nemoguće prikazati na razumljiv način. Uz pretpostavku da su dvije čestice kojima parametri daju jednaku vrijednost funkcije cilja u jednakoj poziciji, dok su dvije čestice čiji parametri daju različite vrijednosti funkcije cilja daleko (što ne mora biti istina), suradnja čestica se onda pojednostavljeno može prikazati preko vrijednosti funkcije cilja.

Uz grafički prikaz, rezultati će se ispisati i tablično: uspješnost klasifikacije uzoraka za učenje i testiranje čestice čija je pozicija u posljednjem koraku imala najbolju vrijednost funkcije cilja, uspješnost klasifikacije uzoraka za učenje i testiranje najbolje pozicije koju je roj uspio pronaći u svim koracima iteracije, te korak u kojem je najbolja pozicija pronađena.

5.2. Problem 1

5.2.1. Vrijednosti parametara algoritama za problem 1

Tablica 4. Vrijednosti parametara algoritama za problem 1

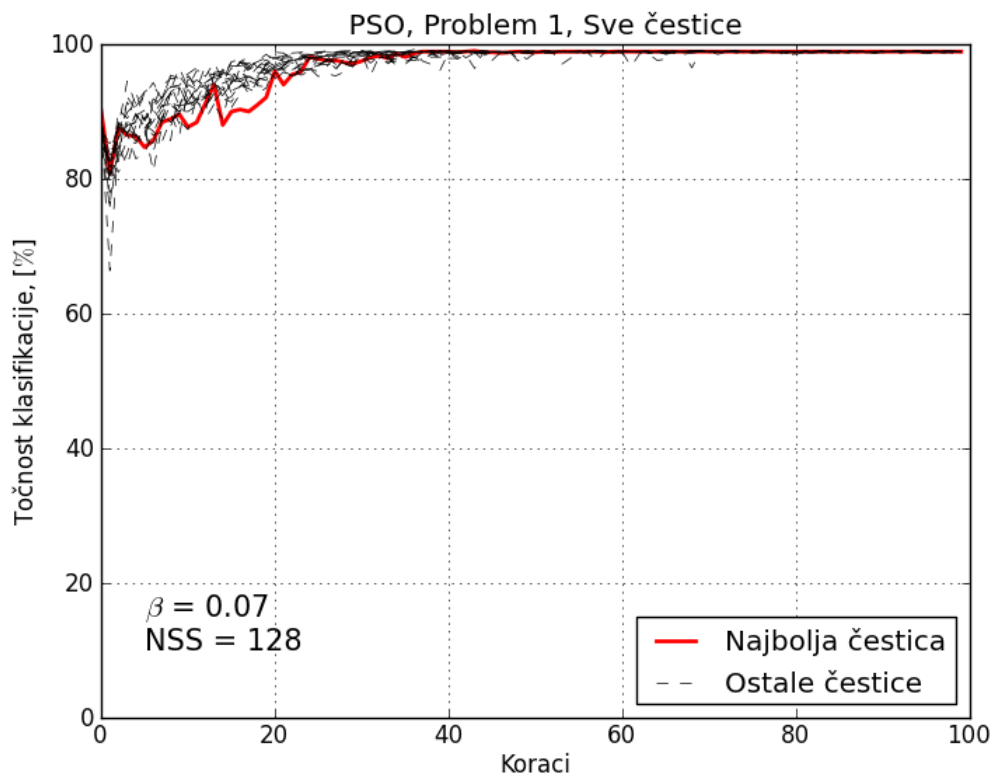
Broj koraka	100
Broj čestica	20
a	1
PSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
UPSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
u	0.3
GCPSO	
$\rho(0)$	1
$s_c(0)$	5
$f_c(0)$	5
OPSO Cauchy	
p_o	0.3

5.2.2. Tablični prikaz rezultata za problem 1

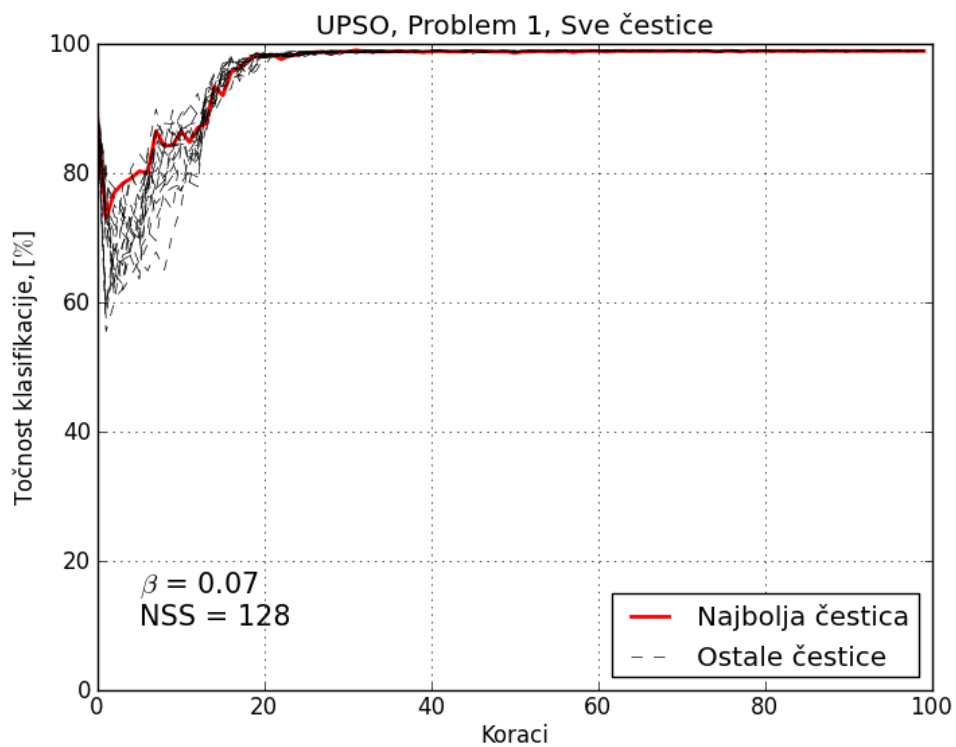
Tablica 5. Rezultati mreža za problem 1

β	NSS	param	MREŽA	učenje	test	učenje	test	Najbolji rezultat u koraku:
				Na kraju učenja [%]		Najbolji rezultat roja [%]		
0.07	128	256	RBF	100	98.3	100	98.3	1
			PSO	100	98.9	99.9	99.2	37
			UPSO	100	98.9	100	99.1	42
			GCPSO	99.7	98.8	99.9	99.1	25
			OPSO Cauchy	99.6	98.5	100	98.9	80
0.2	21	42	RBF	97.65	97.3	97.65	97.3	1
			PSO	99.12	98.1	98.63	98.9	63
			UPSO	98.73	97.7	99.12	98.5	71
			GCPSO	96.48	96.8	97.95	98.5	15
			OPSO Cauchy	95.12	97	99.22	98.5	76

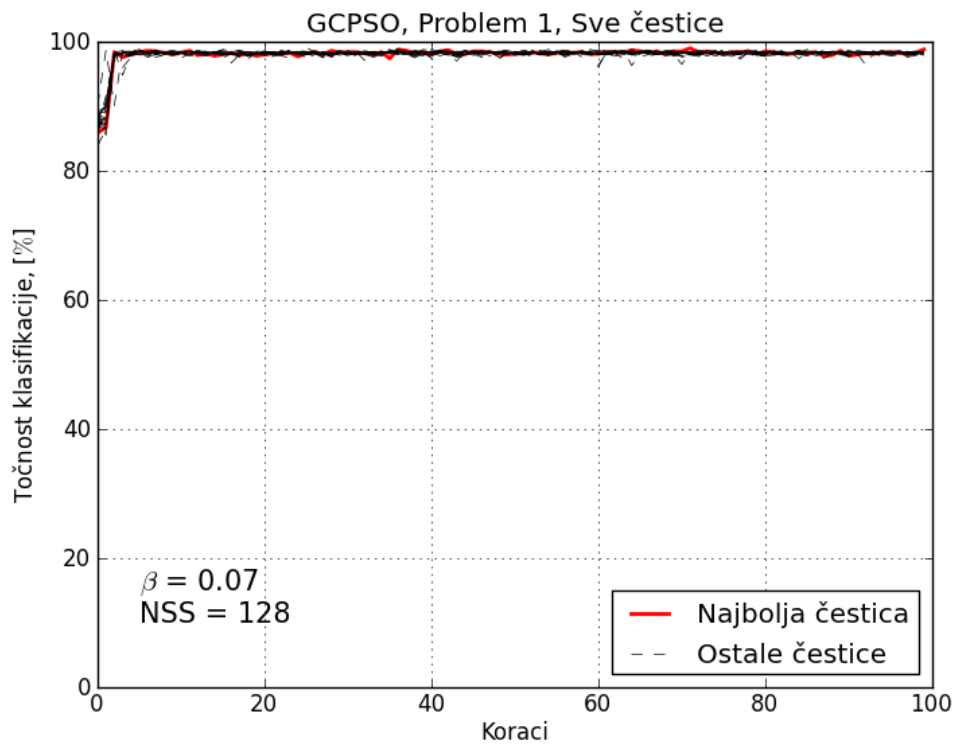
5.2.3. Grafički prikaz učenja algoritama uz $\beta=0.07$



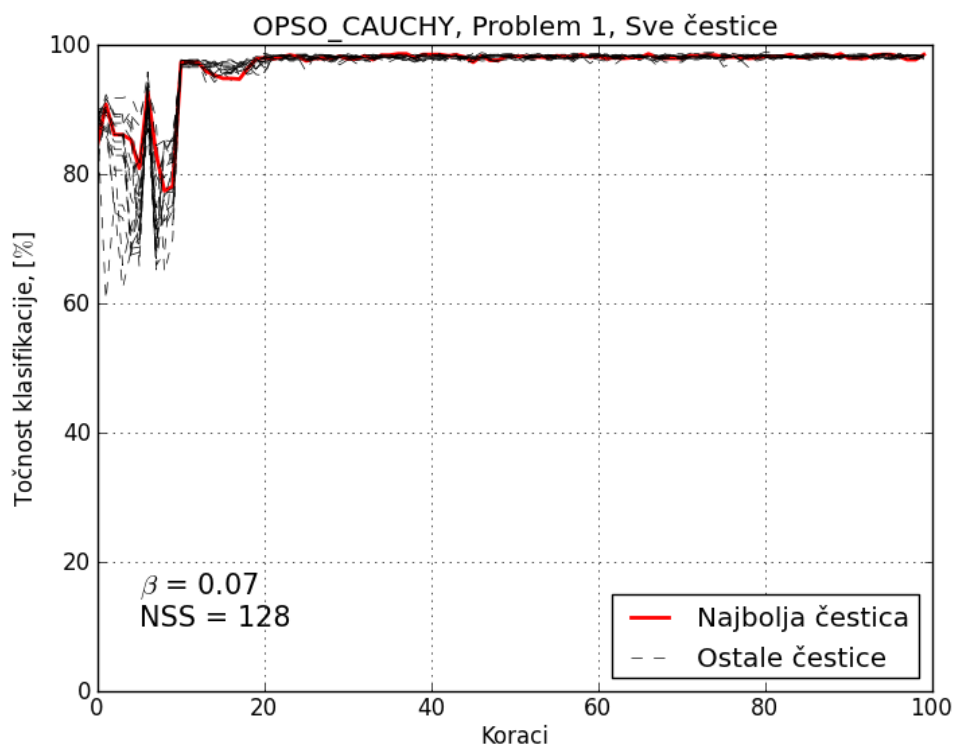
Slika 14. Grafički prikaz učenja PSO algoritma za problem 1 i $\beta = 0.07$



Slika 15. Grafički prikaz učenja UPSO algoritma za problem 1 i $\beta = 0.07$

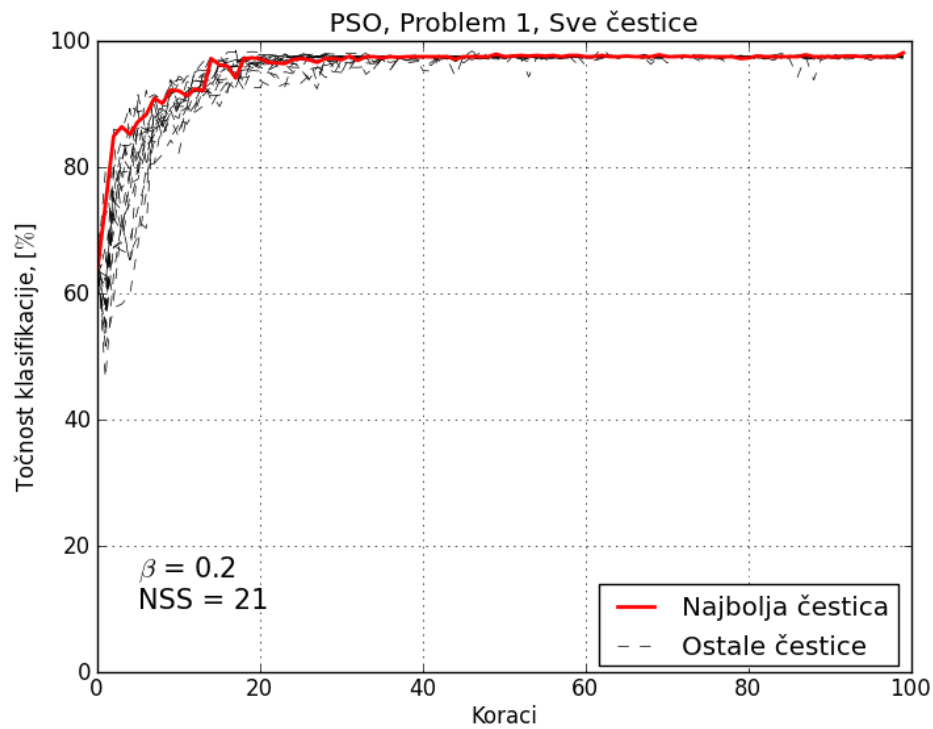


Slika 16. Grafički prikaz učenja GCP SO algoritma za problem 1 i $\beta = 0.07$

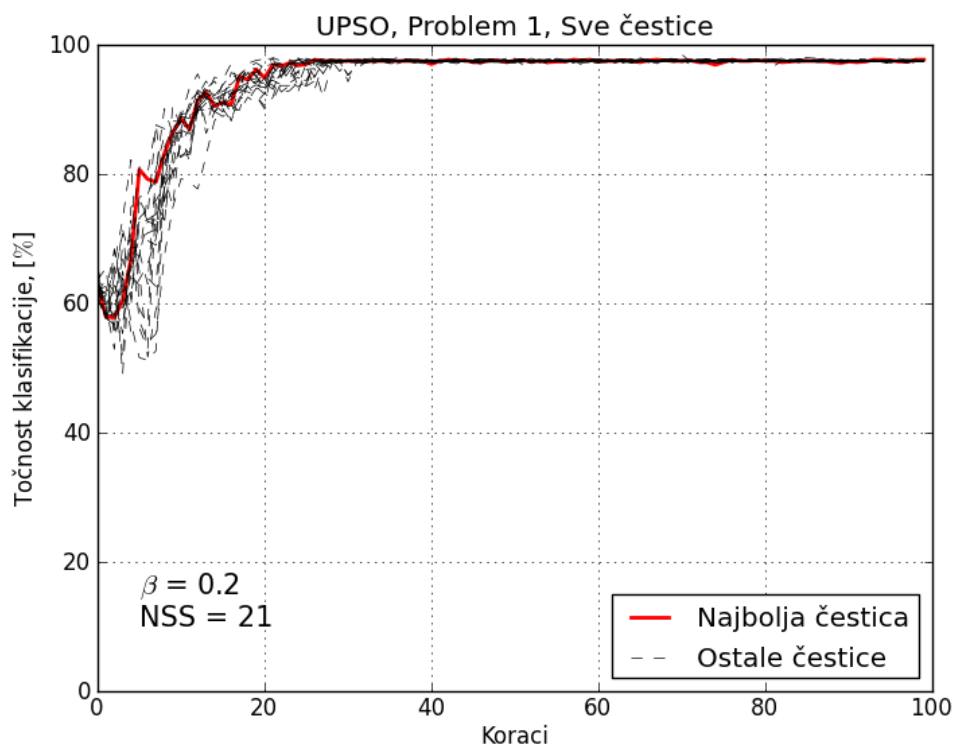


Slika 17. Grafički prikaz učenja OPSO Cauchy algoritma za problem 1 i $\beta = 0.07$

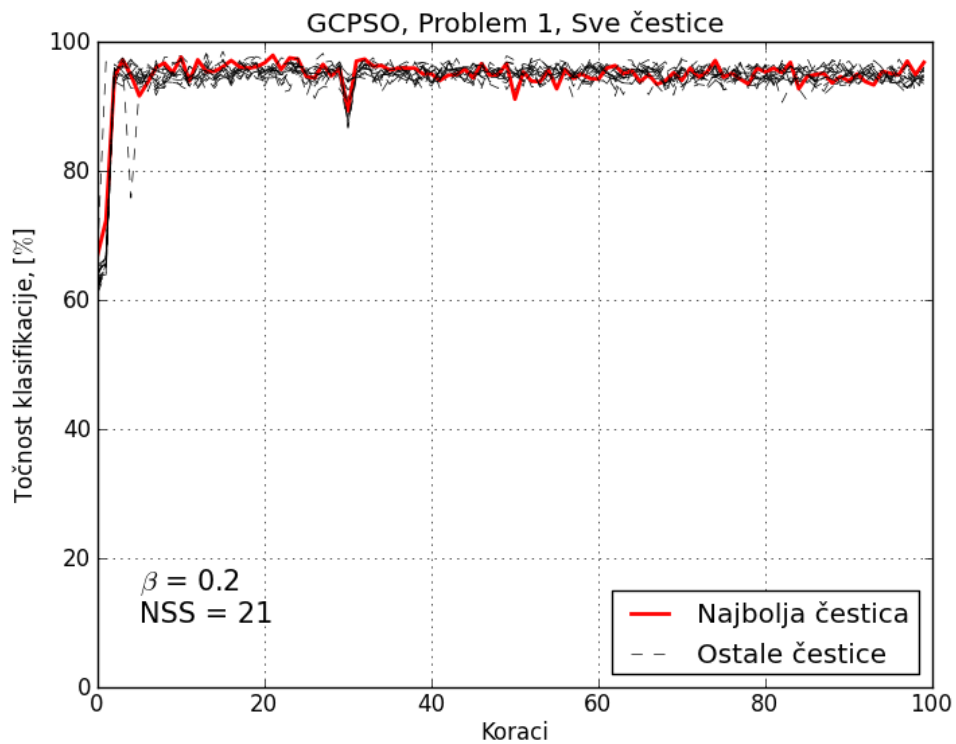
5.2.4. Grafički prikaz učenja algoritama uz $\beta=0.2$



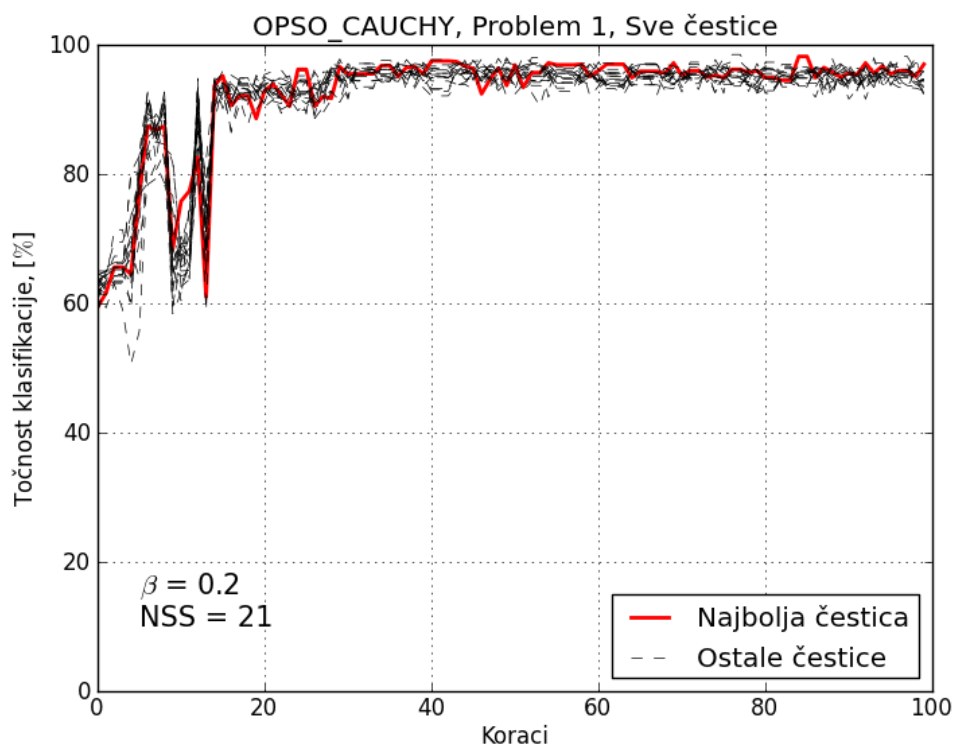
Slika 18. Grafički prikaz učenja PSO algoritma za problem 1 i $\beta = 0.2$



Slika 19. Grafički prikaz učenja UPSO algoritma za problem 1 i $\beta = 0.2$



Slika 20. Grafički prikaz učenja GCP SO algoritma za problem 1 i $\beta = 0.2$



Slika 21. Grafički prikaz učenja OP SO Cauchy algoritma za problem 1 i $\beta = 0.2$

5.3. Problem 2

5.3.1. Vrijednosti parametara algoritama za problem 2

Tablica 6. Vrijednosti parametara algoritama za problem 2

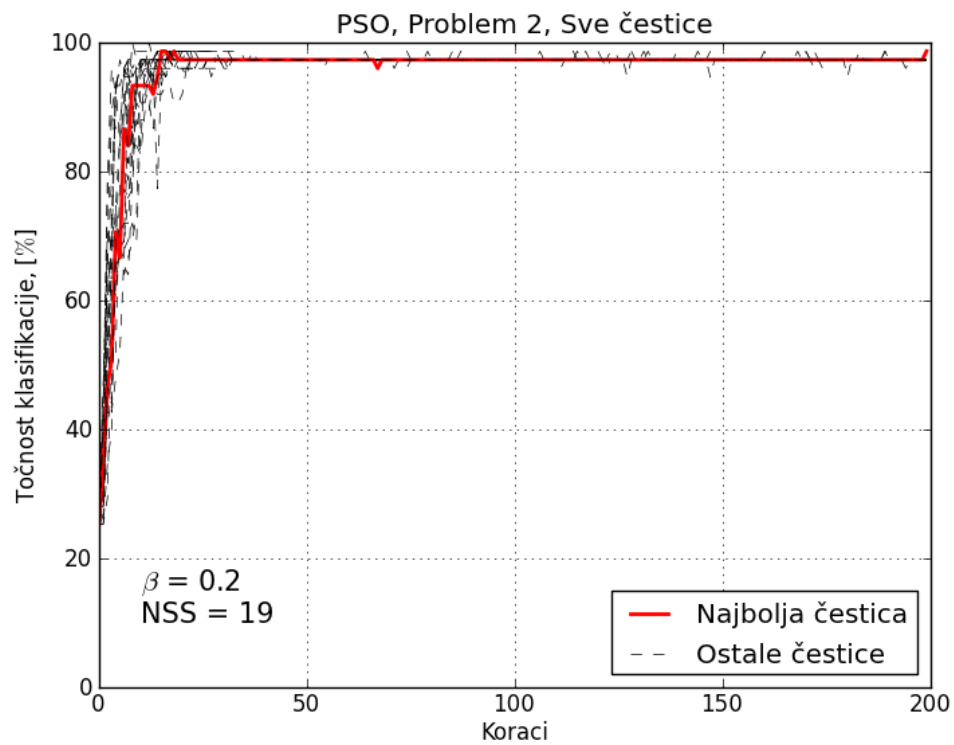
Broj koraka	200
Broj čestica	20
a	1
PSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
UPSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
u	0.3
GCPSO	
$\rho(0)$	1
$s_c(0)$	5
$f_c(0)$	5
OPSO Cauchy	
p_o	0.3

5.3.2. Tablični prikaz rezultata za problem 2

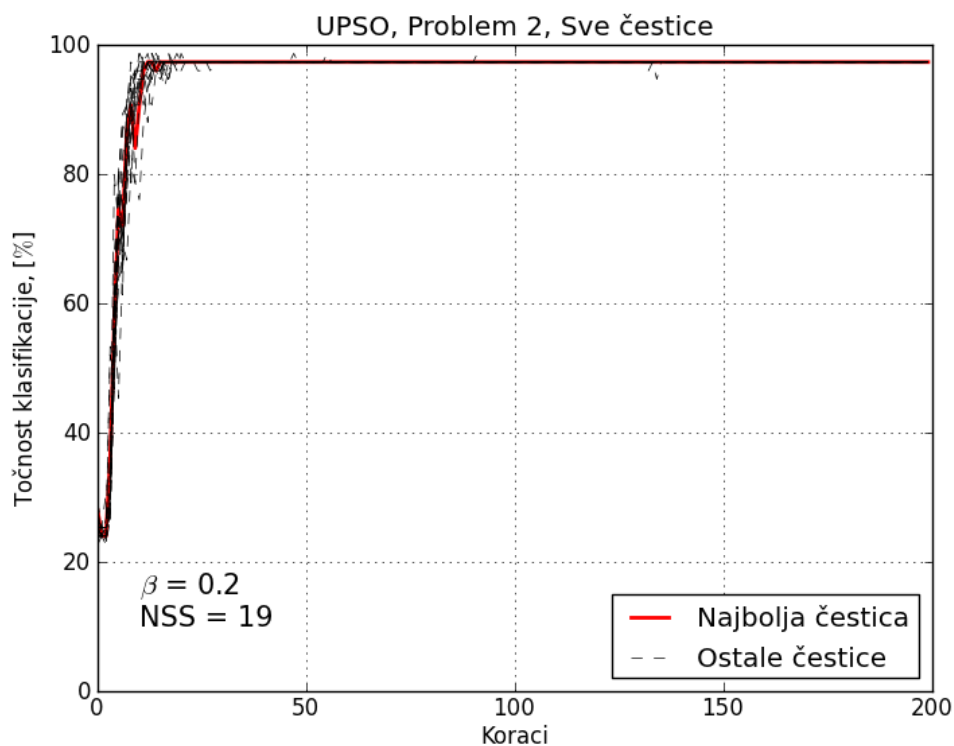
Tablica 7. Rezultati mreža za problem 2

β	NSS	param	MREŽA	učenje	test	učenje	test	Najbolji rezultat u koraku:
				Na kraju učenja [%]		Najbolji rezultat roja [%]		
0.2	19	76	RBF	98.66	97.33	98.66	97.33	1
			PSO	96	98.66	96	100	8
			UPSO	96	97.33	97.33	98.66	10
			GCPSO	97.33	98.66	96	100	11
			OPSO Cauchy	97.33	98.66	97.33	100	39
0.5	7	28	RBF	97.33	93.33	97.33	93.33	1
			PSO	96	93.33	93.33	100	45
			UPSO	96	93.33	90.66	98.66	7
			GCPSO	96	97.33	97.33	100	29
			OPSO Cauchy	96	97.33	89.33	98.33	148

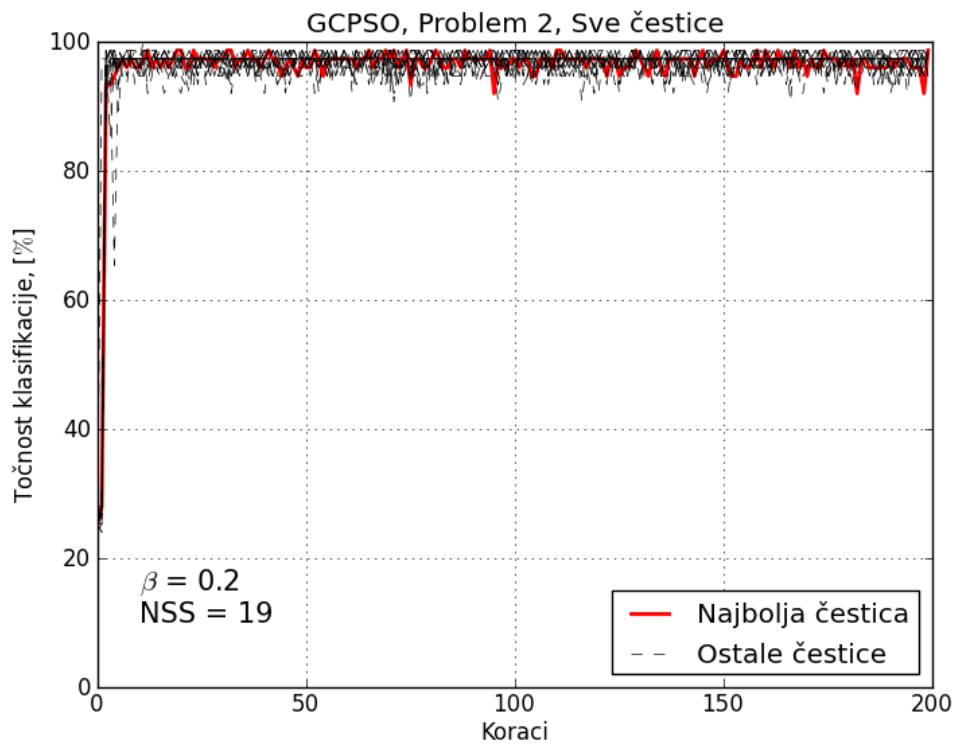
5.3.3. Grafički prikaz učenja algoritama uz $\beta=0.2$



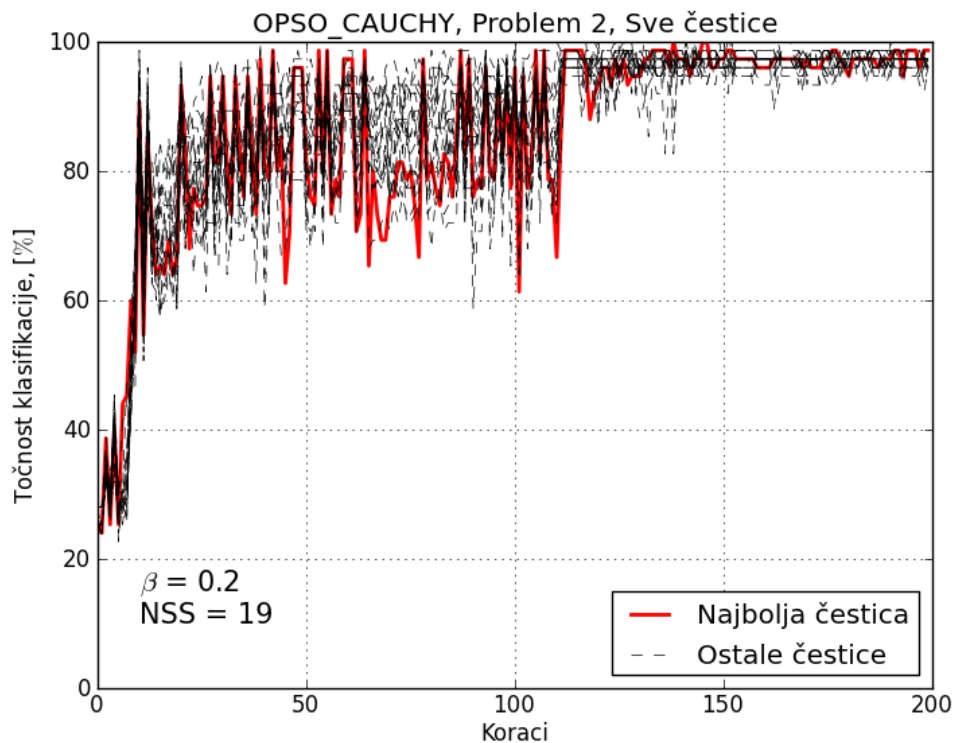
Slika 22. Grafički prikaz učenja PSO algoritma za problem 2 i $\beta = 0.2$



Slika 23. Grafički prikaz učenja UPSO algoritma za problem 2 i $\beta = 0.2$

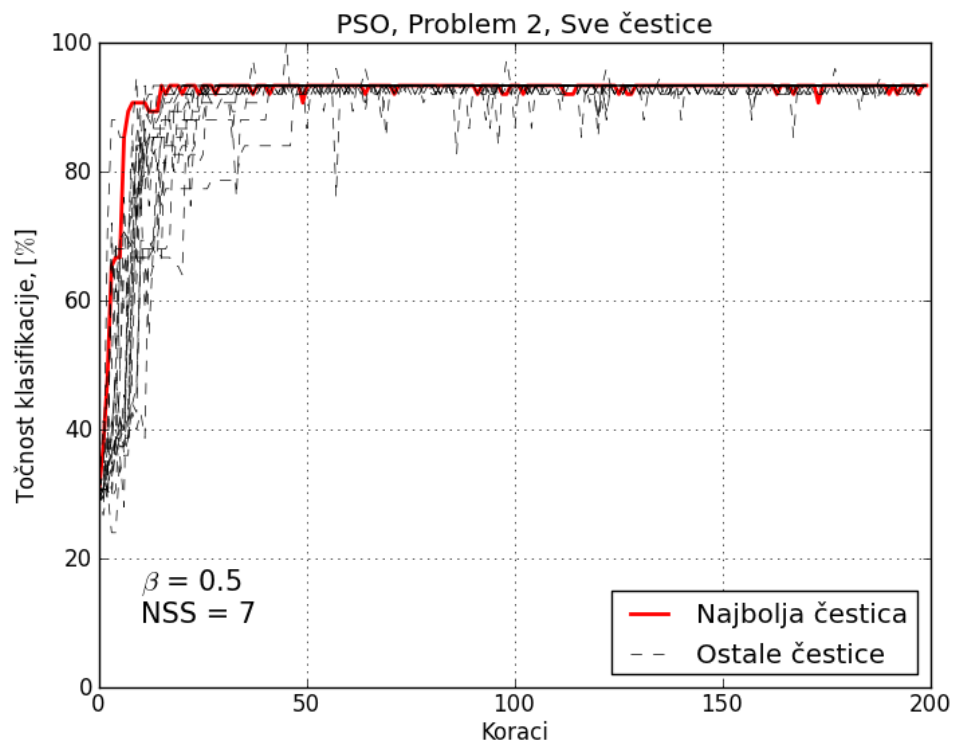


Slika 24. Grafički prikaz učenja GCPSO algoritma za problem 2 i $\beta = 0.2$

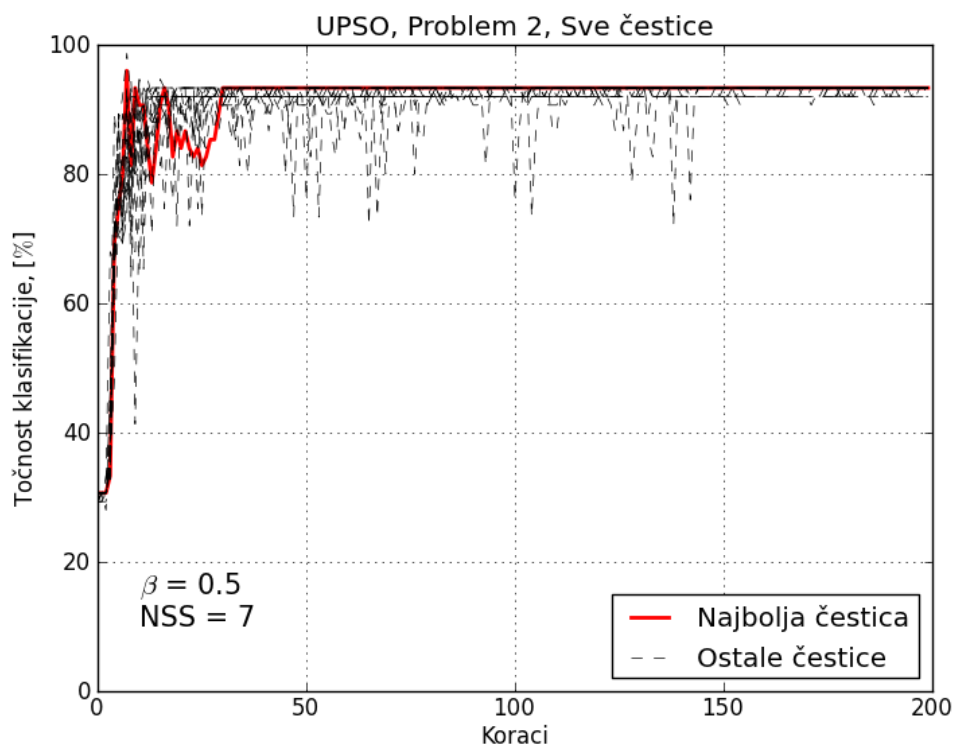


Slika 25. Grafički prikaz učenja OPSO Cauchy algoritma za problem 2 i $\beta = 0.2$

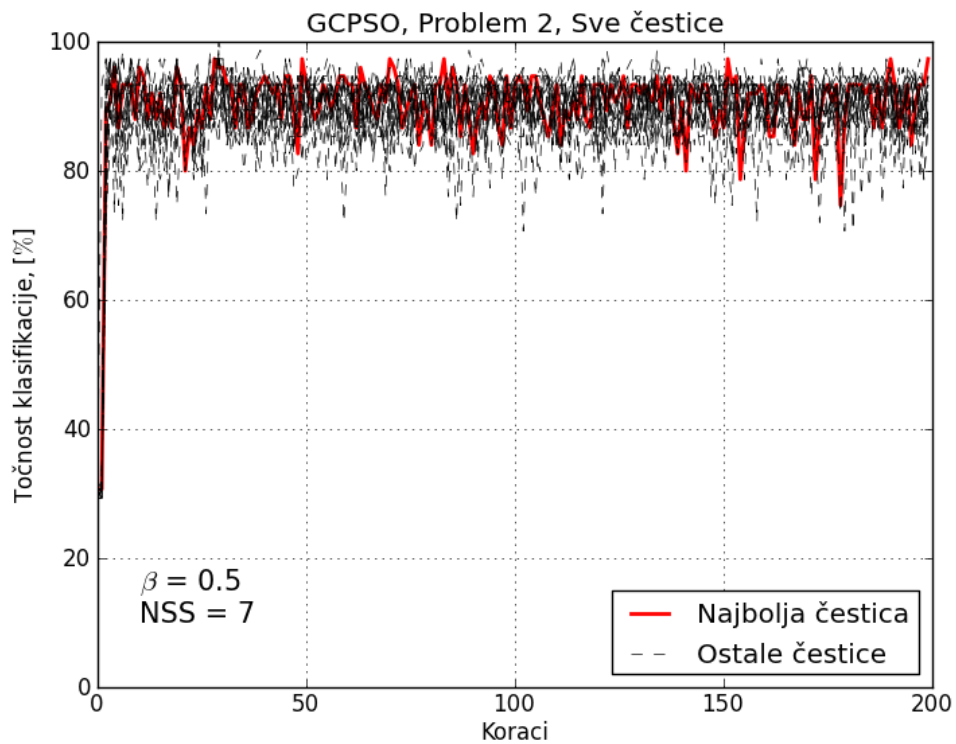
5.3.4. Grafički prikaz učenja algoritama uz $\beta=0.5$



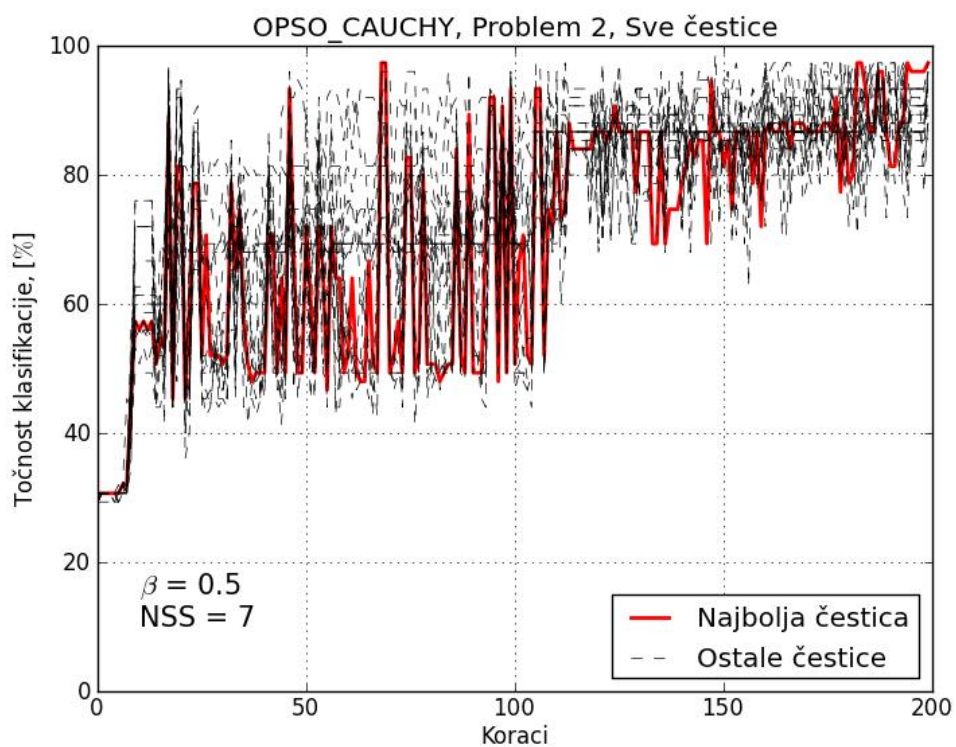
Slika 26. Grafički prikaz učenja PSO algoritma za problem 2 i $\beta = 0.5$



Slika 27. Grafički prikaz učenja UPSO algoritma za problem 2 i $\beta = 0.5$



Slika 28. Grafički prikaz učenja GCPSO algoritma za problem 2 i $\beta = 0.5$



Slika 29. Grafički prikaz učenja OPSO Cauchy algoritma za problem 2 i $\beta = 0.5$

5.4. Problem 3

5.4.1. Vrijednosti parametara algoritama za problem 3

Tablica 8. Vrijednosti parametara algoritama za problem 3

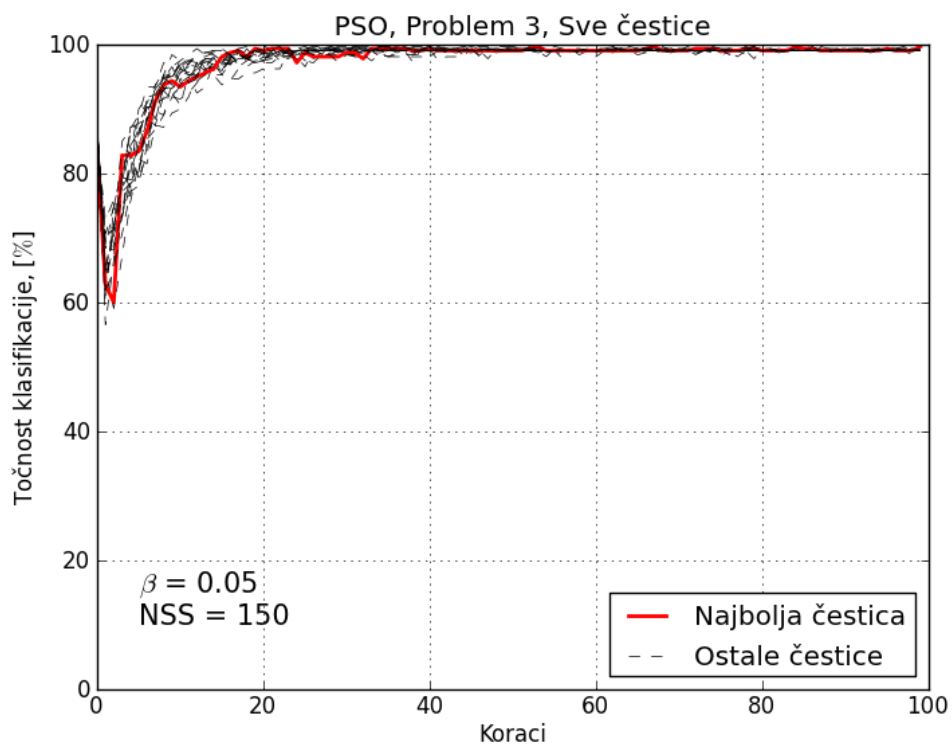
Broj koraka	100
Broj čestica	20
a	1
PSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
UPSO	
Broj susjednih čestica	2
c_1	2.1
c_2	2.1
u	0.3
GCPSO	
$\rho(0)$	1
$s_c(0)$	5
$f_c(0)$	5
OPSO Cauchy	
p_o	0.3

5.4.2. Tablični prikaz rezultata za problem 3

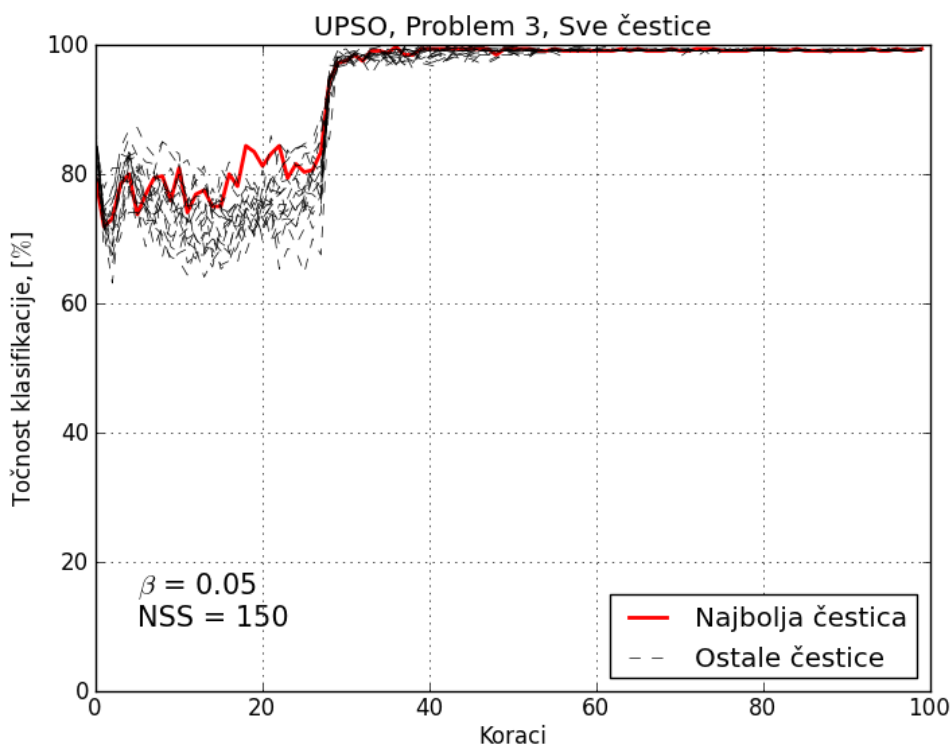
Tablica 9. Rezultati mreža za problem 3

β	NSS	param	MREŽA	učenje	test	učenje	test	Najbolji rezultat u koraku:
				Na kraju učenja [%]		Najbolji rezultat roja [%]		
0.05	150	750	RBF	100	86.88	100	86.88	1
			PSO	100	99.69	100	100	20
			UPSO	100	99.38	100	100	39
			GCPSO	99.06	99.69	100	100	32
			OPSO Cauchy	99.38	99.69	99.69	100	89
0.1	79	395	RBF	99.69	96.25	99.69	96.25	1
			PSO	99.37	99.37	99.06	99.68	22
			UPSO	99.06	99.38	98.75	99.69	35
			GCPSO	98.44	99.69	98.12	100	4
			OPSO Cauchy	98.75	99.69	96.56	100	21

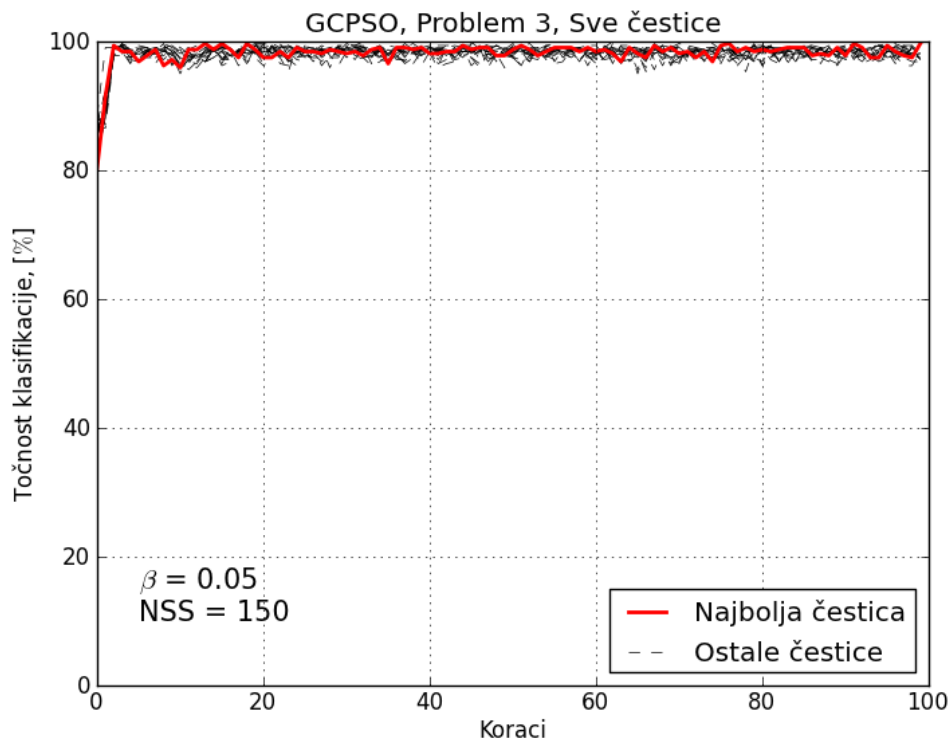
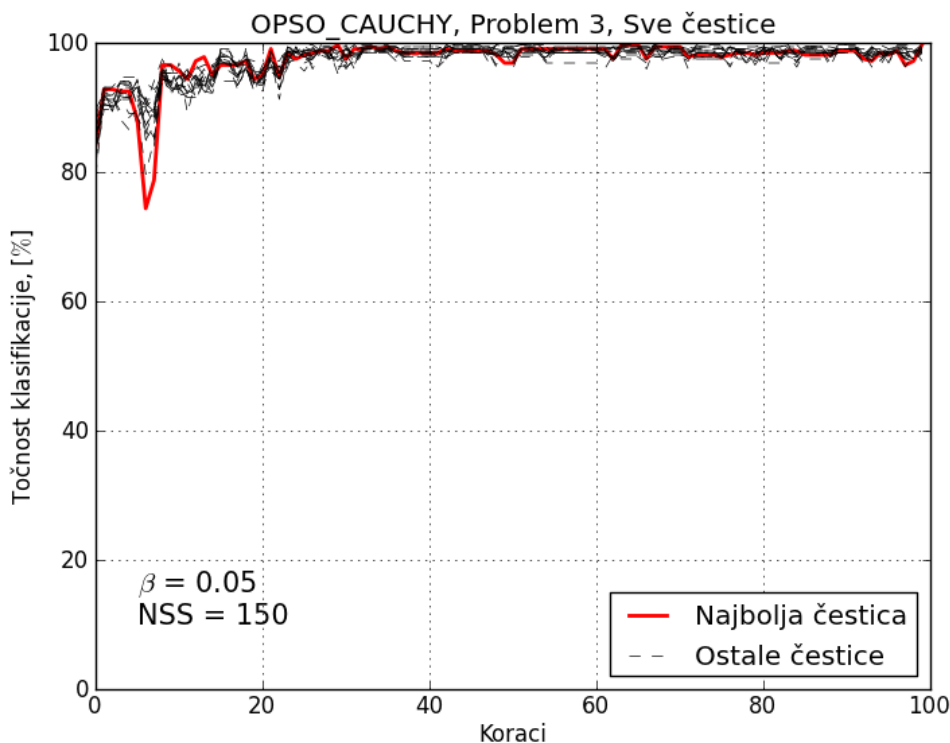
5.4.3. Grafički prikaz učenja algoritama uz $\beta=0.05$



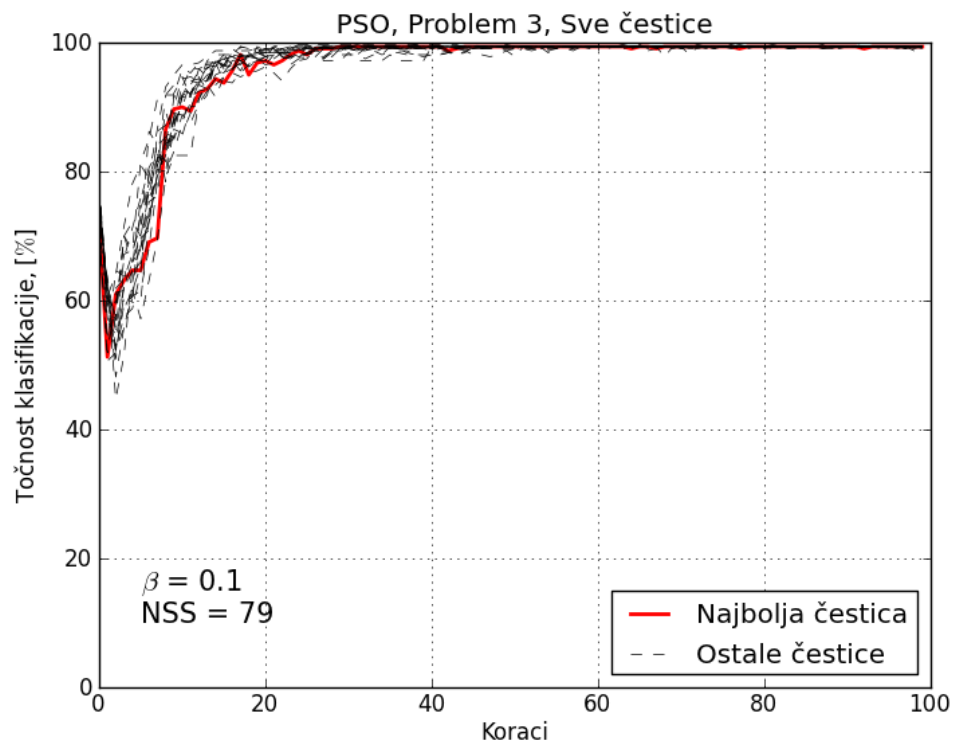
Slika 30. Grafički prikaz učenja PSO algoritma za problem 3 i $\beta = 0.05$



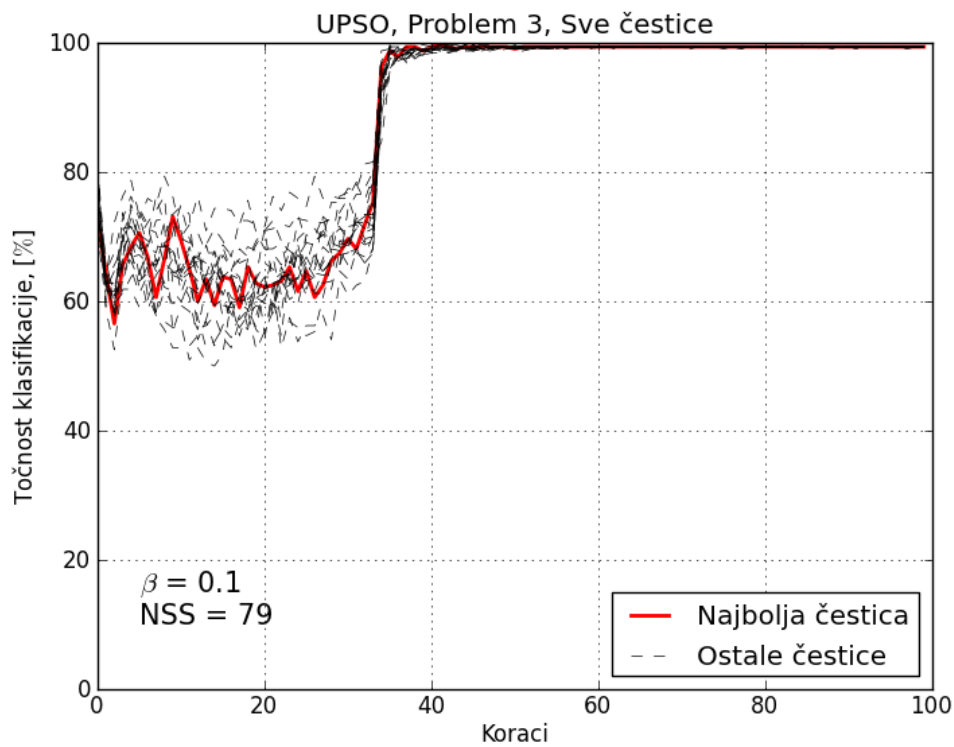
Slika 31. Grafički prikaz učenja UPSO algoritma za problem 3 i $\beta = 0.05$

Slika 32. Grafički prikaz učenja GCP SO algoritma za problem 3 i $\beta = 0.05$ Slika 33. Grafički prikaz učenja OPSO Cauchy algoritma za problem 3 i $\beta = 0.05$

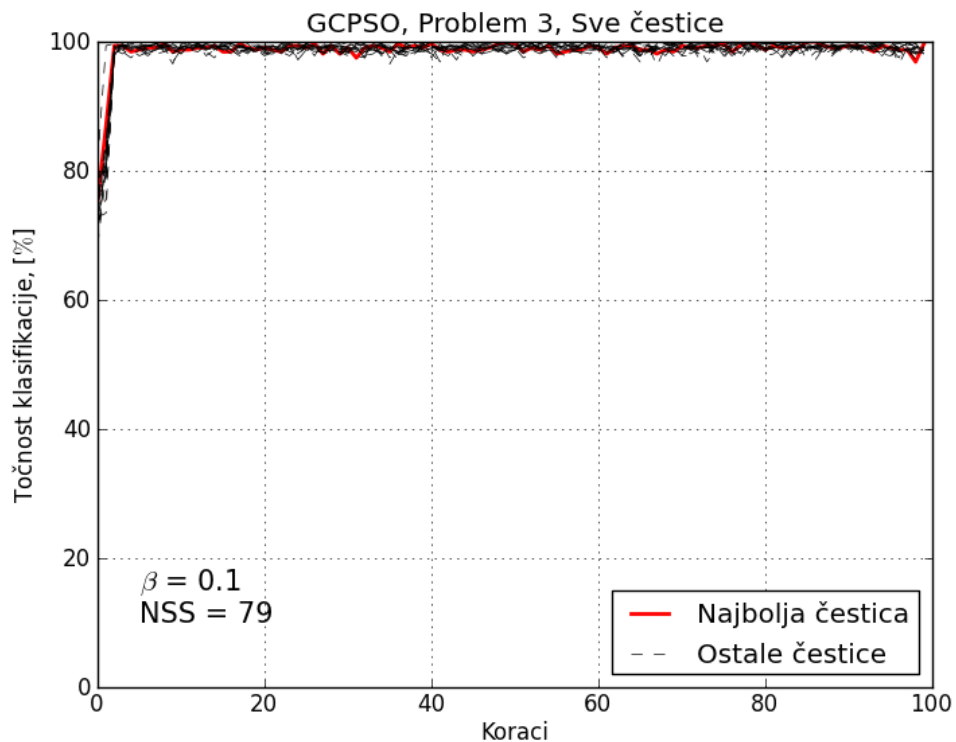
5.4.4. Grafički prikaz učenja algoritama uz $\beta=0.1$



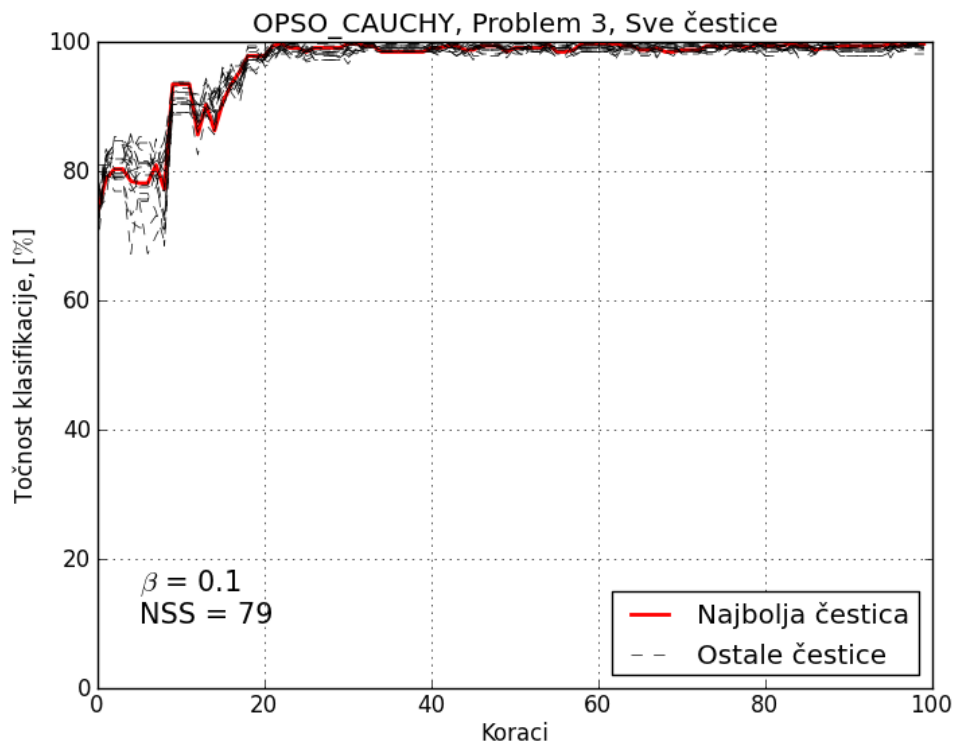
Slika 34. Grafički prikaz učenja PSO algoritma za problem 3 i $\beta = 0.1$



Slika 35. Grafički prikaz učenja UPSO algoritma za problem 3 i $\beta = 0.1$



Slika 36. Grafički prikaz učenja GCP SO algoritma za problem 3 i $\beta = 0.1$



Slika 37. Grafički prikaz učenja OPSO Cauchy algoritma za problem 3 i $\beta = 0.1$

5.5. Analiza rezultata

Iz tabličnih prikaza rezultata vidljivo je da su neuronske mreže proširene PSO algoritmom i njegovim modifikacijama pri rješavanju sva tri problema uspješno naučiti parametre kojima se klasifikacija uzoraka za testiranje obavlja bolje nego klasičnom metodom učenja RBF mreže u jednom koraku. Kod klasifikacije uzoraka za učenje klasična metoda je rješavajući probleme 2 i 3 ostvarivala bolje rezultate, dok je kod problema 1 sa većim iznosom parametra grupiranja ($\beta = 0.2$) svaki PSO algoritam pronašao bolje rješenje, a kod parametra grupiranja $\beta = 0.07$ rješenja algoritama su ostala unutar jedne desetine postotka razlike u odnosu na klasični pristup.

Objašnjenje za takve performanse nalazi se u funkciji cilja – uspješnost čestica roja PSO algoritama bio je postotak uspješno klasificiranih uzoraka za testiranje, dok je RBF mreža koja je koristila klasičnu metodu svoje parametre učila samo na temelju uzoraka za učenje.

Unatoč tome, neuronske mreže proširene PSO algoritmima nisu mnogo zaostajale u klasifikaciji uzoraka za učenje kod problema 1 i 3, dok se problem 2 pokazao kao veći izazov za optimizaciju parametara, te su se tu razlike s klasičnom metodom kretale i do 10%.

S druge strane, nezgodan iznos parametra grupiranja $\beta = 0.05$ kod problema 3 uzrokovao je velik pad generalizacijskih svojstava RBF mreže s klasičnom metodom učenja: dok su uzorci za učenje bili klasificirani savršeno, uzorci za testiranje su bili klasificirani s tek 86.88% uspješnosti. Uzrok tome može biti pretreniranost mreže (skriveni sloj je imao 150 neurona) ili, jednostavno, loše grupiranje uzoraka kod izračuna centara. Iznenađujuće, PSO algoritmi su u tek nekoliko desetaka koraka iteracije pronašli parametre koji savršeno, ili gotovo savršeno, klasificiraju oba seta uzoraka. Tu je najbrža bila RBF mreža proširena lokalnom varijantom PSO algoritma s 20 iteracija, dok je najlošiji, no svejedno impresivan rezultat postigao OPSO Cauchy algoritam s 99.69% uspješno klasificiranih uzoraka za učenje i 100% uspješno klasificiranih uzoraka za testiranje s 89 koraka iteracije. Zanimljivo je spomenuti da su čestice bile sastavljene od čak 750 parametara mreže, što je daleko više od ostalih testova, te se i općenito to može smatrati vrlo velikim problemom optimizacije za PSO algoritam.

Grafičkim rezultatima se dodatno proširuje analiza kvalitete PSO algoritama. Vidljivo je da se rojevi čestica s vrlo malo poteškoća mogu približiti zadovoljavajućim, te čak i superiornijim rješenjima u usporedbi s klasičnom metodom učenja RBF mreže. Efekt brze konvergencije je pogotovo vidljiv u problemima 1 i 3, pa tako GCPSO u trećem problemu, uz $\beta = 0.1$ već u četvrtom koraku iteracije pronalazi svoje najbolje i općenito vrlo kvalitetno

rješenje. U oba problema funkcija cilja koja ocjenjuje uspješnost klasifikacije gotovo odmah po početku učenja počinje rasti, te se nerijetko kod svih algoritama asimptotski približi najboljem rješenju u manje od 20 koraka iteracije. Najbolje rješenje koje roj pronade onda slijedi u nekoliko daljnjih koraka.

Kod rješavanja problema 2, ocjena kvalitete rješenja čestica roja se mijenja kaotičnije nego u ostalim problemima kroz korake iteracije. Uzrok tome može ležati u neoptimalnom odabiru parametara mreže, odnosno parametara PSO algoritama. Također, parametri koji se optimiziraju mogu utjecati na kvalitetu klasifikacije izrazito nelinearno, pa tako malena promjena vrijednosti parametra može imati drastičan utjecaj na kvalitetu rješenja. Problem 2 je jedini u kojem se mrežama dozvolilo da uče 200 koraka iteracije, upravo zbog navedenog nestabilnog ponašanja, pri čemu je OPSO Cauchy imao najizraženije nagle promjene u ocjeni kvalitete rješenja čestica roja. Unatoč tome, vidljivo je da je učenje i u tom slučaju konvergiralo kvalitetnom rješenju.

Zanimljivo je primijetiti utjecaj nasumičnih vrijednosti uključenih u PSO algoritam u pronalasku nenadano dobrih rješenja. To je posebno vidljivo kod lokalne varijante PSO algoritma, kod problema 2, uz $\beta = 0.5$ [Slika 26]. Nakon što je u nekoliko koraka cijeli roj konvergirao određenoj vrijednosti rješenja, brzine čestica su se vrlo brzo smirile, zatočivši pozicije čestica u toj vrijednosti. U osnovi, to je utjecaj komponente brzine koja privlači čestice najboljim rješenjima koje je roj ili susjedstvo pronašlo tokom učenja. Prisutnost nasumičnih vrijednosti u izrazu za brzine čestica je tada uočljiva u naglim promjenama ocjene funkcije cilja određenih čestica. Takvi „skokovi“ često vode lošijim rješenjima, a čestice brzo budu privučene natrag u roj, no ponekad se pronade i bolje rješenje, kao što je slučaj u navedenom primjeru. Analogija se može povući sa operacijom mutacije kod genetskih algoritama, gdje se nasumičnim odabirom mijenjaju vrijednosti jednog ili više parametara gena, čime se stvara prilika za pronalaskom boljeg rješenja.

Jedna od glavnih mana PSO algoritma je zasigurno vrijeme učenja. Za rješavanje sva tri problema generirano je 20 čestica, te se izvršilo 100 koraka učenja za probleme 1 i 3, te 200 koraka za problem 2. Kako se vrijednost funkcije cilja za određene vrijednosti parametara (pozicija čestice) može izračunati samo provođenjem učenja RBF mreže koristeći te parametre, jasno je da provođenje učenja može biti dulje i do nekoliko tisuća puta od klasične metode učenja u jednom koraku. OPSO Cauchy je pritom najsporiji algoritam zato jer uz pozicije čestica, u određenim koracima iteracije generira i suprotne čestice, čije se pozicije također moraju ocijeniti funkcijom cilja.

6. ZAKLJUČAK

Analizom rezultata o uspješnosti umjetne neuronske mreže s radijalnim baznim funkcijama (RBF mreže) u klasificiranju uzoraka tri različita problema, došlo se do sljedećih zaključaka.

RBF mreža proširena algoritmom optimizacije rojem čestica (PSO algoritam) sposobna je postići slične, te u nekim slučajevima i superiornije rezultate od iste mreže koja koristi klasičnu metodu učenja u jednom koraku. Upotreba algoritma optimizacije rojem čestica u učenju parametara mreže također je rezultirala boljim generalizacijskim svojstvima mreže u slučajevima nepovoljno grupiranih uzoraka za učenje. Analizirane modifikacije PSO algoritma, kao i njegova osnovna verzija pokazale su se uspješnima u optimiziranju i do nekoliko stotina parametara mreže, što sugerira da se PSO algoritam može nositi s velikim brojem parametara koji uobičajeno predstavljaju bazu znanja umjetne neuronske mreže. Također pokazalo se da PSO algoritam ne zahtjeva više od nekoliko desetaka koraka učenja kako bi postigao zadovoljavajuće rezultate, dakle algoritam se odlikuje brzom konvergencijom.

Glavna prednost PSO algoritma je to što zahtjeva malo, ili nimalo informacija o problemu optimizacije. Prilagodba algoritma za optimizaciju parametara mreže s drugačijom strukturom, ili optimizaciju drugačijih parametara iste mreže lako je ostvariva zbog matematičke jednostavnosti samog algoritma.

Najveća mana PSO algoritma dolazi do izražaja kod problema čija se kvaliteta rješenja ne može izračunati brzo. Algoritam počiva na ispitivanju većeg broja kandidatskih rješenja, no što je izračun rješenja skuplji, to je upotrebu PSO algoritma teže opravdati.

Daljnji napredak u upotrebi PSO algoritma u problemima učenja parametara mreže predlaže se na dva primjera. Noviji napori istraživača iz područja modifikacije i unaprjeđenja algoritma optimizacije roja čestica kreću se u smjeru dinamički promjenjivih veličina čestica. Promjenjivost broja parametara koji se optimiziraju tako može značiti mogućnost optimiziranja broja neurona skrivenog sloja neuronske mreže, čime se može minimizirati veličina mreže uz zadržavanje zadovoljavajuće kvalitete odaziva [13]. Također, neovisnost PSO algoritma o vrsti optimizacije može poslužiti u zamjeni algoritma učenja povratnim rasprostiranjem pogreške kod višeslojnih neuronskih mreža, gdje gradijentne metode pokazuju lošije performanse ovisno o broju skrivenih slojeva.

LITERATURA

- [1] Broomhead, D. S.; Lowe D.: Radial basis functions, multi-variable functional interpolation and adaptive networks (Technical report), RSRE. 4148, 1988.
- [2] Broomhead, D. S.; Lowe D.: Multivariable functional interpolation and adaptive networks, *Complex Systems* 2: str. 321-355, 1988.
- [3] Novaković, B.; Majetić D.; Široki M.: Umjetne neuronske mreže, Fakultet strojarstva i brodogradnje, Zagreb, 1998.
- [4] LAPACK, službene stranice: <http://www.netlib.org/lapack/>
- [5] Brezak, D. et al.: Tool Wear Monitoring Using Radial Basis Function Neural Network, *Proceedings of International Joint Conference on Neural Networks – IJCNN*, Budapest, Hungary, str. 1859-1863, 2004.
- [6] Kennedy, J.; Eberhart, R.: Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, str.. 1942–1948., 1995.
- [7] Parsopoulos, K. E.; Vrahatis, M. N.: UPSO: A unified particle swarm optimization scheme, *Lecture Series on Computer and Computational Sciences*, Vol. 1, VSP International Science, 2004.
- [8] Carvalho, M.; Ludermir T. B.: An Analysis Of PSO Hybrid Algorithms For Feed-Forward Neural Networks Training, *Proceedings of the Ninth Brazilian Symposium on Neural Networks*, 2006.
- [9] Wang H. et al.: Opposition-based Particle Swarm Algorithm with Cauchy Mutation, *IEEE Congress on Evolutionary Computation (CEC 2007)*, str.. 4750-4756, 2007.
- [10] Fisher, R. A.: The use of multiple measurements in taxonomic problems, *Annals of Eugenics* 7 (2), str. 179-188, 1936.
- [11] Zoričić, M. : Diplomski rad, Fakultet strojarstva i brodogradnje, Zagreb, 2015.
- [12] Python Software Foundation. Python Language Reference, verzija 2.7. dostupna na službenim stranicama: <http://www.python.org>
- [13] Zhao, J.; Li, M.; Wang, Z.: Fuzzy Neural Networks Learning by Variable-dimensional Quantum-behaved Particle Swarm Optimization, *TELKOMNIKA*, Vol. 11, No. 10, str. 6216-6223, 2013.