

ŠOLSKI CENTER VELENJE
ELEKTRO IN RAČUNALNIŠKA ŠOLA
Trg mladosti 3, 3320 Velenje

MLADI RAZISKOVALCI ZA RAZVOJ ŠALEŠKE DOLINE

RAZISKOVALNA NALOGA

PREIZKUŠANJE GENETSKEGA ALGORITMA NEAT

Tematsko področje: RAČUNALNIŠTVO

Avtorja:

Andraž Podpečan, 4. letnik

Rok Žerdoner, 4. letnik

Mentorja:

Samo Železnik

Islam Mušić, prof.

Velenje, 2021

Raziskovalna naloga je bila opravljena na Elektro in računalniški šoli, Šolski Center Velenje.

Mentorja: Samo Železnik,
Islam Mušić, prof.

Datum predstavitve: april 2021

KLJUČNA DOKUMENTACIJSKA INFORMACIJA

- ŠD Šolski Center Velenje, šolsko leto 2020/2021
- KG strojno učenje / umetna nevalna omrežja / genetski algoritem / evolucijski algoritem
- AV PODPEČAN, Andraž / ŽERDONER, Rok
- SA ŽELEZNIK, Samo / MUŠIČ, Islam
- KZ 3320 Velenje, SLO, Trg Mladosti 3
- ZA Šolski Center Velenje
- LI 2021
- IN **PREIZKUŠANJE GENETSKEGA ALGORITMA NEAT**
- TD Raziskovalna naloga
- OP VIII, 29 str., 22 graf., 4 sl., 6 vir.
- IJ SL
- JI sl / en
- IJ NEAT (NeuroEvolution of Augmenting Topologies) je genetski algoritem za ustvarjanje umetnih nevalnih omrežij po evolucijskem postopku. Namen te raziskovalne naloge je bilo ugotoviti, kako parametri algoritma vplivajo na njegovo delovanje in kako se algoritem obnese pri ustvarjanju nevalnega omrežja za vodenje preprostega tekača skozi poligon z ovirami in nagradami. Rezultati niso pokazali močne povezave med fitnessom (ovrednotenjem delovanja nevalnega omrežja) in spremembami posameznih parametrov. Ta je bila izrazitejša pri merjenju kompleksnosti (število vseh povezav v nevalnem omrežju), na katero so imeli največ vpliva trije glavni parametri: delež začetnih povezav, delež mutacij in delež medvrstnega križanja. Vsi trije so lahko pospešili ali upočasnili rast kompleksnosti. Preizkušanje zmožnosti algoritma je pokazalo, da se nevalna omrežja za dano nalogo razvijajo zelo počasi. V 150 generacijah so dosegla le malo napredka in še ta je bil brez velikega pomena, saj z opazovanjem v njihovem delovanju ni bilo mogoče zaslediti sledi inteligence. Preizkus se je izvedel enkrat na stalno enakem poligonu ter drugič na dveh naključno izbranih poligonih in v obeh primerih je bilo očitno, da bi razvoj delujočega omrežja porabil ogromno časa. Tako je nastal zaključek, da NEAT ni najbolj primeren za nalogo v tej raziskavi.

KEY WORDS DOCUMENTATION

ND School Centre Velenje, school year: 2020/2021

CX machine learning / artificial neural networks / genetic algorithm / evolutionary algorithm

AU PODPEČAN, Andraž / ŽERDONER, Rok

AA ŽELEZNIK, Samo / MUŠIČ, Islam

PP 3320 Velenje, SLO, Trg Mladosti 3

PB School Centre Velenje

PY 2021

TI **TESTING THE GENETIC ALGORITHM NEAT**

DT Research work

NO VIII, 29 p., 22 graf., 4 fig., 6 ref.

LA SL

AL sl / en

AB NEAT (NeuroEvolution of Augmenting Topologies) is a genetic algorithm for creation of artificial neural networks by a neuroevolutionary method. The purpose of this research was to find out how the algorithm's parameters affect its function and how well does it perform in creating a neural network for navigating a simple runner through a maze filled with obstacles and rewards. The results did not show a strong connection between fitness (the evaluation of the neural networks' performances) and changes in individual parameters. This one was more apparent in the measurements of complexity (number of all connections in a neural network), where three specific parameters had the greatest effect: initial connection proportion, mutation proportions, and interspecies proportion, all three of which could accelerate or decelerate the rise of complexity. The test of the algorithm's performance showed that the neural networks are adapting very slowly to this specific task. They made miniscule progress in 150 generations and even that had little meaning, as observations of the runners' behaviors showed no sign of intelligence in their actions. The test was performed once on a set maze and once again on two randomly picked mazes and in it became clear in both cases, that the development of a functional network would consume enormous quantities of time. And so, a conclusion was formed that NEAT is not the most suitable for the task in this research.

KAZALO VSEBINE

1.	UVOD	1
1.1	OPIS NEVRALNIH OMREŽIJ	1
1.1.1	HIPOTEZE:	1
2.	PREGLED OBJAV	2
2.1	Dosedanji genetski algoritmi	2
2.2	NEAT	2
2.3	OBJAVLJENI PREIZKUSI neat-algoritma	3
2.3.1	XOR-vrata	3
2.3.2	uravnovešenje palice	4
3.	METODOLOGIJA	5
3.1	RAZVOJ EKSPERIMENTA	5
3.2	UPORABA ORODJA - GODOT	5
3.3	UPORABLJENE KOMPONENTE	6
3.3.1	TEKAČ	6
3.3.2	SESTAVNI DELI POLIGONA	7
3.3.3	EKSPERIMENT	7
3.4	POTEK IZDELAVE	8
3.4.1	Izdelava tekača	8
3.4.2	vkjučitev neat	9
3.4.3	IZDELAVA CELOSTNEGA SISTEMA ZA PREIZKUŠANJE	9
3.5	PREIZKUS Algoritma	9
4.	REZULTATI	13
4.1	PREIZKUS POSAMIČNIH PARAMETROV	13
4.1.1	VELIKOST POPULACIJE	13
4.1.2	ŠTEVILO VRST	14
4.1.3	DELEŽ ELITNIH PRIMERKOV	15
4.1.4	DELEŽ IZBRANIH PRIMERKOV	16
4.1.5	DELEŽ SPOLNEGA RAZMNOŽEVANJA	17
4.1.6	DELEŽ MEDVRSTNEGA RAZMNOŽEVANJA	18

4.1.7	RAZMERJE MUTACIJ	19
4.1.8	DELEŽ ZAČETNIH POVEZAV	20
4.2	STALNI IN NAKLJUČNI POLIGON	21
5.	RAZPRAVA	22
5.1	TRENDI RASTI FITNESA	22
5.2	Vpliv parametrov	23
5.3	KOMPLEKSNOŠT	24
5.4	STALNI IN NAKLJUČNI POLIGON	26
6.	ZAKLJUČKI	27
7.	POVZETEK	28
8.	SUMMARY	29
9.	VIRI	30
9.1	ELEKTRONSKI VIRI	30
9.2	KNJIŽNICA NEAT	30

KAZALO SLIK

Slika 1: Tekaç. Moder kvadrat je njegovo telo, sive črte pa so žarki, ki mu omogočajo vid. (foto: Andraž Podpečan).....	6
Slika 2: Tekaçi na poligonu (foto. Andraž Podpečan).....	8
Slika 3: Prvi preizkusni poligon. (foto: Andraž Podpečan).....	11
Slika 4: Drugi preizkusni poligon. (foto: Andraž Podpečan).....	11

KAZALO GRAFOV

Graf 1: Največji fitnessi pri preizkušanju velikosti populacije.....	13
Graf 2: Povprečni fitnessi pri preizkušanju velikosti populacije.....	13
Graf 3: Največji fitnessi pri preizkušanju števil vrst.....	14
Graf 4: Povprečni fitnessi pri preizkušanju števila vrst.....	14
Graf 5: Največji fitnessi pri preizkušanju deleža elitnih primerkov.....	15
Graf 6: Povprečni fitnessi pri preizkušanju deleža elitnih primerkov.....	15
Graf 7: Največji fitnessi pri preizkušanju deleža izbranih primerkov.....	16
Graf 8: Povprečni fitnessi pri preizkušanju deleža izbranih primerkov.....	16
Graf 9: Največji fitnessi pri preizkušanju deleža spolnega razmnoževanja.....	17
Graf 10: Povprečni fitnessi pri preizkušanju deleža spolnega razmnoževanja.....	17
Graf 11: Največji fitnessi pri preizkušanju deleža medvrstnega križanja.....	18
Graf 12: Povprečni fitnessi pri preizkušanju deleža medvrstnega križanja.....	18
Graf 13: Največji fitnessi pri preizkušanju razmerij mutacij.....	19
Graf 14: Povprečni fitnessi pri preizkušanju razmerij mutacij.....	19
Graf 15: Največji fitnessi pri preizkušanju deleža začetnih povezav.....	20
Graf 16: Povprečni fitnessi pri preizkušanju deleža začetnih povezav.....	20
Graf 17: Meritve fitnessa preizkusa z enim poligonom.....	21
Graf 18: Meritve fitnessa preizkusa z dvema poligonoma.....	21
Graf 19: Največji in povprečni fitness pri preizkusu z samo eno vrsto.....	22
Graf 20: Meritve fitnessa za 75-odstotni delež začetnih povezav.....	23
Graf 21: Največje kompleksnosti za različne nastavitve mutacij.....	25
Graf 22: Največje kompleksnosti za različne nastavitve deleža medvrstnega razmnoževanja.....	25

SEZNAM OKRAJŠAV

NEAT - NeuroEvolution of Augmenting Topologies

t. i. – tako imenovan

MIT - Prosta in odprtokodna licenca za programsko opremo, ki izvira iz Tehnološkega inštituta Massachusettsa

Npr. - na primer

Oz. - oziroma

1. UVOD

1.1 OPIS NEVRALNIH OMREŽIJ

Temelj vsega strojnega učenja so umetna nevrnalna omrežja. Njihove prednosti so še posebej opazne pri zapletenejših nalogah, kjer bi bilo načrtovanje algoritmov prezahtevno in konec koncev tudi neučinkovito. Imajo pa umetna nevrnalna omrežja tudi eno ključno pomanjkljivost: težko jih je usposobiti za njihov zadani namen. Pri uporabi umetnega nevrnalnega omrežja se ukvarjamo samo z njegovimi vhodnimi in izhodnimi podatki, kako pa iz prvih dobimo slednje, pa je nemogoče predvideti. Omrežje je treba naučiti, kako naj opravlja svojo nalogo, kar pa glede na tip le-te zahteva različne pristope.

Namen te raziskovalne naloge je ugotoviti, ali se lahko genetski algoritem NEAT nauči poiskati pot skozi progo z ovirami ter pridobiti podatke, s katerimi lahko določimo, od katerega parametra je najbolj odvisen napredek genetskega algoritma.

1.1.1 HIPOTEZE:

V fazi načrtovanja raziskovalne naloge sva določila tri hipoteze:

- Algoritem bo nalogo opravil bolje, bodo poligoni izbrani naključno.
- Razvoj nevrnalnih omrežij bo viden iz generacije v generacijo.
- Kompleksnost nevrnalnih omrežij ni neposredni pokazatelj njihovih uporabnosti za nalogo.

2. PREGLED OBJAV

2.1 DOSEDANJI GENETSKI ALGORITMI

Genetski algoritmi so naključni iskalni algoritmi, ki poskušajo posnemati načelo preživetja najmočnejšega in najboljšega, ki ga ves čas srečamo pri živih bitij v naravi. Ti algoritmi delujejo na nizih struktur, med katerimi je tudi biološka. Z njimi se razvija algoritem tako, da v vsaki novi generaciji naključno sestavlja nevralne povezave, ki jih pridobi od najuspešnejših osebkov prejšnje generacije. To počenja tako dolgo, dokler ti osebki ne dosežejo zelenega rezultata ali pa ga sami končamo.

Ti algoritmi delujejo na tri različne načine. Prvi način je selekcija, s katerimi potomci podedujejo gene najboljših osebkov prejšnje generacije. Naslednji je prepletenost, pri kateri se geni osebkov te generacije prepletajo med sabo v genih osebkov naslednjih generacij. Enostavno lahko povemo, da ti osebki posnemajo parjenje oziroma razmnoževanje živih bitij v naravi. Tretji ter zadnji način pa so mutacije. To so spremembe ali nepravilnosti, ki se občasno pripetijo pri prenosu genov na naslednjo generacijo. Mutacije so lahko dobre, saj lahko imajo ti osebki večjo možnost za preživetje v nekem okolju. V naravi lahko to zasledimo kot spremembo barve kože pri živalih. Lahko pa so tudi negativne, saj je možno, da se razvoj teh osebkov močno zaustavi.

Najboljši genetski algoritmi vsebujejo vse tri načine razvoja osebkov. Zaradi tega lahko s temi algoritmi pridobimo tudi najnatančnejše rezultate, saj najbolje posnemajo razvoje osebkov, prav tako pa tudi upoštevajo možnosti anomalij, ki se lahko pripetijo.

2.2 NEAT

Eden od teh algoritmov je NEAT, ki ga je leta 2002 razvil Kenneth O. Stanley na univerzi v Teksasu. Za razliko od običajnih nevralnih omrežij NEAT nima ene nespremenljivo vnaprej določene topologije, temveč se ta razvija postopoma med samim izvajanjem. Omrežje na začetku sestoji samo iz vhodnih in izhodnih nevronov, ki imajo že vnaprej določeno število povezav med njimi. Tekom učenja se ustvarjajo nove povezave, ali pa se k obstoječim nevronom dodajajo nove. Prednost tega je ta, da omrežje prej doseže uporabno stanje, zaradi česar pa je učinkovitejše od ostalih nevralnih omrežij.

Druge prednosti NEAT-a so razvrščanje genotipov omrežij v vrste in sledenje genetskim spremembam. Razvrščanje v vrste ohranja genetsko pestrost in omogoča slabše ocenjenim genotipom, da se izboljšajo. Sledenje genetskim spremembam skrbi za to, da se lahko genotipi križajo med sabo brez možnosti napak, ki bi privedle do neuporabnega potomca.

2.3 OBJAVLJENI PREIZKUSI NEAT-ALGORITMA

Pri testiranju NEAT-algoritma nas zanima, če lahko NEAT razvije potrebne struktur in če lahko najde učinkovitejše načine kot ostali nevroevolucijski sistemi. Odgovor na prvo vprašanje lahko dobimo s pomočjo XOR-omrežja, medtem ko za drugo pa s pomočjo testa, imenovanega »balansiranje palice«. Pri obeh testih so bili začetni parametri enaki, razlika je bila samo v številu populacije zaradi večje težavnosti drugega testa.

Vsi testi so imeli 150 NEAT-omrežij (z izjemo enega, ki je imel 1000), medtem ko so bili ostali parametri namenjeni nastavljanju soodvisnosti med omrežji in so bili enaki za vse.

Testi so imeli tudi nekatere omejitve. Te so bile:

- V primeru, da se osebki niso izboljšali po 15 generacijah, se niso smeli več reproducirati.
- Zmagovalec vsake vrste z več kot petimi omrežji je bil brez sprememb kopiran v naslednjo generacijo.
- Geni so imeli 80 % možnost za mutacijo, pri čemer je bila 10 % možnost za dodajanje povsem nove vrednosti k parametru ter 90 %, da se ni nič spremenilo.
- Medvrstno parjenje je imelo 0,001 možnost, medtem ko dodajanje novega gena 0,03 in dodajanje nove vrste mutacije 0,5.

2.3.1 XOR-VRATA

Pri tem testu je moral algoritem ustvariti skrivno skupino s točno določenimi vnesenimi podatki. Namen tega je bilo dokazati, da pri tem algoritmu ni skrbi, da bi se nove mreže zelo razlikovale od starih, ali pa bi se ustvaril elitni primerek z napačnimi povezavami.

Test je pokazal, da v povprečju stotih izvedb najde algoritem pravo vrednost pri 32. generaciji. V povprečju je imela mreža z rešitvijo 2,35 skrivnih ter 7.48 poznanih

povezljivih genov. Pomemben je tudi podatek, da algoritem ni niti enkrat nedokončal naloge (najslabši rezultat je zabeležil pri 90. generaciji). Ta podatek lahko priča o konstantnosti tega algoritma.

2.3.2 URAVNOVEŠENJE PALICE

Pri temu eksperimentu je bil glavni cilj vzdrževati palico na sredini vozička, medtem ko se je ta premikal. Pri dvojnemu balansiranju palice s hitrostjo je bilo potrebno balansirati palico za 100.000 korakov. NEAT-algoritmu je to v povprečju uspelo pri 24. generaciji. Nato so isti test ponovili brez hitrosti in je NEAT opravil preizkus po 286. generaciji. Je pa potrebno poudariti, da je NEAT opravil preizkus v najmanjšem številu evalvacij osebkov, s čimer je potrdil svojo zmago nad ostalimi algoritmi pri tem preizkusu.

3. METODOLOGIJA

3.1 RAZVOJ EKSPERIMENTA

Začela sva z idejo, da bi poskusila narediti program, pri katerem bi lahko videla, kako deluje genetski algoritem NEAT. Za tem sva določila, ali bova izdelala namizno (desktop) aplikacijo ali igro z vpletenim genetskim algoritmom. Odločila sva se za slednjo. Nato sva izbirala programsko okolje, v katerem bova izdelala igro. Izbirala sva med Unity in Godot. Po tehtnem premisleku sva se odločila za Godot, saj je veliko lažji za uporabo pri ustvarjanju platformer iger.

V njem sva pripravila preprosto orodje za oblikovanje dvodimenzionalnih prog z ovirami, namenjeni preizkušanju algoritma, tekača, ki ga bo algoritem nadzoroval, ter sistem za preizkušanje teh tekačev.

3.2 UPORABA ORODJA - GODOT

Uporabila sva odprtokodno programsko orodje za izdelavo iger Godot. Godot je namenjen ustvarjanju tako 2D- kot tudi 3D-iger, vendar pa se največkrat uporablja za izdelavo 2D-iger zaradi svoji preproste logike in enostavnega ustvarjanja. To orodje sta leta 2014 izdala Juan Linietsky in Ariel Manzur. Od takrat pa se letno izdajajo nove verzije, ki omogočajo ustvarjalcu lažjo izdelavo iger ter odpravljajo napake, ki se pojavljajo v predhodnih verzijah. Program je napisan v programskem jeziku C++, ustvarjalci pa lahko izdelujejo igre v C-programskih jezikih (C, C++, C#) ali pa v popolnoma novem jeziku, ki je bil ustvarjen prav za Godot in je zelo podoben programskemu jeziku Python. Ta jezik se imenuje GDScript in se uporablja izključno le v Godotu. Midva sva izbrala programski jezik C#, saj sva ga že dodobra spoznala skozi pretekla leta in tudi sedaj največ programov piševa v njem. Za razliko od GDScript jezika, ki ima editor v samem Godotu, je potrebno pri uporabi C-jezikov imeti naložen Visual studio ali Visual studio code ter tam pisati skripte.

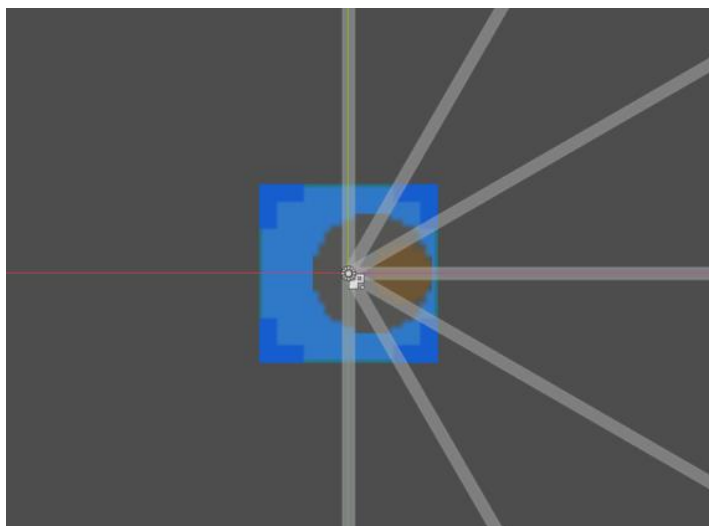
Godot, podobno kot tudi Unity, omogoča izdelavo iger za različne platforme in operacijske sisteme. Omogoča ustvarjanje za Linux, MacOS, Microsoft Windows, BSD,

Android, iOS, Universal Windows Platform, HTML5 in WebAssembly. Na žalost pa ne omogoča ustvarjanje iger za konzole, kot sta PlayStation ali Xbox. Pri nama to ni predstavljajo nikakršnih ovir, saj sva že na začetku določila, da bova ustvarila računalniško igrico.

Glavna funkcija, ki loči Godot od ostalih tovrstnih programov za izdelavo iger, je ta, da arhitektura motorja temelji na konceptu drevesa ("vozlišč"). Vozlišča so organizirana znotraj "prizorov", ki pa so skupine vozlišč. Te prizore in vozlišča je mogoče ponovno uporabiti, podedovati ali združevati. Vsi viri iger, vključno s skriptami in grafičnimi elementi, se shranijo kot del datotečnega sistema računalnika in ne v zbirki podatkov. Ta rešitev za shranjevanje je namenjena olajšanju sodelovanja med skupinami za razvoj iger, ki uporabljajo sisteme za nadzor različic programske opreme.

3.3 UPORABLJENE KOMPONENTE

3.3.1 TEKAČ



Slika 1: Tekač. Moder kvadrat je njegovo telo, sive črte pa so žarki, ki mu omogočajo vid. (foto: Andraž Podpečan)

Tekač ima dvajset žarkov, pritrjenih na t. i. oko, ki se obrača v smeri njegovega gibanja. Ti mu podajo razdaljo do objektov v njegovi okolici. Sedem žarkov zaznava zidove poligona, sedem zaznava nevarnost (žage), trije kovance ter trije zastave. Poleg teh podatkov dobi tekač še svojo hitrost, smer gibanja podano radianih, stanje (na tleh, v

zraku ali na steni) ter razdaljo do laserja.

Tekač je lahko uničen na sledeče načine:

1. Dotakne se žage.
2. Ujame ga laser.
3. Se ne giba dve sekundi.

3.3.2 SESTAVNI DELI POLIGONA

Poligon ima štiri sestavne elemente:

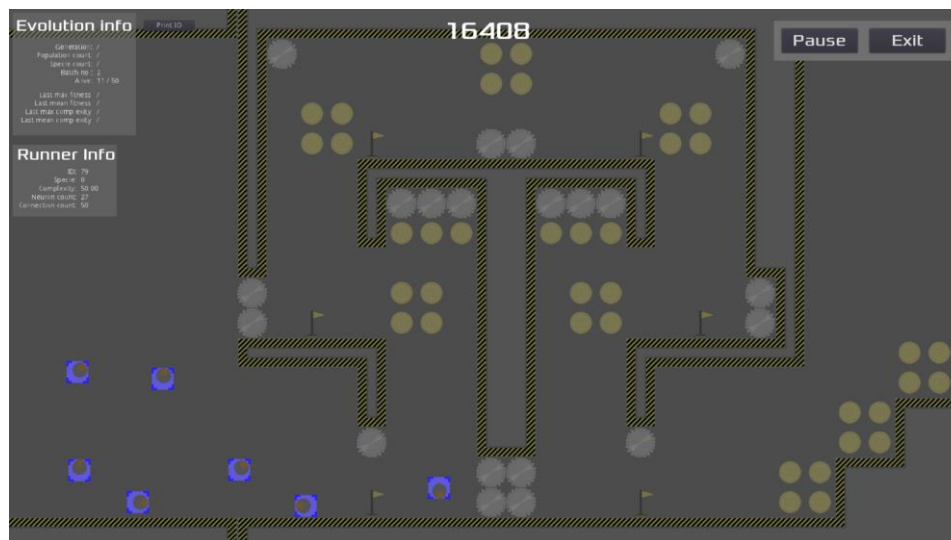
- Tla in zidovi, ki oblikujejo večji del poligona. Tekoč se jih lahko dotakne brez posledic.
- Žage, ki če se jih tekač dotakne, je uničen.
- Kovanci, ki tekaču ob dotiku povišajo fitness. Vsak tekač se lahko vsakega kovanca dotakne samo enkrat.
- Zastave, ki delujejo enako kot kovanci, le da imajo večjo vrednost.
- Laser, ki se giblje desno in uniči tekače, če jih doseže.

3.3.3 EKSPERIMENT

Cikel razvoja genetskega algoritma poteka tako:

1. Generacija tekačev je ustvarjena in postavljena v poligon.
2. Ko preteče 20 sekund ali pa so vsi tekači uničeni, je ustvarjena naslednja generacija tekačev in postopek se ponovi.
3. Razvoj se konča, ko je dosežen določen pogoj (generacija, fitness ali število obdelanih primerkov).

Fitness nevralnih omrežij, ki nadzorujejo tekače, je izračunan glede na čas, ki ga je posamezen tekač uspešno preživel, število dotaknjenih zastav in kovancev (zastava je vredna 10000 točk, kovanec pa 1000) ter prepotovano razdaljo.



Slika 2: Tekalci na poligonu (foto: Andraž Podpečan)

3.4 POTEK IZDELAVE

Izdelava projekta je razdeljena na tri dele: izdelava tekača, vključitev NEAT-a v igro in izdelava sistema za preizkušanje.

3.4.1 IZDELAVA TEKAČA

Pri tekaču je bilo potrebno določiti, kako se bo gibal ter kako bo zaznaval trke in okolico. Poleg tega pa je bilo potrebno poskrbeti tudi za to, da ja čim manj računsko zahteven. Vsi ti sestavni deli so se skozi razvoj večkrat spremenili. Sprva tekači v zraku niso imeli skoraj nobenega nadzora nad svojim gibanjem, kar je bilo zelo omejujoče in prezahtevno. Zaradi tega imajo tekači v končni verziji možnost gibanja tudi v zraku, četudi nekoliko okrnjeno v primerjavi z gibanjem na trdnih tleh.

Tudi zaznavanje trkov je bilo deležno precejšnjega števila sprememb, ki pa niso toliko vplivale na temeljno delovanje sistema, temveč bolj na njegovo učinkovitost in zanesljivost.

Največ sprememb pa je doletelo sistem za zaznavanje okolice. V prvi različici je imel tekač le osem žarkov, razporejenih v polnem krogu okoli njega. Ti žarki so povedali razdaljo in tip prvega objekta, ki ga dosežejo, kar pa ni bilo zadostno. Prvič zato, ker so bili žarki razporejeni prereditko, in drugič zato, ker tekaču niso zagotovili dovolj velikega obsega informacij. V drugi različici je žarkov le sedem, a so razporejeni polkrožno, kar

poveča njihovo gostoto, prav tako pa so pritrjeni na vrteče se oko. To oko se vedno obrača v smeri gibanja tekača. Ta način je rešil prvo težavo prejšnje inačice, ne pa tudi druge. Zato je v zadnji različici sistem narejen tako, da je število žarkov povečano, posamezni žarki pa lahko zaznajo samo določene objekte.

3.4.2 VKLJUČITEV NEAT

Vključiti nevrnalno omrežje v tekače je bilo samo po sebi zelo preprosto. Treba je bilo le določiti število vhodnih in izhodnih nevronov ter ustvariti sistem, ki je omrežju podajal vhodne informacije ter pretvarjal izhodne.

Veliko težje pa je bilo narediti sistem za ustvarjanje, ocenjevanje in evolucijo teh nevrnalnih omrežij.

Uporabila sva knjižnico SharpNEAT avtorja Colina Greena. Prednosti te knjižnice so preprosta vključitev v projekt, saj je napisana v programskem jeziku C# in jo je mogoče namestiti kot "NuGet" paket. Njena slabost, ki je povzročila velik del težav v tem delu projekta, pa je skoraj popolno pomanjkanje dokumentacije. Pomagati sva si morala z redkimi spletnimi vodiči in raziskovanjem izvorne kode. Naposled sva knjižnico le razumela dovolj, da sva jo lahko uspešno uporabila.

3.4.3 IZDELAVA CELOSTNEGA SISTEMA ZA PREIZKUŠANJE

To je bil najobsežnejši, najzapletenejši in temu primerno najbolj časovno potraten del projekta. Najprej sva naredila orodje za izdelavo poligonov. To je bilo še kolikor toliko preprosto, saj orodja ni bilo potrebno vključiti v igro samo, temveč je lahko ostalo kot dodatek pri Godotu.

Večji izziv pa je bilo ustvariti dejanski cikel igre. Pri tem je bilo treba poskrbeti za pravilno generiranje proge, nadzorovano ustvarjanje tekačev, poleg tega pa še ves uporabniški vmesnik za nastavitve evolucije in sledenju njenega napredka.

3.5 PREIZKUS ALGORITMA

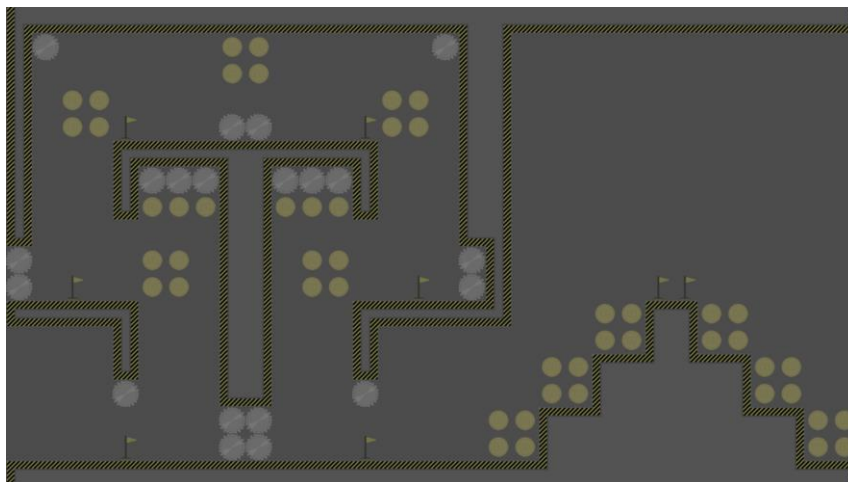
Ko je bilo okolje za preizkušanje končano, sva se lotila preizkušanja algoritma. Najprej sva preizkus zagnala s privzetimi nastavitvami za primerjalni vzorec. Nato sva preizkušala vsako od nastavitvev posebej. Namen tega ni bilo dobiti delujoče nevrnalno

omrežje, temveč zgolj ugotoviti, kako vsaka od nastavitvev vpliva na razvoj. Zato je vsak preizkus trajal le dvajset generacij.

Parametri, ki sva jih nastavljala, so bili:

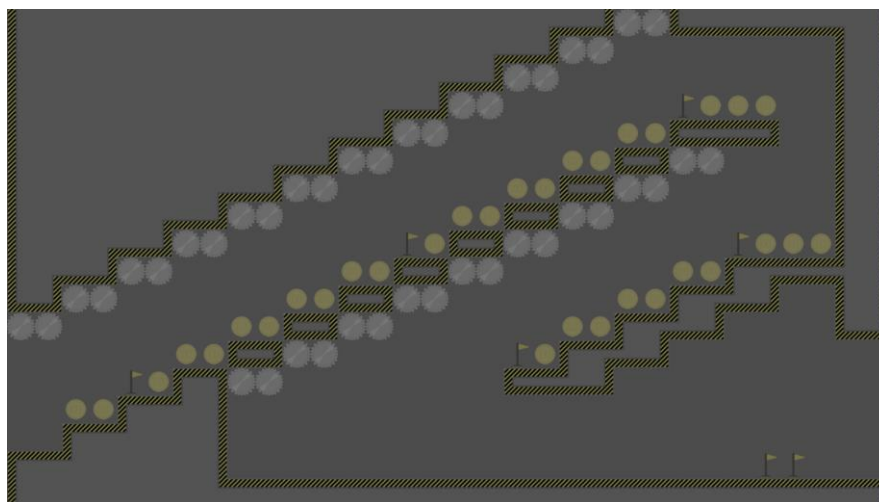
- Velikost populacije. To je število primerkov genetskega algoritma v vsaki generaciji. Privzeta vrednost je bila 200.
- Število vrst. To je, koliko različnih vrst lahko obstaja naenkrat. Privzeta vrednost je bila 10.
- Odstotek elitnih primerkov. Odstotek primerkov z najvišjim fitnessom iz vsake vrste, ki bo nadaljeval v naslednjo generacijo. Ostali primerki so uničeni, da naredijo prostor za potomce. Privzeta vrednost je bila 20 %.
- Izbirni odstotek. Odstotek primerkov z najvišjim fitnessom iz vsake vrste, ki bodo uporabljeni pri ustvarjanju potomcev. Privzeta vrednost je bila 20 %.
- Delež spolnega razmnoževanja. Delež potomcev, ki bodo ustvarjeni s spolnim razmnoževanjem. Preostanek je ustvarjen nespolno (z mutacijami). Privzeta vrednost je bila 50 %.
- Delež medvrstnega razmnoževanja. Delež potomcev, ustvarjenih s spolnim razmnoževanjem, ki bodo nastali s križanjem različnih vrst. Privzeta vrednost je bila 1 %.
- Razmerje mutacij. Mutacija lahko doda novo povezavo, odstrani obstoječo povezavo, spremeni utež na povezavi ali doda nov nevron na povezavo. Razmerje mutacij določi, kakšna verjetnost je, da se bo zgodila določena vrsta mutacije. Privzete vrednosti so bile:
 - 94 % za spremembo uteži na povezavi,
 - 2,5 % za dodajanje nove povezave,
 - 2,5 % za odstranitev obstoječe povezave,
 - 1 % za dodajanje novega nevrona.
- Delež začetnih povezav. Koliko od vseh mogočih povezav med vhodnimi in izhodnimi nevroni bo imela prva generacija primerkov. Privzeta vrednost je bila 5 %.

Vsi preizkusi so se odvijali na enaki progi.



Slika 3: Prvi preizkusni poligon (foto: Andraž Podpečan)

Ko sva opravila s temi osnovnimi meritvami, sva poskusila ugotoviti, kako se bodo nevralna omrežja razvijala, če njihove ovire ne bodo vedno enake. Zato sva načrtovala še en poligon.



Slika 4: Drugi preizkusni poligon (foto: Andraž Podpečan)

Kako si bosta poligona sledila po vrsti, je bilo naključno določeno na začetku vsake nove generacije.

Parametri algoritma so sledeči:

- število vrst: 15,
- velikost populacije: 300,
- delež elitnih primerkov: 20 %,
- delež izbranih primerkov: 20 %,

- delež spolnega razmnoževanja: 50 %,
- delež medvrstnega križanja : 5 %;
- mutacije:
 - dodajanje nevronov: 5 %,
 - dodajanje povezav: 10 %,
 - odstranitev povezav: 5 %.
 - sprememba uteži: 80 %;
- delež začetnih povezav: 50 %.

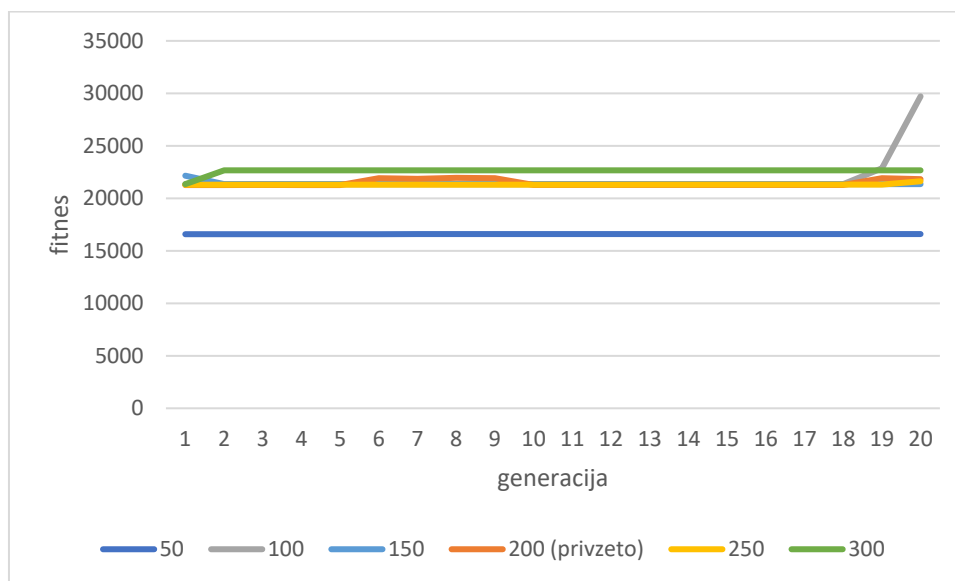
Preizkusa sta trajala 150 generacij.

4. REZULTATI

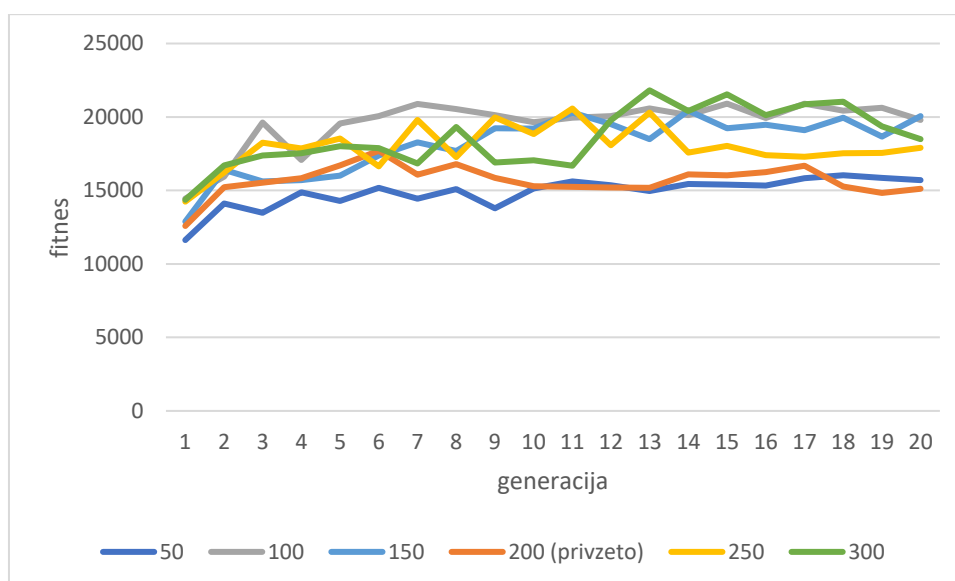
4.1 PREIZKUS POSAMIČNIH PARAMETROV

V tem delu preizkušanja sva spreminjala samo en parameter naenkrat. Ostali so bili nastavljeni na privzete vrednosti.

4.1.1 VELIKOST POPULACIJE

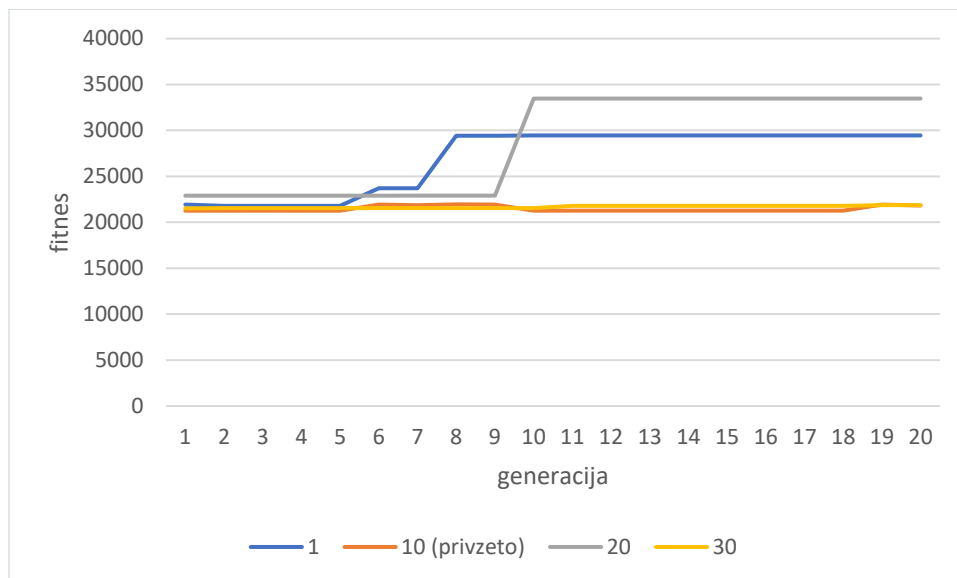


Graf 1: Največji fitnessi pri preizkušanju velikosti populacije

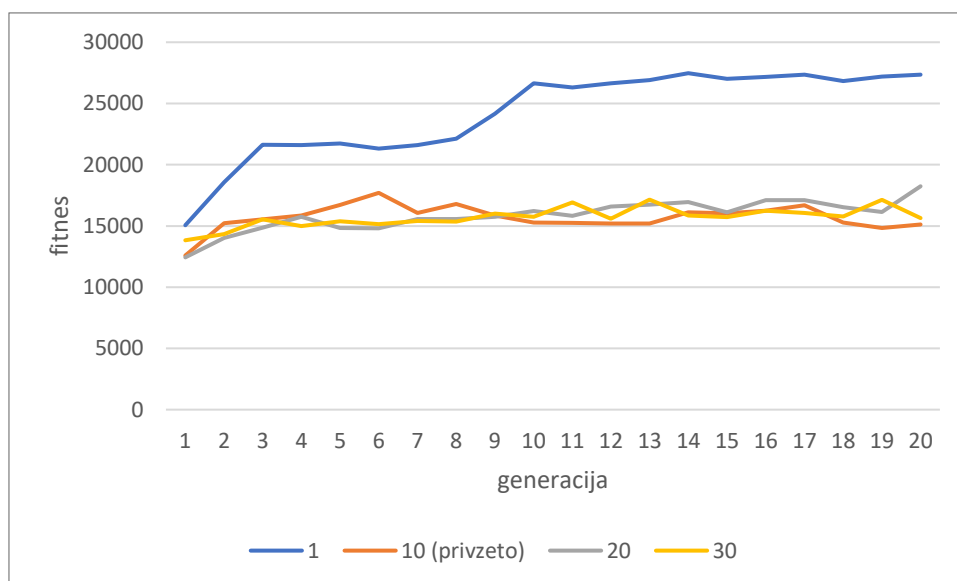


Graf 2: Povprečni fitnessi pri preizkušanju velikosti populacije

4.1.2 ŠTEVILO VRST

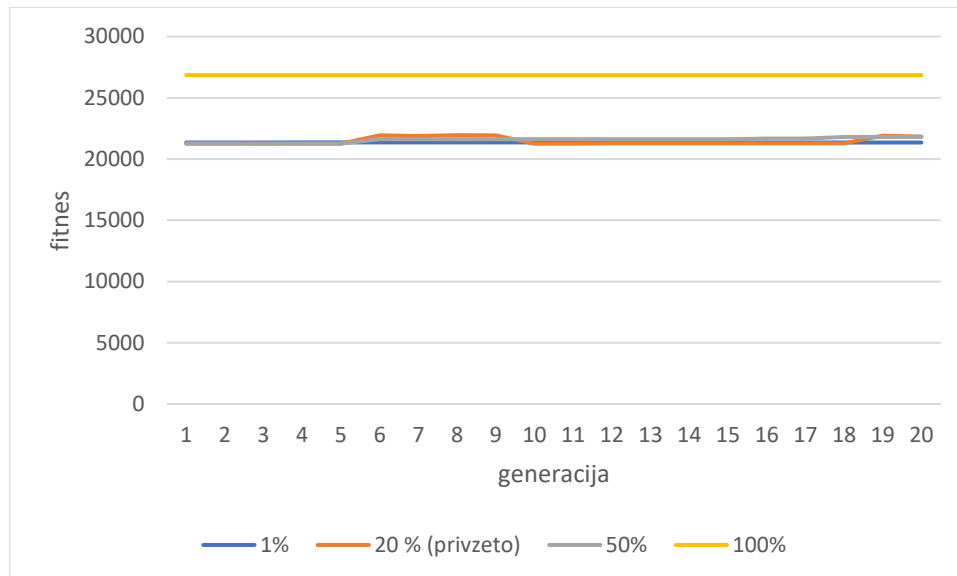


Graf 3: Največji fitnessi pri preizkušanju števil vrst

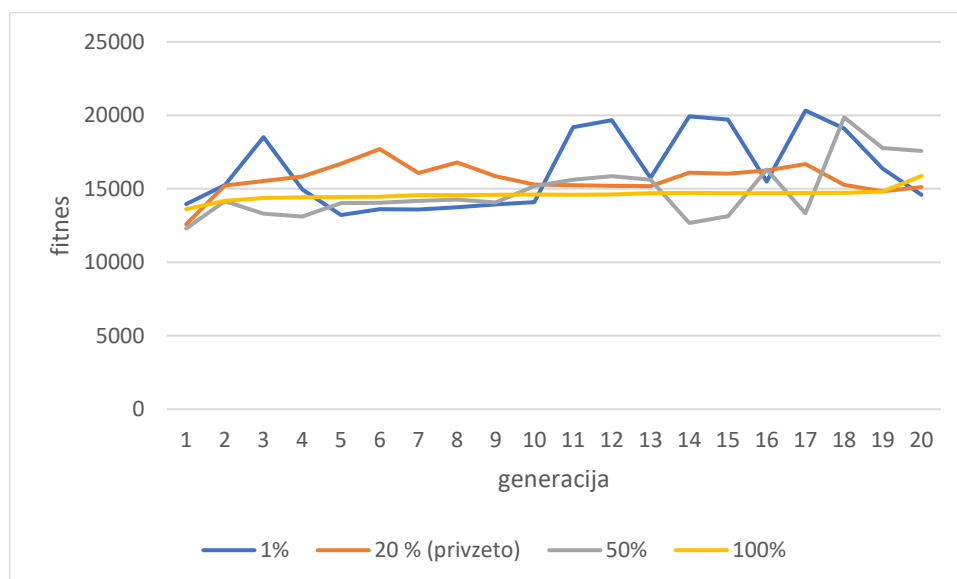


Graf 4: Povprečni fitnessi pri preizkušanju števila vrst

4.1.3 DELEŽ ELITNIH PRIMERKOV

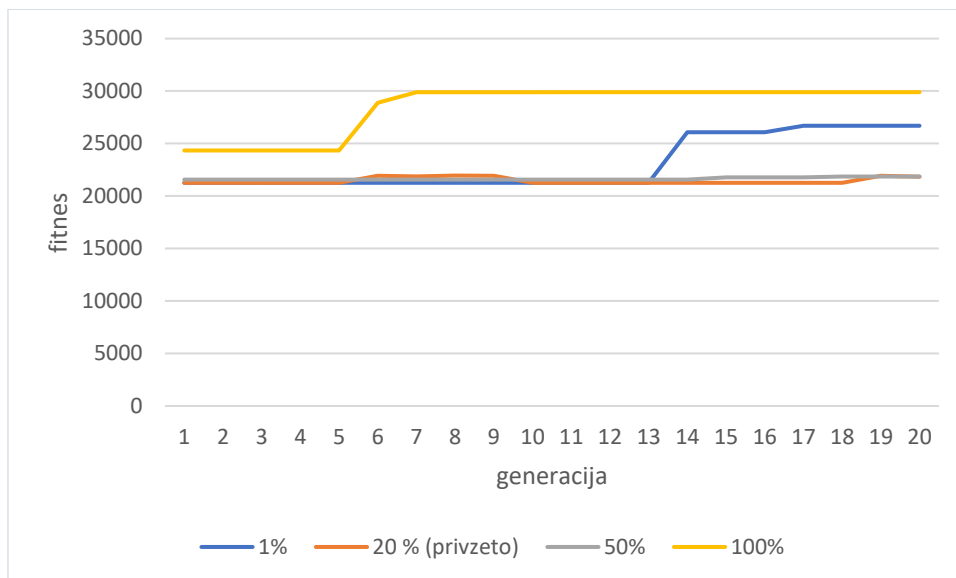


Graf 5: Največji fitnessi pri preizkušanju deleža elitnih primerkov

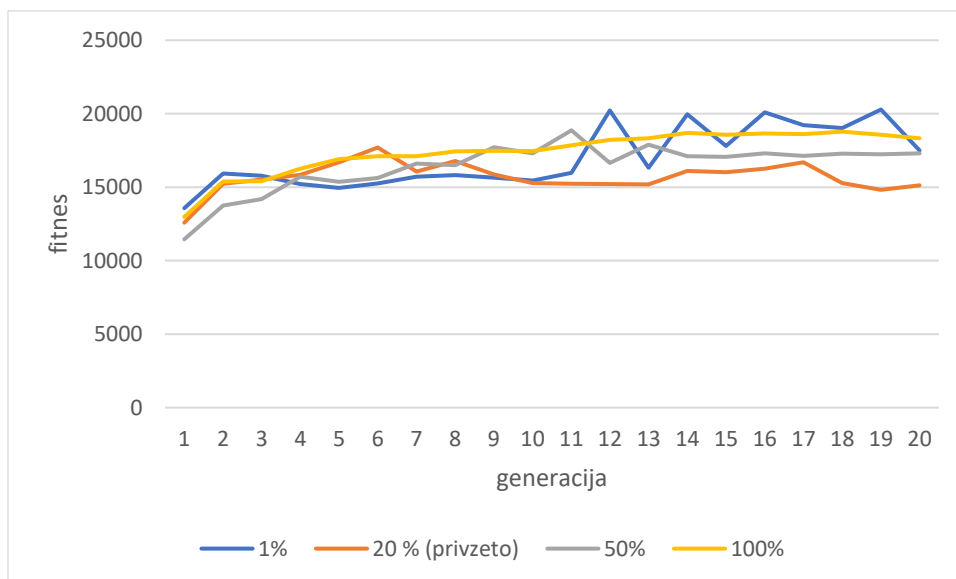


Graf 6: Povprečni fitnessi pri preizkušanju deleža elitnih primerkov

4.1.4 DELEŽ IZBRANIH PRIMERKOV

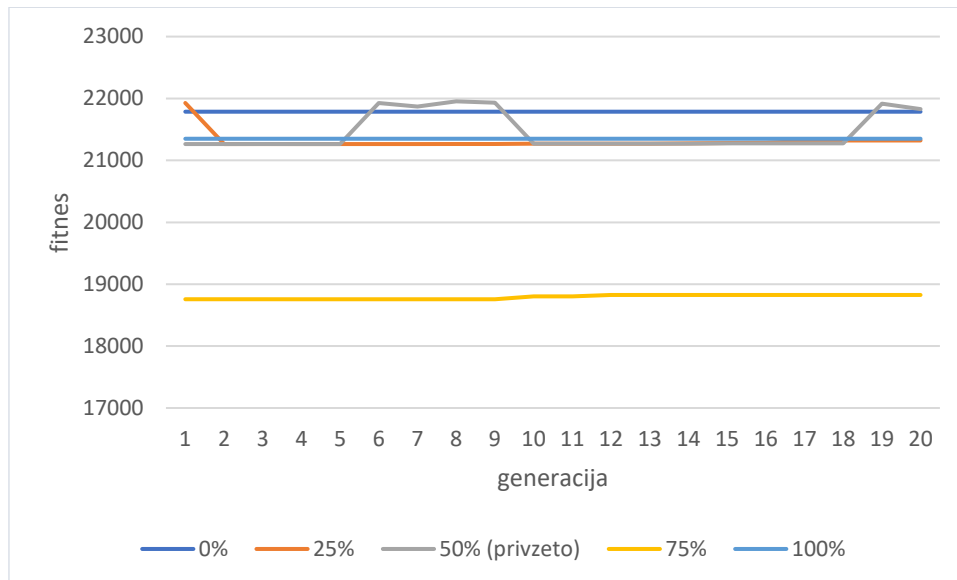


Graf 7: Največji fitnessi pri preizkušanju deleža izbranih primerkov

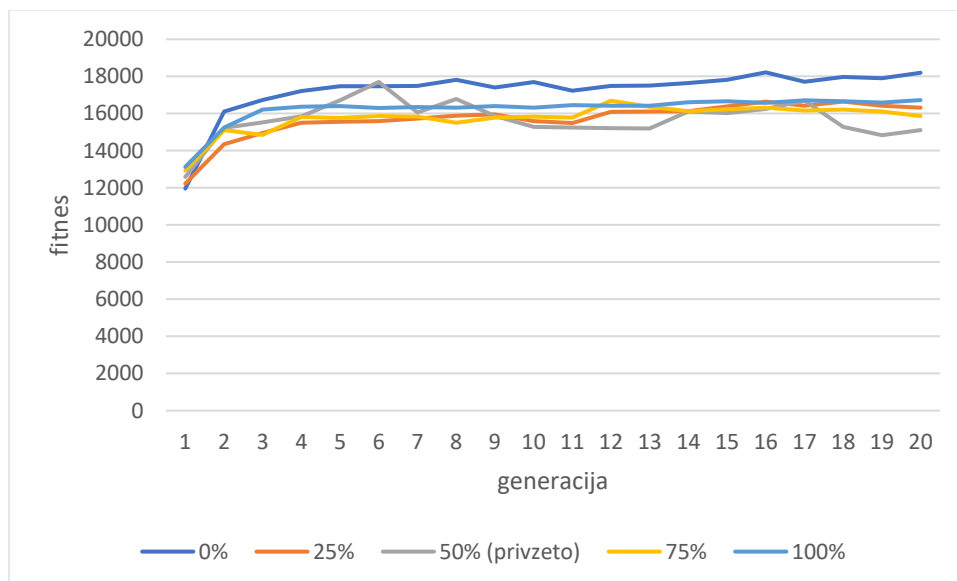


Graf 8: Povprečni fitnessi pri preizkušanju deleža izbranih primerkov

4.1.5 DELEŽ SPOLNEGA RAZMNOŽEVANJA

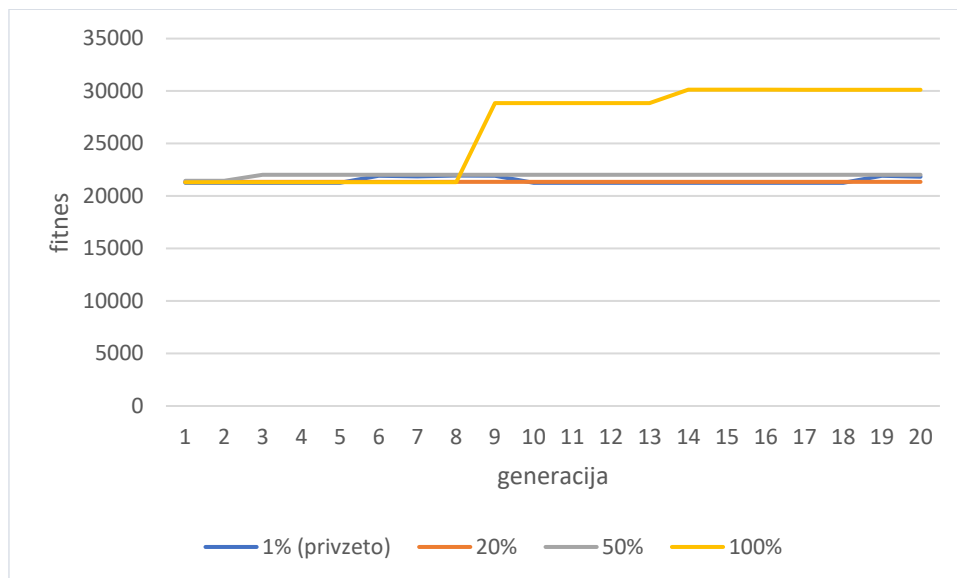


Graf 9: Največji fitnessi pri preizkušanju deleža spolnega razmnoževanja

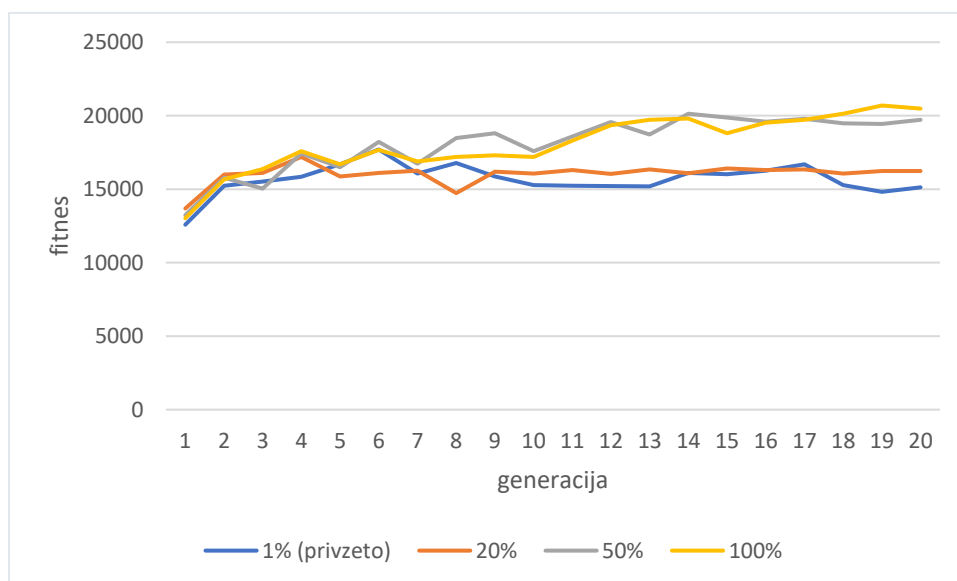


Graf 10: Povprečni fitnessi pri preizkušanju deleža spolnega razmnoževanja

4.1.6 DELEŽ MEDVRSTNEGA RAZMNOŽEVANJA

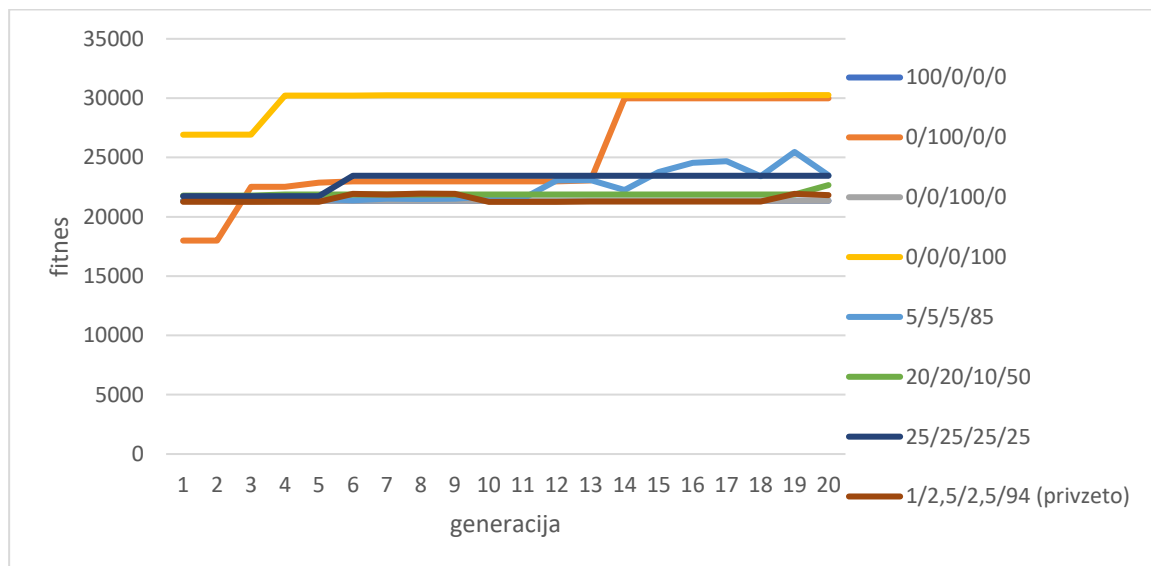


Graf 11: Največji fitnessi pri preizkušanju deleža medvrstnega križanja

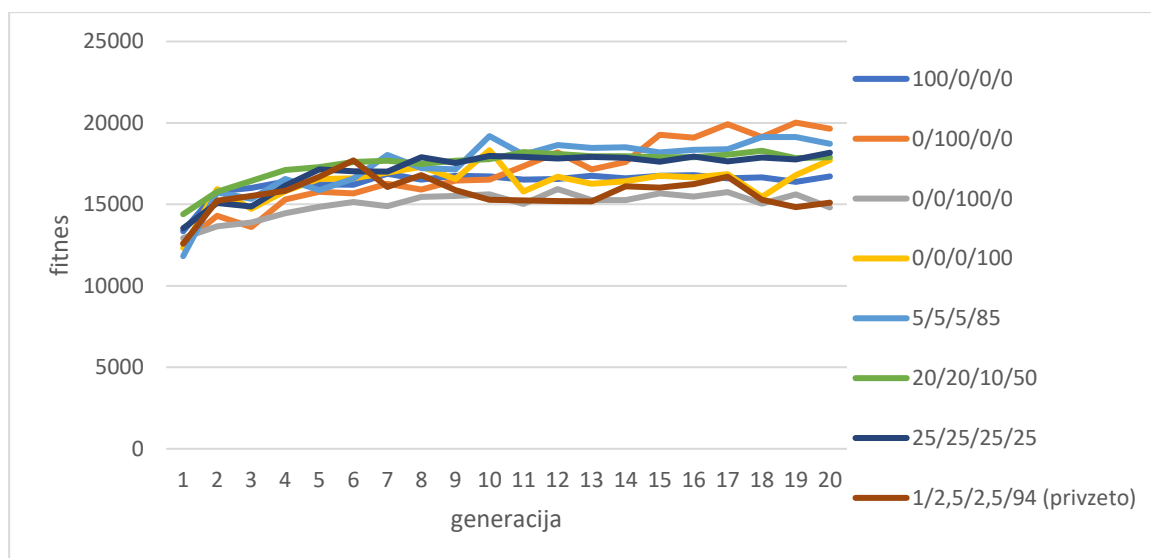


Graf 12: Povprečni fitnessi pri preizkušanju deleža medvrstnega križanja

4.1.7 RAZMERJE MUTACIJ



Graf 13: Največji fitnessi pri preizkušanju razmerij mutacij¹

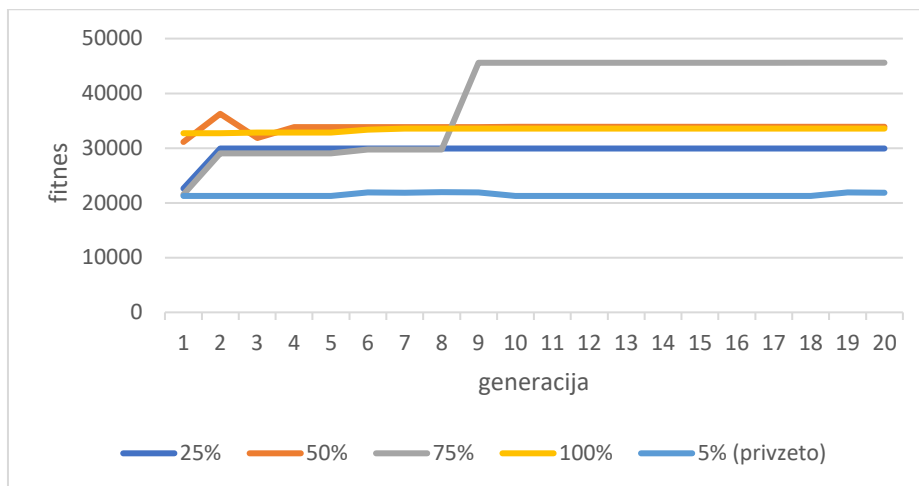


Graf 14: Povprečni fitnessi pri preizkušanju razmerij mutacij²

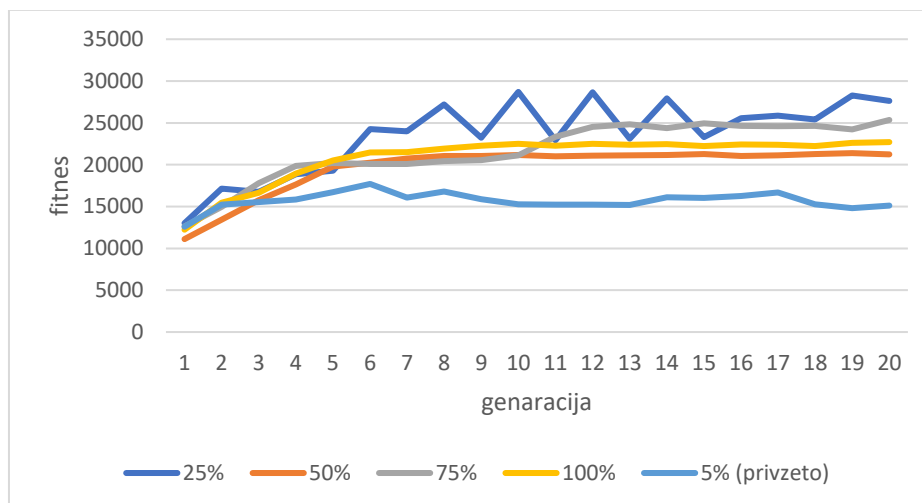
¹ Številke v legendi so odstotki različnih vrst mutacij. Prva je za dodajanje nevrona, druga za dodajanje povezave, tretja za odstranitev povezave in četrta za spremembo uteži.

² Glej opombo 1.

4.1.8 DELEŽ ZAČETNIH POVEZAV

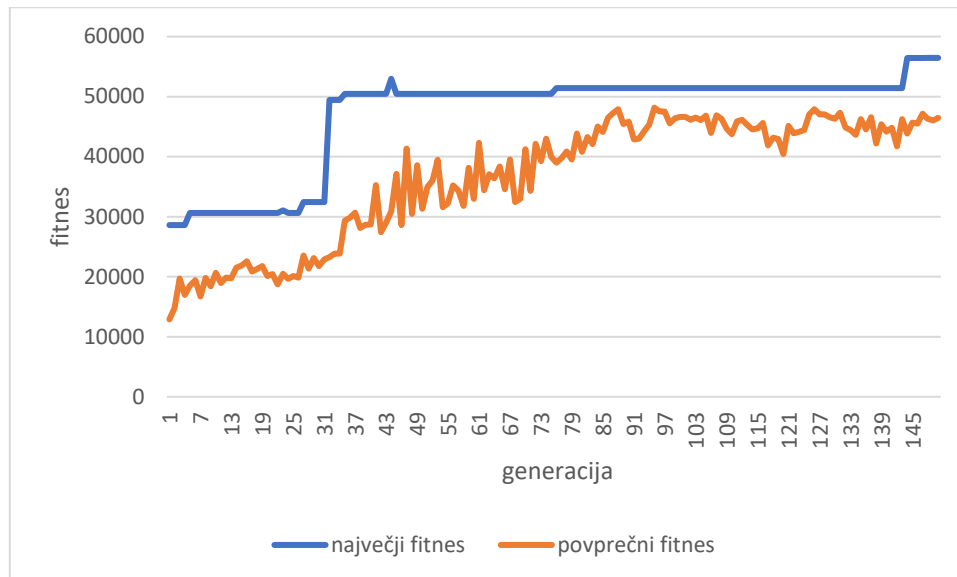


Graf 15: Največji fitnessi pri preizkušanju deleža začetnih povezav

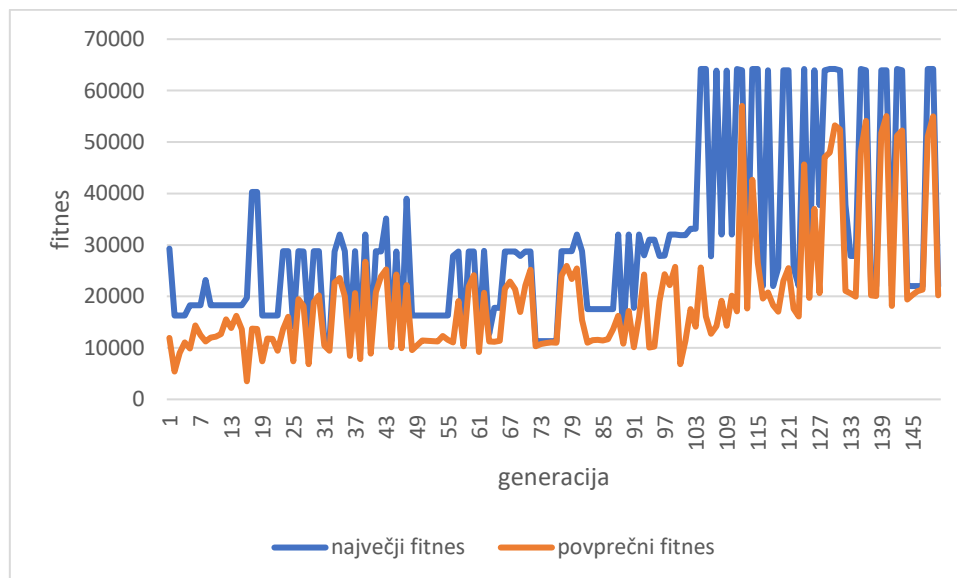


Graf 16: Povprečni fitnessi pri preizkušanju deleža začetnih povezav

4.2 STALNI IN NAKLJUČNI POLIGON



Graf 17: Meritve fitnessa preizkusa z enim poligonom

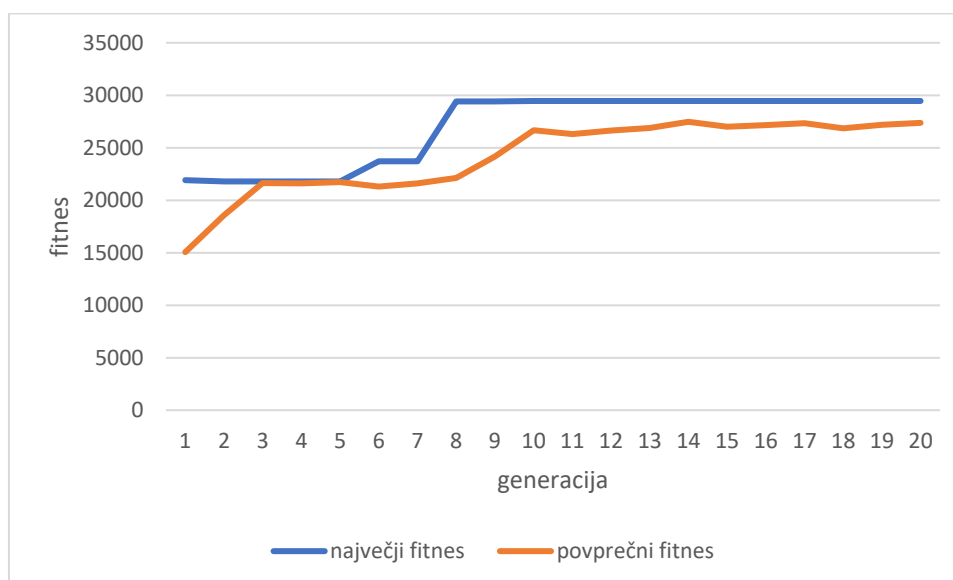


Graf 18: Meritve fitnessa preizkusa z dvema poligonoma

5. RAZPRAVA

5.1 TRENDI RASTI FITNESA

Marsikatera meritev ni pokazala niti najmanjše rasti največjega fitnessa. Tudi v ostalih primerih ni prišlo do naraščanja ali upadanja, ki bi bilo stalno, temveč je bilo skoraj vedno enkratno. Bolj zanimivi so bili podatki o povprečnem fitnessu. V večini primerov se je povprečje postopoma približevalo najvišji vrednosti, dokler se ji ni do neke mere približalo. Potem je tudi to ostalo na bolj ali manj enaki ravni, in če je narasel najvišji fitness, se je posledično povečal še povprečni fitness.



Graf 19: Največji in povprečni fitness pri preizkusu s samo eno vrsto

Razlog za ta vzorec so elitni in izbrani primerki. Elitni primerki so tisti, ki preživijo v naslednjo generacijo, izbrani pa tisti, ki bodo imeli potomce. Ker so oboji izbrani kot delež z najboljšim fitnessom iz vsake generacije in s privzetimi nastavitvami, je ta delež pri obeh parametrih enak. Zaradi tega največji fitness ne pade, saj bo najboljši primerek vedno preživel v naslednjo generacijo, in ker so potomci vedno osnovani na najboljših prednikih, je smiselno, da se bo povprečje približevalo vrhu.

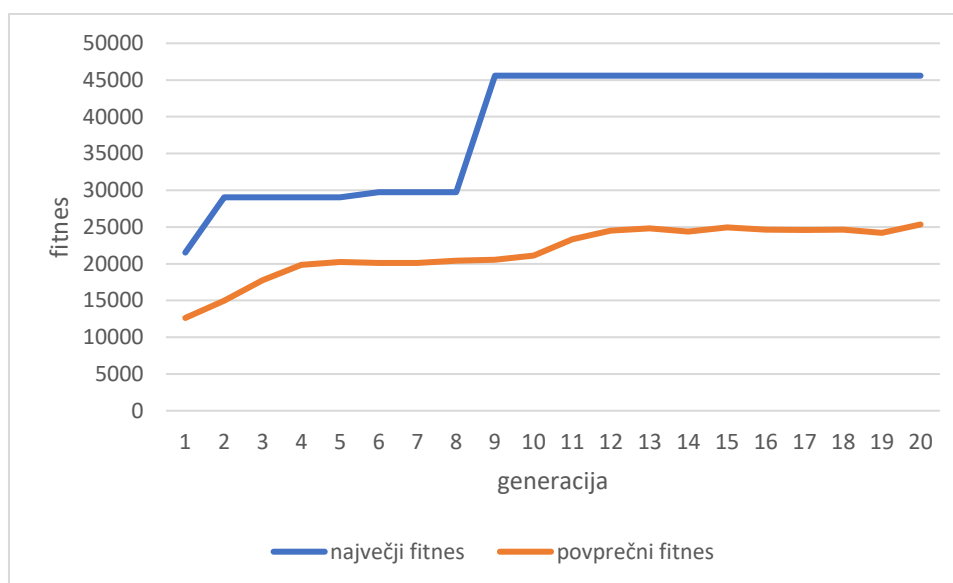
Seveda pride v meritvah tudi do lokalnega odstopanja od tega vzorca, a se ga da pripisati naključju. Nikjer se ni zgodilo, da bi se povprečje začelo odločno oddaljevati od vrha.

5.2 VPLIV PARAMETROV

V procesu razvoja nevralnih omrežij je stalno prisoten element naključja. Poleg tega nekateri parametri vplivajo na druge (npr. če je delež spolnega razmnoževanja majhen, potem je vpliv deleža medvrstnega križanja manj izrazit, saj je slednji odvisen od prvega). Kombinacija teh dveh dejstev onemogoči, da bi lahko iz zbranih podatkov razbrali, kakšen vpliv lahko pripišemo spremembam parametrov.

Bolj nazorno je opazovanje dejanskega delovanja tekačev v poligonu, saj tu lahko vidimo očitne razlike, četudi te niso opazne v izmerjenih podatkih.

Najboljši primer tega je spreminjanje deleža začetnih povezav.



Graf 20: Meritve fitnesa za 75-odstotni delež začetnih povezav

Podatki, ki jih predstavlja graf, niso nič posebnega. Imamo dva večja poskoka v največjem fitnesu in povprečje, ki temu polagoma sledi, tako kot je to opisano pri prejšnji točki. Ampak tekači se dejansko obnašajo izrazito drugače kot pri privzetih nastavitvah. Njihovo delovanje je veliko bolj raznovrstno in kaotično. S privzetim 5-odstotnim deležem začetnih povezav bo imelo nevralno omrežje eno ali dve povezavi³, s 75-odstotnim pa okoli 37. To seveda pomeni, da bodo izhodne vrednosti veliko bolj odvisne od vhodnih. Je pa res, da v omrežju ni še nobenih vmesnih nevronov, ki bi podatke dodatno obdelali, zato je delovanje omrežja prav tako neučinkovito.

³ Poleg dveh nespremenljivih povezav, skupno torej 3 do 4.

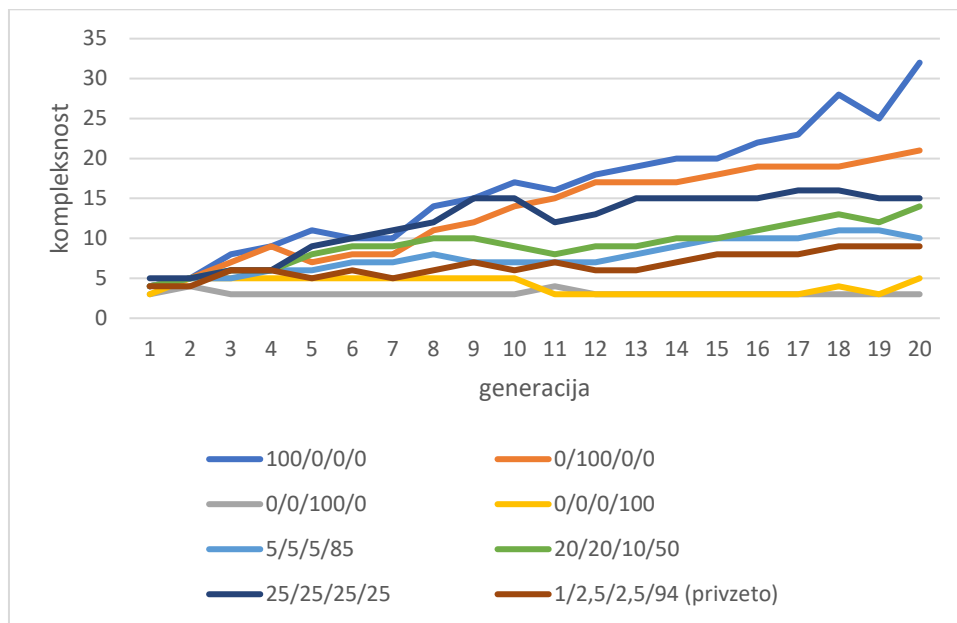
5.3 KOMPLEKSNOŠT

Kompleksnost omrežja je število povezav v nevralnem omrežju in ni na noben način pokazatelj njegove učinkovitosti. Skoraj nikoli ne bo najbolj kompleksno omrežje tisto z največjim fitnessom. Pravzaprav je načeloma veljalo ravno obratno.

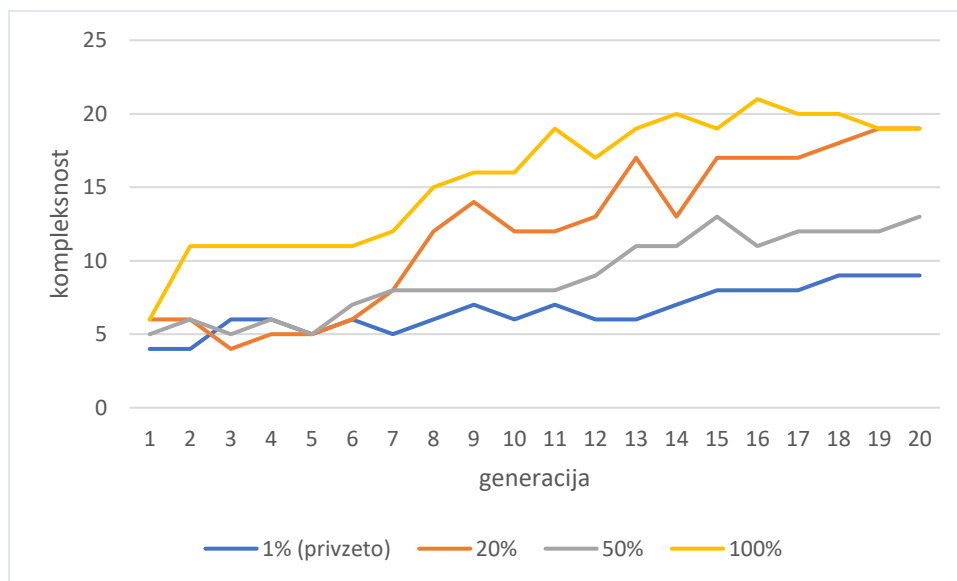
Za razliko od fitnesa so spremembe v kompleksnosti veliko bolj nepredvidljive. Na splošno raste tako v najvišji kot tudi v povprečni vrednosti, toda obe ob tem tudi večkrat upadeta, lahko da za več kot deset točk (povezav). To sploh ni slaba stvar, saj je namen algoritma NEAT ustvariti gosta nevralna omrežja s čim manj elementi. Je pa tudi res, da potrebuje omrežje dovolj zapleteno topologijo, da lahko uspešno opravi nalogo, ki mu je namenjena.

Parametri, ki najbolj izrazito povečujejo kompleksnost, so torej tisti, ki imajo vpliv na topologijo omrežja. Tu je v prvi vrsti seveda delež začetnih povezav, kot je bilo razloženo zgoraj. A ta ima vpliv samo pri ustvarjanju prve generacije. Tekom razvoja so pomembni razmerje mutacij in delež medvrstnega križanja. Visoka možnost za dodajanje nevrona ali povezave bo imela temu primeren učinek, še posebej če je delež spolnega razmnoževanja nizek. Visok delež medvrstnega križanja pa poveča možnost, da se bosta združili dve zelo različni topologiji, katerih potomec bo imel vse njune nevrone in povezave.

Seveda pa pogoste in drastične spremembe topologijam ne pustijo dovolj časa, da bi dejansko postale uporabne.



Graf 21: Največje kompleksnosti za različne nastavitve mutacij⁴



Graf 22: Največje kompleksnosti za različne nastavitve deleža medvrstnega razmnoževanja

⁴ Številke v legendi so odstotki različnih vrst mutacij. Prva je za dodajanje nevrona, druga za dodajanje povezave, tretja za odstranitev povezave in četrta za spremembo uteži.

5.4 STALNI IN NAKLJUČNI POLIGON

Namen tega preizkusa je bilo ugotoviti, če je morda za algoritem bolje, da razvoj ne poteka v nespremenljivem okolju. Povod za ta preizkus je bilo opažanje, da nevrnalna omrežja ne dojamejo, kaj je ima dober in kaj slab učinek na njihov fitness. S tem, da bi tekače naključno izpostavljali različnim okoljem, naj bi vzpodbudili nastanek takšnih smiselnih povezav, saj zgolj ustaljeno gibanje, ki se je obneslo v prvem okolju, ne bi delovalo v drugem.

Rezultati so hipotezo ovrgli. Razvoj v naključnem okolju ni bil nič boljši kot v nenaključnem. Sicer nevrnalna omrežja dosegajo višje fitnese, toda to je zgolj posledica tega, da je drugi poligon preprostejši in ima veliko več kovancev in zastav. Gibanje tekačev je bilo še vedno popolnoma neinteligentno, v prvem poligonu pa so se obnesli še slabše kot pri prvem preizkusu.

Tukaj je očitna pomanjkljivost NEAT-a. Nevralna omrežja nimajo nobene spodbude za razvoj. Evolucija deluje preprosto tako, da se primerki z največjimi fitnessi namnožijo in postanejo dominantni ter tako ostanejo, dokler čas in naključja ne ustvarijo primerka, ki je zmožen doseči večji fitness in cikel se znova ponovi. To deluje dobro za preproste naloge (npr. razvoj XOR-logičnih vrat, uravnoteženje palice), kar pa ta eksperiment pravzaprav ni bil.

Mogoče je, da bi NEAT z dovolj časa vseeno uspel ustvariti omrežje, katerega delovanje bi bilo do neke mere inteligentno. Ampak to bi najbrž trajalo več sto, morda tudi tisoč generacij.

Zato NEAT najverjetneje ni najbolj primeren algoritem za nalogo v tem eksperimentu.

6. ZAKLJUČKI

S pomočjo najine raziskovalne naloge sva ugotovila, da:

- Najvišje vrednosti fitnesa naraščajo v korakih, povprečne vrednosti fitnesa pa postopoma, dokler se do neke mere ne približajo najvišjim.
- Napredek omrežij ni bil viden iz generacije v generacijo.
- Rast kompleksnosti nevralnih omrežij je nepredvidljiva.
- Večja kompleksnost ne zagotovi boljšega delovanja nevralnih omrežij.
- Na kompleksnost najbolj vplivajo parametri deleža medvrstnega križanja in razmerje mutacij (oboje v povezavi z deležem spolnega razmnoževanja) ter delež začetnih povezav (ampak samo na začetku evolucije).
- Razvoj omrežij ni neposredno vzpodbujen, temveč temelji na naključnih mutacijah.
- Naključno okolje ne spodbuja razvoja inteligentnih povezav med vhodnimi in izhodnimi podatki nevralnega omrežja.
- Potrebno bi bilo primerjati NEAT s kakšnim drugačnim algoritmom za strojno učenje (npr. Q-Learning ali kakšen hibridni evlucijski algoritem).

7. POVZETEK

Za to raziskovalno nalogo sva se odločila, ker je področje umetnih nevrlnih omrežij –še posebej evlucijskih algoritmov – zelo zanimivo, aktualno in praktično uporabno. Preizkusila sva algoritem NEAT, ki ga je leta 2002 razvil Stanley O. Kenneth, ker ima določene prednosti pred drugimi nevroevolucijskimi algoritmi (razvrščanje genomov v vrste, označevanje mutacij).

Za eksperiment sva pripravila preprosto igro v Godotu, odprtokodnem razvojnem okolju za videoigre. V tej igri mora preprost tekač, z zmožnostjo gibanja levo in desno ter skakanja od tal in zidov, premagati poligon z nevarnostmi in nagradami, medtem ko beži pred laserjem, ki bi ga uničil. Uporabila sva knjižnico SharpNEAT Colina Greena (MIT-licenca), v glavnem zaradi preproste vključitve v projekt.

V prvem delu preizkusov sva poskusila ugotoviti, kako posamezni parametri vplivajo na razvoj nevrlnih omrežij. Meritve fitnessov niso povedale skoraj ničesar opredeljivega o tem, meritve kompleksnosti pa so pokazale izrazito povezavo med nekaterimi parametri in rastjo oz. zastojem rasti kompleksnosti. Toda kompleksnost ni pokazatelj stopnje učinkovitosti nevrlnega omrežja, zato ta ugotovitev nima bistvenega pomena.

Opazovanje delovanja tekačev je razkrilo, da nevrlna omrežja ne ustvarijo inteligentnih povezav med »videnim« in storjenim. Preizkusila sva, če je mogoče spodbuditi nastanek takšnih povezav z razvojem omrežij v naključno postavljenem okolju. Toda meritve in tudi opazovanja na vzorcu 150 generacij se s tem ne strinjajo in nakazujejo tudi na to, da bi razvoj opazne inteligence najverjetneje trajal zelo dolgo.

Na splošno gledano se zdi, da je bila naloga v eksperimentu preveč zapletena za NEAT. Razvoj nevrlnih omrežij s tem algoritmom je počasen in pasiven postopek, kjer ni prave vzpodbude za izboljšanje. Glavni nosilec razvoja so zato mutacije, ki pa so podvržene naključju. Zato sva prišla do zaključka, da NEAT ni primeren algoritem za ta eksperiment, bi bilo pa treba preizkusiti še kakšen drug algoritem za strojno učenje (npr. Q-learning), da bi lahko to trditev dokončno potrdila.

8. SUMMARY

The reason for starting this research was the fact that artificial neural networks – especially evolutionary algorithms – are a fascinating, relevant, and practical field. The algorithm we tested was NEAT, developed in 2002 by Kenneth O. Stanley, because it has certain advantages compared to some other neuroevolutionary algorithms (speciation, genome tracking)

For the experiment we developed a rudimentary game in Godot, an open-sourced game engine. In this game a simple runner, capable of moving left and right as well as jumping from ground and walls, must navigate a maze filled with danger and rewards, while running away from a laser, which would destroy it. The library we used was SharpNEAT by Colin Green (MIT license), mainly due to it being easy to include into the project.

In the first part of experimentation, we tried to determine, how individual parameters affect the evolution of neural networks. Measuring fitness gave us next to no conclusive results regarding that, but measuring complexity showed a distinct link between some parameters and the rise or stagnation of complexity. However, complexity is not an indicator of the neural networks efficiency, so this discovery has little meaning.

Observing the runners' behaviors revealed that neural networks do not have intelligent relations between what they “see” and what they do. We tested if exposing the specimens to a randomized environment would encourage the formation of such relations. The measurements, taken from samples of 150 generations, however, disagree with that theory and indicate that it would take a very long time for any significant intelligence to evolve. Overall, it seems that the task in our experiment was far too complex for NEAT. The evolution of neural networks with this algorithm is a slow and passive process, without any real incentive for improvement. Therefore, the primary cause of development are mutations, which are governed by random chance. So, we have arrived at the conclusion, that NEAT is not a suitable algorithm for the task in this experiment, though we would need to test another machine learning algorithm (e.g. Q-learning) to fully confirm this claim.

9. VIRI

9.1 ELEKTRONSKI VIRI

- Goyal, Deep Learning vs Neural Networks: Difference Between Deep Learning and Neural Networks.

<https://www.upgrad.com/blog/deep-learning-vs-neural-networks-difference-between-deep-learning-and-neural-networks/> (14. 2. 2021).

- Heidenreich, NEAT: An Awesome Approach to NeuroEvolution.

<https://towardsdatascience.com/neat-an-awesome-approach-to-neuroevolution-3eca5cc7930f> (28. 2. 2021).

- Godot (game engine). [https://en.wikipedia.org/wiki/Godot_\(game_engine\)](https://en.wikipedia.org/wiki/Godot_(game_engine)) (25. 1. 2021).

- Neuroevolution of augmenting topologies.

https://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies (11. 2. 2021).

- Kenneth, S., Miikkulainen, R., Evolving Neural Networks through Augmenting Topologies, Austinska univerza v Texasu, ZDA

<http://nn.cs.utexas.edu/downloads/papers/stanley.ec02.pdf> (15. 3. 2021)

9.2 KNJIŽNICA NEAT

- Collin Green, SharpNEAT. <https://sharpneat.sourceforge.io/> (23. 1. 2021).

ZAHVALA

Za pomoč pri najini raziskovalni nalogi se v prvi vrsti iskreno zahvaljujema najinemu mentorju Samu Železniku, ki nama je ves čas stal ob strani ter nama dajal zelo koristne ter pomembne napotke. Z njegovo pomočjo, predvsem pa njegovimi idejami za izboljšave, sva lahko naredila igro, na katero sva zelo ponosna.

Prav tako se zahvaljujema tudi najinemu somentorju Islamu Mušiću, ki pa nama je dajal predvsem moralno in psihološko podporo ter naju ves čas bodril in opogumljal na najini raziskovalni poti.

Poleg zahvale mentorjema pa se iskreno zahvaljujema tudi najinima družinama, saj so nama ves čas najine raziskave stali ob strani ter naju vzpodbujali.