SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1678

INTELIGENTNI IGRAČ ZA ZATVORENIKOVU DILEMU

Petar Belošević

Zagreb, lipanj, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1678

Pristupnik:	**Petar Belošević (0036538383)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	izv. prof. dr. sc. Marko Čupić

Zadatak: **Računalni inteligentni igrač za igranje Zatvorenikove dileme**

Opis zadatka:

Zatvorenikova dilema je misaoni eksperiment koji igraju dva racionalna agenta koji međusobno mogu ili surađivati ili igrati protiv drugog agenta, a sve u svrhu maksimiziranja nagrade. U okviru ovog završnog rada potrebno je dati detaljni opis problema, istražiti i opisati njegova postojeća rješenja te implementirati računalnog igrača koji nekom tehnikom strojnog učenja pokušava naučiti dobru strategiju igranja. Radu priložiti izvorni i izvršni kod, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod

Zatvorenikova dilema jedan je od najpoznatijih problema iz područja teorije igara. Dilema proučava interakciju između dvaju pojedinaca kroz igru suradnje i izdaje. Zatvorenikova dilema se može pronaći u korijenu mnogih interakcija koje se pojavljuju u društvu, ali i u životinjskom svijetu. Glavno pitanje je kako ta interakcija utječe na dobrobit oba igrača. Paradoks u Zatvorenikove dileme je činjenica da racionalno razmišljanje navodi svakog pojedinca da odabire opciju koja ne odgovara niti jednom od igrača [1].

U stvarnom životu su takve interakcije često ponavljajuće prirode. Tada je potrebno imati na umu da će druga strana vjerojatno imati iskustvo prošlih interakcija što može utjecati na njihovu odluku u novoj interakciji. Stoga je interesantno razmatranje malo složenijeg problema ponavljajuće Zatvorenikove dileme. Zanimljivo je da se u ponavljajućem problemu mijenja pristup optimalnoj strategiji u odnosu na jednu Zatvorenikovu dilemu.

U korijenu Zatvorenikove dileme je napetost između individualnog racionalizma (u pogledu da je objema stranama smisleno biti sebičan) i grupnog racionalizma (objema stranama je isplativija suradnja nego obostrana izdaja) [2].

Mnogo je radova na pisano na temu optimalnog načina igranja Zatvorenikove dileme. Tako je Robert Axelrod iz Sveučilišta u Michiganu održao dva turnira u kojima su sudionici predavali svoje strategije za igranje igre u obliku računalnih programa [2] [3]. Analizom turnira došlo se zanimljivih zaključaka. Axelrod je proučavanjem strategija i njihovih rezultata izlučio nekoliko karakteristika koje imale tendenciju davati bolje rezultate. Te karakteristike su ljubaznost, opraštanje i provokabilnost [3].

Glavni cilj ovog rada je napraviti model inteligentnog igrača koji će naučiti kvalitetno igrati ponavljajuću Zatvorenikovu dilemu. Kod dobivenog modela će se proučiti njegovo

ponašanje te će biti zanimljivo vidjeti je li dobiveni inteligentni igrač razvio karakteristike koje su se pokazale poželjnima u Axelrodovim turnirima.

2. Zatvorenikova dilema

Zatvorenikova dilema je poznati problem iz područja teorije igara koji proučava odnose između pojedinaca. Radi se o igri s dva igrača koji imaju na izboru dva poteza: suradnja s drugim igračem ili izdaja drugog igrača. S obzirom na odabir oba igrača, svakom od njih se daje određena kazna ili nagrada.

Zatvorenikova dilema je dobila naziv po ilustrativnom opisu Alberta Tuckera [4]. Dilema je opisana u kontekstu odvojenog ispitivanja dvojice osumnjičenih kriminalaca. Svaki od osumnjičenih može priznati krivnju (ekvivalentno izdaji) ili šutjeti (ekvivalentno suradnji). Sukladno njihovim potezima dodjeljuju im se zatvorske kazne. Ako i jedan i drugi osumnjičenik šute, dobivanju male zatvorske kazne zbog nedostatka dokaza. Ako i jedan i drugi priznaju krivnju, svatko od njih dobiva veće zatvorske kazne jer su priznali krivnju. Ako pak jedan od njih prizna krivnju a drugi šuti, ovaj koji je priznao krivnju biva oslobođen optužbi kao nagrada za suradnju s pravosudnim organima, a drugi dobiva još veću zatvorsku kaznu zbog toga što je odugovlačio postupak odbijanjem priznavanja krivice [4].

Dilema se naravno može staviti i opisati kroz razne druge slikovite kontekste. No, u ovom radu će se radi jednostavnosti na dilemu gledati samo kao igru u kojoj igrači skupljaju bodove.

Generalna pravila za "bodovanje" je da suradnja oba igrača i obostrana izdaja daju simetričnu podjelu bodova, uz to da suradnja daje veći broj bodova. U slučaju da jedan igrač surađuje dok ga drugi izdaje dolazi do nesimetrične podjele bodova. Igrač koje je izdan mora dobiti manje bodove nego bi dobio obostranom izdajom. Igrač koji ga je izdao tim potezom mora dobiti više nego bi dobio suradnjom oba igrača. Tipična raspodjela bodova [2] koja će se koristiti i u ovom radu dana je u tablici 2.1.

Važno je primijetiti da je Zatvorenikova dilema igra s promjenjivim ishodom [1]. To znači da dobitak jednog igrača nije nužno gubitak drugog igrača i gubitak jednog igrača ne mora nužno biti dobitak drugog igrača [5]. Drugim riječima, mogući su ishodi u kojima oba igrača profitiraju ili oba igrača gube. Upravo ova karakteristika omogućuje suradnji da bude poželjan pristup igranju.

Tablica 2.1.: Bodovanje odluka igrača u Zatvorenikovoj dilemi

Napomena: Bodovi Igrača 1 su dani prvim brojem u svakom od parova.

		Igrač 2	
		Suradnja	Izdaja
Igrač 1	Suradnja	3, 3	0, 5
	Izdaja	5, 0	1, 1

Zbog ove karakteristike Zatvorenikova dilema dobro opisuje mnoge probleme iz stvarnog života i objašnjava zašto je ponekad suradnja s neprijateljem poželjna. Poznati je takav primjer problem prekomjernog naoružavanja i međusobnog nepovjerenja SAD-a i SSSR-a tijekom hladnog rata [2] Upravo je i taj problem bio jedan od motiva nastanka i proučavanja Zatvorenikove dileme.

Zatvorenikova dilema dobro modelira određena ponašanja nekih životinjskih vrsta u prirodi. Tako je primijećeno ponašanje poznato kao *inspekcija predatora* kod nekih vrsta riba kao što su gupi riba (lat. *Poecilia reticulata*) i koljuška (lat. *Gasterosteus aculeatus*). [6]

Također je primijećena suradnja kod impala koje se međusobno timare kako bi se štitile od krpelja. No, zbog nedostatka informacija o cijeni i nagradi za ovakvo ponašanja teško je ovaj primjer modelirati Zatvorenikovom dilemom. [6]

2.1. Dobra strategija

Pitanje koje se prirodno nameće u ponavljajućoj Zatvorenikovoj dilemi je dosta očito: Kako dobro igrati ovu igru? Kojom strategijom pristupiti ovoj igri?

U svrhu istraživanja optimalnih strategija Robert Axelrod je 1980. organizirao turnir

[2] u koji je pozvao znanstvenike iz raznih područja koji su se bavili Zatvorenikovom dilemom. Sudionici su za turnir izradili strategije igranja Zatvorenikove dileme u obliku računalnog programa. Svaka strategija je igrala Zatvorenikovu dilemu u 200 iteracija sa svakom drugom strategijom, sa samom sobom i s dodatnom strategijom koja je donosila nasumične odluke. Kasnije te godine Axelrod je organizirao još jedan sličan turnir u svrhu provođenja dodatne analize [3].

Pobjednik u oba turnira je bila strategija zvana *Tit for Tat*, također poznata i kao *Copycat*. Strategija je vrlo jednostavna – na prvom potezu uvijek surađuje, a dalje uvijek kopira suparnikov prethodni potez [2].

Axelrod je analizom rezultata oba turnira pronašao nekoliko karakteristika koje su bile ključne za uspješnost strategija u oba turnira. Pronađene karakteristike su:

- **Ljubaznost** (eng. *niceness*) [2] – strategija je ljubazna ako nikada neće prva izdati drugog igrača
- **Sklonost opraštanju** (eng. *forgiveness*) [2] – sklonost strategije da surađuje nakon što ju je drugi igrač izdao
- **Provokabilnost** (eng. *provocability*) [3] – sklonost da strategija izda drugog igrača odmah nakon što ju je drugi igrač izdao

Valja primijetiti da su karakteristike sklonosti opraštanju i provokabilnost zapravo u suprotnosti. No one nisu binarne (ili su prisutne u strategiji ili nisu) neko se pojavljuju u strategijama u nekoj mjeri. Zbog toga nisu međusobno isključive. Također, provokabilnost promatra trenutnu reakciju na izdaju od strane drugog igrača, dok sklonost opraštanju može kroz više poteza gledati vjerojatnost da igrač oprosti izdaju [3].

Strategija *Tit for Tat* je bila uspješna jer je bila ljubazna te je imala dobru ravnotežu između sklonosti opraštanju i provokabilnosti. *Tit for Tat* je ljubazna strategija jer počinje sa suradnjom i izdaje samo ako ju je drugi igrač izdao prvi. Strategija oprašta jer će nakon što bude jedom izdana uzvratiti izdajom, ali će nakon toga nastaviti surađivati dok ponovno ne bude izdana – strategija pamti samo posljednji suparnikov potez. Također, strategija je provokabilna jer će nakon što bude izdana uvijek i odmah uzvratiti izdajom.

Valja zamijetiti jednu vrlo zanimljivu činjenicu. Ako malo razmislimo o strategiji *Tit for Tat*, možemo primijetiti da ta strategija nikada ne može igrati bolje od igrača s kojim igra Zatvorenikovu dilemu. U najboljem slučaju strategija *Tit for Tat* može dobiti jednak broj bodova kao i drugi igrač. Ovo pokazuje da kod ponavljajuće Zatvorenikove dileme nije uvijek nužno biti bolji od drugog igrača, već igrati kontinuirano dobro s različitim strategijama.

Axelrod je u svojem radu napomenuo važnu stvar, a to je da ne postoji najbolja strategija za igranje ponavljajuće Zatvorenikove dileme. To proizlazi iz jednostavne činjenice da performansa strategije uvelike ovisi o strategijama s kojima ta strategija igra. Dakle, okruženje u kojem se nalazi je od presudne važnosti [2].

No kroz detaljnu analizu drugog turnira Axelrod je ustanovio da su navedene karakteristike generalno poželjne i u pravilu donose dobre rezultate. Također, iako *Tit for Tat* nije univerzalno najbolja strategija, pokazuje se da generalno daje odlične rezultate u raznim okruženjima zbog dobro kombiniranih poželjnih karakteristika [3].

Axelrod je također napomenuo da je kod kreiranja strategije potrebna analiza na barem 3 dubine [2]. Prva razina je direktna posljedica trenutne odluke, to jest, koliko bodova igrač dobiva na temelju svoje odluke. Druga razina je uzimanje u obzira da će drugi igrač možda odlučiti kazniti izdaju. Treća razina razmatra da reagiranje na izdaju može prouzrokovati eho efekt međusobnih izdaja koji će dugoročno štetiti i jednom i drugom igraču.

3. Umjetne neuronske mreže

Umjetne neuronske mreže su jedan od mnogih koncepata u strojnom učenju, koje je jedno od najvećih grana umjetne inteligencije. Prvi puta su ih predložili Warren McCullough i Walter Pitts sa Sveučilišta u Chicagu davne 1944. Prva neuronska mreža koja se mogla trenirati je bila Perceptron, a predložio ju je psiholog Frank Rosenblatt sa Sveučilišta Cornell 1957. [7]

Umjetne neuronske mreže su inspirirane ljudskim mozgom - sastoje se od velikog broja jednostavnih umjetnih neurona koji su gusto povezani. [7]

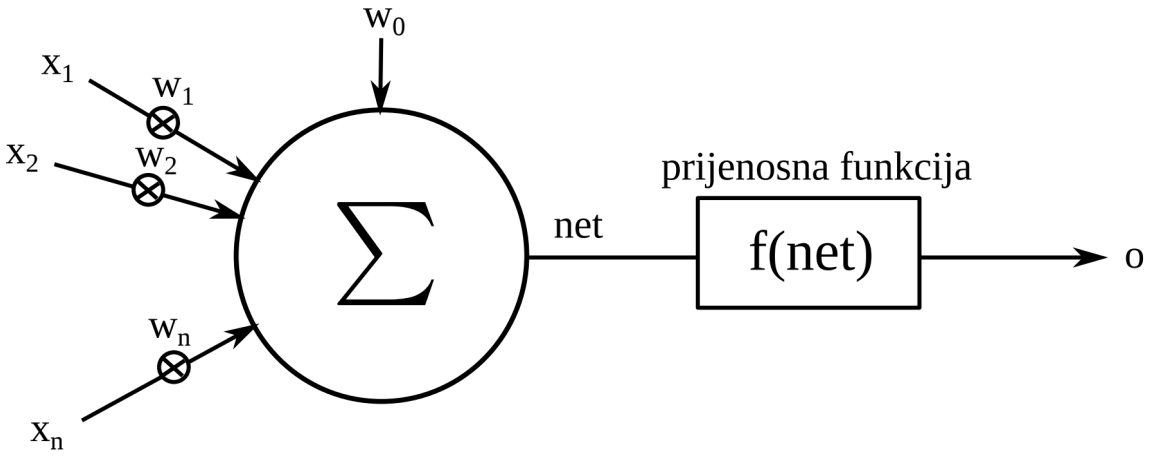
Područje koje se bavi umjetnim neuronskim mrežama zove se neuro-računarstvo, koje čini granu računarstva iz skupine mekog-računarstva. Meko računarstvo objedinjuje pristupe čiji je cilj izgraditi tehničke sustave za rješavanje teških problema s čestim šumom i nepouzdanošću u ulaznim podacima. [8]

Umjetni neuron je najmanja funkcionalna jedinka neuronske mreže i modelira biološki neuron u ljudskom mozgu. Umjetni neuroni imaju ulogu jednostavne procesne jedinice. Svaki umjetni neuron na ulazu prima nekakve numeričke vrijednosti (izlazi drugih neurona ili ulazni podaci), obično označene kao vektor $\vec{x}$. Svaki ulazni podatak u neuron x_i se prvo množe s težinom veze w_i preko koje je došao do trenutnog neurona. Dobiveni umnošci se zbrajaju, te im se dodatno pribraja takozvani pomak (eng. *bias*), obično označavan s w_0. Tako nastala suma se obično označava s net te se može izraziti formulom 3.1 Ako se pak vektor ulaznih podataka umjetno proširi s komponentom x_0 koja ima fiksnu vrijednost 1, onda se formula za računanje težinske sume ulaza može još jednostavnije izraziti, kao što je prikazano izrazom 3.2

$$\text{net} = \sum_{i=1}^{n} w_i \cdot x_i + w_0 \tag{3.1}$$

$$\text{net} = \sum_{i=0}^{n} w_i \cdot x_i = \vec{w} \cdot \vec{x} \tag{3.2}$$

Takva se suma provlači kroz prijenosnu funkciju. Rezultat prijenosne funkcije se postavlja na izlaz umjetnog neurona te se propagira dalje na ulaze drugih neurona ili na izlaz neuronske mreže. Model ovako opisanog neurona prikazan je na slici 3.1. Postoje razne prijenosne funkcije različitih složenosti koje se mogu koristiti kod umjetnih neurona, a neke od njih su: funkcija identiteta, funkcija skoka, sigmoidna funkcija, tangens hiperbolni, funkcija zglobnica, funkcija propusna zglobnica [8].



Slika 3.1. Model umjetnog neuron (preuzeto iz [8])

Umjetni neuroni se povezuju u paralelne strukture različitih arhitektura koje omogućuju paralelnu obradu podataka [8]. Arhitektura umjetne neuronske mreže opisuje kako su povezani umjetni neuroni u mreži i koliko ih ima. Postoje razne arhitekture neuronskih mreža, što jednostavnih, što kompleksnih, a svaka ima svoje prednosti i područja u kojima ima primjenu.

Umjetni neuroni se u umjetnim neuronskim mrežama često organiziraju u slojeve. Tada razlikujemo ulazni sloj, skrivene slojeve i izlazni sloj. Skriveni slojevi i izlazni sloj se sastoje od umjetnih neurona dok ulazni sloj zapravo ne sadrži umjetne neurone kakvi su opisani u odjeljku iznad. Neuroni ulaznog sloja na svoje ulaze samo primaju ulazne podatke neuronske mreže te ih dalje prosljeđuju neuronima prvog skrivenog sloja.

Umjetne neuronske mreže danas imaju vrlo široku primjenu, od obrade slika i signala, prepoznavanja uzoraka, kompresije slika, upravljanja, financija... [8]. Glavne prednosti neuronskih mreža su vrlo dobra sposobnost generalizacije i učenja, čak i uz neprecizne ulazne podatke. Njihova glavna mana je nemogućnost interpretacije njihovog ponašanja, iako su računalni znanstvenici počeli razvijati metode za dedukciju strategija koje neuronske mreže nauče [7]. Znanje koje neuronske mreže nauče iz podataka je implicitno pohranjeno u način povezivanja neurone i težine veza između njih [8]. Stoga je teško razumjeti zašto je neuronska mreža za neki ulaz dala baš taj izlaz. Također, teško je prepoznati kakvu konkretnu ulogu u sveukupnoj obradi ulaza ima pojedini neuron u mreži.

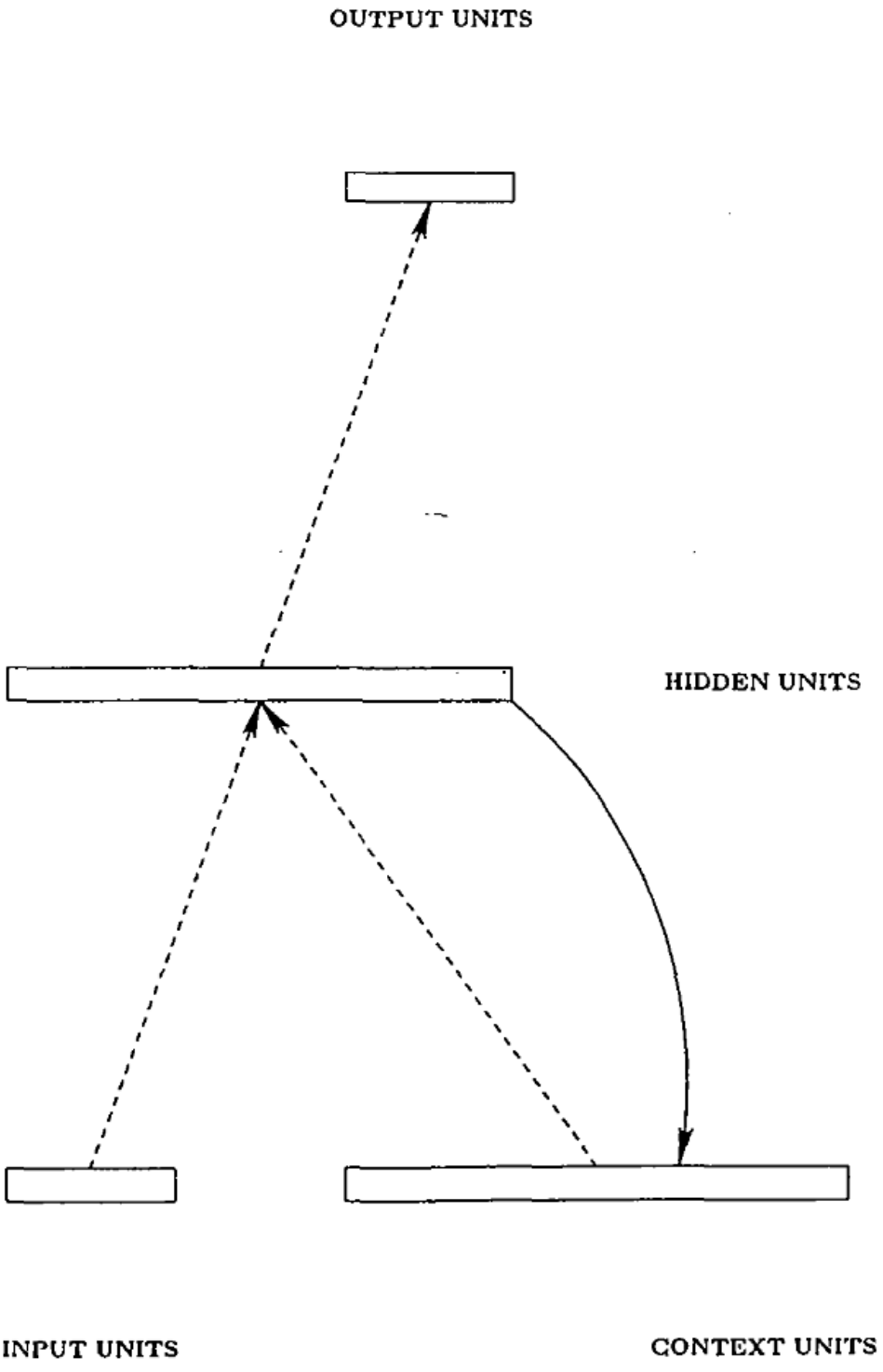
3.1. Elmanova neuronska mreža

Elmanova neuronska mreža je jednostavna povratna neuronska mreža (*eng. Recurrent Neural Network*). Glavna karakteristika takvih mreža je da imaju neku vrstu povratne veze ili vremenski odgođene veze [9].

Elmanova neuronska mreža se u osnovnoj varijanti sastoji od ulaznog sloja, jednog skrivenog sloja i izlaznog sloja. Skriveni sloj ove mreže ima dodatan skriveni kontekst. Kontekst čine neuroni koji služe za pohranu kontekstnih podataka. Dodatno postoje veze iz izlaza skrivenog sloja prema kontekstu i veze iz konteksta prema ulazu u skriveni sloj. Izlaz iz skrivenog sloja se preko tih veza sprema u kontekst. Kod obrade sljedećeg ulaza u neuronsku mrežu skrivenom se sloju na ulaz dodatno stavljaju vrijednosti iz konteksta preko pripadajućih veza te se ukupan ulaz mreže i konteksta dalje obrađuje [10]. Jednostavna shema Elmanove mreže prikazana je na slici 3.2.

Ovime se omogućava da Elmanova mreža može pamtiti prošla stanja. Na ovaj način prijašnji ulazi u neuronsku mrežu imaju utjecaj na rezultat obrade nekog budućeg ulaza.

Ovo je korisno svojstvo za model inteligentnog igrača Zatvorenikove dileme, s obzirom na to da je prirodno da prijašnje interakcije s drugim igračem utječu na nove odluke.



Slika 3.2. Shema Elmanove mreže (preuzeto iz [10])
Napomena: iscrtkane linije označuju veze koje se mogu trenirati

4. Evolucijsko računanje

Evolucijsko računanje je grana umjetne inteligencije koja se najčešće bavi rješavanjem optimizacijskih problema. Začeci ove grane sežu još u kasne 50-e godine prošlog stoljeća. Evolucijsko računanje se može podijeliti u tri grane [11]:

- *evolucijske algoritme*
- *algoritme rojeva*
- *ostale algoritme*

Neki od algoritama iz grane evolucijskih algoritama su evolucijske strategije, evolucijsko programiranje, genetski algoritam i genetsko programiranje [11].

U grani algoritma rojeva nalaze se mravlji algoritmi, algoritam roja čestica, algoritmi pčela i drugi [11].

Neki do značajnih algoritama iz grane ostali algoritmi pripadaju umjetni imunološki algoritmi, algoritam diferencijske evolucije i algoritam harmonijske pretrage [11].

Područje evolucijskog računanja također pripada području metaheuristika. Metaheuristika je skup algoritamskih koncepata koji se koriste za definiranje heurističkih metoda primjenjivih na široki skup problema [11]. Heurističke metode su algoritmi koji pronalaze rješenja problema koja su zadovoljavajuće dobra, ali ne nužno optimalna. Heurističke metode obično imaju relativnu nisku računsku složenost [11].

4.1. Optimizacija

Kao što je ranije navedeno, algoritmi iz područja evolucijskog računanja se često koriste u problemima optimizacije. Optimizaciju možemo definirati kao postupak pronalaženja

najboljeg rješenja problema. Uz svako rješenje se pridružuje funkcija dobrote ili funkcija kazne. Ako se rješenjima pridružuje funkcija dobrote, optimizacijski postupak će tražiti rješenje tako da pokuša maksimizirati funkciju dobrote. Ako se pak rješenjima pridružuje funkcija kazne, optimizacijski postupak će pokušati minimizirati funkciju kazne.[11]

Bitno je reći da kada gledamo optimizacijske algoritme iz područja evolucijskog računanja, niti jedan od njih se ne može istaknuti kao generalno najbolji algoritam. Svaki od tih algoritama je dobar u određenim primjenama, dok u drugim primjenama drugi optimizacijski algoritmi daju bolje rezultate. U prosjeku nema značajnih razlika u globalnim performansama kada se gleda prosjek po svim mogućim funkcijama cijene.[11]

4.2. Genetski algoritam

Genetski algoritam je algoritam iz grane evolucijskih algoritama. Općenito, svi pristupi iz grane evolucijskih algoritama crpe inspiraciju iz Darwinove teorije o postanku vrsta, koja se temelji na pet pretpostavki [11]:

1. potomaka uvijek ima više nego je potrebno
2. veličina populacije je približno stalna u vremenu
3. hrana je ograničen resurs
4. ne postoje dvije potpuno identične jedinke kod vrsta koje se seksualno razmnožavaju – postoje varijacije
5. većina varijacija se prenosi nasljeđem

Genetski algoritam je jedan od algoritama koji rješavaju optimizacijske probleme, a koji izravno utjelovljuje navedene postavke. Genetski algoritam zapravo nije jedinstven, već se može implementirati na različite načine. [11]

Ideja genetskog algoritma je modeliranje evolucijskog procesa. U algoritmu postoji populacija jedinki, koje zapravo predstavljaju neka od mogućih rješenja problema. Cilj algoritma je kroz selekciju roditelja, stvaranje novih jedinki korištenjem operatora križanja nad roditeljima i korištenje operatora mutiranja nad novim jedinkama stvoriti naj-

bolje ili barem dovoljno dobro rješenje problema. Selekcija roditelja bi trebala ovisiti o nekakvoj metrici uspješnosti - uspješniji roditelji bi generalno trebali imati veće izglede da budu odabrani za križanje. Ta metrika se obično naziva dobrota. Takvim postupkom se očekuje da će se kontinuiranim križanjem uspješnih roditelja i mutiranjem njihove djece stvarati bolja rješenja. Ovo je zapravo svojevrstan ekvivalent prirodnoj selekciji, to jest preživljavanju najboljih.

Varijacije u implementaciji se mogu pojavljivati u načinu mutiranja, križanja, selekcije, ali i magnitudi izmjena u populaciji u jednoj iteraciji algoritma.

Tako u jednoj krajnosti imamo takozvani *eliminacijski genetski algoritam* u kojem se u jednoj iteraciji samo jedna jedinka zamjenjuje jednim novonastalim djetetom. Novo dijete već u sljedećem koraku može biti roditelj. [11]

U drugoj krajnosti imamo *generacijski genetski algoritam*. U njemu se u jednoj iteraciji iz jedne populacije biraju roditelji čija se djeca spremaju u novu populaciju sve dok nova populacije ne dosegne veličinu populacije roditelja. Tada populacija djece zamjenjuje populaciju roditelja. U ovoj varijanti dijete može biti roditelj tek kada populacija djece postane populacija roditelja, to jest u sljedećoj iteraciji. [11]

Operator križanja se primjenjuje nad dvije jedinke (roditelji) i iz njih stvara dijete. Može se implementirati na razne načine, ali generalno ovaj operator često vrši kontrakciju nad populacijom. To znači da se primjenom ovog operatora obično gubi raznolikost između jedinki te one u prosjeku postaju sve sličnije. [11]

Operator mutacije ima upravo suprotno svojstvo. On vrši nasumične izmjene nad jedinkama i time im vraća raznolikost. Njegova važna uloga je izbaciti populaciju iz lokalnih maksimuma u prostoru rješenja [11].

5. Implementacija inteligentnog igrača

Inteligentni igrač Zatvorenikove dileme modeliran je Elmanovom neuronskom mrežom. Mreža se sastoji od ulaznog sloja, jednog skrivenog sloja s kontekstom, još jednog skrivenog sloja i izlaznog sloja. Svi slojevi su potpuno povezani. Veze između izlaza skrivenog sloja i konteksta su jedan-na-jedan, to jest izlaz jednog neurona se preslikava u "kopiju" tog neurona u kontekstu. To implicitno znači da skriveni kontekst ima jednak broj neurona kao i sam skriveni sloj. Veze između konteksta i ulaza u neurone skrivenog sloja su svaki-sa-svakim, isto kao i između ostalih slojeva u mreži. Svi neuroni u korištenoj neuronskoj mreži koriste sigmoidnu prijenosnu funkciju.

Korištena mreža je arhitekture 1 x 4 x 2 x 1. Ulaz u mrežu je brojčana vrijednost odluke suparničkog igrača iz prijašnje runde. Ako prijašnje runde još nije bilo, tada se na ulaz dovodi 0. Odluka o suradnji je brojčano kodirana vrijednošću 1, a odluka o izdaju vrijednošću -1. Izlaz mreže je brojčana vrijednost između 0 i 1 koja se interpretira kao potez inteligentnog igrača u ovoj rundi Zatvorenikove dileme. Vrijednost se interpretira na sljedeći način:

- ako je vrijednost veća ili jednaka 0.5, interpretiraj izlaz mreže kao suradnju
- ako je vrijednost manja od 0.5, interpretiraj izlaz mreže kao izdaju

5.1. Učenje inteligentnog igrača

S obzirom na karakteristike Zatvorenikove dileme i dinamiku okruženja u kojem inteligentni igrač mora donositi odluke, često korišteno nadzirano učenje neuronske mreže algoritmom propagacije unazad ovdje nije primjenjivo.

Ako bismo koristili algoritam propagacije unazad, morali bismo za svaki ulaz definirati očekivani (najbolji) izlaz, to jest optimalnu odluku u toj situaciji. S obzirom na to da u ovom problemu postoje samo 3 moguća ulaza u mrežu (0 ako se radi o prvoj rundi, 1 ako je u prijašnjoj rundi suparnički igrač surađivao i -1 ako je izdao ovoga igrača), skup za učenje bi bio premalen. Također, problem u tom pristupu je to što bismo time inteligentnom igraču zapravo predstavljali problem Zatvorenikove dileme sa samo jednom iteracijom, a taj problem ima potpuno različit pristup u odnosu na ponavljajuću Zatvorenikovu dilemu, kao što je prije raspravljeno. Još jedan problem je u tome što je zapravo teško odrediti što je najbolji potez koji igrač može odigrati u nekom okruženju.

Zbog toga je za ovaj problem odlučeno koristiti jednu drugu tehniku, a to je učenje neuronske mreže genetskim algoritmom. Ideja ove tehnike je koristiti genetski algoritam kao algoritam optimizacije nad neuronskim mrežama. Cilj optimizacije je pronaći težine veza između neurona (i pomake svakog neurona) koje će davati najbolje (ili barem dovoljno dobre) performanse neuronske mreže u igri Zatvorenikove dileme. Ova tehnika nam omogućuje da težine veza između neurona prilagođavamo nakon što je inteligentni igrač odigrao nekoliko igra, kada imamo bolju predodžbu o kvaliteti njegovog igranja.

5.1.1. Parametri genetskog algoritma

Za genetski algoritam nad neuronskom mrežom je samo potrebno definirati kriterij i postupak nagrađivanja te rangiranja neuronskih mreža, način selekcije roditelja, operatore križanja i mutacije s njihovim parametrima, magnitudu izmjena u populaciji, veličinu populacije i kriterij za završetak algoritma.

Kao kriterij nagrađivanja i rangiranja neuronskih mreža iz populacije koriste se bodovi dobiveni igranjem Zatvorenikove dileme. Svaka mreža iz populacije igra zadani broj rundi Zatvorenikove dileme sa svakom drugom neuronskom mrežom i sa samom sobom. Dodatno, svaka mreža igra Zatvorenikovu dilemu 4 puta protiv 11 unaprijed pripremljenih strategija za igranje Zatvorenikove dileme. Jedna od tih strategija, nazvana *Detective*, preuzeta je s interaktivne stranice *The Evolution of Trust* koja prezentira zanimljive aspekte Zatvorenikove dileme [12]. Ostalih 10 strategija preuzeto iz prvog Axelrodovog turnira. Strategije su implementirane po opisima dostupnima u Axelrodovom radu u kojem je objavio rezultate prvog turnira [2]. Također, za referencu je korišten i izvorni

kod implementacije nekih od strategija u programskom jeziku Python pronađen na internetu [13]. Implementirane strategije iz Axelrodovog turnira su:

- *Tit for Tat*
- *Nydegger*
- *Grofman*
- *Shubik*
- *Friedman*
- *Davis*
- *Downing*
- *Joss*
- *Tullock*
- *Random*

Valja napomenuti da nije pronađena službena dokumentacija o točnoj implementaciji navedenih strategija. Dodatno, kod testiranja implementacija nisu dobiveni potpuno identični bodovi kod međusobnog igranja Zatvorenikove dileme između strategija kao kod objavljenog Axelrodovog rada. Stoga nema garancije da se sve implementirane strategije ponašaju potpuno identično kao što su se ponašale strategije u Axelrodovom turniru.

Broj rundi koje će mreže odigrati u jednoj igri Zatvorenikove dileme može se podešavati u programskom rješenju. Dobiveni bodovi kod međusobnog igranja između mreža se zbrajaju, dok se kod igranja mreže protiv strategije zbraja prosjek ostvarenih bodova protiv te strategije. Dobivena suma predstavlja takozvanu dobrotu neuronske mreže. Ti bodovi se računaju svaki puta nakon stvaranja nove generacije neuronskih mreža. Ovaj način nagrađivanja savršeno odgovara za problem ponavljajuće Zatvorenikove dileme jer ovime nije nužno da neuronska mreža mora pobjeđivati druge neuronske mreže u direktnom okršaju, već je bitno da kontinuirano može dobivati dosta bodova protiv ne-

uronskih mreža s raznim strategijama.

Za selekciju roditelja je korištena proporcionalan selekcija - vjerojatnost da će neuronska mreža biti odabrana za roditelja je proporcionalna njezinim relativnim bodovima u odnosu na druge neuronske mreže.

Kod operatora križanja se za određivanje parametara[1] djeteta nasumično biraju ekvivalentni parametri jednog od roditelja. Odabir roditelja od kojeg će se naslijediti konkretan parametar se ponovno vrši proporcionalnom selekcijom. Dakle, za svaki parametar vrijedi da je vjerojatnost da će on biti naslijeđen od nekog roditelja je proporcionalna relativnoj dobroti tog roditelja u odnosu na drugog roditelja.

Operator mutacije koristi slučajne brojeve iz Gaussove normalne razdiobe s očekivanom vrijednosti 0 i podesivom standardnom devijacijom. Nakon stvaranja novog djeteta, svaki parametar ima vjerojatnost da će nad njim biti izvršen operator mutacije. Mutacija funkcionira tako da se generira slučajan broj te se dobiveni broj pribraja vrijednosti parametra. Ovime se dobivaju mutacije koje su uglavnom malog intenziteta, no mogu biti i velikog. Tako preko jedne mutacije dobivamo kombinaciju češćih malih mutacija i rjeđih velikih mutacija. Vjerojatnost pojavljivanja mutacije i standardnu devijaciju korištenu kod generiranja slučajnog broja moguće je podešavati u programskom rješenju.

Korišten je *generacijski genetski algoritam*. Dakle, u jednoj iteraciji se cijela populacija jedinki zamjenjuje. Iznimka je jedino da se najbolja mreža iz prošle generacije direktno kopira u novu generaciju bez izmjena. Time se nastoji očuvati svojstvo elitizma [2].

Kriterij zaustavljanja genetskog algoritma je isključivo dosezanje limita za broj generiranih generacija koji se može podesiti u programskom rješenju. Također, u programskom rješenju se može podesiti veličina populacije neuronskih mreža za genetski algoritam.

[1] Za parametre neuronske mreže podrazumijevaju se težine veza između neurona i pomaci (eng. *biases*)
[2] Elitizam je svojstvo algoritma da ne može izgubi najbolje pronađeno rješenje [11]

6. Definiranje pokusa

Pokus je zamišljen tako da se 20 puta pokrene treniranje neuronskim mreža. Na kraju svakog treniranja se izvlači najbolja neuronska mreža iz zadnje generacije te se s njom igra Zatvorenikova dilema s ciljem da se utvrdi postojanje određenih svojstava u strategiji igranja te neuronske mreže. Svojstva koja će se ispitivati su:

- Ljubaznost (eng. *niceness*)
- sklonost opraštanju (eng. *forgiveness*)
- provokabilnost (eng. *provocability*)

Navedena svojstva su opisana ranije u poglavlju 2. Promatrat će se učestalost pojavljivanja i metrika pojedinog svojstva u strategijama najboljih igrača. Iz toga će se pokušati izvesti zaključak o tome jesu li i koliko navedena svojstva robusna i poželjna u generalnoj Zatvorenikovoj dilemi neovisnoj o drugim igračima.

Svojstva će se ispitivati i kvantificirati tako da se protiv svake izabrane neuronske mreže igra Zatvorenikova dilema od 16 rudni nekoliko puta.

Igrač ispitivač protiv inteligentnog igrača prvo igra Zatvorenikovu dilemu tako da uvijek igra suradnju protiv inteligentnog igrača. Ako inteligentni igrač barem jednom odigra izdaju, smatra se da nije ljubazan, a u suprotnom da jest.

U drugoj igri prvi potez igrača ispitivača je suradnja. Zatim ispitivač igra izdaju dok god drugi igrač igra suradnju. Kada inteligentni igrač odigra izdaju, ispitivač igra suradnju dok god se inteligentni igrač ne "smiri" i ponovno igra suradnju. Nakon toga ispitivač ponovno igra izdaju i ponavlja postupak dok igra traje. Omjer broja ispitivačevih izdaja do reakcije inteligentnog igrača i broj reakcija inteligentnog igrača će biti metrika

opraštanja. Na primjer, omjer 15/0 bi značio da je inteligentni igrač cijelu igru surađivao i nikada nije reagirao na 15 izdaja ispitivača. Takav inteligentni igrač bi se smatrao maksimalno opraštajući. Omjer 2/14 bi predstavljao igru u kojoj je nakon prve izdaje ispitivača inteligentni igrač odlučio igrati izdaju do kraja igre. Taj igrač bi bio minimalno opraštajući.

U trećoj igri prvi potez igrača ispitivača je ponovno suradnja. Zatim ispitivač igra jednu izdaju i zatim surađuje dok god inteligentni igrač igra izdaju protiv njega. Nakon što inteligentni igrač prestane igrati izdaju ispitivač ponavlja izdaju i ponavlja postupak. Kao metrika provokabilnosti će se uzimati omjer izdaja ispitivača i trenutnih reakcija inteligentnog igrača na izdaju, neovisno koliko izdaja bilo u reakciji. Tako bi na primjer omjer 5/5 predstavljao inteligentnog igrača koji je nakon svake od 5 ispitivačevih izdaja odmah uzvratio s barem jednom izdajom. Takav igrač bi se smatrao maksimalno provokabilnim. Omjer 8/0 bi predstavljao igru u kojoj je ispitivač 8 puta izdao inteligentnog igrača, a on nije niti jednom uzvratio izdajom u slijećem potezu. Taj igrač se smatra minimalno provokabilnim.

Bitan komentar je da kod inteligentnih igrača koji nisu ljubazni metrike za opraštanje i provokabilnost nisu toliko relevantne i smislene. To je zbog činjenice da kada neljubazan igrač igra izdaju to ne mora biti nužno reakcija na izdaju drugog igrača, već ta izdaja može biti odigrana na vlastitu inicijativu. Unatoč tome, metrike za opraštanje i provokabilnost će biti priložene i za inteligentne igrače koji nisu ljubazni.

Kod treniranja neuronskih mreža korištene su sljedeće postavke genetskog algoritma:

- vjerojatnost mutacije: 0.05
- magnituda mutacije (standardna devijacija kod generiranja slučajnog broja): 0.5
- veličina generacije: 50
- limit broja generiranih generacija: 30
- broj rundi u jednoj igri Zatvorenikove dileme: 50

7. Rezultati i rasprava

Tablica 7.1.: Rezultati pokusa

Redni broj pokusa	**Ljubaznost**	**Opraštanje**	**Provokabilnost**	**Komentar**
1	da	10/5	5/5	sličan strategiji *Tit for Tat*, blaži
2	da	10/5	5/5	sličan strategiji *Tit for Tat*, blaži
3	da	8/8	5/5	jako sličan strategiji *Tit for Tat*
4	da	9/6	8/0	sličan strategiji *Tit for Tat*, blaži; jednom izdao sa odgodom (istovremeno kao i ispitivač)
5	ne	2/14	1/1	nikada ne surađuje
6	ne	2/14	1/1	nikada ne surađuje
7	da	10/5	6/3	kao **1**, ali još blaži
8	da	8/8	5/5	*Tit for Tat*
9	ne	2/14	1/1	nikada ne surađuje
10	da	8/8	5/5	*Tit for Tat*

Continued on next page

Tablica 7.1.: Rezultati pokusa (Continued)

11	ne	2/14	1/1	nikada ne surađuje
12	ne	2/14	1/1	nikada ne surađuje
13	da	8/8	5/5	*Tit for Tat*
14	ne	2/14	1/1	nikada ne surađuje
15	da	8/8	5/5	*Tit for Tat*
16	ne	2/14	1/1	nikada ne surađuje
17	ne	2/14	1/1	nikada ne surađuje
18	da	8/8	5/5	*Tit for Tat*
19	da	12/4	6/3	jako blagi, s vremenom manje oprašta
20	da	8/8	5/5	*Tit for Tat*

12/20 ispitanih inteligentnih igrača je bilo ljubazno. Od toga je 6 njih igralo identično kao strategija *Tit for Tat*. 5 inteligentnih ljubaznih igrača je bilo dosta slično strategiji *Tit for Tat*, ali su bili skloniji opraštanju. Inteligentni igrač pod brojem **19** je bio zanimljiv jer je bio jako blag prema ispitivaču. No, u malo detaljnijem ispitivanju vidjelo da se da taj igrač s vremenom sve više sliči na strategiju *Tit for Tat* kako ga se više izdaje.

Zanimljivo je da su svi neljubazni igrači bili potpuno identični - svi su stalno igrali izdaju neovisno o drugom igraču.

7.1. Rasprava

Iako je kod ljubaznih igrača bilo dosta sličnosti, ipak su postojale razlike u ponašanju kod nekih. Postojala je varijanta igrača koja je bila jako oprostiva i slabo provokabilna, ali je znala s vremenom postajati stroža. Najstroža varijanta inteligentnog igrača je bila ona identična strategiji *Tit for Tat*. Zanimljivo je da niti jedan ljubazan igrač nije bio potpuno naivan tako da je stalno igrao suradnju neovisno o drugom igraču.

Kod neljubaznih igrača nažalost nije bilo nikakve raznolikosti u ponašanju. Svi su

igrali najjednostavniju strategiju stalne izdaje neovisno o potezima drugog igrača. Šteta je što kroz treniranje inteligentnog igrača nije nastao niti jedan neljubazan igrač koji bi imao neku složeniju strategiju kojom bi pokušao iskoristiti drugog igrača. Možda bi složeniji model inteligentnog igrača uspio naučiti složeniju neljubaznu strategiju igranja ponavljajuće Zatvorenikove dileme.

8. Zaključak

Ovakvi rezultati eksperimenta su uglavnom očekivani, ali ne i u potpunosti zadovoljavajući.

U ovom jednostavnom eksperimentu strategija *Tit for Tat* se pojavila čak 6 puta, a preostalih 6 ljubaznih taktika je uglavnom imalo dosta sličnosti s tom strategijom. Zbog toga bismo mogli zaključiti da je ta strategija dosta robusna i na neki način prirodan pristup problemu ponavljajuće Zatvorenikove dileme.

No, ne treba zanemariti činjenicu da se u postupku učenja inteligentnog igrača čak 8 buta istrenirao igrač koji je nikada nije surađivao. To se može objasniti činjenicom da najbolja strategija igranja ponavljajuće Zatvorenikove dileme ovisi o okruženju u kojem se Zatvorenikova dilema igra. S obzirom na to da je inicijalno okruženje kod treniranja inteligentnog igrača bilo nasumično, vjerojatno je puno igrača iz populacije za treniranje prvotno cijelo vrijeme igralo izdaju. Ako je takvih igrača bilo dovoljno, vjerojatno je bilo nemoguće razviti neko drugačije ponašanje koje bi se moglo istaknuti u takvom okruženju. No, ipak činjenica da je 8 od 20 igrača razvila ovakvo nekooperativno ponašanje ukazuje da je pristup stvaranja inteligentnog igrača možda mogao biti i bolji. Možda bi model inteligentnog igrača s neuronskom mrežom složenije arhitekture i s više ulaznih podataka u postupku treniranja rjeđe rezultirao igračem koji nikada ne surađuje. Bilo bi zanimljivo vidjeti ako bi složeniji model ponekada naučio neljubaznu strategiju koja na pametan način pokušava iskoristiti druge igrače.

Literatura

[1] Zatvorenikova dilema. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2013–2024. https://enciklopedija.hr/clanak/zatvorenikova-dilema; pristupljeno 9.3.2024.

[2] R. Axelrod, "Effective choice in the prisoner's dilemma", *The Journal of Conflict Resolution*, sv. 24, br. 1, str. 3–25, ožujak 1980. [Mrežno]. Adresa: https://doi.org/10.1177/002200278002400101

[3] ——, "More effective choice in the prisoner's dilemma", *The Journal of Conflict Resolution*, sv. 24, br. 3, str. 379–403, rujan 1980. [Mrežno]. Adresa: https://doi.org/10.1177/002200278002400301

[4] M. E. I. Barković Bojanić, "Teorija igara i pravo", *Pravni vjesnik*, sv. 29, br. 1, str. 59–76, travanj 2013. [Mrežno]. Adresa: https://hrcak.srce.hr/111004

[5] Igra. Hrvatska enciklopedija, mrežno izdanje. Leksikografski zavod Miroslav Krleža, 2013–2024. https://enciklopedija.hr/clanak/70136; pristupljeno 8.3.2024.

[6] L. A. Dugatkin, "The evolution of cooperation", *BioScience*, sv. 47, br. 6, str. 355–362, 1997. [Mrežno]. Adresa: http://www.jstor.org/stable/1313150

[7] Explained: Neural networks. MIT News, 2017. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414; pristupljeno 23.5.2024.

[8] M. Čupic, Umjetne neuronske mreže, skripta iz kolegija Uvod u umjetnu inteligenciju, 2018. http://java.zemris.fer.hr/nastava/ui/ann/ann-20180604.pdf; pristupljeno 24.5.2024.

[9] Implementation of Elman Recurrent Neural Network in WEKA. ICT Research Blog by John Salatas, 2018. https://jsalatas.ictpro.gr/implementation-of-elman-recurrent-neural-network-in-weka/; pristupljeno 22.5.2024.

[10] J. L. Elman, "Finding structure in time", *Cognitive Science*, sv. 14, br. 2, str. 179–211, ožujak 1990. https://doi.org/https://doi.org/10.1207/s15516709cog1402_1

[11] M. Čupic, Evolucijski računarstvo, skripta iz kolegija Uvod u umjetnu inteligenciju, 2019. http://java.zemris.fer.hr/nastava/ui/evo/evo-20190604.pdf; pristupljeno 24.5.2024.

[12] The Evolution of Trust by Nicky Case, July 2017. https://ncase.me/trust/; pristupljeno 7.6.2024.

[13] Source code for axelrod.strategies.axelrod_first, 2015. https://axelrod.readthedocs.io/en/fix-documentation/_modules/axelrod/strategies/axelrod_first.html; pristupljeno 7.6.2024.

Sažetak

Inteligentni igrač za Zatvorenikovu dilemu

Petar Belošević

Zatvorenikova dilema je poznati problem iz područja teorije igara. U jezgri dileme je sukob između racionalnog razmišljanja maksimiziranja vlastite dobiti dvaju igrača koje ih dovodi do sub-optimalne pozicija te suradnje koja je korisna za oba igrača. Ovaj rad se bavi problemom izrade inteligentnog igrača koji će moći igrati ponavljajući problem Zatvorenikove dileme. Za modeliranje inteligentnog igrača korištena je Elmanova neuronska mreža. Mreža je trenirana generacijskim genetskim algoritmom. Pokusom je pokušano utvrditi učestalost pojedinih karakteristika strategije igranja inteligentnog igrača. Time se želi utvrditi robusnost strategija igranja Zatvorenikove dileme koje posjeduju tražene karakteristike. Karakteristike koje se promatraju su: ljubaznost, sklonost opraštanju i provokabilnost.

Ključne riječi: Zatvorenikova dilema, umjetna inteligencija, umjetne neuronske mreže, inteligentni igrač, Elmanova neuronska mreža

Abstract

Intelligent player for Prisoner's Dilemma

Petar Belošević

Prisoner's Dilemma is famous problem in game theory. The core of the Dilemma is the conflict between rational thinking of maximizing personal gain that brings two players in sub-optimal position and mutual cooperation that brings benefits for both players. This thesis has goal to create intelligent player for iterated Prisoner's Dilemma. For modelling an intelligent player was used Elman's neural network. The network was trained with generational genetic algorithm. Experiment was used to determine frequency of specific characteristics in the strategies used by the intelligent player. Goal of the experiment is to show how robust strategies that have specified characteristics are. Characteristics that were observed are: niceness, forgiveness and provocability.

Keywords: Prisoner's dilemma, Artificial intelligence, Artificial neural networks, Intelligent player, Elman's neural network

Privitak A: Programski kod

U sklopu ovog završnog rada izrađena je jednostavna aplikacija u programskom jeziku *Java*. Aplikacija omogućuje treniranje Elmanove neuronske mreže genetskim algoritmom i igranje Zatvorenikove dileme.

Aplikacija u početnom prozoru prikazuje 4 gumba pomoći kojih nudi sljedeće mogućnosti:

- Treniranje neuronske mreže
- Igranje Zatvorenikove dileme
- Učitavanje postojeće neuronske mreže
- Prikaz kratkog opisa aplikacije u odvojenom prozoru.

Kod izbora treniranja neuronske mreže prikazuje se prozor u kojem se mogu podesiti sljedeći parametri genetskog algoritma:

- Vjerojatnost mutacije
- Magnituda mutacije (standardna devijacija Gaussove normalne razdiobe)
- Veličina populacije
- Najveći broj generiranih generacija prije završavanja algoritma
- Broj iteracija u jednoj igri Zatvorenikove dileme prilikom određivanja dobrote svake mreže

Nakon podešavanja parametara pokreće se genetski algoritam. Program je podešen tako da sprema najbolju, medijalnu i najlošiju neuronsku mrežu svake pete generacije u

direktorij *src/main/resources* Maven projekta. Na dnu prozora je prikazan teoretski najveća moguća dobrota koju neka mreža može ostvariti. Dodatno, za svaku se generaciju ispisuju sljedeći podaci:

- Dobrota najbolje mreže iz te generacije
- Dobrota medijalne mreže iz te generacije
- Dobrota najlošije mreže iz te generacije

Genetski algoritam se može prisilno završiti ranije nego je predviđeno pritiskom na gumb *Stop*.

Po završetku genetskog algoritma omogućeno je da korisnik igra Zatvorenikovu dilemu protiv najbolje mreže iz posljednje generacije genetskog algoritma. Igra se sastoji od 16 rundi Zatvorenikove dileme. Prozor prikazuje ostvarene bodove svakog igrača u svakoj rundi te ukupno ostvarene bodove. Neuronska mreža uvijek igra kao *Player 2*, dok je korisnik *Player 1*.

Kod izbora igranja Zatvorenikove dileme priprema se Zatvorenikova dilema za dva korisnička igrača. Prvo se prikazuje prozor koji prikazuje kontrole za oba igrača. kontrole za svakog igrača su prikazane u tablici A1. i stalno su iste.

Tablica A1.: Kontrole pojedinog igrača

	Suradnja	Izdaja
Player 1	W	S
Player 2	P	L

Nakon toga se prikazuje prozor koji prikazuje osvojene bodove svakog igrača na identičan način kao i prilikom igranja Zatvorenikove dileme s inteligentnim igračem nakon provođenja genetskog algoritma. Igra se ponovno sastoji od 16 rundi.

Kod odabira učitavanja neuronske mreže iz datoteke otvara se prozor za odabir datoteke. Nakon što korisnik odabere datoteku aplikacija će pokušati učitati neuronski mrežu iz datoteke. Po završetku uspješnog učitavanja mreže pokreće se igra identično kao i nakon treniranja neuronske mreže genetičkim algoritmom.

Za programsko rješenje korištena je *Java 17*. Programski kod je organiziran kao Maven projekt. Dodatno je korištena vanjska Maven biblioteka *nd4j-native-platform*. Korištena biblioteka nudi razne optimizirane operacije vezane uz linearnu algebru. Ono što je bilo bitno za ovaj projekt je bilo mogućnost optimiziranog matričnog množenja koje se koristi kod izračunavanja izlaza neuronske mreže. Za izradu grafičkog korisničkog sučelja (eng. *GUI*) korištena je biblioteka *Swing* koja je dio standardne biblioteke programskog jezika *Java*.

Izvorni kod programskog rješenja i sam završni rad dostupni su na javnom Git repozitoriju na poveznici https://github.com/PetarBelosevic/Prisoner-s-Dilemma-AI.git. Za korištenje programskog rješenja potrebno je preuzeti Git repozitorij, prevesti preuzeti Maven projekt (u naredbenom retku pomoću naredbe *mvn compile*) i pokrenuti izvođenje iz razreda *application.GUIApp* (u naredbenom retku pomoću naredbe *mvn exec:java -Dexec.mainClass="application.GUIApp"*).

U programskom kodu dostupno je nekoliko razreda koji nude slične funkcionalnosti kao i aplikacija s grafičkim korisničkim sučeljem. Korišteni su za isprobavanje funkcionalnosti prije izrade grafičkog korisničkog sučelja.

Prvi takav razred je *game.application.ConsoleApp* koji omogućava igranje Zatvorenikove dileme koju igraju dva korisnika preko konzole. Kod oba igrača unos teksta "c" ili "C" se interpretira kao suradnja, a sve ostalo kao izdaja.

Sljedeći takav razred je *evolution.training.ConsoleTestingApp*. Razred omogućuje učitavanje neuronske mreže iz datoteke te igranje Zatvorenikove dileme protiv te mreže preko konzole.

Posljednji takav razred je *evolution.training.ConsoleTrainingApp*. Razred omogućava provođenje genetičkog algoritma kao i kod aplikacije s grafičkim sučeljem. Postupak u konzoli ispisuje dobrote najbolje, medijalne i najlošije neuronske mreže svake 5. generacije te iste neuronske mreže sprema u datoteke na disku.