Ekaia, 2020, 37, 305-325
https://doi.org/10.1387/ekaia.20831

ekaia
ZIENTZIA eta TEKNOLOGIA ALDIZKARIA

ISSN 0214-9001 – eISSN 2444-3255

Adimen artifizialaren erabilera buru-hausgarriak ebazteko: 15 makilen jokoa

(Use of artificial intelligence to solve a 2-player competitive game: the 15 sticks game)

Anne Idigoras, Beñat Galdós, Imanol Echeverría, Josune Ordóñez, Mayi Echeveste, Iñigo Lopez-Gazpio*

Deustuko Unibertsitatea (DU), Donostia, Euskal Herria

LABURPENA: Artikulu honetan Deustuko Unibertsitateko ikasle talde batek Adimen Artifiziala ikasgaiko kontzeptuak aplikatzen dituzte telebistako saio batean planteatzen den buru-hausgarri bat konputazionalki ebazteko: 15 makilen jokoa. 15 makilen jokoa buru-hausgarri motako joko bat da non bi parte-hartzaile azken makila ez hartzeko elkarren kontra lehiatzen diren. Arazo hau ebazteko, adimen artifiziala deskribatzen da, batez ere bilaketa arazoetan sakonduz, 15 makilen jokoa deskribatzen da eta soluzio konputazional baten metodologia planteatzen da ikasketa automatikoko metodoak erabiliz. Honela, inolako kanpo informaziorik erabiltzen ez duen software agente batek jokoa ebazten ikasten du bere buruaren aurka lehiatuz, eta jokoaren egoera irabazleak eta galtzaileak diskriminatzen ikasten du. Bukatzeko, garatutako software agentearen eta honek jarraitzen duen ikasketa prozesuaren gaineko analisi sakon bat egiten da. Azkenik, ondorioak zein etorkizuneko lanak planteatzen dira.

HITZ GAKOAK: Adimen artifiziala, jokoak, bilaketa espazioak, ikasketa automatikoa.

ABSTRACT: *In this paper students from the University of Deusto apply the concepts learned in the artificial intelligence subject to solve a 2-player competitive game proposed recently in a television program: the 15 sticks game. The 15 sticks game consists on a 2-player competitive game in which players must not pick the last stick. For the task, we describe artificial intelligence in the context of search spaces and game heuristics, define the 15 sticks game and propose a computational approach based on machine learning to solve it. The software agent is able to learn to solve the game efficiently by playing against itself through iterations with no external supervision. We finally provide extensive analysis on the developed software agent and remark the conclusions and future work.*

KEYWORDS: Artificial intelligence, games, search spaces, machine learning.

**** Harremanetan jartzeko / Corresponding author:*** Imanol Echeverría. Deustuko Unibertsitatea (DU), Donostia, Euskal Herria. – iecheverria@deusto.es – https://orcid.org/0000-0001-7134-6306.

Nola aipatu / How to cite: Idigoras, Anne; Galdós, Beñat; Echeverría, Imanol; Ordóñez, Josune; Echeveste, Mayi; Lopez-Gazpio, Iñigo (2020). «Adimen artifizialaren erabilera buru-hausgarriak ebazteko: 15 makilen jokoa»; *Ekaia*, 37, 2020, 305-325. (https://doi.org/10.1387/ekaia.20831).

Jasoa: 14 maiatza, 2019; Onartua: 20 abuztua, 2020.

ISSN 0214-9001 - eISSN 2444-3255 / © 2020 UPV/EHU

1. **SARRERA**

Gizakiok munduko usainak hautemateko, soinuak entzuteko, zaporeak dastatzeko, oroimenak gordetzeko, amesteko, hunkitzeko eta abar luze batez gozatzeko aukera daukagu. Honek guztiak gizaki adimendun gisa definitzen gaitu, eta hein berean gure antzinako arbasoetatik desberdindu. Urteetan zientzialariak gizakion burmuinak nola diharduen azaltzen saiatu dira, eta oraindik erabat garbi egon ez arren, gizakion burmuina konputazionalki modelatzeko saiakerak egiten hasiak dira, mundua ulertu ez ezik, erabaki konplexuak egiteko gai den mekanismo artifizial bat sortzeko helburuarekin [10]. Mekanismo konplexu hori konputazionalki modelatzea ez da, ordea, ataza tribiala, gizakion burmuinak era naturalean ebazten dituen arazoak konputazionalki modelatzeko arazo handiak baitaude. Artikulu honetan adimen artifiziala agente adimendunak garatzea helburu duen zientzia gisa definitzen dugu. Adimen artifiziala geroz eta ospe handiagoa bereganatzen ari den lerroa da, agente konplexuagoak eraikitzeko estrategia berriak definitzen ari baita uneoro; esate baterako: auto autonomoen arrakasta [1] edota Go! jolasteko —eta jokalari onenak menperatzeko— gai diren adimen artifizialak [14].

Oro har, adimen artifiziala termino gisa definitzea zaila da, esparru eta zientzia asko bereganatzen baititu; esate baterako: matematikak, informatika, hizkuntzalaritza, neurozientziak eta filosofia; horregatik, askotan artearen egoeran adimen artifizialaren esparru desberdin eta zehatzen inguruan aritzen gara, ez eta adimen artifizial orokor baten inguruan [6]; adibidez, esparru hauek: hizkuntzaren prozesamenduaz, dedukzio eta inferentzia mekanismoez, arrazoinamendu automatikoaz, ikasketa automatikoaz ... eta historikoki adimen artifizialeko azpiataza hauek guztiak lau lerro desberdin erabiliz sailkatuak izan dira [12].

Gizakion modura jokatzen duten sistemak

Lerro honen helburu nagusia gizakion modura jokatuko luketen eta gizakion funtzioak egikarituko lituzketen sistema adimendunak garatzea da. Adar honen baitan kokatuko genuke aski ezaguna den Turing testa[1], test honetan, makina batek iruzurra egin behar dio gizaki bati beste gizaki batekin erlazionatzen ari dela pentsaraziz [13].

Gizakion modura pentsatzen duten sistemak

Lerro honen helburua da sistema adimendunak kognitiboki gizaki gisa joka dezaten lortzea, bai erabakiak hartzeko ahalmenari dagokionez, baita

[1] https://searchenterpriseai.techtarget.com/definition/Turing-test

arazoak ebazteko ahalmenari dagokionez ere. Neurozientziarekin oso lotua dago, eta funtsean sistema adimendunak gizakien modura pentsa dezaten du helburu, zentzurik zuzenenean. Garunaren funtzionalitatea oraindik misteriotsua bada ere, badakigu neurona-sare konplexuek arrazoinamendua, kontzientzia eta jokabideak zehazten dituztela.

Pentsakera arrazionala edo zentzuzkoa duten sistemak

Lerro honetan kognitiboki arrazoitzeko, inguruaz jabetzeko, jarduteko edo inferentzia berriak egiteko helburu duten ikerketa lerroak aurkituko genituzke. Adibide batzuk ematearren, proposizio logikoak, ontologiak eta arrazonamendua egiteko gai diren sistemak aurkituko genituzke. Adar honetan pentsaera arrazionala izateko aukera emango liguketen lege unibertsalak aurkitzea da helburu nagusia. Ondoriozta daitekeen moduan adimen artifizialaren adar hau psikologiarekin eta filosofiarekin zeharo lotua dago, hala eta guztiz ere, zientzia hauetatik datozen ideia guztiak formalizatzeko logika matematikoetan oinarritzea beharrezkoa da.

Jokaera arrazionala edo zentzuzkoa duten sistemak

Adar honen helburua zentzuz jokatzen duten sistema adimendunak garatzea da. Zentzua bera zer den definitzea ez da ataza erraza, ordea. Adimen artifizialaren esparruan zentzuz jokatzea helburu funtzio jakin bat optimizatzen saiatzeko jokabideari jarraitzea dela esan daiteke. Era honetan, zentzuz jokatzen duen sistema adimendun batek bere etekinak maximizatzeko helburuarekin jokatzen duen agente gisa defini daiteke. Gizakion ikuspegitik pentsaera arrazional hau argudiagarria den heinean, etikaren esparruetatik kanpo definitzen delako, adimen artifizialaren esparruan definizio sendotzat hartzen da, eta egun, sistema asko lerro honen filosofiari jarraituz eraikitzen dira [12]. Artikulu honen 2. atalean sistema arrazional hauen inguruan sakontzen dugu, garatuko dugun software agentea adar honetan sailkatzen baita. Hau da, ikuspuntu matematiko batetik konputazionalki modelatutako arazo bat ahalik eta modu optimoenean ebaztea izango baitu helburutzat.

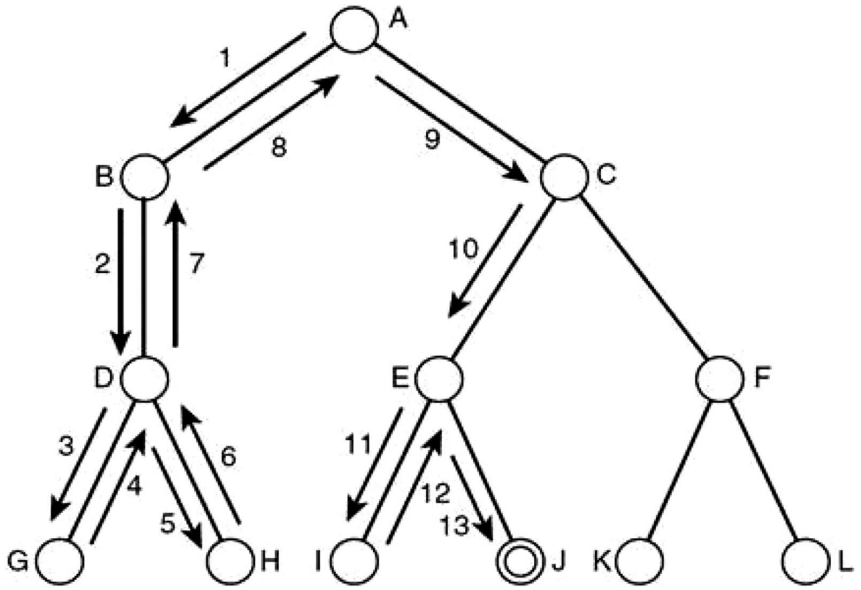
2. JOKAERA ARRAZIONALA ETA BILAKETA ESPAZIOAK

Adimen artifizialaren esparruan jokoekin lotutako arazoak ebazteko ekintza edo erabakien sekuentzia bat zehaztea da ohikoena —bilaketa espazioko ibilbide posible bat—. Bilaketa espazio hau agente batek zehaztuko du eta zatika miatuko du, hasierako egoera batetik egoera terminal batera iritsi arte. Agentearen helburua, betiere, ibilbide posible guztien artean onena identifikatzea da. Ekintza sekuentzia hau zehazteko, bilaketa metodo

edo heuristikoak erabiltzen dira (ikusi 1. eta 2. irudiak). Heuristiko hauek grafo edo zuhaitz motako egitura batean ezaugarri zehatz batzuk dituen nodoa —edo nodo sekuentzia— bilatzea dute helburu, eta bi multzo handitan sailka daitezke:

Informaziorik gabeko bilaketak edo bilaketa itsuak

Bilaketa hauetan agenteak ez du inolako informaziorik eko pauso kopuruaren inguruan, ez baita gai eman beharreko pausoen kostuak estimatzeko. Agenteak ez dauka bilaketa espazioko ibilbide desberdinak saiatzea baino aukera hoberik, hau da, bilaketa espazioa miatu ahala hautematen du.



1. irudia. Sakonera metodoen bilaketa prozesua. Bilaketa prozesu honetan ibilbide oso bat miatu behar da beste ordezko bide bat miatzen hasi aurretik.

Informazio gehigarriarekin egindako bilaketak edo heuristiko jakitunak

Kasu honetan bilaketa optimizatzeko informazio baliagarria eskuragarri du agenteak. Informazio gehigarri honen bitartez, ibilbideen estimazioak egiteko gai da, eta honekin, bilaketak azkartu interesgarriak ez diren ibilbideak baztertuz [12].

Lehenengo multzoari dagokionez, honako bi bilaketa metodo hauek dira ezagunenak: (i) sakonera metodoak eta (ii) zabalera metodoak [3].

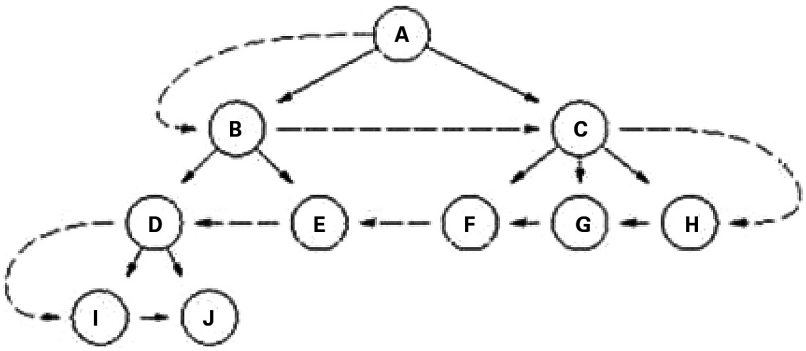
Sakonera Metodoak

Sakonera metodoen funtzionamendua zuhaitz edo grafo baten nodo bakoitza hedatzean datza. Sakonera maila bakoitzean nodo bat aukeratzen da, aztertu egiten da eta soluzioa ez bada hurrengo mailara jaisten da. Ibilbide horretan hedatzeko nodo gehiago ez dagoenean, atzera egiten du —*backtracking*— bukatu arte prozesua errekurtsiboki errepikatuz. 1. irudian sakonera metodoaren funtzionamendua ikus daiteke.

Zabalera Metodoak

Sakonera metodoak ez bezala, zabalera metodoek maila bakoitzeko aukera guztiak agortzen dituzte hurrengo mailara jauzi baino lehen. Hau da, maila bateko nodo guztiak aztertu dituztenean, bat aukeratu eta hurrengo sakonera mailara hedatzen dira; nodo horretan soluziorik ez badago, nodo anaiak prozesatzen dira. 2. irudian ikus daitekeen bezala, prozesu hau errepikatu egiten da maila bakoitzeko nodo guztiekin.

Zabalera metodoaren abantaila nagusia da beti hasierako egoeratik gertuen dagoen soluzioa aurkitzen duela, hala ere, hura aurkitzeko memoria eta denbora esponentzialki hazten dira. Ikusi dugun modura, informaziorik gabeko bilaketetan, metodoak ez ditu nodoak aztertzen helburura iristeko itxaropentsuena zein den ebazteko, oro har, nodo guztiak berdinak balira bezala tratatzen dira eta metodoak markatutako ordenan hedatzen dira.



2. irudia. Zabalera metodoen bilaketa prozesua. Bilaketa prozesu hauetan maila bateko nodo guztiak arakatu behar dira hurrengo mailara jauzi egin aurretik.

Informazio gehigarria duten bilaketa metodoetan, aldiz, informazio gehiago izaten da nodoen inguruan, eta, hortaz, bakoitzaren baliagarritasunari buruz estimazioak egin daitezke soluziora gehien hurbiltzen gaituena aukeratzeko. Metodo hauen barnean nabarmenenak *Hurbilketa oneneko*

bilaketak dira [7]. Kasu hauetan, hedatzeko hautatutako nodoak ebaluazio edo estimazio funtzio batean oinarriturik balio onena duten nodoak dira. Estimazio funtzio honek helburura iristeko kostua aurresaten saiatzen da, eta hortaz, interes altueneko nodoak hautatzen dira. *Hurbilketa oneneko* metodoetan ezagunena *A** metodoa da, zeinak hasierako egoeratik nodo jakin batera iristeko metatutako kostua, eta nodo jakin horretatik nodo terminal batera mugitzeko kostuaren estimazioa kontutan hartzen duen [11]. Benetako kostua eta estimazioak konbinatzeko gaitasuna dela eta, A* metodoak erabilera zabala eta emaitza onak bereganatu ditu adimen artifizialaren alorrean [12]. Hala eta guztiz ere, A* metodoa arrakastatsua izan dadin, behar-beharrezkoa da estimazioak egiteko erabiltzen den funtzioak hurbilketa onak ematea. Funtzio hau ondo modelatzean datza A* metodoa erabiltzearen arrakasta eta zailtasun nagusia, askotan hurbilketa on bat egingo duen funtzio bat idaztea ataza zaila delako —edo ezinezkoa—.

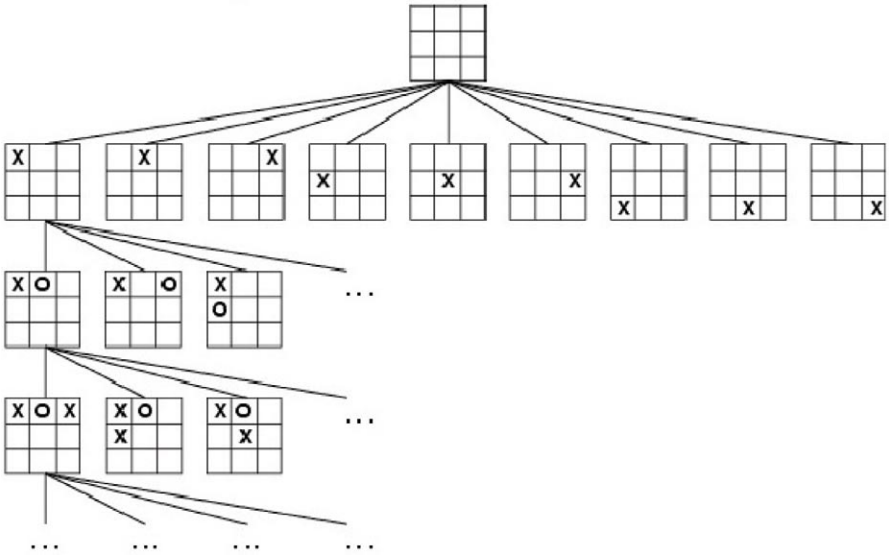
Bi edo aurkari gehiago lehiatzen diren jokoetan *Minimax* algoritmoa erabili ohi da, non agente bakoitzak bere bilaketa espazioa aztertzen eta optimizatzen duen aurkarien aukera ezberdinak aurreikusiz [5].

3. REINFORCEMENT LEARNING: BILAKETA ESPAZIOAK OPTIMIZATZEN

Bilaketa heuristikoen arazo praktiko nagusia da aztertu behar duten espazioaren dimentsioa handiegia dela; ondorioz, bilaketa hori inplementatzen duen agenteak denbora tarte luzea behar du egikarituko duen hurrengo mugimendua zehazteko. Jokoen domeinuan arazo hau oso maiz gertatzen da, har daitekeen aukeren esparrua oso zabala delako. Adibide sinple bat ematearren, *hiru lerrokatu* edo *Noughts and Crosses* jokoaren bilaketa espazioa zuhaitz egitura gisa marraztuko bagenu 9! nodo desberdin izango genituzke, 3. irudian ikus daitekeen moduan. Aipatutako joko hauek garrantzi handia izan dute adimen artifizialaren garapenean, teknika asko eta asko —teorikoki bada ere—, joko hauen gainean ebaluatu direlako. Jarraian azaltzen dugun *reinforcement learning* ikasketa metodoa ere joko hauen gainean ebaluatu zen lehen pauso gisa [4].

Praktikan joko konplexuak ebazteko bilaketetan oinarriturik dauden estrategia aurreratuagoak inplementatzen dira: adibidez, *reinforcement learning* ikasketan oinarritzen direnak [9]. Mekanismo honi esker, software agenteak ez du bilaketa oso bat egin behar bilaketa espazioan zehar, eta zuhaitzaren zati handi bat miatzea saihesten du, bere hurrengo mugimendua zehazteko denbora aurreztuz. Ikasketa prozesuari dagokionez gainbegiratutako edota gainbegiratu gabeko ikasketa automatikoarekiko alde handia dago, ikasketa prozesu honetan agenteak berak ikasi behar duelako erabaki optimoenak egiten bilaketa espazioa miatuz. Hau da, software agenteak

inolako kanpo informaziorik gabe bilaketa espazioan ibilbide desberdinak miatuz doa, eta jokoa amaitzen denean irabazi edo galdu duen kontuan izanda —errefortzua— hartutako ibilbidea kalifikatu behar du. Teorikoki frogatu daiteke jokoan nahikoa denbora trebatzen aritu den agente batek mugimendu optimoak egiten ikasiko duela [8].



3. irudia. Hiru lerrokatu jokoaren bilaketa espazioa zuhaitz gisa irudikatua. Bilaketa espazio honetan 9 maila izango genituzke eta maila bakoitzean taulan marka bat gehiago jarriko genuke taulan espazio librerik gabe geratu arte.

Edozein kasutan, agentearen helburua jokoaren modelo matematiko bat eraikitzea da. Helburu hau lortzeko, agenteak uneoro ahalik eta mugimendu onenak egiten saiatuko da, edo, beste hitz batzuetan esanda, etorkizunean errefortzu altuena lortzera eramango duen ekintza multzoa burutzera. Beraz, esan liteke joko hauek guztiak *egoeren baliagarritasuna* neurtzeko gai diren estrategien bitartez ebatz daitezkeela.

Bilaketa espazioa miatzen ikasten duten agenteak inplementatzeko hainbat estrategia daude; ezagunenak honako bi hauek dira: (i) jokoaren nodoen baliagarritasuna estimatzeko funtzioak ikasten dituztenak (*Value function* edo *Q-learning*), eta (ii) jokoaren nodoak mugimendu zehatzekin erlazionatzen dituzten funtzioak ikasten dituztenak (*Policy learning*). Artikulu honetan garatuko dugun agentea (i) motakoa da, izan ere, bilaketa espazioko zuhaitzaren nodoak baliagarritasunaren arabera sailkatzeko gai izango baita. Baliagarritasun kuantitatibo honi esker egikaritu ditzakeen aukeren artean optimoenak aukeratzeko gai izango da. Gainera, zenbat eta

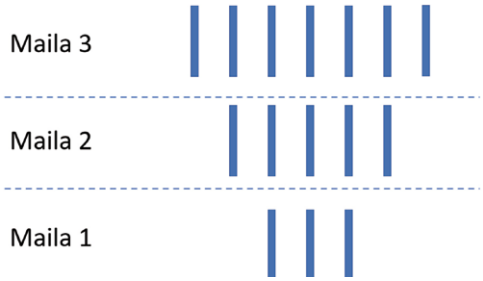
aukera hobeak hautatzeko gaitasunak, orduan eta joko gehiago irabaztera eramango du, eta, era horretan, errekurtsiboki sistema doituz joango da. Prozesu honi ikasketa aktiboa deritzo.

4. irudia. Ezkerrean 15 makilen jokoan aditua den Rogelio jauna, telebista saioko aurkezlea eskuinean.

Ikasketa aktiboaren bitartez ikasten duten agenteek behar-beharrezkoa dute bilaketa espazioa miatzeko aukera izatea. Alegia, ezagutzen ez dituzten ibilbideak miatzeko ausartak izan behar dira. Era honetan, ezagunak diren baliagarritasun altuko nodoak ustiatu ez ezik, ezezagunak diren nodoak ere miatzeko aukera izan behar dute, ikasketa garaian behintzat. Ikasketa prozesu honi jarraiki agenteak jokoaren aukera guztiak modelatzen ikasten du, ibilbide ezagun gutxi batzuk uneoro ustiatzeko. Ustiaketa eta miaketa teknika hau inplementatzeko estrategia desberdinak dauden arren [2], gure kasuan *epsilon-greedy* inplementazioa erabiliko dugu. Inplementazio honi jarraiki, gure software agenteak ausazko zenbaki baten arabera baliagarritasun altuena duen mugimendua ustiatzen saiatuko da, edota, inguruko aukera posibleen artean ibilbide ezezagun bat miatzen ausartuko da. Agentearen egitura eta kodeketa 5. atalean aipatzen dugu sakon, baita aipatutako hiper-parametroen inguruko esperimentuak egikaritu ere.

4. **15 MAKILEN JOKOA: DESKRIBAPENA ETA KODEKETA**

15 makilen jokoa aspaldiko klasiko bat da, Nim jokoan oinarritua dagoena[2], azken egunetan modan jarri dena telebistako saio ezagun batean eguneko gonbidatua jokoan trebatutako aditu batekin lehiatzen delako. Aditu honen hitzetan ez dago pertsonarik bera irabazteko gai izango denik; horregatik, adimen artifizial bat garatzea erabaki genuen, Rogelio jaunarekin lehiatzeko.



5. irudia. Makilen antolaketa jokoan. Makilak hiru mailatan banatzen dira, irudian ikusten den distribuzioari jarraituz.

Jokoan 15 makila daude, hortik izena, eta makila hauek guztiak hiru lerrotan banatzen dira, mailak osatuz. Era honetan, azpiko mailan hiru makil kokatzen dira, erdiko mailan bost eta goiko mailan zazpi. 5. irudian makilek jokoan duten antolaketa azter daiteke. Jokoa sinplea da: lehiatzen ari diren bi jokalarien artean, txandaka, erabakitzen duten maila jakin batetik nahi adina makila ken ditzakete, eta azken makila hartzen duena izango da galtzaile.

4.1. **Jokoa konputazionalki modelatzen**

Lau dira jokoa konputazionalki modelatzeko kontuan hartu ditugun ezaugarriak, *PEAS* gisa ezagutzen direnak [12], hain zuzen ere: agentearen errendimendua neurtzeko mekanismoak, jokoaren egoeran edo ingurunean parte hartzen duten ezaugarriak, agentearen akzio esparrua eta agentearen sentsoreak. Atal honetan erabiltzen dugun terminologia [12] bat dator argitalpenean erabiltzen den terminologiarekin.

— **Performance Measure (P).** Agentearen errendimendua neurtzen du, haren portaera ebaluatzen du eta errefortzua emateaz ere ardu-

[2] http://web.mit.edu/sp.268/www/nim.pdf

ratzen da. Agentea funtzio hau maximizatzen saiatuko da, nahiz eta berarentzat inplementazio zehatza ezezaguna izan. Horregatik, agenteak ibilbide desberdinak miatuko ditu ahalik eta errefortzu altuenak eskuratzeko. 15 makilen jokoa kontuan izanda, 6. irudian agertzen diren egoerak lirateke ordainsari altuenekoak, hau da, aurkariak ziur galduko duen egoeren multzoa.

6. irudia. 15 makilen jokoa irabazteko egoeren multzo minimoa. Agenteak —bere txandako makilak hartu ostean— jokoaren egoera irudian erakusten diren egoeren moduan uztea lortuz gero, jokoa irabazteko ziurtasun osoa dauka. Makilen ordena ez da esanguratsua, mailakako kopurua baizik.

— **Environment (E).** Ingurunearen ezaugarriak aztertzen ditugu hemen. Ingurune lehiakor eta behagarri batean gaudela garbi dago, lehiatzaileek posible baitute partidaren egoera uneoro aztertu. Gainera, Russel eta Norvigen sailkapena erabiliz, ingurunea determinista eta konpetitiboa da —ingurunea agenteen arteko interakzioan soilik oinarritzen delako, hau aldatuko duen ausazko kanpo eraginik ez dagoelako, ez eta agenteen akzioak huts egin dezaketelako—. Akzioei dagokienez ingurunea sekuentziala eta diskretua da, akzio kopurua finitua delako eta hauek bakarrik eragiten dutelako egoeren arteko trantsizioan.
— **Actions (A).** Agentearen akzio posibleen multzoa finitua da, eta uneko partidaren egoeraren arabera bornatua dago. Agente batek posible izango du maila jakin batetik nahi adina makil hartu, maila horretan nahikoa makil badago, noski. Agenteak ez du aukerarik txanda pasa egiteko.
— **Sensors (S).** Agenteak informazioa lortzeko dituen mekanismoak aztertzen ditugu hemen. Ingurunea guztiz behagarria izanik, agenteak partida modelatzeko erabiltzen diren datu egiturak behatzeko aukera izango du, murriztapenik gabe.

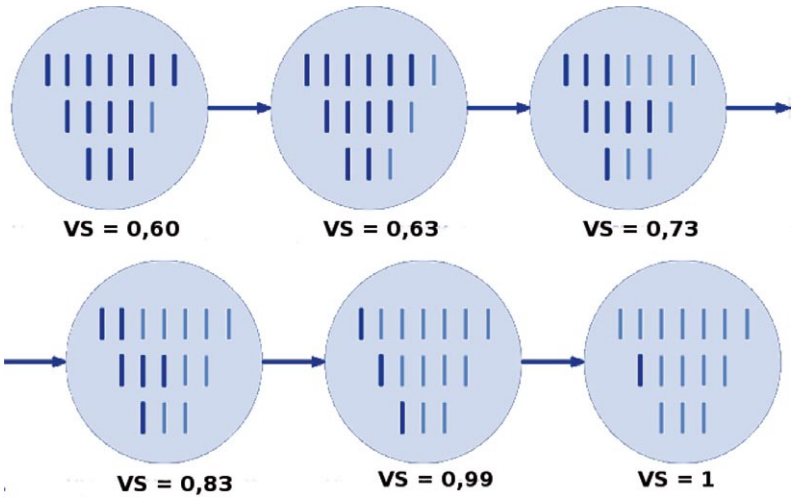
Argibide modura, partida baten definizio modura honako hau hautatu da: bi agente independenteren bi mugimendu sekuentzien segida. Beraz, gerta liteke bi agentek elkarren aurka jokatzea —agenteen entrenamendua— edo agente batek gizaki baten aurka jokatzea. Dena den, bi mugimendu sekuentzien segida bukatzean amaitzen da partida.

4.2. **Agentea jokoan trebatzen**

3. atalean aipatu dugun moduan *reinforcement learning* ikasketan oinarrituko dugu gure inplementazioa. Ustiaketan eta miaketan oinarritutako ikasketa aktibo honi esker agentea jokoa ezagutuz joango da pixkanaka, baliagarritasun altuko eta baxuko egoerak ezagutzen, alegia. Behaketa honek geroz eta optimoagoak diren erabakiak hartzera eramango du agentea, baliagarritasun funtzioa maximizatzea, hain zuzen ere.

Ikasketa prozesu honetan zehar agenteak zeharkatu duen ibilbide zehatzeko nodoen baliagarritasuna eguneratu behar du, partidaren bukaeran jaso duen ordainsari edo errefortzuaren arabera. Honela, partida asko jokatu ostean irabaztera ematen duen nodoen multzoak baliagarritasun altua izango duela espero da, eta galtzera eramaten duen nodoen multzoak, ordea, baliagarritasun txikia. Lehen jokoetan, noski, agenteak ez dauka nodoen baliagarritasunaren informazio zehatzik eta ibilbideak modu itsuan aukeratzera behartuta dago. Partida bakoitzaren ostean agenteak ordainsari edo errefortzu bat jasotzen du ingurunetik, bere gauzatzea ebaluatzen duena. Era honetan, agenteak bilaketa espaziotik egikaritu duen sekuentzia zehatza gorde behar du, errefortzu horren baitan baliagarritasuna eguneratzeko. Errefortzua ona bada, sekuentzia horren baliagarritasuna areagotu behar da; errefortzua txarra bada, ordea, txikitu.

Baliagarritasuna sekuentzian zehar hedatzeko, honako ekuazio hau erabiltzen dugu:



7. irudia. Agenteak exekutatzen duen partida irabazle baten sekuentzia. Kontuan izan aurkariaren sekuentzia galtzailea irudikatu gabe dagoela, agente irabazleak ez baitu ezagutzen. Azken mugimendua burutu ostean makila bakarra uzten dio aurkariari, bere irabazia maximizatuz.

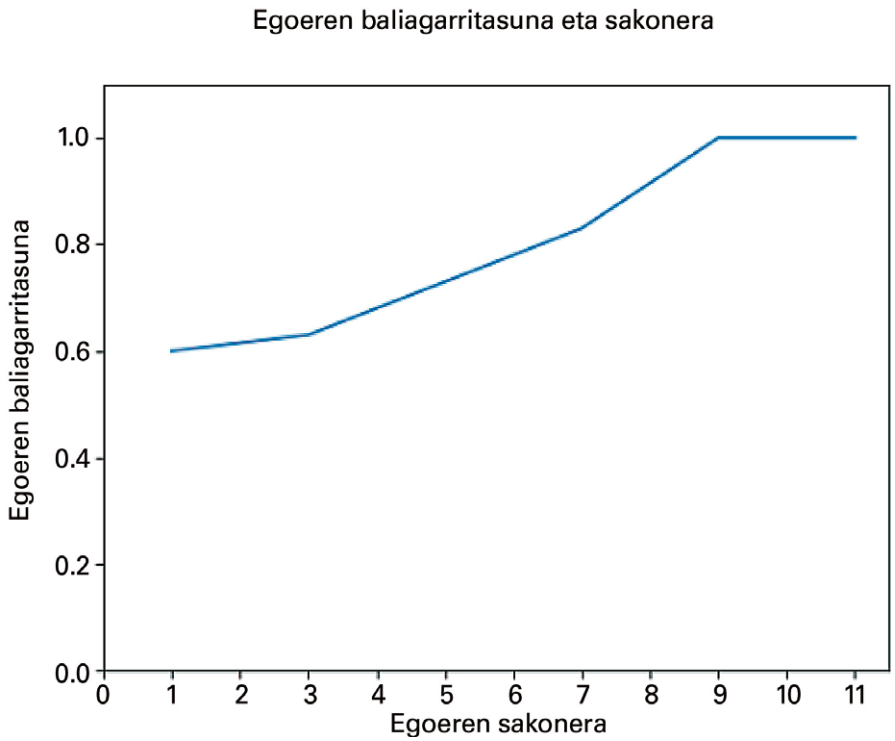
https://doi.org/10.1387/ekaia.20831

$$V(\text{state}) \leftarrow V(\text{state}) + \text{alpha} * (V(\text{statet}) - V(\text{state})) \quad (1)$$

non *V()* funtzioak baliagarritasun funtzioa islatzen duen, *state* agentea exekutatzen ari den sekuentziaren uneko nodoa den, *alpha* ikasteko abiadura hiperparametroa den eta *statet* sekuentziako hurrengo nodoa den.

Adibide gisa 7. irudian hiru mila partidatan trebatutako agente baten sekuentzia irabazle bat azter daiteke. Sekuentziako nodo bakoitzarekin batera, nodoaren baliagarritasuna ere ikus daiteke. Irudian ikusten den moduan, baliagarritasuna geroz eta altuagoa da nodo irabazlera inguratzen garen heinean.

Hortaz, baliagarritasun funtzioa ($V(s)$ funtzioa) egoera guztiei baliagarritasun kuantitatibo bat esleitzeaz arduratzen da, egoera horretara joateko interesa kuantifikatzen duena, hain zuzen ere. Jokaldien sekuentziak kontuan hartuz eguneratzen denez, denbora nahikoa trebatu ondoren baliagarritasun balioak konbergitzera jotzen du egoera bakoitza hautatzeko optimotasuna islatuz. Hau da, baliagarritasun funtzioak agentearentzat egoera jakin batera joatea zein interesgarria den neurtzen du.



8. irudia. Nodoaren baliagarritasuna eta jokoan hartutako ekintzaren sakonera.

7. irudiko nodoen baliagarritasuna ekintzaren sakonerarekiko irudikatzen badugu (8. irudia), ondorio garbi batzuk erauz daitezke; honako hauek, besteak beste: (i) jokoaren lehen ekintzak oso erabakigarriak direla, eta (ii) jokoaren azken ekintzak ez direla hain erabakigarriak. Hau da, hainbat sekuentzia irabazle aztertuz ondoriozta dezakegu partidaren nondik norakoak hasieran erabakitzen direla, eta partidaren amaierarako agente batek irabazteko probabilitate asko (≥ %99) eta besteak, ordea, gutxi dituela. Era honetan, ez dago partidari buelta emateko aukera anitz.

4.3. **Kodeketa eta ebaluazio enpirikoa**

15 makilen jokoa ebazteko garatu dugun softwarea Python programazio lengoaian idatzi dugu. 4.2. atalean azaltzen dugun ikasketa aktiboa inplementatzen du, eta era honetan, jokoan jolastu ahala trebatuz doa, egoeren baliagarritasuna pixkanaka zehaztuz. Agenteak ez du inolako arau berezirik ezagutzen, egoera bakoitzean posible dituen akzioen multzoa bakarrik ezagutzen baitu. Era honetan, erabat berdinak diren bi agente bata bestearen kontra lehiatzen dira, adituak egiten diren arte. Kodea publikoki atzi daiteke Githuben[3].

Gure inplementazioa enpirikoki ebaluatzeko, garatzaileak agentearen aurka lehiatu ginen hamar aldiz. Hiperparametro egokiak aukeratzen diren kasuetan (hiperparametroak 5. atalean aipatzen dira) ez ginen jokaldi bat bera ere irabazteko gai izan. Garbi dago partida guztiak irabaztea ez dela ausazko kontu bat, eta horregatik, jarraian aurkezten dugun atal esperimentala diseinatu genuen, sistema hobeto ulertzeko helburuarekin. Zer eta nola ikasi duen azaltzeko helburuarekin, hain zuzen ere.

5. **ATAL ESPERIMENTALA**

Sekzio honetan algoritmoaren ikasketa prozesuan erabilitako ezaugarrietan aldaketak egingo dira, haien eragina aztertzeko, eta *reinforcement learning* ikasketaren hainbat kontzeptu azaltzeko. Horretarako, hiru esperimentu egingo dira agentearen entrenamentuaren hiru ezaugarri nagusiak —hiperparametroak— ebaluatzeko: (i) trebatzeko iterazio kopurua edo jokatutako partida kopurua (5.3. Sekzioa), (ii) ikasketa abiadura edo *alpha* (5.4. Sekzioa), eta (iii) ustiaketa eta miaketa hiperparametroa edo *epsilon* (5.5. Sekzioa).

Atal esperimental honetan honako galdera hauei erantzuten saiatuko gara: (i) Zein da agentea entrenatzeko iterazio kopuru optimoa? (ii) Zein da hiperparametro desberdinak aldatzearen eragina, eta nola eragiten dio agentearen ikasketa prozesuari? (iii) Zein da hiperparametro garrantzitsuena?

---

[3] https://github.com/lgazpio/15_palos

5.1. **Hasierako egoera**

Garatutako algoritmoak *epsilon-greedy* politika erabiltzen du. Politika honek ustiaketa eta miaketa kontrolatzen du *epsilon* hiperparametroaren bidez, eta haren hasierako balioa %15ekoa da. Honek esan nahi duena da agenteak partiden %15ean ausazko egoerak hartuko dituela, eta ez jokoa irabazteko aukera handienak dituztenak. Honekin, miaketa eta ustiaketa hartzen dira kontuan. Alde batetik, miaketak ingurunearen ezaguera areagotzen du, ingurunea modelatzeko gaitasuna hobetuz, eta bestetik, ustiaketak agenteak irabazteko probabilitatea handitzen du, jadanik ezagutzen diren egoera baliagarrien ezagutza kontuan hartzen baita.

Bestalde, ikasketa abiadura edo *alpha* hiperparametroaren balioa 0,05 da, eta honek egoeren baliagarritasun kuantitatiboa aldatzeko abiadura kontrolatzen du. Azkenik, adimen artifizialak 30.000 iterazio egingo ditu. Iterazio bat partida bati dagokio, jokoa hasten denetik agenteak irabazi edo galtzen duen arte.

Balio hauek lehenetsitako baliotzat hartu dira artearen egoeran askotan sistemak horrela hasieratzen direlako. Miaketa eta ikasketa hiperparametroek ez dituzte balio altuak izan behar hasieran, sistemek ezertxo ere ez ikasteko aukera handia baitute, ikasketa automatikoan gertatzen den moduan.

5.2. **Agentea ebaluatzeko metrikak**

Agentearen errendimendua ebaluatzeko bi metrika sortu ditugu: (i) egoera ziurren kopurua eta (ii) egoeren baliagarritasunaren desbideratze estandarraren batura, jarraian deskribatzen direnak.

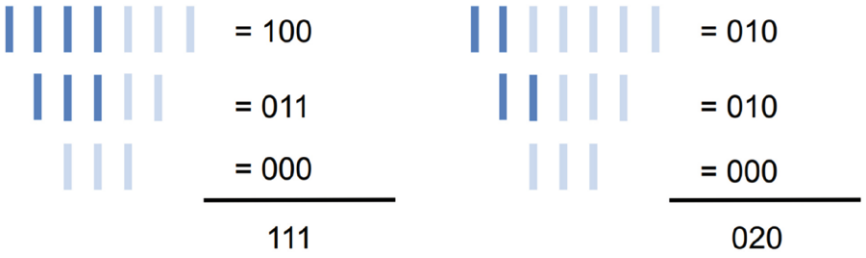
5.2.1. *Egoera ziurren kopurua*

15 makilen jokoa ebazteko bada heuristiko zehatz bat[4] 4 . Heuristiko honek dio edozein egoera irabazletzat edo galtzailetzat har daitekeela maila bakoitzeko makila kopurua sistema bitarrera bihurtzen badugu, eta zutabeen balioak batzen baditugu. Adibidez (2, 5, 7) egoera, hau da, 2 makila daudela 1. mailan, 5 makila 2.ean eta 7 makila 3.ean, sistema bitarrean (010, 101, 111) izango litzateke, eta zenbaki bitar hauen batuketa arrunta eginda batura gisa 122 zenbakia lortuko genuke, zenbaki honen arabera egoera irabazle gisa —zenbaki guztiak bikoitiak— edo egoera galtzaile gisa —zenbaki bakoiti bat gutxienez— sailkatu daiteke. 9. irudian heuristiko honen adibide bat dago.

[4] http://paraisomat.ii.uned.es/paraiso/juegos.php?id=s_753

Jokoa ebazteko filosofia honi jarraiki definitzen dugu lehenengo metrika. Metrika honek egoerak bi multzotan sailkatzen ditu: egoera galtzaileak, baldin eta egoeraren baliagarritasuna [0-0,1] tartean badago; eta egoera irabazleak, baldin eta egoeraren baliagarritasuna [0,9-1] tartean badago. Ondorioz, trebatutako agenteak jokoaren egoerak irabazletzat edo galtzailetzat diskriminatzeko gaitasuna ebaluatuko du metrika honek.

Metrika hau enpirikoki egiaztatu da sistema bitarrera bilakatzeko arauan oinarrituz eta egoera kopuru handia ikertu ondoren ez da ikusi inolako egoerarik gaizki sailkatuta [0-0,1] eta [0,9-1] tarteetan. Hau da, [0-0,1] tartean dauden egoera guztiak egoera galtzaileak dira, eta [0,9-1] tartean dauden guztiak, egoera irabazleak.



9. irudia. Egoera galtzailea ezkerrean eta egoera irabazle bat eskubian jokoa irabazteko heuristiko zehatzari dagokionez.

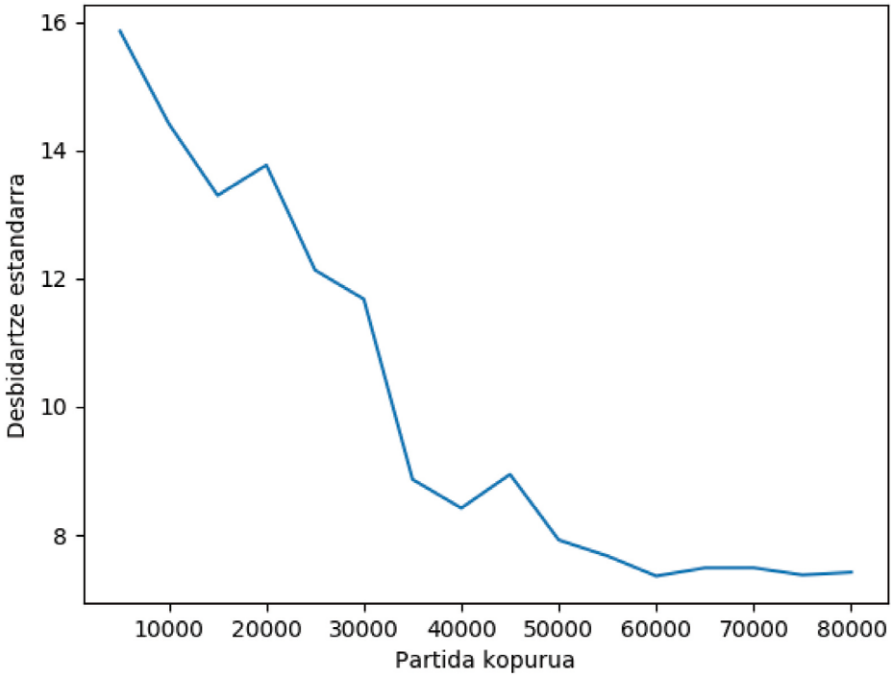
5.2.2. *Egoeren desbideratze estandarraren batura*

Bigarren metrikak ikasketa prozesu ezberdinetan izandako egoeren baliagarritasunaren aldaketa neurtzen du. Horretarako, egoera bakoitzaren desbideratze estandarra lortzen da, konparatu nahi den hiperparametroaren aldaketan oinarrituz. Adibidez, *alpha* hiperparametroaren balioa aldatzen bada, 10 agente desberdin trebatzen dira aldaketa bakoitzeko, horrela 10 balio ezberdin lortuz egoera bakoitzeko, eta hauen bidez desbideratze estandarra kalkula daiteke. Hiperparametro baten aldaketa egitean desbideratze altua lortzeak adierazten du aldaketa hori egiteak ez diola agentearen errendimenduari onik egingo; hau da, agenteak egiten duen aurreikuspena ez da hain zehatza izango.

5.3. sekzioan aipatuko den kasuan, iterazio kopuruaren aldaketekin kalkulatu da desbideratze estandarren batura.

5.3. **Agentea trebatzeko erabilitako partida kopuruaren azterketa**

Atal honetan algoritmoaren bilakaera aztertuko da, agentea trebatzeko erabili den partida kopurua aztertuz. Horretarako, agentea hamasei iterazio balio ezberdinekin trebatu da: 5.000 partidatik 80.000 partidara arte, haien artean 5.000 partidako gehikuntzak eginez. Beste hiperparametroak konstante utzi dira.

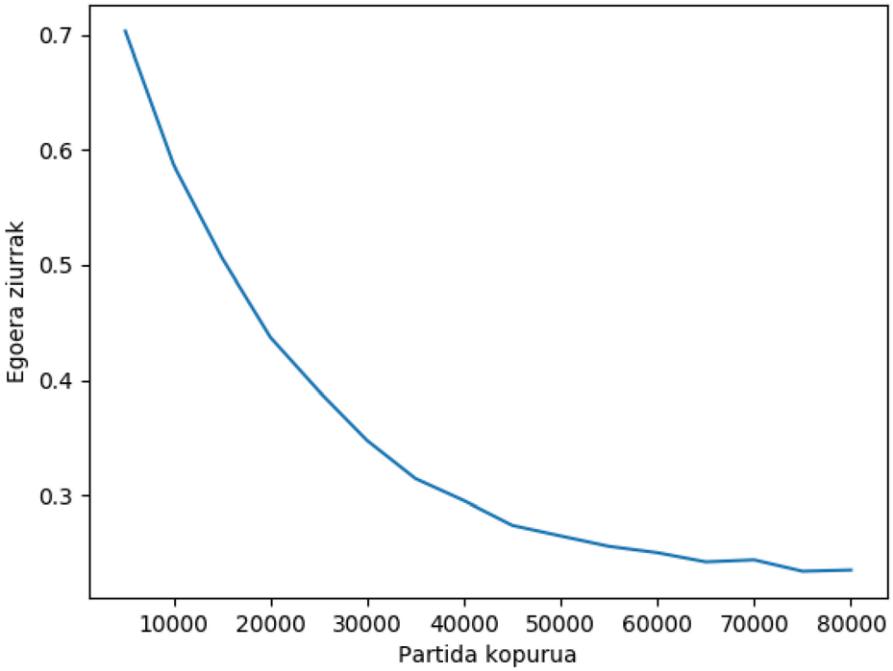


10. irudia. Irudiaren abzisa ardatzean partida kopurua ikus daiteke, eta ordenatuenean desbideratze estandarren batura.

10. irudian ikus daitekeen modura, 5.000 partida eginik soilik lortutako desbideratze estandarra handia da. Dena den, metrikaren gutxitze nabarmena ikusten da 50.000 partidatara iritsi arte. Horren ostean, ordea, jaitsiera konbergentzia puntu batera heltzen da. Joera hau bi arrazoirengatik gertatzen dela uste dugu: (i) ikasketaren hedapenaren ondorioz eta (ii) ausazkotasunaren ondorioz.

Lehenengo arrazoia ulertzeko, gogoratu beharra dago azkeneko pausuek sari edo errefortzu handiagoa ematen diotela agenteari aurrenekoek baino. Izan ere, amaierako egoerek, adibidez (0,1,0), 0 balioko desbideratze estandarra daukate, eta hasierakoek, 0,3 baino handiagoa. Atal esperimentalaren bigarren esperimentua (5.4. sekzioa) honekin lotuta dago.

Desbideratzeen batura ez gutxitzeko bigarren arrazoia epsilon hiperparametroaren %15eko balioa izan daiteke, honek eragiten baitu algoritmoak egoera gehiago miatzea. Dena den, ez dirudi erabat zentzuzkoa uneoro hiperparametroak balio bera izatea, hasieratik jokatzen baitu agenteak pertsona batek baino hobeto. Honen ondorioz, hirugarren esperimentuan aztertuko da epsilon balioaren eragina (5.5. sekzioa).



11. irudia. Irudiaren abzisa ardatzean partida kopurua ikus daiteke, eta ordenatuenean egoera ziurren portzentaia.

Bestalde, egoeren aurreikuspenen asmatzeen portzentaiaren bilakaera azter daiteke 11. irudian. %70 batekin hasteko arrazoia da egoerak 0 edo 1 balioekin hasieratzen direla. Ikus daitekeen bezala, portzentaia %23 aldera jaisten da[5], eta han finkatu; seguruenik, aurretik azaldutako arrazoi berdinengatik.

[5] Irudia intuizioaren kontra doala iruditu arren, egoera ziurren kopuruak behera egitea zeharo arrunta da iterazio kopurua igo ahala, izan ere, agenteak zenbat eta gehiago jokatu egoeren inguruko mesfidantza garatzen baitu partidak galtzeagatik. Horregatik, V (s) funtzioak balio baxuagoak itzultzen ditu.

Proba honen ondorioz ikus daiteke iterazio gehiago egiteak ez duela ikasketa prozesua zertan beti hobetu, eta beharrezkoa da beste hiperparametroak aztertzea, edota egoera aztertzen duen funtzioa aldatzea. Ondorio gisa, ordea, esan daiteke bi metriken balioak iterazio jakin batetik aurrera egonkortzen direla, eta balio hori entrenatzeko iterazio kopuru egokitzat hartu daitekeela.

5.4. **Ikasketa abiadura aldatzearen eragina**

Alda daitekeen bigarren hiperparametroa ikasketa abiadura edo *alpha* da, hau da, egoera baten balioa gehiago aldatzea agenteak irabaztean edo galtzean.

Aipatu beharra dago beti ez dela egokiena *alpharen* balioa handia izatea; izan ere, *alpha* handiegia bada, desbideratzea handia izango da, eta *alpha* txikiegia bada, asko kostatuko da egoera balio egokienera aldatzea.

1. taulan ikusi daitekeenez, desbideratzeen baturak aldaketa handia izaten du, baina egoera ziurren portzentaiak gutxi egiten du gora. Beraz, metrika bakar batean hauteman denez hobekuntza txikia, hiperparametro honek ez du ondorio garbirik erakutsi agentearen hobekuntzarekiko.

1. taula. Ikasketa abiaduraren hiperparametroaren balio desberdinekin lortzen den baliagarritasun egoeren desbideratzea eta egoera ziurren kopurua.

Ikasketa abiadura	Desbideratzea	Ziurrak
0,05	7,40	%25,10
0,10	10,16	%26,64
0,15	10,34	%29,09

5.5. **Ustiaketa eta miaketa hiperparametroaren eragina**

Miaketa hiperparametroaren balioa aldatu nahi bada, hiru estrategia har daitezke kontuan: jaistea, igotzea, edo biak konbinatzea. Hiperparametroa igotzeko kasuan, 2. taulan erakusten den modura, bi metrikek emaitza okerragoak ematen dituzte, egoera ziurren metrikari dagokionez, bereziki. Hau gertatzen da miaketa gehiegi egiten duelako agenteak, eta ustiaketa gutxiegi. Bigarren kasuan, %5era jaisten bada, desbideratzeen batura eta egoera ziurren kopurua gutxi murrizten da. Dena den, bigarren metrika honen kopuruak ez du alde handirik erakusten %15eko ustiaketa portzentaiaren errendimenduarekin konparatuz.

Hobeto jardun lezakeen saiakera aurreko bien konbinazioa egitea litzateke, lehenik miaketari lehentasuna emanez, eta ondoren, ustiaketari,

miaketa hiperparametroaren portzentaia jaitsiz. Hau da, lehenik miaketa 0,15 portzentaiarekin hastea, eta ondoren, iterazioak egin ahala balio hau murriztea. 2. taularen azken lerroan erakusten da esperimentu honen emaitza, eta ikus daitekeen bezala, miaketa portzentaia gutxika-gutxika murriztuz, egoera ziurren kopurua asko igotzen da. Esperimentu honek erakusten du miaketa eta ustiaketaren dilemaren eta haren erabilera zuzenaren garrantzia. Izan ere, miaketa parametro aldakorra erabiliz, egoera ziurren kopurua izugarri igotzea lortu da, %90 baino gehiago.

Hortaz, ondoriozta dezakegu hiperparametro honen erabilera egokiarekin, hau da, epsilon gutxika-gutxika murriztuz, agenteak jokoa ebazteko estrategia modu latentean ikasi duela, egoera irabazleak eta galtzaileak portzentaia oso altu batean baliagarritasunaren bitartez diskriminatuz.

2. taula. Ustiaketa eta miaketa hiperparametroaren eragina egoeren baliagarritasunari eta egoera ziurren kopuruari dagokionez.

Miaketa	Desbideratzea	Ziurrak
%5	7,28	%24,27
%15	7,40	%25,10
%30	8,08	%8,88
Aldakorra	4,48	%91,71

6. **AMAIERA**

Lan honen helburua adimen artifizialaren erabilera azaltzea izan da, bereziki buru-hausgarriak ebazteko, eredu modura 15 makilen jokoa hartuz. Horretarako, lehenik adimen artifiziala ulertzeko modu ezberdinak azaldu dira. Ondoren, jokaera arrazionala duten sistemen ikuspuntutik aztertu da adimen artifiziala, nodoen bilaketa metodoetan sakontzeko. Metodo horiek azaldu ostean, *reinforcement learning* teknikari buruzko argibide batzuk eman dira, bilaketa prozesuen hobekuntzatzat har daitekeena. Ondoren, 15 makilen jokoa deskribatu da, baita haren modelo konputazionala eta hura jokatzeko kodeketa ere, *reinforcement learning* ikasketa teknikan oinarritu dena.

Artikulu honek erakusten du 15 makilen jokoa *reinforcement learning* teknikaren bitartez ebatz daitekeela, eta berau egiteko kodea publikoki zabaltzen du lehen aldiz. Gainera, uste dugu kode hau erraz molda daitekeela antzekoa den beste edozein joko ere ebazteko.

Artikuluaren mamian esperimentu multzo bat egin da jokoaren inplementazioa hobetzeko. Esperimentu hauetatik bik (partida kopuruaren eta

miaketa portzentaiaren aldaketak) emaitza onak eman dituzte, nahiz eta ikasketa abiaduraren aldaketak ondorio garbirik ez eman. Atal esperimentalak erakutsi duen moduan, kontuan hartu beharra dago miaketa eta ustiaketaren dilemaren garrantzia, eta honekin loturiko miaketa eta ustiaketa hiperparametroa.

Ondorio nagusi gisa esan dezakegu lortutako emaitzek erakutsi dutela agentea hiperparametro egokiekin trebatzean egoera irabazleak eta galtzaileak zehaztasun handiz diskriminatzen ikasi duela —sistema bitarrera bilakatzeko arauaren antzera—. Hau da, algoritmoak berak ikasketa aktiboaren bitartez ezagutza hau latente modelatzeko gai izan da egoeren baliagarritasunean oinarrituta.

Proiektu honetan lan egitean eta argitalpena gainbegiratu duten boluntarioei esker esparru gehiago ikertzeko lerroak ere identifikatu dira; besteak beste, hauek: (i) jokoarentzako interfaze grafiko bat diseinatzea; (ii) erabilitako metrikaren baliagarritasuna *gold standard* batekin ebaluatzea, egoera irabazleen eta galtzaileen zehaztasuna izateko; (iii) ideia berrien miaketa egitea ikasketa abiadura hiperparametroan aldaketak egiteko, algoritmoaren errendimendua hobetuz; (iv) egoerak aztertzen dituen funtzioa aldatzea; sare neuronalak har litezke kontuan egoeren baliagarritasuna aztertzeko; eta (v) heuristikoen inplementazioa bilaketa espazioak murrizteko, hala jokoaren konplexutasuna ere murrizteko.

7. **BIBLIOGRAFIA**

[1] James M Anderson, Kalra Nidhi, Karlyn D Stanley, Paul Sorensen, Constantine Samaras, and Oluwatobi A Oluwatola. *Autonomous vehicle technology: A guide for policymakers*. Rand Corporation, 2014.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235-256, 2002.

[3] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[4] Michie D. *Trial and Error*, volume Part 2. Harmondsworth: Penguin, 1961.

[5] Ding-Zhu Du and Panos M. Pardalos. *Minimax and Applications*. Kluwer Academic Publishers, 1995.

[6] Ben Goertzel and Cassio Pennachin. *Artificial general intelligence*, volume 2. Springer, 2007.

[7] Pearl J. *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley, 1984.

[8] Tommi Jaakkola, Michael I Jordan, and Satinder P Singh. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pages 703-710, 1994.

[9] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237-285, 1996.

[10] D Kriesel. A brief introduction to neural networks. dkriesel.com. 2011.

[11] Dechter R. and Pearl J. Generalized best-first search strategies and the optimality of a*. *Journal of the Association for Computing Machinery*, 32:505-536, 1985.

[12] Stuart J Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited, third edition, 2016.

[13] Ayse Pinar Saygin, Ilyas Cicekli, and Varol Akman. Turing test: 50 years later. *Minds and machines*, 10(4):463-518, 2000.

[14] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot *et al*. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.