Szegedi Tudományegyetem

Informatikai Intézet

SZAKDOLGOZAT

Hajdu Katalin

2020

Szegedi Tudományegyetem
Informatikai Intézet

Kombinatorikus játékok számítógépes elemzése a Hanabi társasjátékon keresztül

Computer analysis of combinatory games via the boardgame Hanabi

Szakdolgozat

Készítette:
Hajdu Katalin
programtervező informatikus szakos hallgató

Témavezető:
Dr. Blázsik Zoltán
egyetemi adjunktus

Szeged
2020

Feladatkiírás

A kombinatorikus játékok számítógépes elemzése sok tételt megcáfolhat. Másrészt már kis méretű játékok esetén is rendkívül sok számításra lenne ahhoz szükség, hogy a gép legjobban játsszon. Mit várhatunk így? Ha működik egy játékra egy algoritmus annak működése, eredményei súgnak a játékot elemző embernek. A cél az, hogy a nehéz játékokat jobban játsszuk majd a gépi elemzés következtében.

Tartalmi összefoglaló

- ***A téma megnevezése:***

A kombinatorikus játékok számítógépes elemzése.

- ***A megadott feladat megfogalmazása:***

A Hanabi társasjáték megvalósítása gépi logika alkalmazásával. A mesterséges intelligencia segítségével az emberi játékos megtanulhat jobban játszani.

- ***A megoldási mód:***

Standalone Java alkalmazás.

- ***Alkalmazott eszközök, módszerek:***

MVC modellt alkalmaztam a program szerkezetének megvalósítására.
Felhasználói felülethez (User Interface) Java Swing-et használtam.
Teszteredmények .csv fájlba készültek, amelyekből Excel kimutatásokkal diagrammokat készítettem.

- ***Elért eredmények:***

A mesterséges intelligencia saját magával játszva a maximális 25 pontból átlagosan 18,218 pontot ért el négy játékos esetén.

- ***Kulcsszavak:***

Java, Hanabi, társasjáték, kombinatorikus játék, mesterséges intelligencia

Tartalomjegyzék

Bevezetés

Az alábbi dolgozatban a Hanabi című társasjátékot valósítom meg virtuális környezetben. A Hanabi egy kooperatív kártyajáték, japánul tűzijátékot jelent. Antoine Bauza francia játéktervező adta ki 2010-ben, 2013-ban pedig elnyerte a tekintélyes *„Az év játéka”* díjat. A felhasználó mesterséges intelligenciával vezérelt játékosokkal próbálhatja ki a játékot, választható játékosszámmal és nehézségi szinttel. Emellett lehetőség van csak számítógépes játékosokkal lejátszani a játékot, és az eredményt elemezni.
A dolgozat célja különböző stratégiák elemzése, és a legoptimálisabb megtalálása.
A mesterséges intelligenciát saját tapasztalataim alapján kezdtem el fejleszteni, majd szimulációkat futtatva tovább finomítottam a módszereken. Sokat segített a Mathematics Magazine 2015 februári száma. (Vol. 88, No. 1 – How to Make the Perfet Fireworks Display: Two Strategies for Hanabi) Ez a cikk két stratégiát tartalmaz a tökéletes tűzijáték megvalósításához. Ebből nyertem inspirációt a saját algoritmusom megvalósításához.
Olvastam olyan megvalósításról (Marc G. Bellemare, McGill University, Canada), ahol a mesterséges intelligencia magától tanult és fejlődött. Ennek lényege, hogy durván 300 millió alkalommal játszattak le meccseket a tanuló algoritmussal. Az eredmény, bár két játékosra egész magas pontszámot produkált, (átlagosan 22,73 a maximális 25-ből,) öt játékosszám esetén az amatőr játékosok átlagos pontszámánál alacsonyabban teljesített, körülbelül 15 pontot szerzett. Mivel nincs elég tapasztalatom a saját magát tanító algoritmusokkal kapcsolatban, illetve a kézzel kódolt, lefektetett szabályok alapján cselekvő számítógépes játékosok sokkal magasabb pontszám elérésére képesek, ezért ez utóbbi megvalósítást választottam a dolgozatomban.

1. A Hanabi játék szabályai

1.1 Játék célja

A Hanabi egy kooperatív játék, ami azt jelenti, hogy a játékosok nem versengenek, hanem együtt, egy csapatként játszanak. A saját lapját senki sem láthatja, így kénytelenek a többi játékos tanácsaira hagyatkozni. A játékosoknak a tűzijáték kártyákat színek szerint csoportosítva, növekvő sorrendben kell kijátszaniuk. Egy tűzijátékban minden számból csak egy szerepelhet. Minél több lapot sikerül a játékosoknak helyesen kijátszani, annál több pontot kapnak a játék végén.

1.2 Játékelemek

A játék 50 tűzijáték kártyát tartalmaz.
Az alapjáték 5 színt tartalmaz: piros, kék, zöld, sárga és fehér. Minden színből 10 lap van.
Egy színből 3 darab egyes, 2-2 darab kettes, hármas és négyes, 1 darab ötös van.
A játékosoknak összesen 3 életük és 8 utalásjelzőjük van.

1.3 Játékmenet

Véletlenszerűen választani kell kezdőjátékost, majd az óramutató járásának megfelelően sorra következnek a játékosok. Egy játékos körében 3 féle akció közül választhat:

1. *Lap kijátszása*

   A játékos kiválaszt egyet a saját lapjai közül, és kijátssza azt az 5 tűzijáték megfelelő oszlopába, majd új kártyát húz a pakliból. Ha a lap nem játszható ki szabályosan, egy életet elvesztenek a játékosok.

   Az 5. tűzijáték bármely színből való kijátszásakor egy utalásjelző tokent visszakapnak a játékosok.
2. *Lap eldobása*

   A játékos kiválaszt egyet a saját lapjai közül, és a dobópakliba helyezi, majd új kártyát húz a pakliból. Ezzel egy utalásjelző tokent visszakapnak a játékosok. A dobópakli nyílt, mindenki által látható és bármikor átnézhető.
3. *Információ adás*

   Egy másik játékos kiválasztása, akinek a következő információt lehet adni:

   a. Egy adott szín kiválasztása és a játékos lapjai közül megmutatni neki az összes olyan színű lapot.

b. Egy adott szám kiválasztása és a játékos lapjai közül megmutatni neki az összes olyan számú lapot.

Ez az akció egy utalásjelző tokenbe kerül.

Nem lehet információt adni, ha nincs több utalásjelzője a játékosoknak.

1.4 Játék vége

A játék többféleképpen is befejeződhet:

1. Elfogy mind a 3 élet. A játékosok vesztettek, 0 pontot kapnak.
2. A játékosok mind az 5 színből befejezik a tűzijátékot, nyertek, 25 pontot kapnak (ez a maximum).
3. Egy játékos felhúzza az utolsó lapot a pakliból. Ezután még mindenki egyszer választhat akciót, utoljára az utolsó lapot felhúzó játékos. Ebben a befejező körben már nem tudunk lapot felhúzni, mivel elfogyott a húzópakli. Ezután következik a kiértékelés, a játékosok nyertek.

1.5 Értékelés

Minden színű tűzijáték oszlop annyi pontot ér, amennyi a legnagyobb értékű lap az adott színben. Ezen pontok összessége adja a játék végi pontszámot, maximum 25 pontot.

2. A játék logikája

A játék célja tehát, hogy minden színből kirakjunk egy növekvő számsorozatot. Mivel senki nem látja a saját lapjait, a többi játékos által adott utalások segítségével tudhatjuk meg, mi van a kezünkben. Ebből következően a legfontosabb döntéseket ezen utalások adásakor kell meghozni. A játékosok a kapott információ alapján fogják tudni eldönteni, hogy a kezünkben lévő lapokat inkább kijátsszák vagy eldobják.

A lapok nagy része nem egyedi, csak az ötös lapokból van színenként egy darab. Minden kettes, hármas és négyes lapból színenként két darab van, az egyesekből pedig minden színből három darab van. Ez a nagyobb számosság esélyt ad arra, hogy hamarabb el lehessen kezdeni a tűzijáték oszlopokat.

A játék elején még egy tűzijáték lapot sem raktunk le, így ilyenkor a legfontosabb, hogy az egyes számú lapokat mutassuk meg, hogy játékostársaink elkezdhessék kijátszani őket. Ha van olyan játékos, akinek a kezében több különböző színű egyes lap van, érdemes először neki megmutatni ezeket, mert egy utalás felhasználásával egyszerre több lapról is információt kapnak. Ez általánosan is igaz: ha egy utalással több kijátszható lapot is megmutatunk, akkor az jobb kihasználása az utalásnak, mintha kevesebb lapról adnánk információt.

Figyelni kell arra, hogy ha egy bizonyos lapot már megmutattunk, arról később ne adjunk ismételten információt. A játékos, aki ilyen duplikátumról kapott információt, úgy kezelheti azt, mintha ez a következő tűzijáték lenne a sorban, annak ellenére, hogy korábban egy másik játékos esetleg már lerakta azt. Miután a saját lapjáról tévesen azt az információt kapta, hogy kijátszható, ez egy élet elvesztését vonhatja maga után. Sokat nehezít a helyzeten, ha egy játékos kezében kettő példány van ugyanabból a lapból, hiszen nem lehet kiválasztani az egyiket és csak arról adni utalást. Amíg a duplikátum nem kerül ki a kezéből, csak két utalás felhasználásával lehet egyértelmű információt adni a lapról, ami nem optimális megoldás, és sokszor nincs is rá lehetőség.

Általánosan magasabb pontszámhoz vezet, ha szimultán haladunk mindegyik tűzijátékkal, és nem mindenképpen egy kiválasztott színt akarunk mihamarabb befejezni. Ez azért bölcsebb, mert amikor a kijátszható lapokról adunk információt, az utalásjelző tokenjeink előbb-utóbb elfogynak. Hogy ezeket pótolhassuk, el kell dobnunk lapokat. Annak érdekében, hogy ne olyan lapot dobjunk el, ami később még kelleni fog, esetleg már csak egy példány van belőle a pakliban, azt is meg kell mutatnunk, melyik lapot dobhatjuk el.

Vannak olyan lapok, amelyekre már nem lesz szükség később a játék folyamán, mert egy korábban kijátszott lap duplikátumai. Tegyük fel, a piros tűzijáték oszlopában már az egyes számú lap ki van játszva, akkor a másik kettő piros egyes ilyen felesleges lap, ezekről lehet információt adni. Ez nehezebb feladat, ha még nincs kijátszva az összes egyes, hiszen a játékos hiheti azt, hogy egy kijátszható lapot mutattunk meg. Ezért biztonságosabb addig várni, amíg az összes adott számú lap ki van játszva, mert utána már nincs kétség, hogy kijátszható vagy eldobható-e a lap. Az előbbi példát folytatva, ha már minden színből ki van játszva az egyes, onnantól kezdve biztonsággal meg lehet mutatni az összes egyes lapot, mint eldobható.

Az ötös lapokból csak egy darab van. Éppen ezért ezekre jobban kell vigyázni, nehogy véletlen eldobjuk, mert akkor már nem érhetjük el a maximális 25 pontot. Ezeket a lapokat annak ellenére meg lehet (és néha meg kell) mutatni, hogy épp lejátszható-e, vagy nem, bár ezzel az információval akkor még nem tud mit kezdeni a játékos. Viszont azzal, hogy utalást adtunk valakinek, hogy mely lapjai ötös számúak, indirekten azt az információt is adtuk, hogy a többi lapja nem ötös, tehát azokat nagyobb biztonsággal eldobhatja.

Sokszor az indirekt információ többet ér, mint a tényleges utalás. Ha az emberi játékos is annyira pontosan fejben tudná tartani a kapott információkat, mint a gép, sokkal többet tudna kikövetkeztetni a lapjairól.

3. Program komponensek

A program megvalósításához MVC modellt használtam. Ez három komponensből épül fel: Model, View és Controller.
A Model osztályok reprezentálják a játék állapotát. Itt tárolom a játék logikájához és megjelenítéséhez szükséges adatokat.
A View osztályok a játék képernyőn való megjelenítéséért felelősek.
A Controller felügyeli a játék logikájának változásait, és vezérli játék menetét.

A program szerkezeti felépítése a következő:
A legfontosabb Model osztályok, és lényegesebb változóik:

- Card: egy kártyát reprezentál
    - cardColor: kártya színe
    - cardNumber: kártya száma
    - knownColor: azt tárolja, hogy megmutatták-e a kártya színét
    - knownNumber: azt tárolja, hogy megmutatták-e a kártya számát
- HanabiCards: ez a kártyapakli
- Hand: egy játékos kézben tartott kártyáit tárolja
- Player: egy játékos nyilvántartására szolgál
    - name: a játékos neve
    - hand: a kézben tartott kártyák
    - humanPlayer: megadja, hogy emberi vagy gépi játékos-e
- Players: az összes játékost nyilvántartja
- Fireworks: a kijátszott tűzijátékokat tárolja
- DiscardedCards: az eldobott lapokat tárolja
- Tokens: a játékosok életét és utalásjelző tokenjeit tárolja
- History: a játék során adott információkat tárolja

A legfontosabb View osztályok:

- SetupWindow: a kezdeti játékbeállítások ablaka
- GameTable: a játéktábla megjelenítéséért felelős, a kezdeti beállítások alapján inicializálja a képernyőt. Három részre van osztva az ablak:
    - PlayerPanel: a játékos, és kártyáik megjelenítése
    - FireworksPanel és DiscardedCardsPanel: a kijátszott és eldobott kártyák megjelenítése

    - CluePanel és ControlPanel: a játék technikai elemeinek megjelenítése (életek és utalásjelző tokenek száma, aktuális akció, a korábban adott információk listája)

A Controller osztályok:

- PlayHanabi: az emberi játékossal való játék indítása, felügyelete
- PlayTest: csak gépi játékosokkal való játék indítása, teszteredmények gyűjtése
- AIPlayer: a mesterséges intelligencia algoritmusa

A játékot a HanabiMain osztályból indítom.

4. A mesterséges intelligencia logikája

A mesterséges intelligencia tervezéséhez a Mathematics Magazine által ismertetett stratégiát vettem alapul. Ebben a cikkben a „sapkaszín kitalálós játékot” (*hat guessing game*) vették inspirációnak a tökéletes pontszám eléréséhez. Ennek a játéknak matematikai felhasználása is van.

A szabály a következő: minden játékos fején van egy sapka. Összesen kétféle különböző színű sapka van. Kommunikáció nélkül adott körsorrendben mindenki megtippeli, hogy a saját fején milyen sapka van a többiek fején látott sapkák színe alapján. A feladat, hogy maximalizálják az eltalált sapkaszíneket. Ha megvizsgáljuk, ez a játék nagyban hasonlít a Hanabira. Mindkét játékban a saját lapjainkat, illetve „sapkáinkat” kell kitalálni, és a végpontszámot maximalizálni.

A cikk kétféle stratégiát mutat be a sapkaszín kitalálós játék alapján: az első, amikor az utalás adást arra használják a játékosok, hogy konkrét akciókat javasoljanak játékostársaiknak, ez az „*akció ajánló stratégia*”. A második, amikor az utalás adással információt adnak át a kézben tartott lapokról, hogy a játékos az alapján döntse el a saját akcióját, hogy mit tart a kezében, ez az „*információs stratégia*”. Mindkét stratégia a játékok nagy arányában tökéletes pontszámot eredményezett.

A két stratégiából különböző részeket emeltem át a saját algoritmusom megvalósításához. Az akció ajánló stratégiánál kifejtették, hogy milyen prioritási sorrend szerint lehet utalást adni egy játékos kézben tartott kártyáiról.

Mielőtt ezt a prioritási sorrendet ismertetem, néhány fogalmat szükséges definiálnom. Ezeket később is használom a dolgozatomban.

Kijátszható lap: olyan kártya, amelyet az adott játékálláskor élet elvesztése nélkül, szabályosan ki lehet játszani.

Felesleges lap: olyan kártya, amelyből egy példány már szabályosan ki van játszva.

Duplikátum: olyan kártya, amely nem egyedi, több példány is van belőle a pakliban.

Az utalás adás prioritási sorrendje a következő:

1. Információt adni kijátszható ötös lapról, ha több is van (erre a továbbiakban döntetlenként hivatkozom), a legrégebben kézben tartott ötösről.
2. Információt adni kijátszható (nem ötös számú) lapról, döntetlen esetén az alacsonyabb számú kártyáról. Ha több azonos számú lap van, a legrégebben kézben tartott kártyáról.

3. Információt adni eldobható felesleges lapról, döntetlen esetén a legrégebben kézben tartott kártyáról.
4. Információt adni eldobható duplikátumról, döntetlen esetén a legrégebben kézben tartott kártyáról.
5. Információt adni a legrégebben kézben tartott lapról, hogy eldobható-e.

Ezt a prioritási sorrendet vette alapul a saját programom utalás adás választó része. Az én megvalósításom nem használja az 5. pontot, és míg az akció ajánló stratégia célja, hogy egy kártyáról adjon információt egy utalás adással, számomra ez nem volt fontos. Inkább előnyös, ha egy utalással egyszerre több kézben tartott lapról kap információt a játékos.
Az tehát adott, hogy mi alapján választja ki az algoritmus, hogy milyen utalást ad. De mikor ad információt, játszik ki, vagy dob el lapot? Ennek eldöntésére az információs stratégiánál ismertetett akció választó algoritmust vettem alapul.
A cikkben a következő sorrendet mutatták be:

1. Kijátszani egy kézben tartott kijátszható kártyát, döntetlen esetén a legrégebben kézben tartott lapot.
2. Ha kevesebb, mint öt kártya volt eddig a dobott lapok között, akkor eldobni egy felesleges kártyát, döntetlen esetén a legrégebben kézben tartott lapot.
3. Ha van utalásjelző token, információt adni egy játékostársnak.
4. Eldobni egy kézben tartott felesleges kártyát, döntetlen esetén a legrégebben kézben tartott lapot.
5. Ha a játékos kézben tartott lapjai között ugyanaz a kártya megtalálható, mint egy másik játékos lapjai között, vagyis duplikátum, akkor eldobni ezt a kártyát.
6. Eldobni egy duplikátum kártyát (egy olyan lapot, amelyből van még másik példány, és eldobása után is lehetséges még elérni a tökéletes pontszámot), döntetlen esetén a legrégebben kézben tartott lapot.
7. Eldobni a legrégebben kézben tartott lapot.

Ezt az akció választó algoritmust használja módosítva a programom. A második és ötödik pontot teljes egészében kihagytam, a többit kis mértékben módosítottam.
A következő fejezetekben azt a folyamatot mutatom be, hogyan építettem fel lépésenként ezt a prioritási sorrendet.

5. A mesterséges intelligencia logikájának javítása lépésenként

5.1 Tesztek

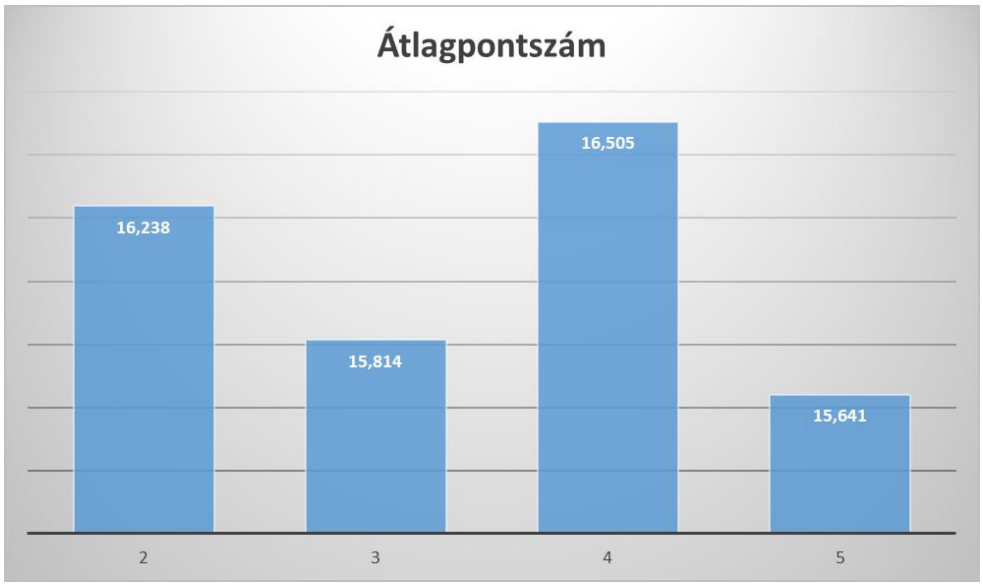
A tesztek elkészítéséhez eltávolítottam az emberi játékost, és csak gépi játékosokkal futtattam szimulációkat. Így lehetőségem volt rövid idő alatt nagy mennyiségű játékot lejátszani, amiből statisztikát állíthatok elő.
Az eredményt a következőképp kell értelmezni:
Minden játék végén valamekkora pontszámot értek el a gépi játékosok. Ha ez a pontszám pozitív, a játék során ennyi kártyát sikerült kijátszaniuk, mielőtt elfogyott a teljes pakli. Ha 0 pontot értek el, akkor elvesztették mind a három életüket.
1000 teszt volt futtatva minden játékosszámra, azaz 1000 2 fős, 3 fős, 4 fős és 5 fős szimulációt futtattam csak gépi játékosokkal.
Minden tesztfuttatáshoz két ábrát készítettem. Az első az átlagpontszámokat mutatja, játékosszámra lebontva, a második a 4000 tesztfuttatás összesített pontszámeloszlása. A vízszintes tengelyen azokat a pontokat ábrázolom, amelyeket a tesztfuttatás elért, a függőleges tengely pedig az adott pontszám előfordulásának számát mutatja.

5.2 Első tesztfuttatás:

A program akció választó algoritmusa a következő:
Elsődleges prioritása a kártya kijátszás. Csak akkor játszhat ki kártyát, ha korábban már megmutatták valamely lap számát vagy színét. Ha ez nem teljesül, a következő akció feltételét vizsgálja.
Második prioritás a segítség adás. Akkor adhat segítséget a játékos, ha van még utalásjelző token, van olyan játékos, aki a kezében tart kijátszható kártyát, és ennek a játékosnak egyértelműen meg lehet mutatni ezt a lapot. Ez azt jelenti, hogy az utalás nem tartalmazhat olyan kártyát, amelyik nem kijátszható. Tehát ha a kiválasztott szín vagy szám nem csak a kijátszható kártyáról ad információt, hanem olyanról is, amit kijátszva az életek száma csökkenne, nem adhatunk segítséget. Ha ezen feltételek valamelyike nem teljesül, az utolsó akciót választja a játékos.
Az utolsó akció a legrégebben kézben tartott kártya eldobása. Ha a fenti két akció közül egyiknek sem teljesült a feltétele, mindenképpen ez lesz végrehajtva.

A pontszámok átlagai játékosszámra lebontva: (1. ábra)

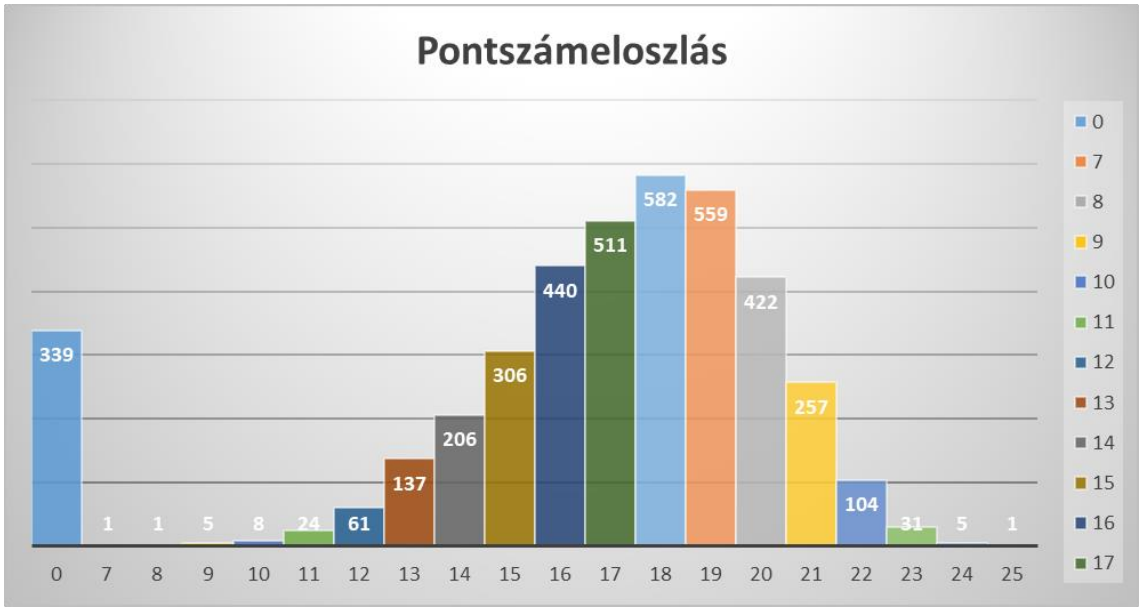
A legmagasabb átlagot 2 és 4 játékosra futtatva érte el a szimuláció. Összességében nem túl nagy a különbség az áltagpontszámok között.



1. ábra

Az összes játék egyesített pontszámeloszlása: (2. ábra)

A diagrammon kirajzolódik egy haranggörbe, melynek legmagasabb pontja 18 pontnál van.



2. ábra

Az eredmények alapján látszik, hogy egész sok esetben vereséggel végződött a játék, tehát a játékosok elvesztették mindhárom életüket. Ez váratlan, mert az algoritmus alapján sosem kapnak olyan utalást, ami nem lejátszható lap. A probléma onnan fakad, hogy ha egy lapból több példány is ugyanannak a játékosnak a kezében van, az algoritmus mindegyiket úgy értelmezi, mintha lejátszható lenne. Illetve amikor utalást adnak a

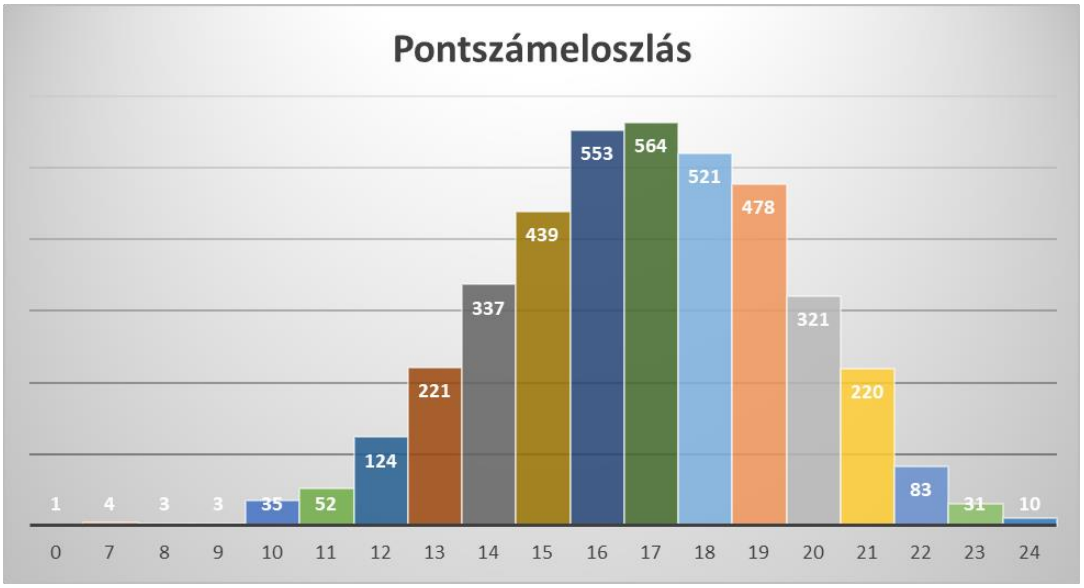
játékosok egymásnak, nem figyelték, hogy korábban már meg volt-e mutatva egy adott kártya, és többször is ugyanarra a kártyára adtak utalást, különböző játékosok kezében. A megoldás tehát, hogy az algoritmus figyelje a duplikátumokat is.

5.3 Javított első tesztfuttatás:

A pontszámok átlagai játékosszámra lebontva: (3. ábra)
Érdekes módon itt már a 2 fős játékok átlagpontszáma is lecsökkent a 3 és 5 fős játékok szintjére, csak a 4 fős szimulációk átlagpontszáma kiemelkedően magas, bár ez csak ~0,5 pontnyi eltérést jelent.

3. ábra

Az összes játék egyesített pontszámeloszlása: (4. ábra)



4. ábra

Jól látható, hogy most már csupán egy szimuláció veszített játékot.

5.4 Második tesztfuttatás

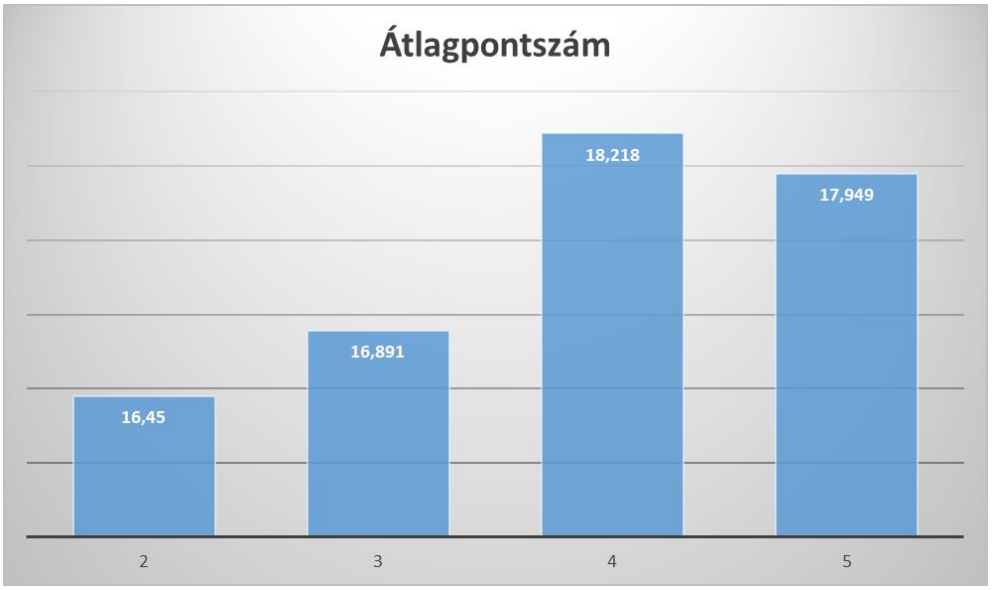
A program akció választó algoritmusa a következőképp módosult:
A legelső prioritás továbbra is a kártyakijátszás. A gépi játékos megvizsgálja, hogy milyen utalást kapott a lapjairól.

- Ha egy lapról a színt és a számot is tudja, megnézi, kijátszható-e, és aszerint cselekszik.
- Ha csak a szám ismert, ellenőrzi, hogy van-e olyan színű tűzijáték, ahol épp a következő lerakható lap lenne az adott szám, és ha igen, kijátssza. Ezért is fontos, hogy ne mutassunk meg idő előtt olyan számot, ami eldobható, mert kijátszhatónak értelmezi a gépi logika.
- Ha csak a színét tudja egy lapnak, és nincs másik lap a kezében ezzel a színnel, akkor kijátssza. Emögött az a logika áll, hogy ha több lap azonos számú, de nem mind kijátszható, akkor úgy tudunk egyértelműen információt adni, hogy a kijátszható lap színét mutatjuk meg. Ez egy olyan konvenció, amely kezdő játékosoknak talán meglepő, de hosszútávon több pontot eredményez.

A második prioritás az utalás adás, bővült az eldobható felesleges lapok mutatásával. Ezt akkor választja az algoritmus, ha nem tudott kijátszani lapot. Kétféle információt adhat: ha csak lehetséges, lerakható lapot mutat, de ha nincs a játékosok kezében ilyen, vagy nem lehet egyértelműen megmutatni, akkor eldobható felesleges lapot mutat meg.
Felesleges lapnak számít az olyan lap, amelyikből már kijátszottak egy példányt a tűzijáték oszlopba, így már nincs rá szükség. Akkor ad utalást felesleges lapról, ha már biztonságos információt adni róla. Megmutatja a számát, ha az összes tűzijáték oszlopban már ki van játszva az adott szám, vagy a színét, ha az adott színből már nem lehet többet kijátszani.
A harmadik prioritás a felesleges kártya eldobása. Ha a gépi játékos nem adott utalást, akkor eldob egy, a kezében tartott felesleges kártyát, feltéve, hogy kapott erről korábban információt.
Az utolsóként választott akció a legrégebben kézben tartott kártya eldobása. Ha a korábbi három akció közül egyiket sem választotta, ezt fogja tenni.

A pontszámok átlagai játékosszámra lebontva: (5.ábra)

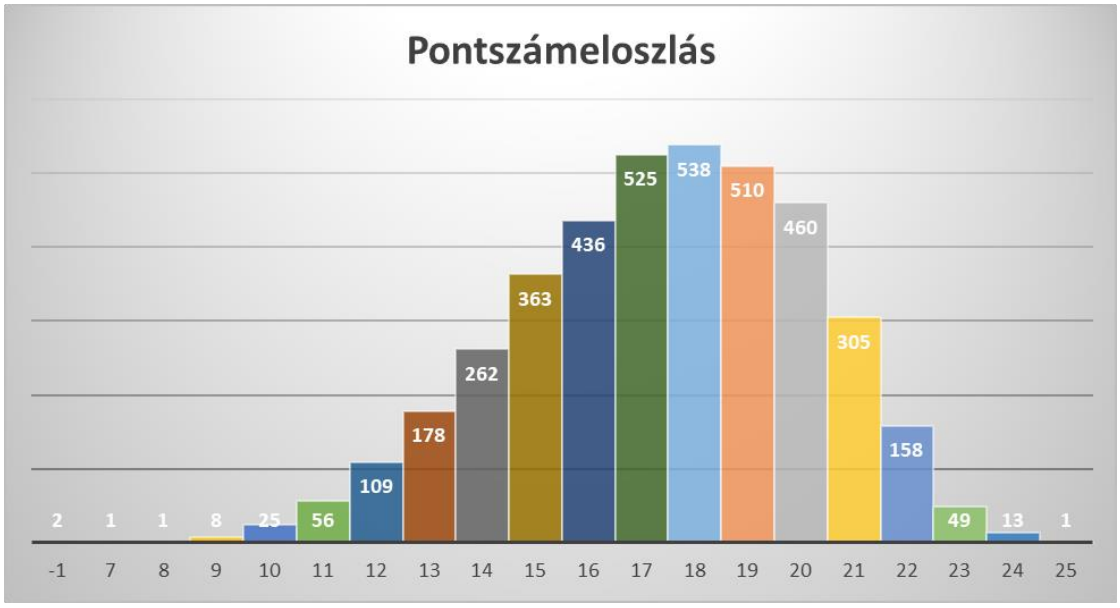
Az első tesztfuttatáshoz képest jelentősen javultak az átlagpontszámok. Különösen a 4-5 fős játéknál, majdnem 1 egész ponttal növekedett az eredmény.



5. ábra

Az összes játék egyesített pontszámeloszlása: (6. ábra)

Legtöbbször 17-18 pontot ért el a mesterséges intelligencia. A haranggörbe sokkal kiegyenlítettebb lett.



6. ábra

5.5 Második javított tesztfuttatás

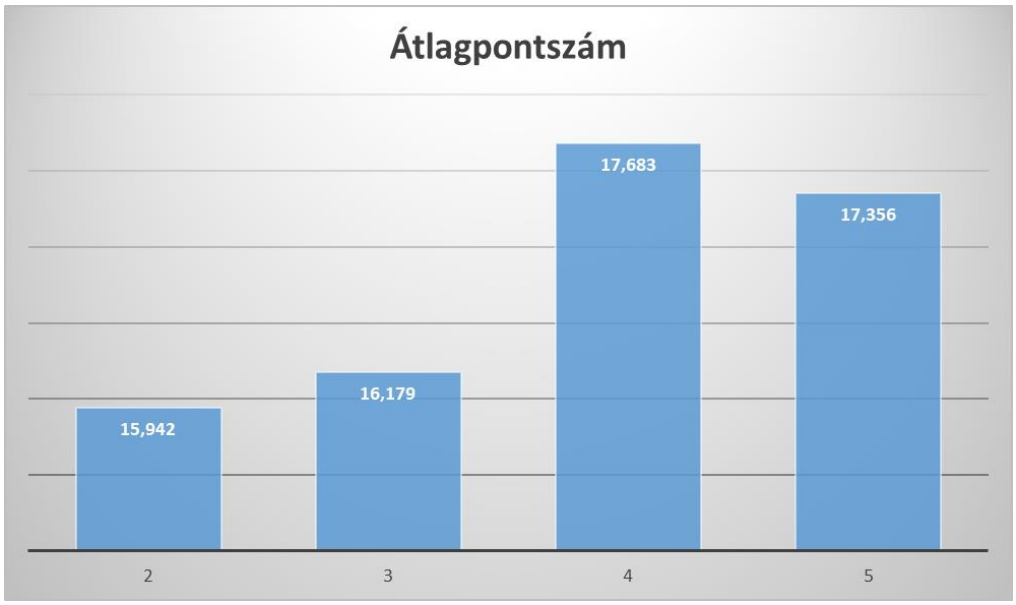
Személyes tapasztalatból kiindulva azon a véleményen voltam, hogy az olyan lapokról, amelyekből már csak egy példány van, információt kell adni, hogy véletlenül se dobja el

az, aki a kezében tartja. A saját játékaink során ezért gyakran megmutattuk egymásnak, mely lapjaink ötös számúak.

A tesztfuttatások során sokszor fordult elő olyan helyzet, hogy az utolsó akciót választotta a mesterséges intelligencia, tehát a legrégebben kézben tartott lapot eldobta. Ilyen esetben gyakran került eldobásra ötös számú lap, amelyekből csak egy példány van, így a maximális pontszám elérése lehetetlen lett. Ennek javítása érdekében módosítottam a programot: amikor utalást ad egy játékos, ha nem tud kijátszható, vagy eldobható felesleges lapról információt adni, akkor megmutatja a kézben tartott ötös lapokat. Először a számáról, majd a kártya színéről is kapnak utalást, hogy ne a helytelen ötös lapot játszák ki, amikor a tűzijáték sorozatokban ahhoz a ponthoz érnek.

A pontszámok átlagai játékosszámra lebontva: (7. ábra)

Az átlagok legalább 0,5 ponttal csökkentek a korábbi verzióhoz képest.

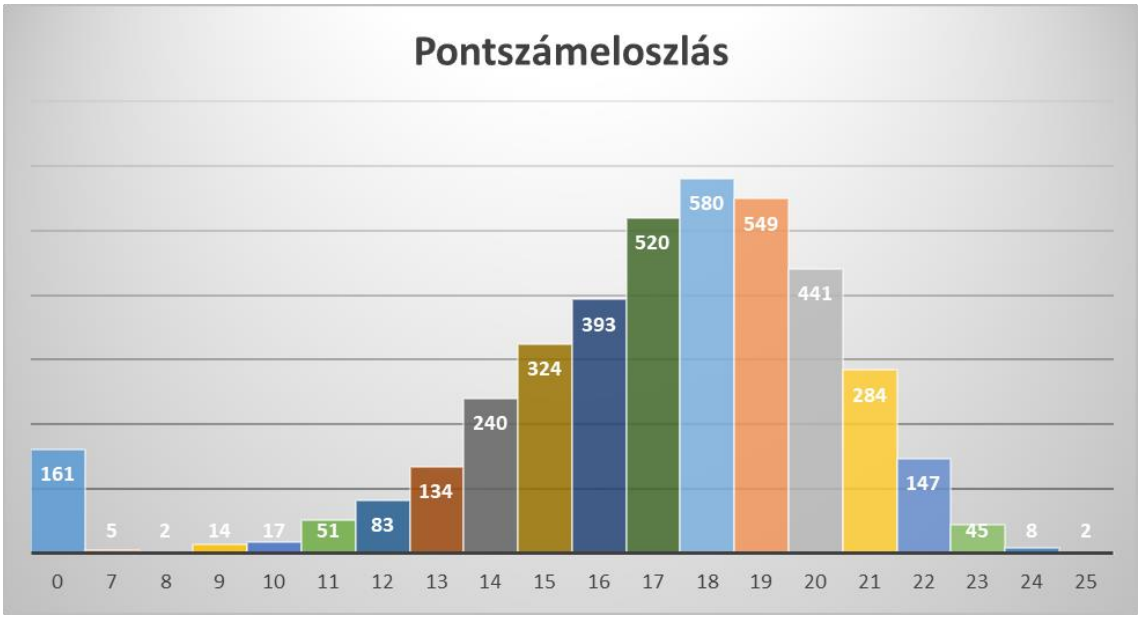


7. ábra

Az összes játék egyesített pontszámeloszlása: (8. ábra)

A tesztfuttatások kis részében az algoritmus még el is veszítette a játékot.

A kapott eredmények alapján levontam a következtetést, miszerint az ötös kártyák megmutatása nem egy jó stratégia. A következő tesztfuttatáshoz kitöröltem az ehhez szükséges kódrészletet, és úgy folytattam a munkát.



8. ábra

5.6 Harmadik tesztfuttatás

Az akció választó algoritmus nagyrészt ugyanaz maradt, csak kis mértékben módosítottam rajta.

Az első prioritás kártya kijátszás, a korábban ismertetett szabályok szerint.

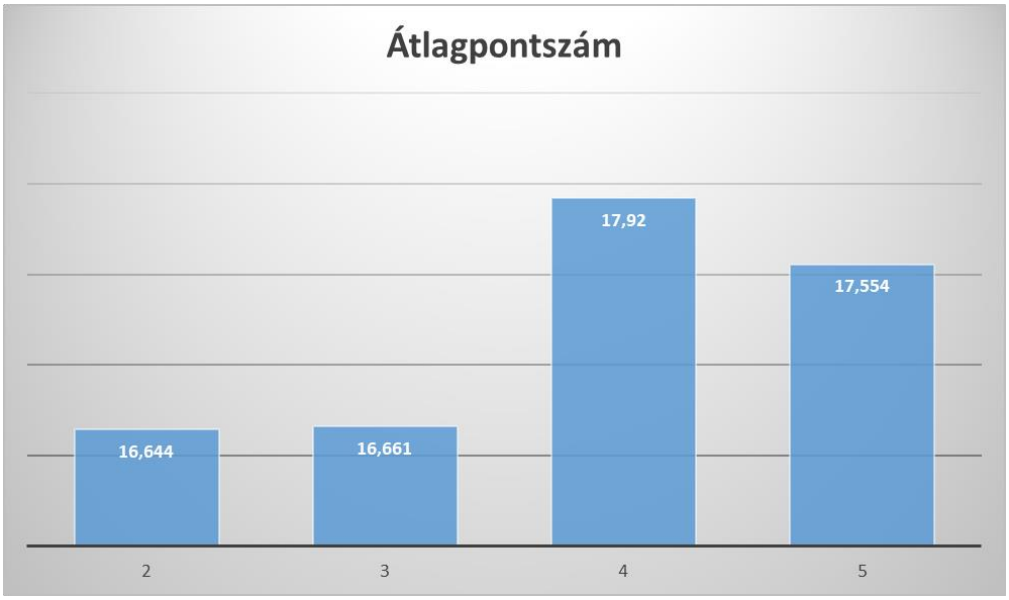
A második prioritás az utalás adás. Elsődlegesen a lerakható lapról ad információt. Ha nincs ilyen, vagy nem lehet egyértelműen megmutatni, akkor eldobható felesleges lapot mutat. Ezt az előző fejezetben ismertetett logika szerint teszi. Ha a játékos kezében nincs ilyen felesleges lap, vagy nem lehet egyértelműen megmutatni, akkor eldobható duplikátumot mutat.

Eldobható duplikátum olyan kártya, amelyből több egyforma példány is van a pakliban, így nincs feltétlen szükség mindegyikre. Amikor a játékos kezében nincs kijátszható vagy felesleges eldobható lap, akkor ezekről a kártyákról adunk információt. Az algoritmus elsősorban a négyes számú lapokat mutatja meg, majd csökkenő sorrendben a hármasokat, vagy ketteseket, ha nem volt magasabb számú lap a játékos kezében. Csak akkor ad információt egy adott számú lapról, hogy ha a szám nem kijátszható. Ez azt jelenti például, hogy nem mutat meg négyes számú lapot, ha már ki van játszva egy hármas számú lap, hiszen akkor kijátszhatóként értelmezné az algoritmus, nem pedig eldobható duplikátumként.

A harmadik prioritás egy felesleges kártya eldobása, ha korábban kapott információt ilyen lapról.

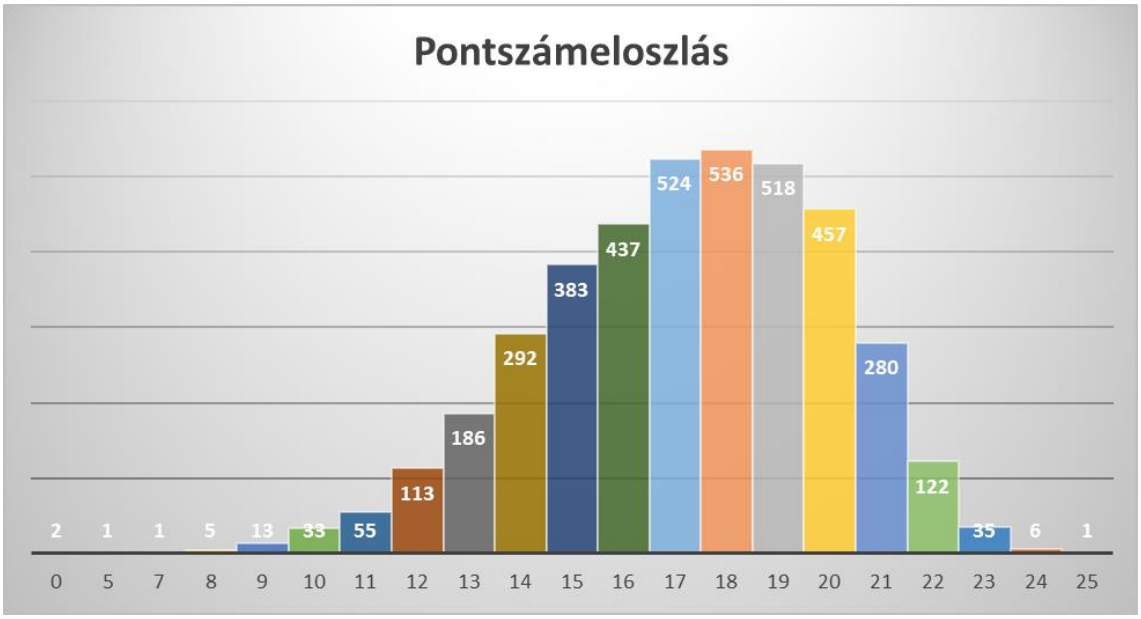
A negyedik prioritás egy duplikátum kártya eldobása, ha korábban kapott információt az itt ismertetett módszer szerint.
Az utolsó prioritás a legrégebben kézben tartott kártya eldobása. Ez az ötös számú lapok megmutatásakor módosult: ha ismeri a legrégebben kézben tartott lap számát, és az ötös, akkor a következő lapot vizsgálja ugyanezzel a feltétellel. Ha talál egy lapot, amelyik nem ötös, akkor azt dobja el.

A pontszámok átlagai játékosszámra lebontva: (9. ábra)
A második tesztfuttatáshoz képest csak a 2 fős játék átlagpontszáma javult (+~0,2), a többi átlaga viszont romlott.



9. ábra

Az összes játék egyesített pontszámeloszlása: (10. ábra)
A korábbi eredményekhez hasonlóan 18 pontot szerzett legtöbbször az algoritmus.



10. ábra

5.7 Negyedik tesztfuttatás

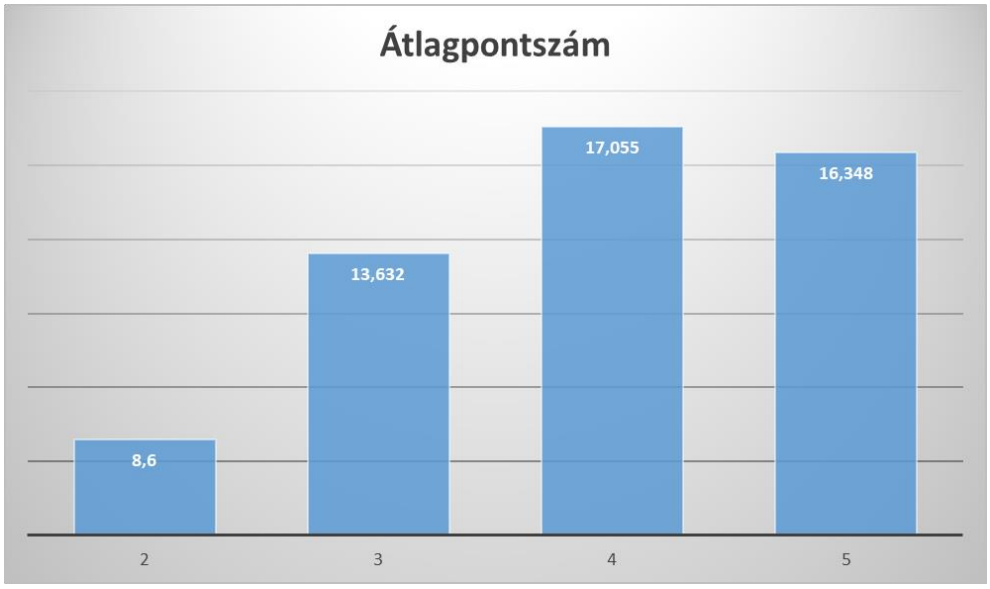
Miután megvalósítottam az 5 lépéses akcióválasztó algoritmust, aminek ötletét a Mathematics Magazine által ismertetett stratégiákból szereztem, a saját tapasztalataim alapján szerettem volna hasonlóvá tenni a gépi intelligenciát ahhoz a játékstílushoz, amit az élő játékban követek.

A tesztfuttatások során sokszor fordult elő olyan helyzet, hogy az algoritmus nem tudott információt adni egy játékos kezében lévő kijátszható lapokról, mert nem tudta egyértelműen megmutatni ezeket. Ha egy adott lap kijátszható, csak akkor ad a számról információt, ha az összes olyan számú lap a kezében kijátszható. Ha volt olyan lap a kezében, amely ugyanolyan számú, de nem kijátszható, akkor megvizsgálja az adott lap színét. Ha nincs másik lap a kezében, amely ilyen színű, akkor ad információt a lap színéről. Minden további esetben nem lenne egyértelmű az utalás adás, így, ha több lapnak egyezett a száma és a színe is az adott kijátszható lappal, nem tudta megmutatni a kijátszható lapot.

Ennek javítása érdekében átírtam az utalás adó algoritmust. Ebben a verzióban amikor csak tud, egyértelműen mutatja meg a kijátszható kártyákat, viszont, ha a fent ismertetett helyzet állna elő, megmutatja a kártya számát. A következő játékos, akinek van lehetősége utalást adni, információt kell adjon a helytelenül megmutatott kártya színéről. Ha egy kártyáról minden információ ismert, nincs lehetőség helytelenül kijátszani.

A pontszámok átlagai játékosszámra lebontva: (11. ábra)

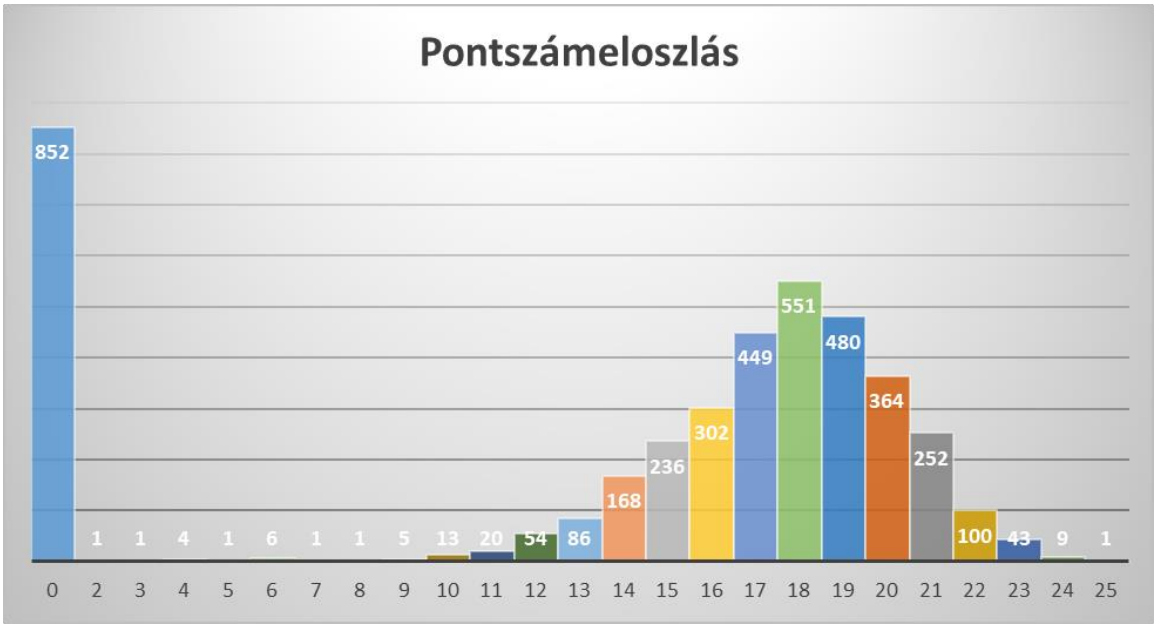
Az átlagpontszámok jelentősen kevesebbek lettek, különösen a 2 fős játéknál.



11. ábra

Az összes játék egyesített pontszámeloszlása: (12. ábra)

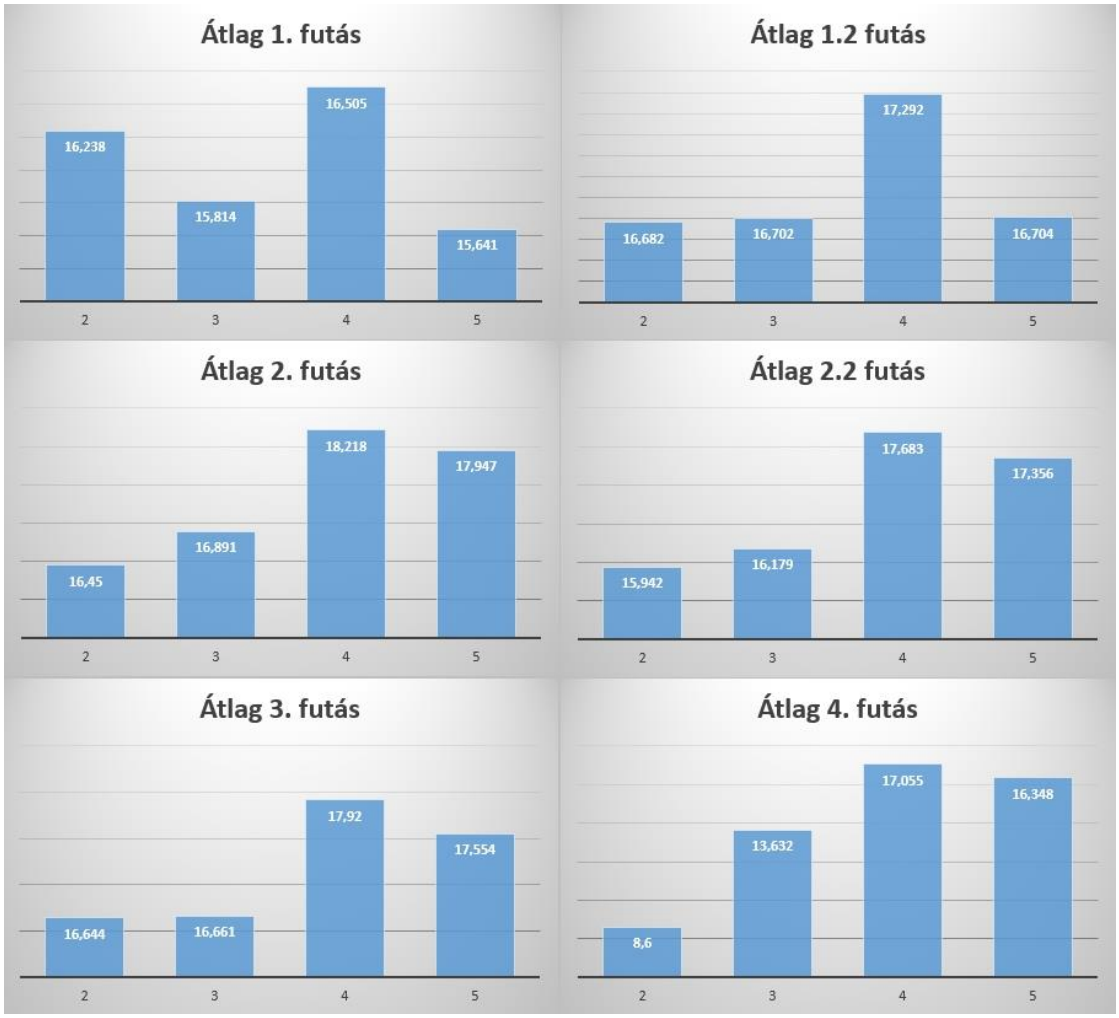
Az eredmény meglepően sokat romlott. A tesztek majdnem negyede vereséggel végződött (852 esetben 0 pontot ért el az algoritmus). Korábban nem volt rá példa, hogy ennyi alkalommal ilyen alacsony pontszámot érjen el az algoritmus.



12. ábra

6. Összegzés

Végeredményben az alábbi következtetéseket vonhatom le a tesztesetek vizsgálatából:



13. ábra

A különböző szimulációk közül a legmagasabb átlagpontszámot az algoritmus a második tesztfuttatáskor érte el. Ekkor a stratégia a következő volt: ha csak lehetséges, kijátszik lapot. Ha nem tud kijátszani kártyát, akkor információt ad, elsődlegesen kijátszható lapról, de ha nincs ilyen, vagy nem lehet egyértelműen megmutatni, akkor eldobható felesleges kártyáról ad utalást. Ha nem tud információt adni, akkor eldob felesleges lapot, ha erről kapott korábban információt. És ha ezt az akciót se tudja választani, akkor a legrégebben kézben tartott lapot dobja el.

A két fős játék kivételével minden játékosszámra ez a futtatás volt a legmagasabb pontszámú.

Az összes teszt közül a négy fős játék érte el a legmagasabb átlagpontszámot, 18,218 pontot. Ez azért meglepő, mert maga a Hanabi játék, és a kooperatív játékok

általánosságban sokkal könnyebbek kevesebb játékosszámmal. Az volt az elvárás, hogy a statisztikák ehhez igazodnak majd, de ez nem így történt. Lehetséges, hogy csak az általam megírt algoritmus az, ami több pontot képes elérni négy fős játékok során, és az emberi, élő játék továbbra is két fővel a legegyszerűbb.

Az élő játékban általában 20 pont körül lehet az átlagpontszám, bár erről pontos statisztikát nem készítettem, és interneten sem találtam. A korábbi, személyes játékban szerzett tapasztalataim alapján azt gondoltam, hogy ha több konvenciót és szabályt írok az algoritmusnak, akkor jobbak lesznek az eredmények. Egy élő játékban több mindenre figyel az ember, mint amennyit a programban megvalósítottam. Így lehet, hogy ha az összes szokást és konvenciót belefoglalnám, mely személyes játszmáinkat jellemzi, a program is jobb eredményt érne el. Bár a tesztfuttatások későbbi verziói rosszabbul teljesítettek, biztos vagyok benne, hogy az elképzelt fejlesztési irány jobb eredményeket hozna.

Célom az volt, hogy egy emberközeli szoftvert fejlesszek, amivel le lehet ülni játszani. Úgy érzem, ezt sikerült megvalósítanom, a program emberi játékosok számára is élvezhető.

Irodalomjegyzék

1. Mathematics Magazine 2015 február, Vol. 88, No. 1 – How to Make the Perfet Fireworks Display: Two Strategies for Hanabi
    a. https://quuxplusone.github.io/blog/images/how-to-make-the-perfect-fireworks-display.pdf
2. Marc G. Bellemare, Research Scientist, Google Brain, Adjunct Professor, McGill University, Canada – A cooperative benchmark: Announcing the Hanabi Learning Environment
    a. http://www.marcgbellemare.info/blog/a-cooperative-benchmark-announcing-the-hanabi-learning-environment/

Nyilatkozat

Alulírott programtervező informatikus szakos hallgató, kijelentem, hogy a dolgozatomat a Szegedi Tudományegyetem, Informatikai Intézet Számítógépes Optimalizálás Tanszékén készítettem, programtervező informatikus Bsc diploma megszerzése érdekében.

Kijelentem, hogy a dolgozatot más szakon korábban nem védtem meg, saját munkám eredménye, és csak a hivatkozott forrásokat (szakirodalom, eszközök, stb.) használtam fel.

Tudomásul veszem, hogy szakdolgozatomat / diplomamunkámat a Szegedi Tudományegyetem Informatikai Intézet könyvtárában, a helyben olvasható könyvek között helyezik el.

Szeged, 2020. május 16.

Hajdu Katalin