



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Hálózati Rendszerek és Szolgáltatások Tanszék

Szabó Ákos Bálint

# **MOZGÓ ÜGYNÖK STRATÉGIÁK HÁLÓZATMENEDZSELÉSRE JÁTÉKELMÉLETI ALAPOKON**

KONZULENS

**Dr. Imre Sándor**

BUDAPEST, 2021

# Tartalomjegyzék

<b>Kivonat.....</b>	<b>4</b>
<b>Abstract.....</b>	<b>5</b>
<b>1 Elméleti bevezető.....</b>	<b>6</b>
1.1 Hálózat típusok .....	6
1.2 Mozgó Ügynök .....	7
1.2.1 Alap koncepció .....	7
1.2.2 Miért jó a mobil ügynök?.....	8
1.3 Játékelmélet .....	10
1.3.1 Játékok csoportosítása.....	11
1.3.2 Egyensúlyi helyzet.....	12
1.3.3 Kevert Stratégiák .....	13
1.3.4 Hogyan lehet felhasználni a játékelméletet?.....	16
<b>2 A szimulációs keretrendszer .....</b>	<b>17</b>
2.1 Előkészületek .....	17
2.2 A keretrendszer felépítése.....	18
2.2.1 Ügynök osztály .....	18
2.2.2 Mozgó ügynökök osztály.....	19
2.2.3 Csomópont osztály.....	19
2.2.4 Gráf osztály.....	20
2.2.5 HUD osztály .....	20
2.2.6 Main osztály.....	21
<b>3 Szimulációs eredmények .....</b>	<b>22</b>
3.1 Első hálózat.....	22
3.1.1 Első mérés.....	23
3.1.2 Második mérés.....	26
3.1.3 Harmadik mérés .....	28
3.2 Második hálózat.....	31
3.2.1 Első mérés.....	31
3.2.2 Második mérés.....	34
3.2.3 Harmadik mérés .....	37
3.3 Összefoglalás .....	39

<b>Irodalomjegyzék.....</b>	<b>40</b>
-----------------------------	-----------

## Kivonat

Az IOT terjedésével egyre nagyobb és nagyobb hálózatok jönnek létre világszerte. A hálózatokat alkotó elemek karbantartása, rendeltetésszerű működtetése egyre nehezebbé válik a hagyományos hálózatmenedzselési eszközökkel. A dolgozatomban az egyik feltörekvő, új menedzsment módszert az elosztott hálózat alapú, mozgó ügynökök segítségével megvalósított rendszert mutatom be.

Az értekezésemben ismertetem a mobil ügynökök alap koncepcióját és, hogy miért érdemes alkalmazni őket. Ezenfelül betekintést nyújtok a Neumann János által megalkotott játékelméletbe, és hogy az ebben definiált eszközök miként segíthetnek a hálózatmenedzselésben felmerülő kérdések megválaszolásában.

A munkám során egy szimulációs keretrendszert valósítottam meg, melyben játékelméleti paradigmák alapján működő mobil ügynököket alkalmazok egy általam generált elosztott hálózat menedzsmentjének ellátására. A dolgozatom részét képezi a szimulációs keretrendszerhez választott eszközök bemutatása és kiválasztásuk indoklása, illetve ismertetem magát a szimulációt és annak alkotó elemeinek működését.

Végül a megírt szimulációs keretrendszerben méréseket végeztem és az ezek során gyűjtött eredményeket és azok kiértékelését ismertetem.

## **Abstract**

With the spread of IoT, larger and larger networks are being created worldwide. The maintenance and proper operation of network components is becoming increasingly difficult with the traditional network management tools. In my dissertation, I present one emerging, new management method using distributed network that is based on mobile agents.

In my paper, I introduce the basic concept of mobile agents and why they should be employed. In addition, I present the Game Theory developed by János Neumann and how the tools defined therein can help answer some network management questions.

As a part of my work I have created a simulation framework in which I use mobile agents that are functioning based on game theory paradigms in order to manage a distributed network. In the course of my paper I review the softwares that I used to create the simulation framework, talk about why I chose them, and of course, present the simulation itself.

As the final part of my dissertation I describe the measurements made in the simulation framework and exhibit the conclusions drawn from the results.

# 1 Elméleti bevezető

Ebben a fejezetben röviden ismertetem a legalapvetőbb hálózat típusokat, bemutatom a mozgó ügynököket, valamint kifejtem, hogy ezek miért használhatóak hálózatmenedzselés céljára. Továbbá ismertetem a játékelmélet alapjait, és azt, hogy ezt miként lehet alkalmazni hálózatmenedzselés terén.

## 1.1 Hálózat típusok

Ahogy az internet szolgáltatásai elérnek minden ipari szektorba, úgy lesz egyre nagyobb az igény az újabb, jobb szolgáltatásokra. A felhasználók elvárásai drasztikusan megváltoznak. A jövőben ezen kritériumok lesznek a technológia fejlődés hajtómotorjai. Ezért a fejlődésmenetének megjósolása fontos a haladási irány meghatározásához.

Az ipari erőviszonyok eltolódtak a szolgáltatóktól a felhasználók irányába. A fogyasztók igényeit többé nem lehet kielégíteni a szolgáltatók által kifejlesztett tipikus módszerekkel. A személyre szabhatóság a legfontosabb igény a jövőre nézve, ehhez a hálózatnak kellően rugalmasnak kell lennie. A flexibilitás megvalósítását a hálózat menedzselésénél kell kezdeni.

A hálózat felügyelete általánosan kétféle lehet, centralizált vagy elosztott. Az elosztott hálózatoknál a jelzések és az erőforrások menedzselése az azt alkotó csomópontokban van lekódolva, míg a központosítottnál az irányítást lehetővé tevő funkciók a központi szerveren foglalnak helyet.

A két típus közötti választást sok tényező befolyásolja, többek között a hálózat mérete, a felhasználók száma, a központi szerverek teljesítőképessége és még számos egyéb. Amennyiben a hálózati forgalom jelentős része kisszámú csomópontból érkezik, központosított hálózat megvalósítása célszerű. A torlódások és adatvesztések elkerülése végett, ha több csomópont hozza létre a forgalmat, akkor már az elosztott hálózati modell megválasztása vezet jobb eredményre. Ez az egyre jobban elterjedő „Internet Of Things” (IoT) miatt fontos, mivel itt sok kihelyezett eszköztől folyamatos forgalmat várunk [1][2].

## 1.2 Mozgó Ügynök

Ebben az alfejezetben a mobil ügynökök tulajdonságait, és használatuk jótékony hatásait fejtem ki.

### 1.2.1 Alap koncepció

Az előző alfejezetben említettem, hogy egyre nagyobb a felhasználók által támasztott igény a hálózatok minőségére és megbízhatóságára. Ezzel párhuzamosan mutatkozik egy trend az egyre heterogénebb hálózatok felé a kommunikációs környezetekben. Ez a diverzitás olyan hálózat menedzsmentet igényel, amely nagyobb tudással és kiképzéssel rendelkezik. Ezen hálózatok igazgatásához nagy mennyiségű adatra, illetve információra van szükség magáról a hálózat működéséről. Ezen adatokat elemezni kell, mielőtt bárminemű menedzsment megtörténhetne. A hálózatokkal szemben támasztott felhasználói igények, és a hálózatoknál jelentkező diverzitás a fő hajtóerői az ügynökök kutatásának, fejlesztésének.

Ahelyett, hogy egy központosított, rendszerint nagy, a rendszer teljes intelligenciáját magába foglaló alkalmazást használnák, több kisebb rendszer, illetve ügynök kapcsolódik össze, hogy kooperatívan megoldjanak egyes problémákat. Ez nem csak egyszerűen azt jelenti, hogy a rendszert felbontjuk több alegységre, hanem például több olyan központosított alkalmazás összekötését egy olyan kommunikációs rendszerrel, mely egyenként más-más részeit tudja egy problémának megoldani. Ez lehetővé tenné, hogy megvitassák nézeteiket és így stratégiát alkossanak a feladat elvégzésére, kiértékelésére. Ez a probléma megoldási metódus a „Distributed Problem Solving” (DPS) azaz „Elosztott Problémamegoldás”, minden egyes résztvevőjét a folyamatnak, ügynöknek tekinthetünk.

Annak ellenére, hogy egy ügynök alapú rendszer megvalósítható bármely kliens/szerver technológiával, különbözik tőlük, mivel nem különül el élesen, hogy ki a kliens és ki a szerver. Minden ügynök részt vesz a feladatok ellátásában, a mindenkori fejlesztő által, vagy a dinamikusan, az ügynököket felügyelő entitás által megadott rendeltetés szerint.

A helyváltoztatás képessége nem általános tulajdonsága az ügynököknek. Lehetnek helyhez kötöttek is, ebben az esetben konvencionális módszerekkel tudnak kommunikálni, mint a „remote procedure calling, azaz távoli eljáráshívással, illetve

üzenetváltással. Az olyan ügynököket melyek nem tudnak mozogni „statikus ügynököknek” hívjuk. A statikus ügynök csak abban a rendszerben lát el feladatokat, amelyben létrejött. Ha más rendszerből van szüksége információra, akkor a fentebb említett módokon kapcsolatba lép egy másik ügynökkel, vagy rendszerrel. Ezzel ellentétben a mozgó ügynök nincs „odaláncolva” a rendszerhez, amelyben megkezdte működését, hanem szabadon mozoghat a hálózat „host”-jai között. A tartózkodási helyéről képes az „állapotát” és kódját egy másik hálózati elemre átszállítani és ott folytatni a működését. Állapot alatt az ügynök olyan attribútumait értjük, melyek a munka folytatásakor segítenek a következő lépés meghatározásában. A kód egy objektum orientált kontextusban azt jelenti, hogy az osztály kódja szükséges az ügynök működéséhez [3].

A mozgó ügynöknek nincsen egy teljesen kiforrott, általános definíciója, de a funkcionalitása jól körül határolható. A mozgó ügynök olyan program, ami különböző tevékenységeket hajt végre a felhasználó helyett. Egy mobil ügynöknek megvan az az egyedülálló képessége, hogy egy hálózat egyik rendszeréről egy másik, ugyanezen a hálózaton lévő rendszerre utazzon. Ez a képesség lehetővé teszi azt, hogy az ügynök egy olyan rendszerhez utazzon, melynek egy objektumával kapcsolatba szeretne lépni, ezzel ki tudja használni azt, hogy ugyanazon a hálózaton van, mint az adott objektum. Ezenfelül a mozgó ügynök aktívan reagál a körülmények alakulására, képes tanulni és együttműködni más ügynökökkel.

## **1.2.2 Miért jó a mobil ügynök?**

A mobil ügynökök mellett szóló érveket gyűjtöttem össze ebben az alfejezetben.

### **1.2.2.1 Csökkenti a hálózat leterheltségét**

Az elosztott rendszerek gyakran támaszkodnak kommunikációs protokollokra, melyek több interakciót használnak, hogy egy adott feladatot elvégezzenek. Ennek következtében jelentősen megnő a hálózati forgalom. A mobil ügynökök lehetővé teszik a felhasználók számára, hogy „becsomagolják” a párbeszédet és elküldjék a cél „host”-hoz, ahol a kommunikáció helyben megtörténhet [3].



#### **1.2.2.2 Aszinkron módon és önállóan működnek**

A mobil eszközök gyakran támaszkodnak drága vagy gyenge hálózati kapcsolatokra. Egy olyan folyamat, aminek folyamatosan nyitott kapcsolatra van szüksége nagy valószínűséggel gazdaságilag vagy technikailag nem kivitelezhető. A problémát orvosolandó a mobil ügynökbe bele lehet ágyazni a folyamatot és el lehet küldeni a hálózatra, majd pedig a feladat végeztével az ügynök begyűjthető.

#### **1.2.2.3 Dinamikusan alkalmazkodnak**

A mobil ügynökök érzékelik a működési környezetüket és automatikusan reagálnak a változásokra. Sok mobil ügynöknek megvan az az egyedülálló tulajdonsága, hogy felosztják maguk között a hálózatot, ezzel biztosítva egy probléma optimális megoldását.

#### **1.2.2.4 Megfigyelés és értesítés**

Egy ügynök megfigyelés alatt tud tartani egy információforrást, anélkül, hogy függne az őt létrehozó rendszertől. Gyakran fontos, hogy a megfigyelő ügynök élettartama, hosszabb, vagy független a számítógépes folyamattól, ami létrehozta.

#### **1.2.2.5 Párhuzamos feldolgozás**

Tekintve, hogy a mobil ügynökök tudják klónozni magukat a hálózatban, így egy másik felhasználási módjuk a párhuzamos folyamatok feldolgozása.

#### **1.2.2.6 Elosztott információ begyűjtés**

Ahelyett, hogy nagy mennyiségű adatot mozgatnánk a keresőmotorhoz, hogy az kereső indexeket kreáljon, az ügynökök elmehetnek a távoli információforrásokhoz, és helyben létrehozhatják a kereső indexeket, amiket később be lehet gyűjteni.

#### **1.2.2.7 Tárhely takarékos**

Az erőforrás felhasználás korlátozott, mivel a mobil ügynök csak egy csomóponton van jelen egy időben. Ezzel ellentétben a statikus szerverek duplikálják a funkcionalitásukat minden egyes csomóponthoz. A mobil ügynökök viszik magukkal a funkcióikat, így azokat nem kell lemásolni.

### **1.2.2.8 Robusztusság és hiba tűrés**

Amikor egy elosztott rendszer elkezd meghibásodni, mobil ügynököket lehet alkalmazni, hogy az érintett területeken, a megfelelő funkciók jobban hozzáférhetőek legyenek. Például a hiba javító és észlelő ügynökök száma könnyedén növelhető. Mindemellett valamilyen fajta ügynök menedzsment szükséges annak biztosítására, hogy az ügynök alapú rendszer megfelelő működése biztosítva legyen.

### **1.2.2.9 Heterogén rendszerek támogatása**

A mobil ügynökök leválasztását a „host”-okról a „mobility framework” (mobilitásért felelő keretrendszer) végzi. Amennyiben a keretrendszer működőképes, az ügynök megcélozhat bárminemű rendszert.

### **1.2.2.10 Kényelmes fejlesztés**

Mobil ügynök alapú elosztott hálózatok készítése viszonylag könnyű. A nehéz rész a „mobility framework”, de amikor már az is elkészült, alkalmazások fejlesztése elképesztően könnyűvé válik. A mobil ügynökök terjedésével szükség lesz a számukra készített magas szintű gyors alkalmazások fejlesztésére alkalmas fejlesztői környezetre is.

### **1.2.2.11 Könnyű frissítési lehetőségek**

Egy mobil ügynököt bármikor az ember kénye-kedve szerint virtuálisan le lehet cserélni. Ezzel ellentétben szerverek frissítése nehézkes, illetve bonyolult, főleg akkor, ha a frissítés ideje alatt meg akarjuk őrizni a szolgáltatásunk minőségét [4].

## **1.3 Játékelmélet**

A játékelmélet az alkalmazott matematika egyik ága, ami olyan helyzetek analizálására alkalmas mely résztvevőinek, a játékosoknak a döntései nem függetlenek egymásétól. Tehát a játékosok a saját stratégiájuk kialakításánál figyelembe veszik a többi játékos választási lehetőségeit. Egy játék végkifejlete, minden játékos által meghozott lehető legoptimálisabb döntéseket írja le. A résztvevők érdekei lehetnek azonosak (együttműködés), ellentétesek(versengés), illetve ezek keveréke.

Annak ellenére, hogy a játékelméletet fellehet használja szalon játékok elemzésére, ennél sokkal sokszínűbb és széleskörűbb felhasználási lehetőségei vannak. A játékelmélet szülő atyjai Neumann János és Oskar Morgenstern, azért hozták létre,

hogy gazdasági problémákat oldjanak meg a segítségével. A "The Theory of Games and Economic Behavior" című könyvükben kifejtették, miszerint a fizikai tudományok leírására létrehozott matematikai modellek nem alkalmasak a gazdasági folyamatok pontos leírására. Megfigyelték, hogy a gazdaság nagyban hasonlít egy játékra, melyben a résztvevők próbálják megjósolni a többi fél döntéseit, de ennek leírásához új modellekre volt szükség.[5]

A játékelméletet azóta széleskörűen alkalmazták, olyan esetekben, ahol a résztvevő felek döntéseinek interakciójával jön létre a végeredmény. A stratégia döntéshozatal és a játékosok által irányított események esetén jóval leiróbb képet ad, mint a klasszikus valószínűségszámítás. Felhasználható politikában, üzleti életben, állatvilágban és az informatikában.

Tehát összegezve a játékelmélet a versengésről és együttműködésről szóló formális kutatás. Elvei akkor használhatóak, ha a résztvevők cselekvései nem függetleníthetők el egymástól. A résztvevők lehetnek önálló emberek, csoportok, cégek, vagy ezek bármilyen kombinációja. A játékelmélet koncepciói, megadják a rendszert, melyben elemezni és értelmezni lehet a stratégiai forgatókönyveket.[6]

### **1.3.1 Játékok csoportosítása**

A játékokat különféle csoportokba lehet sorolni a jellemző tulajdonságaik alapján. A legalapvetőbb ilyen csoportosítás a játékosok száma alapján történik. Megkülönböztetünk egy szereplős, két szereplős és  $n$  szereplős játékokat, ahol minden játéknak ezen kívül még megvannak a saját jellemzői. Ismét leszögezném hogy a szereplőknek nem kell egy személynek lennie, beszélhetünk vállalatokról vagy épp politikai pártokról, illetve minden egyéb csoportosulásról is.

A teljes információs játékok esetén a szemben álló felek egymás minden lehetséges lépésével tisztában vannak, tudnak számolni vele. Egy ellenpélda a póker, ami a nem teljes információs játékok táborát bővíti ugyanis itt az ellenfelünk lapjai az utolsó pillanatig rejtve vannak előttünk.

A játékosok céljainak egyezése és eltérése alapján is lehet csoportosítani a játékokat. A zéró-összegű játékokban, azaz másnéven a tisztán versengő játékokban a szemben álló felek csak egymás kárára tudnak nyereséget szerezni. Erre tökéletes példa a fent említett póker. A változó-összegű játékokban viszont, annak ellenére, hogy

lehetnek győztesek, illetve vesztesek nem kell minden nem nyertes félnek teljesen rosszul járnia.

A változó-összegű játékokat tovább tudjuk csoportosítani együttműködéses és versengéses játékokra. Az együttműködéses játék azt definiálja, hogy az egyes csoportok, koalíciók milyen nyereségeket szerezhetnek a tagok együttműködésével. A csoportok kialakulásának menete nincs meghatározva benne. Például a játékosok lehetnek parlamenti pártok. Minden pártnak van egy meghatározott ereje, mely attól függ hány képviselőjük van a parlamentben. A kooperatív játék leírja, mely pártok koalíciói alkothatnak parlamenti többséget, de az együttszavazásról való tárgyalásokat már nem. A kooperatív játékokkal ellentétben, a versengéses játékok vizsgálata a stratégiai döntések elemzésével foglalkozik. A versengéses játékok paradigmája az, hogy a játékosok döntéseinek sorrendje és ideje életbevágó a végkifejlet meghatározásában. A játékosok együttműködése gyakran megjelenik a versengéses játékokban is, amikor ez mindkét félnek előnyt jelent [7].

#### **1.3.1.1 Példa a változó összegű játékokra**

Kutatásom során talákoztam egy számomra nagyon érdekes konklúziójú játékkal. A feladathoz két személyre van szükség, akik egy számítógépes program segítségével nyúl vadászatot szimulálnak. A játék két fajtáját játszották a kísérleti alanyokkal. Az elsőben teljesen együttműködve kellett a lehető legtöbb nyulat elfogniuk. A másodikban viszont egymással versenyezve volt a cél, hogy ki tud több nyulat megfogni. Ez az opció magába foglalta annak a lehetőségét, hogy a másik játékos elől elhajtsa ellenfele a nyulakat, illetve le vadássza azokat az orra előtt. Ami meglepő volt, hogy a második típus annak ellenére, hogy tartalmaz negatív, a nyúl vadászat hatékonyságát csökkentő elemeket, mégis összességében versengve fogták meg a legtöbb nyulat, nem pedig kooperatíván. Tehát a versengéses nyúl vadászat hatékonyabb volt, mint a kooperatív.

#### **1.3.2 Egyensúlyi helyzet**

Neumann János tétele szerint minden olyan kétszemélyes játéknak, ami megfelel a lentebb felsorolt három kritériumnak, van olyan egyensúlyi helyzete, melybe belekerülve, ha az egyik fél is változtat a stratégiáján az neki kedvezőtlen lesz, ezt az egyensúlyi állapotot nyeregpontnak nevezzük [8].

Véges abban az értelemben, hogy mindkét játékos számára, mindegyik lépés alkalmával véges a választási lehetőségek száma, és abban az értelemben is, hogy a játék véges sok lépésben véget ér.

Zéró összegű, azaz amennyit az egyik játékos nyer, annyit veszít a másik.

Teljes információs, azaz mindkét játékos pontosan ismeri mind a saját maga, mind az ellenfele összes választási lehetőségét, és azt is, hogy a játék melyik kimenetelére mennyire kedvező a saját, illetve ellenfele értékrendje szerint.

Többszemélyes kooperációmentes játékokra John Nash általánosította Neumann tételét. Nash-egyensúlynak nevezzük olyan különböző stratégiák összességét, melyek esetén egyik játékos se jön ki rosszul, ha a stratégiák közül a neki kijelölt stratégiát hajtja végre.

### **1.3.3 Kevert Stratégiák**

Egy olyan játékban, melyben a játékosok egyféle, „tiszta” stratégiát folytatnak, nem mindig van Nash-egyensúlya. A „tiszta” stratégia azt jelenti, hogy, ha egy ember az adott játékon belül, mehet előre, jobbra, balra, illetve hátra, akkor csak az egyik irányba megy, minden esetben. Ellenben, elképzelhető, hogy a játékosok véletlenszerűen választhatnak a „tiszta” stratégiáik közül valamilyen valószínűség szerint. Az ilyen módon sorsolt döntések sorozatát kevert stratégiának hívjuk. Nash bebizonyította 1951-ben, hogy a véges játékoknak van Nash-egyensúlyuk, amennyiben a játék folyamán kevert stratégiákat alkalmazhatnak a játékosok [9][10].

#### **1.3.3.1 Példa a kevert stratégiákra**

Tegyük fel, hogy egy fogyasztó megveszi egy szoftver csomag licencét, beleegyezvén ezzel bizonyos felhasználási feltételekbe, ám szándékában áll megsérteni ezeket. Ellenben az eladó tudni szeretné, hogy a fogyasztó tartja-e magát a szerződésben foglaltakhoz, de ehhez ellenőrzéseket kell tartania, melyek nagy költséggel járnak. Amennyiben az eladó csaláson kapja a fogyasztót, hatalmas büntetést hajthat be rajta. Az 1. ábra mutatja be a nyereségeket egy ilyen ellenőrzés játékhoz. Az I. játékos a szolgáltató, a II. játékos pedig a fogyasztó.

		II betartja → megszegi	
		0	10
I nem ellenőriz ↑	0	-10	↓
	0	-90	
ellenőriz	-1	-6	←

**1. ábra: I. játékos (szolgáltató) és II. játékos (fogyasztó) nyereségének ábrázolása**

A standard végkimenetel, mindkét fél számára 0 nyereséget hoz. Ebben az esetben a szolgáltató nem felügyel, és a fogyasztó betartja a szerződésben foglaltakat. Viszont, amennyiben nem várható ellenőrzés a szolgáltató oldaláról, a fogyasztónak megéri megszegni a feltételeket, mivel ez 10 nyereséget hoz neki. Ellenben ez a szolgáltatónak 10 veszteséget jelent. Abban az esetben, amikor a fogyasztó betartja a szabályokat és a szolgáltató ellenőriz a fogyasztó bevétele 0 marad, ellenben az ellenőrzés -1 veszteséget okoz a másik félnek. Amennyiben a szolgáltató ellenőriz és a fogyasztó szerződést segetett, az az előbbinek -6 veszteséget okoz, mert bár lefűlelte a fogyasztót a büntetés nem fedezi a károkat. Itt a fogyasztó nagyon rosszul jár, mivel súlyos büntetést kell fizetnie, -90-ben zár.

Az I. játékos annak örülne legjobban, ha a II. játékos minden esetben betartaná a szerződést, de a II. játékos döntése felett nincs hatalma. Az I. játékos szívesen ellenőriz, ha a II. játékos megszegi a szerződést, mivel -6 veszteség még mindig jobb, mint -10. Ezt próbálja szemléltetni az 1. ábrán a lefelé mutató nyíl.

Az 1. ábra körüli nyilak azt szemléltetik, hogy ennek a játéknak tiszta stratégiák esetén nincs egyensúlyi helyzete. Ha bármely játékos elkötelezi magát egy opció mellett, és mindig ugyanazt a döntést hozza, például, ha az I. játékos sosem ellenőriz, a legjobb válasz erre a másik féltől, az lenne, hogy elkezd csalni. Ebben az esetben a „nem ellenőrzés” már nem jó stratégia a szolgáltató számára. Mivel a Nash-egyensúlyban mindkét fél úgy választja ki a stratégiáját, hogy az a másik döntése alapján a számára a legtöbb jövedelmet hozza, ezért ebben a játékban a tiszta stratégiák követése esetében nincs egyensúlyi helyzet.

Felmerül a kérdés, hogy a játékosoknak mit kellene tenniük. Az egyik lehetőség, ha mindkét fél felkészül a legrosszabbra, a szolgáltató így mindig megfigyel, mert -6 veszteség jobb, mint -10, illetve a fogyasztó mindig betartja a szerződést, mert 0 nyereség jobb, mint -90 veszteség. Mindazonáltal ez nem Nash-egyensúly.

Az I. játékos kevert stratégiája az lenne, ha csak egy bizonyos valószínűséggel végezne ellenőrzést egyébként pedig nem. Az ellenőrzések végrehajtásának véletlenszerűvé tétele a költségsökkentésnek is hatékony, praktikus megközelítése. Még, ha nem is minden esetben történik meg az ellenőrzés, a tettenérés lehetősége meggátolhatja a fogyasztót a szerződés megszegésében.

A következő megfontolások megmutatják, hogy hogyan válasszuk meg az ellenőrzések valószínűségét, mellyel egyensúlyi helyzetre jutunk. Amennyiben az ellenőrzés valószínűsége túl alacsony, mondjuk 1 százalék, ekkor a II. játékos nyereségre számíthat, abban az esetben, ha mindig megszegi a szerződést. Tehát a fogyasztó mindig megfogja szegni, mint ha nem is lenne ellenőrzés.

Viszont, ha az ellenőrzés valószínűsége túl magas, például 20 százalék, akkor a várható veszteség szerződés megszegés esetén, vagyis a fogyasztónak egyáltalán nem fogja megérni csalni. Tehát ha az ellenőrzés valószínűsége túl magas vagy túl alacsony, akkor a fogyasztónak csak tiszta stratégiát éri meg folytatni, ami nem visz egyensúlyra.

Az egyetlen eset, ahol a II. játékos véletlenszerűen tud a lehetőségei közül választani az, mikor mindkét esetben ugyanazt a nyereséget tudja elérni. Sosem célravezető olyan stratégiát választani, mely gyengébb hozammal rendelkezik a játékosok döntése alapján, a többi opcióhoz képest. Nem nehéz belátni, hogy a fogyasztónak mindegy melyik stratégiát választja, ha a szolgáltató 10 százalék valószínűséggel ellenőriz. Ekkor ugyanis a várható nyereség szerződésszegés esetén ugyanaz, mintha betartaná azt.

Az I. játékos előző bekezdésben kifejtett stratégiájának következtében, a II. játékos is alkalmazhat kevert stratégiát. A szolgáltató ugyanannyi veszteséggel számolhat abban az esetben is, ha ellenőriz és abban az esetben is, ha nem. Tehát ebben az esetben mindkét játékosnak indifferens mindkét opciója. Ezzel definiáltuk ezen játéknak az egyetlen Nash-egyensúlyát. Mivel kevert stratégiákat alkalmazunk, ezért kevert egyensúlynak hívjuk. A várható veszteség a szolgáltatónak -2, a várható nyeresége a fogyasztónak 0 [11][12].

### 1.3.4 Hogyan lehet felhasználni a játékelméletet?

A játékelmélet segítségével sok a természetben az idő és evolúció során kifejlődött mechanikát vehetünk át az informatikába és így az elosztott hálózatok kezelésébe.

Ilyenre egy példa a „pózolás” jelensége. A több állatfajnál megfigyelhető, hogy a hímek, amikor versengenek a nőstényért, nem folytatnak vérremenő harcot, mert az mindkét (vagy több) hím egyed számára nagy kockázattal jár. A viadalt elkerülendő pózolnak egymással szemben. A felek egy előre „kigondolt” véletlenszerűen megválasztott ideig várnak, aki tovább vár az nyer. Minél tovább vár egy fél, annál kevesebb ideje jut táplálék szerzésre vagy másik nőstény keresésére. Tehát egyikük se fog végtelen időt az adott szituációban eltölteni. Hasonló helyzet állhat elő, amikor két mobil ügynök egyazon csomóponttal szeretne kapcsolatot létesíteni.

Fentebb említettem, a nyúl vadászatos játékot. A játék menete könnyen kivethető egy elosztott hálózat menedzsmentjére mobil ügynökökkel. Ebből kiindulva egy olyan szimulációs környezetet valósítottam meg, melyben a menedzsment feladatokat a kooperatívan és kompetitíven és végre tudom hajtani.



## 2 A szimulációs keretrendszer

A dolgozatom elkészítéséhez megalkottam egy szimulációs rendszert melyben a felhasználó által megadott csomópontú és maximális foksámú elosztott hálózaton, menedzsment folyamatokat valósítok mobil ügynökök segítségével. A mobil ügynökök kétféle, a játékelméletből vett stratégiát alkalmazhatnak, kooperatív vagy kompetitívet.

Ebben a fejezetben ismertetem, milyen előkészületeket tettem a szimulációs keretrendszer megírása előtt, milyen eszközök közül válogattam, és végül mi mellett döntöttem. Bemutatom a keretrendszer alap építőelemeit, osztályait, majd magának a keretrendszernek a működését.

### 2.1 Előkészületek

A szimulációs keretrendszerem hatékony megvalósításához megfelelő eszközökre volt szükségem. Korábbi tapasztalataim alapján tudom, hogy egy rossz programozási nyelv vagy fejlesztői környezet sokat elvehet a kész szoftver funkcionalitásából és esztétikusságából. Így mindenképp, olyan eszközöket akartam választani, melyek használatával a lehető legoptimálisabban tudjam a szimulációmtól elvárt funkciókat megvalósítani. Ezután jött a felismerés, hogy a szimulációm lényegében megfelel egy 2D-s játéknak, így a tervezést ennek mentén folytattam a továbbiakban.

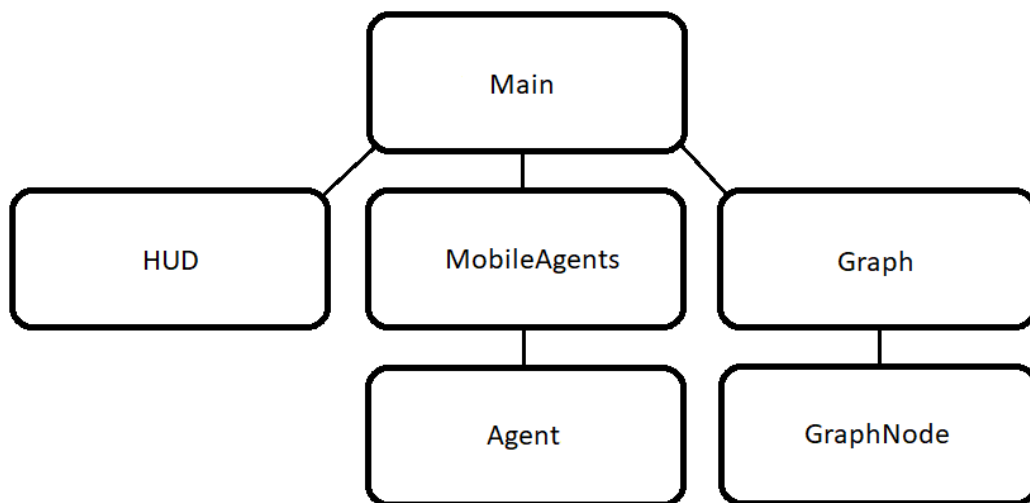
A megfelelő nyelv kiválasztásakor fontos szempont volt számomra, hogy egy könnyen használható, magas szintű, de mégis stabil, megbízható és gyorsan futó nyelvet válasszak, mely széleskörben támogatott a forgalomban lévő fejlesztői környezetek és operációs rendszerek, platformok által. Ezen elvárások alapján a választásom a `C#` nyelvre esett, ami egy jól optimalizált, magasszintű, cross-platform nyelv.

A megfelelő fejlesztési környezet után való keresésem során rátaláltam a Godot nevezetű játék motorra. A játék motor egy olyan szoftver környezet, amit azzal a céllal alkottak meg, hogy segítse a játékok fejlesztését, rendelkezik minden alapvető funkcióval, így a fejlesztőnek tényleg csak a saját elképzeléseinek megírására kell figyelnie. Nyilvánvalóan a motor rendszerét ismerni kell. Lényegében erre volt nekem

is szükségem, a rendszer magjának felépítése nehézkes és időigényes feladat, és ez akár a funkcionalitás és a szimulációk rovására is mehetett volna. Tehát a c# megtanulása után, a Godot alapvető funkcióinak megismerése következett. Szerencsére a játékmotornak tökéletesen érthető és egyértelmű dokumentációja és tanulást segítő segédanyaga van.

## 2.2 A keretrendszer felépítése

A keretrendszer felépítésében objektum orientált szemléletmódot alkalmaztam. Létrehoztam egy Agent (ügynök) osztályt az ügynökök reprezentálására és egy GraphNode (csomópont) osztályt a hálózat csomópontjainak modellezésére. A csomópontokat a Graph (gráf) osztály fogja össze egy elosztott hálózattá. Az ügynökök a szimuláció szempontjából könnyebb kezelhetőségének érdekében a MobileAgents (mozgó ügynökök) osztály alá vannak besorolva. A szimuláció szíve a Main osztály mely alá tartozik az MobileAgents és a Graph osztály, de ezeken felül még a menükért és irányításért felelős HUD osztály is ide tartozik.



2. ábra: A keretrendszer felépítése

### 2.2.1 Ügynök osztály

A mobil ügynökök felelősek a hálózat menedzsmentjéért, azért, hogy ezt a feladatukat el tudják látni több funkcióra is szükségük van. A működésüket három részre osztottam, mozgás, csomópont karbantartása és osztódás, illetve megsemmisülés.

Ezen három lépésen a szimuláció ideje alatt, ameddig része az adott ügynök a szimulációnak annyiszor megy végig, ahányszor ideje engedi.

A mozgás fázisban új célállomást keres magának és a kiválasztott helyre utazik, az utazás azonnal megtörténik, ezzel az egyszerűsítéssel, azért lehet élni, mivel feltehetően az ügynökök egy optikai kábelben közel fénysebességgel mozognak. A célpont kiválasztása játékelméleti paradigmát követ, vagyis az egyes lehetséges célállomások kiválasztásának esélyét súlyozza, majd ezután véletlenszerűen dönt. A lehetséges célállomások az ügynök által már korábban meglátogatott csomópontok és a jelenlegi tartózkodási helyével szomszédos hálózati eszközök. A súlyozás a csomópontok meghibásodási idejének becslésével történik, minél kisebb a következő meghibásodásig a becsült idő, annál valószínűbb a kiválasztás. Kooperatív működés esetén a célállomás kiválasztása előtt az ügynök megkérdezi a többi ügynököt, hogy a közeljövőben valaki tervez-e oda menni, amennyiben igen, akkor új csomópontot keres.

A karbantartás fázisban az éppen meglátogatott csomóponton alaphelyzetbe állítja a meghibásodás esélyét, és amennyiben már hibás volt újra működésbe hozza azt.

A harmadik fázisban először az osztódás lehetősége adott, ez akkor következik be, ha a feladat mennyiségétől befolyásoltan az ügynök véletlenszerűen úgy dönt, minél több a meghibásodott csomópont az ügynök becslései szerint annál nagyobb a hajlam a sokszorozódásra. Ezután, az osztódáshoz hasonlóan eldönti, hogy van-e szükség rá a továbbiakban vagy nincs, amennyiben nincs, akkor megsemmisül.

### **2.2.2 Mozgó ügynökök osztály**

A mozgó ügynökök osztály a szimuláció vezérlését segíti, ennek az osztálynak a segítségével lehet új ügynököt létrehozni a megfelelő paraméterekkel, vagyis, hogy milyen könnyen döntsön úgy, hogy osztódik vagy megsemmisül, illetve, hogy milyen módon, kooperatívban vagy kompetitívben üzemeljen.

### **2.2.3 Csomópont osztály**

A csomópontok, azaz egyes hálózati elemek nem valósítanak meg bonyolult működést, mert a szimuláció szempontjából ez nem lényeges. A feladatuk, hogy időről időre meghibásodjanak, ezzel munkát adva a mozgó ügynököknek, ezek az időközök nincsenek kőbe vésve, de belátható időn belül bekövetkezik a karbantartás szükségessége.

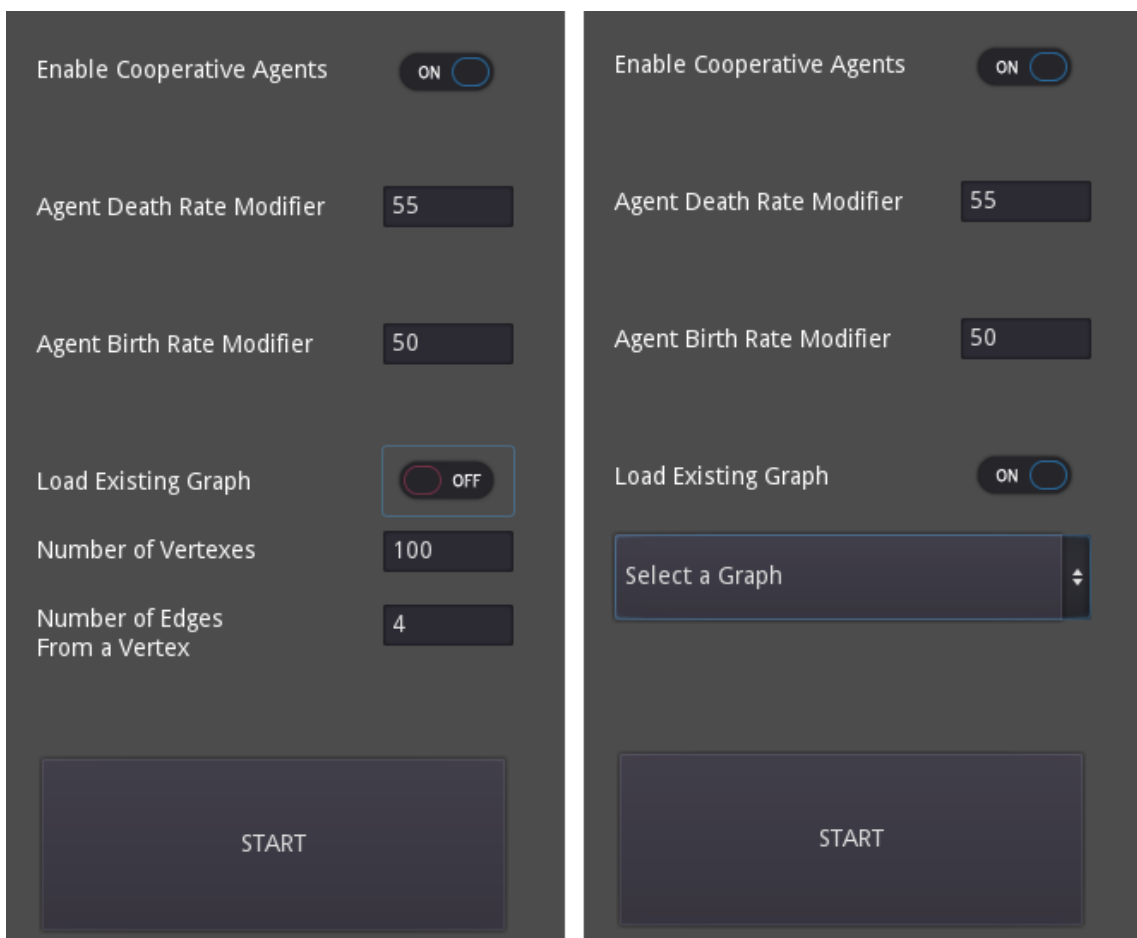
## 2.2.4 Gráf osztály

A gráf osztály segítségével lehet lementeni vagy betölteni egy megadott hálózatot szimuláló gráfot, vagy egy újat generálni a megadott csomópontok számával és maximális fokszámmal.

## 2.2.5 HUD osztály

A HUD osztály feladat, hogy a megfelelő vezérlést a felhasználó kezébe adja, ezzel biztosítva a szoftver könnyű és élvezhető használatát.

Megvalósít egy főmenüt, ahol a futtatni kívánt szimulációt állíthatjuk be. Itt lehetőségünk van beállítani, hogy a csomópontok kompetitíven vagy kooperatívan működjenek. A megsemmisülésre és az osztódásra való hajlandóságukat. Kiválaszthatjuk, hogy új gráfot szeretnénk generálni az általunk megválasztott paraméterekkel, vagy pedig be szeretnénk tölteni egy korábban használt gráfot.



3. ábra: főmenü

A főmenü mellett rendelkezésünk rá áll egy „escape” menü is, melyet a szimuláció futása közben érhetünk el. Itt lehetőség van a szimuláció megállítására, újraindítására, illetve a gráf elmentésére.

### **2.2.6 Main osztály**

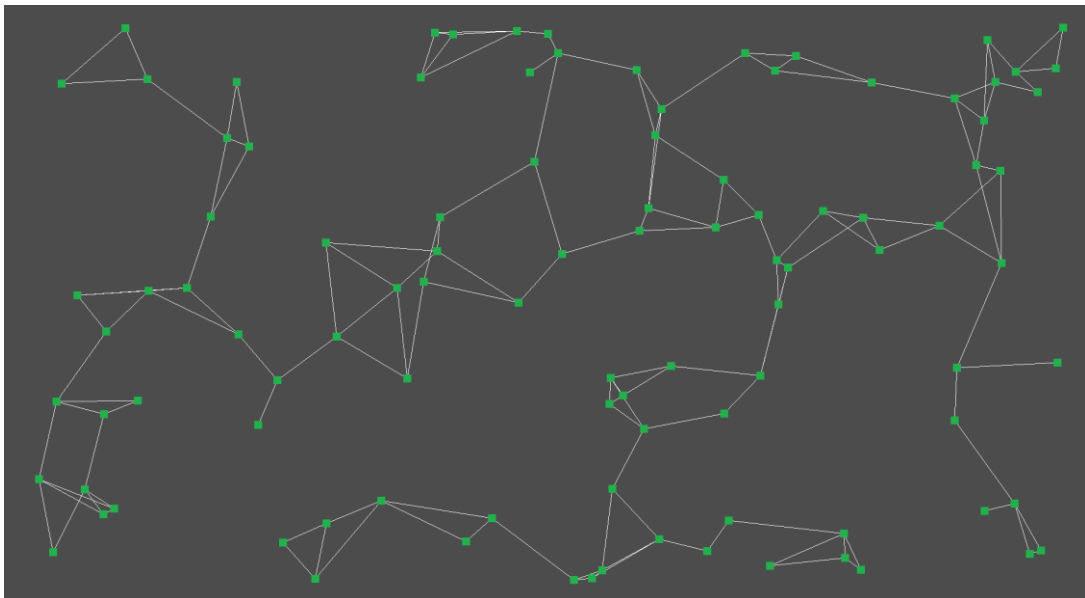
A main osztály a szimuláció szíve, ez irányít mindent és az adatgyűjtés is itt történik. A szimulációk időzítőkkal vannak szabályozva, tehát minden egyes csomópont az időzítő ciklusokkal egyre közelebb kerül a meghibásodáshoz, minden ügynök az időciklusokban hajtja végre a megfelelő feladatát. Ezenfelül, időről időre egy új ügynököt generál, arra az esetre, ha véletlenül kihaltak volna.

## 3 Szimulációs eredmények

Ebben a fejezetben mutatom be az általam generált hálózatokon folytatott méréseimet és elemzem ki azokat. Minden szimulációt tíz percig futtattam így 600 adatból készült minden egyes grafikonom.

### 3.1 Első hálózat

Legelőször egy 100 csomópontból álló gráfot vizsgáltam, ahol egy csomópont maximum 4 éllel rendelkezhet. Ez egy kisebb méretű, de jó, illetve sűrű összeköttetésekkel rendelkező hálózatot hivatott szimulálni.



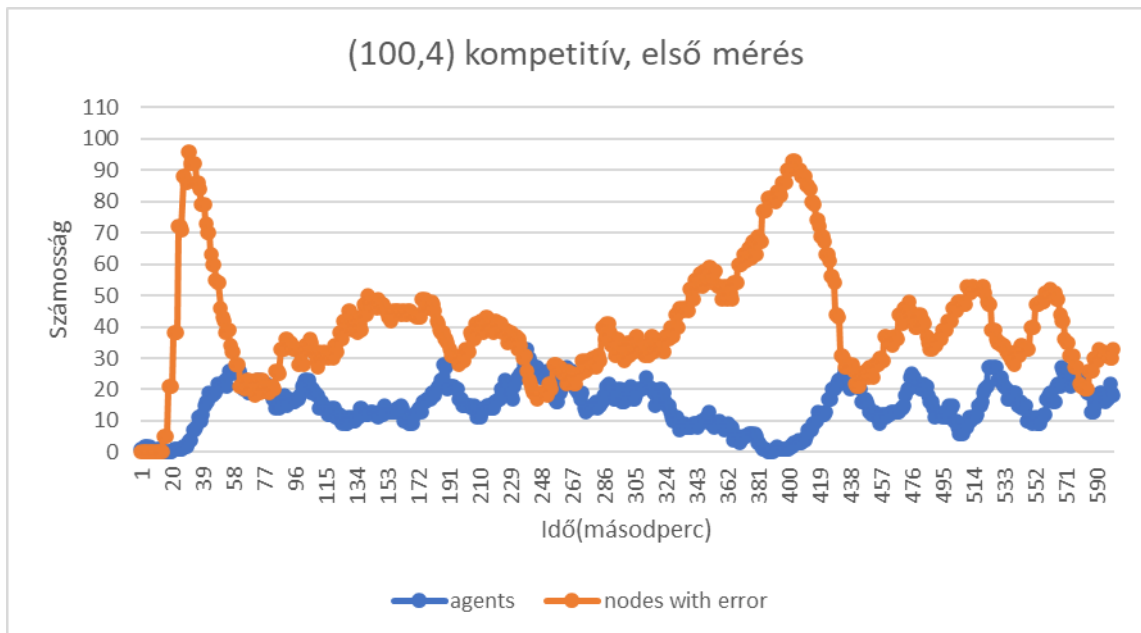
4. ábra: 100 csomópont, maximum 4 él

### 3.1.1 Első mérés

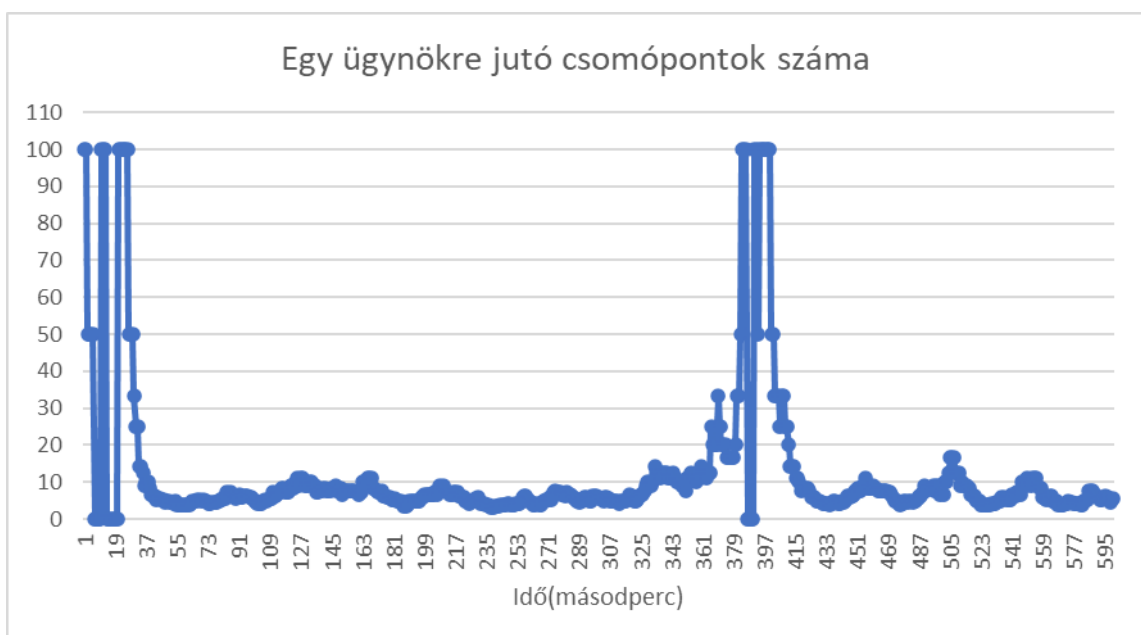
Az első mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 55% míg az osztódásnak 30%.

#### 3.1.1.1 Kompetitív

Az ügynökök kompetitív módban működtek.



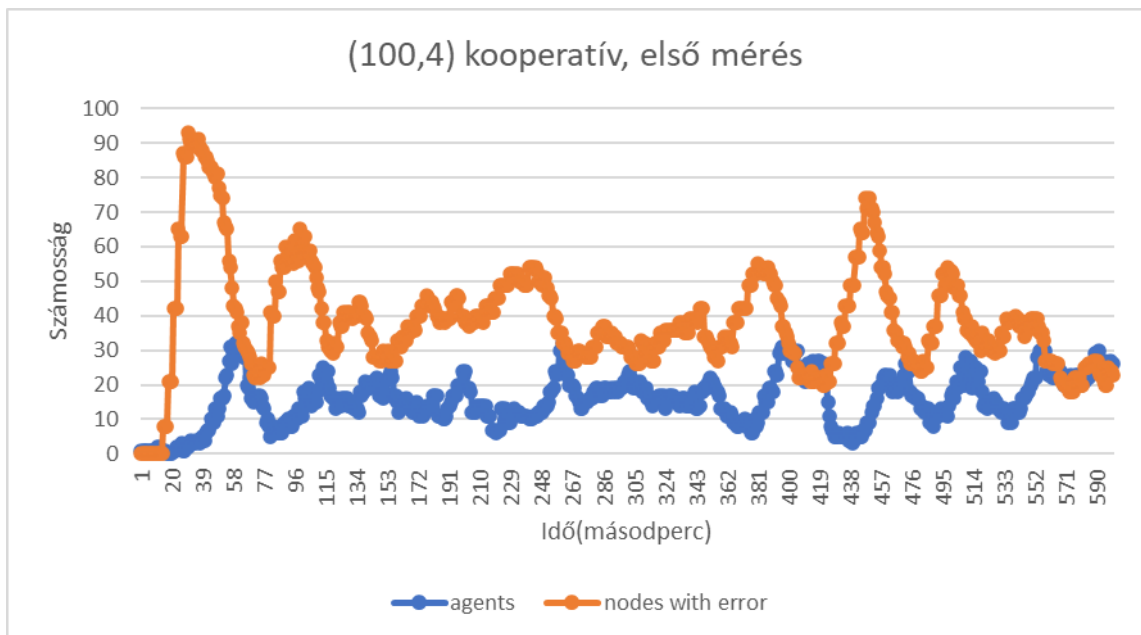
5. ábra: (100,4) kompetitív, első mérés



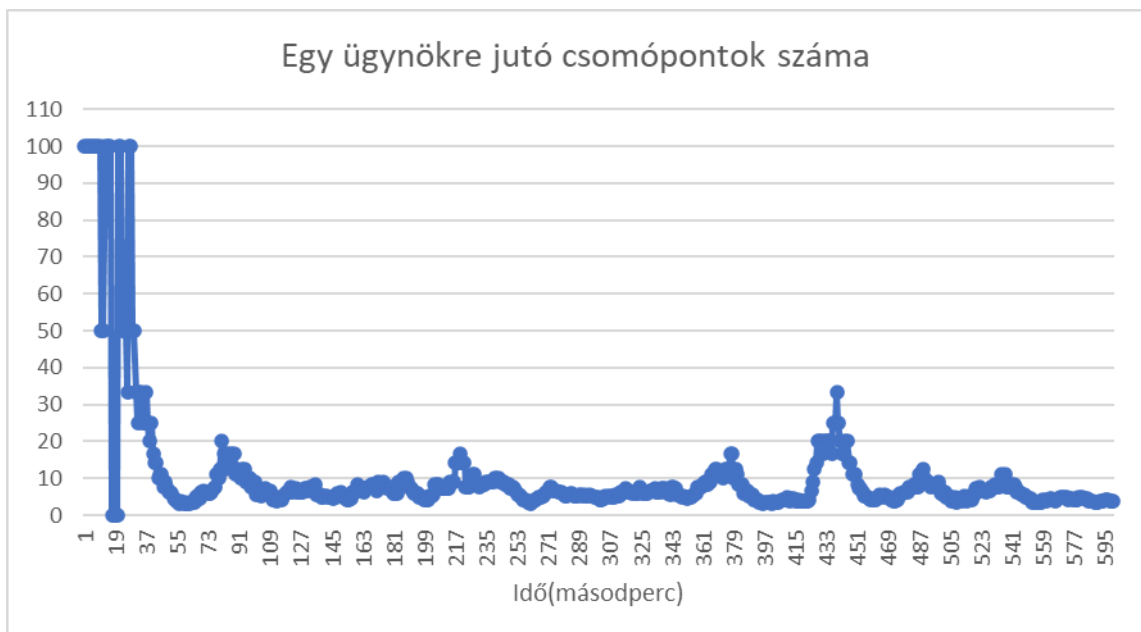
6. ábra: (100,4) kompetitív, első mérés, egy ügynökre jutó csomópontok száma

### 3.1.1.2 Kooperatív

Az ügynökök kooperatív módon működtek.



7. ábra: (100,4) kooperatív, első mérés



8. ábra: (100,4) kooperatív, első mérés, egy ügynökre jutó csomópontok száma

### 3.1.1.3 Kiértékelés

A kompetitív és kooperatív esetben is megfigyelhető a szimuláció elején egy tranzien্স jelenség, mely annak köszönhető, hogy kezdetben a csomópontok időben elég



kis eltéréssel hibásodnak meg, de még nincs elég ügynök a hálózatban, hogy ezt megfelelő sebességgel lereagálják.

Mindkét esetben az egy ügynökre jutó csomópontok száma, néha nulla értéket vesz fel, ez egy hibás érték, ami akkor jön elő, amikor éppen nincs egy ügynök sem a hálózatban. Ahogy korábban említettem ez ellen az állapot ellen egy periodikusan hozzáadott ügynökkel védekezünk.

Kooperatív és kompetitív esetben is nagyon szépen látszik, hogy ahol az ügynökök számának minimuma van ott a meghibásodott csomópontoknak maximuma, illetve ahogy az ügynökök számának maximuma van ott a csomópontoknak minimuma. Vagyis az ügynökök száma leköveti a meghibásodott csomópontok számát.

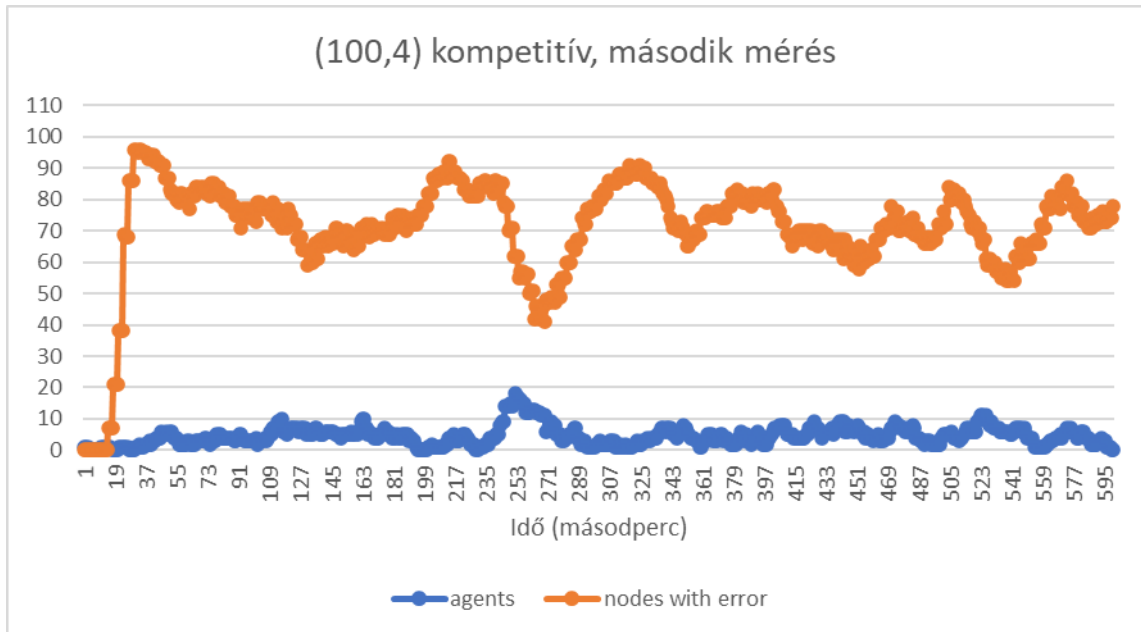
Ami fontos, és az elméleti bevezető alapján meglepetést okozhat, hogy a kooperatív módban működő ügynökök sokkal egyenletesebb hálózat menedzsmentet tudnak biztosítani kisebb csomópont meghibásodási rátával. Kompetitív esetben 41,083 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 38,66, ez a teljes szimuláció során 1902-vel kevesebb meghibásodott csomópontot jelent. Az eredmény azért lehet eltérő az emberekkel végezett nyúl vadászatos játéktól, mert az embereknél a kommunikáció jóval több ideig tart, már csak ha azt az egy ténytet vesszük figyelembe, hogy hang útján kommunikálunk, míg az ügynökök esetében a kommunikációs csatornát egy optikai kábel képezi, vagyis az információ közel fénysebességgel és nem pedig hangsebességgel halad.

### 3.1.2 Második mérés

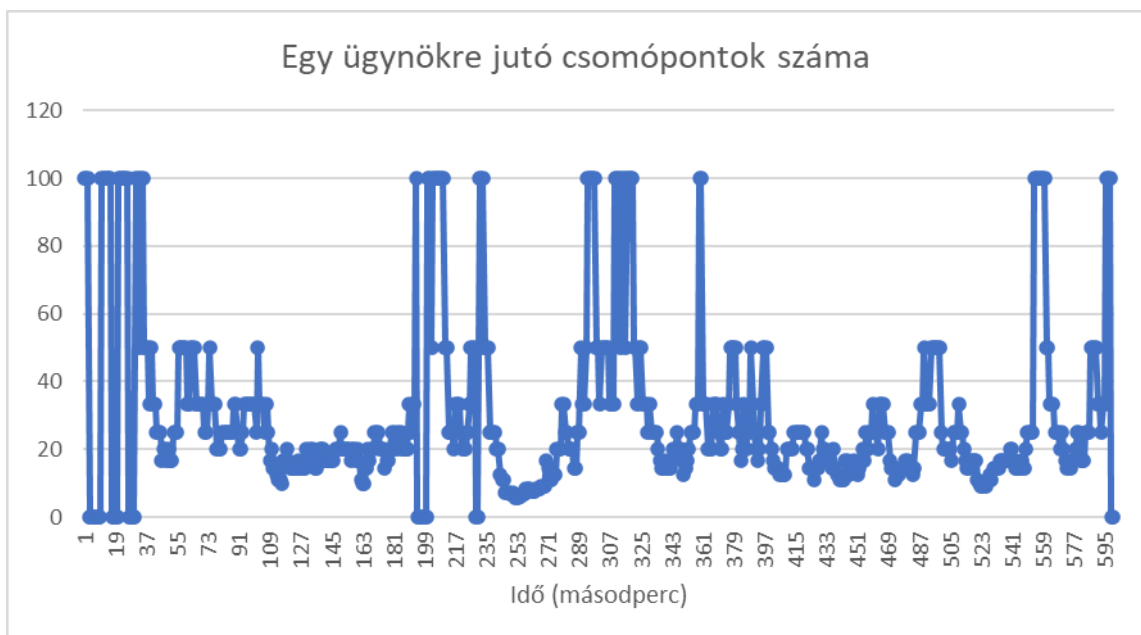
A második mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 80% míg az osztódásnak 30%.

#### 3.1.2.1 Kompetitív

Az ügynökök kompetitív módon működtek.



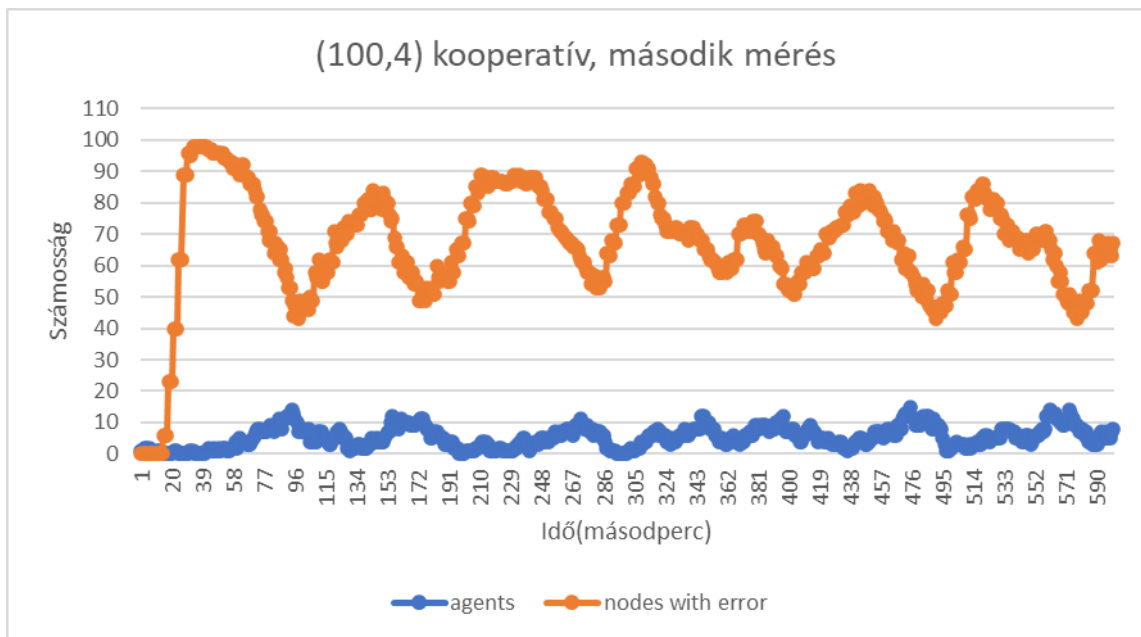
9. ábra: (100,4) kompetitív, második mérés



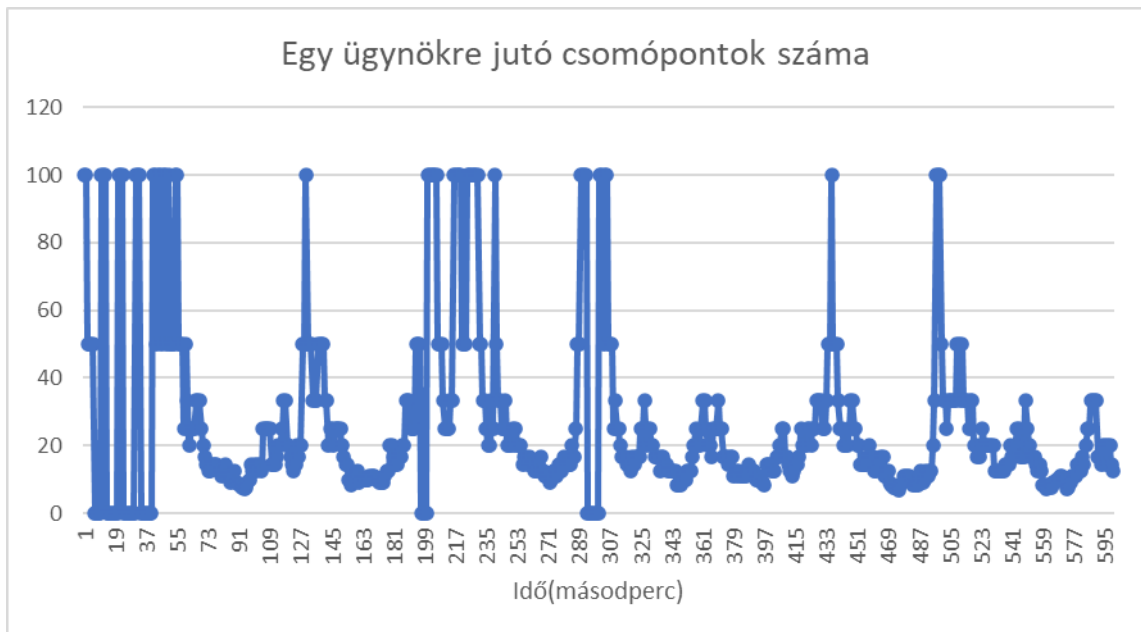
10. ábra: (100,4) kompetitív, második mérés, egy ügynökre jutó csomópontok száma

### 3.1.2.2 Kooperatív

Az ügynökök kooperatív módon működtek.



11. ábra: (100,4) kooperatív, második mérés



12. ábra: (100,4) kooperatív, második mérés, egy ügynökre jutó csomópontok száma

### 3.1.2.3 Kiértékelés

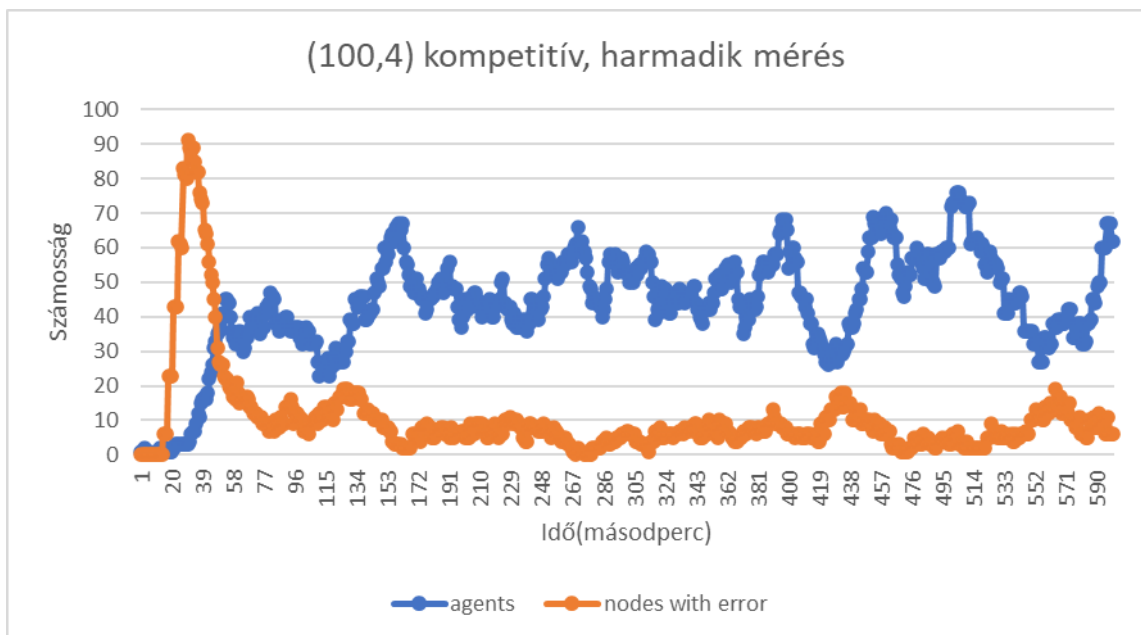
Ebben a mérésben extrém könnyű volt egy ügynök számára a megsemmisülést választani. A mérésből nagyon szépen látszik, hogy minél kevesebb ügynököm van a hálózatban. annál fontosabb, hogy kifinomultabb stratégiát kövessenek. Kompetitív esetben 71,251 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 67,377, ez a teljes szimuláció során 2324,4-gyel kevesebb meghibásodott csomópontot jelent.

### 3.1.3 Harmadik mérés

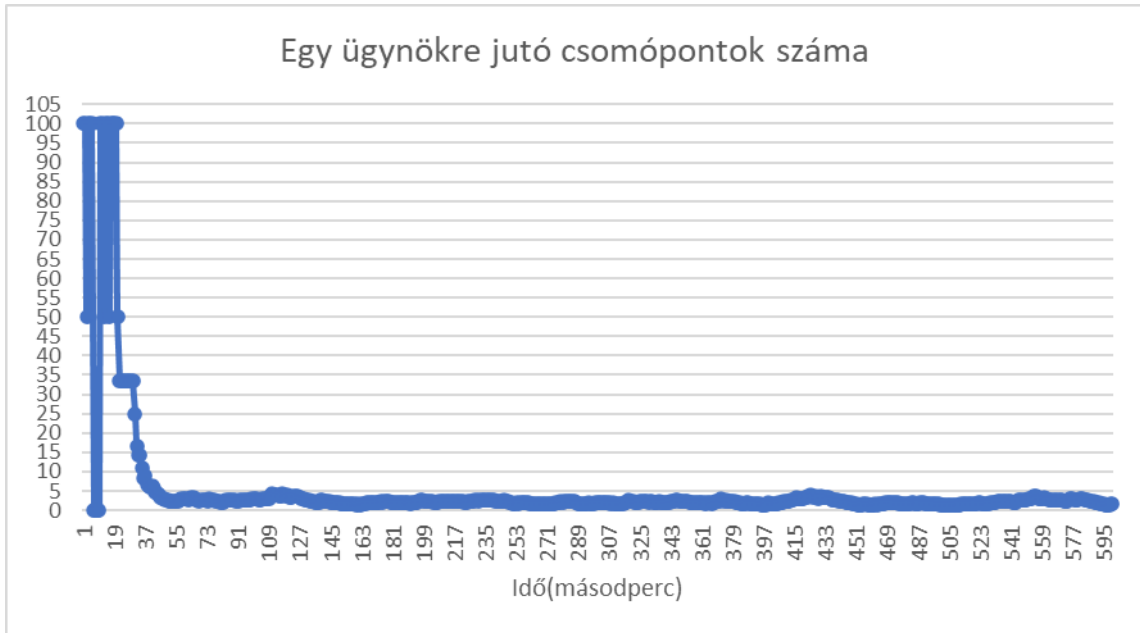
A második mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 55% míg az osztódásnak 50%.

#### 3.1.3.1 Kompetitív

Az ügynökök kompetitív módban működtek.



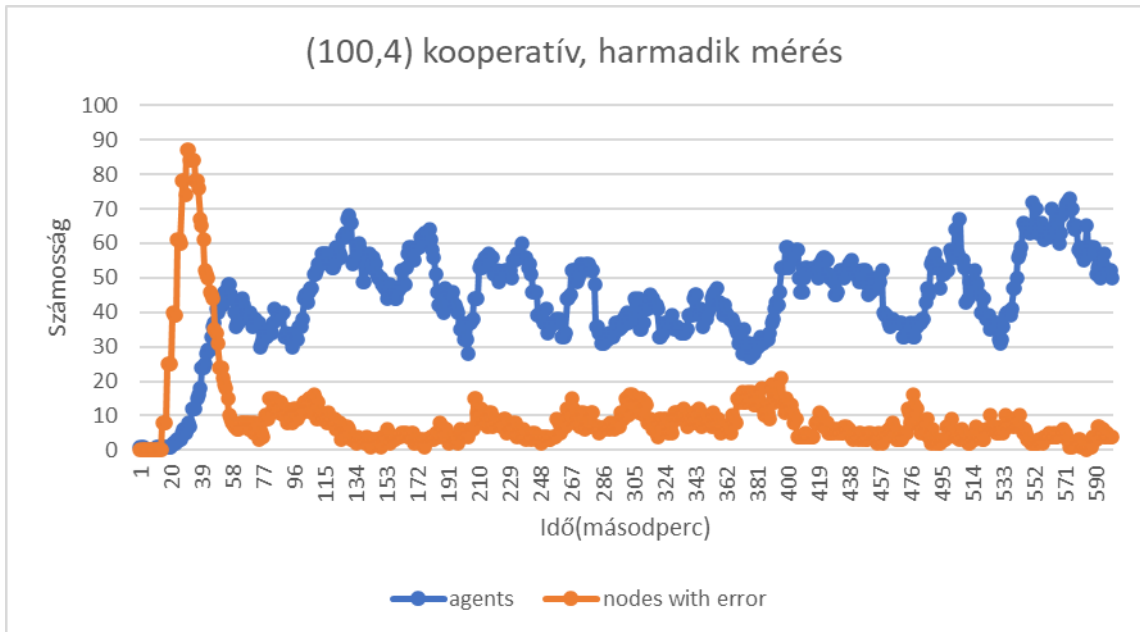
13. ábra: (100,4) kompetitív, harmadik mérés



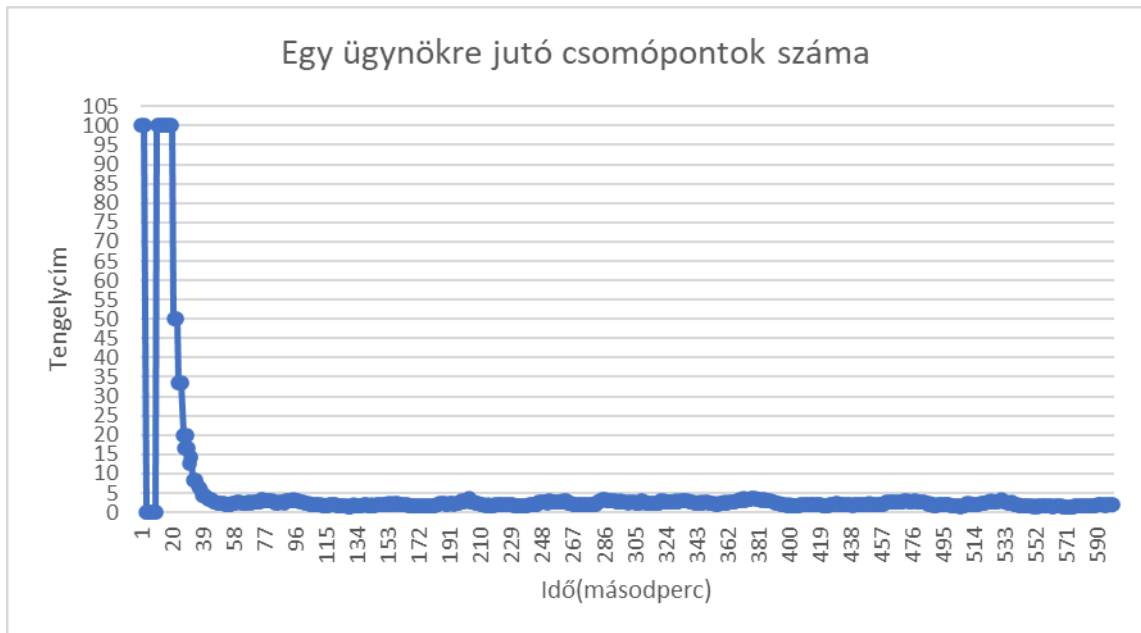
14. ábra: (100,4) kompetitív, harmadik mérés, egy ügynökre jutó csomópontok száma

### 3.1.3.2 Kooperatív

Az ügynökök kooperatív módban működtek



15. ábra: (100,4) kooperatív, harmadik mérés



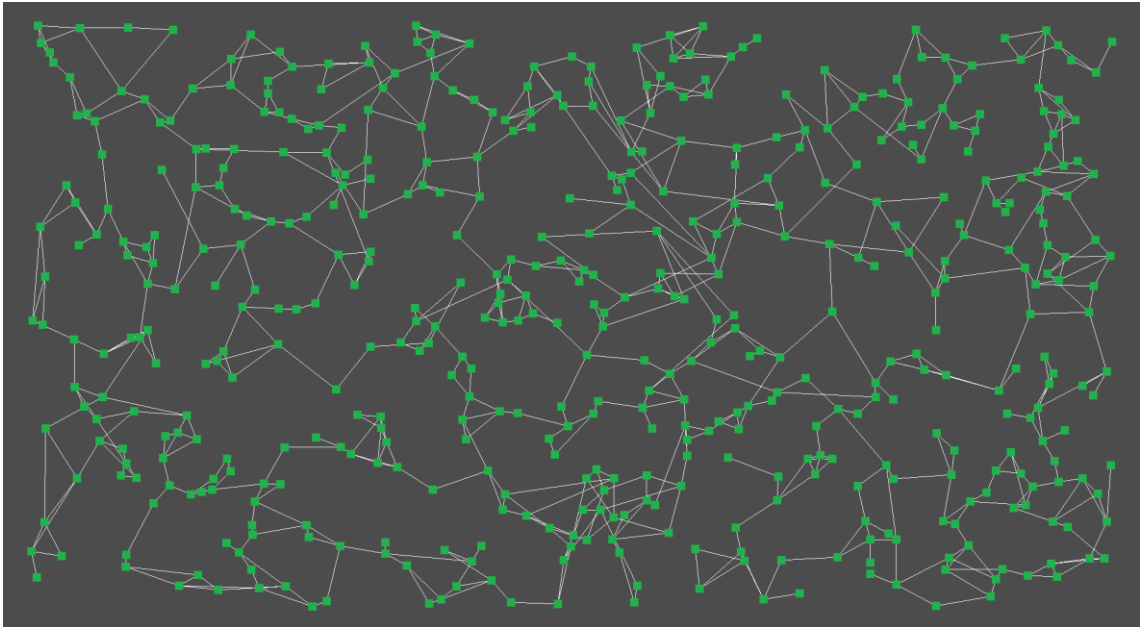
16. ábra: (100,4) kooperatív, harmadik mérés, egy ügynökre jutó csomópontok száma

### 3.1.3.3 Kiértékelés

A második méréssel ellentétben itt extrém könnyű volt az ügynökök számára a sokszorozódás. Ebben az esetben az a következtetés vonható le, hogy egy bizonyos számú ügynökpopuláció elérése után, szinte mindegy melyik menedzsment stratégiát választjuk. Kompetitív esetben 10,39 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 9,668, ez a teljes szimuláció során 434,2-vel kevesebb meghibásodott csomópontot jelent.

## 3.2 Második hálózat

Második hálózati vizsgálatom során egy 500 csomópontból álló gráfot vizsgáltam, ahol egy csomópont maximum 4 éllel rendelkezhet. Ez egy nagyobb méretű, de relatíve rosszabb összeköttetésekkel rendelkező hálózatot hivatott szimulálni.



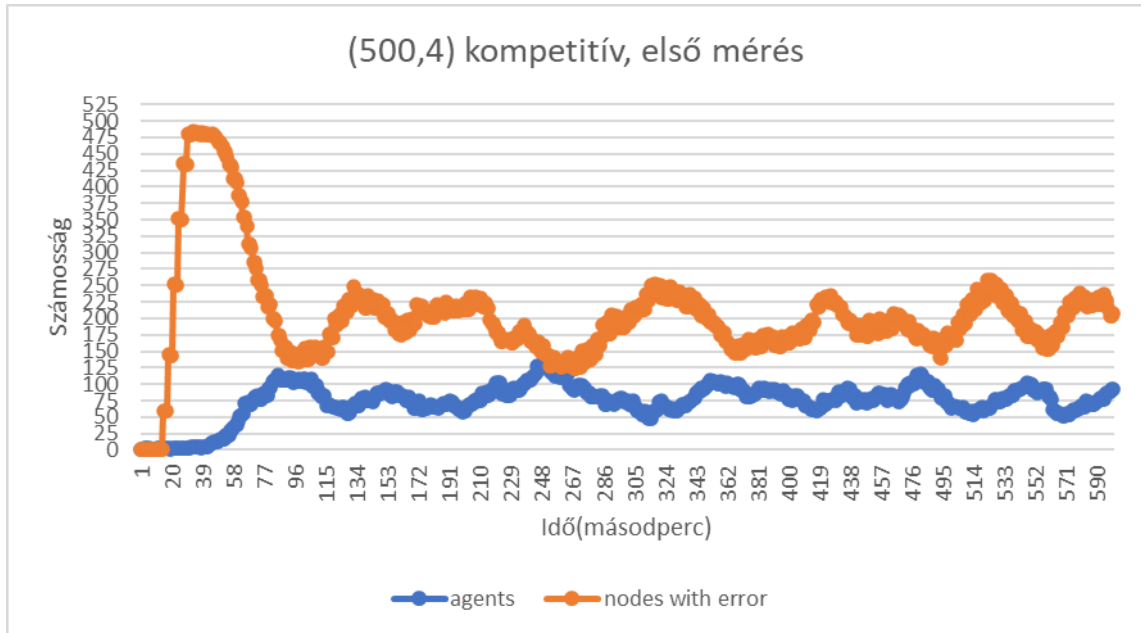
17. ábra: 500 csomópont, maximum 4 él

### 3.2.1 Első mérés

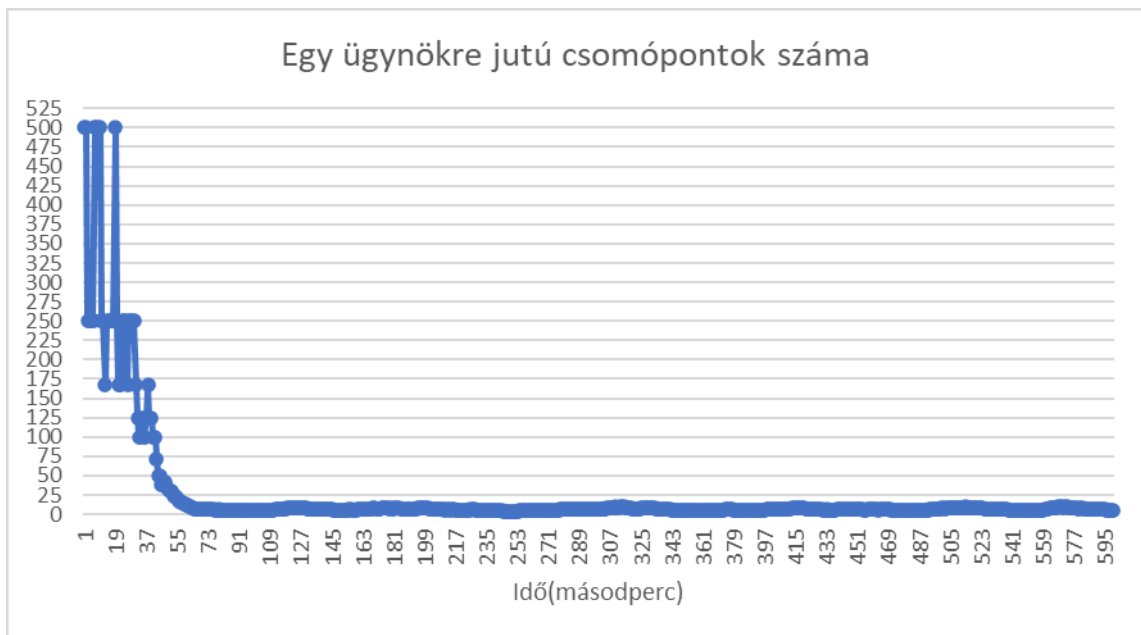
Az első mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 55% míg az osztódásnak 30%.

### 3.2.1.1 Kompetitív

Az ügynökök kompetitív módban működtek.



18. ábra: (500,4) kompetitív, első mérés

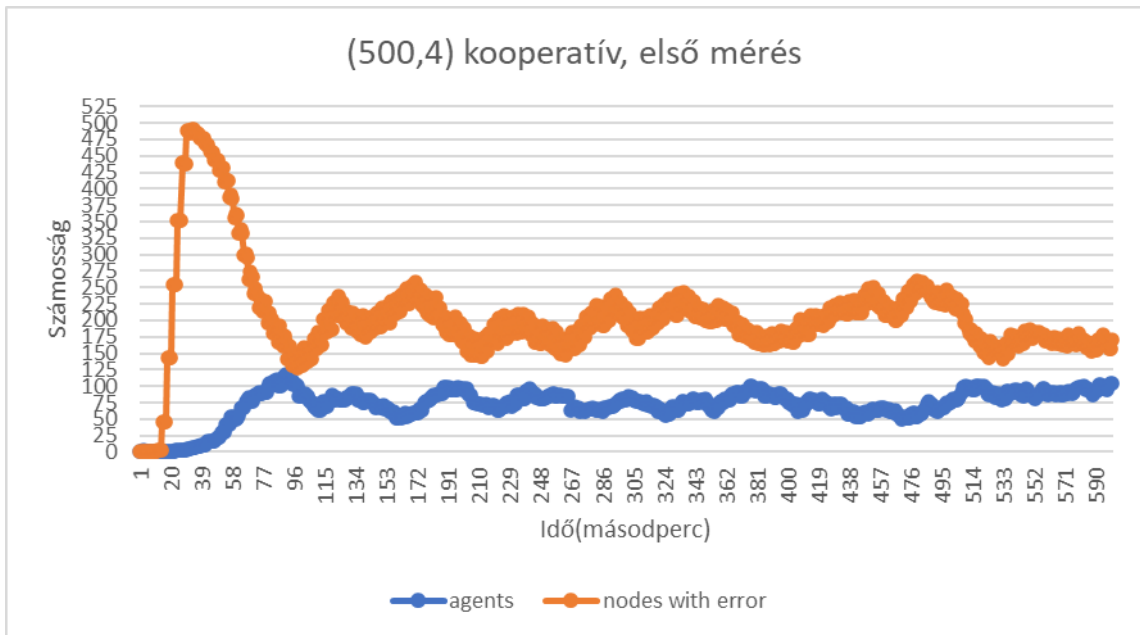


19. ábra: (500,4) kompetitív, első mérés, egy ügynökre jutó csomópontok száma

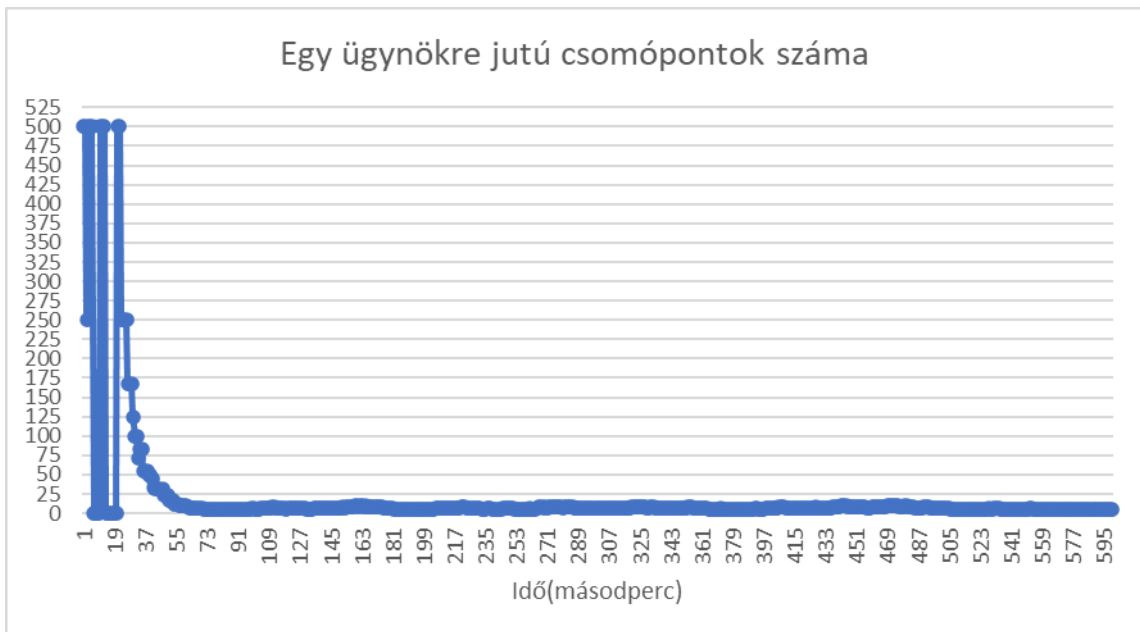


### 3.2.1.2 Kooperatív

Az ügynökök kooperatív módban működtek.



20. ábra: (500,4) kooperatív, első mérés



21. ábra: (500,4) kooperatív, első mérés, egy ügynökre jutó csomópontok száma

### 3.2.1.3 Kiértékelés

Az első mérés eredményeiből az látszik, hogy egy nagyobb, relatíve rosszabb összeköttetésekkel rendelkező hálózat esetén a kooperatív stratégia nem nyújt semmiféle előnyt a kompetitívvel szemben. Ennek a jelenségnek az a valószínű oka,

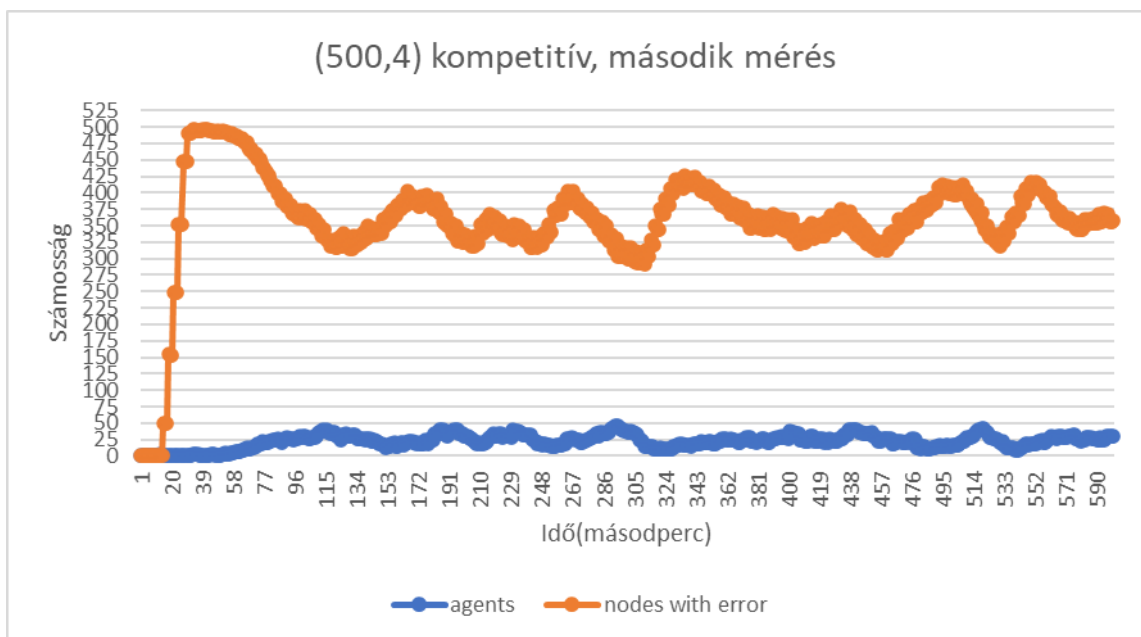
hogy amikor egy ügynök lekérdezi, a többi ügynöktől, hogy akar-e a célpontjával választott csomóponttra menni a közeljövőben, akkor arra valószínűleg nem lesz a válasz, így az üzenetváltás felesleges idővesztés volt. Ez jelenleg úgy mutatkozik meg, az üzenetváltás sebessége miatt, hogy majdnem azonos minőségű menedzsmentet valósít meg a két stratégia. Kompetitív esetben 204,84 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 204,937 ez a teljes szimuláció során 58,2-vel több meghibásodott csomópontot jelent.

### 3.2.2 Második mérés

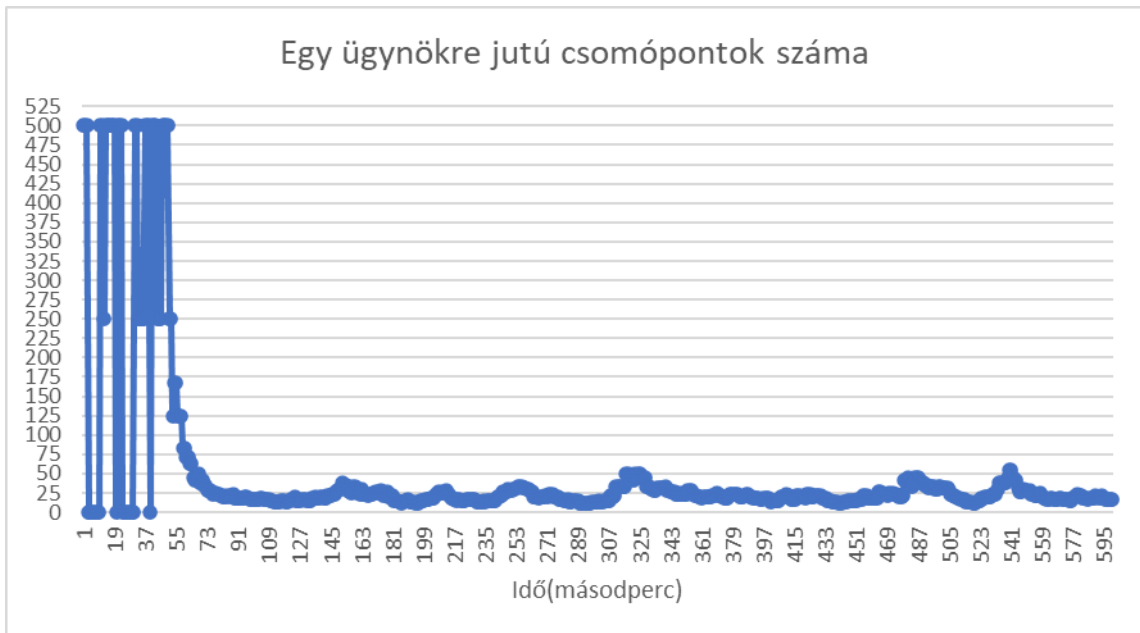
A második mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 80% míg az osztódásnak 30%.

#### 3.2.2.1 Kompetitív

Az ügynökök kompetitív módban működtek.



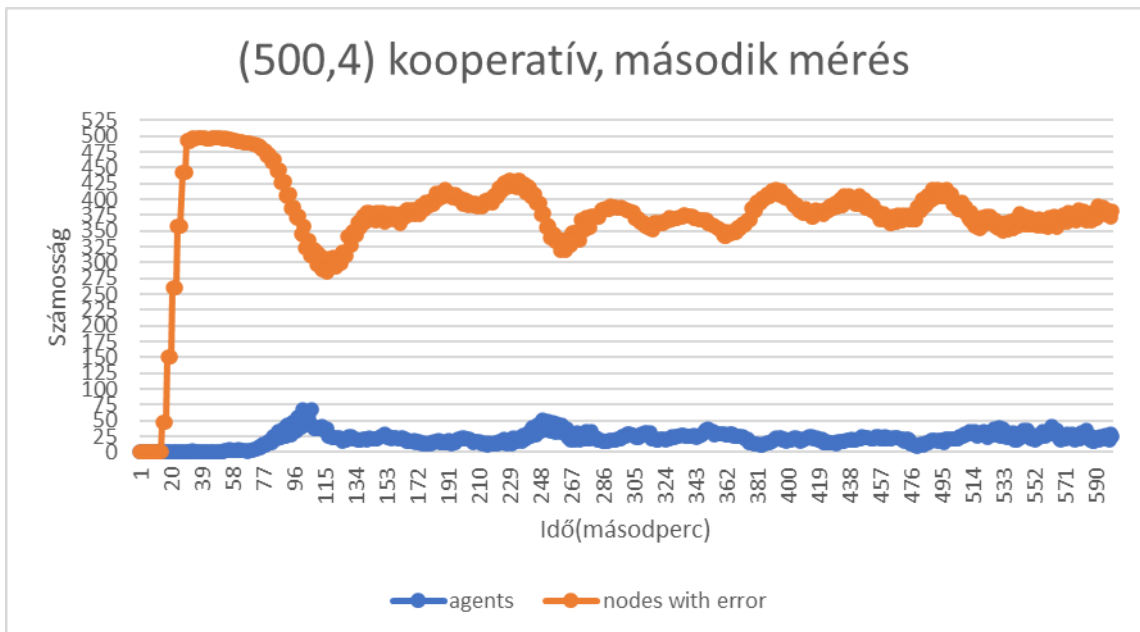
22. ábra: (500,4) kompetitív, második mérés



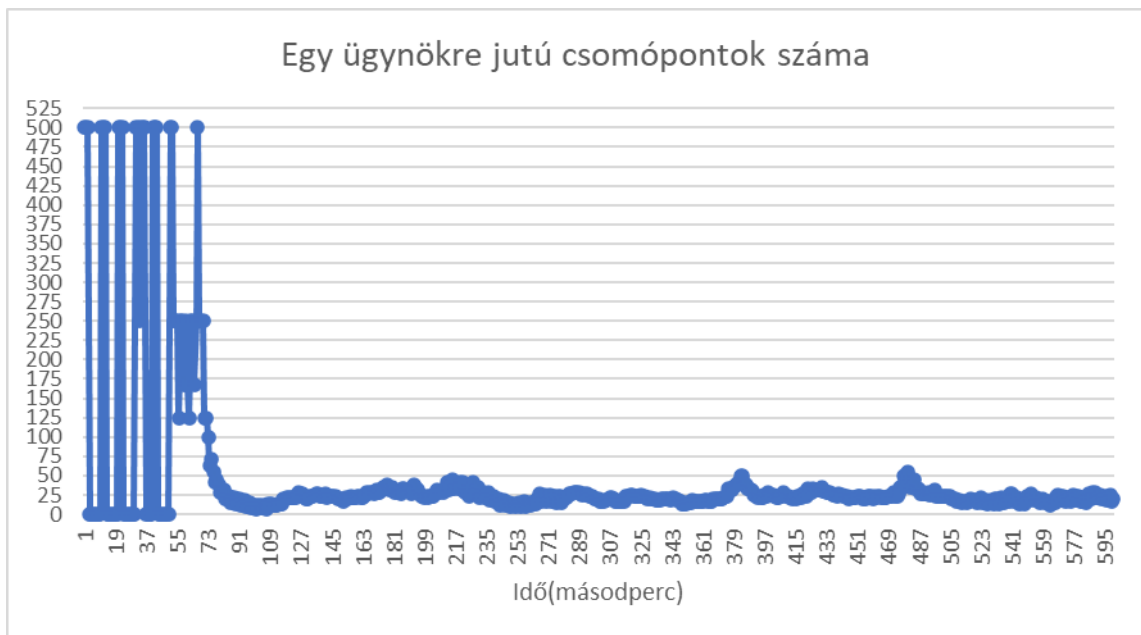
23. ábra: (500,4) kompetitív, második mérés, egy ügynökre jutó csomópontok száma

### 3.2.2.2 Kooperatív

Az ügynökök kooperatív módban működtek.



24. ábra: (500,4) kooperatív, második mérés



25. ábra: (500,4) kooperatív, második mérés, egy ügynökre jutó csomópontok száma

### 3.2.2.3 Kiértékelés

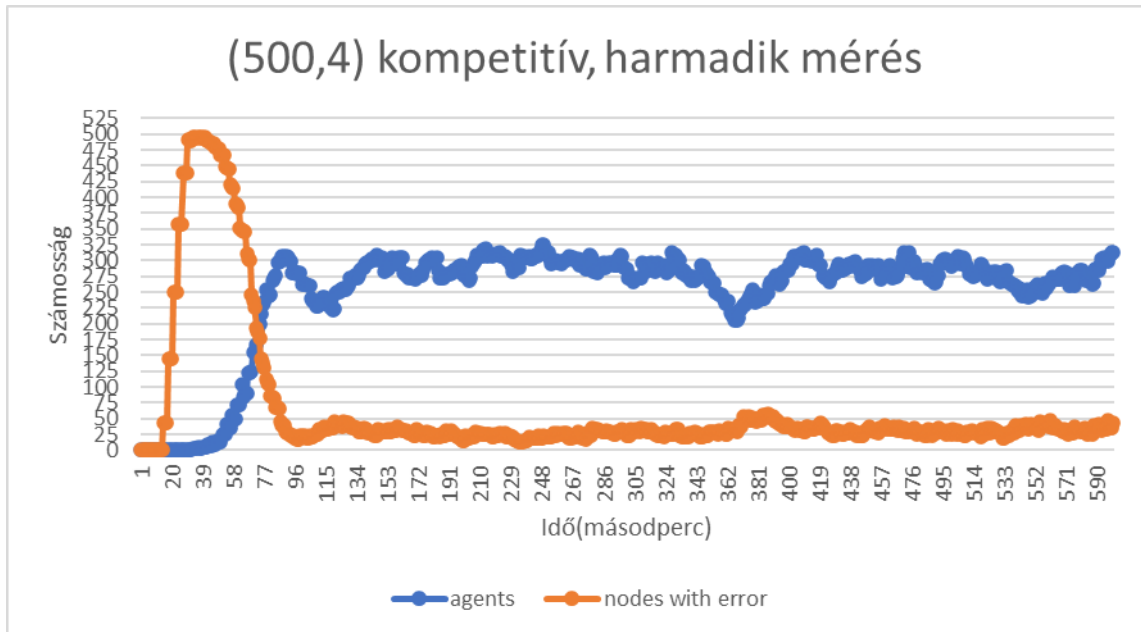
Kevés ügynök esetén nagyobb hálózatban felerősödik az üzenetváltás fölöslegessége, itt már ténylegesen a menedzsment rovására megy. Kompetitív esetben 357,36 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 374,28 ez a teljes szimuláció során 10152-vel több meghibásodott csomópontot jelent.

### 3.2.3 Harmadik mérés

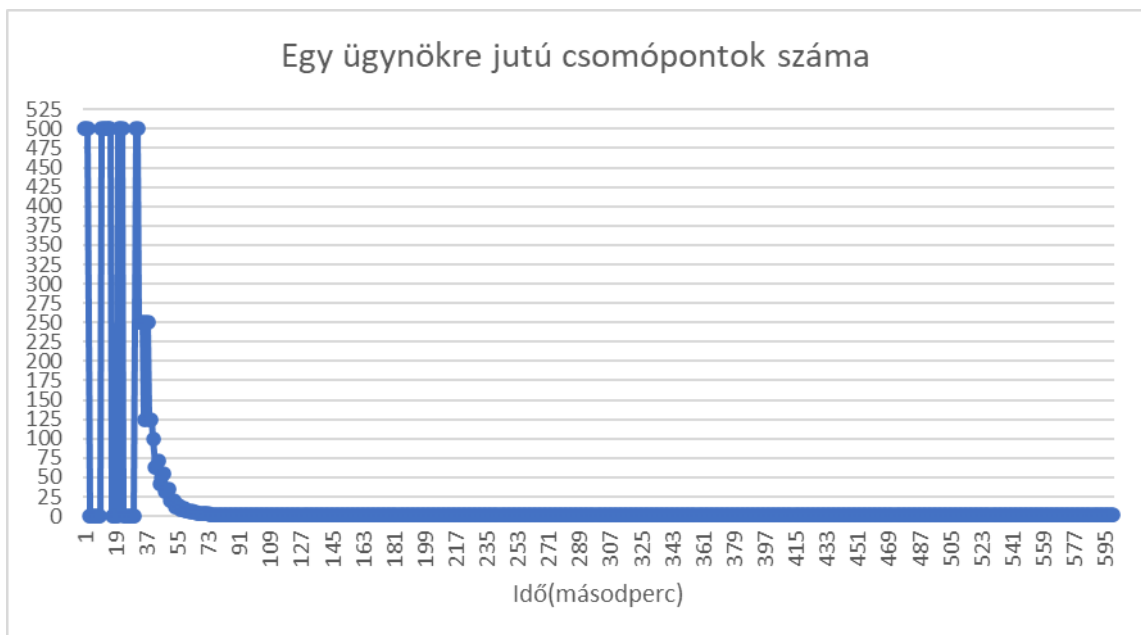
A második mérésnél az alap esélye - módosítás nélkül - egy ügynök megsemmisülésének 55% míg az osztódásnak 50%.

#### 3.2.3.1 Kompetitív

Az ügynökök kompetitív módon működtek.



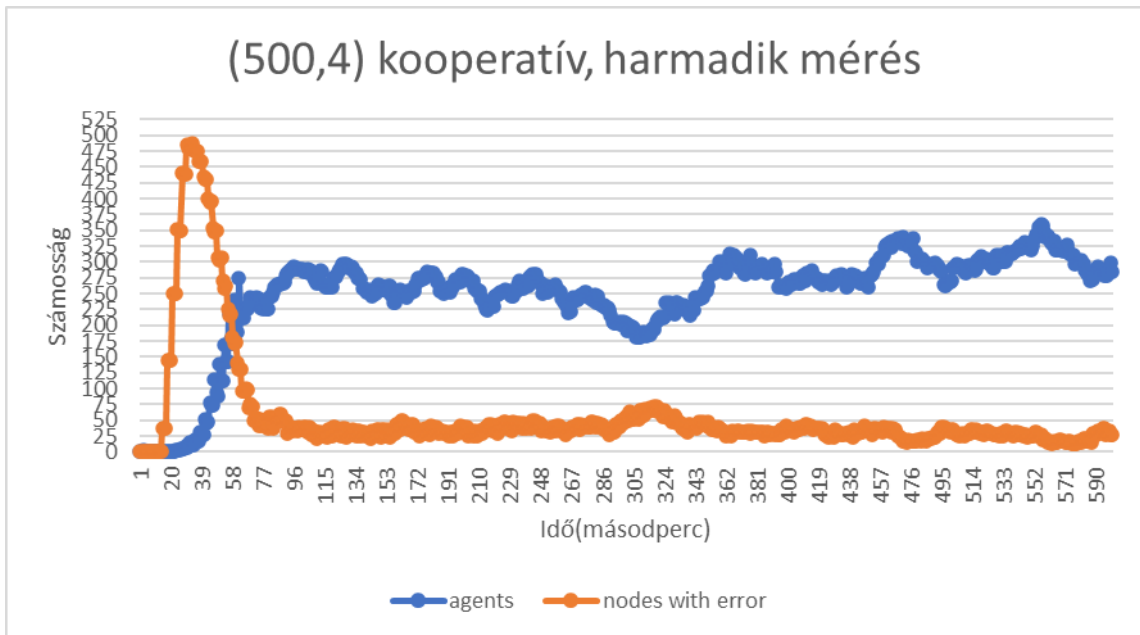
26. ábra: (500,4) kompetitív, harmadik mérés



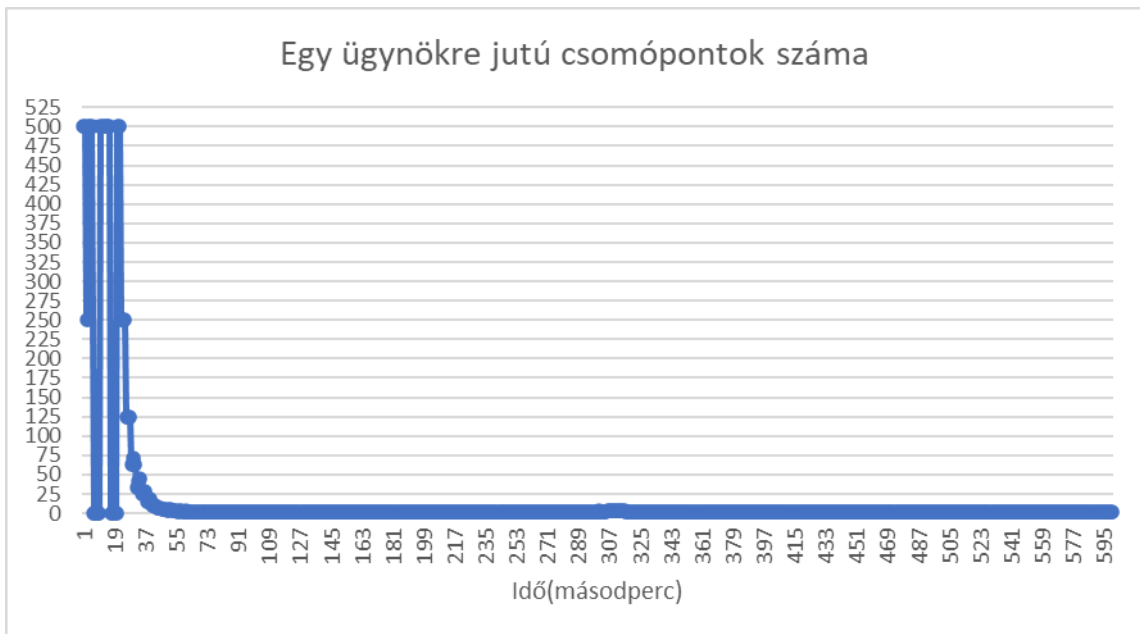
27. ábra: (500,4) kompetitív, harmadik mérés, egy ügynökre jutó csomópontok száma

### 3.2.3.2 Kooperatív

Az ügynökök kooperatív módon működtek



28. ábra: (500,4) kooperatív, harmadik mérés



29. ábra: (500,4) kooperatív, harmadik mérés, egy ügynökre jutó csomópontok száma

### 3.2.3.3 Kiértékelés

Az extrém könnyű sokszorozódási eset eredménye lényegesen változott a kisebb, jobb összeköttetéssel rendelkező hálózathoz képest, ugyanis a nagyobb hálózatban jóval nagyobb ügynök populáció érhető el, így az üzenetváltás vesztesége nagyobb lesz. Kompetitív esetben 64,837 az átlag meghibásodott csomópontok száma a mérés időtartamához mérten, míg kooperatív esetben ez 57,21 ez a teljes szimuláció során 4576,2-vel kevesebb meghibásodott csomópontot jelent.

## 3.3 Összefoglalás

A mérési eredményekből az a következtetés szűrhető le, hogy a kisebb vagy jobb, illetve több összeköttetéssel rendelkező hálózatok esetén, amikor a mozgó ügynököknek van több lehetősége találkozni a csomópontokon, akkor a kooperatív stratégia hoz nagyobb sikert. Ellenben rosszabb, kevesebb összeköttetés esetén egyértelműen a kompetitív stratégia bizonyult előnyösebbnek.

A kompetitív stratégia előnye növekszik a hálózat méretével, népesebb, könnyebben osztódó ügynök populáció esetén. Ezenfelül a ritkább ügynök populáció esetén is csak a kisebb, jobban átjárható hálózat esetén van előnyben a kooperatív stratégia.

A szimulációs rendszer jövőbeni fejlesztésére több tervem is van, lehetne olyan mérési lehetőséget implementálni, ahol hozzáadunk vagy elveszünk csomópontokat, így megmásítva a hálózat méretét futás közben, illetve érdekes lehet az is, ha a stratégiát van lehetőségünk megváltoztatni a szimuláció alatt. Egy felhasználói szempontból hasznos fejlesztés lenne az, ha egyszerre több egymás után futó szimulációt tudnánk sorba állítani, így az egyik végeztével nem kéne manuálisan elindítani a következőt.

## Irodalomjegyzék

- [1] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlam-tac. Internet of things: Vision, applications and research challenges. *Ad Hoc Networks*, 10(7):1497-1516, 2012.
- [2] Jongtae Song and Soon Seok Lee: Comparison of NGN QoS control Models - distributed or centralized. *Proceedings of APCC2008 2008*
- [3] Andrzej Bieszczad, Bernard Pagurek, and Tony White: Mobile agents for network management. *IEEE Communications Surveys and Tutorials*, 1:2-9, 1998.
- [4] Danny B. Lange and Mitsuru Oshima: Seven good reasons for mobile agents. *Communications of the ACM*, 42:88-89, 1999. Steven J. Brams, "Game theory", *Encyclopedia Britannica*, 2020. augusztus 6.
- [5] Steven J. Brams, "Game theory", *Encyclopedia Britannica*, 2020. augusztus 6.
- [6] Simonovits András: Bevezetés a játékelméletbe, MTA, 2007. május 6.
- [7] Christian Julmi: *Introduction to Game Theory*, ISBN 978-87-403-0280-6, 2012
- [8] Mérő László: *Mindenki másképp egyforma*, ISBN 978-963-8453-19-8, 1996.
- [9] Kózczy Á. László: A Neumann-féle játékelmélet, *Közgazdasági Szemle*, LIII, évf., 2006. január (31-45.o.)
- [10] K. Leyton-Brown és Y. Shoham, *Essentials of Game Theory*, Morgan & Claypool, 2008.
- [11] A. MacKenzie és S. Wicker, „Game theory and the design of self-configuring, adaptive wireless networks,” *IEEE Communications Magazine*, 39. kötet, 11. szám, pp. 126 - 131, 2001.
- [12] Theodore L. Turocy, Bernhard von Stengel: *Game Theory*, CDAM Research Report LSE-CDAM-2001-09, 2001. október 8.