

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MEDZIAGENTOVÁ KOMUNIKÁCIA
AKO PROSTRIEDOK DOSAHOVANIA CIEĽOV

Diplomová práca

2014

Peter Vlk

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

MEDZIAGENTOVÁ KOMUNIKÁCIA
AKO PROSTRIEDOK DOSAHOVANIA CIEĽOV

Diplomová práca

Študijný program: kognitívna veda

Študijný odbor: 9.2.11 kognitívna veda

Školiace pracovisko: FMFI.KAI – Katedra aplikovanej informatiky

Vedúci práce: RNDr. Martin Takáč, PhD.

Bratislava, 2014

Peter Vlk

VLOŽ ZADANIE ZÁVEREČNEJ PRÁCE

Pod'akovanie

Rád by som pod'akoval svojmu školiteľovi RNDr. Martinovi Takáčovi, PhD. za jeho rady, ústretovosť a rýchle odpovede. Ďalej chcem pod'akovať Bc. Lenke Burianovej za pomoc s formálnou stránkou tejto práce. V neposlednom rade chcem pod'akovať Bc. Ivanovi Kristeľovi za názory a diskusie ohľadom tejto práce a programovania všeobecne.

ABSTRAKT

V tejto práci porovnávame úspešnosť dvoch skupín kognitívnych agentov. Jedna skupina agentov môže navzájom komunikovať a druhá nemôže. Testujeme našu hypotézu, že agenti z komunikujúcej skupiny budú úspešnejší. Agenti sú vložený do virtuálneho sveta. Každý má k dispozícii senzor, ktorým dokáže pozorovať svoje okolie a sadu akcií, ktoré môže vykonávať. Úlohou agentov je maximalizovať odmenu získavanú z prostredia. Na začiatku agenti nepoznajú stratégiu, ktorou sa majú riadiť. Túto sa musia naučiť. Učiace mechanizmy agentov sú implementované vo forme neurónového aktéra-kritika.

Kľúčové slová: agent, neurónová sieť, aktér-kritik, simulácia, komunikácia

ABSTRACT

In this paper we are comparing success rate of two groups of cognitive agents. Agents in the first group are capable of communication, agents in the second group are not capable of communication. We are testing out hypothesis that the group of agents capable of communication will show better results. Agents are placed into virtual environment. Each agent has a sensor which is allowing him to observe its surroundings and set of actions that he can carry out. Agents' goal is to maximize reward obtained from the environment. At the beginning of simulation agents have no knowledge about strategy that which they should follow. They must learn it. Agents' learning mechanisms are implemented in form of neural actor-critic.

Key words: agent, neural network, actor-critic, simulation, communication

OBSAH

Úvod	10
1 Existujúce prístupy	12
2 Učenie agentov	13
2.1 Neurónové siete	13
2.2 Učenie posilňovaním	14
2.3 Explorácia	16
2.4 Učenie s časovým oneskorením.....	17
2.5 Aktér-kritik	17
3 TerraSim	19
3.1 Plynutie času	19
3.2 Mapa	19
3.3 Senzory	20
3.4 Aktuátory	20
4 Experiment.....	22
4.1 Prostredie	22
4.2 Agent.....	23
4.2.1 Motor	23
4.2.2 Centrum komunikácie	24
4.2.3 Centrum rozhodovania	24
4.2.4 Kamera	25
4.2.5 Zber údajov.....	27
4.3 Fáza 1: Kalibrácia učiacich sa mechanizmov	28
4.4 Fáza 2: Simulácie s komunikáciou	32
5 Implementácia	34
6 Výsledky	35

7 Diskusia	42
7.1 Volanie o pomoc	42
7.2 Vidím/nevidím ovocie	42
7.3 Nápady a ďalšie smerovanie experimentu	43
Záver	45
Použitá literatúra	46
Príloha A.....	47
CD médium.....	47

Zoznam obrázkov

Obrázok 1: Schéma umelého neurónu	13
Obrázok 2: Schéma doprednej neurónovej siete s jednou skrytou vrstvou	14
Obrázok 3: Štandardný model učenia posilňovaním (Kaebling et al. 1996)	15
Obrázok 4: Schéma aktér-kritik	17
Obrázok 5: Vizualizácia virtuálneho sveta v TerraSim-e	20
Obrázok 6: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne hodnoty diskontného faktora.....	29
Obrázok 7: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne rýchlosti učenia	30
Obrázok 8: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne štruktúry ANN	31
Obrázok 9: Graf zobrazujúci zmenu priemernej získanej odmeny počas behu simulácie	35
Obrázok 10: Vývoj dát volania príkazu Help počas behu simulácie (typ a)	37
Obrázok 11: Vývoj dát volania príkazu Help počas behu simulácie (typ b)	38
Obrázok 12: Vývoj dát volania príkazu ISeeNoFruit počas behu simulácie (typ a)	38
Obrázok 13: Vývoj dát volania príkazu ISeeNoFruit počas behu simulácie (typ b)	39
Obrázok 14: Vývoj dát volania príkazu ISeeFruit počas behu simulácie (typ a)	40
Obrázok 15: Vývoj dát volania príkazu ISeeFruit počas behu simulácie (typ b)	40

Zoznam tabuliek

Tabuľka 1: Sumarizácia dát simulácií kontrolnej skupiny	29
Tabuľka 2: Sumarizácia dát simulácií pre rôzne hodnoty diskontného faktora	30
Tabuľka 3: Sumarizácia dát simulácií pre rôzne hodnoty rýchlosti učenia.....	31
Tabuľka 4: Sumarizácia dát simulácií pre rôzne štruktúry ANN	32
Tabuľka 5: Sumarizácia dát priemerných odmien vo fáze 2	35
Tabuľka 6: Sumarizácia dát smerodajných odchýliek vo fáze 2	36

Úvod

Jazyk, ako prostriedok komunikácie a jeho efektívne využívanie, je jedna z hlavných črt, ktoré ľudí odlišujú od ostatných žijúcich organizmov. Téma jazyka ako takého nie je jednoduchá, čo naznačuje aj fakt, že je mu venovaná samostatná vedná disciplína, ktorá je zároveň úzko spätá s mnohými ďalšími.

„Otázky týkajúce sa okolností vzniku a evolúcie jazyka sú jedny z najstarších a najzložitejších, ktorými sa linguisti zaoberajú. Zložitosť týchto otázok je spôsobená dvomi hlavnými faktormi:

- Evolúcia jazyka, takého ako ho dnes poznáme, prebehla iba raz v histórii. To znamená, že tento proces nemáme s čím porovnávať. Nevieme ktoré charakteristiky nášho jazyka vznikli náhodou v priebehu histórie, a prítomnosť ktorých je nevyhnutná vo všetkých komunikujúcich systémoch.
- Evolúcia jazyka po sebe zanecháva len malé množstvo stôp napríklad v porovnaní s biologickou evolúciou.“ (Perfors, 2002)

Nedostatok dôkazov a stôp však neznamená, že o evolúcii jazyka nemožeme uvažovať a na základe dôkazov odvodzovať teórie, matematické modely a podobne.

„Na začiatku devedesiatych rokov 20. storočia sa, vďaka prudkému rastu výpočtovej sily, začala do popredia dostávať nová paradigma modelovania.“ (Steels, 2011)

V tejto práci budeme aj používať spomínanú paradigmu. Vytvorili sme virtuálny model sveta, v do ktorého sme umiestnili dve skupiny kognitívnych agentov. Agentom sme implementovali:

- učiace mechanizmy
- senzory, umožňujúce im vnímať svoje okolie
- aktuátory, ktorými môžu meniť/interagovať so svojím okolím

Vytvorili sme 2 skupiny agentov, ktoré sa líšia iba v tom, že agenti jednej skupiny majú možnosť navzájom komunikovať a agenti druhej skupiny túto možnosť nemajú. Agenti o svete na začiatku nič nevedia, ani stratégiu akou by sa mali riadiť pri výbere svojich akcií. Úspešnosť agenta je považovaná výškou priemernej získanej odmeny.

Máme hypotézu, že agenti v skupine, ktorá má možnosť komunikovať, budú v našom experimente úspešnejší. Cieľom tejto práce je túto hypotézu otestovať.

Prvá kapitola popisuje existujúce prístupy k niektorým problémom, s ktorými sme sa stretli aj my. V druhej kapitole uvádzame prehľad teoretickým základom na ktorom je náš experiment postavený. Čitateľa v stručnosti uvedieme do problematiky neurónových sietí, učenia posilňovaním a učenia s časovým oneskorením. V neposlednom rade vysvetlíme architektúru aktéra-kritika, ktorá je jadrom našich agentov. Tretia kapitola popisuje virtuálny svet, v ktorom sa naši agenti pohybujú. Napriek tomu, že TerraSim umožňuje veľké množstvo nastavení (ako napríklad striedanie dňa a noci, počasie, a tak ďalej), sústredíme sa iba na jeho konfigurovateľné súčasti, ktoré sú relevantné pre náš experiment. Experiment je popísaný vo štvrtej kapitole, ktorá je nasledovaná stručným popisom implementačných detailov tejto práce. V šiestej kapitole sumarizujeme výsledky experimentu a v poslednej kapitole o týchto výsledkoch diskutujeme.

1 Existujúce prístupy

„Na to, aby jednotlivec dokázal ťažiť zo schopnosti komunikovať, musí sa najskôr naučiť prospešnosť koordinovaného komunikačného správania. Napríklad, ak nejaké zviera vyšle nejaký signál, ostatné zvieratá ho s veľkou pravdepodobnosťou vypočujú a primerane naň budú reagovať.“ (Perfors, 2002)

„V roku 1997 uskutočnili Oliphant a Batali výskum so zvieratami, zaoberajúci sa otázkou, ako sa jednotlivec naučí komunikovať. Začali používať termín *komunikačná epizóda*. V takejto epizóde jeden člen populácie spozoruje určitú udalosť a vyšle o nej správu ostatným. Ostatné zvieratá počujú signál a zareagujú naň. Epizóda je úspešná, ak reakcia na signál je adekvátna. Dispozície každého jednotlivca na vyslanie alebo prijatie signálu sú charakterizované dvomi pravdepodobnostnými funkciami *send* a *recieve*. Napríklad, keď náš agent zbadá približujúceho sa predátora, tak sa snaží ostatných agentov na túto skutočnosť upozorniť. Náš agent dokáže produkovať viac druhov signálov. Vtedy pravdepodobnostná funkcia *send* reprezentuje pravdepodobnosti, s akou budú dané signály vyslané. Napríklad [brechot: 0.8; pískanie: 0.15; syčanie: 0,05]. Tu definujú ďalší termín *komunikačná presnosť*, ktorý vyjadruje pravdepodobnosť, s akou bude signál vyslaný *send* funkciou jedného agenta korektne interpretovaný *recieve* funkciou druhého agenta.“ (Oliphant & Batali, 1997)

Zvyšok výskumu venujú analýze rôznych učiacich procedúr. Najjednoduchšiu učiacu procedúru, ktorej dávajú veľkú šancu na úspech, nazvali *Imitate-Choose*. Použitím tejto procedúry agent odošle najčastejšie odosielaný signál pre danú udalosť a interpretuje prijímaný signál ako väčšina populácie agentov. Výsledky používania tejto procedúry ukázali, že ak je populácia hneď od začiatku koordinovaná, tak touto stratégiou si učiaci agent koordináciu osvojí a udrží ju v tomto stave. Ak ale populácia nie je dostatočne koordinovaná, nekoordinovanosť agenta ešte vzrastie.

Ďalšiu spomínanú procedúru nazvali *Obverter*. Táto sa snaží odosielať signál, ktorý je pre danú udalosť najčastejšie interpretovaný správne. Keďže učiaci agent nemá prístup k akciám *send* a *receive* ostatných agentov, musí používať odhady pre každú udalosť. Touto metódou dosiahli agenti veľmi dobré výsledky za veľmi krátky čas.

2 Učenie agentov

V tejto časti sa budeme zaoberať učiacimi mechanizmami, ktoré sme využili pri implementácii našich agentov.

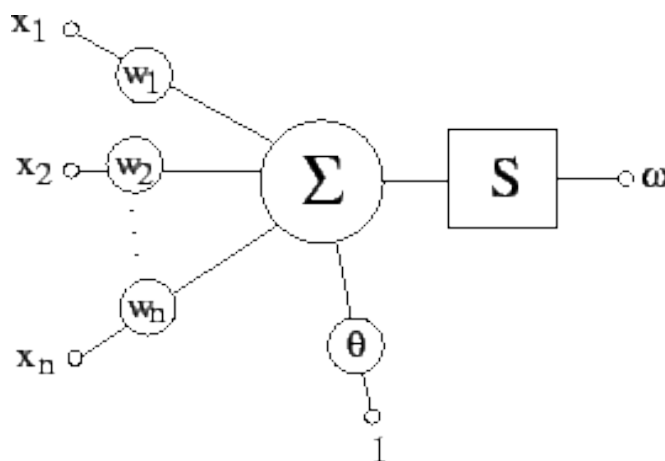
2.1 Neurónové siete

Neurónová sieť (z angličtiny “Artificial Neural Network”, ďalej len ANN), z hľadiska informatiky a informačných technológií je výpočtový model inšpirovaný centrálnym nervovým systémom živých organizmov. ANN sa skladá z výpočtových uzlov (zjednodušený model neurónov), ktoré sú navzájom poprepájané do danej štruktúry. Počet uzlov a štruktúra, do ktorej sú pospájané úzko súvisí s úlohou, ktorú sa daná ANN snaží riešiť.

Neuróny v ANN pracujú na nasledovnom princípe:

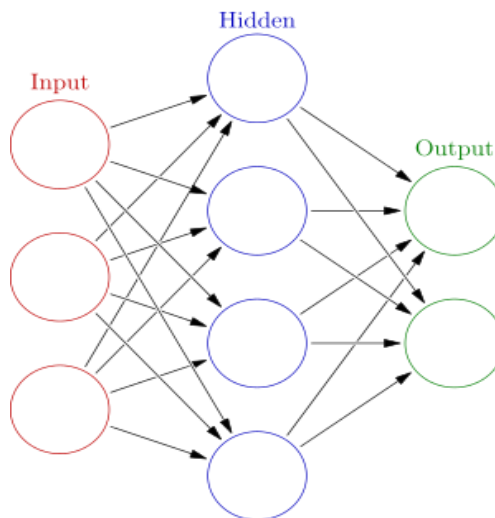
1. Neurón dostane na vstupe vektor hodnôt. Dĺžka vstupného vektora sa pre jednotlivé neuróny nemení. Tieto hodnoty môžu prísť z prostredia na vstupnú vrstvu ANN, alebo prichádzajú z výstupu iných neurónov v rámci ANN.
2. Súčasťou neurónu je vektor váh, ktorého dĺžka je zhodná s dĺžkou vektora vstupov.
3. Neurón vyráta hodnotu skalárneho súčinu vstupného vektora a vektora váh.
4. Hodnotu skalárneho súčinu použijeme ako vstup pre aktivačnú funkciu neurónu.

Ako aktivačná funkcia sa v prípade spojitých neurónov používa sigmoida, poprípade existujú aj iné alternatívy (rovnako aj tu závisí od problému, ktorý sa ANN snaží riešiť). Výstup tejto funkcie je zároveň aj výstupom daného neurónu.



Obrázok 1: Schéma umelého neurónu

Jedna z najjednoduchších štruktúr ANN je dopredná neurónová sieť. Táto sa skladá z neurónov zoskupených vo vrstvách. Dopredná ANN obsahuje vstupnú vrstvu, výstupnú vrstvu a medzi nimi sa môže nachádzať ľubovoľný počet skrytých vrstiev. Vrstvy sú usporiadané postupne, ich poradie sa nemení. Výstup jednej vrstvy sa stáva vstupom pre vrstvu jej nasledujúcu, takže vstupný signál sa šíri stále iba v jednom smere (dopredu).



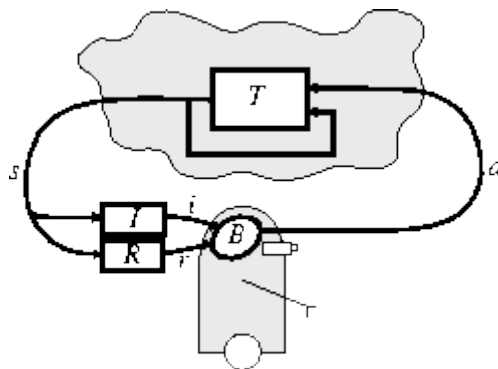
Obrázok 2: Schéma doprednej neurónovej siete s jednou skrytou vrstvou

Trénovanie ANN zabezpečuje mechanizmus spätného šírenia chyby (z angličtiny Backpropagation) (Rumelhart, Hinton, & Williams, 1988). Po zadání vstupu sieť vypočíta výstup (vektor hodnôt veľkosti rovnkej počtu neurónov na výstupnej vrstve), ktorý sa následne porovná s požadovaným výstupom. Ak sa zhodujú (resp. sú vyhodnotené ako dostatočne zhodné), trénovanie nenastáva. V opačnom prípade sa chyba, vypočítaná ako rozdiel medzi skutočným výstupom siete a požadovaným výstupom, spätne šíri od výstupnej vrstvy smerom k vstupnej vrstve. Chyba daného neurónu prispieva k chybe neurónov na predošlej vrstve čiastkou, ktorej veľkosť závisí od váhy príslušného prepojenia. Hodnotu, o ktorú musíme zmeniť váhu, vypočítame ako súčin šírenej chyby, váhy prepojenia a rýchlosti učenia.

2.2 Učenie posilňovaním

Učenie posilňovaním (z angličtiny Reinforcement Learning, ďalej len RL) je podmnožinou strojového učenia, ktorej vznik bol inšpirovaný behaviorálnou psychológiou. Zaoberá sa otázkou učenia agentov, ktorí nemajú k dispozícii príklady optimálneho výberu akcie od učiteľa. Rozdiel medzi učením s učiteľom a RL si môžeme ukázať na príklade učenia agentov hrať šach. V prvom prípade učiteľ prechádza hru s agentom ťah po ťahu

a ukazuje mu optimálne riešenie danej situácie. Pri RL si môžeme agenta predstaviť ako hráča, ktorý nepozná stratégiu, je mu dovolené spraviť iba povolené ťahy a na konci hry dostane agent spätnú väzbu o výhre či prehre. Takúto spätnú väzbu nazývame odmena.



Obrázok 3: Štandardný model učenia posilňovaním (Kaelbling et al. 1996)

„V štandardnom modeli RL je agent v spojení s prostredím prostredníctvom vnemov a akcií, ako je znázornené na obrázku číslo 3. Interakcia s prostredím prebieha nasledovne. Agent dostane z prostredia vstup i a indikáciu stavu prostredia s . Následne agent vyhodnotí tieto vstupy, výsledkom čoho je výber akcie a . Akcia zmení stav prostredia a hodnota prechodu medzi stavmi je komunikovaná na agenta vo forme odmeny r . Agentove správanie B , by malo vyberať akcie tak, aby zvyšovali z dlhodobého hľadiska súčin odmien. Toto sa agent môže postupom času naučiť systematickým využívaním metódy pokus-omyl.

Formálne, model pozostáva z:

- množiny stavov prostredia, S
- množiny akcií, ktoré môže agent vykonať, A
- množiny hodnôt odmien

Na obrázku je taktiež zobrazená funkcia transformácie vstupu I , ktorá určuje ako agent vidí prostredie (napr. vníma iba časť prostredia okolo neho).” (Kaelbling, Littman, & Moore, 1996)

Kaelbling s kolektívom vo svojej prehľadovej štúdii vychádzali z práce Suttona a Bartoa z roku 1998 „Reinforcement Learning: An Introduction”.

2.3 Explorácia

„Výsledkom vykonávania akcií nie je iba získavanie odmien, ale aj učenie sa modelu skutočného prostredia, v ktorom sa agent pohybuje.” (Russell & Norvig, 2002) Aby sme predišli tomu, že agent uviazne s naučeným modelom, ktorý nezodpovedá skutočnému modelu svojho sveta, musíme použiť mechanizmus explorácie. Tento mechanizmus núti agenta vyberať akcie, ktoré nezodpovedajú optimálnej stratégii naučeného modelu sveta, čo v konečnom dôsledku môže zlepšiť naučený model.

Existuje viacero metód, ktoré sa používajú na výber akcie, ktoré v rozumnom pomere vyvažujú výber akcie optimálnej (na základe naučeného modelu) a exploratívnej. Jednou z často používaných je ϵ -greedy metóda výberu, ktorá s pravdepodobnosťou ϵ vyberie náhodnú akciu.

„Napriek tomu, že ϵ -greedy metóda je v RL veľmi populárna, jej nevýhodou je, že pri explorácii majú všetky akcie rovnakú pravdepodobnosť byť vybrané. Toto znamená, že s rovnakou pravdepodobnosťou je vybraná najhoršia možná akcia aj najlepšia možná alternatíva. V úlohách, v ktorých je najhoršia možná akcia veľmi zlá, je takáto metóda výberu nevyhovujúca.” (Sutton & Barto, 1998)

Alternatívna metóda výberu akcie je *softmax*. Táto metóda všetkým akciám určí pravdepodobnosť s akou budú vybrané. Väčšinou sa v softmaxe využíva Boltzmannova, alebo Gibbsova distribúcia. Použitím Boltzmannovej distribúcie sa pravdepodobnosti akcií vypočítajú pomocou vzorca

$$P(a_i|s) = \frac{e^{\beta Q(s,a_i)}}{\sum_{j=1}^n e^{\beta Q(s,a_j)}}$$

V tomto vzorci značí parameter $Q(s,a_i)$ funkciu ohodnotenia výberu akcie a_i v stave s , a β nazývame *inverzná teplota* a n je počet možných akcií. Hodnota β sa časom zvyšuje. Keď je $\beta=0$, výber akcie je úplne náhodný (pravdepodobnosť výberu každej akcie je $1/n$). Čím je hodnota parametra β vyššia, tým je výber akcií menej náhodný. V limite $\beta \rightarrow \infty$ je pravdepodobnosť výberu akcie s najvyšším ohodnotením rovná 1.

2.4 Učenie s časovým oneskorením

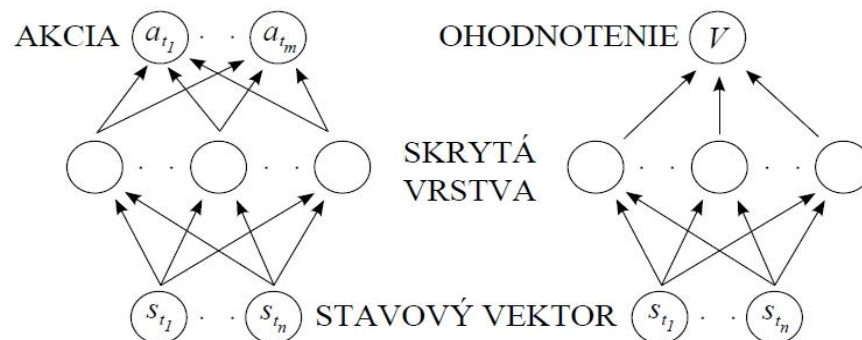
Aby výber akcií agenta konvergoval k optimálnej stratégii, potrebuje sa na základe získavanej odmeny naučiť ohodnocovať utilitu stavov. Učenie ohodnocovania stavov realizujeme metódou učenia s časovým oneskorením (z angličtiny Temporal Difference, ďalej len TD) (Sutton & Barto, 1998).

„Základnou ideou všetkých TD metód je:

1. zdefinovať podmienky, ktoré určia či je aktuálne ohodnotenie stavu správne, alebo ho treba upraviť
2. zdefinovať pravidlo, ktoré v prípade potreby upraví ohodnotenie stavu smerom k jeho ideálnej hodnote” (Russell & Norvig, 2002)

2.5 Aktér-kritik

„Aktér-kritik je jedna zo základných schém TD. Neurónový aktér-kritik väčšinou pozostáva z dvoch neurónových sietí. Prvá, ktorú nazývame kritik, aproximuje ohodnocovacia funkciu, ktorá mapuje stavy na ich predpokladané hodnoty. Druhá, ktorú nazývame aktér, ktorá je zodpovedná za výber akcie v danom stave. Kritik dostáva externú odmenu (z prostredia) a transformuje ju na internú (heuristickú) odmenu. Kritik upravuje svoje váhy na základe TD metódy. Aktér sa snaží naučiť optimálnu stratégiu výberu akcií. Akcie vyberá na základe pravdepodobnosti, čo zabezpečuje exploráciu. Tento spôsob výberu akcií by po čase mal skonvergovať k optimálnym akciám s pravdepodobnosťou jedna. Aktér-kritik sa pokúša nájsť aj optimálnu stratégiu aj optimálnu ohodnocovacia funkciu.” (Mizutani & Dreyfus, 1998) (Benc, 2011)



Obrázok 4: Schéma aktér-kritik

Kritika trénujeme v každom kroku. Jeho váhu upravujeme na základe rovnice

$$V_{t+1}(s_t) = V_t(s_t) + \alpha_t \delta_t \quad ; \quad \delta_t = r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)$$

Hodnoty $V(s)$ reprezentujú kumulatívnu diskontovanú odmenu (ohodnotenie stavu). Rýchlosť učenia α_t a diskontný faktor γ je hodnota z intervalu $(0;1)$. Diskontný faktor určuje, akou mierou kumulatívnu diskontovanú odmenu ovplyvnia odmeny v budúcich stavoch. Zmenu hodnoty stavu po vykonaní akcie označujeme δ_t . Odmenu získanú po vykonaní akcie a_t v stave s_t označujeme r_{t+1} .

To, či v danom kroku máme trénovať aj aktéra je dané hodnotou zmeny stavu δ_t . Ak je hodnota δ_t väčšia ako nula, znamená to, že akcia a_t vykonaná v stave s_t zabezpečí vyššiu kumulatívnu diskontovanú odmenu. V aktérovi výber tejto akcie v danom stave by sme teda mali podporiť. Ak je δ_t menšia, alebo rovná nule, akcia nevedie k vyššej kumulatívnej odmene a aktéra netrénujeme.

3 TerraSim

V tejto časti popisujeme prostredie, v ktorom sa naši agenti pohybujú. Ako sme spomínali pri opise modelu RL, agent sa potrebuje pohybovať v prostredí, ktoré môže skúmať, meniť ho dôsledkom vykonávania svojich akcií a pozorovať ako tieto akcie na dané okolie vplývajú. Pre naše účely sme si vybrali projekt TerraSim, ktorý vznikol ako výsledok diplomovej práce Mgr. Benca v roku 2011.

Výsledkom jeho práce bol sieťový server s názvom TerraSim. Na tomto servri je možné spúšťať simulácie virtuálnych svetov. Tieto svety umožňujú agentom ich skúmať a usudzovať o ich vlastnostiach. Simulácia sveta navyše okrem agentov môže obsahovať rôzne objekty. Záleží len na agentovi, či ich bude iba pozorovať, pokúsi sa ich kategorizovať, pokúsi sa s nimi nejakým spôsobom interagovať a následne pozorovať výsledky svojich akcií. TerraSim podporuje jedno-agentové, ale aj multi-agentové scenáre. Architektúra riešenia umožňuje jednoducho definovať nové senzory a aktuátory agentov. Bez väčších problémov sa dajú definovať nové virtuálne svety. (Benc, 2011)

Prostredie TerraSim-u je veľmi dobre konfigurovateľné, čo umožňuje simuláciu veľkého rozsahu scenárov, ktorým môžu agenti čeliť. TerraSim umožňuje množstvo nastavení prostredia, ktoré ale pre účel tejto práce nie sú podstatné. V nasledujúcom texte preto opíšeme iba tie časti, ktoré sme využili v našom experimente.

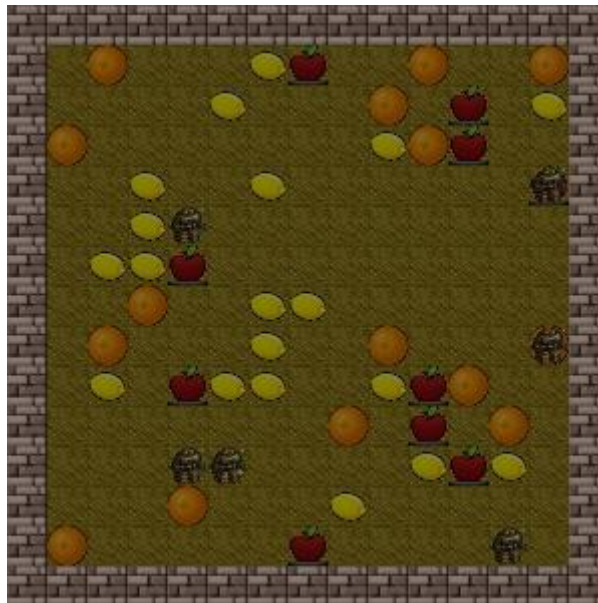
3.1 Plynutie času

Základná časová jednotka simulácii je deň (logická jednotka, trvajúca určený čas reálneho sveta). Deň môžeme rozdeliť na ľubovoľný počet časových krokov rovnakej veľkosti. V každom časovom kroku dostanú všetky entity, nachádzajúce sa vo virtuálnom svete (agenti, ako aj ostatné objekty v prostredí), možnosť vykonať akcie podľa ich implementovanej logiky.

3.2 Mapa

Agenti sa v TerraSime pohybujú po mape reprezentovanej štvorcovou mriežkou. Každý uzol v mriežke reprezentuje pozíciu na mape. Každá pozícia obsahuje zoznam entít, ktoré sú na nej umiestnené (typ terénu, agenti, objekty,...). Záleží na dizajnérovi daného virtuálneho sveta, aké obmedzenia pre pozície vytvorí. Napríklad, bude viac typov terénu,

pričom na pozíciu obsahujúcu typ terénu *stena* už nemôžeme umiestniť žiadny iný objekt a na pozícii obsahujúcej typ terénu *podlaha*, sa môže nachádzať naraz viac entít rôzneho typu.



Obrázok 5: Vizualizácia virtuálneho sveta v TerraSim-e.

3.3 Senzory

Súčasťou každého agenta je zoznam senzorov. Senzory zastupujú agentove zmysly v jeho virtuálnom svete. Senzor je časť agenta, ktorá v každom časovom kroku automaticky zo sveta extrahuje určitú informáciu. Tieto informácie sú potom agentovi k dispozícii a ten ich môže využiť pri jeho ďalšom rozhodovaní. Agent môže napríklad obsahovať senzor typu *kamera*, ktorý simuluje agentov vizuálny vnem jeho okolia.

3.4 Aktuátory

Súčasťou každého agenta je taktiež zoznam aktuátorov. Aktuátory sú nástroje, ktorými môže agent vplývať na svoje okolie. Každý z nich agentovi poskytuje zoznam akcií, ktoré môže vykonať. Agent môže mať, napríklad, k dispozícii aktuátor *motor*, ktorý mu umožňuje meniť svoju polohu vo virtuálnom svete.

Vykonanie akcie aktuátorom prebieha nasledovne:

1. Agent zavolá svoju funkciu *Perform*, ktorá na vstupe dostáva tri parametre. Prvým je názov akcie, ktorú chce agent vykonať. Ďalšie dva parametre sú číselné hodnoty. Tieto sa neskôr použijú ako argumenty pre volanú akciu.

2. V rámci akcie *Perform* agent prejde všetky aktuátory a zistí, či niektorý z nich vie vykonať želanú akciu.

Ak existuje aktuátor, ktorý dokáže danú akciu vykonať, agent prostredníctvom aktuátora vykoná želanú akciu, pričom táto dostane na vstup dva vyššie spomínané argumenty. Záleží od danej implementácie akcie, či vyžaduje argumenty alebo nie.

4 Experiment

Ako sme už spomínali, hlavným cieľom nášho experimentu je pozorovať, či agenti začnú navzájom komunikovať a či sa úspešným používaním tejto možnosti zvýši ich úspešnosť pri riešení danej úlohy. V tejto časti popisujeme detaily nášho experimentu.

4.1 Prostredie

Agenti sú umiestnení do virtuálneho sveta TerraSim-u. Mapa, po ktorej sa pohybujú má rozmery 15x15 políček. Každému políčku je priradený typ terénu. V našom prípade sú to dva typy, a to :

1. Terén typu *stena* – políčko tohto typu v našom svete reprezentuje prekážku. Na toto políčko nemôže byť umiestnený žiadny objekt (agent, ovocie).
2. Terén typu *podlaha* – na políčku tohto typu môže byť umiestnený najviac jeden agent a zároveň najviac jeden objekt typu *ovocie*. Všetky políčka, ktoré nie sú typu *stena*, sú tohto typu.

Mapu sme sa snažili čo najviac zjednodušiť, aby sme agentom čo najviac uľahčili učenie. Skladá sa z políček typu *podlaha* s rozmermi 13x13, pričom je táto plocha ohradená políčkami typu *stena*.

Na mape sú okrem agentov umiestnené objekty typu *ovocie*. Každý objekt tohto typu má atribúty *nutrition* a *množstvo*. Hodnota atribútu *nutrition* je rovná odmene, ktorú agent z prostredia získa, keď sa postaví na políčko s týmto ovocím. Definovali sme tri podtypy ovocia:

1. *Citrón* – *Nutrition* = 1
2. *Pomaranč* – *Nutrition* = 2
3. *Jablko* – *Nutrition* = 4

Hodnota atribútu *množstvo*, ktorého defaultná hodnota je veľkosti 5, sa zníži v každom časovom kroku, v ktorom ovocie zdieľa políčko s agentom. Po tom, čo atribút *množstvo* dosiahne hodnotu rovnú nule, ovocie zmení svoju pozíciu na mape a hodnota atribútu sa nastaví na svoju defaultnú veľkosť.

Pri štarte simulácie sa prednastavený počet agentov a ovocia rozmiestni na náhodné pozície na mape. Vizualizáciu nášho sveta môžete vidieť na obrázku číslo 5.

Počas behu simulácie sme v každom časovom kroku, každému agentovi postupne vyhradili 40 milisekúnd výpočtového času, za ktorý musí stihnúť:

- použiť svoje senzory na zistenie aktuálneho stavu prostredia
- na základe vnemov získaných zo senzorov rozhodnúť o svojej nasledujúcej akcii
- vykonať akciu, pre ktorú sa rozhodol

4.2 Agent

Ako sme už spomenuli, súčasťou agenta je zoznam senzorov, ktorými vníma svoje prostredie a zoznam aktuátorov, ktoré mu umožňujú interagovať so svojím okolím. V nasledujúcich sekciách sa im venujeme vo väčšom detaile.

4.2.1 Motor

Motor je aktuátorom agenta, ktorý mu umožňuje pohyb vo virtuálnom svete. Umožňuje agentovi pohyb v ôsmich smeroch (vertikálne, horizontálne aj diagonálne).

Keďže sa v celom našom experimente snažíme agentom čím viac všetko zjednodušovať, rozhodli sme, že *motor* bude vedieť meniť agentovu pozíciu iba s krokom dĺžky jedna (môže sa pohnúť iba na susediace políčka). Toto zabezpečí, že sa agent nemusí učiť nastavovať parametre určujúce dĺžku kroku.

Kontrolu toho, či je príkaz validný, zabezpečuje samotný aktuátor. Zoznam nepovolených akcií:

- agent sa snaží pohnúť na políčko obsahujúce stenu
- agent sa snaží pohnúť na políčko, na ktorom sa nachádza iný agent

V prípade, že aktuátor dostane príkaz vykonať nevalidný pohyb, je príkaz ignorovaný a agent si nastaví atribút *actionSuccessful* na hodnotu false. Hodnota tohto atribútu sa neskôr používa pri výpočte odmeny pre agenta.

4.2.2 Centrum komunikácie

Centrum komunikácie je aktuátor agenta, ktorý agentovi umožňuje komunikáciu s ostatnými agentmi. Agent komunikuje prostredníctvom signalizácie. Centrum signalizácie obsahuje atribút signál, ktorému nastavuje hodnotu správy, ktorú sa snaží komunikovať.

Tento aktuátor umožňuje agentovi vykonávať štyri akcie:

1. *DoNothing* - Agent nereaguje na pokusy ostatných agentov o komunikáciu. Neurobí nič a hodnotu svojho atribútu signál nastaví na -2.
2. *Help* - Agent vysielá ostatným agentom volanie o pomoc. Hodnotu svojho atribútu signál nastaví na -4.
3. *ISeeFruit* – Agent vysielá ostatným agentom správu, že v jeho perцепte sa nachádza aspoň jedno políčko s ovocím a bez agenta. Hodnotu svojho atribútu signál nastaví na 8.
4. *ISeeNoFruit* – Agent vysielá ostatným agentom signál, že v jeho perцепte sa nenachádza ani jedno políčko s ovocím a bez agenta. Hodnotu svojho atribútu signál nastaví na -8.

Defaultná hodnota atribútu signál je hodnota signálu *DoNothing*. Nastavený signál je agentovi ponechaný počas jedného časového kroku. Toto zabezpečí, že všetci agenti v jeho okolí majú možnosť ho vidieť a na začiatku ďalšieho kroku sa nastaví na defaultnú polohu.

Hodnota atribútu signál je pozorovateľná ostatnými agentmi prostredníctvom ich senzoru *kamera*. To znamená, že agent môže komunikovať iba s agentmi, v ktorých pozorovateľnom okolí sa nachádza. Všetci agenti majú rovnako veľké okolie, ktoré dokážu pozorovať, takže nikdy nenastane situácia, že by komunikácia nemohla prebiehať obojstranne.

Výber hodnôt atribútu *signál* je odôvodnený neskôr, v sekcii *kamera*.

4.2.3 Centrum rozhodovania

Centrum rozhodovania je aktuátor agenta. Tento aktuátor agentovi neposkytuje žiadne ďalšie akcie, ktoré by mohol vykonať. Namiesto toho je jeho súčasťou implementácia agentovej logiky.

Súčasťou centra rozhodovania je štruktúra aktér-kritik. Vstupom pre aktéra-kritika je jednorozmerné pole reálnych čísel, reprezentujúce agentov stav. Tieto dáta sú extrahované senzorom *kamera* z agentovho vizuálneho vnemu. Centrum rozhodovania k tomuto poľu

ešte pripája jeden extra element. Tento je použitý ako indikátor úspechu vykonania vybratej akcie. Indikátor signalizuje neúspech iba ak sa agent pokúsil vykonať invalidnú akciu. Invalidné akcie dokáže vykonávať iba aktuátor *motor* (pre zoznam a popis vid' sekciu *motor*).

Kritik je implementovaný doprednou ANN s jednou skrytou vrstvou. Na vstupnej vrstve má 25 neurónov. Na skrytej vrstve má 50 neurónov (dvojnásobný počet neurónov na vstupnej vrstve) a na výstupnej vrstve je len jeden neurón. Neuróny na výstupnej vrstve používajú lineárnu aktivačnú funkciu, a neuróny na skrytej vrstve používajú sigmoidnú aktivačnú funkciu.

Aktér, rovnako ako kritik, je implementovaný doprednou ANN s jednou skrytou vrstvou. Konfigurácia vstupnej a skrytej vrstvy je rovnaká ako u kritika. Na aktérovej výstupnej vrstve sa ale nenachádza iba jeden neurón, ale 12 (každý symbolizuje jednu akciu, ktorú dokáže aktér vykonať). Neuróny na výstupnej vrstve, rovnako ako u kritika, používajú lineárnu aktivačnú funkciu. Na výstupný vektor aktéra aplikujeme, ako metódu výberu akcie, softmax s Boltzmannovou distribúciou. Jej výstupom je vektor pravdepodobnosti, s akou agent vyberie nasledovnú akciu.

Učenie aktér-kritika prebieha nasledovne:

1. Agent sa nachádza v stave s_t
2. Aktér, za použitia exploračie, vyberie akciu a_t
3. Agent vykoná akciu a_t , čím prejde do stavu s_{t+1}
4. Kritik upraví svoje váhy za použitia rovnice
$$V_{t+1}(s_t) = V_t(s_t) + \alpha_t(r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t))$$
5. Ak akcia viedla k stavu s lepším ohodnotením ako predpokladal kritik
$$V_{t+1}(s_t) > V_t(s_t),$$
upraví svoje váhy aj aktér

4.2.4 Kamera

Kamera je jediný senzor nášho agenta. Reprezentuje vizuálny vnem agentovho okolia. Jeho súčasťou je atribút *distance*, ktorý určuje koľko políčok okolo agenta je schopná kamera zachytiť. V našom experimente používame hodnotu *distance* veľkosti rovnej 2. Agent nevníma svoju pozíciu vo virtuálnom svete. Súradnice všetkých entít vníma relatívne vzhľadom na svoju polohu.

V každom časovom kroku zavolá kamera metódu *Sense()*. *Sense()* zachytí dvojrozmerné pole políček s rozmermi 5x5 ($((distance*2 + 1)^2)$), v ktorého streda sa nachádza políčko s agentom. Z tohto dvojrozmerného poľa extrahuje dáta, ktoré neskôr agent použije ako vstup pre aktér-kritika.

Kamera taktiež zabezpečuje pre agenta transformačnú funkciu *I*, ktorá na vstupe dostáva vnem agentovho okolia a transformuje ho na formát, s ktorým agent dokáže pracovať. Zároveň agentovi sprostredkováva aj funkciu *R*, ktorá z perceptu určí, akú veľkú odmenu agent získal v aktuálnom stave.

Transformačná funkcia používa nasledovné hodnoty:

- -8: podlaha, na ktorej stojí agent, vysielajúci signál *ISeeNoFruit*
- -4: podlaha, na ktorej stojí agent, vysielajúci signál *Help*
- -2: podlaha, na ktorej stojí agent, ktorý sa nesnaží komunikovať
- -1: stena
- 0: podlaha, na ktorej sa nenachádza žiadny objekt
- 1: podlaha, na ktorej sa nachádza citrón
- 2: podlaha, na ktorej sa nachádza pomaranč
- 4: podlaha, na ktorej sa nachádza jablko
- 8: podlaha, na ktorej stojí agent, vysielajúci signál *ISeeFruit*

Ak sa na jednom políčku nachádza zároveň ovocie aj agent, tak sa v extrahovaných dátach toto políčko uvedie s hodnotou agenta. Ovocie zdieľajúce políčko s agentom pre stav sveta nie je relevantné, pretože iný agent sa na toto políčko aj tak nemôže pohnúť.

Dvojrozmerný percept, ktorý *kamera* získa transformujeme nasledovne:

1. Vytvoríme jednorozmerné pole reálnych čísel dĺžky 25 ($((distance*2 + 1)^2)$), pričom všetky elementy v poli inicializujeme na hodnotu -1 (hodnota steny).
2. Prechádzame cez všetky políčka perceptu a tam, kde nájdeme ovocie, alebo čistú podlahu nastavíme v jednorozmernom poli príslušnú hodnotu.
3. Políčka s agentmi nastavíme až na konci, aby sme zabezpečili prekrytie hodnôt čistých podláh a podláh s ovocím, hodnotami agentov
4. Prekrytie ovocia, alebo čistej podlahy neaplikujeme v prípade agenta, pre ktorého sa daný percept vykonáva. Ak by sme prekryli tieto hodnoty, nevedeli by sme správne ohodnotiť stav, v ktorom sa agent nachádza.

Extrahované dáta si uloží do histórie perceptov. V našom experimente nám postačovalo uchovávať históriu dát s jedným časovým krokom dozadu. Históriu perceptov využívame neskôr pri TD.

Funkcia odmien R vracia hodnoty podľa nasledovných pravidiel:

- Ak agent vykonal akciu *Help*, nedostáva žiadnu odmenu.
- Agenti vždy dostanú zápornú odmenu, s hodnotou -1, ak nesprávne signalizujú prítomnosť/neprítomnosť ovocia vo svojom percepte
- Agenti dostanú odmenu, s hodnotou 1, za komunikáciu iba ak správne signalizujú prítomnosť/neprítomnosť ovocia vo svojom percepte a zároveň vidia agenta, ktorý signalizuje volanie *Help*.
- Agenti nedostanú žiadnu odmenu, ak správne signalizujú prítomnosť/neprítomnosť ovocia vo svojom percepte, ale nevidia žiadneho agenta, ktorý by signalizoval volanie *Help*.
- Ak sa agent nachádza na poličku, na ktorom sa nachádza aj ovocie, dostáva odmenu veľkosti hodnoty atribútu *Nutrition* daného ovocia.
- Ak sa agent snaží vykonať nepovolenú akciu, je penalizovaný zápornou hodnotou -1.
- Ak sa dá v jednom stave aplikovať viac pravidiel, výsledná odmena sa rovná súčtu hodnôt ich odmien.

4.2.5 Zber údajov

Pre naše potreby sme vytvorili nástroje na zaznamenávanie údajov, ktoré ukladajú v každom časovom kroku odmenu, ktorú agent získal v danom stave. Dáta potom reprezentujeme metódou *sliding window*. V tejto sa definujú dva parametre:

- veľkosť dátového okna – počet záznamov, ktoré sa pri výpočte hodnoty reprezentujúcej dátové okno berú do úvahy
- veľkosť kroku – počet záznamov, ktoré oddeľujú začiatok daného okna od jeho nasledovníka

Touto metódou získavame priemernú hodnotu priemernej odmeny v danom časovom okne. V našom experimente sme používali veľkosť dátového okna s hodnotou 5000 a veľkosť kroku bola 1000.

Pre dôslednú analýzu komunikačných akcií sme potrebovali zbierať ďalšie parametre. Pre každý typ komunikačnej akcie sme si zaznamenali:

- koľko krát bola v danom časovom okne vykonaná
- či bola v danom kontexte zavolaná správne
- či bola zavolaná ako reakcia na komunikačný akt iného agenta

4.3 Fáza 1: Kalibrácia učiacich sa mechanizmov

V prvej fáze experimentu sme hľadali najvhodnejšiu konfiguráciu parametrov implementovaných mechanizmov. Od hľadanej konfigurácie sme očakávali, že sa v rozumnom čase čím viac priblížili k naučeniu modelu prostredia, v ktorom sa pohybuje. Túto sme neskôr využili v druhej fáze.

Sústredili sme sa najmä na vhodné nastavenie nasledujúcich parametrov:

- Diskontný faktor TD
- Rýchlosť učenia ANN
- Štruktúra ANN

V prvej fáze sme dali agentom možnosť výberu akcií aktuátora motor a akcie, ostat' stať na mieste, bez možnosti komunikovať.

Porovnávali sme dve skupiny agentov:

1. Prvá skupina (kontrolná). Agenti, ktorí robia iba náhodné pohyby.
2. Druhá skupina. Agenti, ktorí sa dokážu učiť.

Druhej skupine sme po jednom menili parametre a pozorovali sme zmenu kontrolnej premennej. Kontrolná premenná bola v našom prípade priemerná získaná odmena za jeden krok.

V tejto fáze sme spúšťali simulácie s nasledovným nastavením:

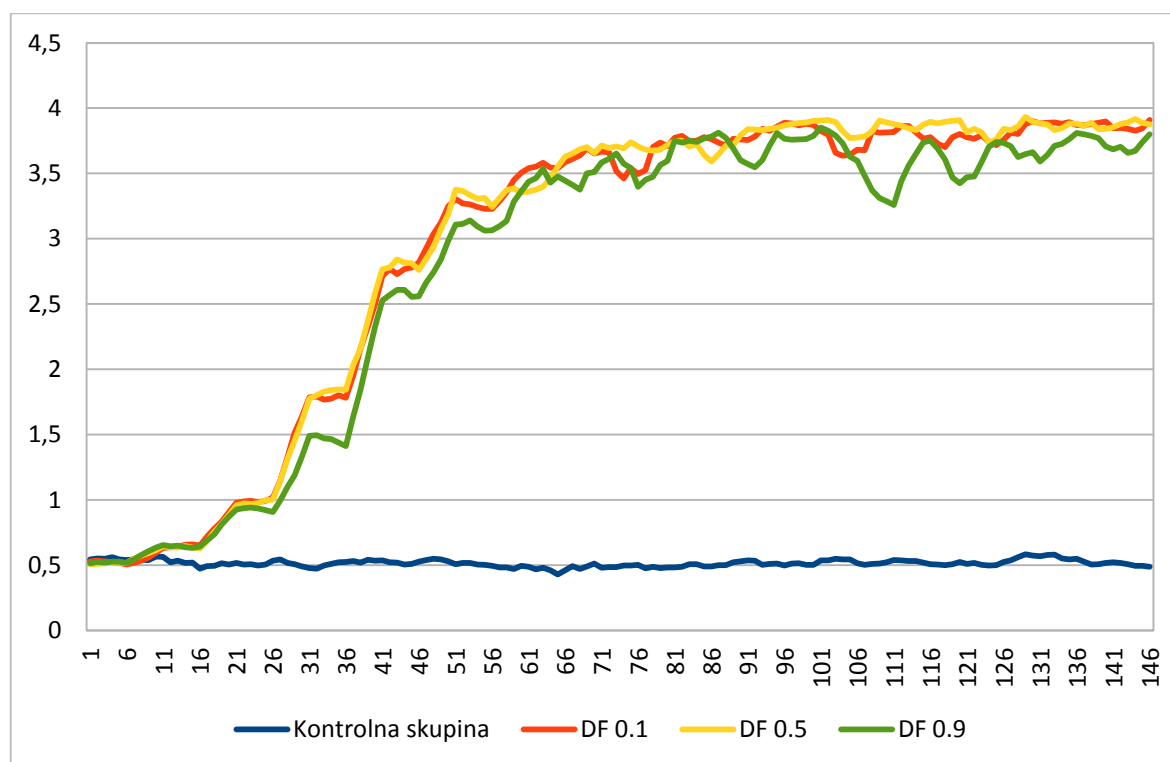
- Do prostredia sme umiestnili 5 agentov.
- V prostredí sa nachádzali všetky podtypy ovocia v množstve:
 - Jablko - 10 objektov
 - Pomaranč – 15 objektov
 - Citrón – 20 objektov

- Políčko, na ktoré sme ovocie presúvali (nastáva pri poklese hodnoty atribútu *množstvo* na 0), bolo vždy vybraté spomedzi voľných políčok, susediacich s agentovou pozíciou. Toto zabezpečí, že ovocie sa po premiestnení nedostane mimo agentov percept, čo umožní agentovi naplno prejavit' efektívnosť učiacich mechanizmov v danej konfigurácii.
- Počiatočná poloha objektov na mape (ovocie a agent) je generovaná náhodne na začiatku každej simulácie.
- V každej simulácii sme agenta nechali spraviť 150000 krokov
- V grafoch zobrazujeme priemer dát piatich agentov

Z nastavení virtuálneho prostredia v prvej fáze vyplýva, že možná odmena sa pohybuje v intervale $\langle -1; 4 \rangle$.

Tabuľka 1: Sumarizácia dát simulácií kontrolnej skupiny

Kontrolná skupina	Celkový priemer	Celkový priemer %	Najvyšší priemer dátového okna	Najvyšší priemer dátového okna %
Náhodné akcie	0,51386438	30,277288	0,5828	31,656

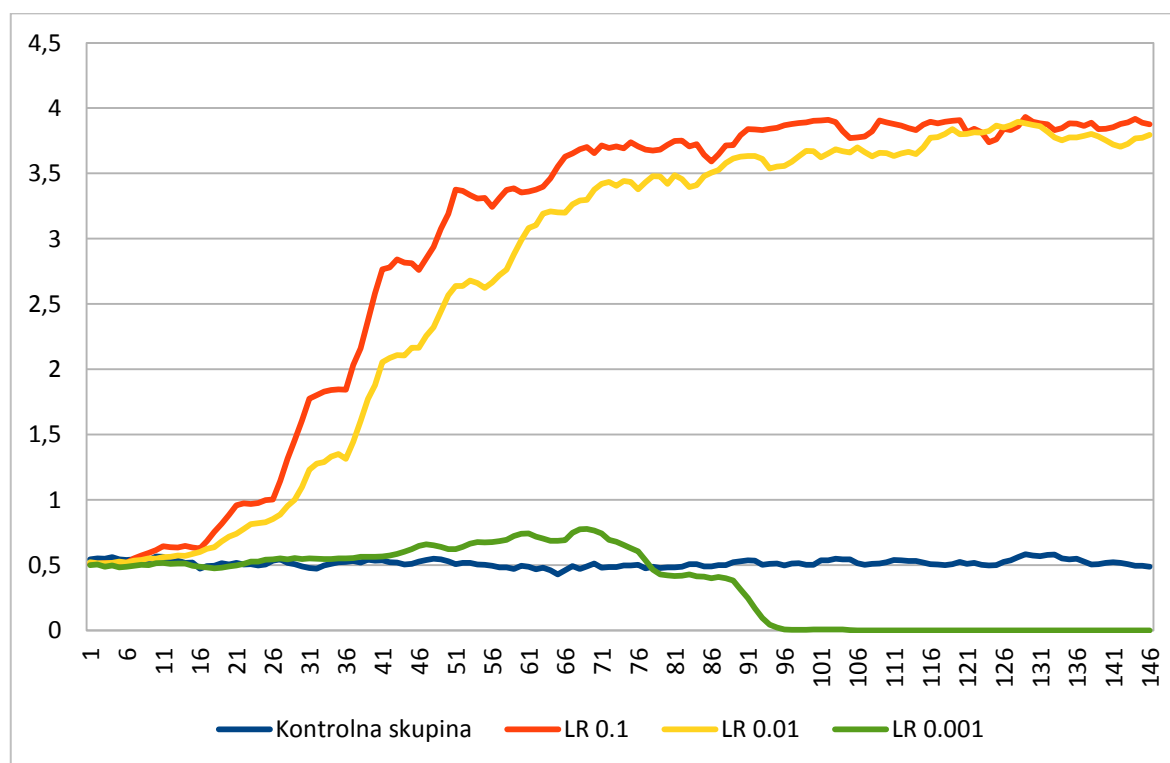


Obrázok 6: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne hodnoty diskontného faktora

Na obrázku číslo 6 vidíme graf efektivity učiacich sa mechanizmov s rôznymi hodnotami diskontného faktoru. Rýchlosť učenia bola v týchto simuláciách nastavená na hodnotu 0.1 a ANN mali na skrytej vrstve 25 neurónov. Z grafu vieme vyčítať, že zmena diskontného faktora nemá na učiace mechanizmy výrazný vplyv. V našich testoch bola ale priemerná hodnota získanej odmeny, počas celého behu simulácie, nepatrne vyššia pri konfigurácii s diskontným faktorom 0,5. Podrobnejšie výsledky sú zobrazené v tabuľke číslo 2.

Tabuľka 2: Sumarizácia dát simulácií pre rôzne hodnoty diskontného faktora

Diskontný faktor	Celkový priemer	Celkový priemer %	Najvyšší priemer dátového okna	Najvyšší priemer dátového okna %
0.1	2,937574247	78,7514849	3,9114	98,228
0.5	2,956852329	79,1370466	3,93272	98,6544
0.9	2,803597808	76,0719562	3,85044	97,0088



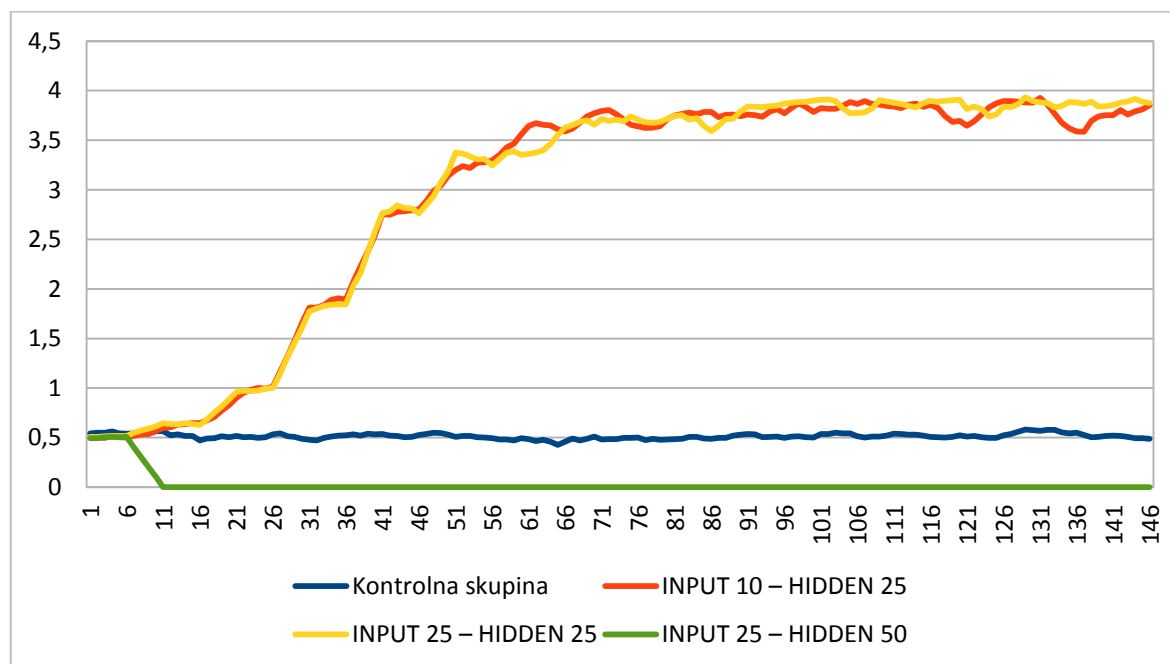
Obrázok 7: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne rýchlosti učenia

Na obrázku číslo 7 vidíme graf efektivity učiacich sa mechanizmov s rôznymi hodnotami rýchlosti učenia. Diskontný faktor bol v týchto simuláciách nastavený na hodnotu 0,5 a ANN mali na skrytej vrstve 25 neurónov. Podrobnejšie výsledky sú zobrazené v tabuľke číslo 3.

Na grafe jasne vidno, že v tomto prípade mali najlepšie výsledky agenti, ktorí sa dokázali učiť najrýchlejšie. Lepšie výsledky začali prejavovať skôr ako ostatné skupiny agentov a rovnako aj priemerná hodnota získanej odmeny, počas celého behu simulácie, bola vyššia.

Tabuľka 3: Sumarizácia dát simulácií pre rôzne hodnoty rýchlosti učenia

Rýchlosť učenia	Celkový priemer	Celkový priemer %	Najvyšší priemer dátového okna	Najvyšší priemer dátového okna %
0.1	2,956852329	79,1370466	3,93272	98,6544
0.01	2,702827397	74,0565479	3,89612	97,9224
0.001	0,352646575	27,0529315	0,77692	35,5384



Obrázok 8: Graf zmeny priemernej odmeny počas behu simulácie pre rôzne štruktúry ANN

V prípade skupiny s najnižšou hodnotou rýchlosti učenia vidíme, že po čase ich získavaná hodnota klesla na nulu. Toto správanie vzniklo v dôsledku nastavenia parametra teplota v našej implementácii softmaxu. Rýchlosť rastu teploty, ktorá spôsobuje znižovanie

náhodnosti výberu akcie v čase neumožnil agentom vykonať dostatočný počet náhodných akcií, čo spôsobilo ich neschopnosť naučiť sa lepšiu stratégiu výberu akcie.

V ďalších simuláciách sme testovali vplyv štruktúry neurónových sietí, konkrétne počtu neurónov na ich skrytej vrstve, na ich schopnosť učiť sa. V simuláciách testujúcich vplyv štruktúry ANN na výsledky agentov sme používali diskontný faktor hodnoty 0,5 a rýchlosť učenia 0,1. Konkrétne výsledky môžeme vidieť na obrázku číslo 8 a v tabuľke číslo 4.

Tabuľka 4: Sumarizácia dát simulácií pre rôzne štruktúry ANN

Štruktúra ANN	Celkový priemer	Celkový priemer %	Najvyšší priemer dátového okna	Najvyšší priemer dátového okna %
INPUT-25 HIDDEN-50	0,027490685	20,5498137	0,50772	0,301544
INPUT-25 HIDDEN-25	2,956852329	79,1370466	3,93272	98,6544
INPUT-25 HIDDEN-10	2,943234521	78,8646904	3,92704	0,985408

Na základe výsledkov získaných v tejto fáze sme vybrali konfiguráciu, ktorú budeme používať v ďalšej fáze nášho experimentu. Rozhodli sme sa pre konfiguráciu parametrov:

- Diskontný faktor s hodnotou 0.5
- Rýchlosť učenia 0.1
- Štruktúru ANN s 25 neurónmi na vstupnej vrstve a 25 neurónmi na skrytej vrstve

4.4 Fáza 2: Simulácie s komunikáciou

V tejto fáze experimentu sme znova porovnávali dve skupiny agentov. Tento krát to boli skupiny s rovnakými konfiguráciami učiacich sa mechanizmov. Rozdiel je v tom, že prvá skupina môže vo svojom prostredí iba meniť svoju polohu, zatiaľ čo druhej skupine je umožnené aj vysielat' signály ostatným agentom.

Výsledky simulácií porovnáваме rovnako ako v prvej fáze. Porovnáваме kontrolnú premennú, ktorou je priemerná získaná odmena na jeden časový krok.

V tejto fáze sme spúšťali simulácie s nasledovným nastavením:

- Do prostredia sme umiestnili 10 agentov.

- V prostredí sa nachádzalo 12 objektov typu *Jablko*. Na potrebu nastavenia tohto faktoru sme prišli pri predbežných testoch. Snažili sme sa ním dosiahnuť vhodný pomer odmeny za komunikačné akcie a odmeny za zbieranie ovocia. Ďalej sme sa snažili dosiahnuť, že sa agenti budú viac snažiť využívať komunikačné akcie. Toto by nenastávalo, keby bolo príliš veľa ovocia okolo nich.
- Políčko, na ktoré sme ovocie presúvali (nastáva pri poklese hodnoty atribútu *Množstvo* na 0), bolo vždy vybrané spomedzi voľných políčok, tesne za hranicou agentovho perzeptu. Toto zvýši pravdepodobnosť, že sa po presune ovocia v agentovom percepte ovocie nenachádza a ten sa musí rozhodovať, či sa pokúsi zistiť pozíciu nového ovocia od ostatných agentov alebo sa ďalšie ovocie pokúsi nájsť sám.
- Počiatočná poloha objektov na mape (ovocie a agent) je generovaná náhodne na začiatku každej simulácie.
- V každej simulácii sme agenta nechali spraviť 350000 krokov
- Keďže je model daného sveta podstatne náročnejší na učenie, zvýšili sme stupeň pravdepodobnosti exploračie tým, že sme spomalili rýchlosť rastu inverznej teploty.

Z vyššie uvedenej konfigurácie vyplýva, že hodnota odmeny v jednom časovom kroku je pre agentov z prvej skupiny (bez komunikácie) v intervale $<-1;2>$ a pre agentov z druhej skupiny (s komunikáciou) v intervale $<-1;3>$.

5 Implementácia

Keďže virtuálny svet TerraSimu je písaný v programovacom jazyku C#, rozhodli sme sa, že na implementáciu našich agentov, tiež použiť tento jazyk ako cestu najmenšieho odporu.

Ako vývojové prostredie sme používali Microsoft Visual Studio 2010.

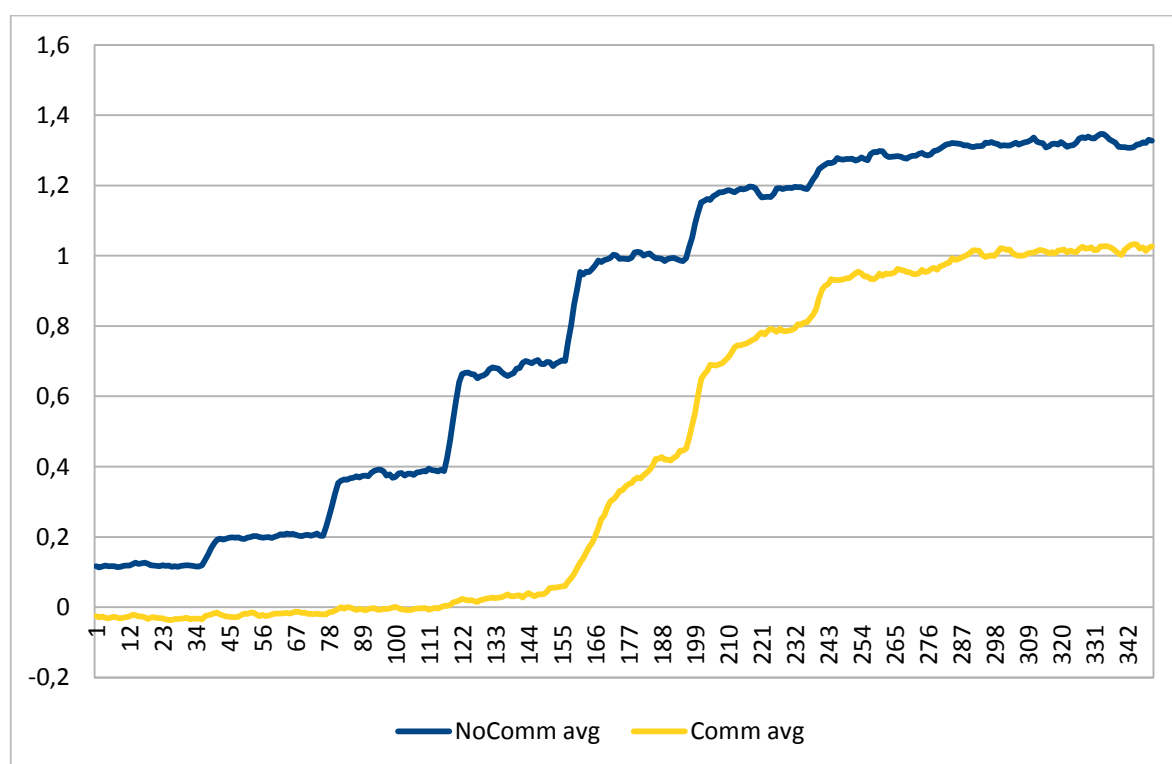
Na tvorbu našich neurónových sietí sme použili opensource framework NeuronDotNet, ktorý poskytuje interface na tvorbu a používanie rôznych typov neurónových sietí. Tento framework je tiež písaný v programovacom jazyku C#. Je vydany pod licenciou GPLv2. Stranka projektu sa nachadza na URL <http://sourceforge.net/projects/neurondotnet/>.

6 Výsledky

Ako sme už spomenuli vyššie, na základe výsledkov z prvej fázy experimentu, sme využívali nasledovnú konfiguráciu učiacich mechanizmov:

- Diskontný faktor s hodnotou 0.5
- Rýchlosť učenia 0.1
- Štruktúru ANN s 25 neurónmi na vstupnej vrstve a 25 neurónmi na skrytej vrstve

V grafoch a tabuľkách označujeme skupinu nepoužívajúcu komunikáciu skratkou *NoComm* a skupinu, ktorej je umožnené komunikovať označujeme skratkou *Comm*.



Obrázok 9: Graf zobrazujúci zmenu priemernej získanej odmeny počas behu simulácie

Tabuľka 5: Sumarizácia dát priemerných odmien vo fáze 2

	Celkový priemer	Celkový priemer %	Najvyšší priemer dátového okna	Najvyšší priemer dátového okna %
NoComm avg	0,82195246	60,731749	1,34656	78,218667
Comm avg	0,436265	35,9066	1,03316	50,829

Na obrázku číslo 9 vidíme graf zobrazujúci vývoj priemerného zisku odmeny oboch skupín, spolu s ich vývojom smerodajnej odchýlky. Tabuľky číslo 5 a 6 sumarizujú hodnoty zobrazené v grafoch. Na predložených údajoch jasne vidieť, že skupina nepoužívajúca komunikáciu má výrazne lepšie výsledky.

Tabuľka 6: Sumarizácia dát smerodajných odchýliek vo fáze 2

	Priemerná smerodajná odchýlka	Najvyššia smerodajná odchýlka dátového okna
NoComm STDEVP	0,026007348	0,059086702
Comm STDEVP	0,045489412	0,113159288

Počet krokov, ktorý agenti z komunikujúcej skupiny potrebovali na prejavenie výrazného pokroku ohľadom získavania odmien, bol skoro 5-krát väčší ako potrebovali agenti v skupine bez komunikácie. Tento fakt nás nezaskočil vzhľadom na to, že agenti sa potrebovali naučiť používanie väčšieho počtu akcií a model sveta, v ktorom sa pohybovali, respektíve, ktorý mohli na základe vykonávania akcií pozorovať a učiť sa ho, bol zložitejší.

Priemerná odmena skupiny bez komunikácie začala na hodnote 0,1. Jej hodnota opakovane nárazovo rástla v rovnakých časových periódach, až kým nedosiahla hodnotu 1,3, okolo ktorej začala oscilovať. Priemerná odmena komunikačnej skupiny mala na začiatku simulácie hodnotu -0,02. Rozdiel v počiatočných priemerných odmenách je tiež pochopiteľný, pretože agenti komunikačnej skupiny môžu získať negatívnu odmenu vo viacerých stavoch ako agenti bez komunikácie. Po tom ako priemerná odmena komunikačnej skupiny začala rásť, sa tento rast nezastavil až kým jej hodnota nezačala oscilovať okolo hodnoty 1.

Po ukončení analýzy získaných odmien sme sa začali venovať analýze komunikačných akcií. Ako sme spomínali v predchádzajúcich kapitolách, ukladali sme si dáta o používaní komunikačných akcií agentmi.

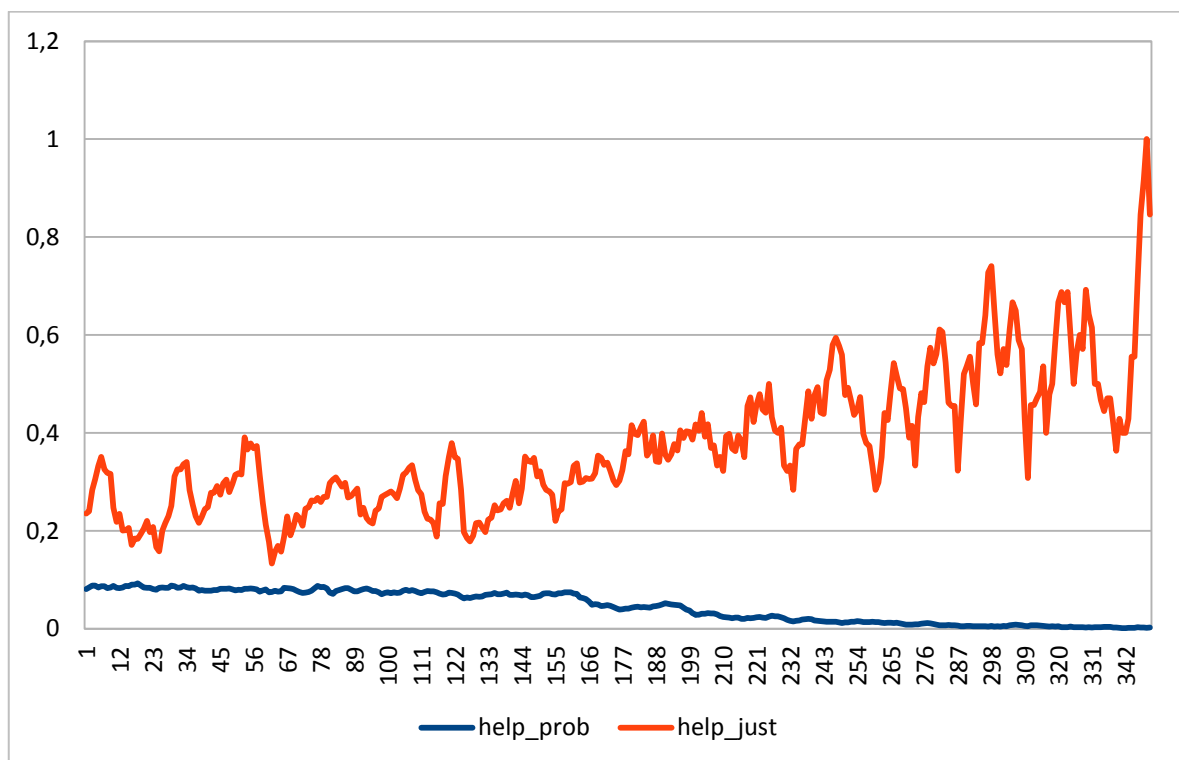
Ukladané dáta ohľadom volania akcie *Help*:

- s akou pravdepodobnosťou bola akcia v danom dátovom okne volaná (v grafoch a tabuľkách ďalej len *help_prob*)
- aké percento volaní akcie *Help*, bolo v danom dátovom okne použité správne. Za správne použitie akcie *Help* sme považovali situáciu, kedy sa v agentovom percepte

nenachádzalo žiadne ovocie, a zároveň sa v ňom nachádzali agenti, z ktorých žiadny nesignalizoval *ISeeFruit/ISeeNoFruit*.

Ukladané dáta ohľadom volaní akcií *ISeeFruit/ISeeNoFruit*:

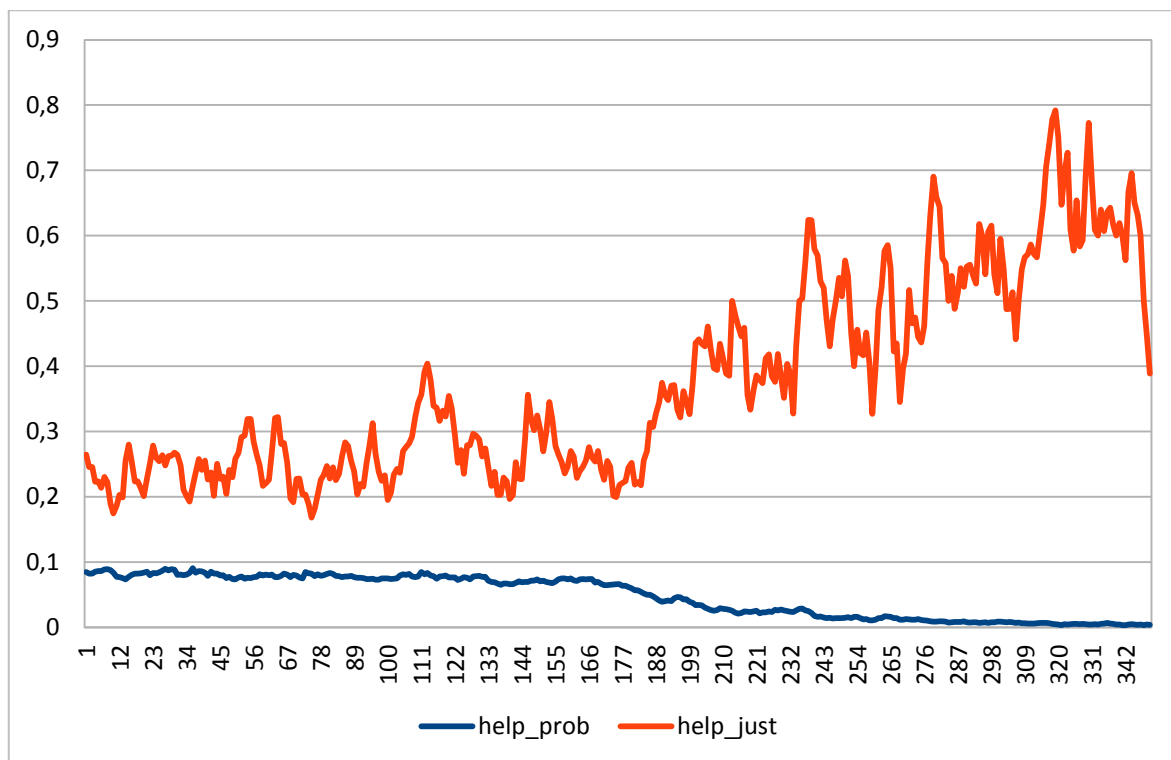
- s akou pravdepodobnosťou bola akcia v danom dátovom okne volaná (v grafoch a tabuľkách ďalej len *fruit_prob/noFruit_prob*)
- aké percento volaní týchto akcií bolo v danom dátovom okne použité správne. Za správne použitie akcie sme považovali situáciu, v ktorej agent správne signalizuje prítomnosť/neprítomnosť ovocia vo svojom percepte. Toto sa nevzťahovalo na ovocie, na ktorom agent stál, pretože toto ovocie bolo obsadené a tým pádom pre ostatných agentov nepoužiteľné.
- aké percento volaní týchto akcií bolo v danom dátovom okne vyslané ako odpoveď na predchádzajúce volanie *Help* iným agentom.



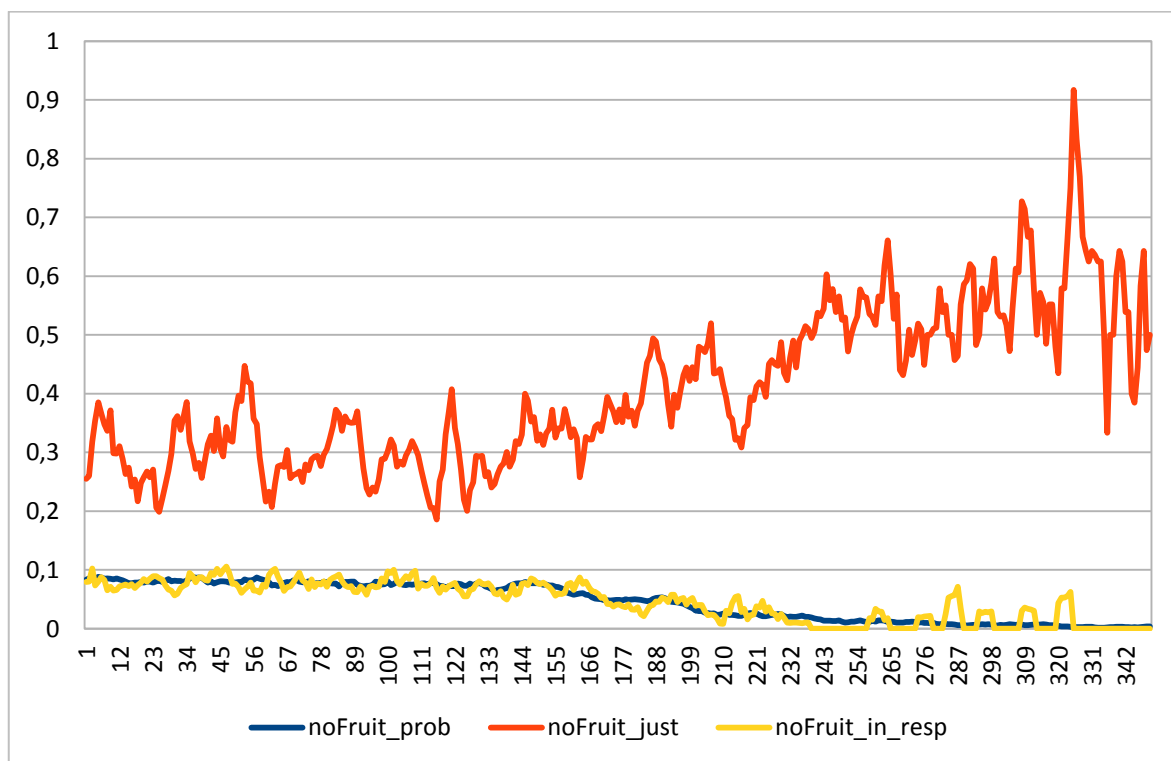
Obrázok 10: Vývoj dát volania príkazu *Help* počas behu simulácie (typ a)

Pri analýze dát komunikačných akcií sme spozorovali vytvorenie dvoch typov správania. Spriemerovaním týchto dát cez celú skupinu agentov, by sme dostali nič nehovoriace priemerné správanie s veľkou smerodajnou odchýlkou vývoja týchto dát. Pre potrebu našej analýzy uvádzame nasledovné dáta jednotlivých agentov, na ktorých vieme

ľahko pozorovať dôležité črty ich vývoja. V prípade záujmu môžete všetky hodnoty aj s priemermi nájsť v digitálnej forme v prílohe A.



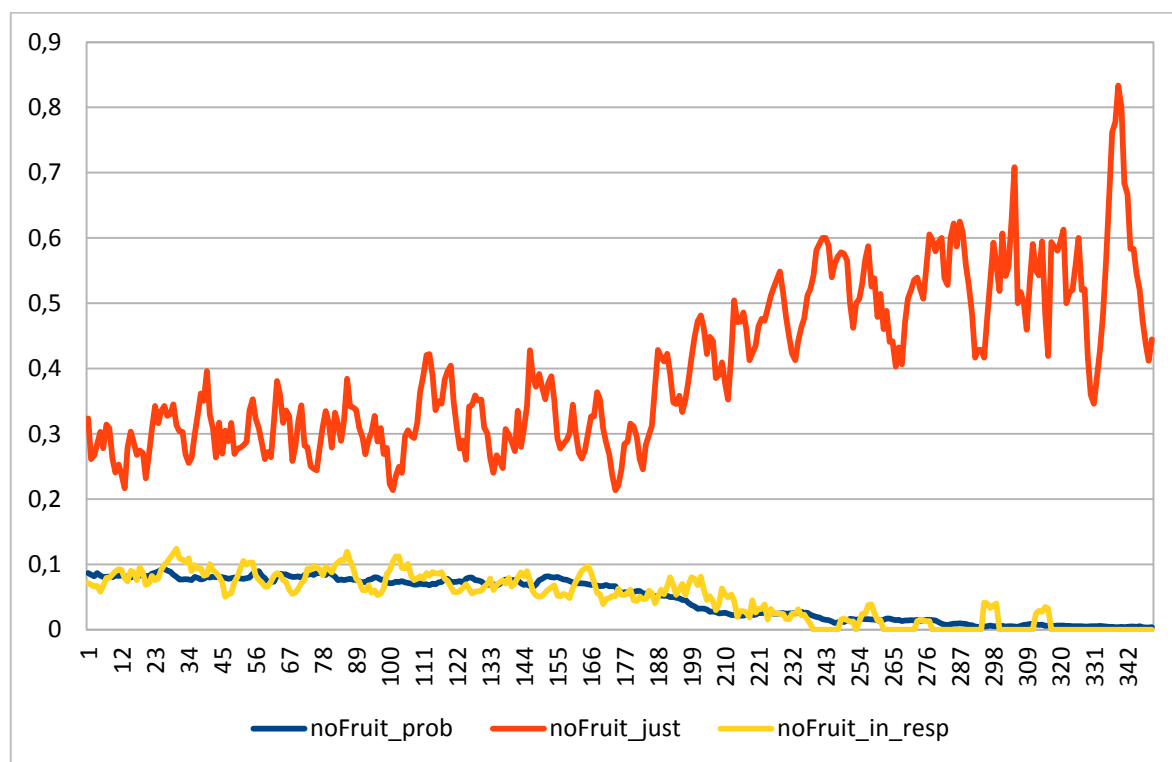
Obrázok 11: Vývoj dát volania príkazu Help počas behu simulácie (typ b)



Obrázok 12: Vývoj dát volania príkazu ISeeNoFruit počas behu simulácie (typ a)

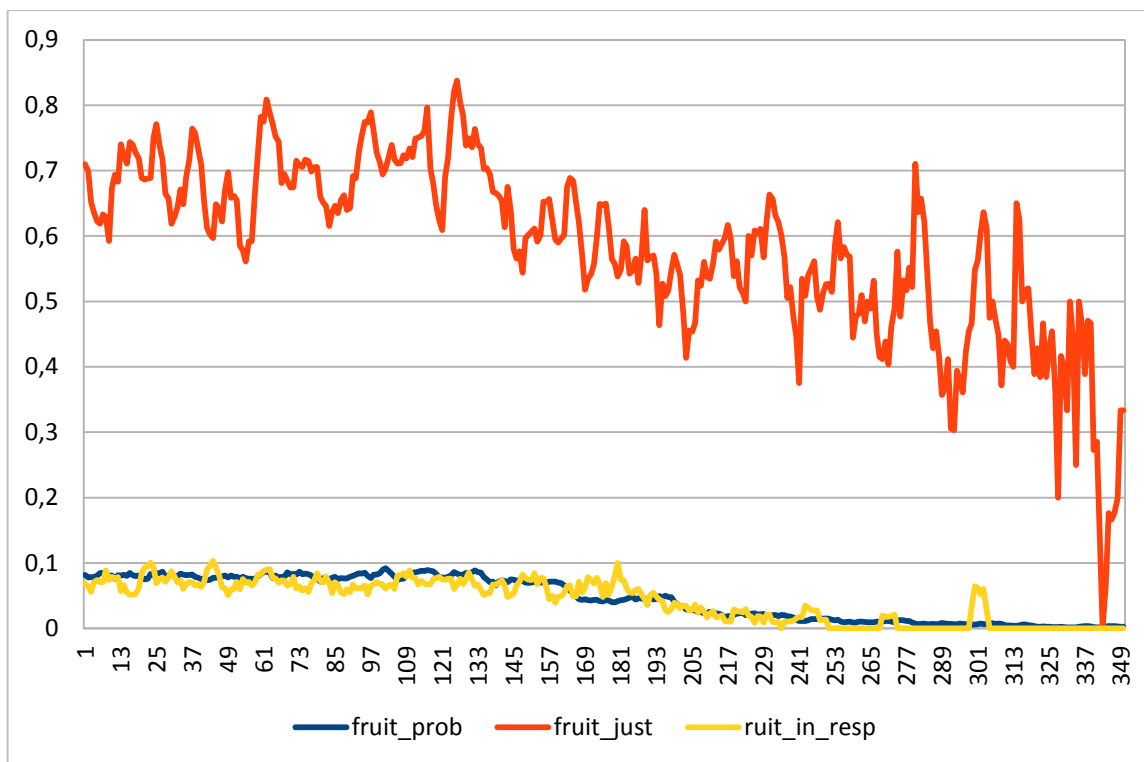
Na obrázkoch číslo 10 a 11 sú zobrazené grafy s hodnotami akcie *Help* agentov z oboch skupín. V obidvoch skupinách sme pozorovali rovnaký vývoj dát súvisiacich s volaním akcie *Help*.

Počiatočná pravdepodobnosť, s akou bola akcia *Help* vyberaná, sa pohybovala okolo 8,5 %, kvôli úplnej náhodnosti výberu. V priebehu celej simulácie táto pravdepodobnosť ani raz nepresiahla hodnotu 10%. Čím menej náhodným sa výber akcií stával, tým sa pravdepodobnosť na výber tejto akcie znižovala a na konci simulácie bola takmer nulová. Čo sa týka správnosti použitia akcie, tak jej úspešnosť mala stúpajúci trend. Výraznejšie zlepšenie v správnosti použitia sme pozorovali približne v rovnakom čase ako začala u komunikujúcich agentov stúpať priemerná odmena. Toto platilo pre obidva typy.

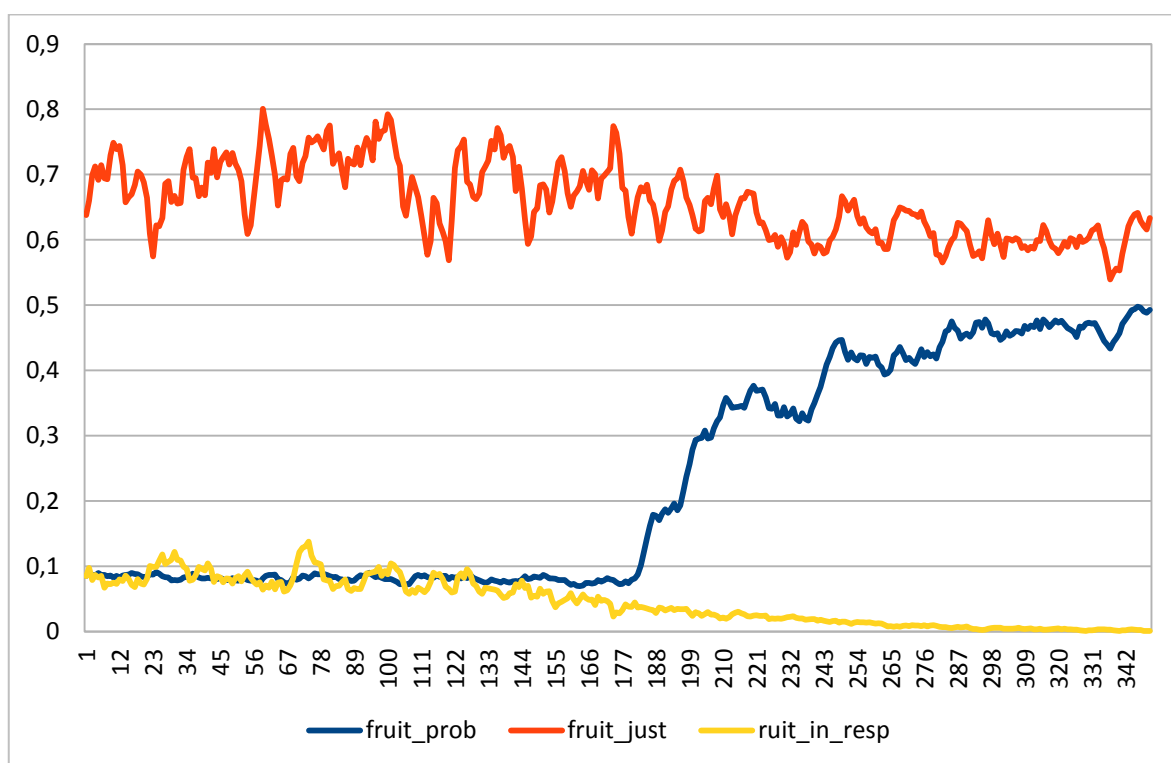


Obrázok 13: Vývoj dát volania príkazu ISeeNoFruit počas behu simulácie (typ b)

Rovnaký vývoj ako mali dáta akcie *Help*, mali aj dáta akcie ISeeNoFruit. Pri tejto sme navyše sledovali parameter, ktorý nám vyjadroval percentuálny podiel volaní, ktorým odpovedali na *Help* signál iných agentov v ich perceptive. Percentuálny podiel odpovedí sa hýbal okolo hodnoty 8 %. Táto hodnota začala výraznejšie klesať približne v rovnakom čase ako začala u komunikujúcich agentov stúpať priemerná odmena. Na konci simulácie sa jej hodnota blížila nule. Tento popis taktiež platí pre obidva typy ako môžete vidieť na grafoch číslo 12 a 13.



Obrázok 14: Vývoj dát volania príkazu ISeeFruit počas behu simulácie (typ a)



Obrázok 15: Vývoj dát volania príkazu ISeeFruit počas behu simulácie (typ b)

Delenie agentov na vyššie spomínané typy vzniklo na základe analýzy výstupov akcie *ISeeFruit*. Dôvodom bolo, že dátové výstupy sa veľmi líšili, čo môžeme pozorovať na obrázkoch číslo 14 a 15. Typy sa zhodovali iba vo vývoji percentuálneho podielu odpovedí na signál *Help*. Tento mal rovnaký priebeh ako vývoj odpovedí pri akcii *ISeeNoFruit*.

Výstup typu *a* má rovnaký vývoj pravdepodobnosti volania akcie *ISeeFruit*, ako akcia *ISeeNoFruit*. Správnosť použitia tejto akcie sa pohybovala okolo hodnoty 0,7 a neskôr začala klesať (viď obrázok 14).

Vo výstupe typu *b* správnosť použitia začne tiež klesať, ale iba veľmi pomaly. No na hodnotách percentuálneho podielu volaní vidíme prudký rast, ktorý na konci simulácie dosahuje hodnotu okolo 50 % (viď obrázok 15).

Možné vysvetlenie pre výskyt typu *b* je také, že sa títo agenti naučili správne používať (s pravdepodobnosťou 70 %) túto akciu. Mohli ju začať používať aj v blízkosti iných agentov, ktorí nevysielali signál *Help*, bez získavania negatívnej odmeny. Takto mohli začať túto akciu využívať namiesto akcie *DoNothing* s rovnakým výsledkom.

Treba poznamenať, že z 10 agentov v komunikujúcej skupine sa správanie typu *b* prejavilo iba u dvoch agentov.

7 Diskusia

V tejto kapitole sa venujeme zhrnutiu a analýze výsledkov, ponúkame nápady na zlepšenie budúceho výskumu a varujeme pred problémami, s ktorými sme sa stretli.

7.1 Volanie o pomoc

Naučiť sa správne používať akciu *Help* a následne sa správať podľa signálov ostatných agentov, za predpokladu, že signalizujú správne, bol základný predpoklad pre dosahovanie lepších výsledkov komunikujúcej skupiny. Zvládnutie tejto akcie by zaručovalo, že v prípade, že agent nemá vo svojom percepte žiadne ovocie, by mu umožnilo pohnúť sa smerom, v ktorom s veľkou šancou uvidí ovocie, alebo naopak nepohnúť sa smerom, v ktorom neúspešne hľadá iný agent.

Problém so zvládnutím tejto akcie agentmi je v tom, že odmena za správne použitie neprichádza hneď po vykonaní akcie, slúži iba ako cesta k cieľu. Vypropagovanie odmeny, ktorá neprichádza bezprostredne po vykonaní akcie, je zložitá úloha, ktorá vyžaduje veľa opakovaní úspešnej postupnosti krokov. Záznamy agentov o používaní tejto akcie ukazujú, že keď už ju použili, tak to bola väčšinou akcia použitá v správnom kontexte (agent nemal vo svojom percepte žiadne ovocie, ale videl agentov, ktorých sa mohol spýtať na pomoc). Napriek tomu sme v našom experimente nevytvorili vhodné podmienky pre agentov na to, aby sa naučili výhodnosť alebo zmysel používania tejto akcie, čo spôsobilo pokles jej volania.

7.2 Vidím/nevidím ovocie

Akcia *Help* bola pre agentov náročná na učenie ako sme spomenuli vyššie, pretože jej úspech či neúspech nebol ohodnotený odmenou bezprostredne po jej vykonaní. Toto neplatilo pre akciu *ISeeNoFruit* a *ISeeFruit*.

Odmena, ktorú agenti dostávali bola na základe nasledovných pravidiel:

- Ak signál, ktorý agenti vysielali, nepravdivo opisoval prítomnosť či neprítomnosť ovocia v ich percepte, boli penalizovaní negatívnou odmenou.

- Ak signál opisoval prítomnosť či neprítomnosť ovocia v agentovom percepte pravdivo, a v jeho percepte sa nachádzal agent, ktorý signalizoval *Help*, tak agent dostal odmenu.
- Ak signál opisoval prítomnosť či neprítomnosť ovocia v agentovom percepte pravdivo, ale v jeho percepte sa nenachádzal žiadny agent, ktorý signalizoval *Help*, tak agent nedostal žiadnu odmenu.

Za kritický problém v našom experimente považujeme malý počet volaní akcie *Help*. Po vykonaní akcii *ISeeNoFruit* a *ISeeFruit* agent síce dostane spätnú väzbu z prostredia, ale ako vidíme v pravidlách uvedených vyššie, pozitívnu odmenu dostane iba v prípade, že táto akcia odpovedá inému agentovi a ku všetkému musí opisovať svoj percept pravdivo.

V neprítomnosti volania signálu *Help* mohli agenti dostať za komunikačné akcie iba dva typy odmeny. V prípade, že použili správnu akciu, dostali nulovú odmenu. V opačnom prípade dostali odmenu zápornú. Agent zároveň nezmení svoju polohu. Jediný prípad, kedy je pre agenta výhodné nezmeniť svoju polohu (ak neberieme do úvahy volanie o pomoc) je, keď stojí na políčku s ovocím. Tu má na výber dve možnosti:

- Zavolá akciu *DoNothing*, za ktorú vždy dostane odmenu 0
- Zavolá komunikačnú akciu, za ktorú (v našom experimente) dostal
 - s veľmi malou pravdepodobnosťou odmenu 1
 - s veľkou pravdepodobnosťou odmenu 0 alebo -1.

Ak by vyberal správne akcie v 50 % prípadov, tak stále sa priemerná získaná hodnota v tomto scenári pohybovala okolo hodnoty -0,5.

Z tohto vyplýva, že používanie komunikačných akcií v neprítomnosti (alebo len veľmi zriedkavej prítomnosti) volania akcie *Help*, nie je pre agentov výhodné.

7.3 Nápady a ďalšie smerovanie experimentu

Ako spomíname vyššie, za kritickú časť nášho experimentu považujeme to, že agenti nevykonávali akciu *Help*. To nastalo v dôsledku toho, že agenti nedostávali odmenu bezprostredne po vykonaní akcie. Jednoducho povedané, agenti neboli motivovaní vyhľadávať pomoc. Túto motiváciu u agentov by sme mohli zvýšiť tak, že by sme im dávali malé odmeny (oveľa menšie, ako za zbieranie ovocia) v prípade, že by svojou akciou zmenšili vzdialenosť od miesta, z ktorého agenti signalizovali *ISeeFruit*, respektíve

zvážili vzdialenosť od miesta, z ktorého agenti signalizovali *ISeeNoFruit*. Popríklad by mohli vymyslieť inú alternatívu, ktorou by sme zvýšili pravdepodobnosť volania akcie *Help*.

Ďalšia zmena, ktorá by mohla pozitívne ovplyvniť množstvo volaní akcie *Help*, je upravenie pravidiel v našej reward funkcii. Momentálne je agent penalizovaný hodnotou -1, ak volá *Help*, a v jeho percepte sa nachádza ovocie, alebo agent signalizujúci prítomnosť/nepítomnosť ovocia.

Doteraz sme sa zaoberali iba dôvodmi, ktoré spôsobili nízky zisk odmeny u komunikujúcich agentov. Poďme sa pozrieť na dôvody, prečo mali agenti bez komunikácie také dobré výsledky.

Do úvahy pri úspechu nekomunikujúcich agentov musíme brať nasledovné faktory:

- veľkosť sveta, v ktorom sa agenti pohybovali
- celkový počet agentov
- celkový počet objektov ovocia

Rozmery sveta, po ktorom sa agenti mohli pohybovať boli 13x13 políček. Každý agent videl plochu 5x5 políček. Na kompletne pokrytie celej tejto plochy nám stačí 9 agentov. My sme ich v experimente mali 10. Z tohto vyplýva, že priemerná dĺžka časového okna, v ktorom sa dané ovocie nenachádzalo v percepte žiadneho agenta, bola pravdepodobne veľmi malá.

V našom experimente sme mali vo svete 12 objektov ovocia. Pri rovnomernom rozložení ovocia po mape vychádza 1 ovocie na 14 políček, čo znamená, že každý agent, pri dokonalom pokrytí mapy, v danom momente môže vidieť skoro 2 objekty ovocia.

Z týchto faktov o našom prostredí vyplýva, že s veľkou pravdepodobnosťou a často, sa v agentovom percepte nachádzalo ovocie. Za takýchto podmienok bolo pre agenta výhodnejšie nepoužívať komunikačné akcie, ale pohnúť sa smerom k ovociu. Tým pádom boli komunikujúci agenti v nevýhode, pretože sa museli naučiť používanie viacerých akcií. Z 100000 náhodných akcií s rovnomernou distribúciou komunikujúci agenti každú vykonajú približne 8300-krát a nekomunikujúci 11100-krát, čo je približne o 33 % viac.

V našom experimente sme príliš dbali na detaily učiacich sa mechanizmov a návrh prostredia ostal v úzadí. Tomuto sa musíme v budúcnosti vyvarovať.

Záver

V tejto práci sme porovnávali úspešnosť dvoch skupín kognitívnych agentov vo virtuálnych prostrediach s identickou konfiguráciou. Agenti sa za pomoci učenia posilňovaním snažili maximalizovať odmenu, ktorú z prostredia získavali na základe vykonávania akcií.

Agenti jednej skupiny mali možnosť navzájom komunikovať a agenti druhej skupiny nie.

Cieľom tejto práce bolo potvrdiť našu hypotézu, že pri konfigurácii, akú sme popísali v našom experimente. Túto hypotézu sa nám potvrdiť nepodarilo, pretože agenti v skupine ktorá mala možnosť komunikovať, dosahovali výraznejšie horšie výsledky ako agenti bez možnosti komunikovať.

Použitá literatúra

- Benc, L. (2011). *Simulované prostredie pre učiacich sa agentov. Diplomová práca FMFI UK*. Bratislava.
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 237-285.
- Mizutani, E., & Dreyfus, S. E. (1998). Totally Model Free Reinforcement Learning by Acto-Critic Elman : Networks in Non-Markovian Domains. *IEEE World Congress*.
- Oliphant, M., & Batali, J. (1997). *Learning and the Emergence of Coordinated Communication*. San Diego.
- Perfors, A. (2002). Simulated Evolution of Language: a Review of the Field. *Journal of Artificial Societies and Social Simulation* vol. 5, no. 2.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning Internal Representations by Error Propagation. In *Cognitive Modeling* (s. 673-695). Cambridge: MIT Press.
- Russell, S., & Norvig, P. (2002). *Artificial Intelligence - A Modern Approach*. Prentice Hall.
- Steels, L. (2011). Exploring Cultural Language Evolution. In L. Steels, *Experiments in Cultural Language Evolution*. Amsterdam: John Benjamins.
- Sutton, R. S., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Príloha A

CD médium

Na priloženom CD sa nachádza text diplomovej práce v elektronickej podobe. Okrem toho sú na ňom zdrojové kódy a výstupné dáta simulácií.