

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Varianty vězňova dilematu
Modely v NetLogu
Diplomová práce

Autor: Jiří Spejchal

Studijní obor: IM2

Vedoucí práce: doc. RNDr. Kamila Štekerová, Ph.D.

Hradec Králové

duben 2016

Prohlášení

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedených zdrojů.

V Hradci Králové dne

.....

Jiří Spejchal

Poděkování

Chtěl bych poděkovat vedoucí diplomové práce doc. RNDr. Kamile Štekerové, Ph.D. za cenné připomínky, vstřícnost při vedení práce a čas, který mé práci věnovala.

Anotace

Diplomová práce se zaměřuje na varianty vězňova dilematu. Pro zvolené varianty jsou vytvořeny odpovídající modely v multiagentním prostředí NetLogo. Modely jsou použity k experimentování se strategiemi, které jsou v rámci práce navrženy nebo využity z citovaných zdrojů. Pro experimenty je využit nástroj BehaviorSpace, který je součástí NetLoga. V teoretické části je věnován prostor základním pojmům a principům z teorie her a multiagentního modelování. Rovněž jsou popsány zvolené varianty vězňova dilematu. Praktická část obsahuje návrh a implementaci modelů a strategií, včetně ukázek kódu. Následuje analýza výsledků experimentů a úspěšnosti strategií v porovnání s ostatními. Jsou uvedeny také vstupní parametry a kritéria pro hodnocení úspěšnosti strategií. Poslední kapitola se věnuje diskuzi ohledně výsledků experimentů s možnými příčinami a dopady principů chování strategií. Jsou uvedeny také další možnosti zkoumání. Práce je doplněna obrázky, tabulkami a grafy, včetně ukázky populačních struktur modelu prostoro-
vého vězňova dilematu.

Klíčová slova

Teorie her, vězňovo dilema, varianty, multiagentové modely, NetLogo, strategie, simulace.

Annotation

Title: Versions of the prisoner's dilemma

Subtitle: Models in NetLogo

The diploma thesis deals with versions of the prisoner's dilemma. Corresponding models are created in multi-agent environment NetLogo. The models are used for experiments with strategies that are designed within the thesis or used from cited sources. The BehaviorSpace is used for the experiments. Theoretical part introduces basic terms and principles about the game theory and multi-agent modelling. There are also described the versions of the prisoner's dilemma. Practical part specifies design and creation of the NetLogo models and strategies with code samples. Then follows results of the experiments with analysis. Strategies are compared to others. Input parameters and criteria for strategies assessment are also given. Finally, follows discussion about experiment results with possible reasons and impacts of behavior principles of the strategies. There are also mentioned possibilities for other research. The thesis is enriched by the pictures, tables and graphs, including some of the population structures of the spatial prisoner's dilemma model.

Keywords

Game theory, prisoner's dilemma, versions, agent-based models, NetLogo, strategy, simulation.

Obsah

1	Úvod.....	1
2	Teoretická část.....	2
2.1	Zkoumání her pomocí agentových modelů	2
2.1.1	Základní principy teorie her.....	2
2.1.2	Klasifikace her	5
2.1.3	Klasifikace strategií.....	7
2.1.4	Multiagentové modely	9
2.1.5	Simulace.....	12
2.1.6	NetLogo.....	13
2.2	Věžňovo dilema, jeho varianty a aplikace	18
2.2.1	Klasické věžňovo dilema.....	18
2.2.2	Opakované věžňovo dilema.....	21
2.2.3	Sosisova hra v kibucech	23
2.2.4	Boj o zdroje.....	23
2.2.5	Dvourozměrné prostorové hry	25
3	Praktická část.....	27
3.1	Návrh modelů.....	27
3.1.1	Sosisova hra v kibucech	27
3.1.2	Boj o zdroje.....	29
3.1.3	Dvourozměrné prostorové věžňovo dilema (SPD).....	30
3.2	Implementace	33
3.2.1	Sosisova hra v kibucech	33
3.2.2	Boj o zdroje.....	39
3.2.3	Dvourozměrné prostorové věžňovo dilema	44
3.3	Experimenty	57
3.3.1	Sosisova hra v kibucech	58
3.3.2	Boj o zdroje.....	61
3.3.3	Dvourozměrné prostorové věžňovo dilema	64
3.4	Diskuze.....	73
3.4.1	Sosisova hra v kibucech	73
3.4.2	Boj o zdroje.....	74
3.4.3	Dvourozměrné prostorové věžňovo dilema	76
4	Závěr.....	78
5	Seznam použitých zdrojů.....	81
6	Přílohy.....	86

Seznam obrázků

Obrázek 1: Schéma prostředí s více agenty	10
Obrázek 2: Ukázka prostředí NetLoga	14
Obrázek 3: Tvorba koalic ve dvourozměrné hře	26
Obrázek 4: Nestabilizovaná populace strategií na konci hry, 10 000 hráčů (model SPD)	72
Obrázek 5: Barevná legenda strategií	a
Obrázek 6: Model SPD, 20 hráčů – stabilizované struktury (po 3 a 7 kolech)	a
Obrázek 7: Model SPD, 100 hráčů – stabilizovaná a nestabilizovaná struktura (14 a 1 000 kol)....	a
Obrázek 8: Model SPD, 900 hráčů – stabilizovaná a nestabilizovaná struktura (41 a 1 000 kol)....	b
Obrázek 9: Model SPD, 2 500 hráčů – stabilizovaná a nestabilizovaná struktura (16 a 1 000 kol) b	

Seznam tabulek

Tabulka 1: Příklad trestů hráčů vězňova dilematu.....	19
Tabulka 2: Obecné vyjádření vězňova dilematu	20
Tabulka 3: Obecná reprezentace výše trestů (užitků)	20
Tabulka 4: Výplatní matice hry Jestřáb-holubice	24
Tabulka 5: Výplatní matice modifikované hry Jestřáb-holubice.....	24
Tabulka 6: Výplatní matice modelu dvourozměrného prostorového vězňova dilematu	31
Tabulka 7: Populace strategií na konci hry (evoluční model SPD)	65
Tabulka 8: Časy vyhynutí strategií a stabilizace populací (evoluční model SPD).....	67
Tabulka 9: Kumulativní skóre strategií (evoluční model SPD).....	69
Tabulka 10: Kumulativní skóre strategií (neevoluční model SPD).....	70

Seznam grafů

Graf 1: Průměrné a mediánové reálné zisky Sosisovy hry	59
Graf 2: Absolutní přírůstky reálných zisků Sosisovy hry	60
Graf 3: Průměrné a mediánové skóre modelu boje o zdroje	62
Graf 4: Vývoj populací strategií v čase u 10 000 hráčů	68
Graf 5: Průměrné skóre hráče na jedno kolo existence (evoluční a neevoluční model)	71
Graf 6: Mediánové skóre hráče na jedno kolo existence (evoluční a neevoluční model)	71

1 Úvod

Teorii her je možné aplikovat na spoustu situací dnešní doby, ať se již jedná o rozhodování jednotlivců s několika málo strategiemi, či konfliktní rozhodovací problémy skupin ve vzájemné interakci za využití mnoha strategií, přičemž různé skupiny mohou mít k dispozici rovněž různé možnosti rozhodování. Hodnotový systém dnešní společnosti se promítá do rozhodování jednotlivců. To může vykazovat značně dynamický ráz a tak je nasnadě, aby metody, které jsou využity pro analýzu rozhodování v určitých situacích, byly adaptabilní a dokázaly reflektovat rychlé změny v preferencích. Rychlejší identifikace změny preferencí v rozhodování může přinést výhodu zejména v oblasti ekonomické.

Za prvotní událost, která vedla k rozvoji teorie her jakožto samostatné matematické disciplíny, je považováno vydání knihy Johna von Neumanna a Oskara Morgensterna *Theory of Games and Economic Behavior* (Teorie her a ekonomického chování) v roce 1944 [1]. Od okamžiku vzniku teorie her jako samostatné vědní disciplíny již uplynulo několik desítek let a za tu dobu bylo publikováno spousta článků a knih. Koncepty teorie her a její aplikace se rozšířily do spousty oblastí a možnosti stále nejsou vyčerpány. Naopak, s rychlým rozvojem výpočetních technologií vyvstávají nové možnosti analýzy situací z oblasti teorie her za využití modelů a simulací. Výpočetní technologie dnešní doby mohou analyzovat mnohem více možností při experimentální změně parametrů, než by tomu bylo možné v případě řešení s využitím exaktních matematických metod.

Cílem této diplomové práce je popsat varianty a aplikace vybrané herní situace, a sice věžňova dilematu, vytvořit odpovídající modely v multiagentním prostředí NetLogo a použít tyto modely k experimentování se strategiemi, které jsou v rámci této práce navrženy nebo použity z citovaných zdrojů. Využita byla verze NetLogo 5.2.1, která byla aktuální v době tvorby modelů. V teoretické části je věnován prostor základním pojmům a principům z oblasti teorie her a tvorby multiagentních modelů. Rovněž jsou popsány zvolené varianty věžňova dilematu. Praktická část je zaměřena na návrh modelů, implementaci, experimenty a diskuzi. V rámci experimentů jsou popsány také vstupní parametry a kritéria pro hodnocení strategií. Pro experimenty byl využit nástroj BehaviorSpace, který je součástí NetLoga.

2 Teoretická část

Tato kapitola objasňuje základní teoretické koncepty a vymezení pojmů souvisejících s teorií her a agentovými modely. Kapitola 2.1 rovněž obsahuje základní rozdělení, podle kterého lze agentové modely a strategie klasifikovat. Kapitola 2.2 popisuje vězňovo dilema včetně možných strategií, variant dilematu a aplikací.

2.1 Zkoumání her pomocí agentových modelů

Multiagentové modely jsou jednou z možných variant pojetí modelování a simulace situací v teorii her. Kapitola 2.1 objasňuje základní principy teorie her a klasifikaci her a strategií.

2.1.1 Základní principy teorie her

Teorie her je matematickou disciplínou. Klasická teorie her zkoumá rozhodovací situace, kterých se účastní jeden či více jedinců nebo organizovaných skupin a u každého je požadováno, aby učinil nějaké rozhodnutí [2]. Rozhodnutí závisí na zvolené strategii účastníka hry [3]. Teorie her pak může usnadnit výběr strategie a lépe dosáhnout požadovaného výsledku [2]. Jak uvádí [3], konflikt nastává, pokud se rozhodovací situace účastní alespoň dva hráči.

Podle [2] je předmětem zkoumání hra, která je dána souborem pravidel. [5] také definují hru jako formální vyjádření zkoumané situace. Ve hře bývají modelováni hráči, jim dostupné informace a jejich rozhodovací preference [5]. Důležité je, že hráči znají strategie, které mohou v průběhu hry zvolit. Nemusejí však nutně znát důsledky, jaké plynou z volby strategie v konkrétním kole hry, byť mohou znát předchozí rozhodnutí soupeře. Nevědí tedy jistě, jaký zisk přinese volba konkrétní strategie v jednom kole hry, neboť závisí i na strategii soupeře. Mohou však zisk odhadovat, stejně jako by mohli po velkém počtu kol s určitou pravděpodobností odhadovat, jakou strategii protihráč využívá v rámci vícekolové hry. Hráči znají více možných dopadů volby konkrétní strategie na jejich výplatu v souvislosti s volbou strategií soupeře.

Jak zmiňuje [3], hra v teorii her nezahrnuje pouze klasické stolní hry typu šachy, dáma a karetní hry, ale také oblasti ekonomické, politické, psychologické a další společenské systémy. Hra může být také vojenského charakteru [3]. Vojenská hra bývá většinou konfliktní situací [3]. Ekonomickou hrou je například rozhodování několika firem v konkurenčním prostředí při změně cen obdobných produktů [3]. V politických hrách se mohou dostávat do konfliktu politické strany. Ty mohou vytvářet také nekonfliktní koalice.

Jak uvádí [4], hráči jsou účastníci rozhodovací situace, tedy hry. V rámci hry volí určité strategie a dílčí rozhodnutí [4]. Hráči se rozhodují také na základě informací, které jsou jim dostupné. Mohou se například rozhodovat na základě předchozích tahů soupeře. Hráči se rozhodují sami za sebe a jsou racionální, leč jejich rozhodování a volba strategie jsou do určité míry závislé na soupeřích [5]. Pokud hráči spolupracují nebo utvářejí koalice, tak se rozhodují s ohledem na prosperitu skupiny nebo koalice [5].

Hráčů může být ve hře více, obvykle minimálně dva. Jak vysvětlují [5], hra s jediným hráčem je obvykle chápána jako rozhodovací problém, například výběr nového kola na internetu s více možnými strategiemi. [3] popisuje jako účastníka rozhodovací situace také neuvědomělý systém, například příroda. Neuvědomělý systém nesleduje cíl (na rozdíl od hráčů), a jeho jednání je možné předpovídat pouze s určitou pravděpodobností nebo může být zcela nejisté [3].

Racionalitu hráče v teorii her lze chápat jako schopnost samostatného rozhodování, volby strategií, schopnost zpětně posoudit důsledky svého rozhodnutí a důsledky rozhodnutí soupeřů, popřípadě odhadnout pravděpodobnost důsledků svých rozhodnutí v jednom kole hry, viz [6]. Takovéto posuzování hráč činí také na základě ohodnocení jeho volby (výplata) v jednom kole [7]. Hráč má k dispozici určité volby, kterými může ovlivnit průběh hry. Podle [8] je jedním ze základních problémů teorie her stanovení výsledků hry, jestliže jsou jejími účastníky právě racionální hráči. Racionální hráč je schopen posuzovat výhry a vytvářet si preferenční vztahy mezi strategiemi [6] a snaží se také maximalizovat svou výplatu (zisk, užitek, ...), případně výplatu skupiny spolupracujících hráčů [7].

Jak uvádí [9], výplata je výsledkem hry z pohledu konkrétního hráče. Výplata hráčů je vyjádřena kvantitativně, například v penězích nebo ve velikosti užitku [9], v případě věžňova dilematu se měří délka trestu [10]. Způsob rozdělení výplaty v souvislosti s interakcí jednotlivých hráčů a jejich strategií v jednom kole hry bývá pevně stanoven [10]. U hry, které se účastní dva hráči, může být výplata vyjádřena ve formě matice [10] nebo výčtu výplat hráčů v pořadí od prvního hráče do posledního hráče [11].

S výplatami hráčů také souvisí pojmy Nashova rovnováha a Paretovská efektivnost, které vysvětluje například [12]. Koncept Nashovy rovnováhy se využívá pro analýzu a predikci výsledku interakce několika subjektů v oblasti sociálních věd. V případě jednoho kola hry je to taková kombinace zvolených strategií, kdy žádný hráč nemůže jednostrannou změnou své strategie v daném kole získat vyšší výplatu. U hry pro dva hráče je dosaženo Nashovy rovnováhy, pokud si oba hráči zvolí své nejlepší (dominantní) strategie. Jestliže

hráči nemají k dispozici jednu jednoznačně nejlepší strategii, potom může nastat více rovnovážných situací [12].

Paretovská efektivnost, viz [13], se v teorii her vyskytuje ve hrách o dvou a více hráčích. V teorii her se tímto pojmem označuje situace, kdy není možné zvýšit výplatu jednoho z hráčů, aniž by se zároveň výplata jakéhokoliv jiného hráče nesnížila. Ze společenského hlediska tak všichni hráči realizují nejvyšší možný užitek. Pokud by někdo chtěl získat lepší postavení, musel by tak učinit pouze na úkor někoho jiného. Dlouhodobě, ve vícekolových hrách, je možné, aby se k tomuto optimu všichni hráči přiblížili nebo jej dosáhli prostřednictvím spolupráce [13].

Strategie [14] je způsob chování hráče ve hře v jednotlivých kolech, který může také vycházet z historických rozhodnutí v předchozích kolech. Podle strategie, kterou si hráči volí, je možné rozdělit výplatu. Strategií může být více a je možné podle různých způsobů rozhodování hráčů hry rovněž klasifikovat [14].

Problémy z teorie her je možné studovat pomocí exaktních matematických metod. Avšak u některých úloh, například při zkoumání konfliktních situací v početné populaci, se jako účinnější metoda jeví využití informačních technologií, mj. simulací a multiagentových modelů.

Obecný postup pro modelování zkoumané situace z oblasti teorie her, který uvádí [4], je následující:

1. Vymezení problému, vč. identifikace klíčových faktorů.
2. Formulace strategií, pokrytí všech možných scénářů.
3. Vytvoření modelu.
4. Provedení experimentů.
5. Analýza výsledků.

Jednotlivé kroky mohou být názorně vysvětleny na hře „Kámen, nůžky, papír“. Vymezením problému se rozumí následující popis hry: Jedná se o hru pro dva a více hráčů. Pro hru jsou využity ruce a tři zástupné symboly, kámen představuje sevřenou pěst, nůžky rozevřený ukazováček a prostředníček a papír rozevřená dlaň s nataženými prsty. Jedním z klíčových faktorů hry „Kámen, nůžky, papír“ je, že si oba (všichni) hráči současně zvolí strategii v daném kole. V případě vícekolové varianty hry se hráči se mohou rozhodovat na základě znalosti historie předchozích tahů soupeřů.

Formulace strategií a pokrytí všech možných scénářů se odvíjí od vymezení problému. V každém kole jsou na výběr tři dílčí strategie, a to kámen, nůžky a papír. Kámen vítězí nad nůžkami, papír nad kamenem a nůžky nad papírem. Pokud jsou zvoleny stejné strategie, je kolo nerozhodné a je nutné jej opakovat, popřípadě jsou vyřazeni hráči, kteří měli méně úspěšnou strategii a hru opakují ti hráči, kteří zvolili nejúspěšnější strategii kola současně.

Vytvoření modelu obnáší přenesení stanovených pravidel do zvoleného prostředí pro modelování a experimenty, například NetLogo, viz [15]. Hráči se v daném kole rozhodují nezávisle na současné volbě strategie soupeřů, vyhodnocení je provedeno až po zvolení strategie všemi hráči.

Experimenty mohou zahrnovat například dva hráče a pevnou volbu strategie jednoho z nich. Pevnou volbou strategie může být například kombinace kámen, nůžky, papír ve třech po sobě jdoucích kolech opakované hry „Kámen, nůžky, papír“. Soupeř volí strategie náhodně na základě pseudonáhodně generovaných čísel. Tato situace je modelována opakovaně a úspěšnost strategií je vyhodnocena v rámci analýzy výsledků. Součástí analýzy může být také stanovení četností vyhraných her pro jednotlivé strategie.

2.1.2 Klasifikace her

Kapitola 2.1.2 obsahuje základní možnosti rozdělení typů her, podle kterých je možné hry klasifikovat. Rozdělení je využito v dalších částech práce. Klasifikace rovněž mohou posloužit pro lepší pochopení a vhled do problematiky teorie her a její šíře.

Prvním kritériem je počet hráčů, viz [3]. Hry s jedním hráčem jsou klasickou rozhodovací situací. Pokud se však hry zúčastní dva a více hráčů, může mezi nimi nastat konflikt v rozhodování. Ten je způsoben tím, že oba hráči sledují rozdílné cíle. Konflikt může nastat také v případě zcela rozdílných strategií, u kterých nevznikne vzájemná spolupráce.

Dva hráči mohou navzájem spolupracovat a u her s vyšším počtem hráčů mohou vznikat spolupracující skupiny, popřípadě samostatný hráč, který nemá zájem o spolupráci a může nakonec získat i výhodnější situaci na úkor spolupracujících jedinců [3]. Podle úrovně spolupráce [3] se hry dělí na

- nekooperativní,
- kooperativní, které se dále dělí na
 - hry s možností rozdělení výher,
 - hry bez možnosti rozdělení výher [3].

V případě kooperativních her mnohou hráči spolupracovat nebo uzavírat různé aliance spolupracujících hráčů [16]. Nekooperativní hry tuto vlastnost u hráčů nevykazují. Spolupráce může hráče zvýhodňovat, ale může také dojít k situaci, kdy je výhodnější zvolit nekooperativní strategii na úkor kooperativní strategie i v krátkodobějším horizontu. U kooperativních her se mohou hráči dopředu dohodnout o vzájemné spolupráci. Tato situace by pro racionální hráče mohla přicházet v úvahu, jestliže oběma stranám uzavřená dohoda přinese větší výhru než v případě, že by dohodu neuzavřeli.

O této situaci pojednává tzv. neantagonistická hra dvou hráčů [3]. Při uzavření dohody mezi hráči je rozdíl mezi tím, zda se mohou dohodnout i na přerozdělení společné výhry nebo pouze na společné strategii (hry s nemožností rozdělení výher) [3]. Jestliže se jedná o hru třech a více hráčů, může ze vzájemné spolupráce dvou a více jedinců profitovat jiný jedinec s nekooperativním chováním.

Podle způsobu dělení výplaty [16] je rozdělení her následující:

- hry s nulovým součtem,
- hry s nenulovým součtem
 - konstantní součet,
 - nekonstantní součet.

Hry s nulovým součtem představují situaci, kdy jeden hráč získá (vyhraje) to, co druhý hráč prohrál a součet výplat je roven nule [16]. Může se jednat například o hru Dáma, jeden hráč prohrál a druhý hráč vyhrál. Oproti tomu mohou mít hry s nenulovým součtem jiný celkový užitek než jen nula [16]. Příkladem je vězňovo dilema, kdy spolupráce obou hráčů přináší jiný celkový užitek, než v případě, že oba hráči nespolečně spolupracují.

Hry s nenulovým součtem mohou mít nejen nekonstantní součet, ale také konstantní součet [16]. Konstantní součet znamená, že celkový užitek všech hráčů je stejný i v případě, že jednotliví hráči mají užitek různý [17]. Reprezentativní situací hry s konstantním součtem může být rozdělení stejné výplaty všem hráčům [17].

Podle způsobu rozhodování hráčů, viz [18], je možné rozlišit

- simultánní hry a
- sekvenční hry.

Účastníci simultánní hry neznají rozhodnutí ostatních hráčů, hráči se rozhodují současně ve stejném kole hry. Sekvenční hry jsou takové hry, ve kterých se hráči rozhodují střídavě.

Typickým příkladem sekvenční hry jsou šachy. Hráči znají předchozí tahy soupeře, a proto může být jejich rozhodování ovlivněno rozhodováním soupeře, ať už by upravili své rozhodnutí vědomě či podvědomě. Nemusejí však znát vícekolovou strategii, kterou si jejich protivník zvolil. Tu by mohli, pokud vůbec, odhalit až po velkém počtu kol hry [18].

Podle pravidel hry [16] se hry dělí na

- symetrické hry,
- asymetrické hry.

Jak uvádí [19], v symetrických hrách všichni hráči využívají stejnou sadu strategií a výplata hráčů závisí jen na zvolených strategiích, nikoliv na hráči, který si strategii zvolil. Pro všechny hráče platí stejná pravidla [19]. Pokud si dva různí hráči symetrické hry zvolí tutéž strategii, tak jim přinese stejný užitek, zatímco hráčům asymetrické hry stejná strategie může přinést různý užitek [16]. Strategie hráčů asymetrické hry mohou být různé nebo nemusejí mít jedinou společnou strategii [16]. Asymetrickou hrou může být například vstup nové firmy na trh [16]. Různé firmy volí různé strategie při vstupu na stejný trh [16]. Příkladem symetrické hry může být vězňovo dilema [19].

Podle počtu kol [2] je možné rozlišit

- jednokolové hry,
- vícekolové hry.

Možným rozdělením her je také podle počtu kol hry. Jestliže hráči hrají stejnou hru opakovaně (více kol), mohou měnit své strategie, případně využít stejnou strategii pro více kol. Jednou z oblastí vícekolových her jsou například evoluční hry. Vícekolové hry je možné dále rozdělit na hry s konečným počtem a nekonečným počtem kol. Jednou z hlavních odlišností obou variant je, že hráči konečné hry mohou v posledních kolech volit jiné chování než v dřívějších kolech. Odlišnost spočívá v tom, že hráči vědí o nastávajícím konci hry a že po skončení hry se již nemusí setkat. Není prostor pro případnou pomstu nebo jinou adaptaci chování [2].

2.1.3 Klasifikace strategií

Stejně tak, jako existuje několik základních typů her, i strategie lze rozdělit do několika základních kategorií, podle kterých by bylo možné klasifikovat strategie vězňova dilematu či jeho variant:

Podle úrovně dominance [20] se strategie dělí na

- dominantní strategie a
- dominované strategie.

Dominantní strategie lze dále rozdělit na silně dominantní a slabě dominantní strategie [20]. Silně dominantní strategie přináší hráči větší užitek, než by měl v případě volby jiné strategie, přičemž spoluhráči mohou volit jakoukoliv strategii [20]. Slabě dominantní strategie může mít stejný užitek jako některé jiné strategie, které si může hráč nebo soupeř zvolit, a vyšší užitek než zbylé strategie, které nepřinášejí stejný užitek jako slabě dominovaná strategie [20]. Hráč dominované strategie získává nižší užitek než v případě některých jiných strategií, které si může vybrat a které by mu přinesly vyšší užitek bez ohledu na to, jaké rozhodnutí by učinili soupeři [21].

Podle komplexnosti strategií je možné rozlišit

- čisté strategie (akce),
- vícekolové strategie složené z jedné čisté strategie,
- smíšené strategie (různé akce).

Strategie v jednokolových hrách je možné označit jako tzv. čisté strategie či akce [2]. Hráč nemůže kombinovat více různých strategií, neboť si pro jednokolovou hru může zvolit pouze jednu akci. Příkladem čisté strategie je rozhodnutí, zda jet autobusem nebo jet na kole, pokud neznáme předpověď počasí. Mezi oběma možnostmi se lze rozhodnout náhodně, například hodem mincí [22]. Pravděpodobnost, že pojedeme autobusem, by byla $p = 0,5$ a pravděpodobnost, že pojedeme na kole $1 - p = 0,5$ [22]. U vícekolových her je možné využít smíšených strategií (kombinace čistých strategií) [2]. Smíšených strategií lze využít u opakovaného vězňova dilematu.

U vícekolových her může strategie využitá pro všechna kola hry obsahovat pouze jednu čistou strategii. Je vhodné rozlišit použití jedné čisté strategie pro všechna kola nebo smíšenou strategii, neboť u smíšené strategie hráč mění své chování v průběhu hry. Je možné analyzovat důvody pro změnu chování a možné následky. Změna chování může také probíhat dle definovaného způsobu. Analyzovány jsou dopady definovaných změn chování na výplaty hráčů a výsledek všech kol hry.

Podle počtu možností je dělení strategií následující:

- konečný počet,
- nekonečný počet.

Jak uvádí [2], hráči mohou mít na výběr konečný nebo nekonečný počet rozhodnutí (strategií). Konečný počet možností znamená, že si hráč v jednom kole hry může zvolit pouze jednu z konečných disktrétních možností, např. ve hře „Kámen, nůžky, papír“. Nekonečný počet možností představují například aukce, ve kterých může hráč nabídnout jakoukoliv cenu za dražený předmět. Obecně představuje nekonečný počet možností hráče volba hodnoty spojité veličiny jako strategie [2].

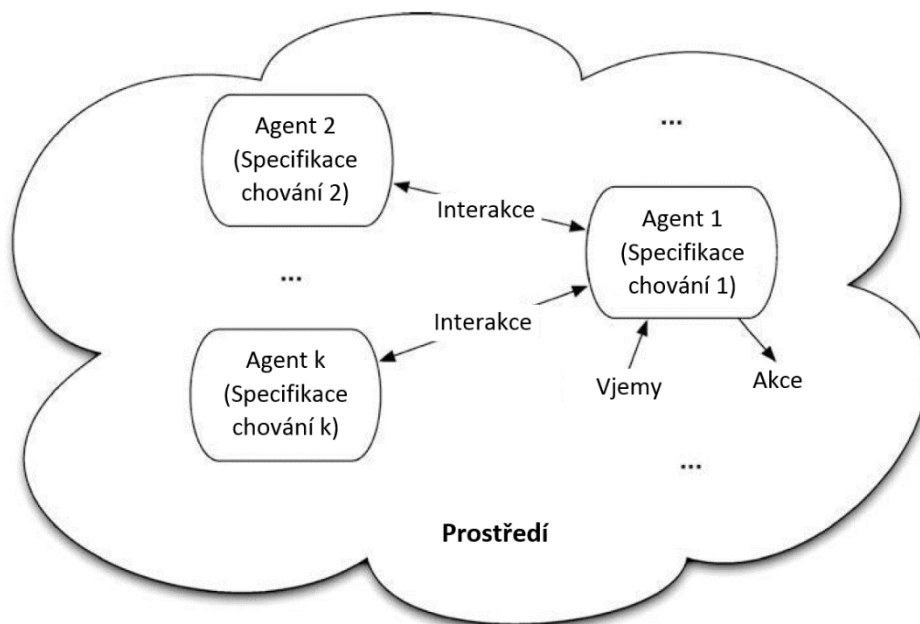
2.1.4 Multiagentové modely

Model je zjednodušenou reprezentací části reálného světa. Může se také jednat o plán reálné části světa, která bude v budoucnu vytvořena. Modely využívají princip abstrakce k odstínění v danou chvíli nepodstatných prvků pro lepší pochopení modelovaného systému či jeho části. Multiagentní modely představují množinu agentů ve sdíleném prostředí, kteří vykonávají určité akce za účelem dosažení určitých cílů [23]. Výsledné chování modelu je závislé na volbě akcí jednotlivých agentů [24].

[25] objasňuje využití multiagentních modelů v souvislosti s teorií her. Hru je možné vyjádřit jako multiagentní model s definováním pravidel a hráčů, kteří jsou v multiagentním modelu chápáni jako agenti. Ti mohou disponovat sadou atributů (vlastností, zjištěných vlastností prostředí nebo ostatních agentů, atd.), do kterých si ukládají důležité informace. Každý agent může ovlivnit průběh hry prostřednictvím metod, které jsou definovány v rámci modelu. Metody mohou přistupovat k hodnotám v atributů. Hra v užším smyslu slova představuje prostředí, ve kterém agenti interagují prostřednictvím nadefinovaných metod. Důležité jsou také vztahy mezi jednotlivými agenty (hráči) [25].

Obrázek 1 reprezentuje základní prvky multiagentových modelů a jejich vzájemné souvislosti. Základními prvky, které je možné obecně v multiagentových modelech identifikovat, jsou [24]:

- agent,
- prostředí,
- interakce,
- vjemy agenta,
- akce agenta.



Obrázek 1: Schéma prostředí s více agenty

Zdroj: [24], upraveno

Agenta je možné chápat jako autonomní entitu v rámci modelovaného systému [24]. Jednou z předpokládaných charakteristik agenta je schopnost samostatného rozhodování, jakou akci (čistou strategií) v konkrétní situaci nebo v konkrétním kole hry zvolí [24]. Každý agent volí akci dle definované části jeho chování, ale také v souvislosti s jeho vnitřním stavem a podněty z prostředí, případně interakce s dalšími agenty (viz schéma na obrázku 1 a [23]).

Jednou z vlastností agenta je autonomie. Tuto vlastnost vysvětlují [26], kteří uvádějí, že při volbě akce pro dosažení cílů je agent nezávislý na ostatních. Rozhoduje se také na základě dostupných informací. Jestliže je agent schopen samostatně volit akce, které mu pomáhají k dosažení jeho cílů, lze jej považovat za proaktivního. Reaktivita agenta znamená, že agent vnímá prostředí, ve kterém se nachází a reaguje na změny v prostředí, a to vhodným způsobem a ve vhodném časovém horizontu od doby, kdy tyto změny zaznamená. Důležitou vlastností agenta je komunikativnost, kterou prokazuje svými schopnostmi interakce s ostatními agenty. V prostředí, ve kterém agent spolupracuje při dosažení stanovených cílů s ostatními, je vzájemná komunikace ještě důležitější. Agent poskytuje také informace uživateli modelu. Dílčí vlastnosti v rámci sociálních schopností agenta jsou spolupráce, koordinace a vyjednávání [26].

Prostředí je sdílená platforma pro všechny agenty, která umožňuje sdílet informace s ostatními, popřípadě informace agentům poskytuje, ať již samo o sobě, nebo prostřednictvím zpráv zanechaných v prostředí ostatními agenty [27]. Agenti sdílené informace

dále zpracovávají a mohou na základě nich volit další akce [27]. Multiagentní prostředí je pro různé typy systémů odlišné [28]. Jednou z nejvíce sledovaných domén vztahu agenta k prostředí jsou mobilní agenti [28]. U těch je žádoucí rozvoj umělé inteligence. Mohou mít odlišné reprezentace prostředí a různě reagovat na překážky [28]. Rostliny a živočichové v ekosystémech jsou také závislí na prostředí, které mohou do jisté míry ovlivňovat [28]. U společenských her jsou spíše sledovány interakce mezi jednotlivými agenty [28]. U ekonomických systémů hraje prostředí také důležitou roli, především s ohledem na hospodářskou politiku [28].

Klíčové vlastnosti prostředí specifikují [26]. Jsou jimi především proměnlivost prostředí, dostupnost informací o prostředí a také míra, s jakou může agent kontrolovat a ovlivnit prostředí. Dynamické prostředí je proměnlivé v průběhu času, statické prostředí nikoliv. Agent může mít kompletní informace o prostředí, ve kterém se nachází (zcela dostupné prostředí). Například u šachů má hráč přehled o celé hrací ploše (prostředí).

Prostředí může být jen částečně dostupné, v tom případě má agent přehled pouze o svém nejbližším okolí, což bývá v realitě častou možností. Jestliže je prostředí ovlivněno i jinými faktory než akcemi agentů, jedná se o stochastické prostředí. Protikladem stochastického prostředí je deterministické prostředí. V deterministickém prostředí má agent 100% jistotu přechodu do následujícího stavu při volbě akce, která přechod mezi stavy umožní. U stochastického prostředí existuje z pohledu agenta pouze určitá pravděpodobnost přechodu prostředí do následujícího stavu [26].

Jak uvádí [29], interakce může probíhat mezi jednotlivými agenty, ale také mezi agenty a prostředím. Agenti spolu mohou komunikovat přímo či nepřímo. Přímou komunikací je například dotázání se druhého agenta ohledně poskytnutí určitého stavu atributu. Agenti spolu mohou navzájem interagovat jak u modelů konkurenční povahy, tak u modelů spolupráce. Interakce konkurenční povahy se vyskytuje například u přírodních modelů, které identifikují vztahy mezi kořistí a predátorem [29].

Nepřímá komunikace probíhá prostřednictvím prostředkovatele zprávy, nejčastěji prostředím. Jestliže spolu agenti nespolupracují při volbě svých akcí, je možné implementovat slabší formu spolupráce, aby si agenti poskytovali informace o sobě navzájem. Toho může být dosaženo implementací principů dotazování agentů na jisté typy informací, které jsou nutné pro rozhodování. Agent může také zjistit určité typy informací o svém konkurentovi už ze samotného chování a zaznamenáváním předchozích akcí agenta. Úplná pozorovatelnost prostředí je výhodná pro přehled změn, které druhý agent (konkurent) v prostředí

provedl. Spolupráce nepřímou komunikací se vyskytuje například u mravenců, kteří zanechávají feromonové stopy v prostředí, pokud naleznou zdroj potravy a vrací se s ním k mraveništi. Jiní mravenci se poté vydávají směrem, ve kterém jsou tyto stopy nejsilnější.

Interakci agentů a prostředí doprovází vjemy agenta a jeho akce, které blíže popisují [30]. Prostřednictvím vjemů získávají agenti informace o prostředí, popřípadě informace o ostatních agentech, pokud je využita komunikace v prostředí. Percepce u modelů sledujících interakci mezi agentem a prostředím na „fyzické“ úrovni se liší od provedení akce vůči prostředí tím, že nedochází ke změně stavu prostředí. Pro vjemy musí mít agent definované procedury, které zjišťují stav prostředí pomocí senzorů nebo přirozených smyslů, pokud se jedná o živou entitu.

Vjemy z prostředí si agent specifickým způsobem reprezentuje, následně vyhodnocuje a poté na ně může reagovat jednou z dostupných akcí. Volbou akce může naopak docházet ke změnám stavu prostředí. U společenských a ekonomických systémů je sledovaným faktorem volby akce spíše dopad na jiné agenty a na získanou výplatu v herní situaci, než dopad na samotné prostředí [30].

V případě společenských a ekonomických systémů dochází při volbě akce ke změnám výsledku hry nebo k ovlivnění chování ostatních agentů. Pokud by bylo například sledováno chování dvou konkurenčních firem při stanovování cen obdobných výrobků a zákazníků, kteří tyto výrobky poptávají, bylo by primárně sledovanými faktory nákupní chování zákazníků a konkurenční chování firem při stanovování různých úrovní cen nebo využití jiných metod marketingu. Vyšší poptávka po určitých typech výrobků způsobuje změny v prostředí, ve kterém firmy působí nebo ve kterém se výrobky prodávají, nepřímo. Takovými nepřímými změnami je například vyšší produkce odpadu a dopad na životní prostředí při výrobě daných výrobků.

2.1.5 Simulace

Multiagentové modelování a simulace je poměrně novým přístupem, který si získává své místo mezi metodami zkoumání umělé inteligence a teorie her díky rychlému rozvoji výpočetních technologií [31]. Zkoumání her pomocí multiagentových modelů je výhodné zejména v případě rychlých simulací s experimentální změnou parametrů. [31] uvádí možnosti a výhody multiagentních modelů a simulací. Při simulacích v multiagentních modelech jsou zkoumány možnosti učení agenta, adaptace v různých podmínkách, emer-

gence nebo využívání specifických strategií, které mohou být u opakovaných situací výhodnější. Může vznikat specifický způsob přechodného chování nebo odchýlení od rovnovážného stavu. Výsledné chování a interakce vzešlé v průběhu simulace mohou být příliš komplikované, spíše než aby se daly předpovědět bez využití metod simulace a výpočetních technologií. Pozorováno je jak chování jednotlivců, tak chování celých skupin. I z jednoduchých modelů interakce mohou vzejít překvapivé výsledky a závěry [31].

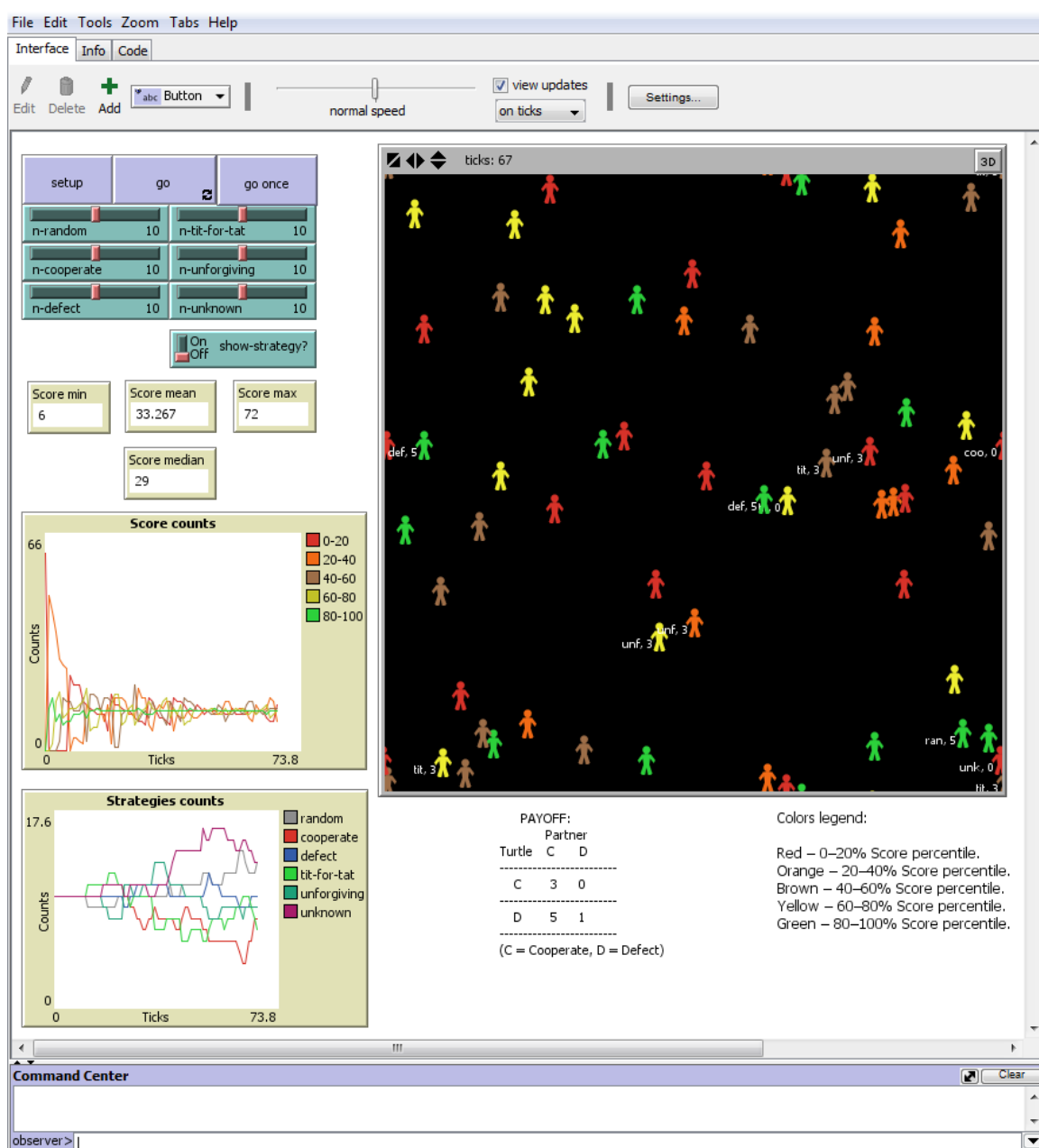
Simulací v multiagentních modelech se poté rozumí spouštění modelu s experimentální změnou parametrů, a to opakované v tom smyslu, že model je spuštěn po určitou dobu, která může být měřena diskrétními nebo spojitými časovými okamžiky. Poté je simulace ukončena, jsou zaznamenány sledované parametry a model je opět spuštěn s jinými počátečními parametry opět na stanovenou dobu. Stav sledovaných proměnných je uveden do výchozího stavu dle nastavených počátečních parametrů. Takto jsou generována výstupní data, která mohou být analyzována a mohou být případně potvrzeny či vyvráceny předpoklady nebo identifikovány nové závěry a vztahy, které nebyly předpokládány. Data jsou experimentálního charakteru, oproti indukční metodě, která využívá pro následnou analýzu empirická data.

S tvorbou modelu a samotnou simulací souvisí také metody dedukce a indukce. Obě metody vysvětluje [32]. Využití dedukce odlišuje následný vývoj modelu oproti indukci. Dedukce vychází z teoretických předpokladů a za využití různých analytických možností se snaží tyto předpoklady logickým způsobem odůvodnit a dokázat jejich platnost či vyvodit určitý závěr. Jedná se tedy o přechod od obecných konceptů ke specifickým způsobům aplikace. Oproti tomu indukce postupuje od specifického k obecnému. Snaží se odvodit obecné závěry z nasbíraných či pozorovaných dat, které slouží jako populační vzorek. Vědci využívají indukce k tvorbě hypotéz a teorií, které jim následně umožňují aplikovat obecné principy ve specifických situacích [32].

2.1.6 NetLogo

NetLogo je programovatelné open source prostředí pro multiagentní modelování a simulace v oblastech společenských a přírodních věd. Vytvořeno je s využitím programovacího jazyku Java. Jak uvádí [31], modelování a simulace v NetLogu umožňují autorovi modelu centralizované řízení všech agentů a agenti si zachovávají nezávislost mezi sebou navzájem. Prostřednictvím NetLoga lze zkoumat chování skupin agentů či globální chování celého modelu v čase. Většinou jsou v NetLogu modelovány právě opakované děje. Dostupné jsou rovněž ukázkové modely z různých vědních oborů a dokumentace [15].

NetLogo má intuitivní prostředí a je možné jej přizpůsobit či vytvořit ovládací prvky modelu. K dispozici je několik prvků uživatelského rozhraní, například tlačítko, posuvník, přepínač, vstupní a výstupní pole, grafy či výběr z několika nadefinovaných možností jako vstupní parametr [15]. Jednoduché a přehledné uživatelské prostředí neodvádí pozornost od podstaty řešeného problému a uživatel se může více zaměřit na samotný model a jeho funkčnost. Složitost vytvořeného uživatelského rozhraní modelu závisí na uživateli a na modelovaném problému. Jestliže je kód modelu uzpůsoben pro opakované experimenty a dílčí experiment není výpočetně náročný, je možné získat výsledky modelovaného problému v řádu sekund. Experimenty složitějších modelů však mohou být časově náročné.



Obrázek 2: Ukázka prostředí NetLoga

Zdroj: vlastní zpracování a [15]

Ukázku prostředí NetLoga reprezentuje obrázek 2. Pro tuto ukázkou byl využit autorem upravený model opakovaného vězňova dilematu s n hráči s různými strategiemi. V levé horní části si lze všimnout ovládacích prvků (tlačítek, posuvníků a přepínače). Pod nimi se nachází výstupní pole (monitory), které sledují definované ukazatele. V levé spodní části jsou grafy, které sledují více barevně odlišených řad najednou v průběhu času. Čas je v NetLogu reprezentován diskretními časovými jednotkami (ticks). Ticks mohou být sekundy, minuty, hodiny atp. Barvy, sledované řady, název grafu, osy, rozsahy os a legendy je možné definovat v kódu modelu.

Pravá část uživatelského rozhraní obsahuje grafický výstup simulace s jednotlivými agenty v prostředí. Agenti se v prostředí pohybují a opakovaně mezi sebou interagují, pokud se setkají, a je jim přidělena výplata dle výplatní matice, popsané na obrázku 2 vpravo dole. U hráčů, kteří se právě setkali, je zobrazen popisek se zkratkou strategie a bodového zisku.

Horní lišta obsahuje záložky Interface, Info a Code. Ve vyobrazené záložce Interface se nachází uživatelské rozhraní modelu s ovládacími prvky. Horní lišta v záložce Interface obsahuje základní nastavení modelů v NetLogu a prvky související s tvorbou uživatelského rozhraní modelu. Prostřední část záložky Interface obsahuje uživatelské rozhraní modelu a spodní část příkazovou řádku s výstupem. U příkazové řádky je možné měnit typy prvků modelu, kterým je dotaz z příkazové řádky určen (observer, turtles, patches, links). V záložce Info je možné vytvořit vlastní popis modelu, popřípadě zde popis modelu nalézt, například u modelů z knihovny ukázkových modelů. Záložka Code obsahuje kód modelu:

```
1  globals
2    [ score-mean
3      score-max
4      score-min
5    ]
6  turtles-own
7    [ strategy
8      score
9    ]
10 ...
11 breed [ sheep a-sheep ]
12 breed [ wolves wolf ]
13 ...
```

Agenty v NetLogu představují tzv. „turtles“, kteří mají nadefinované vlastnosti turtles-own (viz předcházející ukázka kódu). Atributy agentů mohou mít různé formáty, například textový formát, číselný formát nebo seznam. Ve výchozím nastavení při inicializaci modelu jsou atributy nastaveny na celočíselný formát s hodnotou 0, pokud není v kódu modelu v inicializační proceduře definováno jinak. Lze vytvářet vlastní třídy agentů (breed), například vlky a ovečky (wolves, sheep) a těm definovat rozdílné vlastnosti dle potřeby modelování. Globals představují globální proměnné, které jsou dostupné všem agentům a také autorovi modelu. Blok příkazů související s nadřazeným prvkem je ohraničena hranatými závorkami „[“ a „]“

Model v NetLogu je rozdělen na jednotlivá políčka (patches), která mohou být také považována za agenty (typicky u biologických modelů jako buňky). Tato políčka jsou prostředím pro agenty a rovněž mohou mít vlastní atributy, které lze definovat v bloku patches-own. Počet políček modelu je možné určit prostřednictvím změny rozměrů grafického výstupu modelu. Při změně rozměrů je možné změnit také velikost políček. Agent se může nacházet v jakékoliv části políčka. Inicializační procedura modelu může být nazvána „setup“ a pro její spuštění lze definovat příslušné tlačítko v uživatelském rozhraní. Příklad procedury zobrazuje následující ukázka kódu:

```
1  to setup
2    ca
3    reset-ticks
4    ...
5    setup-turtles
6    do-plots
7  end
8
9  to go
10   ...
11  ask turtles [ recolor ]
12  tick
13  do-plots
14 end
```

Počátek každé procedury je deklarován příkazy „to“ a končí „end“. Stejné pravidlo platí pro hlavní aktualizací proceduru modelu, kterou lze pojmenovat „go“ a při jednom kliku na tlačítko, které ji vyvolává, může dojít buď k jednomu provedení této procedury, nebo k opakovanému spouštění v průběhu ticks. V rámci procedury je možné spouštět i jinou proceduru (např. do-plots) deklarovanou odděleně, aby byl kód modelu přehlednější. Proceduru mohou spouštět například pouze agenti či určitý rod agentů po dotázání (ask

turtles). Standardizovanými příkazy jsou například „ca“, které resetuje všechny změny, provedené aktualizací procedurami modelu, a „reset-ticks“, které vynuluje čítač kroků (ticků) modelu. Čítač modelu se navyšuje s každým spuštěním procedury, která obsahuje deklaraci „tick“.

Interakce v NetLogu představuje například vyvolání metody ask turtles se specifickým dotazem na hodnotu atributů ostatních agentů. Agent má oprávnění k zadávání příkazů ostatním agentům a políčkům, a to i v rámci změny hodnot atributů (stavu). Otevřenost modelování je užitečná, avšak musí jí být obezřetně využíváno, aby nedocházelo k nechtěnému chování modelu. Na následující ukázce kódu je uvedena situace, kdy se agent ptá ostatních agentů v okruhu třech políček, aby zobrazili svou strategii. Strategie dotázaných agentů se zobrazí ve výstupu u příkazové řádky.

```
1 ask other turtles in-radius 3 [ show strategy ]
```

Vjemy agenta jsou například zjišťování informací o políčku, na kterém se agent nachází, jak zobrazuje následující ukázka kódu. Dotázané políčko zobrazí své souřadnice a agent zjistí svou přibližnou polohu. Agent v NetLogu svou polohu samozřejmě zná, a to s vyšší přesností než souřadnice políčka, na kterém se nachází. Souřadnice agentů jsou standardními atributy, které není třeba dodatečně definovat.

```
1 ask patch-here
2   [ show pycor
3     show pxcor
4   ]
```

Jak již bylo uvedeno v kapitole 2.1.4, akce jednotlivých agentů mohou ovlivňovat stav prostředí. Na následující ukázce kódu je ilustrován příklad akce agenta, která ovlivňuje stav prostředí. Agent komunikuje s políčkem prostředí, na kterém se nachází, a požaduje, aby políčko změnilo svou barvu na červenou. Pořadí příkazů pro změnu hodnot atributů v NetLogu je „set atribut hodnota“, bez dalších operátorů (např. rovnítko).

```
1 ask patch-here [ set pcolor red ]
```

NetLogo obsahuje také nástroj BehaviorSpace [15], ve kterém je možné vytvářet a ukládat více experimentů s různým nastavením. Nastavení experimentů spočívá především v nastavení hodnot parametrů, které jsou navzájem kombinovány, určení počtu opakování jednotlivých kombinací parametrů a zaznamenání určených výsledků. Hodnoty parametrů se

nastavují z prvků uživatelského rozhraní, jako např. textový vybírací prvek, za jehož názvem se kombinované hodnoty v nastavení experimentů vypisují. Parametry modelu mohou rovněž představovat rozměry grafického výstupu modelu.

Výsledky je možné zaznamenávat do výstupního souboru také při každém provedení aktualizací procedury (odehrání jednoho kola hry). Pokud není časový limit implementován v kódu aktualizací procedur, lze jej nastavit v BehaviorSpace. Dílčí experimenty lze také zastavit určením podmínky zastavení (stop condition) a po dokončení experimentů provést deklarované příkazy v bloku „final commands“. Experimenty v BehaviorSpace podporují vícejádrové procesorové zpracování.

2.2 Vězňovo dilema, jeho varianty a aplikace

Kapitola 2.2 obsahuje vysvětlení klasického a opakovaného vězňova dilematu. Poté jsou popsány varianty vězňova dilematu či jeho aplikace. Součástí kapitoly 2.2 je také popis strategií, které vycházejí z citovaných zdrojů.

2.2.1 Klasické vězňovo dilema

Vězňovo dilema je jednou z nejznámějších situací, studovaných v rámci teorie her [33]. Bylo nalezeno a popsáno mnoho aplikací v různých vědních oborech, ve kterých byl v různé míře identifikován princip vězňova dilematu [33]. Jedním z těchto oborů je například ekonomie nebo management [33].

Jak uvádí [34], klasické vězňovo dilema popisuje situaci, kdy policie zatkla dva podezřelé. Ti jsou při vyšetřování drženi v oddělených místnostech, přičemž každý podezřelý se může rozhodnout, zda toho druhého udá nebo neudá. Udání či neudání podezřelých je jediný rozhodovací aspekt pro soud, neboť policie nemá dostatek důkazů pro usvědčení podezřelých. V teorii her je rozhodnutí udat druhého podezřelého možné reprezentovat jako zrada a volba neudat druhého jako spolupráce. Oba podezřelí jsou navzájem nezávislí ve svém rozhodování, každý se může rozhodnout pro zradu či spolupráci, ať se druhý podezřelý rozhodne jakkoliv [34].

Situace, které mohou u vězňova dilematu nastat, popisují [34]. Pokud se první podezřelý rozhodne neudat druhého a druhý ho udá, potom první podezřelý obdrží nejvyšší definovaný trest a druhý podezřelý vyvázne bez trestu. Pro prvního podezřelého by bylo výhodnější udat druhého, jestliže ten jej také udá. V takovém případě by šli oba do vězení, ale dostali by nižší trest, než v první jmenované situaci. Jestliže se naopak druhý podezřelý

rozhodne neudat prvního, potom může první podezřelý získat svobodu, jestliže druhého podezřelého udá. Pokud by se oba podezřelí rozhodli mlčet, tak nastává z hlediska klasického jednokolového věžňova dilematu nejvýhodnější situace pro obě strany. Sice by oba podezřelí šli do vězení, ale na kratší dobu než v jakémkoliv jiném případě, kdy by se na svobodu nedostali. [33] poté uvádějí také možné výše trestů, resp. odpovídajících užiteků, viz také Tabulka 1.

Tabulka 1: Příklad trestů hráčů věžňova dilematu

		Podezřelý 2	
		Neudat	Udat
Podezřelý 1	Neudat	-1, -1	-7, 0
	Udat	0, -7	-4, -4

Zdroj: vlastní zpracování a [33]

Délky trestů mohou být například 0, 1, 4 a 7 roků. Pokud se oba podezřelí navzájem zradí, jsou odsouzeni každý na 4 roky. Délka trestu prvního podezřelého je 7 let, jestliže se rozhodne neudat druhého podezřelého a ten jej zradí. Jestliže naopak druhý podezřelý chce spolupracovat a první podezřelý jej zradí, tak je první podezřelý svobodný a zrazený podezřelý se stává vězněm na 7 let. Při spolupráci obou podezřelých jsou oba odsouzeni na 1 rok. Tabulka 1 reprezentuje rozdělení trestů pro oba podezřelé, a to jako záporné užítky, protože vyšší trest znamená nižší užitek pro daného podezřelého. Při maximalizaci svého užitku hráči věžňova dilematu hledají nejnižší trest, čili maximální hodnotu ze záporných čísel.

Dominantní strategie obou hráčů je zrada (udání druhého podezřelého), jak uvádí [33]. Kdyby se totiž jakýkoliv hráč rozhodl udat toho druhého, tak to pro něj znamená vždy nižší trest než v případě, že by sám spolupracoval. Dominovanou strategií pro oba hráče je analogicky spolupráce, žádný podezřelý nedosáhne nižšího trestu, pokud místo zrady zvolí spolupráci. Nashova rovnováha nastává v situaci, kdy se oba podezřelí udají navzájem [35]. Tato rovnováha je stabilní, pokud si první podezřelý zvolil zradu, tak druhý podezřelý již nemůže dosáhnout lepšího postavení [35]. Nashova rovnováha u věžňova dilematu však není Paretovsky optimální [35]. Volbou vzájemné spolupráce by oba podezřelí získali nižší trest, čili získali by v této situaci vyšší užitek [35]. Oba podezřelí se ale nemohou do-

mlouvat a Paretovsky optimální strategie neudat-neudat není dlouhodobě stabilní, protože oba vězni jsou si vědomi toho, že je pro ně výhodnější zradit, ať už druhý podezřelý zvolí jakoukoliv možnost [35].

Tabulka 2: Obecné vyjádření věžňova dilematu

		Hráč 2	
		Spolupráce	Zrada
Hráč 1	Spolupráce	B, B	D, A
	Zrada	A, D	C, C

Zdroj: vlastní zpracování a [36]

Tabulka 2 představuje obecné vyjádření věžňova dilematu, které uvádí a vysvětluje [36]. Výplaty jednotlivých hráčů jsou reprezentovány písmenným označením od A do D dle velikosti výplaty. Písmeno A představuje nejnižší trest, neboli nejvyšší užitek hráče, D představuje nejvyšší trest (nejnižší užitek), viz tabulka 3. Udání je nahrazeno obecně zradou a spolupráce představuje mlčení, tedy neudání druhého hráče. Hra vyjádřená výplatní maticí dle tabulky 2 je věžňovým dilematem, jestliže splňuje nerovnost $A > B > C > D$. Zrada je silně dominantní strategií obou hráčů oproti strategii spolupráce a spolupráce je dominovanou strategií.

Silná dominance zrady pro oba hráče by byla porušena, jestliže by byla splněna nerovnost $A \geq B > C > D$ nebo $A > B > C \geq D$. Poté by byla spolupráce pouze slabě dominantní strategií. Racionální hráči sledující pouze vlastní zájmy volí opět strategii zrada-zrada a oba získávají výplatu C . Svoji pozici by si mohli vylepšit vzájemnou spoluprací, kdy by oba obdrželi výhodnější výplatu B . Museli by však splňovat koncept sdílených znalostí. V obecném vyjádření je věžňovo dilema také symetrickou hrou a výplaty obecného věžňova dilematu jsou pouze ordinálního charakteru. Indikují pouze skutečnost, že je jedna z výplat vyšší než druhá, ale již neupřesňují o kolik [36].

Tabulka 3: Obecná reprezentace výše trestů (užitků)

Užitek	Označení
-7	D
-4	C
-1	B
0	A

Zdroj: vlastní zpracování a [36]

2.2.2 Opakované věžňovo dilema

Opakované věžňovo dilema probíhá ve více kolech. Hráči mají možnost analyzovat předchozí tahy soupeře a brát je v úvahu při volbě své strategie v současném kole. Strategie mohou být komplexnější, čítající více čistých strategií. Hráči rovněž mohou spolupracovat a mohou získat v Paretově optimu vyšší výplaty než v případě jednokolové varianty. Vícekolová varianta může obsahovat konečný počet kol nebo nekonečný či neurčený počet kol. Hráči mohou v průběhu hry měnit své čisté strategie. Zajímavé je z dlouhodobého hlediska sledovat, zda dojde ke vzniku trvalé spolupráce či nikoliv. Rovněž je nasnadě analyzovat nejčteněji využívané strategie, popřípadě hráčům strategie určit a sledovat, jak obstojí v porovnání s ostatními strategiemi, které jsou přiřazeny druhému hráči.

Jak uvádějí [37], opakované hry jsou v realitě častou variantou. Opakované věžňovo dilema nachází aplikaci například ve vzájemném střetávání konkurenčních firem na trhu, které mohou měnit svá rozhodnutí v podobě stanovování cen, výše produkce, struktury produkce, apod. Jestliže se hráči setkávají opakovaně, tak je jejich chování jiné, než pokud by se spolu setkali pouze jednou [37].

Existují však určité předpoklady opakované hry, viz [37]:

- Každý hráč má v jednotlivých kolech hry k dispozici stejné čisté strategie, které měl k dispozici v předcházejících kolech.
- Výplaty hráčů nezávisí na kole, které se hraje, ale pouze na zvolených čistých strategiích jednotlivých hráčů.
- Způsob stanovování výplat je ve všech kolech stejný.
- Hráči se v jednotlivých kolech hry rozhodují současně.
- Každý hráč zná předchozí tahy soupeřů.

Jak upřesňují [37], první tři předpoklady naznačují, že výplatní matice pro každé kolo hry je stejná, neboli má stejné hodnoty výplat pro jednotlivé hráče a také stejný rozměr. Z posledních dvou předpokladů vyplývá, že volba akce může být ovlivněna minulými rozhodnutími soupeřů (ne současnými).

Opakované věžňovo dilema rozlišuje různé druhy strategií. Jedny ze základních strategií uvádí [38]. Jsou jimi například trvalá spolupráce nebo trvalá zrada. V těchto strategiích hráč volí spolupráci či zradu v každém kole opakovaného věžňova dilematu bez ohledu na to, jakou strategii volí soupeř.

Jednou ze strategií je také Grim Trigger. Tato strategie představuje spolupráci v úvodním kole s následující spoluprací do té doby, dokud všichni hráči spolupracují v předchozích kolech. Za zradu je tak postižen jak samotný hráč, tak soupeři.

Oko za oko (Tit-for-Tat) předpokládá spolupráci v prvním kole a v dalších kolech volit dle strategie soupeřů v předchozím kole. Pokud soupeř v předchozím kole zradil, tak je doporučeno také zradit, pokud spolupracoval, tak také spolupracovat v současném kole.

Omezená odplata představuje spolupráci v prvním kole a k kol se strategií zrady jako odplatu za zradu soupeře v předchozím kole. Po k kolech následuje návrat k výchozí strategii spolupráce bez ohledu na to, jaké strategie volil soupeř v průběhu k kol odplaty.

Strategie Pavlov doporučuje spolupráci v prvním kole a poté spolupráci pokud oba hráči v předchozím kole zvolili zradu nebo oba zvolili spolupráci, jinak zradit. Jednou podvádět představuje využití strategie oko za oko až do periody kol L , po L kolech jednou podvádět, v kole $L + 1$ spolupracovat a poté opět používat strategii oko za oko.

Model „PD Two Person Iterated“, který je součástí knihovny modelů NetLoga (viz [15]), obsahuje strategie random, tit-for-two-tats a unforgiving. Random představuje náhodnou volbu akce ve všech kolech hry. Strategie tit-for-two-tats je obdobná jako tit-for-tat s tím rozdílem, že hráč zrazuje pouze tehdy, když zradil jeho soupeř v předchozích dvou kolech po sobě. Hráč strategie unforgiving zrazuje tehdy, jestliže sám v předchozím kole zradil nebo jej zradil soupeř.

Spolupráci v sudém kole a v lichém kole zradu popisuje [39]. Střídání spolupráce a zrady uvádí také [40]. Střídat spolupráci a zradu lze po dvou kolech. [40] rovněž zmiňuje kombinaci strategie tit-for-tat s principy ostatních strategií, například vytvořením tolerantní varianty, ve které hráč ve 33 % případů odpustí zradu svému soupeři. [41] popisují kromě 33% šance odpuštění zrady u tit-for-tat také variantu s náhodnou zradou ve 33 % kol.

Jak uvádí [36], nejúspěšnější strategií opakovaného vězňova dilematu na základě Axelrodova turnaje je Oko za oko. Úspěšnost této strategie je založena na čtyřech vlastnostech, viz [36]. První z nich je nezrazovat v prvním kole. Odveta za zradu soupeře nebo oplacení spolupráce je rovněž dlouhodobě výhodné. Doporučením je také schopnost odpuštění zrady, aby bylo možné obnovit spolupráci, která je pro hráče výhodnější. Nelítostná strategie a chamtivost jsou méně výhodné od okamžiku, kdy soupeř jednou zradil. Poslední vlastností je jednoduchost strategie, která je žádoucí pro ostatní hráče, snažící se předpovídat soupeřovo chování.

2.2.3 Sosisova hra v kibucech

První zvolenou alternativou vězňova dilematu je Sosisova hra v kibucech. Kibuc je hebrejský výraz a lze jej přeložit jako skupinu či sdružování. Jak uvádí [42], jedná se o ekonomicky soběstačnou izraelskou osadu, která může čítat 80 až 2 000 obyvatel. Zisky ze zemědělské a průmyslové činnosti kibucu se po uspokojení potřeb obyvatel investují zpět do rozvoje osady.

O Sosisově hře v kibucech pojednává [43]. V každém kole si dva hráči rozdělí částku 100. Každý hráč dopředu zvolí požadovanou částku a o zbytek do 100 se rozdělí napůl. Hráč před vyhodnocením kola nezná požadovanou částku soupeře. Jestliže je celková částka, požadovaná všemi hráči, vyšší než 100, nikdo nedostane nic [43].

Jak uvádí [43], jednou z nejjednodušších strategií opakované Sosisovy hry je požadovat ve všech kolech stejný díl z původního zisku p . Oba hráči mohou požadovat vždy $0,5p$ nebo jeden hráč více než jeho soupeř. Vždy se jedná o konstantní součet p . Podle [43] je hru ještě možné modifikovat o odměnění skromnosti obou hráčů. Jestliže by součet požadovaných částek obou hráčů nepřesáhnul rozdělovaný zisk p , rozdíl bude navýšen o polovinu a rozdělen rovným dílem jako bonus oběma hráčům. Maximální částku by v takovém případě oba hráči získali, jestliže by oba nepožadovali nic, a to $0,75p$ z původního zisku p navýšeného na $1,5p$. Navýšení rozdílu o polovinu je možné chápat jako budoucí výnosy ze současné investice nerozděleného zisku.

Jak popisuje [43], pokud by však hráč věděl o tom, že jeho soupeř nepožaduje žádnou částku ze zisku, tak by mohl požadovat celý zisk a získat o $0,25p$ více než pokud by také nepožadoval žádnou částku. Získal by celý zisk p pro sebe, ale soupeř by nezískal nic. Pokud by se nerozdělená částka násobila třemi a poté dělila mezi oba hráče, existovala by jediná optimální strategie bez ohledu na chování soupeře [43].

2.2.4 Boj o zdroje

Jak popisuje [44], klasická verze hry Jestřáb-holubice (Hawk Dove) představuje situaci, kdy se dva ptáci stejného druhu příležitostně setkávají a bojují spolu o hodnotný zdroj. Zaujmout mohou aktivní (agresivní) či pasivní postoj. Pasivní pták představuje roli „holubice“ a agresivní pták „jestřába“. Pasivní pták se celého zdroje vzdá a získá jej aktivní pták. Dva agresivní ptáci spolu budou bojovat a dva pasivní ptáci budou zdroj sdílet a každý získá polovinu. Předpokladem je, že ptáci pocházejí ze stejné populace a vyvíjejí se za stejných životních podmínek [44].

Příklad užitků při setkávání agresivních a pasivních ptáků je uveden ve výplatní matici v tabulce 4. Vychází z popisu hry dle [44]. Pokud dojde k souboji dvou agresivních ptáků, zlepší se zdatnost ptáků o dvě jednotky užitku. Setkají-li se dva pasivní ptáci, oba získají čtyři užitky. Při setkání agresivního a pasivního ptáka získá agresor všech osm bodů užitku a pasivní pták nezíská nic. [44] vysvětluje, že Nashova rovnováha nastane, pouze pokud oba ptáci zvolí agresi, která je silně dominantní strategií. Oba ptáci by však mohli dosáhnout lepšího užitku, pokud by se rozhodli cenný zdroj sdílet. Výskyt pasivních jedinců v populaci vede k jejich postupné eliminaci při jakémkoliv podílu agresivních jedinců [44].

Tabulka 4: Výplatní matice hry Jestřáb-holubice

		Pták 2	
		Pasivita	Agrese
Pták 1	Pasivita	4, 4	0, 8
	Agrese	8, 0	2, 2

Zdroj: vlastní zpracování a [44]

Jak uvádí [44], hru Jestřáb-holubice je možné modifikovat odečtením několika jednotek užitku ptákům, kteří spolu bojují. Tato modifikace má blíže k reálné situaci, neboť se dva agresivní jedinci mohou při boji zranit oproti pouhému zvýšení zdatnosti a z jejich boje jim tak plyne i záporný užitek. Tabulka 5 reprezentuje výplatní matici, která odečítá čtyři jednotky užitku dvou agresivním jedincům v boji.

Tabulka 5: Výplatní matice modifikované hry Jestřáb-holubice

		Pták 2	
		Pasivita	Agrese
Pták 1	Pasivita	4, 4	0, 8
	Agrese	8, 0	-2, -2

Zdroj: vlastní zpracování a [44]

Jak popisuje [44], modifikovaná verze hry Jestřáb-holubice má tři Nashovy rovnováhy, z nichž jsou dvě čisté a jedna smíšená. Čisté Nashovy rovnováhy nastanou tehdy, když je jeden pták agresivní a druhý pasivní či naopak. Smíšenou Nashovou rovnováhou je podle [44] u opakovaného Jestřába-holubice využití pasivity v jedné třetině případů a agrese ve dvou třetinách.

2.2.5 Dvourozměrné prostorové hry

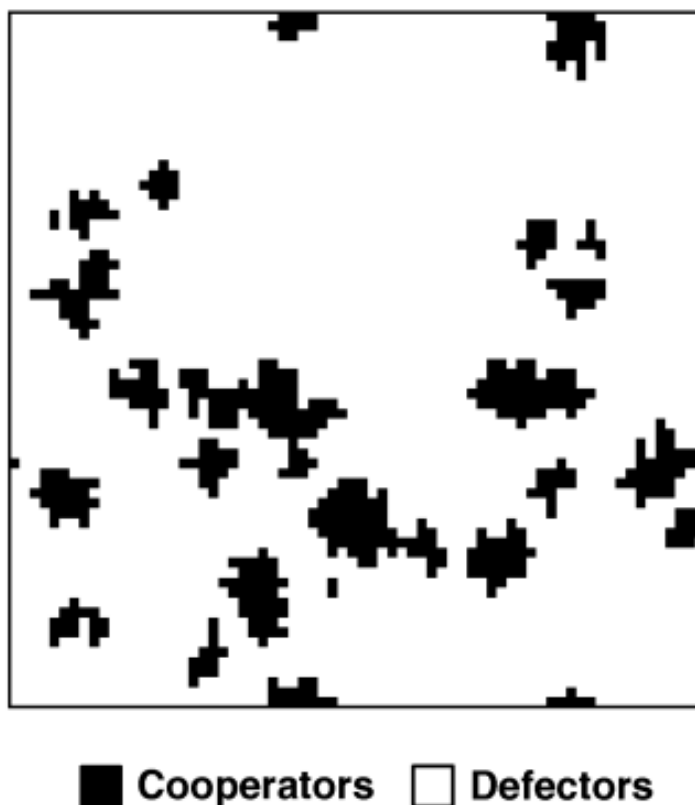
Použití vězňova dilematu pro prostorové hry obdobné buněčným automatům popisují [45]. Tento zdroj byl využit pro zpracování této kapitoly.

Jednotlivci v populaci obývají část z celkového prostoru rozděleného mřížkou se stejnými políčky, obdobně jako reprezentace buňky v případě modelování buněčného automatu. Každý jedinec sousedí s osmi dalšími. Výplata hráčů v jednotlivých kolech hry je určena na základě lokálních interakcí se sousedy. Více sousedů může využívat stejnou strategii a generace jedinců jsou součástí evolučního procesu, při kterém jsou strategie proměnlivé, popř. počet jedinců využívajících tutéž strategii je proměnlivý. Šíření úspěšných strategií představuje reprodukci či učení a evoluci sousedních jedinců, přejímajících úspěšnější strategii, viz [45].

Aktualizace celého herního prostoru probíhá v jednotlivých kolech najednou. Ve stejném kole také proběhne interakce všech jedinců se sousedy a je stanovena jejich dílčí výplata v jednom kole. Změna strategie jedince v oblasti může být provedena na základě různých způsobů vyhodnocení úspěšnějších strategií. [45] uvádějí například porovnání výplaty v jednom kole s jednou náhodně vybranou sousední oblastí. Výplata jedince je vypočítána jako průměr výplat získaných ze všech interakcí se sousedy. Pokud je výplata sousedů vyšší, jedinec méně úspěšnější oblasti přejímá strategii úspěšnějšího souseda.

Dalším způsobem lokální interakce je například vyhodnocení výše výplat všech sousedů a rozšíření nejúspěšnější strategie, popř. zachování současné strategie jedince, pokud žádný soused nezískal vyšší výplatu. Rozšířit se rovněž může nejčteněji zastoupená strategie v sousedních oblastech či také strategie s nejvyšším součtem výplat. Výše výplat lze určit opět jako průměr ze všech dílčích výplat získaných na základě interakcí se sousedními oblastmi [45].

Jak uvádějí [45], zajímavým faktorem interakce na globální úrovni je sledování konvergence k tvorbě komunit stejných strategií, které by mohly lépe přežívat navzdory okolním skupinám či jednotlivcům s jinými strategiemi, viz obrázek 3, který reprezentuje tvorbu komunit ve dvourozměrné hře představující alternativu k vězňově dilematu. Hráči obývající jednotlivé oblasti (políčka) používají dvě základní strategie vězňova dilematu. Bílá políčka představují zrazující hráče a černá políčka spolupracující hráče.



Obrázek 3: Tvorba koalic ve dvourozměrné hře
Zdroj: [45]

Zastoupení jednotlivých strategií může být v prvním kole stejné nebo může být různé. Je možné sledovat rychlost reprodukce či evoluce strategií či vymizení neúspěšných strategií. Šíření úspěšnějších strategií v jednotlivých oblastech je deterministické nebo stochastické [45]. Celou oblast je možné pojmut mimo jiné také jako konečnou a jedinci obývající prostor na okraji oblasti by měli jiný počet sousedů, než jedinci uprostřed, a pravidla hry pro účastníky by tak byla asynchronní [45].

3 Praktická část

Tato kapitola se věnuje modelům variant vězňova dilematu. V kapitole 3.1 je popsán návrh modelů jednotlivých variant. Poté je specifikována implementace modelů a jsou uvedeny ukázky kódu, včetně popisu funkcionality, kterou v rámci vytvořeného modelu reprezentují. Kapitola 3.3 obsahuje návrh a popis experimentů pro varianty vězňova dilematu, popis klíčových aspektů experimentování a analýzu výsledků experimentů. V závěrečné části (kapitola 3.4) je věnován prostor diskuzi o výsledcích experimentů.

3.1 Návrh modelů

Návrh modelů obsahuje prvotní analýzu modelovaných situací a klíčových faktorů pro následnou implementaci. Modely alternativ vězňova dilematu jsou rozděleny do samostatných kapitol, aby si čtenář mohl číst pouze o tom modelu, který jej zajímá. Analogicky u kapitol 3.2, 3.3 a 3.4.

3.1.1 Sosisova hra v kibucech

Cílem odpovídajícího modelu Sosisovy hry je vytvořit modifikovanou variantu, u které jsou hráči odměněni za skromnost, tj. pokud součet požadovaných částek nepřesáhne rozdělovaný zisk. Hry se účastní dva hráči. Volby požadovaných částek by měly být vyhodnoceny nezávisle na hráčích. Navrhovaná výše zisku je shodná s modelem, který je popsán v kapitole 2.2.3, tedy 100. Požadované částky je možné chápat také jako procentuální část ze zisku. Nekooperativní rozhodování představuje nezávislou volbu požadované částky v aktuálním kole mezi agenty navzájem.

Uživatelské rozhraní by mělo obsahovat monitorovací prvky, které zobrazují požadované a reálné zisky obou hráčů, popřípadě celkové zisky. Samozřejmostí jsou ovládací tlačítka pro aktualizací a inicializační procedury modelu. Ovládacím tlačítkem pro spuštění inicializačních procedur může být „setup“ a pro spuštění aktualizací procedur tlačítka „go“ a „go-once“, přičemž „go-once“ provede spuštění aktualizací procedur pouze jednou a je odehráno jedno kolo hry.

Aby bylo možné monitorovat celkové skóre za jednu hru a vyhodnotit úspěšnost jednotlivých strategií, měli by mít agenti mimo jiné další atribut, který by tuto funkcionalitu umožňoval. Požadované zisky jsou hráčům přiděleny dle jejich strategie, kterou je možné z hlediska pozorovatele modelu měnit v jakémkoliv kole hry. Vyhodnocení požadovaných částek je vhodné provést na globální úrovni z pohledu pozorovatele modelu, protože by

hráči neměli mít možnost do tohoto procesu zasáhnout. V grafickém výstupu modelu je možné monitorovat náladu agentů v závislosti na výsledku kola.

Na konci kola by mohla být oběma hráčům známa skutečně přidělená částka, která tvoří požadovanou část a polovinu ze zbylé části do výše zisku, která je navíc navýšena o polovinu. Jestliže je součet požadovaných částek obou hráčů menší nebo roven celkovému zisku, je možné určit přidělené částky v rámci stejné metody, neboť půlená část nulového zbytku je rovněž nulová.

Strategie `random` a `tit_for_tat` byly navrženy na základě opakovaného vězňova dilematu, viz kapitola 2.2.2 a strategie `0.5_profit`, `invest_all` a `take_all` na základě popisu Sosisovy hry, viz kapitola 2.2.3. První navrhouvanou strategií je „`random`“, na základě které si hráč náhodně vybere v jednom kole hry částku od 0 do 100. Požadovaná částka je celé číslo. Strategie, na základě které hráč požaduje polovinu zisku v každém kole, může být nazvána „0.5 profit“. Hráč může také vždy chtít celý zisk pro sebe (`take all`) či celý zisk investovat do rozvoje kibucu (`invest all`).

Jistým zjednodušením strategie „`random`“ z pohledu variant rozhodování je požadovat pouze částky dělitelné deseti. Odpovídající strategii „`take n`“ se může nastavit různá hodnota n , pro $n = \{10, 20, 30, 40, 60, 70, 80, 90\}$. Zajímavá je modifikace strategie „`tit-for-tat`“ z opakovaného vězňova dilematu, hráč může požadovat tolik, co jeho soupeř v předchozím kole či tolik, kolik jeho soupeř v předchozím kole reálně vydělal.

Požadování takové částky, kterou hráč v předchozím kole vydělal, lze považovat za samostatnou strategii „`real as required`“. S tím také souvisí opakování akce, pokud hráč v předchozím kole dosáhl reálného zisku (`repeat if real earn`). Zkombinováním „`repeat if real earn`“ a vyžádáním částky 75 přibližně ve třetině případů vzniká „`repeat if real earn and random75 33%`“. Částka 75 byla zvolena z důvodu přiblížení se strategii zrady u opakovaného vězňova dilematu. Pokud by bylo požadováno 100, byla by menší pravděpodobnost, že by bylo dosaženo dobrých výsledků náhodné volby.

Z hlediska experimentů může být potenciálně úspěšná strategie „`complement`“, která spočítá doplněk do 100 od soupeřovy předchozí požadované částky. Střídání částek u „`take n`“ s osmi variantami je možné také zjednodušit na čtyři alternativy, kdy by hráč požadoval vždy 0, 25, 50 či 75 z celkového zisku 100.

Požadované částky soupeře za předchozí kola se dají také zprůměrovat a nastavit si je jako vlastní požadovanou částku v aktuálním kole. Takovou strategii je možné nazvat například

„mean last 5 required of partner“. Z důvodu toho, že některé strategie pracují s historií tahů hráčů, jsou také navrženy další dva atributy agentů, do kterých jsou ukládány požadované a reálné částky. Experimenty je možné urychlit zkrácením historie na posledních deset tahů. Rovněž je také vhodné atributy reprezentující seznamy nastavit v inicializační proceduře na prázdné seznamy, do kterých je následně možné vkládat více hodnot.

Pro určení úspěšnosti strategií jsou klíčové celkové výplaty obou hráčů za více kol hry, popřípadě jejich porovnání s celkovými požadovanými částkami. Rozdíl mezi celkovou požadovanou částkou ve více kolech a celkovou reálně získanou výplatou je neúspěšně rozdělený zisk. U kooperativní varianty by v případě neúspěšného rozdělení zisku bylo možné dále vyjednávat o snížení požadavků na výplaty.

3.1.2 Boj o zdroje

Cílem modelu je vytvořit modifikovanou variantu hry na jestřába a hrdličku (Hawk Dove). Ve hře se setkávají dva ptáci stejného druhu, kteří mohou při boji o vzácné zdroje zaujmout jeden ze dvou možných postojů, aktivní (agresivní) či pasivní.

Výše zisku je určena dle výplatní matice navržené pro modifikovanou variantu Hawk Dove v tabulce 5. V jednom kole hry jsou jedincům k dispozici dvě možné akce. Agenti by si měli tyto akce v aktuálním kole ukládat pomocí atributů, stejně jako vícekolovou strategii. Změna strategie by měla být umožněna také na základě hodnoty příslušného výběracího prvku v uživatelském rozhraní modelu.

Navrhovanými prvky uživatelského rozhraní jsou tlačítka spouštějící inicializační (setup) a aktualizací procedury (go, go-once). Odehrání jednoho kola lze provést pomocí „go-once“. Pozorovatel modelu by měl také vidět skóre obou jedinců, ať již v aktuálním kole, tak celkově za všechna kola aktuální hry. Navrhovaným způsobem indikace aktuální akce agentů je změna barvy v grafickém výstupu modelu, červená pro agresivitu a zelená pro pasivitu.

Akci agentů mohou monitorovat také příslušné monitorovací boxy. Pro monitorování aktuálního a celkového skóre je vhodné přiřadit agentům oddělené atributy, přičemž celkové skóre by bylo aktualizováno vždy přičtením aktuálního skóre po vyhodnocení výplatní matice. V dalším kole bude dílčí skóre opět přenastaveno.

Strategie byly navrženy na základě strategií opakovaného věžňova dilematu, popsaných v kapitole 2.2.2. Obdobně jako u opakovaného věžňova dilematu jsou strategie stálé spolupráce či zrady, u modelu Hawk Dove mohou tyto strategie představovat trvalá pasivita a trvalá agresivita („passive“ a „aggressive“). Akce může být zvolena také náhodně („random“). Oplácení předchozí agresivity/pasivity představuje „tit for tat“ a předchozích dvou kol agresivity (jinak pasivita) „tit for two tats“. V nemilosrdné strategii „unforgiving“ je hráč agresivní, pokud byl on či soupeř v předchozím kole agresivní. Úspěšnost této strategie by mohla být obdobná jako „aggressive“ či „tit for tat“, avšak může být ovlivněna ztrátou bodů za boj.

Jednou z navrhovaných strategií je také „passive twice aggressive twice“, která po dvou kolech střídá pasivitu a agresivitu. Dalšími kombinovanými strategiemi jsou „tft & random aggressive 33%“ a „tft & random forgive aggressivity 33%“. První jmenovaná je stejná jako „tit for tat“, ale třetinu případů tvoří náhodná složka, kdy může být agent agresivní nezávisle na předchozím tahu soupeře. Druhá jmenovaná strategie kombinuje „tit for tat“ a náhodné odpuštění agresivity ve třetině případů.

Pokud to strategie vyžaduje, hráči se rozhodují pouze na základě známých předchozích tahů soupeře. Aktuální tah není znám. Historii si mohou agenti ukládat do vlastního atributu history redukováného na posledních deset tahů a při rozhodování na základě předchozích tahů soupeře se mohou agenti „dotazovat“. Komunikace agentů je však možná pouze při rozhodování na základě předchozích akcí či při vyhodnocování kola. Modelováno je nekooperativní chování, stejně jako u věžňova dilematu. V aktuálním kole je rozhodnutí agentů vzájemně nezávislé.

3.1.3 Dvourozměrné prostorové věžňovo dilema (SPD)

Cílem modelu dvourozměrného prostorového věžňova dilematu je implementovat hru věžňova dilematu u políček grafického rozhraní v NetLogu a navrhnout a implementovat strategie. Jednotlivé políčka představují hráče obývajících danou oblast. Každé políčko má osm sousedů, přičemž navrhovaným způsobem hry je, že v jednom kole políčko odehraje hru se všemi sousedy, jak bylo popsáno v kapitole 2.2.5. V jednom kole může zvolit jednu ze dvou možných akcí, spolupráci či zradu, stejně jako u opakovaného věžňova dilematu.

Strategie byly navrženy na základě strategií opakovaného věžňova dilematu popsaných v kapitole 2.2.2. Jsou obdobné jako u modelu boje o zdroje. Strategie „random“ volí akci náhodně mezi spoluprací a zradou. Trvalá spolupráce představuje strategii „cooperate“

a trvalá zrada „defect“. Oplacení zrady/spolupráce souseda je „tit for tat“ a oplacení zrady za předchozí dvě kola (jinak spolupráce) „tit for two tats“.

Navrhovanou nelítostnou strategií je „unforgiving“, která nastavuje zradu tehdy, zradil-li v předchozím kole hry hráč či jeho souseď. Spolupráci se souseďem v sudém kole a v lichém kole zradu představuje strategie „even cooperate odd defect“ a dvakrát spolupracovat a dvakrát zradit strategie „cooperate twice defect twice“. Kombinace „tit for tat“ a náhodné zrady souseda ve třetině případů nezávisle na jeho předchozím tahu je „tft & random defect 33%“. Náhodné odpuštění předchozí zrady souseda ve třetině případů, jinak hraní strategie „tit for tat“, může být nazváno „tft & random forgive defect 33%“.

Pro některé strategie je nutné zaznamenávat předchozí tahy s různými souseďy, například dvěma seznamy o shodném počtu prvků, ve kterých by bylo indikováno označení políčka následované tahy za poslední dvě kola hry. Jeden seznam obsahuje tahy hráče s daným souseďem a druhý tahy souseda. Oba obsahují stejné číselné označení osmi souseďů. Celková délka těchto seznamů je tedy 24. Protože jsou políčka v grafickém rozhraní NetLoga jednoznačně indikována souřadnicemi x a y , je pro zjednodušení možné zavést unikátní číselné označení stejně jako u agentů. To může být nastavováno pomocí globální proměnné, jehož hodnotu si políčko nastaví jako vlastní číselné označení a potom hodnotu navýší o jedna, aby ji mělo následující políčko spouštějící inicializační procedury unikátní.

Vyhodnocení jednoho kola může být provedeno na základě výplatní matice navržené v tabulce 1. Záporné výše užítka mohou být převedeny na kladné vztahem

$$\text{nový užitek}_{ij} = (-1 \cdot \text{nejnižší starý užitek}) + \text{starý užitek}_{ij}$$

kde i jsou indexy příslušného řádku a j indexy příslušného sloupce výplatní matice v tabulce 1 a staré užítky jsou záporné či nulové. Výplatní matici pro model prostorového věžňova dilematu reprezentuje tabulka 6.

Tabulka 6: Výplatní matice modelu dvourozměrného prostorového věžňova dilematu

		Hráč 2	
		Spolupráce (neudat)	Zrada (udat)
Hráč 1	Spolupráce (neudat)	6, 6	0, 7
	Zrada (udat)	7, 0	3, 3

Zdroj: vlastní zpracování

Jak lze vidět v tabulce 6, pokud hráč i jeho soused spolupracují, oba získají šest bodů užitku. Nulový užitek získá spolupracující hráč tehdy, pokud jej soupeř zradí. Soupeř získává nejvyšší možný počet bodů, a to sedm. Zrada spolupracujícího souseda je odměněna sedmi body a soused nic nezískává. Jestliže se hráč i jeho soused navzájem zradí, oba získávají tři body. V případě osmi sousedů je možné skóre jednoho kola vypočítat jako součet z dílčích her. Výpočet skóre různých strategií v jednom kole lze určit sečtením skóre jedinců stejné strategie a vydělením počtu jedinců strategie.

Vyhodnocení úspěšnosti strategií je vhodné provést také na základě doby přežití strategií či indikovat vývoj počtu jedinců strategií v čase. Předpokladem je jisté ustálení populací, kdy již neprobíhá žádná evoluce. Na globální úrovni je možné změřit čas, který uplynul od poslední evoluce, přičemž pokud proběhne evoluce, je čas vynulován. Za dostatečnou dobu pro zaznamenání doby ustálení by mohlo být považováno sto kol, ve kterých neproběhla žádná změna struktury populací.

Navrhovanou podkladovou informací pro evoluci políček je maximální hodnota ze souhrnného skóre jedinců sousední strategie vyděleného počtem sousedů hrající danou strategii v jednom kole (skóre sousední strategie na jedince). Vyhodnocení strategií na globální úrovni je možné provádět pomocí seznamu, který bude obsahovat název strategie následovaný hodnotou skóre strategie. Globální skóre strategie se vypočítá jako aktuální hodnota plus suma skóre jedinců dané strategie vydělená jejich počtem v daném kole. Populace strategií evolučního modelu je proměnlivá a tak je vhodné počítat skóre strategie na jednoho jedince.

Model může být porovnán s možností vypnutí evoluce v uživatelském rozhraní. Dalšími navrhovanými prvky uživatelského rozhraní jsou tlačítka spouštějící inicializační procedury (setup) a aktualizací procedury (go, go-once), a rovněž také monitorovací prvky časů od poslední evoluce políček a času ustálení modelu, graf vyobrazující počty jedinců jednotlivých strategií za posledních sto kol a výstupní textová oblast pro vypsání času vymizení jedinců dané strategie.

Počet ani strukturu strategií není možné měnit, po iniciaci modelů je počet jedinců v populacích strategií stejný a měnit se může až v evolučním procesu. Změnou velikosti grafického výstupu modelu je dosaženo změny počtu hráčů. Počet strategií je deset a tak by měl být počet políček dělitelný deseti, což může být v inicializační proceduře ověřováno

a směže být uživateli modelu zobrazena chyba. Nepředpokládá se, že by v modelu existovala políčka bez nastavené strategie, ani že by se takovým políčkům měla nastavovat například náhodná strategie.

Předpokladem evolučního modelu je dobrá adaptace některých strategií, obzvláště „tit for tat“, u ostatních strategií ukáží úspěšnost a přežití experimenty. Jedinci hrající strategii zrady se mohou obohatit na spolupracujících jedincích a tak může spolupracující strategie přežít jedině v uzavřených komunitách či vůbec. Rozložení oblastí jednotlivých strategií je totiž náhodné. Jestliže některé strategie neuspějí v procesu evoluce, stále mohou uspět v neevolučním modelu, kde je struktura generované populace neměnná.

3.2 Implementace

Pro implementaci modelů bylo zvoleno NetLogo ve verzi 5.2.1. Následující podkapitoly obsahují popis implementovaných částí jednotlivých modelů, uživatelského rozhraní i vstupních parametrů. Modely byly implementovány na základě popisu modelů a analýzy klíčových faktorů v kapitole 3.1.

Společnými prvky uživatelského rozhraní všech modelů jsou tlačítka setup, go a go-once. Po stisknutí tlačítka setup je provedeno úvodní nastavení modelu před započítím hry. Tlačítko go-once volá aktualizací proceduru go pouze jednou a je odehráno pouze jedno kolo hry. Go volá proceduru go opakovaně, tlačítko zůstává po kliknutí stisknuté a dalším kliknutím se dá zastavit.

3.2.1 Sosisova hra v kibucech

Soubor s modelem Sosisovy hry v kibucech je pojmenován „Sosis game“ a lze jej nalézt na příloženém CD, viz příloha 2).

Globální proměnné

Model obsahuje jednu globální proměnnou, a to remaining_earnings, do které je ukládána zbylá částka do 100 po součtu požadovaných zisků obou hráčů. Poté je částka navýšena o polovinu a je rozdělena hráčům jako bonus za skromnost. Vyhodnocení tak proběhne z globálního hlediska modelu nezávisle na obou agentech. Podle proměnné remaining_earnings je také nastavována podoba agentů.

Atributy agentů

V multiagentovém modelu Sosisovy hry je implementován jeden druh agentů, pro který jsou generováni dva zástupci se stejnými atributy, neboť se jedná o dva lidské jedince. Atributy hráčů jsou `strategy`, `required_earnings`, `real_earnings`, `required_earnings_sum`, `real_earnings_sum`, `required_earnings_history`, `real_earnings_history`.

Strategy představuje vícekolovou strategii hráče, podle které se nastavuje akce pro jednotlivá kola hry. V atributu `required_earnings` je uložena požadovaná částka z celkového zisku v aktuálním kole hry a v atributu `real_earnings` částka, kterou hráč reálně získá po vyhodnocení kola. Požadované a reálné částky se u jednotlivých hráčů za všechna kola hry sčítají (odpovídající atributy `required_earnings_sum` a `real_earnings_sum`) a slouží také jako kritérium pro vyhodnocení úspěšnosti strategií.

Požadované a reálné zisky si hráči za posledních deset kol hry pamatují (`required_earnings_history` a `real_earnings_history`) a jsou také využity jako podklad pro volbu akce u některých strategií. Atributy `required_earnings_history` a `real_earnings_history` jsou tedy seznamy.

Uživatelské rozhraní

V průběhu hry je možné měnit strategii obou hráčů v uživatelském rozhraní modelu přes textové výběrací prvky `strategy_0` a `strategy_1`. 0 či 1 jsou číselná označení hráčů generovaná při spuštění prvotního nastavení modelu tlačítkem `setup`, 0 je přiřazena hráči vlevo a 1 hráči vpravo v grafickém výstupu.

Dále uživatelské rozhraní modelu Sosisovy hry obsahuje osm monitorovacích prvků. Horní čtyři zobrazují požadované (`required_earnings_0` a `required_earnings_1`) a reálné (`real_earnings_0` a `real_earnings_1`) zisky obou hráčů v aktuálním kole. Názvy monitorovacích prvků obsahují na konci číselné označení agenta. Další čtyři monitory vyobrazují celkové součty požadovaných a reálných zisků za všechna kola hry daného hráče.

Grafický výstup po dokončení inicializační procedury zobrazuje dva hráče, jejichž podoba představuje výraz, který indikuje úspěšnost zvolené akce po vyhodnocení kola. Barva v úvodním nastavení modelu generována pro oba agenty náhodně. Označení agentů je `Player + číselné označení who`, které je výchozím atributem agenta v NetLogu.

Šťastný výraz mají oba hráči, pokud je součet požadovaných částek menší než sto a oba obdrží ještě polovinu ze zbylého zisku navýšeného o polovinu. Hráčům je nastavena zelená barva. Neutrální výraz představuje nulový zbylý zisk, kdy oba agenti v součtu požadovali přesně tolik, kolik tvoří zisk pro rozdělení, tedy 100. V takovém případě je barva agentů šedá. Smutný výraz indikuje větší součet požadovaných částek než 100. Oba hráči mají nulové reálné zisky. Zároveň je oběma agentům nastavena červená barva.

Inicializační procedury

V modelu Sosisovy hry je inicializační procedurou pouze „setup“, která je vyobrazena na následující ukázce kódu.

```
1 to setup
2   ca
3   reset-ticks
4   crt 2
5   [ set size 2
6     set label (word "Player " who)
7     set shape "face neutral"
8     set required_earnings_history []
9     set real_earnings_history []
10    ifelse who = 0
11      [ set heading 270
12        fd 1.5
13      ]
14      [ set heading 90
15        fd 1.5
16      ]
17    ]
18 end
```

Zpočátku prvotního nastavení jsou údaje o předchozí hře vynulovány, včetně počítadla počtu kol ticks. Poté jsou vytvořeni dva hráči, kterým jsou nastaveny počáteční hodnoty atributů. Velikost v grafickém rozhraní je nastavena na 2 a do popisku hráčů je kromě anglického pojmenování Player dynamicky přidáno také jejich číselné označení. Výchozí výraz hráčů je neutrální. Atributy sledující historii požadovaných a reálných zisků jsou nastaveny na prázdná pole, aby do nich bylo možné v aktualizací proceduře vkládat hodnoty. Podle označení hráčů je nastavena i jejich poloha v grafickém výstupu prostřednictvím podmínky ifelse. Hráč s označením 0 je umístěn vlevo (set heading 270 fd 1) a druhý hráč vpravo.

Strategie

Procedura zajišťující volbu akce v daném kole dle zvolené strategie agenta se jmenuje `set_required_earnings`. Obsahuje 19 podmínek, které odpovídají 19 vícekolovým strategiím hráčů. Vždy je splněna pouze jedna z podmínek porovnávající dva textové řetězce. Prvním z nich je atribut hráče `strategy`, který již v momentě porovnávání má uloženou strategii pro aktuální kolo. Druhá je název jedné ze strategií, kterou je možné vybrat. Volbou akce v jednom kole hry je myšleno nastavení požadované částky z celkového zisku 100.

V rámci strategie `random` je hráči nastaven požadovaný zisk na pseudonáhodnou celočíselnou hodnotu od 0 do 100 příkazem „`random 101`“, který generuje přibližně rovnoměrně rozdělené hodnoty mezi 0 a zadaným číslem, viz dokumentace k NetLogu na [46] a vyhodnocení testů generátoru `random` v kapitole 3.3 a v souboru „`Random discrete NetLogo generator test.xlsx`“ na přiloženém CD, obsah viz příloha 2). Strategie `0.5_profit` nastavuje požadovanou částku na 50, `take_all` na 100 a `invest_all` na 0.

U strategií `take_10`, `take_20`, `take_30`, `take_40`, `take_60`, `take_70`, `take_80` a `take_90` je vždy zvolena příslušná částka, která je uvedena v názvu těchto strategií na posledních dvou místech. Implementace těchto strategií nahrazuje původně navrhovanou strategii `take_n`, kdy by se konstanta n vybírala v uživatelském rozhraní pro oba hráče. Dostupné byly tytéž hodnoty. Strategie `take_n` byla rozdělena na dílčí částky v různých strategiích pro zjednodušení experimentů. Hodnoty dalšího výběracího prvku by nebyly při experimentech přiřazeny pouze ke strategii `take_n` pro $n = \{10, 20, 30, 40, 60, 70, 80, 90\}$ ve hře proti ostatním strategiím, ale byla by provedena také kombinace hodnot n se strategiemi, které n při hře nevyžadují.

V prvním kole strategie `tit_for_tat` je akce volena stejně jako u strategie `random` a v dalších kolech hráč požaduje takový zisk, který zvolil jeho soupeř v posledním kole. Počet akcí pro jedno kolo hry se rozšiřuje ze dvou na sto a nelze jednoznačně určit, která částka by znamenala spolupráci se soupeřem jako výchozí situace prvního kola. Strategie `complement` v prvním kole také předpokládá volbu akce také pseudonáhodně (`random`), avšak ve všech ostatních kolech je požadovaný zisk hráče nastaven jako rozdíl 100 a poslední akce soupeře.

Jestliže nebylo dosaženo reálného zisku nebo se hraje první kolo, tak je akce u strategie `repeat_if_real_earn` nastavena stejně jako u `random`, jinak je požadována stejná částka,

kteřá vedla k dosažení reálného zisku. Strategie repeat_4 v jednotlivých kolech hry vybírá náhodně (one-of) mezi hodnotami 0, 25, 50, 75. Implementace strategie real_as_required je realizována tím způsobem, že v prvním kole se hraje strategie random a v následujících je požadováno tolik, jaký byl hráčův poslední reálný zisk.

Předposlední implementovanou strategií je repeat_if_real_earn_and_random75_33%. Nastavení požadované částky na 75 přibližně ve třetině kol hry je dosaženo porovnáním $\text{if random} \leq 0.33$. Tato podmínka je splněna přibližně ve třetině případů, kdy je číslo 0, 1 nebo 2, generované s přibližně stejnou pravděpodobností, rovno 1.

Poslední implementovanou strategií modelu Sosisovy hry je mean_last_5_required_of_partner, jejíž implementaci lze vidět na následující ukázce kódu. První kolo je odehráno stejně jako u strategie random. V dalších kolech je nejprve ověřována podmínka, zda je délka historie soupeřových požadovaných částek menší či rovna než pět. Pokud ano, je částka v aktuálním kole nastavena na hodnotu průměru historických částek, které požadoval soupeř (maximálně pěti).

Jestliže je odehráno více než pět kol, je do dočasně vytvořené proměnné partner_required_earnings_history uložena historie požadovaných zisků soupeře v předchozích kolech. Dokud počet částek v dočasné proměnné přesahuje pět, je z dočasné historie odebrána první položka. Tímto postupem se získá posledních pět požadovaných částek soupeře z historie čítající více než pět kol. Částka v aktuálním kole je poté nastavena na průměr z posledních pěti požadovaných částek ze soupeřovy historie.

```
1 if strategy = "mean_last_5_required_of_partner"
2   [ ifelse ticks = 0
3     [ set required_earnings random 101 ]
4     [ ifelse length [required_earnings_history] of one-of other turtles
5       <= 5
6         [ set required_earnings mean [required_earnings_history] of
7           one-of other turtles ]
8         [ let partner_required_earnings_history
9           [required_earnings_history] of one-of other turtles
10          while [length partner_required_earnings_history > 5]
11            [ set partner_required_earnings_history remove-item 0
12              partner_required_earnings_history ]
13          set required_earnings mean partner_required_earnings_history
14        ]
15      ] ]
```

Aktualizační procedury

Implementace odehrání jednoho kola je provedena s pomocí aktualizační procedury `go`. Deklarace procedury začíná výrazem „`to go`“. Na začátku každého herního kola je oběma hráčům nastavena strategie dle vybrané hodnoty z textového vyběracího prvku příslušného agenta. Tímto je umožněna změna strategie před započítáním libovolného kola hry. V té samé deklaraci volání hráčů je volána procedura `set_required_earnings`. Ta obsahuje volbu akce agenta v daném kole dle jeho strategie. Implementace volby akce byla popsána v předcházející části Strategie.

Následně jsou hráči osloveni, aby si aktualizovali historii požadovaných zisků vložení aktuálního požadovaného zisku za poslední prvek (`lput`) historie požadovaných zisků. Poté je ověřena délka seznamu historie zisků, a pokud je větší než deset, je odebrán první prvek (nejstarší akce – požadovaný zisk o 10 kol zpět) příkazem `remove-item 0`.

Aktualizování historie obou hráčů je provedeno v odděleném volání agentů, neboť některé strategie využívají vyhodnocování předchozích tahů soupeře. NetLogo při volání agentů oslovuje všechny postupně a tak by jeden hráč znal aktuální tah toho, u koho již byla provedena procedura s nastavením akce v daném kole. Aktualizace historie tahů se tak provádí u obou agentů až poté, co mají oba nastavenou akci pro aktuální kolo.

Aktualizační procedura `go` je globální a tak lze rovnou nezávisle na obou hráčích vyhodnotit podmínku, zda součet požadovaných zisků je menší nebo roven 100. Podmínku zobrazuje následující ukázka kódu.

```
1  ifelse ([required_earnings] of turtle 0 + [required_earnings] of turtle
2  1) <= 100
3    [ set remaining_earnings (100 - ([required_earnings] of turtle 0 +
4    [required_earnings] of turtle 1))
5    set remaining_earnings remaining_earnings + (remaining_earnings / 2)
6    ask turtles
7      [ set real_earnings required_earnings + (remaining_earnings / 2)
8        ifelse remaining_earnings = 0
9          [ set shape "face neutral"
10           set color grey
11           ]
12         [ set shape "face happy"
13           set color green
14           ]
15      ]
16  ]
17  [ ask turtles
```

```
18     [ set real_earnings 0
19         set shape "face sad"
20         set color red
21     ]
22 ]
```

Jak lze vidět z ukázky kódu, pokud je podmínka splněna, je do globální proměnné `remaining_earnings` uložen rozdíl $100 - (\text{požadovaný zisk hráče } 0 + \text{pož. zisk hráče } 1)$ a následně je `remaining_earnings` ještě navýšena o polovinu. Poté je oběma hráčům nastaven reálný zisk v aktuálním kole hry na požadovaný zisk + hodnota uložená v `remaining_earnings` dělená dvěma (oba hráči si zbylý zisk navýšený o polovinu rozdělí rovnoměrně).

Následuje již zmíněné nastavování výrazu a barvy hráče dle toho, zda je zbylý zisk `remaining_earnings` nulový či vyšší, odpovídající výrazy neutrální či šťastný, barvy šedá a zelená. Pokud je součet požadovaných zisků větší než 100 (část podmínky `else`), jsou oběma hráčům nastaveny nulové reálné zisky, jejich výraz v grafickém rozhraní je nastaven na smutnou tvář a barva je červená.

Poslední skupina příkazů v proceduře `go` směřovaná na agenty obsahuje deklaraci aktualizace historie reálných zisků. Obdobně jako u historie požadovaných zisků je atribut obsahující historii reálných zisků redukován na posledních 10 tahů a reálný zisk aktuálního kola je vkládán na konec. Aktualizace historie reálných zisků je provedena až poté, co oba hráči znají reálný zisk aktuálního kola.

Ve stejném bloku příkazů jsou také nastavovány součty požadovaných a reálných zisků obou agentů za všechna kola hry. Hodnota příslušných atributů obsahujících dané sumy je jednoduše nastavena na součet vlastní hodnoty a aktuálních požadovaných nebo reálných zisků. Například aktualizace součtu požadovaných zisků vypadá následovně:

```
1 set required_earnings_sum required_earnings_sum + required_earnings
```

3.2.2 Boj o zdroje

Soubor s modelem boje o zdroje je pojmenován „Hawk Dove game modified“ (v následujícím textu zkráceně Hawk Dove) a lze jej nalézt na přiloženém CD, viz příloha 2).

Atributy agentů

Model Hawk Dove nemá implementovány žádné globální proměnné. Atributy agentů jsou `strategy`, `score`, `score_sum`, `aggressive?` a `history`. Atributy obou agentů jsou stejné, neb se jedná o dva ptačí jedince stejného druhu. Jejich odlišnost spočívá především ve strategiích.

Strategy je vícekolovou strategií hráčů, na základě které se nastavují akce v dílčích kolech hry. Score je počet bodů agenta v aktuálním kole a score_sum je součtem bodů získaných za všechna kola hry. Atribut aggressive? obsahuje chování v jednom kole. Jedná se o hodnotu s návratovými typy true či false (pravda/nepravda). True reprezentuje agresivitu ptáka a false pasivitu. Pasivita je analogií ke spolupráci u opakovaného věžňova dilematu a agresivitu je možné chápat jako zradu. Akce hráčů za posledních deset kol hry jsou ukládány do atributu history (seznam).

Uživatelské rozhraní

Kdykoliv v průběhu hry a také na začátku hry lze změnit strategie agentů v uživatelském rozhraní modelu prostřednictvím textových výběracích prvků strategy_0 a strategy_1. 0 či 1 jsou číselná označení generovaná při spuštění procedury setup, 0 je přiřazena hráči vlevo a 1 hráči vpravo v grafickém výstupu modelu.

Uživatelské rozhraní modelu Hawk Dove obsahuje šest monitorovacích prvků. Horní dva monitorovací prvky vyobrazují akci agentů v aktuálním kole. Jsou nazvány „bird 0 aggressive?“ a „bird 1 aggressive?“. Obsahují příkazy „[aggressive?] of turtle 0“ a „[aggressive?] of turtle 1“. Následují dva monitory pro zobrazení aktuálního skóre obou agentů. Monitor „bird 0 score“ se odkazuje na skóre ptáka s označením 0 příkazem „[score] of turtle 0“. Analogicky monitor „bird 1 score“. Spodní dva monitory („bird 0 score sum“ a „bird 1 score sum“) zobrazují celkové skóre agentů za všechna kola hry deklarací „[score_sum] of turtle 0“ či „[score_sum] of turtle 1“.

Grafický výstup modelu zobrazuje po dokončení inicializační procedury dva hráče (ptáky) s ikonou „Bird“. Tato ikona je importována z knihovny Turtle Shapes editor a je nutné ji pro funkčnost modelu importovat v lokální verzi NetLoga či přenastavit ikonku agentů v kódu modelu. Import ikonky „Bird“ je možné provést z horní záložky menu NetLoga Tools, poté Turtle Shape editor, Import from Library..., vybrat ikonku Bird a potvrdit tlačítkem Import.

Agenti jsou v grafickém výstupu modelu pojmenováni jako Bird + číselné označení who, které je výchozí vlastností agentů NetLoga. V průběhu hry je agentům aktualizována barva podle jejich akce, pro agresivního jedince je nastavena červená a pro pasivního zelená. Odpovídající příkazy pro změnu barvy jsou set color red a set color green. Color je výchozím atributem agenta v NetLogu.

Inicializační procedury

Inicializační procedurou je „setup“. V první řadě jsou vynulovány údaje o předchozí hře, včetně počítadla počtu kol ticks. Jsou vytvořeni dva agenti, kterým jsou přiřazeny výchozí hodnoty atributů. Velikost agentů v grafickém rozhraní je nastavena na 1,5 a do popisku hráčů je přiřazeno pojmenování Bird s číselným označením agenta. Ikonka hráčů je nastavena jako „Bird“. Ta je importována již zmíněným způsobem v kapitole Uživatelské rozhraní.

Vzhledem k tomu, že history je seznam, je pro následné úpravy zpočátku nastaven na prázdný seznam. Agent 0 je umístěn v grafickém rozhraní nalevo příkazy set heading 270 a fd 1.5, agent 1 napravo nastavením opačného nasměrování set heading 90 a posunutím vpřed o 1,5 políčka. Při iniciaci mají agenti náhodnou barvu.

Strategie

Procedura zajišťující volbu akce v jednom kole dle zvolené strategie agenta je nazvána set_action a obsahuje deset dílčích podmínek odpovídajících deseti strategiím, z nichž je vždy proveden blok příkazů jen u jedné z nich. Volba akce v jednom kole hry je implementována pomocí nastavení atributu aggressive? na true (agresivita) nebo na false (pasivita). Strategie jsou obdobné jako u opakovaného vězňova dilematu, s tím rozdílem, že stejné strategie mohou být jinak úspěšné, neboť mohou ptáci v modelu Hawk Dove získat i záporné skóre.

Strategie random nastavuje přibližně ve stejném počtu případů pasivitu či agresivitu. Ověření kvality generátoru diskretních pseudonáhodných čísel z rovnoměrného rozdělení viz vyhodnocení testů generátoru random v kapitole 3.3 a příložené CD, příloha 2), soubor „Random discrete NetLogo generator test.xlsx“. Volba akce dle strategie passive je provedena příkazem „set aggressive? false“. Agresivita ve všech kolech hry je nastavena prostřednictvím „set aggressive? true“.

V prvním kole strategie tit_for_tat je nastavena pasivita (obdoba spolupráce) a v ostatních kolech je ověřována podmínka, zda při posledním setkání byl soupeř agresivní či nikoliv. Pokud ano, je zvolena agresivita v aktuálním kole, pokud ne, tak pasivita. U tit_for_two_tats je v prvních dvou kolech je nastavena pasivita a agresivita je v dalších kolech zvolena pouze tehdy, pokud byl soupeř v posledních dvou kolech agresivní. Ověření podmínky agresivity představuje následující ukázka kódu.


```

1  ifelse (last [history] of one-of other turtles)
2  AND (item (length history - 2) [history] of one-of other turtles)
3    [ set aggressive? true ]
4    [ set aggressive? false ]

```

Podmínka má dvě části, první ověřuje poslední prvek historie akcí soupeře a druhá vrací prvek na předposlední pozici historie soupeřových tahů. Oba ptáci mají stejnou délku seznamu historie tahů z totožných předchozích kol. Předposlední prvek soupeřovy historie je tak možné získat jako položku (item) na pozici *length history - 2* v historii soupeře, kde history je vlastní historie akcí, length její délka a -2 je odečítáno pro získání prvku seznamu na pozici $n - 1$ (předposlední), neboť indexy prvků v seznamu v NetLogu začínají od 0. Agresivita je nastavena jen tehdy, jsou-li splněny obě části podmínky, jinak je akcí v aktuální kole pasivita.

Unforgiving nastavuje pasivitu v prvním kole a v dalších kolech postačí splnit vlastní či soupeřovu agresivitu v posledním kole, obě části jsou odděleny deklarací OR. Změna mezi pasivitou v sudém kole a agresivitou v lichém kole je provedena na základě ověření celočíselného zbytku po dělení čítače počtu kol ticks a čísla 2. Jestliže je zbytek po tomto dělení nulový, jedná se o sudé kolo, jinak o liché.

Implementovaná strategie passive_twice_aggressive_twice předpokládá odehrání prvních dvou kol s pasivitou, dalších dvou s agresivitou, dvě s pasivitou atd. Přepínání agresivity a pasivity po dvou kolech zajišťuje podmínka sudosti čítače kol a zároveň ověření, zda bylo předchozí kolo odehráno s pasivitou (not last history) či agresivitou (last history). V prvním případě je na následující dvě kola nastavena agresivita a ve druhém případě pasivita.

Předposlední strategií modelu Hawk Dove je tft_&_random_aggressive_33%. Přibližně ve třetině případů je random 3 = 1, tehdy je akcí pro aktuální kolo agresivita, jinak je kolo odehráno dle strategie tit_for_tat. V prvním kole je nastavena pasivita.

U poslední strategie tft_&_random_forgive_aggressivity_33% je první kolo rovněž odehráno s pasivitou. V následujících kolech je nejprve ověřena podmínka, zda soupeř v předchozím kole byl agresivní. Pokud ano, je přibližně ve třetině případů nastavena akce na pasivitu i přes předchozí zradu soupeře a ve zbylých přibližně dvou třetinách případů je nastavena agresivita v souladu se strategií tit_for_tat. Jestliže soupeř v předchozím kole nebyl agresivní, tak je v aktuálním kole zvolena pasivita.

Aktualizační procedury

V modelu je implementována jedna aktualizační procedura (go) a jedna vnořená procedura set_action, která je spuštěna vždy oběma agenty v jednom kole po nastavení příslušných strategií. Procedura go představuje odehrání jednoho kola. Strategie je hráčům nastavena dle vybrané hodnoty z textového vybíracího prvku příslušného agenta (strategy_0 či strategy_1). Ve stejné deklaraci volání hráčů je spuštěna procedura set_action, jejíž účelem je nastavit akci v jednom kole. Implementace volby akce byla popsána v předchozí části Strategie.

Následuje vyhodnocení hry, které obsahuje všechny možné situace implementované ve vnořených podmínkách. Vyhodnocení je provedeno v odděleném volání než nastavení akce. Implementace výplatní matice a vyhodnocení viz následující ukázka kódu.

```
1  ifelse aggressive?
2    [ ifelse ([aggressive?] of one-of other turtles)
3      [ set score -2 ]
4      [ set score 8 ]
5    ]
6  [ ifelse ([aggressive?] of one-of other turtles)
7    [ set score 0 ]
8    [ set score 4 ]
9  ]
10 set score_sum score_sum + score
11 ifelse aggressive?
12 [ set color red ]
13 [ set color green ]
```

Podmínka na prvním řádku kódu ověřuje aktuální volbu hráče. Návrátovou hodnotou je true („jsem agresivní“) či false („jsem pasivní“). Pokud je navráceno true, je proveden blok příkazů v první části podmínky ifelse, pro false druhý blok příkazů. V první i ve druhé části podmínky ifelse je následně vyhodnocována akce soupeře, opět s návratovými typy true či false.

Snížení celkového skóre je provedeno pouze tehdy, jsou-li oba agenti agresivní a bojují o cenné zdroje. Implementaci vyhodnocení situace, kdy jsou oba ptáci agresivní lze vidět na třetím řádku předcházející ukázky kódu. Celkové skóre je v tomto případě sníženo o 2 body. Pták v boji se soupeřem ztratil energii. Jestliže je soupeř pasivní, získává agresivní pták nejvyšší možný počet bodů v jednom kole hry, a to 8. Tuto situaci představuje implementace na čtvrtém řádku ukázky kódu.

Na sedmém řádku kódu není provedena žádná změna skóre, jedná se o situaci, kdy pasivní pták při setkání s agresivním soupeřem nezíská žádné body, veškeré zdroje získá agresivní soupeř, který je ochoten bojovat. Pasivní pták ustupuje agresivnímu. Osmý řádek reprezentuje vyhodnocení střetu dvou pasivních ptáků, kteří se o zdroj rovnoměrně podělí. Je splněna „else“ část i u nadřazené podmínky. Na desátém řádku je přičteno aktuální skóre k celkovému a poté je aktualizována barva příslušného ptáka v grafickém rozhraní modelu. Pokud je pták agresivní (`aggressive? = true`, zkráceně lze využít pouze `aggressive?`), je mu nastavena červená barva, jinak zelená.

Po vyhodnocení aktuální hry probíhá aktualizace historie akcí. Ta je implementována odděleně od nastavení akcí stejně jako vyhodnocení hry, neboť některé strategie využívají vyhodnocování předchozích tahů soupeře. Pokud by aktualizace historie byla ve stejném volání agentů jako aktualizace akcí, došlo by k situaci, že prvně oslovený agent provede nastavení akce, ale druhý agent by již znal tah soupeře v aktuálním kole. Aktualizace historie je provedena vložím aktuální akce za poslední prvek v seznamu history. Rovněž je ověřena podmínka délky seznamu. Pokud počet prvků přesahuje deset, je odebrán první prvek v historii, který představuje nejstarší akci agenta.

3.2.3 Dvourozměrné prostorové věžňovo dilema

Soubor s modelem dvourozměrného prostorového věžňovo dilematu je pojmenován „Spatial prisoners dilemma“ a lze jej nalézt na příloženém CD, viz příloha 2).

Globální proměnné

V modelu bylo implementováno osm globálních proměnných. První je `strategies`, která slouží jako pomocný seznam obsahující názvy všech strategií modelu. Celkový počet strategií je ukládán v globální proměnné `num_of_strategies` a barvy, které jsou jednotlivým strategiím přiřazeny, jsou uloženy v seznamu `colors`. Jednou z dalších pomocných proměnných je i `set_pwho` sloužící jako čítač při nastavování unikátních číselných identifikátorů políček.

Výpočet skóre jednotlivých strategií je implementován pomocí seznamu místo deseti atributů hodnotového typu. Skóre je ukládáno v `strategies_score_sum`, který obsahuje název strategie a hodnotu skóre strategie. Celková délka seznamu je tak 20 prvků. Obdobně je implementováno i ukládání času vymizení strategií pro evoluční model. Posledními globální proměnnými jsou `time_since_evolution` reprezentující počet kol od poslední proběhlé evoluce a `stabilization_time` pro určení času „stabilizace“ evolučního procesu.

Atributy políček

Políčka mají v první řadě atribut číselného označení pwho (zkrácené patch-who, obdoba who u agentů). Strategy reprezentuje vícekolovou strategii hry, která se může u tohoto modelu měnit. Seznam current_action obsahuje aktuální akci se všemi osmi sousedy. Do current_action je vkládáno pwho souseda a příslušná akce, false (spolupráce) či true (zrada). Atribut history ukládá historii posledních dvou tahů hráče se všemi sousedy v seznamu.

Score je dílčí skóre hráče v jednom kole. Každý hráč v jednom kole odehraje hru se všemi osmi sousedy. Score je součtem všech dílčích výplat. Score_sum je součtem skóre hráčů ve všech kolech hry. Skóre sousedních strategií je ukládáno do atributu neighbors_strategies_score. Představuje průměrné skóre jedinců dané strategie dělené počtem jedinců. Tato informace je důležitá pro evoluci.

Uživatelské rozhraní

Spuštění inicializačních procedur v uživatelském rozhraní modelu je možné pomocí tlačítka setup a aktualizacních procedur pomocí go a go-once. Go spouští aktualizacní procedury opakovaně a je odehráno více kol po sobě, dokud není tlačítko vypnuto. Pod základními tlačítky se nachází dva přepínače (switch). Horní přepínač je nazván evolution? a slouží pro vypnutí či zapnutí evolučních procedur. Spodní přepínač plabel vypíná (hodnota „off“) a zapíná (hodnota „on“) popisky políček, které obsahují označení pwho po iniciaci či skóre v průběhu hry.

Model obsahuje dva monitorovací prvky. „Time since last evolution“ zobrazuje čas od poslední proběhlé evoluce jakéhokoliv políčka modelu, pokud je evoluce zapnutá. „Stabilization time“ zobrazuje čas stabilizace struktury strategií u evoluční varianty po sto kolech od poslední proběhnuvší evoluce. Ve spodní části modelu se nachází graf „Strategies counts“, jehož součástí je také barevná legenda strategií. Graf vyobrazuje vývoj počtu jedinců jednotlivých strategií v průběhu času (Ticks). Napravo od grafu se nachází výstupní textová oblast, ve které jsou vypisovány doby vymizení jedinců strategií a také průměrné skóre vymizevších strategií za kola hry, ve kterých byla populace strategií nenulová.

Grafický výstup modelu zobrazuje po dokončení inicializační procedury políčka modelu, která představují jednotlivé hráče. Barva je nastavena dle strategie a v evolučním procesu

je aktualizována. Lze také vidět strukturu populací strategií. Pokud je přepínač plabel? nastaven na hodnotu „on“, hráči mají po dokončení inicializační procedury nastaven popis s číselným označením. Vpravo od grafického rozhraní se nachází textová legenda strategií.

Inicializační procedury

Model obsahuje dvě inicializační procedury, základní setup a vnořenou `set_strategy` s dvěma parametry. Setup je spouštěna stejnojmenným tlačítkem v uživatelském rozhraní modelu. Zpočátku jsou vynulovány údaje o předchozí hře a počítadlo počtu kol ticks. Poté je do atributu `strategies` uložen seznam všech strategií v textové podobě a proměnná `num_of_strategies` je nastavena na délku seznamu `strategies`.

Pro následnou práci se seznamem celkového skóre všech strategií je `strategies_score_sum` nastaveno na seznam strategií v textové podobě, přičemž za každou strategií je číselná hodnota skóre, výchozí hodnota je 0. Stejným způsobem je nastavena proměnná `strategies_extinction_time`. Výchozí hodnota je také 0, přičemž následná aktualizace reprezentuje okamžik hry, od kterého již neexistuje žádný zástupce strategie. Proměnná `colors` po dokončení procedury `setup` obsahuje strategie a barvy, které jim jsou na základě tohoto seznamu nastaveny.

Následuje implementace upozornění uživatele v případě, že je nastaven počet políček modelu, který není dělitelný deseti, nebo je nastaven menší rozměr grafického výstupu než 3 * 3 políčka. Dělitelnost deseti je ověřena celočíselným zbytkem po dělení počtu políček počtem strategií. Pokud je zbytek modulo jiný než nula (\neq), nelze nastavit rovnoměrné výchozí počty populací strategií a uživateli se po spuštění procedury `setup` zobrazí chybové hlášení „Modulo after dividing the number of patches and 10 must be zero.“. Poté se provádění procedury ukončí.

Rozměry grafického výstupu jsou zjištěny podmínkou „if world-width < 3 OR world-height < 3“, při jejímž splnění se zobrazí chybové hlášení „World size must be at least 3x3.“. Jestliže by šířka či výška byla dvě, potom by měli hráči stejného souseda dvakrát. Aby byl zajištěn stejný počet sousedů u všech hráčů, mají políčka na kraji obrazovky jako sousedy protilehlá políčka.

Nastavení populace strategií ve vnořené metodě `set_strategy` je implementováno v podmínce `while`. Je deklarována dočasná proměnná `i` s hodnotou 0. `While` cyklus je prováděn opakovaně, dokud je `i` menší než délka seznamu strategií (tolikrát, jaký je počet strategií).

Na konci cyklu je i navyšováno o jedna. Uvnitř cyklu je volána procedura `set_strategy` s parametry `strategy_to_set` a `color_to_set`, které jsou dočasnými proměnnými vytvořenými uvnitř cyklu `while` za současného nastavení hodnot. `strategy_to_set` je nastavena na prvek pozice i v seznamu všech strategií, tedy aktuální strategii, jejíž populaci je třeba vytvořit.

Dočasné proměnné `color_to_set` je přiřazena barva strategie deklarací „item $((2 * i) + 1)$ colors“. Výpočet uprostřed reprezentuje index barvy v seznamu `colors`. Hodnota i je vynásobena dvěma, neboť `colors` obsahuje kromě názvu strategií také příslušné barvy a má dvounásobnou délku. Indexy seznamů začínají na nule, tudíž je nutné přičíst jedna. Například barva strategie `cooperate` se v `colors` nachází na čtvrtém místě (index 3). Nastavována je v druhém cyklu, kdy je $i = 1$. Vztahem $2 * 1 + 1$ je získán požadovaný index 3.

```
1 to set_strategy [ strategy_to_set color_to_set ]
2   ask n-of (count patches / num_of_strategies) patches with [strategy =
3     0]
4     [ set strategy strategy_to_set
5       set pcolor color_to_set
6       set pwho set_pwho
7       set set_pwho set_pwho + 1
8       if plabel?
9         [ set plabel pwho
10          set plabel-color black
11          ]
12     set history []
13     set current_action []
14   ]
15 end
```

Předcházející ukázka kódu obsahuje zmíněnou inicializační proceduru, kterou jsou nastavovány výchozí populace strategií. Ke spuštění procedury jsou zapotřebí dva parametry `strategy_to_set` a `color_to_set`, jejichž hodnoty jsou nastavovány již zmíněným způsobem. Uvnitř procedury je osloveno n hráčů z celkového počtu děleného počtem strategií (rovnoměrné populace), které zároveň ještě nemají nastavenou jinou strategii. „Volným“ políčkům se přiřadí strategie z prvního příchozího parametru `strategy_to_set` a barva z druhého `color_to_set`. U políček je výchozí atribut barvy pojmenován `pcolor`.

Atribut `pwho` je nastaven na aktuální hodnotu globálního čítače `set_pwho`, který je poté navyšěn o jedna a je tak připraven pro další políčko. Jestliže je v uživatelské rozhraní modelu zapnuto zobrazení popisek, jsou popisky (`plabel`) po iniciaci nastaveny na unikátní

označení pwho. Po zapnutí či vypnutí plabel? je nutné opětovně spustit inicializační procedury tlačítkem setup. Z důvodu následného zpracování history a current_action příkazem lput („vlož na poslední pozici“), který pracuje pouze se seznamy, jsou tyto atributy v proceduře set_strategy nastaveny jako prázdný seznam.

Po nastavení výchozí struktury populace strategií a označení pwho je možné nastavit výchozí hodnoty dalších atributů. Neighbors_strategies_score je nastaven na stejný seznam jako strategies_score_sum, neboť sousední oblasti jsou vyhodnocovány dle průměrného skóre na jednoho jedince strategie. Pro výchozí nastavení historie tahů je nutné znát čísla všech sousedů. Po oslovení hráče jsou dotázáni i všichni jeho sousedé a poté je provedeno výchozí nastavení seznamů history a current_action v bloku příkazů směřujících na hráče, který souseda oslovil. Právě tehdy jsou známy údaje o obou hráčích.

Do seznamu history je vkládáno pwho souseda a dvakrát výchozí akce spolupráce (defect? = false), která je při hře nahrazována historií tahů. Current_action obsahuje o jednu akci méně, vkládá se pwho souseda a jednou aktuální akce. Postup zjišťování informací o hráči i jeho sousedovi je možné nahradit přiřazením sousedních hráčů do dočasného seznamu a následné nastavování atributů a smazání hráče z dočasného seznamu, o němž byly údaje zjištěny. Počet prováděných operací je však přibližně stejný, pro n políček a m sousedů se provádí $n \cdot m$ výchozích nastavení history a current_action.

Strategie

Hráčům je nastavena akce procedurou set_action na základě nastavené strategie. Oslovení jsou všichni hráči a následně jejich sousedé, kteří proceduru set_action pro hráče spouští. Údaje sousedů jsou nutné pro aktualizaci current_action. Prvními implementovanými strategiemi v proceduře set_action jsou cooperate a defect, u kterých je pro hráče v každém kole zvolena spolupráce příkazy, které vyobrazuje následující ukázka kódu.

```
1  if [strategy] of myself = "cooperate"
2    [ ask myself [ set current_action replace-item (1 + position [pwho] of
3      myself current_action) current_action false ] ]
4  if [strategy] of myself = "defect"
5    [ ask myself [ set current_action replace-item (1 + position [pwho] of
6      myself current_action) current_action true ] ]
```

Příkazy provádí soused pro hráče, který jej volal. Na atributy hráče je tedy nutné se odkazovat deklarací „[atribut] of myself“, příkaz myself se odkazuje na nadřazeného agenta, který volal toho, kdo hodnoty ověřuje. Návratovou textovou hodnotou je strategie, která je

porovnává s deseti textovými hodnotami všech strategií a jedna strategie provádí vždy jen jí příslušný blok příkazů.

Ve strategii *cooperate soused* dotazuje hráče, aby si aktualizoval *current_action* nahrazením prvku na pozici $1 + \text{pozice } pwho \text{ souseda ve vlastním listu } current_action$ hodnotou *false* (spolupráce) či *true* (zrada). Jedna je přičítáno pro navrácení příslušné hodnoty akce za unikátním číselným označením *pwho souseda*. Stejná deklarace aktualizace *current_action* je implementována u všech strategií, které obsahují pouze jiné podmínky vedoucí k nastavení spolupráce či zrady s dotazovaným sousedem.

Strategie *random* předpokládá přibližně ve stejném počtu případů spolupráci či zradu pomocí ověření $random\ 2 = 0$. Splnění podmínky znamená nastavení spolupráce, jinak zrady. Ověření kvality generátoru diskrétních pseudonáhodných čísel z rovnoměrného rozdělení viz vyhodnocení testů generátoru *random* v kapitole 3.3 a příložené CD, příloha 2), soubor „Random discrete NetLogo generator test.xlsx“.

Strategie *even_cooperate_odd_defect* v *set_action* ověřuje podmínku sudosti herního kola (*ticks*) implementací „*ifelse (ticks mod 2 = 0)*“. Pokud je celočíselný zbytek po dělení kola dvěma nulový, jedná se o sudé kolo a je nastavena spolupráce, jinak zrada. Další implementovanou strategií je *cooperate_twice_defect_twice*, která v prvních dvou kolech předpokládá spolupráci.

V ostatních kolech je přepínána spolupráce na zradu v sudém kole při současném splnění spolupráce v předchozím kole, která je ověřována deklarací „*not (item (2 + position pwho [history] of myself) [history] of myself)*“. Návratovou hodnotou části kódu za „*not*“ je *false/true*. Zjišťován je poslední prvek hráčovy historie tahů se sousedem, který se nachází na pozici $2 + \text{pozice } pwho \text{ souseda v nadřazeném hráčově listu } history$. Aby byla spolupráce přepnuta na zradu, musí být navrácenou hodnotou *false* (spolupráce v předchozím kole).

Blok příkazů uvnitř podmínky je dosažen, neboť „*not false*“ je stejné jako *true*. K pokračování zrady z předchozích kol nedochází, neboť pro zradu je návratovou hodnotou *true* a „*not true*“ je nesplněnou částí podmínky AND, tudíž není dosaženo bloku příkazů pro nastavení zrady v současném kole. Analogicky u druhé podmínky sudosti herního kola a návratové hodnoty *true* (zrada v předchozím kole) je akce v současném kole nastavena na „*false*“ (spolupráce) a u návratové hodnoty *false* není splněna část podmínky pro pokračování spolupráce ve více než dvou kolech. U evolučního modelu může strategie při

rozšíření na hráče, který hrál původně jinou strategii, hrát pro různé sousedy různé akce i ve stejném kole a až podle souseda přepínat spolupráci a zradu po dvou kolech.

```
1 ask myself [ show item (2 + position [pwho] of myself history) history ]
2 =
3 show item (2 + position pwho [history] of myself) [history] of myself
```

Jak lze vidět na předcházející ukázce kódu, obě deklarace příkazu prováděného z pohledu souseda zobrazují stejnou hodnotu, neboť „ask myself“ se odkáže zpět na hráče, který souseda volal, a je zjištěn prvek na pozici $2 + \text{pozice } pwho \text{ nadřazeného souseda v hráčově historii akcí se sousedy}$. Totožný prvek je z pohledu souseda navrácen deklarací $2 + \text{pozice sousedova pwho v historii akcí nadřazeného hráče se sousedy}$.

Strategie `tit_for_tat` nastavuje v prvním kole spolupráci se všemi sousedy. V následujících kolech ověřuje poslední akci souseda v jeho historii s identifikátorem `pwho` nadřazeného hráče. Oproti strategii `cooperate_twice_defect_twice` zjišťuje `tit_for_tat` sousedovu akci s hráčem a ne hráčovu akci se sousedem. Jestliže v předchozím kole soused hráče zradil, tak hráč souseda v současném kole také zradí, jinak se sousedem spolupracuje.

Strategie `tit_for_two_tats` se od strategie `tit_for_tat` liší v tom, že místo prvního kola nastaví spolupráci v prvních dvou kolech a poté ověřuje poslední dva tahy souseda s hráčem. Pokud soused hráče zradil ve dvou předchozích kolech současně, hráč jej v aktuálním kole také zradí, jinak spolupracuje. Akce posledních dvou kol souseda s hráčem jsou uloženy na pozici $1 + \text{pwho nadřazeného hráče v sousedově historii}$ (akce o dvě kola zpět) a $2 + \text{pwho nadřazeného hráče v sousedově historii}$ (akce o jedno kolo zpět).

Unforgiving v prvním kole nastavuje spolupráci a v následujících kolech zradu tehdy, jestliže je navráceno `true` (zrada) z posledního prvku historie souseda s hráčem nebo z posledního prvku hráčovy historie se sousedem. Jinak hráč se sousedem v aktuálním kole spolupracuje. Strategie `tft_&_random_defect_33%` vychází z implementace `tit_for_tat`, ke které přidává ověření „random 3 = 1“, při jehož splnění v přibližně třetině kol hry se nastaví zrada nezávisle na předchozích akcích soupeře, jinak se hraje `tit_for_tat`.

```
1 if [strategy] of myself = "tft_&_random_forgive_defect_33%"
2   [ ifelse ticks = 0
3     [ ask myself [ set current_action replace-item (1 + position
4       [pwho]
5       of myself current_action) current_action false ] ]
6     [ ifelse item (2 + position [pwho] of myself history) history
7       [ ifelse random 3 = 1
```

```

8      [ ask myself [ set current_action replace-item (1 +
9          position [pwho] of myself current_action) current_action
10         false ] ]
11     [ ask myself [ set current_action replace-item (1 + position
12         [pwho] of myself current_action) current_action true ] ]
13     ]
14     [ ask myself [ set current_action replace-item (1 + position
15         [pwho] of myself current_action) current_action false ]
16     ] ] ]

```

Poslední implementovanou strategií je `tft_&_random_forgive_defect_33%`, jejíž kód zobrazuje předchozí ukázka. V prvním kole hráč spolupracuje se všemi sousedy. V ostatních kolech je stejně jako u strategie `tit_for_tat` zjišťována poslední akce hráče v sousedově historii za použití identifikátoru `pwho` hráče, který souseda volá. U návratové hodnoty `true` (zrada) je následně porovnáváno `random 3 = 1`, což je přibližně ve třetině případů splněno a sousedovi je odpuštěna jeho zrada.

Hráč si ve vlastním seznamu současných akcí `current_action` nastavuje akci na pozici `1 + sousedivo pwho` na `false` (spolupráci). Ve zbylých přibližně dvou třetinách případů zradu sousedovi neodpouští a souseda také zradí. Pokud soused hráče v předchozím kole nezradil, není blok příkazů na řádcích 7–13 proveden vůbec a hráči je nastaveno `false` (spolupráce) se sousedem v aktuálním kole, viz řádek 14–16 předcházející ukázky kódu.

Aktualizační procedury

Model obsahuje celkem šest aktualizačních procedur, z nichž hlavní procedurou je „go“, ze které jsou ostatní procedury spouštěny. Procedura `set_action` implementuje nastavení akce v jednom kole hry dle strategie hráče. Nastavení akce bylo popsáno v předcházející části Strategie. Procedury vyhodnocování jsou `evaluate_neighbors_strategies` a `evaluate_strategies` a změny strategií jsou implementovány v `evolve`. Aktualizaci grafu provádí `do_plots`.

Po nastavení akce aktuálního kola (viz Strategie) probíhá vyhodnocení aktuálního kola a aktualizace historie. Vyhodnocení probíhá až po nastavení akce všem hráčům, tedy v oddělené deklaraci volání hráčů. Na počátku jim je vynulováno skóre aktuálního kola. Poté se zeptají svých sousedů. Na jejich úrovni jsou porovnány všechny možné situace dle výplatní matice. Implementace viz následující ukázka kódu.

```

1  ifelse item (1 + position pwho [current_action] of myself)
2  ([current_action] of myself)
3  [ ifelse item (1 + position [pwho] of myself current_action)

```

```

4     current_action
5         [ ask myself [ set score score + 3 ] ]
6         [ ask myself [ set score score + 7 ] ]
7     ]
8     [ ifelse item (1 + position [pwho] of myself current_action)
9         current_action
10        [ ask myself [ set score score + 0 ] ]
11        [ ask myself [ set score score + 6 ] ]
12    ]

```

Na úrovni sousedů je nejdříve zjišťována akce na pozici identifikátoru pwho souseda, zvětšené o jedna, v seznamu současných akcí hráče se sousedem pwho current_action. Návratová hodnota true představuje situaci, kdy hráč souseda v aktuálním kole zradil a je proveden blok příkazů na 3–7. řádku ukázky kódu. False reprezentuje hráčovu spolupráci, první podmínka s false hodnotou není splněna a je spuštěn blok příkazů v else části podmínky, viz 8–12. řádek. Na 3–7. řádku je zjišťována souseдова akce v jeho seznamu current_action s pwho nadřazeného dotazujícího hráče.

Jestliže se hráč i jeho souseď navzájem zradili, je hráči zvětšeno skóre o tři, viz pátý řádek. Pro aktualizaci skóre je z úrovně dotázaného souseďa hráč volán zpět příkazem „ask myself“. Šestý řádek předchozí ukázky kódu reprezentuje zradu spolupracujícího souseďa, hráči je navýšeno skóre o nejvyšší možný počet bodů, a to sedm. Pokud naopak hráč se souseďem spolupracuje, ale souseď jej zradí, hráč nezískává nic (desátý řádek). Vzájemná spolupráce je pro hráče odměněna šesti body (jedenáctý řádek). Souseďovi je skóre aktualizováno až v momentě, kdy je osloven na nadřazené úrovni hráče a poté oslovuje své souseďy. Prvky seznamů současných akcí jsou pro hráče a jeho souseďa porovnávány dvakrát, ale skóre je aktualizováno pouze hráči dotázanému na první úrovni.

Ve stejné deklaraci dotazování souseďa a vyhodnocování aktuálního kola je implementována také aktualizace historie hráče se souseďem dotázaným na druhé úrovni. Hráč je z úrovně souseďa zpět dotázán, aby znal souseďovo pwho a mohl si aktualizovat příslušnou akci se souseďem. Historie obsahuje poslední dva tahy a aktualizace historie má dvě fáze. Nejdříve je nahrazen předposlední (dvě kola zpět) tah hráče se souseďem posledním tahem a poté je poslední tah nahrazen aktuálním.

```

1 set history replace-item (position [pwho] of myself history + 1) history
2 item ( (position [pwho] of myself history + 2) ) history
3
4 set history replace-item (position [pwho] of myself history + 2) history
5 (item (1 + position [pwho] of myself current_action) current_action)

```

Předchozí ukázka kódu zobrazuje obě fáze aktualizace historie. Předposlední hodnota historie hráče je nahrazena příkazem `replace-item`. `Replace-item` vyžaduje tři parametry, index prvku, který se má nahradit, název seznamu s nahrazovaným prvkem a hodnotu, jakou se má nahradit prvek na indexu. Index je zjištěn jako pozice sousedova `pwho` v hráčově historii. Zvětšením o jedna je zjištěn index uložené akce se sousedem o dvě kola zpět.

Druhý parametr je hráčův seznam history (historie) a třetí parametr je akce o jedno kolo zpět, jejíž hodnota je získána jako položka na pozici `sousedova pwho` v seznamu history, pozice zvětšená o dva. Ve druhé fázi je nahrazována poslední akce historie, nacházející se na pozici se sousedovým `pwho` v history, pozice zvětšená o dva. Ta je nahrazena prvkem na pozici se sousedovým `pwho` v seznamu aktuálních akcí hráče se sousedem `pwho`.

Po vyhodnocení aktuálního kola a aktualizaci historie hráč již se svého pohledu nastavuje popisek na součet skóre ze všech dílčích her se sousedy (`score`). Jestliže není zapnuto nastavování popisů v uživatelském rozhraní (`plabel?`), je popisek hráče nastaven na prázdný textový řetězec. Po nastavení popisů je aktualizováno celkové skóre políček (`score_sum`) sečtením současné hodnoty a skóre v aktuálním kole (`score`).

V samostatném oslovení všech hráčů je poté spouštěna vyhodnocovací procedura `sousedních strategií evaluate_neighbors_strategies` s jedním parametrem, do kterého jsou ve `while` cyklu postupně přiřazovány názvy strategií ze `strategies`. Před cyklem `while` je definován dočasný čítač `i`, jehož hodnota je na konci cyklu navyšována o jedna. Dokud je `i` menší než délka seznamu všech strategií, je blok příkazů `while` cyklu opakovaně spouštěn. Proveden je tolikrát, jaký je počet strategií. Uvnitř bloku příkazů je do nově vytvořené dočasné proměnné `neighbor_strategy_to_evaluate` přiřazen prvek na pozici `i` v seznamu strategií. Potom je spuštěna procedura `evaluate_neighbors_strategies`, které je předán parametr `neighbor_strategy_to_evaluate`.

V proceduře `evaluate_neighbors_strategies` jsou zprvu spočítáni sousedé s aktuálně vyhodnocovanou strategií. Pokud je počet sousedů nenulový, je do dočasné proměnné `pos_of_neighbor_strategy_score` za účelem zpřehlednění kódu uložena pozice vyhodnocované strategie v seznamu `neighbors_strategies_score`, zvětšená o jedna. Následně je aktualizováno skóre sousední strategie v hráčově seznamu `neighbors_strategies_score` nahrazením prvku na pozici `pos_of_neighbor_strategy_score` součtem bodů sousedů hrajících aktuálně vyhodnocovanou strategií `neighbor_strategy_to_evaluate` dělený jejich počtem. Jestliže je počet sousedů s vyhodnocovanou strategií nulový, je průměrné skóre

sousední strategie v seznamu `neighbors_strategies_score` vynulováno. Skóre sc_g sousední strategie g v jednom kole je určeno dle vztahu

$$sc_g = \frac{\sum_{k=1}^{c_g} sc_k}{c_g}$$

kde c_g je populace sousedů hrajících strategii g a sc_k je skóre suseda k se strategií g .

Procedura `evaluate_strategies` je obdobně jako `evaluate_neighbors_strategies` spouštěna v cyklu `while` tolikrát, jaký je počet strategií. Rozdílem je, že je globální a nespouští se při dotazování všech hráčů. Místo parametru `neighbor_strategy_to_evaluate` je předán `strategy_to_evaluate` se stejnými textovými hodnotami strategií jako u procedury `evaluate_neighbors_strategies`. Po spuštění `evaluate_strategies` jsou spočítána všechna políčka s vyhodnocovanou strategií `strategy_to_evaluate`. Pokud existuje alespoň jeden takový hráč, je do dočasné proměnné `pos_of_strategy_score` uložena pozice hodnocené strategie v globálním seznamu `strategies_score_sum`, zvětšená o jedna. V dalším kroku je aktualizován prvek s indexem `pos_of_strategy_score` ve `strategies_score_sum` přičtením sumy skóre políček s hodnocenou strategií dělené počtem těchto políček. Skóre sc_c strategie c je určeno dle vztahu

$$sc_c = \sum_{t=0}^n \frac{\sum_{k=1}^{c_t} sc_{tk}}{c_t}$$

kde n je počet kol hry, c_t je populace strategie c v kole t a sc_{tk} je skóre jedince k hrajícího strategii c v kole t .

Jestliže je počet hráčů vyhodnocované strategie `strategy_to_evaluate` nulový, populace této strategie již vyhynula a nemůže znovu vzniknout. Rozšiřují se pouze existující strategie. Při vyhynutí strategie se zaznamená herní kolo do globálního seznamu `strategies_extinction_time`. Pro zpřehlednění kódu je opět deklarována dočasná proměnná, do které je přiřazena pozice hodnocené strategie v seznamu `strategies_extinction_time`, zvětšená o jedna. Aby k zaznamenání času vyhynutí došlo, musí být prvek na pozici `pos_of_extinction_time` v seznamu `strategies_extinction_time` nulový, což znamená, že doposud nebyl zaznamenán čas vyhynutí strategie.

Pakliže je prvek nulový, je seznam `strategies_extinction_time` přenastaven nahrazením prvku na pozici `pos_of_extinction_time` aktuálním herním kolem `ticks`. Ve výstupní textové oblasti jsou vypsaný název hodnocené strategie, čas vyhynutí a skóre hodnocené strategie

ze seznamu `strategies_score_sum` zaokrouhlené na dvě desetinná místa. Skóre strategie se dále neaktualizuje a průměrné skóre na jedno kolo lze zjistit vydělením skóre strategie ze `strategies_score_sum` a dobou existence strategie ze `strategies_extinction_time`.

Procedura změny strategií `evolve` na začátku `go` je spouštěna až od druhého kola a vyžaduje informace o vyhodnocení prvního kola. Aby byla evoluce strategií spuštěna, musí být také zapnutý přepínač `evolution?` (zapnutí vrací hodnotu `true`). Evoluce strategií probíhá na začátku herního kola, aby hráči měli na konci kola tu strategii, kterou v průběhu kola hráli. Evoluce na počátku druhého kola proběhne na základě vyhodnocení prvního kola atd.

Uvnitř podmínky ověřující zapnutý přepínač `evolution?` a nenulovost herního kola (`ticks > 0`) je navyšován čítač `time_since_evolution` o jedna. Poté je z pohledu všech hráčů spuštěna procedura `evolve` a pokud nedošlo po sto kolech k žádné změně strategií a čas stabilizace nebyl doposud nastaven, je také nastaven čas stabilizace evolučního procesu. Stabilizace je vyhodnocována zpětně a čas je nastaven na aktuální počet kol hry `ticks` snížený o 100 („set stabilization_time ticks - 100“). Předpokladem je, že sto kol je dostatečně dlouhá doba, po kterou neproběhla žádná změna, aby byl nastaven čas stabilizace. Ke stabilizaci nemusí dojít vůbec.

```
1  to evolve
2    let best_neighborhood_strategy max neighbors_strategies_score
3    if best_neighborhood_strategy > score
4      [ let nsc neighbors_strategies_score
5        ifelse length filter [? = best_neighborhood_strategy] nsc > 1
6          [ let best_strategies []
7            while [length nsc > 0]
8              [ if item 1 nsc = best_neighborhood_strategy
9                [ set best_strategies lput (item 0 nsc) best_strategies ]
10             set nsc remove-item 0 nsc
11             set nsc remove-item 0 nsc
12           ]
13           let best_strategy one-of best_strategies
14           if best_strategy != strategy
15             [ set strategy best_strategy
16               set pcolor item ((position strategy colors) + 1) colors
17               set time_since_evolution 0
18             ]
19         ]
20     [ let best_strategy item ((position best_neighborhood_strategy
21       neighbors_strategies_score) - 1) neighbors_strategies_score
22       if best_strategy != strategy
```

```

23         [ set strategy best_strategy
24           set pcolor item ((position strategy colors) + 1) colors
25           set time_since_evolution 0
26         ]
27     ] ]
28 end

```

Proceduru `evolve` vyobrazuje předchozí ukázka kódu. Po spuštění `evolve` hráč zjišťuje maximální hodnotu skóre sousedních strategií z `neighbors_strategies_score` a ukládá ji do dočasné proměnné `best_neighborhood_strategy`. Aby mohlo dojít ke změně strategií, musí být skóre nejlepší sousední strategie větší než skóre hráče (3. řádek). Jestliže je podmínka splněna, je zjištěno, zda existuje více sousedních strategií se stejným maximálním skóre. Z toho důvodu je nejdříve duplikován seznam `neighbors_strategies_score` do dočasné proměnné `nsc` a poté je ověřena délka seznamu, ze kterého byly filtrováním odstraněny hodnoty, které nemají maximální hodnotu skóre sousedních strategií (viz řádek 5). Tím je zjištěn počet sousedních strategií, které mají stejnou maximální hodnotu skóre v jednom kole.

Jestliže je tento počet větší než jedna, je náhodně vybrána jedna z více nejlepších sousedních strategií. Pro tento postup je zprvu vytvořen dočasný seznam `best_strategies`. Následuje opakované spuštění `while` cyklu, ve kterém je porovnáván druhý prvek (skóre sousední strategie) seznamu `nsc` s hodnotou `best_neighborhood_strategy`. Při splnění rovnosti je do `best_strategies` vložen první prvek (index 0) z `nsc`, na kterém se nachází název příslušné strategie s maximálním skóre na pozici 1 v seznamu `nsc` (řádek 8–9). Poté je `nsc` zkrácen o dva první prvky dvounásobnou deklarací „`set nsc remove-item 0 nsc`“. `While` cyklus je opakovaně spuštěn, dokud je redukován seznam `nsc` nenulový. Je prozkoumáno skóre všech sousedních strategií a do seznamu `best_strategies` jsou vloženy názvy nejlepších strategií.

Z nejlepších strategií se náhodně vybere jedna, ta je uložena do lokální proměnné `best_strategy` (řádek 13). Mezi nejlepšími sousedními strategiemi může být i strategie, kterou hráč aktuálně má. Do seznamu nejlepších strategií by bylo možné přidávat pouze ty, které jsou jiné než současná strategie hráče, ale potom by neměly všechny strategie stejnou šanci. Stejná strategie by měla mít stejnou šanci na své zachování (rozšíření na sousedního hráče, který již tuto strategii má) jako je šance rozšíření jiné strategie, než kterou má sám hráč.

Na 14. řádce je zjištěno, zda je zvolená nejlepší strategie `best_strategy` jiná než současná hráčova strategie a pokud ano, je hráči nastavena s následnou aktualizací barvy a vynulování čítače `time_since_evolution`, neb právě proběhl evoluční proces strategií. Jestliže je `best_strategy` stejná jako současná hráčova strategie, pak sice nedojde ke změně struktury strategií, ale strategie obstála v evolučním procesu.

20–28. řádek reprezentuje blok příkazů, který je proveden v případě, že je nejlepší sousední strategie pouze jedna. Do dočasně vytvořené `best_strategy` se uloží prvek na pozici maximálního skóre sousedních strategií `best_neighborhood_strategy` v seznamu `neighbors_strategies_score`, snížené o jedna. Zde se nachází název nejlepší strategie. Jestliže je nejlepší strategie jiná než současná hráčova strategie, nastaví se jeho strategie na `best_strategy`, jinak si zachovává svoji. Po změně atributu `strategy` je aktualizována barva hráče a čítač `time_since_evolution` je vynulován.

Aktualizační procedurou `do_plots` je od druhého kola aktualizován graf „Strategies counts“. Vykreslováno je deset datových řad, pro každou strategii jedna. Deklarací `set-current-plot-pen` je nastavena datová řada, u které je v následujícím řádku kódu implementováno vykreslení počtu hráčů s příslušnou strategií. Na konci kódu `do_plots` je porovnáno číslo herního kola. Jestliže je větší než 100, u grafu je změněn rozsah osy `x`, reprezentující průběh hry, na posledních sto kol příkazem „`set-plot-x-range (ticks - 100) (ticks)`“.

3.3 Experimenty

Experimenty byly provedeny za pomoci nástroje `BehaviorSpace`, který je součástí `NetLogo`. V simulaci bylo využito vícejádrového zpracování jednotlivých her a pokusy byly ve výstupním souboru seřazeny dle jedinečného identifikátoru hry. Výsledky experimentů byly zpracovány v `Excelu`. `Excel` má omezení v počtu řádků na listu, ale to nebylo pro účely experimentů limitující. Pro analýzu výsledků simulací byly využity základní statistické metody.

Strategie implementované v modelech využívají generátor diskretních pseudonáhodných čísel `random`. Hodnoty mezi nulou a zadaným číslem jakožto argumentem této funkce by měly mít přibližně rovnoměrné rozdělení, tedy přibližně stejné četnosti. Kvalita generátoru `random` byla ověřena na vzorku 100 000 generovaných celých čísel mezi 0 a 100 (příkaz „`random 101`“) Kolmogorov-Smirnovovým testem a Chí-kvadrát testem shod

empirického a teoretického modelu, viz příložené CD, příloha 2), soubor „Random discrete NetLogo generator test.xlsx“.

Ani na základě testů ani na základě grafické interpretace (četnosti jednotlivých hodnot se pohybují okolo 1 000) nebyly odhaleny statisticky významné rozdíly v teoretických a pozorovaných četnostech čísel od 0 do 100. Pro Kolmogorov-Smirnovův test byly počítány relativní a pro Chí-kvadrát test absolutní četnosti.

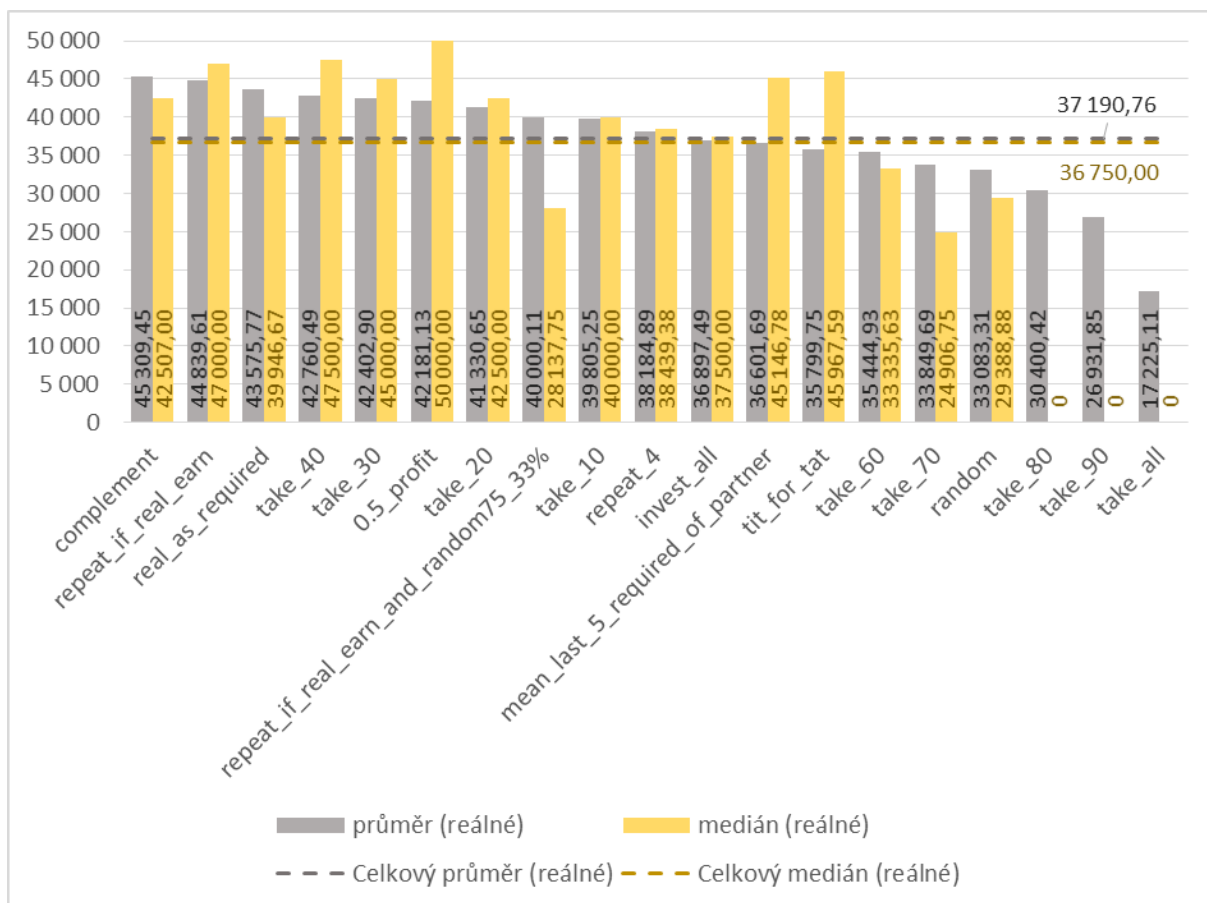
3.3.1 Sosisova hra v kibucech

Cílem experimentů modelu Sosisovy hry je porovnat úspěšnost jednotlivých strategií ve hře proti všem ostatním. Strategie popsané v kapitolách 3.1.1 a 3.2.1, byly navrženy a implementovány pro účely experimentů. Strategie mezi sebou nejsou porovnávány na základě výsledků z jednoho kola hry, ale na základě celých her, přičemž jedna hra má 1 000 kol. Strategie mají v různých kolech různé dílčí chování a v průběhu hry mohou procházet vývojem na základě tahů protihráče. Globální chování strategie se projeví až ve více kolech hry.

Strategie byly hodnoceny na základě výsledků hráče 0, kterému byly postupně přiřazovány jednotlivé strategie, které byly změněny až po odehrání strategie se všemi 19 strategiemi přiřazenými hráči 1. Strategie hrála i se sebou samou. Jak bylo popsáno v kapitolách 3.1.1 a 3.2.1, některé strategie obsahují prvky náhodnosti. Aby byl vliv náhodnosti na výsledky experimentů omezen, byla každá kombinace hry strategií proti ostatním opakována stokrát. Celkový počet her byl tedy $19 * 19 * 100 = 36\ 100$.

Klíčové indikátory pro hodnocení strategií jsou průměrné a mediánové reálné zisky her a průměrný a mediánový absolutní přírůstek reálných zisků oproti požadovaným. Souhrnné hodnoty jednotlivých strategií byly vypočítány z 1 900 dílčích výsledků her. Souhrnné výsledky experimentů modelu Sosisovy hry v kibucech obsahují grafy 1 a 2. Všechny výsledky obsahuje příložený soubor „Sosis game experiment.xlsx“, viz příloha 2).

Graf 1 na vedlejší ose indikuje průměrné (mediánové) reálné zisky za všechny hry jednotlivých strategií s dalšími 19, včetně jich samých. Průměr i medián byly pro jednotlivé strategie vypočítány na základě 1 900 dílčích výsledků her. Strategie jsou vyobrazeny na hlavní ose, název přesně odpovídá implementovanému pojmenování v modelu. Strategie jsou seřazeny sestupně dle průměrných a poté dle mediánových reálných zisků her.



Graf 1: Průměrné a mediánové reálné zisky Sosisovy hry

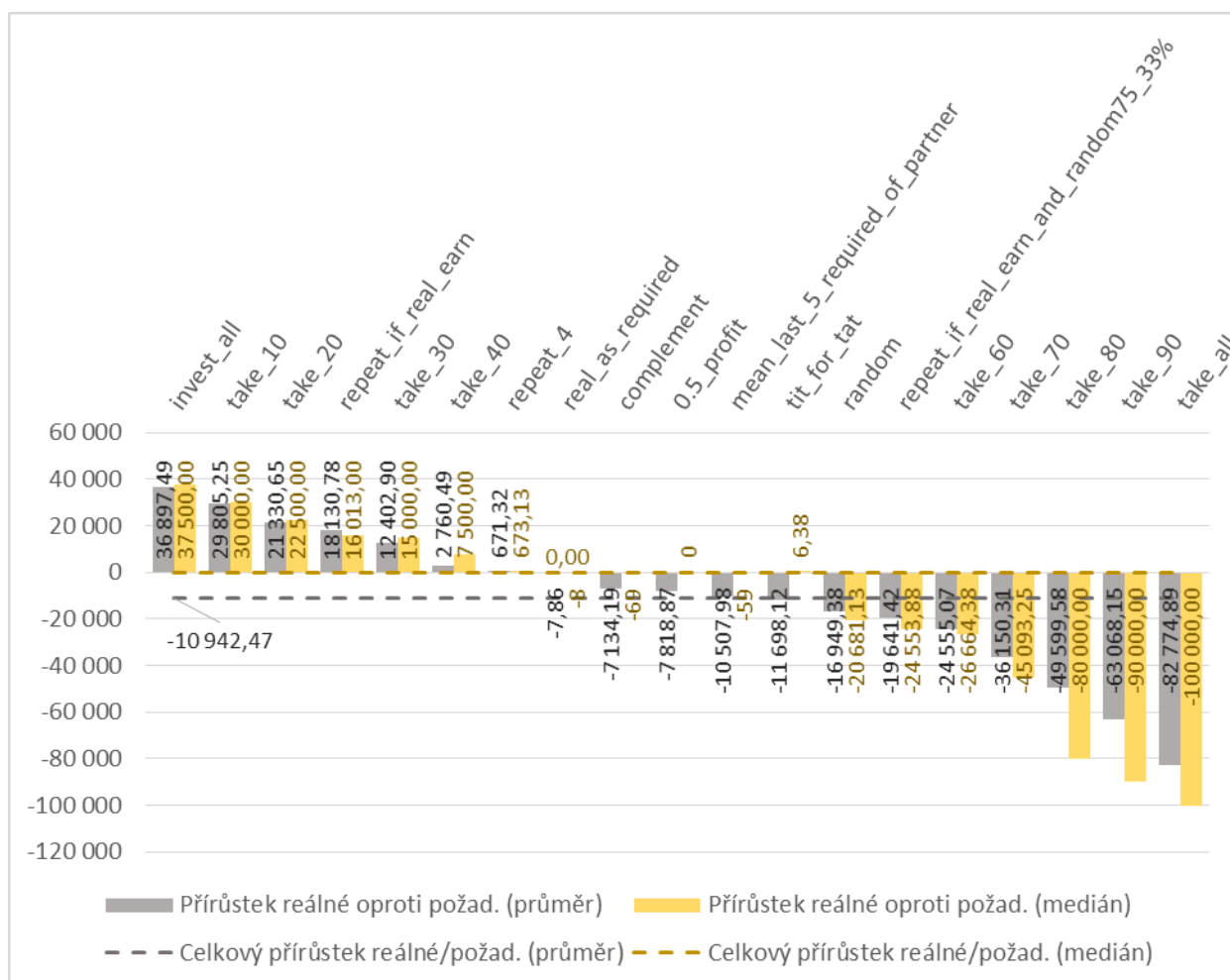
Zdroj: vlastní zpracování

Jak lze vyčíst z grafu 1, nejúspěšnější strategií podle průměrných reálných zisků her je complement následovaná strategií repeat_if_real_earn. Hráč požadující doplněk do 100 k předchozí částce požadované soupeřem (complement) vlastnil na konci všech her průměrně 45 309,45. Na třetím místě je real_as_required. Nejméně úspěšné jsou strategie take_80, take_90 a take_all. Strategie repeat_if_real_earn je také třetí nejlepší strategií podle mediánových reálných zisků. V polovině her hráč obdržel vyšší reálné zisky než 47 000. Mediánově nejúspěšnější je 0.5_profit a druhou nejlepší strategií je take_40. Hráč požadující polovinu rozdělovaného zisku v každém kole reálně vlastnil v polovině her částku 50 000. V polovině her poté strategie take_80, take_90 a take_all skončily s nulovým skóre.

Mediánové hodnoty jsou oproti průměrným výrazněji rozdílné například u strategie repeat_if_real_earn_and_random75_33%. Mediánové reálné zisky jsou nižší než průměrné, neboť existuje poměrně hodně nadprůměrných hodnot reálných výdělků při hrách se strategiemi invest_all, take_10 a repeat_if_real_earn a třemi dalšími. Šest strategií z 19 nepředstavuje ani třetinu celkového počtu her. Analogicky u strategie take_70.

Vyšší mediánové částky oproti průměrným se vyskytují u strategií 0.5_profit, mean_last_5_required_of_partner a tit_for_tat. Způsobeny jsou vyšším počtem podprůměrných hodnot než nadprůměrných, které jsou zároveň i více rozdílné od průměru, u her se strategiemi take_60, take_70, take_80, take_90 a take_all. Posouzeno na základě všech výsledků přiloženého souboru „Sosis game experiment.xlsx“, viz příloha 2).

Nad celkovým průměrem je deset strategií a pod celkovým průměrem devět strategií. Celkový průměr je mírně vyšší než celkový medián, což je způsobeno vysokými reálnými zisky her strategií proti invest_all a také tím, že minimální hodnota skóre hry je 0, zatímco maximální 100 000, průměrné skóre hry je 37 190,76. Mediánové reálné zisky dvanácti strategií jsou vyšší než celkové mediánové reálné zisky a u sedmi strategií nižší. Pravděpodobně to je způsobeno tím, že 40,68 % nulových zisků všech her se nachází u strategií take_80, take_90 a take_all, jejichž mediánové reálné zisky jsou nulové. Posouzeno na základě všech výsledků přiloženého souboru „Sosis game experiment.xlsx“, viz příloha 2).



Graf 2: Absolutní přírůstky reálných zisků Sosisovy hry

Zdroj: vlastní zpracování

Hlavní osa grafu 2 reprezentuje implementovaný název strategií. Vedlejší osa vyobrazuje průměrné (mediánové) absolutní přírůstky reálných zisků oproti požadovaným. Strategie jsou seřazeny sestupně dle průměrných absolutních přírůstků a na druhé úrovni řazení dle mediánových absolutních přírůstků.

Jak lze vidět na grafu 2, průměrně nejvíce reálných zisků oproti požadovaným získaly strategie invest_all, take_10 a take_20. Strategie invest_all získala průměrně o 36 897,49 více, než požadovala. V polovině případů získala alespoň o 37 500 více. Mediánové pořadí prvních třech strategií podle průměru je stejné. Čtvrtou nejúspěšnější podle průměrných i mediánových přírůstků je repeat_if_real_earn. Tato strategie je rovněž druhou nejlepší podle průměrných a třetí nejlepší podle mediánových reálných zisků. Nejméně úspěšnými strategiemi z hlediska absolutních přírůstků reálných zisků jsou take_80, take_90 a take_all. Strategie take_all získala průměrně o 82 774,89 méně, než požadovala, přičemž minimálně v polovině případů se jedná o 100 000 méně.

Pokud by byly strategie posuzovány z hlediska procentuálních přírůstků, invest_all by nebyla zařazena, neboť požaduje vždy nulovou částku a při výpočtu procentuálních přírůstků by docházelo k dělení nulou. Procentuální přírůstky jsou navíc v poměru k požadovaným ziskům značně zkresleny extrémními hodnotami her adaptabilních strategií se strategií invest_all. Adaptabilní strategie v prvním kole požadují náhodnou částku a v následujících kolech opakují nulovou požadovanou částku po strategii invest_all. Požadovaná částka za celou hru je tak v poměru k reálným ziskům značně nižší. Posouzeno na základě všech výsledků příloženého souboru „Sosis game experiment.xlsx“, viz příloha 2).

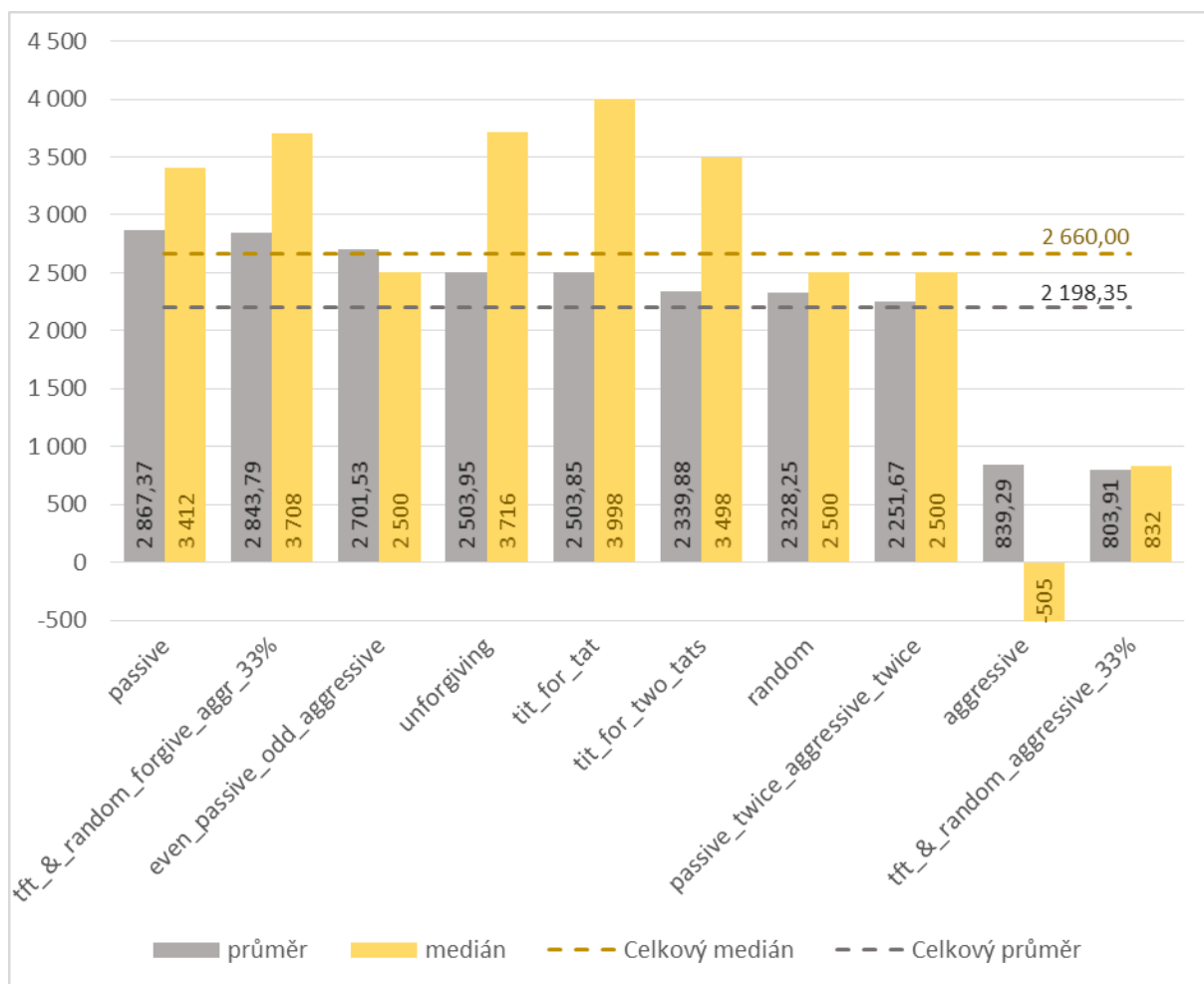
3.3.2 Boj o zdroje

Cílem experimentů modelu boje o zdroje je porovnat úspěšnost strategií ve hře proti všem ostatním. Strategie uvedené v kapitolách 3.1.2 a 3.2.2, byly navrženy a implementovány pro účely experimentů. Strategie mezi sebou nejsou porovnávány na základě výsledků z jednoho kola hry, ale na základě celých her, přičemž jedna hra má 1 000 kol. Strategie mají v různých kolech různé dílčí chování a v průběhu hry mohou procházet vývojem na základě tahů protihráče. Globální chování strategie se projeví až ve více kolech.

Strategie byly hodnoceny na základě výsledků hráče 0, kterému byly postupně přiřazovány jednotlivé strategie, které byly změněny až po odehrání strategie se všemi 10 strategiemi přiřazenými hráči 1. Strategie hrála i se sebou samou. Jak bylo uvedeno v kapitolách

3.1.2 a 3.2.2, některé strategie obsahují prvky náhodnosti. Aby byl vliv náhodnosti na výsledky experimentů omezen, byla každá kombinace hry strategií proti ostatním opakována stokrát. Celkový počet her byl tedy $10 * 10 * 100 = 10\ 000$.

Klíčové indikátory pro hodnocení strategií jsou průměrné a mediánové skóre her. Souhrnné výsledky experimentů modelu boje o zdroje obsahuje graf 3. Všechny výsledky obsahuje příložený soubor „Hawk Dove game modified experiment.xlsx“, viz příloha 2).



Graf 3: Průměrné a mediánové skóre modelu boje o zdroje

Zdroj: vlastní zpracování

Graf 3 na hlavní ose zobrazuje implementované názvy jednotlivých strategií. Název strategie tft_&_random_forgive_aggressivity_33% byl zkrácen na tft_&_random_forgive_aggr_33%, aby se vešel do oblasti grafu importovaného z Excelu. Vedlejší osa představuje průměrné a mediánové skóre za všechny hry jednotlivých strategií s dalšími 10, včetně jich samých. Průměr i medián byly pro jednotlivé strategie vypočítány na základě 1 000 dílčích výsledků her. Strategie jsou seřazeny sestupně dle průměrného a poté dle mediánového skóre.

Jak lze vidět na grafu 3, nejvíce úspěšnou strategií podle průměrného skóre je pasivita (passive) s průměrným skóre her 2 867,37. Druhou nejúspěšnější strategií je tft_&_random_forgive_aggressivity_33 a třetí even_passive_odd_aggressive. Rozdíl průměrných skóre prvního a druhého místa není příliš zásadní. Na prostřední pozici se nachází velmi úspěšná strategie vězňova dilematu (tit_for_tat) modifikovaná pro účely modelu boje o zdroje. Nejméně úspěšnými strategiemi jsou passive_twice_aggressive_twice, agresivita (aggressive) a tft_&_random_aggressive_33%. Průměrné skóre první zmíněné je ještě poměrně vysoké, avšak za ní následuje prudký propad na skóre 829,29 a poslední strategie s průměrným skóre 803,91.

Podle mediánového skóre je však nejúspěšnější strategie tit_for_tat, která měla v polovině her skóre alespoň 3 998. Rozdíl mediánové a průměrné hodnoty skóre strategie je pravděpodobně způsoben vysoce podprůměrnými výsledky hry se strategiemi aggressive a tft_&_random_aggressive_33%. Jedná se o 20 % her a tak je průměrné skóre ovlivněno podstatně větší mírou než medián. Posouzení na základě všech výsledků přiloženého souboru „Hawk Dove game modified experiment.xlsx“, viz příloha 2).

Znatelně vyšší medián než průměr se vyskytuje také u strategií unforgiving, tit_for_two_tats, tft_&_random_forgive_aggressivity_33%, passive a aggressive. U aggressive je medián dokonce záporný. Způsob ovlivnění průměru strategií unforgiving a tit_for_two_tats je stejný jako u strategie tit_for_tat. Strategie tft_&_random_forgive_aggressivity_33% a passive však mají vyšší medián než průměr kvůli značně podprůměrným hodnotám skóre her pouze proti aggressive. Jedná se o 10 % her a medián není natolik rozdílný jako například u tit_for_tat. Aggressive má druhý nejrozdílnější medián od průměru. Znatelně vyšší průměr než medián je zde pravděpodobně způsoben vysoce nadprůměrnými výsledky her (přibližně desetinásobek průměru) se strategií passive, jedná se o 10 % her.

Na druhé pozici dle imaginárního sestupného řazení mediánového skóre grafu 3 se nachází nelítostná strategie unforgiving se skóre 3 716 v minimálně polovině her těsně následovaná tft_&_random_forgive_aggressivity_33%, která je rovněž druhou nejlepší podle průměrného skóre her. Na základě mediánového skóre jsou nejhoršími strategiemi tft_&_random_aggressive_33% a aggressive se záporným mediánovým skóre -505. Celkově má osm strategií nadprůměrné skóre. Celkové průměrné skóre je nižší než celkový medián, neboť poslední dvě strategie mají značně podprůměrné skóre.

3.3.3 Dvourozměrné prostorové věžňovo dilema

Experimenty prostorového věžňovo dilematu (zkráceně „SPD“ – Spatial prisoners dilemma) byly provedeny v evoluční a neevoluční variantě. Parametrem obou variant je celkový počet hráčů, mezi které jsou rovnoměrně rozděleny strategie. Počet hráčů lze nastavit změnou velikosti grafického výstupu modelu. Musí být dělitelný deseti a minimální velikost je $3 * 3$. Počáteční počty byly $n = \{20, 100, 900, 2\ 500, 10\ 000\}$. Různé počáteční velikosti populací měly odhalit případný vliv na úspěšnost některých strategií. Obzvláště strategie spolupráce by mohla být úspěšnější v menších komunitách. Výsledky experimentů počáteční velikosti populace n se nacházejí v příslušném listu „SPD exp n “ souborů „SPD experiment evolutionary.xlsx“ a „SPD experiment non-evolutionary.xlsx“, viz příloha 2).

Strategie popsané v kapitolách 3.1.3 a 3.2.3, byly navrženy a implementovány pro účely experimentů. Některé strategie obsahují prvky náhodnosti. Aby byl vliv náhodnosti na výsledky experimentů omezen, byla každá kombinace vstupních parametrů opakována stokrát. Celkový počet her byl $2 * 5 * 100$ pro evoluční a neevoluční model a pro pět různých velikostí počátečních populací. Každá hra proběhla na 1 000 kol a výsledky každého kola byly zaznamenány na samostatném řádku příložených souborů „SPD experiment evolutionary.xlsx“ a „SPD experiment non-evolutionary.xlsx“, viz příloha 2).

Cílem modelů je porovnat úspěšnost strategií ve hře proti sousedním oblastem. Úspěšnost byla posouzena ve čtyřech dílčích oblastech. Jednou z nich je průměrná a mediánová velikost populace strategií v posledním kole hry, která je proměnlivá pouze u evolučního modelu. Dalším kritériem evolučního modelu jsou průměrné a mediánové časy vyhynutí strategií, které jsou určeny na základě hodnot seznamu `strategies_extinction_time` v posledním kole hry. Pro evoluční model jsou také zaznamenány průměrné a mediánové časy stabilizace populací v posledním kole.

Kritériem pro hodnocení experimentů je v neposlední řadě také průměrné a mediánové kumulativní skóre strategií v posledním kole sta různých her. Výpočet kumulativního skóre viz kapitola 3.2.3, skóre je uloženo v atributu `strategies_score_sum`. Kumulativní skóre strategií evolučního a neevolučního modelu by nebylo vhodné porovnávat, neboť strategie neodehrají stejný počet kol. Lze je porovnat v posledním kritériu, které představuje průměrné a mediánové skóre jedince strategie na jedno kolo období existence. Všechny výsledky obsahují příložené soubory „SPD experiment evolutionary.xlsx“ a „SPD experiment non-evolutionary.xlsx“, viz příloha 2).

Tabulky 7, 8, 9 a 10 obsahují v záhlaví různé počáteční počty hráčů. První sloupec obsahuje názvy strategií a poslední sloupec souhrny ze všech velikostí počátečních populací pro strategii na daném řádku. Průměrné hodnoty se nachází na zeleně zvýrazněném řádku, mediánové na následujícím nezvýrazněném řádku. Hodnoty byly vypočítány na základě dat posledního kola sta her, sloupcové souhrny z 10 * 100 hodnot pro 10 strategií a 100 opakování a řádkové souhrny z 5 * 100 hodnot pro pět různých velikostí populací a sto opakování hry. Celkový mediánový souhrn řádků i sloupců je vypočítán z 10 * 5 * 100 hodnot. Celkový průměrný souhrn je možné vypočítat jako průměry kategorií, neboť jednotlivé řádkové průměry jsou spočítány ze stejného počtu 500 hodnot (celkem 10 * 500 hodnot za řádky) a jednotlivé sloupcové souhrny z 1 000 hodnot (celkem 5 * 1 000 hodnot za sloupce).

Tabulka 7: Populace strategií na konci hry (evoluční model SPD)

Strategie Počáteční počet hráčů	20	100	900	2 500	10 000	Celkem
random	průměr	0	0	0	0	0
	medián	0	0	0	0	0
cooperate		0	0,05	1,04	22,63	4,85
		0	0	0	0	0
defect		19,61	86,04	763,30	1 568,97	5 961,92
		20,00	95,00	795,00	1 539,50	5 967,00
tit_for_tat		0	1,60	16,33	82,30	20,53
		0	0	0	0	0
tit_for_two_tats		0	0,21	1,70	1,70	0
		0	0	0	0	0
unforgiving		0	3,77	26,74	27,81	0
		0	0	0	0	0
even_cooperate_odd_defect		0,24	1,67	25,17	407,74	2 110,96
		0	0	0	441,50	2 098,00
cooperate_twice_defect_twice		0	0	0	0	0
		0	0	0	0	0
tft_&_random_defect_33%		0,15	5,96	62,06	370,21	1 901,74
		0	0	50,50	441,50	1 913,00
tft_&_rand_forgive_defect_33%		0	0,70	3,66	18,64	0
		0	0	0	0	0
Průměrně přeživších strategií	1,03	1,84	2,81	3,11	3,01	2,36
Mediánově přeživších strategií	1	2	3	3	3	3

Zdroj: vlastní zpracování

Tabulka 7 obsahuje průměrné a mediánové populace jednotlivých strategií na konci hry. Poslední dva řádky obsahují průměrné a mediánové počty přeživších strategií. Pro přeživší strategie byl vypočítán počet nenulových populací strategií, průměr a medián příslušného modelu velikosti populace je spočítán ze sta hodnot opakování hry. Počáteční

populace strategií jsou rovnoměrné. Tabulka 7 neobsahuje sloupcové souhrny, neboť součty průměrných populací strategií představují počáteční velikost populace daného sloupce (například $19,61 + 0,24 + 0,15 = 20$) a průměry sloupců reprezentují velikost populace jedné strategie.

Největší konečné populace měla podle průměru i mediánu strategie defect. Pro 20 hráčů existovalo na konci kola průměrně 19,61 a mediánově 20 hráčů a pro 10 000 hráčů 5 961,92 a medián 5 967. Celkový průměr 1 679,97 a medián 787. Druhou nejúspěšnější strategií podle průměrné konečné populace je even_cooperate_odd_defect a třetí tft_&_random_defect_33%. Strategie even_cooperate_odd_defect měla na rozdíl od třetí strategie v polovině her na konci nulovou populaci. První tři strategie dle průměrného skóre mají celkové průměrné populace značně vyšší než mediánové, což je způsobeno hodně nadprůměrnými populacemi na konci kol z modelu 10 000 hráčů.

S průměrně i mediánově nulovými konečnými populacemi skončily strategie random a cooperate_twice_defect_twice. Třetí nejméně úspěšná je tft_&_random_forgive_defect_33% s průměrnou populací 4,6 a v polovině případů s nulovou. Konečné počty hráčů prvních třech strategií v pořadí jsou rostoucí se zvětšující se počáteční velikostí populace. Konečné populace poloviny strategií byly průměrně nižší u 10 000 hráčů než u 2 500, mediánové populace jsou však u všech nulové. Průměrný počet přeživších strategií u 20 hráčů je přibližně jedna, u 100 hráčů dvě a s větší počáteční populací přibližně tři průměrně i mediánově.

Tabulka 8: Časy vyhynutí strategií a stabilizace populací (evoluční model SPD)

Strategie Počáteční počet hráčů	20	100	900	2 500	10 000	Celkem	
random	průměr	2,22	4,73	8,04	11,45	15,29	8,35
	medián	2,00	4,00	7,00	10,00	14,00	7,00
cooperate		1,36	12,41	15,33	81,49	57,23	33,56
		1,00	2,00	4,00	6,00	12,00	4,00
defect		1000	970,49	982,21	981,04	1000	986,75
		1 000,00	1 000,00	1 000,00	1 000,00	1 000,00	1 000,00
tit_for_tat		1,26	182,2	440,58	403,57	363,13	278,15
		1,00	2,00	86,50	186,00	334,00	73,00
tit_for_two_tats		1,36	32,39	145,59	90,28	121,19	78,16
		1,00	2,00	5,00	48,00	109,50	4,00
unforgiving		1,25	232,45	384,89	249,9	183,75	210,45
		1,00	2,00	77,00	122,00	168,00	83,00
even_cooperate_odd_defect		21,71	27,54	395,96	822,57	1000	453,56
		1,00	5,00	256,00	1 000,00	1 000,00	81,00
cooperate_twice_defect_twice		1,32	2,63	6,92	9,3	13,11	6,66
		1,00	2,00	7,00	9,00	11,00	7,00
tft_&_random_defect_33%		11,36	392,53	731,96	835,81	991,17	592,57
		1,00	3,50	1 000,00	1 000,00	1 000,00	1 000,00
tft_&_rand_forgive_defect_33%		1,38	12,48	18,46	54,78	65,24	30,47
		1,00	2,00	6,00	11,50	34,00	5,00
Celkem průměr		104,32	186,99	312,99	354,02	381,01	267,87
Celkem medián		1,00	2,50	15,00	74,50	151,00	10,00
Průměrný čas stabilizace populací		3,87	10,99	168,84	21,56	0	41,052
Mediánový čas stabilizace populací		3,00	8,00	42,00	0	0	3,00

Zdroj: vlastní zpracování

Tabulka 8 reprezentuje časy vyhynutí strategií a v posledních dvou řádcích průměrné a mediánové časy stabilizace populací. Průměr a medián časů byl vypočítán na základě sta hodnot příslušné počáteční populace, celkový souhrn z 5 * 100 hodnot pro pět různých velikostí počátečních populací. Nulové časy vyhynutí strategií (populace se zachovaly až do konce hry), byly nahrazeny hodnotou 1 000, aby bylo možné zjistit „výdrž“ strategie.

Průměrně i mediánově nejdéle vydržela strategie defect následovaná tft_&_random_defect_33%. Třetí strategií podle průměrné výdrže je even_cooperate_odd_defect. Podle mediánového pořadí je třetí nejúspěšnější unforgiving s mediánovou hodnotou času vyhynutí strategie 83. Even_cooperate_odd_defect přežila v polovině případů alespoň 81 kol. Nejméně úspěšné podle průměru jsou strategie cooperate_twice_defect_twice, random a tft_&_rand_forgive_defect_33% v pořadí od nejnižšího do nejvyššího průměrného času. Mediánově nejméně úspěšné jsou cooperate, tit_for_two_tats a tft_&_rand_forgive_defect_33%.

Průměrný čas stabilizace strategií se od velikosti počáteční populace 20 do 900 zvyšuje a poté klesá, stejně tak podle mediánových hodnot. Celkový průměrný čas stabilizace je vyšší než mediánový z důvodu vysoce nadprůměrných hodnot her, které se stabilizovaly až v 700. či 800. kole. Z obdobného důvodu je vyšší i celkový průměr časů vyhynutí strategie než medián. Strategie defect v polovině případů nevyhynula vůbec a její nenulový čas vyhynutí nahrazený hodnotou 1 000 (nejvyšší výdrž) zvyšuje celkový průměr.



Graf 4: Vývoj populací strategií v čase u 10 000 hráčů

Zdroj: vlastní zpracování

Na grafu 4 lze vidět vývoj struktury populací v prvních a posledních 40 kolech hry. Vyobrazen je průměrný vývoj populací strategií ze sta her v modelu s počátečním počtem 10 000 hráčů. Ostatní velikosti populací mají vývoj obdobný. Zpočátku se rychle rozšíří strategie defect a poté se ještě zmenšuje s evolucí jiných strategií, zejména even_cooperate_odd_defect (druhá nejčetnější v posledních 40 kolech) a tft_&_random_defect_33% (třetí nejčetnější v posledních 40 kolech). Stejnou úspěšnost těchto strategií v konečné četnosti populací potvrzuje i tabulka 7.

Tabulka 9: Kumulativní skóre strategií (evoluční model SPD)

Strategie Počáteční počet hráčů	20	100	900	2 500	10 000	Celkem	
random	průměr	72,91	133,86	214,35	304,37	396,97	224,49
	medián	68,88	122,39	194,19	254,22	354,61	189,59
cooperate		46,87	290,84	591,92	3 167,24	2 182,95	1 255,96
		42,00	63,60	110,08	162,86	354,20	107,88
defect		24 159,57	23 346,14	23 978,18	24 448,22	24 988,33	24 184,09
		24 037,23	24 041,68	24 297,28	24 999,89	24 991,21	24 178,46
tit_for_tat		45,34	4 398,17	11 314,23	11 160,63	9 718,39	7 327,35
		39,00	64,20	2 340,94	5 049,81	8 861,74	1 928,30
tit_for_two_tats		45,59	794,69	3 626,16	2 440,17	3 305,40	2 042,40
		39,00	62,80	142,26	1 257,41	2 835,31	116,97
unforgiving		44,09	6 077,28	9 949,71	6 830,12	4 950,21	5 570,28
		39,00	65,10	1 919,27	3 257,82	4 446,52	2 123,70
even_cooperate_odd_defect		657,03	954,36	12 794,16	26 775,59	32 583,37	14 752,90
		39,00	147,97	8 335,42	32 454,80	32 580,61	2 523,24
cooperate_twice_defect_twice		45,59	76,16	183,10	237,58	334,11	175,30
		42,00	61,20	179,29	218,83	281,12	172,58
tft_&_random_defect_33%		289,21	9 456,69	18 233,95	21 782,20	26 042,46	15 160,90
		42,00	108,10	24 042,87	26 271,05	26 262,57	24 037,58
tft_&_rand_forgive_defect_33%		47,85	548,02	698,67	2 082,67	2 247,73	1 124,99
		42,00	63,00	166,53	312,78	1 031,02	152,15
Celkem průměr		2 545,40	4 607,62	8 158,44	9 922,88	10 674,99	7 181,87
Celkem medián		42,00	74,20	418,32	1 949,32	4 289,65	262,97

Zdroj: vlastní zpracování

Jak lze vidět v tabulce 9, průměrné kumulativní skóre na jedince potvrzuje úspěšnost prvních třech strategií podle konečných populací na konci hry, viz tabulka 7. Rozdíl je v pořadí strategií na druhém a třetím místě. Tft_&_random_defect_33% je podle průměrného kumulativního skóre druhá a even_cooperate_odd_defect třetí. Stejné pořadí strategií potvrzují také mediánová skóre. Vysoké kumulativní skóre získala také strategie tit_for_tat. Průměrná skóre jsou vyšší než mediánová, neboť jsou průměry značně ovlivněny několika značně nadprůměrnými hodnotami kumulativního skóre. U strategií cooperate, tit_for_tat a unforgiving je kumulativní skóre počáteční populace 10 000 nižší oproti 2 500.

Vyšší skóre v porovnání k velké četnosti nižších skóre vzniká ve hrách, ve kterých strategie přežije až do konce hry a skóre od přeživších jedinců je stále přičítáno v atributu kumulativních skóre. Strategie jsou tedy porovnány dle řádkových mediánových souhrnů. Je potvrzeno pořadí prvních třech přiček v pořadí defect, tft_&_random_defect_33% a even_cooperate_odd_defect od nejvyššího kumulativního skóre po nejnižší.

Nad celkovým průměrem 7 181,87 se nachází průměry konečných populací čtyř strategií, průměr je zvýšen nad medián 262,97 v důsledku cca. 6 % nadprůměrných hodnot, které

jsou v rozsahu od 24 000 do 33 000. Nadprůměrné hodnoty kumulativního skóre na konci her vznikly u modelu 10 000 hráčů u třech strategií – defect, even_cooperate_odd_defect a tft_&_random_defect_33% (3 * 100 opakování z 10 * 5 * 100 hodnot pro 10 strategií, pět různých počátečních populací a sto různých pokusů stejné kombinace = 6 %).

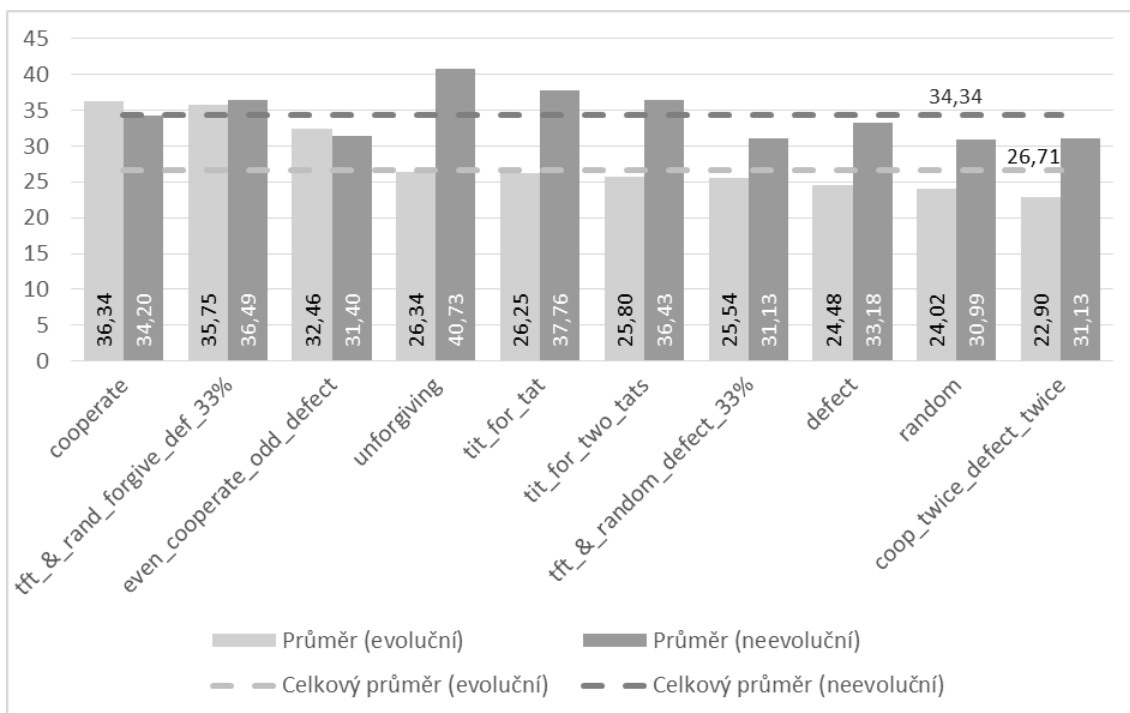
Tabulka 10: Kumulativní skóre strategií (neevoluční model SPD)

Strategie Počáteční počet hráčů	20	100	900	2 500	10 000	Celkem	
random	průměr	31 075,66	30 945,13	31 016,49	31 052,87	31 003,42	31 018,71
	medián	31 308,25	30 977,95	30 981,26	31 037,15	31 003,62	31 012,65
cooperate		33 789,18	34 196,65	34 405,97	34 361,06	34 411,85	34 232,94
		34 020,00	34 002,30	34 403,37	34 363,60	34 423,68	34 378,01
defect		33 451,62	33 341,59	33 113,22	33 108,91	33 074,00	33 217,87
		33 419,00	33 370,00	33 106,09	33 114,02	33 075,23	33 116,68
tit_for_tat		37 099,24	37 868,93	38 000,17	38 010,08	38 022,63	37 800,21
		37 007,25	37 891,85	38 004,04	38 009,78	38 011,44	37 990,01
tit_for_two_tats		35 823,85	36 664,49	36 622,69	36 611,87	36 607,13	36 466,01
		35 297,00	36 796,45	36 618,47	36 629,22	36 614,59	36 581,45
unforgiving		40 671,20	40 774,30	40 787,48	40 802,24	40 806,50	40 768,34
		40 962,75	40 842,70	40 795,12	40 816,52	40 827,31	40 817,27
even_cooperate_odd_defect		31 278,41	31 453,03	31 444,26	31 462,91	31 502,39	31 428,20
		31 142,00	31 480,60	31 443,21	31 491,75	31 496,29	31 467,27
cooperate_twice_defect_twice		30 992,53	31 145,65	31 212,27	31 204,60	31 243,00	31 159,61
		30 764,00	31 186,70	31 283,39	31 192,80	31 249,97	31 236,98
tft_&_random_defect_33%		31 501,42	31 156,15	31 082,84	31 031,01	31 053,10	31 164,90
		31 487,75	31 226,35	31 127,96	31 029,12	31 052,97	31 080,48
tft_&_rand_forgive_defect_33%		35 931,72	36 460,35	36 708,32	36 790,61	36 764,41	36 531,08
		35 830,50	36 594,95	36 729,84	36 794,80	36 778,97	36 740,02
Celkem průměr		34 161,48	34 400,63	34 439,37	34 443,62	34 448,84	34 378,79
Celkem medián		33 955,75	33 726,80	33 651,81	33 642,00	33 567,59	33 736,45

Zdroj: vlastní zpracování

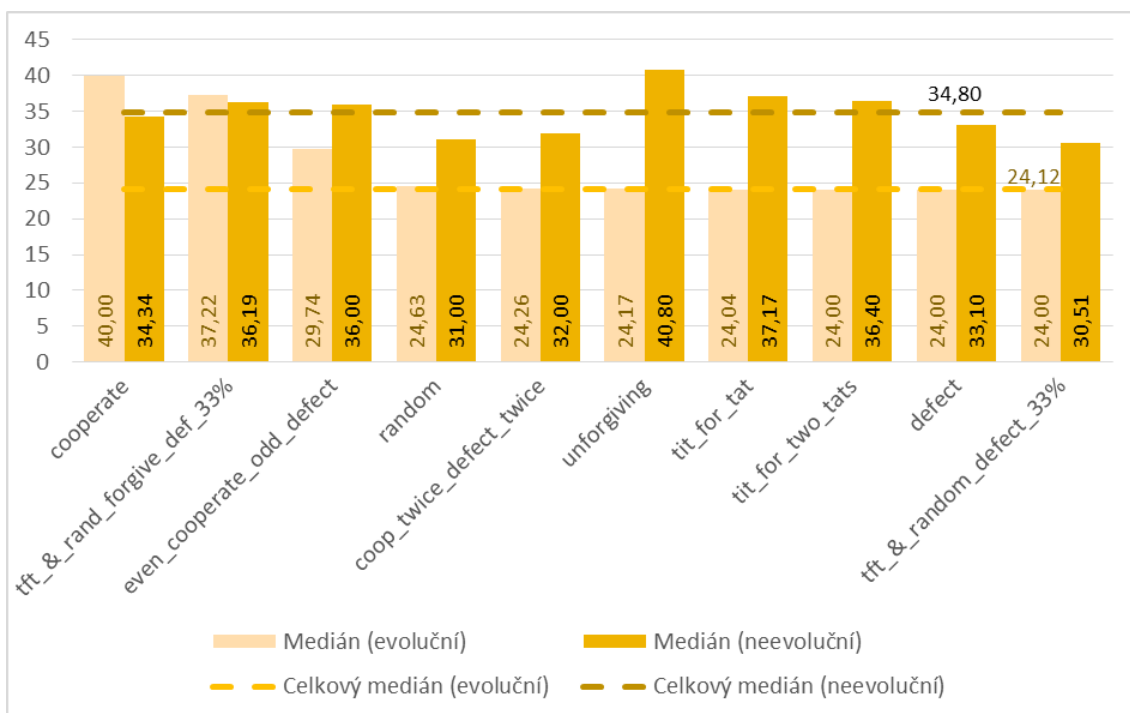
Tabulka 10 obsahuje kumulativní skóre strategií neevolučního modelu SPD pro různé počáteční velikosti populací. Oproti evolučnímu modelu jsou na prvních třech pozicích jiné strategie. Podle sestupného řazení průměrného i mediánového kumulativního skóre strategií je nejlepší strategií unforgiving, druhou nejlepší tit_for_tat a třetí nejlepší tft_&_random_forgive_defect_33%. Jedná se o adaptabilní strategie, které umí oplácet zrady.

Nejméně úspěšné podle průměru i mediánu jsou strategie tft_&_random_defect_33%, cooperate_twice_defect_twice a random. Tft_&_random_defect_33% je u evolučního modelu podle průměrného kumulativního skóre druhou nejúspěšnější strategií. U neevolučního modelu je podle kumulativního skóre na předposledním místě a podle mediánu na osmém. Strategie cooperate je na pátém místě sestupného řazení průměrného i mediánového kumulativního skóre. Všechny souhrnné průměry i mediány jsou přibližně stejné.



Graf 5: Průměrné skóre hráče na jedno kolo existence (evoluční a neevoluční model)

Zdroj: vlastní zpracování



Graf 6: Mediánové skóre hráče na jedno kolo existence (evoluční a neevoluční model)

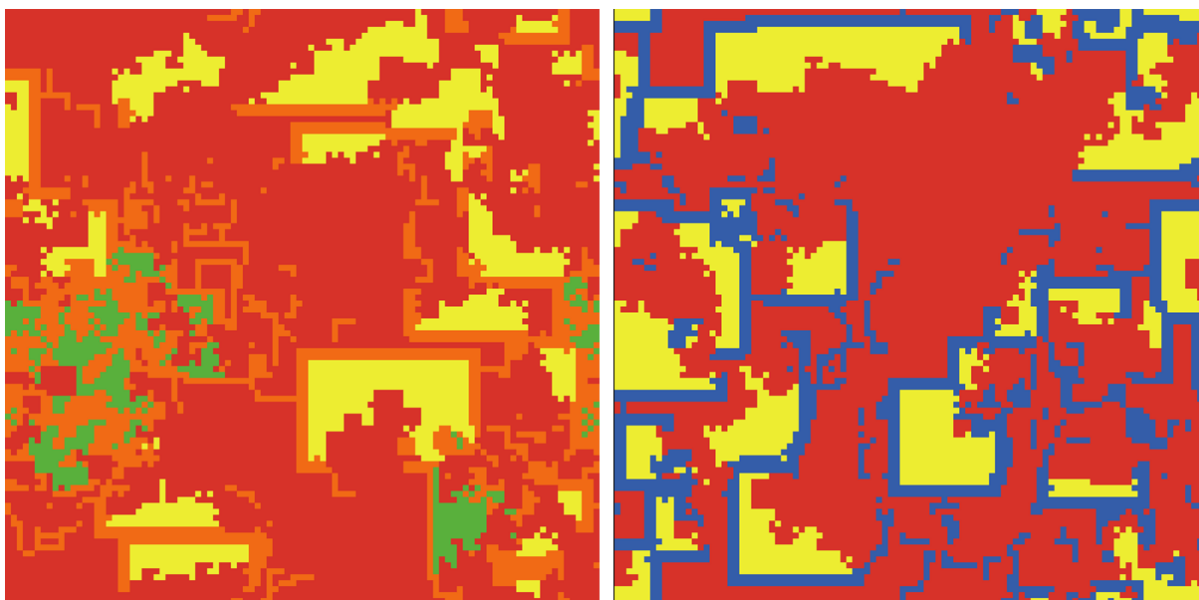
Zdroj: vlastní zpracování

Graf 5 a graf 6 vyobrazuje porovnání evolučního a neevolučního modelu v průměrném a mediánovém skóre hráče na jedno kolo existence strategie. Počty kol, které odehrály strategie evolučního modelu, jsou jiné než u neevolučního modelu, u kterého všechny strategie odehrály tisíc kol. Hodnoty za různé velikosti počátečních populací byly agregovány.

Hodnoty byly zjištěny z $5 * 100 * 1\,000$ řádků pro pět různých počátečních populací, sto opakování každé kombinace vstupních parametrů a 1 000 kol hry. Nejdříve byla u řádku ověřena podmínka nenulovosti populace strategie, poté podmínka, zda se jedná o řádek s daty z experimentů (soubory obsahují i řádky hromadně vkládaných souhrnů). Pokud byly obě podmínky splněny, Excel započítal do celkového průměru či mediánu hodnotu celkového skóre strategie v daném kole vydělenou celkovým počtem hráčů strategie.

Jak lze vidět z grafu 5, v porovnání s evolučním modelem byly u neevolučního modelu značně úspěšnější strategie unforgiving, která měla o 14,38 vyšší skóre jednoho hráče v jednom kole, tit_for_tat s rozdílem 11,51 a tit_for_two_tats s 10,63. U zbylých strategií byly výsledky méně rozdílné. Strategie cooperate a even_cooperate_odd_defect měly o něco vyšší skóre u evolučního modelu než u neevolučního modelu. Strategie defect je úspěšná v jiných kritériích než průměrné skóre jedince v jednom kole.

Rozdíl ve výsledcích strategií unforgiving, tit_for_tat a tit_for_two_tats potvrzují také mediánové hodnoty u grafu 6. Rozdíly neevolučního modelu oproti evolučními jsou kladné, v polovině případů měli hráči strategie v jednom kole vyšší skóre než u evolučního modelu. Hodnoty činí 16,63, 13,13 a 12,40. Mediánově je strategie cooperate u evolučního modelu v jednom kole hry nejúspěšnější.



Obrázek 4: Nestabilizovaná populace strategií na konci hry, 10 000 hráčů (model SPD)

Zdroj: vlastní zpracování

Obrázek 4 reprezentuje nestabilizované populace strategií po 1 000 kolech modelu 10 000 hráčů. Levá část obrázku představuje ojedinělý výsledek hry, ve kterém strategie cooperate (zelená) přežila až do konce (jedná se o jednu ze sta her). Obklopena je hráči

strategie `tit_for_tat`. Spolupráce s touto strategií je jednou z možných šancí na přežití, neboť `tit_for_tat` dokáže adaptovat zlé i hodné strategie. Žlutá představuje strategii `even_cooperate_odd_defect`, která je v levé části ohraničena strategií `tit_for_tat` a v pravé části `tft_&_random_defect_33%` (modrá).

Pravá část obrázku reprezentuje nejčastěji vzniklé vzory struktury populace, která vznikla na konci her. `Even_cooperate_odd_defect` mění po jednom kole spolupráci a zradu a má tak šanci na přežití mezi zlými strategiemi. Okolo ní se v hranatých strukturách shromažďuje strategie `tit_for_tat` zkombinovaná s náhodnou zradou v přibližně třetině případů, která pravděpodobně z části dokáže odstínit zlé chování strategie `defect` a zároveň přežít ve většině zrazující populace lépe než samotná `tit_for_tat` díky větší míře nezávislých zrad. Většinu populace obou situací tvoří strategie `defect` (červená).

Vzorové grafické výstupy konečné struktury ostatních počtů hráčů a barevnou legendu strategií obsahuje příloha 1). Výstupy k experimentům byly získány za pomoci dočasně implementované funkce zachycující grafický výstup modelu po zadaném počtu kol hry do souboru s příponou „.PNG“. Soubory byly rozlišeny číslem od jedné do 9 mil. (sloupec `viewID` v souborech experimentů), generovaným pomocí funkce `random`. Funkce automatického ukládání obrázků byla po provedení experimentů odebrána, neboť by při nechtěném zapnutí mohla zahltit disk či složku ve stejné lokaci, ve které je uložen model v NetLogu.

3.4 Diskuze

Kapitola obsahuje shrnutí některých výsledků, diskuzi o možných příčinách, možné dopady principů chování strategií, postřehy autora a podněty pro další zkoumání.

3.4.1 Sosisova hra v kibucech

Hodnotícími kritérii strategií Sosisovy hry byly průměrné a mediánové reálné zisky her a průměrné a mediánové absolutní přírůstky reálných zisků oproti požadovaným. Z výsledků experimentů vyplývá, že je poměrně úspěšné a napříč klíčovými hodnotícími faktory stabilní požadovat v současném kole předchozí reálně získanou částku, pokud byla nenulová (`repeat_if_real_earn`). Strategie se umístila na druhém místě podle průměrných reálných zisků, na třetím podle mediánových reálných zisků a na čtvrtém dle absolutních průměrných i mediánových přírůstků. Kladné výsledky této strategie mohou být způsobeny rychlou adaptací. Když hráč soupeří se strategií, se kterou nezískává reálné zisky, tak zvolí požadované částky náhodně. V takovém procesu je možné nalézt částku, kterou by měl hráč průměrně požadovat.

Výsledky experimentů sice potvrdily předpoklad úspěšnosti strategie complement v získávání reálných zisků, avšak strategie reálně získala méně, než kolik požadovala, průměrně o 7 134,19, přičemž v polovině případů minimálně o 69 méně. Pravděpodobně je to dáno tím, že complement ostatním strategiím ustupuje, pokud požadují více než polovinu rozdělovaného zisku. Skromnost v požadavcích částky 10 či 20 a investování všeho do rozvoje kibucu se navrácí absolutně nejvíce, byť se v reálných ziscích nejedná o největší získané částky. Strategie take_20 je sedmá, take_10 devátá a invest_all jedenáctá ze šestnácti strategií v průměrných reálných částkách.

I přesto, že výsledky strategie tit_for_tat jsou u vězňova dilematu velmi dobré, implementací jejích principů nebylo dosaženo dobrých výsledků a strategie se umístila na prostředních až spodních příčkách. Obdobně dopadla strategie náhodné volby v každém kole, která v jednom kole průměrně požadovala přibližně 50. Reálně požadavků ale nebylo dosaženo. Volba částky 75 náhodně přibližně ve třetině případů, jinak opakování požadavků, pokud bylo dosaženo reálného zisku, u strategie repeat_if_real_earn_and_random75_33% nenaplnila očekávání dobrých výsledků vyšší požadované částky v kombinaci s adaptačními principy strategie repeat_if_real_earn.

Nejméně úspěšné v Sosisově hře je být lakomý, i když je dosaženo převahy nad skromnými strategiemi. Skromných je málo a lakomé strategie mohou jinak získat převahu pouze nad ústupnými strategiemi complement, repeat_if_real_earn a real_as_required, jak potvrzují dílčí výsledky her lakomých strategií se všemi ostatními v přiloženém souboru „Sosis game experiment.xlsx“, viz příloha 2).

Šest z 19 navržených strategií překonalo požadované částky v reálném výdělku průměrně i mediánově. Real_as_required je těsně pod touto hranicí. Strategie požadování poloviny zisku nejméně v 50 % případů skutečně polovinu zisku získala. V získaných částkách jsou však nejbližší k požadavkům repeat_4 a real_as_required (medián i průměr). Mohlo by se jednat o jistá východiska při volbě strategií. Podnětem pro další zkoumání by mohlo být zkombinování jiných principů s nejstabilnější strategií repeat_if_real_earn či zahrnout do stabilní strategie principy doplňků k soupeřovým požadovaným částkám (complement).

3.4.2 Boj o zdroje

Z výsledků experimentů modelu boje o zdroje vyplývá, že je výhodnější se o cenné zdroje raději podělit než o ně bojovat. Pasivita a agresivita jsou podle průměrného skóre téměř zcela protikladné, pasivita neúspěšnější a agresivita druhá nejhorší. Značné rozdíly těchto

dvou strategií potvrdily i mediánové hodnoty. Předpoklad obdobného skóre strategií unforgiving, aggressive a tit_for_tat navzájem se naplnil pouze z části. Podobné skóre mají strategie unforgiving a tit_for_tat a to jak průměrné, tak mediánové.

Kombinace strategie tit_for_tat s náhodným odpuštěním agresivity přibližně ve třetině kol se jeví úspěšně a poměrně stabilně napříč kritérii průměrného i mediánového skóre. Jestliže tft_&_random_forgive_aggressivity_33% hraje proti adaptabilním strategiím a jedinci by bojovali ve více kolech po sobě, je možné se z boje navrátit zpět ke spolupráci. Přibližně ve třetině případů odpuštění však ztrácí body proti trvale agresivní strategii. Oproti strategii pasivity má jen o málo nižší průměrné skóre a zvýšením procenta případů, ve kterých dochází k odpuštění agresivity, by strategie mohla dosáhnout lepších výsledků.

Strategie oplácení předchozího soupeřova tahu tit_for_tat je u vězňova dilematu značně úspěšná, avšak implementováním záporných užitků představujících zranění z boje si pravděpodobně přihorší při soupeření s agresivní či částečně agresivní strategií. I když je agresivita nevýhodná, strategie pasivity v sudém kole a agresivity v lichém kole skončila na třetím místě dle průměrného skóre her. Strategie může být jistým východiskem pro zavědění se klidným strategiím a odrazení agresivních strategií či agresivity v adaptabilních.

Strategie even_passive_odd_aggressive, random a passive_twice_aggressive_twice měly v polovině případů skóre alespoň 2 500. Společně by se tak řadily na šestou příčku sestupně řazeného mediánového skóre. Střídání pasivity v sudém a agresivity v lichém kole je třetí nejúspěšnější strategie podle průměrného skóre. Pokud by bojující ptáci neztratili část své energie, pravděpodobně by even_passive_odd_aggressive bylo ještě úspěšnější. Proti pasivním jedincům by získala přibližně stejné skóre, ale oproti agresivním lepší.

Poslední dvě strategie doplatily na agresivitu proti ostatním. Poslední strategie podle průměrného skóre je tft_&_random_aggressive_33%. Strategie je sice náhodně agresivní pouze ve třetině případů, ale proti hře s adaptabilními strategiemi měla náhodná agresivita fatální dopady na výši skóre. Jakmile začne strategie tft_&_random_aggressive_33% s náhodnými zradami, tak adaptabilní soupeř začne tahy opakovat a navzájem nakonec mohou oba hráči dojít k trvalé agresivitě ve zbytku hry. Chování agresivních strategií je opláceno stejně jako u ostatních adaptabilních strategií. Snížením procenta případů náhodné agresivity by mohlo být dosaženo lepších výsledků.

3.4.3 Dvourozměrné prostorové věžňovo dilema

Důležitými kritérii pro vyhodnocení výsledků experimentů se strategiemi byly velikost populace strategií na konci hry, časy vyhynutí strategií, kumulativní skóre strategie a skóre jedince strategie na jedno kolo období existence. U evolučního modelu strategií bylo předpokládáno ustálení evolučního procesu. Se zvětšujícím se počtem hráčů se zvyšuje i doba ustálení a vzhledem ke stanovenému počtu kol hry na 1 000 u vyšších počátečních populací ani nedošlo ke stabilizaci. Zajímavým podnětem pro další zkoumání je, zda by vůbec došlo ke stabilizaci u velkých počtů hráčů. Z původních deseti strategií zůstalo na konci kola průměrně 2,36, u menších modelů jedna strategie, u větších tři. Modely tak poukazují na evoluci odlišných strategií do několika málo přeživších skupin.

Celkově největší populace na konci hry měla strategie defect, která se na počátku hry poměrně rychle rozšíří v důsledku počáteční spolupráce některých strategií. Spolupráce jako úvodní prvek hry v prostorovém věžňově dilematu není úspěšná. Předmětem dalšího zkoumání by mohla být volba jiné "výchozí" akce strategií než spolupráce, aby bylo zabráněno rychlému rozšíření zrady. Strategie střídající spolupráci v sudém kole a zradu v lichém kole měla však dobré výsledky i v konkurenci velkého počtu zrazujících hráčů. Strategie tit_for_tat se dokáže dobře adaptovat, pokud je modifikována náhodnou zradou přibližně ve třetině případů. V nemodifikované variantě nebyla nejúspěšnější strategií, ale podle průměrné konečné populace je čtvrtá.

Častou nestabilizovanou strukturou na konci hry modelu 10 000 hráčů byly hranaté útvary strategie even_cooperate_odd_defect ohraničené strategií tft_&_random_defect_33%. Přibližná tloušťka ohraničení činí dvě políčka (hráči). V jednom ze sta případů přežila strategie spolupráce až do konce hry. Při nestejném počátečním rozdělení strategií pro hráče by bylo možné získat více bodů ze vzájemné spolupráce. Zrada by se však na úkor spolupráce u velkého počtu hráčů pravděpodobně stejně rozšířila a je vhodné, aby populace obsahovala i adaptabilní strategie či strategie, které své chování periodicky mění, pro vykompenzování zlého chování a ochránění hodných jedinců. Hodné strategie přežívají spíše v uzavřených komunitách ohraničených adaptabilními strategiemi či strategiemi, které mění své chování nezávisle na okolí.

Časy vyhynutí potvrzují úspěšnost stejných strategií jako u konečných populací s tím rozdílem, že druhou nejúspěšnější strategií průměrně je tft_&_random_defect_33% a třetí even_cooperate_odd_defect. Průměrný a mediánový čas stabilizace populací odhalil ros-

toucí hodnoty stabilizace do velikosti populace 900 a u větších populací častěji nedocházelo vůbec ke stabilizaci. Do velikosti populace 2 500 se průměrná populace pěti strategií zvětšovala a u modelu 10 000 hráčů byla menší. Mediánové populace tento vztah nepotvrzují. Průměrné kumulativní skóre třech strategií bylo u modelu 10 000 hráčů nižší než u 2 500, medián je však rostoucí. Pro případnou korelační a regresní analýzu pro odhalení míry vztahu a regresní funkce popisující vztah by bylo nutné provést více experimentů s více velikostmi počátečních populací. Korelační a regresní analýza nebyla cílem práce.

Podle kumulativního skóre u evolučního modelu jsou úspěšné jiné strategie než u neevolučního modelu. Nejlepšími strategiemi podle průměru i mediánu neevolučního modelu jsou `unforgiving`, `tit_for_tat` a `tft_&_rand_forgive_defect_33%` a nejlepší strategie evolučního jsou `defect`, `tft_&_random_defect_33%`, `even_cooperate_odd_defect`. `Unforgiving` je podle kumulativního skóre evolučního modelu pátá podle průměru a čtvrtá podle mediánu. Strategie zrady je u neevolučního modelu na šestém místě. Spolupráce je podle kumulativního skóre u evolučního modelu nejméně úspěšná, u neevolučního modelu se nachází na pátém místě. Strategie `even_cooperate_odd_defect` je třetí nejúspěšnější strategií evolučního modelu, avšak u neevolučního modelu je podle sestupného řazení průměru i mediánu sedmá.

Původním předpokladem pro experimenty pěti různých počátečních populací byla úspěšnost strategie spolupráce v menších komunitách. V kritériích velikosti populace a době výdrže strategie se úspěšnost nepotvrdila ani u nejmenšího možného počtu hráčů 20. Podle průměrného i mediánového kumulativního skóre strategie je `cooperate` na šesté pozici v sestupném řazení. U evolučního modelu byla strategie spolupráce průměrně i mediánově nejúspěšnější v kritériu skóre hráče na jedno kolo existence. Pravděpodobně je to způsobeno tím, že velký počet strategií začíná spoluprací. Při rozšiřování strategie `defect` strategie spolupráce rychle mizí, ale dobré výsledky na jedno kolo existence strategie jsou zachovány. U neevolučního modelu se ve stejném kritériu umístila na pátém místě podle sestupného průměrného řazení a na šestém místě podle mediánového řazení.

Celkový průměr skóre na jednoho hráče v jednom kole je u neevolučního modelu vyšší než u evolučního pravděpodobně z důvodu poměrně rychlého rozšíření strategie `defect` u evolučního modelu. S tou poté hraje více strategií v porovnání s neevolučním modelem. Vzájemnou zradou s adaptabilními strategiemi samozřejmě získají méně bodů než u neevolučního modelu, ve kterém je větší prostor pro spoluprací.

4 Závěr

Cílem diplomové práce bylo zvolit a popsat varianty vězňova dilematu a vytvořit odpovídající modely v NetLogu. Tyto modely následně použít k experimentování se strategiemi, které byly v rámci této práce navrženy nebo převzaty z citovaných zdrojů. V teoretické části byly popsány základní pojmy a principy z teorie her a tvorby multiagentních modelů. Rovněž bylo popsáno vězňovo dilema a zvolené varianty, a sice Sosisova hra v kibucech, boj o zdroje (modifikovaná varianta Hawk Dove) a dvourozměrné prostorové vězňovo dilema (zkr. SPD).

Kapitoly praktické části jsou rozděleny podle zvolených variant vězňova dilematu, aby se čtenář mohl případně věnovat těm, které jej zajímají. Kapitola 3.1 je zaměřena na návrh modelů. Funkčnost strategií zajišťuje implementace v kódu modelu a tak bylo vhodné uvést strategie pro experimenty již v návrhu modelů a implementaci. Strategie random a tit_for_tat Sosisovy hry byly odvozeny od opakovaného vězňova dilematu (kapitola 2.2.2) a strategie 0.5_profit, invest_all a take_all na základě popisu Sosisovy hry (kapitola 2.2.3), ostatní strategie byly navrženy v rámci práce. Strategie modelů boje o zdroje a dvourozměrného prostorového vězňova dilematu vycházejí z opakovaného vězňova dilematu. Výplatní matice jsou stanoveny na základě popisu příslušné varianty v teoretické části.

Za návrhem modelů následuje popis implementace (globální proměnné, aktualizací procedury atp.), součástí jsou ukázky kódu. Pro implementaci byla využita verze NetLogo 5.2.1.

Modely byly použity k experimentům v nástroji BehaviorSpace. Byly popsány vstupní parametry a klíčová kritéria pro hodnocení strategií. U analýzy výstupů experimentů byly mj. uvedeny také příčiny některých rozdílů průměrných a mediánových výsledků. Protože strategie implementované v modelech využívají generátor diskretních pseudonáhodných čísel random, byl generátor ověřen Kolmogorov-Smirnovovým testem a Chí-kvadrát testem shod empirického a teoretického modelu. Generované hodnoty jsou přibližně rovnoměrně rozdělené.

Cílem experimentů modelu Sosisovy hry a boje o zdroje bylo porovnat úspěšnost jednotlivých strategií ve hře proti všem ostatním. Klíčové indikátory pro hodnocení strategií

Sosisovy hry byly průměrné a mediánové reálné zisky her a průměrný a mediánový absolutní přírůstek reálných oproti požadovaným ziskům. Strategie modelu boje o zdroje byly hodnoceny na základě průměrných a mediánových skóre her.

Z hlediska reálných zisků byly nejúspěšnějšími strategiemi Sosisovy hry podle průměru complement, repeat_if_real_earn a real_as_required a podle mediánu 0.5_profit, take_40 a repeat_if_real_earn. Z hlediska průměrných i mediánových absolutních přírůstků byly nejúspěšnější strategie invest_all, take_10 a take_20. Předpoklad úspěšnosti strategie complement byl potvrzen jen z části, strategie je úspěšná při získávání reálných zisků, ale získala průměrně o 7 134,19 a mediánově o 69 nižší částku, než jakou požadovala. Kombinace strategie repeat_if_real_earn s náhodnou volbou požadované částky 75 přibližně ve třetině případů nenaplnila očekávání dobrých výsledků.

Nejvyšší průměrné skóre her modelu boje o zdroje měly strategie passive, tft_&_random_forgive_aggressivity_33% a even_passive_odd_aggressive. Druhá zmíněná strategie byla rovněž třetí nejúspěšnější podle mediánového skóre. Na prvním místě podle sestupného řazení mediánového skóre je strategie tit_for_tat a na druhém unforgiving. Předpoklad obdobných výsledků strategií unforgiving, aggressive a tit_for_tat se naplnil pouze z části. Podobné průměrné i mediánové skóre měly unforgiving a tit_for_tat.

Cílem modelu dvourozměrného prostorového vězňova dilematu je porovnat úspěšnost strategií ve hře proti sousedním oblastem. Model byl rozlišen na evoluční a neevoluční variantu. Úspěšnost byla posouzena ve čtyřech dílčích oblastech na základě průměrných i mediánových hodnot. U evolučního modelu se jedná o velikost populace strategií v posledním kole hry a časy vyhynutí strategií. Kritérium kumulativního skóre strategií bylo vyhodnoceno u evolučního i neevolučního modelu. Obě varianty modelu byly porovnány z hlediska skóre hráčů na jedno kolo období existence, neboť strategie evolučního a neevolučního modelu neodehrají stejný počet kol. Pro evoluční model byly také zaznamenány průměrné a mediánové časy stabilizace populací strategií pro různé počty hráčů.

Nejvíce hráčů na konci kol průměrně měly strategie defect, even_cooperate_odd_defect a tft_&_random_defect_33%. Mediánové pořadí je defect, tft_&_random_defect_33% a dále následují pouze nulové mediánové populace strategií. U evolučního modelu bylo předpokládáno s ustálením struktury. Bylo zjištěno, že se zvětšujícím se počtem hráčů se zvyšuje i doba ustálení a vzhledem ke stanovenému počtu kol hry na 1 000 u vyšších počátečních populací ani nedošlo ke stabilizaci. Adaptace strategie tit_for_tat byla poměrně dobrá,

podle průměrné konečné populace je strategie čtvrtá. Úspěšnější byla modifikovaná varianta s náhodnou zradou přibližně ve třetině případů.

Úspěšnost strategií z hlediska časů vyhynutí byla obdobná jako u konečných populací, s tím rozdílem, že druhou nejúspěšnější strategií průměrně byla `tft_&_random_defect_33%` místo třetí `even_cooperate_odd_defect`. V kritériu kumulativního skóre byly evoluční a neevoluční modely rozdílné. Nejlepšími strategiemi podle průměru i mediánu u neevolučního modelu jsou `unforgiving`, `tit_for_tat` a `tft_&_rand_forgive_defect_33%` a evolučního modelu `defect`, `tft_&_random_defect_33%`, `even_cooperate_odd_defect`.

Předpokladem pro experimenty s pěti různými počátečními populacemi byla úspěšnost strategie spolupráce v menších komunitách. Úspěšnost se nepotvrdila v kritériích velikosti populace strategie, době výdrže a kumulativního skóre, avšak z hlediska skóre hráče na jedno kolo existence u evolučního modelu byla strategie `cooperate` nejúspěšnější. U neevolučního modelu byla strategie `cooperate` pátá podle sestupného řazení průměrných hodnot a na šestém místě podle mediánových.

Průměrné i mediánové pořadí strategií podle skóre hráče na jedno kolo existence u evolučního modelu je `cooperate`, `tft_&_rand_forgive_defect_33%`, `even_cooperate_odd_defect`, zatímco u neevolučního modelu `unforgiving`, `tit_for_tat` a na třetím místě podle průměru je `tft_&_rand_forgive_defect_33%` a podle mediánu `tit_for_two_tats`. Pro evoluční model byly uvedeny příklady vzniklých populačních struktur na konci hry (kapitola 3.3.3 a příloha 1).

Nakonec praktické části byla věnována pozornost diskuzi ohledně výsledků experimentů. V diskuzi byl rovněž věnován prostor možným příčinám výsledků a dopadům klíčových principů chování strategií. Byly také uvedeny další možnosti zkoumání. Modely a výsledky experimentů jsou součástí přílohy práce na CD.

5 Seznam použitých zdrojů

- [1] VON NEUMANN, J. – MORGENSTERN, O.: *Theory of Games and Economic Behavior* (60th Anniversary Commemorative edition) (eBook and Paperback). *Princeton University Press Home Page* [online]. Princeton University Press, 41 William Street, Princeton, New Jersey 08540 USA: Princeton University Press, 2015, 2015-11-13 [cit. 2015-11-14]. Dostupné z: <http://press.princeton.edu/titles/7802.html>.
- [2] CHVOJ, M. *Pokročilá teorie her ve světě kolem nás*. 1. vyd. Praha: Grada, 2013, s. 15–39, 227 s. ISBN 978-80-247-4620-3.
- [3] VOLEK, J. *Modelování a řešení rozhodovacích situací*. 1. vyd. Pardubice: Univerzita Pardubice, 2010, iii, s. 1–4, 106 s. ISBN 978-80-7395-311-9.
- [4] KACHRU, U. *Strategic Management: Concepts and Cases*. First Edition. New Delhi, India: Excel Books, 2005, s. 319, 819 s. ISBN 81-744-6424-7.
- [5] TUROCY, T. L. – VON STENGEL, B. *Game Theory* [online]. CDAM Research Report LSE-CDAM-2001-09, 2001, s. 2–6 [cit. 2015-11-05]. Dostupné z: <http://www.cdam.lse.ac.uk/Reports/Files/cdam-2001-09.pdf>.
- [6] PELIŠ, M. *Teorie her jako formální teorie racionálního rozhodování. Filozofická fakulta Univerzity Karlovy v Praze* [online]. Filozofická fakulta, Univerzita Karlova v Praze, 2015, s. 1–3 [cit. 2015-11-06]. Dostupné z: <http://web.ff.cuni.cz/~pelis/gt-pelis.pdf>.
- [7] SHOR, M. *Rationality. Game Theory .net - Resources for Learning and Teaching Strategy for Business and Life* [online]. Game Theory .net (c) Mike Shor 2001–2006, 2005-08-15 [cit. 2015-11-06]. Dostupné z: <http://www.gametheory.net/dictionary/Rationality.html>.
- [8] RAPOPORT, A. *N-person game theory: concepts and applications*. Dover ed. Mineola, New York: Dover Publications, 2001, 2013-06-17, s. 87, 331 s. ISBN 978-0-486-41455-3.
- [9] MCNULTY, D. *The Basics Of Game Theory. Investopedia - Educating the world about finance* [online]. (c) 2015, Investopedia, LLC. [cit. 2015-11-06]. Dostupné z: <http://www.investopedia.com/articles/financial-theory/08/game-theory-basics.asp>.
- [10] GALLEGO, L., et al. *Game theory I: Strategic form. Policonomics, economics made simple* [online]. (c) 2012 Policonomics [cit. 2015-11-06]. Dostupné z: <http://www.policonomics.com/lp-game-theory1-strategic-form/>.

- [11] GALLEGO, L., et al. Game theory I: Extensive form. *Policonomics, economics made simple* [online]. (c) 2012 Policonomics [cit. 2015-11-06]. Dostupné z: <http://www.policonomics.com/lp-game-theory1-extensive-form/>.
- [12] SETHI, R. Nash Equilibrium. In: DARITY, W. A. *International Encyclopedia of the Social Sciences* [online]. 2nd Edition. Detroit: Macmillan Reference USA, 2008, s. 540–542 [cit. 2015-11-19]. ISBN 9780028659671. Dostupné z: <http://www.columbia.edu/~rs328/NashEquilibrium.pdf>.
- [13] CARMICHAEL, F. *A guide to game theory*. New York: Financial Times Prentice Hall, 2005, xiii, s. 60–61, 286 s. ISBN 0-273-68496-5.
- [14] MYERSON, R. B. *Game Theory: Analysis of Conflict*. 1st edition. London, England: Harvard University Press, 2013, s. 325. ISBN 9780674728622.
- [15] WILENSKY, U. 1999. *NetLogo*. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
- [16] NITISHA. 5 Types of Games in Game Theory (With Diagram). *Economics Discussion - Discuss Anything About Economics* [online]. (c) 2015 EconomicsDiscussion.net [cit. 2015-09-12]. Dostupné z: <http://www.economicsdiscussion.net/game-theory/5-types-of-games-in-game-theory-with-diagram/3827>.
- [17] GUO, R. *Cross-Border Management: Theory, Method and Application*. New York: Springer, 2015, s. 149. ISBN 978-3-662-45156-4.
- [18] HUGHES, B. Simultaneous and sequential games. *Game Theory Strategies | Real life examples of game theory to improve your business and your life* [online]. (c) 2011-13 Barry Hughes, 2015 [cit. 2015-10-15]. Dostupné z: <http://www.gametheorystrategies.com/2011/09/15/simultaneous-and-sequential-games/>.
- [19] CHENG, Shih-Fen, et al. Notes on Equilibria in Symmetric Games. In: *Proceedings of Workshop on Game Theory and Decision Theory* [online]. University of Michigan, Michigan, USA, 2011, s. 1 [cit. 2015-10-15]. Dostupné z: <http://spider.sci.brooklyn.cuny.edu/~parsons/events/gtdt/gtdt04/reeves.pdf>.
- [20] GALLEGO, L., et al. Game theory II: Dominant strategies. *Policonomics, economics made simple* [online]. (c) 2012 Policonomics [cit. 2015-10-16]. Dostupné z: <http://www.policonomics.com/lp-game-theory2-dominant-strategy/>.
- [21] DIXIT, A. – Nalebuff, B. Game Theory. *Library of Economics and Liberty* [online]. (c) 2008 Liberty Fund, Inc., 2008 [cit. 2015-10-16]. Dostupné z: <http://www.econlib.org/library/Enc/GameTheory.html>.

- [22] HEAP, S. H. – VAROUFAKIS, Y. *Game theory: a critical text*. 2nd ed., rev. ed. New York: Routledge, 2004, s. 44, 369 s. ISBN 0-415-25094-3.
- [23] MARWALA, T. *Multi-agent Approaches to Economic Modeling: Game Theory, Ensembles, Evolution and the Stock Market*. London: Springer, 2013, s. 195–198. DOI: 10.1007/978-1-4471-5010-7_11.
- [24] BANDINI, S. – MANZONI, S. – VIZZARI, G. *Agent Based Modeling and Simulation. Computational Complexity*. New York, NY: Springer New York, 2012, s. 105–117. DOI: 10.1007/978-1-4614-1800-9_7. ISBN 978-1-4614-1799-6.
- [25] CURRAN, K. *Pervasive and ubiquitous technology innovations for ambient intelligence environments*. Hershey, PA: Information Science Reference, 2013, xxv, s. 173, 290 s. ISBN 978-146-6620-438.
- [26] HADZIC, M., et al. *Ontology-based multi-agent systems*. Online-Ausg. Berlin: Springer, 2009, s. 16–17. ISBN 978-3-642-01904-3.
- [27] NORTH, M. J. – MACAL, CH. M. *Agent Based Modeling and Computer Languages. Computational Complexity*. New York, NY: Springer New York, 2012, s. 60. DOI: 10.1007/978-1-4614-1800-9_4. ISBN 978-1-4614-1799-6.
- [28] ANTUNES, L. – TAKADAMA, K. (eds) – ANTUNES, L. *Multi-agent-based simulation VII international workshop, MABS 2006, Hakodate, Japan, May 8, 2006: revised and invited papers*. Berlin: Springer, 2007, x, s. 16–17, 187 s. Lecture notes in computer science, 4442. ISBN 978-354-0765-394.
- [29] WILENSKY, U. – RAND, W. *An introduction to agent-based modeling: Modeling natural, social, and engineered complex systems with NetLogo*. Cambridge, Massachusetts: The MIT Press, 2015, xxii, s. 258–260, 482 s. ISBN 978-026-2731-898.
- [30] KUDENKO, D. – KAZAKOV, D. – ALONSO, E. *Adaptive agents and multi-agent systems II adaptation and multi-agent learning*. Online-Ausg. Berlin: Springer-Verlag, 2005, s. 96. ISBN 978-354-0322-740.
- [31] ALLEN, T. T. *Agents and New Directions. Introduction to Discrete Event Simulation and Agent-based Modeling*. London: Springer London, 2011, s. 179–181. DOI: 10.1007/978-0-85729-139-4_12. ISBN 978-0-85729-138-7.
- [32] BRADFORD, A. *Deductive Reasoning vs. Inductive Reasoning. Live Science: Scientific News, Articles and Current Events* [online]. 2015 [cit. 2015-11-28]. Dostupné z: <http://www.livescience.com/21569-deduction-vs-induction.html>.
- [33] GALLEGO, L., et al. *Prisoner's dilemma. Policonomics, economics made simple* [online]. (c) 2012 Policonomics [cit. 2015-11-14]. Dostupné z: <http://www.policonomics.com/prisoners-dilemma/>.

- [34] DIXIT, A. – NALEBUFF, B. Prisoners' Dilemma. *Library of Economics and Liberty* [online]. (c) 2008 Liberty Fund, Inc., 2008 [cit. 2015-11-14]. Dostupné z: <http://www.econlib.org/library/Enc/PrisonersDilemma.html>.
- [35] CEJTHAMR, V. – DĚDINA, J. *Management a organizační chování. 2.*, aktualiz. a rozš. vyd. Praha: Grada, 2010, s. 247, 344 s. Expert (Grada). ISBN 978-80-247-3348-7.
- [36] KUHN, S. Prisoner's Dilemma. *Stanford Encyclopedia of Philosophy* [online]. (c) 2014 by Steven Kuhn, 1997, revision Fri Aug 29, 2014 [cit. 2015-11-15]. Dostupné z: <http://plato.stanford.edu/entries/prisoner-dilemma/>.
- [37] DLOUHÝ, M. – FIALA, P. *Úvod do teorie her. 2.*, přeprac. vyd. Praha: Oeconomica, 2009, s. 47–48, 119 s. ISBN 978-80-245-1609-7.
- [38] SLANTCHEV, B. L. *Game Theory: Repeated Games* [online]. Department of Political Science, University of California, San Diego, 2004, s. 6–7 [cit. 2016-01-06]. Dostupné z: <http://users.auth.gr/~kehagiat/Research/GameTheory/02Courses/Course3/07-repeated-games.pdf>.
- [39] NAMATAME, A. *Agent based modelling and network dynamics*. New York, NY: Oxford University Press, 2015, s. 22, 320 s. ISBN 9780198708285.
- [40] IBA, H. *Agent-Based Modeling and Simulation with Swarm*. Hoboken: CRC Press, 2013, s. 89, 317 s. ISBN 9781466562400.
- [41] ONAL, E., et al. Decision-making in Abstract Trust Games: A User Interface Perspective. In: *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)* [online]. San Antonio: IEEE., 2014, s. 21–27 [cit. 2016-04-11]. Dostupné z: <http://www.cs.ucsb.edu/~holl/pubs/Onal-2014-CogSIMA.pdf>.
- [42] ALBRECHTOVÁ, J. Cestopis Izrael aneb na zkušenou do Kibucu. In: *PlanetaCestovani.cz - odborník (nejen) na levné zájezdy ✈* [online]. 2016 (c) PlanetaCestovani.cz, 2015 [cit. 2016-01-20]. Dostupné z: <http://planetacestovani.cz/cestopis-izrael-kibucu/>.
- [43] SOSIS, R. The Adaptive Value of Religious Ritual. *American Scientist* [online]. č. 3-4/2004, s. 166–172 [cit. 2016-01-20]. Dostupné z: <http://people.uncw.edu/bruce/hon%20210/pdfs/Religious%20ritual%20%20Sosis%202004.pdf>.
- [44] BINMORE, K. *Teorie her: ...a jak může změnit váš život. 1. vyd. v českém jazyce*. Praha: Dokořán, 2014, s. 159–162, 230 s. Edice Aliter (Argo: Dokořán). ISBN 978-80-7363-549-7.

- [45] DOEBELI, M. – HAUERT, Ch. Models of cooperation based on the Prisoner's Dilemma and the Snowdrift game. *Ecology Letters*. 2005, 8(7), s. 748-766. DOI: 10.1111/j.1461-0248.2005.00773.x. ISSN 1461023x.
- [46] WILENSKY, U. NetLogo 5.3.1 User Manual: NetLogo Dictionary. NetLogo Home Page [online]. (c) 1999-2016 Uri Wilensky, 2016 [cit. 2016-02-03]. Dostupné z: <https://ccl.northwestern.edu/netlogo/docs/dictionary.html>.

6 Přílohy

- 1) Vzorové struktury populací strategií modelu SPD
- 2) Obsah přiloženého CD
- 3) Zadání práce

1) Vzorové struktury populací strategií modelu SPD



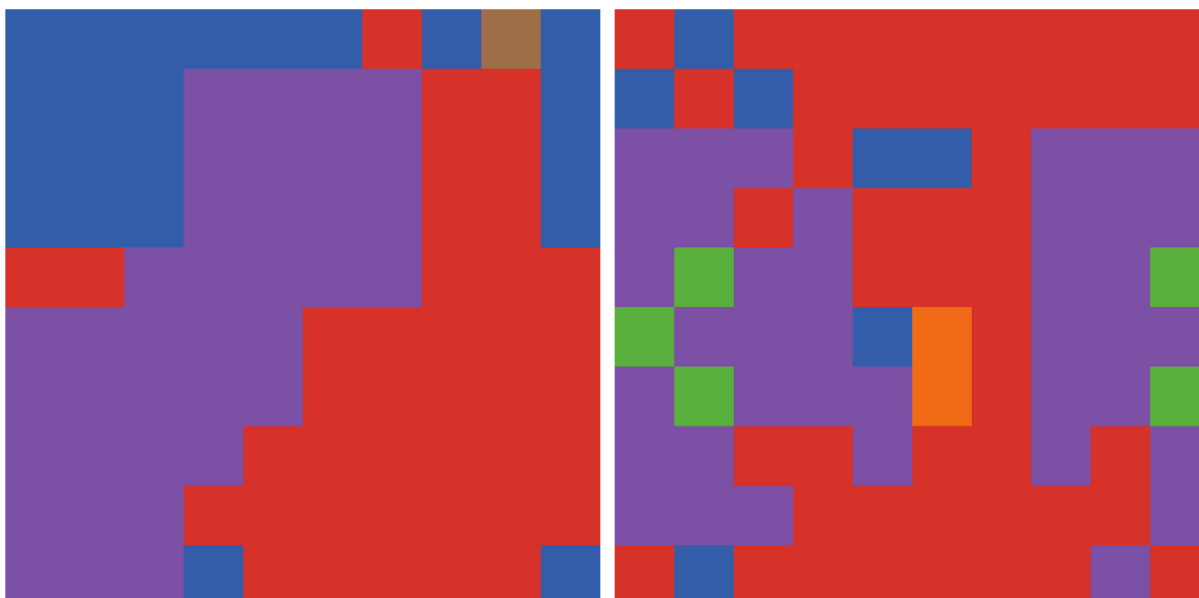
Obrázek 5: Barevná legenda strategií

Zdroj: vlastní zpracování



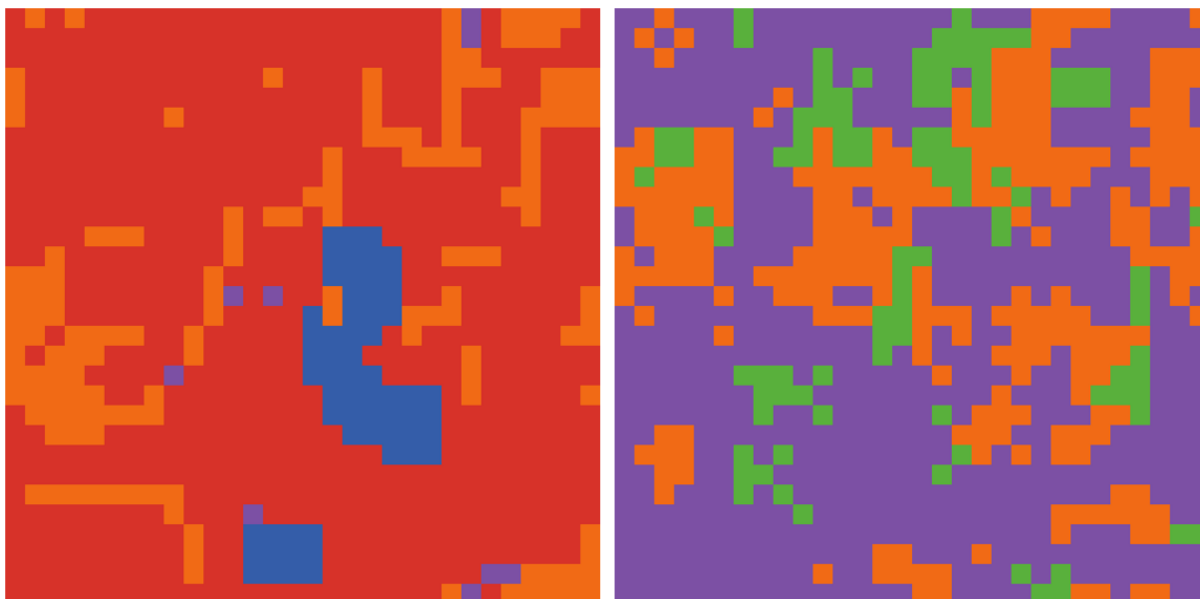
Obrázek 6: Model SPD, 20 hráčů – stabilizované struktury (po 3 a 7 kolech)

Zdroj: vlastní zpracování



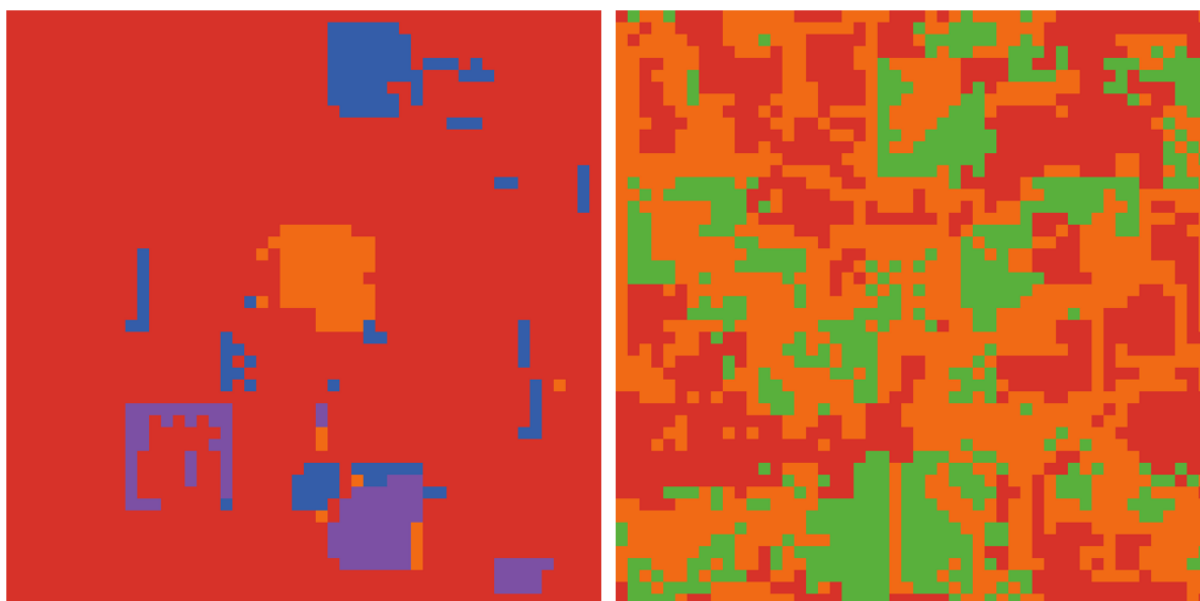
Obrázek 7: Model SPD, 100 hráčů – stabilizovaná a nestabilizovaná struktura (14 a 1 000 kol)

Zdroj: vlastní zpracování



Obrázek 8: Model SPD, 900 hráčů – stabilizovaná a nestabilizovaná struktura (41 a 1 000 kol)

Zdroj: vlastní zpracování



Obrázek 9: Model SPD, 2 500 hráčů – stabilizovaná a nestabilizovaná struktura (16 a 1 000 kol)

Zdroj: vlastní zpracování

2) Obsah přiloženého CD

Diplomová práce:

- Spejchal Jiří – Varianty vězňova dilematu – Modely v NetLogu.pdf

Modely alternativ vězňova dilematu v NetLogu:

- Sosis game.nlogo
- Hawk Dove game modified.nlogo
- Spatial prisoners dilemma.nlogo

Soubory Excelu – výsledky experimentů:

- Sosis game experiment.xlsx
- Hawk Dove game modified experiment.xlsx
- SPD experiment evolutionary.xlsx
- SPD experiment non-evolutionary.xlsx
- Random discrete NetLogo generator test.xlsx

3) Zadání práce

Univerzita Hradec Králové
Fakulta informatiky a managementu
Akademický rok: 2015/2016

Studijní program: Systémové inženýrství a informatika
Forma: Prezenční
Obor/komb.: Informační management (im2-p)

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Spejchal Jiří	Růžová 346, Opatovice nad Labem	I1448

TÉMA ČESKY:

Varianty vězňova dilematu - modely v NetLogu

TÉMA ANGLICKY:

Versions of the prisoner's dilemma - models in NetLogo

VEDOUcí PRÁCE:

doc. RNDr. Kamila Štekerová, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce:

Cílem této diplomové práce je popsat varianty vězňova dilematu, vytvořit odpovídající modely v prostředí NetLogo a použít tyto modely k experimentování se strategiemi.

Osnova:

1. Úvod
2. Teoretická část
 - 2.1. Zkoumání her pomocí agentových modelů
 - 2.1.1. Vězňovo dilema, jeho varianty a aplikace
3. Praktická část
 - 3.1. Návrh modelů
 - 3.2. Implementace
 - 3.3. Experimenty
 - 3.4. Diskuze
4. Závěr
5. Seznam použitých zdrojů

SEZNAM DOPORUČENÉ LITERATURY:

- [1] CHVOJ, Martin. Pokročilá teorie her ve světě kolem nás. 1. vyd. Praha: Grada, 2013, 227 s. ISBN 978-80-247-4620-3.
- [2] VOLEK, Josef. Modelování a řešení rozhodovacích situací. 1. vyd. Pardubice: Univerzita Pardubice, 2010, iii, 106 s. ISBN 978-80-7395-311-9.
- [3] MICHAEL DOEBELI a CHRISTOPH HAUERT. Models of cooperation based on the Prisoner's Dilemma and the Snowdrift game. Ecology Letters [online]. 2005, 8(7): 748-766. DOI: 10.1111/j.1461-0248.2005.00773.x. Dostupné z: <http://onlinelibrary.wiley.com/doi/10.1111/j.1461-0248.2005.00773.x/full>.
- [4] HOUSER, Pavel. Variace na Vězňovo dilema: Sosisova hra v kibucech. Science World.cz | Novinky ze světa vědy a techniky: technologie, neživá příroda, člověk, biologie [online]. 2014. Dostupné z: http://www.scienceworld.cz/neziva-priroda/dalsi-variace-na-veznovo-dilema-sosisova-hra-v-izraelskych-kibucech-377/?switch_theme=desktop.
- [5] Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.