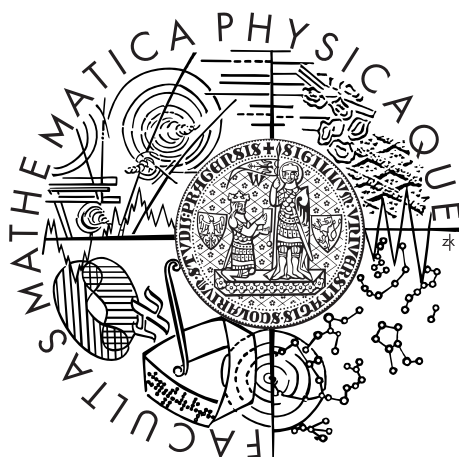


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Tobiáš Hudec

Vytváranie obchodných stratégií pre dvojité aukcie pomocou genetického programovania

Katedra teoretické informatiky a matematické logiky

Vedoucí bakalářské práce: Mgr. Martin Pilát, Ph.D.

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2014

Na tomto mieste by som chcel poďakovať svojmu vedúcemu bakalárskej práce, pánovi Mgr. Martinovi Pilátovi, Ph.D., za jeho inšpirujúce podnety a odborné vedenie tejto práce.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Název práce: Vytváranie obchodných stratégií pre dvojité aukcie pomocou genetického programovania

Autor: Tobiáš Hudec

Katedra: Katedra teoretickej informatiky a matematickej logiky

Vedoucí bakalárskej práce: Mgr. Martin Pilát, Ph.D., Katedra teoretickej informatiky a matematickej logiky

Abstrakt: Výpočtová ekonómia skúma ekonomické deje na základe počítačových simulácií. Pre tieto simulácie potrebuje počítačové programy, ktoré simulujú správanie ľudí. Simulovať správanie ľudí je možné použitím umelej inteligencie. Preskúmame spôsoby použitia formy umelej inteligencie nazývanej genetické programovanie v systéme obchodovania nazývanom dvojitá aukcia. Navrhujeme úpravu genetického programovania s cieľom zlepšiť jeho fungovanie v dvojitej aukcii. Experimentálne overíme, či má obchodovanie riadené genetickým programovaním vlastnosti, ktoré by sme čakali od obchodovania ľudí. Ukáže sa, že genetické programovanie dobre aproximuje správanie ľudí, a teda je vhodným nástrojom pre výpočtovú ekonómiu.

Kľúčová slova: dvojitá aukcia, multiagentný systém, genetické programovanie, obchodné stratégie

Title: Creating bidding strategies for double auction markets using the genetic programming

Author: Tobiáš Hudec

Department: Department of Theoretical Computer Science and Mathematical Logic

Supervisor: Mgr. Martin Pilát, Ph.D., Department of Theoretical Computer Science and Mathematical Logic

Abstract: Computational economics investigates economic phenomena by computer simulations. For these simulations it needs computer programs which can simulate behaviour of people. It is possible to use artificial intelligence to simulate behaviour of people. We investigate ways of using a type of artificial intelligence called genetic programming in a trading system called double auction. We devise a modification of genetic programming in order to improve its performance in double auction. We experimentally verify whether trading controlled by genetic programming has properties that we would expect from trading conducted by people. It turns out that genetic programming approximates behaviour of people well, and is therefore a feasible tool for computational economics.

Keywords: double auction, multi-agent system, genetic programming, bidding strategy

Obsah

Úvod	1
1 Súvisiace práce	3
2 Výpočtová ekonómia a dvojitá aukcia	5
2.1 Dvojitá aukcia	6
2.2 Simulácia dvojitej aukcie	7
2.3 Meranie výkonu dvojitej aukcie	9
2.4 Heuristiky pre dvojitú aukciu	12
3 Evolučný algoritmus a genetické programovanie	15
3.1 Evolučný algoritmus	15
3.2 Genetické operátory	17
3.3 Selekcia	18
3.4 Elitizmus	19
3.5 Typy evolučných algoritmov	19
3.6 Použitie genetického programovania v dvojitej aukcii	21
4 Bloky	23
4.1 Blokované genetické operátory	24
5 Experimenty	25
5.1 Parametre experimentov	26
5.2 Ziskovosť najlepšieho agenta	30
5.3 Adaptácia na pevné prostredie	35
5.4 Overenie vlastností koevolučného modelu	40
Záver	44
Literatúra	45

Úvod

Experimentálna ekonómia je časť ekonómie, ktorá sa snaží získavať poznatky o ekonomických dejoch na základe experimentov, ktoré tieto deje simulujú. V experimentoch vystupujú *agenti*, ktorí reprezentujú ekonomické subjekty. Príkladom experimentu môže byť simulované obchodovanie, v ktorom agenti zastávajú úlohu kupujúcich a predávajúcich. V našej práci sme sa zamerali na simulované obchodovanie na trhu, ktorý sa nazýva dvojité aukcia [16].

Agenti v ekonomických experimentoch môžu byť buď ľudia alebo počítačové programy. Experimenty s ľuďmi sú časovo aj finančne náročné, preto je vhodné nahradiť ich experimentami s počítačovými programami. Takýmito experimentami sa zaoberá výpočtová ekonómia. Výpočtová ekonómia teda potrebuje počítačové programy, ktoré by boli schopné verne reprezentovať ľudských agentov.

Existujú dva postupy, ako vytvárať počítačových agentov pre ekonomické simulácie. Prvým je písanie heuristik. Heuristika je počítačový program, ktorý obchoduje na základe pevne danej stratégie stanovenej programátorom. Táto stratégia môže byť odvodená napríklad zo skúseností z reálneho obchodovania alebo z teoretických znalostí ekonómie. Druhým postupom je použitie umelej inteligencie. V tomto prípade počítač sám vyvíja stratégie pre obchodovanie. Hlavnou výhodou umelej inteligencie je jej schopnosť adaptovať sa. Agent riadený umelou inteligenciou by mal byť schopný odhaliť slabiny protistrany a využiť ich vo svoj prospech. Naopak, ak protistrana odhalí slabinu agenta riadeného umelou inteligenciou, tento agent by mal upraviť svoju stratégiu a slabinu odstrániť. Takéto správanie by sme čakali od ľudí, a preto je vhodné, aby ho mali aj počítačovní agenti.

V tejto práci sa zameriame na použitie jednej formy umelej inteligencie nazývanej genetické programovanie. Navrhujeme úpravu genetického programovania, ktorá by mohla pomôcť pri vývoji stratégií pre obchodovanie. Preskúmame dva spôsoby použitia genetického programovania v dvojitej aukcii. Celkom teda získame štyri konfigurácie genetického programovania (pôvodné aj upravené genetické programovanie použijeme oboma spôsobmi).

Agentov riadených genetickým programovaním otestujeme v simulovanej dvojitej aukcii. Cieľom je zistiť, či agenti riadení genetickým programovaním majú vlastnosti, ktoré by sme očakávali od ľudí, a teda či je genetické programovanie vhodným nástrojom pre výpočtovú ekonómiu. Porovnáme preto zisky agentov riadených genetickým programovaním so ziskmi heuristik. Takisto porovnáme výsledky simulácií so známymi teoretickými vlastnosťami dvojitej aukcie, ako aj so známymi výsledkami experimentov s ľuďmi. Ďalším cieľom je porovnať jednotlivé konfigurácie genetického programovania medzi sebou a určiť tak, či je niektorá z nich lepšia než ostatné.

Štruktúra práce je nasledovná. V prvej kapitole podáme základný prehľad

o prácach, ktoré boli v tejto oblasti publikované.

V druhej kapitole predstavíme podrobnejšie výpočtovú ekonómiu. Takisto detailne popíšeme fungovanie dvojitej aukcie a odvodíme niektoré jej teoretické vlastnosti, ako je zákon jednej ceny a konvergencia cien k rovnovážnej cene. Ďalej definujeme veličiny, ktoré charakterizujú obchodovanie v dvojitej aukcii. Nakoniec uvedieme heuristiky pre dvojitú aukciu.

V tretej kapitole popíšeme evolučný algoritmus ako algoritmus na riešenie optimalizačných problémov. Potom popíšeme genetický algoritmus a genetické programovanie ako špeciálny prípad evolučného algoritmu. V závere kapitoly ukážeme, ako sa dá genetické programovanie použiť v dvojitej aukcii.

Vo štvrtej kapitole navrhujeme úpravu genetického programovania, ktorá by mala pomôcť pri jeho použití v dvojitej aukcii. V tejto kapitole popíšeme, ako je možné genetickému programu odovzdať informáciu, ktorú získame analýzou heuristík.

V piatej kapitole pojednáme o experimentoch, ktoré sme vykonali. Popíšeme tri experimenty: priame porovnanie genetického programovania a heuristík, overenie schopnosti genetického programovania adaptovať sa a overenie, či má trh s agentmi riadenými genetickým programovaním vlastnosti, ktoré predpovedá ekonomická teória. Nakoniec naše zistenia zhrnieme v závere.

Kapitola 1

Súvisiace práce

V tejto kapitole podáme základný prehľad prác, ktoré pojednávajú o výpočtovej ekonómii a o použití umelej inteligencie v ekonomických simuláciách. Dobrým prehľadovým článkom v tejto oblasti je [6].

Heuristiky pre riadenie agentov v ekonomických simuláciách sa začali rozvíjať predovšetkým pomocou ekonomických turnajov. Ekonomický turnaj je súťaž, v ktorej heuristiky spolu interagujú v simulovanom ekonomickom procese s cieľom získať čo najväčší zisk. Tento simulovaný proces je zvyčajne nejaká forma obchodovania, ale môže ísť aj o iný proces. Jedným z prvých turnajov bol Axelrodov turnaj iterovanej väzňovej dilemy [5].

Väzňova dilema je hra dvoch hráčov. Každý hráč sa nezávisle na druhom rozhodne, či bude s druhým spolupracovať, alebo ho zradí. Na základe ich rozhodnutí je hráčom pridelený zisk. Najväčší zisk dostane hráč v prípade, že protihráča zradí, ale protihráč sa rozhodne spolupracovať. Menší zisk dosiahne, ak sa obaja rozhodnú spolupracovať. Ešte menší zisk dosiahne v prípade, že sa obaja hráči rozhodnú zradiť, a najmenší zisk dosiahne, ak sa rozhodne spolupracovať, ale protihráč sa rozhodne zradiť. Iterovaná väzňova dilema je hra, v ktorej hráči spolu opakovane hrajú väzňovu dilemu, pričom vedia, ako sa protihráč rozhodol v predchádzajúcich kolách. V tejto hre je nutné nájsť rovnováhu medzi prílišnou sebeckosťou, ktorá môže viesť k suboptimálnym výsledkom, a prílišnou ochotou spolupracovať, ktorá môže byť zneužitá protihráčom. Najlepšia heuristika Axelrodovho turnaja bola prekvapivo jednoduchá: v prvom kole spolupracovala, a v každom ďalšom kole urobila to, čo urobil protihráč v predchádzajúcom kole.

Ďalším z turnajov bol turnaj dvojitej aukcie, ktorý sa uskutočnil v Inštitúte Santa Fe [18]. Tento turnaj používal model trhu veľmi podobný tomu, ktorý používame v našej práci. Preto aj v našej práci používame heuristiky z tohto turnaja. Je zaujímavé podotknúť, že najlepšie heuristiky boli relatívne jednoduché programy, a to aj napriek tomu, že turnaja sa zúčastnili aj komplikované programy, ktoré využívali napríklad neurónové siete alebo štatistické predikcie. Tento výsledok je podobný tomu, ktorý získal Axelrod.

Neskoršie turnaje sa zameriavali na komplikovanejšie ekonomické simulácie. Napríklad [19] popisuje turnaj, v ktorom heuristiky hrajú úlohu cestovných agentov, ktorý sa snažia naplánovať čo najlacnejší a najuspokojivejší výlet pre svojich klientov. Ďalším príkladom je simulácia popísaná v [3], v ktorej heuristiky hrajú úlohu výrobcov, ktorý musia nakupovať rôzne súčiastky pre svoje výrobky na rôznych trhoch, a na ďalšom trhu svoje výrobky predávajú.

Jedným z prvých použití umelej inteligencie v ekonomických simuláciách je použitie genetických algoritmov v iterovanej väzňovej dileme [4]. V tejto aplikácii sa každý agent rozhodne spolupracovať alebo zradiť na základe histórie posledných troch kôl. Keďže existuje $2^6 = 64$ možných spôsobov, ako mohli prebehnúť posledné tri kolá (oba hráči mali v každom kole dve možnosti), stratégie agentov sú reprezentované ako 64-bitové reťazce.

Genetické programovanie v dvojitej aukcii prvýkrát použili Andrews a Prager [2]. V ich simuláciách je jeden agent vyvíjaný genetickým programovaním, pričom všetci ostatní agenti sú heuristiky získané z ekonomických turnajov. Dawid [9] popisuje použitie genetických algoritmov v dvojitej aukcii. V jeho experimentoch je každý agent reprezentovaný binárnym reťazcom, ktorý kóduje cenu, za ktorú chce obchodovať. Kupujúci resp. predávajúci sú chápaní ako populácia agentov, na ktorú je aplikovaný genetický algoritmus popísaný v kapitole 3. Nakoniec Chen [7] navrhuje použitie viacpopulačného genetického programovania pre dvojité aukcie. V tomto modeli sú všetci obchodujúci vyvíjaní genetickým programovaním. Tento model je podrobnejšie popísaný na konci kapitoly 3.

Okrem použitia genetického programovania a genetických algoritmov je tiež možné použiť iné formy umelej inteligencie. Napríklad Roth a Erev [17] vyvinuli algoritmus založený na učení so spätnou väzbou (reinforcement learning). Tento algoritmus bol použitý v [15] na simuláciu obchodovania s elektrickou energiou.

Kapitola 2

Výpočtová ekonómia a dvojitá aukcia

Výpočtová ekonómia je časť ekonómie, ktorá sa zaoberá počítačovými simuláciami ekonomických procesov. V týchto simuláciách hrajú kľúčovú úlohu *agenti*, ktorí môžu reprezentovať jednotlivcov, firmy, sociálne skupiny a pod. Ekonomický proces je simulovaný ako dynamický systém interagujúcich agentov. Príkladom ekonomického procesu je obchodovanie, kde agenti zastávajú úlohu kupujúcich a predávajúcich.

Pre niektoré ekonomické procesy existujú teoretické modely, ktoré predpovedajú, ako sa budú tieto procesy vyvíjať v čase. V takom prípade sa môže výpočtová ekonómia uplatniť na experimentálne overenie teoretických modelov. O mnoho dôležitejšie uplatnenie však má výpočtová ekonómia v prípadoch, keď sú ekonomické procesy príliš zložité, aby ich bolo možné popísať teoretickým modelom. V takom prípade sa môžu simuláciami zisťovať vlastnosti týchto procesov.

Ako príklad použitia výpočtovej ekonómie môžeme uviesť experiment popísaný v [15]. Autori tohto experimentu sa zaoberali otázkou deregulácie trhu s elektrickou energiou. Na jednej strane by deregulácia mohla priniesť väčšiu konkurenciu a tým zníženie cien. Na druhej strane by sa mohlo stať, že veľký výrobcovia elektrickej energie by mohli byť schopní zneužiť svoje dominantné postavenie a umelo zvyšovať ceny. Obchodovanie na trhu s elektrickou energiou je komplikované, pretože je obmedzené kapacitou prenosovej sústavy. To je jeden z dôvodov, pre ktorý je teoretická analýza takéhoto trhu náročná, a je vhodné radšej použiť simuláciu.

Pre použitie výpočtovej ekonómie je nutné, aby agenti čo najvernejšie reprezentovali reálnych obchodujúcich, ktorých majú simulovať. Základnou vlastnosťou obchodujúcich je, že sa snažia maximalizovať svoj vlastný zisk, a to bez ohľadu na ostatných. Túto vlastnosť dobre reprezentujú ľuďmi naprogramovaní agenti (heuristický agenti). Takýto agenti sú programy, ktoré obchodujú podľa pevne danej stratégie. Problémom týchto agentov je to, že sa nedokážu adaptovať. Reálni obchodujúci sú totižto schopní odhaliť slabiny svojich súperov a využiť ich vo svoj prospech.

Pre prekonanie tohto problému je možné použiť autonómnych agentov riadených umelou inteligenciou. Takýto agenti sú schopní vyvíjať svoje stratégie a tým sa prispôbiť okolnostiam, a preto sú realistickejšou reprezentáciou reálnych obchodujúcich. Ako autonómni agenti sa zvyčajne používajú agenti vyvíjaní ge-

netickým programovaním. Rozlišujeme dva modely použitia autonómnych agentov [6]. Prvým je evolučný model, v ktorom jedna strana (napríklad kupujúci) používa autonómnych agentov, zatiaľ čo druhá strana (predávajúci) používa heuristických agentov. V takomto modeli sa autonómni agenti snažia adaptovať na pevné prostredie, ktoré predstavujú heuristickí agenti. Druhým modelom je koevolučný model, v ktorom obidve strany používajú autonómnych agentov. Tento model reprezentuje reálny trh, na ktorom sa kupujúci aj predávajúci neustále vyvíjajú a adaptujú.

V tejto práci sa zameriame na model trhu nazývaný *dvojitá aukcia*. Tento model je z teoretického hľadiska relatívne jednoduchý a preto sú známe teoretické predpovede o jeho správaní. Navyše sa tento systém používa pri reálnom obchodovaní¹, takže sú k dispozícii empirické údaje, ktoré potvrdzujú teoretické predpovede. Preskúmame správanie agentov vyvíjaných genetickým programovaním v tomto modeli a porovnáme ho s teoretickými predpoveďami. Na základe toho budeme schopní rozhodnúť, či sú takéto agenti dobrou reprezentáciou skutočných obchodujúcich.

2.1 Dvojitá aukcia

Dvojitá aukcia je systém obchodovania *kupujúcich* a *predávajúcich* s istou komoditou. Kupujúci *ponúkajú* ceny (*bid*), za ktoré chcú kupovať, a tiež množstvá, ktoré chcú kúpiť. Predávajúci *žiadajú* ceny (*ask*), za ktoré chcú predávať, a tiež množstvá, ktoré chcú predať. Následne *aukcionár* vyberie dvojicu (kupujúci, predávajúci), ktorá bude spolu obchodovať. Aukcionár tiež stanoví cenu a množstvo, ktoré bude obchodované. V praxi je aukcionár zvyčajne počítačový program, ktorý sa riadi pevnými, verejne známymi pravidlami.

Pre jednoduchosť budeme predpokladať, že komodita je diskrétna, t.j. existuje nedeliteľná jednotka množstva a je možné obchodovať len s celočíselným počtom jednotiek. Ďalej budeme predpokladať, že agenti budú môcť naraz kúpiť resp. predať iba jednu jednotku komodity. V prípade, že agent bude chcieť obchodovať s n jednotkami, uskutoční n samostatných obchodov. Tieto predpoklady zjednodušia úlohu aukcionára, pretože nebude musieť stanovovať množstvo obchodu.

Pri dvojitej aukcii má každý kupujúci zaručené, že nebude kupovať za vyššiu cenu, než ktorú ponúkol. Podobne má každý predávajúci zaručené, že nebude predávať za nižšiu cenu, než žiadal. Úlohou aukcionára je spárovať ceny stanovené agentmi tak, aby boli tieto podmienky splnené. V prípade, že najvyššia ponuka je menšia než najnižšia požiadavka, je zrejmé, že žiaden obchod nemôže prebehnúť bez porušenia horeuvedených podmienok. V opačnom prípade existuje kupujúci a predávajúci tak, že cena k ponúkaná kupujúcim je vyššia alebo rovná cene p požadovanej predávajúcim. Táto dvojica môže obchodovať za ľubovoľnú cenu v intervale $[p, k]$. Takáto dvojica však nemusí byť určená jednoznačne. Je úlohou aukcionára stanoviť dvojicu, ktorá bude obchodovať, a tiež stanoviť cenu, za ktorú bude obchodovať. Zvyčajne aukcionár vyberie kupujúceho s najvyššou ponukou a predávajúceho s najnižšou požiadavkou. Čo sa týka zvolenia ceny, sú dve základné

¹Pri reálnom obchodovaní sa zvyčajne používajú komplikovanejšie pravidlá než tie, ktoré používame v simuláciách. Hlavná myšlienka a vlastnosti modelu však zostávajú zachované.

možnosti: zvoliť cenu náhodne v intervale $[p, k]$, alebo zvoliť cenu ako aritmetický priemer p a k .

Dvojitá aukcia môže prebiehať buď v spojitom čase alebo v diskretnom čase. V prípade spojitého času agenti oznamujú svoje ceny v ľubovoľných časových okamihoch a aukcionár rozhodne o obchode hneď, ako najvyššia ponuka na nákup prevýši najnižšiu požiadavku na predaj. Pri tomto systéme obchodovania je nutné zohľadniť časovú náročnosť použitých algoritmov, pretože počas výpočtu sa situácia na trhu môže zmeniť. V prípade, že sa agent nachádza na odlišnom mieste než aukcionár, je tiež nutné zohľadniť časové zdržanie, ku ktorému dôjde pri komunikácii. Tento systém tak do určitej miery zvyhodňuje rýchlych agentov, ktorí sú schopní okamžite reagovať na situáciu na trhu. Tento systém sa zvyčajne používa pri reálnom obchodovaní na burzách.

V prípade diskretného času je obchodovanie rozdelené do *krokov*. V každom kroku agenti oznámia svoje ceny (prípadne sa rozhodnú, že v danom kroku nechcú obchodovať) a aukcionár následne rozhodne o obchode. Tento systém nijako nezvyhodňuje rýchlych agentov. Taktiež v prípade, že sa agenti nachádzajú na inom mieste než aukcionár, nemajú agenti s lepším spojením výhodu oproti ostatným. Výsledok obchodovania závisí len na kvalite použitých algoritmov, nie na ich rýchlosti. Takisto je tento systém jednoduchší na implementáciu. Z týchto dôvodov sa tento systém zvyčajne používa pri simuláciách.

2.2 Simulácia dvojitej aukcie

Pri simulácii dvojitej aukcie sa predpokladá, že agenti obchodujú s fiktívnou diskretnou komoditou zvanou *token*. Obchoduje sa v diskretnom čase, teda obchodovanie je rozdelené do krokov. Budeme predpokladať, že každý agent je buď kupujúci alebo predávajúci, ale žiaden agent nezastáva obidve úlohy. V praxi je síce možné, aby agent kúpil komoditu, ktorú neskôr predá, ale takúto situáciu pre jednoduchosť nebudeme uvažovať.

Predstavme si, že predávajúci sú výrobcovia tokenov. S výrobou sú spojené určité náklady. Je však nereálne predpokladať, že tieto náklady budú konštantné. Poľnohospodár môže vlastniť niekoľko polí s rôznou kvalitou pôdy a náklady na pestovanie plodín na menej kvalitnej pôde budú zrejme vyššie. Podobne v továrni sa môže nachádzať viac výrobných liniek, pričom moderné linky budú vyrábať s nižšími nákladmi než zastaralé. Aby sme takúto situáciu reprezentovali v simulácii, priradíme predávajúcemu *rezervačné ceny*. Počet rezervačných cien predstavuje počet tokenov, ktoré je predávajúci schopný vyrobiť, pričom každá rezervačná cena predstavuje náklady na výrobu daného tokenu. Ak predávajúci predá i -ty token za cenu p_i , pričom jeho rezervačná cena pre daný token je r_i , dosiahne tým zisk $z_i = p_i - r_i$. V prípade, že sa predávajúcemu nepodarí i -ty token predáť, zisk pre tento token definujeme ako $z_i = 0$. Toto je možné interpretovať napríklad tak, že výrobcovia predajú svoj výrobok ešte predtým, než ho vyrobia, a následne vyrobia a dodajú kupujúcim iba toľko výrobkov, koľko predali. Celkový zisk predávajúceho potom definujeme ako $Z = \sum_{i=0}^M z_i$, kde M je počet rezervačných cien predávajúceho. Je rozumné predpokladať, že ak výrobcovia predajú menej než je ich maximálna kapacita výroby, budú prednostne využívať lacnejšie spôsoby výroby. Tento fakt sa v simulácii prejaví tak, že predávajúci budú predávať tokeny v poradí podľa rezervačných cien od najmenej

po najväčšiu.

Takisto kupujúcim priradíme pri simulácii rezervačné ceny. Ich interpretácia je o niečo náročnejšia. Môžeme si napríklad predstavovať, že dvojitá aukcia je veľkoobchod, kde kupujúci nakúpia tovar, ktorý ďalej predávajú koncovým zákazníkom. V tomto prípade rezervačná cena predstavuje cenu, za ktorú kupujúci tovar predá koncovému zákazníkovi. Je tiež možné si predstavovať, že kupujúci nakupuje polotovary, ktoré ďalej spracúva a predáva. V takom prípade rezervačná cena predstavuje predajnú cenu hotového výrobku zníženú o náklady na spracovanie. Zisk pre i -ty token kupujúceho je prirodzené definovať ako $z_i = r_i - p_i$, kde r_i je rezervačná cena a p_i je nákupná cena i -teho tokenu. V prípade, že kupujúci nepoužije i -tu rezervačnú cenu, definujeme $z_i = 0$. Celkový zisk kupujúceho definujeme ako $Z = \sum_{i=0}^M z_i$. Rovnako ako to bolo v prípade predávajúcich, je aj v prípade kupujúcich rozumné predpokladať, že rezervačné ceny budú nekonštantné. Je totiž možné, že rôzni koncoví zákazníci budú platiť rôzne ceny. V príklade spracovania polotovarov môže mať zas kupujúci k dispozícii rôzne výrobné procesy s rôznymi nákladmi tak, ako sme to uvažovali u predávajúcich. Racionálny kupujúci bude prednostne používať procesy s nižšími nákladmi a obsluhovať zákazníkov, ktorí platia najviac. To sa prejaví tak, že kupujúci bude v simulácii používať rezervačné ceny od najväčšej po najmenšiu.

Pri simulácii, v ktorej použijeme N kupujúcich, pričom každý z nich má mať M rezervačných cien, vygenerujeme najskôr NM rezervačných cien (r_1, \dots, r_{NM}) , ktoré budú zoradené od najväčšej po najmenšiu. Rozdelíme ich do N skupín tak, že v i -tej skupine budú rezervačné ceny $(r_i, r_{N+i}, \dots, r_{(M-1)N+i})$. Týmto spôsobom sa zaručí, že rezervačné ceny budú približne rovnomerne rozmiestnené do všetkých skupín. Nenastanú teda prípady, kedy by v jednej skupine bolo M najväčších cien a v inej M najmenších, čo by výrazne zvýhodnilo kupujúceho s prvou menovanou skupinou. Stále však bude platiť, že skupiny s nižším poradovým číslom budú obsahovať vyššie ceny. Keďže chceme, aby zisk agentov bol závislý len na kvalite ich stratégie a nie na tom, či náhodou dostanú dobré rezervačné ceny, rozdelíme obchodovanie do *periód*. Každá perióda bude pozostávať z pevného počtu krokov. Na začiatku každej periódy dostanú kupujúci náhodne vybranú skupinu rezervačných cien, s ktorou budú v danej perióde obchodovať. Obdobným spôsobom priradíme rezervačné ceny predávajúcim.

Pri dvojitej aukcii s diskretným časom je obchodovanie v každej perióde rozdelené do krokov. V každom kroku agenti oznámia ponúkané resp. žiadané ceny aukcionárovi. Najvyššiu cenu, aká bola ponúknutá v ľubovoľnom kroku od uskutočnenia posledného obchodu až do aktuálneho kroku, označíme ako *súčasná ponuka* (*current bid*). Kupujúceho, ktorý túto cenu ponúkol, označíme ako *súčasného kupujúceho* (*current bidder*). Podobne najnižšiu cenu, aká bola žiadaná v ľubovoľnom kroku od uskutočnenia posledného obchodu až do aktuálneho kroku, označíme ako *súčasnú požiadavku* (*current ask*). Predávajúceho, ktorý túto cenu žiadal, označíme ako *súčasného predávajúceho* (*current asker*). V prípade, že súčasná ponuka je v aktuálnom kroku menšia ako súčasná požiadavka, obchod sa v danom kroku neuskutoční. V opačnom prípade prebehne obchod medzi súčasným kupujúcim a súčasným predávajúcim za cenu, ktorú stanoví aukcionár medzi súčasnou požiadavkou a súčasnou ponukou. My budeme používať aukcionára, ktorý v takomto prípade stanoví cenu obchodu ako aritmetický priemer súčasnej ponuky a požiadavky.

2.3 Meranie výkonu dvojitej aukcie

Súhrn rezervačných cien predávajúcich nazývame *ponuka*. Ponuku môžeme graficky znázorniť ako skokovú funkciu, v ktorej bude každý skok reprezentovať jednu rezervačnú cenu predávajúcich. Keďže predávajúci používajú rezervačné ceny od najmenej po najväčšiu, budeme pri grafickom znázornení rešpektovať toto usporiadanie a ponuku budeme znázorňovať ako neklesajúcu funkciu. Súhrn rezervačných cien kupujúcich nazývame *dopyt*. Analogicky ako v prípade ponuky budeme dopyt znázorňovať ako nerastúcu skokovú funkciu. Ponuka a dopyt spolu tvoria *topológiu trhu*. Príklady topológií trhu sú na obrázku 2.1.

Zákon jednej ceny hovorí, že všetky obchody by na trhu mali prebiehať za jednu cenu. Ak by totiž obchody prebehli za rôzne ceny, predávajúci, ktorí predali za nízke ceny by tým získali informáciu, že existujú kupujúci, ktorí sú ochotní platiť viac. Preto by títo predávajúci v ďalších krokoch žiadali o niečo vyššie ceny. Podobne kupujúci, ktorí nakúpili príliš drahú, by v ďalších krokoch žiadali nižšie ceny. Týmto spôsobom budú ceny konvergovať k jednej cene, ktorú označíme p .

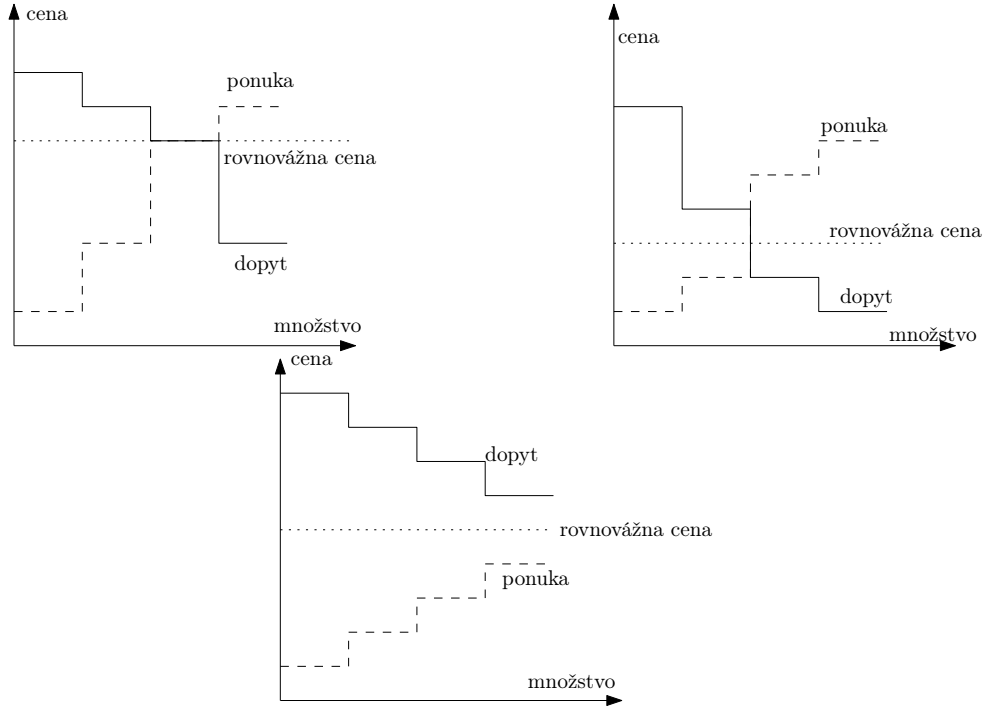
V nasledujúcich odstavcoch zavedieme dôležitý pojem nazývaný *rovnovážna cena*. Nie všetci kupujúci budú ochotní kupovať za cenu p . Ak je rezervačná cena kupujúceho menšia než p , potom by nákupom za p prerobil, a tak je pre neho výhodnejšie nenakupovať. Naopak ak je jeho rezervačná cena väčšia než p , potom bude mať z nákupu zisk, a preto je pre neho výhodné obchod uskutočniť. Ak je rezervačná cena presne rovná p , potom mu je jedno, či obchod uskutoční alebo nie. Pre jednoznačnosť budeme v takomto prípade predpokladať, že sa obchod uskutoční. Označme $Q_K(p)$ počet rezervačných cien kupujúcich, ktoré sú väčšie alebo rovné p . Je to teda množstvo, ktoré chcú kupujúci kúpiť za cenu p . Podobnou úvahou zistíme, že predávajúci sú ochotní predávať, len ak je ich rezervačná cena menšia alebo rovná p . Analogicky teda označíme $Q_P(p)$ počet rezervačných cien predávajúcich, ktoré sú menšie alebo rovné p . Z definície je zrejmé, že $Q_K(p)$ je nerastúca funkcia a $Q_P(p)$ je neklesajúca funkcia.

Uvažujme najskôr situáciu, keď $Q_K(p) > Q_P(p)$. V tomto prípade chcú kupujúci nakúpiť viac, než sú predávajúci ochotní predávať. Z tohto dôvodu sa niektorým kupujúcim nepodarí nakúpiť za cenu p a budú musieť ponúknuť vyššiu cenu. Zo zákona jednej ceny plyní, že aj ostatní kupujúci budú musieť zvýšiť svoje ceny. Ak by to totiž neurobili, predávajúci by prednostne obchodovali s tými, ktorí ponúkajú vyššie ceny, a tak na tých, ktorí ponúkajú nižšie ceny, by už nezvýšilo. Toto spôsobí, že cena p bude rásť, a teda $Q_K(p)$ bude klesať a $Q_P(p)$ bude rásť až kým sa nedosiahne rovnosť

$$Q_K(p) = Q_P(p). \quad (2.1)$$

Rozoberme teraz opačnú situáciu, teda $Q_K(p) < Q_P(p)$. Keďže predávajúci chcú predávať viac, než sú kupujúci ochotní kúpiť, niektorým predávajúcim sa nepodarí svoje tokeny predávať. Títo predávajúci v snahe prilákať kupujúcich znížia predajnú cenu. Analogickou úvahou ako v predchádzajúcom prípade teda dostávame, že cena p bude klesať, čo spôsobí rast $Q_K(p)$ a pokles $Q_P(p)$ až kým nebude platiť 2.1.

Vidíme teda, že cena, za ktorú sa bude na trhu predávať a kupovať, by mala konvergovať k cene spĺňajúcej rovnicu 2.1. Takúto cenu nazveme rovnovážnou cenou a označíme ju p^* . Označíme tiež $Q^* = Q_K(p^*) = Q_P(p^*)$. Q^* teda označuje množstvo, ktoré bude zobchodované pri obchodovaní za rovnovážnu cenu.



Obr. 2.1: Príklady topológií trhu

Pozrime sa teraz na riešenie rovnice 2.1. Predpokladajme, že rezervačné ceny kupujúcich sú $r_1^K \geq r_2^K \geq \dots \geq r_{N_1}^K$ a rezervačné ceny predávajúcich sú $r_1^P \leq r_2^P \leq \dots \leq r_{N_2}^P$. Najskôr predpokladajme, že existuje index i , v ktorom sa rezervačné ceny kupujúcich a predávajúcich pretnú (ak takýchto indexov existuje viac, uvažujme najväčší možný), teda

$$r_i^K = r_i^P. \quad (2.2)$$

Túto spoločnú hodnotu rezervačných cien s indexom i označíme \tilde{p} . Platí $Q_K(\tilde{p}) = Q_P(\tilde{p}) = i$. Pre akékoľvek $p > \tilde{p}$ je $Q_K(p) < i$ a $Q_P(p) \geq i$ a pre $p < \tilde{p}$ je $Q_K(p) \geq i$ a $Q_P(p) < i$. \tilde{p} je teda jediným riešením rovnice 2.1, takže $p^* = \tilde{p}$. Vo väčšine prípadov však index spĺňajúci 2.2 neexistuje. V takom prípade sa môže stať, že rovnica 2.1 má nekonečne veľa riešení, alebo že nemá žiadne riešenie. Keďže chceme mať rovnovážnu cenu jednoznačne definovanú aj v tomto prípade, v analógii s predchádzajúcim prípadom nájdeme bod, ktorý by najlepšie aproximoval bod pretnutia ponuky a dopytu. Nech i je najväčší možný index, pre ktorý platí $r_i^K \geq r_i^P$. Rovnovážnu cenu definujeme ako

$$p^* = \frac{r_i^K + r_i^P}{2}.$$

V ďalšom odvodzovaní budeme pre jednoduchosť predpokladať, že ponuka a dopyt sa pretnú v jednom bode, a teda je možné za rovnovážnu cenu považovať riešenie 2.1. Odvođené výsledky sú však platné obecné.

Je nutné si uvedomiť, že zákon jednej ceny a rovnovážna cena sú teoretické koncepty, ktoré nemusia byť nutne splnené. V reálnom obchodovaní sa ukazuje, že tieto teórie skutočne platia, ale v simuláciách by mohli byť porušené. Mohlo by sa napríklad stať, že jeden kupujúci by nebol dosť inteligentný na to, aby si

všimol, že platí viac než všetci ostatní. Tým by bol porušený zákon jednej ceny. Simulácia s takýmto „hlúpym“ kupujúcim zrejme nie je realistickým modelom skutočného trhu. Pre náš cieľ posúdiť vhodnosť genetického programovania ako nástroja pre ekonomické simulácie je teda vhodné zistiť, ako veľmi sa simulácia líši od teoretických predpovedí. Za týmto účelom definujeme *alfa hodnotu* [8] ako

$$\alpha = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - p^*)^2}}{p^*},$$

kde n je počet uskutočnených obchodov a p_i je cena i -teho obchodu. Veľké alfa hodnoty teda signalizujú veľké odchýlky od rovnovážnej ceny.

Ak sú alfa hodnoty veľké, je to buď preto, že nie je splnený zákon jednej ceny, alebo preto, že zákon jednej ceny síce splnený je, ale na trhu sa obchoduje za inú než rovnovážnu cenu. Problém alfa hodnoty je ten, že neumožňuje rozlíšiť, ktorá z týchto situácií nastala. Preto definujeme *modifikovanú alfa hodnotu*

$$\alpha_{mod} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - \bar{p})^2}}{\bar{p}} = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n p_i^2 - \bar{p}^2}}{\bar{p}},$$

kde $\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i$ je priemerná cena. Modifikovaná alfa hodnota meria vzdialenosť cien, za ktoré sa obchoduje, od ich priemeru. Veľké hodnoty teda signalizujú porušenie zákona jednej ceny.

Ďalšou vlastnosťou, ktorú je možné merať, je efektivita trhu. Predpokladajme, že kupujúci s rezervačnou cenou r^K obchodoval za cenu p s predávajúcim s rezervačnou cenou r^P . Zisk kupujúceho je $r^K - p$ a zisk predávajúceho je $p - r^P$. Celkový zisk, t. j. súčet ziskov kupujúceho a predávajúceho je $r^K - r^P$. Všimnime si, že celkový zisk nezávisí na cene obchodu, iba na rezervačných cenách. V ďalšej časti budeme opäť uvažovať, že všetky rezervačné ceny kupujúcich sú $r_1^K \geq r_2^K \geq \dots \geq r_{N_1}^K$ a rezervačné ceny predávajúcich sú $r_1^P \leq r_2^P \leq \dots \leq r_{N_2}^P$.

Uvažujme najskôr situáciu, keď je splnený zákon jednej ceny a obchoduje sa za rovnovážnu cenu. Z definície Q^* plynie, že kupujúci použijú rezervačné ceny $r_1^K, \dots, r_{Q^*}^K$ a predávajúci použijú rezervačné ceny $r_1^P, \dots, r_{Q^*}^P$. Celkový zisk teda bude

$$\sum_{i=1}^{Q^*} (r_i^K - r_i^P).$$

Uvažujme teraz situáciu, keď $p > p^*$. Vieme, že platí $Q_K(p) \leq Q_P(p)$ a tiež $Q_K(p) \leq Q^*$. Limitujúcim faktorom je teda množstvo, ktoré chcú kupujúci nakúpiť. Kupujúci použijú rezervačné ceny $r_1^K, \dots, r_{Q_K(p)}^K$, pretože to sú rezervačné ceny väčšie než p . Predávajúci majú istú voľnosť v tom, ktoré rezervačné ceny použijú, ale najväčší celkový zisk sa dosiahne, ak použijú prvých $Q_K(p)$ rezervačných cien. Celkový zisk teda môžeme zhora odhadnúť

$$\sum_{i=1}^{Q_K(p)} (r_i^K - r_i^P) \leq \sum_{i=1}^{Q^*} (r_i^K - r_i^P).$$

Vidíme teda, že celkový zisk dosiahnutý v prípade $p > p^*$ je zhora obmedzený celkovým ziskom dosiahnutým pri obchodovaní za rovnovážnu cenu. Analogicky dostaneme rovnaký výsledok pre prípad $p < p^*$. Dá sa dokázať, že väčší zisk nie je možné dosiahnuť ani pri porušení zákona jednej ceny.

Definujme *potenciálny zisk kupujúcich* a *potenciálny zisk predávajúcich* ako zisk, ktorý by dosiahli pri obchodovaní za rovnovážnu cenu, t. j.

$$PZ_K = \sum_{i=0}^{Q^*} (r_i^K - p^*)$$

$$PZ_P = \sum_{i=0}^{Q^*} (p^* - r_i^P).$$

Celkový potenciálny zisk definujeme ako

$$PZ = PZ_K + PZ_P = \sum_{i=0}^{Q^*} (r_i^K - r_i^P).$$

Definujme *realizovaný zisk kupujúcich* RZ_K resp. *realizovaný zisk predávajúcich* RZ_P ako súčet ziskov, ktoré kupujúci resp. predávajúci skutočne dosiahli pri obchodovaní. *Celkový realizovaný zisk* potom definujeme ako

$$RZ = RZ_K + RZ_P.$$

Z predchádzajúcich úvah plynie, že $RZ \leq PZ$. Ako mieru efektivity trhu definujeme *beta hodnotu*, tiež nazývanú beta pomer alebo pomer efektivity, ako

$$\beta = \frac{RZ}{PZ}.$$

Platí teda $\beta \in [0, 1]$, pričom beta hodnoty blízke 1 teda signalizujú, že obchodovanie na trhu je efektívne.

Alfa a beta hodnoty charakterizujú trh ako celok. Niekedy však potrebujeme skúmať kupujúcich a predávajúcich zvlášť. Ak sú napríklad kupujúci „šikovnejší“ než predávajúci, môže sa im podariť prinútiť predávajúcich obchodovať za nižšie ceny než je rovnovážna cena. V takomto prípade bude alfa hodnota veľká, pretože ceny budú ďaleko od rovnovážnej ceny a beta hodnota bude malá, pretože niektorým predávajúcim sa pri tak nízkyh cenách neoplatí obchodovať. Efektivita trhu ako celku bude teda malá, napriek tomu bude tento stav pre kupujúcich výhodný, pretože im zaručí veľké zisky. Pre skúmanie takýchto stavov použijeme *trhovú silu* [15]. *Trhovú silu kupujúcich* definujeme ako

$$TS_K = \frac{RZ_K - PZ_K}{PZ_K},$$

a *trhovú silu predávajúcich* definujeme analogicky ako

$$TS_P = \frac{RZ_P - PZ_P}{PZ_P}.$$

Ak je trhovú sila kupujúcich kladná, znamená to, že kupujúcim sa podarilo dosiahnuť väčší zisk než v rovnovážnom stave.

2.4 Heuristiky pre dvojitú aukciu

Základné príklady heuristik pre dvojitú aukciu pochádzajú z Turnaja dvojitej aukcie (Double auction tournament), ktorí sa uskutočnil v Inštitúte Santa Fe v roku 1990 [18]. Účastníci turnaja vytvorili programy, ktoré následne súťažili v simulovanej dvojitej aukcii.

Algoritmus 2.1 Kaplanov algoritmus pre nákup.

```
most = min(current_ask, reservation_price - 1)
expected_profit = reservation_price - (most + current_ask) / 2
if (there is no current bid)
    bid min_allowed_price + 1
else if (most <= current_bid) don't trade
else if (current_ask <= previous_max_price
    && expected_profit > 0.02 * reservation_price
    && current_ask - current_bid < 0.1 * current_ask)
    bid most
else if (current_ask <= previous_min_price) bid most
else if (time_left <= 1) bid most
else don't trade
```

Algoritmus 2.2 Stratégia Skeleton pre nákup.

```
alpha = 0.25 + 0.1 * random(0,1)
if (there is current bid){
    most = min(current_ask, reservation_price - 1)
    if (most <= current_bid) don't trade
    else bid (1 - alpha) * (current_bid + 1) + alpha * most
}
else{
    most = min(current_ask, last_reservation_price - 1)
    bid most - alpha * (first_reservation_price - last_reservation_price)
}
```

Kaplan Kaplanova stratégia je najlepšou stratégiou z turnaja. Pseudokód pre prípad, že je táto heuristika kupujúcim, je uvedený v algoritme 2.1. Prípad, keď je heuristika použitá pre predaj, je symetrický. Táto heuristika využíva stratégiu „čakania v úzadí“. Počká, kým ostatní agenti privedú najlepšiu ponuku a najlepšiu požiadavku dostatočne blízko k sebe. Potom sa im Kaplanova stratégia pokúsi „ukradnúť“ obchod tým, ponúkne minimum zo súčasnej požiadavky a rezervačnej ceny - 1. Ponúknutím súčasnej požiadavky si Kaplanova stratégia zaistí, že v danom kroku bude obchodovať (ak nikto iný neponúkne viac). Nikdy však neponúkne viac než rezervačná cena - 1, inak by nemala zisk.

Skeleton Pseudokód stratégie Skeleton je uvedený v algoritme 2.2. Táto stratégia v každom kroku navýši najlepšiu ponuku o náhodnú sumu. Dáva však pozor, aby nikdy neponúkla cenu, pri ktorej by obchodovala so stratou.

Ringuette Stratégia Ringuette je uvedená v algoritme 2.3. Táto heuristika, podobne ako Kaplan, čaká v úzadí, až kým sa najlepšia ponuka nepriblíži najlepšej požiadavke. Keď sa tak stane, Ringuette na rozdiel od Kaplana ponúkne najlepšiu požiadavku navýšenú o náhodnú sumu. Tým si zaistí šancu na obchod aj v prípade, že sa niekto ďalší pokúsi obchodovať ponúknutím väčšej ceny než je najlepšia požiadavka. V prípade, že tento postup nefunguje, alebo ak dochádza

Algoritmus 2.3 Stratégia Ringuette pre nákup.

```
span = first_reservation_price - last_reservation_price + 10
if (time_left <= 10 || time_since_my_last_trade >= 10)
    use Skeleton strategy
else{
    alpha = random(0,1)
    if (there is no current ask) don't trade
    else if (current_ask - current_bid < span/5
            && reservation_price > current_ask + span/20)
        bid current_ask + 1 + 0.05 * alpha * span
    else don't trade
}
```

Algoritmus 2.4 Neinteligentný kupujúci.

```
min_price = min(current_bid + 1, reservation_price)
bid random(min_price, reservation_price)
```

čas, začne Ringuette obchodovať podľa stratégie Skeleton.

Gamer Heuristika Gamer vždy ponúkne resp. požiadajú takú cenu, pri ktorej bude mať 10% zisk. Kupujúci teda ponúkne o 10% menej než je jeho aktuálna rezervačná cena a podobne predávajúci požiadajú o 10% viac než je jeho rezervačná cena.

Pravduvraviaci (Truth teller) Pravduvraviaci vždy ponúkne resp. požiadajú svoju rezervačnú cenu. Ponuky resp. požiadavky všetkých heuristik do istej miery odrážajú ich rezervačné ceny, ale väčšina heuristik sa snaží svoje rezervačné ceny nevyzradiť a namiesto toho ponúkajú resp. žiadajú ceny, ktoré budú pre nich výhodnejšie. Ostatné heuristiky teda v istom zmysle klamú, ale pravduvraviaci hovoria pravdu.

Mohlo by sa zdať, že pravduvraviaci bude nutne obchodovať s nulovým ziskom, ale nie je to pravda. Ak napríklad budú spolu obchodovať dvaja pravduvraviaci (pričom kupujúci bude mať vyššiu rezervačnú cenu než predávajúci, inak by nebolo možné obchodovať), obchod prebehne za cenu, ktorá je priemerom rezervačných cien agentov. Každý agent teda dostane polovicu potenciálneho zisku.

Neinteligentný agent Pseudokód neinteligentného kupujúceho je v algoritme 2.4. Neinteligentný agent vždy ponúkne náhodnú cenu, pri ktorej bude mať zisk. Vždy sa však snaží ponúknuť cenu, ktorá je vyššia než súčasná ponuka, pretože inak ho určite aukcionár v danom kroku nevyberie a teda nebude obchodovať.

Kapitola 3

Evolučný algoritmus a genetické programovanie

V tejto kapitole popíšeme spôsob riešenia optimalizačných problémov nazývaný evolučný algoritmus. Tento algoritmus je do značnej miery inšpirovaný biológiou. Na začiatku kapitoly sa zameriame hlavne na popis základnej myšlienky evolučného algoritmu a jeho biologickú motiváciu, pričom technické detaily popíšeme až v ďalších sekciách. Popíšeme genetické programovanie ako špeciálny prípad evolučného algoritmu, ktorý je vhodný pre vyvíjanie stratégií pre obchodovanie v dvojitej aukcii. V závere kapitoly popíšeme rôzne možnosti použitia genetického programovania v dvojitej aukcii.

3.1 Evolučný algoritmus

Evolučný algoritmus [14] je optimalizačný algoritmus, ktorý sa snaží nájsť riešenie daného optimalizačného problému s použitím princípov inšpirovaných biologickou evolúciou. Takýto algoritmus nevytvára iba jedno riešenie, ktoré postupne vylepšuje. Namiesto toho vytvára skupinu možných riešení. V súlade s biologickým náhľadom sa táto skupina nazýva *populácia*. Jednotliví členovia populácie sa potom nazývajú *jedince*.

Biologické organizmy sú do značnej miery určené svojou genetickou výbavou. Táto genetická výbava je tvorená jedným alebo viacerými chromozómami. Základným stavebným kameňom chromozómov sú gény. Gén kóduje istú základnú, ďalej nedeliteľnú informáciu o jedincovi. Každý gén sa môže vyskytovať vo viacerých variantách. Napríklad gén kódujúci farbu očí sa môže vyskytovať vo variantách „modrá“, „hnedá“ a iné. Jednotlivé varianty génu sa nazývajú alely. Evolučný algoritmus preberá tento pohľad a jedince stotožňuje s ich genetickou výbavou. V najjednoduchšom prípade obsahuje jedinec iba jeden chromozóm, ktorý je reprezentovaný postupnosťou pevnej dĺžky nad zvolenou abecedou. Pozícia v postupnosti určuje, o ktorý gén sa jedná, a samotný znak, ktorý sa nachádza na danej pozícii, určuje alelu.¹ Existujú tiež iné reprezentácie jedincov, ktorými sa budeme zaoberať neskôr.

Biologická populácia sa časom vyvíja, a to tak, že jej jedince sa rozmnožujú, čím vznikajú jedince nové, a staré jedince zomierajú. Tento vývoj v evolučnom al-

¹Tento zjednodušený pohľad teda predpokladá, že každý gén má rovnaký počet alel.

goritme zachytíme nasledovne: evolučný algoritmus bude iteratívny, pričom v každej iterácii vznikne nová populácia. Iterácie evolučného algoritmu budeme nazývať *kolá*.² Populáciu, ktorá vznikne v i -tej iterácii, budeme nazývať i -ta generácia. Spravidla i -ta generácia vznikne vývojom z generácie $i - 1$. Biologicky sa toto dá interpretovať tak, že jedince sa v každej generácii rozmnožia, čím dajú vznik novej generácii, a potom hneď zahynú. Tento pohľad je samozrejme dosť zjednodušený. V mnohých biologických populáciách žije niekoľko generácii súčasne, pričom sa dokonca môžu páriť jedince rôznych generácii.

Jedince si v biologickej populácii navzájom konkurujú. Snažia sa získať jedlo, vodu, úkryt, utiecť pred predátormi a podobne. Niektorým sa to darí lepšie, iným horšie. Lepší jedinci majú väčšiu šancu rozmnožovať sa. Je to dané tým, že horší jedinci môžu zomrieť skôr, než sú schopní sa rozmnožovať. U niektorých druhov (napríklad u vlkov) navyše panuje hierarchia, ktorá dovoľuje rozmnožovať sa iba tým najlepším. Tento mechanizmus, ktorým si príroda vyberá najlepšie jedince, ktorým dovolí rozmnožovať sa, sa nazýva prirodzený výber.

Pozrime sa teraz na prirodzený výber z teoretického hľadiska. Každý jedinec rieši problém, ako čo najlepšie prežiť v prírodnom prostredí. Toto riešenie si jedinec nevymýšľa, ale má ho (do istej miery) zakódované v genetickej výbave. Nekvalitné riešenia sú prirodzeným výberom odstránené, kvalitné riešenia postupujú do ďalšej generácie.

Evolučný algoritmus obsahuje analógiu prirodzeného výberu. Najskôr každému jedincovi danej generácie priradí *zdatnosť* (*fitness*). Toto je číslo reprezentujúce, ako dobre daný jedinec rieši optimalizačný problém. Čím je jedinec lepší, tým má väčšiu zdatnosť. Konkrétna implementácia výpočtu zdatnosti vždy závisí na riešenom probléme. Pri jednoduchých problémoch môže byť výpočet zdatnosti priamočiary. Pri zložitejších problémoch (ako je napríklad vyvíjanie stratégií pre dvojité aukcie) je výpočet zdatnosti komplikovanejší, a je len odhadom skutočnej kvality jedincov. Keď má každý jedinec priradenú zdatnosť, prebehne *selekcia*. Jej úlohou je vybrať jedince, ktoré vytvoria základ ďalšej generácie. Ide o výber s opakovaním, teda každý jedinec môže byť vybraný viackrát. Selekcia vyberá predovšetkým jedince s veľkou zdatnosťou, aby nasledujúca generácia bola vytvorená predovšetkým z kvalitných jedincov. Selekcia tiež vyberie zopár jedincov s menšou zdatnosťou. Dôvodom je zachovanie rôznorodosti populácie. Na prvý pohľad nemusí byť zrejmé, prečo je rôznorodosť dôležitá. Jej význam bude jasný, keď popíšeme genetické operátory.

Jedince vybrané selekciou nemôžu priamo vstúpiť do ďalšej generácie. Ak by sa tak stalo, populácia by sa nevyvíjala. Ani v prírode nie sú potomkovia presnou kópiou rodičov, ale vznikajú kombináciou (a prípadne modifikáciou) genetického materiálu rodičov. Evolučný algoritmus používa na kombináciu a modifikáciu jedincov *genetické operátory*. Každý genetický operátor prijíma ako vstup jedného alebo viac jedincov. Tieto jedince sú genetickým operátorom skombinované alebo inak upravené, a tieto nové jedince sú vrátené ako výstup. Evolučný algoritmus má k dispozícii niekoľko genetických operátorov. Pri vytváraní novej generácie opakovane náhodne vyberá genetické operátory.³ Každému dá na vstup jedince

²Tento názov nie je inšpirovaný biológiou, ale naším zámerom použiť evolučný algoritmus v dvojitej aukcii.

³Rôzne genetické operátory majú rôznu pravdepodobnosť výberu. Tieto pravdepodobnosti sú dopredu nastavené, evolučný algoritmus si ich neurčuje sám.

vybrané selekciou z aktuálnej generácie, a výstup vloží do nasledujúcej generácie. Tento postup opakuje, až kým nie je v nasledujúcej generácii dostatok jedincov.

Popísali sme, ako evolučný algoritmus iteratívne vytvára nové generácie s cieľom vytvoriť jedincov, ktorí budú riešiť zadaný problém lepšie než predchádzajúce generácie. Zostáva vyriešiť otázku, ako bude vytvorená počiatočná (nultá) generácia. Táto je zvyčajne vygenerovaná náhodne. Čo presne znamená „náhodný jedinec“ samozrejme závisí na použitej reprezentácii.

3.2 Genetické operátory

Základnými genetickými operátormi sú kríženie, mutácia a reprodukcia. Tieto sú popísané v nasledujúcom texte. Pri jeho čítaní je nutné vziať do úvahy, že tieto operátory boli síce inšpirované biologickými dejmi a zachovávajú ich hlavnú myšlienku, ale v detailoch sa od týchto dejov odlišujú.

Kríženie je genetický operátor inšpirovaný prekrížením chromozómov, čo je biologický dej, ktorý nastáva u niektorých organizmov počas tvorby pohlavných buniek. Tieto organizmy majú v každej bunke dve verzie každého typu chromozómu. Tieto dve verzie sa nazývajú homologický pár. Homologické chromozómy teda obsahujú rovnaké gény, ale môžu obsahovať rôzne alely. Počas prekríženia si dva homologické chromozómy (v rámci jednej bunky) medzi sebou vymenia niektoré alely. Následne sa bunka rozdelí takým spôsobom, že dcérske bunky obsahujú už len jednu verziu každého typu chromozómu. Z týchto buniek vzniknú pohlavné bunky. Pri párení splynú dve pohlavné bunky, čím opäť vznikne bunka obsahujúca dve verzie každého typu chromozómu. Dôsledkom prekríženia chromozómov je to, že alely génov, ktoré sa nachádzajú na jednom chromozóme, nemusia byť dedené ako celok. Bez prekríženia by totiž každý rodič musel odovzdať potomkovi alely len z jedného chromozómu daného homologického páru. Vďaka prekríženiu však rodič odovzdáva potomkovi kombináciu alel nachádzajúcich sa na homologickom páre. Výsledkom je väčšia variabilita v genetickom materiáli potomkov a tiež možnosť, aby sa každý gén pri evolúcii vyvíjal samostatne.

Evolučný algoritmus využíva operátor kríženia ako nejakú analógiu prekríženia chromozómov. Operátor kríženia prijíma na vstupe dva jedince. Medzi týmito jedincami vymení náhodne vybranú časť genetického materiálu, a výsledné jedince vráti na výstup.⁴ Je dôležité si uvedomiť, že tento postup je výrazne odlišný od pôvodného biologického procesu. Biologické prekríženie chromozómov totiž nastáva v rámci jedného jedinca, ale genetický operátor kríženia kombinuje genetický materiál dvoch jedincov.

Motivácia operátoru kríženia je už vo väčšom súlade s biológiou. Predstavme si, že optimalizačný problém, ktorý evolučný algoritmus rieši, sa dá rozdeliť na niekoľko podproblémov. Pre jednoduchosť budeme uvažovať iba dva podproblémy. Predpokladajme, že riešenie pre každý podproblém je kódované jedným génom. Majme dva jedince. Nech prvý jedinec má dobrú alelu pre prvý podproblém a zlú alelu pre druhý podproblém. Symetricky, nech druhý jedinec má dobrú alelu pre druhý podproblém a zlú pre prvý. Keď si takéto jedince pri krížení vymenia

⁴V niektorých implementáciách vráti kríženie iba jedného jedinca, a druhého zahodí. Deje sa tak zrejme preto, že ostatné genetické operátory vracajú tiež len jedného jedinca. Táto verzia kríženia tak môže byť jednoduchšia na implementáciu.

jednu z týchto alel, vznikne jedinec, ktorý má dobré alely pre oba podproblémy, a je teda lepší než obaja jeho rodičia.⁵ Cieľom kríženia je teda kombinovať jedince v nádeji, že každý z nich obsahuje nejakú dobrú časť, a pri krížení sa tieto dobré časti spoja. Tu vidíme, prečo je nutné pri selekcii občas vyberať aj horšie jedince. Horšie jedince totiž tiež môžu obsahovať dobré alely niektorých génov. Ak by sme pri selekcii boli príliš striktní, vybrali by sme veľa kópií pár najlepších jedincov, a kríženie by malo moc málo možností, ako kombináciou vytvoriť nové jedince. V extrémnom prípade by sme vždy vybrali najlepšieho jedinca a kríženie by nemalo žiaden efekt.

Kríženie je náhodná operácia, preto sa môže stať, že sa jedince skombinujú nevhodne. Je napríklad možné, že dva gény na sebe silno závisia a iba niektoré kombinácie alel týchto génov sú prospešné. Predstavme si, že dva gény môžu mať alely A a B, pričom kombinácie AA a BB vytvoria zdatného jedinca, a kombinácie AB a BA vytvoria nezdatného jedinca. Ak si pri krížení jedinca AA a BB vymenia iba jeden z daných génov, vzniknú jedince AB a BA. Zo zdatných jedincov teda krížením vzniknú menej zdatné jedince. Takéto kríženie budeme nazývať deštruktívne kríženie. Deštruktívnemu kríženiu je možné predísť vhodnou voľbou kódovania jedincov, aby boli gény medzi sebou nezávislé. V prírode sa toto darí, a k deštruktívnemu kríženiu prakticky nedochádza. Pri použití evolučného algoritmu sa nie vždy podarí takéto kódovanie nájsť.

Kríženie kombinuje existujúci genetický materiál, ale nie je schopné vytvoriť nič nové. Ak sa nejaká alela daného génu v populácii nenachádza, kríženie ju nedokáže vyrobiť. Z tohto dôvodu sa používa genetický operátor mutácia. Mutácia prijíma jedného jedinca, ktorému náhodne zmení jednu (prípadne niekoľko) alel. Takáto náhodná zmena môže byť prospešná, ale často býva deštruktívna. Preto je nutné mutáciu používať iba s malou frekvenciou.

Ako sme už uviedli, výsledok kríženia aj mutácie je do istej miery náhodný a môže byť deštruktívny. Niekedy použitie týchto operátorov vedie ku zlepšeniu jedincov, inokedy sú výsledné jedince horšie. Ak by sme používali iba tieto dva genetické operátory, mohlo by sa stať, že by sme v jednom kole evolučného algoritmu zničili príliš veľa jedincov, a nezachovalo by sa dosť dobrých jedincov pre ďalšie generácie. Aby sme sa takejto situácii vyhli, používame reprodukciu. Reprodukcia je genetický operátor, ktorý prijíma na vstupe jedného jedinca, ktorého bez zmeny vráti na výstup. Ak sa teda stane, že kríženie a mutácia sú občas náhodou príliš deštruktívne, použitie reprodukcie zaručí, že sa do ďalšej generácie dostane niekoľko jedincov, ktoré neboli poškodené krížením a mutáciou.

3.3 Selekcia

Cieľom selekcie je uprednostniť zdatnejšie programy pred menej zdatnými, aby sa zvýšila celková zdatnosť ďalšej generácie. Zároveň je však nutné zachovať rôznorodosť jedincov, ktorá je potrebná pre dobré fungovanie kríženia. Základné typy selekcie sú *selekcia úmerná zdatnosti* (*fitness proportionate selection*) a *turnajová selekcia* (*tournament selection*).

Pri selekcii úmernej zdatnosti sa pri každom výbere i -ty jedinec vyberie s

⁵Súčasne vznikne jedinec, ktorý má zlé alely pre oba podproblémy, ale ten nás netrápi, pretože v ďalšom kole dosiahne malú zdatnosť a tak bude pravdepodobne pri selekcii odstránený.

pravdepodobnosťou

$$p_i = \frac{f_i}{\sum_{j=0}^N f_j}$$

kde f_j je zdatnosť j -teho jedinca a N je veľkosť populácie.

Pri turnajovej selekcii sa pevne zvolí prirodzené číslo T , ktoré sa nazýva veľkosť turnaju. Potom sa náhodne vyberie skupina obsahujúca T jedincov z aktuálnej generácie, pričom každý jedinec má rovnakú pravdepodobnosť byť vybraný. Z tejto skupiny sa následne vyberie jedinec s najvyššou zdatnosťou.

3.4 Elitizmus

Pri aplikácii genetických operátorov sa môže stať, že dobrý jedinec sa mutáciou alebo krížením zhorší. Takýmto spôsobom sa celková kvalita nasledujúcej generácie môže zhoršiť. Aby sa tomuto javu predišlo, zaviedli sme operátor reprodukcie. Reprodukcia, tak ako všetky genetické operátory, prijíma ako vstup jedinca náhodne vybraného selekciou. Reprodukcia teda zaručí, že v ďalšej generácii bude dostatok dobrých jedincov. Nijako však nezaručí, že sa do ďalšej generácie dostane najlepší jedinec. Môže sa totiž stať, že najlepší jedinec nebude vybraný pre reprodukciu (pretože selekcia je náhodná), a ostatné genetické operátory vytvoria iba horších jedincov než je najlepší jedinec aktuálnej generácie.

Aby sa takejto situácii predišlo, používa sa technika nazvaná *elitizmus*. Pri elitizme sa zvolí prirodzené číslo E . V každej iterácii evolučného algoritmu sa najskôr vyberie E najlepších jedincov, a tí sa skopírujú do nasledujúcej generácie bez zmeny. Zvyšní jedinci sa pre novú generáciu vytvoria použitím selekcie a genetických operátorov tak, ako sme to popísali vyššie.

3.5 Typy evolučných algoritmov

Existuje niekoľko typov evolučných algoritmov. Tieto typy sú v podstate rôzne spôsoby použitia základných myšlienok evolučného algoritmu, ktoré sú popísané vyššie. Líšia sa napr. reprezentáciou jedincov a inými technickými detailami. Rôzne typy môžu byť vhodné pre rôzne aplikácie. V tomto texte sa nebudeme zaoberať všetkými typmi, ale popíšeme iba dva. Prvým je genetický algoritmus, ktorý bol jedným z prvých typov evolučných algoritmov. Ostatné sú od neho odvodené, alebo sú ním inšpirované – to je dôvod, prečo ho popisujeme. Druhým typom je genetické programovanie. Tento typ je vhodný pre použitie v dvojitej aukcii, a preto sa týmto typom budeme zaoberať vo zvyšku tejto práce.

3.5.1 Genetický algoritmus

Generický algoritmus [11] používa jednoduchú reprezentáciu jedincov, ktorú sme už popísali vyššie: reprezentuje jedince ako postupnosti pevnej dĺžky nad zvolenou abecedou. Každý znak reprezentuje alelu daného génu. Táto reprezentácia je vhodná pre problémy, ktorých riešenia sú jednoznačne určené pevným počtom parametrov. Každý parameter je možné zakódovať jedným génom.

Použitie genetických operátorov na takúto reprezentáciu je relatívne jednoduché. Mutácia náhodne vyberie jeden znak z postupnosti, a nahradí ho náhodne

vybraným znakom z abecedy. Kríženie je o niečo komplikovanejšie a existujú dve základné možnosti. Prvou je náhodne vybrať jednu pozíciu a vymeniť medzi danými jedincami všetky znaky, ktoré sa nachádzajú za touto pozíciou. Druhou možnosťou je náhodne vybrať dve pozície a vymeniť znaky, ktoré sa nachádzajú medzi týmito dvomi pozíciami.

Vytvorenie nulte generácie je taktiež priamočiare. Jednoducho sa vytvorí súbor náhodne vygenerovaných postupností.

3.5.2 Genetické programovanie

Hlavný problém genetického algoritmu je jeho pevná reprezentácia. V mnohých prípadoch nie je možné reprezentovať riešenie problému pevným počtom parametrov. Príkladom je vytváranie stratégií pre obchodovanie v dvojitej aukcii. Riešenia tejto úlohy je vhodnejšie reprezentovať ako počítačový program. Tento program dostane na vstupe informácie o stave na trhu, a na výstupe vráti cenu, za ktorú chce obchodovať.

Za účelom riešenia takýchto druhov úloh vzniklo *genetické programovanie* (GP) [12]. Genetické programovanie chápe jedince ako počítačové programy. Z tohto dôvodu sú pri použití genetického programovania jedince tiež nazývané *genetické programy*. Pre jednoduchosť budeme uvažovať iba také programy, ktoré majú na vstupe n čísel, kde n je pevne dané číslo rovnaké pre všetky programy, a ktorých výstupom je jedno číslo.

Genetický program je reprezentovaný ako strom. Listy stromu reprezentujú vstupy programu. Keďže program môže mať viacero vstupov, je nutné medzi nimi rozlišovať. Jednotlivým vstupom priradíme identifikátory (názvy), ktoré budeme nazývať *terminálne symboly*. Každý list teda reprezentuje jeden terminálny symbol. Vnútorne vrcholy reprezentujú funkcie. Každá funkcia má určený počet parametrov, ktorý nazývame *aritou* funkcie. Každý vnútorný vrchol musí mať práve toľko synov, koľko je arita jeho funkcie.

Vyhodnotenie genetického programu prebieha nasledovne. Listy vrátia hodnoty terminálnych symbolov, ktoré program dostal na vstupe. Každý vnútorný vrchol najskôr vyhodnotí svojich synov. Hodnoty, ktoré jeho synovia vrátia, dosadí do funkcie, ktorú reprezentuje, a vráti výsledok. Výstupom programu je hodnota, ktorú vráti jeho koreň.

Implementácia genetických operátorov je o niečo náročnejšia ako v prípade genetického algoritmu. Ak by sme chceli priamo zovšeobecniť genetický algoritmus, mohli by sme vymyslieť nejakú reťazcovú reprezentáciu stromov. Takáto reprezentácia by samozrejme musela používať reťazce premennej dĺžky, ale inak by spĺňala predpoklady genetického algoritmu. Bolo by teda možné použiť genetické operátory v rovnakej forme, v akej sú použité v genetickom algoritme. Tento postup však nijako nerešpektuje stromovú štruktúru genetického programu, a to by viedlo ku komplikáciám. Napríklad pri krížení by si programy vymenili súvislý úsek reťazcovej reprezentácie, ale po spätnej transformácii na strom by tento úsek mohol byť rozhádzaný po celom programe. Takáto drastická zmena programov by pravdepodobne bola deštruktívna. Preto je nutné genetické operátory implementovať iným spôsobom, ktorý bude rešpektovať vnútornú štruktúru programov.

Pri krížení sa najskôr náhodne vyberie jeden vrchol z každého genetického programu. Potom si genetické programy vymenia podstromy zakorenené v týchto

vrcholoch. Táto implementácia kríženia lokalizuje zmenu do jednej časti programu, a tým znižuje pravdepodobnosť deštruktívneho kríženia. Deštrukcia tým samozrejme nie je zďaleka vylúčená.

Mutácia sa v genetickom programovaní vyskytuje vo dvoch variantách, a to *bodová mutácia* a *podstromová mutácia*. Pri bodovej mutácii sa náhodne vyberie vrchol genetického programu, a nahradí sa iným vrcholom s rovnakou aritou. Pri podstromovej mutácii sa náhodne vybraný vrchol z genetického programu odstráni aj s celým podstromom, ktorý je v ňom zakorenený. Na jeho miesto sa náhodne vygeneruje nový podstrom.

3.6 Použitie genetického programovania v dvojitej aukcii

Pri použití genetického programovania v simulácii dvojitej aukcie sú niektorí, prípadne všetci agenti reprezentovaní genetickými programami. Pre použitie genetického programovania je nutné rozdeliť obchodovanie do kôl. Každé kolo bude pozostávať z pevného počtu periód. V každom kole prebehne obchodovanie v dvojitej aukcii a genetickým programom sa priradí zdatnosť odvodená od zisku, ktorý dosiahli. Po skončení obchodovania prebehne evolúcia, ktorá vytvorí agentov pre ďalšie kolo.

Najjednoduchším použitím GP v dvojitej aukcii je *jednopolulačné genetické programovanie*. V tejto konfigurácii je skupina agentov reprezentovaná jednou populáciou genetických programov. Každý genetický program reprezentuje jedného agenta v danej skupine. Ako zdatnosť genetického programu sa použije priamo zisk, ktorý program v danom kole dosiahol. Je dôležité si uvedomiť, že jednopolulačná konfigurácia neznamena, že všetky genetické programy použité pre obchodovanie tvoria jednu populáciu. Napríklad v prípade koevolučného modelu (t. j. kupujúci aj predávajúci sú reprezentovaní genetickými programami) používame samostatnú populáciu pre kupujúcich a samostatnú pre predávajúcich. Je to tak preto, že stratégie na nákup sa líšia od stratégií na predaj a preto nemá zmysel kombinovať ich pri krížení. Charakteristickou vlastnosťou jednopolulačnej konfigurácie je to, že istá skupina agentov je reprezentovaná jedinou populáciou.

Problémom jednopolulačného programovania je, že je z ekonomického hľadiska nerealistické. Kríženie v takejto populácii totiž znamená, že dôjde ku kombinácii stratégií rôznych agentov. Reálni obchodujúci si však svoje stratégie strážia a nezverejňujú ich, a preto takáto kombinácia stratégií nie je možná. Z tohto dôvodu bolo navrhnuté *viacpopulačné genetické programovanie* [7]. V tejto konfigurácii má každý agent svoju vlastnú populáciu genetických programov.⁶ Tieto populácie sa vyvíjajú nezávisle jedna od druhej. Pri takomto použití teda nedochádza ku kombinácii stratégií rôznych agentov. Agent si na začiatku každej periódy náhodne vyberie genetický program zo svojej populácie⁷, ktorý používa počas danej periódy. Na konci kola sa vypočíta zdatnosť programu, ktorá je stanovená ako priemerný zisk daného programu. Zdatnosť je teda celkový zisk, ktorý daný program vyprodukoval, podelený počtom periód, v ktorých bol vybraný. Túto

⁶Skupina agentov je teda reprezentovaná viacerými populáciami – odtiaľ je odvodený názov tejto konfigurácie.

⁷Každý program má rovnakú pravdepodobnosť, že bude vybraný.

konfiguráciu môžeme interpretovať tak, že každý agent má k dispozícii niekoľko stratégií (každá z nich je reprezentovaná genetickým programom), pričom tieto stratégie používa pri obchodovaní. Na základe výsledkov obchodovania vyhodnotí kvalitu svojich stratégií a snaží sa ich vylepšovať.

Kapitola 4

Bloky

Pri predbežných experimentoch sme si všimli, že vyvinuté programy sú často-krát len jednovrcholové stromy, ktoré vrátia hodnotu terminálneho symbolu, ako napríklad rezervačná cena (čo je ekvivalentné heuristike Pravduvraviaci). Pokúsili sme sa preto vymyslieť vlastné genetické operátory, ktoré by mohli pomôcť vyvinúť komplikovanejšie programy.

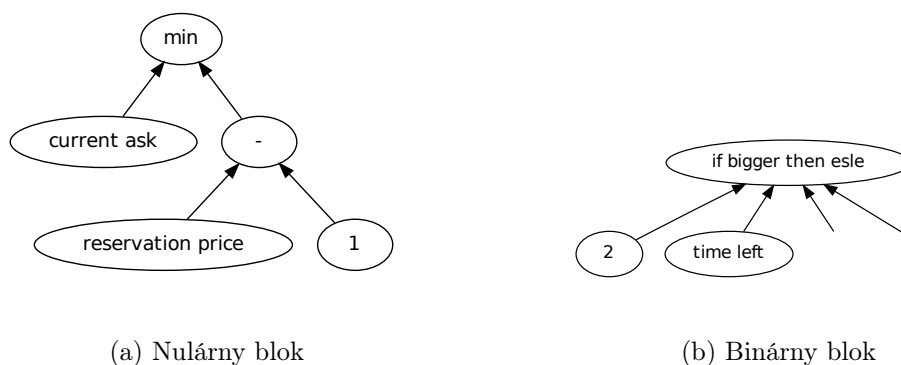
Dôvod, prečo sa vyvíjajú predovšetkým malé programy, môže byť ten, že aj tieto malé programy fungujú dobre. Ak sa aj vytvoria veľké programy, tieto spočiatku počítajú nezmysly a chvíľu trvá, než sa vyvinú. Pri selekcii sa teda spočiatku preferujú malé a funkčné programy, ktoré môžu úplne vytlačiť veľké programy. Pokúsili sme sa preto nájsť spôsob, ako urýchliť vývoj dobre fungujúcich veľkých programov.

Náš postup sa opiera o analýzu heuristik. Pri tejto analýze sme identifikovali niektoré konštrukcie, ktoré je možné použiť ako stavebné kamene pre zložitejšie stratégie. Niektoré konštrukcie sa dokonca vyskytujú vo viacerých heuristikách. Príkladom je \min (súčasná požiadavka, rezervačná cena -1). Táto hodnota má zmysel pre kupujúceho, pretože ponúknutím súčasnej požiadavky si zaistí, že v danom kroku bude obchodovať (ak nikto iný neponúkne viac). Zároveň však kupujúci nechce ponúknuť viac ako rezervačná cena -1 , inak by nemal zisk. Ďalšia konštrukcia, ktorú sme našli vo viacerých heuristikách, je „ak dochádza čas, použi stratégiu 1, inak použi stratégiu 2“.

Pre formalizáciu pojmu „konštrukcia“ zavedieme pojem *blok*. Blokom rozumíme strom zložený z vrcholov reprezentujúcich funkcie a terminálne symboly. Na rozdiel od genetického programu, v bloku môžu mať vrcholy menej synov, než je ich arita. Na to, aby sa z bloku stal genetický program, je nutné takýmto vrcholom doplniť synov. Celkový počet synov, ktorý je nutné takto do bloku doplniť, nazveme *aritou bloku*. Špeciálnym prípadom n -árneho bloku je n -árny vrchol. Príklady blokov sú na obrázku 4.1.

Bloky sú v istom zmysle podobné vrcholom. Podobne ako n -árny vrchol je funkcia, ktorá má n parametrov, je tiež n -árny blok funkcia s n parametrami. Z blokov sa dajú stavať stromy podobne, ako sa dajú stavať z vrcholov. Je teda možné vytvoriť operátory mutácie, ktoré budú používať bloky namiesto vrcholov. Týmto spôsobom umožníme evolučnému algoritmu, aby skladal genetické programy z blokov, čím urýchlime vývoj komplikovanejších programov.

Samozrejme, že by bolo možné namiesto bloku poskytnúť jeden vrchol, ktorý spočíta tú istú funkciu, ako blok. Význam blokov však spočíva v tom, že po vložení



Obr. 4.1: Príklady blokov

do stromu blok nie je nedeliteľný. Strom si totiž „nepamätá“, ktoré vrcholy sa do neho dostali ako súčasť bloku, a ktoré samostatne. Preto, keď je blok vložený do stromu, môže byť ďalej upravovaný v ďalších kolách evolučného algoritmu.

Na tomto mieste je vhodné podotknúť, že nami definované bloky sú odlišné od automaticky definovaných funkcií (ADF), ktoré sú popísané napríklad v [13]. Myšlienka za automaticky definovanými funkciami spočíva v tom, že každý genetický program obsahuje niekoľko stromov, z ktorých jeden reprezentuje hlavný program, a ostatné reprezentujú funkcie. Tieto funkcie môžu byť volané z hlavného programu, a môžu sa tiež volať navzájom. Výhodou takejto štruktúry je, že hlavný program môže opakovane volať tú istú funkciu (prípadne s rôznymi parametrami). Automaticky definované funkcie však riešia iný problém než bloky. Zatiaľ čo ADF sa snažia o to, aby evolučný algoritmus sám vyvinul vhodné funkcie, ktoré bude genetický program používať, blokové operátory sa snažia poskytnúť vhodné funkcie napísané človekom, aby sa urýchlil vývoj GP.

4.1 Blokové genetické operátory

V tejto časti popíšeme funkciu blokových operátorov. Blokové operátory majú k dispozícii množinu blokov, z ktorých náhodne vyberajú podobne, ako „obyčajné“ operátory náhodne vyberajú z množiny vrcholov. Množina blokov je vopred pripravená experimentátorom, nie je vytváraná evolučným algoritmom.

Bloková bodová mutácia Podobne ako pri obyčajnej bodovej mutácii sa pri blokovej bodovej mutácii náhodne vyberie jeden vrchol stromu. Tento vrchol sa následne nahradí náhodne vybraným blokom s rovnakou aritou.

Bloková podstromová mutácia Pri blokovej podstromovej mutácii sa náhodne vyberie jeden vrchol stromu a odstráni sa celý podstrom, ktorý je v ňom zakorenený. Následne sa z blokov vygeneruje nový podstrom, ktorý sa vloží na jeho miesto.

Kapitola 5

Experimenty

V tejto kapitole popíšeme experimenty, ktoré sme vykonali. Pre uskutočnenie experimentov sme vyvinuli software, ktorý sa spolu s dátami z experimentov nachádza na priloženom CD. V experimentoch sme použili jednopopulačné aj viacpopulačné genetické programovanie. V oboch prípadoch sme použili evolučný algoritmus bez blokových operátorov, a tiež evolúciu s blokovými operátormi. Celkom teda máme štyri konfigurácie genetického programovania. Cieľom tejto kapitoly je na základe experimentov rozhodnúť, ktoré z týchto konfigurácií sú vhodné modely reálnych obchodujúcich, prípadne či sú niektoré konfigurácie lepšie než iné.

V niektorých experimentoch sme použili koevolučný model, teda predávajúci aj kupujúci boli vyvíjaní GP. V týchto experimentoch sme kupujúcich aj predávajúcích vyvíjali tou istou konfiguráciou GP. Inde sme použili evolučný model, v ktorom sme kupujúcich vyvíjali genetickým programovaním a ako predávajúcích sme použili heuristiky popísané v prvej kapitole. Niektoré z týchto heuristik (ako napr. pravduvraviaci) sú jednoduché a dalo by sa očakávať, že budú dosahovať malé zisky pri obchodovaní. Mohlo by sa teda zdať vhodné takéto heuristiky vyradiť a nechať len tie najlepšie. Problémom je to, že niektoré veľmi kvalitné heuristiky môžu v tomto prípade fungovať veľmi zle. Príkladom je Kaplanova heuristika, ktorá čaká v úzadí, kým sa najlepšia ponúkaná cena dostatočne nepriblíži najlepšej požadovanej cene. Keď sa tak stane, požiadala cenu, ktorá bude nižšia než najlepšia ponuka, a tým pádom dôjde k obchodu. Táto heuristika implicitne predpokladá, že nejaké iné heuristiky dostanú najlepšiu požiadavku dostatočne blízko k najlepšej ponuke. Ak budú všetci predávajúci riadení Kaplanovou heuristikou (prípadne inou heuristikou, ktorá čaká v úzadí), budú všetci čakať, než niekto iný niečo urobí, a takéto obchodovanie bude príliš neefektívne. Pri predbežných experimentoch sa tiež ukázalo, že niektoré GP obsahovali jediný vrchol, ktorým bola rezervačná cena, teda tieto GP boli pravduvraviaci. Táto heuristika teda nemusí byť až také zlá, ako na prvý pohľad vyzerá. Preto sme sa rozhodli pre obchodovanie použiť všetky heuristiky.

V každom experimente sme nechali najskôr prebiehať obchodovanie predpísaný počet kôl, aby sa genetické programy mohli vyvinúť. Potom sme skúmali vlastnosti genetických programov na základe výsledkov obchodovania v poslednom kole. Skúmali sme teda vždy poslednú (najvyvinutejšiu) generáciu genetických programov.

Pre experimenty s genetickými programami je typické, že sa zvyčajne zisťuje

kvalita najlepšieho vyvinutého genetického programu. Takýto experiment sme pre úplnosť zaradili aj do tejto práce. Je však nutné si uvedomiť, že takýto experiment nie je príliš relevantný z hľadiska výpočtovej ekonómie. V ekonomických simuláciách sa zvyčajne nevyberá najlepší genetický program, ale nechávajú sa obchodovať všetky, pričom sa na základe výsledkov obchodovania zisťujú vlastnosti daného trhu. Je preto dôležitá skôr kvalita celej populácie, než kvalita najlepšieho jedinca. Z tohto dôvodu sme sa v experimentoch zamerali viac na charakterizácie populácie ako celku.

5.1 Parametre experimentov

Parametre, ktoré sme v experimentoch použili, sú zhrnuté v tabuľke 5.1. Vo všetkých experimentoch bol počet kupujúcich rovný počtu predávajúcich. V prípade jednopopulačného programovania musí byť táto hodnota rovná veľkosti populácie. Keďže počet heuristik, ktoré používame, je menší než veľkosť populácie, použili sme v evolučnom modeli jednopopulačného programovania každú heuristiku viackrát. V prípade viacpopulačného programovania počet agentov nezávisí na veľkosti populácie a môže byť ľubovoľný. Pre porovnanie jedno- a viacpopulačného programovania by bolo ideálne, keby sme aj v tomto prípade nastavili počet agentov ako veľkosť populácie. Problém je v tom, že viacpopulačné programovanie by sa tak stalo výpočetne o mnoho náročnejšie než jednopopulačné, pretože by používalo príliš veľa populácií. Je teda nepravdepodobné, že by niekto použil viacpopulačné programovanie s takýmto veľkým počtom agentov (v literatúre [6, 8] sme našli viacpopulačné GP so 4 agentmi). Preto sme sa aj my rozhodli použiť menší počet agentov, konkrétne 6, čo je počet heuristik, ktoré používame.

Kvôli rôznemu počtu agentov v jedno- a viacpopulačnej konfigurácii používame tiež rôzny počet rezervačných cien pre každého agenta. Viacpopulačná konfigurácia používa menej agentov než jednopopulačná, preto každý agent dostane viac rezervačných cien. Tieto hodnoty sú nastavené tak, aby bol celkový počet rezervačných cien rovnaký v oboch konfiguráciách. To nám umožní v oboch prípadoch použiť tú istú topológiu, vďaka čomu budeme môcť medzi nimi jednoducho porovnať výsledky.

Ďalším parametrom, ktorý sa líši pre jednopopulačnú a viacpopulačnú konfiguráciu, je počet períód. Ako sme uviedli v kapitole 2, použitie väčšieho počtu períód zabezpečí, že každý agent bude v každom kole vyhodnotený viackrát s rôznymi rezervačnými cenami. Cieľom je, aby jeho zisk nezávisel na tom, či šťastnou náhodou dostane dobré rezervačné ceny, ako by tomu bolo v prípade použitia len jednej períódy. Je teda nutné zvoliť dostatočne veľký počet períód, aby každý agent dostal reprezentatívnu vzorku rezervačných cien. V prípade jednopopulačnej konfigurácie sme sa rozhodli použiť 10 períód na každé kolo. V prípade viacpopulačnej konfigurácie musí byť počet períód väčší. Každý agent totiž obsahuje 48 genetických programov, a každý z týchto programov musí byť vyhodnotený na dostatočne veľkej vzorke rezervačných cien. Mohlo by sa zdať, že pre dosiahnutie rovnakého efektu ako pri jednopopulačnom programovaní bude treba použiť až $48 \cdot 10 = 480$ períód. Tento počet je však príliš veľký a spôsobil by, že viacpopulačné programovanie by sa stalo výpočetne o mnoho náročnejšie než jednopopulačné. Pri bližšom pohľade však zistíme, že takýto veľký počet períód vôbec nie je potrebný. Viacpopulačné programovanie totiž používa iba 6 agentov,

	Jedna populácia	Viac populácií
Veľkosť populácie		48
Počet agentov	48	6
Počet kôl		100
Počet periód	10	144
Počet krokov	4-krát počet rezervačných cien	
Počet rezervačných cien každého agenta	4	32
Celkový počet rezervačných cien	192	192

Tabuľka 5.1: Parametre dvojitej aukcie.

+, -, *, /	Základné aritmetické operácie.
abs	Absolútna hodnota.
exp, log, sin, cos	Exponenciála, logaritmus a goniometrické funkcie.
max, min	Maximum a minimum dvoch hodnôt.
if bigger then else	Ak je hodnota prvého argumentu väčšia než hodnota druhého argumentu, vráti tretí argument, inak vráti štvrtý argument.
constant	Reprezentuje celočíselnú konštantu.
pass	Žiadosť genetického programu neobchodovať.

Tabuľka 5.2: Funkcie použité v GP

a teda rezervačné ceny sú rozdelené do 6 skupín. Ak by bol každý program vyhodnocovaný v 10 periódach, dostal by každú skupinu v priemere viac než raz, čo je zbytočne veľa. Pre viacpopulačné programovanie postačí, ak bude každý program vyhodnotený v priemere 3-krát, teda bude vyhodnotený na polovici rezervačných cien. Celkový počet periód viacpopulačného programovania sme preto nastavili na $48 \cdot 3 = 144$.

5.1.1 Evolučný algoritmus

Pre použitie genetického programovania je najskôr nutné špecifikovať základné funkcie a terminálne symboly, z ktorých budú genetické programy zostavené. Funkcie, ktoré sme sa rozhodli použiť, sú zobrazené v tabuľke 5.2 a terminálne symboly v tabuľke 5.3. Náš výber je inšpirovaný predovšetkým [6].

Používame dve konfigurácie evolučného algoritmu - bez blokových operátorov a s blokovými operátormi. Pri konfigurácii bez blokových operátorov sa používajú operátory reprodukcia, kríženie, bodová mutácia, podstromová mutácia. Ich pravdepodobnosti sú zobrazené v tabuľke 5.4. Pri konfigurácii s blokovými operátormi sa namiesto bodovej a podstromovej mutácie používa bloková bodová mutácia a bloková podstromová mutácia. Pravdepodobnosti operátorov v tejto konfigurácii sú zobrazené v tabuľke 5.5. Blokové operátory využívajú ako bloky okrem špeciálne navrhnutých blokov aj všetky terminálne symboly a funkcie. Pri konfigurácii s blokovými operátormi sa tiež nultá generácia vytvára pomocou blokov.

5.1.2 Topológia trhu

Použili sme štyri rôzne topológie trhu, ktoré sú zobrazené na obrázku 5.1. V symmetrickej topológii rezervačné ceny kupujúcich klesajú a rezervačné ceny predávajúcich rastú. V topológii s plochým dopytom sú rezervačné ceny kupujúcich

reservation price, first reservation price last reservation price	Aktuálna, prvá, a posledná rezervačná cena.
current bid, current ask	Súčasná ponuka a požiadavka.
max bid, min bid, max ask, min ask	Maximálna a minimálna ponuka a požiadavka v predchádzajúcom kroku.
trade price	Cena, za ktorú sa obchodovalo v predchádzajúcom kroku.
time non trade	Počet po sebe idúcich krokov, v ktorých nedošlo k obchodu.
time left	Počet krokov do konca periódy.
previous max bid, previous min bid previous max ask, previous min ask	Maximálna a minimálna ponuka a požiadavka v predchádzajúcej perióde.
previous max prize, previous min prize previous average prize	Maximálna, minimálna a priemerná cena, za ktorú sa obchodovalo v predchádzajúcej perióde.
total time	Celkový počet krokov periódy.
min allowed price, max allowed price	Minimálna a maximálna cena, ktorú môžu obchodujúci ponúkať.

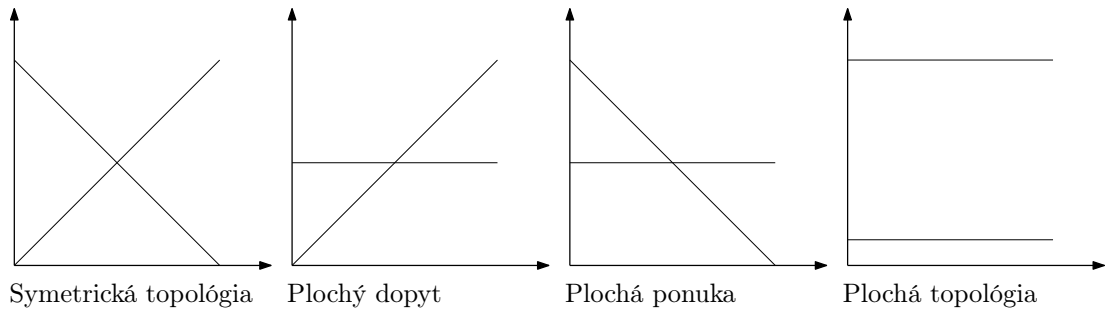
Tabuľka 5.3: Terminálne symboly

Reprodukcia	10%
Kríženie	85%
Bodová mutácia	4%
Podstromová mutácia	1%

Tabuľka 5.4: Pravdepodobnosti výberu operátorov v konfigurácii bez blokových operátorov.

Reprodukcia	10%
Kríženie	85%
Bloková bodová mutácia	4%
Bloková podstromová mutácia	1%

Tabuľka 5.5: Pravdepodobnosti výberu operátorov v konfigurácii s blokovými operátormi.



Obr. 5.1: Topológie trhu použité pri simuláciách.

konštantné, pričom rezervačné ceny predávajúcich rastú. V topológii s plochou ponukou sú rezervačné ceny predávajúcich konštantné, pričom rezervačné ceny kupujúcich klesajú. V plochej topológii sú rezervačné ceny kupujúcich aj predávajúcich konštantné, pričom ceny kupujúcich sú vyššie než ceny predávajúcich (inak by nebolo možné obchodovať).

Ako sme uviedli v prvej kapitole, topológia trhu je reprezentovaná skokovými funkciami, ale funkcie na obrázku 5.1 skokové nie sú, preto je nutné ich diskretizovať. Popíšeme diskretizáciu dopytu, ponuka sa diskretizuje analogicky. V našom postupe využijeme fakt, že všetky použité topológie obsahujú iba lineárne funkcie, ktoré sa dajú jednoducho aproximovať skokovými funkciami. Predpokladajme teda, že máme lineárny dopyt začínajúci na hodnote a a končiaci na hodnote b , kde $a \geq b$. Najskôr zvolíme celkový počet rezervačných cien kupujúcich (označme ho N). Rezervačné ceny určíme ako vektor (r_1, \dots, r_N) taký, že $r_1 = a$, $r_N = b$ a $r_i - r_{i+1} = \frac{a-b}{N-1}$, $i = 1, \dots, N-1$.

Parametre experimentov volíme tak, aby bol celkový počet rezervačných cien kupujúcich aj predávajúcich rovnaký pre všetky konfigurácie. Vďaka tomu sa každá topológia zdiskretizuje rovnako vo všetkých konfiguráciách. Táto voľba je motivovaná tým, že niektoré charakteristiky topológie sú závislé na počte rezervačných cien, ktoré sa diskretizáciou vytvoria.

Príkladom takejto závislosti je rovnovážna cena. Hoci vo všetkých nediskretizovaných topológiách s výnimkou plochej topológie sa ponuka a dopyt pretnú v jednom bode (a rovnovážna cena je teda jednoznačne určená), v diskretizovaných topológiách to tak byť nemusí. Diskretizácia môže spôsobiť, že ponuka a dopyt sa nepretnú v jednom bode, a v takom prípade sa rovnovážna cena definuje ako aproximácia pretnutia ponuky a dopytu. Presná hodnota tejto aproximácie závisí na hodnotách rezervačných cien, ktoré zas závisia na diskretizácii.

Ďalšou charakteristikou závislou na diskretizácii je potenciálny zisk kupujúcich a z neho odvodená trhová sila kupujúcich (a obdobne tiež potenciálny zisk a trhová sila predávajúcich). Potenciálny zisk kupujúcich sa počíta z rezervačných cien kupujúcich a z rovnovážnej ceny. Keďže oba tieto parametre závisia na diskretizácii, je zřejmé, že potenciálny zisk bude závisieť tiež.

Táto závislosť sa obzvlášť prejaví v topológii s plochým dopytom. „Nediskretizovaná“ rovnovážna cena je totiž v tejto topológii rovná rezervačným cenám kupujúcich, takže „nediskretizovaný“ potenciálny zisk kupujúcich je nulový. Kvôli diskretizácii je však rovnovážna cena o niečo nižšia než rezervačné ceny kupujúcich, a preto je potenciálny zisk kupujúcich veľmi malé kladné číslo. Toto číslo veľmi závisí na diskretizácii – aj malá zmena počtu rezervačných cien môže vy-

volať relatívne veľkú zmenu potenciálneho zisku a tým veľmi zmeniť trhovú silu. Použitie trhovej sily kupujúcich pre hodnotenie obchodovania v topológii s plochým dopytom teda považujeme za nevhodné, pretože jej hodnota veľmi závisí nie len na obchodovaní agentov, ale aj na počte rezervačných cien. Navyše veľmi malá zmena celkového zisku kupujúcich môže vyvolať veľkú zmenu ich trhovej sily, pretože potenciálny zisk (ktorým sa delí), je veľmi malý. Toto považujeme za zavádzajúce. V topológii s plochým dopytom teda použijeme inú charakteristiku, ktorá na parametroch experimentu až tak nezávisí.

5.1.3 Štatistické spracovanie

Pri štatistickom spracovaní narážame na dva typy úloh. Prvým je zisťovanie, či sa namerané hodnoty štatisticky významne líšia od konštanty. Príkladom je určenie, či je trhovú silu kupujúcich rôzna od nuly. Na takéto typy úloh používame t-test.

Druhým typom úlohy je porovnanie viacerých nameraných vzoriek. Príkladom môže byť porovnanie trhovej sily 4 rôznych konfigurácií genetického programovania. V takomto porovnaní testujeme nulovú hypotézu, že všetky konfigurácie sú rovnaké, proti alternatíve, že niektoré konfigurácie sa od seba líšia. V prípade zamietnutia nulovej hypotézy chceme samozrejme určiť, ktoré dvojice konfigurácií sa od seba významne líšia. Na prvý pohľad by sa mohlo zdať, že stačí použiť 6 dvojvýberových t-testov (jeden pre každú dvojicu). Problém tohto postupu je, že nedodrží predpísanú hladinu testu. Predpokladajme, že platí nulová hypotéza, že žiadne dve konfigurácie sa od seba nelíšia. Každý z t-testov môže nulovú hypotézu zamietnuť s pravdepodobnosťou α , kde α je predpísaná hladina testu. Nulovú hypotézu potom zamietneme, ak ju zamietne aspoň jeden t-test. Celková pravdepodobnosť zamietnutia nulovej hypotézy je potom ale väčšia než α . Z tohto dôvodu je použitie t-testov nevhodné. Namiesto nich používame Tukeyho metódu (tiež nazývanú Tukey's honest significant difference test) popísanú v [1].

Tukeyho metóda spočíta pre každú dvojicu (modifikovanú) p-hodnotu. Ak je táto p-hodnota menšia než α , považujeme danú dvojicu za štatisticky významne rozdielnu. Rozdiel oproti sérii t-testov je ten, že pravdepodobnosť, že aspoň jedna modifikovaná p-hodnota je menšia než α , je za platnosti nulovej hypotézy najviac α . Tukeyho metóda teda dodržiava hladinu testu.

V grafoch a tabuľkách označujeme jednopopulačnú konfiguráciu bez blokov ako J, jednopopulačnú konfiguráciu s blokmi ako JB, viacpopulačnú konfiguráciu bez blokov ako V a viacpopulačnú konfiguráciu s blokmi ako VB. Vo všetkých testoch používame hladinu $\alpha = 0.05$.

5.2 Ziskovosť najlepšieho agenta

V prvom experimente sme zisťovali, či genetické programovanie dokáže vyvinúť lepších agentov než sú agenti reprezentovaní heuristikami. Použili sme evolučný model, teda ako kupujúcich sme použili agentov riadených GP a ako predávajúcich heuristiky. Obchodovanie prebiehalo v symetrickej topológii, ktorá je najjednoduchšia a najrealistickejšia zo všetkých použitých topológií. Zisťovali sme vlastnosti najlepšieho agenta vyvinutého GP. Hlavným cieľom tohto experimentu je určenie ziskovosti, preto sme porovnávali zisk najlepšieho agenta vyvinutého

GP so ziskom najlepšej heuristiky. Okrem zisku sme tiež zisťovali zložitosť najlepšieho genetického programu.

Ako sme uviedli v kapitole 4, dôvodom pre použitie blokových genetických operátorov je snaha o vytvorenie komplikovanejších stratégií pre obchodovanie. Ako mieru zložitosti agentov používame ich veľkosť. V prípade jednopopulačného programovania veľkosť agenta definujeme ako počet vrcholov jeho genetického programu. V prípade viacpopulačného programovania je veľkosť agenta priemerný počet vrcholov genetických programov jeho populácie.

Zložitosť agentov je nutné interpretovať opatrne. Zložitosť predstavuje potenciál pre vytvorenie lepších stratégií pre obchodovanie, pretože väčšie stromy môžu kódovať komplikovanejšie programy. Väčšie stromy však samé o sebe nezaručia, že programy budú schopné obchodovať lepšie. Naopak sa môže stať, že najlepšia stratégia bude reprezentovaná jednoduchým programom. Väčšie stromy tiež znamenajú, že vývoj a vyhodnocovanie agentov budú časovo náročnejšie. Nárast zložitosti agentov má teda význam len v prípade, že dôjde k zodpovedajúcemu nárastu ziskovosti, inak je takáto zložitosť zbytočná. Zložitosť je teda nutné chápať len ako potenciál, ale primárnym kritériom musí zostať ziskovosť.

5.2.1 Možnosti porovnania najlepších agentov

Ponúka sa niekoľko možností, ako je možné porovnávať najlepších agentov. Prvou je vybrať najlepších agentov z posledného kola a následne ich nechať obchodovať samostatne, aby sme zistili, ktorý z nich je lepší. Druhou je vybrať najlepšiu heuristiku dopredu a najlepší GP z posledného kola a opäť ich nechať obchodovať samostatne. Treťou je vybrať najlepších agentov z posledného kola a na porovnanie použiť výsledky z tohto kola bez samostatného obchodovania. V nasledujúcom texte popíšeme tieto možnosti podrobnejšie.

Prvou možnosťou je najskôr vybrať kupujúceho a predávajúceho, ktorí dosiahli najväčší zisk v poslednom kole obchodovania. Týchto agentov potom necháme obchodovať samostatne (t. j. bez prítomnosti ostatných agentov) a porovnáme ich zisk v tomto samostatnom obchodovaní. Výhodou tejto možnosti je, že odstraňuje vplyv ostatných agentov. Jej problém spočíva v tom, že testuje genetický program v prostredí, v ktorom nebol vyvíjaný. Je totiž možné, aby genetické programovanie vyvinulo stratégiu, ktorá funguje dobre pri obchodovaní s inou než najlepšou heuristikou. Počas vývoja, keď sú prítomné všetky heuristiky, môže táto stratégia dosahovať veľké zisky, a môže byť dokonca najlepšia, a pritom nemusí nikdy obchodovať s najlepšou heuristikou. Ak však bude táto stratégia vybraná pre samostatné obchodovanie s najlepšou heuristikou, môže dosahovať malé zisky, pretože na takúto situáciu nebola vyvinutá.

Druhou možnosťou je a priori vybrať, ktorú heuristiku považujeme za najlepšiu. Pri vývoji genetických programov potom použijeme iba túto heuristiku (t. j. všetci predávajúci budú riadení touto heuristikou). Následne vyberieme najlepšieho kupujúceho a predávajúceho a necháme ich spolu obchodovať rovnako ako v predchádzajúcom prípade. Tento postup rieši problém, ktorý vznikol v prvom porovnaní, pretože všetky genetické programy sú vyvinuté tak, aby obchodovali s najlepšou heuristikou. Ako sme však už uviedli v úvode tejto kapitoly, niektoré kvalitné heuristiky nemusia v takomto použití fungovať dobre, pretože vyžadujú prítomnosť ďalších heuristik.

Ďalšou možnosťou je porovnávať najlepšieho kupujúceho a predávajúceho priamo podľa zisku, ktorý dosiahli v poslednom kole obchodovania. V takomto porovnaní budú agenti obchodovať presne v takom prostredí, na aké boli vyvinutí. Problémom však je, že na výsledok tohto porovnania majú vplyv všetci agenti, a nie len najlepší, ktorých chceme porovnať. Môže sa napríklad stať, že najlepší genetický program bude využívať slabinu nejakej slabej heuristiky, a vďaka tomu dosiahne veľký zisk. Takisto zisk najlepšej heuristiky bude závisieť na tom, či budú existovať slabé genetické programy, ktoré môže obrať o zisk, alebo či budú všetky genetické programy kvalitné. Hoci táto možnosť priamo porovnáva iba najlepších agentov, vidíme, že nepriamo do porovnania vstupujú aj všetci ostatní agenti. Ako sme už uviedli v úvode tejto kapitoly, je pre nás dôležitejšie správanie trhu ako celku než správanie jednotlivcov, a preto tento problém nepovažujeme za zásadný. Preto sme sa v experimente rozhodli použiť túto možnosť porovnania.

5.2.2 Výsledky experimentu

Experiment sme s každou konfiguráciou GP spustili 50-krát. Zakaždým sme zaznamenali zisk najlepšieho kupujúceho a predávajúceho, a tiež veľkosť najlepšieho kupujúceho. Zisky najlepších agentov sme vydělili počtom rezervačných cien, ktoré agenti dostali v jednom kole.¹ Viacpopulačné konfigurácie mali v každom kole viac rezervačných cien než jednopolulačné, preto je rozumné očakávať, že dosiahli väčšie zisky. Toto normovanie nám teda umožní porovnávať zisky medzi konfiguráciami napriek rôznym počtom rezervačných cien.

Normované zisky najlepších agentov sme vyhodnotili obojstranným párovým t-testom, t. j. pre každý experiment sme spočítali rozdiel ziskov najlepších agentov, tieto rozdiely sme následne spriemerovali a určili sme p-hodnotu. Zvolili sme obojstrannú variantu, pretože dopredu nie je jasné, či sa genetickým programom podarí prekonať heuristiky, alebo či to bude naopak. Obojstranná varianta umožňuje detekovať aj prípad, keď budú heuristiky štatisticky významne lepšie než GP.

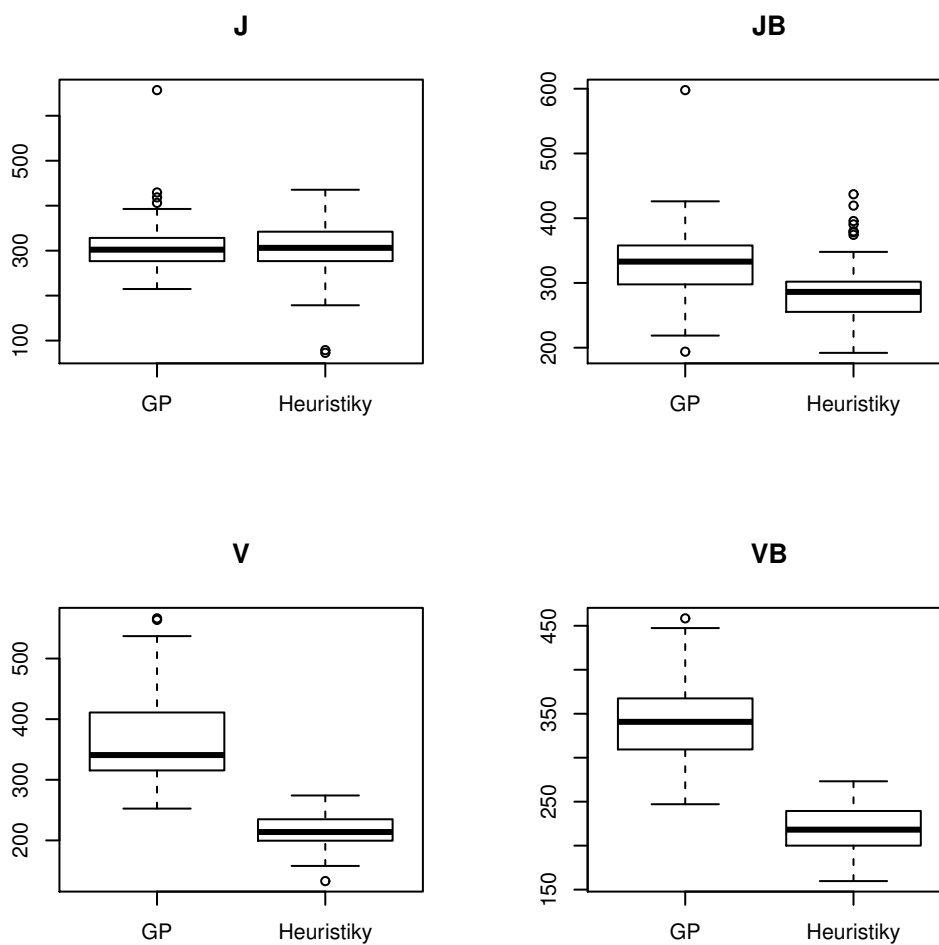
Porovnávali sme tiež najlepších agentov naprieč konfiguráciami. Pre tento účel sme použili Tukeyho metódu. Problém je v tom, že pre každý experiment máme k dispozícii dve hodnoty (zisk najlepšieho kupujúceho a predávajúceho), ale Tukeyho metóda vyžaduje iba jednu hodnotu. Preto sme sa rozhodli vyhodnotiť rozdiely normovaných ziskov najlepšieho kupujúceho a najlepšieho predávajúceho. Konfigurácie, ktoré vyprodukovali väčšie rozdiely, považujeme za lepšie.

Veľkosti najlepších kupujúcich sme vyhodnotili Tukeyho metódou. Porovnávali sme, či niektoré konfigurácie vytvorili zložitejších agentov než iné.

Zisky najlepších agentov sú zhrnuté v tabuľke 5.6. Pre lepšiu predstavu tiež uvádzame krabičkový graf na obrázku 5.2. Sú na ňom zobrazené zisky najlepšieho GP a najlepšej heuristiky z každého opakovania experimentu.

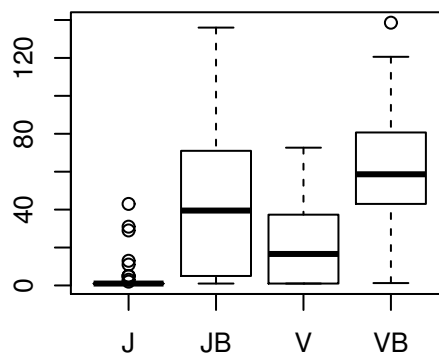
Pri pohľade do tabuľky vidíme, že nárast ziskov genetických programov je spojený s poklesom ziskov heuristik. Je to tak preto, že celkový realizovaný zisk je obmedzený celkovým potenciálnym ziskom, ktorý je určený topológiou trhu. Ak chcú teda genetické programy zvýšiť svoj zisk, musia tak nutne urobiť na úkor zisku heuristik.

¹Tento počet získame vynásobením počtu periód na jedno kolo a počtu rezervačných cien jedného agenta na jednu periódu.



Obr. 5.2: Krabičkový graf ziskov najlepších agentov.

Veľkosť najlepšieho agenta



Obr. 5.3: Priemerná veľkosť genetických programov

	J	JB	V	VB
priemer ziskov najlepších GP	313	328	368	345
priemer ziskov najlepších heuristik	302	291	214	219
priemer rozdielu ziskov najlepších agentov	11	36	153	125
p-hodnota t-testu	0.570	0.031	<0.001	<0.001

(a) Priemery normovaných ziskov najlepších agentov.

	JB	V	VB
J	0.591	<0.001	<0.001
JB		<0.001	<0.001
V			0.520

(b) P-hodnoty rozdielov ziskovosti medzi konfiguráciami.

Tabuľka 5.6: Normované zisky najlepších agentov.

J	JB	V	VB
3.74	44.72	20.93	61.56

(a) Priemerné veľkosti najlepších agentov.

	JB	V	VB
J	<0.001	0.007	<0.001
JB		<0.001	0.008
V			<0.001

(b) P-hodnoty Tukeyho metódy.

Tabuľka 5.7: Veľkosť najlepších agentov.

Jednopolulačná konfigurácia bez blokov nedosiahla zisky, ktoré by sa významne líšili od ziskov heuristik. Naproti tomu všetky ostatné konfigurácie vyprodukovali významne väčšie zisky než heuristiky. Z Tukeyho metódy vidíme, že viacpopulačné konfigurácie dosiahli významne väčšie zisky než jednopolulačné. Rozdiel medzi jednopolulačnou konfiguráciou bez blokov a s blokmi nie je významný, a rovnako je na tom rozdiel medzi viacpopulačnou konfiguráciou bez blokov a s blokmi.

Vidíme teda, že primárnym kritériom pre ziskovosť je počet populácií, pretože obe konfigurácie viacpopulačného programovania dokázali vytvoriť lepších agentov než než obe konfigurácie jednopolulačného programovania. Použitie blokov je sekundárnym kritériom. V prípade viacpopulačného programovania nemá použitie blokov vplyv na kvalitu agentov. Analýza jednopolulačného programovania je o niečo komplikovanejšia. Jednopolulačné programovanie s blokmi je totiž významne lepšie než heuristiky, ale jednopolulačné programovanie bez blokov sa významne nelíši ani od heuristik ani od JB. Je zrejmé, že ak sa JB líši od heuristik, potom sa J musí líšiť od heuristik alebo od JB (prípadne od oboch), ale v našich experimentoch sme ani jednu odlišnosť nezaznamenali. Na zistenie odlišnosti by teda bol nutný väčší počet experimentov. S prihliadnutím na túto skutočnosť nie je možné jednoznačne určiť, či má použitie blokových operátorov vplyv na kvalitu programov vyvinutých v jednopolulačnej konfigurácii.

Pre porovnanie veľkostí vyvinutých agentov sme v každom opakovaní experimentu zaznamenali veľkosť najlepšieho agenta vyvinutého genetickým programovaním. Tieto veľkosti sme vyhodnotili Tukeyho metódou. V tabuľke 5.7 uvádzame priemernú veľkosť najlepšieho agenta pre každú konfiguráciu, a tiež p-hodnoty Tukeyho metódy. Krabičkový graf znázorňujúci veľkosť najlepšieho agenta v každom experimente je na obrázku 5.3. Vidíme, že jednopolulačné programovanie

bez blokov vyvinulo najmenších agentov. O niečo väčších agentov vyvinulo viacpopulačné programovanie bez blokov. Za nimi nasledujú konfigurácie s blokmi, pričom opäť jednopopulačná konfigurácia vyvinula menších agentov než viacpopulačná. Z Tukeyho metódy vieme, že všetky tieto rozdiely sú štatisticky významné.

Na rozdiel od zisku agentov, primárnym kritériom pre veľkosť agentov je použitie blokov, pretože obe konfigurácie s blokmi vytvorili väčších agentov než obe konfigurácie bez blokov. Vzhľadom na to, že použitie blokov nemá preukázateľný vplyv na zisk, je tento rozdiel neopodstatnený. Použitie blokových operátorov totiž vytvorí väčších a zložitejších agentov, ale táto zložitosť je zbytočná, pretože takto vytvorení agenti nie sú o nič ziskovejší. Ako sme už uviedli, takýto stav považujeme za nežiadúci.

Sekundárnym kritériom pre veľkosť agentov je počet populácií. Toto kritérium je významné v konfiguráciách bez blokov aj s blokmi. Viacpopulačné konfigurácie vytvorili väčších agentov, a tiež dosiahli väčšie zisky než jednopopulačné. Tento stav interpretujeme tak, že viacpopulačným konfiguráciám sa podarilo nájsť kvalitné stratégie pre obchodovanie, ktoré nebolo možné zakódovať malými stromami. Z tohto dôvodu považujeme rozdiel vo veľkosti agentov medzi jedno- a viacpopulačnými konfiguráciami za opodstatnený, pretože nárast zložitosti je spojený s väčšími ziskmi.

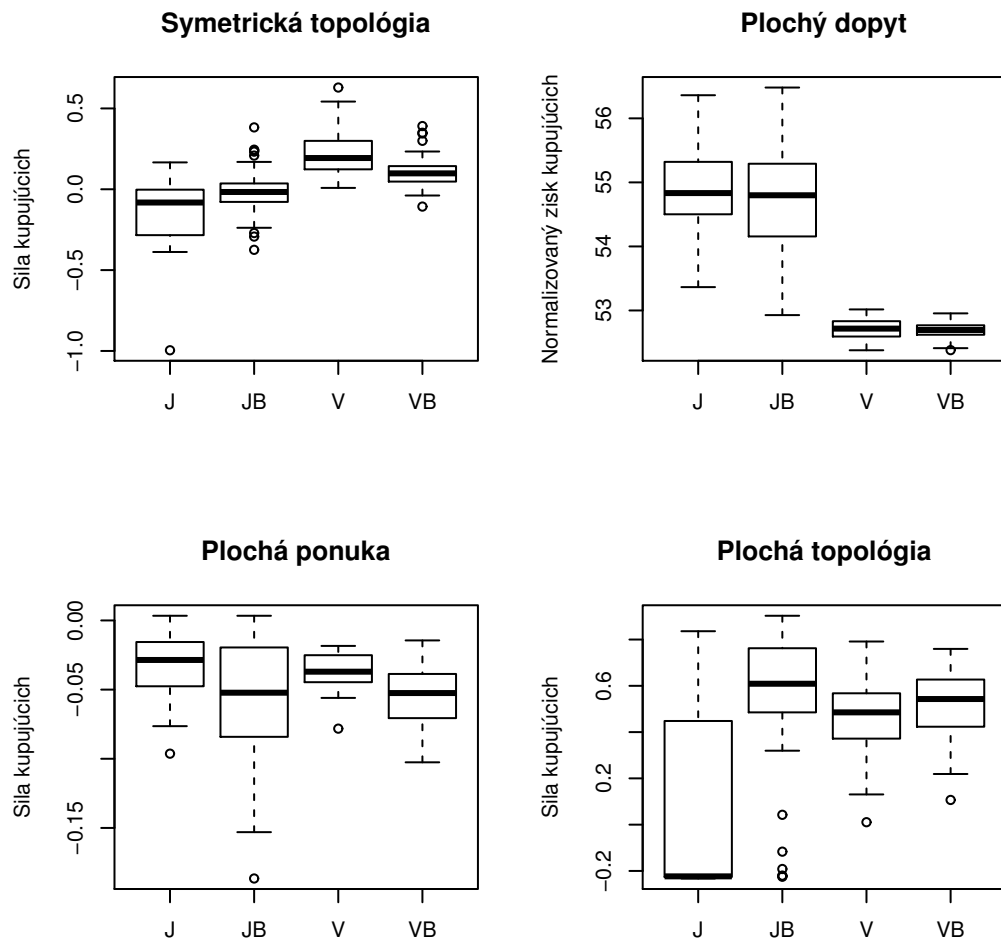
Z nášho experimentu teda celkovo vyplýva, že viacpopulačné programovanie dokáže vytvoriť kvalitnejších agentov než jednopopulačné, a preto ho považujeme za lepšie. Naproti tomu použitie blokov nemá na kvalitu programov žiaden preukázateľný vplyv, ale zbytočne vytvára zložité programy. Preto považujeme použitie blokov za nevhodné.

5.3 Adaptácia na pevné prostredie

V tomto experimente sme použili evolučný model, teda ako kupujúcich sme použili agentov riadených GP a ako predávajúcich heuristiky. Heuristiky predstavujú pevné prostredie, na ktoré sa genetické programy snažia adaptovať. Keďže heuristiky sa nemenia, je potenciálne možné, aby GP odhalili ich slabiny a využili ich vo svoj prospech. Takéto správanie by sme očakávali, keby kupujúci boli ľudia, preto je žiadúce, aby sme ho pozorovali aj v prípade GP.

Hodnotili sme ziskovosť konfigurácií. Ako mieru ziskovosti kupujúcich sme použili ich trhovú silu (s výnimkou topológie s plochým dopytom, kde sme z technických dôvodov použili normovaný celkový zisk, viď nižšie). Kladné hodnoty trhovej sily kupujúcich signalizujú, že kupujúcim sa podarilo získať viac, než by získali v rovnovážnom stave, a teda sa im podarilo vhodne sa adaptovať na danú situáciu. Trhová sila je normovaná, t. j. vyjadruje rozdiel medzi skutočným a rovnovážnym stavom v percentách. Preto je vhodným nástrojom na porovnanie konfigurácii navzájom napriek tomu, že viacpopulačné konfigurácie majú nastavený väčší počet períód a teda majú k dispozícii viac tokenov, čo má za následok väčšie potenciálne zisky. Trhová sila je tiež vhodná preto, že umožňuje na prvý pohľad zistiť, či je rozdiel oproti rovnovážnemu stavu výrazný, čo by napríklad celkový zisk, prípadne rozdiel celkového zisku a potenciálneho zisku v rovnovážnom stave, neumožňoval.

Konfigurácie sme navzájom porovnali Tukeyho metódou, aby sme zistili, či sú niektoré z nich ziskovejšie než iné. V prípadoch, kde to malo zmysel, sme tiež



Obr. 5.4: Krabičkový graf ziskovosti kupujúcich.

pre každú konfiguráciu obojstranným t-testom zisťovali, či je jej ziskovosť významne odlišná od rovnovážneho stavu. Obojstrannú variantu sme použili preto, lebo dopredu nevieme, či bude ziskovosť GP väčšia alebo menšia než ziskovosť v rovnovážnom stave a chceme byť schopní detekovať obidve možnosti.

Experimenty sme vykonali vo všetkých štyroch topológiách trhu. Hlavným dôvodom je, že každá topológia ponúka rôzne možnosti pre kupujúcich zvýšiť svoj zisk nad potenciálny zisk rovnovážneho stavu, a tiež má rôzne úskalia. V ideálnom prípade by mali byť genetické programy schopné adaptovať sa na ľubovoľnú topológiu. Je však možné, že niektorým konfiguráciám sa podarí adaptovať sa len na niektoré topológie. Môže sa dokonca stať, že pre každú topológiu bude vhodná iná konfigurácia, ktorá na danej topológii bude obchodovať lepšie než ostatné.

Experiment sme v každej topológii opakovali 50-krát. Výsledky experimentu sú zobrazené na obrázku 5.4. Numerické zhrnutie a tiež výsledky t-testov sú v tabuľke 5.8. Porovnanie konfigurácii Tukeyho metódou je v tabuľke 5.9.

		J	JB	V	VB
Symetrická topológia	priemer trhovej sily	-0.134	-0.015	0.231	0.110
	p-hodnota	<0.001	0.459	<0.001	<0.001
Plochý dopyt	priemer normovaných ziskov	54.887	54.714	52.706	52.682
Plochá ponuka	priemer trhovej sily	-0.031	-0.057	-0.036	-0.054
Plochá topológia	priemer trhovej sily	0.041	0.550	0.465	0.519
	p-hodnota	0.460	<0.001	<0.001	<0.001

Tabuľka 5.8: Priemery ziskovosti kupujúcich a p-hodnoty t-testov.

	JB	V	VB
J	<0.001	<0.001	<0.001
JB		<0.001	<0.001
V			<0.001

(a) Symetrická topológia

	JB	V	VB
J	0.482	<0.001	<0.001
JB		<0.001	<0.001
V			0.997

(b) Plochý dopyt

	JB	V	VB
J	<0.001	0.801	<0.001
JB		0.002	0.963
V			0.011

(c) Plochá ponuka

	JB	V	VB
J	<0.001	<0.001	<0.001
JB		0.378	0.938
V			0.736

(d) Plochá topológia

Tabuľka 5.9: P-hodnoty rozdielov medzi ziskovosťami kupujúcich.

5.3.1 Symetrická topológia

V tomto experimente používame symetrickú topológiu. Táto topológia je dôležitá predovšetkým preto, že sa často vyskytuje v praxi. Samozrejme, na reálnom trhu nemožno očakávať perfektne symetrickú topológiu, a v tomto zmysle je náš experiment istým zjednodušením reality. Zo všetkých topológií, ktoré používame, je však táto najbližšie skutočnému stavu na trhu.

V symetrickej topológii dopadla najhoršie jednopopulačná konfigurácia bez blokov – v priemere získala o 13% menej než v rovnovážnom stave, a tento rozdiel sa ukázal ako štatisticky významný. O niečo lepšie dopadla jednopopulačná konfigurácia s blokmi, ktorá síce tiež dosiahla záporné hodnoty trhovej sily, ale tieto hodnoty sú malé a nie sú štatisticky významné. Táto konfigurácia teda nedosiahla zisky, ktoré by sa preukázateľne odlišovali od rovnovážneho stavu. Viacpopulačné konfigurácie naopak dosiahli významne vyššie zisky než je rovnovážny stav – konfigurácia s blokmi prekonala rovnovážny stav o 11%, a konfigurácia bez blokov až o 23%. Tukeyho metódou sme zistili, že všetky konfigurácie sa navzájom od seba významne líšia. Je teda preukázateľné, že poradie, v ktorom sme jednotlivé konfigurácie práve vymenovali, je poradie od najhoršej po najlepšiu.

Z nášho experimentu plynie, že v symetrickej topológii je vhodnejšie použiť viacpopulačné programovanie. Toto zistenie je v súlade s výsledkami experimentu 5.2. Na rozdiel od uvedeného experimentu sme však teraz zistili tiež vplyv blokových operátorov na ziskovosť agentov. Je zaujímavé, že tento vplyv závisí na počte populácií. Jednopopulačnej konfigurácii totiž použitie blokov pomohlo, viacpopulačnej to naopak uškodilo.

Môžeme povedať, že ani jednej jednopopulačnej konfigurácii sa nepodarilo adaptovať sa na pevné prostredie reprezentované heuristikami. Takéto správanie je nežiadúce, a preto tieto konfigurácie považujeme za nevhodné ekonomické

modely reálnych obchodujúcich. Naopak viacpopulačné konfigurácie sa na toto prostredie adaptovali veľmi dobre, a preto ich použitie môžeme odporučiť.

5.3.2 Plochý dopyt

Ako sme uviedli v sekcii 5.1.2, v topológii s plochým dopytom je použitie trhovej sily kupujúcich nevhodné, pretože trhová sila je veľmi citlivá na parametre experimentu. Preto sme v tomto experimente namiesto trhovej sily kupujúcich merali ich celkový zisk. Pretože viacpopulačné konfigurácie obchodujú vo viacerých periódach než jednopopulačné, môžu predať viac tokenov a preto môžu ľahšie dosiahnuť veľký zisk. Aby sme tento fakt zohľadnili v porovnávaní konfigurácii medzi sebou, celkový zisk sme normovali. Vzhľadom na to, že celkový počet rezervačných cien, ktoré majú agenti k dispozícii na jednu periódu, je rovnaký pre všetky konfigurácie, pri normovaní by stačilo vydeliť zisk počtom periód. Z interpretačného hľadiska sa nám však zdalo vhodnejšie vydeliť celkový zisk celkovým počtom rezervačných cien pripadajúcich na jedno kolo.² Takto totiž dostaneme priemerný zisk, ktorý agenti vyprodukovali z jednej rezervačnej ceny.

Keďže potenciálny zisk je (takmer) nulový, akýkoľvek kladný zisk je pre kupujúcich úspech. Nemá preto zmysel porovnávať realizovaný zisk kupujúcich s rovnovážnym stavom. V tomto experimente sme preto len konfigurácie porovnávali medzi sebou.

Jednopopulačné konfigurácie dosiahli priemerný zisk 55 na jednu rezervačnú cenu, zatiaľ čo viacpopulačné konfigurácie dosiahli priemerný zisk 53 na jednu rezervačnú cenu. Ukázalo sa, že tento rozdiel je štatisticky významný. Naproti tomu rozdiel medzi konfiguráciou bez blokov a zodpovedajúcou konfiguráciou s blokmi je nevýznamný. Použitie blokov teda nemalo žiaden vplyv na kvalitu agentov, naproti tomu počet konfigurácií vplyv mal. Je zaujímavé všimnúť si, že vplyv počtu konfigurácií je opačný než v predchádzajúcich experimentoch – zatiaľ čo doteraz boli lepšie viacpopulačné konfigurácie, v tomto experimente sa ukázali ako lepšie jednopopulačné. Vidíme teda, že kvalita agentov vyvinutých danou konfiguráciou závisí na použitej topológii trhu.

5.3.3 Plochá ponuka

Topológia s plochou ponukou je obrátenou verziou topológie s plochým dopytom. Potenciálny zisk predávajúcich je veľmi malé kladné číslo. Keďže predávajúci nemôžu obchodovať so záporným ziskom a súčet realizovaných ziskov kupujúcich a predávajúcich nemôže presiahnuť súčet potenciálnych ziskov, môžu kupujúci prekonať rovnovážny stav iba o kúsok. V tejto topológii je však už samotné dosiahnutie (a nie ešte prekonanie) rovnovážneho stavu komplikované, pretože predávajúci by v takom prípade museli obchodovať s takmer nulovými ziskmi, čo zrejme nebudú ochotní urobiť. Kupujúci sa teda budú musieť snažiť prinútiť predávajúcich obchodovať s čo najmenšími ziskmi, aby sa aspoň priblížili rovnovážnemu stavu. Keďže dosiahnutie aj prekonanie rovnovážneho stavu je veľmi ťažké, v tejto topológii sme agentov s rovnovážnym stavom neporovnávali. Iba

²Tento počet dostaneme vynásobením počtu rezervačných cien v jednej perióde počtom periód.

sme porovnávali jednotlivé konfigurácie medzi sebou podobne, ako sme to urobili v topológii s plochým dopytom.

Všetky konfigurácie dosiahli trhovú silu blízku 0. Takýto stav je žiaduci a znamená, že GP sa podarilo prinútiť predávajúcich obchodovať s minimálnymi ziskmi. Najlepšie dopadla jednopopulačná konfigurácia bez blokov, ktorá získala len o 3% menej než v rovnovážnom stave. Za ňou nasleduje viacpopulačná konfigurácia bez blokov, ktorá získala o 4% menej než v rovnovážnom stave. Rozdiel medzi týmito dvomi konfiguráciami nie je významný. Viacpopulačná konfigurácia s blokmi dosiahla o 5% menšie zisky než v rovnovážnom stave, a je štatisticky významne horšia než predchádzajúce dve. Najhoršie dopadla jednopopulačná konfigurácia s blokmi, ktorá získala o 6% menej než v rovnovážnom stave, čo je významne menej než získali konfigurácie bez blokov, ale nie je to významne odlišné od trhovej sily viacpopulačnej konfigurácie s blokmi.

Na rozdiel od predchádzajúcich výsledkov sú v tomto experimente primárnym kritériom blokové operátory, pričom ich použitie vedie ku zhoršeniu ziskovosti agentov. Počet populácií je až sekundárnym kritériom, ktoré navyše nie je štatisticky významné.

5.3.4 Plochá topológia

Plochá topológia je zo všetkých prípadov najzaujímavejšia. V tomto prípade sa totiž ponuka a dopyt nikde nepretnú (a to ani v nediskretizovanej topológii). Rovnovážna cena je v tejto topológii podľa našej definície v strede medzi rezervačnými cenami predávajúcich a kupujúcich, takže v rovnovážnom stave by si kupujúci a predávajúci rozdelili zisk napoly. Nie je ale žiaden dôvod, prečo by mala cena, za ktorú sa bude obchodovať, konvergovať práce do tohto stredu. Obchodovanie môže prebiehať za ľubovoľnú cenu medzi rezervačnými cenami predávajúcich a kupujúcich. Zisk agentov teda závisí len na ich šikovnosti a nie je žiadnym spôsobom limitovaný topológiou trhu.

V plochej topológii dopadla najhoršie jednopopulačná konfigurácia bez blokov, ktorá dosiahla o 4% viac než v rovnovážnom stave. Táto hodnota ale nie je významne odlišná od nuly, a preto konštatujeme, že GP sa nepodarilo v tejto konfigurácii prekonať heuristiky. Druhá najhoršia skončila viacpopulačná konfigurácia bez blokov so 47%, nasleduje viacpopulačná konfigurácia s blokmi s 52%, a najlepšie dopadla jednopopulačná konfigurácia s blokmi s 55%. Zisky týchto troch konfigurácií sú významne vyššie než v rovnovážnom stave, teda môžeme povedať, že GP sa v tomto prípade vhodne adaptovalo na dané prostredie. Čo sa týka porovnania medzi konfiguráciami, jednopopulačná konfigurácia bez blokov dosiahla významne menšie zisky než všetky ostatné. Ostatné rozdiely vyšli nevýznamne. V tejto topológii teda skončili najlepšie obe viacpopulačné konfigurácie a jednopopulačná konfigurácia s blokmi, pričom rozdiely medzi nimi už nie je možné na základe nameraných dát porovnať.

5.3.5 Záver

Pri skúmaní vhodnosti použitia jednotlivých konfigurácií sa ukazuje, že jednopopulačná konfigurácia bez blokov je v symetrickej aj v plochej topológii vyslovene nevhodná. V plochej topológii sa totiž jej trhovú silu nelíšila významne od nuly,

v symetrickej topológii bola dokonca preukázateľne záporná. Tejto konfigurácii sa teda vo vymenovaných topológiách nepodarilo adaptovať na pevné prostredie. Naopak jednopopulačnej konfigurácii s blokmi sa v symetrickej aj v plochej topológii adaptovať podarilo. V tomto prípade teda použitie blokov viedlo k lepším výsledkom. Všetky konfigurácie s výnimkou jednopopulačnej bez blokov v uvedených topológiách dosiahli významne kladné hodnoty trhovej sily, a je teda možné považovať ich za vhodné modely reálnych agentov. Vo zvyšných topológiách testovanie nenulovosti trhovej sily nedáva zmysel, a preto o „vhodnosti“ z hľadiska trhovej sily v týchto topológiách nemôžeme hovoriť. Vo všetkých topológiách však na základe porovnania medzi konfiguráciami môžeme povedať, ktorá konfigurácia je najlepšia.

Ukázalo sa, že pre každú topológiu je vhodná iná konfigurácia genetického programovania. V symetrickej topológii dopadla najlepšie viacpopulačná konfigurácia bez blokov, v topológii s plochým dopytom boli najlepšie obe jednopopulačné konfigurácie, v topológii s plochou ponukou to zas boli obe konfigurácie bez blokov, a nakoniec v plochej topológii skončili najlepšie všetky konfigurácie s výnimkou jednopopulačnej bez blokov. Z hľadiska trhovej sily môžeme tieto konfigurácie v uvedených topológiách považovať za najlepšie modely reálnych obchodujúcich.

Za zmienku ešte stojí, že jednopopulačná konfigurácia je síce v symetrickej aj plochej topológii najhoršia a, ako sme uviedli, vyslovene nevhodná, v topológiách s plochou ponukou a plochým dopytom patrí naopak do skupiny najlepších. Tento fakt asi najlepšie ilustruje, ako veľmi sú výsledky obchodovania závislé na použitej topológii.

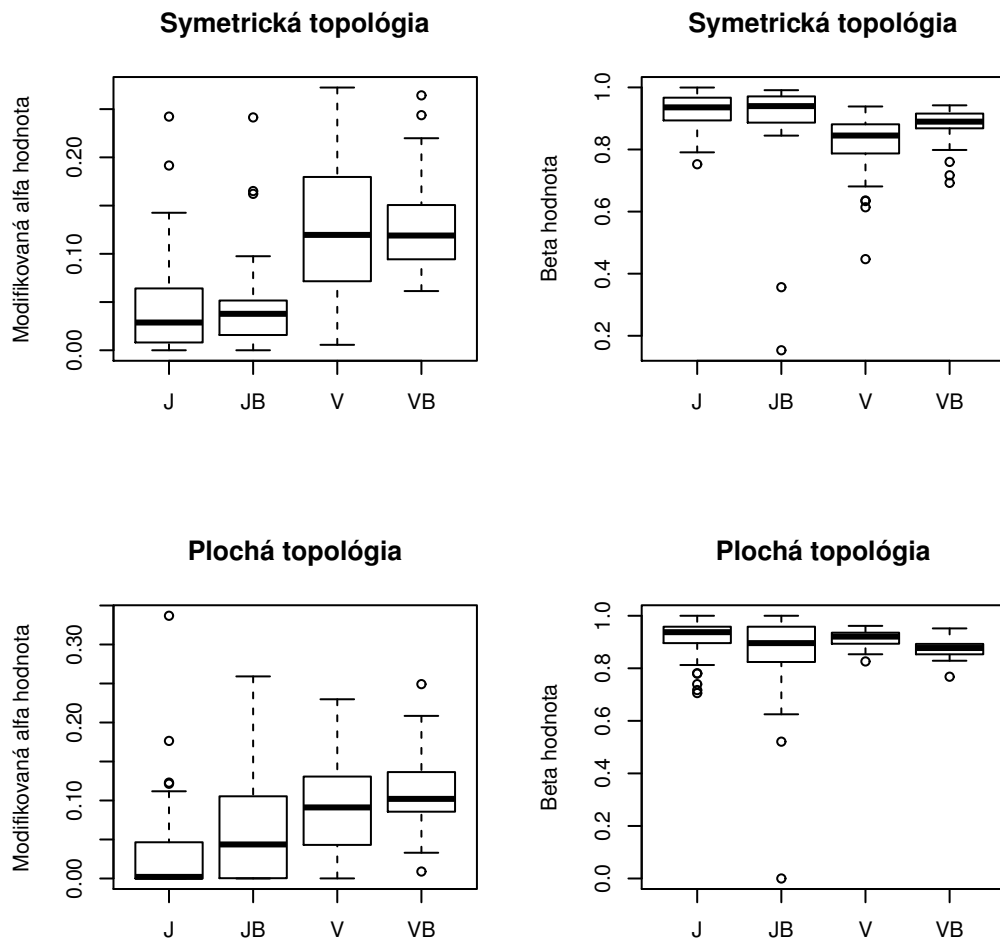
Správanie genetického programovania teda preukázateľne závisí na použitej topológii trhu a výber konfigurácie GP je preto nutné prispôbiť výberu topológie.

5.4 Overenie vlastností koevolučného modelu

V tomto experimente sme použili koevolučný model, teda kupujúci aj predávajúci boli vyvíjaní genetickým programovaním. Zisťovali sme, či takýto model spĺňa teoretické vlastnosti, ktoré sú známe o dvojitej aukcii. Zamerali sme sa na zákon jednej ceny a efektivitu trhu.

Platnosť zákona jednej ceny sme merali pomocou modifikovanej alfa hodnoty. Ak je modifikovaná alfa hodnota veľká, znamená to, že niektorí kupujúci nakupujú príliš draho v porovnaní s ostatnými. Takýto kupujúci si nie sú schopní uvedomiť, že existujú predávajúci ochotní predať lacnejšie. Podobne niektorí predávajúci musia v takejto situácii predávať príliš lacno, hoci existujú kupujúci ochotní zaplatiť viac. Neplatnosť zákona jednej ceny teda znamená, že agenti nie sú schopní plne využiť informácie o trhu, ktoré majú k dispozícii. Konkrétne nevyužívajú informáciu o cenách, za ktoré obchodovali ostatní agenti.

Ako mieru efektivity trhu sme použili beta hodnotu. Beta hodnoty blízke jednej znamenajú, že trh realizoval takmer všetok svoj potenciálny zisk. Naopak nízke beta hodnoty znamenajú, že niektorí agenti nevyužili všetky svoje potenciálne ziskové tokeny. Takáto situácia môže znamenať, že kupujúci aj predávajúci sa snažia obchodovať príliš agresívne s veľkými požiadavkami na svoj zisk. To vedie k tomu, že sa nevedia dohodnúť na spoločnej cene a tak nakoniec k obcho-



Obr. 5.5: Krabičkové grafy modifikovanej alfa hodnoty a beta hodnoty.

du nedôjde a daní agenti majú z daného tokenu nulový zisk. V tomto prípade by bolo pre oboch výhodné, aby aspoň jeden z nich ustúpil a znížil požiadavku na svoj zisk, čo by viedlo k úspešnému obchodu, na ktorom by obaja získali.

Použili sme všetky štyri konfigurácie GP, pričom sme vždy nechali obchodovať proti sebe tú istú konfiguráciu. Experiment sme uskutočnili v symetrickej a v plochej topológii. Ako sme už uviedli, symetrická topológia je zo všetkých najrealistickejšia, preto sú výsledky experimentov v tejto topológii najrelevantnejšie. Topológie s plochým dopytom resp. plochou ponukou sú problematické v tom, že pri maximálnej efektívite trhu by všetci kupujúci resp. predávajúci museli mať takmer nulové zisky, čo zrejme nebudú ochotní urobiť. Nie je preto jasné, aké výsledky by pre tieto topológie boli žiadúce. Plochá topológia je však zaujímavá tým, že sa v nej ponuka a dopyt nepretnú, a hoci je rovnovážna cena definovaná ako stred medzi rezervačnými cenami predávajúcich a kupujúcich, neexistuje mechanizmus, ktorý by obchodovanie na trhu „tlačil“ do tohto stredu. Preto dodržanie zákona jednej ceny môže byť komplikované.

Experiment sme v každej topológii opakovali 50-krát. Namerané hodnoty sú zhrnuté v tabuľke 5.10 a sú tiež zobrazené na obrázku 5.5. V symetrickej topológii dosiahli jednopopulačné konfigurácie priemerné modifikované alfa hodnoty

		J	JB	V	VB
Symetrická topológia	modifikovaná α hodnota	0.044	0.044	0.123	0.128
	β hodnota	0.921	0.905	0.820	0.884
Plochá topológia	modifikovaná α hodnota	0.035	0.065	0.094	0.109
	β hodnota	0.911	0.858	0.915	0.874

Tabuľka 5.10: Priemerné hodnoty α a β .

	JB	V	VB
J	>0.999	<0.001	<0.001
JB		<0.001	<0.001
V			0.973

(a) P-hodnoty rozdielov modifikovaných alfa hodnôt v symetrickej topológii.

	JB	V	VB
J	0.061	<0.001	<0.001
JB		0.069	0.002
V			0.617

(b) P-hodnoty rozdielov modifikovaných alfa hodnôt v plochej topológii.

Tabuľka 5.11: Upravené p-hodnoty rozdielov modifikovaných alfa hodnôt.

0.044. To znamená, že odchýlky od priemernej ceny boli zhruba na úrovni 4%. Toto je dosť malé číslo a môžeme povedať, že nedošlo k žiadnemu významnému porušeniu zákona jednej ceny. Naproti tomu v prípade viacpopulačných konfigurácií boli odchýlky na úrovni 12-13%, čo už signalizuje mierne porušenie zákona jednej ceny. Na základe porovnania Tukeyho metódou, ktorého výsledky sú v tabuľke 5.11, vidíme, že rozdiel medzi jednopopulačnými a viacpopulačnými konfiguráciami je štatisticky významný. Môžeme teda povedať, že viacpopulačné konfigurácie sa v symetrickej topológii dopustili väčšieho porušenia zákona jednej ceny než jednopopulačné.

V plochej topológii je situácia podobná, ale jednopopulačná konfigurácia s blokmi má modifikované alfa hodnoty o niečo väčšie než jednopopulačná konfigurácia bez blokov. Následkom toho je štatisticky neodlíšiteľná nie len od jednopopulačnej konfigurácie bez blokov, ale aj od viacpopulačnej konfigurácie bez blokov. V tejto konfigurácii teda predeľ medzi dobrými (t. j. malými) a zlými modifikovanými alfa hodnotami nie je jasný, pretože jednopopulačná konfigurácia s blokmi leží na hranici.

Z tabuľky 5.10 tiež vidíme, že v experimentoch boli priemery beta hodnôt v rozmedzí približne 82-92%. Takéto beta hodnoty sú menšie než zvyčajné beta hodnoty pozorované v experimentoch s ľuďmi. Napríklad [10] uvádza experimenty v 5 topológiách trhu, pričom v 4 z nich dosiahli ľudia priemerné beta hodnoty nad 99% a v piatej bol priemer beta hodnôt 90%. Z toho vyplýva, že najväčšia efektivita obchodovania, akú sme dosiahli v našich experimentoch, je zhruba

	JB	V	VB
J	0.833	<0.001	0.190
JB		<0.001	0.654
V			0.004

(a) P-hodnoty rozdielov beta hodnôt v symetrickej topológii.

	JB	V	VB
J	0.015	0.998	0.159
JB		0.009	0.791
V			0.109

(b) P-hodnoty rozdielov beta hodnôt v plochej topológii.

Tabuľka 5.12: Upravené p-hodnoty rozdielov beta hodnôt.

rovnaká ako najmenšia efektivita dosahovaná v experimentoch s ľuďmi. Na obrázku 5.5 tiež vidíme veľké množstvo odľahlých pozorovaní, ktoré majú o mnoho menšiu beta hodnotu než priemer. Z toho plynie, že v mnohých prípadoch bola efektivita obchodovania výrazne nižšia než usudzujeme z priemeru, a v týchto prípadoch bola teda efektivita výrazne nižšia než sa zvyčajne dosahuje v experimentoch s ľuďmi.

Rozdiely medzi beta hodnotami sme opäť vyhodnotili Tukeyho metódou. Upravené p-hodnoty sú zhrnuté v tabuľke 5.12. V symetrickej topológii dosiahla viacpopulačná konfigurácia bez blokov významne nižšiu beta hodnotu než všetky ostatné konfigurácie. V plochej topológii naopak jednopopulačná konfigurácia s blokmi dosiahla významne nižšiu beta hodnotu než obe konfigurácie bez blokov. Všetky ostatné rozdiely boli nevýznamné.

Z hľadiska efektivity trhu sa ukázalo, že genetické programovanie je o niečo horšie, než by sme čakali v prípade reálnych agentov. Tento nedostatok však nie je veľký. Tukeyho metóda ukázala, že rozdiely medzi konfiguráciami zväčša nie sú významné, preto je pri ekonomických simuláciách možné použiť ľubovoľnú konfiguráciu.

Záver

V tejto práci sme skúmali genetické programovanie z hľadiska jeho použiteľnosti vo výpočtovej ekonómii. Pracovali sme so 4 rôznymi konfiguráciami GP. Zisťovali sme, či tieto konfigurácie majú vlastnosti, ktoré sú vyžadované od agentov používaných v ekonomických simuláciách. Okrem toho sme zisťovali, či niektoré konfigurácie genetického programovania majú lepšie vlastnosti než ostatné, aby sme tieto konfigurácie mohli odporučiť pre použitie vo výpočtovej ekonómii.

Zistili sme, že genetické programovanie dokáže vo väčšine prípadov vyprodukovať agentov, ktorí dokážu poraziť ľuďmi naprogramované heuristiky. Takisto sa ukázalo, že väčšina konfigurácií je schopná adaptovať sa. Pri overovaní zákona jednej ceny sme zistili jeho mierne porušenie viacpopulačnými konfiguráciami, toto porušenie však nebolo nijako zásadné. Čo sa týka efektivity obchodovania, genetické programovanie síce obchoduje s menšou efektivitou než ľudia, ale opäť sa nejedná o veľký rozdiel. Z týchto zistení usudzujeme, že genetické programovanie dokáže vytvoriť vhodných agentov pre ekonomické simulácie.

Na základe porovnania medzi konfiguráciami usudzujeme, že viacpopulačné konfigurácie sú vo všeobecnosti lepšie než jednopopulačné. Toto tvrdenie podporujeme tým, že jednopopulačná konfigurácia (predovšetkým tá bez blokov) mala problém adaptovať sa na pevné prostredie, a mala tiež problém prekonať heuristiky. Je síce pravda, že v niektorých experimentoch dopadli naopak jednopopulačné konfigurácie lepšie než viacpopulačné, ale tieto rozdiely boli väčšinou malé. Navyše viacpopulačné konfigurácie majú lepšiu ekonomickú interpretáciu než jednopopulačné. Z uvedených dôvodov odporúčame v ekonomických experimentoch používať viacpopulačné konfigurácie.

V otázke použitia blokov sú výsledky nejednoznačné. Vo väčšine prípadov použitie blokov nemalo zásadný vplyv na výsledky experimentov. Našli sa však aj prípady, keď použitie blokov pomohlo, ale naopak v niektorých prípadoch použitie blokov viedlo ku zhoršeniu výsledkov. Nie je preto možné jednoznačne povedať, či je lepšie používať konfigurácie s blokmi alebo bez nich.

V experimentoch, ktoré sme popísali v našej práci, by sa dalo pokračovať. Jedným zo zaujímavých experimentov by bolo otestovanie genetického programovania v nelineárnych topológiách. Rovnako zaujímavé by mohlo byť obchodovanie s rôznym počtom predávajúcich a kupujúcich. Napríklad by bolo možné simulovať monopol, t. j. trh s jedným predávajúcim a veľkým množstvom kupujúcich. Taktiež je možné porovnať genetické programovanie s inými formami umelej inteligencie. Toto porovnanie je možné urobiť priamo, t. j. nechať proti sebe obchodovať genetické programovanie a inú formu umelej inteligencie. Je tiež možné porovnávať nepriamo, napríklad nechať obchodovať genetické programovanie a inú umelú inteligenciu proti heuristikám.

Literatúra

- [1] ANDĚL, J. *Základy matematické statistiky*, s. 209–210. Matfyzpress, 2., opr. vyd., 2007.
- [2] ANDREWS, M. – PRAGER, R. Genetic programming for the acquisition of double auction market strategies. *Advances in genetic programming*. 1994, 1, s. 355–368.
- [3] ARUNACHALAM, R. – SADEH, N. M. The supply chain trading agent competition. *Electronic Commerce Research and Applications*. 2005, 4, 1, s. 66–84.
- [4] AXELROD, R. The evolution of strategies in the iterated prisoner’s dilemma. *The dynamics of norms*. 1987, s. 1–16.
- [5] AXELROD, R. M. *The evolution of cooperation*. Basic books, 2006.
- [6] CHEN, S.-H. Varieties of agents in agent-based computational economics: A historical and an interdisciplinary perspective. *Journal of Economic Dynamics and Control*. 2012, 36, 1, s. 1–25.
- [7] CHEN, S.-H. Toward an agent-based computational modeling of bargaining strategies in double auction markets with genetic programming. In *Intelligent Data Engineering and Automated Learning–IDEAL 2000. Data Mining, Financial Engineering, and Intelligent Agents*, s. 517–531. Springer, 2000.
- [8] CHEN, S.-H. – TAI, C.-C. Trading restrictions, price dynamics and allocative efficiency in double auction markets: Analysis based on agent-based modeling and simulations. *Advances in Complex Systems*. 2003, 6, 03, s. 283–302.
- [9] DAWID, H. On the convergence of genetic learning in a double auction market. *Journal of Economic Dynamics and Control*. 1999, 23, 9, s. 1545–1567.
- [10] GODE, D. K. – SUNDER, S. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*. 1993, 101, 1, s. 119.
- [11] GOLDBERG, D. E. et al. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Reading Menlo Park, 1989.
- [12] KOZA, J. R. *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992.

- [13] KOZA, J. R. *Genetic programming II: automatic discovery of reusable programs*. MIT press, 1994.
- [14] MICHALEWICZ, Z. *Genetic algorithms+ data structures= evolution programs*. Springer, 1996.
- [15] NICOLAISEN, J. – PETROV, V. – TESHATSIAN, L. Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *Evolutionary Computation, IEEE Transactions on*. 2001, 5, 5, s. 504–523.
- [16] PARSONS, S. et al. Everything you wanted to know about double auctions, but were afraid to (bid or) ask. *Draft Paper, Department of Computer & Information Science, Brooklyn College [Online] Dostupné z: <http://www.sci.brooklyn.cuny.edu/~parsons/projects/mech-design/publications/cda.pdf>*. 2006.
- [17] ROTH, A. E. – EREV, I. Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and economic behavior*. 1995, 8, 1, s. 164–212.
- [18] RUST, J. – MILLER, J. H. – PALMER, R. Characterizing effective trading strategies: Insights from a computerized double auction tournament. *Journal of Economic Dynamics and Control*. 1994, 18, 1, s. 61–96.
- [19] WELLMAN, M. P. et al. The 2001 trading agent competition. *Electronic Markets*. 2003, 13, 1, s. 4–12.