Väzenská dilema
(evolúcia stratégií hry)

Predmet: Evolučné algoritmy
Vypracoval: Radoslav Tausinger
Email: 5tausinger@st.fmph.uniba.sk
Matematicko-fyzikálna fakulta UK
Zimný semester 1999/2000

Úloha

Zobrazte evolúciu stratégií hry väzenská dilema, ktorá je modelovaná genetickým algoritmom, pričom každá dvojica väzňov bude interagovať vždy 10-krát medzi sebou. Tabuľka platieb je

	kooperuje (C)	nekooperuje (D)
Kooperuje (C)	3,3	0,5
Nekooperuje (D)	5,0	1,1

Napríklad keď prvý väzeň nekooperuje (D) a druhý väzeň kooperuje (C), prvý dostane 5 bodov a druhý 0 bodov. Väzni majú pamäť veľkosti 1, pamätajú si iba posledný ťah súpera. Napríklad stratégia CCD (chromozóm o troch zložkách) znamená: v prvom kroku agent kooperuje (1. položka v chromozóme je C), v ďalších ťahoch kooperuje (2. zložka je C), ak v predchádzajúcom ťahu súper kooperoval, alebo nekooperuje (3. zložka je D), ak súper v predchádzajúcom ťahu nekooperoval. Pri populácii 1000 urobte vždy náhodne 500 skupín po 10, nechajte ich všetky po dvojiciach interagovať medzi sebou, s tým, že sa každému budú sčitovať platby a zoberte do ďalšej generácie iba 2 najlepších. Výsledkom má byť graf, kde na osi x bude číslo generácie (teda čas) a na osi y početnosť zástupcov tej ktorej stratégie v generácii.

Teória hier

Problémami ako Väzenská dilema sa zaoberá špeciálna vedná disciplína teória hier. Teória hier vo všeobecnosti rozlišuje dva druhy hier: hry s nulovým súčtom a hry s nenulovým súčtom. V prvom type hier sa jedná o hry typu výhra-prehra, teda ak jeden hráč vyhrá, druhý musí nutne prehrať a naopak. K tomuto druhu hier by sme mohli zaradiť napríklad hru Nim, kde hráči striedavo odoberajú z kôpky 1, 2 alebo 3 zápalky a vyhráva ten, ktorý zoberie poslednú zápalku. Zo zložitejších hier sa sem dá priradiť napríklad šach. Na vysvetlenie názvu by som dodal, že hráč buď vyhrá (+1b) alebo prehrá (-1b), čo dáva vždy celkový súčet 0 (pri remíze môžeme každému hráčovi udeliť 0 bodov). Druhý typ sa vyznačuje tým, že takého pravidlo o nulovom (ani konštantnom) súčte výsledkov neplatí. A práve sem patrí aj Väzenská dilema, v ktorej jednotlivé súčty sú 2 (pri obojstrannej kooperácii), 5 (pri jednostrannej kooperácii) respektíve 6 (pri obojstrannej nekooperácii). Práve Väzenská dilema, ktorá bola prvýkrát formulovaná v roku 1950, patrí k najskorším, pomerne jednoduchým hrám, ktoré majú navyše aj množstvo aplikácií v reálnom svete.

Teória hier s nenulovým súčtom má blízko k teórii rozhodovania. Problémy riešené teóriou rozhodovania sa vyznačujú väčším počtom alternatív na riešenie nejakého problému a väčším počtom udalostí ovplyvňujúcich výsledky jednotlivých alternatív. Týmto udalostiam sú priradené pravdepodobnosti a na základe toho sa dá rozhodnúť, ktorá stratégia je najlepšia. Ako príklad možno uviesť farmára, ktorý sa má rozhodnúť, či si dá alebo nedá zaviesť zavlažovací systém. Rozhodujúcim faktorom je počasie, u ktorého treba odhadnúť pravdepodobnosti a na základe toho sa rozhodnúť, či zavlažovanie má zmysel. Teória hier však prináša nový prvok, tým je druhý hráč (prípadne aj tretí, štvrtý, atď.). Tu už odhadovanie pravdepodobností je nepomerne zložitejšie, lebo odhadujeme, ako sa zachová iný živý človek.

Klasická väzenská dilema

Vráťme sa však k Väzenskej dileme. Príbeh je o dvoch zločincoch, ktorí sú zatknutí za menší priestupok, za ukradnutie auta. Policajti ich po prichytení pri čine zatknú, odvedú na policajnú stanicu a umiestnia do oddelených miestností. Zločinci teda nemôžu medzi sebou komunikovať. Detektív však verí, že obaja sú vinný z väčšieho priestupku, z vykradnutia banky. Z toho dôvodu im obom nechá možnosť udať toho druhého ako zločinca, ktorý vykradol banku (možnosť D, defect alebo nekooperovať so svojím spoločníkom) alebo ho neudať (možnosť C, cooperate alebo kooperovať so svojím spoločníkom). Ak ani jeden neprehovorí (možnosť CC), detektív im prišije krádež auta a obaja si pôjdu sadnúť na jeden rok. Ak obaja nahlásia toho druhého (možnosť DD), obaja dostanú po tri roky. Ak prvý udá druhého, no druhý neudá prvého (možnosť DC), prvý bude odmenený za spoluprácu s políciou slobodou a ten druhý si pôjde odsedieť päť rokov. Možnosť CD funguje presne naopak ako možnosť DC. Obaja teda stoja pred rovnakým problémom, udať či neudať.

Klasická dilema	kooperuje (C)	nekooperuje (D)
Kooperuje (C)	1,1	5,0
Nekooperuje (D)	0,5	3,3

Väzenská dilema, teda spĺňa jednu dôležitú podmienku pre preferencie výberu alternatív:

- DC>CC>DD>CD

Ako sa teda rozhodnúť? Obaja majú sebecký cieľ, a to stráviť vo väzení čo najkratší čas. Predstavme si, že prvý zločinec vidí do cely, v ktorej sedí druhý a počuje, ako sa ten rozhoduje. Nech je teda druhý „správnym“ zločincom a bude zatĺkať, a teda detektívovi nič nepovie. V tom prípade si prvý vyberá medzi slobodou a jedným rokom väzenia, čo ho pri maximalizácii svojho úžitku vedie k spolupráci s políciou. Predpokladajme pre zmenu, že druhý zločinec by chcel ostať slobodný, a tak udá prvého. Čo má robiť prvý v tomto prípade? Rozhoduje sa medzi tromi a piatimi rokmi vo väzení, samozrejme tri roky sú prijateľnejšie, takže opäť udá toho druhého. V oboch prípadoch sme teda boli svedkami, že prvému sa viac oplatí toho druhého udať. Obdobne však uvažuje aj druhý väzeň, čo vedie k stavu DD. Tento stav sa nazýva rovnovážny. Vznik tohoto rovnovážneho stavu umožnila vyššie uvedená podmienka preferencií. Tragédiou však je, že rovnovážny stav DD je horší ako stav CC pre oboch hráčov. Obaja by si teda dosť polepšili, ak by prešli do stavu CC. Bohužiaľ však neexistuje dôvod, ktorý by ich do toho stavu viedol.

Nie každá hra má rovnovážny stav, v nasledujúcej tabuľke uvádzam pre porovnanie rôzne druhy hier. Pod názvom hry je uvedené poradie preferencií možností.

Prisoner's Dilemma (PD)	Chicken (CH)	Stag Hunt (SH)	Deadlock (DL)
DC	DC	CC	DC
CC	CC	DC	DD
DD	CD	DD	CC
CD	DD	CD	CD

Spolupráca, o ktorej sme dokázali jej výhodnosť sa o mnoho jednoduchšie dosahuje v hre typu SH a je najskôr nedosiahnuteľná v hre typu DL.

Iterovaná väzenská dilema

Klasická väzenská dilema sa hrá raz. Obaja väzni sa musia rozhodnúť bez toho, aby sa mohli dohovoriť. Skúsme trošku zvoľniť tieto podmienky a predpokladajme, že títo dvaja sú nepoučiteľní a dostanú sa do podobnej situácie mnohokrát po sebe. Toto by snáď mohlo viesť k tomu, že sa obaja poučia a dostanú sa z nevýhodného rovnovážneho stavu DD do výhodnejšieho stavu CC. Z uvedeného dôvodu autori obyčajne pridávajú pri iterovanej verzii novú podmienku pre preferencie:

- CC>(DC+CD)/2

Táto podmienka zabezpečuje, že zatknutým sa viac oplatí spolupracovať, ako sa striedavo udávať a brať na seba vinu.

Zdá sa teda, že z dlhodobého hľadiska je pre oboch výhodné spolupracovať. Obaja sa totiž boja, že v nasledujúcom kroku ich ten druhý udá, čo spôsobuje príklon k spolupráci. Náklonnosť k spolupráci sa dá meniť zmenami v podmienkach úlohy. Predstavme si, že zločin, z ktorého ich detektív upodozrieva, nie je vykradnutie banky, ale brutálna vražda, potom DC znamená slobodu, CC 1 rok, DD 10 rokov a CD 20 rokov. Tu je hrozba, že moja dobrá vôľa nebude opätovaná až príliš veľká, takže spoločníci budú menej náklonní k spolupráci. Pritvrdiť by sa dalo v niektorých amerických štátoch, kde je možný aj trest smrti. Naopak podmienky sa dajú aj zjemniť a navodiť vhodnejšiu situáciu pre kooperáciu.

Stratégie

Ak sa hra neustále opakuje, hráč si môže vybrať v podstate z neobmedzeného množstva stratégií. V ďalšom rozoberiem najzákladnejšie z nich.

Náhodná stratégia: Hráč si pri každej príležitosti hodí mincou a podľa toho sa rozhodne, či bude alebo nebude spolupracovať. Táto stratégia je jednoduchá na prevedenie, no nedá sa od nej očakávať, že bude miminalizovať hráčov čas strávený vo väzení. Protihráča totiž náhodné udávania unavia a po určitom čase prestane byť ochotný spolupracovať.

Kooperatívna stratégia: Jedná sa vlastne o stratégiu „kto do teba kameňom, ty doňho chlebom“. Hráč sa teda vôbec nezaujíma o súperove ťahy, ale jednoducho vždy spolupracuje. Výhodou tejto stratégie je presvedčenie protihráča o svojich dobrých úmysloch, čo spôsobí, že sa prestane báť najväčšej hrozby

CD a bude viac náklonný k spolupráci. Nevýhodou je, ak si protihráč overí, že hráč používa čisto kooperatívnu stratégiu, môže sa rozhodnúť využiť situáciu a vyťažiť pre seba maximum.

Nekooperatívna stratégia: Jedná sa vlastne o stratégiu „kto do teba chlebom, ty doňho kameňom“. Hráč nikdy s druhým nespolupracuje. Táto stratégia sa zvykne aplikovať hlavne pri veľmi nevýhodnom CD, keď je to až príliš penalizované. Nevýhody tejto stratégie sú zrejmé. Jednak znemožňuje nastolenie obojstranne výhodného stavu CC a potom v podstate núti súpera, aby adoptoval rovnakú stratégiu, čo vedie k rovnovážnemu stavu DD.

Stratégia Tit-for-Tat (TFT): V tejto stratégii hráč spolupracuje v prvom kroku a potom vždy opakuje predošlý ťah súpera. Niekoľko rôznych štúdií ukázalo, že práve táto stratégia maximalizuje spokojnosť hráča. Robert Axelrod z University of Michigan zorganizoval v roku 1981 súťaž každý s každým rôznych algoritmov, ktoré dodali vedci z rôznych amerických škôl. Najprv sa turnaja zúčastnilo okolo 20 stratégií, z ktorých práve TFT vyšla víťazne. V neskoršom kole sa už zišlo nad 60 stratégií a práve jednoduchá TFT bola opäť najúspešnejšia. Prečo je TFT taká úspešná? Po prvé, TFT začína priateľsky, teda spolupracuje, čo vytvára predpoklad k nastoleniu rovnovážneho stavu CC. Po druhé, je to trestajúca stratégia, teda trestá súperove podrazy. Po tretie, je to odpúšťajúca stratégia, čo znamená, že v prípade prejavenej súperovej dobrej vôle, zabudne na predošlé podrazy. Napriek týmto výhodám, TFT má aj jednu drobnú chybičku. Ak sa stretnú dve TFT, tak je to ideálne, no predstavme si, že z istého dôvodu je raz súper donútený nespolupracovať, potom sa TFT dostáva do nepríjemného cyklu CD, DC, z ktorého niet východiska.

Vylepšená TFT: Vylepšená TFT sa môže napríklad v malom percente prípadov rozhodnúť zabudnúť na súperovu nespoluprácu, čím sa vyhne práve spomínanému efektu. To percento sa obyčajne volí v rozmedzí 5-10%. Inou možnosťou sú rôzne mechanizmy, ktoré majú kontrolovať, či súper náhodou nepoužíva len čisto kooperatívnu stratégiu, potom by sa to dalo zneužiť, inak sa vrátime k používaniu TFT.

Aplikácie

Doteraz sme sa zaoberali klasickou väzenskou dilemou, no tento problém má mnoho iných aplikácií v reálnom svete. Ako som už spomenul, hodnoty v jednotlivých kolónkach možno nastavovať ľubovoľne. To vedie k rôznym verziám väzenskej dilemy, jednou z variácií je vlastne aj zadanie tejto úlohy, kde väzni nie sú trestaní ale odmeňovaní. Nič to však nemení na poradí preferencií, takže celá teória ostáva zachovaná.

Kde všade možno nájsť aplikácie tejto hry? Začnime napríklad so zbrojením dvoch krajín. Stratégia spolupráce znamená nízke zbrojenie, naopak nespolupracovanie vedie k vysokým výdavkom za armádu. Splnenie podmienok uvádzaných pre väzenskú dilemu si možno jednoducho overiť. Najlepšie pre krajinu je mať veľkú armádu, keď súper nemá nič, potom nastupuje situácia dvoch malých armád, ktoré sú bezpečnejšie ako dve veľké armády napríklad s jadrovými zbraňami a nakoniec je situácia, keď my armádu nemáme, no protivník ju má. Takáto situácia v podstate nastala v období studenej vojny medzi ZSSR a USA. Najprv boli obe krajiny v stave DD, až príchodom Gorbačova nastal prechod ku spolupráci.

Inú aplikáciu je možno vidieť v ekonomickej spolupráci krajín. Krajina buď chráni svoje trhy tým, že má vysoké clá, a teda nekooperuje alebo uvoľní zahraničný obchod, a teda kooperuje. Moderní ekonómovia už dokázali výhodnosť liberálnej politiky, čím sa dokazuje výhodnosť CC oproti DD. Ďalším príkladom môže byť situácia dvoch kmeňov, ktoré môžu buď podnikať nájazdy jeden voči druhému alebo s ním obchodovať. Jednoducho sa dá overiť, že podmienky sú opäť splnené.

Evolúcia

Na nasledujúcich stranách popíšem výsledky evolúcie jednotlivých stratégií. Budem sa zaoberať verziou problému podľa zadania ako aj klasickou väzenskou dilemou. Mohlo by sa zdať, že výsledky oboch verzií by mali byť zhodné, však použité čísla sú rovnaké, rozdiel je len v tom, či znamenajú odmeny alebo tresty. Ako sa však ukáže, výsledky zhodné nebudú.

Väzenská dilema s odmenami

Na úvod analýzy by som uviedol tabuľku vzájomných súbojov jednotlivých stratégií.

Tabuľka 1. Turnaj stratégií pre verziu s odmenami

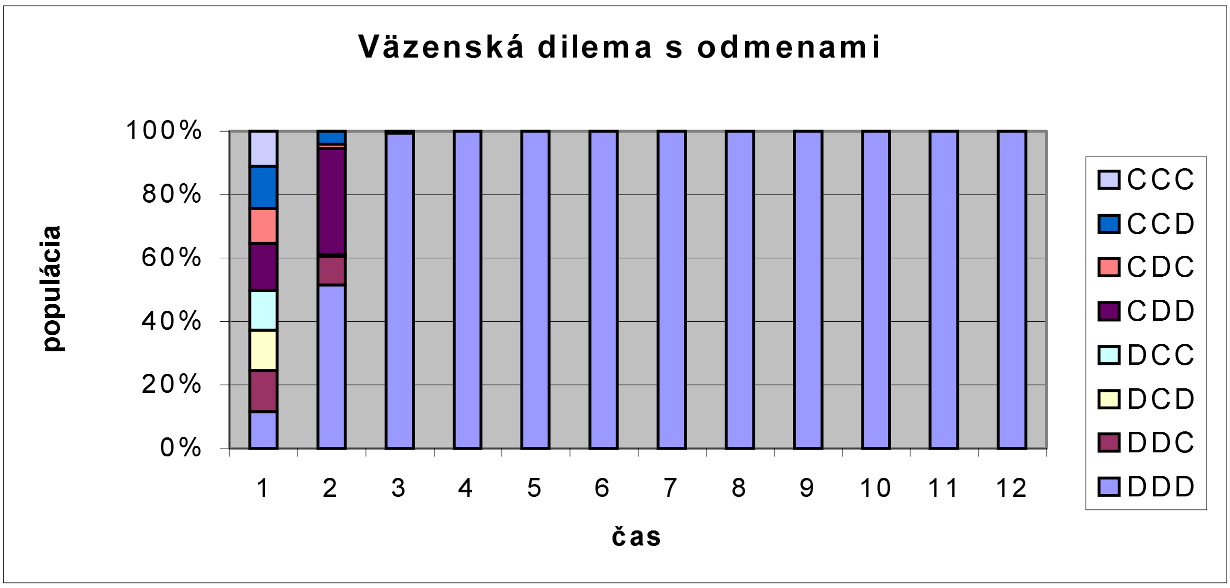
Stratégia	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC	Spolu
DDD	10 : 10	46 : 1	10 : 10	46 : 1	14 : 9	50 : 0	14 : 9	50 : 0	240
DDC	1 : 46	20 : 20	19 : 24	44 : 4	6 : 41	50 : 0	24 : 19	50 : 0	214
DCD	10 : 10	24 : 19	10 : 10	30 : 25	13 : 13	26 : 21	25 : 25	32 : 27	170
DCC	1 : 46	4 : 44	25 : 30	28 : 28	5 : 45	8 : 43	29 : 29	32 : 27	132
CDD	9 : 14	41 : 6	13 : 13	45 : 5	12 : 12	44 : 4	16 : 11	48 : 3	228
CDC	0 : 50	0 : 50	21 : 26	43 : 8	4 : 44	20 : 20	26 : 21	48 : 3	162
CCD	9 : 14	19 : 24	25 : 25	29 : 29	11 : 16	21 : 26	30 : 30	30 : 30	174
CCC	0 : 50	0 : 50	27 : 32	27 : 32	3 : 48	3 : 48	30 : 30	30 : 30	120

Pozrime sa teda na dosiahnuté výsledky jednotlivých stratégií, ktoré sú uvedené v tabuľke 1. Víťazom sa s pomerne veľkým náskokom stala stratégia DDD, teda už spomínaná nekooperatívna. Na druhom mieste nasleduje CDD, čo je vlastne opäť nekooperatívna stratégia len s tým rozdielom, že v prvom kroku „omylom“ spolupracuje. Na treťom mieste je podivná stratégia, ktorá udáva súpera, ak predtým súper spolupracoval a kooperuje, ak predtým súper dohadzoval polícii. Ostatné spôsoby už dosť strácajú. Toľko ospevovaná TFT stratégia (označená ako CCD) je až na štvrtom mieste. Úplne katastroficky dopadli všetky kooperatívne alebo viac-menej kooperatívne stratégie, CCC a DCC zohrievajú dno tabuľky.

Skúsme sa zamyslieť nad príčinami takýchto výsledkov. Ako už bolo v teoretickej časti vysvetlené, udávanie znamená neprehrávanie. Ak nahliadneme do prvého riadku tabuľky 1, kde sú uvedené nahrané body vždy v porovnaní so súperovými bodmi, tak DDD nemá nikdy pasívne skóre, čo je nie síce nutný ale dobrý predpoklad pre celkové víťazstvo. Podobne je na tom aj výber CDD. Na druhom konci kooperatívne stratégie nedokážu súpera nikdy poraziť, ich skóre je ako vidno v poslednom riadku tabuľky 1 pre možnosť CCC vždy pasívne. Zaujímavé je si aj všimnúť, že stratégia DCD v priamom súboji nikdy neprehrá, no celkovo nie je moc úspešná, je dokonca horšia ako CCD, teda naša známa TFT, ktorá môže prehrať, no nikdy o viac ako o 5 bodov. Obe tieto tvrdenia sú ľahko dokázateľné.

Ako teda bude asi prebiehať samotná evolúcia? Na začiatku máme 1000 náhodne vybraných stratégií, teda pri rozdeľovaní do skupín bude v každej skupine s veľkou pravdepodobnosťou zástupca každého výberu. Každá stratégia nebude mať asi zástupcu v každej skupine, no aspoň vo väčšine áno. Z toho sa dá očakávať, že do druhého kola postúpia hlavne zástupcovia DDD a prípadne aj CDD. Na druhej strane CCC nepostúpi skoro vôbec a podobne bude na tom aj DCC. U ostatných bude záležať na rozložení jednotlivých skupín. Ako budú vyzerať ďalšie kroky? Dalo by sa očakávať, že DDD bude víťaziť, no pozrime si radšej jednu simuláciu.

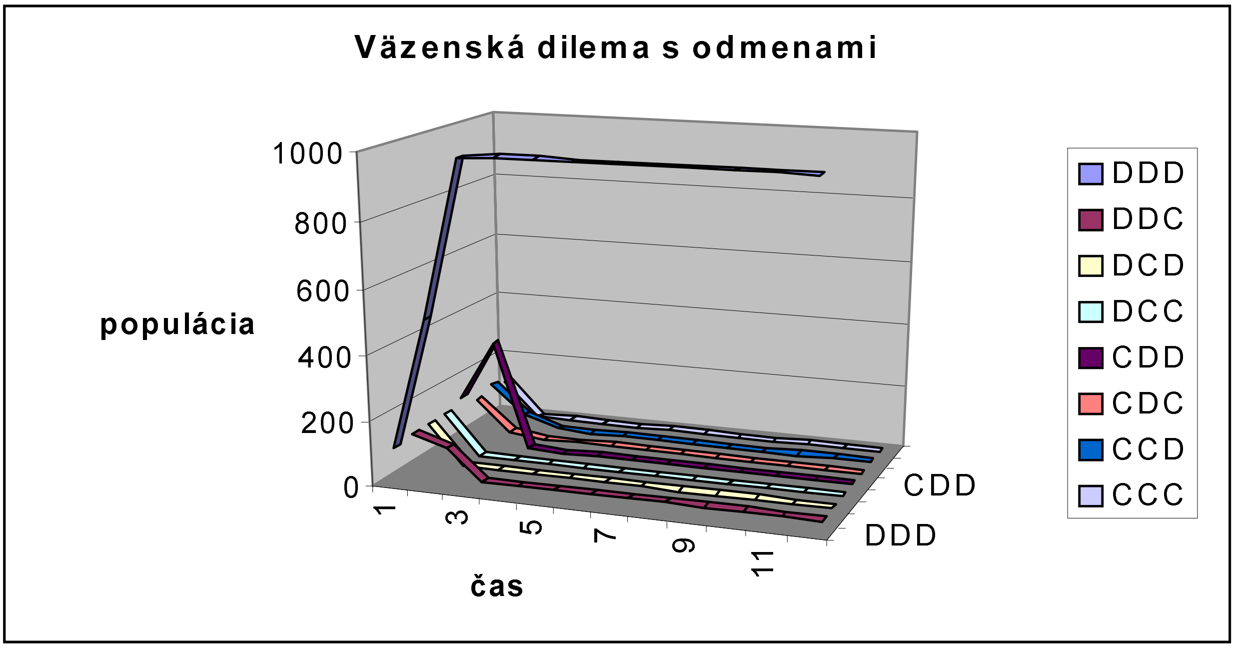
Graf 1. Ukážková simulácia väzenskej dilemy s odmenami



Tabuľka 2. Ukážková simulácia väzenskej dilemy s odmenami

Čas	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
1	116	128	127	128	147	110	132	112
2	515	92	2	2	335	14	39	1
3	993	0	2	0	2	0	3	0
4	1000	0	0	0	0	0	0	0

Graf 2. Ukážková simulácia väzenskej dilemy s odmenami



Túto ukážkovú simuláciu možno vidieť v tabuľke 2 a graficky znázornenú aj na grafoch 1 a 2. Naša analýza pred spustením simulácie bola teda správna a udavač zvíťazil. Nepotreboval dokonca ani moc krokov, všetkých ostatných odstránil za 3 kroky. Dokonca aj druhý v poradí CDD bol v prvom kroku podľa očakávania pomerne úspešný a polepšil si svoju pozíciu. Musíme si však uvedomiť, že evolučný algoritmus, ktorý používame na simuláciu v tejto práci je stochastický, a teda sa nám mohlo stať, že prvá simulácia vrátila vysoko nepravdepodobný výsledok. Jediný spôsob, ktorým si výsledok možno overiť, je spustiť simuláciu viackrát. Nasledujúca tabuľka udáva víťazov a priemerné počty krokov, ktoré potrebovali na dosiahnutie úplného víťazstva pre 10 000 opakovaní evolúcie.

Tabuľka 3. Víťazi väzenskej dilemy s odmenami

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	9896	0	0	0	0	0	104	0
Kroky	2.40						11.05	

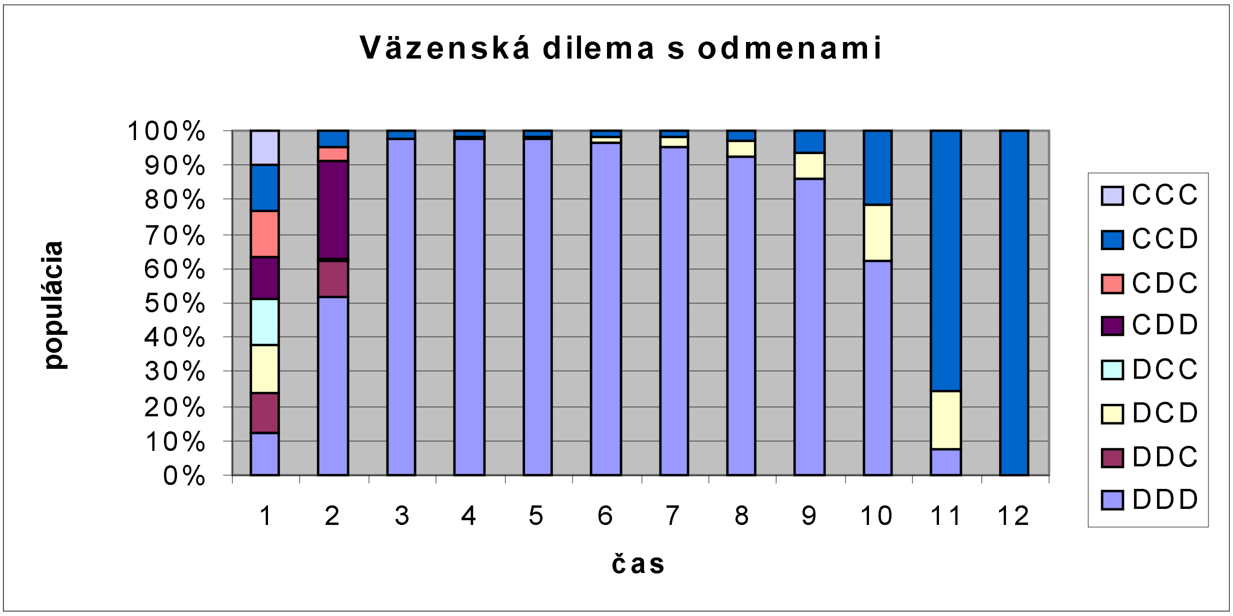
Tabuľka 3 ilustruje výsledky. Hlavným víťazom je podľa očakávania nekooperatívny výber. Vyhráva v 99% prípadov a vyhráva pomerne hladko. Zaujímavé však je, že pôvodne proklamovaná stratégia TFT tiež dokáže zvíťaziť. Darí sa jej to len v 1% prípadov, no aj to je lepšie ako nič. Otázka samozrejme stojí, ako je to možné. Skúsme si teda pozrieť jednu evolúciu, v ktorej víťazí TFT.

Tabuľka 4. Ukážková simulácia výhry TFT pre väzenskú dilemu s odmenami

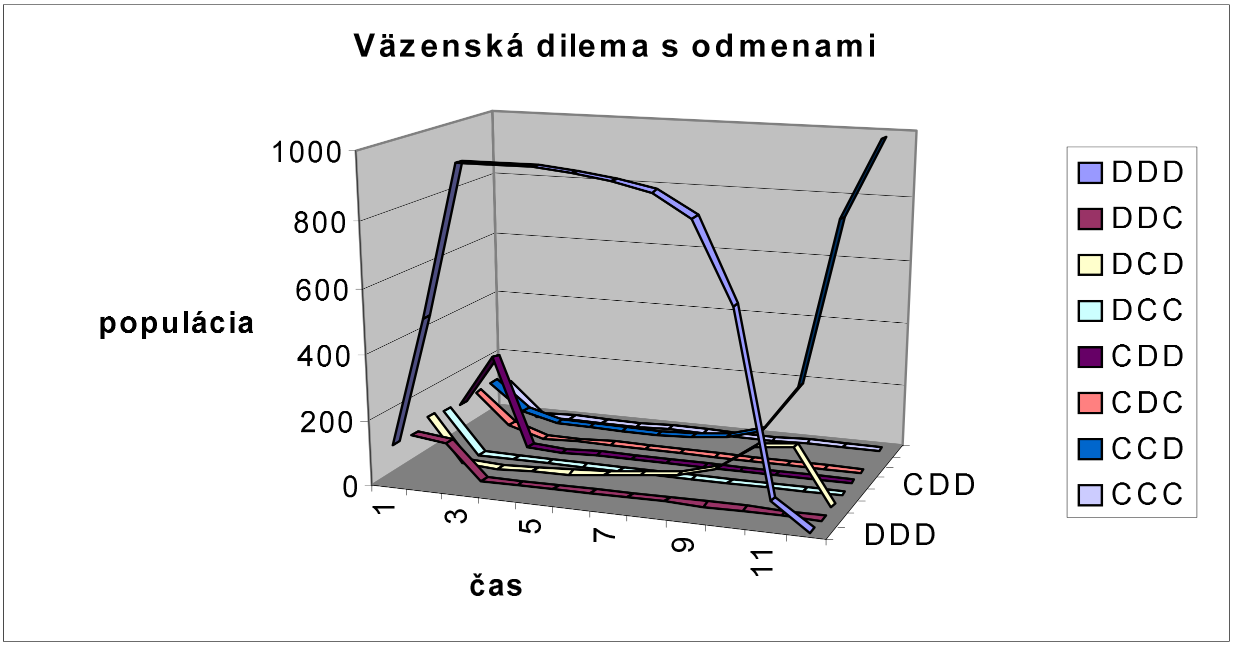
Čas	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
1	120	117	143	133	122	130	137	98
2	517	106	5	0	287	36	49	0
3	974	0	3	0	2	0	21	0
4	975	0	6	0	0	0	19	0
5	974	0	11	0	0	0	15	0
6	966	0	18	0	0	0	16	0
7	951	0	32	0	0	0	17	0

8	925	0	45	0	0	0	30	0
9	861	0	76	0	0	0	63	0
10	624	0	158	0	0	0	218	0
11	78	0	168	0	0	0	754	0
12	0	0	0	0	0	0	1000	0

Graf 3. Ukážková simulácia výhry TFT pre väzenskú dilemu s odmenami



Graf 4. Ukážková simulácia výhry TFT pre väzenskú dilemu s odmenami



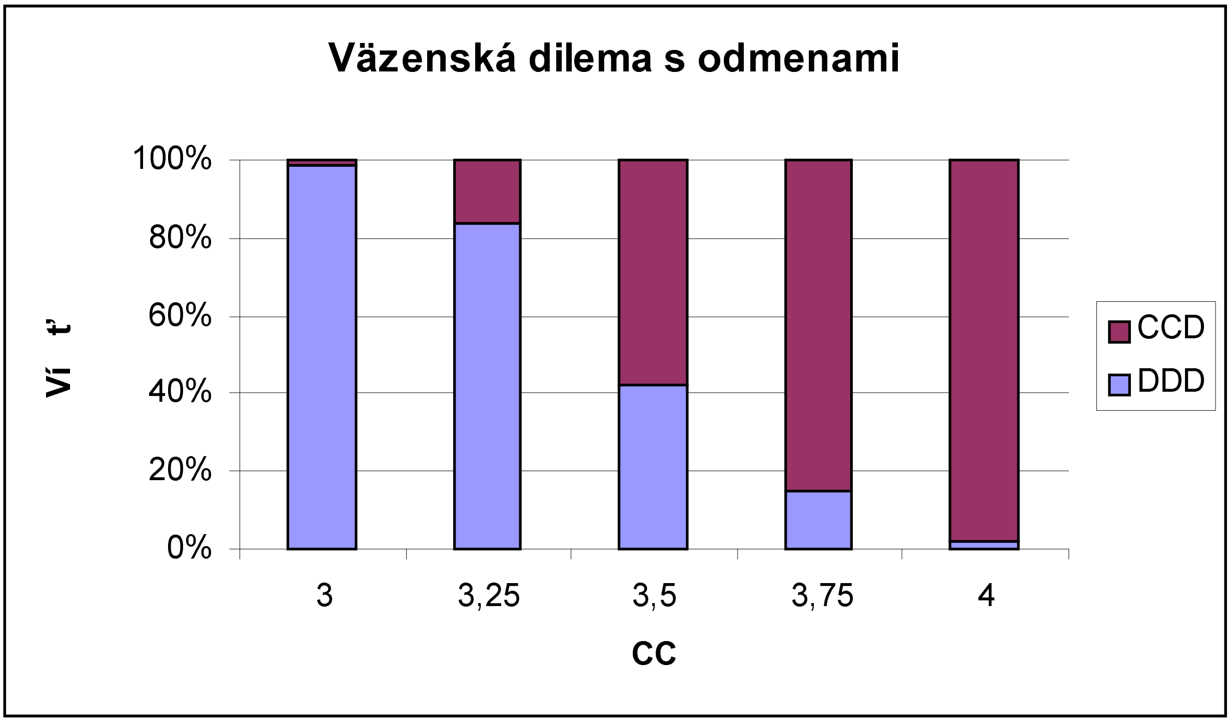
Ak sa pozorne pozrieme na tabuľku 4 a grafy 3 a 4, uvidíme, že vývoj bol po dva kroky podobný ako pri úvodnej simulácii. Zmena však nastala v tom, že stratégia DCD sa od toho momentu začala rozmáhať, čo využila CCD a pohodlne zvíťazila. Ako si však možno vysvetliť úspech DCD? Jednoducho, DCD sa v súboji proti DDD správa ako DDD, takže v tomto súboji remizuje. Potom už v skupine záleží na poradí, lebo v prípade rovnosti bodov, postupuje hráč s nižším poradovým číslom. Tým pádom sa aspoň v jednom prípade zo sto začne rozmáhať DCD proti DDD. Ak by ale TFT stratégia nemala dostatočnú populáciu, tak by asi zanikla a súboj by ostal dlhodobo nerozhodný medzi DDD a DCD, čo obyčajne vďaka lepšej štartovacej pozícii vyhrá nekooperatívna stratégia, teda DDD. Ak však TFT má dostatočnú populáciu, tak sa udrží, narastie a vyhrá. Vyhrá vďaka tomu, že je proti DCD úspešnejšia ako proti DDD. Ako vidno z tabuľky 1, proti DCD získa 25 bodov a proti DDD získa iba 9 bodov.

Pre lepšie pochopenie si rozanalyzujme rôzne situácie. Ak je 9 DDD a 1 CCD, potom vyhrávajú DDD. Ak je 9 DDD a 1 DCD, potom rozhoduje náhoda, teda DCD postúpi v 10% prípadov. Prípad 8 DDD a 2 DCD je podobný, akurát DCD postúpi v 20% prípadov. Čo ak je v skupine 8 DDD a 2 CCD, alebo ako je úspešná TFT proti nekooperatívnosti? Po spočítaní hodnôt zistíme, že TFT získa 102 bodov a nespolupracovník len 98 bodov, teda CCD vyhráva a postupujú obaja jej zástupcovia. Ešte jedna zaujímavá možnosť, nech je v skupine 8 DDD, 1 DCD a 1 CCD. V tomto prípade vyhrá DCD so 105 bodmi pred CCD s 97 bodmi a až na konci skončí všetkých 8 verzií DDD s 94 bodmi. Vidíme teda, že 8 DDD v skupine s niekým z CCD alebo DCD pomerne často vypadáva. Nasleduje teda otázka, koľko zástupcov uvedených dvoch stratégií potrebujeme, aby boli schopní rastu a víťazstva.

Predpokladajme, ak je v skupine 8 alebo menej zástupcov DDD, potom vypadnú a postúpia ich dvaja súperi. Nech je **x** zástupcov CCD a DCD, potom ostáva **1000-x** zástupcov DDD. Nech C(n,k) sú kombinácie n nad k, potom šanca, že všetci v skupine budú z DDD je: **C(1000-x,10)/C(1000,10).**
Šanca, že v skupine bude práve 9 chlapcov z DDD je: **C(1000-x,9)*C(x,1)/C(1000,10).**
My chceme, aby počet zástupcov DDD klesol. Ak teda sčítam uvedené dve pravdepodobnosti a urobím ich doplnok, dostanem pravdepodobnosť výskytu 8 a menej DDD v jednej skupine, označme ju **p**. Podľa predpokladov z každej takejto skupiny postupujú DCD alebo CCD, keďže skupín je 500 a z každej postupujú 2, dostávame nerovnicu **500*2*p>x**. Po vyriešení tejto nerovnice dostávame **x>26**, teda 27 by už malo stačiť. Bohužiaľ úvodný predpoklad nie je splnený, lebo pri výbere dvoch DCD a 8 DDD v skupine rozhoduje náhoda a DCD väčšinou dokonca vypadne, takže bude potrebné ešte o niečo väčší počet iných stratégií. Ak sa však pozrieme späť na tabuľku 4, vidíme, že v štvrtom resp. piatom kroku bola populácia CCD a DCD spolu len 24 resp. 25. K tomu neostáva len dodať, že vypočítané hodnoty sú priemerné pravdepodobnosti a skutočná simulácia sa od nich bude vždy o niečo líšiť. Ako však vidno po prekonaní týchto dvoch krokov už uvedené stratégie pomerne rýchlo rastú a CCD neskôr vyhubí aj svojho pôvodného pomocníka DCD a vyhrá.

Týmto by som analýzu preberanej hry uzavrel. Je jasné, že analýza prevedená v poslednom odstavci by sa dala poopraviť, aby presne vyhovovala podmienkam úlohy, no to by bolo treba pridať ďalšiu premennú a riešiť nerovnicu o dvoch neznámych. Toto však už presahuje rámec tejto práce. Mohli by sme sa však ešte zamyslieť, ako sa zmení evolúcia, ak zmeníme podmienky úlohy. Konkrétne ak si spomenieme na druhú podmienku, ktorá hovorila, že CC musí byť viac ako priemer DC a CD, v našej úlohe to bolo 3 voči 2.5, čo podmienku spĺňa, no môžeme túto nerovnosť ešte zväčšiť a tým zvýhodniť spoluprácu. V tabuľke 5 resp. 6 sú uvedené výsledky analýzy pre CC=3.5 resp. 4.

Graf 5. Víťazi väzenskej dilemy s odmenami pre rôzne hodnoty CC



Tabuľka 5. Víťazi väzenskej dilemy s odmenami pre CC=3.5

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	4235	0	0	0	0	0	5765	0
Kroky	4.13						7.45	

Tabuľka 6. Víťazi väzenskej dilemy s odmenami pre CC=4

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	186	0	0	0	0	0	9814	0
Kroky	5.55						5.47	

Uvedené výsledky len potvrdzujú všeobecné očakávania. Čím je spolupráca lepšie ohodnotená, tým lepšie dopadajú kooperačne orientované stratégie. Tu by som chcel upozorniť, že zmena podmienok nechala v platnosti všetky pôvodné úvahy. Rozdiel oproti verzii s menším CC je v tom, že u vyšších CC postúpi viac zástupcov TFT do druhého a tretieho kola a potom si už jednoduchšie poradia v boji proti nekooperatívnym stratégiám, ktorých je tam aj pri zmenených podmienkach väčšina. Táto úvaha je podporovaná aj klesajúcim priemerným počtom krokov pri výhre CCD a rastúcim pri výhre DDD.

Klasická väzenská dilema

V ďalších riadkoch sa pokúsim stručne rozanalyzovať klasickú väzenskú dilemu a poukázať na jej rozdiely voči väzenskej dileme s odmenami. Na prvý pohľad symetrické zadanie s minimalizovaním trestov namiesto maximalizovania odmien by mohlo dávať rovnaké výsledky. Ak nahliadneme do tabuľky 7, v ktorej sú uvedené výsledky turnaja každý s každým klasickej väzenskej dilemy. Poradie je úplne zhodné s poradím u väzenskej dilemy s odmenami. Opäť sa teda dá očakávať, že stratégia DDD bude víťaziť. Ak sa vrátime k poznámkam uvedeným na konci predošlého odseku, zistíme, že otázkou ostáva, nakoľko je spolupráca atraktívna, aby sa rozhodlo, či nejaká TFT prežije prvé kolá, keď je pod veľkým náporom nespolupracujúcich stratégií.

Tabuľka 7. Turnaj stratégií pre klasickú verziu

Stratégia	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC	Spolu
DDD	30 : 30	3 : 48	30 : 30	3 : 48	27 : 32	0 : 50	27 : 32	0 : 50	120
DDC	48 : 3	20 : 20	26 : 21	4 : 44	43 : 8	0 : 50	21 : 26	0 : 50	162
DCD	30 : 30	21 : 26	30 : 30	11 : 16	29 : 29	19 : 24	25 : 25	9 : 14	174
DCC	48 : 3	44 : 4	16 : 11	12 : 12	45 : 5	41 : 6	13 : 13	9 : 14	228
CDD	32 : 27	8 : 43	29 : 29	5 : 45	28 : 28	4 : 44	25 : 30	1 : 46	132
CDC	50 : 0	50 : 0	24 : 19	6 : 41	44 : 4	20 : 20	19 : 24	1 : 46	214
CCD	32 : 27	26 : 21	25 : 25	13 : 13	30 : 25	24 : 19	10 : 10	10 : 10	170
CCC	50 : 0	50 : 0	14 : 9	14 : 9	46 : 1	46 : 1	10 : 10	10 : 10	240

Tabuľka 8. Víťazi klasickej väzenskej dilemy

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	10000	0	0	0	0	0	0	0
Kroky	2.00							

Výsledok nie je až taký prekvapujúci. Dokumentuje, že táto situácia je síce podobná tej pre väzenskú dilemu s odmenami, no táto verzia je z hľadiska spolupráce ešte o niečo horšia. V tabuľke 8 vidíme, že jednoznačným víťazom je DDD. Ak sa chceme dostať k príčinám, tak si stačí urobiť malé prepočty. Rozdiely nastávajú vo výsledkoch skupín s 8 DDD a 2 CCD. V tomto prípade si totiž DDD odsedia len 264 rokov, kdežto CCD až 266, čo robí zo zástupcov DDD víťazov. K podobnej situácii dochádza v skupine zloženej z 8 DDD, 1 CCD a 1 DCD. Víťazí DCD s 265 rokmi väzenia pred DDD s 267 rokmi a CCD je ďaleko vzadu s 281 rokmi.

Aby sme dodržali symetriu s verziou s odmenami, skúsme teraz pohnúť hodnotou DD. Pýtame sa teda, koľko musí daná hodnota byť, aby stratégia TFT vôbec mala šancu na úspech. Pre možnosť 8 DDD a 2 CCD dostávame hodnotu **88DD** pre DDD a **50+72DD** pre CCD, čo vedie k výsledku **DD>3.125.** Pre možnosť 8 DDD, 1 CCD a 1 DCD je výsledok ešte nepriaznivejší, lebo hodnota pre DDD je **89DD** a pre CCD to je **65+72DD**, čo vedie k nerovnici **DD>3.824.** Pri pôvodnej hodnote DD bolo na úspech TFT potrebné, aby sa vyskytovali v dostatočnom počte skupín v trojiciach. Podobnou analýzou s kombinačnými číslami ako vo verzii s odmenami dostanem, že počet stratégií TFT musí byť aspoň 131. Toto je však štatisticky nereálne, lebo TFT v prvých kolách pomerne rýchlo vymiera a prebudiť sa dokáže až v ďalších, k čomu však práve z uvedeného dôvodu nepríde.

Skúsme teda najprv hodnotu DD=4, ktorá vyhovuje obom vypočítaným podmienkam. Jej hodnoty sú v tabuľke 9. A potom vezmime ešte aj hodnotu DD=3.5, ktorej výsledky sú v tabuľke 10.

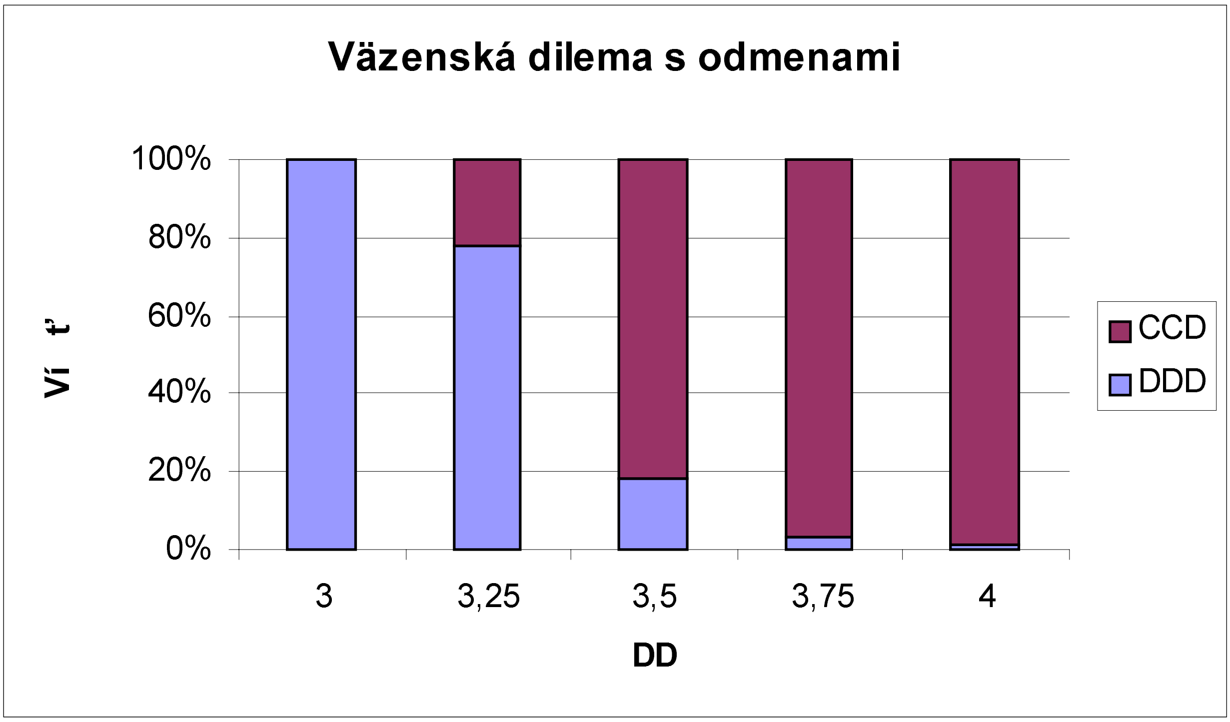
Tabuľka 9. Víťazi klasickej väzenskej dilemy pre DD=4

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	122	0	0	0	0	0	9878	0
Kroky	5.12						5.29	

Tabuľka 10. Víťazi klasickej väzenskej dilemy pre DD=3.5

Víťaz	DDD	DDC	DCD	DCC	CDD	CDC	CCD	CCC
Počet	1799	0	0	0	0	0	8201	0
Kroky	4.30						6.41	

Graf 6. Víťazi klasickej väzenskej dilemy rôzne hodnoty DD



Výsledky nie sú veľmi prekvapujúce, aj keď predsa len sa taký výrazný úspech TFT už pri hodnote 3.5 nedal očakávať. Výsledok len dokumentuje to, že v tejto verzii hry sa TFT nemusí spoliehať na pomoc DCD, ale vystačí si aj sama, pretože pre 3.5 vyhráva vo vyše 80% prípadov a pri 3.75 dokonca až vo viac ako 95% prípadov.

Implementácia

V prílohe k tejto práci prikladám program v jazyku Pascal, ktorý rieši pôvodnú úlohu, teda väzenskú dilemu s odmenami. Zmena na klasickú väzenskú dilemu je pomerne triviálna, stačí zmeniť typ úlohy z Max na Min a zmeniť ohodnotenie podmienok CC, CD, DC, DD. Všetky dôležité údaje sú ako konštanty v úvode programu, pri ich zmenení sa dá program upraviť na riešenie ľubovoľnej úlohy typu 2x2. Pôvodná verzia programu vždy pre každú dvojicu robila 10 stretnutí a na základe ich výsledku pripočítavala hodnoty. Vylepšená verzia si vždy na začiatku vypočíta tabuľku hodnôt turnaja každý s každým a hodnoty z tejto tabuľky využíva pri interakcii každej dvojice, toto je veľmi výhodné, lebo súťaží celkovo len 8 stratégií. Príkladmi takýchto tabuliek sú tabuľky 1 a 7. Týmto vylepšením sa celkový čas výpočtu mnohonásobne zmenšil. Jediným problémom teda bolo rozhodnúť, kedy sa niektorá stratégia prehlási za víťaza turnaja. Ideálne by to malo byť po dosiahnutí plného stavu, teda 1000 zástupcov. Toto nie je problém pre TFT, ale pre nekooperatívnu stratégiu DDD problém nastáva. Ten problém je práve v tom, že poraziť DCD nie je jednoduché, záleží tam totiž dosť od náhody. Z toho dôvodu som obmedzil víťazstvo na stav, keď súčet druhých mocnín počtov zástupcov stratégií je viac ako 980000, čo zodpovedá prekročeniu hodnoty 990 jednou stratégiou. V prípade, že ani toto nestačí, som obmedzil počet kôl na 15. Ak si však pozrieme ľubovoľnú z analyzovaných tabuliek, vidíme, že táto hodnota prekračovaná nebola, dokonca sa k nej stratégie ani len neblížili, takže obyčajne bola využívaná podmienka druhých mocnín. K tejto podmienke ma viedli skôr uvedené výpočty, ktoré hovoria, že zásadný obrat môže nastať iba pri populácii súperov poriadne cez 20, takže chyby by sa vyskytovať nemali.

Záver

Hlavnou otázkou tejto práce bolo zistiť, čo je výhodnejšie spolupracovať alebo nespolupracovať. V oboch preberaných verziách väzenskej dilemy sa vyskytli podmienky, pri ktorých bolo výhodnejšie spolupracovať ale aj podmienky, pri ktorých bolo výhodnejšie nespolupracovať. Najzaujímavejším úkazom je však to, že kooperácia vzniká z čisto sebeckých dôvodov, teda maximalizovať vlastný úžitok. Z uvedených analýz vidno, že spolupráca vzniká evolúciou a nemusí nastať priamo od začiatku. Evolúcia nie je hrou s nulovým súčtom a vďaka tomu môže kooperácia priniesť výrazné výhody pre všetkých zúčastnených.

Ako som už spomenul vyššie, väzenská dilema má množstvo aplikácií v reálnom svete. Vo väčšine prípadov platí, že obojstranná spolupráca je výhodnejšia ako obojstranné škodenie si. Pri spolupráci je tu vždy však hrozba, že druhá strana z nejakého dôvodu odstúpi, čo by nás veľmi poškodilo. A práve relatívna veľkosť tejto škody je jedným z rozhodujúcich faktorov, ktoré ovplyvňujú náchylnosť jednotlivých hráčov k spolupráci.

Už pán Axelrod empircky zistil, že TFT je jednou z najlepších a pomerne jednoduchou stratégiou. V tejto práci sme analyzovali len stratégie s pamäťou 1, možné rozšírenie môže viesť k stratégiám s väčšou pamäťou, ktoré môžu byť pre menšie čísla riešené rovnakým spôsobom, aký je popísaný v tejto práci. Pre väčšie hodnoty sa dá pristúpiť k iným algoritmom, napríklad ku genetickému algoritmu.

```
Program Vaznova_dilema; {Evolucny agloritmus na hladanie najlepsej strategie}
Uses Crt, Dos;
Const pv  = 1000;   {pocet vaznov}
      ps  = 500;    {pocet skupin}
      pvs = 10;     {pocet vaznov v skupine}
      pt  = 10;     {pocet tahov}
      pn  = 2;      {pocet najlepsich/postupujucich}
      pk  = 10000;  {pocet opakovani hry}
      out   : String = 'vd3.txt'; {vystupny subor}
      typ   : String = 'Max';     {typ hry Max (odmena)/ Min (trest)}
      trest : Array [0..1,0..1] of Real = ( (1,5),(0,3) ); {trest/odmena}
      {0 znamena kooperaciu a 1 znamena nekooperaciu}

Var   s    : Array [1..pv] of Byte;      {strategie vaznov}
      ns   : Array [1..pv] of Byte;      {nove strategie vaznov}
      svs  : Array [1..pvs] of Byte;     {strategie vaznov v skupine}
      sedi : Array [1..pvs] of Real;     {tresty z interakcii v skupine}
      poc  : Array [0..7] of Longint;    {pocty strategii}
      vitaz: Array [0..7] of Integer;    {pocty vitazstiev}
      prt  : Array [0..7] of Real;       {priemerne pocty tahov}
      tab  : Array [0..7,0..7] of Real;  {tabulka vysledkov}
      f    : Text;                       {vystupny subor}
      tah1,tah2   : Byte;           {tahy vaznov}
      ptah1,ptah2 : Byte;           {predosle tahy vaznov}
      sedi1,sedi2 : Real;           {dokopy tresty}
      i,j,k,l,q,p : Integer;
      eps         : Real;

Begin
 {inicializacia}
 Assign(f,out);
 Rewrite(f);
 Clrscr;
 Randomize;
 For i:=0 to 7 do
  Begin
   vitaz[i]:=0;
   prt[i]:=0;
  End;
 {generovanie tabulky vysledkov vzajomnych subojov}
 For i:=0 to 7 do
  For j:=i to 7 do
   Begin
    tah1:=i Div 4;           {prvy tah prveho vazna}
    tah2:=j Div 4;           {prvy tah druheho vazna}
    sedi1:=trest[tah1,tah2]; {trest za prvy tah prveho vazna}
    sedi2:=trest[tah2,tah1]; {trest za prvy tah druheho vazna}
    For l:=2 to pt do
     Begin
      ptah1:=tah1;
      ptah2:=tah2;
      tah1:=ptah2*((i Div 2) Mod 2)+(1-ptah2)*(i Mod 2); {tah prveho vazna}
      tah2:=ptah1*((j Div 2) Mod 2)+(1-ptah1)*(j Mod 2); {tah druheho vazna}
      sedi1:=sedi1+trest[tah1,tah2]; {plus trest za tah prveho vazna}
      sedi2:=sedi2+trest[tah2,tah1]; {plus trest za tah druheho vazna}
     End;
    tab[i,j]:=sedi1;
    tab[j,i]:=sedi2;
   End;
 {spustenie simulacii}
 For q:=1 to pk do
  begin
   For i:=1 to pv do
```

```
    s[i]:=Random(8); {generuj nahodne strategie}
   For i:=0 to 7 do
    poc[i]:=0;
   For i:=1 to pv do
    Inc(poc[s[i]]);
   p:=0;
   Repeat
    For i:=1 to ps do
     Begin
      For j:=1 to pvs do
       Begin
        svs[j]:=s[Random(1000)+1]; {nahodny vyber do skupiny}
        sedi[j]:=0;
       End;
      For j:=1 to pvs do          {kazdy s ...}
       For k:=(j+1) to pvs do     {... s kazdym}
        Begin
         {pripocitanie hodnot z tabulky}
         sedi[j]:=sedi[j]+tab[svs[j],svs[k]];
         sedi[k]:=sedi[k]+tab[svs[k],svs[j]];
        End;
      For l:=1 to pn do
       Begin
        j:=1;
        For k:=2 to pvs do
         If (typ='Max') and (sedi[k]>sedi[j]) then j:=k;
         If (typ='Min') and (sedi[k]<sedi[j]) then j:=k;
        ns[pn*i-l+1]:=svs[j];
        sedi[j]:=0;
       End;
     End;
    For i:=0 to 7 do
     poc[i]:=0;
    For i:=1 to pv do
     Inc(poc[ns[i]]);
    {For i:=0 to 7 do {vypis momentalneho stavu}
    { Begin
      Writeln(poc[i]);
      Write(f,poc[i]:4,';');
     End;
    Writeln('--');
    Writeln(f);
    Delay(1000);}
    For i:=1 to pv do
     s[i]:=ns[i];
    eps:=0;
    For i:=0 to 7 do
     eps:=eps+Sqr(poc[i]);
    Inc(p);
   Until (eps>980000) or (p=15); {zaverecna podmienka}
   j:=0;
   For k:=1 to 7 do
    If (poc[k]>poc[j]) then j:=k;
   Inc(vitaz[j]);
   prt[j]:=prt[j]+p;
  end;
 {vypis vysledkov}
 For i:=0 to 7 do
  If (vitaz[i]<>0)
   Then Writeln(f,vitaz[i]:5,';',(prt[i]/vitaz[i]):5:2)
   Else Writeln(f,'    0;');
 Close(f);
End.
```