

Optimal abstraktion i heads up no-limit Texas hold'em

Dan Eriksson

May 21, 2012

E-mail: dane2@kth.se

Kurs: DD143X, Examensarbete inom datalogi, grundnivå

Skola: Skolan för datavetenskap och kommunikation

Universitet: Kungliga Tekniska Högskolan

Handledare: Mikael Goldmann

Abstract

En bra pokerspelare förknippas ofta med en förmåga att läsa motståndare. Trots detta har alla pokerspel en lösning i form av en nashjämviktsstrategi, som inte tar hänsyn till hur motståndaren spelar. En nashjämviktsstrategi för heads up no-limit poker är inte beräkningsbar med dagens datorer. En metod för att kringgå detta problem är att abstrahera spelet till ett spel där en nashjämviktsstrategi är lättare att beräkna. Den här uppsatsen jämför olika abstraktionsmetoder och redovisar skillnaderna i resultat mellan strategierna framtagna från respektive abstraktionsmetod. Abstraktionsmetoden som visade störst potential ställdes mot en icke-trivial pokerbot och lyckades vinna med god marginal.

Abstract

A good poker player is often associated with an ability to read the opponent. There is despite this a solution to the game of poker. The solution is called a nash equilibrium strategy. Calculating a nash equilibrium strategy for heads up no-limit Texas hold'em is not possible with the computers of today. A method to tackle this is to abstract the game to a simplified game where a nash equilibrium strategy can be calculated. Different ways to abstract the game were compared in this essay. The most promising abstraction was matched against a non-trivial poker bot and managed to win by a good margin.

Contents

1	Inledning	4
1.1	Syfte	4
1.2	Regler och begrepp för heads up no-limit Texas hold'em	4
2	Pokerstrategier	5
2.1	Komponenter i heads up no-limit hold'em	5
2.2	Definition av en strategi	6
2.3	Nashjämviktsstrategi	6
2.4	Exploatering av motståndare	6
2.5	Motivering till att använda nashjämviktsstrategier	7
3	Att hitta en nashjämviktsstrategi	7
3.1	Counterfactual regret minimization	7
3.2	Abstraktion	7
3.2.1	Handlingsabstraktion	8
3.2.2	Hinkabstraktion	8
3.2.3	Handstyrka	8
3.2.4	Handstyrka i kvadrat	9
3.2.5	Procentuella hinkar	9
3.2.6	Minmaxhinkar	9
3.2.7	Historikhinkar	10
3.2.8	Vad händer när motståndaren inte följer abstraktionsreglerna?	10
4	Resultat	10
4.1	Hur man utvärderar strategier	10
4.2	Resultat mellan hinkabstraktioner	11
4.3	hhs ² mot SimpleBot	12
5	Slutsats	13
	References	14

1 Inledning

1.1 Syfte

Poker på Internet är en industri som omsätter miljardtals kronor årligen[1]. Med stora summor pengar inblandade är det viktigt att möjligheterna för fusk är låga. Det har de senaste åren spekulerats kring fusk genom användandet av så kallade pokerbotar.[8]. Trots det lever pokern vidare genom tron att pokerbotar inte är kapable att konkurrera med toppskiktet av pokerspelare i den mest spelade formen av poker - no-limit Texas hold'em. Studier av poker AI är därför intressant, inte bara ur en spelteoretisk synvinkel, utan även för Internetpokerindustrins fortsatta legitimitet.

1.2 Regler och begrepp för heads up no-limit Texas hold'em

Heads up no-limit Texas hold'em är en variant av Texas hold'em där enbart två spelare deltar, och där spelarna har möjlighet att satsa ett fritt antal mörkar vid varje satsningstillfälle. Målet är att vinna motståndarens marker, vilka oftast har ett monetärt värde. Båda spelarna väljer själv innan spelomgången hur mycket marker de vill ha möjlighet att satsa. Dessa marker kallas för spelarens stack. Oftast bestäms både en minimum- och maximumgräns för stacken. Om den ena spelarna väljer en större stack än motståndaren kan han dock inte vinna mer än hela motståndarens stack. Den effektiva stacken är således den minsta stacken av spelarna.

Båda spelarna tilldelas vid början av spelomgången två kort var osynliga för motståndaren, och båda spelarna placerar obligatoriska satsningar. De obligatoriska satsningarna kallas för mörkar. Den ena mörken är dubbelt så stor som den andra. Gränserna för stacken sätts i relation till den stora mörken, vanligtvis en minimumgräns på 20 stora mörkar och en maximumgräns på 100 stora mörkar. Efter det följer satsningsrundor, där spelarna i tur och ordning har möjlighet att antingen lägga sig (fold), syna (call), eller satsa ett godtyckligt antal marker (bet). En satsningsrunda avslutas när ena spelaren synat motståndarens satsning, eller om ingen satsat och båda agerat. I vilken ordning spelarna agerar beror på vem som har knappen. Efter rundan tilldelas båda spelarna gemensamma kort synliga för båda spelarna.

- Spelaren med knappen placerar lilla mörken, och motståndaren placerar den stora mörken. Två enskilda kort tilldelas båda spelarna, osynliga för motståndaren. Dessa kort kallas för spelarens hålkort.
- Satsningsrunda. Spelaren som placerade lilla mörken agerar först.
- Tre gemensamma kort tilldelas (flop).
- Satsningsrunda. Spelaren som placerade stora mörken agerar först.
- Ett gemensamt kort tilldelas (turn).
- Satsningsrunda. Spelaren som placerade stora mörken agerar först.
- Ett gemensamt kort tilldelas (river).

- Satsningsrunda. Spelaren som placerade stora mörken agerar först.
- Förutsatt att ingen spelare lagt sig så vinner spelaren med bästa möjliga femkortshand den samlade potten av marker.
- Knappen förflyttas och en ny spelomgång påbörjas.

2 Pokerstrategier

2.1 Komponenter i heads up no-limit hold 'em

En formell beskrivning av komponenterna i no-limit Texas hold 'em, som även är applicerbar för alla ändliga omfattande spel med ofullständig information, ges av Osborne och Rubinstein[7].

- Ett ändligt antal spelare, N . $N = 2$ i heads up Texas hold 'em.
- En ändlig mängd historiksekvenser, H . $Z \subseteq H$ innefattar alla historiksekvenser som avslutar spelet.
- En funktion, P , som för varje historiksekvens h antingen anger vilken spelares tur det är att agera, eller c om chansen anger nästkommande historiksekvens. I Texas hold 'em är $P(h) = c$ ifall kort ska delas ut till spelarna.
- En funktion f_c som för varje historiksekvens h där $P(h) = c$ anger sannolikheten för alla möjliga nästkommande historiksekvenser.
- För varje spelare $i \in N$ en mängd I_i med alla historiksekvenser där $h \in H$ och $P(h) = i$, med egenskapen att spelare i inte har möjlighet att särskilja historiksekvenserna i I_i från varandra. Denna mängd kallas för spelarens informationsmängd.
- För varje spelare $i \in N$ en funktion u_i som för varje avslutande historiksekvens $h \in Z$ anger ett värde. Då heads up Texas hold 'em är ett nollsummespel med två spelare har vi $u_1 = -u_2$.

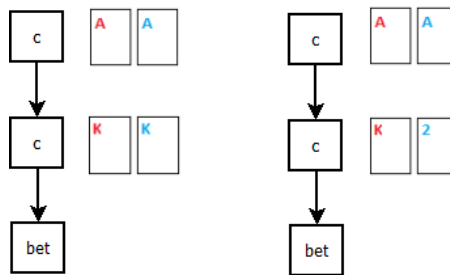


Figure 1: Två historiksekvenser.

Figuren ovan visar två olika historiksekvenser i Texas hold 'em. Korten som delas ut representeras av c och första spelaren väljer att satsa. Notera att båda historiksekvenserna ligger i samma informationsmängd för den första spelaren, då denna spelaren inte vet utgången av motståndarens kortutdelning. För den andra spelaren ligger historiksekvenserna i olika informationsmängder, då denna spelaren har informationen för att särskilja dem.

2.2 Definition av en strategi

En strategi är en mappning från varje informationsmängd till en sannolikhetsfördelning över de möjliga handlingarna vid informationsmängden. En strategi är statisk och ändras inte med tiden. En *ren strategi* är en strategi som för varje informationsmängd alltid väljer en och samma handling av de möjliga handlingarna vid informationsmängden. En *blandad strategi* är en strategi som enligt en sannolikhetsfördelning väljer olika handlingar för minst en informationsmängd. En strategi som ibland höjer och ibland synar med ett par i ess innan floppen är ett exempel på en blandad strategi.

2.3 Nashjämviktsstrategi

I no-limit hold 'em finns det en strategi som inte har ett negativt förväntat värde mot någon annan strategi, även om man öppet informerar motspelaren om hur ens strategi ser ut[2]. Denna strategi kallas spelteoretiskt optimal, eller för en nashjämviktsstrategi. När två spelare spelar mot varandra med en nashjämviktsstrategi, innebär det att ingen av spelarna har något att tjäna på att ändra sin strategi.

2.4 Exploatering av motståndare

Poker är vanligtvis ett spel som förknippas med exploatering av motståndaren, antingen genom att man försöker tolka ansiktsuttryck, eller genom att försöka klura ut motståndarens spelstrategi och därefter anpassa sig till den. Eftersom syftet med poker inte bara är att undvika att förlora, men även att vinna så mycket som möjligt, har spelare som effektivt exploaterar motståndare varit framgångsrika.

2.5 Motivering till att använda nashjämviktsstrategier

I takt med att spelare på de höga nivåerna blir allt bättre och svårare att exploatera, har balanserade nashjämviktsstrategier förespråkats. Detta beror på att så fort man byter strategi för att exploatera en motståndares strategi, så öppnar man upp sig själv för att bli exploaterad. Att försöka exploatera en motståndare är därför riskabelt och det finns inga garantier för att ens uppfattning om motståndarens förmåga är riktig, oavsett hur länge man sett denna personen spela. En liknelse med schack kan göras. När man föreställer sig den perfekta schackspelaren bör detta vara den som vid varje möjlig position, vinner de positioner som går att vinna, och håller remi vid de positioner som inte går att vinna. Låt säga att denna perfekta schackspelaren har uppnått en position där inget drag forcerat leder till vinst, där fem drag leder till remi och 20 drag förlorar under förutsättning att motståndaren spelar optimalt. En exploaterande schackspelare skulle här tänkbart kunna välja ett av de förlorande dragen i hopp om att motståndaren inte hittar den optimala fortsättningen och förlorar istället för att vinna. Denna spelare som hoppas på att motståndaren är dålig bör dock inte längre anses spela optimalt, eftersom den skulle förlora mot sig själv. På samma sätt kommer en person som försöker exploatera motståndaren alltid ha ett negativt förväntat värde mot en motståndare som spelar en nashjämviktsstrategi i poker.

3 Att hitta en nashjämviktsstrategi

3.1 Counterfactual regret minimization

En känd algoritm för beräkandet av en nashjämviktsstrategi är counterfactual regret minimization (CFRM), funnen av Michael Bradley Johanson[3]. Implementationsdetaljer rörande algoritmen är utanför avgränsningen för detta arbete, men vissa restriktioner är viktiga att nämna. För att implementera algoritmen krävs att man lagrar sannolikhetsfördelningen av handlingar för varje informationsmängd i minnet. I limit hold 'em, där man enbart får satsa ett bestämt antal marker, finns det $3.19 * 10^{14}$ informationsmängder. Eftersom det krävs att man lagrar en float för de möjliga handlingarna (fold, call, bet), krävs mer än 4.5 petabytes minne[3]. I no-limit är denna siffra betydligt högre, eftersom antalet möjliga handlingar är fler. Därför är det inte möjligt att hitta en nashjämviktsstrategi för det fullständiga spelet med denna, eller någon annan känd algoritm.

När informationsmängderna har lagrats i minnet förekommer en form av självspelningsprocess som iterativt konvergerar mot en nashjämviktsstrategi. Hur snabbt strategin konvergerar mot nashjämviktsstrategin beror på antalet informationsmängder i spelet. I spelen vi är intresserade av att lösa är det inte genomförbart att nå en exakt nashjämviktsstrategi. Resultatet blir istället en ϵ -nashjämviktsstrategi, där ϵ anger strategins exploaterbarhet. Trots detta har botar producerade genom CFRM varit framgångsrika. En bot kallad Hyperborean som skapades av University of Alberta med denna metod, vann 2009 års Computer Poker Competition[5].

3.2 Abstraktion

För att använda sig av ovan nämnda algoritm krävs det att man förminskar antalet informationsmängder till ett antal som är möjligt att lagra i minnet. Det gör man genom att abstrahera

spelet. Det innebär att man inför regler som spelaren måste förhålla sig till, för att på så vis minska antalet informationsmängder som kan förekomma. Resultatet av dessa införda regler är ett abstraherat spel där en nashjämviktsstrategi kan beräknas. Strategin för det abstraherade spelet används sen som en strategi till det fullständiga spelet. För att denna strategi ska vara effektiv på det fullständiga spelet krävs det att reglerna man inför är väl genomtänkta.

3.2.1 Handlingsabstraktion

De strategiska olikheterna mellan att satsa närliggande antal mörkar är ofta försumbara. I praktiken genomförs oftast satsningar runt storleken på potten. En ofta använd abstraktion är därför att bara tillåta satsningar på storleken av potten, eller att gå all in. Med denna abstraktion, kallad *fcpa* (fold, call, pot, allin), reduceras de möjliga handlingssekvenserna till 9 på första rundan, 56 på floppen, 184 på turnen, och 456 på rivern. Det totala antalet informationsmängder blir då

$$19 * \binom{52}{2} + 92 * \binom{52}{2} * \binom{50}{3} + 276 * \binom{52}{2} * \binom{50}{3} * 47 + 636 * \binom{52}{2} * \binom{50}{3} * 47 * 46 = 3.6 * 10^{13}$$

Trots denna abstraktion har vi alltså ett antal informationsmängder som inte är möjligt att lagra i minnet.

3.2.2 Hinkabstraktion

Hinkabstraktion är en metod använd med framgång för att ytterligare abstrahera spelet. Händer med strategiska likheter slås ihop för att på så vis minska antalet informationsmängder som kan förekomma. Olika metoder för beräkna vilka händer som har strategiska likheter har tidigare beprovats. I det fullständiga spelet finns det en hink för varje hand, det vill säga $\binom{52}{2}$ innan floppen, $\binom{50}{3}$ på floppen, 47 på turnen, och 46 på rivern. Händer som enbart skiljer sig från varandra med färgen på korten har identiska strategiska egenskaper och kan därför paras ihop utan att någon information går förlorad. Denna ihopparning förminskar antalet informationsmängder med som mest 4![6] och räcker inte för att lagra informationsmängderna i minnet.

Antalet informationsmängder med hinkabstraktion och handlingsabstraktion beräknas med nedstående formel, där $P = innanflop$, $F = flop$, $T = turn$, $R = river$ och där index s anger antalet sekvenser på satsningsrundan, och index h anger antalet hinkar på satsningsrundan.

$$(P_s * P_h) + (F_s * P_h * F_h) + (T_s * P_h * F_h * T_h) + (R_s * P_h * F_h * T_h * R_h)$$

Med *fcpa*-handlingsabstraktionen nämnd tidigare är 8 hinkar på varje satsningsrunda rimliga antal hinkar för lagring i minnet på en dator med 8 gb minne.

3.2.3 Handstyrka

Handstyrka definieras som chansen att vinna mot en slumpmässig hand när resterande gemensamma kort har delats ut. För att beräkna handstyrkan delar man ut alla möjliga fortsättningar på given, och delar antalet vunna potter med antalet möjliga fortsättningar. Resultatet är ett värde mellan 0 och 1. Händer med liknande handstyrka kan därefter paras ihop, eftersom de oftast

har strategiska likheter. En enkel metod är sedan att låta varje hink på satsningsrundorna delas upp i ett enskilt intervall med storlek ($1/\text{hinkantal}$), där alla händer med handstyrka i intervallet hamnar i hinken.

3.2.4 Handstyrka i kvadrat

Problemet med att para ihop händer baserat på deras handstyrka är att händer som inte är starka på den aktuella satsningsrundan, men som har potential att förbättras till en stark hand paras ihop med händer som är medelstarka, men som inte har någon chans att förbättras på senare satsningsrundor.

6♥5♥	9♠3♥4♥
K♠4♦	9♠3♥4♥
Q♠5♦	9♠3♥5♥

Figure 2: Tre flophänder med närliggande handstyrkevärde. Den första handen är olik de övriga två på grund av dess potential.

Genom att kvadrera handstyrkorna som en hand kan uppnå på nästkommande satsningsrunda, ger vi högre värde till händer med potential. På så vis kommer händer med mycket potential paras ihop med starka händer.

3.2.5 Procentuella hinkar

När hinkarna utgör ett likformigt intervall mellan 0 och 1 enligt någon av ovannämnda metriker så blir antalet händer som hamnar i hinkarna olika för varje hink. Det behöver inte nödvändigtvis vara dåligt, då hinkar som utgör ett starkt intervall får ett lägre antal händer i sig för att på så vis ge större fokus på dessa viktiga händer. Ett problem är dock att vissa hinkar helt och hållet blir oanvända. Pondera att vi använder oss av 8 hinkar per satsningsrunda. Handen med lägst handstyrka på satsningsrundan innan floppen är 3♣2♥. Den har handstyrkevärde 0.32 och hamnar i hink 3. Handen med högst handstyrka är A♣A♥ med handstyrkevärde 0.85 och hamnar i hink 7. Hinkarna 1, 2 och 8 är således helt oanvända innan floppen. För att åtgärda detta kan man använda sig av procentuella hinkar. Med det menas att varje hink tilldelas ett lika antal händer. Innan floppen finns $\binom{52}{2}$ möjliga händer. Den starkaste hinken av N hinkar tilldelas de bästa $\binom{52}{2}/N$ händerna. Den näst starkaste hinken tilldelas de bästa kvarvarande $\binom{52}{2}/N$ händerna, och så vidare. Samma metod används på övriga satsningsrundor.

3.2.6 Minmaxhinkar

En annan lösning på problemet att vissa hinkar blir oanvända med ett likformigt intervall mellan 0 och 1 är att införa minimum- och maximumvärden för hinkarnas intervall. För händer på respektive satsningsrunda delar man in hinkarna i ett likformigt intervall mellan minimum- och maximumvärdet på satsningsrundan. På så vis försäkras man sig om att alla hinkar blir använda,

men till skillnad från procentuella hinkar innehåller de inte ett lika antal händer. Nedan följer en tabell över värdena för de sämsta respektive bästa händerna på varje satsningsrunda, beroende på om man använder handstyrka eller handstyrka i kvadrat.

	Preflop	Flop	Turn	River
Min HS	0.32	0.11	0.04	0
Max HS	0.85	1	1	1
Min HS ²	0.14	0.02	0.01	0
Max HS ²	0.73	1	1	1

Table 1: Tabell över minimum- och maximumvärden på respektive satsningsrunda.

3.2.7 Historikhinkar

Procentuella hinkar och minmaxhinkar ser till att alla hinkar blir använda, men inte att alla hinksekvenser blir använda. En hand som ligger i den högsta hinken innan floppen kommer så gott som aldrig hamna i den lägsta hinken på floppen. En hand som ligger i en stark hink på en satsningsrunda, kommer med stor sannolikhet ligga i en stark hink på nästkommande satsningsrunda med de ovannämnda hinkabstraktionerna. Historikhinkar ser till att alla hinksekvenser används med samma sannolikhet genom att sätta intervall för hinksekvenser, och inte bara för hinkar. Till exempel kommer intervallet för den starkaste hinken på floppen vara väldigt snävt om handen låg i den starkaste hinken innan floppen. På samma sätt kommer hinkintervallet för den svagaste hinken på floppen vara snävt om handen låg i en svag hink innan floppen.

3.2.8 Vad händer när motståndaren inte följer abstraktionsreglerna?

Så fort motståndaren avviker från abstraktionen finns risken att strategin som är framtagen blir exploaterbar. När motståndaren agerar med en handling som inte var möjlig i abstraktionen måste vi tolka detta till en av de möjliga handlingarna. Med fcpa-handlingsabstraktionen kommer alla satsningar tolkas till en satsning av storleken på potten. En motståndare som vet om att denna tolkning sker kan därför enkelt exploatera detta genom att satsa minimumbeloppet. För att förhindra en sådan enkel exploatering krävs det att fler handlingar görs möjliga i abstraktionen.

4 Resultat

4.1 Hur man utvärderar strategier

I poker beror resultatet till stor del på variansen. Den exakta standardavvikelsen är svår att beräkna, då den beror på strategierna som spelar. Den har tidigare uppskattats till 6 små mörkar per giv[9]. Det betyder att det behövs miljontals givar för att försäkra sig om att resultatet är riktigt. Det är inget problem när strategier producerade av CFRM ställs mot varandra, då en

miljon givar inte tar mer än en minut att spela igenom.

Människor, och andra sorters botar, behöver betänketid för att välja sin handling. Därför är det svårt att inom rimlig tid försäkra sig om det förväntade resultatet mellan en bot och en människa. Dubbelsidiga givar är en metod för att minska standardavvikelsen per giv till 1.6 små mörkar[9]. Dubbelsidiga givar betyder att varje giv spelas två gånger med kortleken sorterad på samma sätt. Den andra gången växlar spelarna positioner. Om den ena spelaren hade tur första gången, kommer således den andra spelaren ha tur andra gången.

Resultatet anges i stora mörkar vunna per 100 givar (bb/100).

4.2 Resultat mellan hinkabstraktioner

Alla strategier som implementerades gjordes med en stackstorlek på 100 stora mörkar. Handlingsabstraktionen för samtliga var fcpa och antalet hinkar var 8 per satsningsrunda. Följande botar implementerades.

p_{hs}

Procentuella hinkar efter handstyrka.

mm_{hs}

Minmaxhinkar efter handstyrka.

mm_{hs}²

Minmaxhinkar efter handstyrka i kvadrat.

h_{hs}

Historikhinkar efter handstyrka.

h_{hs}²

Historikhinkar efter handstyrka i kvadrat.

	p _{hs}	mm _{hs}	mm _{hs} ²	h _{hs}	h _{hs} ²
p _{hs}		-4.2	-7.2	-13.4	-10.2
mm _{hs}	4.2		-3.2	1.6	-2.7
mm _{hs} ²	7.2	3.2		-4.5	-1.3
h _{hs}	13.4	-1.6	4.5		-0.5
h _{hs} ²	10.2	2.7	1.3	0.5	

Table 2: Resultat i bb/100 mellan hinkabstraktionerna över 1 miljon dubbelsidiga givar.

4.3 hhs² mot SimpleBot

Följande match simulerades i Open Meerkat Poker Testbed. Open Meerkat Poker Testbed är ett öppet källkodsprogram där pokerbotar som implementerar Meerkat API kan ställas mot varandra. Meerkat API är ett java API (Application Programming Interface) skapat av University of Alberta, som bland annat används av botar i det kommersiella programmet Poker Academy. Med Meerkat API medföljer en enkel bot kallad SimpleBot. Trots sin enkelhet utgör resultatet mot SimpleBot ett bra riktmärke på grund av botens tillgänglighet för alla. När SimpleBot väljer att satsa så satsar den alltid storleken på potten. Det betyder att ingen handlingstolkning behöver göras. Stackstorleken återställdes vid början av varje giv till 100 stora mörkar var.

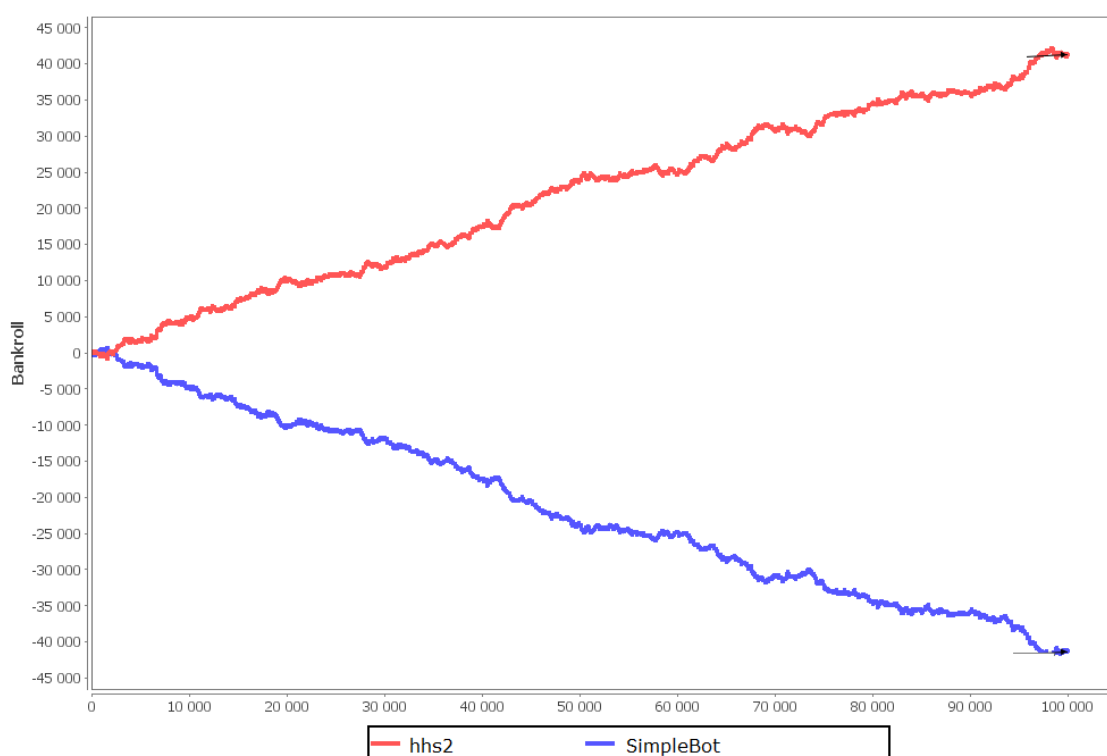


Figure 3: Resultatutveckling över 100000 dubbelsidiga givar. hhs² vann med 10.3bb/100.

5 Slutsats

Botar som använde sig av historikhinkar visade sig prestera bättre än övriga hinkabstraktioner. Handstyrka i kvadrat utgjorde en bättre metrik än handstyrka. Boten med historikhinkar baserade på handstyrka i kvadrat lyckades besegra en icke-trivial pokerbot med god marginal. Resultaten som presenterades i det här arbetet gjorde vissa restriktioner. För att producera en fullständig bot krävs det att handlingsabstraktionen utökas från fcpa-handlingsabstraktionen. Varje giv i matcherna påbörjades med en stackstorlek på 100 stora mörkar. I verkligheten varierar stackstorleken i takt med att ena spelaren vinner delar av den andras stack. För spel i verklig poker krävs därför att strategier beräknas för olika stackstorlekar. Det finns inga absoluta garantier för att samma resultat som de som gavs i denna uppsats gäller för andra stackstorlekar, eller för större handling- och hinkabstraktioner. Produktionen av en fullständig pokerbot är en tidsmässigt kostsam process både implementations- och beräkningstidsmässigt. Resultaten givna här ger därför en bra indikation på var man bör lägga sin tid.

References

- [1] *Going All In for Online Poker*.
Newsweek Magazine, 2005.
<http://www.newsweek.com/id/56438>
Hämtad 2012-02-12.
- [2] Bill Chen, Jerrod Ankenman.
The Mathematics of Poker.
ISBN: 1886070253, 2006.
- [3] Michael Bradley Johanson.
Robust strategies and counter-strategies: Building a champion level computer poker player.
M. Sc. Thesis, University of Alberta, 2007.
<http://webdocs.cs.ualberta.ca/games/poker/publications/johanson.msc.pdf>
Hämtad 2012-02-12.
- [4] Kevin Waugh.
Abstraction In Large Extensive Games.
M. Sc. Thesis, University of Alberta, 2009.
<http://webdocs.cs.ualberta.ca/games/poker/publications/waugh.msc.pdf>
Hämtad 2012-02-12.
- [5] David Paul Schnizlein.
State Translation in No-Limit Poker.
M. Sc. Thesis, University of Alberta, 2009.
<http://poker.cs.ualberta.ca/publications/schnizlein.msc.pdf>
Hämtad 2012-02-12.
- [6] Darse Billings.
Algorithms and Assessment in Computer Poker.
PhD thesis, University of Alberta, 2006.
- [7] M. Osborne and A. Rubenstein.
A Course in Game Theory.
The MIT Press, Cambridge, Massachusetts, 1994.
- [8] Mike Brunner.
Are poker 'bots' raking online pots?
http://www.msnbc.msn.com/id/6002298/ns/technology_and_science-internet_roulette/t/are-poker-bots-raking-online-pots/#.T3dwUNUtpkc
Hämtad 2012-03-31.
- [9] Darse Billings, Morgan Kan.
A tool for the direct assessment of poker decisions.
The International Association of Computer Games Journal, 2006.