ROYAL INSTITUTE
OF TECHNOLOGY

Optimal spelstrategi för yatzy

en jämförande studie av mindre regelförändring

Optimal game strategy for yatzy
a comparative study of minor rule changes

Henriksson, Sam	Engström, Gustav
Kärrstigen 6	Atlasvägen 31
142 51 Skogås	131 34 Nacka
0763379309	0704949447
hensa@kth.se	guseng@kth.se
910530-2773	860811-7555

Examensarbete inom datalogi, grundnivå (Kandidatexjobb) - DD143X – 6 hp
Kungliga Tekniska Högskolan - CSC
Handledare: Wikström, Douglas
Examinator: Björkman, Mårten

12 april 2013

Abstract

This report intends to analyze the difference between the original rules and a modification of the rules for the Scandinavian version of the world famous dice game yatzy (yahtzee). Several works are made previously existing strategies for yatzy and Yahtzee.

The study was conducted using the optimal strategy calculated and implemented in the programming language Java. The execution of the application to create the file containing the optimal strategy was running on leased powerful servers from Amazon and took just over six hours each.

A simulated player played 100,000 games based on the strategy, to obtain data which then was used to plot graphs and analyze the expected score and its standard deviation. The expected value (mean) and standard deviation of the score for the possibility of three or four throws were 233.00 ± 44.82 and 280.07 ± 41.25.

In view of the results drawn the expected conclusion that the four possible throws provides increased probability of getting a high score, unlike the case where the player may only be able to throw the dice three times. The fact that the variance is lower in the case of four throw session is equivalent to that the likelihood to get close to the expected value increases.

Sammanfattning

Denna rapport har för avsikt att analysera skillnaden mellan orginalregelverket och en modifikation av detta för den skandinaviska varianten av det världskända tärningsspelet yatzy (yahtzee). Ett flertal arbeten är tidigare gjorda gällande strategier för yatzy och yahtzee.

Undersökningen har utförts med hjälp av den optimala strategin som beräknades och implementerades i programmeringsspråket java. Exekveringen av programkoden för att skapa filen innehållande den optimala strategin kördes på hyrda kraftfulla servrar från Amazon och tog drygt sex timmar vardera.

En simulerad spelare läts spela 100 000 spel utifrån denna strategi, för att få data som sedan används för att plotta tydliga grafer och analysera förväntat poängsumma och dess standardavvikelse. Förväntat värde (medelvärde) och standardavvikelse av poängsumman för möjligheten till tre respektive fyra kast blev 233.00±44.82 respektive 280.07±41.25.

Med hänsyn till resultaten dras den förväntade slutsatsen att fyra möjliga kast ger ökad sannolikhet att få en hög poängsumma, till skillnad från det fall då spelaren endast får möjlighet till tre kast. Det faktum att variansen är lägre för fallet med fyra slag är ekvivalent med att sannolikheten att komma nära det förväntade värdet ökar.

Sammarbete inom gruppen

Detta arbete har utförts av en grupp om två personer. Stor del av det arbete som gjorts har vi utfört tillsammans vid varsin dator i samma arbetsrum alternativt vid samma dator.

Detta gäller både implementation av alla delprogram, analys av bakgrund och tidigare arbeten inom samma område samt författandet av denna rapport.

Trots det omfattandet samarbete inom alla områden har uppdelning skett. Henriksson har under den inledande analysen av problemet sammanställt uträkningar för sannolikheter för olika slag. Engström var under den senare tiden av implementationen mer involverad i just denna, samtidigt som Henriksson ägnade mer tid åt författandet av rapporten. Då den algoritmen skrivits förhållandevis klart har fullständigt sammarbete skett i syfte att finna fel och rätta till dessa.

Innehållsförteckning

Introduktion

Yatzy är den skandinaviska versionen av tärningsspelet "yahtzee" vilket uppfanns på 1950-talet och går ut på samla poäng från tärningar, under en 15 varv lång spelomgång. [1] Yatzy spelas vanligtvis mot andra spelare, men inget hindrar en ensam spelare att spela ett visst antal omgångar för att jämföra och försöka slå sina tidigare resultat. Det faktum att antalet spelare inte är en begränsande faktor har gjort det till ett mycket omtyckt och ofta spelat tärningsspel.

Det stora flertalet spel grundas på någon sorts strategi och inte enbart på tur. Somliga spel, så som schack, har så pass hög komplexitet att inte ens dagens kraftfulla datorer klarar att skapa en felfri robotspelare, medan andra spel, så som "tre i rad", är så pass "enkla" att en människa kan lära sig att bemästra den optimala strategin snabbt. Mellan dessa ytterligheter av spel placerar sig tärningsspelet yatzy. Den optimala strategin, vad spelaren skall göra i varje möjligt läge för att maximera de slutgiltiga poängen, för yatzy är inte självklar, men är möjligt att beräkna med hjälp av datorns hjälp, vilket ett flertal redan gjort. [2] [3] [4] [5] [6] [7]

När den optimala strategin är beräknad och lagrad kan ett datorprogram simulera en ensam spelare som slår tärningarna slumpar och sedan sparar alternativt skriver in poäng på den plats som är optimalt enligt den optimala strategin för det specifika fallet. Ett flertal av de som programmerat den optimala strategin har simulerat den ett stort antal gånger, och på sådant sätt fått fram ett medelvärde för slutsumman, det så kallade väntevärdet. Larsson och Sjöberg fick det värde till cirka 248. Detta värde kan även beräknas direkt från den optimala strategin utan simulering, vilket enligt deras arbete är det exakta värdet. Det kalkylerade värdet fick Larson och Sjöberg till 248.92. [4, p. 7] Utifrån dessa beräkningar kan även variansen fås, vilken indikerar hur stora svängningar som kan förväntas. Låg varians tyder på att sannolikheten att komma nära väntevärdet är högt. [8]

För att finna det bästa valet i precis varje möjligt läge i spelet räknar ett program ut alla möjliga tillstånd som kan tänkas finnas i spelet för att sedan ta ut det bästa valet i just det tillstånd spelet befinner sig.

Syfte

Då den optimala strategin för den skandinaviska versionen av yatzy redan beräknats och analyserats ett flertal gånger har vi valt att undersöka den förväntade slutsumman när en mindre förändring av spelet görs, i det fall den optimala strategin används. Vi har valt att undersöka påverkan av förändringen som lyder att en spelare under varje omgång får fyra möjligheter att slå tärningarna, istället för tre. Resultatet av denna modifikation skall jämföras med utkomsten från den optimala strategin för originalregelverket.

Analysen kommer behandla skillnad i väntevärde och i varians. Hur stor blir den faktiska påverkan av att ge en spelare ett extra slag varje omgång, utifrån det faktum att denna spelar efter den optimala strategin?

Den huvudsakliga begränsning vi valt att göra är endast undersöka orginalregelverket mot att spelaren får fyra kast per runda. Det förväntas att det är mer fördelaktigt att tillåtas slå tärningarna fyra gånger istället för tre då det ökar sannolikheten att nå mer fördelaktiga kategorisummor. Detta eftersom varje ”dålig” tärning som slås om har en viss sannolikhet att bli ”fördelaktig”, vilken är strikt högre än noll.

Bakgrund

En bidragande anledningen till valet av problemformuleringen och området yatzystrategier i stort är det behandlar stora delar at de matematikkurser författarna nyligen studerat. Detta tillsammans med det faktum att problemetlösningen innefattar en stor del programmering och logiskt tänkande gjorde valet förhållandevis enkelt. En ytterligare orsak till ämnesvalet är att båda författarna kontinuerligt spelar yatzy och att det uppskattas mycket.

Historia

Det skandinaviska spelet yatzy härstammar från det mycket likartade spelet yahtzee, vilket innan kommersialiseringen kallades för "yacht spelet". Spelet uppfanns av ett rikt kanadensiskt par som spelade det med sina vänner på sin yacht år 1954. Två år senare sålde de rättigheterna till spelmakaren Lowe som senare kom att ge spelet namnet "yahtzee". [1] I den utomskandinaviska varianten av tärningsspelet som än idag heter yahtzee är det ett flertal saker som skiljer sig i reglerna mot den skandinaviska varianten yatzy. Bland annat ger bonusen 100 poäng, varken "par" eller "tvåpar" existerar samt att det för många kategorier där antalet prickar på tärningen inte spelar någon roll. Exempelvis ger kåk alltid 25 poäng oavsett vilka tärningar man fått. [9]

Enligt Hasbro som numera äger rättigheterna till den utomskandinaviska versionen av yahtzee spelar upp till 100 miljoner spelet regelbundet. [10] Då yatzy blev ett mycket populärt spel även i skandinavien släpptes två varianter av spelet, av speltillverkaren Alga. Den första av dessa två kom att kallas "maxi-yatzy" i vilket man har sex tärningar och reglerna är utformade så att strategin blir mer avancerad och spelarna får tänka mer. På 1980-talet släppte de även varianten "crazy-yatzy" vilket utger sig för att vara ett annorlunda spel med många märkliga regler. [11]

Spelregler

Fem tärningar slås totalt tre gånger, mellan vilka man kan välja att spara ett visst antal tärningar. När de tre omgångarna är klara bokförs poängen och turen går över till nästa spelare. Poängen grundas på vilka siffror tärningarna visar, vilka korresponderar mot olika kategorier som skall fyllas. Det är totalt 15 kategorier, vilket är ekvivalent med 15 varv bland spelarna. Maximal poängsumma i skandinaviska yatzy är 374, till skillnad från den utomskandinaviska versionen som har maximal summa på 1575. Fullständiga regler för den skandinaviska varianten av yatzy finns återgivet i [appendix d].

Tidigare studier

Då yatzy och yahtzee är mycket populära spel har det gjorts ett flertal arbeten om spelen. Många av dessa handlar om strategier och då bland många andra även den optimala strategin. Yahtzee som är den mest utbredda varianten av spelet och det finns ett flertal arbeten som behandlar detta, exempelvis ”An optimal strategy for yahtzee” skriven av James Glenn år 2006. Där beskrivs hur den optimala strategin för yahtzee beräknas samt resultatanalys av denna strategi i jämförelse med andra strategier så som den “giriga algoritmen” eller att endast satsa på att få yatzy vilket kan göras med hjälp av den optimala strategin. [2]

Trots att yatzy inte är lika utbrett som den utomskandinaviska varianten så finns det arbeten som behandlar denna variant med avseende på den optimala strategin, till exempel ”Optimal yatzy strategy” skriven av Marcus Larsson och Andreas Sjöberg, vilken gjordes som ett kandidatexamensarbete vid Kungliga Tekniska Högskolan i Stockholm år 2012. Detta arbete beskriver i detalj hur den optimala strategin beräknats och resultatet analyseras genom att simulera ett stort antal spel där programmet använder denna strategi. När den optimala strategin beräknas fås en förväntad poäng för resterande omgångar vid varje tillstånd. Vid simulering av 20 000 spel fick de i genomsnitt en slutsumma på 248.92. [4]

Sjöberg och Larsson har i sitt arbete använt sig av två skilda tekniker för att optimera de mycket tidskrävande beräkningarna. Den första av dessa två är parallellisering vilket går ut på att dela upp arbetet i separata delar som sedan beräknas parallellt var för sig. Detta ger en avsevärd sänkning av tidsåtgång då de utvecklat programmet så att datorn kan göra ett flertal oberoende beräkningar samtidigt.

En mer invecklad men mycket effektiv optimering är att använda det som de kallar “keepers”, vilket berör beräkningen av det förväntade värdet för olika steg i spelet. De beskriver i arbetets sektion 3.6 vad skillnaden är mellan den “vanliga” beräkningsvägen som hela tiden tar fram det viktade väntevärdet med denna mer effektiva metod som lagrar en del av beräkningarna och på så sätt undviker upprepningar av lika beräkningar. Ett exempel de tar upp som belyser förbättringen är när samma beräkning behöver göras. Om en spelare har handen {1,1,3,4,5} och väljer att slå om den första 1an så är sannolikheten och därmed väntevärdet identiskt med om spelaren valt att slå om den andra 1an.

Optimeringen går som tidigare nämnts ut på att spara undan och återanvända värden från beräkningar. Tack vare “keepern” slapp de göra explicita beräkningar av sannolikheter, då de görs inbäddat i denna optimering. [4, p. 6] Just denna metod med “keepers” används exempelvis i arbetet ”A Nearly Optimal Computer Player in Multi-player Yahtzee” skriven av Jakub Pawlewiez, vilken Larsson och Sjöberg refererar till i sin text. [6] [4]

De övriga arbeten som gjorts har kommit fram till liknande resultat med liknande implementation av algoritmen. De har i likhet med Glenn och Pawlewicz analyserat den utomskandinaviska varianten. Pawlewicz beräknade väntevärdet till 254.589 vilken bortsett från avrundningen överensstämmer med Glenns värde, 254.59, vilket har en tillhörande standardavvikelse om 59.61. Detta värde är även identiskt med resultatet från Verhoeffs arbete, som dock fått standardavvikelsen till 60, vilket kan bero på avrundning.

I Glenns senare arbete, från år 2007, har han gått djupare in på själva implementationen av beräkningen och gör ytterligare förbättringar av det tidigare arbetet. En studie lik många av de andra gjordes år 2012 som ett kandidatexamensarbete vid Kungliga Tekniska Högskolan av Felldin och Sood. Dessa fick efter simulering väntevärdet till 221.67906, vilket även de analyserade den utomskandinaviska varianten. [5] [2] [3] [4] [6] [7]

Anledningen till den stora skillnaden mellan somliga resultat beror på vilken variant som är analyserad, den skandinaviska eller utomskandinaviska.

Optimal strategi

Det inom detta område ofta använda uttrycket “optimal strategi” syftar till ett sorts regelverk som en tänkt spelare skall följa. För varje möjligt tillstånd i ett spel, vilket beror på följande faktorer:

- hur många kategorier som är ifyllda
- om bonus är uppnådd eller hur långt ifrån bonus spelaren är
- vilka tärningar som slagits
- vilket slag i omgången spelaren befinner sig på

På denna nod finns en instruktion lagrad som säger vad spelaren skall göra i just detta skede. Som tidigare nämnts så har just denna strategi beräknats ett flertal gånger, så väl för den skandinaviska varianten som för den utomskandinaviska. I båda dessa fall är beräkningarna mycket likartade, då den stora skillnaden endast är skillnad i hur många poäng en viss tärningskombination ger. [4] [6]

Metod

Den metod som använts i detta arbete för att beräkna den optimala strategin är av typen ”brute force”, att gå igenom varje tänkbar möjlighet och skeende i spelet. Spelet kan representeras av en graf bestående av noder och kanter, vilket på fil lagras som en stor matris.

Grafrepresentation

Som gjorts i många av de tidigare studierna representeras en hel spelomgång som en graf bestående av hörn och kanter. Varje hörn motsvarar ett av det stora antalet tillstånd i spelet, vilket grundas på följande faktorer:

- antal poäng som saknas i de sex överst kategorierna för att nå bonus: $|0...63| = 64$
- vilken “hand” vi har: (alla unika kombinationer av 5 tärningar) $|1...252| = 252$, upprepning utan avseende på ordning, [12, p. 12] $\binom{n+r-1}{r-1} = \binom{5+6-1}{6-1} = \binom{10}{5} = 252$, r=antal utfall, n=antal tärningar
- vilka kategorier som är ifyllda: (alla kombinationer av fyllda respektive icke fyllda kategorier) $2^{15} = 32768$
- vart spelaren ligger i sin tur: efter första slaget, efter andra och efter tredje.[2] [4]

Kanterna representerar vilken “väg” spelaren tar, vilka tärningar som sparas respektive i vilken kategori spelaren väljer att sätta poäng. Antal kanter som representerar behållning av tärningar är maximalt 467(beräknas som antalet unika händer med n=5, r=7), men varierar beroende på utgångshanden, eftersom endast det unika sparandet av händer är relevant. Från noden som representerar att det sista slaget för turen är slaget finns det endast mellan 1 och 15 noder, alla permutationer av fyllda kategorier med endast en ofylld.

Det som utgör den optimala strategin är att det på noderna som representerar tillståndet efter det första och det andra slaget finns information om vilka tärningar som skall slås om, medan det på noden som representerar tillståndet efter det tredje slaget finns information om vilken kategori poängen skall sättas på. Detta innebär att det kan visualiseras som att det på varje nod finns information om just det optimala draget, för varje möjligt tillstånd i spelet.

Gällande de faktorer som påverkar vilket tillstånd spelaren befinner sig i, så spelar de faktiska poängen spelaren har ingen roll, med undantag för de i de sex översta kategorierna. Detta är ekvivalent med att det inte spelar någon roll var spelaren “kom” från, utan endast det nuvarande läget. Detta fenomen av “minneslöshet” innebär att grafen benämns som markovsk i matematiska termer. [13, p. 2]

Denna metod att visualisera grafen är den samma som använts av Larsson och Sjögren samt Glenn, vilken är en så kallad “brute force” metod, vilket innebär att exakt alla möjliga tillstånd spelaren kan hamna i är representerat och att beräkningen görs genom att hela grafen traverseras för att finns den bästa vägen i alla lägen. [4] [2]

Basfall

Traverseringen inleds i den ände av den tänkta grafen där det finns så få möjligheter som möjligt, det vill säga det tillstånd när alla tre slag är slagna, alla kategorier utom en är ifyllda. Dessa tillstånd kallas basfall och utgör grunden för traverseringen. Algoritmen går igenom alla olika permutationer av fyllda kategorier där endast en kategori är ofylld, (15 stycken), för varje möjlig summa hos de översta sex raderna (64 stycken) och för varje möjlig hand spelaren kan ha (252 stycken).

Under själva traverseringen beräknas förväntade poäng på varje tillstånd, vilket representerar hur många ytterligare poäng spelaren kan “förvänta” sig om den får en viss hand. Det speciella med basfallen är det förväntade värdet är det exakta värdet, eftersom spelaren vid ett sådant fall endast har möjlighet att sätta tärningarna på en enda kategori. De poäng som lagras som de förväntade är således så många poäng som ges för att spara den valda handen i just den kategorin, vilket är anledningen till att traverseringen kan börja just här - ingen annan data behövs.

Rekursion

När basfallen är beräknade är resterande del av algoritmen mycket lik under hela den återstående beräkningen. Steget efter basfallen är för det fall då spelaren har slagit sitt andra slag medan resterande faktorer är lika de i basfallen. De värden som nu skrivs till varje nod är denna gång det högsta förväntade värdet, då spelaren väljer att behålla en viss hand.

Detta beräknas genom att algoritmen går igenom alla olika sätt tärningarna kan behållas och för varje sådant fall tar fram sannolikheten för att nå de olika händerna i nästa steg i grafen från en förkalkylerad databas. Förväntat värde för behållningen beräknas genom att ta ut sannolikheten ur

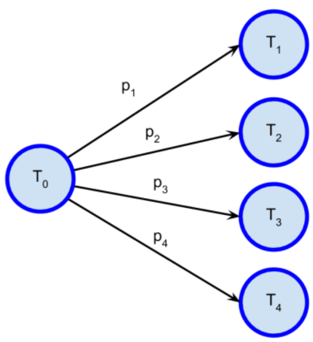
databasen för att få en viss hand och multiplicerar det med det från tidigare lagrade förväntade värdet för en viss nästa hand. Alla dessa sannolikheter summeras till ett och det förväntade värdet fås genom följande formel:
Ett tillstånd $T = \{h, kat, k\}$, h=unik hand, kat=poängkort med unikt ifyllda kategorier och poängsumma av sex översta kategorierna, k=kast (1,2,3(,4)).

a. $Exp(T_0) = \sum_{j=1}^{252} \sum_{i=0}^{31} p(T_j | M(m_i, T_0)) * Exp(T_j)$ (Se figur 1)

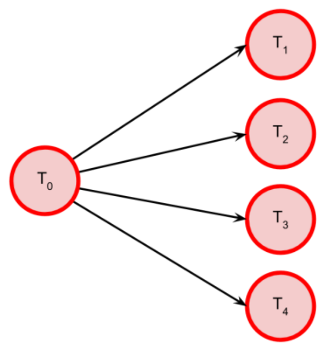
$M(m, T)$ = återstående hand efter maskning m.

$Exp(x)$ = förväntad kommande poängsumma för tillstånd x, där x är tillståndet efter första respektive andra kastet.

b. $Exp(T_0) = \max_{i=1}^{15} Exp(T_i)$ (Se figur 2)



Figur 1 - förväntat poäng vid kast



Figur 2 - förväntat poäng vid placering av poäng i kategori

Förutom förväntat värde skrivs information om det drag som ger just detta maxvärde alternativt för det tredje kastet vilken kategori spelaren skall sätta handen på.

Detta förfaringssätt fortlöper för alla tillstånd, det följande steget är när spelaren endast slagit ett slag. När detta är gjort för alla möjliga poäng för de övre raderna, för varje hand och för varje tom kategori fortlöper algoritmen med alla kombinationer av två ofyllda kategorier och så vidare fram till att alla kategorier är ofyllda, vilket är så poängkortet ser ut när spelaren börjar.

Efter varje runda sparas ett förväntat värde undan för just den rundan med avseende på hur många kategorier som är ifyllda, samt för varje hand som är möjlig att få på det första slaget.

Detta görs genom att precis som tidigare iterera över alla möjliga händer men istället för att ta fram sannolikheten för att få en viss hand med en viss maskning tas sannolikheten att få en viss hand på första slaget, det vill säga sannolikheten att få en viss hand utan att hålla någon tärning.

Det sista fallet är när spelaren ska välja en kategori att sätta sina tärningar på. Detta liknar mycket det fallet där algoritmen undersöker vilken maskning som är en mest fördelaktiga. Till skillnad från maskningen undersöks istället vilken kategori som är mest fördelaktig, i avseende att ge högst förväntat värde. Genom att testa att sätta handen på alla kategorier som ej ännu är ifyllda kan det optimala valet av kategori fås eftersom det förväntat värde för de då unikt ifyllda kategorierna finns sparat.

Beräkning av sannolikheter

Sannolikheterna beräknas för varje tänkbart kast med noll till fem tärningar enligt följande:

Sanolikheten för ett visst fall kan formuleras matematiskt som $\varepsilon * (1/6)^d$ utifrån en hand tärningar D. Låt H_0 vara handen med tärningar spelaren har från början och H_1 den hand spelaren har för avsikt att få.

$\varepsilon =$ antal permutationer som kan skapas av D.

(Exempel: $D = \{1,1\} \; -> \; \varepsilon = 1 \; och \; D = \{1,2\} \; -> \; \varepsilon = 2$).

$d =$ antal omslagna tärningar.

$D =$ de omslagna tärningarna = $H_1 \backslash H_0$ `

$P\{x_0, x_1, ..., x_n\} = \varepsilon$

De sex fall från att ingen tärning slås om till att alla tärningar slås om, bearbetas enligt följande exempel.

1. Då ingen tärning slås om är sannolikheten 1 om $H_0 = H_1$, annars 0.
2. Då en tärning slås om finns ett möjligt utfall utifrån att $\{x\} = H_1 \backslash H_0$.
3. Då två tärningar slås om finns två möjliga utfall utifrån att $\{x, y\} = H_1 \backslash H_0$:
    a. $x = y$, vilket nås med sannolikheten $1 * \left(\frac{1}{6}\right)^2 = \frac{1}{36}$ 1*(1/6)^2=(1/36), då $P\{x, x\} = 1$.

b $x! = y$, vilket nås med sannolikheten

$$2*\left(\frac{1}{6}\right)^2 = \frac{1}{18}, \text{ då } P\{x,y\} = 2$$

4 Då tre tärningar slås om finns tre möjliga utfall utifrån att $\{x,y,z\} = H1 \setminus H0$:

a $x = y = z$, vilket nås med sannolikheten

$$1*\left(\frac{1}{6}\right)^3 = \frac{1}{216}, \text{ då } P\{x,x,x\} = 1.$$

b $x \neq y,\ x \neq z,\ y \neq z$, vilket nås med sannolikheten

$$6*\left(\frac{1}{6}\right)^3 = \frac{1}{7776} \text{ , då } P\{x,y,z\} = 6 \text{ .}$$

c $x \neq y,\ x = z,\ y \neq z$, vilket nås med sannolikheten

$$3*\left(\frac{1}{6}\right)^3 = \frac{1}{72}, \text{ då } P\{x,y,x\} = 3.$$

5 Då fyra tärningar slås om finns fem möjliga utfall utifrån att $\{x,y,z,w\} = H_1 \setminus H_0$:

a $x = y = z = w$, vilket nås med sannolikheten

$$1*\left(\frac{1}{6}\right)^4 = \frac{1}{1296}, \text{ då } P\{x,x,x,x\} = 1 \text{ .}$$

b $x = y,\ z = w,\ x \neq z$, vilket nås med sannolikheten

$$6*\left(\frac{1}{6}\right)^4 = \frac{1}{216}, \text{ då } P\{x,x,y,y\} = 6 \text{ .}$$

c $x \neq y,\ z \neq y,\ w \neq z,\ x \neq w,\ y \neq w$, vilket nås med sannolikheten

$$24*\left(\frac{1}{6}\right)^4 = \frac{1}{54}, \text{ då } P\{x,y,z,w\} = 24 \text{ .}$$

d $x = y = z\ x \neq w$, vilket nås med sannolikheten

$$4*\left(\frac{1}{6}\right)^4 = \frac{1}{324}, \text{ då } P\{x,x,x,w\} = 4.$$

e $x = y,\ x \neq z,\ x \neq w,\ z \neq w$, vilket nås med sannolikheten

$$12*\left(\frac{1}{6}\right)^4 = \frac{1}{108}, \text{ då } P\{x,x,z,w\} = 12 \text{ .}\backslash$$

6 Då fem tärningar slås om finns sju möjliga utfall utifrån att $\{x,y,z,w,q\} = H_1 \setminus H_0$:

a $x = y = z = w = q$, vilket nås med sannolikheten

$$1*\left(\frac{1}{6}\right)^5 = \frac{1}{7776}, \text{ då } P\{x,x,x,x,x\} = 1 \ .$$

b $x = y = z = w,\ x \neq q$, vilket nås med sannolikheten

$$5*\left(\frac{1}{6}\right)^5 = \frac{5}{7776}, \text{ då } P\{x,x,x,x,q\} = 5 \ .$$

c $x = y = z\ x \neq w,\ w = q$, vilket nås med sannolikheten

$$10*\left(\frac{1}{6}\right)^5 = \frac{5}{3888}, \text{ då } P\{x,x,x,w,w\} = 10 \ .$$

d $x = y = z,\ x \neq w,\ x \neq q,\ w \neq q$, vilket nås med sannolikheten

$$20*\left(\frac{1}{6}\right)^5 = \frac{5}{1944}, \text{ då } P\{x,x,x,w,q\} = 20 \ .$$

e $x = y,\ x \neq z,\ z = w,\ x \neq q,\ z \neq q$, vilket nås med sannolikheten

$$30*\left(\frac{1}{6}\right)^5 = \frac{5}{1296}, \text{ då } P\{x,x,z,z,q\} = 30 \ .$$

f $x = y,\ x \neq z,\ x \neq w,\ x \neq q,\ z \neq w,\ z \neq q,\ w \neq q$, vilket nås med sannolikheten

$$60*\left(\frac{1}{6}\right)^5 = \frac{1}{648}, \text{ då } P\{x,x,z,w,q\} = 60 \ .$$

g $x \neq y,\ x \neq z,\ x \neq w,\ x \neq q,\ y \neq z,\ y \neq w,\ y \neq q,\ z \neq w,\ z \neq q,\ w \neq q$, vilket nås med sannolikheten

$$120*\left(\frac{1}{6}\right)^5 = \frac{5}{324}, \text{ då } P\{x,y,z,w,q\} = 120.$$

Implementation av regelförändringar

I syfte att undersöka vad som sker när varje spelare under varje tur låts slå tärningarna fyra gånger istället för tre görs mindre förändringar i programkoden. Detta påverkar inte förfaringssättet i stort, men antalet tillstånd som behöver beaktas ökar med en faktor $4/3$, då det finns ett tillstånd som representerar varje av de fyra fallen: efter första slaget, efter andra slaget, efter tredje slaget och efter fjärde slaget. Då denna förändring ökar antalet beräkningar så kommer beräkningstiden också öka med ovan nämnd faktor.

Skapandet och användandet av den optimala strategin

I den representativa grafen finns nu ett stort antal noder innehållande väntevärde och instruktion om det optimala draget. Vid användande av den optimala strategin är väntevärdena ej nödvändiga längre, utan det enda som behöver sparas är instruktioner för vad spelaren skall göra, för varje möjligt tillstånd i spelet. Den faktiska optimala strategin är således en fil av instruktioner för spelaren. Instruktionen är antingen en kategori som spelaren skall sätta sin hand på eller hur spelaren skall behålla tärningarna innan nästa kast.

Varje instruktion utgörs av en byte, vilket är utrymmesmässigt tillräckligt eftersom kategoriindex går mellan $0-14$ (0-indexerat) och maskningen som består av ett fem-bitars binärt tal, där noll på en plats betyder slå om och ett betyder behåll tärningen på samma plats i handen. Varje hand krävs vara sorterad för att maskningen skall peka på rätt tärning. Då ett fem-bitars tal går från $0-31$ så kan även detta representeras av en byte. Under beräkning lagras dessa bytes i en lista av dimension tre. Efter beräkning av alla tillstånd skrivs listan till fil, byte för byte hexadecimalt.

För att exempelvis låta en simulator använda den optimala strategin låts den slumpa fram en hand, vilken sorteras, varpå information om antal poäng från bonus, vilka kategorier som är ifyllda och vilket slag spelaren tillsammans med handen pekar, på en unik plats i filen.

Optimeringar

Trots dagens kraftfulla datorer och det faktum att yatzy inte är något av de mest komplexa spelen så tar beräkningen av den optimala strategin lång tid. Utifrån detta förhållande är ett antal optimeringar implementerade. De två allmänt begränsande faktorerna vid beräkning är arbetsminne och processorkraft.

För att undvika kontinuerliga beräkningar av samma sak, vilket belastar processorerna mycket, så kan många värden förberäknas och lagras i arbetsminnet. Detta är klart fördelaktigt så länge det förberäknade går att lagra på det snabba arbetsminnet. I annat fall lagras det på det virtuella arbetsminnet vilket lagras på disk och är mycket långsammare vid läsning.

Förberäknade poäng för händer

En tidsbesparing kan uppnås genom förberäkna och lagra data som används ofta i arbetsminnet. En stor del av den information som behövs vid beräkning av strategin är hur mycket en viss hand H maximalt kan ge då den lagras i en viss kategori K, utifrån ett visst antal poäng i de sex översta raderna P. Alla dessa förberäknas och lagras i en tredimensionell lista av bytes. H går från $1-252$ (antal möjliga händer, 1-indexerad), K från 1 till 15 (antal kategorier i yatzy, 1-indexerad) och det möjliga antalet väsentliga poäng för de övre sex raderna är $0-63$ (0-indexerat), där 63 representerar att tillräckligt med poäng för bonus har uppnåtts och att denna har utdelats till spelaren. Detta ger, utifrån att en byte är 8 bitar och de vanliga reglerna, en lista som kräver drygt 0.24 mb i arbetsminnet.

Förberäknade sannolikheter

Samma teknik som ovan används även för alla de sannolikheter som behöver beräknas: sannolikheten att från en hand H_0 få handen H_1 utifrån att en viss uppsättning av tärningarna i H_0 behålls. Både H_0 och H_1 kan väljas på 252 olika sätt och antalet möjliga sätt att spara tärningar är 32. Denna databas med information är mer minneskrävande då sannolikheterna måste lagras som doubles (64 bitar på ett 64-bitars operativsystem) eftersom det är decimaltal. Det ger en total minnesåtgång om drygt 16 mb.

Parallellt arbete

På grund av att en mycket stor andel av beräkningarna inte är sekventiellt beroende av varandra kan ett flertal av dessa beräknas parallellt. Med hjälp av den inbyggda funktionaliteten i java för uppdelning av körningar på olika processortrådar kan körtiden sänkas betydligt. Vid beräkning av den optimala strategin för standardregler är det användbart med upp till 411840 parallella trådar. Denna siffra kommer från binomialkoffecienten för 15,6 och/eller 15,7 vilket representerar antalet permutationer av kategorier med sex respektive sju ifyllda. Detta multipliceras med 64 då varje sådant poängkort finns i 64 varianter på grund av summa av de övre kategorierna.

$$\binom{15}{6} = \binom{15}{7} = 6435$$

$$6435 * 64 = 411840$$

I teorin ger detta en minskad tidsåtgång med faktorn 411840 på just dessa två fall, vilket dock ej har varit möjligt att testa på grund av avsaknandet av resurser, till skillnad från endast en tråd körs åt gången.

Teknik

En viktig faktor i detta arbete är beräkningskraft då det är en mycket stor mängd beräkningar som krävs. Till alla dessa har olika datorer använts.

Mjukvara

Programkoden som använts för att beräkna den optimala strategin är skrivet i programmeringsspråket java. Programmet har sedan exekverats på en dator med Ubuntu 12.10 med java-7 runtime enviroment.

Hårdvara

På grund av det höga kravet på beräkningshastighet för att kunna genomföra beräkningarna inom rimlig tid har koden för skapandet av den optimala strategin körts på fjärrservrar hyrda från den amerikanska nätbutiken Amazon. Produkttypen kallas "Cluster Compute Instances" vars tekniska specifikation återfinns i appendix a. Simulering av spelade omgångar är relativt "lätta" beräkningar och har därför utförts på mindre kraftfulla salsdatorer på Kungliga Tekniska Högskolan i Stockholm. Specifikation för dessa återfinns i [appendix b].

Simulering utifrån optimal strategi

När den optimala strategin beräknats lät vi en salsdator på KTH simulera 100 000 spel, där tärningarna slumpvis valdes med hjälp av javas inbyggda slumpgenerator. Slutgiltig poängsumma från varje spel lagras i en fil. Utifrån dessa värden kan väntevärdet beräknas (genomsnittlig poängsumma) samt även spridningen, i form av varians och standardavvikelse.

Inspiration och lånad kod

Den algoritm som skall användas för att lösa den ställda frågeställningen har beräknats av ett flertal personer. På grund av detta i samband med att frågeställningen i huvudsak inte handlar om algoritmen i sig, utan själva jämförelsen av regelförändringar har valet gjorts att ta inspiration från källkod som publicerats som öppen källkod på Github. Denna källkod har en medföljande textfil [appendix c] som behandlar upphovsrätt för arbetet, i vilken det tydligt framgår att koden både får modifieras och användas, så länge hänvisningar görs till upphovsmännen. [14] [4]

De huvudsakliga delar som lånats från det ovan nämnda projektet är programstruktur gällande parallellisering och datastrukturer. Mycket av detta har varit tvunget att modifieras för att frågeställnigne skall vara möjligt att svara på, samtidigt som valet gjorts att inte använda sig av deras optimeringsmetoden kallad ”keeper”.

Den mest betydande delen i algoritmen, det vill säga den som beräknar det viktade poängsummorna har implementerats utan direkt inspiration från Sjögren och Larsson, då ”keeper” ej används och därmed blir algoritmerna mycket olika.

Resultat

Efter implementation av den optimala strategin exekverades beräkningarna först på en dator med 8 processorkärnor som körde Windows 7 som operativsystem. Körningen för tre kast tog då en bit över 15 timmar att beräkna medan körningen för fyra kast tog över 20 timmar.
Två fjärrservrar hyrdes då in med vardera 32 virtuella processorkärnor som körde Ubuntu 12.10 som operativsystem. Detta resulterade i en markant minskning av tidåtgången för de båda körningarna. Beräkningarna för tre kast tog då 6.5 timmar och strax över 11 timmar för beräkningen för fyra kast.

Resultaten av beräkningarna sparas till fil. Storleken på filen är därför direkt beroende av antalet tillstånd. Storleken på beräkningarna för tre respektive fyra slag blir därför olika.
Där varje tillstånd består av en byte kommer filens storlek vara antalet unika tillstånd, räknat i byte.

För filen med tre slag gäller:

$$\textit{Antal slag} * \textit{antal unika händer} * \textit{antal unika poängkort} =$$

$$3 * 253 * (2^{15} * 64) = 1591738368\, B \approx 1.59 GB$$

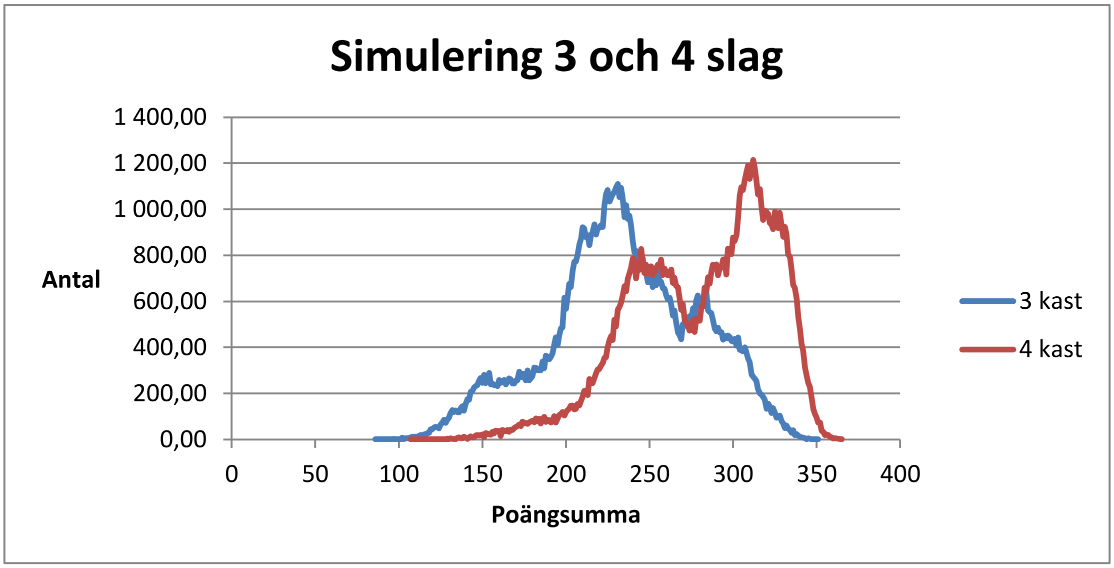
För fyra slag gäller:

$$4 * 253 * (2^{15} * 64) = 2122317824\, B \approx 2.12 GB$$

När beräkningarna var färdiga kunde simuleringar köras med hjälp av ett java program som ”spelar” yatzy och beroende på vilket tillstånd i spelet programmet befinner sig i, behålla vissa tärningar eller sätta tärningarna på en viss kategori som den optimala strategin föreslår.

Simulering

Simuleringen kördes 100 000 gånger med den optimala strategin för tre respektive fyra kast. Utifrån dessa körningar gavs medelpoäng (väntevärde) samt en lista med alla slutsummor. Utifrån denna lista beräknades varians och standardavvikelse, samtidigt som den oftast förekommande poängsumman kunde finnas, se (Tabell 1). Dessa värden användes även för att rita en illustrerande graf för förhållandet mellan poängsumma och antal förekomster, för tre respektive fyra kast, se (Graf 1).



Graf 1

Väntevärde och varians

Väntevärdet $V(X)$ och standardavvikelsen $D(X)$ ges av följande formler och är jämförelsevärden för att analysera hur hög sannolikheten är att få ett värde nära det förväntade värdet $E(X)$, där $X = \{x_0, x_1, ..., x_n\}$ i vilken x_n representerar en poängsumma från simuleringen.

$$V(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$D(X) = \sqrt{V(X)}$$

[8]

	3 kast	4 kast
Genomsnittlig slutsumma	233.00	280.07
Varians	2009.20	1701.17
Standardavvikelse	44.82	41.25
Mest förekommande poängsumma	231	312

Tabell 1

Korrekthet

Korrektheten för den optimala strategin är fullständig på grund av att rekursionen som görs går igenom alla möjliga tillstånd i spelet och eftersom denna process är markovsk så är vägen till ett tillstånd oväsentligt. Från varje givet tillstånd är det säkert att den optimala vägen väljs i ett givet tillstånd, eftersom alla möjliga vägar är testade och den väg med högst förväntat värde är den som följs.

Allt detta förutsätter självklart att tärningarna som används är korrekta, med vilket menas att de sex sidorna numrerade från ett till sex uppkommer med identisk sannolikhet, om (1/6). Statistiskt sätt är sannolikheten att under ett helt spel enbart slå "bra" eller "dåliga" händer är så pass låg, att sannolikheten för att få ett resultat i närheten av det förväntade värdet är högt.

Diskussion

Det resultat vi har fått för spelomgångar där spelaren slår tre slag per omgång har en genomsnittligt slutsumma som skiljer sig från tidigare studier. Att det skulle skilja sig från de utomskandinaviska studierna var något vi hade räknat med eftersom reglerna är annorlunda och poängutdelningen är högre. Larsson och Sjögrens studie behandlar dock den skandinaviska versionen av yatzy och är den enda vi har funnit som analyserar den optimala strategin. Här skiljer sig vårt resultat från deras med cirka 11 poäng, det vill säga drygt fyra procent. Jämförs däremot graferna kan tydliga likheter i fördelningen ses. (Se [appendix e] och Graf 1).

Vi anser precis som Larsson och Sjögren att den optimala strategin har implementerats, men uppenbarligen har våra implementationer skillnader som gör att resultatet också skiljer sig. För vår frågeställning anser vi dock att eftersom de två fall vi undersöker är implementerade med samma algoritm kan en jämförande studie utföras.

Som förväntat är det fördelaktigt att få möjligheten att slå fyra slag istället för tre, vilket både kan uttydas från Graf 1 och ur data från Tabell 1. Enkom det faktum att väntevärdet för fyra slag är högre än det för tre slag är tecken på fördelarna. Det faktum att variansen är lägre för reglerna med fyra slag tyder på att sannolikheten att komma nära det förväntade värdet är högre, än det är för reglerna med tre slag.

Som tydligt ses så skiljer sig kurvorna för tre respektive fyra slag sig åt, inte enbart med avseende på poäng utan även på form och fördelning. Ett enligt oss tydlig korrelation mellan graferna är de förhållandevis lågt förekommande värdena för poängsummor i intervallet 260-290. Detta tyder på att dessa poängsummor är mer osannolika att få, oavsett om tre eller fyra kast tillåts.

Det faktum att majoriteten av poängsummorna för fyra slag ligger närmare maximal poängsumma (375) kan tänkas vara självklart, och detta påvisas i vår undersökning. Utifrån detta kan spekulationer göras om huruvida allt fler slag leder till en allt tydligare fördelning av poängsummor nära den maximala summan, vilket är ekvivalent med att väntevärdet höjs. Dessa spekulationer är dock inget vi genom detta arbete kunnat utröna.

Slutsats

Korrektheten för den optimala strategin är fullkomliga, men under detta arbete har möjligheten att kontrollera om denna faktiskt uppnåtts inte kunnat undersökas. Trots avsakande av fullständiga bevis för total korrekthet anses strategin vara ett fullgott verktyg för att jämföra de två regelförutsättningarna. Den implementerade strategin påvisar dock tydligt att variansen minskar vilket innebär att sannolikheten för högre poängsumma ökar när fyra slag tillåts till skillnad från tre slag. Variansen för möjligheten att slå fyra slag istället för tre minskade, vilket tyder på ökad sannolikhet att resultera i en poängsumma nära väntevärdet.

Sammanfattningsvis blev resultatet som förväntat, att då i yatzy få möjligheten att slå fyra slag per omgång istället för tre, är fördelaktigt i avseende för att få höga poäng och minimera risken för låga.

Citerade arbeten

[1] Hasbro Inc., ”Fascinating facts about the invention of the Yahtzee board game”,[Webbsida], 2006,20130412 http://www.ideafinder.com/history/inventions/yahtzee.htm

[2] J. Glenn, ”An optimal strategy for yahtzee,” Loyala college, Computer Science Department,[Online],20060510, 20130412 http://www.cs.loyola.edu/~jglenn/research/optimal_yahtzee.pdf.

[3] J. Glenn, ”Computer Strategies for Solitaire Yahtzee,” [Online], 2007,20130412, http://cswww.essex.ac.uk/cig/2007/papers/2046.pdf

[4] M. Larsson och A. Sjöberg, ”Optimal yatzy strategy”,[Online], 2012,20130412, http://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand12/Group89Michael/report/Larsson+Sjoberg.pdf

[5] M. Fleddin och V. Sood, ”Optimal Yahtzee Strategies and Heuristics,”,[Online], 2012,20130412, http://www.csc.kth.se/utbildning/kth/kurser/DD143X/dkand12 /Group5Mikael/final/Markus_Felldin_and_Vinit_Sood.pdf

[6] J. Pawlewicz, ”A Nearly Optimal Computer Player in Multi-player Yahtzee,” [Online], 2010,20130412, http://www.mimuw.edu.pl/~pan/papers/yahtzee.pdf

[7] T. Verhoeff, ”Optimal Solitaire Yahtzee Strategies,”,[Online], 2004,20130412, www.win.tue.nl/~wstomv/talks/game-analysis/Yahtzee-talk-NWD.pdf

[8] B. Ringnér, ”Väntevärden,”,[Online], 2006,20130412, http://www.maths.lth.se/matstat/kurser/fms033/fms033_06/expectation.pdf

[9] Milton Bradley Company, ”Yahtzee,” Milton Bradley Company,— [Webbsida], http://www.hasbro.com/common/instruct/Yahtzee.pdf. [Använd 11 04 2013].

[10] I. Peterson, ”Ivars Peterson's MathTrek, Mathematical association of America,” [Webbsida] 20030317,20130412. http://www.maa.org/mathland/mathtrek_03_17_03.html. [Använd 12 04 2013].

[11] Alga, ”Superyatzy,”,[Webbsida], 2009,20130412, http://www.algaspel.se/~/media/Alga/Files/Rules/flersprakiga/38018949_QUBE_Superyatzy.ashx

[12] L.-C. Böiers, Lunds Tekniska Högskola, [Webbsida], 2001, 20130412, www.maths.lth.se/matematiklth/valfria/diskret/huvudfil.pdf

[13] G. Rundqvist, ”INTRODUKTION TILL MARKOVKEDJOR,” [Online], 20130412, www.math.kth.se/~goranr/goran2.pdf

[14] A. Sjögren och M. Larsson, ”Optimal-yatzy-Github,[Online],

2013,20130412, https://github.com/ansjob/optimalt-yatzy.git

Appendix

a - specifikation av CCI

Teknisk specifikation av “Cluster Compute Instances”:

- Internminne: 60.5 GB
- Processor: 88 EC2 Compute Units (2 x Intel Xeon E5-2670,8 kärnor)
- Lagringsutrymme: 3370 GB
- Plattform: Ubuntu 12.10 - 64-bit
- Skrivning/läsning: mycket snabb (10 Gigabit Ethernet)
- AP-namn: cc2.8xlarge

b - specifikation av salsdatorer på KTH

Teknisk specifikation av använda datorer på KTH:

- Processor: Intel(R) Core(TM) i7-2600, 3.4 GHz (8 kärnor)
- Interminne: 16.0 GB
- Plattform: Windows 7 Enterprise - 64-bit
- Grafikkort: NVIDIA GeForce 8500 GT

c – upphovsrättlig fil

Copyright (C) 2012, Marcus Larsson, Andreas Sjöberg

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

d - spelregler

Den skandinaviska versionen av yatzy spelas med fem tärningar av en eller flera personer. Poängblocket som skall fyllas i består av 15 kategorier vilka representerar olika konstellationer av tärningsresultatet efter det sista av de tre slagen. Ett helt spel består av 15 varv där varje spelare har en tur per varv.

Varje tur inleds med att spelaren kastar de fem tärningarna. Mellan varje av de två följande kasten kan spelaren välja att kasta om alla tärningarna eller behålla en eller flera av dem. Efter det tredje och sista kastet skall spelaren placera utkomsten av tärningarna i någon av de 15 kategorier som representerar spelarens poäng.

Varje kategori har olika kriterier som måste uppfyllas för att få sätta poäng där.

Kategorierna är följande:

- "Ettor"-"Sexor": sex rader vilka var och en endast kan fyllas av summan av de tärningar som visar EN prick, respektive TVÅ prickar och så vidare.
- "Par": fylls med summan av två tärningar som visar samma antal prickar.
- "Tvåpar": fylls med summan av två unika par, till exempel {1,1} och {4,4} men inte {1,1} och {1,1}.
- "Tretal": fylls med summan av tre tärningar vilkas antal prickar är lika.
- "Fyrtal": fylls med summan av fyra tärningar vilkas antal prickar är lika.
- "Kåk": fylls med summan av tre lika tärningar och ett par av annat antal prickar.
- "Liten stege": kan endast fyllas med 15 poäng, då tärningarna visar {1,2,3,4,5}.
- "Stor stege": kan endast fyllas med 20 poäng, då tärningarna visar {2,3,4,5,6}.
- "Chans": summan av alla tärningarna.
- "Yatzy": kan endast fyllas med 50 poäng då alla tärningar visar samma antal prickar.

En bonus om 50 poäng ges till en spelare om summan av de sex översta raderna överstiger 62. (En tumregel är att gränsen precis överstigs då summorna utgörs av tre tärningar i respektive rad).

I det fall tärningarna inte uppfyller kraven för att skriva in poäng är spelaren tvungen att "stryka" en rad, vilket innebär att denne sätter noll poäng på någon rad, vilket gör den förbrukad. Detta alternativ får även väljas trots att spelaren har möjligheten att sätta poäng på någon rad.

Eftersom en rad blir "förbrukad" när poäng sätts i kategorin är en viktig del i strategin att alltid sätta utkomsthanden av tärningar rätt. För att efter det tredje slaget få ha en så bra hand som möjligt är det viktigt att alltid spara de tärningar som medför att det förväntade slutgiltiga resultatet är så högt som möjligt. [11]

e – simuleringsgraf Larsson & Sjöberg