Optimal Yahtzee

Strategies and Heuristics

Markus Felldin and Vinit Sood

Statement of collaboration

The project can be broken down into two phases, the research and development phase and the report writing and result gathering phase. During the R&D phase we worked more or less independently with strategically placed meetings that coordinated our efforts. This was done to decrease the likelihood of “group-think” and increase the diversity of our algorithm. Near the end of this phase we discussed our findings and constructed an algorithm that was a summation of both of our findings. During the report writing and result gathering phase we worked on one section each at a time and then proof read and corrected each other’s sections. The introduction, abstract and conclusions were written together. To summarize we have both put in roughly the same amount of work and time into this project.

Abstract

The purpose of this project is to explore and employ statistically sound strategies for maximizing the average result of solitaire Yahtzee. The game was broken down into its statistical base and analyzed from a strategic point of view. Based on the findings from the analysis an efficient and strategically sound algorithm was designed and tested thoroughly. The results and major parts of the strategy used are presented. The results gathered throughout this project support the idea that Yahtzee is a *rational* game for which there are sound strategies that yield favorable results. The algorithm proved to be efficient, both in terms of complexity and results, but was not optimal.

Sammanfattning

Avsikten med projektet är att utforska och tillämpa statistiskt motiverade strategier för att maximera det högsta medelvärdet i spelet solitär Yahtzee. Spelet bryts ned till dess strategiska grunder och analyseras från ett strategisk perspektiv. Från analysen skapades en effektiv och strategisk algoritm som testades grundligt. Resultaten och även stora delar av den slutliga strategin presenteras i uppsatsen. Från resultaten och analysen som dragits ur under projektets utförande stöds idén om att Yahtzee är ett rationellt spel som kan spelas med statistiskt motiverade strategier för att uppnå ett bra resultat. Den slutliga algoritmenen visade sig vara effektiv men inte optimal.

Table of Contents

Introduction

The information age has brought forth many changes in our society; previously incalculable problems have all of a sudden become solvable with readily available pieces of equipment. One such problem is Yahtzee. In contrast to games like chess and blackjack it is a light-hearted game that seldom puts money or prestige on the line and has, much because of this, eluded the attention of many in the academic world. Despite this Yahtzee is a perfect candidate for analysis as it is far too complicated to perform the necessary calculations without access to a computer, but is on the other hand much less complicated than games like chess. Beneath the surface Yahtzee is a game of skill and chance where good strategies rely on making sound choices grounded in probability.

Since Yahtzee is a rational game, good choices are based on the data at hand, common sense dictates that there should exist an optimal strategy, or at the very least strategies that on average perform well.

Background

Yahtzee is a popular turn-based dice game that is currently marketed by Hasbro. It was originally invented by a wealthy Canadian couple, in 1956, to play aboard their yacht, hence giving rise to the name “Yacht game” and later on Yahtzee[1]. The game consists of five six-sided dice and a scorecard with 13 boxes, each with its own set of scoring rules. The game can be played solitaire or in groups of two or more players. For solitaire Yahtzee the object of the game is to obtain the highest possible score, i.e. utilizing an optimal strategy. When playing with two or more players the objective is simply to beat the opponent’s score.

According to results posted on Yahtzee UK’s homepage[2] a good player seldom reports scores lower than 100 points with the majority of scores being over 200 points.

There has been some research done on the subject previously and Tom Verhoeff[3], a professor at Eindhoven University of Technology, has created what is believed to be an optimal algorithm for playing solitaire Yahtzee.

Aim and Objectives

The aim of this project is to construct an algorithm making use of efficient heuristics for playing solitaire Yahtzee. The algorithm should make sound choices based on statistics and strategy. The algorithm should also require minimal computational power as this project will be conducted on standard desktop computers.

Delimitation of Study

Due to the limited time frame this project will be limited to finding an efficient heuristic rather than providing an optimal solution. The project will also be limited to solitaire Yahtzee as the additional time needed to implement a multiplayer algorithm would most certainly decrease the quality of our solitaire algorithm which is indeed the main focus of this project.

---

[1](Hasbro 2012, "The History of YAHTZEE.")
[2](The Yahtzee Page 2012,"Yahtzee Rules, Probability, Statistics and More.")
[3](Verhoeff 2012, "Optimal Solitaire Yahtzee Player: Trivia.")

Breaking Down the Game

A game of Yahtzee consists of thirteen rounds. At the start of each round the player rolls all five dice. The player is then given the opportunity to reroll any of these dice. After this the player may again opt to reroll as many dice as desired. Once the second reroll has been completed, or at any point after the first roll, the player must place the score in one of the thirteen boxes. The resulting score is determined based on the scoring conditions of the chosen box. Each box can be used only once, and you cannot switch scores if a more favorable outcome were to appear later on in the game.

The Yahtzee score card consists of thirteen different scoring options divided into two sections: the upper and lower sections. The upper section consists of six boxes, one for each of the six face values of the dice. The score in each box is the sum of all dice with the corresponding face value. An additional 35 point bonus is rewarded if the total score of the upper section is at least 63. A total upper section score of 63 points corresponds to scoring three-of-a-kind in each of the six boxes.

The lower section is a little more complex in nature and each box has its own set of poker-themed scoring rules:

- **Three of a kind –** The score is equal to the sum of all dice if and only if the set of dice includes three or more with the same face value.
- **Four of a kind –** The same as three of a kind except the set of dice must include at least four dice with the same face value.
- **Full House –** The set of dice must include both a three of a kind and a two of a kind (a pair). The score is always 25 if these conditions are met regardless of the face values. The set cannot consist of only one face value; a five of a kind is not a full house.
- **Small straight –** The set of dice must include 4 consecutive face values, for example: 2-3-4-5. The score is always 30 if these conditions are fulfilled.
- **Large Straight –** The set of dice must include 5 consecutive face values. The score is always 40 if these conditions fulfilled.
- **Chance –** The score is the sum of all faces and there are no conditions that need to be fulfilled.
- **Yahtzee –** The score is always 50 if all five dice show the same face.

If the scoring conditions are not met for a box in the lower section, the resulting score is always zero. It is also worth noting that there exists certain variations to these rules, the most common of which being the addition of “Yahtzee bonus”. The Yahtzee bonus entails that after the initial Yahtzee has been scored, an additional bonus of one hundred points is scored for each consecutive Yahtzee. This project will not include the rule set for Yahtzee bonuses.

The Strategy

In order to find a good strategy for Yahtzee we must first define the game in a scope that allows for computable analysis. For the sake of this project we will envision the game as a state graph where each vertex represents a state or outcome of the dice. For each state there are zero or more ways to transition from this state to another, these transitions can be thought of as weighted edges. The weights will be the expected value [Equation 1] for each possible transition given the current state.

Every game of Yahtzee starts off with an initial roll of the dice from which there is 6^5 state transitions. After the initial roll the player is allowed to select zero to five dice to keep, increasing the size of the state graph to $6^5 2^5$ vertices. The player is then allowed to reroll the selected dice, increasing the amount of vertices to $6^5 2^5 3.5^5$ (=$6^5 7^5$). Finally the player is allowed to keep and then reroll any combination of dice again resulting in a total of $6^5 7^5 7^5$ vertices per turn. Since there are 13 turns in a game of Yahtzee the total amount of vertices in a complete Yahtzee game tree is $(6^5 7^5 7^5) * 13!$. However since the dice are not sensitive to order the size of the graph can be reduced.

Unfortunately the scope of this project does not permit a complete analysis of the state graph; instead the focus will lie on optimizing the transitions for one turn at a time. For each turn there will be an initial state, achieved by rolling the dice at the beginning of the turn. Given this initial state and the scoring rules of the remaining boxes the expected value for each of these can be calculated.

Equation 1: Def. Expected value: $\sum_{i=1}^{\infty} x_i p_i$ where x is the value of a given outcome and p is the probability of reaching this outcome from the current state.

It would seem like calculating all of these expected values would be a rather laborious task given the sheer amount of states. However, given the specific scoring rules of each box; many of these states will have no value for the chosen box, hence the expected value is simply the sum of all non-zero scoring reachable states multiplied by the probability of reaching them. Additionally the outcomes are independent of order, thus further reducing the scope of outcomes.

A turn in Yahtzee is comprised of two reroll phases, i.e. the player can at two separate occasions per turn decide which dice to keep in order to reach a desired outcome. This means that during the first reroll phase the player must take into consideration that there are two opportunities to reach an outcome, meaning an increased probability of reaching desired outcomes. This must be taken into consideration when the expected values are calculated.

Equation 2: Def. binomial coefficient: The amount of ways to pick m things out of n

$$C_{n,m} = \frac{n!}{m!\,(n-m)!}$$

Equation 3: Def. probability of going from n beneficial dice to m beneficial dice in one roll:

$$p(n,m) = C_{n,m} p^m (1-p)^{n-m}$$

Equation 4: Def. probability of going from i beneficial dice to j beneficial dice in two rolls is the sum of all ways of going from n to m in two tries:

$$p^2(i,j) = \sum_{k=1}^{5} p(i,k)p(k,j)$$

The equations above account for the probabilities of going from one state to another given one or two remaining rerolls, and will be employed in the algorithm in the following section.

Breaking down the game into smaller graphs, one for each turn, allows for more complete analysis of that specific part of the game. Conversely it makes it hard to account for the soundness of choices over more than one turn. This is where clever heuristics come into play, how do we get the algorithm to recognize the worth of a choice over several turns without actually computing it?

The Algorithm

Designing a successful solitaire Yahtzee algorithm means that the algorithm should make strategically sound choices given any outcome at any point in the game. In order to make a choice we must first establish a set of values to choose from. As described in the section above this can be done by calculating the expected value for each of the thirteen boxes for each roll. The starting point of each turn is the outcome of the initial dice roll. At this point in the game the player has an outcome accompanied by two optional rerolls and a set of empty score boxes to base their actions on. The expected values for each of the remaining boxes can be calculated using equation 4 with the addition of another multiplicative factor indicating the value of the postulated outcome:

$\sum_{k=1}^{n} p^2(i,j)x_k$ *Where x_k stands for the value of the postulated outcome, i is the current amount of beneficial dice and j is the amount of beneficial dice we wish to achieve after the two rerolls.*

In the actual algorithm mutual benefit is also factored into the expected values for each outcome. A perfect example of this is that when a player decides to try to achieve a good score in sixes they are also selecting dice in a way which may result in beneficial outcomes for three-of-a-kind, four-of-a-kind, Yahtzee and chance. Additionally the boxes in the upper section are affected by their impact on the upper section bonus. Consider for example that the player rolls two ones and three sixes in the beginning of the turn, the natural inclination would be to keep the sixes and reroll the ones. If however the player needs at least three ones in order to secure the upper section bonus, the value of three ones would be 38, since the bonus cannot be achieved without them.

Using the data obtained from the calculations mentioned above the algorithm chooses the box with the most favorable predicted outcome and rerolls the dice from which the box does not benefit. The reroll yields a new outcome (or state) from which the algorithm must yet again decide how to proceed. The method is similar to the undertakings performed after the initial outcome was reached with the difference being that all rerolls will have been exhausted after this phase. The expected outcomes are calculated in the same way as above, with the sole distinction that equation 4 is replaced by equation 3:

$\sum_{k=1}^{n} p(n,m)x_k$ *Where x_k stands for the value of that particular outcome*

Yet again this data is used as grounds for the decision making process. The box with the most favorable predicted outcome is selected and all unbeneficial dice are rerolled. At this point all rerolls have been exhausted and the resulting outcome must be placed in one of the remaining boxes. The values for each remaining box are calculated and the result is placed where it is deemed most favorable. The degree of favorability for a given result is determined by comparing it to the range of possible results for that particular box.

The choice processes performed during each phase of a turn is illustrated in Figure 1.

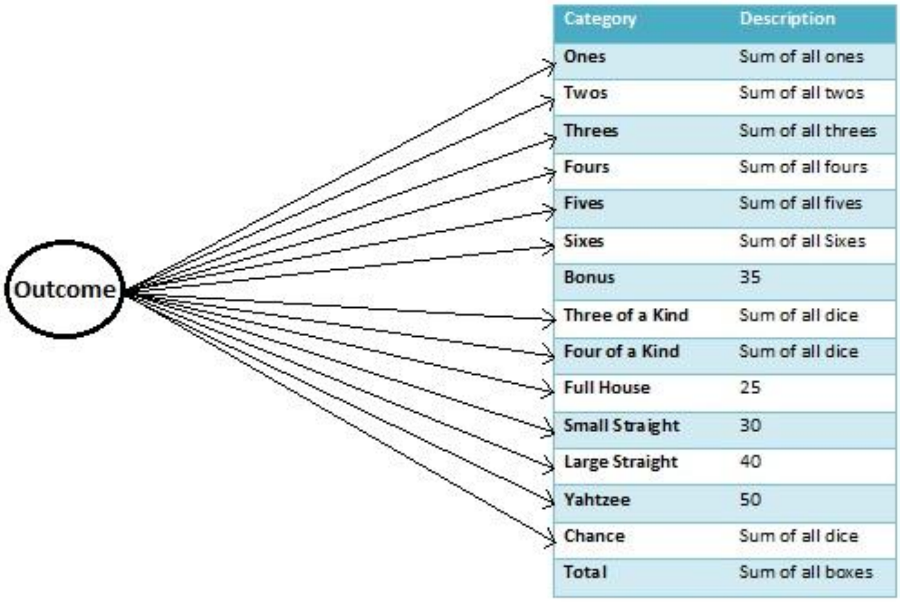

Figure 1 – This simple diagram illustrates how an outcome is related to each of the boxes of the scorecard. The arrows have a weight corresponding to the expected value of that particular choice.

The procedure described above is repeated for each of the thirteen turns that encompass the game of Yahtzee. As the game progresses the weights of certain choices differ as boxes they share outcomes with are filled, this means that the favorability of choices varies along with the progression of the game.

After all thirteen turns have been completed we are left with exactly one score per box (this score can indeed be zero if the conditions required to score it were never met) and a total score which is the sum of all boxes.

Program and Lab Environment

The algorithm was coded in Python version 2.7.2 and all trial runs were run through idle. Idle is pythons standard IDE[4]. The random module of Python was used to generate the dice values[5]. All times were recorded on a desktop computer running a 64bit Windows 7 Professional Operating System with a 4,78GHz Intel i7 2600K CPU[6] and 16 GB of RAM memory.

[4] IDE – Stands for Integrated Development Environment and is in this case a compiler, and source code editor.
[5] Python uses an algorithm called Mersenne Twister to generate pseudo-random numbers.
[6] CPU – Stands for Central Processing Unit and is the main source of a computers data processing power.

Results

Throughout the project the algorithm has been run frequently in various stages of completion in order to check progression. Empirically we found that the average results of less than one thousand runs vary wildly[7], and for fine tuning no less than one hundred thousand results were required to yield credible results. Thus we subsequently decided to perform one million runs for our results acquisition phase. Comparing 10 independent results, at one million runs a piece, the maximum difference between any two results was 0.03 points over the average total scores. The complete results for one million runs took a total of 4hours and 22 minutes to obtain, running the program on one thread.

The results show that the algorithm has an average total score of 221.67906 [Table 1] and that nearly 70% of all reported scores are over 200 points [Figure 2].

Category	Description	Average Score
Ones	Sum of all ones	1.277551
Twos	Sum of all twos	4.00933
Threes	Sum of all threes	7.742268
Fours	Sum of all fours	11.17878
Fives	Sum of all fives	14.24434
Sixes	Sum of all Sixes	17.762082
Bonus	35	10.159975
Three of a Kind	Sum of all dice	22.59869
Four of a Kind	Sum of all dice	16.636411
Full House	25	16.230025
Small Straight	30	29.06337
Large Straight	40	31.71348
Yahtzee	50	17.0288
Chance	Sum of all dice	22.033958
Total	Sum of all boxes	221.67906

Table 1 – The average score per box over 1 million runs

The results in Table 1 show that the relative scores for the upper section increase in correlation to the face value associated with the box. Furthermore the average score for the upper section Bonus is a mere 10.16 indicating that the bonus is only achieved in less than one out of every three games. A Yahtzee was scored a little over once in every three games.

Also evident from Table 1 is that the two largest contributors to the average score are the Large and Small Straights.

[7] At most +/- 3 points over one thousand runs. This value was determined based on data obtained over 10 runs.

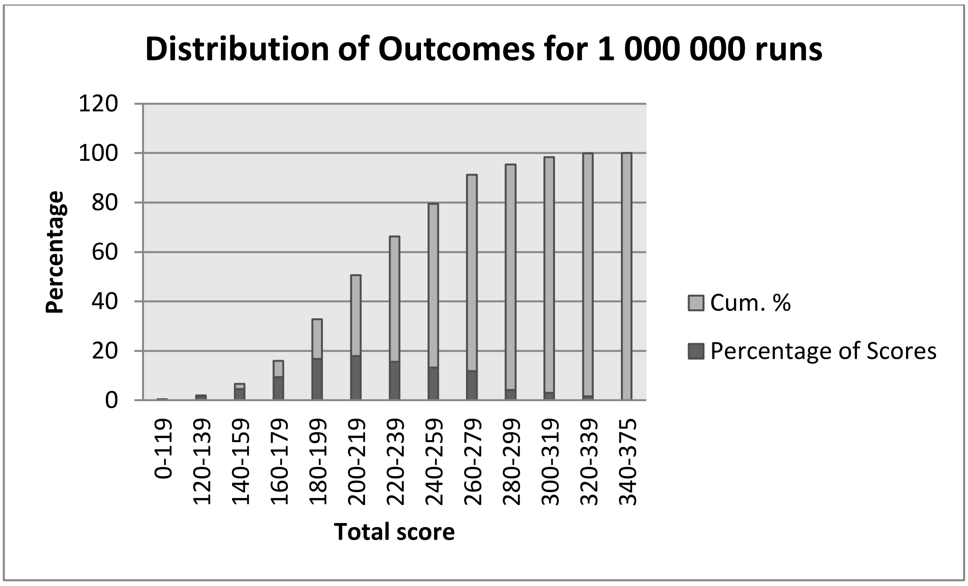

Figure 2 – The percentage per 20 point interval and the cumulative percentages for 1 000 000 runs of our algorithm

As is made evident by Figure 2 the distribution of scores closely resembles a bell curve, something which is to be expected when dealing with probabilistic distributions over a number of samples. The algorithm very seldom scores beneath 119 points, this portion of the results account for a mere 0.36% of the total outcomes. Also evident is that nearly 80% of all scores fall in the range 180-279.

Additional noteworthy results that are not shown in Table 1 or Figure 2 are the minimum, maximum and median scores achieved over the test in question. The results were the following:

Minimum	Median	Maximum
75	219	353

Table 2 – The minimum, median, and maximum total scores over 1 million runs

There are additional results, including comparisons to other similar algorithms and results from previous runs included in Appendix B. The results brought forth here are the ones considered most relevant, but the Appendix can be consulted for additional insight.

Discussion

The intent of this project was to construct an efficient algorithm that plays solitaire Yahtzee well. The term well, in this instance, means that the algorithm should perform better than playing the game at random and that the results should approach those of an optimal algorithm. This was, as shown in the results, achieved. The algorithm is quick to compute, taking only 4 hours and 22 min to perform 1 million runs. This is approximately 15 milliseconds per run, a respectable result considering the program was coded in Python, which is known to be a relatively slow programming language. Furthermore the total scores were distributed in a bell curve shape with a median score of 219, all evidence of the validity of the heuristics employed. A median score of 219, in combination with the bell curve distribution, is very similar to, if not better than, that of an expert human player[8].

Most of the results were in line with what is to be expected of an algorithm of this nature. The low average score of the upper section bonus may however be a little surprising. By studying the behavior of the algorithm over several runs it could be determined that the low average score of the upper section bonus was simply due to the algorithm consistently posting low scores in the boxes for ones and twos before their respective bonus contributions had the chance to be taken into account. This is of course one of the major problems of basing the decision making process on one turn at a time. This problem, if you will, could be countered in several ways; the most natural might be to consider a larger portion of the game in the decision making process or by placing more weight on scoring well in ones and twos. The latter proved rather inefficient and resulted in lower average scores due to a greater loss of points in the categories three and four of-a-kind. The former was deemed too time consuming to implement during the scope of this project, but definitely seems like a rational solution that could be employed to improve the algorithm.

The large difference between maximum and minimum scores may seem alarming at a first glance; however, Yahtzee is a very complex game with a large amount of possible outcomes. These extremes represented very few outcomes and can be explained by the probability at work.

It may be worth noting that Python's own random generator was used to generate the dice rolls throughout this project. Although this algorithm provides an equal distribution across its range over time it is pseudo random; meaning that its results are completely deterministic. This is a possible source of error, but because of the nature of this experiment it should have none to minimal impact on the results presented in this report. Python's random number generator shows a very similar statistical distribution to that of a true random number generator. Furthermore the deterministic quality of the generator is not exploited in the algorithm, insuring minimal impact on the results of the report.

Compared to Other Algorithms

There are a couple of algorithms for playing solitaire Yahtzee, the most noteworthy of which being the one designed by Tom Verhoeff. This algorithm is thought to be optimal, i.e. it plays solitaire Yahtzee perfectly for any set of outcomes. The algorithm designed by this project runs considerably

[8] As mentioned in the Background a good (or expert) player will score over 200 points across a majority (>50%) of their games.

faster than Verhoeff's optimal solution at the price of scoring a lower average score. It is also important to note that Verhoeff's algorithm is designed for a different version of the game; one which allows multiple Yahtzees, which increases the possible maximum score.

Verhoeff also has a webpage[9] where it is possible to test yourself or your algorithm against his. Although designed for two different versions of the game; the games are similar enough to be accurately compared, as long as those differences are taken into account. The algorithm employed in this project was tested against this webpage, albeit on a scale far too small to be of statistical significance, and made very similar recommendations. An extract of Verhoeff's results is provided in Appendix A.

[9](Verhoeff 2012, "Optimal Solitaire Yahtzee Player: Trivia.")

Conclusion

The algorithm designed for this project successfully employed the probability based strategies discussed throughout this report, and did in the end show results indicating that Yahtzee is indeed a rational[10] game. The algorithm did not perform optimally[11], but did achieve comparable results by making use of heuristics. It might however prove to be better at playing solitaire Yahtzee than human players, but as there was insufficient data on the subject a comparison could not be made. Despite the fact that the algorithm was not optimal, an optimal algorithm was found through a survey of the field, thus cementing the view that there is always an optimal move to be made at any given state. An interesting step forward would be to employ smarter heuristics in an attempt to secure the upper section bonus more often without damaging the averages of other boxes.

[10] The term rational in this case means that correct choices can be made based on accurate mathematical calculations.

[11] The algorithm did not make perfect choices for all possible outcomes, and can as such not be deemed as optimal. Perfect choices are choices that take every eventuality into account and based on correct mathematics chooses the action most favorable for the desired result.

References

Articles

• Thomason, Richmond, "Logic and Artificial Intelligence", *The Stanford Encyclopedia of Philosophy (Spring 2009 Edition)*, Edward N. Zalta (ed.), <http://plato.stanford.edu/archives/spr2009/entries/logic-ai/>.

• McCarthy, John. *WHAT IS ARTIFICIAL INTELLIGENCE?* Tech. CSD Stanford University, 12 Nov. 2007. Web. 12 Feb. 2012. <http://www.formal.stanford.edu/jmc/whatisai/whatisai.html>.

Literature

• Kleinberg, Jon, and Éva Tardos. *Algorithm Design*. Boston Mass.: Pearson Addison Wesley, 2006. Print.

• Blom, Gunnar. *Sannolikhetsteori Och Statistikteori Med Tillämpningar*. Lund: Studentlitteratur, 2005. Print.

Websites

• Verhoeff, Tom. "Optimal Solitaire Yahtzee Player: Trivia." *Technische Universiteit Eindhoven*. TUE, 1999. Web. 12 Feb. 2012. <http://wwwset. win.tue.nl/~wstomv/misc/yahtzee/trivia.html>.

• Hasbro. "The History of YAHTZEE." *Hasbro.com*. Web. 25 Mar. 2012. <http://web.archive.org/web/20070808043322/http://www.hasbro.com/default.cfm?page=ci_history_yahtzee>.

• "Yahtzee Rules, Probability, Statistics and More." *The Yahtzee Page*. Web. 30 Mar. 2012. <http://www.yahtzee.org.uk/index.html>.

Optimal Yahtzee

Appendix A

Markus Felldin and Vinit Sood

Rules of Yahtzee

Objective

In the game of Yahtzee the player has to score as much points as possible by rolling five die and achieving certain combinations. The game consists of thirteen rounds and for each round the player has three opportunities to roll the dice; the first time he or she *must* roll all dice. During the second and third roll the player can choose which die to reroll in order to achieve a desired combination. After the third roll, one combination on the scoreboard *must* be filled with yielded points of the round and in the worst case scenario the player may need to score zero points for a combination. After thirteen rounds the player with the most points wins. Yahtzee can be played alone with other players.

Scoreboard

The combinations that can be achieved are presented on a scoreboard that is divided in two sections: the **upper section** and the **lower section**.

Upper Section

The upper section consists of 6 combinations, where each combination is named after a face on the die. Each combination is scored by adding the value of all of dice faces for the matching combination. If a player for example would roll the combination of {3,3,3,4,5} he or she could score a total of 9 points in the “threes” combination. A bonus score of 35 points is awarded the total amount of points in the upper section exceeds 63.

Lower Section

The lower section consists of poker-inspired combinations that for many yield a fixed amount of points.

Combination	Description	Score
Three of A Kind	A minimum of three dice are showing the same face	Sum of all the dice
Four of A Kind	A minimum of four dice are showing the same face	Sum of all the dice
Full House	A three of a kind and a pair of different face	25
Small Straight	Four sequential dice	30
Large Straight	Five sequential dice	40
Yahtzee	All the dice show the same face	50
Chance	Any combination	Sum of all the dice

Scoring variations

In the Finnish, Danish, Norwegian and Swedish versions of the game, the scoreboard is slightly different:

- The bonus for achieving 63 points in the upper section is 50.
- The lower section also consists of the combinations *pair* and *two pair.*
- Points for *Three of a kind* and *Four of a Kind* are determined by the sum of all the beneficial dice.
- A Small Straight is strictly defined as 1-2-3-4-5 and awards 15 points.
- A Large straight is strictly defined as 2-3-4-5-6 and awards 20 points.
- The score for a full house is the sum of all the dice.

Combinatronics

A fair dice has six different faces, and in the game of Yahtzee the player rolls 5 die. There are thusly 7776 patterns of faces for five dice. The following table shows the distribution of face patterns that can occur. A Yahtzee can for example occur in 6 different ways.

A	A	A	A	A	6/7776
A	A	A	A	B	150/7776
A	A	A	B	B	300/7776
A	A	A	B	C	1200/7776
A	A	B	B	C	1800/7776
A	A	B	C	D	3600/7776
A	B	C	D	E	720/7776

Table 1: A table showing the amount of combinations for every face pattern for five dice.

Probabilities

Equations

Binomial coefficient	$C_{n,m} = \frac{n!}{m!\,(n-m)!}$	Equation *1.1*
PMF for binomial distribution	$C_{n,m}p^m(1-p)^{n-m}$	Equation *1.2*
Applied binomial distribution for 5 dice and 1 throw	$C_{5-i,k}\left(\frac{1}{6}\right)^k\left(\frac{5}{6}\right)^{5-(i+k)}$	Equation *1.3*
Applied binomial distribution for 5 dice and 2 throws	$p^2(i,j) = \sum_{k=1}^{5} p(i,k)p(k,j)$	Equation *1.4*

Tables

Formula 1.3 gives the probability of going from *i* dice with the same face to *k* dice with the same faces in one single roll.

$$C_{5-i,k}\left(\frac{1}{6}\right)^k\left(\frac{5}{6}\right)^{5-(i+k)}$$

From(i) To (k)	1	2	3	4	5
1	0.482	0.386	0.116	0.0154	0.000772
2	0	0.579	0.347	0.0694	0.00463
3	0	0	0.694	0.278	0.0278
4	0	0	0	0.833	0.167
5	0	0	0	0	1.0

Table 2: The probabilities of going from *i* dice with the same face to *k* in one roll.

Formula 1.4 gives the probability of going from *i* dice with the same face to *k* dice with the same face but with two available rolls.

$$p^2(i,j) = \sum_{k=1}^{5} p(i,k)p(k,j)$$

From(i) To(j)	1	2	3	4	5
1	0.233	0.409	0.270	0.0792	0.00872
2	0	0.335	0.442	0.195	0.0285
3	0	0	0.482	0.424	0.0934
4	0	0	0	0.694	0.306
5	0	0	0	0	1.0

Table 3: The probabilities of going from *i* dice with the same face to *k* in two rolls.

Python Code

- ***Equation 1.1***

```
def select(n,m):
    C = math.factorial(n)/(math.factorial(m)*math.factorial(n-m))
    return C
```

- ***Equation 1.3***

```
def prob1(fro,to):
''' Calculates the chance of going from one amount of a number to
another in one dice roll'''

    n = NUM_DICE-fro
    m = to-fro
    C = select(n,m)

    prob = C*math.pow(1.0/6.0,(m))*math.pow(5.0/6.0,(n-m))
    return prob
```

- ***Equation 1.4***

```
def prob2(fro,to):
'''Calculates the chance of going from one amount of a number to
another in two dice rolls'''
    prob = 0
    for i in range(fro,to+1):
        prob += prob1(fro,i)*prob1(i,to)
    return prob
```

Results from Tom Verhoeffs Optimal Yahtzee Player

Category	Average Score
Aces	1.88
Twos	5.28
Threes	8.57
Fours	12.16
Fives	15.69
Sixes	19.19
Upper Section Bonus	23.84
Three of a Kind	21.66
Four of a Kind	13.10
Full House	22.59
Small Straight	29.46
Large Straight	32.71
Yahtzee	16.87
Chance	22.01
Extra Yahtzee Bonus	9.58
Grand Total	254.59

Table 3: The average scores for each category from Tom Verhoeff's own Optimal Solitaire Yahtzee Player

References

Literature

• Blom, Gunnar. Sannolikhetsteori Och Statistikteori Med Tillämpningar. Lund: Studentlitteratur, 2005. Print.

Websites

• Verhoeff, Tom. "Optimal Solitaire Yahtzee Player: Trivia." Technische Universiteit Eindhoven. TUE, 1999. Web. 12 Feb. 2012. <http://wwwset. win.tue.nl/~wstomv/misc/yahtzee/trivia.html>.

• Berry, Nick. "Yahtzee Probability." Web log post. DataGenetics. 4 Jan. 2012. Web. 3 Apr. 2012. <http://www.datagenetics.com/blog/january42012/index.html>.

• Hasbro. "Yahtzee Instructions." Hasbro.com. Web. 15 Mar. 2012. <http://www.hasbro.com/common/instruct/Yahtzee.pdf>.

Optimal Yahtzee

Appendix B

Markus Felldin and Vinit Sood

Average results from all runs

The following tables display the progression of the results as the algorithm was modified and improved over time. The tables show the average result for every category for a specified number of games. Bellow every table, excluding Table 1, is also the minimum and maximum value that was obtained during the session.

Table 1: 100 000 runs 06/04-2012

Category	Average Score
Aces	1.192
Twos	4.62698
Threes	8.03166
Fours	11.72856
Fives	15.17005
Sixes	18.8772
Upper Section Bonus	14.1827
Three of a Kind	21.43568
Four of a Kind	5.09466
Small Straight	28.7664
Large Straight	28.0452
Yahtzee	15.698
Full House	14.632
Chance	22.06977
Grand Total	**209.55086**

Data for the minimum and maximum values were not taken for this session.

Table 2: 100 000 runs, first try, 07/04-2012:

Category	Average Score
Aces	1.18364
Twos	4.62188
Threes	7.80996
Fours	11.33776
Fives	14.313
Sixes	17.54634
Upper Section Bonus	10.4853
Three of a Kind	21.96368
Four of a Kind	14.64658
Small Straight	29.0274
Large Straight	29.3592
Yahtzee	15.5805
Full House	16.15575
Chance	22.2666
Grand Total	**216.29759**

Max: 351
Min: 67

Table 3: 100 000 runs, second try, 07/04-2012:

Category	Average Score
Aces	1.18551
Twos	4.61752
Threes	7.80747
Fours	11.33144
Fives	14.3159
Sixes	17.54322
Upper Section Bonus	10.3852
Three of a Kind	21.98945
Four of a Kind	14.6449
Small Straight	29.0139
Large Straight	29.5152
Yahtzee	15.6325
Full House	16.0795
Chance	22.28037
Grand Total	**216.34208**

Max: 351

Min: 76

Table 4: 10 000 runs, first try, 08/04-2012

Category	Average Score
Aces	1.4274
Twos	4.2728
Threes	8.2089
Fours	11.4224
Fives	14.509
Sixes	17.6628
Upper Section Bonus	11.445
Three of a Kind	22.2307
Four of a Kind	14.9582
Small Straight	29.064
Large Straight	29.448
Yahtzee	16.22
Full House	16.28
Chance	21.9473
Grand Total	**219.0965**

Max: 341

Min: 94

Table 5: 10 000 runs, second try, 08/04-2012

Category	Average Score
Aces	1.43044
Twos	4.29866
Threes	8.1336
Fours	11.48564

Fives	14.48965
Sixes	17.74722
Upper Section Bonus	11.59305
Three of a Kind	22.22842
Four of a Kind	14.94804
Small Straight	29.0688
Large Straight	29.7372
Yahtzee	15.96
Full House	16.42175
Chance	21.91043
Grand Total	**219.4529**

Max: 347

Min: 75

Table 6: 100 000 runs 09/04 -2012

Category	Average Score
Aces	1.27755
Twos	4.01566
Threes	7.75026
Fours	11.15588
Fives	14.2591
Sixes	17.78934
Upper Section Bonus	10.2102
Three of a Kind	22.60264
Four of a Kind	16.62778
Small Straight	29.085
Large Straight	31.8656
Yahtzee	16.983
Full House	15.93375
Chance	22.02806
Grand Total	**221.58382**

Max: 350

Min: 84

Table 7: 1 000 000 runs, first try, 10/04 -2012

Category	Average Score
Aces	1.277186
Twos	4.005758
Threes	7.74822
Fours	11.177524
Fives	14.23326
Sixes	17.765148
Upper Section Bonus	10.16288
Three of a Kind	22.600491
Four of a Kind	16.638983
Small Straight	29.06226
Large Straight	31.68448
Yahtzee	17.04175

Full House	16.19965
Chance	22.031328
Grand Total	**221.628918**

Max: 348

Min: 68

Table 8: 1 000 000 runs, second try, 10/04 -2012

Category	Average Score
Aces	1.277551
Twos	4.00933
Threes	7.742268
Fours	11.17878
Fives	14.24434
Sixes	17.762082
Upper Section Bonus	10.159975
Three of a Kind	22.59869
Four of a Kind	16.636411
Small Straight	29.06337
Large Straight	31.71348
Yahtzee	17.0288
Full House	16.230025
Chance	22.033958
Grand Total	**221.67906**

Max: 353

Min: 75