Optimal Yatzy Strategy

Kalle Sederblad
kallese@kth.se

Johan Törnebohm
torneb@kth.se

Abstract

Yatzy is a famous dice game that is played with a scorecard and five dice. By rolling the dice the player's goal is to achieve a combination of dice and place the score in one of 15 categories. At the end of the game when all categories are filled the points for every category are summarized. We have developed an algorithm that makes up the optimal strategy of the game by calculating the expected value for each state of the game, recursing backwards from the last roll of the game to the first. The implementation was unsuccessful, as the algorithm was too inefficient to run in an acceptable amount of time. Instead, a part of the algorithm was run, starting the game with 10 randomly chosen categories filled with 0 points and collecting the average scores for the remaining categories using 1 million simulations.
The result was 167.60 points, which sets a lower boundary to the expected value for the complete game.

Sammanfattning

Yatzy är ett känt tärningsssspel som spelas med ett poängblock och fem tärningar. Genom att rulla tärningarna önskar spelaren uppnå en tärningskombination och placera poängen i någon av 15 kategorier. I slutet av spelet när alla kategorier är fyllda summeras alla kategorier ihop. Vi har utvecklat en algoritm som utgör spelets optimala strategi genom att beräkna väntevärdet för varje tillstånd i spelet och rekursera från spelets sista kast till det första. Implementationen misslyckades, då algoritmen var för ineffektiv för att köras inom rimlig tid. Istället kördes en del av algoritmen, där spelet börjar med 10 slumpvis valda kategorier fyllda med 0 poäng och snittpoängen för de resterande kategorierna samlades genom 1 miljon simulationer.
Resultatet var 167.60 poäng, vilket fungerar som undre gräns för spelets totala väntevärde.

Contents

1 Introduction

Yatzy is a game which is very popular in Scandinavia, with similar games being played all over the world. What is probably most encapsulating about the game for many people is that it is a good mix of luck and strategy. There being an optimal strategy to the game is not obvious, but given its straightforward form of play it can be shown that every choice in the game can optimized.

Problem statement

Our objective was to construct an algorithm able to decide which moves will generate the highest expected score in the Scandinavian version of the game Yatzy. The results would be measured in the average number of points obtained in a game, calculated by running multiple simulations of our algorithm a sufficient number of times. The focus was solely on the solitary version of Yatzy, as taking an opponent into account would require a completely different approach.

2 Background

History

Yahtzee, previously known as “the yacht game”, was developed by a wealthy couple aboard a yacht in Canada during the 1950s. The couple used to play the game whenever their friends were visiting. Their friends started to like the game so much that they wanted their own copies of the game. In order to make some copies of the game the couple approached the game entrepreneur and toy maker Edwin S. Lowe. Lowe became a fan as well and he later purchased the rights to the game in 1956. The couple did not want any money in return and instead contented themselves with the first couple of games that were produced to give to their friends. Lowe changed the name to “Yahtzee” which still remains as the game’s name. The game has since that day entertained millions of people around the world. The game now exists on many different platforms and in many different versions [1].

The number of players is unlimited and the player that reaches the highest score is the winner. Yahtzee is a turn-based game and during one turn the player has three possibilities to roll the dice; before each time, he decides which dice he wants to re-roll and and which ones to keep. The goal is to achieve one of the combinations that is written on the piece of paper. These are divided in an upper and lower section. In the upper section you try to gather as many of one kind as possible. In the lower section the categories consists of specific combinations of dice. In the end, the score from all the different criteria are summarized as the player’s total score.

Yatzy is a variant of Yahtzee which is especially popular in the Scandinavian countries. There are only a few differences in rules, regarding how points are calculated and some differences in categories. The major difference in calculating points regards three-of-a-kind, four-of-a-kind and full house, where instead of all five dice being counted for the two former and a constant number of points being awarded for the latter as in Yahtzee, the sum of the relevant dice are counted. This means that in three-of-a-kind, only the three dice which are the same are counted, giving the score of that dice number multiplied by three. The same applies for four-of-a-kind. For a full house, all five dice are relevant, and therefore the sum of all dice is needed to make up the score. The only category to retain

the Yahtzee trait to award a constant number of points that is not the sum of the dice is the Yahtzee/Yatzy category (all dice having the same value), which rewards 50 points, the same as in the original game.
Another difference is the rules for small and large straight. In Yahtzee a small straight consist of any four dice in a sequence, whereas a large straight is made up by any five dice in a sequence. In Yatzy there is only one way to fulfill each of these rules. For small straight all dice from one through five for are required. For large straight the required dice are two through six.
There are also two additional combinations; one pair and two pair, where the score is also only counted for the relevant dice. See Appendix A for a further explanation of the Yatzy rules.

Probability theory

Many people would claim that Yatzy is a game of chance, but that is not entirely true. There are opportunities during a round to change the outcome of the game and that is when it is decided which dice to roll or to keep and where to put the score. The main object for this project is to develop an algorithm that calculates the optimal strategy for Yatzy. Since Yatzy is a dice game it is strongly connected to probability theory. Expected value is a common concept in probability and it could be seen as an average. The expected value can be calculated as

Sum(x*P(x))

where x represents the value and P(x) is the probability of x happening. In our case, by calculating the expected value for every combination and choosing the outcome that has the highest value we will maximize the score.

Previous work

Algorithms to find an optimal way to play Yahtzee have been done many times before. One of the most famous authors in this area is Tom Verhoeff, who has written a report on the subject [2] and has a website where one can see all expected values for every possible choice while playing the game [3]. Regarding the Scandinavian version, Yatzy, the subject is not as fully explored, but algorithms to find an optimal strategy for Yatzy have been constructed before at Kungliga Tekniska Högskolan [4].

3 Approach

From a player's perspective a game of Yatzy consists of two parts; rolling dice and making choices. The former, assuming the dice are fair, is a completely random event, while the latter makes up the strategy of the game. After the first two rolls, the player can choose between two actions; rerolling any subset of the previously rolled dice and filling in points in one of the remaining categories. If the current dice combination does not award any points in that category, 0 points are given. After the third roll, the player must fill in a category. To keep the algorithm straightforward, the first two choices can be simplified as just choosing any subset of dice, including all dice. The dice are therefore carried forward to next round, and the choice of category is not made until after three "rolls", even if one or two of those does not include any actual re-roll of dice.

In our implementation, we consider the game as a graph consisting of nodes which depend on three aspects:

- Which categories are filled in and which are empty. There are 15 different categories, each independent of the others, which makes 2^15 = 32,768 number of combinations.
- The values of the dice thrown. Since the order of the dice does not matter it can be seen as an ordered set. The number of ways to pick *k* elements out of *n* distinct elements is bin(n+k-1,k)[1], which with *k* as 5 and *n* as 6 makes bin(10,5) = 252 different combinations.
- Current roll number. Corresponding to how many times the dice have been rolled (or, as stated above, been left unrolled) this can be 1, 2 or 3. However, to make the algorithm faster, nodes with roll number 0 are introduced. In these nodes the dice number are irrelevant, and their purpose is to keep the expected value before the first throw each round.

In total this makes 32,768*252*3 + 32,768*1 = 24,805,376 nodes, where the latter addend represents the nodes with roll number 0.

The algorithm is made recursive, starting with the last choice of the game

---

[1] bin(x,y) is the number of ways to select y elements out of x distinct elements. bin(x,y) = x!/(y!*(x-y)!)

and going backwards in a breadth-first manner. The base case is when there is only 1 category left and the roll number is 3. This is easily calculated as the points awarded for the remaining category for the current dice. The reason for the algorithm going backwards is because by doing so we can predict possible future states and thereby calculate the optimal solution.

For the rolling choice nodes, the expected value of each possible connecting node is looked up, by adding 1 to the roll number and setting the dice to all possible subsets of the current dice. The probability of getting to another node is for each subset the probability to throw the dice to make up that exact dice combination. From this, the expected value is calculated, by multiplying the probabilities for reaching each node with the expected value that node corresponds to.

For the third throw of each round, where a category has to be filled in, every empty category is checked, by setting the roll number to 0 (to get the probabilities before the first throw in the next round) and all dice to 0 (since no dice have been thrown before the first roll) and filling one category at a time. This takes at most 15 lookups, one for each unfilled category. The node with the highest expected value is chosen as the optimal one.

Implementation

The program implementation was made in Java. The graph was represented by two different HashMaps, a Java class mapping keys to values. The first HashMap maps each node to the expected value of that node. The other HashMap maps each node to an array containing the optimal choice to be made for that node; for the last roll that is an integer representing which category to fill, while for the others it is an array of the optimal dice to save. The index for dice to be saved are set to one, and those not to be saved are set to zero. For example, if in a certain node we have the dice 1,4,4,5,6 and the optimal strategy is to save the dice 4,4 and 6, the array is filled in as 0,1,1,0,1.

4 Results

The algorithm was too inefficient to finish running in a reasonable amount of time on a normal desktop computer. To make up for this, we only built a small part of the graph; namely the nodes which contain from 1 up until a certain number of empty categories. The simulations were run with this number starting as 1 and ending as 5. This is justified by the reasoning that the higher this number is, the closer to the actual optimal strategy we get. The empty categories are randomly chosen so that after a large number of simulation each combination of empty categories have been run approximately the same number of times.

The results are presented in figures 1 and 2 and Table 1. The explanation for the points varying between each step in the graphs lies in the fact that even scores are more common than uneven scores. The reason behind this is that many categories (twos, fours, sixes, pair, two pair, four of a kind, large straight and Yatzy) always awards an even number of points. Conversely, there is no category always awarding an uneven number of points.[2]
Table 1 clearly demonstrates that the average score for each category increases with the number of categories unfilled from the beginning of the game. We expect this trend to continue over the entire algorithm, as a greater number of options to choose from gives a higher probability for obtaining high scores. The exception in this case is the category collecting ones, which is explained by it being a lower priority the more categories there are to choose from, as both its average and maximum score are very low compared to other categories. This is also partially true for twos.

[2] Small straight awards either 0 (an even number) or 15 points. As can be seen in the category breakdown i Table 1, the average value is circa 4, meaning the category is filled about 4 times out of 15, and not filled for the remaining 11. Therefore it can not be seen as evening out discrepancy between even and uneven scores, but actually doing the opposite.

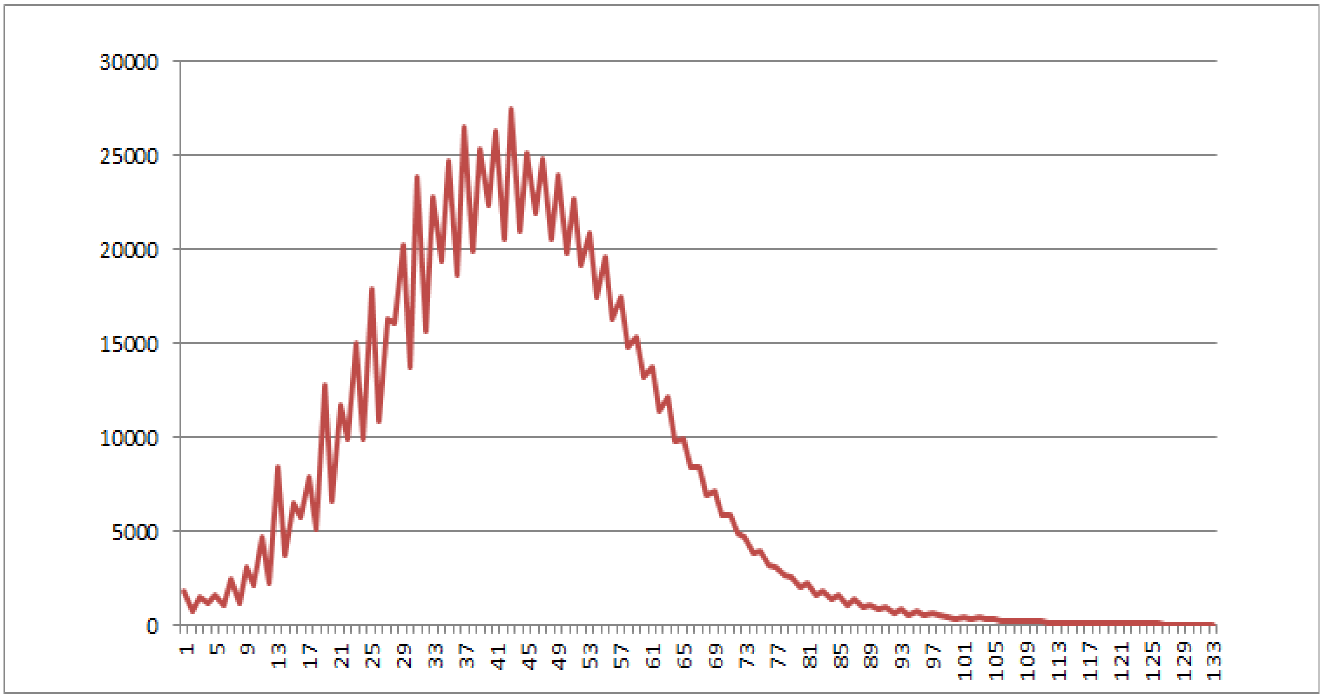


Figure 1. Points obtained from 1,000,000 simulations starting with 4 empty categories.

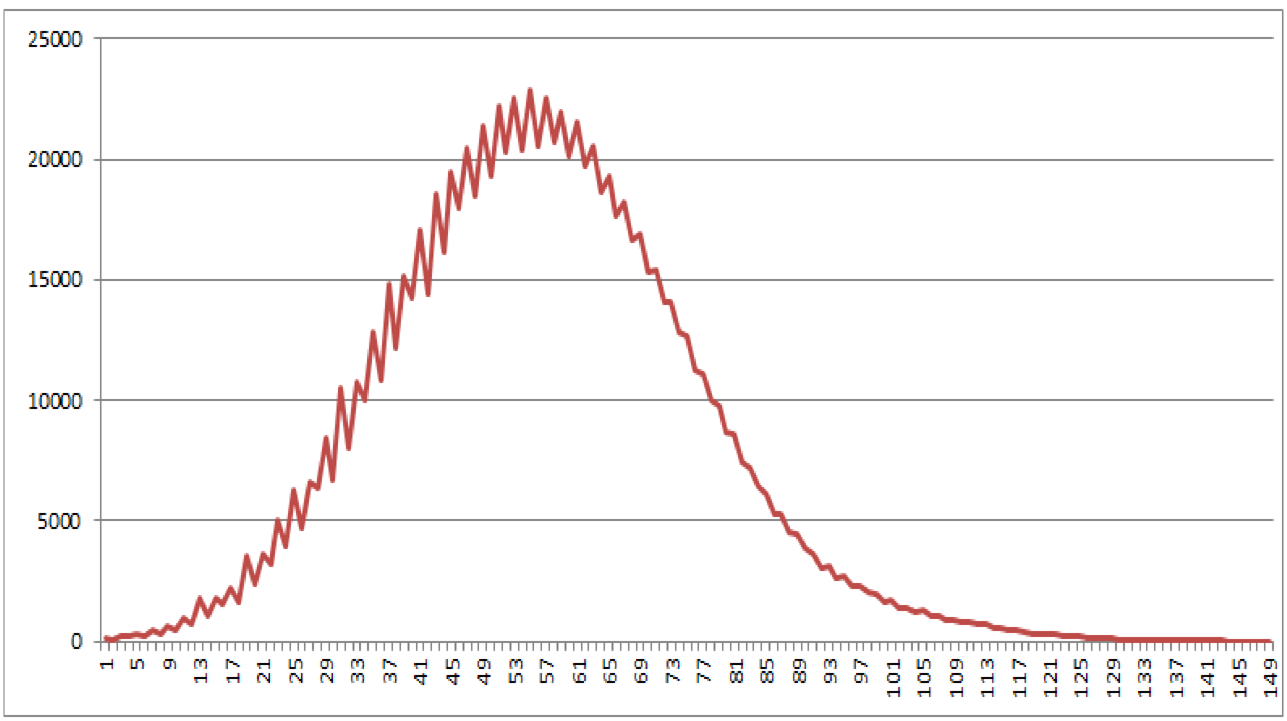


Figure 2. Points obtained from 1,000,000 simulations starting with 5 empty categories.

Category	1	2	3	4	5
Ones	2.11	1.80	1.61	1.49	1.43
Twos	4.21	4.15	4.18	4.23	4.27
Threes	6.32	6.66	6.92	7.10	7.32
Fours	8.42	9.17	9.67	9.97	10.22
Fives	10.54	11.60	12.27	12.62	13.09
Sixes	12.68	14.16	15.00	15.46	15.98
Bonus	0	0	0	0	0.000001
Pair	10.63	10.88	11.02	11.10	11.16
Two Pair	11.95	15.41	17.00	17.91	18.42
Three of a kind	9.59	12.22	13.62	14.22	14.90
Four of a kind	4.56	6.65	8.54	10.01	11.72
Small straight	2.93	3.23	3.55	3.56	4.06
Large straight	3.94	4.94	5.93	6.92	7.84
Full house	6.98	10.46	13.31	15.37	16.97
Chance	23.35	23.58	23.70	23.87	23.83
Yatzy	2.30	2.90	3.87	4.84	6.38
Total	**120.50**	**137.86**	**150.19**	**158.67**	**167.60**
Avg./ category	**8.02**	**9.19**	**10.01**	**10.64**	**11.17**

Table 1. Average scores for 1,000,000 simulations. The columns marked 1-5 represents number of categories unfilled by the start of the game.

5 Conclusions

Since we were not able to run a complete game simulation we do not have an expected value for an entire game. However, as we ran the program with a small number of categories left, we could see that the average scores increased along with that number. We therefore have a minimal estimate, which is the sum of all average scores from all categories. For the highest number of empty categories we ran (5) this score was 167.60.

Comparison to other algorithms

We can compare the correlation between the category scores to those of Tom Verhoeff's report *Optimal Solitaire Yahtzee Strategies* [2]. In that report, the rules of Yahtzee are used, as opposed to Yatzy, but some categories can still be used for comparison. For example, we can see that those scores for the single value categories are consistently around 25-50% higher than ours.[3] In the report *Optimal Yatzy Strategy* [4], the game of Yatzy with Scandinavian rules is analyzed, giving an expected score of 248.63. We think it is reasonable for our algorithm to achieve an average near this, considering our actual results and the fact that a similar approach to the problem was used.

Another aspect our algorithm does not take into account is the bonus which is awarded when the upper section of the scoreboard (the categories consisting of ones through sixes) has a score of at least 63 points. It is counted in the total number of points during simulation, but the algorithm makes no attempt to optimize the game to take that factor into account when building the graph.

In conclusion, we did not reach our goal of finding an efficient algorithm to calculate an expected value for a full game of Yatzy, but instead made an algorithm which works for parts of the game, and serves of use for a player during the last few rounds of the game.

[3] The exception is the category *ones*, see section 4 for further explanation.

References

1. Hasbro. The history of Yahtzee. [homepage on the Internet] 2007 [updated 8 August 2007; cited 11 April 2013] Available from: http://web.archive.org/web/20070808043322/http://www.hasbro.com/default.cfm?page=ci_history_yahtzee
2. Verhoeff, T. Optimal Solitaire Yahtzee Strategies. Technische Universiteit Eindhoven, 2004.
3. Verhoeff, T and Scheffers, E. Optimal Solitaire Yahtzee Player. [homepage on the Internet] 1999 [updated 1999; cited 11 April 2013] Available from: http://www-set.win.tue.nl/~wstomv/misc/yahtzee/osyp.php
4. Larsson, M. and Sjöberg, A. Optimal Yatzy Strategy. Kungliga Tekniska Högskolan, 2012.
5. Alga. Yatzy Resespel. [homepage on the Internet] 2011 [updated 2 February 2011; cited 11 April 2013]. Available from: http://www.algaspel.se/~/media/Alga/Files/Rules/flersprakiga/38018737_ReseYatzy.ashx

Appendix A - Official Yatzy Rules

The following section contains the official Yatzy rules from game developers Alga [5]. It should be noted that our algorithm follows the third option in the list of game variants.

The aim of Yatzy is to roll as many of the dice combinations on the Yatzy scorecard as possible. The winner is the player who scores the most points.

Playing a turn

The player makes an initial throw (by throwing 5 dice), and can then make 1 or 2 rethrows (by throwing any number of dice). Add up the number of points scored, and enter the total on the scorecard.

Choose one of the three following options before starting.

1

Play in order, from top to bottom.

2

First play the upper section in any order, and then play the lower section in any order.

3

Play in any order.

If no points are scored, the player must place a zero in the appropriate box (or in any box if option 3 has been chosen).

Dice combinations:

Collect ones-sixes:

Only the total of all the dice which are the same is counted. (E.g. throwing 1-1-1-4-2 gives three points.)

Bonus:

A player who scores 63 points or more in the upper section of the scorecard gets 50 bonus points!

One pair/Two pairs:
The total of the pair(s) is counted. N.B. Throwing 4-4-4-4-6 does not count as two pairs. Both pairs must be different.

Three-of-a-kind/Four-of-a-kind:
Only the total of all the dice which are the same is counted, e.g. three 4s (= 12 points) or four 2s (= 8 points).

Small straight:
Throw 1-2-3-4-5. This throw scores 15 points.

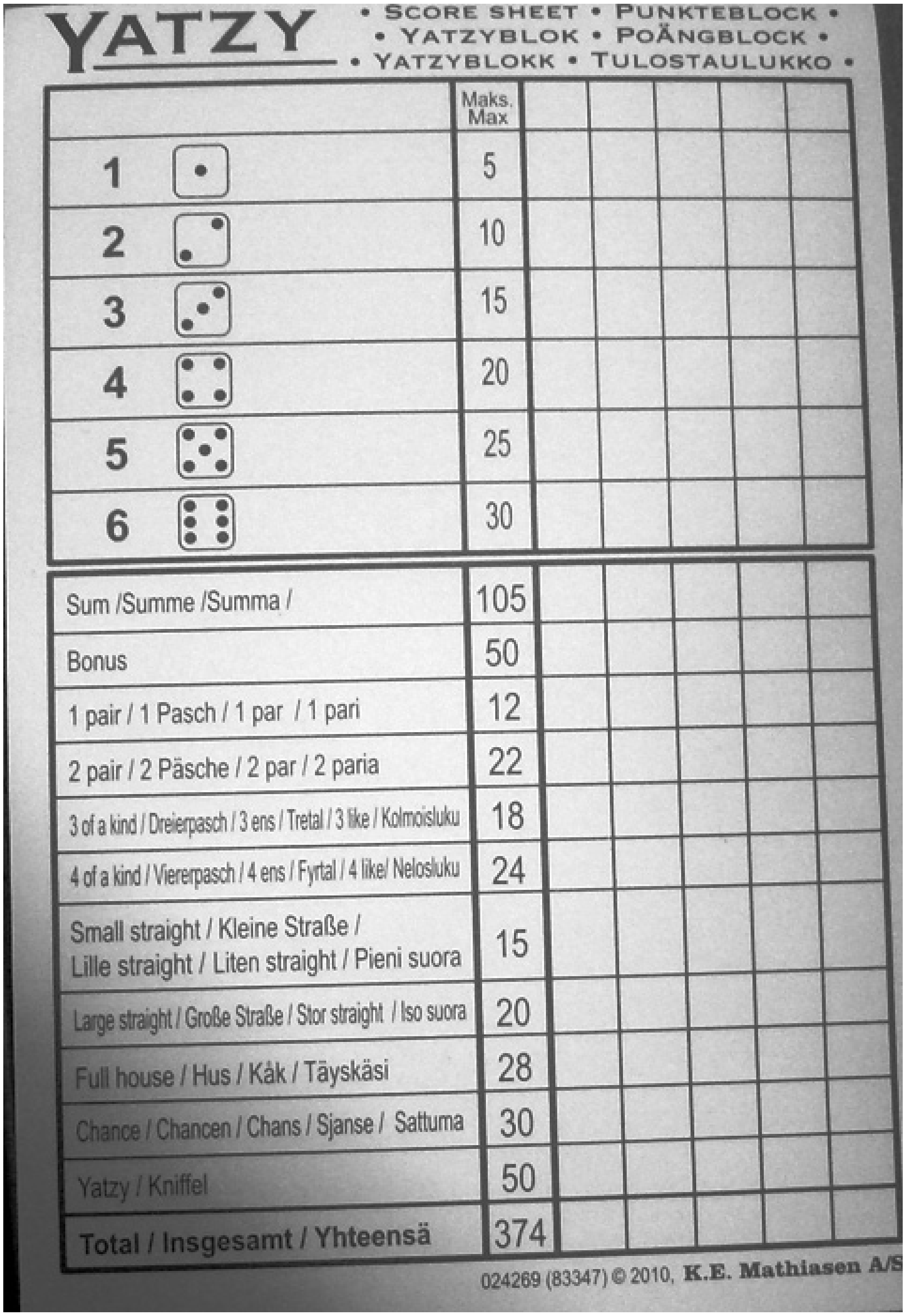
Large straight:
Throw 2-3-4-5-6. This throw scores 20 points.

Full house:
Throw any pair and any three-of-a-kind, e.g. 2-2-5-5-5. Note that the pair and the three-of-a-kind must be different! Count up the total and enter it on the scorecard.

Chance:
Score the total number of points shown on all the dice.

Yatzy:
Throw all five dice the same. This throw scores 50 points

Appendix B - Scoresheet


YATZY • SCORE SHEET • PUNKTEBLOCK • YATZYBLOK • POÅNGBLOCK • YATZYBLOKK • TULOSTAULUKKO •

	Maks. Max					
1	5					
2	10					
3	15					
4	20					
5	25					
6	30					
Sum /Summe /Summa /	105					
Bonus	50					
1 pair / 1 Pasch / 1 par / 1 pari	12					
2 pair / 2 Päsche / 2 par / 2 paria	22					
3 of a kind / Dreierpasch / 3 ens / Tretal / 3 like / Kolmoisluku	18					
4 of a kind / Viererpasch / 4 ens / Fyrtal / 4 like/ Nelosluku	24					
Small straight / Kleine Straße / Lille straight / Liten straight / Pieni suora	15					
Large straight / Große Straße / Stor straight / Iso suora	20					
Full house / Hus / Kåk / Täyskäsi	28					
Chance / Chancen / Chans / Sjanse / Sattuma	30					
Yatzy / Kniffel	50					
Total / Insgesamt / Yhteensä	374					

024269 (83347) © 2010, K.E. Mathiasen A/S

Scoresheet for Yatzy showing the name and maximum value of every category in English, German, Danish, Swedish, Norwegian and Finnish.