Farkle Project

Operations Research II

Fall 2009

By: Jason Hwa

Hon Ming Quek

Shuang Chen Shen

Michael Yang

**The Game**

 The Farkle dice game is an old game which has evolved through different names but has maintained the same principle. The game involves a player rolling 6 dice, and then trying to make combinations out of the result which are worth points. Each time a die is taken to make a combination it cannot be used again to roll the next round unless there are no dice left, in which case all 6 dice can be re-rolled. Before or after each roll the player can choose which dice to pick up and whether or not to continue rolling, or save the points. The reason one would want to save the points is because if no combinations can be made from a roll, called a farkle, all points that round are lost. If the player gets 3 farkles in a rown, 500 points are deducted from the players saved points. The goal is to accumulate the most points through 10 rounds. The list of combinations and point values are shown in the table below:

|  |  |  |
| --- | --- | --- |
| Dice | Explanation | Point Value |
| One | One’s are worth 100 points | 100 |
| Five | Five’s are worth 50 points | 50 |
| Three of a kind One’s | 3 one’s are worth 1000 points | 1000 |
| Three of a kind 2,3,4,5, or 6 | Three of a kind x:{2,3,4,5,6} are worth x\*100 points. | Number\*100 |
| Straight(one of each number) | This combination can only be formed on the first roll because it requires 6 dice. | 1500 |
| 3 pair | If the 3 pair consists of One’s or Five’s their individual values are not added. | 750 |
| 4 of a kind, 5 of a kind, 6 of a kind | For each additional die of the same number after 3 of a kind, the points are multiplied. For example 4 Two’s is worth 200+200.  | Value of 3 of a kind\*(1+number of addition die of the same value)  |

 For our project we are trying to answer two questions: which dice to pick up and whether or not to roll again. We are trying to find the optimal strategy for this game. Initially, we figured it would be a simple task of finding the probabilities of different outcomes and calculating the expected values. But as we tried to find the optimal game, we ran into some problems. It wasn’t as simple as just calculating the probabilities of the 6 dice roll and then 5 dice and so on because we also had to consider the next roll and cases where we picked up all the dice for a new roll. The fact that the game had the potential to go on forever was the basis of many complications.

 In order to calculate the expected value of a roll, we had to make a few simplifying assumptions. We assumed that on the last roll, the player would pick up all available points before banking his score. We could then calculate the player's expected score as if the next roll was his last. This enabled us to find the optimal strategy at any given point instead of looking at the game's potentially infinite outcomes. Near the end of our project, we experimented with calculating the expected points when the player is able to pick up all the dice for a new game. We found out that the only difference with this scheme is the ability to roll 3 dice with less than 385 points instead of having to bank anything over 300.

**Dynamic Programming Problem**

 Farkle can be casted as a dynamic programming problem. Each roll of the dice corresponds to a different state each with various possible outcomes and the choice of whether to continue rolling. The states are defined by the player's current score, the number of dice left on the table, and whether the previous two games were farkles. The number of farkles is important because if the player already has two farkles, he might stand to lose 500 points if he does not have a scoring combination on the next roll. We define our decision for whether to roll as a function of the current score, the number of dice left on the table, and whether there has been two farkles:

f(c = current score, d = die left of table, h = whether there has been 2 farkles)

f(c, d, h) = max{c, $\sum\_{ωϵΩ}^{}P(w)\*f(c + s(w), d(w), h) $where s is a score function.

 The expected payoff of rolling the next set of dice is the payoff of all events weighted by their probabilities. The function s(w) returns the additional points we stand to win given a certain outcome. Hence, our function tells us whether to continue rolling by comparing whether our current score is higher than the expected payoff from rolling once more.

Example:

 Assuming we have 500 points currently, 5 dice left on the table and no farkles so far:

f(500, 5, N) = max{500, P(1xxxx)\*f(600, 4, N) + P(5xxxx)\*f(550, 4, N) + P(11xxx)\*f(700, 3, N) + P(55xxx)\*f(600, 3, N) + P(222xx)\*f(700, 2, N) + P(333xx)\*f(800, 2, N) + P(444xx)\*f(900, 2, N) + P(555xx)\*f(1000, 2, N) + P(666xx)\*f(1100, 2, N) + …}

 After rolling, we're faced with a set of dice on the table. Our optimal strategy is given by choosing the maximum payoff given the various possible scoring combinations. A is the subset of possible scoring combinations, given the faces of the dice after our roll:

$max\_{aϵA}${f(c + s(a), d(a), h)} where A = subset of Omega which are possible choices

Example:

Assuming our dice faces are {1, 2, 2, 2, 4, 6} after the first roll, our optimal strategy is given by:

max$\left\{\begin{array}{c}f\left(100, 5, N\right)if we pick \{1\}\\f(200, 3, N) \&if we pick \{2, 2, 2\}\\f(300, 2, N) \&if we pick \{1, 2, 2, 2\}\end{array}\right.$

**The program**

 The above calculations are tedious and time consuming which one cannot calculate on the spot during a game. Therefore, our group has created a program which calculates the expected value of certain choices. We chose to use excel for our program because it is quick to formulate tables for all possible outcomes, however, the programming is limited to the capabilities of Visual Basic. After listing all the possible outcomes for 6 dice, 5 dice, 4 dice, 3 dice, 2 dice, and 1 die, we created a function which calculated the additional points one would earn for that particular outcome. In the column next to points, we listed the amount of points one would end up with given their current points. This is important because if the player farkles, he loses all of his current points instead of simply not gaining points. After all the possible scores are calculated, the program returns the average score based on how many dice need to be rolled.

 In the top left corner of the tab labeled "6 Dice Game" are the cells for entering the inputs. Current score is entered into cell A2, number of dice remaining into cell B2, and "y" or "n" is entered in C2 for whether or not the previous two rolls were farkles. The expected value of the next roll is displayed in cell A5. We will use the example from the previous page to show how the program works. In this example, we are facing a 6 dice outcome which shows {1, 2, 2, 2, 4, 6}. We use the calculator to find the expected value of each option:

max$\left\{\begin{array}{c}f\left(100, 5, N\right)=317.69\\f\left(200, 3, N\right)=251.94\\f\left(300, 2, N\right)=300\end{array}\right.$

 It is now clear that the optimal strategy is to pick up the one for 100 points and reroll the other 5 dice. A simple dice simulator is provided with the program and while playing around with different situations, we came up with a list of interesting findings.

**Interesting Findings**

 The most important finding to remember when playing are the rerolling thresholds. With a threshold of 17000 points, it is virtually always optimal to roll six dice when given the chance. If the player's current score is less than 2900 points, he should roll five dice. The player should roll four dice if his current score is 900 points or less and he should roll three dice if his score is under 385. Finally, if the player has enough points to bank with two or less dice left, he should bank instead of rerolling. These thresholds are only estimates but they will be the most useful information to remember when playing the game. If there has already been two farkles, these thresholds change to 16000, 2400, 400, 300, 300, 300 respectively.

 One interesting case we discovered is when a player faces the dice roll {1, 2, 2, 2, X, X}, where X denotes a 3,4, or 6. This case was already analyzed in the previous section but it is interesting to note that if the outcome was {5, 2, 2, 2, X, X}, it would still be optimal to pick up the 5 for 50 points and reroll five dice. If the outcome had been {1, 2, 2, 2, 5, X} instead, it would be optimal to pick up all the points and move on to the next game.

 Another interesting situation is which dice to pick up given the player has 100 points and faces the five dice situation {1, 5, X, X, X}, where the 3 X's don't give any points. The player can either pick up the 1 for 200 total points and reroll four dice or pick up both for 250 total points and reroll 3 dice. It is inadvisable to pick up just the 5 for 50 points and reroll four dice because picking up just the 1 is trivially a superior decision. It turns out that f(200, 4, N) = 312.04 and f(250, 3, N) = 287.98 which indicates the optimal strategy is to pick up an extra 100 points and reroll 4 dice.

**Conclusion**

 Using game theory and dynamic programming, we analyzed the dice game of Farkle. Our function computes the expected value of a decision based on the amount of points the player currently has, the number of dice the player will reroll, and whether there has been two farkles or not. We implemented this function as an excel program and it computes the expected value when given inputs. This program does not exactly model the game since we made assumptions to simplify the programming but it is very useful for gaining insight into certain situations. One suggestion for future study is to create the program using a different language such as C++ or Java in order to deal with recursions.