Claremont Colleges
Scholarship @ Claremont

All HMC Faculty Publications and Research

HMC Faculty Scholarship

1-1-1995

Optimal Klappenspiel

Arthur T. Benjamin
Harvey Mudd College

Derek Stanford '93
Harvey Mudd College

Recommended Citation

Benjamin, A.T., & Stanford, D. (1995) Optimal Klappenspiel. Journal for Undergraduate Mathematics and Its Applications, 16(1): 11-20.

This Article is brought to you for free and open access by the HMC Faculty Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in All HMC Faculty Publications and Research by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.

The UMAP Journal

The Journal of Undergraduate Mathematics and Its Applications

Optimal Klappenspiel

Page 11

COMAP

Spring 1995

1

ıes.
vill

the

m-

he

nt-

ht
ou

Optimal Klappenspiel

Arthur T. Benjamin
Dept. of Mathematics
Harvey Mudd College
Claremont, CA 91711
benjamin@hmc.edu

Derek Stanford
Dept. of Statistics
University of Washington
Seattle, WA 98195
stanford@stat.washington.edu

Introduction

The game Klappenspiel ("flipping game") is a traditional German game of flipping tiles according to dice rolls. In this paper, we derive the optimal strategy for this game by using dynamic programming. We show that the probability of winning using the optimal strategy is 0.30%.

The Rules of the Game and Notation

The game begins with ten tiles, numbered 1 through 10, initially all face up. The object of the game is to reach the position where all of the tiles are face down. Two six-sided dice are rolled, and the player has a choice:

- flip over two face-up tiles corresponding to the individual die values, or
- flip the face-up tile equal to the sum of the two dice.

Thus, once a tile is face down, it remains face down for the rest of the game. For brevity, we shall call the above choices the IND play and the SUM play. If one of these choices is not possible, then the play is forced. If neither play is possible, then the game ends; otherwise, the player rolls again. Note that if doubles are rolled, then only the SUM play is possible. (In fact, double-six immediately ends the game.) Since the IND play requires flipping two tiles, it cannot be used if there is only one tile left; a single remaining tile must be flipped on a SUM play. Tiles will be referred to by their number (e.g., "the 5 tile"); unflipped tiles will be "up," and tiles that have already been flipped will be "down."

For example, if tiles 4, 6, and 10 are up and the player rolls 4,6 (see **Figure 1**), then there is a choice between flipping tile 10 or tiles 4 and 6. It turns out that the optimal play is to flip the 4 and 6 tiles, since the probability of winning with only the 10 tile up is .083, while the probability of winning with the 4 and 6 tiles up is .079.

But what if other tiles are up? The presence of other tiles can greatly affect the optimal play. If we add the 1 tile to the example above, so that

Roll: 4,6										Optimal Play
			4		6				10	IND
1			4		6				10	SUM

Figure 1. For a given roll, the best play for each of two sets of face-up tiles.

tiles 1, 4, 6, and 10 are up, then the optimal play for the roll 4,6 changes to SUM. In fact, if tiles 4 and 6 were flipped, it would become impossible to win (since the 10 tile can be flipped only on a SUM play and the 1 tile can be flipped only on an IND play).

Methods

To find the optimal strategy, we will use dynamic programming to compare the win probabilities of the SUM play vs. the IND play, for every set of face-up tiles (called a "board") and every possible dice roll. We wrote a Pascal program to create and output an array OPT that stores the optimal play for each possible circumstance.

Dynamic programming can be thought of as the art of working backwards. We find optimal strategies to problems by building on already-computed optimal strategies to "smaller" problems. Usually, we need to specify the optimal solutions to only the simplest (often trivial) problems, together with a recurrence that builds upon solutions to smaller problems to create solutions to larger ones. Klappenspiel is naturally suited for dynamic programming, since once a tile is flipped down, it can never be flipped up again. For more examples and information about dynamic programming, see Benjamin and Huggins [1993], Denardo [1982], and Dreyfus and Law [1977].

There are 10 tiles, which can each be up or down, so there are $2^{10} = 1024$ possible boards. We can think of these boards as binary numbers with up tiles as 1 digits and down tiles as 0 digits. Tile 1 is in the 1s column, tile 2 in the 2s column, tile 3 in the 4s column, and so on, up to tile 10 in the 2^9=512s column. Thus, each board can be converted to an ordinary integer by assigning a value of 2^{j-1} to tile j and adding the values of all face-up tiles.

Example 1:

0	=	0000000000	All tiles down (Goal Position)
1	=	0000000001	Tile 1 up, all others down
199	=	0011000111	Tiles 1, 2, 3, 7, and 8 up, all others down
1023	=	1111111111	All tiles up (Initial Position)

Notice that a legal play can only reduce a board's number. Specifically, if from board number X, we have to play roll Y consisting of die values j and k, then the IND play (if legal) results in board number $X - 2^{j-1} - 2^{k-1}$, while the SUM play (if legal) results in board $X - 2^{j+k-1}$.

Example 2: We begin at board 1023 (all tiles up), and we want to flip tile 10. We have $1023 - 2^{10-1} = 511$, and we get board 511 = 0111111111, with Tile 10 down and all others up.

Dice Rolls

Since the order of the dice roll does not matter (i.e., rolling 2,1 is the same as rolling 1,2), there are only 21 distinct dice rolls possible. The probability of each doubles roll is 1/36, and the probability of any other roll is 2/36. For notational convenience, define prob(Y) to be the probability of dice roll Y.

Dynamic Program

Define

$$P[X] = \text{probability of winning from board } X \text{ using optimal strategy.}$$

Our base case is board 0, since $P[0] = 1$. Here are two other examples that we can compute quickly.

- If $X = 32$ (board consists of a lone 6 tile), then $P[32] = 5/36$, since we win if and only if our next roll sums to 6.
- If $X = 17$ (only tiles 1 and 5 up), then our only hope of winning is to roll 1,5 immediately. Thus, $P[17] = 2/36$.

Next, we define

$$\text{OPT}[X, Y] = \text{the optimal decision at board } X \text{ with roll } Y.$$

For example, if $X = 49$ (tiles 1, 5, and 6 up) and roll $Y = 1, 5$, then OPT$[X, Y]$ = IND, since the resulting IND position is superior to the resulting SUM position, i.e., $P[32] > P[17]$.

To find OPT$[X, Y]$ and $P[X]$, we must first determine if the play is forced. Say roll Y corresponds to die values j and k. We examine board X to see if tiles j, k, and $(j + k)$ are up. The player will have a decision to make only if all three tiles are up and j and k are distinct; otherwise, the play is forced. There are three types of forced plays: forced SUM, forced IND, and no play possible.

If j and k are distinct and tiles j, k, and $(j+k)$ are all up, then both SUM and IND are possible; we can find the optimal play by comparing the win probabilities of the outcomes of each play.

$$\mathrm{OPT}[X,Y] = \begin{cases} \text{IND}, & \text{if } P[X - 2^{j-1} - 2^{k-1}] \geq P[X - 2^{j+k-1}]; \\ \text{SUM}, & \text{otherwise.} \end{cases}$$

Notice that this calculation is recursive; it builds on the fact that we have computed P for all boards less than X.

Define

$$E[X,Y] = \begin{cases} P[X - 2^{j-1} - 2^{k-1}], & \text{if OPT}[X,Y] = \text{IND or IND is forced;} \\ P[X - 2^{j+k-1}], & \text{if OPT}[X,Y] = \text{SUM or SUM is forced;} \\ 0, & \text{if no play possible and } X > 0; \\ 1, & \text{if no play possible and } X = 0. \end{cases}$$

So $E[X,Y]$ is the probability of winning using the optimal strategy starting at board X with dice roll Y. Once we have found the optimal and forced plays for every possible dice roll at a certain board X, we can take a weighted average of the resulting win probabilities after each play Y to find $P[X]$:

$$P[X] = \sum_{\text{all } Y} \mathrm{prob}(Y) \cdot E[X,Y],$$

where Y varies over all 21 possible distinct dice rolls.

Note that $P[1023]$ is the expected value with all tiles up (i.e., at the beginning of the game).

Results

We wrote a Pascal program to employ the dynamic programming techniques discussed above. According to output from the program, $P[1023] = .0030$—the probability of winning this game is exceedingly small, even when the optimal strategy is used! There are seven rolls that are always forced; these are the double rolls and the 5,6 roll. Double 6 ends the game, and other doubles (if playable) must be SUM. Since there is no 11 tile, a roll of 5,6 (if playable) must be IND.

A sample section of the OPT array output is included as **Table 1**. The main pattern that emerges from these data is that

- *Rolls involving lower numbers tend to have an optimal play of IND, while*
- *rolls involving larger numbers tend to have an optimal play of SUM.*

The reason is consistent with our intuition: The more extreme the tile number, the less likely that the player will obtain a roll that would allow flipping that tile (see **Table 2**).

Table 1.
Sample of OPT. Since doubles rolls and the 5,6 roll are always forced, they are not listed here. F is the probability of winning using the optimal strategy from the given board. "I" indicates that IND is the optimal play, "S" indicates that SUM is the optimal play, and "." indicates that the play is forced or no play is possible.

Board	F	1,2	1,3	1,4	1,5	1,6	2,3	2,4	2,5	2,6	3,4	3,5	3,6	4,5	4,6
0	1.000	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1	.000	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	.028	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	.056	.	.	.	.	.	.	.	.	.	.	.	.	.	.
4	.056	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	.056	.	.	.	.	.	.	.	.	.	.	.	.	.	.
6	.059	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	.0062	I	.	.	.	.	.	.	.	.	.	.	.	.	.
8	.083	.	.	.	.	.	.	.	.	.	.	.	.	.	.
⋮	⋮														
1008	.0038	.	.	.	.	.	.	.	.	.	.	.	.	.	.
1009	.0016	.	.	.	I	I	.	.	.	.	.	.	.	.	.
1010	.0021	.	.	.	.	.	.	.	I	S	.	.	.	.	.
1011	.0020	.	.	.	I	S	.	.	S	S	.	.	.	.	.
1012	.0026	.	.	.	.	.	.	.	.	.	.	S	S	.	.
1013	.0023	.	.	.	I	S	.	.	.	.	.	S	S	.	.
1014	.0024	.	.	.	.	.	I	.	S	S	.	S	S	.	.
1015	.0020	I	.	.	I	S	I	.	S	S	.	S	S	.	.
1016	.0032	.	.	.	.	.	.	.	.	.	.	.	.	S	S
1017	.0023	.	.	I	I	S	.	.	.	.	.	.	.	S	S
1018	.0027	.	.	.	.	.	.	I	S	S	.	.	.	S	S
1019	.0024	.	.	I	I	S	.	S	S	S	.	.	.	S	S
1020	.0031	.	.	.	.	.	.	.	.	.	I	S	S	S	S
1021	.0025	.	I	I	I	S	.	.	.	.	S	S	S	S	S
1022	.0031	.	.	.	.	.	I	I	S	S	S	S	S	S	S
1023	.0030	I	I	I	I	S	I	I	S	S	S	S	S	S	S

Table 2.
Probability of a roll that would allow flipping with all tiles up (board 1023).

Tile	IND	SUM or forced SUM	Total
1	10/36	0	10/36
2	10/36	1/36	11/36
3	10/36	2/36	12/36
4	10/36	3/36	13/36
5	10/36	4/36	14/36
6	10/36	5/36	15/36
7	0	6/36	6/36
8	0	5/36	5/36
9	0	4/36	4/36
10	0	3/36	3/36

In other words, the more extreme tiles are more difficult to flip, and it is not surprising that we would want to flip the most difficult tiles possible on each roll.

Although this heuristic rule might lead us to try to find a ranking of the tiles in order of difficulty, the rule is not as simple as it first appears. Flipping any of the tiles 1 through 6 can make it more difficult to flip the remaining tiles; also, as we move to boards with fewer tiles, we increase the probability of obtaining an unusable roll, which would end the game. If we try to account for all of the factors of difficulty, we find that the optimal strategy for this game becomes too complex for a simple description. However, there are some features of OPT that are worthy of further investigation, and the optimal strategy can be well approximated.

The Nine Consistent Rolls

There are nine rolls that have a consistent optimal strategy; these are the seven forced rolls, the 1,2 roll, and the 1,3 roll. The 1,2 roll is the only roll for which the expected value of the IND play sometimes equals the expected value of the SUM play; when this occurs, either play is optimal, so the choice does not matter. Other than these cases, inspection of the entire OPT array shows that the 1,2 and 1,3 rolls are always forced or IND.

The Twelve Inconsistent Rolls

The remaining 12 rolls are more difficult to characterize. The optimal strategy cannot be described in a simple way because the optimal decision for roll j, k can change depending on the presence of tiles other than j, k, and $(j + k)$. Let us call this property *interference*.

Example 3: We roll 1,6 and the play is not forced. Is IND or SUM the optimal play? The answer depends on which other tiles are up.

Tiles up	Roll	Optimal play
1, 6 ,7	1,6	IND
1, 2, 6, 7	1,6	SUM
1, 3, 6, 7	1,6	SUM
1, 2, 3, 6, 7	1,6	SUM
1, 4, 6, 7	1,6	IND

We see from the example that the presence of some tiles causes a switch in the optimal play from IND to SUM, while the presence of others do not. Is there some consistency in the interference caused by a given tile? **Tables 3a** and **3b** combine to show that every tile can cause a change in optimal play from IND to SUM, and **Table 4** shows that every tile except tile 1 can cause a change from SUM to IND.

Table 3.
Interference caused by an additional tile can change optimal play from IND to SUM.

a.
Boards with only tiles j, k, and $(j + k)$ up.

Tiles up	Roll	Optimal play	With tile	Optimal play
1, 2, 3	1,2	IND		(always IND or does not matter)
1, 3, 4	1,3	IND		(always IND)
1, 4, 5	1,4	IND	2, 3, or 6	SUM
1, 5, 6	1,5	IND	2, 4, or 8	SUM
1, 6, 7	1,6	IND	2	SUM
2, 3, 5	2,3	IND	1 or 4	SUM
2, 4, 6	2,4	IND	1, 5, 7, or 10	SUM
2, 5, 7	2,5	IND	1	SUM
2, 6, 8	2,6	IND	3	SUM
3, 4, 7	3,4	IND	2	SUM
3, 5, 8	3,5	IND	4	SUM
3, 6, 9	3,6	IND	5	SUM
4, 5, 9	4,5	IND	6	SUM
4, 6, 10	4,6	IND	7	SUM

b.
Other boards.

Tiles up	Roll	Optimal play	With tile	Optimal play
4, 6, 10	4,6	IND	8	SUM
4, 6, 10	4,6	IND	9	SUM
1, 2, 3, 4, 5, 7, 9	2,3	IND	10	SUM

Table 4.
Interference from any one of tiles 2 through 10 can change optimal play from SUM to IND.

Tiles up	Roll	Optimal play	With tile	Optimal play
1, 2, 4, 6, 7, 8	1,6	SUM	10	IND
1, 2, 4, 5, 6, 7, 8, 10	1,4	SUM	9	IND
3, 4, 5, 6, 7, 10	3,4	SUM	8	IND
1, 2, 4, 5, 6, 8, 9, 10	1,5	SUM	7	IND
1, 2, 3, 4, 5, 7, 9	1,4	SUM	6	IND
1, 2, 3, 4, 6, 7, 8, 9, 10	2,4	SUM	5	IND
1, 2, 6, 7, 8, 10	1,6	SUM	4	IND
1, 2, 6, 7, 8	1,6	SUM	3	IND
1, 4, 6, 7	1,6	SUM	2	IND

The Boards with Only Tiles j, k, and $(j + k)$ Up

When the only tiles up are those directly involved in the dice roll (see **Table 3a**), then the optimal play is always IND (except for the board with tiles 1, 2, and 3 up, for which the expected value of the IND play equals the expected value of the SUM play). Consider a roll j, k for which this can occur (i.e., not doubles or 5,6). If we flip the $(j + k)$ tile, then we must subsequently either roll j, k, or we must flip both tiles on SUM plays. It makes sense intuitively that this would be more difficult than flipping the $(j + k)$ tile on a single SUM play. In fact, we can prove this directly by considering that if we make the SUM play, then our winning chances are exactly $P(\text{Roll } j, k) + 2{*}P(\text{sum} = j){*}P(\text{sum} = k)$, which is less than or equal to $2/36 + 2(3/36)(5/36) = 102/1296$. (The $(3/36)(5/36)$ term comes from the "best-case" scenario $j = 4$, $k = 6$.) If, however, we make the IND play, then our chance of winning is exactly $P(\text{sum} = j + k)$, which is at least $3/36$ ("worst-case scenario"), which exceeds $102/1296$.

Suboptimal Strategies

We have seen that the optimal strategy is too complex to be captured in only a few sentences. However, it is possible to find a suboptimal strategy that does almost as well as the optimal one. With the optimal strategy, the probability of winning is .0030; but a relatively simple heuristic yields a probability of winning of .0029. For comparison, a few other simple strategies are also examined, with results shown in **Table 5**.

Table 5.

Suboptimal strategies.

Strategy	Probability of winning
Anti-optimal	.0011
Always play IND	.0016
Random	.0017
Always play SUM	.0019
Heuristic	.0029
Optimal	.0030

First, to obtain a lower bound on winning the game, we look at the "anti-optimal" strategy. In this strategy, every time the player has a choice, the play with the lower winning chances is chosen. Thus, by **Table 5**, no matter how hard you try to lose (but playing legally), your winning chances are at least .0011.

Always playing SUM fares better than always playing IND. The SUM-only strategy would flip tiles 7 to 10 whenever possible, whereas the IND-only strategy would prefer tiles 1 to 6. Since the player usually encounters

more opportunities to flip tiles 1 to 6 than tiles 7 to 10 (see, for example, **Table 3**), it follows that tiles 7 to 10 tend to be more difficult to flip. Hence, we would intuitively expect the SUM-only strategy to do better than the IND-only strategy, since the former more often corresponds to flipping the more difficult tiles.

The random strategy, as its name implies, randomly selects a play whenever there is a choice. This strategy yields a probability of winning between the SUM-only and IND-only strategies.

The heuristic strategy chooses a play based on the dice roll: If the sum is 6 or less, play IND; if the sum is 7 or more, play SUM. We devised this strategy by inspecting the OPT array to find the most common optimal choice for each dice roll. This heuristic approximates quite well the results of the optimal strategy.

The optimal strategy offers an improvement of 172% over the anti-optimal strategy and 76% over random play. The heuristic strategy comes close to this, doing 163% better than the anti-optimal strategy and 71% better than random.

Hence, we must conclude that although Klappenspiel is a game that can be described simply, its optimal strategy is quite involved. Although the dynamic programming technique makes it possible to find explicitly the optimal strategy for this game, it offers no guarantees that the data can be easily reduced to a few rules. Nonetheless, inspection of the data allows us to find a good approximation of the optimal strategy in a simple heuristic.

References

Benjamin, Arthur T., and Eric Huggins. 1993. Optimal blackjack strategy with "lucky bucks." *The UMAP Journal* 14 (4): 309–318.

Denardo, Eric V. 1982. *Dynamic Programming: Models and Applications*. Englewood Cliffs, NJ: Prentice-Hall.

Dreyfus, Stuart E., and Averill M. Law. 1977. *The Art and Theory of Dynamic Programming*. New York: Academic Press.

About the Authors

Arthur Benjamin received a B.S. from Carnegie-Mellon University in applied mathematics and the M.S.E. and Ph.D. degrees in mathematical sciences from Johns Hopkins University. In 1988, he received the Nicholson Prize of the Operations Research Society of America for his paper "Graphs, maneuvers, and turnpikes" (*Operations Research* 18 (1990) 202–216). In addition to teaching mathematics at Harvey Mudd, he enjoys playing tournament backgammon, racing against calculators, and performing magic.

Derek Stanford is working toward a Ph.D. in statistics at the University of Washington. His research includes image analysis and stochastic modeling. While attending Harvey Mudd College and the Claremont Graduate School, he developed an interest in experimental music, which he continues to pursue in Seattle.