Moving Knife Algorithms

A Comparison between Discrete and Exact Solutions

John Fang, Zheng Luo, Sreeharsha Sistla

Abstract

Fair division algorithms, especially the ones that require continuous observation, tend to assume synchronicity in decision making and knife movements in theory. However, in practice, it is not always possible to give an exact solution to a fair division algorithm without making some assumptions or error bounds. As such, we would like implement both an exact and discrete solution to given algorithms and compare how good of an approximation the discrete solution is.

In this experiment, we will be implementing this concept to two such algorithms, both Moving Knife algorithms. The first of the two is the classic Moving Knife algorithm, also known as the Dubins-Spanier moving knife. The second of the two algorithms is a custom variation to this moving knife algorithm, which we have coined the "Sistla-Fang" Moving Knife algorithm. In both cases, we will compare the discrete and exact solutions and determine how good an approximation the discrete version is. Further, we will compare the two algorithms mentioned and attempt to show that the "Sistla-Fang" Moving Knife is a more viable option when considering envy-freeness.

1. Algorithms

Dubins-Spanier

The Dubins-Spanier moving knife algorithm guarantees proportionality and is a continuous algorithm. There are no bounds on the maximum amount of players in this algorithm, thus it is a good candidate to run repeated simulations on while varying other parameters. However, the Dubins-Spanier moving-knife algorithm does not guarantee envy-freeness, as the last person to receive a piece will have the possibility of getting more than proportional in everyone's view. Specifically, all players will satisfy the property $v_i\,(\pi_i) = \frac{1}{n}$, but the last player to cut will receive a piece that is $v_{last}\,(\pi_{last}) \geq \frac{1}{n}$. The procedure for Dubins-Spanier moving knife is as follows:

(i) Start the knife on the very left side of the cake. Move the knife to the right side at a continuous pace. For discrete simulations, the knife would take a fixed step size every time.

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

(ii) Any player that sees v_i $(\pi_{left\ of\ knife}) = \frac{1}{n}$ will immediately call stop, and obtain the piece that is currently to the left of the knife.

(iii) Players that have obtained a piece drop out of the process while those who have not gotten a piece will continue to partake in the algorithm until everyone has a piece. The last player does not need to call stop, since he will be getting anything left over by default.

The exact version of the Dubins-Spanier algorithm can be calculated by figuring out where each individual player would call "stop". Whichever player calls "stop" first would then get the piece at exactly where "stop" was called. To figure out exactly where player i would call stop ***b***, we use the following relationship shown below, given the preference function ***$f_i(x)$***, the position of the last cut ***a***, and the total number of players **n**.

$$\frac{1}{n} = \int_b^a f_i(x)$$

Since player preference functions are continuous, piecewise, and linear equations, it is fairly easy to solve for **b** by using simple integration and algebra.

Sistla-Fang

As we observe in the classical Moving Knife algorithm, the last person to get a piece will always receive the most value. Without loss of generality, let the n^{th} player be the last to get a piece. This algorithm makes it so that v_n $(\pi_n) \geq \frac{1}{n}$, and v_i $(\pi_i) = \frac{1}{n}$ $\forall i \neq n$. However, one thing that we observed was that there is a simple variation to this algorithm guaranteeing n-1 of the n players 1/n of the cake while one player will receive exactly 1/n. The algorithm is as follows:

(i) Ask each of the n players to create markers (n-1 markers in addition to the right end of the cake being an implicit marker) that mark the cake into n equal pieces

(ii) Move a knife from left to right until it hits the first of a set of corresponding markers (by corresponding, we mean that value to the left of these markers is equal for all players)

   a. If the player corresponding to this first marker does not have a piece, give him the piece to the left of the knife

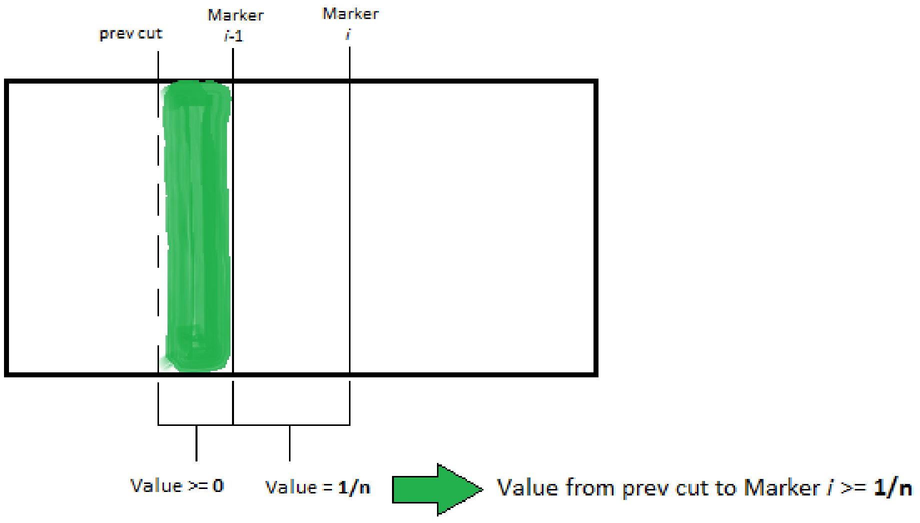
   b. If the corresponding player does in fact have a piece, continue moving the knife until it hits a marker corresponding to a player that does not have a piece

(iii) Continue this procedure until there is one player left, at which point we give the rest of the cake to that player

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

Proof of Result:

Assume there are *n* players. It is trivial to show that the first player to receive a piece receives value $\frac{1}{n}$. We want to show that the subsequent players receive value $\geq \frac{1}{n}$. Let us consider any player from the set of *n*. Let this player receive piece *i,* meaning that he receives a piece directly to the left of his *i*th marker. From the first step described above, we know that this player's value of the piece between his *i*th and (*i*-1)th markers is $\frac{1}{n}$. But because he did not receive the previous piece, the previous cut of the cake occurred before his (*i*-1)th marker. This means that he gets the piece between his *i*th and (*i*-1)th markers and then some. I.e. he receives value at least $\frac{1}{n}$. A figure corresponding to this is shown below. ■



* **Note**: because the player received a piece directly to the left of his *i*th marker, the prev cut MUST be to the left of his *(i-1)*th marker

Figure 1. Graphical representation of the Sistla-Fang Moving Knife algorithm.

The exact solution for this algorithm wasn't necessarily the most efficient solution, but definitely the simplest in our mind. To determine the exact solution, the most non-trivial step was determining the exact locations of each player's markers. This was done through multiple iterations over the set of all intervals from each player's preferences in addition to using the quadratic formula. We had to first determine in which specific interval a given marker was located. After we found this interval, we needed to figure out where the marker should be placed within this interval to make the value between the current marker and the previous marker $\frac{1}{n}$. This was where the quadratic formula came in. The formula was necessary because the values in each interval were not uniform, and it was sufficient because each

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

interval could be represented by a linear equation in the form *y = mx+b*. In terms of complexity, the exact solution is far more demanding than the discrete.

2. Analysis of Error

Error for our discrete algorithms will be defined as the following sum of squared differences (SSD),

$$SSD\ Error = \Sigma\ [v_n\ (\pi_{discrete}) - v_n\ (\pi_{exact})]^2$$

where error is summed over all players in the same trial. Both the discrete and exact version of the moving knife variant is ran on the same set of randomized data. By varying the amount of discrete steps as well as the amount of players, we find interesting data regarding the differences in discrete and exact solutions. Each of the figures below show the SSD averaged over 50 trials for each step size and number of players.

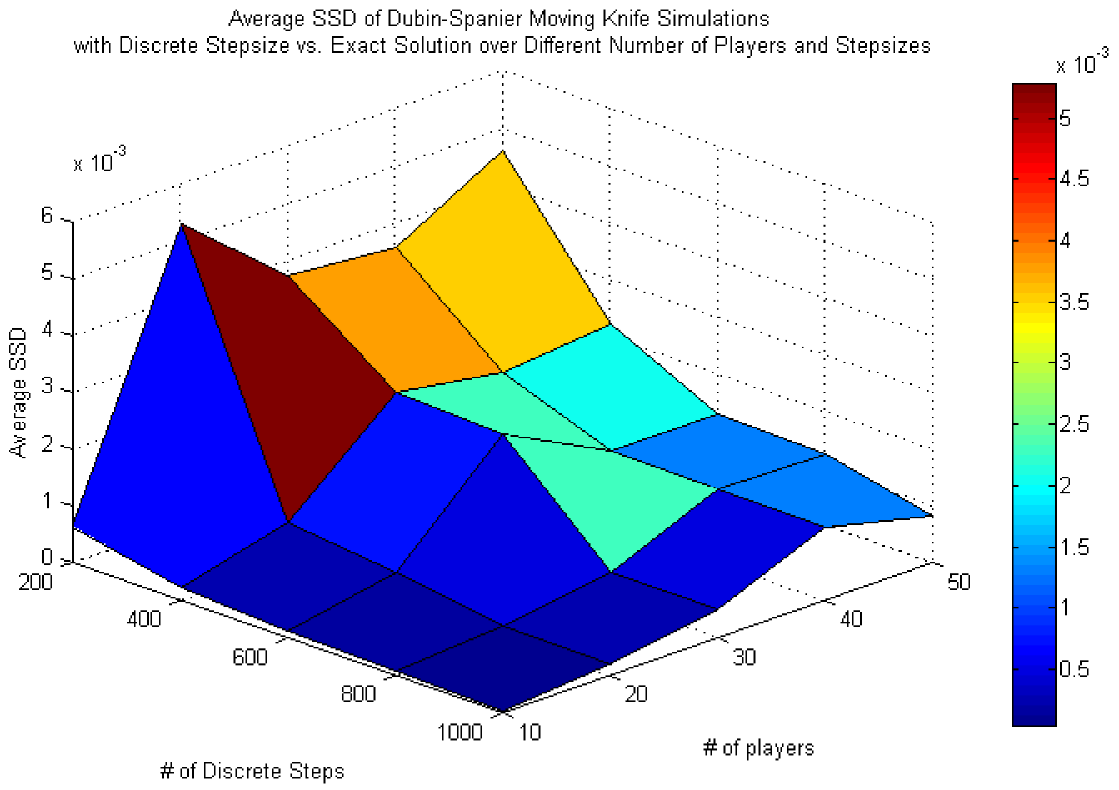


Figure 2. Average SSD of Dubins-Spanier discrete simulation in comparison to its exact solution.

*__NOTE__: Piece X refers to Player X's piece, not the piece X that was cut from the cake

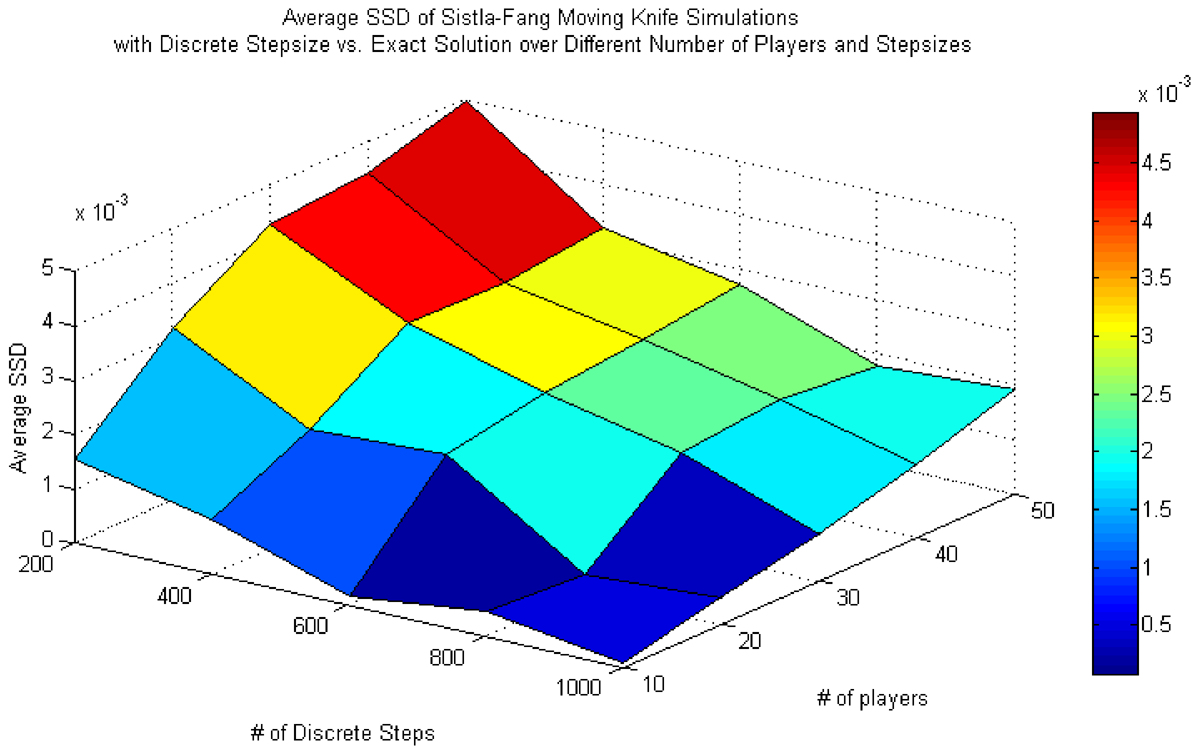


Figure 3. Average SSD of Sistla-Fang discrete simulation in comparison to its exact solution.

The trend of increasing error in the Sistla-Fang version is fairly trivial and expected, as seen in Figure 3. The decreased step size allows for more potential cases of over-allocation and under-allocation. Increasing the number of players has the same effect despite the errors being weighted by squaring. This would imply that doubling the amount of players would more than double the absolute amount of squared error.

However, the average SSD of the Dubins-Spanier for different step sizes and number of players seems much more irregular than the Sistla-Fang. A possible reason for this could be due to the fact that since the last player gets the entire surplus in the Dubins-Spanier algorithm, large fluctuations in the last player's preference function could be very sensitive to the precision of discrete steps. Increasing the number of steps taken seem to reduce this distortion, however.

On average, the SSD for both Dubins-Spanier and Sistla-Fang algorithms are roughly on the same magnitude. This would imply that for a given fixed step size and a given number of players both algorithms would be equally faithful at simulating the exact solution on average, although Sistla-Fang is potentially more consistent.

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

3. Envy(Sistla-Fang) $\leq$ Envy(Dubins-Spanier)?

One of the major reasons we wanted to consider a variation to the classical Moving Knife algorithm was to fix the envy issue. More often than not, we have players whose utility functions over the cake greatly vary. This tends to create a good deal of envy. Numerous simulations were run to see what the envy was like in both cases. On the next page, there are two graphs representing envy when given the same set of player preferences. Now keep in mind that these two graphs are just one sample. However, we felt that it was fairly representative of what we saw in MANY of our simulations. Both of the algorithms cannot create an envy-free situation. But, the Sistla-Fang algorithm, as seen in the two graphs tends to have "less" envy. What exactly do we mean by this? From the many test cases we ran, we noticed something about the pieces the players receive through the Sistla-Fang (SF) algorithm as compared to the classic Dubins-Spanier algorithm. The SF Moving Knife seems to limit how much a player can value another player's piece. That is, it seems to limit the degree of envy a player can have. Let us consider the two graphs below. One thing that clearly pops out in the Dubins-Spanier graph is Piece 3. The amount of envy for that piece is incredible. On the other hand, look at the corresponding Sistla-Fang graph. There isn't anything that really jumps out at you. Yes, Piece 1 seems to have some envy, but it is considerably less than what we saw in the Dubins-Spanier graph for Piece 3.

Why does this said "phenomena" occur? Suppose there are *n* players, and without loss of generality, let Player 1 get the very first piece. In Dubins-Spanier, *n-2* of the remaining *n-1* get $\frac{1}{n}$ while the last player to get a piece receives all the "spoils." On the other hand, in the Sistla-Fang algorithm, these "spoils" are distributed between the remaining *n-1* players instead of just the last player. A better distribution of the "spoils" seems to create a situation with less envy. We would like to point out that a lot of this is just speculation. Theoretically, we have not been able to prove our case, but from the data we have gathered, we feel that this is a much better Moving Knife algorithm than is Dubins-Spanier.

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

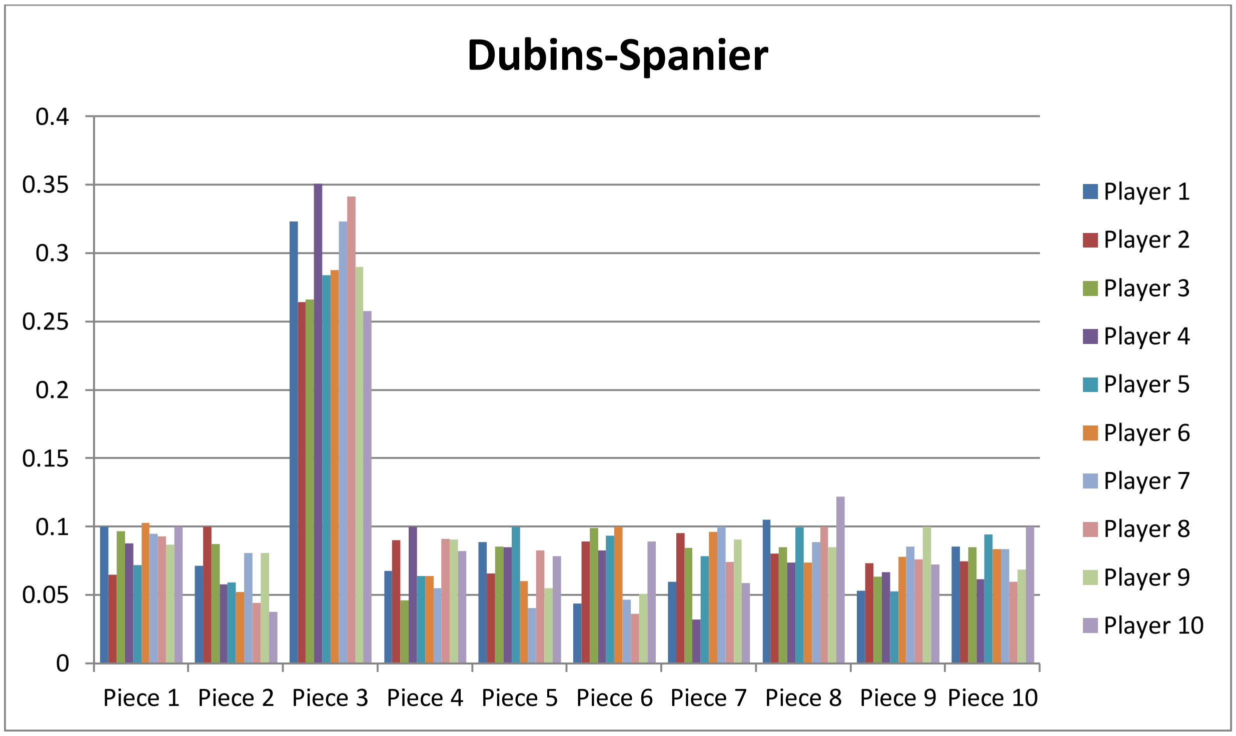


Figure 4: Utility distribution of each player's piece versus every other player's piece for the Dubins-Spanier moving knife algorithm.

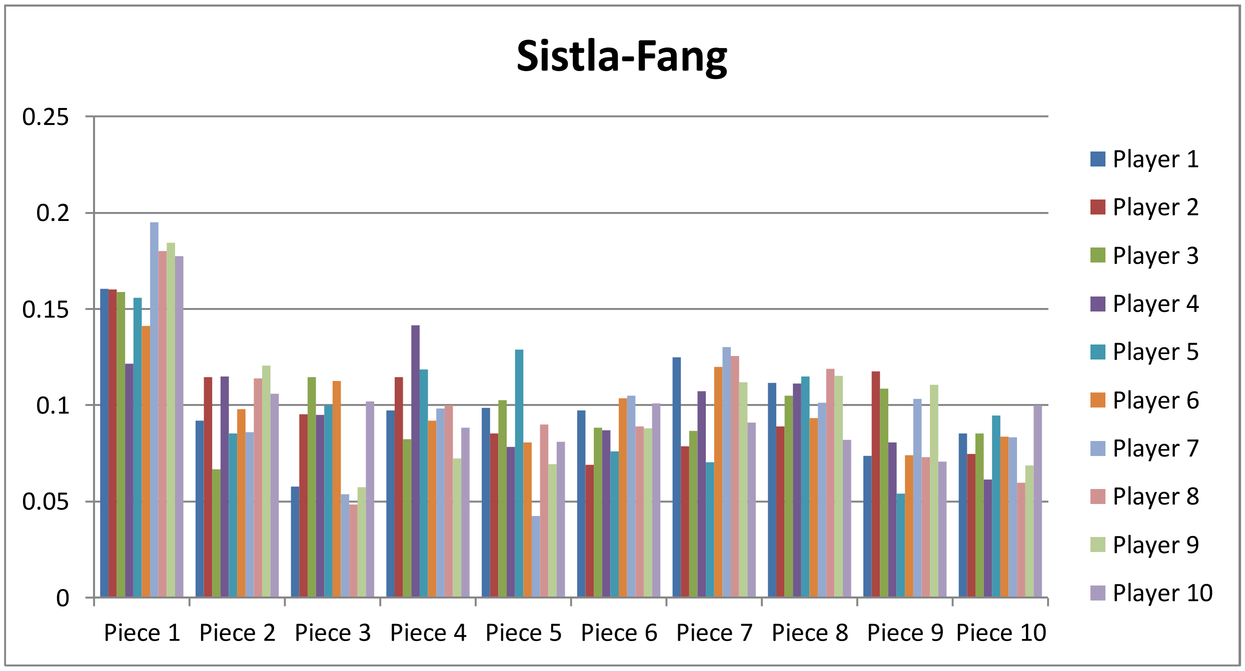


Figure 5: Utility distribution of each player's piece versus every other player's piece for the Sistla-Fang moving knife algorithm.

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake

Conclusion

By increasing the step size resolution as well as reducing the amount of total players, discrete moving knife simulations will be increasingly more accurate with respect to its exact solution on the same set of player preferences. Data shows that this trend is generally true for both variants of the moving knife algorithm shown in this paper, though much more consistent in the Sistla-Fang version.

If more time was given, we may have had the chance to compare other continuous algorithms that can be scaled for n players, as well as examining the differences in envy-freeness. We are also curious as to how these results would differ given more complicated preference files, ones that are not just linear piece-wise. The incredible amount of time it took to ran simulations also cut into our deadline. The data generated for the Sistla-Fang simulations took roughly three hours. This speed may be improved by better coding as well as by using a higher performance programming language. With faster simulations, we could potentially generate more smooth figures, as more samples can be sampled for each fixed step size and number of players.

***NOTE**: Piece X refers to Player X's piece, not the piece X that was cut from the cake