Minimal Number of Cuts for Fair Division

Jack M. Robertson and William A. Webb
Washington State University
Pullman, WA 99164-2930

Abstract. The problem of fairly dividing a piece of cake apparently originates with Hugo Steinhaus in 1948 at which time he raised the question of the number of cuts required in fair division algorithms. In this paper an algorithm requiring $O(n \log n)$ cuts is given, improving known algorithms which require $O(n^2)$ or more cuts. The algorithm is shown to be optimal in a certain class, and general algorithms are shown to allow a certain freedom of participants to choose pieces.

1. Introduction

The problem of fairly dividing a cake among n persons apparently originates with Hugo Steinhaus in 1948 [5]. He wrote at the time; "Having found during the war a solution for three partners, I proposed the problem for n partners to B. Knaster and S. Banach." They solved the general problem but Steinhaus went on to say; "Interesting mathematical problems arise if we are to determine the minimal number of cuts necessary for fair division." If he said anything more on the latter issue we apparently have no record of it.

Following the formulation given by Woodall [7], the general problem of fair division among n players $\{P_1, \ldots, P_n\}$ assumes a "cake" X which is a compact set in Euclidean space and probability measures μ_i, $1 \leq i \leq n$, on X where the μ_i-measurable subsets of X are the Lebesgue-measurable subsets of X. It is also assumed that if the Lebesgue measure of a subset is zero so is its μ_i measure for each i.

A fair division requires a partition $X = S_1 \cup \cdots \cup S_n$ of the cake into μ_i-measurable subsets so that person P_i thinks piece S_i is "fair." Different degrees of fairness can be sought. For example "fair" can mean:

I. $\mu_i(S_i) \geq \frac{1}{n}$ for $i = 1, 2, \ldots, n$ or
II. $\mu_i(S_i) > \frac{1}{n}$ for $i = 1, 2, \ldots, n$ or
III. $\mu_i(S_i) \geq \mu_i(S_j)$ for all i and j or
IV. $\mu_i(S_j) = \frac{1}{n}$ for all i and j.

Clearly requirement I is the weakest of the four. A number of algorithms are known for I, see [4], [5] and [7]. An existence solution for II is given in [2] and [3] provided the measures are not all identical; assuming the existence of a piece S for which $\mu_i(S) \neq \mu_j(S)$ for some i, j an algorithm for II is given in [8]. An algorithm for III is known for three persons, see [6] and [7], while existence of solutions for III and IV is proved in [1],[2], [6] and [7]. Alon [1] proves the existence of a solution for IV involving only $n^2 - n$ cuts, the best possible, but it seems implausible that there could exist an algorithm for IV that would terminate in any finite number of cuts. Concerning Steinhaus's remark, we note that classical algorithms for problem I generate $n!$ or $\frac{1}{2}(n^2 - n)$ pieces of cake [7] . In this paper we give an algorithm which requires $O(n \log n)$ pieces.

We first place restrictions on the algorithm which will guarantee that it be finite. The algorithm is allowed to branch but it is required that:

1. Only a finite number of times will any player be asked to cut an existing piece into two smaller pieces.
2. Each time P_i is required to make a cut the values of μ_i on the two new pieces are prescribed in advance of the cut by the algorithm. It is assumed P_i can make the required cut.
3. Only finitely many times will each player be required to evaluate the finite number of pieces at that stage.

In particular, the well known "moving knife" algorithm is eliminated. "Piece" is interpreted to mean a subset of X with non-empty interior. The boundaries are ignored in light of the assumption that sets with Lebesgue measure zero also have μ_i measure zero.

We now return to Steinhaus's question. Let us denote by $M(n)$, the minimal number of cuts required in any fair division algorithm of the cake for n players, where "fair" is defined by requirement I above. The algorithm is assumed to satisfy 1-3. "You cut I choose" establishes $M(2) = 1$.

We next show $M(3) = 3$. Suppose an algorithm exists which uses only two cuts. We may suppose that P_1 makes the first cut, resulting in pieces A and B such that $\mu_1(A) = r$ and $\mu_1(B) = 1 - r$ where the value r, $\frac{1}{2} \leq r \leq 1$, is specified by the algorithm. Since the algorithm must allow for any evaluations by P_2 and P_3 we may assume that $\mu_2(A) = \mu_3(A) = \frac{3}{5}$ and $\mu_2(B) = \mu_3(B) = \frac{2}{5}$.

If any player now cuts B into pieces B_1 and B_2, then the other two players may both evaluate both B_1 and B_2 as less than $\frac{1}{3}$. But since no further cuts are allowed, one of these two players must be assigned either B_1 or B_2, and so would not be satisfied. Similarly if P_1 cuts A then both P_2 and P_3 may evaluate both of the resulting pieces as less than $\frac{1}{3}$. Again some player would not be satisfied.

Finally, if P_2 (or equivalently P_3) cuts A into pieces A_1 and A_2 such that $\mu_2(A_1) = s, \mu_2(A_2) = \frac{3}{5} - s$, where $0 \leq s \leq \frac{3}{10}$ is specified by the algorithm, then we may assume $\mu_1(A_1) = \mu_3(A_1) = \frac{3}{10}$ since the algorithm must allow any evaluation. Now no player will accept piece A_1.

Algorithms requiring at most three cuts are known. In particular we will now see a class of algorithms for any number of players which yields a three cut solution for three players.

2. The Class of Algorithms $\mathcal{A}$.

We now describe a specific class of algorithms $\mathcal{A}$ for fair division (for interpretation I) which satisfies 1-3. Let $m(n)$ be the minimum number of cuts required for fair division for n players, where the minimum is taken over all algorithms in $\mathcal{A}$ (so $M(n) \leq m(n)$); we will establish a closed form for $m(n)$. The class $\mathcal{A}$ is recursive, breaking the problem for n players into two smaller problems for k_1 and k_2 players where $k_1 + k_2 = n$. It is assumed $m(k_1)$ and $m(k_2)$ are known for $k_i < n$.

Note that any collection of pieces of cake can be thought of as a single piece for cutting purposes. For, if $H = A_1 \cup \cdots \cup A_k$ is a holding of k pieces and $s + t = \mu_i(H)$, then P_i can divide H in the ratio $s : t$ by cutting a piece of size $\mu_i(A_1) + \cdots + \mu_i(A_j) - s$ off the first A_j for which this is positive.

Assume we are given positive integers k_1 and k_2 such that $k_1 + k_2 = n$, and a piece (or holding) A of cake. Our algorithm $D(k_1, k_2)$ will produce two holdings H_1 and H_2 as well as sets of players K_1 and K_2 such that:

$$\left.\begin{array}{l} H_1 \cup H_2 = A, \quad H_1 \cap H_2 = \emptyset, \\ |K_1| = k_1, \quad |K_2| = k_2, \quad K_1 \cap K_2 = \emptyset, \\ \mu_i(H_1) \geq (k_1/n)\mu_i(A) \text{ if } P_i \in K_1, \text{ and} \\ \mu_i(H_2) \geq (k_2/n)\mu_i(A) \text{ if } P_i \in K_2. \end{array}\right\} \qquad (*)$$

The algorithm is based on the idea of having $n-1$ of the players make parallel cuts in the cake dividing it in the ratio k_1: k_2, and letting the remaining player decide whether H_1 is to be the union of the first k_1 or the first $k_1 - 1$ pieces. Indeed, in the worst case our algorithm is no better than this. But its average performance is better because we can often identify players who do not need to make a cut.

At a typical stage, A will be of the form $A_1 \cup \cdots \cup A_j \cup X \cup Y \cup B_k \cup \cdots \cup B_1$, where the last cut was made between X and Y. Let $H_1 = A_1 \cup \cdots \cup A_j \cup X$ and $H_2 = B_1 \cup \cdots \cup B_k \cup Y$. We will say that P_i *accepts* H_1 if $\mu_i(H_1) \geq (k_1/n)\mu_i(A)$, that P_i is *stable* on H_1 if P_i accepts $A_1 \cup \cdots \cup A_j$ (in the same sense), and that P_i *initials* X if P_i accepts H_1 but is not stable on H_1. We say that H_1 is *deficient* if fewer than k_1 players accept H_1. Identical terminology is used for H_2. Not both H_1 and H_2 can be deficient at any stage since each player must accept either H_1 or H_2.

Consider a partition of A into two holdings H_1 and H_2 satisfying:

I_1. $H_1 = A_1 \cup \cdots \cup A_j \cup X, H_2 = B_1 \cup \cdots \cup B_k \cup Y$.
I_2. At least one player initials both X and Y.
I_3. Only s_1 players are stable on H_1 with $s_1 < k_1$.
Only s_2 players are stable on H_2 with $s_2 < k_2$.

I_4. One holding is deficient.

I_5. If H_1 is deficient some player initials Y but not X, if H_2 is deficient some player initials X but not Y.

Lemma.

(a) A partition that satisfies $I_1 - I_4$ also satisfies I_5.

(b) A partition that satisfies $I_1 - I_3$ but not I_4 yields a solution to (*).

Proof:

(a) Assume H_1 is deficient but all t players who initial Y also initial X. Let x be the number of players who initial X but not Y. Then $s_1 + s_2 + t + x = n = k_1 + k_2$ so $s_1 + t + x = k_1 + (k_2 - s_2) > k_1$ which contradicts the assumption that H_1 was deficient.

(b) Assume n_1 players accept H_1, n_2 accept H_2, r players accept both H_1 and H_2, $n_1 \geq k_1$ and $n_2 \geq k_2$. Assign the $n_1 - r$ players accepting only H_1 as well as $k_1 - (n_1 - r)$ accepting both holdings to set K_1. Assign the other players to K_2.

∎

The Deficit Reduction Algorithm $D(k_1, k_2)$

Initial Step: With A, k_1 and k_2 given, ask P_1 to cut $A, A = X \cup Y$ so that $\mu_1(X) = (k_1/n)\mu_1(A)$ and $\mu_1(Y) = (k_2/n)\mu_1(A)$. Form holdings $H_1 = X, H_2 = Y$. Note that $I_1 - I_3$ hold. I_1 is obvious, P_1 initials both X and Y so I_2 holds, and no player is stable yet so I_3 holds.

Iterative Step: Suppose that $I_1 - I_3$ hold. If neither holding is deficient then stop; else assume H_1 is deficient and player P_y initials Y but not X. (P_y exists by part (a) of the Lemma.) Instruct P_y to cut Y in pieces Y_1 and Y_2 such that $\mu_y(A_1 \cup \cdots \cup A_j \cup X \cup Y_1) = (k_1/n)\mu_y(A)$ and $\mu_y(B_1 \cup \cdots \cup B_k \cup Y_2) = (k_2/n)\mu_y(A)$. Relabel X as A_{j+1}, Y_1 as X, and Y_2 as Y. Redefine $H_1 = A_1 \cup \cdots \cup A_{j+1} \cup X, H_2 = B_1 \cup \cdots \cup B_k \cup Y$.

Theorem 1. *In at most $n - 1$ cuts $D(k_1, k_2)$ produces holdings and sets satisfying (*).*

Proof: To prove that the algorithm is well defined, we must check that $I_1 - I_3$ hold after every iterative step. I_1 is obvious, as is I_2 because P_y initials both X and Y. To see that I_3 holds, assuming H_1 was previously deficient, we see that s_2 does not change since $B_1 \cup \cdots \cup B_k$ does not change. The new stable set on H_1 is increased by the players who initialed X at the previous stage. Thus s_1 remains smaller than k_1 because the previous H_1 was deficient. Thus, the algorithm continues until $I_1 - I_3$ hold but I_4 doesn't, when it yields a solution to (*) by part (b) of the Lemma.

It only remains to show the algorithm stops after $n-1$ or fewer cuts. We see at each stage after the first, a new player becomes stable, namely the person at the previous stage who initialed both X and Y. After $n-1$ cuts we are sure of $n-2$ stable players. If H_1 was the deficient holding before this last cut we know $s_1 \leq k_1 - 2$ then, otherwise H_1 would not have been deficient (since at least one player initials X by I_2.) Since at that stage there were at least $n-3 = (k_1-2)+(k_2-1)$ stable players it follows from I_3 that $s_1 = k_1 - 2$ while $s_2 = k_2 - 1$. Adding the new stable player on H_1 after the $(n-1)st$ cut we have $s_1 \geq k_1 - 1$ and $s_2 \geq k_2 - 1$. There are now at most two nonstable players including the last cutter who accepts both H_1 and H_2. If the remaining player accepts H_i, let K_i comprise that player and the $k_i - 1$ players who are stable on H_i. The other set is the cutter and the players stable on the other holding. This completes the proof. ∎

Theorem 2. *For any fixed values of k_1 and $k_2 (k_1 + k_2 = n)$, $D(k_1, k_2)$ may require all $n-1$ cuts.*

Proof: Assume $k_1 \leq k_2$. Let P_i denote the cutter in the *ith* step. The following situation may occur.

After the *ith* step $(1 \leq i \leq k_2 - 1)$, P_i accepts H_1 and $H_2, P_1 \cdots, P_{i-1}$ are stable on H_2, the remaining $n-i$ players accept H_1 but not H_2, so H_2 is deficient.

After the *ith* step $(k_2 \leq i \leq n-2)$, P_i accepts H_1 and $H_2, P_1 \cdots, P_{k_2-1}$ are stable on $H_2, P_{k_2}, \ldots, P_{i-1}$ are stable on H_1, the remaining $n-i$ players accept H_2 but not H_1, so H_1 is deficient.

Thus we see that the $(n-1)st$ cut may be required. ∎

Notice that $D(1,1)$ is "one cuts the other chooses," so $m(2) = 1$. For larger n, $D(k_1, k_2)$ produces sets K_1 and K_2 whose players can perform $D(s_i, t_i)$ on H_i where s_i and t_i are positive integers satisfying $s_i + t_i = k_i$. Repeated applications produce a fair division for the n players. To compute $m(3)$ perform $D(1,2)$ (the only choice) arriving at sets $K_1, K_2, |K_1| = 1, |K_2| = 2$ after two cuts. Thus $m(3) = 2 + m(1) + m(2) = 3 = M(3)$.

Theorem 3. *The sequence $m(n)$ is generated from $m(1) = 0$ by the recursion $m(n) = (n-1) + m(k_1) + m(k_2)$, where $k_1 = \lfloor \frac{1}{2}n \rfloor$ and $k_2 = \lceil \frac{1}{2}n \rceil$*

Proof: We have seen that $m(1) = 0, m(2) = 1$ and $m(3) = 3$, so that the sequence $m(1), m(2), m(3)$ is convex. It is evident that if the sequence $m(1), \cdots, m(n-1)$ is convex, then the choice of k_1 and k_2 in the theorem minimizes $m(k_1) + m(k_2)$ over all choices such that $k_1 + k_2 = n$. This choice also preserves the convexity of m, since

$$m(n) - m(n-1) = 1 + m(\lceil \frac{1}{2}n \rceil) - m(\lceil \frac{1}{2}n \rceil - 1),$$

and this is (weakly) increasing with n by the convexity of $m(1), \cdots, m(n-1)$. ∎

We now can give the closed form for the sequence $m(n)$.

Theorem 4. *The minimal number of cuts required in the class A of algorithms for fair division is* $m(n) = n(\lfloor log_2 n \rfloor + 1) - 2^{\lfloor log_2 n \rfloor + 1} + 1$.

This follows easily from Theorem 3 using induction on n. ∎

3. Choice versus Assignment of Holdings - An Application of Matching in Graphs.

Three versions of "fair" given earlier can be interpreted as matching problems in graphs. With n players, an algorithm produces n holdings which are to be distributed, one holding to each player. If we let V_1 be the set of players and V_2 the set of holdings to be distributed, we can define three bipartite graphs with vertex set $V_1 \cup V_2$ whose only edges are given by:

G_I: Join vertex $P_i \in V_1$ to holding $H_j \in V_2$ if and only if $\mu_i(H_j) \geq \frac{1}{n}$.

G_{II}: Join vertex $P_i \in V_1$ to holding $H_j \in V_2$ if and only if $\mu_i(H_j) > \frac{1}{n}$.

G_{III}: Join vertex $P_i \in V_1$ to holding $H_j \in V_2$ if and only if $\mu_i(H_j) \geq \mu_i(H_k)$ for all $k \neq j$.

The algorithm has produced a solution for the particular interpretation of "fair" if and only if it has assigned a matching in the corresponding graph. But in G_I and G_{II} some edges could be preferred by the players to the ones assigned by the algorithm since they would represent a larger holding than another acceptable holding. To what extent can choices be allowed and a matching still exist? Simple examples show that solutions to problem I do not always produce solutions to problem III. However in any solution to problem I some choosing of holdings is always possible. In particular an ordering of the players exists so that the players can choose acceptable holdings in that order subject to the restriction that no one selects a previously chosen holding. The following theorem formalizes the result.

Theorem 5. *If an algorithm produces holdings which can be assigned to n players so that fair division is thereby accomplished under interpretation I above, then the players and holdings can be ordered so that*

$$\mu_i(H_i) = max\{\mu_i(H_j) : j \geq i\} \geq \frac{1}{n} \text{for } i = 1, 2, \ldots, n.$$

Proof: In order to have a solution to problem I for two players, G_I must be either $K_{2,2}$, $K_{2,2}$ less one edge, or $K_2 \cup K_2$. In any case the result holds. Assuming the result for $2, 3, \cdots, n-1$ players look at a graph G_I for n players for which a V_1 matching accomplishes a fair division. For a subset $S \subset V_1$ of players let $R(S) = \{H_j \in V_2 : H_j \text{ is adjacent to some } P_i \in S\}$. If there exists a non-empty proper subset S for which $|S| = |R(S)|$ then any V_1 matching must assign players in S to holdings in $R(S)$. In this case the result follows by letting players in S choose in the order guaranteed on the smaller problem by the induction hypothesis followed by the players in $V_1 - S$ choosing on holdings $V_2 - R(S)$ in the order guaranteed by the induction hypothesis.

On the other hand if $|R(S)| > |S|$ for all $\emptyset \neq S \subset V_1$, let any player, say P_1, choose his largest holding, say H_1. Then the bipartite subgraph $(V_1 - \{P_1\}, V_2 - \{H_1\}) = G'$ satifies $|R(S)| \geq |S|$ for all $S \subset V_1 - P_1$ so by Hall's Theorem a $V_1 - \{P_1\}$ matching exists in G'. Applying the induction hypothesis to G' completes the proof. ∎

Suppose that by this process of choosing, members of S receive holdings $R'(S) \subset R(S)$. Then the solution is stable in the sense that for any S, not every player in S would prefer a different holding in $R'(S)$ to the one received. We finally note further that standard optimization techniques provide matchings maximizing satisfaction in the sense that $\sum_1^n \mu_i(H_i)$ is maximal.

Finally note the relation that Theorem 5 gives between conditions I and III. Any solution for I gives a partition $X = S_1 \cup S_2 \cup \cdots \cup S_n$ so that for all i, $\mu_i(S_i) \geq \frac{1}{n}$, and $\mu_i(S_i) \geq \mu_i(S_j)$ for $j \geq i$. Condition III requires that for all i, $\mu_i(S_i) \geq \mu_i(S_j)$ *for all* j.

Acknowledgement

The authors appreciate the considerable improvement made by the referee.

References

1. N. ALON, *Splitting necklaces*, Advances in Mathematics **63** (1987), 247–253.
2. L.E. DUBINS AND E.H. SPANIER, *How to cut a cake fairly*, Amer. Math Monthly, **68** (1961), 1–17.
3. K. REBMAN, *How to get (at least) a fair share of the cake*, in "Mathematical Plums", Dolciani Math Ep MAA 4 (1979), pp. 22–37.
4. T.L. SAATY, *Optimization in Integers and Related Extremal Problems*, McGraw-Hill, New York, (1970).
5. H . STEINHAUS, *The problem of fair division*, Econometrica **16** (1948), 101–104.
6. W. STROMQUIST, *How to cut a cake fairly*, Amer. Math. Monthly **87** (1980), 640–644.
7. D.R. WOODALL, *Dividing a cake fairly*, J. of Math Anal and App. **78** (1980), 233–247.
8. D . R. WOODALL, *A Note on the cake-division problem*, J. of Comb. Theory, Serie A **42** (1986), 300–301.