Lecture Notes on Cake Cutting & Fair Division, CS70

The cake-cutting problem:
  We have a cake, and n people who want to split it amongst themselves.
  However, each person might value different portions of the cake
  differently. (I like flowers; you hate them. I hate icing; you
  prefer it.) What's worse, we don't trust each other! What can we
  do?

A protocol is *fair* for X if the following property is true:
If X follows the party, then X gets at least 1/n-th of the cake
(by X's measure), no matter how the other parties behave.
The protocol is *fair* if it is fair for all parties.

For n=2: cut-and-choose protocol
1. Alice cuts the cake into two equal pieces (equal by her measure)
2. Bob chooses whichever piece looks larger (by his measure)
3. Alice takes the remaining one
Theorem: Cut-and-choose is fair.
Proof: If Alice follows the protocol, she gets exactly 1/2 (by her
measure), no matter how Bob behaves.
Next Let's consider Bob. When it is time for him to choose,
he sees two pieces, one worth W and the other worth 1-W (by his measure).
It is guaranteed that either W >= 1/2 or 1-W >= 1/2 (if W < 1/2 and
1-W < 1/2, then 1 = W+1-W < 1/2 + 1/2 = 1, which is impossible).
Therefore, if Bob follows the protocol, he gets at least 1/2 by his
measure. QED

Puzzle: Is there a general algorithm for fair cake-cutting for n>2?

For general n:
   If n=2, use cut-and-choose. Otherwise:

   Let first n-1 people divide the cake using a recursive
   call to this procedure.
   Then the nth person steps in and
   asks each of the first n-1 people to divide her share into
   n equal pieces (by her measure).
   Finally, the nth person goes around and collects the
   largest (in his view) of the n pieces from each of the other people.

   Does each think they've got at least 1/n of the whole pile
   (by their measure)?
   For first n-1 people, yes, since they get
   >= 1/(n-1) after recursive call, divide this into pieces each of
   size >= 1/n(n-1), and then get n-1 of these pieces.
   For last person, yes, since he gets at least x_i/n from the ith
   person, if ith person's whole share is worth x_i. Also,
   x_1 + ... + x_{n-1} = 1, so last person's total share is
   x_1/n + ... + x_{n-1}/n >= 1/n. Therefore:
Theorem: This protocol is fair.
Proof: By induction on n. We've proved the base case (n=2).
The inductive step is exactly what appears in the previous paragraph. QED.

This general protocol can be implemented as a recursive program.
  Q: What's the running time?
  A: n! > 2^n ... which is exponential. yuck.

Q: Can it be done in time polynomial in n?

Puzzle: Find a fair cake-cutting protocol with running time polynomial in n.
One answer: A moving knife.
Move a knife slowly from left to right. When area to the left of the knife covers at least 1/n-th of the cake by your measure, yell "Stop!". First person to yell "Stop!" gets everything to the left of the knife, and we continue.

However, we might prefer to avoid using moving knifes.
Q: Why not use a moving knife?
A: hard to implement in a distributed system without time synchronization

Puzzle: Find a fair cake-cutting protocol with running time polynomial in n, with no use of moving knives.
One solution:
  Take the n-party moving knife algorithm, and translate it into a non-moving-knife protocol by having each person cut at first point they'd yell "Stop!". Take the smallest piece and give it to the person who cut there. Then, continue dividing up the rest of the cake up among the remaining n-1 parties.
Running time: O(n^2) basic operations

A protocol is *envy-free* for X if it has the following property:
if X follows the protocol, then X gets at least as much (by X's measure) as anyone else gets (by X's measure), no matter how the other parties behave.
A protocol is *envy-free* if it is envy-free for all participants.

Q: Is 2-party cut-and-choose envy-free?
A: Yes (do you see why?)

Q: Is the recursive algorithm I gave last time envy-free for n=3?
  (i.e., A+B cut-and-choose, then divide their pieces into 3 sub-pieces; C gets to choose one sub-piece from A & one from B)
A: No.
  (A+B might conspire to give whole cake to A; then C gets 1/3, but A gets 2/3, not envy-free)

Q: Is the 3-party moving knife algorithm envy-free?
  (i.e., move knife slowly to right until someone yells Stop!, repeat)
A: No.
  (suppose A yells Stop! first. then B+C might conspire to give all of remainder to B and none to C, not envy-free)

By the way, this is a non-zero-sum game.
Non-zero sum games often allow everyone to come away with better than 1/n-th of the cake.

An example of "arbitrage" and a non-zero-sum game:

Alice believes that Gore will win the election with probability 5/8. Bob believes that Bush will win the election with probability 3/4.

Assuming that Alice and Bob are both willing to accept any bet that gives them a positive expectation of winning, did you know that there's a way to place bets with both of them so that you can make money for certain?

Here's what you can do. Bet with Alice that you'll pay her $2 if Gore wins and she'll pay you $3 otherwise. Alice agrees because her expectation is: $2(5/8)-$3(3/8)=$1/8.

Bet with Bob that you'll pay him $2 if Bush wins, and he'll pay you $3 otherwise. Bob agrees because his expectation is $2(3/4)-$3(1/4)=$3/4.

Alice and Bob both believe they have positive expectation, but you will win for certain: if either Bush or Gore wins, you will net a dollar!

Credits: http://www.math.hmc.edu/funfacts/ffiles/30003.6.shtml

An envy-free cake-cutting protocol for n=3 people:
http://www.math.hmc.edu/funfacts/ffiles/30001.4-8.shtml

Problem: polynomial-time envy-free cake-cutting protocol for all n
(This might be open?)