Fair-Cake-Cutting

Roi Sibony

March 2025

1 Question 2

We are using the Cut-and-Choose method, and we have precise information about our opponents evaluation. A strategic manipulation Algorithm works as follows:

Algorithm 1 Manipulation Algorithm

procedure MANIPULATION($A, B, Object, n$) ▷ $O(n * log(n)$
  let A be the Evaluation function to Hero and B for the Villan (Opponent)
  $Arr \leftarrow Cut - To - Pieces(Object, B, n)$ ▷ Cut the object to n pieces equal let n be big number
  Sort Arr using B as key Desc tie Breaker using "A" ACS
  $counter \leftarrow 0$ ▷ Initialize counter to 0
  $max \leftarrow B(Object)$
  for $i \in Arr$ **do**
    if $A(i) \leq B(i)$ **then**
      $B \leftarrow B \cup \{i\}$
      $counter \leftarrow counter + B(i)$
    else
      $A \leftarrow A \cup \{i\}$
    end if
    $Arr \leftarrow Arr \setminus \{i\}$
    if $max/2 < counter$ **then**
      break
    end if
  end for
  $A \leftarrow A \cup \{Arr\}$ ▷ We put what's left in A
  return A,B ▷ each Bucket would contain what's where
end procedure

procedure CUT TO PIECES(Object,B,n) ▷ *O(n*log(n)*
  Cut to n equal pieces to B ▷ Even paz algorithm with randomized ties breakers
end procedure

procedure EXPONENTIAL-SEARCH($Object, A, B$) ▷ $O(n * log^2(n))$
  $Threashold \leftarrow 10^7$ ▷ Could be bigger Depends on Resources
  for int n = 1; n $< Threshold$; $n* = 2$ **do** ▷ $log(n)$ *iterations*
    $B_A, B_B \leftarrow Manipulation(Object, A, B, n)$ ▷ tuple unpack to A and B buckets
    if $A(B_A) > 0.5$ & $B(B_B) > 0.5$ **then** ▷ We verify that we got it right and that n is big enough
      return B,A
    end if
  end for
end procedure

This is a greedy algorithm that is designed to maximize the amount of value $Player_1$ accumelate into one piece, while giving to his opponent more than 50% in the other piece. Manipulation procedure requires

a big enough n, We order the arr by B DESC so the most valuable pieces to B would be in the start, the tie breaker is "A ASC" Because we prefer to lose as much less value while giving the opponent as much value so he won't need to take Valuable Pieces of mine.
We split the pieces using B Evalution function in order to give the minimum to $Player_2$ Incase $A \cap B \neq \phi$ to maximize further A final Piece.
In order to find a Big enough "n", we do *Exponential-Search* that operates in *O(log(n))* there's also a threshold for n if we can't waste too much time on that search.

Example: Given the following Object [1|1|1|2|2|2] that we would want to split between two players ($Player_1$ cuts and $Player_2$ chooses.

1. **Iteration-1:** We start from Exponential-Search(Object,A,B)
   The for loop starts from 1 then n would be equal to 1.
   then we call $Manipulation(Object, A, B, n(1))$
   $Arr \leftarrow single - piece$
   there's nothing to order
   it would be worth the same then we enter to the first if
   $B \leftarrow B \cup single - Piece$
   for loops end return A,B
   We then Check if B_A, B_B both worth more than half and we will see that A is not then we would increment to the next iteration.

2. **Iteration-2** n == 2
   $Manipulation(Object, A, B, n(2))$
   Cut-equivalent-Pieces $Arr \leftarrow [Piece_1, Piece_2]$; $Piece_1$ would be [1,1,1,2] and $Piece_2$ would be [2,2], used randomized tie breaker
   Sort lead to $Arr \leftarrow [Piece_1, Piece_2]$
   Sort $Arr \leftarrow [Piece_2, Piece_1]$
   $Arr \leftarrow [Piece_1, Piece_2]$ // $Piece_1$ would be the one that A evaluate 100%
   first iteration $Piece_2$ goes to B, and immediately breaks the for loops since B has more than 50% (66%)
   $A \leftarrow A \cup \{Piece_1\}$
   return $A(Piece_1), B(Piece_2)$
   Both of them above 50% then we could return the answer(from *Exponential-Search*).
   We also got some of the value of B as Bonus :)

1.1 b

The risk of engaging in manipulation based on imprecise estimates of your opponents valuation is that you might without noticing(As a cause of the imprecise information) give them the opportunity to secure a portion that represents more than half of your total value. In order to manipulate your opponent you need to present him a good piece that he would always want, and in the same time try to maximize your value in the other piece which he should want, thus risking more than 50%. If we don't have opponent precise valuation we could create "golden piece" that would be worth more than 50% for the opponent and for myself, which he would pick and and leave me with the less valuable piece.

1.2 c

Consider using the Even–Paz algorithm with a total of 8 participants (you plus 7 others). In order to be sure of doing a safe manipulation, we need to know atleast 4 participants Evaluation Function to manipulate.

First I would explain why 3 doesn't work (Smaller number would be trivial if we prove that 3 doesn't work), given the following [1|0|2|0|3|0|4], where there's 0 every place there is a person and the number 1, 2, 3, 4 would be the exact half of ours. Let address each Case:

1. **Case 1,4:** If we would try to manipulate(1) $W.L.O.G$ and move right $W.L.O.G$ (meaning that we would gain to the right half of ours more value) and decrease the other half, the opponents could put all of the remaining players on the left(4 left) thus excluding us from that half.

2. **Case 2,3:** If we were to try to manipulate(2) $W.L.O.G$ and move right $W.L.O.G$, the opponents would put everything on the far left, and thus exclude us from the left half (and left with the less valuable piece for further cuts)

Let us examine the case where we know the Evaluation Function of 4 people. Like before we would look at all 1..5 places we our exact half can be and 0's would be the opponents [1|0|2|0|3|0|4|0|5]

1. **1,5** $W.L.O.G$ I would choose 1 but the same goes for 5. if our half is in the 1 area, we could move our exact location to the most left zero (the one close to 1) minus some epsilon, thus for the following possibility get more value, if all the other opponents would put in the left of us, we would gain that distance(1,0) more than we would normally do. If they place not only on our left, it is trivial that we would gain more than half because the median would further move to the right (gain more value).

2. **2,3,4** We cannot manipulate at these positions because our unknown opponents could steal our piece. then we would just pick our exact half.

As you can see, if we get information on the right opponents (meaning that we get 1,5) we could always improve our valuation, if we get put into the 2..4 positions we would perform as we would normally do pick our exact half.