Obsolete Results

Alek Westover

March 2, 2020

These are results that I typed up and then kind of decided were useless ish. For example, there is a proposition in this document that is a special case of a more general lemma; once I wrote the proof of that lemma this proof became redundant. But I dunno, I think its nice to save the work. Yeah, nothing is really ever gone with GitHub, but if there is ever a major thing that you know you need to cut from a writeup, you can store it here and rest assured that it's not gone :). **Note that these results are not guaranteed to be: complete, quality writing, useful, etc. In fact, they almost certainly fail multiple of these categories.**

In the ***variable-processor cup game*** the filler is allowed to change p, the amount of water that the filler can add and the number of cups from which the emptier can remove water. Apriori this offers neither the filler nor the emptier a clear advantage: lower values of p mean that the emptier is at more of a discretization disadvantage but also mean that the filler can anchor fewer cups. We hoped that the variable-processor cup game could be simulated in the vanilla multiprocessor cup game, because the extra ability given to the filler does not seem very strong. The new version of the cup game arose as we tried to get a bound of $\Omega(\log p)$ backlog in the multiprocessor game against an off-line filler, which would combine with previous results to give us a lower bound that matches our upper bound: $O(\log\log n+\log p)$. This new version seemed promising in this respect because of the following Proposition:

However, attempts at simulating the variable-processor cup game are futile because of the following results, which show that the variable-processor cup game is–surprisingly–fundamentally different from the multiprocessor cup game, and thus impossible to simulate:

Proposition 1. *In the variable-processor cup game on n cups against an online filler, the filler can force backlog to be $\Omega(\log^2 n)$.*

In fact Proposition **??** gives a weak lowerbound on backlog; the filler can achieve much higher fill, as high as $\text{poly}(n)$, which follows as a corollary of the ***Amplification Lemma:***

Proof of Proposition **??**. If at any point in the process that will be described backlog is greater than $\frac{1}{8}\log^2 n$ then the filler instantly stops whatever it is doing, and just maintains this backlog. Henceforth we do not consider this case where the filler finishes achieving the backlog before the end of our procedure.

The main idea of this analysis is to use a subroutine for achieving fill $\frac{1}{2}\log n/2^l$ on half of the ***active*** cups (of which there are $n/2^l$), and then halving the number of processors, and the number of cups that we focus on, and recursing on this set. By recursing $\log n$ times we achieve backlog $\frac{1}{2}(\log n/2+\log n/4+\log n/8+\cdots)\geq\Omega(\log^2 n)$.

The key technical challenge is how to achieve fill at least $\frac{1}{2}\log n/2^l$ at each level of recursion, becasue ***negative fill***–fill below the average fill of the active cups–is allowed. Note that it is strictly easier for the filler to achieve large fill if negative fill is not allowed (i.e. cups can zero out); we need to allow for negative fill because we are recursing.

Let $h_l=\frac{1}{2}\log n/2^l$; we will achieve fill at least $h_l/4$ in $n_l=n/2^l$ cups at the l-th level of recursion. Let a cup be designated ***bad*** if it has fill less than $-h_l$, ***good*** if it has fill at least $\frac{1}{4}h_l$ and ***fine*** if it has fill in $(-\frac{1}{4}h_l,\frac{1}{4}h_l)$. Let b be the number of bad cups, g the number of good cups, and f the number of fine cups.

If $g\geq n_l/4$ then we simply recurse on the good cups.

If $f\geq n_l/4$. Which is fine. So we do the following, with the assurance that the cups are pretty much fine basically.

We now outline the filler's strategy to achieve $n_l/2$ cups with fill h_l.

To accomplish this, the filler anchors the $n_l/2$ fullest cups, sets the number of processors to be $n_l/2+1$ and then repeats the following algorithm for each anchored cup i:
If the fill in anchor cup i is alreday at least h_l, then we do nothing.

At every step the filler places 1 fill in each anchored cup. For $n_l \log^2 n+1$ rounds the filler plays a single processor cup game on the $n_l/2$ non-anchored cups. If among these games the emptier always neglects at least 1 anchor cup at least once, then the average fill of the anchor cups will have increased by $\Omega(\log^2 n)$, hence we have the desired backlog in an anchor cup.

If on the other hand there exists a game where the emptier always empties a unit of water from each of the $n_l/2$ anchor cups, then the emptier and the filler are genuinely engaged in a single processor cup game on the remaining cups. By a well known construction the filler can achieve backlog at least $\log n_l/2$ in the single processor cup game on $n_l/2$ cups.

Upon achieving such backlog the filler will swap this cup into the anchor set. At the conclusion of this process each anchor cup will have fill at least $\log n_l/2$ greater than the average fill of the non-anchor cups.

However, this does not quite imply that the anchor cups have increased in fill by $\log n_l/2$ from the start of the process, as the average fill in the non-anchor cups has been depleted as water is siphoned off into the anchor cups. In particular, if the average fill of the non-anchor cups ends up as μ', started as μ, and the average fill has increased by $\epsilon \geq 0$ from μ ($\epsilon > 0$ could occur if cups zero out, which as remarked upon makes it strictly easier to achieve high fill), then we have $\mu' + (\log n_l/2)/2 \leq \mu + \epsilon$. If μ' is at least $(\log n_l/2)/2$ below the new average, then the average fill among anchor cups must be at least $(\log n_l/2)/2$ above this new average fill, which is at least the average fill. Hence by the end of this process all anchor cups are at least $(\log n_l/2)/2$ above the average fill at the beginning of the process.

After applying this process the filler cuts the number of processors in half and recurses on the anchor cups. Note that in recursion fill is relative to the new average fill of the set that we recurse on. The recursion depth possible is $\log n$, so this strategy gives backlog at least $\frac{1}{2}(\log n/2 + \log n/4 + \log n/8 + \cdots) \geq \Omega(\log^2 n)$ as desired.

And finally, if $g < n_l/4$ and $f < n_l/4$, then as $g+b+f = n_l$, we have $b > n_l/2$ then the negative fill of the bad cups is at least $\frac{1}{4} b h_l$; this must be offset by some set of cups with positive fill. If there are less than $b/\log n_l$ cups that have positive fill, then the average fill in these cups is $\Omega(\log^2 n)$, so the filler would instantly win. Unfortunately, if say, $n/\sqrt{\log n}$ of the cups hae fill $\log^{3/2} n$ or something then like we're kind of screwed. There are pretty much 2 options at this point. You can recurse on a set of things that have fill at least $\log^{3/2} n$, and there are $n/\sqrt{\log n}$ of them, so you can get like $\log n/\mathrm{loglog} n$ levels of recursion ish, or you can like do smoothing.

□