An Adaptive Filling Strategy for the Multi-Processor Cup Game

Alek Westover

Abstract

We give an adaptive filling strategy for the p-processor cup game on n cups that achieves backlog $\Omega(\log n)$.

1 Introduction

In the p-processor cup game on n cups is a multi-round game with two players, the **filler** and the **emptier**, that take turns adding and removing water from the cups. In particular, on each round the filler distributes p units of fill arbitrarily amongst the cups, subject only to the restriction that the filler cannot place more than 1 unit of fill into any given cup. Next, the emptier chooses p cups and removes at most 1 unit of water from each of these. The **backlog** is the height of the fullest cup; the emptier aims to minimize backlog while the filler aims to maximize backlog.

Kuszmaul showed that in the p-processor cup game on n cups the greedy emptying strategy—i.e. empty from the p fullest cups—never lets backlog exceed $O(\log n)$. Kuszmaul also provided a filling strategy that achieves backlog $\Omega(\log(n-p))$. For small p, e.g. $p \leq n/2$, the upper-bound and lower-bound match, but for large p the bounds do not match.

Many suspected that the $O(\log n)$ upper bound could be reduced to $O(\log(n-p))$ even for large p; however, we show that this is not the case.

2 Preliminaries

Let $H_n = 1/1 + 1/2 + \cdots + 1/n$ denote the n-th harmonic number. It is well known that $H_n = \Theta(\log n)$, and in fact $H_n = \ln n + \Theta(1)$.

The **mass** of the cups is the sum the fills of all of the cups.

3 Lower Bound on Backlog

Theorem 1. *There exists an adaptive filling strategy for the p-processor cup game on n cups that achieves backlog $\Omega(\log n)$ against any emptier.*

Proof. Kuszmaul's construction shows that there is a filling strategy that achieves backlog $\Omega(\log(n-p))$. If $p \leq n - \sqrt{n}$, then $n-p \geq \sqrt{n}$, so $\log(n-p) \geq \frac{1}{2}\log(n)$, hence $\Omega(\log(n-p)) = \Omega(\log(n))$. Thus the result holds for $p \leq n - \sqrt{n}$; we proceed to consider the case where $p > n - \sqrt{n}$.

We give a filling strategy, that we call **IncMass**, with running time $O(p/(n-p))$, that increases the mass of the cups in n rounds by at least $1/2$, provided that backlog starts below $O(\log n)$. The filler's strategy for achieving backlog $\Omega(\log n)$ is to repeatedly use IncMass as many as $\Theta(n \log n)$ times, terminating if backlog $\Omega(\log n)$ is ever achieved (after which the filler changes to the strategy of placing a unit of water in the p fullest cups on each round). The filler's strategy either terminates before finishing, which only happens if it has achieved backlog $\Omega(\log n)$, or does not terminate early, in which case the mass of the cups has increased by $\Omega(n \log n)$, implying that which means that average fill and also backlog have increased by at least $\Omega(\log n)$. We now describe IncMass.

Let S denote the set of cups. The filler will maintain a set $U \subset S$ throughout the algorithm. The algorithm's procedure will ensure that once a cup enters U its fill never decreases for the rest of the process (by placing 1 unit of water in such a cup each round). Furthermore, $|U|$ will increase by $n-p$ at each iteration of the process. U is initialized to $\varnothing$. For each of $\lfloor (p+1)/(n-p) \rfloor$ steps the filler will:

1. Distribute $p - |U|$ water equally among the cups in $S \setminus U$ (thus, each such cup receives $\frac{p-|U|}{n-|U|}$ fill)
2. Distribute $|U|$ water equally among the cups in U (thus, each such cup receives 1 fill)

Then the emptier must chose p cups to empty from, and hence $n-p$ cups to **neglect**, i.e. not empty from. Let N be the set of neglected cups on this step, with $|N| = n-p$. The emptier adds all cups in $N \setminus U$ to U, and then adds the $(n-p) - |N \setminus U|$ fullest cups in $S \setminus U$ to U.

The number of cups in $S \setminus U$ decreases by $n-p$ on of the $\lfloor (p+1)/(n-p) \rfloor$ steps, so in the end we have $|S \setminus U| \geq p+1$.

Claim 1. *Any cup $c \in S \backslash U$ at the end of this process has 0 fill, provided that the backlog started as at most $O(\log n)$.*

Proof. On the i-th step of the filler's process the fill of a cup in $S \setminus U$ increases by $(p-|U|)/(n-|U|)$, and then decreases by 1 on this round, resulting in a net change of

$$-\frac{n-p}{n-(n-p)\cdot i}.$$

The total amount that the fill of c has changed since the start of the filler's process is simply the sum of the amounts that the fill changes on each step, which is

$$\sum_{i=0}^{\left\lfloor \frac{p+1}{n-p} \right\rfloor -1} \frac{n-p}{n-(n-p)\cdot i}. \tag{1}$$

We aim to show that (1) is $\Omega(\log n)$; combined with the fact that the backlog started as $O(\log n)$ this will imply that c has fill 0.

For $p = n-1$ (1) is simply the difference of harmonic numbers; this prompts the idea of lower bounding (1) by a difference of harmonic numbers. We lower bound the i-th term in the sum by:

$$\frac{n-p}{n-i(n-p)} = \sum_{j=0}^{n-p-1} \frac{1}{n-i(n-p)} \geq \sum_{j=n-i(n-p)}^{n-(i-1)(n-p)-1} \frac{1}{j}.$$

Applying this to (1) we get a difference of harmonic numbers:

$$\sum_{i=0}^{\left\lfloor \frac{p+1}{n-p} \right\rfloor -1} \sum_{j=n-i(n-p)}^{n-(i-1)(n-p)-1} \frac{1}{j}.$$

This is now a difference of harmonic numbers. When $i = 0, j = n-(i-1)(n-p)-1$ we get the smallest term in the sum:

$$\frac{1}{n+(n-p-1)}.$$

When $i = \lfloor (p+1)/(n-p) \rfloor - 1, j = n - i(n-p)$ we get the largest term in the sum, which is at least

$$\frac{1}{n-\left(\frac{p+1}{n-p}-1\right)(n-p)} = \frac{1}{2(n-p)-1}.$$

Thus, c has lost fill at least

$$H_{2n-p-1} - H_{2(n-p-1)} \geq \Omega\left(\log \frac{p}{n-p}\right).$$

Because $p > n - \sqrt{n}$ we have that c has lost fill at least

$$\Omega\left(\log \frac{n-\sqrt{n}}{\sqrt{n}}\right) \geq \Omega(\log n),$$

as desired. Because the backlog, and hence fill of c, started less than $O(\log n)$ by assumption, c must now have 0 fill. □

There are now (at least) $p+1$ cups with fill 0. The final step of the filler's procedure is to add $1/(p+1)$ fill to $p+1$ cups with fill 0. The emptier now must waste resources on emptying one of these cups with fill below 1. In total on this round the filler adds p units of fill to the cups, and then the emptier removes at most $p-1+1/(p+1) \leq p-1/2$ water to the cups. Hence the mass increases by at least 1/2. As mass is monotonically increasing, this increase of $\frac{1}{2}$ in mass persists, as desired.

We have shown that IncMass can increase the mass given that backlog starts below $O(\log n)$. As shown previously, by repeatedly using IncMass we can achieve backlog $O(\log n)$.

We remark that the running time of IncMass is $O(p/(n-p))$ which satisfies

$$\sqrt{n} \leq \frac{p}{n-p} \leq n.$$

The running time of the full strategy is

$$O\left(\frac{p}{n-p} n \log n\right) \leq O(n^2 \log n).$$

□

4 Conclusion

Our analysis shows that $\Theta(\log n)$ is a tight bound on backlog in the p-processor cup game on n cups with an adaptive filler.

However, many open questions remain. The bounds on backlog in the Multi-Processor cup game with an oblivious filler have *not* yet been made tight. In particular, there is a lower bound of $\Omega(\log \log n)$, and an upper bound of $O(\log \log n + \log p)$; for $p \leq \log n$ the bounds are tight, but for large p they are not.

Further, analysis of the variant of the game where p can change, potentially with resource augmentation, is also of great interest.