Variable-Processor Cup Games

Alek Westover

April 8, 2020

1 Introduction

The cup game is a classic game in computer science that models work-scheduling. In the cup game a filler and an emptier take turns adding and removing water (i.e. work) to the cups. We investigate a variant of the vanilla multiprocessor cup game which we call the ***variable-processor cup game*** in which the filler is allowed to change the number of processors p (the amount of water that the filler can add and the number of cups from which the emptier can remove water. This is a natural extension of the vanilla multi-processor cup game to when the resources available are variable. Note that although the restriction that the filler and emptier's resources vary together may seem artificial, this is the only way to conduct the analysis; the rationale behind giving the emptier and filler equal resources in the classical vanilla multi-processor cup game is that this is the only way to achieve upper and lower bounds. The equivalent rational holds for the motivation of the variable-processor cup game. Analysis of this game does provide information about how real-world systems will behave.

A priori the fact that the number of processors can vary 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 oblivious filler, which would combine with previous results to give us a lower bound that matches our upper bound: $O(\log\log n + \log p)$. In Proposition 2 we prove that there is an oblivious filling strategy in the variable-processor cup game on n cups that achieve backlog $\Omega(\log n)$ as desired. [1]

However, we also show that attempts at simulating the variable-processor cup game are futile because the variable-processor cup game is–surprisingly–fundamentally different from the multiprocessor cup game, and thus impossible to simulate. This follows as a corollary of an ***Amplification Lemma*** for both the adaptive and oblivious filler. Section 2 and Section 4 follow the structure:

1. Proposition: Base case of inductive argument in corollary
2. Lemma: Amplification Lemma, allows for inductive step in inductive argument
3. Corollary: Repeatedly amplify the base case backlog to get very large backlog

We proceed with our results.

2 Adaptive Lower Bound

Proposition 1. *There exists an adaptive filling strategy for the variable-processor cup game on n cups that achieves backlog at least $\frac{1}{4}\ln(n/2)$, where fill is relative to the average fill of the cups, with negative fill allowed.*

Proof. Let $h = \frac{1}{4}\ln(n/2)$ be the desired fill. Once a cup with fill at least h is achieved the filler stops, the process completed. Denote the fill of a cup i by $\text{fill}(i)$. Let the ***positive tilt*** of a cup i be $\text{tilt}_+(i) = \max(0, \text{fill}(i))$, and let the positive tilt of a set S of cups be $\sum_{i\in S}\text{tilt}_+(i)$. Let the ***mass*** of a set of cups S be $m(S) = \sum_{i\in S}\text{fill}(i)$. Denote the average fill of a set of cups S as $\mu(S) = m(S)/|S|$. Let A consist of the $n/2$ fullest cups, and B consist of the rest of the cups.

[1] Note that we have $\Omega(\log n)$ in this proposition instead of $\Omega(\log p)$ because the filler can increase the number of processors, so it increases the number of processors to $n-1$ to start.

A nearly identical construction could be used to show that backlog $\Omega(\log p_{\max})$ can be achieved, where the number of processors starts at $p_{\max}$ and the filler does not ever increase the number of processors. However, using $p_{\max} = n$ is natural in the variable-processor cup game, so we do not consider the game with the restriction that the filler can not increase the number of processors above some $p_{\max} < n$.

If the process is not yet complete, that is $\text{fill}(i) < h$ for all cups i, then $\text{tilt}_+(A \cup B) < h \cdot n$. Assume for sake of contradiction that there are more than $n/2$ cups i with $\text{fill}(i) \leq -2h$. The mass of those cups would be less than $-hn$, but there isn't enough positive tilt to oppose this, a contradiction. Hence there are at most $n/2$ cups i with $\text{fill}(i) \leq -2h$.

We set the number of processors equal to 1 and play a single processor cup game on $n/2$ cups that have fill at least $-2h$ (which must exist) for $n/2-1$ steps. We initialize our "active set" to be A, noting that $\text{fill}(i) \geq -2h$ for all cups $i \in A$, and remove 1 cup from the active set at each step. At each step the filler distributes water equally among the cups in its active set. Then, the emptier will choose some cup to empty from. If this cup is in the active set the filler removes it from the active set. Otherwise, the filler chooses an arbitrary cup to remove from the active set.

After $n/2-1$ steps the active set will consist of a single cup. This cup's fill has increased by $1/(n/2) + 1/(n/2-1) + \cdots + 1/2 + 1/1 \geq \ln n/2 = 4h$. Thus such a cup has fill at least $2h$ now, so the proposition is satisfied. □

Lemma 1 (The Adaptive Amplification Lemma)**.** *Let f be an adaptive filling strategy that achieves backlog $f(n)$ in the variable-processor cup game on n cups (relative to average fill, with negative fill allowed). Let $n_0 \in \mathbb{N}$ be a constant such that $\frac{1}{4}\ln(n_0/2) \geq 1$. Let $\delta \in (0,1)$ be constant, and let $M \in \mathbb{N}$ be a constant such that $n_0 \leq (1-\delta)\delta^M n \leq n_0/\delta$.*

Then, there exists an adaptive filling strategy that achieves backlog

$$f'(n) \geq \max\left(1, (1-\delta)\sum_{\ell=0}^{M} f((1-\delta)\delta^\ell n)\right)$$

in the variable processor cup game on $n \geq n_0$ cups.

Proof. The basic idea of this analysis is as follows:

1. Using f repeatedly, achieve average fill at least $(1-\delta)f(n(1-\delta))$ in a set of $n\delta$ cups.

2. Reduce the number of processors to $n\delta$.

3. Recurse on the $n\delta$ cups with high average fill.

Let A the ***anchor set*** be initialized to consist of the $n\delta$ fullest cups, and let B the ***non-anchor set*** be initialized to consist of the rest of the cups (so $|B| = (1-\delta)n$). Let $n_\ell = n\delta^{\ell-1}$, $h_\ell = (1-\delta)f(n_\ell(1-\delta))$; the filler will achieve a set of at least $n_\ell\delta$ cups with average fill at least h_ℓ on the ℓ-th level of recursion. On the ℓ-th level of recursion $|A| = \delta \cdot n_\ell, |B| = (1-\delta) \cdot n_\ell$.

We now elaborate on how to achieve Step 1. Our filling strategy always places 1 unit of water in each anchor cup. This ensures that average fill in the anchor set is non-decreasing.

On the ℓ-th level of recursion we will repeatedly apply f to B, and then take the cup generated by f within B to have large backlog and swap it with a cup in A until A has the desired average fill. Note that

$$\mu(A) \cdot |A| + \mu(B) \cdot |B| = 0$$

so

$$\mu(A) = -\mu(B) \cdot (1-\delta)/\delta.$$

Thus, if at any point in this process B has average fill lower than $-h_\ell \cdot \delta/(1-\delta)$, then anchor set has average fill at least h_ℓ, so the process is finished. So long as B has average fill at least $-h_\ell \cdot \delta/(1-\delta)$ we will apply f to B.

It is somewhat complicated to apply f to B however, because we need to guarantee that in the steps that the algorithm takes while applying f the emptier always empties the same amount of water from B as the filler fills B with. This might not be the case if the emptier does not empty from each anchor cup at each step. Say that the emptier ***neglects*** the anchor set on an application of f if there is some step during the application of f in which the emptier does not empty from some anchor cup.

We will apply f to B at most $h_\ell n_\ell \delta + 1$ times, and at the end of an application of f we only swap the generated cup into A if the emptier has not neglected the anchor set during the application of f.

Note that each time the emptier neglects the anchor set the mass of the anchor set increases by 1. If the emptier neglects the anchor set $h_\ell n_\ell \delta + 1$ times, then the average fill in the anchor set increases by more than h_ℓ, so the desired average fill is achieved in the anchor set.

Otherwise, there must have been an application of f for which the emptier did not neglect the anchor set. We only swap a cup into the anchor set if this is the case. In this case we achieve fill

$$-h_\ell \cdot \delta/(1-\delta) + f(n_\ell(1-\delta)) = (1-\delta)f(n_\ell(1-\delta)) = h_\ell$$

in a non-anchor cup, and swap it with the smallest cup in the anchor set.

We achieve average fill h_ℓ in the anchor set for M levels of recursion. Note that as $n \geq n_0$ we can always simply use Proposition 1 to achieve backlog 1. We will revert to this option if it gives larger fill than we get by repeatedly applying f. □

Corollary 1. *There is an adaptive filling strategy for the variable-processor cup game on n cups that achieves backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon \in (0,1)$, in running time $2^{O(\log^2 n)}$.*

Proof.

Basic Idea. Let

$$f_0(k) = \begin{cases} \lg k, & k \geq 1, \\ 0 & \text{else.} \end{cases}$$

Note that we can achieve backlog $f_0(k)$ on k cups by Proposition 1. Let f_{m+1} be the result of applying the Amplification Lemma to f_m with $\delta = 1/2$. The function $f_{\lg n^{1/9}}(k)$ satisfies

$$\text{for } k \geq n,\ f_{\lg n^{1/9}}(k) \geq 2^{\lg n^{1/9}} \lg k. \qquad (1)$$

In particular, using $f_{\lg n^{1/9}}(n)$ (applying the function to all of the cups) we achieve backlog $\Omega(n^{1/9} \lg n) \geq \Omega(\text{poly}(n))$ as desired. To prove Equation 1, we prove the following lower bound for f_m by induction:

$$f_m(k) \geq 2^m \lg k,\ \text{for } k \geq (2^9)^m.$$

The base case follows from the definition of f_0. Assuming the property for f_m, we get the following by Lemma 1: for $k > (2^9)^{m+1}$,

$$\begin{aligned}
&f_{m+1}(k) \\
&\geq \frac{1}{2}(f_m(k/2) + f_m(k/4) + \cdots + f_m(k/2^9) + \cdots) \\
&\geq \frac{1}{2}(f_m(k/2) + f_m(k/4) + \cdots + f_m(k/2^9)) \\
&\geq \frac{1}{2}2^m(\lg(k/2) + \lg(k/4) + \cdots + \lg(k/2^9)) \\
&\geq \frac{1}{2}2^m(9\lg(k) - \frac{9 \cdot 10}{2}) \\
&\geq 2^{m+1}\lg(k),
\end{aligned}$$

as desired. Hence the inductive claim holds, which establishes that $f_{\lg n^{1/9}}$ satisfies the desired condition, which proves that backlog can be made $\Omega(\text{poly}(n))$.

Running Time Analysis. The recursive construction requires quite a lot of steps, in fact a superpolynomial number of steps. If we consider the tree that represents computation of $f_{\log n^{1/\alpha}}(n)$ we see that each node will have at most α (some constant, e.g. $\alpha = 9$, α is the number of terms that we keep in the sum) children (the children of $f_k(c)$ are $f_{k-1}(c/2), f_{k-1}(c/4), \ldots f_{k-1}(c/2^\alpha)$), and the depth of the tree is $\log n^{1/\alpha}$. Say that the running time at the node $f_{\log n^{1/\alpha}}(n)$ is $T(n)$. Then because $f_k(n)$ must call each of $f_{k-1}(n/2^i)$ $n/2^i$ times for $1 \leq i \leq \alpha$, we have that $T(n) \leq \frac{\alpha n}{2}T(n/2)$. This recurrence yields $T(n) \leq \text{poly}(n)^{\log n} = O(2^{\log^2 n})$ for the running time.

Generalizing Our Approach. Generalizing our approach we can achieve a (slightly) better polynomial lower bound on backlog. In our construction the point after which we had a bound for f_m grew further out by a factor of 2^9 each time. Instead of 2^9 we now use 2^α for some $\alpha \in \mathbb{N}$, and can find a better value of α. The value of α dictates how many iterations we can perform: we can perform $\lg n^{1/\alpha}$ iterations. The parameter α also dictates the multiplicative factor that we gain upon going from f_m to f_{m+1}. For $\alpha = 9$ this was 2. In general it turns out to be $\frac{\alpha-1}{4}$. Hence, we can achieve backlog $\Omega\left(\left(\frac{\alpha-1}{4}\right)^{\lg n^{1/\alpha}} \lg n\right)$. This optimizes at $\alpha = 13$, to backlog $\Omega(n^{\frac{\lg 3}{13}} \log n) \approx \Omega(n^{0.122} \log n)$.

We can further improve over this. Note that in the proof that f_{m+1} gains a factor of 2 over f_m given above, we lower bound $9\lg k - 9 \cdot 10/2$ with $2\lg k$. Usually however this is very loose: for small m a significant portion of the $9\lg k$ is annihilated by the constant $1+2+\cdots+9$ (or in general $\alpha \lg k$ and $1+2+\cdots+\alpha$), but for larger values of m because k must be large we can get larger factors between steps, in theory factors arbitrarily close to α. If we could gain a factor of α at each step, then the backlog achievable would be $\Omega(\alpha^{\lg n^{1/\alpha}} \log n) = \Omega(n^{(\lg \alpha)/\alpha} \log n)$ which optimizes (over the naturals) at $\alpha = 3$ to $n^{(\lg 3)/3} \approx n^{0.528}$. However, we can't actually gain a factor of α each time because of the subtracted constant. But, for any $\epsilon > 0$ we can achieve a $\alpha - \epsilon$ factor increase each time (for sufficiently large m). Of course ϵ can't be made arbitrarily small because m can't be made arbitrarily large, and the "cut off" m where we start achieving the $\alpha - \epsilon$ factor increase must be a constant (not dependent on n). When the cutoff m, or equivalently ϵ, is constant then we can achieve backlog $\Omega((\alpha-\epsilon)^{\lg n^{1/\alpha}} \log n) = \Omega(n^{(\lg(\alpha-\epsilon))/\alpha} \log n)$. For instance, with this method we can get backlog $\Omega(\sqrt{n})$ for appropriate ϵ, α choice, or $\tilde{\Omega}(n^{(\lg(3-\epsilon))/3})$ for any constant $\epsilon > 0$.

Existential Improvement. We now (non-constructively) demonstrate the existence of a filling strategy that achieves backlog $cn^{1-\epsilon}$ for constant $\epsilon \in (0,1)$ and $c \ll 1$.

Let $f^*(n)$ be the supremum over all filling strategies of the fill achievable on n cups. Clearly $f^*(n)$

satisfies the Amplification Lemma, i.e.

$$f^*(n) \geq (1-\delta)\sum_{\ell=0}^{M} f^*((1-\delta)\delta^\ell n).$$

Assume for the sake of deriving a contradiction that there is some n such that $f^*(n) < cn^{1-\epsilon}$, let n_* be the minimum such n.

Then we have

$$cn_*^{1-\epsilon} > f^*(n_*) \geq (1-\delta)\sum_{\ell=0}^{M} f^*((1-\delta)\delta^\ell n_*).$$

However,

$$\begin{aligned}
&(1-\delta)\sum_{\ell=0}^{M} f^*((1-\delta)\delta^\ell n_*) \\
&\geq cn_*^{1-\epsilon}(1-\delta)\sum_{\ell=0}^{M}((1-\delta)\delta^\ell)^{1-\epsilon} \\
&\geq cn_*^{1-\epsilon}(1-\delta)\frac{(1-\delta)^{1-\epsilon}}{1-\delta^{1-\epsilon}}.
\end{aligned}$$

We will now show that there is an appropriate choice of $\delta \in (0,1)$ such that

$$\frac{(1-\delta)^{2-\epsilon}}{1-\delta^{1-\epsilon}} \geq 1,$$

which contradicts the assumption that $cn_*^{1-\epsilon} > f^*(n_*)$. Rearranging, we desire

$$(1-\delta)^{2-\epsilon} + \delta^{1-\epsilon} \geq 1.$$

For any ϵ we will show that there is an appropriate choice of $\delta \ll 1$ satisfying this inequality.

Consider the Taylor series for $(1-\delta)^{2-\epsilon}$:

$$(1-\delta)^{2-\epsilon} = 1-(2-\epsilon)\delta - O(\delta^2).$$

By taking δ sufficiently small, the $O(\delta^2)$ term becomes negligible compared to the $(\alpha+1)\delta$ term. In particular, say that the $O(\delta^2)$ term is less than $c\delta^2$ for some constant c. Taking δ small enough such that $\delta^2 c < \delta$, we have that $(1-\delta)^{2-\epsilon} > 1-(2-\epsilon)\delta - \delta$.

So, to find a δ where $g(\delta) \geq 1$ it suffices to find a δ with

$$\delta^{1-\epsilon} \geq (3-\epsilon)\delta.$$

The equality is achieved at $\delta = (\frac{1}{3-\epsilon})^{1/\epsilon}$.

This establishes the existence of a filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$.

Modifying the Existential Argument to achieve backlog $n^{1-\epsilon}$ in finite time. We can modify the existential argument to get a guarantee on how long it will take to achieve the desired backlog. Fix an $\epsilon > 0$, and choose a $\delta \in (0,1)$ satisfying $(1-\delta)^{2-\epsilon}/1-\delta^{1-\epsilon} \geq 1$. Fix $c \ll 1$. Say we aim to achieve backlog at least $cn^{1-\epsilon}$. Note that the choice of δ is motivated by the fact that

$$(1-\delta)\sum((1-\delta)\delta^i)^{1-\epsilon} = \frac{(1-\delta)^{2-\epsilon}}{1-\delta^{1-\epsilon}},$$

and, as in the existential argument it will be useful to assert that this quantity is at least 1. ok I'm kind of worried about things not being integers being a problem. We start with

$$f_0(k) = \begin{cases} \lg k, & k \geq 1, \\ 0 & \text{else.} \end{cases}$$

Then we construct f_n as the amplification of f_{n-1}. We claim the following regarding this construction:

$$f_\ell(k) \geq cn^{1-\epsilon} \text{ for all } k > n/(1-\delta)^\ell.$$

This is clearly true in the base case with f_0. If $f_\ell(k) \geq cn^{1-\epsilon}$ for all k then we are already done. Otherwise, let k_*+1 be the smallest k such that $f_\ell(k) < cn^{1-\epsilon}$. Note that by assumption we have $k_* > n/(1-\delta)^\ell$. Now consider the amplification $f_{\ell+1}$ of f_ℓ.

$$\begin{aligned}
&f_{\ell+1}(k_*/(1-\delta)) \\
&\qquad \geq (1-\delta)\sum f_\ell((1-\delta)\delta^i n) \\
&\qquad \geq cn^{1-\epsilon}\frac{(1-\delta)^{2-\epsilon}}{1-\delta^{1-\epsilon}} \\
&\qquad \geq cn^{1-\epsilon}.
\end{aligned}$$

This is as desired. Thus, by taking $f_{(\log n)/\log(1/(1-\delta))}$ we achieve backlog $cn^{1-\epsilon}$.

Achieving backlog $\Omega(n^{\lg 3/2})$. Recall the recursive procedure that we use in the proof of the Amplification Lemma: to achieve the desired fill we must call $f(n/2^\ell)$ for $\ell = 0,1,2,\ldots$. As f_{m+1} recursively calls f_m, there is even more recursion.

Let $\#(m,i)$ denote the number of times $f_m(n/2^i)$ occurs in the recursive construction. Let there be $M = \lg(n/2)$ levels of recursion. The first level in the tree has $\#(M,i) = 1$ for all i. Note that we have

$$\#(m-1,i) = \sum_{j>i}\#(m,j)$$

for any level m, because any expression $f_m(n/2^j)$ will call $f_{m-1}(n/2^i)$ for $j > i$.

This is very reminiscent of the hockey stick identity:

$$\binom{n}{i} = \sum_{i-1\leq j\leq n-1}\binom{j}{i-1}.$$

In fact we claim that if you look at it right (i.e. sideways) the $\#(m,i)$'s form Pascal's triangle! Specifically the bijection is

$$\#(m,i) = \binom{i}{M-m}.$$

This is true because of the Hockey Stick Identity and the base case like $\#(M,i) = 1$ for all i. We induct on the diagonals of Pascal's triangle. The inductive hypothesis is that $\#(m,i) = \binom{i}{M-m}$ for all i for some m. Then by the Hockey Stick Identity we get

$$\begin{aligned}\#(m-1,i) &= \sum_{j>i} \#(m,j) \\ &= \sum_{j>i} \binom{j}{M-m} = \binom{i}{M-(m-1)}\end{aligned}$$

as desired.

We can also prove this with a simple combinatorial argument: there is a bijection between terms of the form $f_{M-m}(n/2^{m+i-m})$ and integer partitions of $i-m$ into m integers, as you must divide up the array subdivisions among the different levels of recursion. This demonstrates that

$$\#(m,i) = \binom{i-m+m}{M-m} = \binom{i}{M-m}.$$

We know that $f_m(n/2^M) \geq 1$ by design in Lemma 1, so to determine the total backlog we add up the occurrences of $f_m(n/2^M)$ on each level, weighted by the $1/2$ decay factor. Then the backlog we get is

$$\sum_{i=0}^{M} \binom{M}{i} \frac{1}{2^i} = (3/2)^M = n^{\lg(3/2)}.$$

This is optimal for $\delta = 1/2$.

Constructively achieving backlog $\Omega(n^{1-\epsilon})$ The existential proof that backlog $\Omega(n^{1-\epsilon})$ suggests that we will need to take $\delta \ll 1$ to achieve this backlog. The analysis from the case $\delta = 1/2$ doesn't immediately apply here; that analysis was significantly simplified by the fact that $\delta = 1-\delta$ for $\delta = 1/2$. However, we use some similar ideas.

□

3 Adaptive Upper Bounds

Let $[n] = \{1, 2, \dots, n\}$, let $i+[n] = \{i+1, i+2, \dots, i+n\}$. The cup game consists of many successive rounds. On the t-th round the state starts as S_t. The filler chooses the number of processors p_t for the round. Then the filler distributes p_t units of water among the cups (with at most 1 unit of water to any particular cup). After this, we are at an intermediate state in the t-th round, which we call state I_t. Then the emptier chooses p_t cups to empty 1 unit of water from. After this the round is over, and the state is S_{t+1}.

Let the rank of a cup at a given state be its position in a sorted list of the cups, breaking ties arbitrarily but consistently. For example, the fullest cup at a state has rank 1, and the least full cup has rank n. Let $\mu_S(X), m_S(X)$ denote the average fill and mass of a set of cups X respectively at state S (S could be S_t or I_t for any $t \in \mathbb{N}$) [2]. Let $S_t(\{r_1, \dots, r_m\})$ and $I_t(\{r_1, \dots, r_m\})$ denote the cups of ranks $r_1, r_2, \dots, r_m$ at states S_t and I_t respectively. We establish the following Lemma:

Lemma 2. *The greedy emptier maintains the invariant $\mu_{S_t}(S_t([k])) \leq n-k$ for all $t \in \mathbb{N}, k \leq n$. In particular, for $k=1$, this means that the emptier never lets backlog exceed $O(n)$.*

Proof. First note that the invariant is trivial when $k = n$, as the average fill of the set of all cups is by definition 0.

We will prove the invariant by induction on t. The invariant holds trivially for $t = 1$ (the base case of our recurrence): the cups start empty so $\mu_{S_1}(S_1([k])) = 0 \leq n-k$.

Fix a round $t \geq 1$. We assume all the invariants for state S_t (we will only use two of the invariants, but the invariants that we need depend on the choice of p_t by the filler, so we need all of them) and show that $\mu_{S_{t+1}}(S_{t+1}([k])) \leq n-k$.

Note that as the emptier is greedy it always empties from the cups $I_t([p_t])$.

Let A, with $a = |A|$, be $A = I_t([\min(k, p_t)]) \cap S_{t+1}([k])$, that is, A consists of cups among the k fullest cups in I_t that were emptied from and ended up in the k fullest cups in S_{t+1}. Let B, with $b = |B|$, be $I_t([\min(k, p_t)]) \setminus A$, that is B consists of the cups among the k fullest cups at state I_t that the emptier empties from that do not end up in the k fullest cups in S_{t+1}. Let $C = I_t(a+b+[k-a])$, with $c = k-a = |C|$ (Note that $k-a \geq 0$ as $a+b \leq k$).

Note that $a+b = \min(k, p_t)$. Note that $A = I_t([a])$ and $B = I_t(a+[b])$, as every cup in A must have higher fill than all cups in B in order to remain above the cups in B after 1 unit of water is removed from all

[2]Note that previously when we used the notation m, μ for mass and average fill, we left off the subscript indicating the state at which the properties were measured. Previously it was sufficient to leave the round implicit as it was understandable from context, however in this Section the state is crucial, and needs to be explicit in the notation.

cups in $A \cup B$. Further, note that $S_{t+1}([k]) = A \cup C$, because once the cups in B are emptied from cups in C take their place among the k fullest cups.

With these definitions made, we proceed to prove the Lemma.

First we prove the following key property, which we call the ***switcheroo-ability of cups***. The property is: without loss of generality $S_t([a+b]) = I_t([a+b])$.

Proof. Say there are cups x, y with $x \in S_t([a+b]) \setminus I_t([a+b]), y \in I_t([a+b]) \setminus S_t([a+b])$. Let the fills of cups x, y at state S_t be f_x, f_y; note that $f_x > f_y$. Let the amount of fill that the emptier adds to these cups be $\Delta_x, \Delta_y \leq 1$; note that $f_x + \Delta_x < f_y + \Delta_y$.

Define a new state S'_t where cup x has fill f_y and cup y has fill f_x. Let the amount of water that the filler places in these cups from the new state be $f_x - f_y + \Delta_x$ and $f_y - f_x + \Delta_y$ for cups x, y respectively. Note that this is valid as $f_y - f_x + \Delta_y \leq 1$ and $f_x - f_y + \Delta_x < 1$ which is true because, $f_x - f_y + \Delta_x < \Delta_y \leq 1$ and $f_y - f_x + \Delta_x < \Delta_x \leq 1$.

We can repeatedly apply this process to swap each cup in $I_t([a+b]) \setminus S_t([a+b])$ into being one of the $a+b$ fullest cups in the new state S'_t. At the end of this process we will have some "fake" state S_t^f. Note that S_t^f must satisfy the invariant if S_t satisfied the invariant, because we are essentially just renaming variables, or in other words swapping the fill of certain cups.

It is without loss of generality that we start in state S_t^f because from state I_t we could equally well have come from state S_t or state S_t^f. $\square$

Now we proceed with the proof of the Lemma.

First we consider the case $b = 0$. If $b = 0$, then $S_{t+1}([k]) = S_t([k])$. The emptier has removed a units of fill from the cups in $S_t([k])$ (specifically the cups in A), and the filler has distributed at most a units of among the cups in $S_t([k])$. Thus the invariant holds:

$$m_{S_{t+1}}(S_{t+1}([k])) \leq m_{S_t}(S_t([k])) + a - a \leq k(n-k).$$

Now consider $b \neq 0$. In particular, note that this implies $a < k$ as $a + b \leq k$. Recall that $S_{t+1}([k]) = A \cup C$, so the mass of the k fullest cups at S_{t+1} is the mass of $A \cup C$ at S_t plus any water added to cups in $A \cup C$ by the filler, minus any water removed by the emptier. The emptier removes exactly a units of water from $A \cup C$, and the filler adds at most $a + b \leq p_t$ (recall $a + b = \min(p_t, k)$) units of water to $A \cup C$. Thus:

$$m_{S_{t+1}}(S_{t+1}([k])) \leq m_{S_t}(A) + m_{S_t}(C) + b.$$

This is easy to bound if $m_{S_t}(C) \leq m_{S_t}(B \cup C) - b$, becuase

$$m_{S_t}(A) + m_{S_t}(B \cup C) = m_{S_t}(A \cup B \cup C) \leq m_{S_t}([k])$$

which would imply the invariant for S_{t+1}, k. If $\mu_{S_t}(C)$ is not significantly less than $\mu_{S_t}(B \cup C)$ we have more difficulty. The key insight is to notice that larger values for $m_{S_t}(A)$ correspond to smaller values for $m_{S_t}(C)$ becase of the invariants; the higher fill in A ***pushes down*** the fill that C can have. By quantifying exactly how much higher fill in A pushes down fill in C we can arrive at the desired invariant. We can upper bound $\mu(C)$ by $\frac{c}{b+c} m_{S_t}(BC) = (m_{S_t}(ABC) - m_{S_t}(A)) \frac{c}{b+c}$ as $\mu_{S_t}(C) \leq \mu_{S_t}(B)$ (we established that this was without loss of generality because switcheroo-ing is possible). Thus, we have

$$m_{S_t}(A) + m_{S_t}(C) \leq \frac{c}{b+c} m_{S_t}(ABC) + \frac{b}{b+c} m_{S_t}(A).$$

Now we set $\mu_{S_t}(ABC)$ and $\mu_{S_t}(A)$ extremally, that is $\mu_{S_t}(ABC) = n - |ABC|$ and $\mu_{S_t}(A) = n - |A|$. At this point the inequality can be verified by straightforward algebra, however this is not elegant; instead, we combinatorially interpret the sum. Consider $\mu_{S_t}(BC)$. By setting A extremally we have raised its average fill $|BC|$ above $n - |ABC|$, the average fill of ABC. If we initially had equal water level in all cups, then to equalize an increase in $\mu_{S_t}(A)$ of $|BC|$, there would need to be a corresponding decrease in the average fill of $B \cup C$ by $|A|$. Thus we have

$$\mu_{S_t}(BC) \leq n - |ABC| - |A|.$$

Thus we have the following bound:

$$\begin{aligned} m_{S_t}(A) + m_{S_t}(C) & \\ & \leq m_{S_t}(A) + c\mu_{S_t}(BC) \\ & \leq a(n-a) + c(n - |ABC| - a) \\ & (a+c)(n-a) - c(a+c+b) \\ & (a+c)(n-a-c) - cb. \end{aligned}$$

Recall that we were considering $b > 0, c \geq 1$ (we previously handled the case where $b = 0$, and if $b > 0$ then $c = k - a \geq b \geq 0$). Hence we have

$$m_{S_t}(A) + m_{S_t}(C) \leq k(n-k) - b$$

So

$$m_{S_t}(A) + m_{S_t}(C) + b \leq k(n-k).$$

As shown previously the left hand side of the above expression is an upper bound for $m_{S_{t+1}}([k])$. Hence the invariant holds.

Conclusion:

The proof was for arbitrary k, so given that the invariants all hold at state S_t they also must all hold at state S_{t+1}. Thus, by induction we have the invariant for all rounds $t \in \mathbb{N}$. $\square$

4 Oblivious Lower Bounds

An important theorem that we use repeatedly in our analysis is Hoeffding's Inequality:

Theorem 1 (Hoeffding's Inequality)**.** *Let X_i be independent bounded random variables with $X_i \in [a, b]$. Then,*

$$P\left(\Big|\frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X_i])\Big| \geq t\right) \leq 2\exp\left(-\frac{2nt^2}{(b-a)^2}\right)$$

Hoeffding also proved that this is true even if X_i are drawn without replacement from some finite population. This is intuitive as drawing without replacement clearly has less variance than sampling with replacement, i.e. sampling without replacement should be more tightly concentrated around the mean than sampling with replacement. This is a corollary of his Theorem 4, see page 28 of his seminal work [1].

Call an emptying strategy (ℓ, δ)***-greedy-like*** if is satisfies the following property when the number of processors is p: for any cup i, if $\text{fill}(i) + \delta > \ell$, and there are at least p cups containing fill greater than $\text{fill}(i) + \delta$, the emptier does not empty from cup i. Of particular interest is the smoothed greedy emptier, which is $(1, 0)$-greedy-like.

Proposition 2. *There exists an oblivious filling strategy in the variable-processor cup game on n cups that achieves backlog $\Omega(\log n)$ against a (ℓ, δ)-greedy-like emptier (where $\ell, \delta \leq O(1)$ are constants known to the filler), with probability at least $1 - 1/\text{polylog}(n)$.*

Proof. Let A, the ***anchor*** set, be a subset of the cups chosen uniformly at random from all subsets of size $n/2$ of the cups, and let B, the ***non-anchor*** set, consist of the rest of the cups ($|B| = n/2$). Let $h = 2\ell + g$ where g is a sufficiently large constant. At each level of our recursive procedure we will achieve fill h in some fraction of the cups in A, and because the emptier is greedy, we can turn this into a known set of cups with fill at least $h' = \ell + (g + \delta)/2$. Our strategy to achieve backlog $\Omega(\log n)$ overall is roughly as follows:

- **Step 1:** Obtain large positive tilt in B, either by repeatedly making cups in B have a constant probability of having fill at least h and then transferring these cups into A, or by exploiting high expected positive tilt.
- **Step 2:** Reduce the number of processors to a constant fraction nc of n and raise the fill of nc cups to h'. This step relies on the emptier being greedy.
- **Step 3:** Recurse on the nc cups that are known to have fill at least h'.

We can perform $\Omega(\log n)$ levels of recursion, achieving constant backlog at each step (relative to average fill); doing so yields backlog $\Omega(\log n)$.

Now we detail how to achieve Step 1. For each anchor cup i we will perform a ***switching-process***. First we choose an index $j \in [n^2]$; the process proceeds for n^2 ***rounds***, j is the index of the switching-process at which we will switch a cup into the anchor set. On each of the n^2 rounds, the filler selects a random subset $C \subset B$ of the non-anchor cups and plays a single processor cup game on C. On most rounds, all rounds except the j-th the filler does nothing with the cup that it achieves at the end of the single processor cup game. On round j with $1/2$ probability the filler swaps the winner of the single processor cup game into the anchor set, and with $1/2$ probability the filler swaps a random cup from B into the anchor set.

We say that a cup is ***overpowered*** if it contains fill $\geq \sqrt{\frac{nh}{\log\log n}}$. If there is ever an overpowered cup, then the proposition is trivially satisfied. Note that we don't need to know which cup is overpowered because it will take $\Omega(\text{poly}(n))$ rounds for the emptier to reduce the fill below $\text{poly}(n)$. Hence, we can assume without loss of generality that no cup is ever overpowered.

We consider two cases:

- **Case 1:** For at least $1/2$ of the switching-processes, at least $1/2$ of the cups $i \in B$ have $\text{fill}(i) \geq -h$.
- **Case 2:** For at least $1/2$ of the switching-processes, less than $1/2$ of the cups $i \in B$ have $\text{fill}(i) \geq -h$.

Claim 1. *In Case 1, with probability at least $1 - e^{-\Omega(n)}$, we achieve fill at least h in a constant fraction of the cups in A, which in particular implies that we can achieve positive tilt nhk for some known constant $k \in (0, 1)$ (k is a complicated function of h).*

Proof. Consider a switching-process where at least $1/2$ of the cups $i \in B$ have $\text{fill}(i) \geq -h$.

Say the emptier ***neglects*** the anchor set in a round if on at least one step of the round the emptier does not empty from every anchor cup. By playing the single-processor cup game for n^2 rounds, with only one round when we actually swap a cup into the anchor set, we strongly disincentives the emptier from neglecting the anchor set on more than a constant fraction of the rounds.

The emptier must have some binary function, $I(k)$ that indicates whether or not they will neglect the anchor set on round k if the filler has not already swapped. Note that the emptier will know when the filler perform a swap, so whether or not the emptier neglects a round k depends on this information. This is the only relevant statistic that the emptier can use to decide whether or not to neglect a round, because on any round when we simply redistribute water amongst the non-anchor cups we effectively have not changed anything about the game state.

If the emptier is willing to neglect the anchor set for at least 1/2 of the rounds, i.e. $\sum_{k=1}^{n^2} I(k) \geq n^2/2$, then with probability at least 1/4, $j \in ((3/4)n^2, n^2)$, in which case the emptier neglects the anchor set on at least $n^2/4$ rounds ($I(k)$ must be 1 for at least $n^2/4$ of the first $(3/4)n^2$ rounds). Each time the emptier neglects the anchor set the mass of the anchor set increases by at least 1. Thus the average fill of the anchor set will have increased by at least $(n^2/2)/(n/2) \geq \Omega(\text{poly}(n))$ over the entire process in this case, implying that we win automatically as there must be an overpowered cup.

Otherwise, there is at least a 1/2 chance that the round j, which is chosen uniformly at random from the rounds, when the filler performs a switch into the anchor set occurs on a round with $I(j) = 0$, indicating that the emptier won't neglect the anchor set on round j. In this case, the round was a legitimate single processor cup game on C_j, the randomly chosen set of e^{2h} cups on the j-th round. Then we achieve fill increase $\geq 2h$ by the end of the game with probability at least $1/e^{2h}!$, the probability that we correctly guess the sequence of cups within the single processor cup game that the emptier empties from.

The probability that the random set $C_j \subset B$ contains only elements with fill $\geq -h$ is basically $1/2^{e^{2h}}$, because at least half of the elements of B have fill $\geq -h$ (in reality the selection of elements of C are not independent events, but as h is constant here this does not matter). If all elements of C_j have fill $\geq -h$, then the fill of the winner of the cup game has fill at least $-h + 2h = h$ if we guess the emptier's emptying sequence correctly.

Combining the results, we have that for such a switching-process there is a constant probability of the cup which we switch into the anchor set has fill $\geq h$.

Say that this probability is $k \in (0, 1)$. Then the expectation of the number of cups $i \in A$ with $\text{fill}(i) \geq h$ is at least $kn/2$. Let X_i be the binary random variables, with X_i taking value 1 if the i-th switching-process succeeded, and 0 if it failed. Then by a Chernoff Bound (Hoeffding's Inequality applied to Binary Random Variables),

$$P\left(\sum_{i=1}^{n/2} X_i \leq nk/4\right) \leq e^{-n(k/2)^2}.$$

That is, the probability that less than $nk/4$ of the anchor cups have fill at least h is exponentially small in n.

□

Claim 2. *In Case 2, with probability at least $1 - 1/\text{polylog}(n)$, we achieve positive tilt $hn/8$ in the anchor set.*

Proof. Consider a switching-process where we have less than 1/2 of the cups $i \in B$ with $\text{fill}(i) \geq -h$.

We assume for simplicity that the average fill of B is 0. In reality this is not the case, but by a Hoeffding bound and the fact that overpowered cups don't exist, the fill is really tightly concentrated around 0, so this is almost without loss of generality.

Let the positive tilt of a cup i be $\text{tilt}_+(i) := \max(\text{fill}(i), 0)$. We have

$$\mathbb{E}[\text{tilt}_+(X)] = \frac{1}{2}\,\mathbb{E}[|\,\text{fill}(X)|] \geq h$$

(because negative tilt is at least $nh/2$ and positive tilt must oppose this).

Let Y_i be the random variable $Y_i = \text{tilt}_+(X)$ where X is a randomly selected cup from the non-anchor set at the start of the i-th round of playing single processor cups games. Note that the Y_i are not really independent, but it is probably ok. Note that $0 \leq Y_i \leq hn/\lg\lg n$. Now we have, by Hoeffding's inequality, that

$$P\left(\left|\frac{1}{n/2}\sum_{i=1}^{n/2}(Y_i - \mathbb{E}[Y_i])\right| \geq h/2\right) \leq$$

$$2\exp\left(-\frac{n(h/2)^2}{(\sqrt{hn/\lg\lg n})^2}\right)$$

$$P\left(\frac{1}{n/2}\sum_{i=1}^{n/2} Y_i \leq h/4\right) \leq 1/\text{polylog}(n)$$

□

In both cases we achieve, with probability at least $1 - 1/\text{polylog}\, n$, positive tilt at least hnk in the anchor set for some known $k \in (0, 1)$. Using the positive tilt, with one processors, we can transfer over the fill into nk cups. Note, we use one processor because we do not know how many cups the fill is concentrated in.

The filler repeatedly distributes 1 unit of fill to each of the nk cups in succession, and continues until h' fill has been distributed. We cannot continue beyond this point because we have used up the positive tilt. Now we recurse on this set of nk cups.

Note that this is the only part of this proof that was specific to a greedy emptier: when we wanted to achieve known fill in some cups. Against an arbitrary opponent we can't assume that just because they are far behind means that they won't oppose our attempts to achieve cups with known fill. Extending this result to non-greedy emptiers, or showing that it cannot be extended is an important open question.

We can perform $\Omega(\log n)$ levels of recursion, and gain $\Omega(1)$ fill at each step. Hence, overall, backlog of $\Omega(\log n)$ is achieved. □

Lemma 3 (The Oblivious Amplification Lemma). *Given an oblivious filling strategy for achieving backlog $f(n)$ in the variable-processor cup game on n cups that succeeds with probability at least $1/2$, there exists a strategy for achieving backlog*

$$f'(n) \geq \frac{1}{32}(f(n/2) + f(n/4) + f(n/8) + \cdots)$$

that succeeds with constant probability.

Proof. We essentially perform the same proof as Proposition 2, but some new issues arise, which we proceed to highlight and address.

Claim 3. *Let a cup be* ***verysad*** *if it has fill $< -nh/\lg\lg n$. WLOG there are no verysad cups.*

Proof. First note that because WLOG there are no overpowered cups, there fewer than $n/2$ verysad cups.

Consider 2 cases:

- If the mass of the verysad cups is less than $nh/8$ then we can ignore them and accept a $-h/8$ penalty to the average fill.
- On the other hand, if the mass of the verysad cups is greater than $nh/8$, then by the end the average fill of everything else is already $h/8$ which is also basically as desired.

□

Claim 4. *WLOG A, B have average fill $\geq -h/8$. In particular, we can construct a subset of $n/2$ cups with average fill $\geq -h/8$ with high probability in n.*

Proof. Recall the definition of an overpowered cup as a cup with fill $\geq nh/\lg\lg n$, and the fact that WLOG there are no overpowered cups. So, If we randomly pick B then this means that we are pretty good. Formalizing this, let X_i be the fill of the $n/2$-th randomly chosen cup for B. Unfortunately these are not quite independent events.

Lets say we pick $2n$ things from n things with replacement. Claim: with exponentially good probability we have $n/2$ distinct things. Proof: chernoff bound. Let X_i be indicator variable for cup i (whether it was chosen or not). Probability that X_i was chosen: $1-((n-1)/n)^n \approx 1-1/e > 1/2$ for large n. Then by a Chernoff Bound we have that $\sum_i X_i$ is tightly concentrated around its mean, which is larger than n. In particular, with probability exponentially close to 1 in n we have that at least $n/2$ cups were chosen.

initially solution: no overpowered cups wlog, so if we pick them randomly star holds by Hoeffding's. (kinda, bc stuff isnt really independent, can probably swap with replacement to fix this tho)

□

Claim 5. *What if C needs to be big because we need big backlog?*

Proof. this isn't a problem because the base case is the only case that needs to explicitly deal with positive and negative fill. □

These concerns resolved, the exact same argument as in Proposition 2 gives the desired result.

□

Corollary 2. *There is an oblivious filling strategy for the variable-processor cup game on n cups that achieves backlog $2^{\Omega(\sqrt{\log n})}$ in running time $O(n)$*

Proof. We must reduce want to reduce $\log^2 n$ to $\log n$ to achieve the appropriate running-time, so we reduce n to $n' = 2^{\sqrt{\log n}}$. This detail taken care of we apply exactly the same recursive construction of $f_{\theta(\log n)}$ as in Corollary 1, but using repeated application of the Oblivious Amplification Lemma rather than the Adaptive Amplification Lemma, which yields the disclaimer that the backlog is only achieved with constant probability. So we achieve backlog $\Omega(2^{\log n'})$ in running time $O(2^{\log^2 n'})$. By design, expressing this in terms of n we have running time $O(n)$ (randomized lower bounds are not supposed to take longer than poly(n) time), and as a consequence we get backlog $\Omega(2^{\sqrt{\log n}})$. □

References

[1] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, page 28, 1962.