Variable-Processor Cup Games

William Kuszmaul*[1] and Alek Westover†[2]

MIT[1], MIT PRIMES[2]
kuszmaul@mit.edu, alek.westover@gmail.com

Abstract

In a ***cup game*** a filler and an emptier take turns adding and removing water from cups, subject to certain constraints. In one version of the cup game, the p-processor cup game, the filler distributes p units of water among the n cups with at most 1 unit of water to any particular cup, and the emptier chooses p cups to remove at most one unit of water from. Analysis of the cup game is important for applications in processor scheduling, buffer management in networks, quality of service guarantees, and deamortization.

We investigate a new variant of the classic multi-processor cup game, which we call the ***variable-processor cup game***, in which the resources of the emptier and filler are variable. In particular, in the variable-processor cup game the filler is allowed to change p at the beginning of each round. Although the modification to allow variable resources seems small, we will show that it drastically alters the outcome of the game.

We construct a filling strategy that an adaptive filler can use to get backlog $\Omega(n)$ in running time $2^{O(n)}$. We also construct a filling strategy that an adaptive filler can use to get backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ in running time $2^{O(\log^2 n)}$. Not only is this bound shockingly large, but the steep trade off-curve between running-time and backlog is very surprising: the time required goes up from quasi-polynomial time to exponential time!

Furthermore, we demonstrate that this lower bound on backlog is tight: using a novel set of invariants we prove that a greedy emptier never lets backlog exceed $O(n)$.

We also investigate bounds on an oblivious filler. We show, using concentration bounds for random variables (Hoeffding's Inequality), that – surprisingly – an oblivious filler can achieve essentially the same lower bounds as an adaptive filler. However, oblivious lower bounds are only supposed to take $O(\text{poly}(n))$ time, so we apply the strategies from the adaptive case to a subset of the cups. Doing so, an oblivious filler can achieve backlog $2^{\Omega(\sqrt{\log n})}$ in running time $O(\text{poly}(n))$ with constant probability against any "greedy-like" emptier.

*Supported by a Hertz fellowship and a NSF GRFP fellowship
†Supported by MIT PRIMES

1 Introduction

Definition and Motivation. The ***cup game*** is a multi-round game in which the two players – the ***filler*** and the ***emptier*** – take turns adding and removing water from cups. On each round of the classic ***p-processor cup game*** on n cups, the filler first distributes p units of water among the n cups with at most 1 unit to any particular cup (without this restriction the filler can trivially achieve unbounded backlog by placing all of its fill in a single cup every round), and then the emptier removes 1 unit of water from each of p cups.

The cup game naturally arises in the study of processor-scheduling. The incoming water added by the filler represents work added to the system at time steps. At each time step after the new work comes in each of p processors must be allocated to a task, which they will achieve 1 unit of progress on before the next time step. The assignment of processors to tasks is modeled by the emptier deciding which cups to empty from. The backlog of the system is the largest amount of work left on any given task. To model this, in the cup game, the ***backlog*** of the cups is defined to be the fill of the fullest cup at a given state. It is important to know bounds on how large backlog can get.

Previous Work. The bounds on backlog are well known for the case where $p = 1$, i.e. the ***single-processor cup game***. In the single-processor cup game an adaptive filler can achieve backlog $\Omega(\log n)$ and a greedy emptier never lets backlog exceed $O(\log n)$. The bounds are much better against an oblivious filler. In the randomized version of the single-processor cup game, which can be interpreted as a smoothed analysis of the deterministic version, the emptier never lets backlog exceed $O(\log\log n)$, and a filler can achieve backlog $\Omega(\log\log n)$.

Recently Kuszmaul has achieved bounds on the case where $p > 1$, i.e. the ***multi-processor cup game*** [2]. Kuszmaul showed that in the p-processor cup game on n cups a greedy emptier never lets backlog exceed $O(\log n)$. He also proved a lower bound of $\Omega(\log(n-p))$. Recently we showed a lower bound of $\Omega(\log n - \log(n-p))$. Combined these bounds imply a lower bound of $\Omega(\log n)$. Kuszmaul also established an upper bound of $O(\log\log n + \log p)$ against oblivious fillers, and a lower bound of $\Omega(\log\log n)$ (note that tight bounds on backlog against an oblivious filler are not yet known for large p).

Our Variant. We investigate a variant of the vanilla multi-processor cup game which we call the ***variable-processor cup game***. In the variable-processor cup game the filler is allowed to change p (the total amount of water that the filler adds, and the emptier removes, from the cups per round) at the beginning of each round. Note that we do not allow the resources of the filler and emptier to vary separately; just like in the classic cup game we take the resources of the filler and emptier to be identical. Although this restriction may seem artificial, it is crucial; if the filler has more resources than the emptier, then the filler could trivially achieve unbounded backlog, as average fill will increase by at least some positive constant at each round. Taking the resources of the players to be identical makes the game balanced, and hence interesting.

A priori having variable resources offers neither player 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[1]. We hoped that the variable-processor cup game could be simulated in the vanilla multi-processor cup game, because the extra ability given to the filler does not seem very strong.

However, we show that attempts at simulating the variable-processor cup game are futile because the variable-processor cup game is–surprisingly–vastly different from the multi-processor cup game.

Outline and Results. In Section 2 we establish the conventions and notations we will use to discuss the variable-processor cup game.

In Section 3 we provide an inductive proof of a lower bound on backlog in Corollary 1. The base case of the argument is a direct consequence of Proposition 1, and the inductive step follows from the "Adaptive Amplification Lemma" (Lemma 1). Corollary 1 gives a lower bound of $\Omega(n)$ on backlog. Corollary 1 provides two algorithms: one algorithm with running time $2^{O(n)}$ that achieves backlog $\Omega(n)$, and another with running time $2^{O(\log^2 n)}$ that achieves backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$.

In Section 4 we prove a novel invariant: the average fill of the k fullest cups is at most $n-k$. In particular this implies (setting $k=1$) that backlog is $O(n)$. Thus, our analysis is tight.

Section 5 has similar macro-structure to Section 3: We lower bound backlog in Corollary 3, using Proposition 2 as the base case of the inductive argument and the "Oblivious Amplification Lemma" (Lemma 3) to facilitate the inductive step. Corollary 3 gives the lower bound $2^{\Omega(\sqrt{\log n})}$ on backlog, against a "greedy-like" emptier. In particular the corollary asserts that we can achieve this backlog in time $O(\text{poly}(n))$. Note that the restriction on runtime of the filler is the main way in which this bound differs from the adaptive case.

[1] A useful part of many filling algorithms is maintaining an "anchor" set of "anchored" cups. The filler always places 1 unit of water in each anchored cup. This ensures that the fill of an anchored cup never decreases after it is placed in the anchor set.

2 Preliminaries

The cup game consists of a sequence of 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, the game is in an intermediate state, which we call state I_t. Then the emptier chooses p_t cups to empty 1 unit of water from. This concludes the round; the state of the game is now S_{t+1}.

Denote the fill of a cup c by $\text{fill}(c)$. Let the ***positive tilt*** of a cup c be $\text{tilt}(c) = \max(0, \text{fill}(c))$, and let the positive tilt of a set X of cups be $\text{tilt}(X) = \sum_{c \in X} \text{tilt}(c)$. Let the ***mass*** of a set of cups X be $m(X) = \sum_{c \in X} \text{fill}(c)$. Denote the average fill of a set of cups X by $\mu(X)$. Note that $\mu(X)|X| = m(X)$.

Let the ***rank*** of a cup at a given state be its position in a list of the cups sorted by fill at the given state, breaking ties arbitrarily but consistently. For example, the fullest cup at a state has rank 1, and the least full cup has rank n.

We adopt the convention of allowing for negative fill: i.e. regardless of the fills of the cups the emptier always empties exactly 1 unit of water from p cups. [2]

3 Adaptive Filler 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

[2] Allowing for cups to have negative fill makes the game strictly harder for the filler, as it means that none of the emptier's emptying is ever wasted by cups "zeroing-out". In variants of the cup game without negative fill, when the emptier empties from a cup with fill less than 1 the cups fill goes to 0. This however implies that the total mass of water removed by the emptier is less than p. Our filling strategy does not rely on cups zeroing out to achieve large backlog however, so the lower bounds hold regardless of our choice of allowing or not allowing negative fill.

On the other hand, the proof of the upper bound does depend on the fact that cups never zero out. In particular, throughout the proof of the upper bound we call the average fill of the cups 0 and say that it never changes. We chose to allow negative fill (which has no physical analog in work scheduling) because it makes the problem more elegant.

completed. Let A consist of the $n/2$ fullest cups, and B consist of the rest of the cups (at any given state, so A,B are implicitly functions of the round t).

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

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*** –the set of cups that we place fill in– to be A. Note that that $\text{fill}(c) \ge -2h$ for all cups $c \in A$, as A consists of the $n/2$ fullest cups. We will 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 $1/(n/2)+1/(n/2-1)+\cdots+1/2+1/1 \ge \ln n/2 = 4h$. 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 we can achieve backlog 1 on n_0 cups. Let $\delta \in (0,1)$ be a parameter, and let $L \in \mathbb{N}$ be a constant such that $n_0 \le (1-\delta)\delta^L n \le n_0/\delta$.*

Then, there exists an adaptive filling strategy that achieves backlog $f'(n)$ satisfying

$$f'(n) \ge (1-\delta)\sum_{\ell=0}^{L} f((1-\delta)\delta^\ell n)$$

and $f'(n) \ge 1$, in the variable-processor cup game on $n \ge n_0$ 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 the filler uses the following ***process*** to achieve the desired average fill in A: 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; repeat 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 the process B has average fill lower than $-h_\ell \cdot \delta/(1-\delta)$, then A 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 L levels of recursion. Summing h_ℓ for $0 \le \ell \le L$ yields the desired result.

Note that as $n \ge n_0$ 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 are adaptive filling strategies for the variable-processor cup game on n cups that achieve backlog $\Omega(n)$ in time $2^{O(n)}$ and backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon>0$ in time $2^{O(\log^2 n)}$.*

Proof.
Fix $\epsilon \in (0,1/2)$, and let c,δ be parameters, with $c \in (0,1), 0<\delta \ll 1/2$ – these will depend on ϵ,n. Say that we aim to achieve backlog at least $cn^{1-\epsilon}$. Observe that if we apply the Amplification Lemma to the function satisfying $f(k) \geq ck^{1-\epsilon}$ for $k \leq g$ then for any k_0 with $k_0(1-\delta) \leq g$ (which enforces $k_0 \leq g/(1-\delta)$) we have the following:

$$\begin{aligned} f'(k_0) \geq & \\ & (1-\delta)\sum_{\ell=0}^{L} c(((1-\delta)\delta^\ell)k_0)^{1-\epsilon} \\ & = ck_0^{1-\epsilon}(1-\delta)^{2-\epsilon}\sum_{\ell=0}^{L}(\delta^\ell)^{1-\epsilon}, \end{aligned}$$

where L is the greatest integer such that $(1-\delta)\delta^L n \geq n_0$ where n_0 is a constant such that we can achieve backlog 1 on n_0 cups (this definition is identical to the definition in the statement of Lemma 1). Note that as δ will be very small, $\sum_{\ell=0}^{L}(\delta^L)^{1-\epsilon}$ is very well approximated by $1+\delta^{1-\epsilon}$, so we will not loose much by relaxing our lower bound on $f'(k_0)$ to only use the first 2 terms of the sum. Then we have

$$f'(k_0) \geq ck_0^{1-\epsilon}(1-\delta)^{2-\epsilon}(1+\delta^{1-\epsilon}).$$

Let

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

We prove the following claim:

Claim 1. *There exists an appropriate choice of δ that is small enough such that $h(\delta)>1$ and large enough such that $L \geq 1$, when ϵ is chosen to be $4/\lg n$, or a positive constant. In particular, if ϵ is chosen to be $4/\lg n$ then we will choose $\delta \leq O(1/n)$, and if ϵ is chosen to be a positive constant then we will choose $\delta \leq O(1)$.*

Note that if $h(\delta) \geq 1$, then $f'(k_0) \geq ck_0^{1-\epsilon}$, meaning we have constructed from f a new function f' that satisfies the inequality $f'(k) \geq ck^{1-\epsilon}$ for $k \leq g/(1-\delta)$, as opposed to only for $k \leq g$ as in the case of f. [3] Then by repeatedly amplifying a function, we should be able to arbitrarily extend the support, which would help us attain the desired backlog.

We now prove Claim 1.

[3] Note that although $f'(k) \geq ck^{1-\epsilon}$ holds for at least as many k as $f(k) \geq ck^{1-\epsilon}$, it does not necessarily hold for strictly more; in particular, if $\lfloor g/(1-\delta) \rfloor = g$ then the inequality on f' holds for no more k than the inequality on f, as f and f' are functions on $\mathbb{N}$. We have to be somewhat careful about the fact that there are an integer number of cups.

Proof. Consider the Taylor series for $(1-\delta)^{2-\epsilon}$ about $\delta=0$:

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

So, to find a δ where $h(\delta) \geq 1$ it suffices – note that, again, we choose to neglect the δ^2 term as it does not help us substantially because it is so small – to find a δ with

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

Rearranging we have

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

This clearly is true for sufficiently small δ, as $\delta^{1-\epsilon}$ will be much greater than δ or $\delta^{2-\epsilon}$. However it will be beneficial to have a more explicit criterion for possible choices of δ in terms of ϵ. To get this, we enforce a much stronger inequality on $\delta^{1-\epsilon}$ by vastly overestimating $\delta^{2-\epsilon}$ as δ. Surprisingly even with this overestimate we are still able to get the desired value of ϵ to work, as we will demonstrate later. We have,

$$\delta \leq \frac{1}{(2(2-\epsilon))^{1/\epsilon}}. \tag{1}$$

In addition to the constraint that δ must be small enough such that $h(\delta) \geq 1$, the only other constraint on δ is that δ must be large enough that the sum from the Amplification Lemma has at least two terms, i.e. such that $L \geq 1$. The condition $L \geq 1$ enforces

$$\delta(1-\delta)n \geq n_0.$$

Recall that we choose $\delta < 1/2$, so $1-\delta > 1/2$. Thus to make δ sufficiently big it suffices to chose δ with

$$\delta \geq 2n_0/n. \tag{2}$$

Any choice of δ that is sufficiently large to make $L \geq 1$ and simultaneously small enough to make $h(\delta) \geq 1$ is a valid choice of δ. That is, δ is valid if and only if it satisfies

$$2n_0/n \leq \delta \leq \frac{1}{(2(2-\epsilon))^{1/\epsilon}}. \tag{3}$$

To achieve the desired backlog of $\Omega(n)$ we can use $\epsilon = \gamma/\lg n$ for appropriate constant γ, as

$$n^{1-\gamma/\lg n} = n/2^\gamma = \Omega(n).$$

We show that there is a valid choice of γ such that the following inequality is satisfied:

$$2n_0/n \leq \frac{1}{(2(2-\gamma/\lg n))^{(1/\gamma)\lg n}}. \tag{4}$$

Note that

$$(2(2-\gamma/\lg n))^{(1/\gamma)\lg n} \leq 4^{(1/\gamma)\lg n} \leq n^{2/\gamma}$$

Clearly by choosing e.g. $\gamma = 4$ we have the desired inequality. Inequality 4 implies that there is a valid choice of δ when we chose $\epsilon = \gamma/\lg n$. When proving that we can achieve backlog $\Omega(n)$ we use $\epsilon = 4/\lg n$, and $\delta = O(1/n)$ satisfying Inequality 3, based on our choice of ϵ. When proving that we can achieve backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ we choose δ to be a constant satisfying Inequality 1, and δ, being constant, is trivially not too small, hence satisfies Inequality 2.

□

Now we proceed to show that with the appropriate values of δ, ϵ we can achieve a filling strategy that achieves backlog $cn^{1-\epsilon}$ on n cups. First we present a simple existential argument which asserts that a strategy that achieves the desired backlog exists. Then we provide two constructive arguments: one achieving backlog $\Omega(n)$ in running time $2^{O(n)}$, the other achieving backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ in running time $2^{O(\log^2 n)}$. Both constructive arguments rely on repeated application of the Amplification Lemma.

Existential Argument. Let $\epsilon > 0$ be constant. By Claim 1, there is a valid constant setting of δ; let $\delta \ll 1/2$ be appropriate constant. Let f^* be the supremum over all filling strategies of the backlog achievable on n cups. Then f^* must be greater than or equal to the amplification of f^*. Assume for contradiction that there is some least n_* such that

$$\begin{cases} f^*(k) < ck^{1-\epsilon}, & k > n_* \\ f^*(k) > ck^{1-\epsilon}, & k < n_* \end{cases}$$

Note that $n_*(1-\delta)\delta \geq n_0$ by appropriate choice of constant c, and Proposition 1, which states that we can get backlog $O(\log n_*)$ on n_* cups[4]. Because f^* satisfies the Amplification Lemma we have:

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

which is a contradiction. Hence f^* achieves backlog $cn^{1-\epsilon}$ for all n.

Constructive Argument achieving backlog $\Omega(n^{1-\epsilon})$ (for constant $\epsilon > 0$) in time $2^{O(\log^2 n)}$. It is desirable to have an algorithm for achieving this backlog with bounded running time; we now modify the existential argument to make it constructive, which yields an algorithm for achieving backlog $cn^{1-\epsilon}$ on n cups in finite running time. We again use constant $\epsilon > 0$ and appropriate constant δ.

We start with the algorithm given by Proposition 1 for achieving backlog

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

Then we construct an algorithm that achieves better backlog using the Amplification Lemma (Lemma 1): we construct f_{i+1} as the amplification of f_i.

Define a sequence $g_i \in \mathbb{N}^\infty$ with

$$g_i = \begin{cases} \lceil 1/\delta \rceil \gg 1, & i = 0, \\ \lceil g_{i-1}/(1-\delta) \rceil - 1 & i \geq 1 \end{cases}$$

Note that is, g_{i+1} is the greatest integer strictly less than $g_i/(1-\delta)$. Note that $(1/\delta)/(1-\delta) > (1+\delta)/\delta = 1/\delta + 1$. Thus $g_1 = 1 + g_0$, and in general, $g_{i+1} > g_i$, because the difference $g_{i+1} - g_i$ can only grow as i grows.

We claim the following regarding this construction:

$$f_i(k) \geq ck^{1-\epsilon} \text{ for all } k < g_i. \qquad (*)$$

We shall prove Claim * by induction on i.

Claim * is true in the base case of f_0 by taking c sufficiently small, in particular small enough that $f_0(k) \geq ck^{1-\epsilon}$ holds for $k < g_0$.[5] As our inductive hypothesis we assume Claim * for f_i; we aim to show that Claim * holds for f_{i+1}. Note the key property of g_i, that $g_{i+1} \cdot (1-\delta) < g_i$. Also note that (without loss of generality) the f_i are monotonically increasing functions: given more cups we can always achieve higher fill than with fewer cups. Thus we have, for any $k < g_{i+1}$,

$$\begin{aligned} &f_{i+1}(k) \\ &\quad \geq (1-\delta)\sum_{\ell=0}^{L} f_i((1-\delta)\delta^\ell k) \\ &\quad \geq ck^{1-\epsilon} h(\delta) \\ &\quad \geq ck^{1-\epsilon}, \end{aligned}$$

as desired.

Note that $g_{i+1} \geq g_i + 1$ so by continuing this process we eventually reach some f_{i_*} such that $f_{i_*}(n) \geq cn^{1-\epsilon}$. Note that $i_* \leq n$. Let the running time $f_{i_*}(n)$ be $T(n)$. Note that $f_{i_*}(n)$ must call $f_{i_*-1}(n(1-\delta)\delta^\ell)$ as many as $n(1-\delta)\delta^\ell$ times, for all $0 \leq \ell \leq L$. However, we only use the terms of the sum where $\ell = 0, 1$, so we could use a modified version of the Amplification Lemma in which we truncate the sum. This we have the following (loose) recurrence bounding $T(n)$:

$$T(n) \leq \delta n \cdot T(n(1-\delta)) + T(\delta n).$$

[4]Note: this is where it is crucial that ϵ, δ are constants.

[5]Note: this is where it is crucial that ϵ, δ are constants.

We can upper bound this by

$$n^{\frac{\log n}{\log(1/(1-\delta))}}.$$

Continuing for $O(\log n)$ levels of recursion should be sufficient to achieve the desired backlog. This gives running time

$$T(n) \le ((1+\delta)n)^{O(\log n)} \le 2^{O(\lg^2 n)}$$

as desired. ok, technically I'm ignoring integer problems in saying lets only do $O(\lg n)$ levels of recursion, it'll be enough. I should prove it. But for δ constant it seems pretty obvious.

Constructive Argument for backlog $\Omega(n)$ in time $2^{O(n)}$. We describe a simple filling strategy that gives the desired backlog. Let $n_0 \le O(1)$ be a constant such that we can achieve backlog 1 on n_0 cups, and note that this is possible by Proposition 1. We construct a function that achieves large backlog on n cups. To achieve large backlog on n cups we first recursively apply our function to $(1-\delta)n$ cups repeatedly (for each of the δn cups that we are attempting to get high fill in), as described in the proof of the Amplification Lemma, and transfer over the cups that we get. Then we achieve backlog 1 on the δn cups whose average fill has been increased. The backlog we achieve satisfies the following recurrence:

$$f(n) \ge \begin{cases} (1-\delta) f((1-\delta)n) + 1, & \text{if } n\delta(1-\delta) > n_0 \\ 0, \text{ else.} \end{cases}$$

Let $(1-\delta)^c = \delta$, let $\delta^2 n < n_0 < (1-\delta)^{2c-1} n$ by our choice of $\delta = O(1/n)$. We can get backlog

$$\sum_{i=1}^{c} (1-\delta)^i.$$

To see this, consider a binary tree representing our algorithm. At every branch we both proceed to recurse on a $1-\delta$ fraction of the cups, and achieve backlog 1 on a δ fraction of the cups.

The sum evaluates to

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

which, if we chose $\delta = 1/n$, becomes $\Omega(n)$.

The running time satisfies the recurrence

$$T(n) = \delta n T((1-\delta)n) + O(1)$$

because to achieve backlog $f(n)$ we must achieve backlog $f((1-\delta)n)$ δn times, and then achieve backlog 1 on the remaining cups. Solving this recurrence yields that the running time is

$$\frac{(\delta n)^c - 1}{\delta n - 1}.$$

Recalling that $\delta = O(1/n)$ this becomes

$$2^{O(n)}.$$

□

4 Adaptive Filler Upper Bound

Let $\mu_S(X)$ and $m_S(X)$ denote the average fill and mass of a set of cups X at state S (e.g. $S = S_t$ or $S = I_t$).[6] 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. Let $[n] = \{1,2,\dots,n\}$, let $i+[n] = \{i+1, i+2, \dots, i+n\}$. We will use concatenation of sets to denote unions, i.e. $AB = A \cup B$.

We establish the following Lemma:

Lemma 2. *The greedy emptier maintains the invariant*

$$\mu_{S_t}(S_t([k])) \le n-k \text{ for all } t \ge 1, k \in [n].$$

In particular, for $k=1$, this says that the greedy 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 for the inductive proof): the cups start empty so $\mu_{S_1}(S_1([k])) = 0 \le n-k$.

Fix a round $t \ge 1$, and any $k \in [n-1]$. We assume invariant for all values of $k \in [n]$ for state S_t (we will only explicitly use two of the invariants for each k, but the invariants that we need depend on the choice of p_t by the filler, so we actually need all of them) and show that the invariant on the k fullest cups holds on round $t+1$, i.e. that

$$\mu_{S_{t+1}}(S_{t+1}([k])) \le n-k.$$

Note that because the emptier is greedy it always empties from the cups $I_t([p_t])$. Let A, with $a = |A|$, be $A = I_t([\min(k,p_t)]) \cap S_{t+1}([k])$; A consists of cups that are among the k fullest cups in I_t, are emptied from, and are among the k fullest cups in S_{t+1}. Let B, with $b = |B|$, be $I_t([\min(k,p_t)]) \setminus A$; B consists of the cups that are among the k fullest cups in state I_t, are emptied from, and aren't among the k fullest cups in S_{t+1}. Let $C = I_t(a+b+[k-a])$, with $c = k-a = |C|$ (Note that $k-a \ge 0$ as $a+b \le k$).

Note that $a+b = \min(k,p_t)$. We also have that $A = I_t([a])$ and $B = I_t(a+[b])$, as every cup in A must

[6] Note that in the lower bound proofs (e.g. Section 3) when we used the notation m (for mass) and μ (for average fill), we omitted the subscript indicating the state at which the properties were measured. In those proofs it was sufficiently clear to leave the state implicit. However, in this section the state is crucial, and needs to be explicit in the notation.

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 cups in AB. Further, note that $S_{t+1}([k])=AC$ because, once the cups in B are emptied from, the cups in B are not among the k fullest cups, so the cups in C take their places among the k fullest cups.
With these definitions made, we proceed to prove the Lemma.

First we prove that without loss of generality $S_t([a+b]) = I_t([a+b])$; we call this fact the ***interchangeability of cups***.

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 Δ_x,$\Delta_y \le 1$; 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. This is valid as both fill amounts are at most 1: $f_x-f_y+\Delta_x<\Delta_y\le 1$ and $f_y-f_x+\Delta_x<\Delta_x\le 1$.

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 invariants if S_t satisfied the invariants, because our process can be thought of as just relabelling the cups; in particular $\text{fill}(S_t^f(r))=\text{fill}(S_t(r))$ for all ranks $r\in[n]$.

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. Thus we consider state I_t to have come from 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:

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

Now consider the case $b\ne 0$. Because $b>0$, and $a+b\le k$ we have that $a<k$, and $c=k-a>0$. Recall that $S_{t+1}([k])=AC$, so the mass of the k fullest cups at S_{t+1} is the mass of AC at S_t plus any water added to cups in AC by the filler, minus any water removed from cups in AC by the emptier. The emptier removes exactly a units of water from AC. The filler adds no more than p_t units of water from AC (because the filler adds at most p_t total units of water per round) and the filler also adds no more than $k=|AC|$ units of water from AC (because the filler adds at most 1 unit of water to each of the k cups in AC). Thus, the filler adds no more than $a+b=\min(p_t,k)$ units of water to AC. Combining these observations we have:

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

The key insight necessary to bound this is to notice that larger values for $m_{S_t}(A)$ correspond to smaller values for $m_{S_t}(C)$ because 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 achieve the desired inequality. We can upper bound $m_{S_t}(C)$ by

$$\frac{c}{b+c}m_{S_t}(BC)=(m_{S_t}(ABC)-m_{S_t}(A))\frac{c}{b+c}$$

because $\mu_{S_t}(C)\le\mu_{S_t}(B)$ without loss of generality by the interchangeability of cups. Thus we have

$$m_{S_t}(AC)\le m_{S_t}(A)+\frac{c}{b+c}m_{S_t}(BC) \tag{5}$$

where

$$\begin{aligned} &m_{S_t}(A)+\frac{c}{b+c}m_{S_t}(BC)\\ &=\frac{c}{b+c}m_{S_t}(ABC)+\frac{b}{b+c}m_{S_t}(A). \end{aligned} \tag{6}$$

Note that the expression in Equation 6 is monotonically increasing in both $\mu_{S_t}(ABC)$ and $\mu_{S_t}(A)$. Thus, by numerically replacing both average fills with their extremal values $(n-|ABC|,n-|A|)$ we upper bound $m_{S_t}(A)+m_{S_t}(C)$. At this point the inequality can be verified by straightforward algebra, however this is not elegant; instead, we combinatorially interpret the sum.

We define a new "fake" state F, which may not represent a valid configuration of cups (i.e. might not satisfy the invariants), where $\mu_F(A)=n-|A|$ and $\mu_F(ABC)=n-|ABC|$, in particular with all the cups in A having identical fill, and all the cups in BC having identical fill. We can think of F as having come from a state where every cup has fill $\mu_F(ABC)=n-|ABC|$. To reach F from this state where every cup has identical fill we must increase the fill of each cup in A by some amount, and decrease the fill of each cup in BC by an amount such that the mass added to A is taken away from BC. To reach fill $\mu_F(A)=n-|A|$, the cups in A must have been increased by $|BC|$ from their previous fill of $n-|ABC|$. To equalize an increase in $\mu_F(A)$ of $|BC|$, we need a corresponding decrease in $\mu_F(BC)$ by $|A|$. That is,

$$\mu_F(BC)=n-|ABC|-|A|.$$

Thus we have the following bound:

$$\begin{aligned} &m_{S_t}(A)+m_{S_t}(C)\\ &\qquad\le m_F(A)+c\mu_F(BC) \qquad (*)\\ &\qquad\le a(n-a)+c(n-|ABC|-a)\\ &\qquad\le (a+c)(n-a)-c(a+c+b)\\ &\qquad\le (a+c)(n-a-c)-cb, \end{aligned}$$

where (*) follows from Equation 6.

Consider a new configuration of fills F achieved by starting with state S_t, and moving water from BC into A until $\mu_F(A)=n-|A|$. [7] This transformation increases (strictly increases if and only if we move a non-zero amount of water) the mass in AC because water in BC counts less towards mass in AC than water in A by Inequality 5. In particular, if mass $\Delta\geq 0$ fill is moved from BC to A, then the mass of AC increases by $\frac{b}{b+c}\Delta\geq 0$.

Since $\mu_F(A)$ is above $\mu_F(ABC)$, the greater than average fill of A must be counter-balanced by the lower than average fill of BC. In particular we must have

$$(\mu_F(A)-\mu_F(ABC))|A|=(\mu_F(ABC)-\mu_F(BC))|BC|.$$

Note that

$$\mu_F(A)-\mu_F(ABC)\geq(n-|A|)-(n-|ABC|)=|BC|.$$

Hence we must have

$$\mu_F(ABC)-\mu_F(BC)\geq|A|.$$

Thus

$$\mu_F(BC)\leq\mu_F(ABC)-|A|\leq n-|ABC|-|A|.$$

Thus we have the following bound:

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

Recall that we were considering $b>0$, and since $b>0$ we have that $c=k-a\geq b>0$, i.e. $c\geq 1$. 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.

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 $t\in\mathbb{N}$. $\square$

[7] Note that whether or not F satisfies the invariants is irrelevant.

5 Oblivious Filler Lower Bound

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

Theorem 1 (Hoeffding's Inequality)**.** *Let X_i for $i=1,2,\ldots,k$ be independent bounded random variables with $X_i\in[a,b]$ for all i. Then,*

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

Let S be a finite population, let X_i for $i=1,2\ldots,k$ be chosen uniformly at random from $S\setminus\{X_1,\ldots,X_{i-1}\}$, and let Y_i for $i=1,2,\ldots,k$ be chosen uniformly at random from S. Note that $\{X_1,\ldots,X_k\}$ represents a sample of S chosen without replacement, whereas $\{Y_1,\ldots,Y_k\}$ represents a sample with replacement. Note that as the Y_i are independent random variables Hoeffding's Inequality provides a bound on the probability of $\sum_{i=1}^{k}Y_i$ deviating from its mean by more than t.

The same bound can be given on the probability of $\sum_{i=1}^{k}X_i$ deviating significantly from its mean, because the probability of $\sum_{i=1}^{k}X_i$ deviating from it's expectation by more than t is at most the probability of $\sum_{i=1}^{k}Y_i$ deviating from it's mean by t. Formally we can write this as

Corollary 2. *Let S be a finite set with $\min(S)\geq a,\max(S)\leq b$, and let X_i for $i=1,2\ldots,k$ be chosen uniformly at random from $S\setminus\{X_1,\ldots,X_{i-1}\}$. Then*

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

Hoeffding proved Corollary 2 in his seminal work [1] (the result follows from his Theorem 4, combined with Hoeffding's Inequality for independent random variables). The intuition behind Corollary 2 is that samples drawn without replacement should be more tightly concentrated around the mean than samples drawn with replacement.

Call an emptying strategy (T,Δ)-***greedy-like*** if is satisfies the following property: for any cup c, if $\text{fill}(c)+\Delta>T$, and there are at least p cups containing fill greater than $\text{fill}(c)+\Delta$, the emptier does not empty from cup c. Combinatorially the quantity T is the threshold above which the filler "notices" cups, and $\pm\Delta$ is the tolerance in cup fills within which the emptier is allowed to not be greedy.

Of particular interest is the greedy emptier, which is (0,0)-greedy-like, and 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 (T,Δ)-greedy-like emptier (where $T,\Delta\leq O(1)$ 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 $h=2T+1$. Step 1 of our procedure is to achieve high positive tilt in A. Exploiting the fact that the emptier is (T,Δ)-greedy-like, we can transform this positive tilt into a known set of cups with fill at least $h' = T + (1+\Delta)/2$. More specifically, the filling algorithm proceeds as follows:

- **Step 1:** Obtain large positive tilt in A. If at least half of the cups $c\in B$ have $\text{fill}(c)\geq -h$, then we can, with constant probability, achieve a cup with fill h in B – which we will then swap to A – by playing a single-processor cup game on a constant-size ($\lceil e^{2h}\rceil$ suffices) subset of B. On the other hand, if at least half of the cups in B have fill less than $-h$, then the expectation of the positive tilt of a cup chosen randomly is high; this property can be exploited to get high positive tilt in the anchor set.

  Because the filler is oblivious it cannot know which of these states it is in; thus the filler employs a hybrid of these strategies, switching between them with certain probabilities, which works in both cases.

- **Step 2:** Reduce the number of processors to a constant fraction c 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'.

By performing $\Omega(\log n)$ levels of recursion, achieving constant backlog $h'\geq 1/2$ at each step (relative to the average fills), the filler achieves backlog $\Omega(\log n)$.

The strategy fails if fill is extremely concentrated in a very small number of cups; however, in this case the proposition is trivially satisfied. In particular, we call a cup ***overpowered*** if it contains fill at least $h\sqrt{n}/\log\log n$. If there is ever an overpowered cup, then the proposition is trivially satisfied, as backlog is $\Omega(\text{poly}(n))$. Note that the filler doesn'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.

Now we detail how to achieve Step 1. We set apart a constant fraction $\alpha n = n/1000$ of the cups in A, which we call the ***storage-block***, or simply C. Let $\gamma=(\text{lglg}n)^{1/3}$; we will store between $\gamma-\gamma/2$ and $\gamma+\gamma/2$ sets of $2\beta = \alpha n/(\gamma+\gamma/2)$ randomly chosen cups at certain randomly chosen rounds. For each anchor cup c we will perform a procedure called a ***swapping-process***. With probability $\frac{\gamma}{n/2}$ the swapping-processes consists of what we term a ***storing-operation***; in a storing-operation the filler takes a random subset of β cups from B, and a random subset of β cups from $A\setminus C$, and swap these cups into the storage-block. Note that, by a Chernoff bound, the probability that the number of swapping-processes where we perform a storing-operation lies within $\pm\gamma/2$ of its expectation γ is at least $1-2e^{-n(\gamma/2)^2}$, which is better than exponentially small in n, and in particular much better than the probability of success that we need. Thus we assume that the number of swapping-processes that consist of a storing-operation is between $\gamma-\gamma/2$ and $\gamma+\gamma/2$.

The swapping-process does not entail performing a storing-operation with probability $1-\frac{\gamma}{n/2}$. In this case the swapping-process is composed of a substructure, repeated many times, which we call a ***round-block***. A round-block is a set of rounds. At the beginning such a swapping-process we choose a round-block $j\in[n^2]$ uniformly at random from all the round-blocks. The swapping-process proceeds for n^2 round-blocks; on the j-th round-block we swap a cup into the anchor set.

On each of the n^2 round-blocks, the filler selects a random subset $C\subset B$ of the non-anchor cups and plays a single processor cup game on C. In this single-processor cup game the filler employs the classic adaptive strategy for achieving backlog $\Omega(\log|B|)$ on a set of $|B|$ cups, however modified because it is an oblivious filler. In particular, the filler's strategy in the single-processor cup games is to distribute water equally among an ***active set*** of cups, and then after the emptier removes water from some cup the filler removes a random cup from the active set. There is at least constant probability that this results in the active set having a single cup at the end, with fill that has increased by at least $1/|B|+1/(|B|-1)+...+1/1\geq \ln|B|$ since the start of the round-block.

On most round-blocks – all but the j-th – the filler does nothing with the cup that it achieves in the active set at the end of the single processor cup game. However, on the j-th round-block the filler swaps the winner of the single processor cup game into the anchor set.

We consider two cases:

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

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

We now prove that in either case we can achieve high positive tilt in A with good probability.

Claim 2. *Let $q\geq\Omega(1)$ be an appropriately small constant (q is a function of $h\leq O(1)$). In Case 1, with probability at least $1-e^{-nq^2/1024}$, we achieve fill at least h in at least $nq/16$ of the cups in A (i.e. a constant fraction of the cups in A). In particular, this implies that we achieve positive tilt $hnq/16\geq\Omega(n)$ in A.*

Proof. Consider a swapping-process where the filler does not perform a storing-operation where at least $1/2$ of the cups $c \in B$ have $\text{fill}(c) \geq -h$. Note that by assumption there are at least $n/4-3/2\gamma$ such rounds.

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

The emptier must have some binary function, $I(i)$ that indicates whether or not they will neglect the anchor set on round-block i 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-block i depends on this information. However, j is the only parameter of the swapping-process, so there is no other information that the emptier can use to decide whether or not to neglect a round-block, because on any round-block 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 round-blocks, i.e. $\sum_{i=1}^{n^2} I(i) \geq n^2/2$, 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 $n^2/4$ round-blocks ($I(k)$ must be 1 for at least $n^2/4$ of the first $(3/4)n^2$ round-blocks). 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 $(n^2/2)/(n/2) \geq \Omega(n)$ over the entire swapping-process in this case, implying that we achieve the desired backlog.

Otherwise, there is at least a $1/2$ chance that the round-block j, which is chosen uniformly at random from the round-blocks, when the filler performs a swap into the anchor set occurs on a round-block with $I(j)=0$, indicating that the emptier won't neglect the anchor set on round-block j. In this case, the round-block was a legitimate single processor cup game on C_j, the randomly chosen set of $\lceil e^{2h} \rceil$ cups on the j-th round. Then we achieve fill increase $\geq 2h$ by the end of the round-block with probability at least $1/\lceil e^{2h} \rceil!$ – 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 cups that are among the $n/4$ fullest cups in B is

$$\binom{n/2}{\lceil e^{2h} \rceil} \Big/ \binom{n}{\lceil e^{2h} \rceil} = O(1).$$

Note that because, by assumption, at least half of the cups $c \in B$ have $\text{fill}(c) \geq -h$, then the $n/4$ fullest cups in B must have fill at least $-h$. If all cups $c \in C_j$ have $\text{fill}(c) \geq -h$, then the fill of the cup in the active set at the end of the round-block is at least $-h+2h=h$, if the filler guesses the emptier's emptying sequence correctly.

Say that a swapping-process where at least half of the cups $c \in B$ have $\text{fill}(c) \geq -h$ ***succeeds*** if C_j is a subset of the $n/4$ fullest cups in B, and if the filler correctly guesses the emptier's emptying sequence. Note that if a swapping-process succeeds, then the filler is able to swap a cup with fill at least h into A. We have shown that there is a constant probability of a given swapping-process succeeding. Let X_i be the binary random variable indicating whether or not the i-th swapping process where the filler does not perform a storing-operation where at least half of the cups $c \in B$ have $\text{fill}(c) \geq -h$ succeeds. Let $q \geq \Omega(1)$ be the probability of a swapping-process succeeding, i.e. $P(X_i=1)$. Note that the random variables X_i are clearly independent, and identically distributed.

Clearly

$$\mathbb{E}\left[\sum_{i=1}^{n/8} X_i\right] = qn/8.$$

Note that we do not use all the X_i; we know there must be at least $n/4-3/2\gamma$ swapping-processes that do not consist of a storing-operation, but only use $n/8$ of the X_i. We make this choice because the particular constants that we get do not matter, and because it substantially simplifies the analysis. By a Chernoff Bound (i.e. Hoeffding's Inequality applied to binary random variables),

$$P\left(\sum_{i=1}^{n/8} X_i \leq nq/16\right) \leq e^{-nq^2/1024}.$$

That is, the probability that less than $nq/16$ of the anchor cups have fill at least h is exponentially small in n, as desired.

□

Claim 3. *In Case 2, with probability at least $1-1/polylog(n)$, we achieve positive tilt $nh\alpha/96 \geq \Omega(n)$ in the anchor set.*

Proof. Consider a swapping-process on which the filler does a storing-operation and at least half of the cups $c \in B$ have $\text{fill}(c) < -h$.

If the storage-block C already has positive tilt at least $nh/8$ then the filler has already succeeded, as no water ever exits the storage-block.

Otherwise, at least one of A or B must have large positive tilt in order to offset the many cups in B with fill less than h. In particular, the positive tilt of $A \cup B \setminus C$ must be at least $nh/4 - nh/8 = nh/8$. Let D be B if $\text{tilt}(B) > \text{tilt}(A)$ and A otherwise. Note that $\text{tilt}(D) \geq nh/16$. Because there are no overpowered cups, this positive tilt is distributed among many cups. Note that if X is a cup chosen uniformly randomly from D then

$$\mathbb{E}[\text{tilt}(X)] \geq h/8.$$

Let $Y_1, Y_2, \ldots, Y_\beta$ be the positive tilts of a set of β cups drawn from D, with all subsets equally likely; equivalently, consider $Y_1,\ldots,Y_\beta$ to be the positive tilts of β cups sampled from D, sampling without replacement. We apply Hoeffding's Inequality for samples drawn with replacement (stated in Corollary 2) to $\sum_{i=1}^{\beta} Y_i$. Note that $0 \leq Y_i \leq h\sqrt{n}/\text{lglg}n$, as positive tilt is non-negative, and as there are no overpowered cups (without loss of generality).

Thus we have,

$$P\left(\frac{1}{\beta}\sum_{i=1}^{\beta}(Y_i - \mathbb{E}[Y_i]) \leq -h/16\right) \leq \exp\left(-\frac{2\beta(h/16)^2}{(h\sqrt{n}/\text{lglg}n)^2}\right).$$

Simplifying this gives,

$$P\left(\frac{1}{\beta}\sum_{i=1}^{\beta} Y_i \leq h/16\right) \leq \frac{1}{\text{polylog}(n)}.$$

Now we will show that with probability at least $1 - 1/\text{polylog}(n)$ there are at least $\gamma/4$ swapping-processes on which the filler does a storing-operation. Recall that at least $\gamma/2$ storing-operations happen. Choose a random set of $\gamma/2$ of the storing-operations that happen. Note that the distribution of these sets will clearly be identical to the distribution we would get if we simply selected a random subset of $\gamma/2$ of the swapping-process, or equivalently, if we had sampled $\gamma/2$ of the swapping-processes sampling without replacement. Let Z_i be the indicator random variable for whether or not the i-th of these swapping-processes occurs on a round where at least half of the cups $c \in B$ have $\text{fill}(c) < -h$. Clearly $\mathbb{E}\left[\sum_{i=1}^{\gamma/2} Z_i\right] \geq \gamma/4$. Then a Chernoff bound gives that

$$P\left(\sum_{i=1}^{\gamma/2} Z_i \geq \gamma/8\right) \geq 1 - e^{-2\gamma/2(\gamma/8)^2},$$

and recalling that $\gamma = (\text{lglg}n)^{1/3}$ this simplifies to

$$P\left(\sum_{i=1}^{\gamma/2} Z_i \geq \gamma/8\right) \geq 1 - \frac{1}{\text{polylog}(n)}.$$

Combining these, with probability at least

$$\left(1 - \frac{1}{\text{polylog}(n)}\right)\left(1 - \frac{1}{\text{polylog}(n)}\right) = 1 - \frac{1}{\text{polylog}(n)}$$

we achieve positive tilt at least $\beta h/16$ in at least $\gamma/8$ sets of β cups. In total, this means that means we have positive tilt $\beta\gamma h/128 = \frac{\alpha n/2}{(3/2)\gamma}\gamma h/128 = n\alpha h/96 \geq \Omega(n)$ in the anchor set.

□

By Claim 2 and Claim 3, in both Case 1 and Case 2, we achieve, with probability at least $1 - 1/\text{polylog}n$, the filler achieves positive tilt at least nc_0 in the anchor set for some constant c_0 (which is a function of h).

Using the positive tilt, setting the number of processors to 1, the filler can obtain high fill in a set of nc *known* cups. In particular, the filler chooses a set of nc cups randomly, which by Hoeffding's Inequality will have average fill close to 0, and then the filler distributes a unit of water equally among these cups. The filler continues until the average fill of that set of cups has increased by h'. The filler uses one processor because it doesn't know how many cups the positive tilt is concentrated in. Then the filler recurses on the set of nc known cups with average fill.

Note that this transformation from a set with high positive tilt to a set of known cups with high positive average fill is the only part of the proof of Proposition 2 that is specific to a greedy-like emptier. Against a general emptier it is not true that the emptier will necessarily focus on the set of cups with high positive tilt; an arbitrary emptier can of course foil our attempts to achieve high fill in any fixed set of p cups, at a given setting of p. Extending Proposition 2 to apply to non-greedy-like emptiers is an important open question. □

Lemma 3 (The Oblivious Amplification Lemma)**.** *Let f be an oblivious filling strategy that achieves backlog $f(n)$ in the variable-processor cup game on n cups with constant probability (relative to average fill, with negative fill allowed). Let $\delta \in (0,1)$ be a parameter. Then, there exists an adaptive filling strategy that, with constant probability, either achieves backlog*

$$f'(n) \geq (1-\delta)(f((1-\delta)n) + f((1-\delta)\delta n))$$

or achieves backlog at least $\Omega(poly(n))$ in the variable processor cup game on n cups.

Proof. Note that the statement of Lemma 3 is very similar to the statement of Lemma 1. We have made a few simplifications, such as not allowing more than 1 "level of recursion", for simplifying the proof, at no cost to the final backlog that we will derive. The more major deviation from the Adaptive Amplification Lemma, is of course that the filler is oblivious in this case. This necessitates working with functions that only succeed at achieving the desired backlog with certain probabilities. Nevertheless, the proof is remarkably similar to the proof of Lemma 1. We now establish the Lemma.

First we establish some important facts.

Again, we call a cup ***overpowered*** if it has fill at least $h\sqrt{n}/\lg\lg n$. As in Proposition 2, without loss of generality there are no overpowered cups, because existence of an overpowered cup means that we already have poly(n) backlog.

Let a cup be ***verysad*** if it has fill $< -h\sqrt{n}/\mathrm{lglg}n$.

Claim 4. *Without loss of generality there are no verysad cups.*

Proof. First note that because 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 5. *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 $\geq nh/\mathrm{lglg}n$, 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.

Initial 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 6. *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 3. *There is an oblivious filling strategy for the variable-processor cup game on n cups that achieves backlog at least $2^{\Omega(\sqrt{\log n})}$ in running time $O(n)$ with constant probability.*

Proof. Given the Oblivious Amplification Lemma we could try to apply the same strategy as outlined in the proof of Corollary 1 to achieve backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ in time $2^{O(\log^2 n)}$. Because of our definition of an overpowered cup as a cup with fill at least $\tilde{\Omega}(\sqrt{n})$, we can't get quite as close to linear as an adaptive filer could. However, the more pressing problem is that of running time: randomized algorithms are traditionally supposed to have polynomial-running time. By artificially reducing n, i.e. ignoring some portion of the cups, we can get an algorithm that achieves high backlog, but in polynomial time. In particular, we want to choose a subset of n' of the cups to focus on, where $2^{O(\log^2 n')} = O(n)$. An appropriate choice is $n' = 2^{\sqrt{\log n}}$.

With n' chosen, we apply the exact same strategy as given in the paragraph in the proof of Corollary on the $2^{O(\log^2 n)}$-time construction for achieving backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$, 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. Thus, 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 $O(n)$ and backlog $\Omega(2^{\sqrt{\log n}})$. □

6 Conclusion

Many important open questions remain open.

References

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

[2] William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. *SIAM*, 2020.