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 p-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 show that it drastically alters the game.

We construct an adaptive filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ of our choice in running time $2^{\log^2 n}$. This is enormous compared to the upper bound of $O(\log n)$ that holds in the classic p-processor cup game! We also present a simple adaptive filling strategy that is able to actually achieve backlog $\Omega(n)$ in extremely long games: it has running time $2^{O(n)}$.

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 oblivious filling strategies. We are able to derive exactly the same lower bound as in the adaptive case for games of length $2^{O(\log^2 n)}$: namely an oblivious filler can achieve backlog $\Omega(n^{1-\epsilon})$ for $\epsilon > 0$ a constant of our choice, with probability at least $1 - 2^{-\operatorname{polylog}(n)}$. This result only holds against a certain class of emptiers however: "greedy-like" emptiers. A lower bound against greedy-like emptiers is very interesting; for instance, greedy-like emptiers are used in the upper bound proof.

*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 at most 1 unit of water from each of p cups[1].

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. In analyzing the cup game we aim to prove upper and lower bounds on backlog.

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

[1] Note that negative fill is not allowed, so if the emptier empties from a cup with fill below 1 that cup's fill becomes 0.

$O(\log\log n + \log p)$ 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 classic p-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. This restriction 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[2]. Furthermore, at any fixed value of p upper bounds have been proven. For instance, regardless of p a greedy emptier prevents an adaptive filler from having backlog greater than $O(\log n)$. Switching between different values of p, all of which the filler cannot individually get backlog larger than $O(\log n)$ is not obviously going to help the filler achieve larger backlog. 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 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 with an adaptive filler. Theorem 1 states that a filler can achieve backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ in quasi-polynomial running time. Theorem 1 also provides an extremal strategy that achieves backlog $\Omega(n)$ in incredibly long games: it has exponential running time.

In Section 4 we prove a novel invariant maintained by the greedy emptier. In particular Theorem 2 establishes that a greedy emptier keeps the average fill of the k fullest cups at most $2n-k$. In particular this implies (setting $k=1$) that a greedy emptier prevents backlog from exceeding $O(n)$.

The lower bound and upper bound agree; our analysis is tight for adaptive fillers!

[2] 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.

In Section 5 we prove a lower bound on backlog with an oblivious filler. Theorem 4 states that an oblivious filler can achieve backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ in quasi-polynomial time with probability at least $1 - 2^{-\operatorname{polylog}(n)}$. Theorem 4 only applies to a certain class of emptiers: "greedy-like emptiers". Nonetheless, this class of emptiers is very interesting; it contains the emptiers that are used in upper bound proofs. It is shocking that randomization doesn't help the emptier in this game; usually the lower bound for backlog in the randomized version of a cup game is log of the backlog in the deterministic setting!

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 at most 1 unit of water from. Note that if the fill of a cup that the emptier empties from is less than 1 the emptier reduces the fill of this cup to 0 by emptying from it; we say that the emptier ***zeros out*** a cup if it empties from a cup with fill less than 1. Note that whenever the emptier zeros out a cup the emptier has removed less fill than the filler has added; hence the average fill will increase. 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 $\text{tilt}(c) = \max(0, \text{fill}(c))$. Let the ***mass*** of a set of cups X be $m(X) = \sum_{c \in X} \text{fill}(c)$. Let the positive tilt of a set X of cups be $\text{tilt}(X) = \sum_{c\in X} \text{tilt}(c)$. 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.

Many of our lower bound proofs will adopt the convention of allowing for negative fill. We call this the ***negative-fill variable-processor cup game***. Specifically, in the negative-fill variable-processor cup game, when the emptier empties from a cup its fill always decreases by exactly 1: there is no zeroing out. Negative-fill can be interpreted as fill below some average fill. Measuring fill like this is important however, as our lower bound results are used recursively, building on the average fill already achieved. Note that it is strictly easier for the filler to achieve high backlog when cups can zero out, because then some of the emptiers resources are wasted. On the other hand, during the upper bound proof we show that a greedy emptier maintains the desired invariants even if cups zero out. This is crucial as the game is harder for the emptier when cups can zero out.

3 Adaptive Filler Lower Bound

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

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. Let A consist of the $n/2$ fullest cups, and B consist of the rest of the cups. Note that 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 c with $\text{fill}(c) \leq -2h$. The mass of those cups would be at most $-hn$ but the mass of the remaining cups cannot exceed $hn/2$. This implies that average fill is negative when it is in fact 0 by definition, a contradiction. Hence there are at most $n/2$ cups c with $\text{fill}(c) \leq -2h$.

The filler sets the number of processors equal to 1 and plays a single processor cup game on $n/2$ cups that each have fill at least $-2h$ (which must exist) for $n/2 - 1$ steps. Throughout this process the filler maintains a set of cups called the ***active set***: the set of cups that the filler will place fill in. The filler initializes the active set to be A. Note that that $\text{fill}(c) \geq -2h$ for all cups $c \in A$, as A consists of the $n/2$ fullest cups. The filler removes 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 $1/(n/2) + 1/(n/2 - 1) + \cdots + 1/2 + 1/1 \geq \ln n/2 = 4h$. This cup's fill started as at least $-2h$. Thus this cup has fill at least $2h$ now, as desired. □

Lemma 1 (The Adaptive Amplification Lemma)**.** *Let f be an adaptive filling strategy that achieves backlog $f(n)$ in the negative-fill variable-processor cup game on n cups. Let $\delta \in (0, 1)$ be a parameter, and let $L \in \mathbb{N}$ be a parameter.* [3]

Then, there exists an adaptive filling strategy that achieves backlog

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

in the negative-fill variable-processor cup game on n cups.

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

- **Step 1:** Using f repeatedly, achieve average fill at least $(1-\delta)f(n(1-\delta))$ in a set of $n\delta$ cups.
- **Step 2:** Reduce the number of processors to $n\delta$, and 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. The filling strategy always places 1 unit of water in each anchor cup. This ensures that no cups in the anchor set ever have their fill decrease.

On the ℓ-th level of recursion the filler uses the following procedure, termed a ***swapping-process***, to achieve the desired average fill in A: repeatedly apply f to B, and then take the cup generated by f within B with fill at least $f(|B|)$ 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 $-h_\ell \cdot \delta/(1-\delta)$ the filler will applies f to B.

It is somewhat complicated to apply f to B however, because we must guarantee that the emptier removes the same mass from B as the filler adds on all rounds during the filler's application of f to B. This might not be the case if the emptier does not empty from each anchor cup at each step. We 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.

The filler applies f to B at most $h_\ell n_\ell \delta + 1$ times. At the end of an application of f the filler swaps the generated cup into A only if the emptier has not neglected the anchor set during this 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. In this case the filler achieves fill
$-h_\ell \cdot \delta/(1-\delta) + f(n_\ell(1-\delta)) = (1-\delta)f(n_\ell(1-\delta)) = h_\ell$
in some non-anchor cup, and swaps it with the least full cup in the anchor set.

The filler achieves average fill h_ℓ in the anchor set for L levels of recursion. Summing h_ℓ for $0 \leq \ell \leq L$ yields the desired backlog.

□

Theorem 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 $\epsilon > 0$ of our choice in running time $2^{\log^2 n}$.*

[3] Note that n must be sufficiently large, and δ, L must be chosen such that $(1-\delta)\delta^L n \in \mathbb{N}$ because it doesn't make sense to talk of "fractional cups".

There is also an adaptive filling strategy for achieving backlog $\Omega(n)$ in running-time $2^{O(n)}$.

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 a function f 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:

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

for appropriate choice of L. We will choose δ to be very small, so $\sum_{\ell=0}^{L}(\delta^L)^{1-\epsilon}$ is well approximated by $1+\delta^{1-\epsilon}$, and thus we don't lose much by relaxing our lower bound on $f'(k_0)$ to only using $L = 1$ (i.e. taking two terms of the sum). We have the bound

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

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 $h(\delta) \geq 1$ and large enough such that $(1-\delta)\delta n \geq n_0$, 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 $\delta = \Theta(1/n)$, and if ϵ is chosen to be a positive constant then we will choose $\delta = \Theta(1)$.

Note that in order for $L = 1$ to make sense it must be that $n(1-\delta)\delta n \geq n_0$, or else this term from the sum would be contributing essentially 0 backlog to the sum. If $L = 1$ and 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. [4] Thus by repeatedly amplifying a function, we should be able to arbitrarily extend the region where the function satisfies the desired inequality, which will allow us to attain the desired backlog. We now prove Claim 1.

Proof. First we show that making $h(\delta) > 1$ is possible. 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 we choose to neglect the δ^2 term as it does not strengthen the lower bound substantially – 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 stronger inequality on $\delta^{1-\epsilon}$ by overestimating $\delta^{2-\epsilon}$ as δ. Then, δ satisfying

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

will make $h(\delta) \geq 1$.

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 can have at least two terms, i.e. such that $L \geq 1$. We need $L \geq 1$ because otherwise the Amplification Lemma doesn't give a larger function. That is, we want

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

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

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

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

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

Thus, 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 for our choice of ϵ. When proving that we can achieve backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ we choose $\delta > 0$ to be a constant satisfying Inequality 1, and δ, being constant, trivially 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 to show that an algorithm achieving backlog $\Omega(n^{1-\epsilon})$ for ϵ constant exists. Next we modify the existential proof to achieve an algorithm achieving the same backlog, but in bounded running time, specifically running time $2^{O(\log^2 n)}$. Finally we present an algorithm

[4] 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 $\mathbb{N}$. In general we have to be somewhat careful about the fact that there are an integer number of cups throughout this proof (this issue was deferred from earlier proofs to be dealt with here).

where δ is set extremally; doing so, along with other modifications of the approach, we can achieve backlog $\Omega(n)$ in running time $2^{O(n)}$.

Proposition 2. *There exists a filling algorithm that achieves backlog $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$.*

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

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

Note that $n_*(1-\delta)\delta \geq n_0/(1-\delta)$ by appropriate choice of constant c, and Proposition 1, which states that we can get backlog $\Omega(\log n_*)$ on n_* cups[5]. Because f^* satisfies the Amplification Lemma we have:

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

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

Proposition 3. *There is a filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ in time $2^{O(\log^2 n)}$.*

Proof. 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 with

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

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:

Claim 2.

$$f_i(k) \geq ck^{1-\epsilon} \textit{ for all } k < g_i. \tag{5}$$

Proof. We prove Claim 2 by induction on i. Claim 2 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 $k < g_0$.[6] As our inductive hypothesis we assume Claim 2 for f_i; we aim to show that Claim 2 holds for f_{i+1}. Note the key property of g_i, that $g_{i+1} \cdot (1-\delta) < g_i$. Also note that (at least 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 $k < g_{i+1}$,

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

as desired. □

Note that $g_{i+1} \geq g_i + 1$ so by continuing this process we eventually reach some f_{i_*} such that $f_{i_*}(n) \geq cn^{1-\epsilon}$; trivially $i_* \leq n$. In fact we can show $i_* \leq O(\log n)$. The recurrence is almost simply the geometric sequence $g_i = g_{i-1}/(1-\delta)$ with $g_0 = 1/\delta$, for which it is trivially true that $g_i = 1/\delta(1/(1-\delta))^i$, hence we would have $g_{\log_{1/(1-\delta)} n} = n \cdot 1/\delta > n$ as desired. The actual sequence is not quite this large. The sequence is lower bounded by the sequence g_i' defined as $g_0' = 1/\delta, g_i' = g_{i-1}'/(1-\delta) - 1$. Even the sequence g_i' grows exponentially though, hence only a logarithmic recursion depth is necessary. In particular, g_i' satisfies

$$g_i' = \frac{1/\delta}{(1-\delta)^i} - \frac{1/(1-\delta)^{i-1} - 1}{1/(1-\delta) - 1},$$

which simplifies to

$$g_i' = \frac{1}{(1-\delta)^i} + \frac{1-\delta}{\delta}.$$

Thus $g'_{\log_{1/(1-\delta)} n}$, and hence also $g_{\log_{1/(1-\delta)} n}$ both are at least n, as desired.

However, we have not addressed the issue that the number of cups must be an integer! Although it is relatively clear that this will not affect the asymptotic results, we briefly carefully address this here. Luckily, by careful choice of δ and a slight (constant factor) reduction of n we can get the argument to work. In particular, let ψ be the smallest natural such that $1/2^\psi < \delta$; define $\delta' = 1/2^\psi$. Let τ be the integer such that $2^{\tau\psi} \leq n < 2^\psi 2^{\tau\psi}$; define $n' = 2^{\tau\psi}$. Note that $O(n') = O(n)$ as n' is at worst a factor of 2^ψ less than n, but δ being constant implies that ψ is also constant. Note that

$$\tau = \left\lfloor \frac{\lg n}{\psi} \right\rfloor \geq \Omega(\log n).$$

Thus, we have that $n'(\delta')^{i_1}(1-\delta')^{i_2} \in \mathbb{N}$ for any integers i_1, i_2 satisfying $i_1 + i_2 \leq \tau$, which means that

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

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

$i_1, i_2 \geq \Omega(\log n)$ is possible. Now we can genuinely guarantee that throughout the levels of recursive application of the Amplification Lemma – of which there will be at most $O(\log n)$– the number of cups is always an integer.

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 need the terms of the sum where $\ell = 0, 1$, so we can take $L = 1$. Thus we have the following (loose) recurrence bounding $T(n)$:

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

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

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

as desired.

□

Proposition 4. *There is a filling strategy that achieves backlog $\Omega(n)$ in time $2^{O(n)}$.*

Proof. We now describe an algorithm f that achieves backlog

$$f(k) = \begin{cases} \left(1 - \frac{n_0}{k}\right) f(k - n_0) + 1, & k \geq 2n_0 \\ 1, & n_0 \leq k < 2n_0 \\ 0, & k < n_0. \end{cases}$$

Clearly our strategy is possible for $k < 2n_0$ by Proposition 6. Assuming that our algorithm works for $k < mn_0$ we can extend it to work for $k < (m+1)n_0$ by considering the amplification of our algorithm using $\delta = n_0/n$. By induction our algorithm thus works for all n.

Expanding the recurrence, we find that our algorithm achieves backlog

$$f(n) = 1 + 1 - \frac{n_0}{n} + 1 - \frac{n_0}{n - n_0} + 1 - \frac{n_0}{n - 2n_0} + \cdots$$

Clearly

$$f(n) \geq n - n_0 \log n \geq \Omega(n).$$

The recurrence for running time is

$$T(n) = n_0 T(n - n_0) + O(1).$$

Clearly $T(n) = 2^{O(n)}$.

This algorithm can be interpreted very simply. To achieve large backlog on n cups we create an anchor set A of n_0 cups and a set B of $n - n_0$ cups; We recursively apply our strategy to B for each cup in A. In order for the average fill difference between A and B to be $f(n - n_0)$, $\mu(A)$ must rise by $\frac{n-n_0}{n}$ of this difference whereas $\mu(B)$ must sink by $\frac{n_0}{n}$ of this difference. Hence we achieve average fill $\frac{n-n_0}{n} f(n - n_0)$ in A. Then, using the strategy from Proposition 1 we can achieve backlog 1 on these cups. The Amplification Lemma makes this argument rigorous.

□

□

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 $S = I_t$).[7] Let $S(\{r_1, \ldots, r_m\})$ denote the cups of ranks $r_1, r_2, \ldots, r_m$ at state S. Let $[n] = \{1, 2, \ldots, n\}$, let $i + [n] = \{i+1, i+2, \ldots, i+n\}$. We will use concatenation of sets to denote unions, i.e. $AB = A \cup B$. The ***greedy emptier*** always empties from the p fullest cups.
We establish the following Theorem:

Theorem 2. *In the variable-processor cup game on n cups, the greedy emptier maintains the invariant*

$$\mu_{S_t}(S_t([k])) \leq 2n - k \text{ for all } t \geq 1, k \in [n].$$

In particular, for $k = 1$, this implies that the greedy emptier never lets backlog exceed $O(n)$.

Proof. We 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 $\mu_{S_1}(S_1([k])) = 0 \leq 2n - k$ for all $k \in [n]$.

Fix a round $t \geq 1$, and any $k \in [n]$. 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

$$\mu_{S_{t+1}}(S_{t+1}([k])) \leq 2n - k.$$

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 $k - a \geq 0$ as $a + b \leq k$, and also $|ABC| = k + b \leq n$, because by definition the b cups in B must not be among the k fullest cups in state S_{t+1} so there are at least $k + b$ cups. 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 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.

We now establish the following claim, which we call the ***interchangeability of cups***:

Claim 3. *Without loss of generality*

$$S_t([r]) = I_t([r])$$

[7] 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.

for any rank $r \in [n]$.

Proof. Say there are cups x, y with $x \in S_t([r]) \setminus I_t([r]), y \in I_t([r]) \setminus S_t([r])$. 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. This is valid as both fill amounts are at most 1: $f_x - f_y + \Delta_x < \Delta_y \leq 1$ and $f_y - f_x + \Delta_x < \Delta_x \leq 1$.

We can repeatedly apply this process to swap each cup in $I_t([r]) \setminus S_t([r])$ into being one of the r 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. □

Now we proceed with the proof of the theorem.

Claim 4. *If some cup in A zeros out then invariant holds.*

Proof. Say a cup in A zeros out. Then

$$m_{S_{t+1}}(A) \leq (a-1)(2n-(a-1))$$

because the $a-1$ fullest cups must have satisfied the invariant on round S_t, and $\text{fill}_{S_{t+1}}(I_{t+1}(a)) = 0$. Furthermore, the fill of all cups in C must be at most 1 in I_t to be less than the fill of the cup in A that zeroed out.

Thus,

$$m_{S_{t+1}}(S_{t+1}([k])) \leq (a-1)(2n-(a-1)) + k - a.$$

We claim that this is less than $a(2n-a)$. This follows from simple manipulation of the above expression:

$$\begin{aligned}
&(a-1)(2n-(a-1)) + k - a \\
&= a(2n-a) + a - 2n + a - 1 + k - a \\
&= a(2n-a) + (k-n) + (a-n) - 1 \\
&< a(2n-a)
\end{aligned}$$

as desired.

As k increases from 1 to n, $k(2n-k)$ strictly increases (it is a quadratic in k that achieves its maximal value at $k = n$). Thus $a(2n-a) \leq k(2n-k)$ because $a \leq k$.

And we have

$$m_{S_{t+1}}(S_{t+1}([k])) \leq k(2n-k).$$

□

Claim 5. *If no cups in A zero out and $b = 0$ the invariant holds.*

Proof. 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. The filler cannot have added more than k fill to these cups, because it can add at most 1 fill to any given cup. Also, the filler cannot have added more than p_t fill to the cups because this is the total amount of fill that the filler is allowed to add. Hence the filler adds at most $\min(p_t, k) = a + b = a + 0 = a$ fill to these cups. Thus the invariant holds:

$$m_{S_{t+1}}(S_{t+1}([k])) \leq m_{S_t}(S_t([k])) + a - a \leq k(2n-k).$$

□

Claim 6. *If no cups in A zero out and $b > 0$ the invariant holds.*

Proof. Because $b > 0$ and $a + b \leq k$ 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:

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

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

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

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

$$m_{S_t}(AC) \leq m_{S_t}(A) + \frac{c}{b+c} m_{S_t}(BC) \tag{6}$$

where

$$\begin{aligned}
&m_{S_t}(A) + \frac{c}{b+c} m_{S_t}(BC) \\
&= \frac{c}{b+c} m_{S_t}(ABC) + \frac{b}{b+c} m_{S_t}(A).
\end{aligned} \tag{7}$$

Note that the expression in Equation 7 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 $(2n - |ABC|, 2n - |A|)$ 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 $\mu_F(A) = 2n - |A|$ and $\mu_F(ABC) = 2n - |ABC|$, 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 $\mu_F(ABC) = 2n - |ABC|$. 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 $\mu_F(A) = 2n - |A|$, the cups in A must have been increased by $|BC|$ from their previous fill of $2n - |ABC|$. To equalize an increase in $\mu_F(A)$ of $|BC|$, we need a corresponding decrease in $\mu_F(BC)$ by $|A|$. That is,

$$\mu_F(BC) = 2n - |ABC| - |A|.$$

Thus we have the following bound:

$$\begin{aligned} m_{S_t}(A) + m_{S_t}(C) & \\ &\leq m_F(A) + c\mu_F(BC) \qquad (*) \\ &\leq a(2n-a) + c(2n - |ABC| - a) \\ &\leq (a+c)(2n-a) - c(a+c+b) \\ &\leq (a+c)(2n-a-c) - cb, \end{aligned}$$

where (*) follows from Equation 7.

Consider a new configuration of fills F achieved by starting with state S_t, and moving water from BC into A until $\mu_F(A) = 2n - |A|$. [8] 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 6. 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

$$\mu_F(BC) \leq \mu_F(ABC) - |A| \leq 2n - |ABC| - |A|.$$

Thus we have the following bound:

$$\begin{aligned} m_{S_t}(A) + m_{S_t}(C) & \\ &\leq m_F(A) + c\mu_F(BC) \\ &\leq a(2n-a) + c(2n - |ABC| - a) \\ &\leq (a+c)(2n-a) - c(a+c+b) \\ &\leq (a+c)(2n-a-c) - cb. \end{aligned}$$

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

$$m_{S_t}(A) + m_{S_t}(C) \leq k(2n-k) - b$$

So

$$m_{S_t}(A) + m_{S_t}(C) + b \leq k(2n-k).$$

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.

□

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

We have shown the inductive hypothesis 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 $t \in \mathbb{N}$. □

5 Oblivious Filler Lower Bound

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

Theorem 3 (Hoeffding's Inequality)**.** *Let X_i for $i = 1, 2, \dots, k$ be independent bounded random variables with $X_i \in [a, b]$ for all i. Then,*

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

Let S be a finite population, let X_i for $i = 1, 2 \dots, k$ be chosen uniformly at random from $S \setminus \{X_1, \dots, X_{i-1}\}$, and let Y_i for $i = 1, 2, \dots, k$ be chosen uniformly at random from S. Note that $\{X_1, \dots, X_k\}$ represents a sample of S chosen without replacement, whereas $\{Y_1, \dots, Y_k\}$ 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 1. *Let S be a finite set with $\min(S) \geq a, \max(S) \leq b$, and let X_i for $i = 1, 2 \dots, k$ be chosen uniformly at random from $S \setminus \{X_1, \dots, X_{i-1}\}$. Then*

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

Hoeffding proved Corollary 1 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 1 is that samples drawn without replacement should be more tightly concentrated around the mean than samples drawn with replacement.

Another important, yet very trivial, corollary of Hoeffding's Inequality is the Chernoff Bound (i.e. Hoeffding's Inequality applied to binary random variables):

Corollary 2. *Let X_i for $i = 1, 2, \dots, k$ be independent identically distributed binary random variables (i.e. $X_i \in \{0, 1\}$). Then*

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

We proceed with our analysis of oblivious lower bounds.

Call a cup configuration M-***flat*** if the fill of every cup is in the interval $[-M, M]$.

We call an emptier Δ-***greedy-like*** if, when there are two cups c_1, c_2 with fills satisfying $\text{fill}(c_1) > \text{fill}(c_2)+\Delta$ the emptier never empties from c_2 without emptying from c_1 on the same round. Intuitively, a Δ-greedy-like emptier has a $\pm\Delta$ range where it is allowed to "not be greedy". Note that a perfectly greedy emptier is 0-greedy-like. We call an emptier ***greedy-like*** if it is Δ-greedy-like for $\Delta \le O(1)$.

In the randomized setting we are only able to prove lower bounds for backlog against greedy-like emptiers; whether or not our results can be extended to a more general class of emptiers is an interesting open question.

We now prove a crucial property of greedy-like emptiers: that they are ***flattenable***, i.e.:

Proposition 5. *Given a cup configuration that is M-flat, an oblivious filler can, in running time $2M$, achieve a $2(2+\Delta)$-flat configuration of cups against a Δ-greedy-like emptier, in the negative-fill variable-processor cup game on n cups.*

Proof. The filler sets $p = n/2$ and distributes fill equally amongst all cups at every round, in particular placing $1/2$ units of water in each cup. Let $\ell_t = \min_{c \in S_t} \text{fill}_{S_t}(c)$, $u_t = \max_{c\in S_t} \text{fill}_{S_t}(c)$. Let L_t be the set of cups c with $\text{fill}_{S_t}(c) \le l_t + 2 + \Delta$, and let U_t be the set of cups c with $\text{fill}_{S_t}(c) \ge u_t - 2 - \Delta$.

There are two ways to think of U_t. First we can consider U_t as capturing cups in the union of intervals of length 1, Δ, and 1. Note the key property that if a cup with fill in $[u_t - \Delta - 2, u_t - \Delta - 1]$ is emptied from, then all cups with fills in $[u_t - 1, u_t]$ must be emptied from, because the emptier is Δ-greedy-like. On the other hand, we can consider U_t as capturing cups with fill in the union of $[u_t - 2, u_t]$ and $[u_t - \Delta - 2, u_t - 2]$. This is useful as the interval of width Δ serves as a "buffer". In particular, if there are more than $n/2$ cups outside of U_t then all cups in $[u_t - 2, u_t]$ must be emptied from because the emptier is Δ-greedy-like. L_t is of course completely symmetric to U_t.

First we prove a key property of the sets U_t and L_t: once a cup is in U_t or L_t it is always in $U_{t'}, L_{t'}$ for all $t' > t$. This follows immediately from the following claim:

Claim 7.

$$U_t \subseteq U_{t+1}, L_t \subseteq L_{t+1}.$$

Proof. Consider a cup $c \in U_t$.

If c is not emptied from, i.e. $\text{fill}(c)$ has increased by $1/2$, then clearly $c \in U_{t+1}$, because backlog has increased by at most $1/2$, so the fill of c must still be within $2+\Delta$ of the backlog on round $t+1$.

On the other hand, if c is emptied from, i.e. $\text{fill}(c)$ has decreased by $1/2$, we consider two cases.

Case 1: If $\text{fill}_{S_t}(c) \ge u_t - \Delta - 1$, then, as $u_{t+1} \le u_t + 1/2$,

$$\text{fill}_{S_{t+1}}(c) \ge u_t - \Delta - 1 - 1/2 \ge u_{t+1} - \Delta - 2.$$

Case 2: On the other hand, if $\text{fill}_{S_t}(c) < u_t - \Delta - 1$, then every cup with fill in $[u_t - 1, u_t]$ must have been emptied from. The fullest cup at round $t+1$ is the same as the fullest cup on round t, because the fills of all cups with fill in $[u_t - 1, u_t]$ have decreased by $1/2$, and no cup with fill less than $u_t - 1$ had fill increase by more than $1/2$. Hence $u_{t+1} = u_t - 1/2$. Because both the fill of c and the backlog have decreased by the same amount, the distance between them is still at most $\Delta + 2$, hence $c \in U_{t+1}$. The argument for $L_t \subseteq L_{t+1}$ is essentially identical. $\square$

Now that we have shown that L_t and U_t never lose cups, we will show that they eventually gain a substantial number of cups.

Claim 8. *As long as $|U_t| \le n/2$ we have $u_{t+1} = u_t - 1/2$. Identically, as long as $|L_t| \le n/2$ we have $\ell_{t+1} = \ell_t + 1/2$.*

Proof. If there are more than $n/2$ cups outside of U_t then there must be some cup with fill less than $u_t - \Delta - 2$ that is emptied from. Because the emptier is Δ-greedy-like this means that the emptier must empty from every cup with fill at least $u_t - 2$. Thus $u_{t+1} = u_t - 1/2$: no cup with fill less than $u_t - 2$ could have become the fullest cup, and the previous fullest cup has lost $1/2$ units of fill.

The proof is identical for L_t. $\square$

By Claim 8 we see that both $|U_t|$ and $|L_t|$ must eventually exceed $n/2$ at some times $t_u, t_\ell \le 2M$, by the assumption that the initial configuration is M-flat. Since by Claim 7 $|U_{t+1}| \ge |U_t|$ and $|L_{t+1}| \ge |L_t|$ we have that there is some round $t_0 = \max(t_u, t_\ell) \le 2M$ on which both $|U_{t_0}|$ and $|L_{t_0}|$ exceed $n/2$. Then $U_{t_0} \cap L_{t_0} \ne \varnothing$. Furthermore, the sets must intersect for all $t_0 \le t \le 2M$. In order for the sets to intersect it must be that the intervals $[u_t - 2 - \Delta, u_t]$ and $[\ell_t, \ell_t + 2 + \Delta]$ intersect. Hence we have that

$$\ell_t + 2 + \Delta \ge u_t - 2 - \Delta.$$

Since $u_t \ge 0$ and $\ell_t \le 0$ this implies that all cups have fill in $[-2(2+\Delta), 2(2+\Delta)]$.

$\square$

Given a Δ-greedy-like filler, let $R_\Delta = 2(2+\Delta)$. By Proposition 5, if a filler is given a M-flat configuration of cups they can achieve a R_Δ-flat configuration of cups.

Now we are equipped to prove the following proposition:

Proposition 6. *Let $H \le O(1)$, $M \le \text{poly}(n)$, $\Delta \le O(1)$, $n \ge \Omega(1)$ at least a sufficiently large constant.*

There exists an oblivious filling strategy for the negative-fill variable-processor cup game on n cups that can achieve backlog H on a M-flat configuration of cups in running time poly(n) against a Δ-greedy-like emptier, with probability at least

$$1 - 2^{-\Omega(n)}.$$

Proof. The filler starts by flattening all the cups, using the flattening procedure detailed in Proposition 5.

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 $h = 16\Delta + 16$, and let $h' = 2$. Note that the average fill of A and B both must start as at least $-R_\Delta$ due to the flattening.

The filler's strategy is roughly as follows:

- **Step 1:** Make a constant fraction of the cups in A have fill at least h by playing single processor cup games on constant-size subsets of B and then swapping the cup within B that has high fill, with constant probability, into A. By a Chernoff bound this makes a constant fraction of A, say nc cups, have fill at least h with exponentially good probability. Between single-processor cup games the filler flattens B.
- **Step 2:** Reduce the number of processors to nc, and raise the fill of nc *known* cups to fill h'. The emptier must first empty from the cups with fill h before emptying from the cups that the filler is attempting to get fill h' in.
- **Step 3:** Recurse on the nc cups that are known to have fill at least h'.

To achieve Step 1 the filler performs a series of ***swapping-process***, which are procedures that the filler uses to get a new cup –hopefully with high fill– in A. A swapping-process is composed of a substructure, repeated many times, which we call a ***round-block***; a round-block is a set of rounds. A swapping-process will consists of $n \cdot c_\Delta$ round-blocks ($c_\Delta = \Theta(1)$ a function of Δ to be specified); at the beginning of each swapping-process the filler chooses a round-blocks j uniformly at random from $[n \cdot c_\Delta]$.

For each round-block $i \in [n \cdot c_\Delta]$, the filler selects a random subset $D_i \subset B$ of the non-anchor cups and plays a single processor cup game on D_i. In this single-processor cup game the filler essentially employs the classic adaptive strategy for achieving backlog $\Omega(\log |B|)$ on a set of $|B|$ cups, with slight modifications for the fact that it is oblivious. In particular, the filler will only achieve this fill with constant probability. While doing this, the filler always places 1 unit of fill in each cup in the anchor set. Note that the filler sets $p = n/2 + 1$.

At the end of each round-block the filler applies the flattening procedure to flatten the non-anchor set. Note that this will not affect the running-time beyond a multiplicative factor (of e.g. 3).

On most round-blocks – all but the j-th – the filler does nothing with the cup that it achieves with constant probability in its 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 (with constant probability there is a winner).

Claim 9. *With probability at least $1 - 2^{-\Omega(n)}$, the filler achieves fill at least h in at least $nc = \Theta(n)$ of the cups in A.*

Proof. Consider a particular swapping-process. Let j, the round-block on which the filler will perform the swap, be chosen uniformly randomly from $[n \cdot c_\Delta]$ (c_Δ to be determined).

Say the emptier ***neglects*** the anchor set during a round-block if on at least one round of the round-block the emptier does not empty from every cup in the anchor set. By playing the single-processor cup game for many round-blocks with only one round-block when the filler actually swaps a cup into the anchor set, the filler prevents the emptier from neglecting the anchor set too often.

The fill of any cup in the anchor set can clearly never exceed $R_\Delta + \Delta$ because B is R_Δ-flat at the start of each round-block (a cup with fill this high would necessarily be emptied from). Let $\mu_\Delta = 2R_\Delta + \Delta$; the emptier can neglect the anchor set no more than $(n/2)\mu_\Delta$ times. might need to mess with the size of A and B a bit to make this more true... Furthermore, the average fill of B is thus always at least $-\mu_\Delta$. actually this isn't quite true, the average fill sinks a (very) little bit (a very small constant). As B is R_Δ-flat this also means that the fills of cups in B at the start of each round-block are at least $-\mu_\Delta - R_\Delta$.

On each round-block the filler chooses a random subset $D_i \subset B$ of $\lceil e^{2h} \rceil$ cups. If the emptier does not neglect the anchor set on round-block i then the filler plays a legitimate single-processor cup game on n cups. The filler maintains an ***active-set*** of cups, which is a subset of D_i initialized to D_i. On each round of the round-block the filler distributes 1 unit of fill equally among all cups in the active set. Then the emptier removes fill from some cup in B. The filler chooses a random cup to remove from the active set. The probability that the cup the emptier emptied from is not in the active set after a random cup is removed from the active set by the filler is at least constant. By the end of the round-block the active-set will consist of a single cup. With constant probability, in particular probability at least

$$q_0 = 1/\lceil e^{2h} \rceil!,$$

this cup has gained fill at least $\ln\lceil e^{2h} \rceil \geq 2h$. Recalling that the cups fill started as at least $-\mu_\Delta - R_\Delta$, we have that this cup now has fill at least $2h - \mu_\Delta - R_\Delta$; by design in choosing h this quantity is at least h.

Now we shall choose c_Δ, choosing it large enough such that with constant probability there is some round-block on which the emptier doesn't neglect the anchor set on which the filler succeeds.

We choose

$$c_\Delta = 2\frac{1}{q_0}\mu_\Delta.$$

By having $n \cdot c_\Delta$ round-blocks, we make it so that there should be at least $n\mu_\Delta$ round-blocks on which the filler correctly guesses the emptier's emptying sequence. Formally this is due to a Chernoff bound: the expectation of the number of rounds when the filler correctly guesses the emptier's emptying sequence is at least $2n\mu_\Delta$, and the probability that it deviates from its expectation by more than $n\mu_\Delta$ is hence exponentially small in n. As shown before, the emptier cannot neglect the anchor set more than $(n/2)\mu_\Delta$ times. The filler correctly guesses the emptiers emptying sequence on the j-th round-block. Conditioned on this event, the j is chosen uniformly randomly from all the round-blocks on which the filler correctly guesses the emptiers emptying sequence. Since the emptier can neglect the anchor set on at most half of these round-blocks there is at least a $1/2$ chance that j is chosen on a round-block where the filler does not neglect the anchor set. Thus, overall, there is at least a constant probability of achieving fill h in a cup in A.

Say that a swapping-process ***succeeds*** if 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 succeeds. Let $q \geq \Omega(1)$ be the probability of a swapping-process succeeding, i.e. $\Pr(X_i = 1)$. Note that the random variables X_i are clearly independent, and identically distributed.

Clearly

$$\mathbb{E}\left[\sum_{i=1}^{n/4} X_i\right] = qn/4.$$

By a Chernoff Bound (i.e. Hoeffding's Inequality applied to binary random variables),

$$\Pr\left(\sum_{i=1}^{n/4} X_i \leq nq/8\right) \leq e^{-nq^2/128}.$$

That is, the probability that less than $nq/8$ of the anchor cups have fill at least h is exponentially small in n, as desired.

□

Hence Step 1 is possible.

Step 2 is easily achieved by setting $p = nc$ and uniformly distributing the fillers fill among a chosen set S of nc cups. The greedy nature of the emptier will force it to focus on the cups which must exist in A with large positive fill until the cups in S have sufficiently high fill. In particular, the fills of the cups in S must start as at least $-\mu_\Delta - \Delta \geq -h/2$ by design in choosing h. After removing from the very full cups for $\lceil h/2 + h' \rceil$ rounds the fills of these new cups are clearly at least h'. Note that throughout this process the emptier cannot have emptied from the cups in S until they attained fill h' because there would be $p = nc$ cups at least $\lceil h/2 + h' \rceil \geq h' + \Delta$ by design in choice of h.

Step 3, which is to recurse, is of course possible. By performing H levels of recursion, increasing the fill by $h' = 2$ and reducing the problem size by a factor of c at each level of recursion, the filler achieves backlog at least $2H$. Say the probability of Step 1 succeeding is at least $1 - e^{-nk}$. Then (by the union bound) the probability that any of H levels of recursion fail is bounded above by

$$e^{-nk} + e^{-nck} + e^{-nc^2k} + \cdots + e^{-nc^Hk} \leq 2^{-\Omega(n)}.$$

Hence the probability that every level of recursion succeeds is at least $1 - 2^{-\Omega(n)}$.

□

Lemma 2 (The Oblivious Amplification Lemma). *Let $\Delta \leq O(1)$, $M, M' \geq R_\Delta$, $q \geq \Omega(1)$, f be an oblivious filling strategy that achieves backlog $f(n)$ in the negative-fill variable-processor cup game on n cups with probability at least $1 - 2^{-qn}$ in running time $T(n) \leq poly(n)$ when given a M-flat configuration, against a Δ-greedy-like emptier.*

Let $0 < \delta \ll 1/2$, $L \in \mathbb{N}$, $\eta \in \mathbb{N}$ be constant parameters, appropriately chosen. Let $1/2 \ll \phi < 1$ be a constant parameter chosen as close to 1 as desired.

There exists an oblivious filling strategy that achieves backlog

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

in the negative-fill variable-processor cup game on n cups given a M'-flat configuration of cups in running time

$$T'(n) \leq O(M') + (\delta L) n^{\eta+1} T((1-\delta)n)$$

against a Δ-greedy-like emptier with probability at least

$$1 - 2^{-\Omega(n)}.$$

Proof. The proof is quite similar to the proof of Lemma 1, but more complicated because the filler's strategy must be randomized.

The filler starts by flattening all the cups, using the flattening procedure detailed in Proposition 5.

Let A, the ***anchor*** set, be a subset of the cups chosen uniformly at random from all subsets of size $n\delta$ of the cups, and let B, the ***non-anchor*** set, consist of the rest of the cups ($|B| = n(1-\delta)$). Let $n_\ell = n\delta^{\ell-1}, h_\ell = (1-\delta) f(n_\ell(1-\delta))$; the filler will achieve average fill h_ℓ on a set of $n_\ell \delta$ cups on the ℓ-th level of its recursive process. Note that the average fill of A and B both must start as at least $-R_\Delta$ due to the flattening.

The filler's strategy is essentially as follows:

- **Step 1:** Using f repeatedly, achieve fill $(1-\delta) f(n(1-\delta))$ in cups in the non-anchor set and then swap these cups into the anchor set.
- **Step 2:** Decrease the number of processors to $p = \delta n$ and recurse on the anchor set.

First we show how to achieve Step 1. The filler's strategy will be to always place 1 fill in each cup in the anchor-set while applying f to B. As always, the filler cannot directly apply f to B; the filler must ensure that the emptier is using the appropriate amount of resources on B.

For each cup in A the filler performs a procedure called a ***swapping-process***, which consists of a sub-structure repeated many times that we call a ***round-block***. Each round-block consists of an ***attempt*** to apply f. We say that the emptier ***neglects*** the anchor set on a round-block if there is at least 1 round on which the emptier does not empty from each cup in the anchor set. The mass of the anchor set increases by at least 1 on each round-block that the anchor set is neglected. This cannot happen more than $n\delta(2R_\Delta + \Delta) = n\delta\mu_\Delta$ times. Thus, by making each swapping-process consist of n^η round-blocks (note: we do n^η round-blocks on all levels of recursion, everything else changes to n_ℓ but not this) and then choosing a single round-block among these (uniformly at random) to swap a cup in to A, we guarantee that with probability at least $\delta\mu_\Delta/n^\eta$ this round-block occurs on a round-block when the emptier does not neglect the anchor set. On this round-block f is legitimately applied, and succeeds with probability at least $1-2^{-qn}$. At the end of each round-block the filler flattens B, so that f is receiving as input a flattened set of cups as needed. Over the course of this process the average fill of B will decrease a little. In the most extreme case f may have succeeded up to δn times, in which case the mass transfered from B to A would be $\delta n f((1-\delta)n)$. In order for their to be an increase in the difference of the average fills of A and B by this amount B would have had to contribute $|A|/n = \delta$ of the difference, with A contributing $|B|/n = (1-\delta)$ of the difference. Hence the average fill of A would have actually only increased by $(1-\delta)f((1-\delta)n)$. the whole preceding part is a little bit sketchy.

For Step 2 the filler simply recurses. The run-time bound is clear: The initial smoothing takes time $O(M')$, and after that, at each level of recursion for each cup in the anchor set the filler applies f to the non-anchor set n^η times. Hence the running time due to this is

$$\sum_{\ell=0}^{L}(\delta^\ell n\delta)n^\eta T((1-\delta)\delta^\ell n).$$

This is quite complicated, a simpler bound suffices; bounding each term in the sum with the first term, which is clearly the largest, we have

$$T'(n) \le O(M') + (L\delta)n^{\eta+1}T((1-\delta)n).$$

It is almost clear that the desired backlog is achieved; if every swapping process succeeded then we would achieve fill $(1-\delta)f((1-\delta)\delta^\ell n)$ in each cup in the anchor set at each level of recursion hence achieving backlog

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

overall. However each swapping process has some (very small) probability of failing; we computed probability of failure this to be at most $\delta\mu_\Delta/n^\eta$. Consider the probability that more than a constant fraction $w=\Theta(1)$ of the $s=\sum_{\ell=0}^{L} n\delta^{\ell+1}$ swapping-processes fail. Let X_i be the random variable indicating whether the i-th swapping-process succeeds (note: this is swapping-processes on all levels of recursion), and let $X=\sum_{i=1}^{s} X_i$. Clearly $\mathbb{E}[X] = s(1-\delta\mu_\Delta/n^\eta)$. Success of the swapping-processes are not independent events: a swapping-process is in-fact more likely to succeed given that previous swapping-processes have failed. Hence we can upper bound the probability of more than a w-fraction of the swapping-processes failing by a Chernoff Bound:

$$\Pr\left(\frac{1}{s}X \ge \frac{1}{s}\mathbb{E}[X] - w/2\right) \ge 1-2e^{-sw^2/2} \ge 1-2^{\Omega(n)}$$

By appropriately large choice for $\eta \le O(1)$,

$$\delta\mu_\Delta/n^\eta \le w/2$$

no matter how small $w \ge \Omega(1)$ is chosen. In particular this implies that $\Pr[X \ge s(1-w)] \ge 1-2^{\Omega(n)}$.

Now we will define ϕ such that success of $s(1-w)$ of the swapping-processes guarantees backlog

$$\phi\cdot(1-\delta)\sum_{\ell=0}^{L} f(n(1-\delta)\delta^\ell).$$

In the worst case the failed swapping-processes bring very negative cups into the anchor-set, potentially as negative as $-\delta f((1-\delta)\delta^\ell n)$ on the ℓ-th level of recursion. However, clearly this is equivalent to removing at most 2 cups worth of mass from the anchor set. Overall we thus remove at most $2w$ cups worth of mass from the anchor set. Hence choosing $\phi = 1-2w$ works. Noting that the constant $w>0$ was arbitrary we have that ϕ can be made any constant arbitrarily close to 1.

In order to achieve this backlog however, not only does the filler need to be able to swap over $s(1-w)$ cups on rounds where the emptier neglects the anchor set, but no applications of f can fail; failure happens with probability $2^{-n(1-\delta)\delta^\ell q}$ for an application of f to $n(1-\delta)\delta^\ell$ cups. Taking a union bound over the poly(n) applications of f clearly still gets probability failure at most $2^{-\Omega(n)}$.

Thus overall the filler succeeds with at least probability $1-2^{-\Omega(n)}$.

□

Theorem 4. *There is an oblivious filling strategy for the variable-processor cup game on n cups that achieves backlog at least $\Omega(n^{1-\epsilon})$ for any constant $\epsilon>0$ in running time $2^{O(\log^2 n)}$ with probability at least $1-2^{-\Omega(n)}$.*

Proof. The proof is quite similar to that of Theorem 1, except we do not achieve the extremal backlog $\Omega(n)$ due to its extremely long running time being so long as to make our probabilities not good enough. Nonetheless it is quite remarkable that the filler is still able to achieve poly(n) backlog, and in fact the same asymptotic backlog as the adaptive filler when the games are restricted to the reasonable length of $2^{O(\log^2 n)}$.

We aim to achieve backlog $cn^{1-\epsilon}$ for $\epsilon>0$ a constant of our choice, and $0<c\ll 1$ an appropriate constant that we will choose. As in the proof of Theorem 4 we achieve large backlog by repeated amplification of a base case.

The base case is given by Proposition 6. However, unlike in the adaptive case, we cannot do the base case on a constant size subset of the cups: this would destroy our probability of success. Recalling that the running time of the algorithm is going to be $2^{\text{polylog}(n)}$ it seems reasonable to want to union bound over $2^{\text{polylog}(n)}$ events. Hence the probability of failure needs to also be at most $2^{-\text{polylog}(n)}$. Hence as our base case we use a polylog(n) size subset of the cups. Let $n_b = \text{polylog}(n)$ be the size of our base case.

Our base case strategy is

$$f_0(k) = \begin{cases} 2, & k > n_b \\ 0, & k \le n_b \end{cases}.$$

This is possible by Proposition 6. Then we construct f_{i+1} as the amplification of f_i using Lemma 2.

Define g_i as

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

Claim 10. *$f_i(k \cdot n_b) \geq ck^{1-\epsilon}$ for all $k < g_i$.*

We prove Claim 10 by induction. Clearly by appropriate choice of c the base case is satisfied.

Assume the claim for f_i, consider f_{i+1}. For any $k < g_{i-1}/(1-\delta)$ we have

$$\begin{aligned} &f_{i+1}(k \cdot n_b) \\ &\geq \phi \cdot (1-\delta) \sum_{\ell=0}^{L} f_i((1-\delta)\delta^\ell k \cdot n_b) \\ &\geq \phi \cdot (1-\delta) \sum_{\ell=0}^{L} c((1-\delta)\delta^\ell k)^{1-\epsilon} \\ &\geq ck^{1-\epsilon}\phi \cdot (1-\delta)^{2-\epsilon} \sum_{\ell=0}^{L} (\delta^\ell)^{1-\epsilon} \\ &\geq ck^{1-\epsilon}\phi \cdot (1-\delta)^{2-\epsilon}(1+\delta^{1-\epsilon}). \end{aligned}$$

Now we must prove the following claim, which is nearly identical to Claim 1, but slightly complicated by the ϕ factor out front:

Claim 11. *Let*

$$h(\delta, \phi) = \phi \cdot (1-\delta)^{2-\epsilon}(1+\delta^{1-\epsilon})$$

There exists constant choices of δ, ϕ such that $h(\delta, \phi) > 1$.

Proof. As before we lower bound h by

$$h(\delta, \phi) \geq \phi \cdot (1-(2-\epsilon)\delta)(1+\delta^{1-\epsilon}).$$

Recall from before that

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

is positive for $\delta \in (0, 1/(2(2-\epsilon))^{1/\epsilon})$. Choosing δ as the midpoint of this interval, $(1-(2-\epsilon)\delta)(1+\delta^{1-\epsilon})$ is some value strictly larger than 1, say $1+z$. Then choosing

$$\phi = \frac{1+z/2}{1+z}$$

guarantees that $h(\delta, \phi) = 1 + z/2 > 1$ as desired. $\square$

Now we can complete the proof of Claim 10,

$$\begin{aligned} &f_{i+1}(k) \\ &\geq ck^{1-\epsilon}h(\delta, \phi) \\ &\geq ck^{1-\epsilon}. \end{aligned}$$

Recursing for $O(\log n)$ levels of recursion is sufficient to achieve a function f with

$$f(n) \geq c(n/n_b^{1-\epsilon}$$

As $n_b \leq \text{polylog}(n)$ this is still $\Omega(n^{1-\epsilon})$ as desired.

Hence the desired backlog is achieved. For a resolving of the integer issues, see the proof of Theorem 1.

By identical analysis to before we get running-time $2^{O(\log^2 n)}$. As stated previously, the probability result is guaranteed by a union bound, so we have probability at least $1 - 2^{-\text{polylog}(n)}$ of success. $\square$

6 Conclusion

Many important open questions remain open. Can our oblivious cup game results be improved, e.g. by expanding them to apply to a broader class of emptiers? Can the classic oblivious multi-processor cup-game be tightly analyzed? These are interesting questions.

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.