Variable-Processor Cup Games

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

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

Abstract

In a ***cup game*** two players, the ***filler*** and the ***emptier***, take turns adding and removing water from cups, subject to certain constraints. In the classic 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 $\Omega(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$ of our choice in running time $2^{O(\log^2 n)}$. 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 achieve backlog $\Omega(n)$ in extremely long games: it has running time $O(n!)$.

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 construct an oblivious filling strategy that achieves backlog $\Omega(n^{1-\varepsilon})$ for $\varepsilon > 0$ constant of our choice in time $2^{O(\log^2 n)}$ against any "greedy-like" emptier with probability at least $1 - 2^{-\operatorname{polylog}(n)}$. Whereas classically randomization gives the emptier a large advantage, in the variable-processor cup game the lower bound is the same!

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 game has been studied for ***adaptive*** fillers, i.e. fillers that can observe the emptier's actions, and for ***oblivious*** fillers, i.e. fillers that cannot observe the emptier's actions.

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; in the cup game the ***backlog*** of the cups is the fill of the fullest cup at a given state. In analyzing a 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 $O(\log n)$. In the randomized version of the single-processor cup game, i.e. when the filler is oblivious, 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 $\Omega(\log\log n)$.

*Supported by a Hertz fellowship and a NSF GRFP fellowship

†Supported by MIT

[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.

Recently Kuszmaul has established bounds on the case where $p > 1$, i.e. the ***multi-processor cup game*** [4]. Kuszmaul showed that a greedy emptier never lets backlog exceed $O(\log n)$. He also proved a lower bound of $\Omega(\log(n-p))$ on backlog. Recently we showed a lower bound of $\Omega(\log n - \log(n-p))$. Combined, these lower bounds 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 $\Omega(\log \log n)$. Tight bounds on backlog against an oblivious filler are not yet known for large p.

The Variable-Processor Cup Game. We investigate a new 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.

The variable-processor cup game models the natural situation where many users are all on a server, and the number of processors allocated to each user is variable as other users get some portion of the processors.

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 use to 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 classic 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 classic 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 $\Omega(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$ in quasi-polynomial running time. Proposition 2 also provides an extremal strategy that achieves backlog $\Omega(n)$ in incredibly long games: it has $O(n!)$ 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!

In Section 5 we prove a lower bound on backlog with an oblivious filler. Theorem 3 states that an oblivious filler can achieve backlog $\Omega(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$ in quasi-polynomial time with probability at least $1 - 2^{-\operatorname{polylog}(n)}$. Theorem 3 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; being oblivious seems like a large disadvantage for the filler!

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 on round t, 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 ***zeroes out*** a cup at round t if the emptier empties, on round t, from a cup with fill at state I_t that is less than 1. Note that on any round where the emptier zeroes out a cup the emptier has removed less fill than the filler has added; hence the average fill will increase. This concludes the round;

[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.

the state of the game is now S_{t+1}.

Denote the fill of a cup c by $\text{fill}(c)$. 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. Let $[n] = \{1, 2, \ldots, n\}$, let $i + [n] = \{i+1, i+2, \ldots, i+n\}$.

Many of our lower bound proofs will adopt the convention of allowing for negative fill. We call this the ***negative-fill cup game***. Specifically, in the negative-fill 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 emptier's 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. Some results are proved for the variable-processor negative-fill cup game, and some results are proved for the single-processor negative-fill cup game.

3 Adaptive Filler Lower Bound

In this section we give a $2^{\text{polylog}\, n}$-time filling strategy that achieves backlog $n^{1-\varepsilon}$ for any positive constant ε. We also give a $O(n!)$-time filling strategy that achieves backlog $\Omega(n)$.

We begin with a simple proposition that gives backlog 1/2 for two cups.

Proposition 1. *Consider an instance of the negative-fill 1-processor cup game on 2 cups, and let the cups start in any state where the average fill is 0. There is an $O(1)$-step adaptive filling strategy that achieves backlog at least 1/2.*

Proof. Let the fills of the 2 cups start as x and $-x$ for some $x \geq 0$. If $x \geq 1/2$ the algorithm need not do anything. Otherwise, the filling strategy adds $1/2 - x$ fill to the cup with fill x, and adds $1/2 + x$ fill to the cup with fill $-x$. This results in 2 cups both having fill 1/2; the emptier then empties from one of these, and leaves a cup with fill 1/2, as desired. □

Next we prove the ***Amplification Lemma***.

Lemma 1 (Adaptive Amplification Lemma)**.** *Let $\delta \in (0, 1)$ be a parameter. Let alg(f) be an adaptive filling strategy that achieves backlog $f(n) < n$ in the negative-fill variable-processor cup game on n cups in running time $T(n)$ starting from any initial cup state where the average fill is 0.*

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

$$f'(n) \geq (1-\delta) f(\lfloor (1-\delta)n \rfloor) + f(\lceil \delta n \rceil)$$

and $f'(n) \geq f(n)$ in the negative-fill variable-processor cup game on n cups in running time

$$T'(n) \leq n^2 \delta \cdot T(\lfloor (1-\delta)n \rfloor) + T(\lceil \delta n \rceil)$$

starting from any initial cup state where the average fill is 0.

Before proving the Amplification Lemma, we briefly motivate it. We call *alg*(f'), the filling strategy created by the Amplification Lemma, the ***amplification*** of *alg*(f). As suggested by the name, *alg*(f') will be able to achieve higher backlog than *alg*(f). In particular, we will show that by starting with a filling strategy *alg*(f_0) for achieving constant backlog and then recursively forming a sufficiently long sequence of filling strategies *alg*(f_0), *alg*$(f_1), \ldots,$ *alg*(f_{i_*}) with *alg*(f_{i+1}) the amplification of *alg*(f_i), we eventually get a filling strategy for achieving $\text{poly}(n)$ backlog.

Proof of Amplification Lemma. The algorithm defaults to using *alg*(f) if $f(n) \geq (1-\delta) f(\lfloor (1-\delta)n \rfloor) + f(\lceil \delta n \rceil)$; in this case using *alg*(f) achieves the desired backlog in the desired running time. In the rest of the proof, we describe our strategy for achieving backlog $(1-\delta) f(\lfloor (1-\delta)n \rfloor) + f(\lceil \delta n \rceil)$.

Let A, the ***anchor set***, be initialized to consist of the $\lceil n\delta \rceil$ fullest cups, and let B the ***non-anchor set*** be initialized to consist of the rest of the cups (so $|B| = \lfloor (1-\delta)n \rfloor$). Let $h = (1-\delta) f(\lfloor (1-\delta)n \rfloor)$.

The filler's strategy is roughly as follows:
Step 1: Get $\mu(A) \geq h$ by using *alg*(f) repeatedly on B to achieve cups with fill at least $\mu(B) + f(|B|)$ in B and then swapping these into A.
Step 2: Use *alg*(f) once on A to obtain some cup with fill $\mu(A) + f(|A|)$.
Note that in order to use *alg*(f) on subsets of the cups the filler will need to vary p.

We now describe how to achieve Step 1, which is complicated by the fact that the emptier may attempt to prevent the filler from achieving high fill in a cup in B.

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. If the emptier wishes to keep the average fill of the anchor cups from increasing, then emptier must empty from every anchor cup on each step. If the emptier fails to do this on a given round, then we say that the emptier has ***neglected*** the anchor cups.

We say that the filler ***applies*** *alg*(f) to B if it follows the filling strategy *alg*(f) on B while placing 1 unit of water in each anchor cup. An application of *alg*(f) to B is said to be ***successful*** if A is never neglected during the application of *alg*(f) to B. The filler uses a procedure that we call a ***swapping-process*** to achieve the desired average fill in A. In a swapping-process, the filler repeatedly applies *alg*(f) to B until a successful application occurs, and then takes the cup generated by *alg*(f) within B on this successful application with fill at least $\mu(B) + f(|B|)$ and swaps it with the least full cup in A. If the average fill in A ever reaches h, then the algorithm immediately halts (even if it is in the middle of a swapping-process) and is complete.

Note that

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

so

$$\mu(A) = -\mu(B) \cdot \frac{\lfloor (1-\delta)n \rfloor}{\lceil \delta n \rceil} \geq -\frac{1-\delta}{\delta}\mu(B).$$

Thus, if at any point B has average fill lower than $-h \cdot \delta/(1-\delta)$, then A has average fill at least h, so the algorithm is finished. Thus we can assume in our analysis that

$$\mu(B) \geq -h \cdot \delta/(1-\delta). \qquad (1)$$

We will now show that during each swapping process, the filler applies *alg*(f) to B at most $hn\delta + 1$ times. Each time the emptier neglects the anchor set, the mass of the anchor set increases by 1. If the emptier neglects the anchor set $hn\delta+1$ times, then the average fill in the anchor set increases by more than h, so the desired average fill is achieved in the anchor set. Thus the swapping process consists of at most $hn\delta+1$ applications of *alg*(f).

Consider the fill of a cup c swapped into A at the end of a swapping-process. Cup c's fill is at least $\mu(B) + f(|B|)$, which by (1) is at least

$$-h \cdot \frac{\delta}{1-\delta} + f(\lfloor n(1-\delta) \rfloor) = (1-\delta) f(\lfloor n(1-\delta) \rfloor) = h.$$

Thus the algorithm for Step 1 succeeds within $|A| = \lceil \delta n \rceil$ swapping-processes, since at the end of the $|A|$-th swapping process every cup in A has fill at least h, or the algorithm halted before $|A|$ swapping-processes because it already achieved $\mu(A) \geq h$.

Now the filler performs Step 2, i.e. the filler applies *alg*(f) to A, and hence achieves a cup with fill at least

$$\mu(A) + f(|A|) \geq (1-\delta) f(\lfloor (1-\delta)n) \rfloor) + f(\lceil \delta n \rceil),$$

as desired.

Now we analyze the running time of the filling strategy *alg*(f'). First, recall that in Step 1 *alg*(f') calls *alg*(f) on a set of size $\lfloor (1-\delta)n \rfloor$ as many as $hn\delta + 1$ times. Because we mandate that $h < n$, Step 1 contributes no more than $(n \cdot n\delta) \cdot T(|B|)$ to the running time. Step 2 requires applying *alg*(f) to $|A|$ cups one time, and hence contributes $T(|A|)$ to the running time. Summing these we have

$$T'(n) \leq n^2 \delta \cdot T(\lfloor (1-\delta)n \rfloor) + T(\lceil \delta n \rceil).$$

□

We next show that by recursively using the Amplification Lemma we can achieve backlog $n^{1-\varepsilon}$.

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

Proof. Take constant $\varepsilon \in (0, 1/2)$. Let c, δ be parameters, with $c \in (0,1), 0 < \delta \ll 1/2$, these will be chosen later as functions of ε. We show how to achieve backlog at least $cn^{1-\varepsilon} - 1$.

By Proposition 1 there exists a constant n_0 such that a filler can achieve backlog 1 on n_0 cups (e.g., $n_0 = 1000$ works). Let *alg*(f_0) by the filling strategy described in Proposition 1, where $f_0(k) \geq 1$ for all $k \geq n_0$.

Next, using the Amplification Lemma we recursively construct *alg*(f_{i+1}) as the amplification of *alg*(f_i) for $i \geq 0$.

Define a sequence g_i with

$$g_i = \begin{cases} \lceil 16/\delta \rceil, & i = 0, \\ \lfloor g_{i-1}/(1-\delta) \rfloor & i \geq 1 \end{cases}$$

We claim the following regarding this construction:

Claim 1. *For all $i \geq 0$,*

$$f_i(k) \geq ck^{1-\varepsilon} - 1 \;\; \text{for all} \;\; k \in [g_i]. \qquad (2)$$

Proof. We prove Claim 1 by induction on i. For $i = 0$, the base case, (2) can be made true by taking c and δ sufficiently small. In particular, we we choose $c = \Theta(1)$ small enough to make $cn_0^{1-\varepsilon} - 1 \leq 0$, which implies (2) holds for $k \in [n_0]$ by monotonicity of $ck^{1-\varepsilon} - 1$; we also choose δ small enough to make $g_0 \geq n_0$, and we choose c small enough to make

$cg_0^{1-\varepsilon} - 1 \leq f_0(g_0) = 1$, which implies (2) holds for $k \in [n_0, g_0]$ by monotonicity of $ck^{1-\varepsilon} - 1$. [3]

As our inductive hypothesis we assume (2) for f_i; we aim to show that (2) holds for f_{i+1}. Note that, by design of g_i, if $k \leq g_{i+1}$ then $\lfloor k \cdot (1-\delta) \rfloor \leq g_i$. Consider any $k \in [g_{i+1}]$. First we deal with the trivial case where $k \leq g_0$. In this case

$$f_{i+1}(k) \geq f_i(k) \geq \cdots \geq f_0(k) \geq ck^{1-\varepsilon} - 1.$$

Now we consider the case where $k \geq g_0$. Since f_{i+1} is the amplification of f_i we have

$$f_{i+1}(k) \geq (1-\delta) f_i(\lfloor (1-\delta)k \rfloor) + f_i(\lceil \delta k \rceil).$$

By our inductive hypothesis, which applies as $\lceil \delta k \rceil \leq g_i$, $\lfloor k \cdot (1-\delta) \rfloor \leq g_i$, we have

$$f_{i+1}(k) \geq (1-\delta)(c \cdot \lfloor (1-\delta)k \rfloor^{1-\varepsilon} - 1) + c \lceil \delta k \rceil^{1-\varepsilon} - 1.$$

Dropping the floor and ceiling, incurring a -1 for dropping the floor, we have

$$f_{i+1}(k) \geq (1-\delta)(c \cdot ((1-\delta)k - 1)^{1-\varepsilon} - 1) + c(\delta k)^{1-\varepsilon} - 1.$$

Because $(x-1)^{1-\varepsilon} \geq x^{1-\varepsilon} - 1$, due to the fact that $x \mapsto x^{1-\varepsilon}$ is a sub-linear sub-additive function, we have

$$f_{i+1}(k) \geq (1-\delta)c \cdot (((1-\delta)k)^{1-\varepsilon} - 2) + c(\delta k)^{1-\varepsilon} - 1.$$

Moving the $ck^{1-\varepsilon}$ to the front we have

$$f_{i+1}(k) \geq ck^{1-\varepsilon} \cdot \left((1-\delta)^{2-\varepsilon} + \delta^{1-\varepsilon} - \frac{2(1-\delta)}{k^{1-\varepsilon}} \right) - 1.$$

Because $(1-\delta)^{2-\varepsilon} \geq 1 - (2-\varepsilon)\delta$, a fact called Bernoulli's Identity, we have

$$f_{i+1}(k) \geq ck^{1-\varepsilon} \cdot \left(1 - (2-\varepsilon)\delta + \delta^{1-\varepsilon} - \frac{2(1-\delta)}{k^{1-\varepsilon}} \right) - 1.$$

Of course $-2(1-\delta) \geq -2$, so

$$f_{i+1}(k) \geq ck^{1-\varepsilon} \cdot \left(1 - (2-\varepsilon)\delta + \delta^{1-\varepsilon} - \frac{2}{k^{1-\varepsilon}} \right) - 1.$$

Because

$$-2/k^{1-\varepsilon} \geq -2/g_0^{1-\varepsilon} \geq -2(\delta/16)^{1-\varepsilon} \geq -\delta^{1-\varepsilon}/2,$$

which follows from our choice of $g_0 = \lceil 16/\delta \rceil$ and the restriction $\varepsilon < 1/2$, we have

$$f_{i+1}(k) \geq ck^{1-\varepsilon} \cdot \left(1 - (2-\varepsilon)\delta + \delta^{1-\varepsilon} - (1/2)\delta^{1-\varepsilon} \right) - 1.$$

Finally, combining terms we have

$$f_{i+1}(k) \geq ck^{1-\varepsilon} \cdot \left(1 - (2-\varepsilon)\delta + (1/2)\delta^{1-\varepsilon} \right) - 1.$$

Because $\delta^{1-\varepsilon}$ dominates δ for sufficiently small δ, there is a choice of $\delta = \Theta(1)$ such that

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

Taking δ to be this small we have,

$$f_{i+1}(k) \geq ck^{1-\varepsilon} - 1,$$

completing the proof. We remark that the choices of c, δ are the same for every i in the inductive proof, and depend only on ε. □

To complete the proof, we will show that g_i grows exponentially in i. Thus, after there exists $i_* \leq O(\log n)$ such that $g_{i_*} \geq n$, and hence we have an algorithm $alg(f_{i_*})$ that achieves backlog $cn^{1-\varepsilon} - 1$ on n cups, as desired.

We lower bound the sequence g_i with another sequence g'_i defined as

$$g'_i = \begin{cases} 4/\delta, & i = 0 \\ g'_{i-1}/(1-\delta) - 1, & i > 0. \end{cases}$$

Solving this recurrence, we find

$$g'_i = \frac{4 - (1-\delta)^2}{\delta} \frac{1}{(1-\delta)^i} \geq \frac{1}{(1-\delta)^i},$$

which clearly exhibits exponential growth. In particular, let $i_* = \left\lceil \log_{1/(1-\delta)} n \right\rceil$. Then,

$$g_{i_*} \geq g'_{i_*} \geq n,$$

as desired.

Let the running time of $f_i(n)$ be $T_i(n)$. From the Amplification Lemma we have following recurrence bounding $T_i(n)$:

$$\begin{aligned} T_i(n) &\leq n^2\delta \cdot T_{i-1}(\lfloor (1-\delta)n \rfloor) + T_{i-1}(\lceil \delta n \rceil) \\ &\leq 2n^2 T_{i-1}(\lfloor (1-\delta)n \rfloor). \end{aligned}$$

It follows that $alg(f_{i_*})$, recalling that $i_* \leq O(\log n)$, has running time

$$T_{i_*}(n) \leq (2n^2)^{O(\log n)} \leq 2^{O(\log^2 n)}$$

as desired.

□

Now we provide a very simple construction that can achieve backlog $\Omega(n)$ in very long games. The construction can be interpreted as the same argument

[3] Note that it is important here that ε and δ are constants, that way c is also a constant.

as in Theorem 1 but with an extremal setting of δ to $\Theta(1/n)$. [4]

Proposition 2. *There is an adaptive filling strategy that achieves backlog $\Omega(n)$ in time $O(n!)$.*

Proof. We start, as in the proof of Theorem 1, with an algorithm $alg(f_0)$ for achieving backlog $f_0(k) \geq 1$ on $k \geq n_0$ cups, which is possible by Proposition 1. For $i > 0$ we construct $alg(f_i)$ as the amplification of $alg(f_{i-1})$ using the Amplification Lemma with parameter $\delta = 1/(i+1)$.

We claim the following regarding this construction:

Claim 2. *For all $i \geq 0$,*

$$f_i((i+1) \cdot n_0) \geq \sum_{j=0}^{i} \left(1 - \frac{j}{i+1}\right). \tag{3}$$

Proof. We prove Claim 2 by induction on i. When $i = 0$, the base case, (3) becomes $f_0(n_0) \geq 1$ which is true. Assuming (3) for f_{i-1}, we now show (3) holds for f_i. Because f_i is the amplification of f_{i-1} with $\delta = 1/(i+1)$, we have by the Amplification Lemma

$$f_i((i+1) \cdot n_0) \geq \left(1 - \frac{1}{i+1}\right) f_{i-1}(i \cdot n_0) + f_{i-1}(n_0).$$

Since $f_{i-1}(n_0) \geq f_0(n_0) \geq 1$ we have

$$f_i((i+1) \cdot n_0) \geq \left(1 - \frac{1}{i+1}\right) f_{i-1}(i \cdot n_0) + 1.$$

Using the inductive hypothesis we have

$$f_i((i+1) \cdot n_0) \geq \left(1 - \frac{1}{i+1}\right) \sum_{j=0}^{i-1} \left(1 - \frac{j}{i}\right) + 1.$$

Note that

$$\begin{aligned} \left(1 - \frac{1}{i+1}\right) \cdot \left(1 - \frac{j}{i}\right) &= \frac{i}{i+1} \cdot \frac{i-j}{i} \\ &= \frac{i-j}{i+1} \\ &= 1 - \frac{j+1}{i+1}. \end{aligned}$$

Thus we have

$$f_i((i+1) \cdot n_0) \geq \sum_{j=1}^{i} \left(1 - \frac{j}{i+1}\right) + 1 = \sum_{j=0}^{i} \left(1 - \frac{j}{i+1}\right),$$

as desired. □

Let $i_* = \lfloor n/n_0 \rfloor - 1$, which by design satisfies $(i_* + 1)n_0 \leq n$. By Claim 2 we have

$$f_{i_*}((i_* + 1) \cdot n_0) \geq \sum_{j=0}^{i_*} \left(1 - \frac{j}{i_* + 1}\right) = i_*/2 + 1.$$

As $i_* = \Theta(n)$, we have thus shown that $alg(f_{i_*})$ can achieve backlog $\Omega(n)$ on n cups.

Let T_i be the running time of $alg(f_i)$. The recurrence for the running running time of f_{i_*} is

$$T_i(n) \leq n \cdot n_0 T_{i-1}(n - n_0) + O(1).$$

Clearly $T_{i_*}(n) \leq O(n!)$.

□

4 Upper Bound

In this section we analyze the ***greedy emptier***, which always empties from the p fullest cups. We prove in Corollary 1 that the greedy emptier prevents backlog from exceeding $O(n)$.

In order to analyze the greedy emptier, we establish a system of invariants that hold at every step of the game.

Let $\mu_S(X)$ and $m_S(X)$ denote the average fill and the mass, respectively, of a set of cups X at state S (e.g. $S = S_t$ or $S = I_t$).[5] Let $S(\{r_1, \ldots, r_m\})$ denote the set of cups of ranks $r_1, r_2, \ldots, r_m$ at state S. We will use concatenation of sets to denote unions, i.e. $AB = A \cup B$.

The main result of the section is the following theorem.

Theorem 2. *In the variable-processor cup game on n cups, the greedy emptier maintains, at every step t, the invariants*

$$\mu_{S_t}(S_t([k])) \leq 2n - k \tag{4}$$

for all $k \in [n]$.

By applying Theorem 2 to the case of $k = 1$, we arrive at a bound on backlog:

Corollary 1. *On a game with n cups, the greedy emptying strategy achieves backlog $O(n)$.*

Proof of Theorem 2. We prove the invariants by induction on t. The invariants hold trivially for $t = 1$

[4] Or more precisely, setting δ in each level of recursion to be $\Theta(1/n)$, where n is the subproblem size; note in particular that δ changes between levels of recursion, which was not the case in the proof of Theorem 1.

[5] Note that in the lower bound proofs (i.e. Section 3 and Section 5) when we use the notation m (for mass) and μ (for average fill), we omit the subscript indicating the state at which the properties are measured. In those proofs the state is implicitly clear. However, in this section it will be useful to make the state S explicit in the notation.

(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 the invariants 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) 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 the 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 are not among the k fullest cups in S_{t+1}. Let $C = I_t(a + b + [k - a])$, with $c = k - a = |C|$; C consists of the cups with ranks $a + b + 1, \ldots, k + b$ in state I_t. The set C is defined so that $S_{t+1}([k]) = AC$, since once the cups in B are emptied from, the cups in B are not among the k fullest cups, so cups in C take their places among the k fullest cups.

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.

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

Claim 3. *There exists a cup state S'_t such that: (a) S'_t satisfies the invariants* (4)*, (b) $S'_t(r) = I_t(r)$ for all ranks $r \in [n]$, and (c) the filler can legally place water into cups in order to transform S'_t into I_t.*

Proof. Fix $r \in [n]$. We will show that S_t can be transformed into a state S^r_t by relabelling only cups with ranks in $[r]$ such that (a) S^r_t satisfies the invariants (4), (b) $S^r_t([r]) = I_t([r])$ and (c) the filler can legally place water into cups in order to transform S^r_t into I_t.

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

$$f_x > f_y. \tag{5}$$

Let the amount of fill that the filler adds to these cups be $\Delta_x, \Delta_y \in [0, 1]$; note that

$$f_x + \Delta_x < f_y + \Delta_y. \tag{6}$$

Define a new state S'_t where cup x has fill f_y and cup y has fill f_x. Note that the filler can transform state S'_t into state I_t by placing water into cups as before, except changing the amount of water placed into cups x and y to be $f_x - f_y + \Delta_x$ and $f_y - f_x + \Delta_y$, respectively.

In order to verify that the transformation from S'_t to I_t is a valid step for the filler, one must check three conditions. First, the amount of water placed by the filler is unchanged: this is because $(f_x - f_y + \Delta_x) + (f_y - f_x + \Delta_y) = \Delta_x + \Delta_y$. Second, the fills placed in cups x and y are at most 1: this is because $f_x - f_y + \Delta_x < \Delta_y \leq 1$ (by (6)) and $f_y - f_x + \Delta_x < \Delta_x \leq 1$ (by (5)). Third, the fills placed in cups x and y are non-negative: this is because $f_x - f_y + \Delta_x > \Delta_x \geq 0$ (by (5)) and $f_y - f_x + \Delta_y > \Delta_x \geq 0$ (by (6)).

We can repeatedly apply this process to swap each cup in $I_t([r]) \setminus S_t([r])$ into being in $S'_t([r])$. At the end of this process we will have some state S^r_t for which $S^r_t([r]) = I_t([r])$. Note that S^r_t is simply a relabeling of S_t, hence it must satisfy the same invariants (4) satisfied by S_t. Further, S^r_t can be transformed into I_t by a valid filling step.

Now we repeatedly apply this process, in descending order of ranks. In particular, we have the following process: create a sequence of states by starting with S^{n-1}_t, and to get to state S^r_t from state S^{r+1}_t apply the process described above. Note that S^{n-1}_t satisfies $S^{n-1}_t([n-1]) = I_t([n-1])$ and thus also $S^{n-1}_t(n) = I_t(n)$. If S^{r+1}_t satisfies $S^{r+1}_t(r') = I_t(r')$ for all $r' > r+1$ then S^r_t satisfies $S^r_t(r') = I_t(r')$ for all $r > r$, because the transition from S^{r+1}_t to S^r_t has not changed the labels of any cups with ranks in $(r+1, n]$, but the transition does enforce $S^r_t([r]) = I_t([r])$, and consequently $S^r_t(r+1) = I_t(r+1)$. We continue with the sequential process until arriving at state S^1_t in which we have $S^1_t(r) = I_t(r)$ for all r. Throughout the process each S^r_t has satisfied the invariants (4), so S^1_t satisfies the invariants (4). Further, throughout the process from each S^r_t it is possible to legally place water into cups in order to transform S^r_t into I_t.

Hence S^1_t satisfies all the properties desired, and the proof of Claim 3 is complete.

□

Claim 3 tells us that we may assume without loss of generality that $S_t(r) = I_t(r)$ for each rank $r \in [n]$. We will make this assumption for the rest of the proof.

In order to complete the proof of the theorem, we break it into three cases.

Claim 4. *If some cup in A zeroes out in round t, then the invariant $\mu_{S_{t+1}}(S_{t+1}([k])) \leq 2n - k$ holds.*

Proof. Say a cup in A zeroes out in step t. Of course

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

because the $a-1$ fullest cups must have satisfied the invariant (with $k = a-1$) on round t. Moreover, because $\mathrm{fill}_{S_{t+1}}(I_{t+1}(a)) = 0$

$$m_{S_{t+1}}(I_t([a])) = m_{S_{t+1}}(I_t([a-1])).$$

Combining the above equations, we get that

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

Furthermore, the fill of all cups in C must be at most 1 at state I_t to be less than the fill of the cup in A that zeroed out. Thus,

$$\begin{aligned} m_{S_{t+1}}(S_{t+1}([k])) &= m_{S_{t+1}}(AC) \\ &\leq (a-1)(2n-(a-1)) + k - a \\ &= a(2n-a) + a - 2n + a - 1 + k - a \\ &= a(2n-a) + (k-n) + (a-n) - 1 \\ &< a(2n-a) \end{aligned}$$

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

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

□

Claim 5. *If no cups in A zero out in round t and $b = 0$, then the invariant $\mu_{S_{t+1}}(S_{t+1}([k])) \leq 2n-k$ holds.*

Proof. If $b = 0$, then $S_{t+1}([k]) = S_t([k])$. During round t the emptier removes 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:

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

□

The remaining case, in which no cups in A zero out and $b > 0$ is the most technically interesting.

Claim 6. *If no cups in A zero out on round t and $b > 0$, then the invariant $\mu_{S_{t+1}}(S_{t+1}([k])) \leq 2n-k$ 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 to 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 to 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:

$$m_{S_{t+1}}(S_{t+1}([k])) \leq m_{S_t}(AC) + b. \qquad (7)$$

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 capturing the pushing-down relationship combinatorially we will achieve the desired inequality.

We can upper bound $m_{S_t}(C)$ by

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

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

$$\begin{aligned} m_{S_t}(AC) &\leq m_{S_t}(A) + \frac{c}{b+c} m_{S_t}(BC) && (8) \\ &= \frac{c}{b+c} m_{S_t}(ABC) + \frac{b}{b+c} m_{S_t}(A). && (9) \end{aligned}$$

Note that the expression in (9) 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, $2n - |ABC|$ and $2n - |A|$. At this point the claim can be verified by straightforward (but quite messy) algebra (and by combining (7) with (9)). We instead give a more intuitive argument, in which we examine the right side of (8) combinatorially.

Consider a new configuration of fills F achieved by starting with state S_t, and moving water from BC into A until $\mu_F(A) = 2n - |A|$. [6] This transformation increases (strictly increases if and only if we move a non-zero amount of water) the right side of (8). In particular, if mass $\Delta \geq 0$ fill is moved from BC to

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

A, then the right side of (8) increases by $\frac{b}{b+c}\Delta \geq 0$. Note that the fact that moving water from BC into A increases the right side of (8) formally captures the way the system of invariants being proven forces a tradeoff between the fill in A and the fill in BC—that is, higher fill in A pushes down the fill that BC (and consequently C) can have.

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

$$\begin{aligned}
&\mu_F(A) - \mu_F(ABC) \\
&= (2n - |A|) - \mu_F(ABC) \\
&\geq (2n - |A|) - (2n - |ABC|) \\
&= |BC|.
\end{aligned}$$

Hence we must have

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

Thus

$$\mu_F(BC) \leq \mu_F(ABC) - |A| \leq 2n - |ABC| - |A|. \quad (10)$$

Combing (8) with the fact that the transformation from S_t to F only increases the right side of (8), along with (10), we have the following bound:

$$\begin{aligned}
m_{S_t}(AC) &\leq m_F(A) + c\mu_F(BC) \\
&\leq a(2n-a) + c(2n - |ABC| - a) \\
&\leq (a+c)(2n-a) - c(a+c+b) \\
&\leq (a+c)(2n-a-c) - cb. \qquad (11)
\end{aligned}$$

By (7) and (11), we have that

$$\begin{aligned}
m_{S_{t+1}}(S_{t+1}([k])) &\leq m_{S_t}(AC) + b \\
&\leq (a+c)(2n-a-c) - cb + b \\
&= k(2n-k) - cb + b \\
&\leq k(2n-k),
\end{aligned}$$

where the final inequality uses the fact that $c \geq 1$. This completes the proof of the claim. □

We have shown the invariant holds 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 $t \in \mathbb{N}$. □

TODO: emptier turn skipping RIP **TODO:** integer stuff

5 Oblivious Filler Lower Bound

In this section we prove that, surprisingly, an oblivious filler can achieve backlog $n^{1-\varepsilon}$, although only against a certain class of "greedy-like" emptiers.

We call a cup configuration ***M*-flat** if the difference between the fill of the fullest cup and the fill of the least full is at most M; note that in an M-flat cup configuration with average fill 0 all cups have fills in $[-M, M]$. We say an emptier is **Δ-*greedy-like*** if, whenever there are two cups with fills that differ by at least Δ, the emptier never empties from the less full cup without also emptying from the more full cup. That is, if on some round t, there are cups c_1, c_2 with $\text{fill}_{I_t}(c_1) > \text{fill}_{I_t}(c_2) + \Delta$, then a Δ-greedy-like emptier doesn't empty from c_2 on round t unless it also empties from c_1 on round t. Note that a perfectly greedy emptier is 0-greedy-like. We call an emptier ***greedy-like*** if it is Δ-greedy-like for $\Delta \leq O(1)$.

With an oblivious filler we are only able to prove lower bounds on 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. Nonetheless, greedy-like emptiers are of great interest because all the known randomized algorithm for the cup game are greedy-like [1, 4].

As a tool in our analysis we define a new variant of the cup game: In the p-processor ***E*-extra-empties** negative-fill cup game on n cups, the filler distributes p units of water amongst the cups, and then the emptier empties from p *or more* cups. In particular the emptier is allowed to empty E extra cups over the course of the game. Note that the emptier still cannot empty from the same cup twice on a single round.

We now prove a crucial property of greedy-like emptiers:

Lemma 2. *Let $R_\Delta = 2(2+\Delta)$. Consider an M-flat cup configuration in the p-processor E-extra-empties negative-fill cup game on $n = 2p$ cups. An oblivious filler can achieve a R_Δ-flat configuration of cups against a Δ-greedy-like emptier in running time $2(M+E)$. Furthermore, throughout this process the cup configuration is M-flat on every round.*

Proof. If $M \leq R_\Delta$ the algorithm does nothing, since the desired flatness is already achieved; for the rest of the proof we consider $M > R_\Delta$.

The filler's strategy is to distribute fill equally amongst all cups at every round, placing $p/n = 1/2$ fill 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 $\text{fill}_{S_t}(c) \leq l_t + 2 + \Delta$, and let U_t be the set of cups c with $\text{fill}_{S_t}(c) \geq u_t - 2 - \Delta$.

Now 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 Claim 7.

Claim 7.

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

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

If c is not emptied from, i.e. $\text{fill}(c)$ has increased by $1/2$ from the previous round, then clearly $c \in U_{t+1}$, because backlog has increased by at most $1/2$, so $\text{fill}(c)$ 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 $\text{fill}_{S_t}(c) \geq u_t - \Delta - 1$, then $\text{fill}_{S_t}(c)$ is at least 1 above the bottom of the interval defining which cups belong to U_t. The backlog increases by at most $1/2$ and the fill of c decreases by $1/2$, so $\text{fill}_{S_{t+1}}(c)$ is at least $1 - 1/2 - 1/2 = 0$ above the bottom of the interval, i.e. still in the interval.
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 because the emptier is Δ-greedy-like. Therefore the fullest cup on round $t + 1$ is the same as the fullest cup on round t, because every cup with fill in $[u_t - 1, u_t]$ has had its fill decrease by $1/2$, and no cup with fill less than $u_t - 1$ had its fill increase by more than $1/2$. Hence $u_{t+1} = u_t - 1/2$. Because both $\text{fill}(c)$ and the backlog have decreased by $1/2$, the distance between them is still at most $\Delta + 2$, hence $c \in U_{t+1}$.
The argument for why $L_t \subseteq L_{t+1}$ is symmetric. □

Now that we have shown that L_t and U_t never lose cups, we will show that they each eventually gain more than $n/2$ cups.

Claim 8. *On any round t, if $|U_t| \leq n/2$ then $u_{t+1} = u_t - 1/2$. On any round t where the emptier doesn't use extra resources, if $|L_t| \leq n/2$ then $\ell_{t+1} = \ell_t + 1/2$.*

Proof. If there are at least $n/2$ cups outside of U_t, i.e. cups with fills in $[\ell_t, u_t - 2 - \Delta]$, then all cups with fills in $[u_t - 2, u_t]$ must be emptied from; if one such cup was not emptied from then by the pigeon-hole principle some cup outside of U_t was emptied from, which is impossible as the emptier is Δ-greedy-like. This clearly implies that $u_{t+1} = u_t - 1/2$: no cup with fill less than $u_t - 2$ has gained enough fill to become the fullest cup, and the fullest cup from the previous rounds has lost $1/2$ units of fill.

Now consider a round where the emptier doesn't use extra resources and where $|L_t| \leq n/2$. Clearly no cup with fill in $[\ell_t, \ell_t + 2]$ can be emptied from; if one such cup were emptied from, then by the pigeonhole principle some cup outside of L_t was not emptied from, which is impossible as the emptier is Δ-greedy-like. Hence we have $\ell_{t+1} = \ell_t + 1/2$.

We remark however that we cannot guarantee that $\ell_{t+1} = \ell_t + 1/2$ on all rounds where $|L_t| \leq n/2$, because the emptier could do extra emptying; in the most extreme case the emptier could empty from every cup in which case we would have $\ell_{t+1} = \ell_t - 1/2$. □

We call a round where the emptier uses extra resources an ***emptier-extra-resource*** round. At most E of the $2(M + E)$ total rounds are emptier-extra-resource rounds. When the emptier uses extra resources it can potentially hurt the filler's efforts to achieve a flat configuration of cups, in particular by making $\ell_{t+1} < \ell_t$. However, the affect of emptier-extra-resource rounds is countered by rounds where the emptier does not use extra resources. In particular, we now define what it means for a non-emptier-extra-resource round j to cancel an emptier-extra-resource round $i < j$. For $i = 1, 2, \ldots, 2(M + E)$, if round i is an emptier-extra-resource round then the first non-emptier-extra-resource round $j > i$ that has not already cancelled some emptier-extra-resource round $i' < i$ in this sequential labelling process, provided such a round exists, is said to ***cancel*** round i. Each emptier-extra-resource round is cancelled by at most one later round, some emptier-extra-resource rounds may not be cancelled at all.

Consider rounds of the form $2M + i$ for $i \in [2E + 1] - 1$. We claim there is some i such that there are $2M$ non-emptier-extra-resource rounds among rounds $[2M + i]$ that are not cancelling other rounds. Assume for contradiction that this is not so. Then every non-emptier-extra-resource round $2M + i$ is necessarily a cancelling round. Hence by round $2(M + E)$, there must have been E cancelled tasks, so on round $2(M + E)$ all emptier-extra-resource rounds are cancelled.

Let t_e be some round by which there are $2M$ non-emptier-extra-resource, non-cancelling rounds. The value of u_t cannot have shrunk by more than M because the configuration started M-flat. Hence by Claim 8 there is some round $t_u \in [t_e]$ such that $|U_t| \geq n/2$. Identically, there is some round $t_\ell \in [t_e]$ such that $|L_t| \geq n/2$. Since by Claim 7 $|U_{t+1}| \geq |U_t|$ and $|L_{t+1}| \geq |L_t|$, we have that there is some round $t_0 = \max(t_u, t_\ell)$ on which both $|U_{t_0}|$ and $|L_{t_0}|$ exceed $n/2$. Then $U_{t_0} \cap L_{t_0} \neq \varnothing$. Furthermore, the sets must intersect for all $t_0 \leq t \leq [2(M + E)]$. 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

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

Or, rearranging,

$$u_t - \ell_t \leq 2(2 + \Delta) = R_\Delta.$$

Thus the cup configuration is R_Δ-flat.

Now we establish that throughout this flattening process the cup configuration is always M-flat. Consider a round where $u_{t+1} - \ell_{t+1} > u_t - \ell_t$. For this to happen the emptier must have used less than $n - p$ extra emptying, or else the fill of every cup would simply decrease by $1/2$ which wouldn't affect the difference $u_{t+1} - \ell_{t+1}$. In order for the difference $u_t - \ell_t$ to increase either a) some cup with fill in $[\ell_t, \ell_t + 1/2]$ was emptied from and some cup with fill in $[u_t - 1, u_t]$ was not emptied from, or b) some cup with fill in $[u_t - 1/2, u_t]$ was not emptied from and some cup with fill in $[\ell_t, \ell_t + 1]$ was emptied from. In either case, because the emptier is Δ-greedy-like, such an action implies

$$u_{t+1} - \ell_{t+1} \leq u_t + 1/2 - (\ell_t - 1/2) \leq \Delta + 5/2 \leq M.$$

Since the cup configuration starts M-flat, and after any round where the distance $u_t - \ell_t$ increases it increases to a value at most M, we have that the cups are always M-flat. □

Next we describe a simple oblivious filling strategy that will be used as a subroutine in Lemma 3; this strategy is very well-known, and similar versions of it can be found in [1, 2, 3, 4].

Proposition 3. *Consider an R-flat cup configuration in the negative-fill single-processor cup game on n cups with average fill 0. Let $d = \sum_{i=2}^{n} 1/i$.*

There is an oblivious filling strategy that achieves fill at least $-R + d$ in a randomly chosen cup with probability at least $1/n!$. This filling strategy guarantees that the chosen cup has fill at most $R + d$, and has running time $n - 1$. Further, when applied against a Δ-greedy-like emptier with $R = R_\Delta$, this filling strategy guarantees that the cups always remain $(R + d)$-flat.

Proof. The filler maintains an ***active set***, initialized to being all of the cups. Every round the filler distributes 1 unit of fill equally among all cups in the active set. Then the emptier removes 1 unit of fill from some cup. Finally, the filler removes a cup uniformly at random from the active set. This continues until a single cup c remains in the active set.

We now bound the probability that c has never been emptied from. On the i-th step of this process, i.e. when the size of the active set is $n - i + 1$, consider the cup c' that the emptier empties. If c' is in the active-set, then with probability at least $1/(n - i + 1)$ the filler removes it from the active set. If c' is not in the active set, then it is irrelevant. Hence with probability at least $1/n!$ the final cup in the active set, c, has never been emptied from. In this case, c will have gained fill $d = \sum_{i=2}^{n} 1/i$ as claimed. Because c started with fill at least $-R$, c now has fill at least $-R + d$.

Further, c has fill at most $R + d$, as c starts with fill at most R, and c gains at most $1/(n - i + 1)$ fill on the i-th round of this process.

Now we analyze this algorithm specifically for a Δ-greedy-like emptier. Consider a round on which the minimum fill of the cups becomes lower, i.e. where the emptier empties from some cup c with fill less than 1 above the lowest fill. On such a round, because the emptier is Δ-greedy-like, the backlog is no more than Δ greater than fill(c). Hence the cups are $(\Delta + 1)$-flat on such a round. If the cups are always $(\Delta+1)$-flat then we are done. Otherwise, consider the round after the last round on which the cups are $(\Delta+1)$-flat, (or the first round of the process if the cups are never $(\Delta+1)$-flat). From this round on, the emptier cannot empty a cup with fill within 1 of the fill of the lowest cup, hence the fill of the lowest cup cannot decrease. Consider how much the backlog could increase. The backlog could increase by d from whatever value it starts at. The backlog starts as at most $\max(\Delta + 1, R) = R$, and hence throughout the process the cups remain $(R + d)$-flat, as desired.

□

Now we are equipped to prove Lemma 3, which shows that we can force a constant fraction of the cups to have high fill; using Lemma 3 and exploiting the greedy-like nature of the emptier we can get a known cup with high fill (we show this in Proposition 4).

Lemma 3. *Let $\Delta \leq O(1)$, let $h \leq O(1)$ with $h \geq 16 + 16\Delta$, let n be at least a sufficiently large constant determined by h and Δ, and let $M \leq poly(n)$. Consider an M-flat cup configuration in the negative-fill variable-processor cup game on n cups with average fill 0. Let A, B, A' be disjoint constant-fraction-size subsets of the cups with $|A| = \Theta(n)$ sufficiently small and with $|A| + |B| + |A'| = n$. These sets will change over the course of the filler's strategy, but $|A|$ will remain fixed and $|A| \ll |A|$ will always hold.*

There is an oblivious filling strategy that makes an unknown set of $\Theta(n)$ cups in A have fill at least h with probability at least $1 - 2^{-\Omega(n)}$ in running time *poly(n) against a Δ-greedy-like emptier. The filling strategy also guarantees that $\mu(B) \geq -h/2$.*

Proof. We refer to A as the ***anchor*** set, B as the ***non-anchor*** set, and A' as the ***garbage*** set. Throughout the proof the filler uses $p = |A| + 1$. The filler initializes the sets as $A' = \varnothing$, and B is all the cups besides the cups in A. The set A is chosen to satisfy

$$|A| \leq (n - 2|A|)/(2e^{2h+1} + 1). \tag{12}$$

We denote by *randalg* the oblivious filling strategy given by Proposition 3. We denote by *flatalg* the oblivious filling strategy given by Lemma 2. We say that the filler ***applies*** a filling strategy *alg* to a set of cups $D \subseteq B$ if the filler uses *alg* on D while placing 1 unit of fill in each anchor cup.

We now describe the filler's strategy.

The filler starts by flattening the cups, i.e. using *flatalg* on $A \cup B$ for $2M$ rounds. After this, the filling strategy always places 1 unit of water in each anchor cup. The filler performs a series of $|A|$ ***swapping-processes***, one per anchor cup, which are procedures that the filler uses to get a new cup—which will sometimes have high fill—in A. On each swapping-process the filler applies *randalg* many times to arbitrarily chosen constant-size sets $D \subset B$ with $|D| = \lceil e^{2h+1} \rceil$. The number of times that the filler applies *randalg* is chosen at the start of the swapping-process, chosen uniformly at random from $[m]$ ($m = \text{poly}(n)$ to be specified). At the end of the swapping-process, the filler does a "swap": the filler takes the cup given by *randalg* in B and puts it in A, and the filler takes the cup in A associated with the current swapping-process and moves it into A'. Before each application of *randalg* the filler flattens B by applying *flatalg* to B for poly(n) rounds (exactly how many rounds will be specified later in the proof).

We remark that this construction is similar to the construction in Lemma 1, but has a major difference that substantially complicates the analysis: in the adaptive lower bound construction the filler halts after achieving the desired average fill in the anchor set, whereas the oblivious filler cannot halt but rather must rely on the emptier's greediness to guarantee that each application of *randalg* has constant probability of generating a cup with high fill.

We proceed to analyze our algorithm.

First note that the initial flattening of $A \cup B$ succeeds because the emptier is not allowed to do any extra emptying on $A \cup B$ so by setting $p = n/2$ the filler makes the flattening happen in the 0-extra-empties, $(n/2)$-processor cup game, which by Lemma 2 gets an R_Δ-flat cup configuration in running time $2M$.

We say that a property of the cups has ***always*** held if the property has held since the start of the first swapping-process; i.e. from now on we only consider rounds after the initial flattening of $A \cup B$.

We say that the emptier ***neglects*** the anchor set on a round if it does not empty from each anchor cup. We say that an application of *randalg* to $D \subset B$ is ***successful*** if the emptier does not neglect the anchor set during any round of the application of *randalg*. We define $d = \sum_{i=2}^{|D|} 1/i$ (recall that $|D| = \lceil e^{2h+1} \rceil$). We say that an application of *randalg* to D is ***lucky*** if it achieves backlog at least $\mu(B) - R_\Delta + d$; note that by Proposition 3 any successful application of *randalg* where B started R_Δ-flat has at least a $1/|D|!$ chance of being lucky.

Now we prove several important bounds on fills of cups in A and B.

Claim 9. *If all applications of flatalg so far have made B be R_Δ-flat, then B has always been $(R_\Delta + d)$-flat and $\mu(B)$ has always been at most* 1.

Proof. Given that the application of *flatalg* immediately prior to an application of *randalg* made B be R_Δ-flat, by Proposition 3 we have that B will stay $(R_\Delta + d)$-flat during the application of *randalg*. Given that the application of *randalg* immediately prior to an application of *flatalg* resulted in B being $(R_\Delta + d)$-flat, we have that B remains $(R_\Delta + d)$-flat throughout the duration of the application of *flatalg* by Lemma 2. Given that B is $(R_\Delta + d)$-flat before a swap occurs B is clearly still $(R_\Delta + d)$-flat after the swap, because the only change to B during a swap is that a cup is removed from B which cannot increase the backlog in B or decrease the fill of the least full cup in B. Note that B started R_Δ-flat before the first application of *flatalg* because $A \cup B$ was flattened. Hence we have by induction that B has always been $(R_\Delta + d)$-flat.

Now consider how high $\mu(B)$ could rise. The only time when $\mu(B)$ rises is at the end of a swapping-process. The cup that B evicts at the end of a swapping-process has fill at least $\mu(B) - R_\Delta - (|D| - 1)$, as the running time of *randalg* is $|D| - 1$, and B started R_Δ-flat by assumption. The highest that $\mu(B)$ can rise is clearly if a cup with fill as far below $\mu(B)$ as possible is evicted at every swapping-process. Evicting a cup with fill $\mu(B) - R_\Delta - (|D| - 1)$ from B changes $\mu(B)$ by $(R_\Delta + |D| - 1)/(|B| - 1)$ where $|B|$ is the size of B before the cup is evicted from B. Even if this happens on each of the $|A|$ swapping processes $\mu(B)$ cannot rise higher than $|A|(R_\Delta + |D| - 1)/(n - 2|A|)$ which by design in choosing $|B| \gg |A|$, as was done in (12), is at most 1.

□

Let $u_A = 1 + (R_\Delta + d) + \Delta + 1$, $\ell_A = -|B| - u_A \cdot (|A| - 1)$.

Claim 10. *If B has always been $(R_\Delta + d)$-flat and $\mu(B)$ has never exceeded 1, then the fills of cups in A have always been in $[\ell_A, u_A]$ and we have always had $\mu(B) > -h/2$.*

Proof. First consider how high the fill of a cup $c \in A$ could be. Let $u_B = 1 + (R_\Delta + d)$; note that the assumptions on B guarantee that u_B has always been an upper bound for the backlog in B. If c came from B then when it is swapped into A its fill is at most $u_B < u_A$. Otherwise, c started with fill at most $R_\Delta < u_A$. Now consider how much the fill of c could increase while being in A. Because the emptier is Δ-greedy-like, if a cup $c \in A$ has fill more than Δ higher than the backlog in B then c must be emptied from, so any cup with fill at least $u_B + \Delta = u_A - 1$ must be emptied from, and hence u_A upper bounds the backlog in A.

Of course an upper bound on backlog in A also serves as an upper bound on the average fill of A as well, i.e. $\mu(A) \le u_A$. Then, because $A \cup B$ has average fill 0, we have that

$$\mu(B) = -\frac{|A|}{|B|}\mu(A) \ge -u_A \frac{|A|}{|B|}. \qquad (13)$$

Note that $|B| \gg |A|$ so (13) is not very negative. In particular, by (12) (13) can be loosened to $\mu(B) \ge -h/2$.

Because $\mu(B) \le 1$ we have $\mu(A) \ge -|B|/|A|$. The mass of a subset of $|A| - 1$ of the cups is at most $(|A| - 1)u_A$, so we can lower bound the fill of any particular cup $c \in A$ by

$$\text{fill}(c) \ge -|B| - u_A \cdot (|A| - 1).$$

□

Let $r = |A|(\ell_A + u_A)$; note that $r = \Theta(n^2) = \text{poly}(n)$.

Claim 11. *If at the start of an application of flatalg B is $(R_\Delta + d)$-flat, $\mu(B) \le 1$, and all cups in A have fills in $[\ell_A, u_A]$ then by applying flatalg for $2((R_\Delta + d) + r)$ rounds, the filler guarantees that B will be R_Δ-flat at the end of the application of flatalg.*

Proof. If all cups in A start the application with fill at least ℓ_A and the emptier uses $r + 1$ extra empties during the application of *flatalg*, then by the pigeonhole principle some cup in A will have fill higher than u_A by the end of the application, as no cup in A loses water during the application of *flatalg*.

Lemma 2 says that B will remain $(R_\Delta + d)$-flat throughout the application of *flatalg*, and $\mu(B)$ obviously cannot rise during the application of *flatalg*. Hence throughout this process it will still be the case that $\mu(B) \le 1$ and that B is $(R_\Delta + d)$-flat. Then Claim 10 says that the backlog in A cannot have grown larger than u_A. Hence it is impossible for the emptier to have done $r + 1$ extra emptyings.

Thus, we can consider the application of *flatalg* as happening in the $(|B|/2)$-processor r-extra-empties cup game. By Lemma 2 we thus have that the cup configuration at the end of the application of *flatalg* will be R_Δ-flat by applying *flatalg* for $2((R_\Delta + d) + r)$ rounds. □

Now we combine Claim 9, Claim 10, and Claim 11 to get the following:

Claim 12. *All applications of flatalg make B be R_Δ-flat.*

Proof. This follows by induction on the flattening processes. Assume that all previous flattening processes have made B be R_Δ-flat. Then by Claim 9 we have that $\mu(B) \le 1$ has always held and that B has always been $(R_\Delta + d)$-flat. Thus by Claim 10 the fills of cups in A have always been in $[\ell_A, u_A]$. Thus by Claim 11 the next flattening successfully makes B be R_Δ-flat.

Note that the first application of *flatalg* makes B be R_Δ-flat because $A \cup B$ is flattened (so the application of *flatalg* lasts for 0 rounds, i.e. it finishes immediately). Hence by induction all applications of *flatalg* make B be R_Δ-flat. □

Now we show that this guarantees that with constant probability the final application of *randalg* on a swapping-process is both lucky and successful.

Claim 13. *There exists choice of $m = poly(n)$ such that with at least constant probability the final application of randalg on any fixed swapping-process is both lucky and successful.*

Proof. Fix some swapping-process. By Claim 12 we have that the fill of each cup in A starts the swapping-process with fill at least ℓ_A, and never exceeds u_A throughout the course of the swapping-process. This places an upper bound of r on the number of applications of *randalg* on which A can be neglected.

The filler chooses $m = 4r|D|!$. By a Chernoff bound, there is exponentially high probability that of $4r|D|!$ applications of *randalg* at least $2r$ would be lucky if the emptier didn't neglect A during the application. The emptier can choose at most r of these to neglect, so there is at least a $1/2$ chance that the randomly chosen final application of *randalg* is successful, conditioning on it lucky. The final application is lucky with probability $1/|D|!$. Hence overall this choice of m makes the final round lucky and successful with constant probability $1/(2|D|!)$. □

Claim 14. *With probability at least $1 - 2^{-\Omega(n)}$, the filler achieves fill at least h in at least $\Theta(n)$ of the cups in A.*

Proof. By Claim 13 each swapping-process has at least constant probability of swapping a cup with fill at least $\mu(B) + d - R_\Delta$ into A. The events that the swapping-processes swap such a cup into A are independent, so by a Chernoff bound there is exponentially high probability that at least a constant fraction of them succeed. By Claim 10 $\mu(B) \geq -h/2$. Recalling that $d \geq 2h$ and $h \geq 16(1 + \Delta)$, we have that such a swapped cup has fill at least h, as desired. □

We now briefly analyze the running time of the filling strategy. There are $|A|$ swapping-processes. Each swapping-process consists of poly(n) applications of *randalg*, which take constant time, and the poly(n) flattening procedure, which take poly(n) time. Clearly overall the algorithm takes poly(n) time, as desired.

□

Finally, using Lemma 3 we can show in Proposition 4 that an oblivious filler can achieve constant backlog. We remark that Proposition 4 plays a similar role in the proof of the lower bound on backlog as Proposition 1 does in the adaptive case, but is vastly more complicated to prove (in particular, Proposition 1 is trivial, whereas we have already proved several lemmas and propositions as preperation for the proof of Proposition 4).

Proposition 4. *Let $H \leq O(1)$, let $\Delta \leq O(1)$, let n be at least a sufficiently large constant determined by H and Δ, and let $M \leq poly(n)$. Consider an M-flat cup configuration in the negative-fill variable-processor cup game on n cups with average fill 0. Given this configuration, an oblivious filler can achieve fill H in a chosen cup in running time poly(n) against a Δ-greedy-like emptier with probability at least $1 - 2^{-\Omega(n)}$.*

Proof. The filler starts by performing the procedure detailed in Lemma 3, using $h = H \cdot 16(1 + \Delta)$. Let the number of cups which must now exist with fill h be of size $nc = \Theta(n)$.

The filler reduces the number of processors to $p = nc$. Now the filler exploits the filler's greedy-like nature to to get fill H in a set $S \subset B$ of nc chosen cups.

The filler places 1 unit of fill into each cup in S. Because the emptier is greedy-like it must focus on the nc cups in A with fill at least h until the cups in S have sufficiently high fill. In particular, $(5/8)h$ rounds suffice. Over $(5/8)h$ rounds the nc high cups in A cannot have their fill decrease below $(3/8)h \geq h/8 + \Delta$. Hence, any cups with fills less than $h/8$ must not be emptied from during these rounds. The fills of the cups in S must start as at least $-h/2$ as $\mu(B) \geq -h/2$. After $(5/8)h$ rounds the fills of the cups in S are at least $h/8$, because throughout this process the emptier cannot have emptied from them until they got fill at least $h/8$, and if they are never emptied from then they achieve fill $h/8$.

Thus the filling strategy achieves backlog $h/8 \geq H$ in some known cup (in fact in all cups in S, but a single cup suffices), as desired.

□

Next we prove the ***Oblivious Amplification Lemma***. The idea is quite similar to that of the Adaptive Amplification Lemma, but the proof is somewhat more complicated because the filler is oblivious.

Lemma 4 (Oblivious Amplification Lemma)**.** *Let $0 < \delta \ll 1/2, 1/2 \ll \phi < 1$ be constant parameters, and let $\eta \in \mathbb{N}$ be a function of ϕ. Let $\Delta \leq O(1)$, $M, M' \geq R_\Delta$. Let alg(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 $1 - 2^{-\Omega(n)}$ in running time $T(n) \leq poly(n)$ when given a M-flat configuration, against a Δ-greedy-like emptier.*

There exists an oblivious filling strategy alg(f') that achieves backlog $f'(n)$ satisfying

$$f'(n) \geq (1-\delta)(\phi - 1/(\delta n))(f(\lfloor (1-\delta)n \rfloor) - R_\Delta) + f(\lceil \delta n \rceil)$$

and $f'(n) \geq f(n)$, in the negative-fill variable-processor cup game on n cups with probability at least $1 - 2^{-\Omega(n)}$ in running time

$$T'(n) \leq O(M') + 6\delta n^{\eta+1} T(\lfloor (1-\delta)n \rfloor) + T(\lceil \delta n \rceil)$$

when given a M'-flat configuration of cups against a Δ-greedy-like emptier.

Proof. The algorithm defaults to using *alg*(f) on all the cups if doing so results in greater backlog than the strategy that we will outline in the rest of the proof; in this case applying *alg*(f) to all the cups trivially achieves the desired backlog in the desired running time. We now outline the filler's strategy if this is not the case.

The filler starts by flattening all the cups, using the flattening procedure detailed in **??**.

Let A, the ***anchor*** set, be a subset of $\lceil \delta n \rceil$ cups chosen arbitrarily, and let B, the ***non-anchor*** set, consist of the rest of the cups ($|B| = \lfloor (1-\delta)n \rfloor$). 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 *alg*(f) repeatedly on B, achieve a cup with fill $\mu(B) + f(|B|)$ in B and then swap this cup into A.
Step 2: Use *alg*(f) once on A to obtain a cup in A with fill $\mu(A) + f(|A|)$.
Note that in order to use *alg*(f) on subsets of the cups the filler will need to vary p.

We now describe how to achieve Step 1, which is complicated by the fact that the emptier may attempt to prevent the filler from achieving high fill in a cup in B, and further by the fact that the filler, being oblivious, can not know if the emptier has done this. In particular, Step 1 may not succeed time, but we show that with exponentially good probability is works almost every time.

The filler's strategy will be to always place 1 fill in each cup in the anchor-set while applying *alg*(f) to B.

For each cup in A the filler performs a procedure called a ***swapping-process***. Let A_0 be initialized to $\varnothing$; during each swapping-process the filler will get some cup in B to have high fill (with very good probability), and then swap this cup into A, and place the cup in A_0 too. We say that the filler ***applies*** *alg*(f) to B if it follows the filling strategy *alg*(f) on B while placing 1 unit of fill in each anchor cup; during a swapping-process the filler repeatedly applies *alg*(f) to B, flattening $B \cup (A \setminus A_0)$ and then flattening B too before each application. We say that the emptier ***neglects*** the anchor set on a round if the emptier does not empty from every anchor cup on this round. The mass of the anchor set increases by at least 1 each round that the anchor set is neglected. An application of *alg*(f) to B is said to be ***successful*** if A is never neglected during the application of *alg*(f) to B. We say that a swapping-process is ***successful*** if the application of *alg*(f) on which the filler swaps a cup into A is a successful application of *alg*(f).

Let $\mu_\Delta = 2R_\Delta + \Delta$; the emptier, being Δ-greedy-like, cannot neglect the anchor set more than $n\delta\mu_\Delta$ times. Thus, by making each swapping-process consist of n^η applications of *alg*(f) to B and then choosing a single application among these (uniformly at random) after which to swap a cup into A (and we also place the cup in A_0; A_0 consists of all cups in A that were swapped into A from B), we guarantee that with probability at least $n\delta\mu_\Delta / n^\eta$ this swap occurs at the end of a successful application of *alg*(f) to B.

If an application of *alg*(f) is successful, then with probability at least $1 - 2^{-\Omega(n)}$ it generates a cup with fill $f(|B|) + \mu(B)$ in B, because equal resources were put into B on each round while *alg*(f) was used, and the cup state started as R_Δ-flat (relative to $\mu(B)$) and hence also started as M-flat (as $M \geq R_\Delta$).

Now we aim to show that $\mu(A)$ is large; we do so by showing that $\mu(B)$ is small (i.e. very negative). Because the probability of an application of *alg*(f) being successful is only $1 - 1/\operatorname{poly}(n)$, which is in particular not as good as the $1 - 2^{-\Omega(n)}$ that we will guarantee, we will not be able to actually assume that every such application of *alg*(f) is successful. However, (as we will show later) we can guarantee that at least a constant fraction ϕ of the swapping-processes are successful with exponentially good probability.

The filler swaps δn cups into B. Consider how $\mu(B \cup A \setminus A_0)$ changes when a new cup is swapped into A and placed in A_0. Let initial value of $\mu(B \cup A \setminus A_0)$ be μ_0. Say that initially $|A_0| = i$ (i.e. i swapping processes have occured so far). If the swapping-process is successful then the swapped cup has fill at least $\mu_0 - R_\Delta + f(|B|)$. Hence the new average fill of $B \cup A \setminus A_0$ after the swap is

$$\frac{\mu_0 \cdot (n-i) - (\mu_0 - R_\Delta + f(|B|))}{n-i-1} = \mu_0 - \frac{f(|B|) - R_\Delta}{n-i-1}.$$

This recurrence relation allows us to find the value of $\mu(B \cup A \setminus A_0) = \mu(B)$ after $|A|$ swapping processes (i.e. once $A \setminus A_0 = \varnothing$):

$$\mu(B) \leq -\sum_{i=1}^{|A|\phi} \frac{f(|B|) - R_\Delta}{n-i}.$$

Now we bound $H_{n-1} - H_{n-|A|\phi-1}$ where H_i is the i-th harmonic number. Using the fact that

$$H_n = \ln n + \gamma + 1/(2n) - 1/(12n^2) + 1/(120n^4) - \dots$$

we have,

$$\begin{aligned}
&H_{n-1} - H_{n-|A|\phi-1} \\
&\geq \ln \frac{n-1}{n-|A|\phi-1} - \frac{1}{2(n-|A|\phi-1)} \\
&\geq \ln \frac{n}{n-|A|\phi} - \frac{1}{n} \\
&= \ln \frac{n}{n - \lceil \delta n \rceil \phi} - \frac{1}{n} \\
&\geq \ln \frac{1}{1-\delta\phi} - \frac{1}{n} \\
&\geq \delta\phi - \frac{1}{n}.
\end{aligned}$$

Hence we have,

$$\mu(A) \geq \frac{(1-\delta)}{\delta}\left(\delta\phi - \frac{1}{n}\right)(f(|B|) - R_\Delta). \quad (14)$$

Now we establish that we can guarantee that $\phi|A|$ of the $|A|$ swapping-process succeed for any choice of $\phi = \Theta(1)$ by sufficiently large choice of η, i.e. by performing enough applications of *alg*(f) within each swapping-process. Recall that by construction of μ_Δ the emptier cannot neglect the anchor set on more than $n\delta\mu_\Delta$ applications of *alg*(f) to B.

Let X_i be the random variable that indicates the event that the i-th swapping-process was not successful; note that the X_i are independent, because the filler's random choices of which applications of *alg*(f) within each swapping-process on which to swap a cup into the anchor set are independent. We have, for any constant ϕ,

$$\Pr\left[\left|\frac{1}{|A|}\sum_{i=1}^{|A|} X_i - \frac{n\delta\mu_\Delta}{n^\eta}\right| \geq 1 - 2\phi\right] \leq 2e^{-2|A|(1-2\phi)^2}$$
$$\leq 2^{-\Omega(n)}.$$

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

$$n\delta\mu_\Delta / n^\eta \leq \phi$$

no matter how small $w \geq \Omega(1)$ is chosen. In particular this implies that

$$\Pr\left[\sum_{i=1}^{|A|} X_i \geq |A|(1-\phi)\right] \geq 1 - 2^{\Omega(n)}.$$

That is, with exponentially good probability $|A|\phi$ of the swapping processes succeed. Taking a union bound over all applications of *alg*(f) we have that there is exponentially good probability that all applications of *alg*(f) succeeded. Thus, with exponentially good probability, by (14), Step 1 achieves backlog

$$(1-\delta)(\phi - 1/(\delta n)(f(\lfloor(1-\delta)n\rfloor - R_\Delta)$$

To achieve Step 2 the filler simply applies *alg*(f) to A. This clearly achieves backlog

$$f(|A|) = f(\lceil\delta n\rceil)$$

with exponentially good probability.

Since both Step 1 and Step 2 succeed with exponentially good probability, the entire process succeeds with exponentially good probability.

We now analyze the running time of *alg*$(()f')$. The initial smoothing takes time $O(M')$. Step 1 entails $n^\eta \cdot (n\delta)$ swapping-processes, each of which takes time $f(|B|)$. Due to flattening at the beginning of each application of *alg*(f) the running time may be increased by a multiplicative factor of at most 6. Step 2 takes time $T(|A|)$. Adding these times we have that the running time $T'(n)$ of *alg*(f') is

$$T'(n) \leq O(M') + 6\delta n^{\eta+1}T(\lfloor(1-\delta)n\rfloor) + T(\lceil\delta n\rceil).$$

Having proved that *alg*(f') achieves the desired backlog with the desired probability in the desired running time, the proof is now complete.

□

Finally we prove that an oblivious filler can achieve backlog $n^{1-\varepsilon}$. The proof is very similar to the proof of Theorem 1, but more complicated because in the oblivious case we must guarantee that the result holds with good probability, and also more complicated because the Oblivious Amplification Lemma is more complicated than the Adaptive Amplification Lemma. We remark that it is quite remarkable that an oblivious filler is still able to achieve poly(n) backlog, just as an adaptive filler can, because intuitively being oblivious is a large disadvantage.

Theorem 3. *There is an oblivious filling strategy for the variable-processor cup game on n cups that achieves backlog at least $\Omega(n^{1-\varepsilon})$ for any constant $\varepsilon > 0$ in running time $2^{O(\log^2 n)}$ with probability at least $1-2^{-\Omega(n)}$ against any Δ-greedy-like emptier for $\Delta \leq O(1)$.*

Proof. Take constant $\varepsilon \in (0, 1/2)$. We aim to achieve backlog $(n/n_b)^{1-\varepsilon} - 1$ for some constant n_b on n cups. Let δ, ϕ be constants, chosen as functions of ϵ.

By Proposition 4 there is an oblivious filling strategy that achieves backlog $\Omega(1)$ on n cups with exponentially good probability in n; we call this algorithm *alg*(f_0). However, unlike in the proof of Theorem 1, we obviously cannot use the base case with a constant number of cups: doing so would completely destroy our probability of success! Because the running time of our algorithm will be $2^{\text{polylog}(n)}$, we will be required to take a union bound over $2^{\text{polylog}(n)}$ events. By making the size of our base case $n_b = \text{polylog}(n)$ we get that the probability of the algorithm failing in the base case is at most $2^{-\text{polylog}(n)}$. Then, taking a union bound over $2^{\text{polylog}(n)}$ events can give us the desired probability. By Proposition 4 *alg*(f_0) achieves backlog $f_0(k) \geq H \geq \Omega(1)$ for all $k \geq n_b$, for some constant $H \geq \Omega(1)$ to be determined (H is a function of δ).

Then we construct f_{i+1} as the amplification of f_i using Lemma 4.

Define a sequence g_i as

$$g_i = \begin{cases} n_b \lceil 16/\delta \rceil, & i = 0 \\ \lfloor g_{i-1}/(1-\delta) \rfloor, & i \geq 1 \end{cases}.$$

We claim the following regarding our construction:

Claim 15.

$$f_i(k) \geq (k/n_b)^{1-\varepsilon} - 1 \textit{ for all } k \leq g_i. \qquad (15)$$

Proof. We prove Claim 15 by induction on i.

First we derive a simpler (more loose) form of the lower bound on $alg(f')$'s backlog in terms of $alg(f)$'s backlog that hold if $\lfloor (1-\delta)n \rfloor \geq n_b$. We choose $n_b = \text{polylog}(n)$ making $n_b > 1/\delta^2$ and hence also $\delta > 1/(\delta n_b)$; this means that there is a choice of $\phi \in (1/2, 1)$ making $\phi - 1/(\delta n_b) > 1 - \delta$. Note that for any $n \geq n_b$ this same ϕ satisfies

$$(1-\delta) \leq \phi - \frac{1}{\delta n_b} \leq \phi - \frac{1}{\delta n}.$$

We choose $\phi = 1 - \delta + 1/(\delta n_b)$. Further, we choose $H \geq \Omega(1)$ to make

$$H - R_\Delta \geq (1-\delta)H.$$

This ensures that

$$f_0(\lfloor (1-\delta)n \rfloor) - R_\Delta \geq (1-\delta) f_0(\lfloor (1-\delta)n \rfloor)$$

so long as $\lfloor (1-\delta)n \rfloor \geq n_b$. Combining this, we have that if $\lfloor (1-\delta)n \rfloor \geq n_b$ then

$$f'(n) \geq (1-\delta)^3 f(\lfloor (1-\delta)n \rfloor) + f(\lceil \delta n \rceil). \qquad (16)$$

We also choose H large enough so that $H \geq (g_0/n_b)^{1-\epsilon} - 1 = \lceil 16/\delta \rceil^{1-\epsilon} - 1$.

When $i = 0$, the base case of our induction, (15) is trivially true as $(k/n_b)^{1-\epsilon} - 1 \leq H$ by definition of H for $k \leq g_0$.

Assume (15) for f_i, consider f_{i+1}.

Note that, by design of g_i, if $k \leq g_{i+1}$ then $\lfloor k \cdot (1-\delta) \rfloor \leq g_i$. Consider any $k \in [g_{i+1}]$.

First we deal with the trivial case where $k \leq g_0$. In this case

$$f_{i+1}(k) \geq f_i(k) \geq \cdots \geq f_0(k) \geq (k/n_b)^{1-\varepsilon} - 1.$$

Now we consider $k \geq g_0$. Note that in this case $\lfloor (1-\delta)k \rfloor \geq n_b$. Since f_{i+1} is the amplification of f_i, and k is sufficiently large, we have by (16) that

$$f_{i+1}(k) \geq (1-\delta)^3 f_i(\lfloor (1-\delta)k \rfloor) + f_i(\lceil \delta k \rceil).$$

By our inductive hypothesis, which applies as $\lceil \delta k \rceil \leq g_i$, $\lfloor k \cdot (1-\delta) \rfloor \leq g_i$, we have

$$f_{i+1}(k) \geq (1-\delta)^3 (\lfloor (1-\delta)k/n_b \rfloor^{1-\varepsilon} - 1) + \lceil \delta k/n_b \rceil^{1-\varepsilon} - 1.$$

Dropping the floor and ceiling, incurring a -1 for dropping the floor, we have

$$f_{i+1}(k) \geq (1-\delta)^3 (((1-\delta)k/n_b - 1)^{1-\varepsilon} - 1) + (\delta k/n_b)^{1-\varepsilon} - 1.$$

Because $(x-1)^{1-\varepsilon} \geq x^{1-\varepsilon} - 1$, due to the fact that $x \mapsto x^{1-\varepsilon}$ is a sub-linear sub-additive function, we have

$$f_{i+1}(k) \geq (1-\delta)^3 (((1-\delta)k/n_b)^{1-\varepsilon} - 2) + (\delta k/n_b)^{1-\varepsilon} - 1.$$

Moving the $(k/n_b)^{1-\varepsilon}$ to the front we have

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} \cdot \left((1-\delta)^{4-\varepsilon} + \delta^{1-\varepsilon} - \frac{2(1-\delta)^3}{(k/n_b)^{1-\varepsilon}} \right) - 1.$$

Because $(1-\delta)^{4-\varepsilon} \geq 1 - (4-\varepsilon)\delta$, a fact called Bernoulli's Identity, we have

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} \cdot \left(1 - (4-\varepsilon)\delta + \delta^{1-\varepsilon} - \frac{2(1-\delta)^3}{(k/n_b)^{1-\varepsilon}} \right) - 1.$$

Of course $-2(1-\delta)^3 \geq -2$, so

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} \cdot \left(1 - (2-\varepsilon)\delta + \delta^{1-\varepsilon} - 2/(k/n_b)^{1-\varepsilon} \right) - 1.$$

Because

$$-2/(k/n_b)^{1-\varepsilon} \geq -2/(g_0/n_b)^{1-\varepsilon} \geq -2(\delta/16)^{1-\varepsilon} \geq -\delta^{1-\varepsilon}/2,$$

which follows from our choice of $g_0 = \lceil 8/\delta \rceil n_b$ and the restriction $\varepsilon < 1/2$, we have

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} \cdot \left(1 - (4-\varepsilon)\delta + \delta^{1-\varepsilon} - (1/2)\delta^{1-\varepsilon} \right) - 1.$$

Finally, combining terms we have

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} \cdot \left(1 - (4-\varepsilon)\delta + (1/2)\delta^{1-\varepsilon} \right) - 1.$$

Because $\delta^{1-\varepsilon}$ dominates δ for sufficiently small δ, there is a choice of $\delta = \Theta(1)$ such that

$$1 - (4-\varepsilon)\delta + (1/2)\delta^{1-\varepsilon} \geq 1.$$

Taking δ to be this small we have,

$$f_{i+1}(k) \geq (k/n_b)^{1-\varepsilon} - 1,$$

completing the proof. □

The sequence g_i is n_b times the sequence g_i from the proof of Theorem 1; we thus have that $g_{i_*} \geq n$ for some $i_* \leq O(\log n)$. Hence $alg(f_{i_*})$ achieves backlog

$$f_{i_*}(n) \geq (n/n_b)^{1-\varepsilon} - 1.$$

As $n_b \leq \text{polylog}(n)$ we have

$$f_{i_*}(n) \geq \Omega(n^{1-\varepsilon}),$$

as desired.

Let the running time of $f_i(n)$ be $T_i(n)$. From the Amplification Lemma we have following recurrence bounding $T_i(n)$:

$$\begin{aligned} T_i(n) &\leq 6n^{\eta+1}\delta \cdot T_{i-1}(\lfloor(1-\delta)n\rfloor) + T_{i-1}(\lceil \delta n \rceil) \\ &\leq 7n^{\eta+1}T_{i-1}(\lfloor(1-\delta)n\rfloor). \end{aligned}$$

It follows that $alg(f_{i_*})$, recalling that $i_* \leq O(\log n)$, has running time

$$T_{i_*}(n) \leq (7n^{\eta+1})^{O(\log n)} \leq 2^{O(\log^2 n)}$$

as desired.

As noted, because the running time is $2^{\text{polylog}(n)}$ and the base case size is $n_b \geq \text{polylog}(n)$, a union bound guarantees the probability of success is at least $1 - 2^{-\text{polylog}(n)}$. □

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] Michael A Bender, Martín Farach-Colton, and William Kuszmaul. Achieving optimal backlog in multi-processor cup games. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1148–1157, 2019.

[2] Michael A Bender, Sándor P Fekete, Alexander Kröller, Vincenzo Liberatore, Joseph SB Mitchell, Valentin Polishchuk, and Jukka Suomela. The minimum backlog problem. *Theoretical Computer Science*, 605:51–61, 2015.

[3] Paul Dietz and Rajeev Raman. Persistence, amortization and randomization. 1991.

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