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 $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ 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 $\Omega(n^{1-\epsilon})$ for $\epsilon > 0$ 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 $\Omega(\log \log n)$.

Recently Kuszmaul has established bounds on the case where $p > 1$, i.e. the ***multi-processor cup game*** [2]. 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

*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.

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 $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ 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 $\Omega(n^{1-\epsilon})$ for any constant $\epsilon > 0$ 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, 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; 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 $[n] = \{1, 2, \dots, n\}$, let $i+[n] = \{i+1, i+2, \dots, i+n\}$.

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 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.

[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.

3 Adaptive Filler Lower Bound

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

We begin by modifying the proof of the classic lower bound for the single-processor cup game to work with a cup configuration that includes negative fills; allowing for negative fills will be important when applying the result in later constructions.

Proposition 1. *Consider an instance of the negative-fill 1-processor cup game on n cups, and let the cups start in any state where the average fill is 0. There is an $O(n)$-step adaptive filling strategy that achieves backlog at least $\frac{1}{4}(\ln(n/2) - 1)$.*

Proof. Let $h = \frac{1}{4}(\ln(n/2) - 1)$ be the desired fill. Once a cup with fill at least h is achieved the filler stops, and the algorithm is done.

We claim that, at the beginning of the game, either the algorithm is already done, or there are at least $n/2$ cups containing fills greater than $-2h$. If the filling process is not yet complete then $\text{fill}(c) < h$ for all cups c, and thus the mass of the cups with non-negative fills is less than hn. Assume for contradiction that at least $n/2$ cups have fills $-2h$ or smaller. The mass of those cups is less than or equal to $-hn$. Since the mass of the cups with non-negative fills is less than hn, it follows that the total mass of the cups is negative, a contradiction.

We now describe the filler's strategy. If the algorithm is not already done at the beginning of the game, then at least $n/2$ cups must have fills greater than $-2h$. At the beginning of the algorithm, $n/2$ such cups are labeled as ***active cups***. In each of step of the algorithm, the filler distributes 1 unit of water evenly among the active cups. The emptier then removes water from some cup c. Finally, the filler reduces the number of active cups by 1, removing cup c if it was active, and otherwise removing an arbitrary cup. Now consider the state of the final active cup after $n/2 - 1$ steps. This cup's fill has increased by $1/(n/2) + 1/(n/2-1) + \cdots + 1/2 \geq \ln(n/2) - 1 = 4h$. This cup's fill started as at least $-2h$. Thus this cup now has fill at least $2h$, 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 $\text{alg}_{f'}$ 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 $\text{alg}_{f'}$, the filling strategy created by the Amplification Lemma, the ***amplification*** of alg_f. As suggested by the name, $\text{alg}_{f'}$ 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 $\text{alg}_{f_0}, \text{alg}_{f_1}, \ldots, \text{alg}_{f_{i_*}}$ with $\text{alg}_{f_{i+1}}$ 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 step, 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

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

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

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

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 $\mu(A)\ge h$.

Now the filler performs Step 2, i.e. the filler applies alg_f to A, and hence achieves a cup with fill at least

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

as desired.

Now we analyze the running time of the filling strategy $\text{alg}_{f'}$. First, recall that in Step 1 $\text{alg}_{f'}$ 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

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

□

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

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 $2^{O(\log^2 n)}$.*

Proof. Take constant $\epsilon\in(0,1/2)$. 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 $cn^{1-\epsilon}-1$.

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 $f_0(k)\ge 1$ for all $k\ge n_0$.

Next, using the Amplification Lemma we recursively construct $\text{alg}_{f_{i+1}}$ as the amplification of alg_{f_i} for $i\ge 0$.

Define a sequence g_i with

$$g_i = \begin{cases}\lceil 4/\delta\rceil, & i=0,\\ \lfloor g_{i-1}/(1-\delta)\rfloor & i\ge 1\end{cases}$$

We claim the following regarding this construction:

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

$$f_i(k)\ge ck^{1-\epsilon}-1 \;\; \textit{for all} \;\; k\in[g_i]. \qquad (2)$$

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 $cn_0^{1-\epsilon}-1\le 0$, which implies (2) holds for $k\in[n_0]$ by monotonicity of $ck^{1-\epsilon}-1$; we also choose δ small enough to make $g_0\ge n_0$, and we choose c small enough to make $cg_0^{1-\epsilon}-1\le f_0(g_0)=1$, which implies (2) holds for $k\in[n_0,g_0]$ by monotonicity of $ck^{1-\epsilon}-1$. [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 $k\le g_{i+1}$ then $\lfloor k\cdot(1-\delta)\rfloor\le g_i$. Consider any $k\in[g_{i+1}]$. First we deal with the trivial case where $k\le g_0$. In this case

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

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

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

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

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

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

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

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

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

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

$$f_{i+1}(k)\ge ck^{1-\epsilon}\cdot\big((1-\delta)^{2-\epsilon}+\delta^{1-\epsilon}-2(1-\delta)/k^{1-\epsilon}\big)-1.$$

Because $(1-\delta)^{2-\epsilon}\ge 1-(2-\epsilon)\delta$, which follows by inspection of the first three terms of the Taylor Series of $(1-\delta)^{2-\epsilon}$, we have

$$f_{i+1}(k)\ge ck^{1-\epsilon}\cdot\big(1-(2-\epsilon)\delta+\delta^{1-\epsilon}-2(1-\delta)/k^{1-\epsilon}\big)-1.$$

By choosing $\delta<1/2$ we make $-2(1-\delta)\ge -1$, so we have

$$f_{i+1}(k)\ge ck^{1-\epsilon}\cdot\big(1-(2-\epsilon)\delta+\delta^{1-\epsilon}-1/k^{1-\epsilon}\big)-1.$$

Because $-1/k^{1-\epsilon}\ge -1/g_0^{1-\epsilon}\ge -(\delta/4)^{1-\epsilon}\ge -\delta^{1-\epsilon}/2$, which follows from our choice of $g_0=\lceil 4/\delta\rceil$ and the restriction $\epsilon<1/2$, we have

$$f_{i+1}(k)\ge ck^{1-\epsilon}\cdot\big(1-(2-\epsilon)\delta+\delta^{1-\epsilon}-(1/2)\delta^{1-\epsilon}\big)-1.$$

Finally, combining terms we have

$$f_{i+1}(k)\ge ck^{1-\epsilon}\cdot\big(1-(2-\epsilon)\delta+(1/2)\delta^{1-\epsilon}\big)-1.$$

Because $\delta^{1-\epsilon}$ dominates δ for sufficiently small δ, there is a choice of $\delta=\Theta(1)$ such that $1-(2-\epsilon)\delta+(1/2)\delta^{1-\epsilon}\ge 1$. Taking δ to be this small we have,

$$f_{i+1}(k)\ge ck^{1-\epsilon}-1,$$

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

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. □

We claim that g_i grows exponentially, so after recursing with this process for $i_* \le O(\log n)$ levels of recursion we have $g_{i_*} \ge n$ and hence we have an algorithm $\text{alg}_{f_{i_*}}$ that achieves backlog $cn^{1-\epsilon} - 1$ 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

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

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

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

as desired.

Let the running time of $f_{i_*}(n)$ be $T(n)$. From the Amplification Lemma we have following recurrence bounding $T(n)$:

$$T(n) \le 2n^2 \cdot T(n(1-\delta)).$$

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

$$T(n) \le (2n^2)^{O(\log n)} \le 2^{O(\log^2 n)}$$

as desired.

□

Now we provide a very simple construction that can achieve backlog $\Omega(n)$ in very long games. Although the running time is very large, it is still an interesting result. Further, it can be interpreted as the same argument 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 $f_0(k) \ge 1$ on $k \ge n_0$ cups, which is possible by Proposition 1. For $i > 0$ we construct alg_{f_i} as the amplification of $\text{alg}_{f_{i-1}}$ using the Amplification Lemma with parameter $\delta = 1/(i+1)$.

We claim the following regarding this constuction:

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

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

Proof. We prove Claim 2 by induction on i. When $i = 0$, the base case, (3) becomes $f_0(n_0) \ge 1$ 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

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

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

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

Using the inductive hypothesis we have

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

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

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

as desired. □

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

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

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

The recurrence for the running running time of f_{i_*} is

$$T(n) \le n \cdot n_0 T(n - n_0) + O(1).$$

Clearly $T(n) \le O(n!)$.

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. □

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.

[4] Or more precisely, setting δ to be a constant divided by the number of cups at the level of recursion; note in particular that δ changes at the different levels of recursion as opposed to in the proof of Theorem 1.

$S = S_t$ or $S = I_t$).[5] Let $S(\{r_1, \dots, r_m\})$ denote the set of cups of ranks $r_1, r_2, \dots, r_m$ 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 $t = 1$ (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 $a+b+1, \dots, k+b$ 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_t^r by relabelling only cups with ranks in $[r]$ such that (a) S_t^r satisfies the invariants (4), (b) $S_t^r([r]) = I_t([r])$ and (c) the filler can legally place water into cups in order to transform S_t^r 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_t^r for which $S_t^r([r]) = I_t([r])$. Note that S_t^r is simply a relabeling of S_t, hence it must satisfy the same invariants (4) satisfied by S_t. Further, S_t^r 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_t^{n-1}, and to get to state S_t^r from state S_t^{r+1} apply the process described above. Note that S_t^{n-1} satisfies $S_t^{n-1}([n-1]) = I_t([n-1])$ and thus also $S_t^{n-1}(n) = I_t(n)$. If S_t^{r+1} satisfies $S_t^{r+1}(r') = I_t(r')$ for all $r' > r + 1$ then S_t^r satisfies $S_t^r(r') = I_t(r')$ for all $r > r$, because the transition from S_t^{r+1} to S_t^r has not changed the labels of any cups with ranks in $(r+1, n]$, but the transition does enforce $S_t^r([r]) = I_t([r])$, and consequently $S_t^r(r + 1) = I_t(r + 1)$. We continue with the sequential process until arriving at state S_t^1 in which we have $S_t^1(r) = I_t(r)$ for all r. Throughout the process each S_t^r has satisfied the invariants (4), so S_t^1 satisfies the invariants (4). Further, throughout the process from each S_t^r it is possible to legally place water into cups in order to transform S_t^r into I_t.

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

□

[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.

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 $\text{fill}_{S_{t+1}}(I_{t+1}(a)) = 0$

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

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

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

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 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

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

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

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:

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

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 $t \in \mathbb{N}$. □

5 Oblivious Filler Lower Bound

In this section we prove a lower bound on backlog with an oblivious filler.

Call a cup configuration M***-flat*** if the fill of every cup is in the interval $[-M, M]$. Call an emptier Δ***-greedy-like*** if whenever there are two cups c_1, c_2 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 $\Delta \leq O(1)$. 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. Nonetheless, greedy-like emptiers are of great interest; for example, a greedy emptier (which is 0-greedy-like) is used in the upper bound proof.

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

Proposition 3. *Given a cup configuration that is M-flat, an oblivious filler can achieve a $2(2+\Delta)$-flat configuration of cups against a Δ-greedy-like emptier in running time $2M$ 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 $1/2$ 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 $\text{fill}_{S_t}(c) \geq u_t - 2 - \Delta$.

There are two ways to think of U_t. First, U_t contains cups in the union of intervals of length 1, Δ, and 1. This ensures that if any 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. Second, U_t contains cups with fill in the union of $[u_t - 2, u_t]$ and $[u_t - \Delta - 2, u_t - 2]$. Thus 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.

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 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 $\text{fill}_{S_t}(c) \geq u_t - \Delta - 1$, then $\text{fill}_{S_t}(c)$ is at least 1 above the bottom of the interval. 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. 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 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. *As long as $|U_t| \leq n/2$ we have $u_{t+1} = u_t - 1/2$. Identically, as long as $|L_t| \leq n/2$ 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 symmetric for L_t. □

By Claim 8 we see that both $|U_t|$ and $|L_t|$ must eventually exceed $n/2$ at some times $t_u, t_\ell \leq 2M$, by the assumption that the initial configuration is M-flat. Since by Claim 7 $|U_{t+1}| \geq |U_t|$ and $|L_{t+1}| \geq |L_t|$, we have that there is some round $t_0 = \max(t_u, t_\ell) \leq 2M$ 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 $t_0 \leq t \leq 2M$. 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

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

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

□

Given a Δ-greedy-like filler, let $R_\Delta = \lceil 2(2+\Delta) \rceil$. [7] By Proposition 3, 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 4. *Let $H \leq O(1)$, $M \leq$ poly(n), $\Delta \leq O(1)$, $n \geq \Omega(1)$ at least a sufficiently large constant.*

There exists an oblivious filling strategy for the negative-fill variable-processor cup game on n cups that achieves 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 the cups, using the flattening procedure detailed in Proposition 3.

Let A, the ***anchor*** set, be an arbitrary (e.g. randomly chosen) subset of $n/32$ cups and let B, the ***non-anchor*** set, consist of the rest of the cups ($|B| = n{\cdot}31/32$). 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 sets $p = |A| + 1$. 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 $|A|$ ***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 $|A|{\cdot}c_\Delta$ 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-block j uniformly at random from $[|A|{\cdot}c_\Delta]$.

For each round-block $i \in [|A| \cdot c_\Delta]$, 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.

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).

Now we formally prove that the Step 1 succeeds (with exponentially good probability).

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 $[|A| \cdot c_\Delta]$ (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.

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 $\ln\lceil e^{2h} \rceil \geq 2h$.

[7] It is convenient to have this be an integer, and there is no drawback to taking a slightly larger R_Δ than necessary. In fact, this value of R_Δ is already not tight.

Consider what this cups fill started as at the beginning of the round-block. By the flattening it was within $\pm\Delta$ of $\mu(B)$. However, $\mu(B) \geq -R_\Delta$ isn't necessarily true; although before any swapping-process were performed the entire set of cups was R_Δ flat, the filler is moving cups from B –hopefully with high fill– into A which could cause the average fill of B to sink. However, it cannot have sunk very much. At most, if every swapping process succeeds, we would still have less than, say, $|A| \cdot 4h$ mass removed from B. Thus the average fill of B is certainly never less than

$$\ell_b = -R_\Delta - \frac{|A|}{|B|}4h = -R_\Delta - \frac{1}{31}4h \geq -h/2. \quad (12)$$

Thus, a cup with fill that has increased by at least $2h$ from the start of the round-block has fill at least h.

It is also useful to know how high $\mu(B)$ could possible rise, which will help us bound how far $\mu(A)$ could possibly sink. A cup removed from B has fill strictly more than $s = -h - R_\Delta - \lceil e^{2h} \rceil$. The fills of cups in A have fill no more than R_Δ initially. Hence $\mu(B)$ can certainly rise no higher than $u_b = \frac{|A|}{|B|}(f_b + R_\Delta) + R_\Delta$. It is conceivable that $\mu(A)$ could sink during this process. In particular, the cup swapped into A from B might have quite negative fill: if the emptier emptied from it on each round of the round-block the cups fill would have decreased by strictly less than $\lceil e^{2h} \rceil$. As $\mu(B)$ never drops below $-h$ and B starts each single-processor cup game R_Δ-flat, we have that any cup swapped into A has fill at least s. We thus have (the quite loose) lower bound that $\mu(A)$ can certainly never drop below $\ell_a = s - R_\Delta$.

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.

Let $\mu_0 = -\ell_a + \mu_b + R_\Delta + \Delta$. The emptier can neglect the anchor set no more than $|A|\mu_0$ times, because it is Δ-greedy-like and neglecting the anchor set $|A|$ times would increase the mass of the anchor set by μ_0, and consequently make each cup in A have fill high enough that the emptier, being Δ-greedy-like would be forced to empty from that cup; in particular, a cup in A starts with fill at least ℓ_a, so after having fill increased by μ_0 such a cup with have fill at least $R_\Delta + \Delta$ than μ_b (the maximum average fill of B). Hence such a cup must be emptied from.

We choose

$$c_\Delta = 2\frac{1}{q_0}\mu_0.$$

By having $|A| \cdot c_\Delta$ round-blocks, we make it so that there should be at least $|A|\mu_0$ 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 $2|A|\mu_0$, and the probability that it deviates from its expectation by more than $|A|\mu_0$ is hence exponentially small in $|A|$ and hence n as $|A| = \Theta(n)$. As shown before, the emptier cannot neglect the anchor set more than $|A|\mu_0$ 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,

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

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, $S \subset B$ chosen arbitrarily. 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 $\ell_b \geq -h/2$ by (12). 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 $H \leq O(1)$ 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

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

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

□

Lemma 2 (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

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

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 3.

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 sounds 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

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

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

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

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:

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

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

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

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

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

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

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 $2^{-n\delta^\ell(1-\delta)q}$ for an application of f to $n\delta^\ell(1-\delta)$ 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 3. *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. It is quite remarkable that the filler is 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 $2^{O(\log^2 n)}$. Because of the similarity to Theorem 1's proof, we ignore minor technical details in this section, specifically with regards to there being an integer number of cups. These can be dealt just as in the proof of Theorem 1: by modifying n and δ slightly to be powers of 2.

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 1 we achieve large backlog by repeated amplification of a base case.

The base case is given by Proposition 4. 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 4. Then we construct f_{i+1} as the amplification of f_i using Lemma 2.

Define g_i as

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

Claim 10. *$f_i(k\cdot n_b) \ge ck^{1-\epsilon}$ 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

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

Now we must prove the following claim (something similar was proven in the adaptive version of this proof, but the following proof is 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

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

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. □

Now we can complete the proof of Claim 10,

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

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

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

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

Hence the desired backlog is achieved. See the proof of Theorem 1 for how to treat the integers more carefully.

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. □

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.