Variable-Processor Cup Games

Alek Westover

April 20, 2020

Abstract

The **cup game** is a classic problem in computer science that models work-scheduling. In the cup game there are two players: the **emptier** and the **filler**. The cup game consists of many rounds. In a given round, the filler distributes p of water among the n cups (with at most 1 unit to any particular cup) and then the emptier removes 1 unit of water from p of the cups. We investigate a variant of the classic multi-processor cup game, the **variable-processor cup game**, in which the resources are variable: the filler is allowed to change p.

Although the modification to allow variable resources seems small, we will show that it drastically alters the outcome of the game. We demonstrate a recursive algorithm that an adaptive filler can use to achieve backlog $\Omega(n)$. We also demonstrate that this lower bound on backlog is tight: using a novel set of invariants we prove that a greedy emptier never lets backlog exceed $O(n)$.

We also investigate bounds on an oblivious filler. We show that an oblivious filler can achieve backlog $2^{\Omega(\sqrt{\log n})}$ in running time $O(\text{poly}(n))$ with constant probability against any "greedy-like" emptier. This result surprisingly uses very similar techniques as in the analysis of the adaptive filler, in addition to Hoeffding's Inequality for proving that the amount of water that we get in certain sets is tightly concentrated around its mean.

1 Introduction

The **cup game** is a classic problem in theoretical computer science that models work-scheduling. In the cup game, there are two players: the **filler** and the **emptier**. The cup game consists of many rounds. On each round the filler first distributes p units of water among the n cups with at most 1 unit to any particular cup (without this restriction the filler can trivially achieve unbounded backlog by placing all of its fill in a single cup every round), and then the emptier removes 1 unit of water from each of p cups.[1]

We investigate a variant of the vanilla multi-processor cup game which we call the **variable-processor cup game** in which the filler is allowed to change the p –the 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; like in the classic cup game we make the resources of the filler and emptier identical. Although this restriction may seem artificial, it is crucial; for example, if the filler had more resources than the emptier in the classic cup game, then the filler could trivially achieve unbounded backlog, as average fill would increase by at least some positive constant at each round. Thus, analysis of this game does provide useful information about how real-world systems behave.

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]. We hoped that the variable-processor cup game could be simulated in the vanilla multi-processor cup game, because the extra ability given to the filler does not seem very strong.

We invented the new version of the cup game arose as we tried to get a bound of $\Omega(\log p)$ backlog in the multi-processor game against an oblivious filler, which would combine with previous results to give us a lower bound that matches our upper bound: $O(\log \log n + \log p)$ [2]. In Proposition 2 we prove that there is an oblivious filling strategy in the variable-processor cup game on n cups that achieve backlog $\Omega(\log n)$ as desired. [3]

---

[1] Water represents work, at every time step some amount of new work comes in, and then we must allocate resources, in particular p processors, to certain processes, with the goal of minimizing backlog.

[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] Note that we have $\Omega(\log n)$ in this proposition instead of

However, we also show that attempts at simulating the variable-processor cup game are futile because the variable-processor cup game is–surprisingly–vastly different from the multi-processor cup game. This follows as a corollary of an **Amplification Lemma** for both the adaptive and oblivious filler.

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

In Section 3 we provide an inductive proof of a lower bound on backlog in Corollary 1. The base case of the argument is a direct consequence of Proposition 1, and the inductive step follows from the "Adaptive Amplification Lemma" (Lemma 1). Corollary 1 gives the lower bound $\Omega(n)$ on backlog.

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

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

2 Preliminaries

The cup game consists of a sequence of rounds. On the t-th round the state starts as S_t. The filler chooses the number of processors p_t for the round. Then the filler distributes p_t units of water among the cups (with at most 1 unit of water to any particular cup). After this, the game is in an intermediate state, which we call state I_t. Then the emptier chooses p_t cups to empty 1 unit of water from. This concludes the round; the state of the game is now S_{t+1}.

Denote the fill of a cup c by $\mathrm{fill}(c)$. Let the **positive tilt** of a cup c be $\mathrm{tilt}(c) = \max(0, \mathrm{fill}(c))$, and let the positive tilt of a set X of cups be $\sum_{c \in X} \mathrm{tilt}(c)$. Let

the **mass** of a set of cups X be $m(X) = \sum_{c \in X} \mathrm{fill}(c)$. 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.

In the proof of the upper bound it is important to note that we adopt the convention of allowing for negative fill.[4]

3 Adaptive Filler Lower Bound

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

Proof. Let $h = \frac{1}{4} \ln(n/2)$ be the desired fill. Once a cup with fill at least h is achieved the filler stops, the process completed. Let A consist of the $n/2$ fullest cups, and B consist of the rest of the cups (at any given state, so A, B are implicitly functions of the round t).

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

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

---

$\Omega(\log p)$ because the filler can increase the number of processors, so it increases the number of processors to $n-1$ to start. A nearly identical construction could be used to show that backlog $\Omega(\log p_{\max})$ can be achieved, where the number of processors starts at $p_{\max}$ and the filler does not ever increase the number of processors. However, using $p_{\max} = n$ is natural in the variable-processor cup game, so we do not consider the game with the restriction that the filler can not increase the number of processors above some $p_{\max} < n$.

---

[4]It is not important in the proofs of the lower bounds. Allowing for cups to have negative fill makes the game harder for the filler, as it means that none of the emptier's emptying is ever wasted by cups "zeroing-out". Our filling strategy does not rely on cups zeroing out to achieve large backlog however, so the lower bounds hold regardless of if fill is allowed to zero out or not. On the other hand the proof of the upper bound does depend on the fact that cups never zero out. In that proof we call the average fill of the cups 0 and say that it never changes. We chose to allow negative fill (which has no physical analog in the work scheduling analogy) because it makes the problem more elegant.

chooses an arbitrary cup to remove from the active set.

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

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

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

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

and $f'(n) \geq 1$, in the variable processor cup game on $n \geq n_0$ cups.

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

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

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

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

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

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

On the ℓ-th level of recursion the filler uses the following **process** to achieve the desired average fill in A: repeatedly apply f to B, and then take the cup generated by f within B to have large backlog and swap it with a cup in A; repeat until A has the desired average fill. Note that

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

so

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

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

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

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

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

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

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

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

We achieve average fill h_ℓ in the anchor set for L levels of recursion. Summing h_ℓ for $0 \leq \ell \leq L$ yields the desired result.

Note that as $n \geq n_0$ we can always simply use Proposition 1 to achieve backlog 1. We will revert to this option if it gives larger fill than we get by repeatedly applying f. $\qquad\square$

Corollary 1. *There is an adaptive filling strategy for the variable-processor cup game on n cups that achieves backlog $\Omega(n)$ in bounded running time.*

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

$k_0 \leq g/(1-\delta))$ we have the following:

$$f'(k_0) \geq$$

$$(1-\delta)\sum_{\ell=0}^{L} c(((1-\delta)\delta^\ell)k_0)^{1-\epsilon}$$

$$= ck_0^{1-\epsilon}(1-\delta)^{2-\epsilon}\sum_{\ell=0}^{L}(\delta^\ell)^{1-\epsilon},$$

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

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

Let

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

We will show that $h(\delta) \geq 1$ is satisfiable for sufficiently small δ given any choice of $\epsilon \in (0,1)$. Note that if $h(\delta) \geq 1$, then $f'(k_0) \geq ck_0^{1-\epsilon}$, meaning we have constructed from f a new function f' that satisfies the inequality $f'(k) \geq ck^{1-\epsilon}$ for $k \leq g/(1-\delta)$, as opposed to only for $k \leq g$ as in the case of f. [5] Consider the Taylor series for $(1-\delta)^{2-\epsilon}$ about $\delta = 0$:

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

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

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

Rearranging we have

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

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

---

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

overestimate we are still able to get the desired value of ϵ to work, as we will demonstrate later. We have,

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

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

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

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

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

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

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

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

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

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

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

Note that

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

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

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

other achieving backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ in running time $2^{O(\log^2 n)}$. Both constructive arguments rely on repeated application of the Amplification Lemma.

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

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

Note that $n_*(1-\delta)\delta \geq n_0$ by appropriate choice of constant c, and Proposition 1, which states that we can get backlog $O(\log n_*)$ on n_* cups[6]. Because f^* satisfies the Amplification Lemma we have:

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

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

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

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

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

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

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

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

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

We claim the following regarding this construction:

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

We shall prove Claim * by induction on i.

Claim * is true in the base case of f_0 by taking c sufficiently small, in particular small enough that $f_0(k) \geq ck^{1-\epsilon}$ holds for $k < g_0$[7]. As our inductive hypothesis we assume Claim * for f_i; we aim to show that Claim * holds for f_{i+1}. Note the key property of g_i, that $g_{i+1} \cdot (1-\delta) < g_i$. Also note that (without loss of generality) the f_i are monotonically increasing functions: given more cups we can always achieve higher fill than with fewer cups. Thus we have, for any $k < g_{i+1}$,

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

as desired.

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

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

We can upper bound this by

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

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

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

as desired. ok, technically I'm ignoring integer problems in saying lets only do $O(\lg n)$ levels of recursion,

---

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

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

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

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

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

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

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

The sum evaluates to

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

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

The running time satisfies the recurrence

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

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

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

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

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

□

4 Adaptive Filler Upper Bound

Let $\mu_S(X), m_S(X)$ denote the average fill and mass of a set of cups X respectively at state S (S could be S_t or I_t for any $t \in \mathbb{N}$) [8]. Let $S_t(\{r_1, \ldots, r_m\})$ and $I_t(\{r_1, \ldots, r_m\})$ denote the cups of ranks $r_1, r_2, \ldots, r_m$ at states S_t and I_t respectively. Let $[n] = \{1, 2, \ldots, n\}$, let $i + [n] = \{i + 1, i + 2, \ldots, i + n\}$. We will use concatenation of sets to denote unions, i.e. $AB = A \cup B$.

We establish the following Lemma:

Lemma 2. *The greedy emptier maintains the invariant*

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

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

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

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

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

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

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

Note that $a + b = \min(k, p_t)$. Note that $A = I_t([a])$ and $B = I_t(a + [b])$, as every cup in A must have higher fill than all cups in B in order to remain above the cups in B after 1 unit of water is removed from all cups in AB. Further, note that $S_{t+1}([k]) = AC$, because once the cups in B are emptied from cups in C take their place among the k fullest cups.

---

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

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

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

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

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

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

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

Now we proceed with the proof of the Lemma.

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

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

Now consider $b \neq 0$. In particular, note that this implies $a < k$ as $a + b \leq k$. Recall that $S_{t+1}([k]) = AC$, 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 by the emptier. The emptier removes exactly a units of water from AC, and the filler adds at most $a + b \leq p_t$ (recall $a + b = \min(p_t, k)$) units of water to AC. Thus:

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

This is easy to bound if $m_{S_t}(C) \leq m_{S_t}(BC) - b$, because

$$m_{S_t}(A) + m_{S_t}(BC) = m_{S_t}(ABC) \leq m_{S_t}([k])$$

which would imply the invariant for S_{t+1}, k. If $\mu_{S_t}(C)$ is not significantly less than $\mu_{S_t}(BC)$ we have

more difficulty. The key insight is to notice that larger values for $m_{S_t}(A)$ correspond to smaller values for $m_{S_t}(C)$ because of the invariants; the higher fill in A ***pushes down*** the fill that C can have. By quantifying exactly how much higher fill in A pushes down fill in C we can arrive at the desired invariant. We can upper bound $m_{S_t}(C)$ by $\frac{c}{b+c}m_{S_t}(BC) = (m_{S_t}(ABC) - m_{S_t}(A))\frac{c}{b+c}$ as $\mu_{S_t}(C) \leq \mu_{S_t}(B)$ (we established that this was without loss of generality because interchanging is possible). Thus, we have

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

where

$$\begin{aligned} &m_{S_t}(A) + \frac{c}{b+c}m_{S_t}(BC) \\ &= \frac{c}{b+c}m_{S_t}(ABC) + \frac{b}{b+c}m_{S_t}(A). \end{aligned} \tag{5}$$

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

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

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

Thus we have the following bound:

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

7

where (*) follows from Equation 5. Recall that we were considering $b > 0, c \geq 1$ (we previously handled the case where $b = 0$, and if $b > 0$ then $c = k - a \geq b \geq 0$). Hence we have

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

So

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

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

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

5   Oblivious Filler Lower Bound

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

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

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

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

Call an emptying strategy (T, Δ)-***greedy-like*** if is satisfies the following property when the number of processors is p: for any cup i, if $\text{fill}(i) + \Delta > T$, and there are at least p cups containing fill greater than $\text{fill}(i) + \Delta$, the emptier does not empty from cup i. Combinatorially the quantity T is the threshold above which the filler "notices" cups, and $\pm\Delta$ is the tolerance in cup fills within which the emptier is allowed to not be greedy. Of particular interest is the smoothed greedy emptier, which is $(1, 0)$-greedy-like.

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

Proof. Let A, the ***anchor*** set, be a subset of the cups chosen uniformly at random from all subsets of size $n/2$ of the cups, and let B, the ***non-anchor*** set, consist of the rest of the cups ($|B| = n/2$). Let $h = 2T + g$ where g is a sufficiently large constant. At each level of our recursive procedure we will achieve fill h in some fraction of the cups in A, and because the emptier is (T, Δ)-greedy-like, we can turn this into a known set of cups with fill at least $h' = T + (g + \Delta)/2$. Our strategy to achieve backlog $\Omega(\log n)$ overall is roughly as follows:

- **Step 1:** Obtain large positive tilt in B, either by repeatedly making cups in B have a constant probability of having fill at least h and then transferring these cups into A, or by exploiting high expected positive tilt.

- **Step 2:** Reduce the number of processors to a constant fraction nc of n and raise the fill of nc cups to h'. This step relies on the emptier being greedy.

- **Step 3:** Recurse on the nc cups that are known to have fill at least h'.

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

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

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

We consider two cases:

- **Case 1:** For at least $1/2$ of the switching-processes, at least $1/2$ of the cups $c \in B$ have

$\text{fill}(c) \geq -h$.

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

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

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

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

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

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

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

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

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

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

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

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

$\square$

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

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

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

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

We have

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

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

Now we have, by Hoeffding's inequality, that

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

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

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

$\square$

In both cases we achieve, with probability at least $1 - 1/\operatorname{polylog} n$, positive tilt at least hnq in the anchor set for some known $q \in (0,1)$. Using the positive tilt, with one processors, we can transfer over the fill into nk cups. Note, we use one processor because we do not know how many cups the fill is concentrated in. The filler repeatedly distributes 1 unit of fill to each of the nq cups in succession, and continues until h' fill has been distributed. We cannot continue beyond this point because we have used up the positive tilt. Now we recurse on this set of nq cups.

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

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

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

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

that succeeds with constant probability.

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

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

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

Consider 2 cases:

- If the mass of the verysad cups is less than $nh/8$ then we can ignore them and accept a $-h/8$ penalty to the average fill.

- On the other hand, if the mass of the verysad cups is greater than $nh/8$, then by the end the average fill of everything else is already $h/8$ which is also basically as desired.

$\square$

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

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

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

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

$\square$

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

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

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

$\square$

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

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

6   Conclusion

Many important open questions remain open.

References

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

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