Variable Processor Cup Games

Alek Westover

Belmont High School

June 1, 2020

What is the Cup Game?

Definition

The ***p-processor cup-game*** on n cups is a multi-round game in which two players take turns emptying and removing water from the cups.

On each round

- The ***filler*** distributes p units of water among the cups (with at most 1 unit to any particular cup).
- Then the ***emptier*** chooses p cups to remove (at most) one unit of water from.

What is the Cup Game?

Definition

The ***backlog*** of the system is the amount of water in the fullest cup; The emptier aims to minimize backlog whereas the filler aims to maximize backlog.

Note: The emptier's resources must be allocated discretely whereas the filler can continuously distribute resources.

Why is it important?

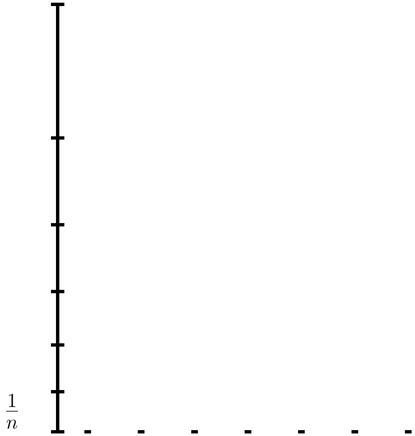
The cup game models ***work scheduling***:

- The n cups represent tasks that must be performed.
- At each time step:
  - p new units of work come in, distributed arbitrarily among the n tasks (with the constraint that no task gets more than 1 unit of work)
  - p processors must be allocated to a subset the tasks, on which they will achieve 1 unit of progress.

The cup game is also an interesting mathematical object.

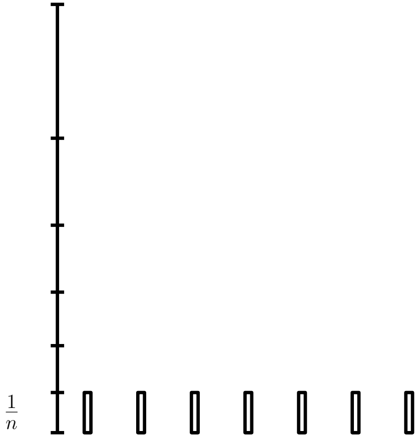
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



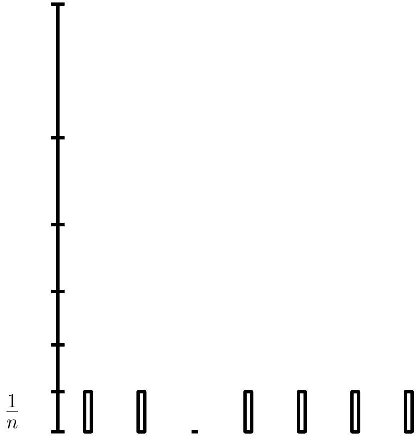
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



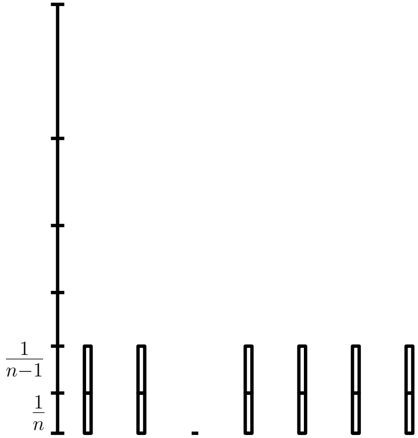
SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



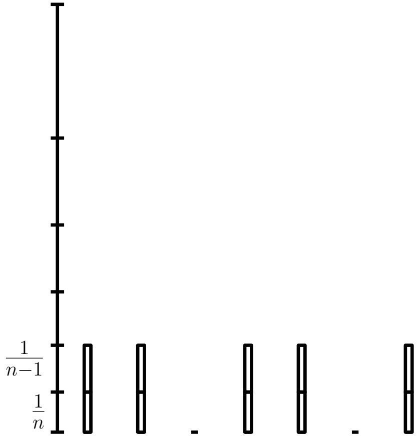
SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



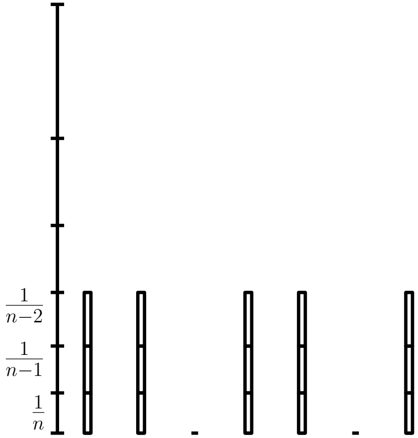
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



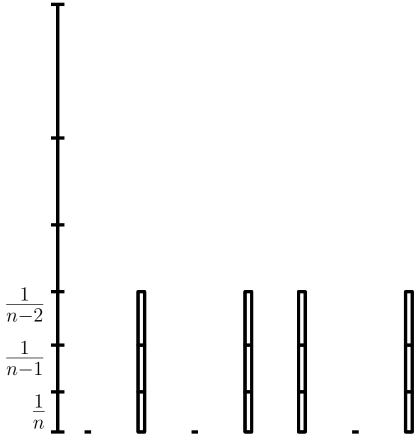
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



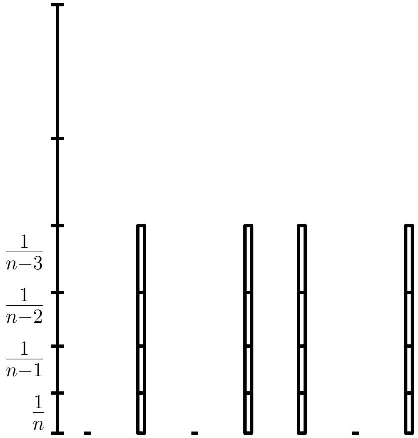
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



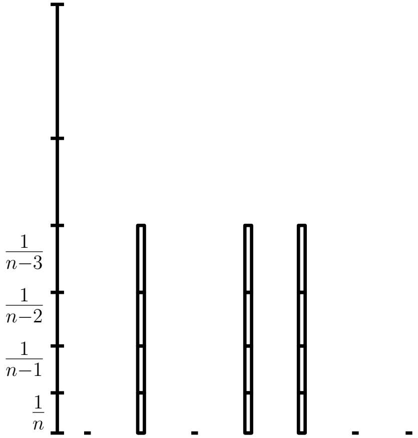
SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



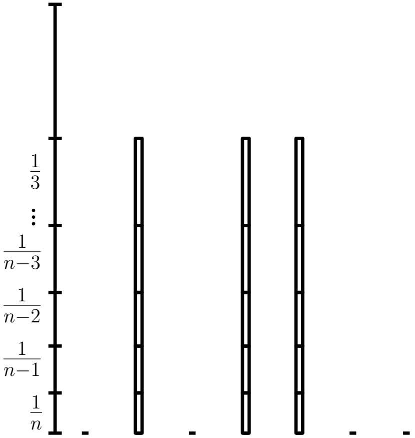
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



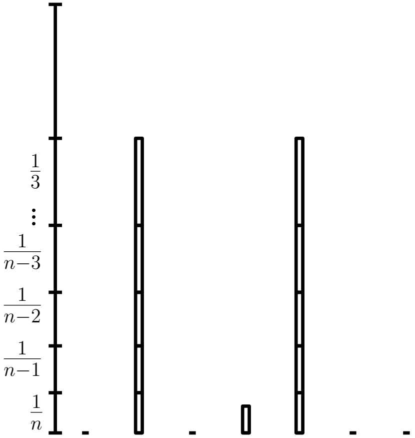
SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



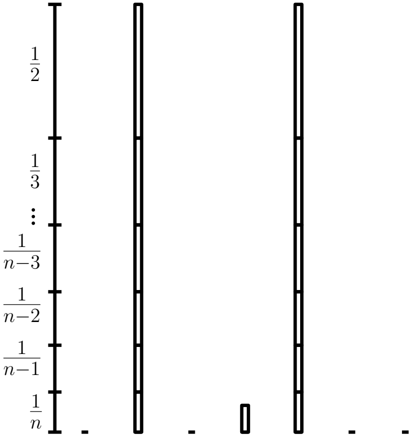
Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



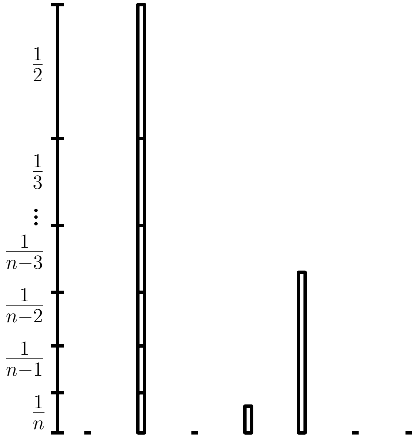
SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.



Single-Processor Lower Bound

Filling strategy: distribute water equally amongst cups not yet emptied by the emptier.

Achieves backlog:

$$\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} = \Omega(\log n).$$

Single-Processor Upper Bound

A ***greedy emptier*** – an emptier that always empties from the fullest cup – never lets backlog exceed $O(\log n)$.

Definitions

- Let S_t denote the cup state at the start of round t
- Let I_t denote the state after the filler has added water on round t but before the emptier has emptied from cups
- Let $\mu_k(S_t)$ denote the average fill of the k fullest cups in S_t.

Single-Processor Upper Bound Proof

Proof: Inductively prove a set of invariants:

$$\mu_k(S_t) \leq \frac{1}{k+1} + \dots \frac{1}{n}.$$

Let a be the cup that the emptier empties from in state S_t.

Case 1: a is the fullest cup in S_{t+1}

$$\mu_k(S_{t+1}) \leq \mu_k(S_t).$$

Case 2: a is not the fullest cup in S_{t+1}

$$\mu_k(S_{t+1}) \leq \mu_{k+1}(I_t) \leq \mu_{k+1}(S_{t+1}) + \frac{1}{k+1}.$$

Previous work on Cup Games

- The Single-Processor cup game ($p = 1$) has been tightly analyed with ***oblivious*** and ***adaptive*** fillers (i.e. fillers that can't and can observe the emtpier's actions).
- The Multi-Processor cup game ($p > 1$) is substantially more difficult. With an adaptive filler:
  - Kuszmaul established upper bound of $O(\log n)$.[1]
  - We established a matching lower bound of $\Omega(\log n)$.
- The multi-processor cup game with an oblivious filler has not yet been tightly analyzed.
- Variants where valid moves depend on a graph have been studied.
- Variants with resource augmentation have been studied.
- Variants with clairvoyance have been studied.

---

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

Our Variant of the Cup Game

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 prove very surprising bounds on backlog.

Negative Fill

In lower bound proofs we allow negative fill

- This allows for measuring fill relative to average fill
- This is important for the recursive nature of our strategies
- The game is strictly easier for the filler if cups can zero out

Amplification Lemma

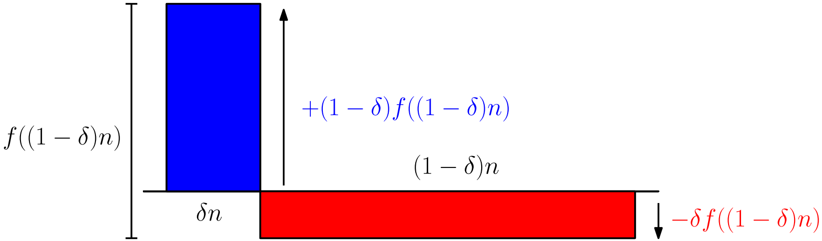
Lemma

Given a strategy f, we can construct a new strategy that achieves backlog

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

for appropriate parameters $L \in \mathbb{N}, 0 < \delta \ll 1/2$.

Amplification Lemma Intuition



Proof Sketch of Amplification Lemma

- A starts as the δn fullest cups, B as the $(1-\delta)n$ other cups.
- Repeatedly apply f to B and swap the cup with fill increased by $f(|B|)$ into A.
- Decrease p to $|A|$ and recurse on A.

Lower Bound against Adaptive Filler

By repeated amplification, using $\delta = \Theta(1)$, we get:

Theorem

There is an adaptive filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ of our choice in running-time $2^{O(\log^2 n)}$.

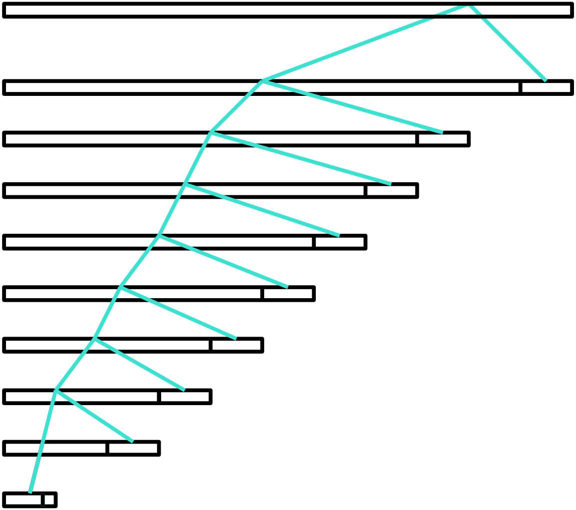
Lower Bound against Adaptive Filler

Setting $\delta = \Theta(1/n)$ – which is quite extremal – and recursively using the Amplification Lemma we prove:

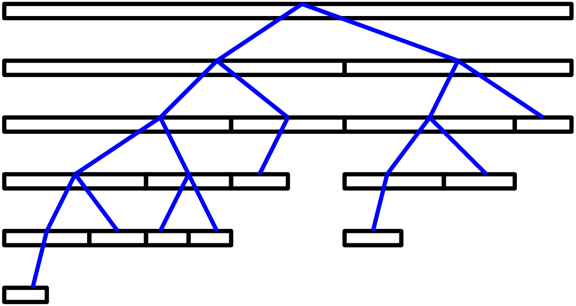
Theorem

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

Lower Bound Intuitively



Lower Bound Intuitively



Upper Bound against Adaptive Filler

We prove a novel set of invariants:

Theorem

A greedy emptier maintains the invariant:

$$\mu_k(S_t) \leq 2n - k.$$

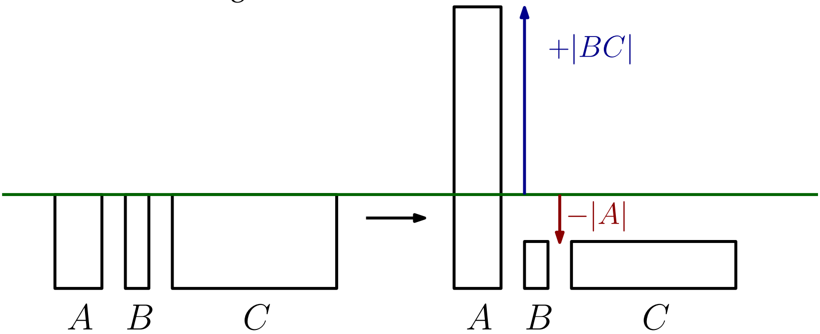
In particular this implies that backlog is

$$O(n).$$

Note: this matches our lower bound!

Upper Bound Proof Idea

Extremal fill configuration:



Lower Bound Against Oblivious Filler

Classically the emptier can do much better (e.g. $\log\log n$ vs $\log n$ backlog) in the randomized setting.

In the variable-processor cup game–shockingly–an oblivious filler can achieve an identical lower bound for games of length $2^{\operatorname{polylog}(n)}$ with probability at least $1 - 2^{-\operatorname{polylog}(n)}$, although only against ***greedy-like*** emptiers.

Oblivious Filler: Constant Fill

Getting constant fill in a *known* cup is hard now. Strategy:

- Play lots of single-processor cup games on constant numbers of cups blindly. Each succeeds with some constant probability.
- By a Chernoff Bound, with *exponentially* good probability, i.e. $1 - 2^{-\Omega(n)}$, at least a constant fraction, say nc, of the single-processor cup games succeed.
- Set $p = nc$.
- Exploiting the greedy-like nature of the emptier fill a set of nc known cups while the emptier is forced to focus on the set of nc with high fill.
- Recurse for a constant number of levels on the nc cups with known high fill.

Oblivious Amplification Lemma

Almost identical to the Adaptive Amplification Lemma!

Lemma

Given a strategy f, we can construct a new strategy that achieves backlog

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

for appropriate parameters $L \in \mathbb{N}, 0 < \delta \ll 1/2$ and constant $\phi \in (0,1)$ of our choice.

Note that the Lemma is actually substantially more complicated than this: There are conditions on the starting configurations, everything succeeds with certain probabilities, the emptier must be greedy-like, we aren't discussing run-time here, and more.

Lower Bound Against Oblivious Filler

By a similar argument as for the adaptive case we have:

Theorem

There is an oblivious filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$ for constant $\epsilon > 0$ with probability at least $1 - 2^{-\operatorname{polylog}(n)}$ in running time $2^{O(\log^2 n)}$.

Note that for the union bound to guarantee the appropriate probability the base case in our recursive argument has to be larger in the oblivious case than the adaptive case, but this doesn't harm our backlog result.

Open Questions

- Can we extend the Oblivious lower-bound construction to work against arbitrary emptiers?
- Are there shorter more simple constructions?

Acknowledgements

- My mentor, William Kuszmaul!
- MIT PRIMES
- My Parents