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?

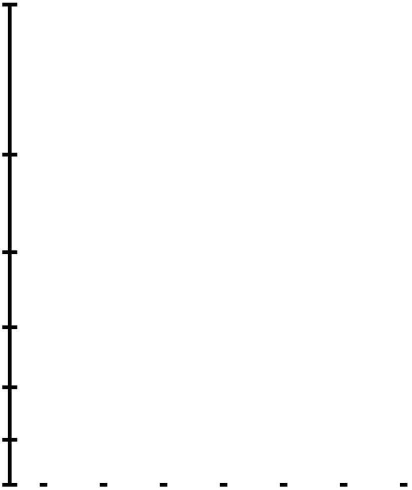
The cup game models ***work scheduling***:

- 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.

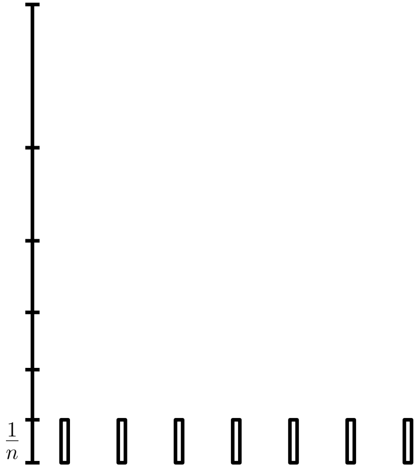
SINGLE-PROCESSOR LOWER BOUND

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



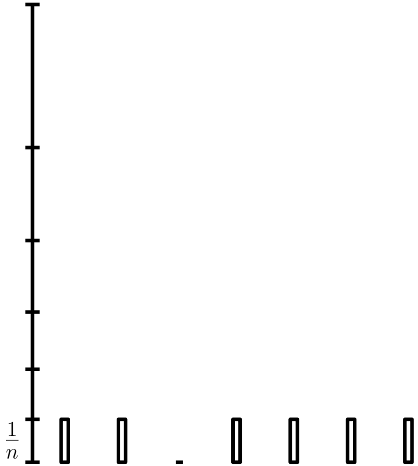
SINGLE-PROCESSOR LOWER BOUND

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



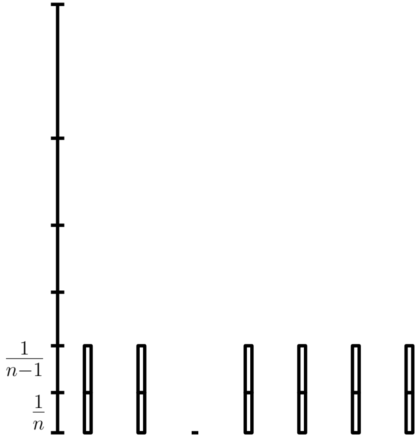
Single-Processor Lower Bound

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



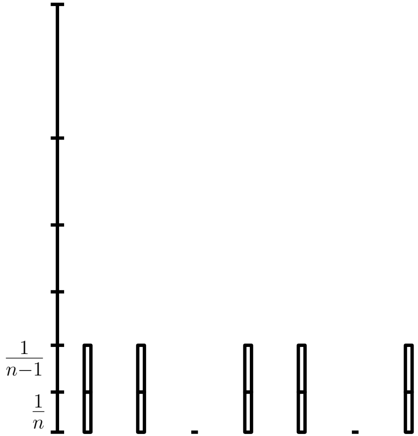
SINGLE-PROCESSOR LOWER BOUND

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



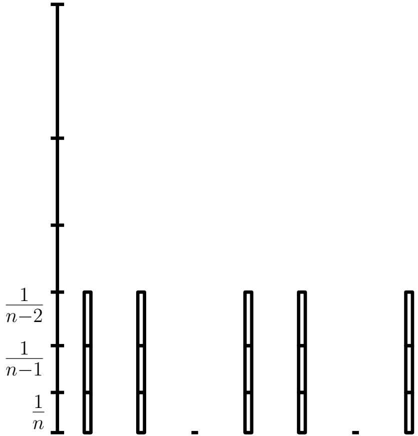
SINGLE-PROCESSOR LOWER BOUND

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



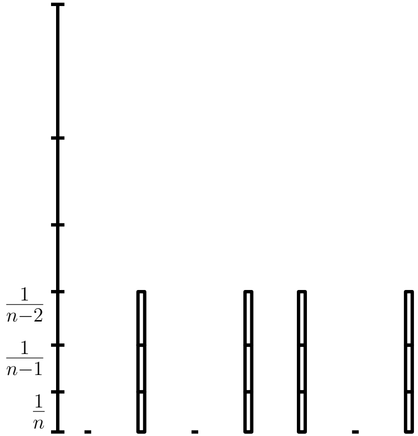
Single-Processor Lower Bound

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



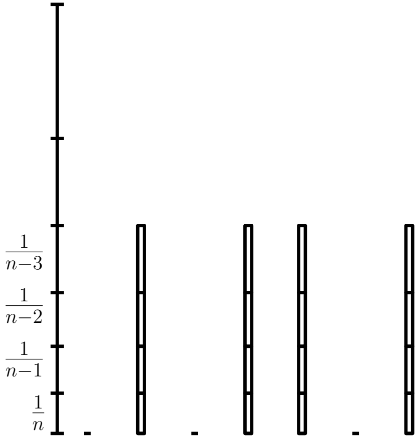
SINGLE-PROCESSOR LOWER BOUND

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



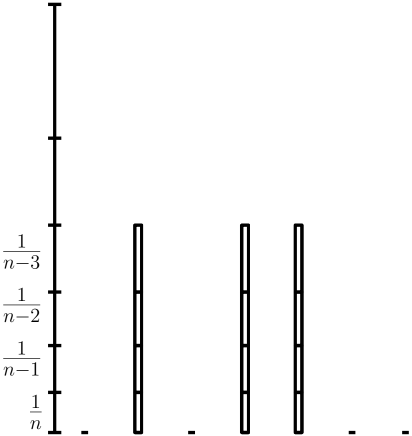
SINGLE-PROCESSOR LOWER BOUND

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



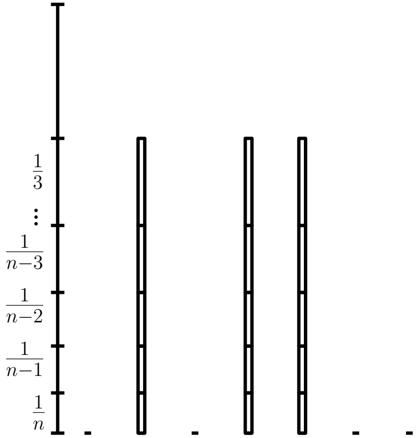
Single-Processor Lower Bound

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



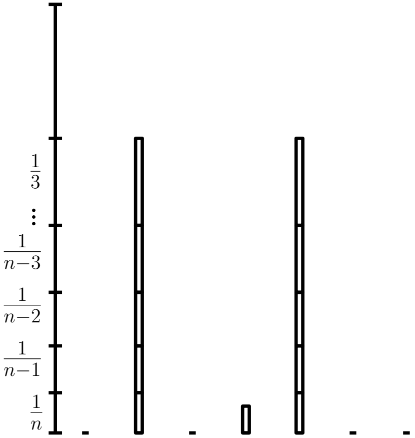
Single-Processor Lower Bound

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



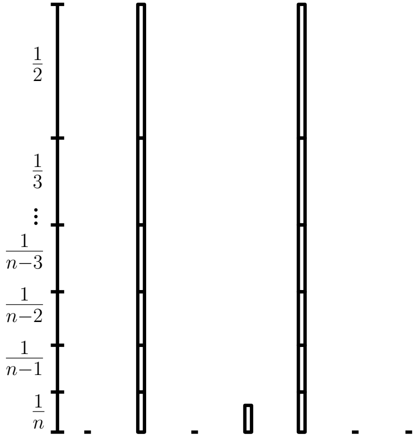
Single-Processor Lower Bound

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



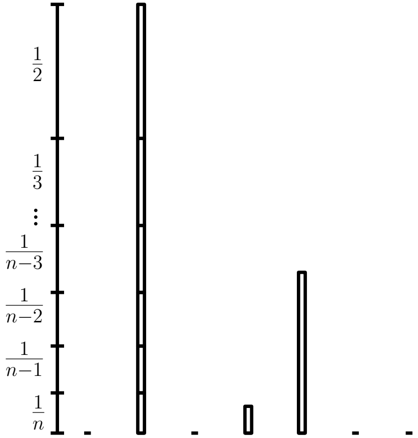
SINGLE-PROCESSOR LOWER BOUND

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:

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

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}

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

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

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

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.

Amplification Lemma

Lemma

Given a filling strategy for achieving backlog $f(n)$ on n cups, we can construct a new filling strategy that achieves backlog

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

for any parameter δ with $0 < \delta \ll 1/2$ of our choice, where L is the largest integer such that we can achieve backlog 1 on $(1-\delta)\delta^L n$ cups.

Remark: WLOG the number of cups in the recursive calls is an integer.

Proof Sketch of Amplification Lemma

- Let A be the δn fullest cups and B be the $(1-\delta)n$ other cups.
- By repeatedly applying f to each cup in B, and transfering over the cup generated in B with backlog $f((1-\delta)n)$ to A, while maintaining the fill of cups in A, we make $\mu(A) - \mu(B) \geq f((1-\delta)n)$. This is accomplished while maintaining $\mu(A \cup B)$. The mass of A is guaranteed to be obve $a\mu(A \cup B)$ by the same amount that the mass of B is guaranteed to be depressed from $b\mu(A \cup B)$, thus fraction of the difference that $\mu(A)$ gets is $|B|/|A \cup B| = (1-\delta)$. So $\mu(A)$ is at least $(1-\delta)f((1-\delta)n)$ above $\mu(A \cup B)$.
- We then recursively apply this procedure to A. Summing over $\ell = 0, 1, \ldots, L$ we have the desired result.

Note: we are ignoring a lot of details here, e.g. ensuring that we actually are playing a cup game on B when applying f to it.

Lower Bound against Adaptive Filler

Setting $\delta = O(1/n)$ and analyzing a filling strategy that recursively applies the Amplification Lemma we proved

Corollary

The filler can achieve backlog

$$\Omega(n)$$

Upper Bound against Adaptive Filler

We prove a novel set of invariants:

Theorem

A greedy emptier maintains the invariant:

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

In particular this implies that backlog is

$$O(n).$$

Note: this matches our lowerbound!

Strategy Even Works with Oblivious filler!

Using Hoeffding's Concentration Inequality[2], we can surprisingly prove the same lower-bound for an Oblivious filler as for an Adaptive filler, although only against ***greedy-like*** emptiers.

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

Open Questions

- Can we extend the Oblivious lower-bound construction to work against a broader class of emptiers?
- Can we extend the Oblivious lower-bound construction to work against arbitrary emptiers?

Acknowledgements

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