

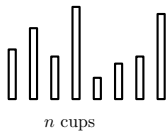
The Variable Processor Cup Game

Alek Westover

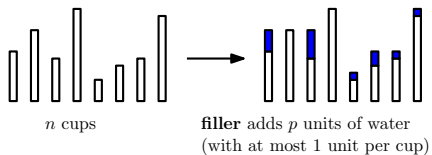
Belmont High School

June 7, 2020

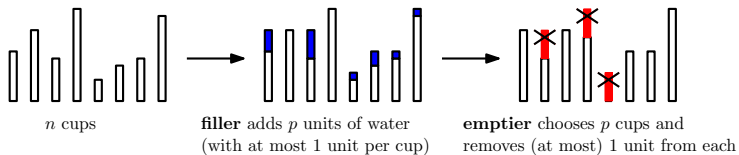
p -PROCESSOR CUP GAME ON n CUPS



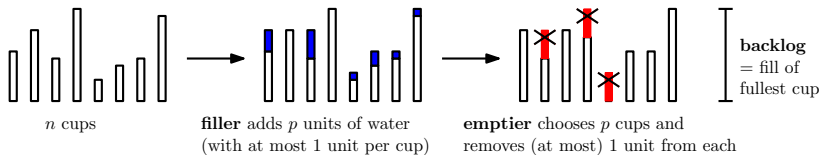
p -PROCESSOR CUP GAME ON n CUPS



p -PROCESSOR CUP GAME ON n CUPS

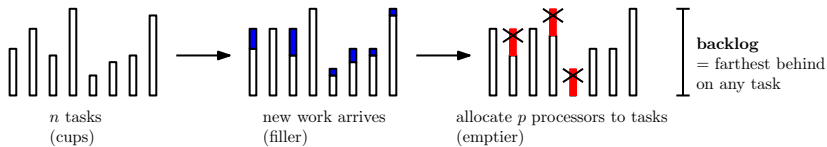


p -PROCESSOR CUP GAME ON n CUPS



WHY IS THE CUP GAME IMPORTANT?

work scheduling:



PREVIOUS WORK — $p = 1$

Single-processor cup game

Adaptive filler:

- ▶ $\Omega(\log n)$ lower bound
- ▶ $O(\log n)$ upper bound

Oblivious filler (can't see emptier's actions): ¹

- ▶ $\Omega(\log \log n)$ lower bound
- ▶ $O(\log \log n)$ upper bound (with good probability in short games)

¹[M. Bender, M. Farach-Colton, and W. Kuszmaul. Achieving optimal backlog in multi-processor cup games. In Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC), 2019.]

PREVIOUS WORK — RESTRICTED VERSIONS

Cup flushing game (emptier can completely empty cups):²

- ▶ $\Omega(\log \log n)$ lower bound
- ▶ $O(\log \log n)$ upper bound

Bamboo Garden Trimming (filler always adds same amount):³

- ▶ 2 lower bound
- ▶ 2 upper bound

²[P. F. Dietz and R. Raman. Persistence, amortization and randomization. In Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 78–88, 1991.]

³[Bilò, Davide, Luciano Gualà, Stefano Leucci, Guido Proietti, and Giacomo Scornavacca. "Cutting Bamboo Down to Size." arXiv preprint arXiv:2005.00168 (2020).]

PREVIOUS WORK — $p > 1$

Multi-processor cup game: ⁴

- ▶ $\Omega(\log n)$ lower bound
- ▶ $O(\log n)$ upper bound

Multi-processor cup game with *oblivious* filler: ⁵

- ▶ $\Omega(\log \log n)$ lower bound
- ▶ $O(\log \log n + \log p)$ upper bound (with good probability in short games)

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

⁵[William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. SIAM, 2020.]

THIS TALK

Our Question: What if p can change?

Variable-Processor Cup Game:

Each round the filler can change p

Modification seems small...

OUR RESULT

The variable-processor cup game is *fundamentally different* than the p -processor cup game!

ADAPTIVE FILLER LOWER BOUND

Theorem

There is an adaptive filling strategy that achieves backlog

$$\Omega(n^{1-\epsilon})$$

for any constant $\epsilon > 0$ in running-time

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

ADAPTIVE FILLER LOWER BOUND

Theorem

There is an adaptive filling strategy that achieves backlog

$$\Omega(n)$$

in running-time

$$O(n!).$$

UPPER BOUND

Theorem

A greedy emptier maintains the invariant:

$$\text{Average fill of } k \text{ fullest cups} \leq 2n - k.$$

Corollary

A greedy emptier never lets backlog exceed

$$O(n).$$

This matches our lower bound!

OBLIVIOUS FILLER LOWER BOUND

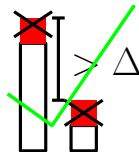
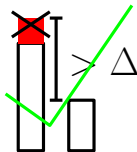
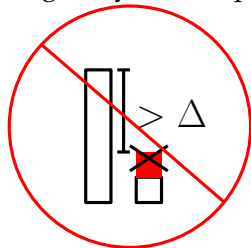
Theorem

There is an oblivious filling strategy that achieves backlog

$$\Omega(n^{1-\epsilon})$$

for constant $\epsilon > 0$ with probability at least $1 - 2^{-\text{polylog}(n)}$ in running time $2^{O(\log^2 n)}$ against a greedy-like emptier.

Δ -greedy-like emptier:



Adaptive Filler Lower Bound Proof Sketch

AMPLIFICATION LEMMA

Lemma

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

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

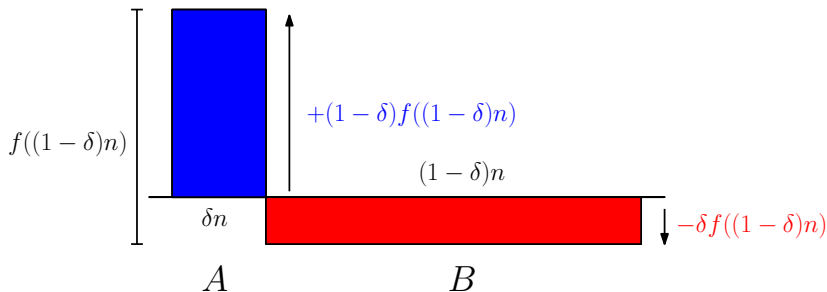
for appropriate parameters $L \in \mathbb{N}$, $0 < \delta \ll 1/2$.

And, if the running time of $f(n)$ is $T(n)$ the running time of $f'(n)$ satisfies

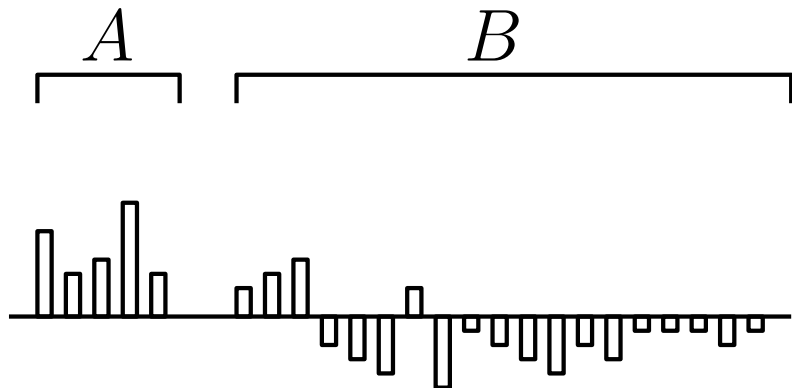
$$T'(n) \leq (L + 1)n^2T(n(1 - \delta)).$$

PROOF META-STRUCTURE

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

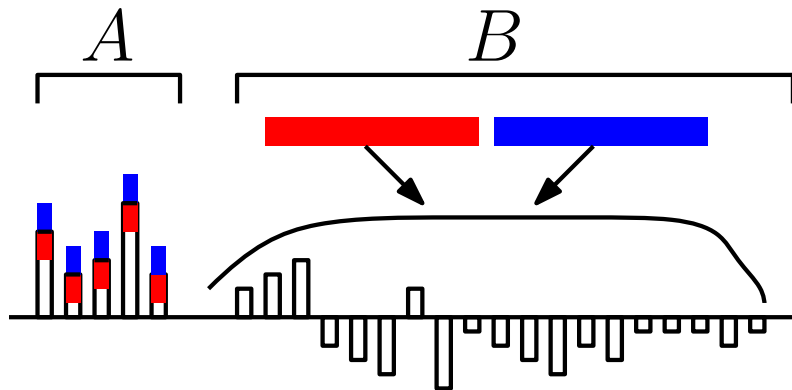


AMPLIFICATION LEMMA PROOF SKETCH



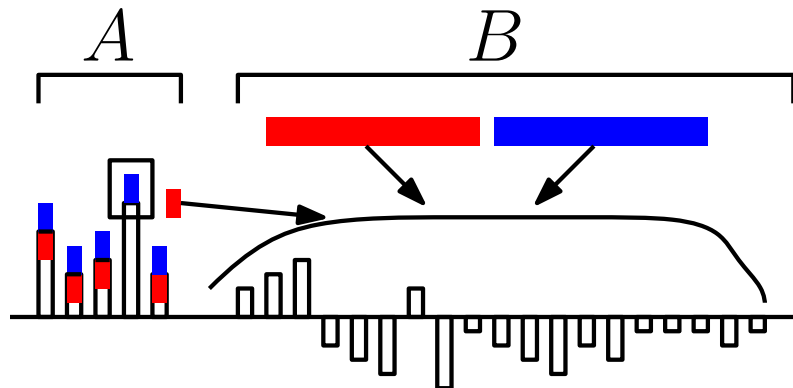
Instantiate A and B

AMPLIFICATION LEMMA PROOF SKETCH



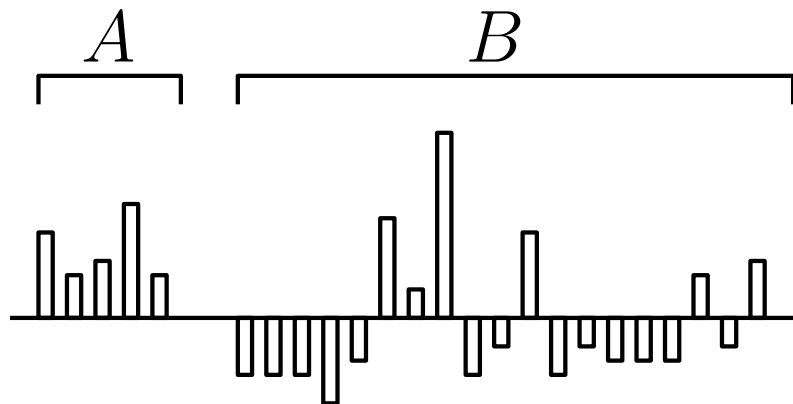
Will try to apply f to B while placing 1 fill in each cup in A .

AMPLIFICATION LEMMA PROOF SKETCH



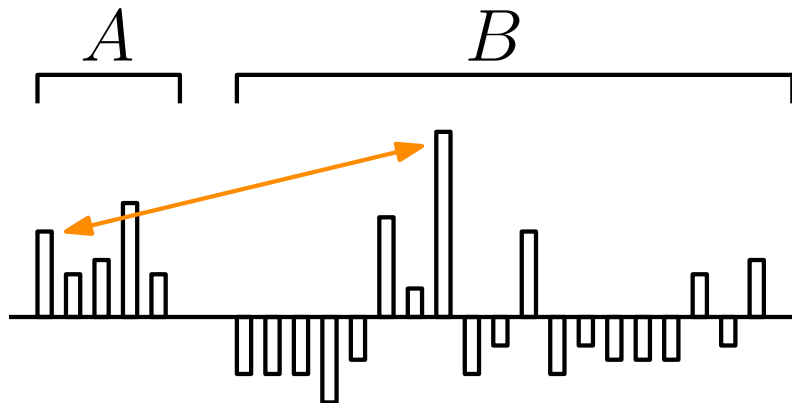
If the emptier doesn't empty from each cup in A (we say it *neglects* A) then the average fill of A rises!

AMPLIFICATION LEMMA PROOF SKETCH



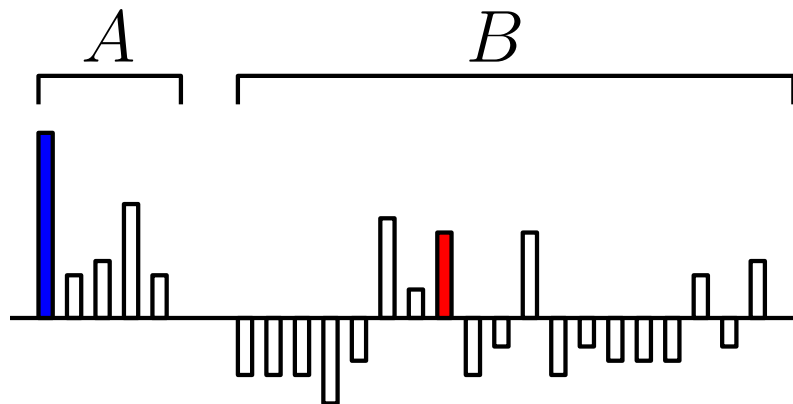
If emptier doesn't neglect A filler can apply f to B

AMPLIFICATION LEMMA PROOF SKETCH



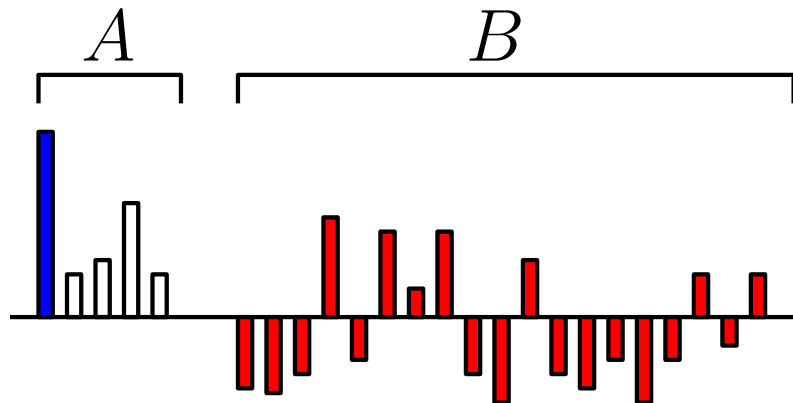
this gets a cup with high fill in B : swap it into A

AMPLIFICATION LEMMA PROOF SKETCH



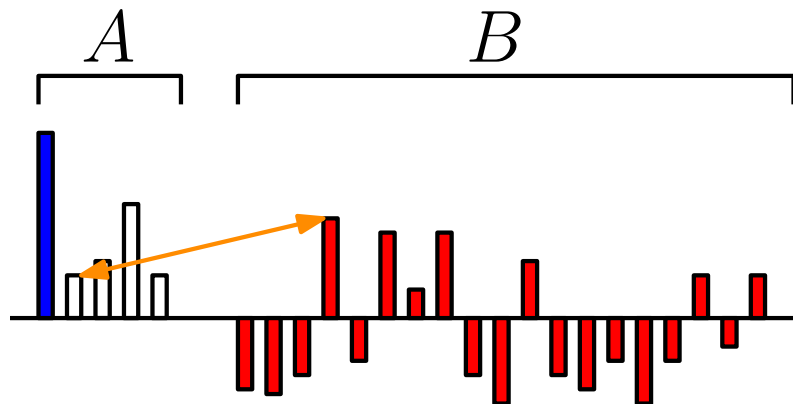
Note: swaps increase average fill of A , decrease average fill of B .

AMPLIFICATION LEMMA PROOF SKETCH



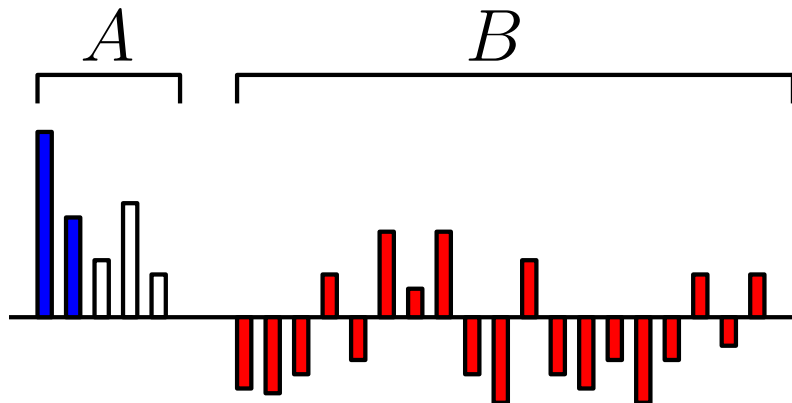
Apply f to B again

AMPLIFICATION LEMMA PROOF SKETCH



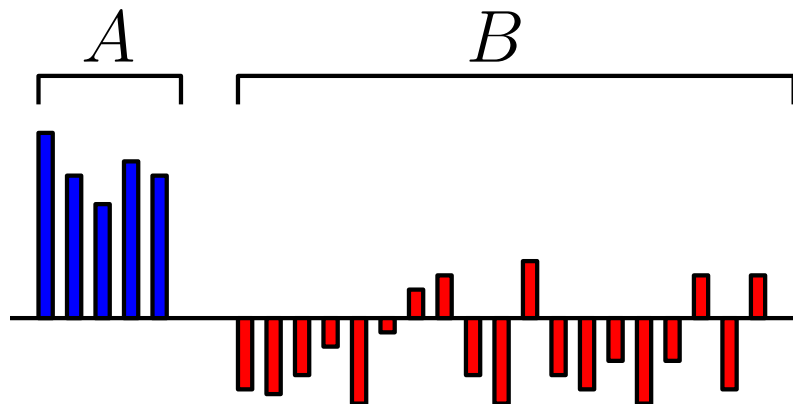
If emptier doesn't neglect A we have a high cup in B again

AMPLIFICATION LEMMA PROOF SKETCH



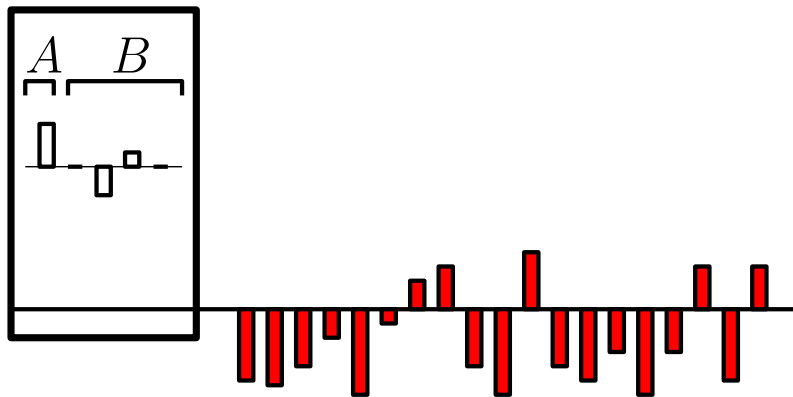
Swap this cup into A .

AMPLIFICATION LEMMA PROOF SKETCH



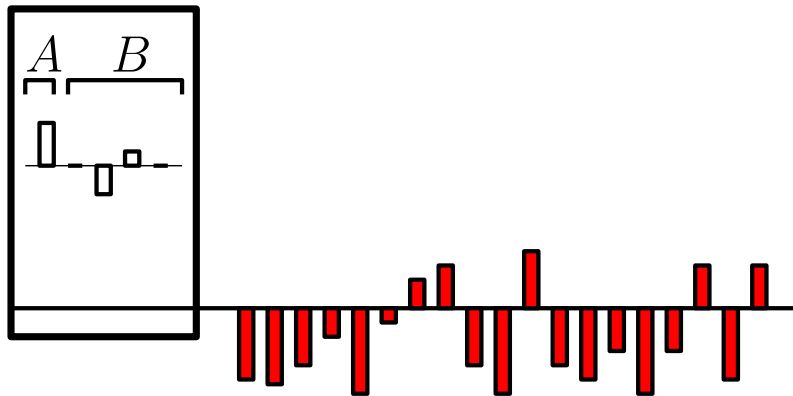
After repeating this process for each cup in A , the difference of the average fills of A and B is at least $f(|B|)$; A accounts for $|B|/n$ of this difference. Thus average fill of A has increased by at least $(1 - \delta)f((1 - \delta)n)$

AMPLIFICATION LEMMA PROOF SKETCH



Recurse on A for L levels of recursion. Problem size shrinks by a factor of δ each time.

AMPLIFICATION LEMMA PROOF SKETCH



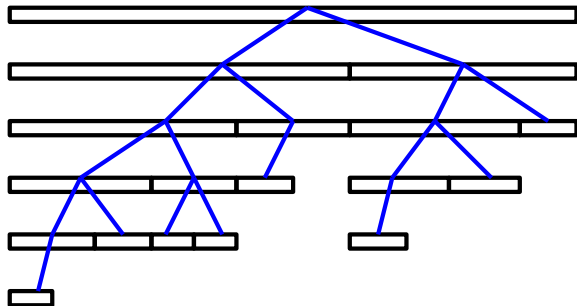
$$f'(n) \geq (1 - \delta) \sum_{\ell=0}^L f((1 - \delta)\delta^\ell n)$$

ADAPTIVE FILLER LOWER BOUND

Let $\epsilon > 0$ be any constant. Then there is some $\delta = \Theta(1)$ such that by repeated amplification we get:

Theorem

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



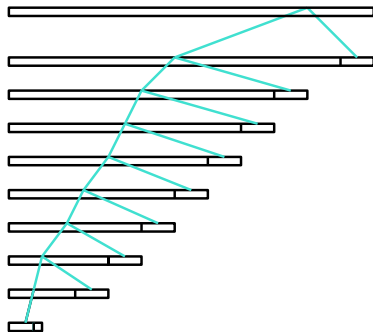
ADAPTIVE FILLER LOWER BOUND

Extremal strategy:

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

Theorem

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



OPEN QUESTIONS

- ▶ Can we extend the oblivious lower bound construction to work with arbitrary emptiers?
- ▶ Are there shorter more simple constructions?

ACKNOWLEDGEMENTS

- ▶ My mentor William Kuszmaul
- ▶ MIT PRIMES
- ▶ My Parents

Question Slides

WHAT IS THE CUP GAME?

Definition

p-processor cup-game on n cups:
multi-round game. every round:

- ▶ *filler* adds water
- ▶ *emptier* removes water

Note:

Emptier must allocate resources discretely

Filler can allocate resources continuously

GOALS

- ▶ Emptier tries to *minimize* backlog
- ▶ Filler tries to *maximize* backlog

WHY IS THE CUP GAME IMPORTANT?

Models *work scheduling*:

- ▶ Cups represent tasks
- ▶ At each time step:
 - ▶ new work comes in distributed among the tasks (*filler*)
 - ▶ must allocate processors to work on tasks (*emptier*)

ANALYSIS OF THE CUP GAME

Prove *lower bounds*

- ▶ Exhibit a filling strategy that achieves large backlog (against any emptier)

Prove *upper bounds*

- ▶ Exhibit an emptying strategy that prevents backlog from growing large (against any filler)

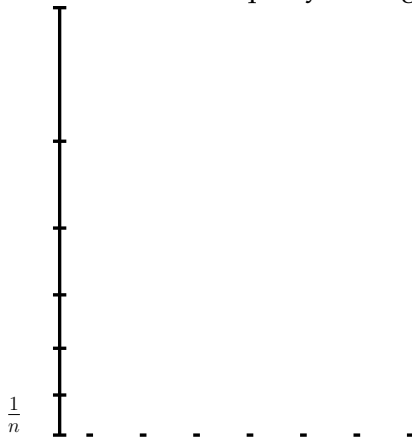
Quick example: Single-processor cup game:

- ▶ $\Omega(\log n)$ lower bound
- ▶ $O(\log n)$ upper bound

SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

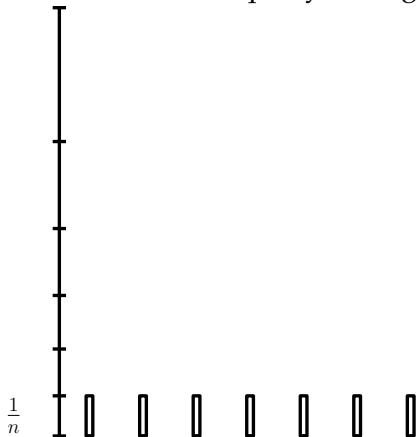
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

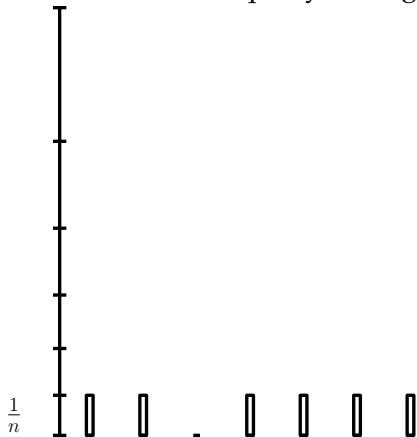
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

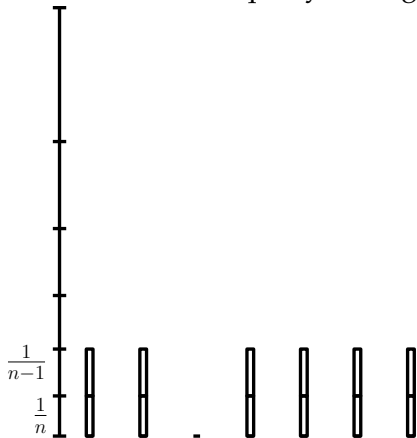
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

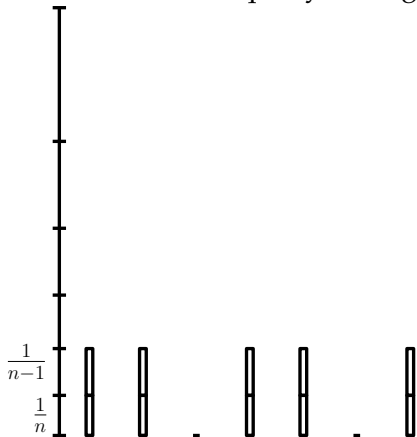
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

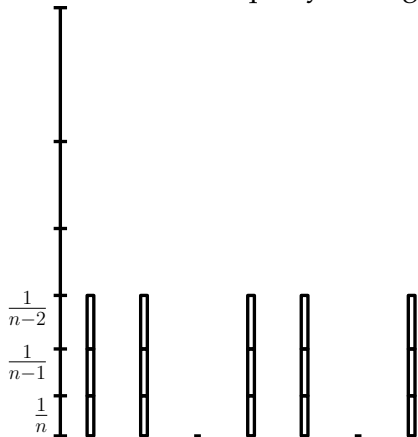
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

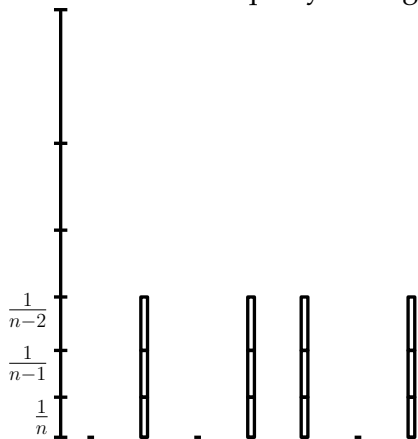
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

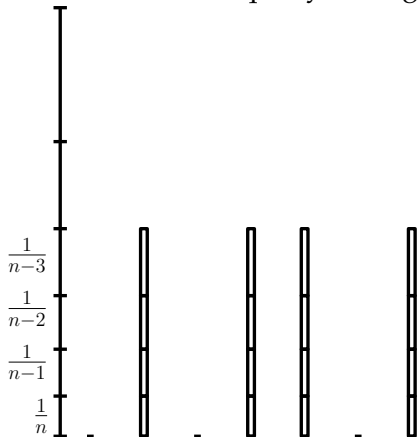
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

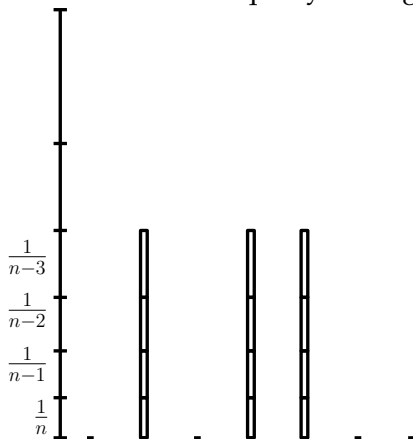
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

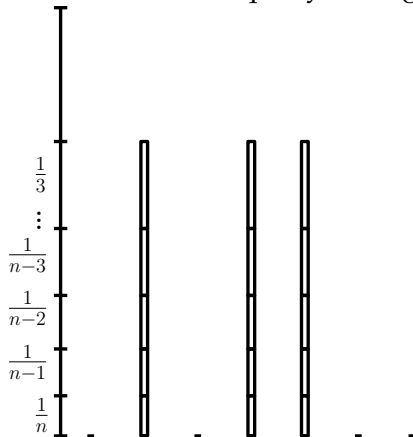
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

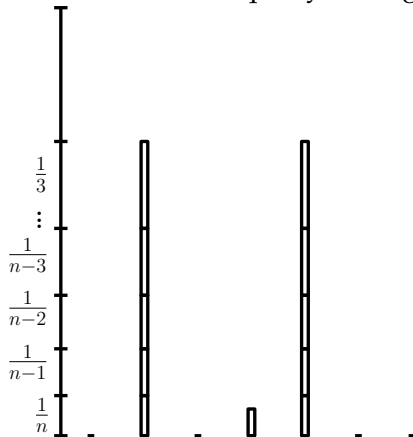
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

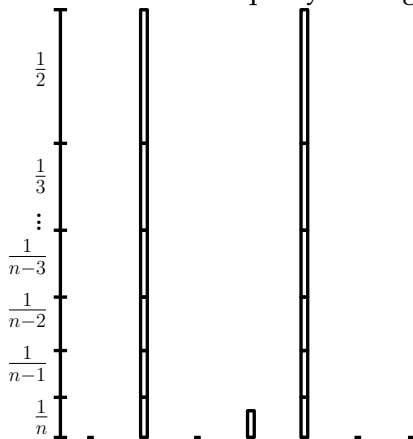
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

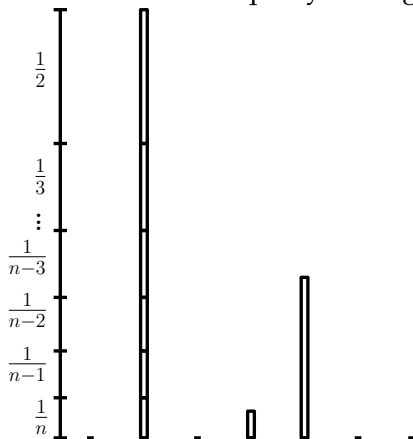
Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

Distribute water equally amongst cups not yet emptied from.



SINGLE-PROCESSOR LOWER BOUND

Filling strategy:

Distribute water equally amongst cups not yet emptied from.

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

- ▶ S_t : state at start of round t
- ▶ I_t : state after the filler adds water on round t , but before the emptier removes water
- ▶ $\mu_k(S)$: average fill of k fullest cups at state S .

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 on round t

If a is one of the k fullest cups in S_{t+1} :

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

Otherwise:

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

PREVIOUS WORK ON CUP GAMES

- ▶ The Single-Processor cup game ($p = 1$) has been tightly analysed with *oblivious* and *adaptive* fillers (i.e. fillers that can't and can observe the emptier's actions).
- ▶ The Multi-Processor cup game ($p > 1$) is substantially more difficult. With an adaptive filler:
 - ▶ Kuszmaul established upper bound of $O(\log n)$.⁶
 - ▶ 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 semi-clairvoyance have been studied.

⁶William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. SIAM, 2020.

OUR VARIANT

Definition

Variable-Processor Cup Game:

Each round filler can change p

Modification may seem small, but it drastically alters the game

Adaptive Filler Lower Bound

NEGATIVE FILL

In lower bound proofs we allow *negative fill*

- ▶ Measure fill relative to average fill
- ▶ Important for recursion
- ▶ Strictly easier for the filler if cups can zero out

AMPLIFICATION LEMMA

Lemma

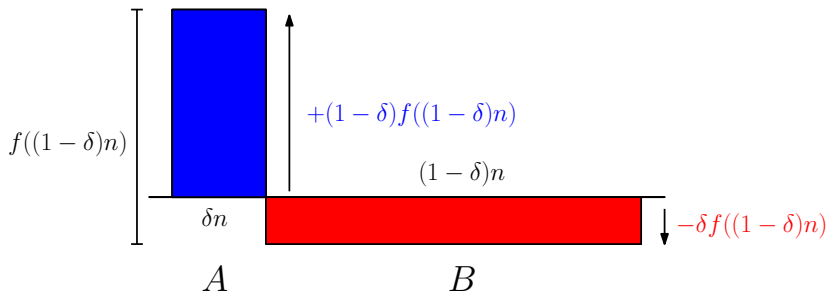
Given a strategy for achieving backlog $f(n)$ on n cups, we can construct a new strategy that achieves backlog

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

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

AMPLIFICATION LEMMA PROOF SKETCH

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

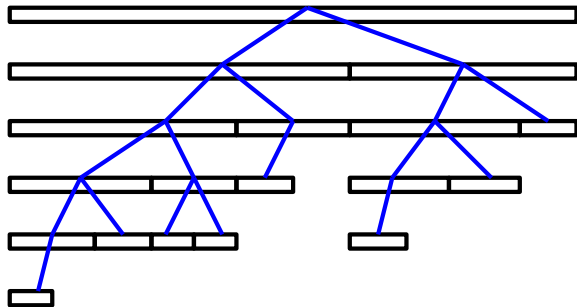


ADAPTIVE FILLER LOWER BOUND

Let $\epsilon > 0$ be any constant. Then there is some $\delta = \Theta(1)$ such that by repeated amplification we get:

Theorem

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



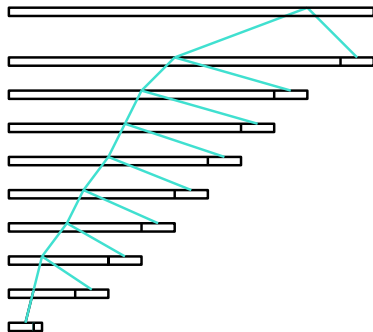
ADAPTIVE FILLER LOWER BOUND

Extremal strategy:

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

Theorem

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



Upper Bound

UPPER BOUND

We prove a novel set of invariants:

Theorem

A greedy emptier maintains the invariant:

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

Corollary

A greedy emptier never lets backlog exceed

$$O(n).$$

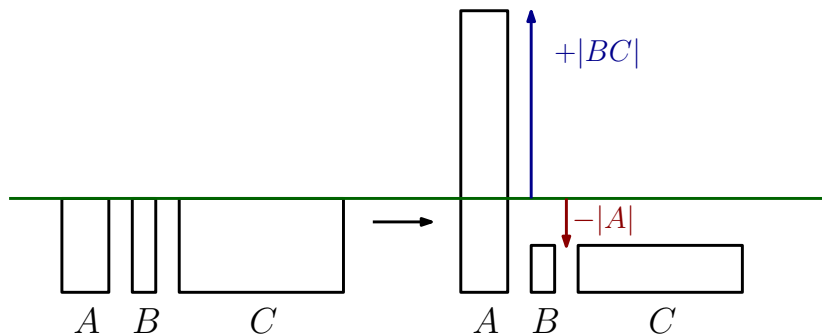
Note: this matches our lower bound!

UPPER BOUND PROOF SKETCH

Induct on t . Fix k . Define sets of cups:

- ▶ A : (emptied from) \cap (k fullest in S_t) \cap (k fullest in S_{t+1})
- ▶ B : (emptied from) \cap (k fullest in S_t) \cap (**not** k fullest in S_{t+1})
- ▶ C : AC is the k fullest cups in S_{t+1}

$\mu_k(S_{t+1})$ is largest if fill from BC is pushed into A



Oblivious Filler Lower Bound

OBLIVIOUS FILLER LOWER BOUND

Definition

Oblivious Filler: Can't observe the emptier's actions

- ▶ Classically emptier does better in the randomized setting.
- ▶ But not in the variable-processor cup game!
- ▶ We get the same lower bound as with an adaptive filler in quasi-polynomial length games!

OBLIVIOUS FILLER LOWER BOUND

Definition

Δ -greedy-like emptier:

Let x, y be cups. If $\text{fill}(x) > \text{fill}(y) + \Delta$ then a Δ -greedy-like emptier empties from y *only if* it also empties from x .

Oblivious filler can achieve backlog $\Omega(n^{1-\epsilon})$ for $\epsilon > 0$ constant in running time $2^{\text{polylog}(n)}$ against a Δ -greedy-like emptier ($\Delta \leq O(1)$) with probability at least $1 - 2^{-\text{polylog}(n)}$.

FLATTENING

Definition

A cup configuration is R -flat if all cups have fills in $[-R, R]$.

Proposition

Oblivious filler can get a $2(2 + \Delta)$ -flat configuration from an R -flat configuration against a Δ -greedy-like emptier in running time $O(R)$.

OBLIVIOUS FILLER: CONSTANT FILL

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

- ▶ Play many single-processor cup games on $\Theta(1)$ cups blindly. Each succeeds with constant probability.
- ▶ By a Chernoff Bound with probability $1 - 2^{-\Omega(n)}$ at least a constant fraction nc of these succeed.
- ▶ Set $p = nc$.
- ▶ Fill nc known cups; because emptier is greedy-like it must focus on the nc cups with high fill before these cups.
- ▶ Recurse on the nc known cups with high fill.

OBLIVIOUS AMPLIFICATION LEMMA

Almost identical to the Adaptive Amplification Lemma!

Lemma

Given a strategy f for achieving backlog $f(n)$ on n cups, 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 against a greedy-like emptier.

(Note: Lemma is actually more complicated than this.)

OBLIVIOUS FILLER LOWER BOUND

Theorem

There is an oblivious filling strategy that achieves backlog

$$\Omega(n^{1-\epsilon})$$

for constant $\epsilon > 0$ with probability at least $1 - 2^{-\text{polylog}(n)}$ in running time $2^{O(\log^2 n)}$ against a greedy-like emptier.

Achieve this probability by a union bound on $2^{\text{polylog}(n)}$ events.

Proof notes:

- ▶ Similar to adaptive filler proof
- ▶ need larger base case for union bound to work; this doesn't harm backlog though