The Variable-Processor Cup Game

Alek Westover

Belmont High School

June 7, 2020
$p$-PROCESSOR CUP GAME ON $n$ CUPS

Filler: wants high backlog
Emptier: wants low backlog

In this talk we take the side of the filler (we want high backlog)
$p$-PROCESSOR CUP GAME ON $n$ CUPS

$n$ cups

filler adds $p$ units of water (with at most 1 unit per cup)
$p$-PROCESSOR CUP GAME ON $n$ CUPS

$n$ cups

**filler** adds $p$ units of water (with at most 1 unit per cup)

**emptier** chooses $p$ cups and removes (at most) 1 unit from each

Filler: wants high backlog

Emptier: wants low backlog

In this talk we take the side of the filler (we want high backlog)
**$p$-PROCESSOR CUP GAME ON $n$ CUPS**

- **Filler**: wants high backlog
- **Emptier**: wants low backlog

In this talk we take the side of the filler (we want high backlog)

---

$n$ cups

**Filler** adds $p$ units of water
(with at most 1 unit per cup)

**Emptier** chooses $p$ cups and
removes (at most) 1 unit from each

backlog
= fill of fullest cup
$p$-PROCESSOR CUP GAME ON $n$ CUPS

$n$ cups

**Filler**: wants high backlog

**Emptier**: wants low backlog

In this talk we take the side of the filler (we want high backlog)
**Important Application: Work Scheduling**

- New work arrives
- Allocate $p$ processors to tasks
- Backlog = farthest behind on any task
**Previous Work** ¹,²,³

*Adaptive filler:* can see emptier’s actions

**Theorem**

*With an adaptive filler optimal backlog is* $\Theta(\log n)$.

*Oblivious filler:* can not see emptier’s actions ("blindfolded")

**Theorem**

*With an oblivious filler optimal backlog is between* $\Omega(\log \log n)$ and $O(\log \log n + \log p)$ (with high probability in short games).

---

¹[C. L. Liu. Scheduling algorithms for multiprocessors in a hard real-time environment. JPL Space Programs Summary, 1969.]


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 and the $p$-processor cup game are *fundamentally different*!
Theorem

There is an adaptive filling strategy that achieves backlog

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

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

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

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

*A greedy emptier never lets backlog exceed*

\[ O(n). \]

This matches our lower bound!

**Corollary follows from more general theorem:**

**Theorem**

*A greedy emptier maintains the invariant:*

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

$\Delta$-greedy-like emptier:
Adaptive Filler
Lower Bound
Proof Sketch
**Amplification Lemma**

**Lemma**

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

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

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

*If the running time of $f(n)$ is $T(n)$ the running time of $f'(n)$ satisfies*

\[
T'(n) \leq n \sum_{\ell=0}^{L} n\delta^\ell T(n\delta^\ell(1 - \delta)).
\]
Proof Meta-Structure

- $A$ starts as the $\delta n$ 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 $A$. 

\[ (+ (1 - \delta)f((1 - \delta)n) - \delta f((1 - \delta)n) ) \delta n \]
Amplification Lemma Proof Sketch

Instantiate $A$ and $B$
Filling Strategy: Place 1 fill in each cup in $A$, try to apply $f$ to $B$. 
If the emptier *neglects* $A$ then the average fill of $A$ rises!
We repeat our strategy many times; if the emptier neglects $A$ too many times we get the desired backlog in $A$. 
If emptier doesn’t neglect $A$ filler can apply $f$ to $B$
Get 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
Swap cup into $A$ again
Amplification Lemma Proof Sketch

Swap this cup into $A$. 
Eventually average fill of $A$ is at least $(1 - \delta)f(n(1 - \delta))$.
Average fill of $B$ is $-(\delta)f(n(1 - \delta))$. 
Recurse on $A$ for $L$ levels of recursion.
Problem size shrinks by a factor of $\delta$ each time.
Amplification Lemma Proof Sketch

\[ f'(n) \geq (1 - \delta) \sum_{\ell=0}^{L} f(n\delta^\ell (1 - \delta)) \]
**Adaptive Filler Lower Bound**

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

**Theorem**

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

\[ f(\log_{1/(1-\delta)} n) 
\]

\[ f(\log_{1/(1-\delta)} n) - 1 
\]

\[ \vdots 
\]

\[ f(\log_{1/(1-\delta)} n) - 1 
\]

\[ \Theta(\log n) \]
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 $O(n!)$. 

\[ \Theta(n) \]

\[ \frac{f_n}{n_0} \]
\[ \frac{f_n}{n_0-1} \]
\[ \frac{f_n}{n_0-2} \]
\[ \vdots \]
\[ \frac{f_n}{n_0-k} \]
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
**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 $A$
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
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.

\[
\frac{1}{n-3} \quad \frac{1}{n-2} \quad \frac{1}{n-1} \quad \frac{1}{n}
\]
SINGLE-PROCESSOR LOWER BOUND

Filling strategy:
Distribute water equally amongst cups not yet emptied from.

\[
\frac{1}{n} \quad \frac{1}{n-1} \quad \frac{1}{n-2} \quad \frac{1}{n-3} \quad \ldots
\]
**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.

\[
\frac{1}{n} \quad \frac{1}{n-1} \quad \frac{1}{n-2} \quad \frac{1}{n-3} \quad \ldots \quad \frac{1}{2} \quad \frac{1}{1} \]

**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:

\[ \mu_k(S_t) \leq \frac{1}{k+1} + \ldots + \frac{1}{n}. \]

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 analyzed 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 \(O(\log n)\).
  - 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.

---

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

---

Previous Work — Restricted Versions

Cup flushing game (emptier can completely empty cups):\(^6\)
- \(\Omega(\log \log n)\) lower bound
- \(O(\log \log n)\) upper bound

Bamboo Garden Trimming (filler always adds same amount):\(^7\)
- 2 lower bound
- 2 upper bound

Cups are nodes in a graph, moves restricted based on graph structure. \(D\) is the diameter of the graph.
- \(\Omega(D)\) lower bound
- \(O(D)\) upper bound

---


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 $\Delta$-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 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