An Analysis of Call Admission Problems on Grids
LSD & LAW

Hans-Joachim Böckenhauer, Dennis Komm, Raphael Wegner

Department of Computer Science
ETH Zürich

February 9, 2018
Outline

1. Motivation and Definitions

2. Results
   - Lower Bounds
   - Upper Bounds

3. Conclusion
Online Problems

Definition (Online Maximization Problem $\Pi$)

- Sequence of requests
- Satisfy one request before the next one arrives
- Maximize the gain
The Disjoint Path Allocation Problem (DPA)
The Disjoint Path Allocation Problem (DPA)
The Disjoint Path Allocation Problem (DPA)
The Disjoint Path Allocation Problem (DPA)
The Disjoint Path Allocation Problem (DPA)
How much information are we missing

- ...to be optimal?
- ...to achieve some competitive ratio?

⇒ New measure for complexity of online problems
Definition (Online Algorithm $\text{ALG}$ with Advice for $\Pi$)

- Adversary chooses online input instance
- Oracle with unlimited power knows instance and chooses infinite advice string
- $\text{ALG}$ can read an arbitrary long, but finite prefix
- $q(\cdot)$ is the advice complexity of $\text{ALG}$ $\iff \text{ALG}$ reads at most first $q(\cdot)$ bits of advice from start
- Advice complexity $s(n)$ of $\Pi$: maximum over all inputs of length $n$, for best pair of oracle and algorithm
Definition (Competitive ratio with advice)

- $\Pi$ is online maximization problem
- $\text{ALG}$ is online algorithm with advice for $\Pi$
- $\text{OPT}(I)$ is an optimal (offline) solution for instance $I$ of $\Pi$

$\text{ALG}$ is $c$-competitive for $\Pi$ if there exists a constant $\alpha \geq 0$ such that

$$\text{gain (OPT}(I)) \leq c \cdot \text{gain (ALG}(I)) + \alpha$$

for all instances $I$ of $\Pi$. 
Extended to Grids

- Height: $m - 1$
- Length: $n - 1$
The Call Admission Problem on Grids (CAPG)

Definition (CAPG)

- Online maximization problem $\Pi_{\text{CAPG}}$
- Request is a pair of servers asking for a connection
- Every connection is fixed (no termination or modification)
- Only one connection per wire
- Goal: maximize the number of granted connections
Outline

1. Motivation and Definitions
2. Results
   - Lower Bounds
   - Upper Bounds
3. Conclusion
Theorem ([BBF+14])

To solve DPA optimally, $|E| - 1$ advice bits are necessary and sufficient.
$|E| - 1$ Advice Bits for DPA

- Optimal solution $\text{OPT}(I)$ indicated by bit string: $1 \iff$ end one request, start another one
- $P_i$ contains all requests of length $|E| - i + 1$ which do not contradict requests in $\text{OPT}(I)$ of earlier phase
$|E| - 1$ Advice Bits for DPA

- Optimal solution is unique
- $2^{|E|-1}$ different instances
- Optimal solutions $S_1, S_2$ have to differ before instances $I_1, I_2$ are distinct on their asked prefixes of requests
  \[ \implies \log_2(2^{|E|-1}) = |E| - 1 \text{ advice bits required for optimality} \]
Almost $|E|$ Advice Bits for CAPG

Can we just ask these instances on each column and row for CAPG?

- Consider long request in solution indicated by bit string
- If not satisfied we have much space for detours
  \[ \Rightarrow \text{bit string solution is not optimal anymore} \]

Mitigation:

- Ask only sufficiently small requests, i.e., only last four phases
  \[ \Rightarrow \text{bit string solution is optimal again} \]
- Still no unique optimal solution in general
Lemma

There are $t_{n+2}$ bit strings of length $n \in \mathbb{N}$ which contain at most three consecutive 0s, where $t_n$ denotes the $n$th tetranacci number $^a$.

\[ ^a t_n = t_{n-1} + t_{n-2} + t_{n-3} + t_{n-4}, \quad t_0 = 1, \quad t_1 = t_2 = 0, \quad t_3 = 1 \]
Proof sketch:

- Optimal solutions differ only in requests satisfied with paths of length 4 and have specific forms.
- Every optimal solution has to grant a detour before rejecting a request intended by bit string.
  \[ \implies \text{optimal solutions are distinct before the prefixes of respective instances are different} \]
- \( t_m^n \cdot t_n^n \) instances with different optimal solutions.
  \[ \implies \geq m \cdot \log_2(t_n) + n \cdot \log_2(t_m) \] advice bits required for optimality.
Theorem

Every optimal online algorithm with advice for CAPG on an \((m \times n)\)-grid \(G\) has to read at least \(m \cdot \log_2(t_n) + n \cdot \log_2(t_m)\) advice bits.
Corollary

*Every optimal online algorithm with advice for CAPG on an $(m \times n)$-grid $G$ has to read at least $0.94677 \cdot |E(G)| - m - n$ advice bits.*
Theorem

Every online algorithm with advice for CAPG which achieves a competitive ratio of $c \leq \frac{12}{11}$ on a grid $G$ has to read at least

\[
\left(1 + \left(6 - \frac{6}{c}\right) \log_2 \left(6 - \frac{6}{c}\right) + \left(\frac{6}{c} - 5\right) \log_2 \left(\frac{6}{c} - 5\right)\right) \frac{|E(G)|}{2}
\]

bits of advice.
Outline

1. Motivation and Definitions

2. Results
   - Lower Bounds
   - Upper Bounds

3. Conclusion
Trivial Bound

Theorem

There is an optimal online algorithm with advice for CAPG which reads at most $2|E| \cdot \lceil \log_2(|V|) \rceil \leq 2|E| \cdot \log_2(|E| + m + n)$ bits of advice for every $(m \times n)$-grid $G = (V, E)$.

- Oracle chooses some optimal solution
- $\leq |E|$ requests
- Encode both endpoints of every granted request
  $\implies 2 \cdot \lceil \log_2(|V|) \rceil$ bits per satisfied request
- Overall: $\leq 2|E| \cdot \lceil \log_2(|V|) \rceil$
How can we improve?

- Knowing which edges are used in an optimal solution sometimes not helpful (e.g., in case all edges are used, but some requests are contradicting)
- Need to transmit the “membership” to a request
- “Neighbouring” paths need to be distinguishable

$\Rightarrow$ Coloring problem in some auxiliary graph
The Auxiliary Graph $\hat{G}$

**Definition** ($\hat{G}(S) = (\hat{V}, \hat{E})$)

- Path $p$ in $S$ satisfying a request $\implies v_p \in \hat{V}$
- $\{v_g, v_h\} \in \hat{E} \iff g$ and $h$ share some vertex in $G$
- Edges of $G$ unused by $S$ are split up into connected components, $s.t.$ component $q$ corresponds to $v_q \in \hat{V}$ and the chromatic number $\chi(\hat{G}(S))$ is minimized
The Auxiliary Graph $\hat{G}$
Theorem

Let \( \mathcal{I} \) denote all possible instances of CAPG on a grid \( G = (V, E) \), and let \( S_{\text{opt}}(I) \) be the set of optimal solutions for an instance \( I \in \mathcal{I} \). Then, there is an optimal online algorithm with advice for CAPG using at most

\[
\max_{I \in \mathcal{I}} \min_{S \in S_{\text{opt}}(I)} \left[ |E| \cdot \log_2(\chi(\hat{G}))\right] + 2\left[ \log_2(\chi(\hat{G}(S))) \right]
\]

advice bits.
Oracle:

- Can compute all optimal solutions
- Selects optimal solution $S$, s.t.
  \[
  \lceil |E| \cdot \log_2(\chi(\hat{G})) \rceil + 2\lceil \log_2(\chi(\hat{G}(S))) \rceil
  \]
  is minimal
- Uses $2\lceil \log_2(\chi(\hat{G}(S))) \rceil$ bits to transmit $\chi(\hat{G}(S))$ in a self-delimiting encoding
- Colors corresponding connected components of $G$ according to $\chi(\hat{G}) \Rightarrow \chi(\hat{G})^{|E|}$ possibilities
- $\lceil |E| \cdot \log_2(\chi(\hat{G})) \rceil$ bits for transmitting the coloring
Algorithm (receiver):
- Recomputes the length of the encoding
- Reads off the coloring
- Decides accordingly
Corollary

There is an optimal online algorithm with advice for CAPG that reads at most \( \lceil |E| \cdot \log_2(\frac{1}{3}(2|E| + 7)) \rceil \) bits of advice.
Motivation and Definitions

Results

Conclusion

Lower Bounds

Upper Bounds

\( \hat{G} \) Bound – Limitations

- All paths are “neighboring”
  \[ \implies \hat{G} \text{ contains } n \text{ clique} \]

- Since \( \chi(\hat{G}) \geq \omega(\hat{G}) \) this upper bound can not be stronger than
  \[ \lceil |E| \cdot \log_2(n) \rceil \]
  \[ = \lceil |E| \cdot \log_2(\sqrt{|V|}) \rceil \]
  \[ \in O(|E| \cdot \log(|E|)) \]
Can we still improve?

It suffices to be able to

- Distinguish the end vertices of different satisfied requests
- Follow the path that is used to satisfy the request
  - Only three possible direction at every inner vertex
Theorem

There is an online algorithm with advice for CAPG that computes an optimal solution using at most $3|E|$ advice bits.

Corollary

Let $\mathcal{I}$ denote all possible instances of CAPG on a grid $G = (V, E)$, and let $S_{\text{opt}}(I)$ be the set of optimal solutions for an instance $I \in \mathcal{I}$. Then, there is an optimal online algorithm with advice for CAPG that uses at most

$$\left\lceil \log_2(5) \cdot k + \log_2(3) \cdot |V| \right\rceil + \left\lceil 2 \log_2(k) \right\rceil$$

advice bits, where $k$ is the number of requests in $I$. 
Proof Sketch

- Choose some optimal solution
- Treat each row and each column separately
- Cut off requests
- Color edges as before using the auxiliary graph

⇒ Can distinguish aligned paths in solution
Proof Sketch

- Color edges of unaligned paths in optimal solution additionally red
  \[\Rightarrow\] Can distinguish aligned and non-aligned paths in solution
Proof Sketch

- **Yellow**: next edge in clockwise direction belongs to same path, pivot around lower, left vertex of edge
- **Cyan**: next edge in clockwise direction belongs to same path, pivot around lower, left vertex of edge

implies Can follow non-aligned paths in solution
Proof Sketch

\[ \Rightarrow \] Eight color combinations

\[ \Rightarrow \] \( \log_2(8) = 3 \) bits per edge
1 Motivation and Definitions

2 Results
   - Lower Bounds
   - Upper Bounds

3 Conclusion
Conclusion

- Lower and upper bound already close
- Upper bounds applicable for other graphs