Maximizing Revenue for the Online Dial-A-Ride Problem

Ananya Das Christman
Middlebury College

Outline
• Background: What is an Online Problem?
• Analyzing Online Algorithms: Competitive Analysis
• Online Dial a Ride Problem
• Related Open Problems

Classical Computer Science Problems
• Sorting

All input known in advance – Offline Problems

• Closest Points

• Shortest Paths

Classical Computer Science Problems
• Sorting

• Closest Points

• Shortest Paths

Classical Computer Science Problems
• Sorting

Online problem - input arrives over time

• Closest Points

• Shortest Paths

Ski Rental Problem
Ski Rental Problem

Rent or Buy?

Offline Ski Rental Problem

• Ski resort – cheap but owner often shuts down in the middle of the season to go to Florida...
• Cost to Rent: $1 /day, Cost to Buy: $10
• Input: $d$ = number of days resort will stay open
• Goal: Decide whether to Rent or Buy to achieve cheapest cost

\[
\text{Alg-Check}_d(\text{input}: d): \\
\quad \text{if } (d < 10) \quad \text{Rent (Cost = } d) \\
\quad \text{else} \quad \text{Buy (Cost = } 10) \\
\]

Optimal Algorithm

Online Ski Rental Problem

Everyday for a ski season, you decide:
Rent – or – Buy?

Input (# days open) arrives over time.
Need a “good” Online Algorithm

Outline

• Background: What is an Online Problem?
• Analyzing Online Algorithms: Competitive Analysis
• Online Dial a Ride Problem
• Related Open Problems
Optimistic Algorithm for Ski Rental

Optimist-Alg:
If open on first day: BUY

Worst input?
Resort closes on Day 2

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th></th>
<th>Day 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Optimist-Alg (ON) $10 X X X

OPT $1 X X X

Pessimistic Algorithm for Ski Rental

Pessimist-Alg:
while (open):
RENT

Worst input?
Resort stays open all season

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th></th>
<th>Day 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
</tr>
</tbody>
</table>

Pessimist-Alg (ON) $1 $1 $1 $1

OPT $10

Cautious-Optimist for Ski Rental

Cautious-Optimist:
For the first 3 days: RENT
If still open on day 4, BUY

Worst input?
Resort closes on Day 5

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
<th></th>
<th>Day 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1</td>
<td>$1</td>
<td>$10</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Cautious-Alg (ON) $1 $1 $10 X X X

OPT $1 $1 $1 $10 X X X

How to Analyze an Online Algorithm?

• Compare its cost to the optimal offline cost, given the worst possible input.

• Compare using ratio: \[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})}
\]

• If \[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} \leq c
\] then ON is c-competitive

Cautious-Optimist:

For the first 3 days: RENT
If still open on day 4, BUY

Worst input?
Resort closes on Day 5

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
<th></th>
<th>Day 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1</td>
<td>$1</td>
<td>$10</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

Cautious-Alg (ON) $1 $1 $10 X X X

OPT $1 $1 $1 $10 X X X

How to Analyze an Online Algorithm?

• Compare its cost to the optimal offline cost, given the worst possible input.

• Compare using ratio: \[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})}
\]

• If \[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} \leq c
\] then ON is c-competitive

Competitive Analysis

Optimist Competitive Ratio

Optimist-Alg:
If open on first day: BUY

Worst input?
Resort closes on Day 2

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th></th>
<th>Day 90</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>$10</td>
</tr>
</tbody>
</table>

OPT $1 X X X

\[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} = 10
\]

\[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} = 1
\]
Pessimist Competitive Ratio

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>...</th>
<th>Day 90</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pessimist (ON)</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$10</td>
</tr>
<tr>
<td>OPT</td>
<td>$10</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

\[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} = \frac{90}{10} = 9
\]

Cautious-Optimist Competitive Ratio

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>Day 3</th>
<th>Day 4</th>
<th>Day 5</th>
<th>...</th>
<th>Day 90</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cautious (ON)</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$10</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OPT</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Cost(ON) = 13
Cost(OPT) = 4

\[\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} = \frac{13}{4} = 3.25\]

New-Cautious-Optimist Competitive Ratio

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>...</th>
<th>Day 9</th>
<th>Day 10</th>
<th>Day 11</th>
<th>...</th>
<th>Day 90</th>
</tr>
</thead>
<tbody>
<tr>
<td>New-Cautious (ON)</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$10</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>OPT</td>
<td>$6</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>$6</td>
</tr>
</tbody>
</table>

Cost(ON) = 19
Cost(OPT) = 10

\[\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} = \frac{19}{10} = 1.9\]

Generalized Competitive Ratio

<table>
<thead>
<tr>
<th>Day 1</th>
<th>Day 2</th>
<th>...</th>
<th>Day b-1</th>
<th>Day b</th>
<th>Day b+1</th>
<th>...</th>
<th>Day 90</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generalized (ON)</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$1</td>
<td>$b</td>
<td>X</td>
<td>X</td>
<td>$(b-1) + b</td>
</tr>
<tr>
<td>OPT</td>
<td>$b</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>$b</td>
<td></td>
</tr>
</tbody>
</table>

Cost(ON) = $(b - 1) + b = 2b - 1$

\[
\frac{\text{Cost}(\text{ON})}{\text{Cost}(\text{OPT})} < 2 - \frac{1}{b}
\]

Best ratio for any online algorithm!
(no other number of rental days yields a better ratio)

ON is: \((2 - \frac{1}{b})\)-competitive

Outline

- Background: What is an Online Problem?
- Analyzing Online Algorithms: Competitive Analysis
- Online Dial a Ride Problem
- Related Open Problems
Applications

OFFline vs ONline

OFFline ~ "Reservation Service": Receives all requests in advance

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
<th>Time</th>
<th>Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oak Street</td>
<td>Airport</td>
<td>9am</td>
<td>$15</td>
</tr>
<tr>
<td>Maple Ave</td>
<td>Grocery</td>
<td>3pm</td>
<td>$10</td>
</tr>
<tr>
<td>Pine Stree</td>
<td>Airport</td>
<td>12pm</td>
<td>$5</td>
</tr>
<tr>
<td>Elm Rd.</td>
<td>2nd Street</td>
<td>2:30pm</td>
<td>$50</td>
</tr>
<tr>
<td>1st Street</td>
<td>Main Street</td>
<td>6:30pm</td>
<td>$12</td>
</tr>
</tbody>
</table>

ONline ~ Uber: Receives each request only when it is released

Revenue = $15

OFFline vs ONline

ONline ~ Uber: Receives each request only when it is released

Revenue = $15

Revenue = $12
OFFline vs ONline

ONline ~ Uber: Receives each request only when it is released

Revenue = $15  $12  $17

Accept

ONline ~ Uber: Receives each request only when it is released

Revenue = $15  $12  $17  $100

Accept

Online-Dial-A-Ride

Model with a complete weighted Graph

Node (Location)  Edge (Road)

Online-Dial-A-Ride with Revenue

Input:
- Complete Weighted Graph
- Initial Location of Server (origin)
- Requests: (source, destination, release time, revenue)
- Time Limit (T)

Goal: Serve requests to Maximize Total Revenue within T

Offline version is NP-hard

Worst Input?

- Every request takes a long time to serve $V_{\text{max}}$
  (where $V_{\text{max}}$ is the maximum edge weight in the graph)
- After each request we serve, we have to move for a long time $V_{\text{max}}$

Segmented-Best-Path (SBP) Algorithm

1. Let $V_{\text{max}}$ be the maximum edge weight in the graph.
2. Split the total time $T$ into $T/V_{\text{max}}$ “segments”, each of length $V_{\text{max}}$
3. While there is still time remaining:
   Alternate between:
   a) Find the request set with the highest revenue that can be served within time $V_{\text{max}}$
   Let $S$ denote this set. Move to the source of the first request in $S$.
   (Wait until the end of the time segment)
   b) Serve the requests in set $S$.
   (Wait until the end of the time segment)
SBP Example

$T = 24, \ max = 6$

$\text{Revenue earned} = \$11$

$t = 12$

$t = 18$

$t = 24$

$\text{Revenue earned} = \$11 + \$6 + \$11 = \$28$
**SBP Example**

Revenue earned = $11 + $6 + $11 = $28
OPT = $11 + $6 + $11 + $10 = $38

**SBP is 1/5-Competitive Proof**

- Compare schedules of SBP and OPT, one segment at a time:
  - Consider the segments where SBP earns less than OPT
- Create a modified schedule SBP (initially SBP = SBP) that will contain additional copies of requests
  - For now: assume all requests served by OPT during a segment are available to SBP

**SBP is 1/5-competitive**

- SBP serves in only every other time segment, OPT may serve the entire time
  - For a pair of consecutive segments, SBP misses at most $\frac{1}{2}$ OPT’s revenue
  - Initially: revenue(SBP) $\leq$ 2 revenue(SBP)
Overlapping Request

• Both $r^*$ and $R$ can separately be served within a time segment
• So SBP will serve a set with revenue at least $\max(r^*, R)$
  \[ \text{revenue(SBP)} \geq \max(r^*, R) \]
• So SBP misses at most $\frac{1}{2}$ OPT

SBP is 1/5-competitive

• Recall: we assumed all requests served by OPT during a segment are available to SBP
• Need another factor of revenue(SBP) to compensate when this assumption is removed

\[
\text{revenue(SBP)} \leq 4 \text{revenue(SBP)} + \text{revenue(SBP)} \\
\leq 5 \text{revenue(SBP)}
\]

Uniform Revenue

• Useful when all requests have equal priority
• SBP is 1/4-competitive

Bipartite Graphs

SBP is $k$-competitive, where $k$ is the ratio between the minimum and maximum edge weights in the input graph
Outline

- Background: What is an Online Problem?
- Analyzing Online Algorithms: Competitive Analysis
- Online Dial a Ride Problem
- Related Open Problems

Summary of Results + Open Problems

<table>
<thead>
<tr>
<th>Competitive ratio of SBP for OLDARP</th>
<th>Uniform Revenue</th>
<th>Nonuniform Revenue</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted General Graphs</td>
<td>1/4</td>
<td>1/5</td>
</tr>
<tr>
<td>Weighted Bipartite Graphs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Can we achieve better ratios?

\*k is a fraction used to bound the minimum edge weight in the input graph

Segmented-Best-Path (SBP) Algorithm

1. Let $\text{max}$ be the maximum edge weight in the graph.
2. Split the total time $T$ into $\text{max}$ disjoint segments.
3. While there is still time remaining:
   Alternate between:
   a) Find the request set with the highest revenue that can be served within time $\text{max}$; let $S$ denote this set. Move to the source of the first request in $S$. (Wait until the end of the time segment)
   b) Serve the requests in set $S$. (Wait until the end of the time segment)

Many Open Problems...

A fleet of Uber drivers

Serve all requests while minimizing route complexity

Thank You!

Lambus Li
Middlebury '20

Nick Jaczko
Middlebury '19

 항기 푸두
Middlebury '20

Amika Xu
Middlebury '20

Anita Vashistha
'17

Scott Westvold

Dr. Barbara Anthony
Southwestern University

Dr. Christine Chung
Connecticut College

Dr. David Yuen

Will Forcier
Lake Forest College '13

Annika Xu
Middlebury '20

Aspyn Poudel '18

Nick Jaczko
Middlebury '19

Anna Vashistha '17

Scott Westvold

Dr. Barbara Anthony
Southwestern University

Dr. Christine Chung
Connecticut College

Dr. David Yuen

Lambus Li
Middlebury '20

Nick Jaczko
Middlebury '19

Annika Xu
Middlebury '20

Aspyn Poudel '18

Dr. Barbara Anthony
Southwestern University

Dr. Christine Chung
Connecticut College

Dr. David Yuen

Lambus Li
Middlebury '20

Annika Xu
Middlebury '20

Aspyn Poudel '18

Dr. Barbara Anthony
Southwestern University

Dr. Christine Chung
Connecticut College

Dr. David Yuen

Lambus Li
Middlebury '20

Annika Xu
Middlebury '20

Aspyn Poudel '18

Dr. Barbara Anthony
Southwestern University

Dr. Christine Chung
Connecticut College

Dr. David Yuen