# Has this been solved?

## Some Open Problems in Online Algorithms

Arkaprava Choudhury

University of Toronto

*arka@cs.toronto.edu*

*https://www.cs.toronto.edu/∼arka/*

DCS Undergraduate Summer Research
August 29, 2022

# Presentation Overview

# Online algorithms

## The online framework

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case).

# Online algorithms

## The online framework

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case). After each input agent is presented, the algorithm makes a decision (either *revocable* or *irrevocable*).

# Online algorithms

## The online framework

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case). After each input agent is presented, the algorithm makes a decision (either *revocable* or *irrevocable*).

## Analysis

We are not concerned with the time complexity of online algorithms. Neither are we concerned with their space complexity...

# Online algorithms

## The online framework

A model of algorithms accepting an input instance given as an unknown sequence of inputs (agents, in this case). After each input agent is presented, the algorithm makes a decision (either *revocable* or *irrevocable*).

## Analysis

We are not concerned with the time complexity of online algorithms. Neither are we concerned with their space complexity...
So... how does one analyse such an algorithm?

# Analysing online algorithms

## Performance of online algorithms

We analyse online algorithms with respect to their performances on sequences of inputs, as compared with the optimal offline solution, if the items were already known.

## Competitive ratio

The worst case ratio between the result of an online algorithm and the best-case result.

# Input order models

## Adversarial order model

In analysis, adversary can choose both the input items and the sequence of items to be fed to the online algorithm.

## Random order model

In analysis, adversary can choose the input items but the sequence to be fed to the online algorithm is chosen uniformly at random.

# Secretary problem

## Classic formulation

You are sequentially presented, in random order, items from an adversarially selected set of items. Upon encountering each item, you can either choose to reject it permanently, or accept it, and end the program. You wish to maximize the value of the item you accept.

# Prophet secretary

## Prophet...?

The word prophet implies that the items are drawn independently from a set of distributions.

## Prophet secretary

Simply the secretary problem with the input items drawn independently from some (set of) distribution(s)

## Prophet inequalities

Prophet secretary, but with adversarial order of items.

## Multi-item prophet secretary / inequalities

Generalization to a case where we can accept multiple items subject to some feasibility constraints

# Feasibility constraints

# Feasibility constraints

## Matroids

A matroid $\langle E, \mathscr{I} \rangle$ is a tuple of a base set $E$ and a downward-closed collection of subsets $\mathscr{I} \subseteq \mathscr{P}(E)$ such that

- $\emptyset \in \mathscr{I}$
- If $A, B \in \mathscr{I}$ with $|A| < |B|$, then there exists $x \in B \backslash A$ s.t. $A \cup \{x\} \in \mathscr{I}$

# Feasibility constraints

## Matroids

A matroid $\langle E, \mathscr{I} \rangle$ is a tuple of a base set $E$ and a downward-closed collection of subsets $\mathscr{I} \subseteq \mathscr{P}(E)$ such that

- $\emptyset \in \mathscr{I}$
- If $A, B \in \mathscr{I}$ with $|A| < |B|$, then there exists $x \in B \backslash A$ s.t. $A \cup \{x\} \in \mathscr{I}$

## Matroid feasibility constraint

Given a matroid constraint $\langle E, \mathscr{I} \rangle$, you can take a subset $A$ of items from the set of items $E$ only if $A \in \mathscr{I}$.

# Knapsack and bin-packing

## Online knapsack problem

You have one knapsack with a fixed capacity, and you are presented items sequentially. You can either choose to pack an item, or revoke it forever. You wish to maximize the amount of items you packed.

## Online bin-packing problem

You have a fixed number of items, of varying sizes, and an unlimited supply of bins to pack the items into. After allocating an item to a bin, you may not change its position. You wish to minimize the number of bins used in the end.

# Generalized assignment problem

## Intuition

*Think of this as just a generalization of both the knapsack and bin packing problem.*

# Open problems in ROM

| Problem | Upper bound | Lower bound |
|---|---|---|
| Matroid prophet secretary | — | $1-1/e$ |
| Matroid prophet inequality | 0.745 | $1-1/e$ |
| Online knapsack | — | $1/6.65$ |
| Online bin-packing | $10/11$ | $2/3$ |
| Online GAP | — | $1/6.69$ |

Table: Online problems and known bounds on competitive ratio

# Open problems in ROM

| **Problem** | **Upper bound** | **Lower bound** |
|---|---|---|
| Matroid prophet secretary | – | $1-1/e$ |
| Matroid prophet inequality | 0.745 | $1-1/e$ |
| Online knapsack | – | $1/6.65$ |
| Online bin-packing | $10/11$ | $2/3$ |
| Online GAP | – | $1/6.69$ |

Table: Online problems and known bounds on competitive ratio

## Gaps in bounds

Any differences in these bounds mentioned above are open problems yet to be resolved in online algorithm design.

# References


S. Ehsani, M. Hajiaghayi, T. Kesselheim, S. Singla

Prophet Secretary for Combinatorial Auctions and Matroids

*Symposium on Discrete Algorithms (SODA) 2018.*

J. Correa

Posted Prices for a Random Stream of Customers

*ACM Proceedings EC'17 (2017).*

H. Beyhaghi, N. Golrezaei, R. Paes Leme, M. Pal, B. Sivan

Improved Revenue Bounds for Posted-Price and Second-Price Mechanisms

*Operations Research 69(6) (2021).*

# References

S. Albers, A. Khan, L. Ladewig

Improved Online Algorithms for Knapsack and GAP in the Random Order Model

*Algorithmica 83(6) 2021.*

S. Albers, A. Khan, L. Ladewig

Best Fit Bin Packing with Random Order Revisited

*Algorithmica 83(9) 2021.*

C. Kenyon

Best Fit Bin Packing with Random Order

*SODA 1996.*

# Acknowledgements

**UofT**

- Prof. Allan Borodin

**Funding**

- UofT Dept of Comp Sci

# The End

Questions? Comments?