

# Secretary Problems with Convex Cost

Siddharth Barman, Shuchi Chawla, David Malec, Seeun Umboh  
 {sid, shuchi, dmalec, seeun}@cs.wisc.edu



WISCONSIN  
 UNIVERSITY OF WISCONSIN-MADISON

## Problem Definition

- Given  $U$ , pick  $A$  to maximize profit

$$\max_{A \subseteq U} \sum_{e \in A} v(e) - c \left( \sum_{e \in A} s(e) \right)$$

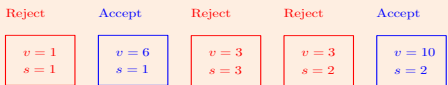
$$\text{s.t. } A \in \mathcal{F}$$

- $v$ : value function,  $s$ : size function;  $c$ : convex cost function
- $\mathcal{F}$  is a known feasibility constraint;  $v$  and  $s$  are adversarial
- Elements arrive one by one in random order
- Algorithm is online

Challenge: Objective function can take negative values; breaks all previous techniques

## Example

Online Stream



Cost function:  $c = s^2$   
 Total Profit =  $16 - 3^2 = 7$

## Applications

- Resource allocation problems:
- Wireless access point accepting connection requests
  - Cloud computing services accepting jobs
  - Sponsored search auctions accepting bids for ad slots

## Extension – Multi-Dimensional Costs

- $\ell$  cost functions  $c_1, \dots, c_\ell$ ;  $\ell$  sizes  $s_1, \dots, s_\ell$
- Given  $U$ , pick  $A$  to solve

$$\max_{A \subseteq U} \sum_{e \in A} v(e) - \sum_{i=1}^{\ell} c_i \left( \sum_{e \in A} s_i(e) \right)$$

$$\text{s.t. } A \in \mathcal{F}$$

## Results – Competitive Ratios

Single-Dimensional Cost

- Unconstrained  $O(1)$
- Matroid/knapsack constraints  $O(\alpha)^*$

Multi-Dimensional Cost

- Unconstrained  $O(\ell)$ , ← Tight to within a constant factor
- Matroid/knapsack constraints  $O(\alpha \ell)^*$

Extends work on submodular matroid secretary [Gupta et al. 10, Bateni et al. 10]

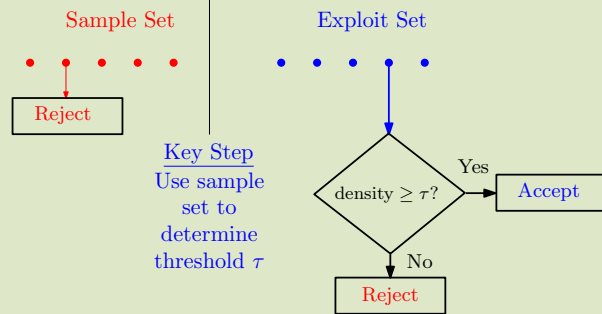
\*  $\alpha$  is the competitive ratio for the corresponding matroid/knapsack secretary problem

## Unconstrained Setting

Challenge:

Initial bad decisions may drive cost too high to accept even high-value elements later on

High-Level Approach



Intuition

- Define density = value/size, as in the greedy knapsack algorithm
- Greedy by decreasing density is near-optimal
- With high probability, we do not accept any element with density below cut-off for greedy by density

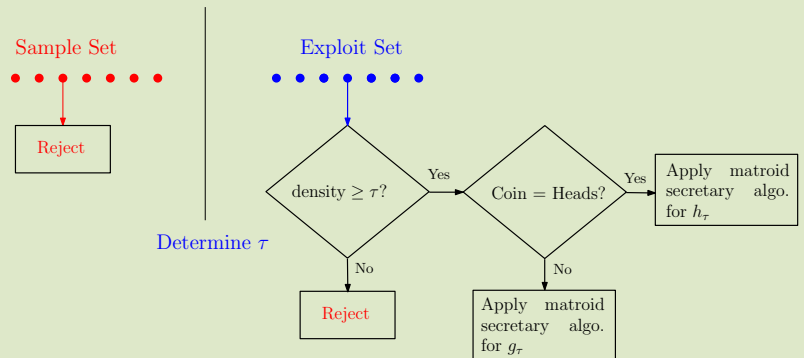
## Constrained Setting – Matroids

Challenge:

Greedy by density is no longer near-optimal

High-Level Approach

At the beginning of the algorithm, we toss a fair coin



Intuition

- We can decompose profit  $\pi(A) = g_\tau(A) + h_\tau(A)$ ; both  $h_\tau$  and  $g_\tau$  are sum-of-values type functions
- If  $\tau$  is "good", then optimizing both  $h_\tau$  and  $g_\tau$  are non-negative value matroid secretary problems