Online Algorithms for packing and covering problems

Amit Kumar

IIT Delhi

Joint work with

Anupam Gupta (CMU), Guru Guruganesh (CMU),
Ravishankar Krishnaswamy (MSR India) Debmalya Panigrahi (Duke),
David Wajc (CMU)

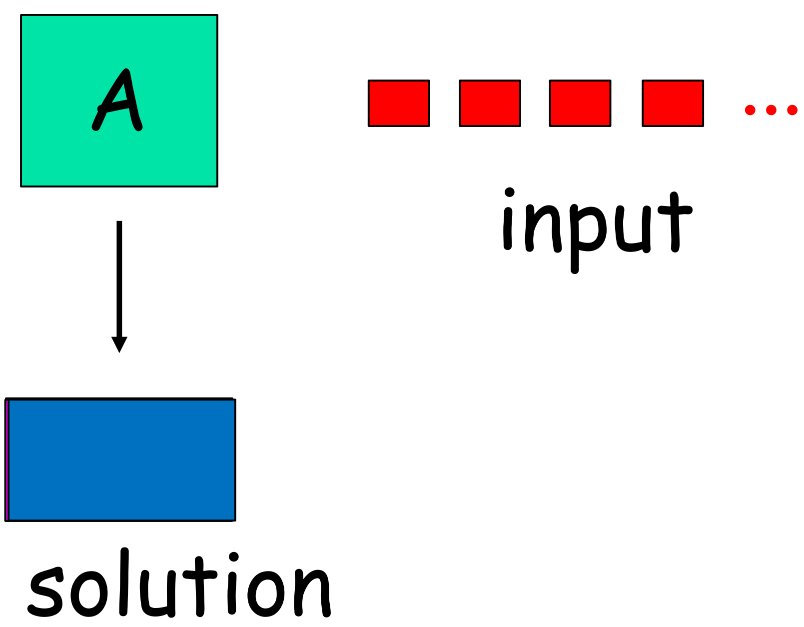
Outline

1. Online algorithms and competitive analysis
2. Online Bin packing
3. Online Set cover
4. Future Directions

What is “online algorithm“

The input to the algorithm arrives online….

"Shouldn't we be doing this online?"



Umbrella Rental Problem

Buying umbrella costs Rs 10

Can rent at Rs 1 per day

It will keep raining for n days, and then rain will stop.

Rent or buy ?

If we knew future, optimum strategy costs min(n, 10).

"off-line algorithm"

Umbrella Rental Problem

Buying umbrella costs Rs 10

Can rent at Rs 1 per day

Rent Rent Rent Buy

A strategy decides how long to rent

Competitive ratio: max $\dfrac{\text{Cost of Online strategy}}{\text{Optimal offline cost}}$

Umbrella Rental Problem

Buying umbrella costs Rs 10

Can rent at Rs 1 per day

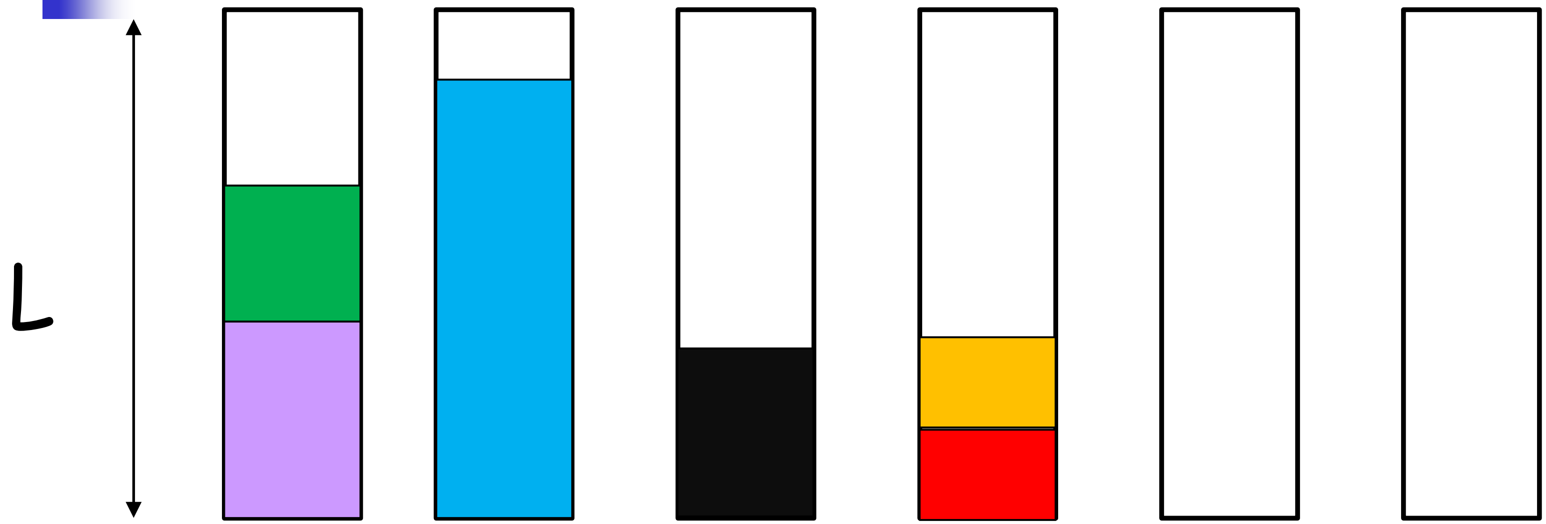
Rent for 10 days, and then buy

Competitive ratio = 2

Outline

1. Online algorithms and competitive analysis
2. Online Bin packing
3. Online Set Cover
4. Future Directions

Bin Packing Problem



Unlimited number of bins of size L each.

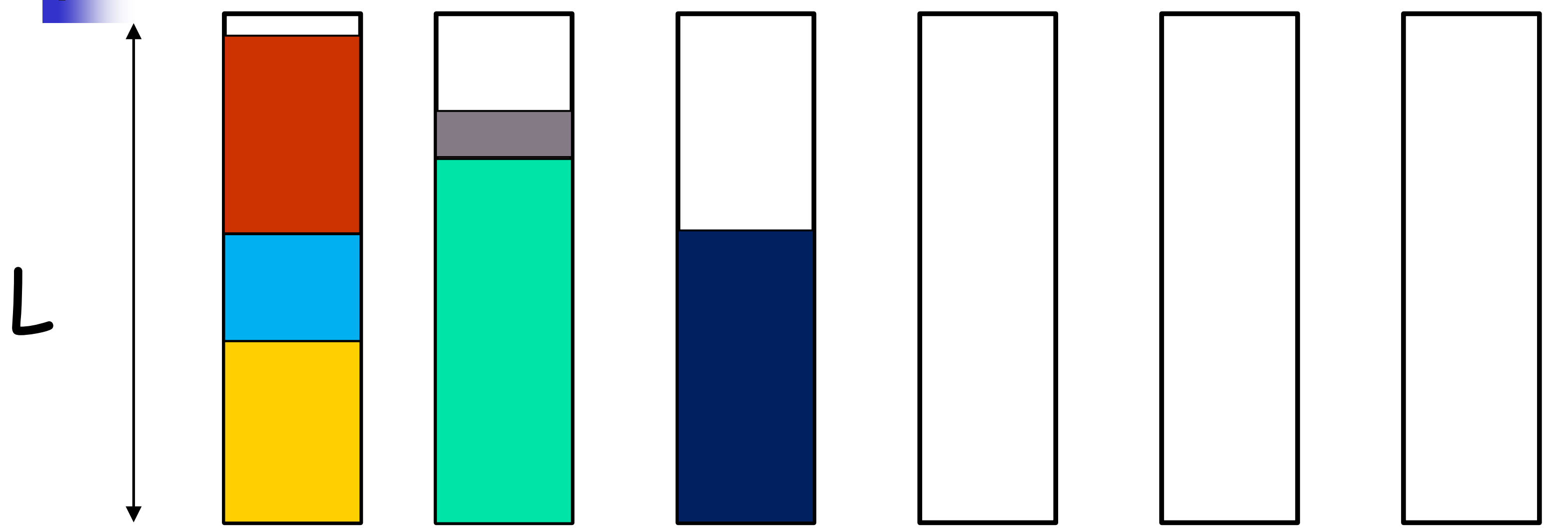
Items of varying size arrive on-line

Pack them using min. number of bins.

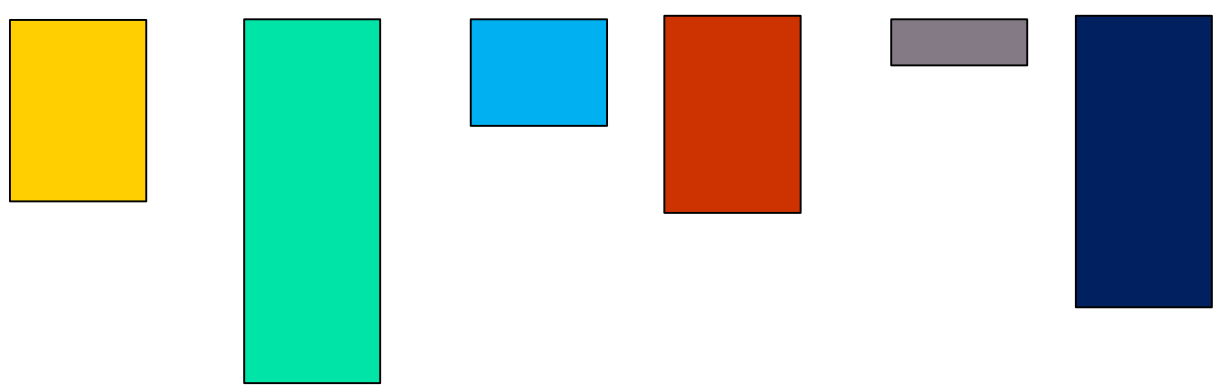
Who cares?

- Central problem in operations research and theoretical CS (since 1940's)
- Fertile ground for many algorithmic ideas
  "problem that won't go away" [CGJ '84]
- Applications in
  cutting stock, disk allocation, job scheduling
- Multidimensional versions used in
  VLSI layout, VM scheduling in data centers, container packing…
- Off-line version computationally hard
  Many rich techniques

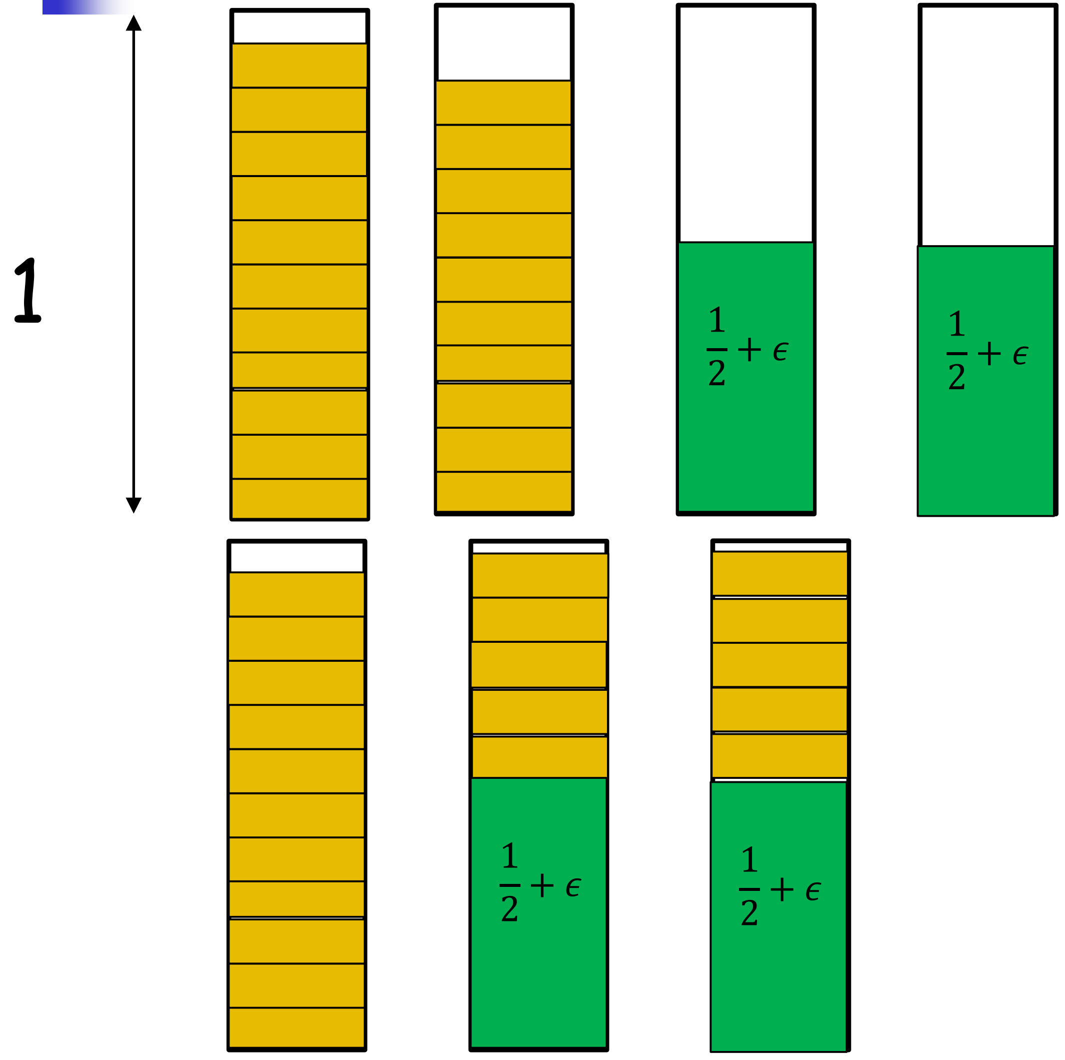
First Fit Algorithm



Pack the item in the first available bin

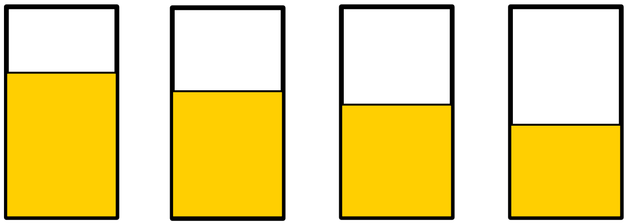


What can go wrong ?



What can we prove ?

- [Garey et al. '72] First Fit is 1.7-competitive.
- [Yao '82] Revised First Fit is 1.66-competitive.
- [Lee, Lee '85] Harmonic Algorithm is 1.69-competitive.



Leave some extra space deliberately

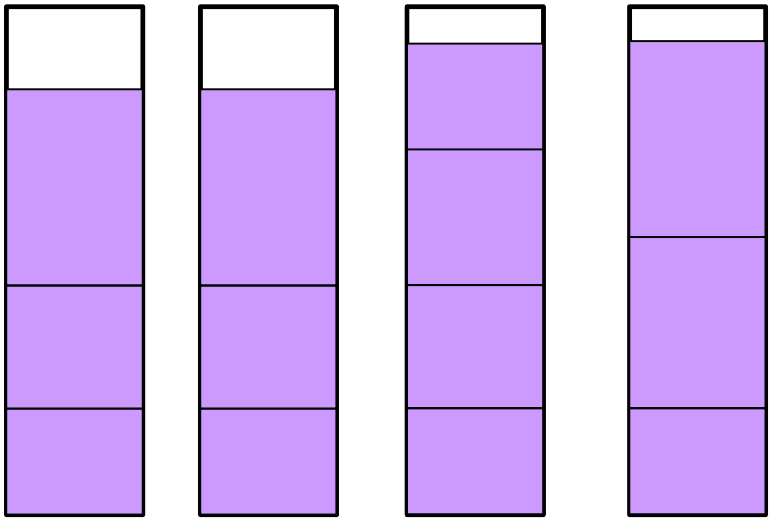
- …[Seiden '02] Harmonic++ 1.588-competitive

  (40 manually set parameters!)
- …[Heydrick and Stee] 1.5813-competitive

- Lower bound: 1.54 [Balogh et al. '12]

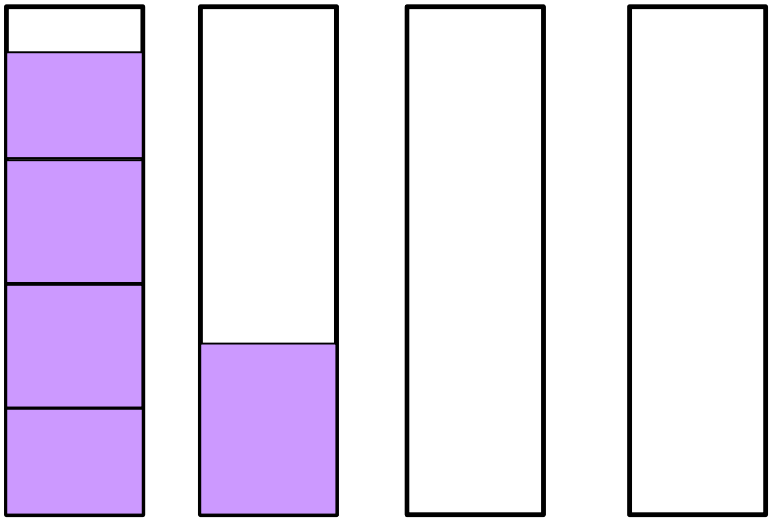
All ratios are "asymptotic"

Two New Twists



- Items can go away
- Allow some items to be repacked

Two New Twists

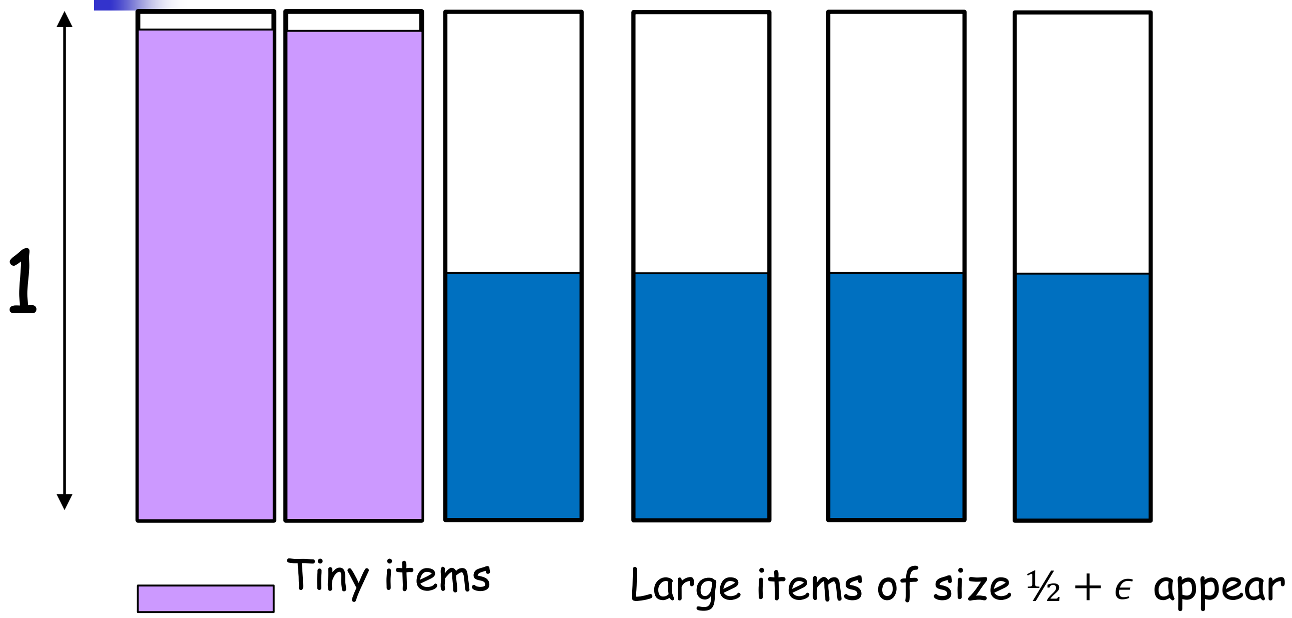


- Items can go away
- Allow some items to be repacked
- Want to understand trade-off between extent of repacking and quality of solution

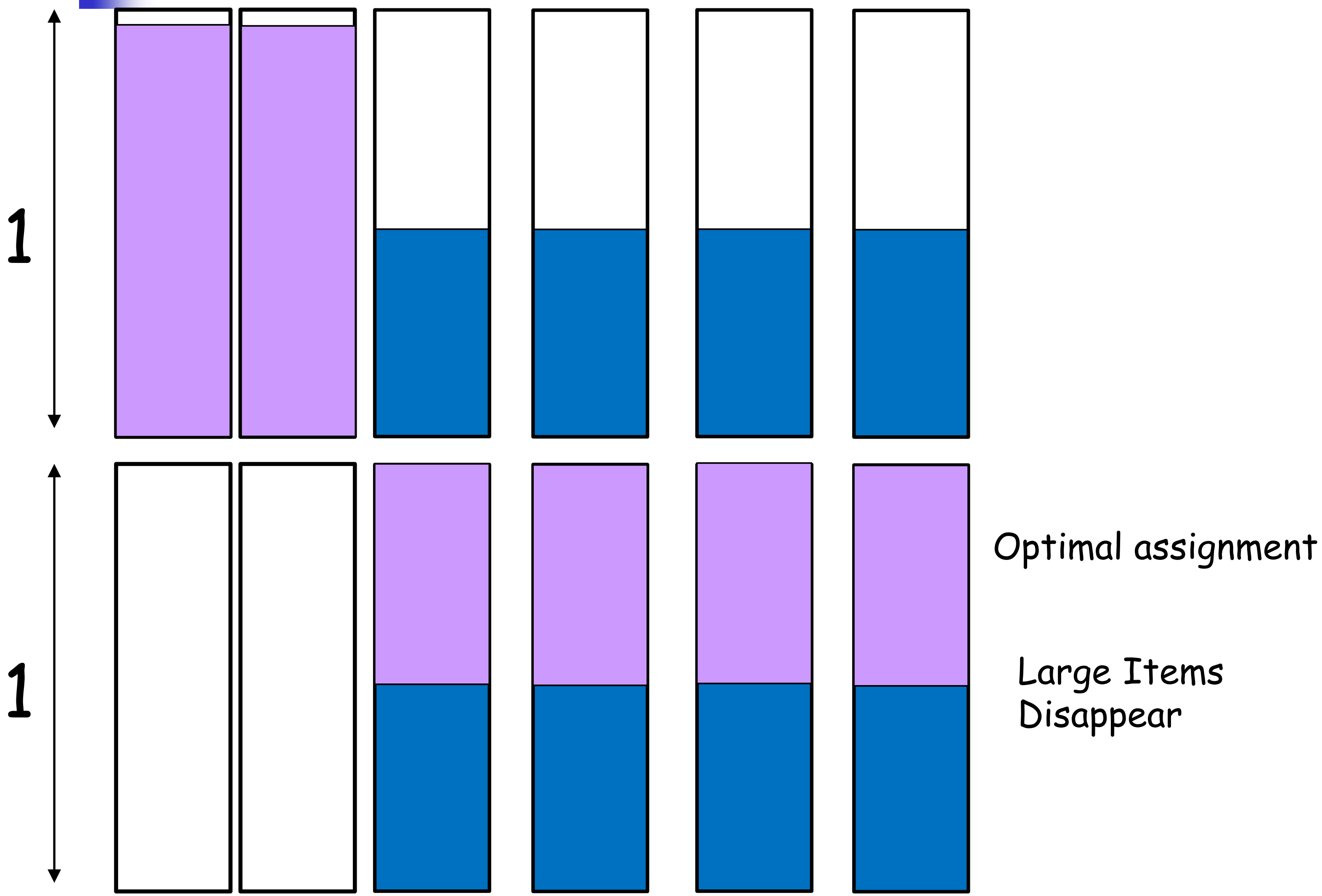
A concrete question

- Suppose we can repack 1 item whenever a new item arrives/departs.
  - [Gupta, Guruganesh, K., Wajc '18] : 1.38 competitive.
  - Results obtained independently by [Feldkord, Feldotto, Riechers '18]
  - No such algorithm can be better than 1.38-competitive.

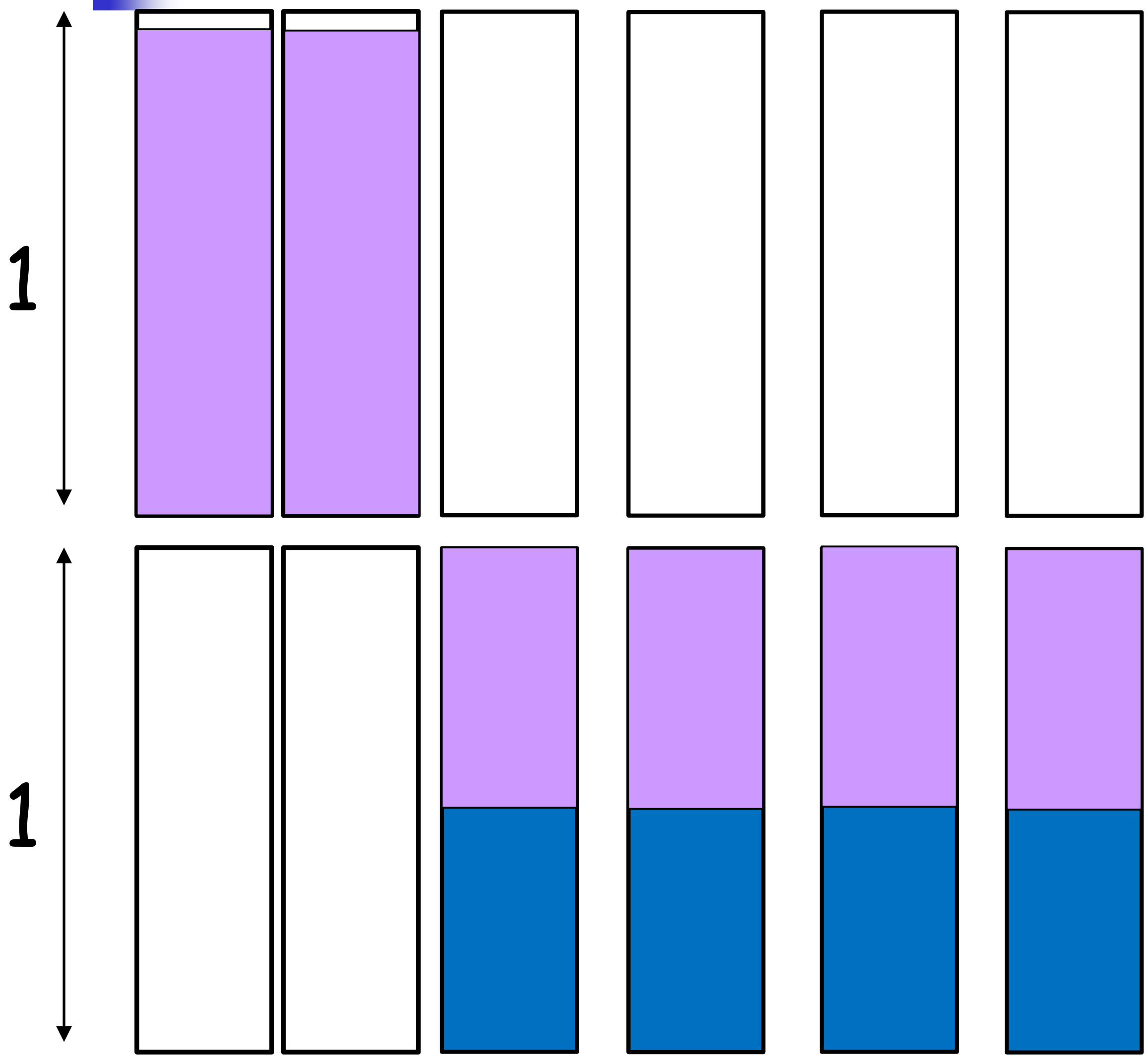
An example



An example

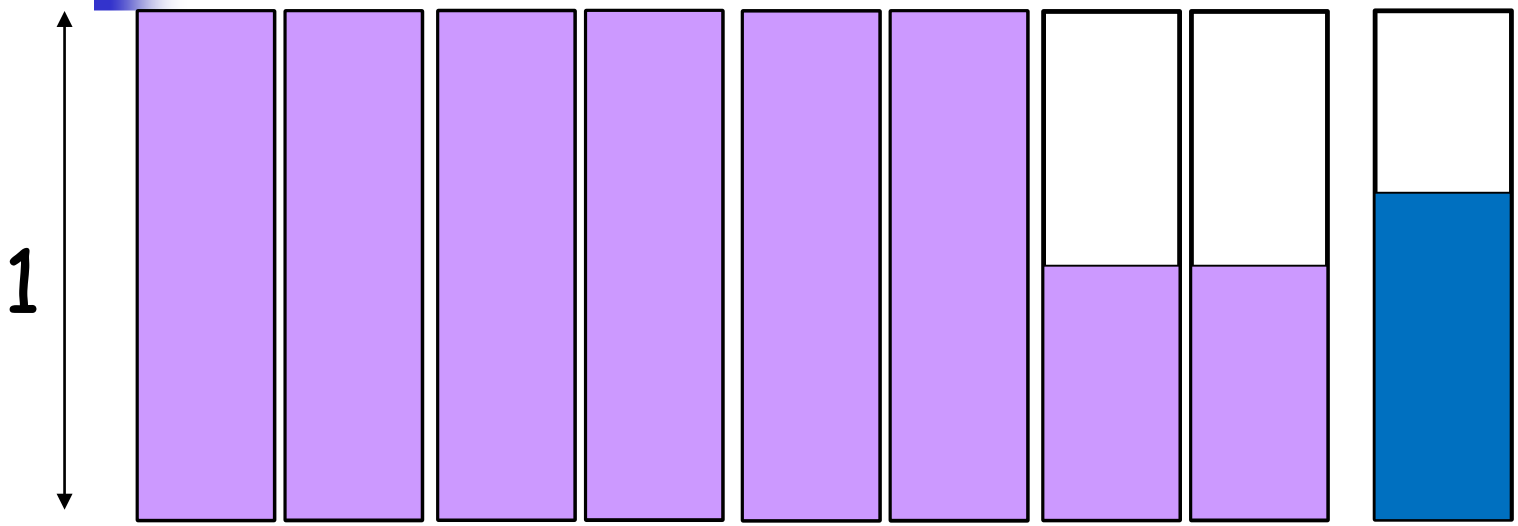


An example

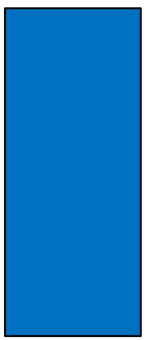


Need to go back and forth between these two solutions.

Algorithm Idea



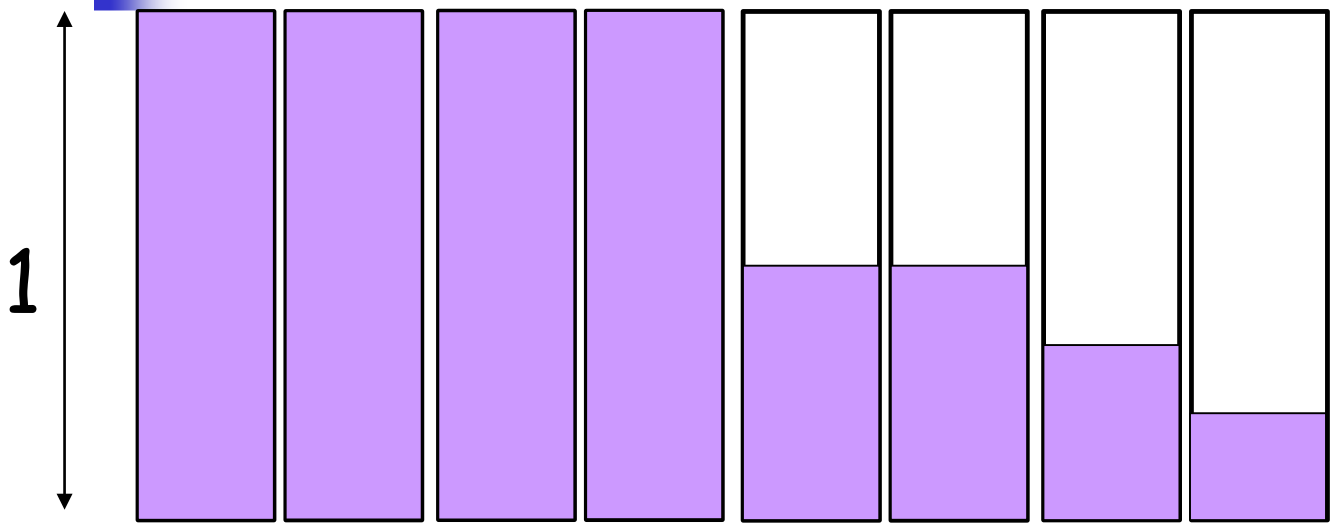
Keep a few bins half filled.



What if slightly larger items arrive ?

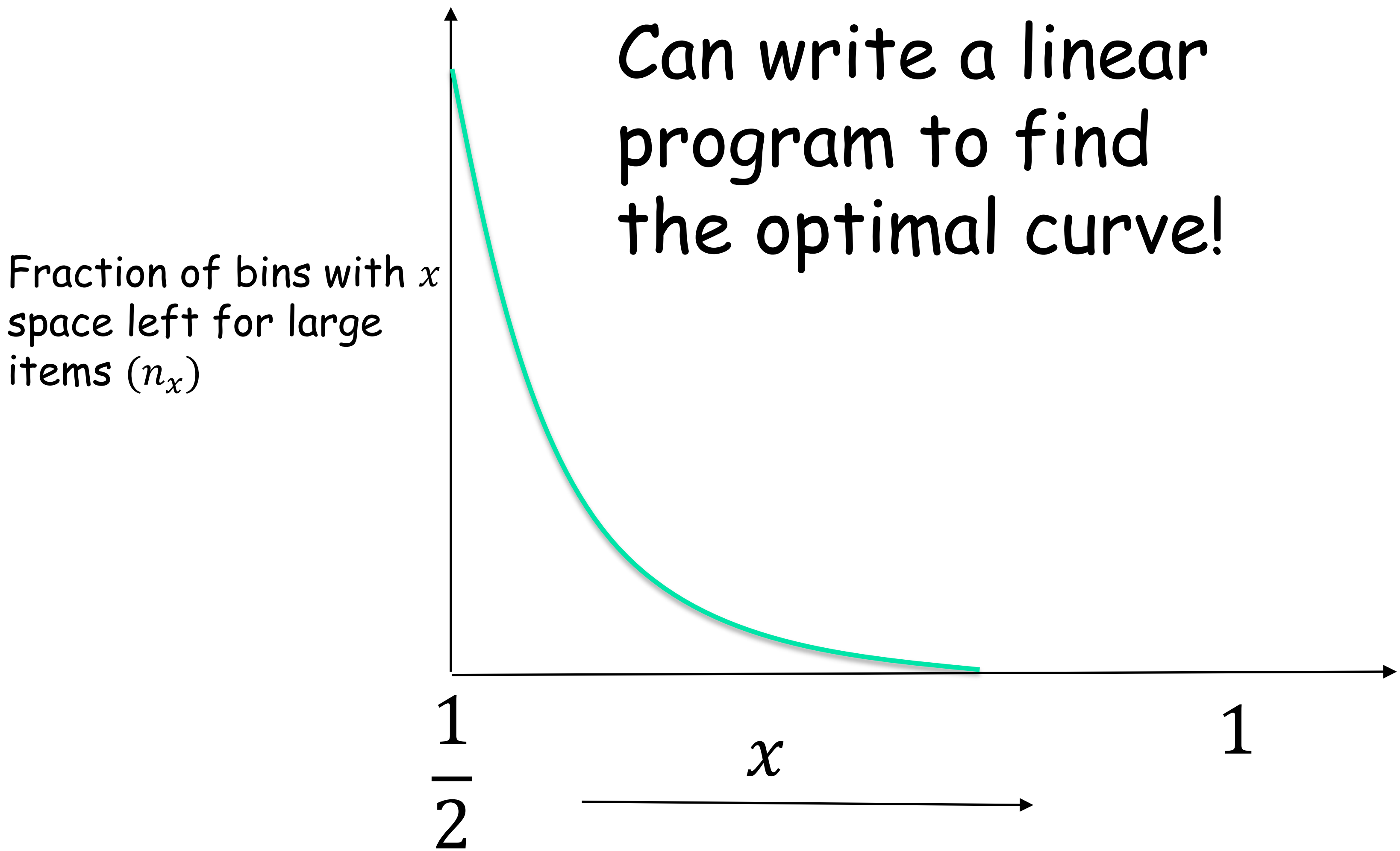
The fact that we left out space did not really help!

Algorithm Idea



For several sizes $s_1, s_2, \ldots, s_c$, keep some bins with s_i free capacity.

Algorithm Idea



Can write a linear program to find the optimal curve!

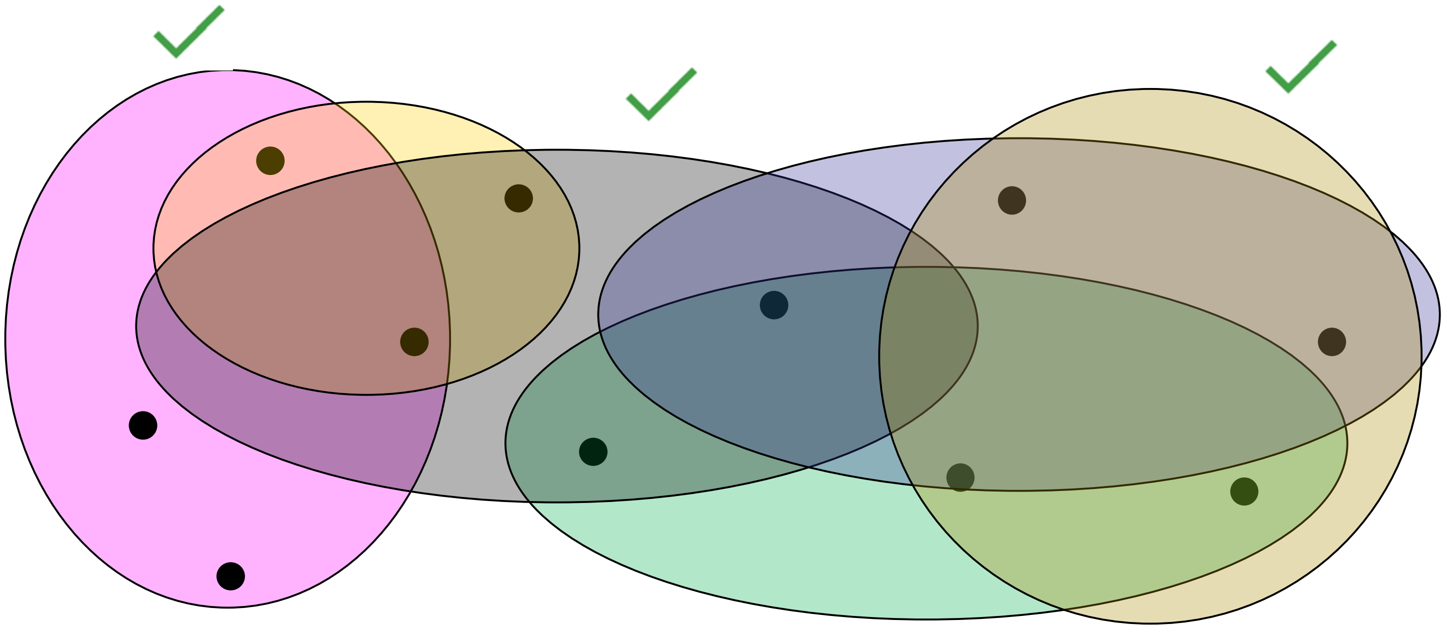
Extensions

- Algorithm can be made non-amortized
- Different items can have different movement costs.
- Items are multi-dimensional objects (or vectors) ?

Outline

1. Online algorithms and competitive analysis
2. Online Bin packing
3. Online Set Cover
4. Future Directions

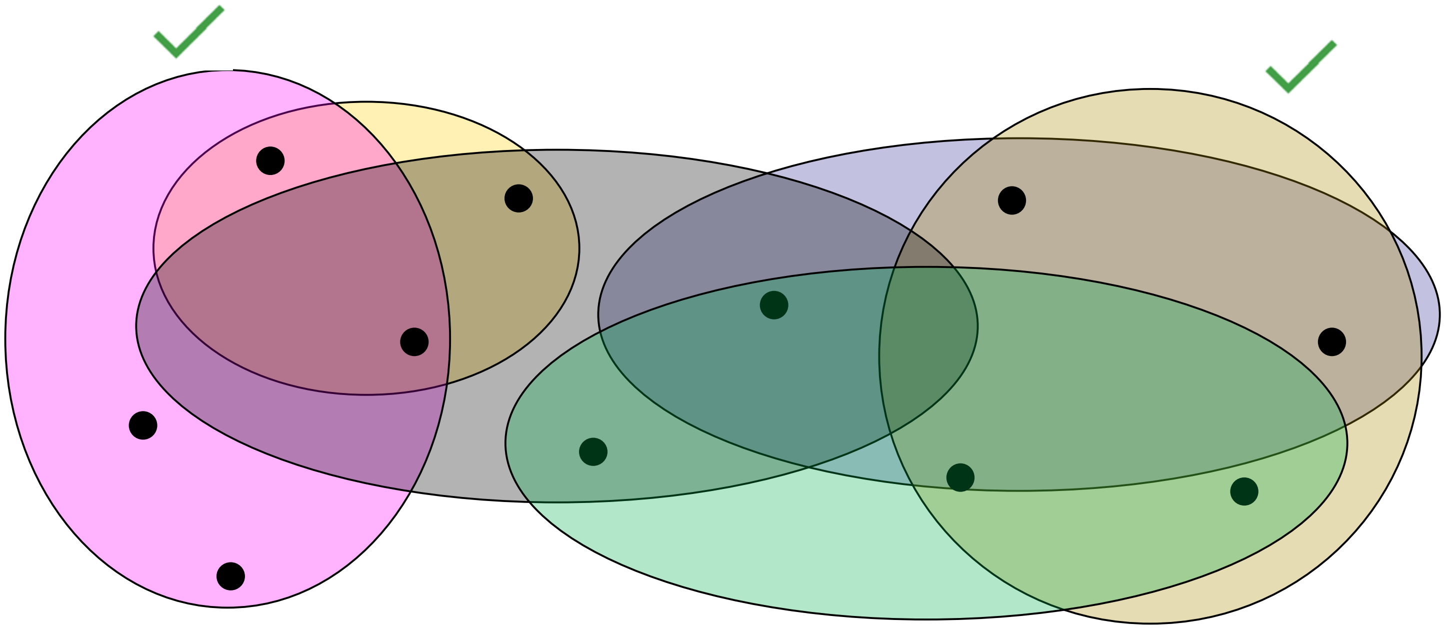
Set Cover Problem



A set-system defined by a set of points U and subsets containing them.

Find the smallest number of subsets which cover each element.

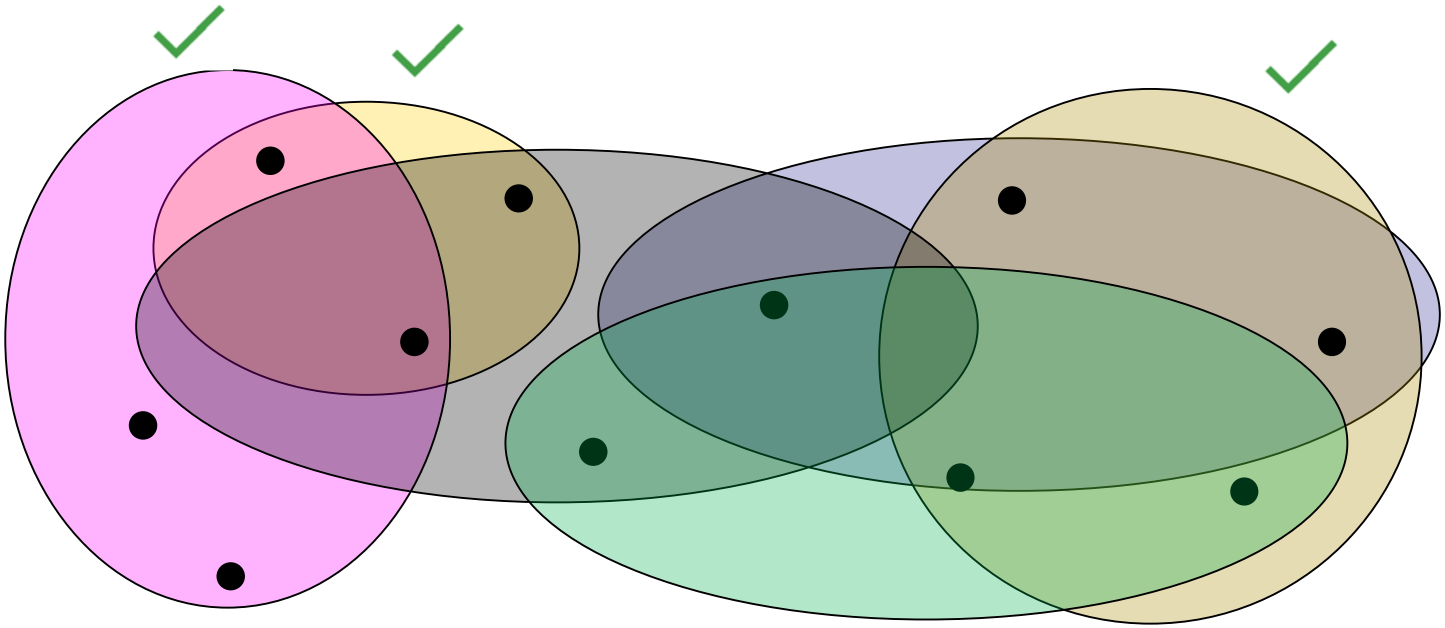
Online Set Cover Problem



Points arrive one by one.

Maintain a set cover of points which have arrived so far.

Our model formally…



- At each time, a point arrives or an existing point departs.
- Whenever a point arrives/departs, we are allowed to add or remove O(1) sets (amortized) in our solution.

Known work and Our results

Offline set-cover:

Greedy is $O(\log n)$-approximation.

Hardness of $\Omega(\log n)$ under complexity theoretic assumptions

[Lund, Yannakakis, 91] [Feige 98] [Dinur, Steurer '13]

Online set-cover:

$O(\log m \log n)$ –competitive algorithm [Alon et al. '09]

Matching hardness for deterministic/efficient algorithms.

[Alon et al. '09, Korman '05]

Our Result: O(log n)-competitive algorithm with O(1) change in solution on every arrival/departure of an element.

[Gupta, K., Panigrahi, Krishnaswamy '17]

Extensions

Can extend to weighted setting as well.

$O(f)$ competitive with $O(1)$ amortized change per arrival/departure.

Each arrival/departure can be implemented in $O(\log n)$ time

Non amortized ?

Conclusions

Interplay between competitive ratio and “recourse” a fundamental issue in online algorithms.

Bin packing, set-cover [this talk]

Minimum spanning tree [Gu, Gupta, K. ‘13]

Load balancing, matching, network flows [Gupta, K. ‘14]

Facility location [Fotakis ‘06, ’07], call admission [Adler, Azar ’01]

Extensions to many other problems remains a challenging future direction.

THANK YOU!