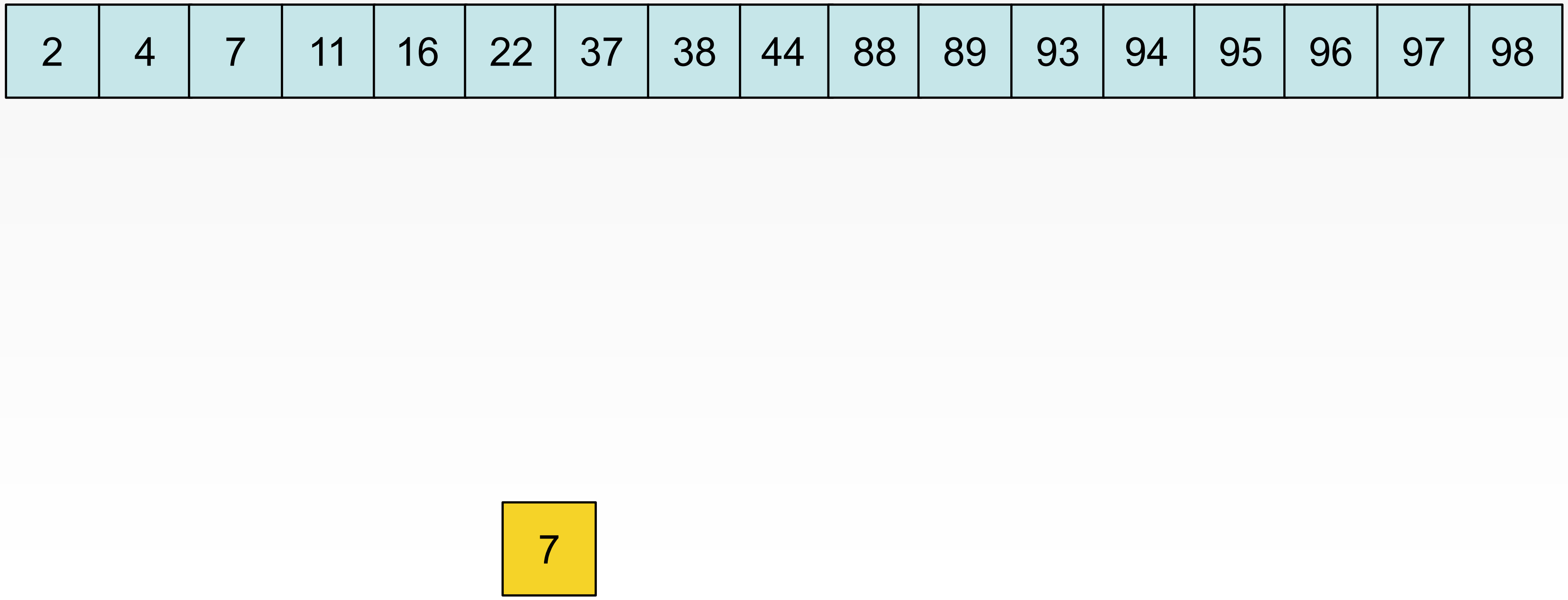
Lecture 3
Algorithms with Predictions

Slide by Sergei Vassilvitskii

Warm-up

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



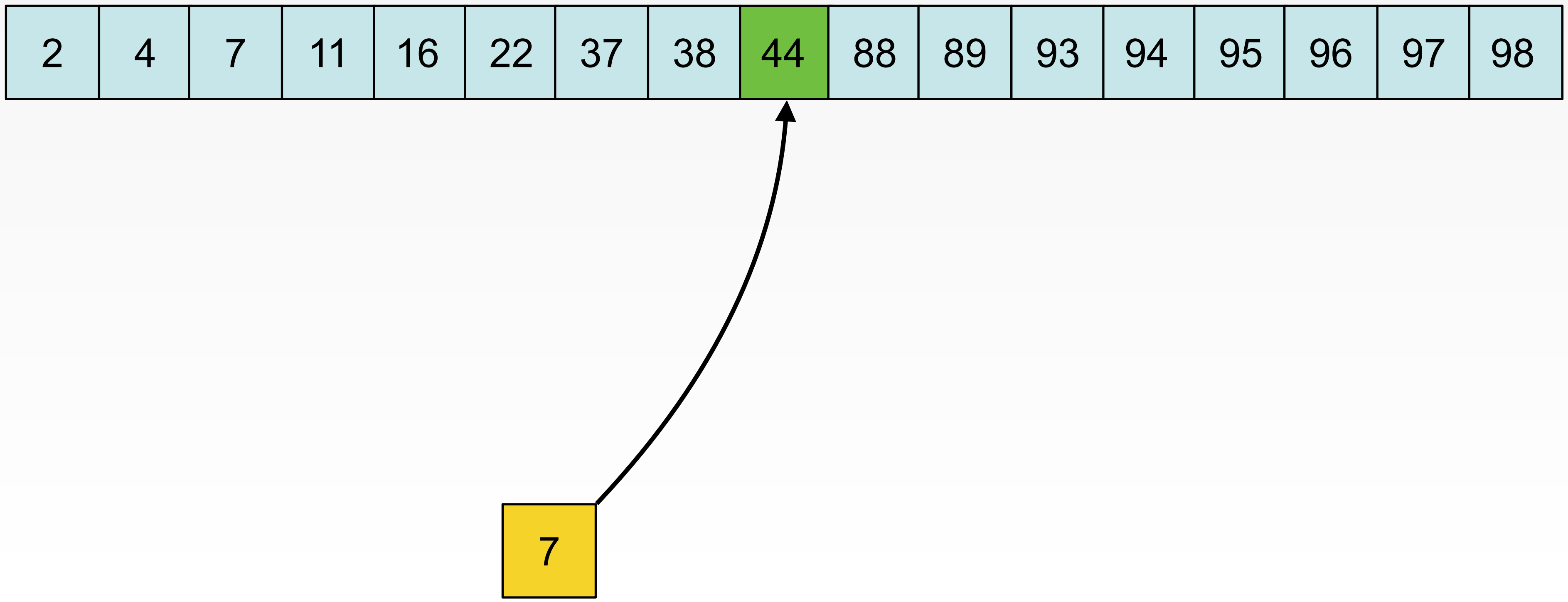
2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

7

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



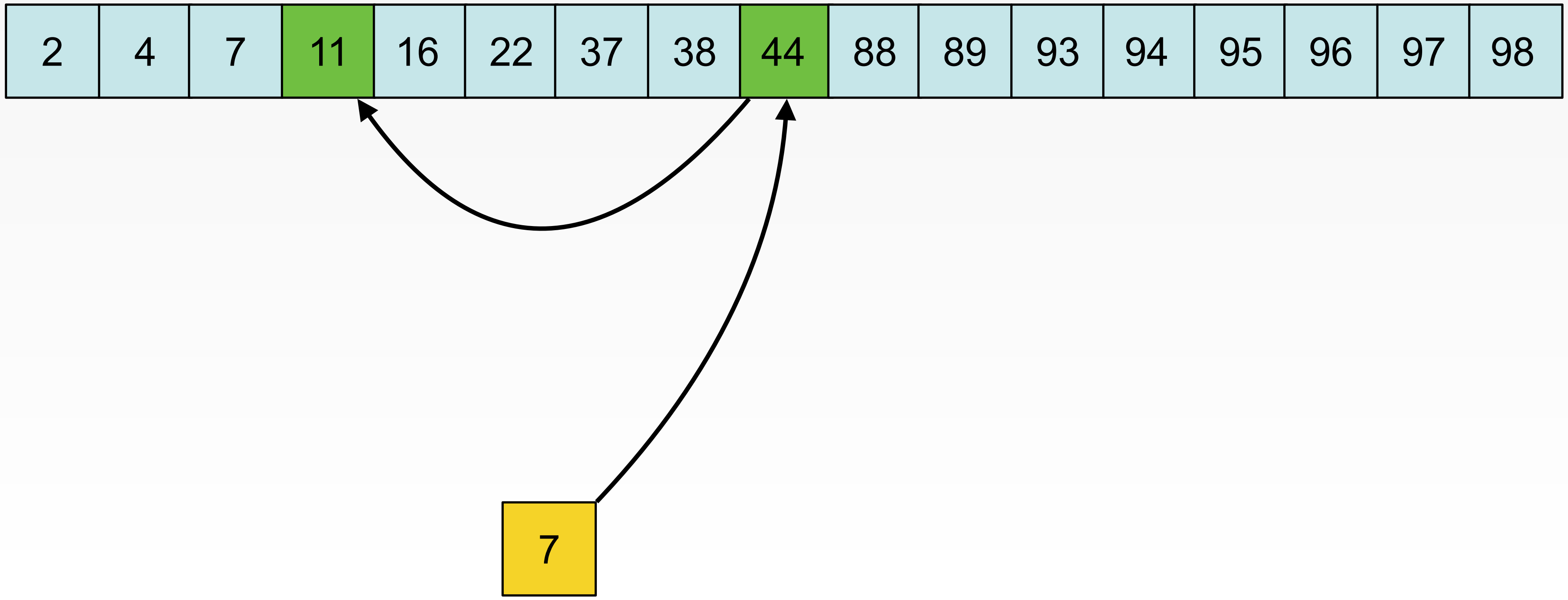
2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

7

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



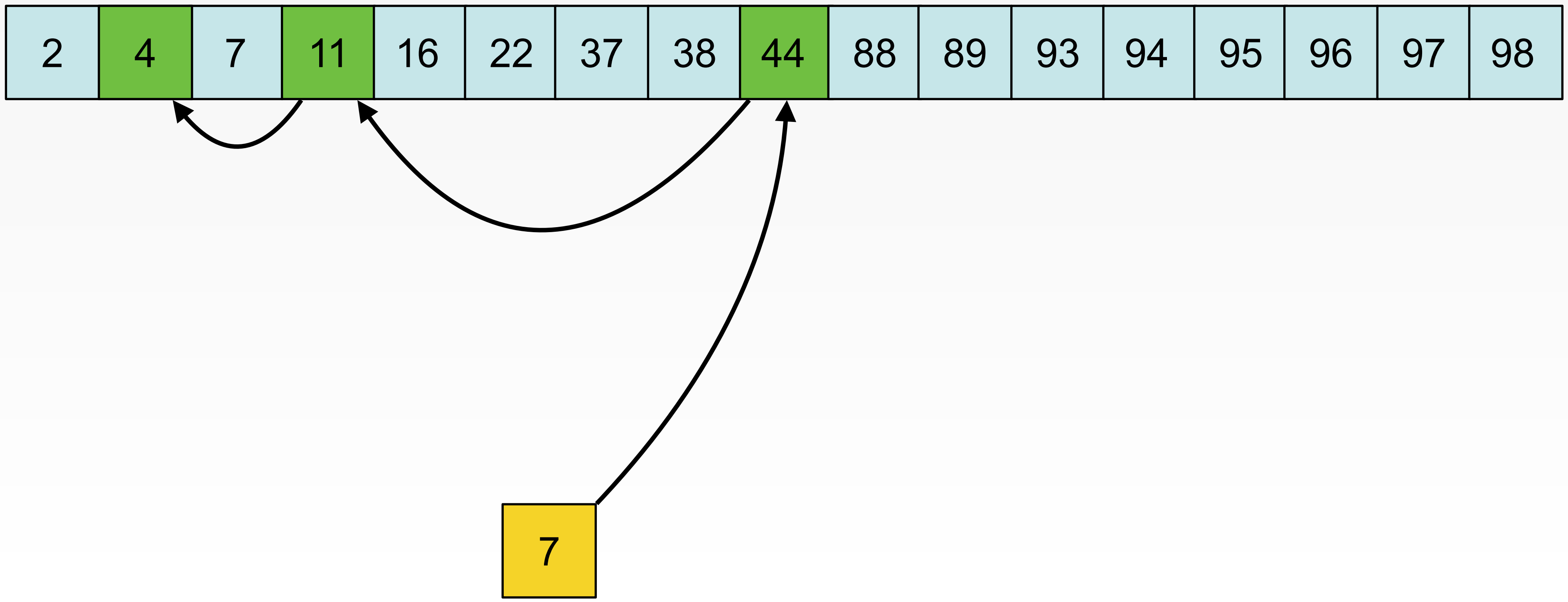
2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

7

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



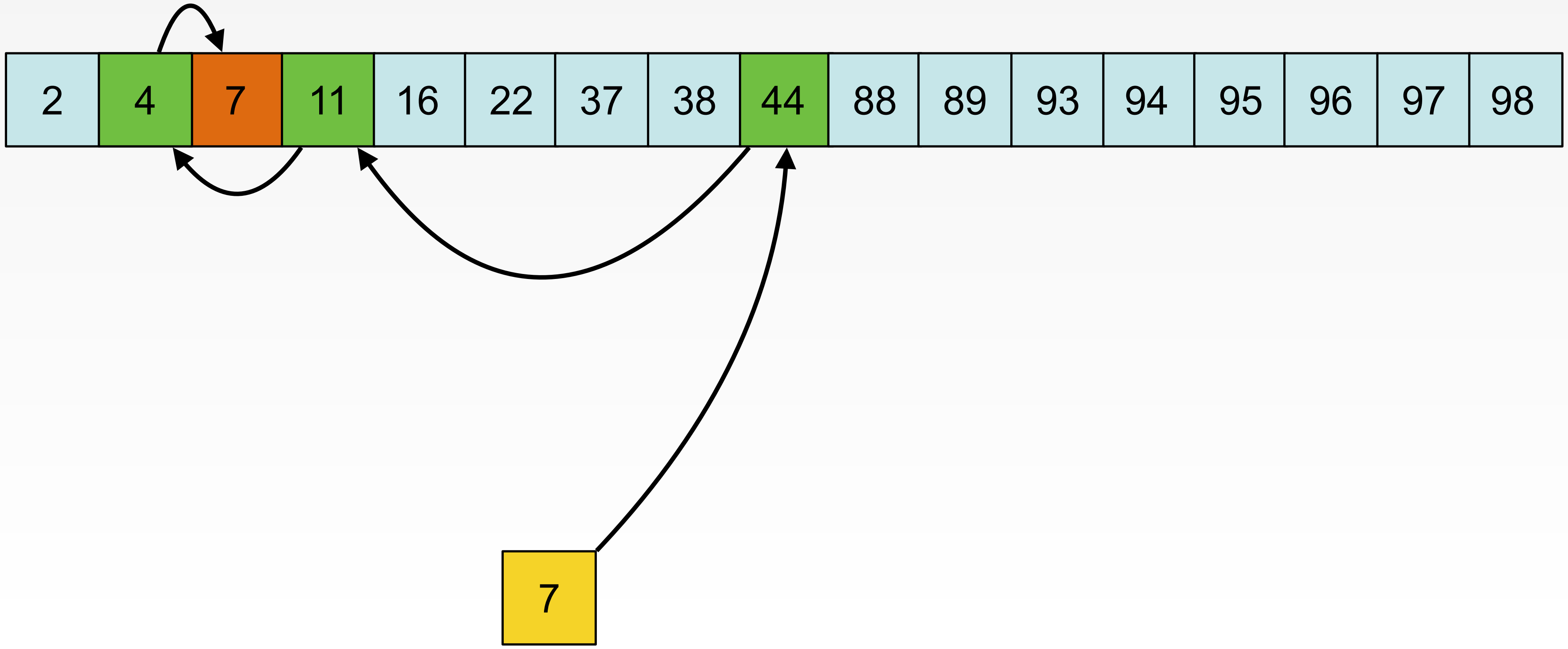
2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

7

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



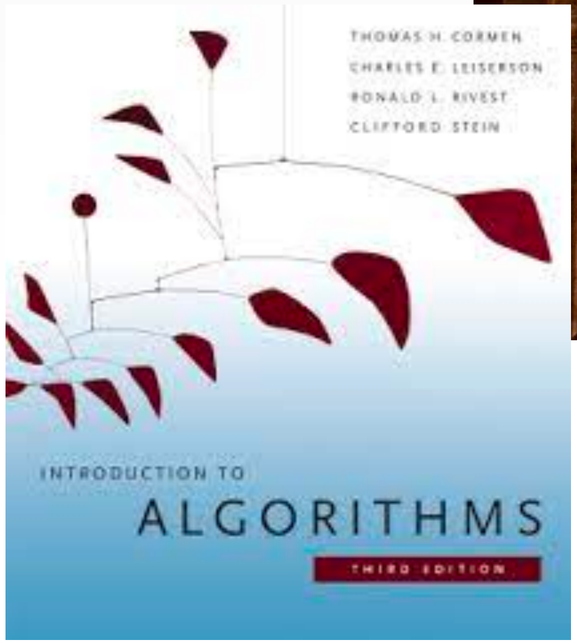
– Look up time: $O(\log n)$

Slide by Sergei Vassilvitskii

Finding a book in the library…

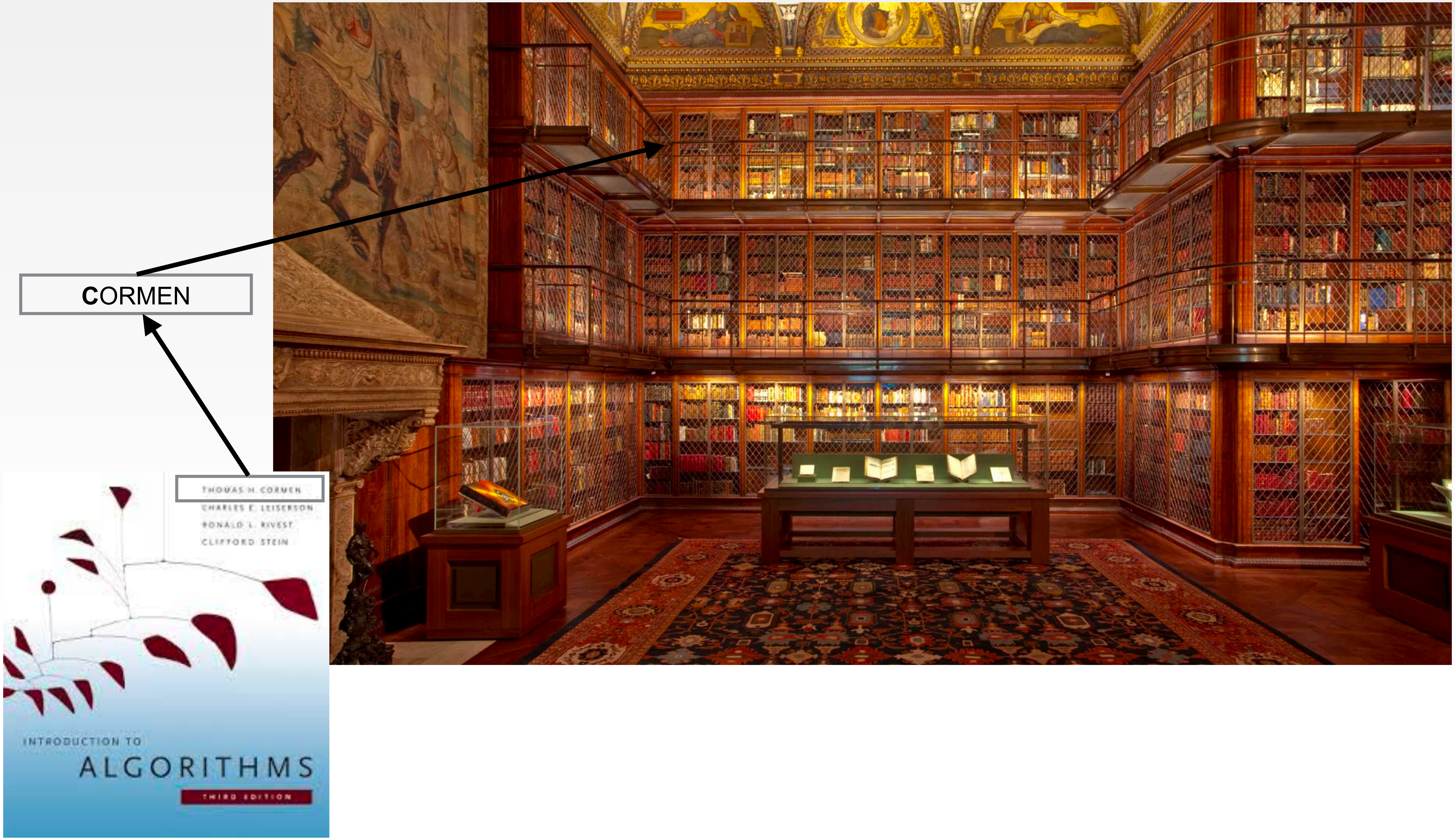
Slide by Sergei Vassilvitskii

Finding a book in the library…



Slide by Sergei Vassilvitskii

Finding a book in the library…

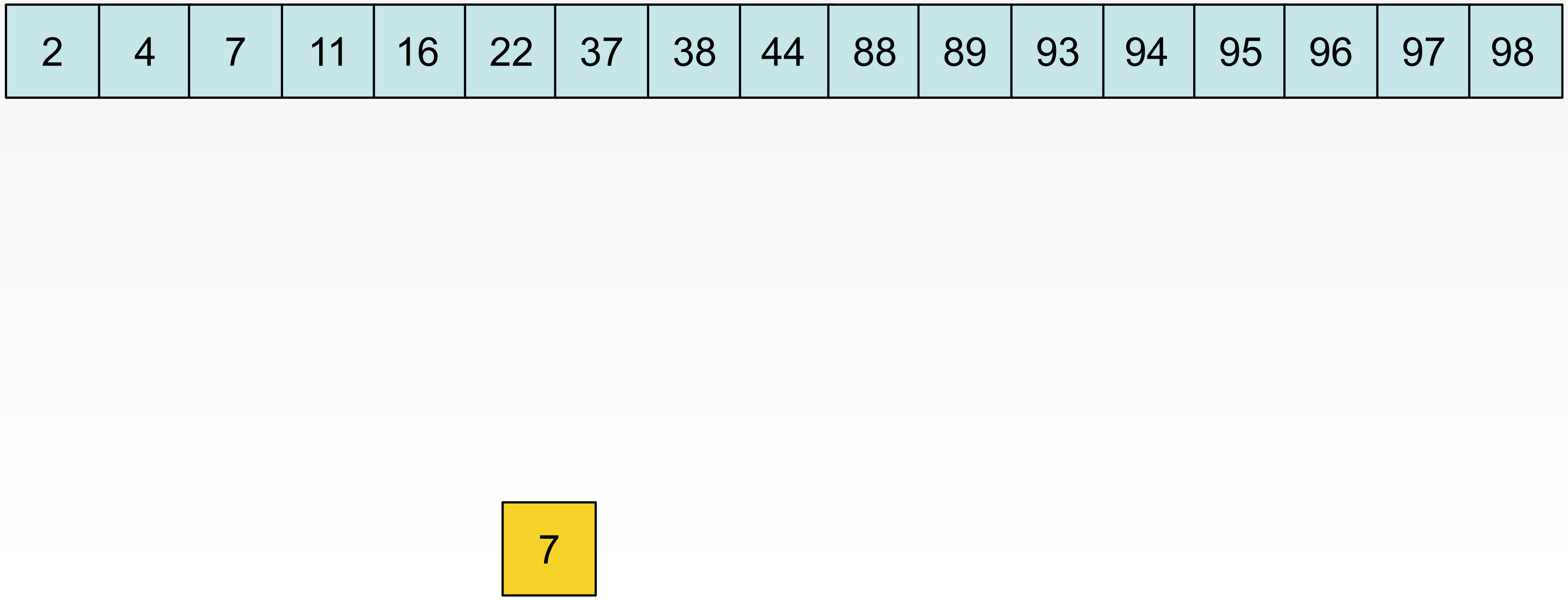


CORMEN

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

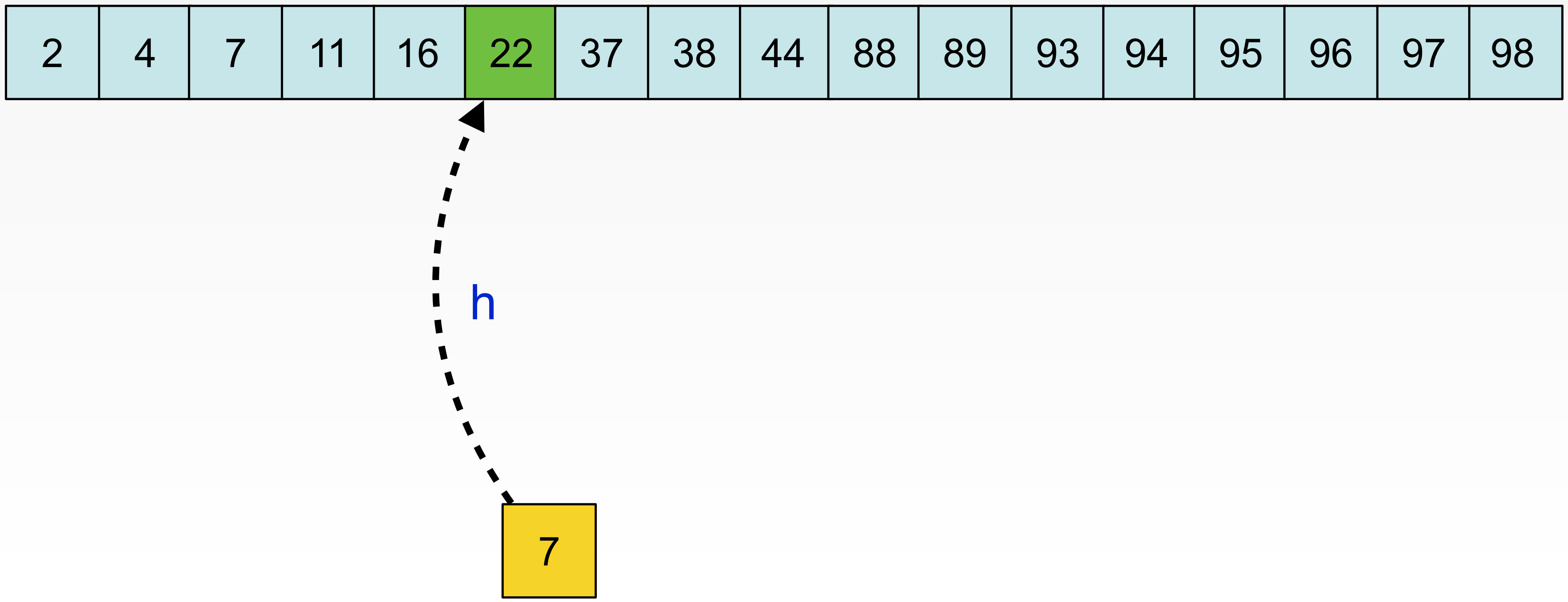
7

- Train a predictor h to learn where q should appear. [Kraska et al.’18]
- Then proceed via doubling binary search

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.

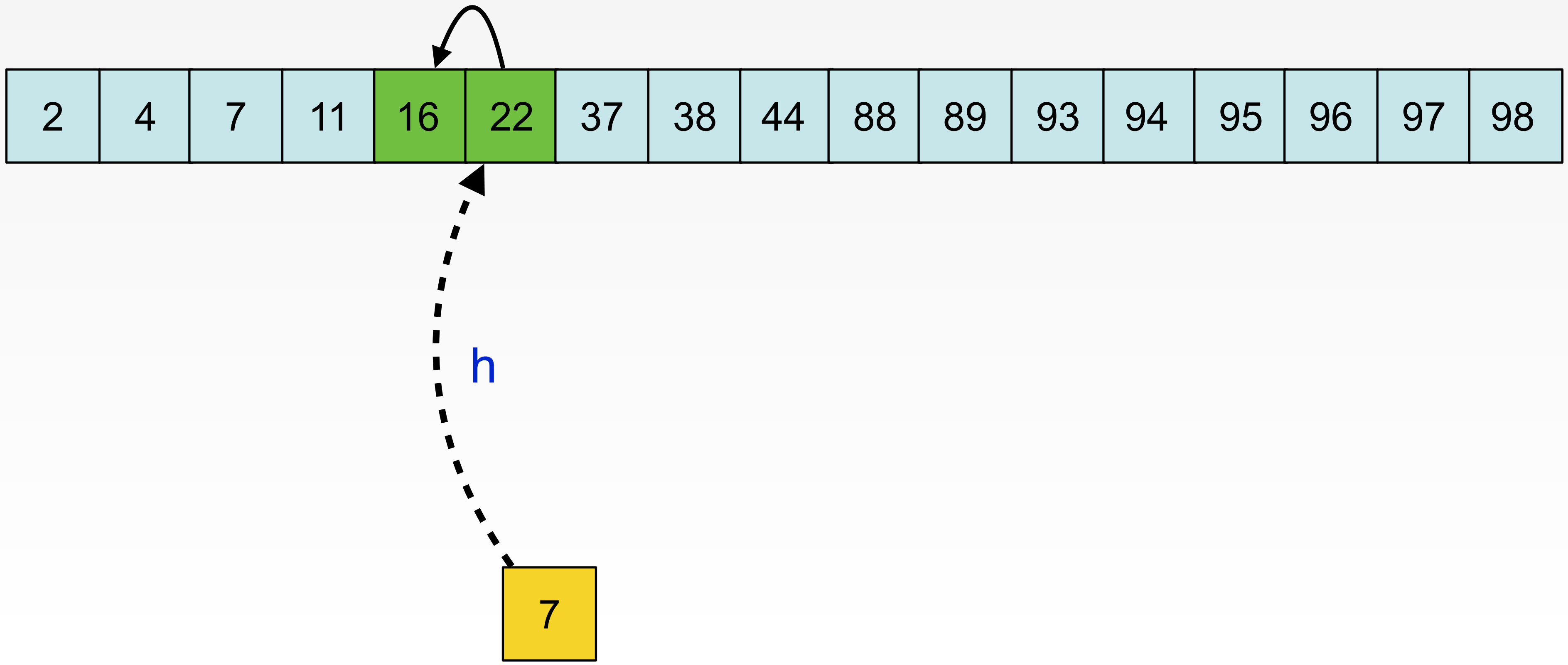


- Train a predictor h to learn where q should appear. [Kraska et al.’18]
- Then proceed via doubling binary search

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

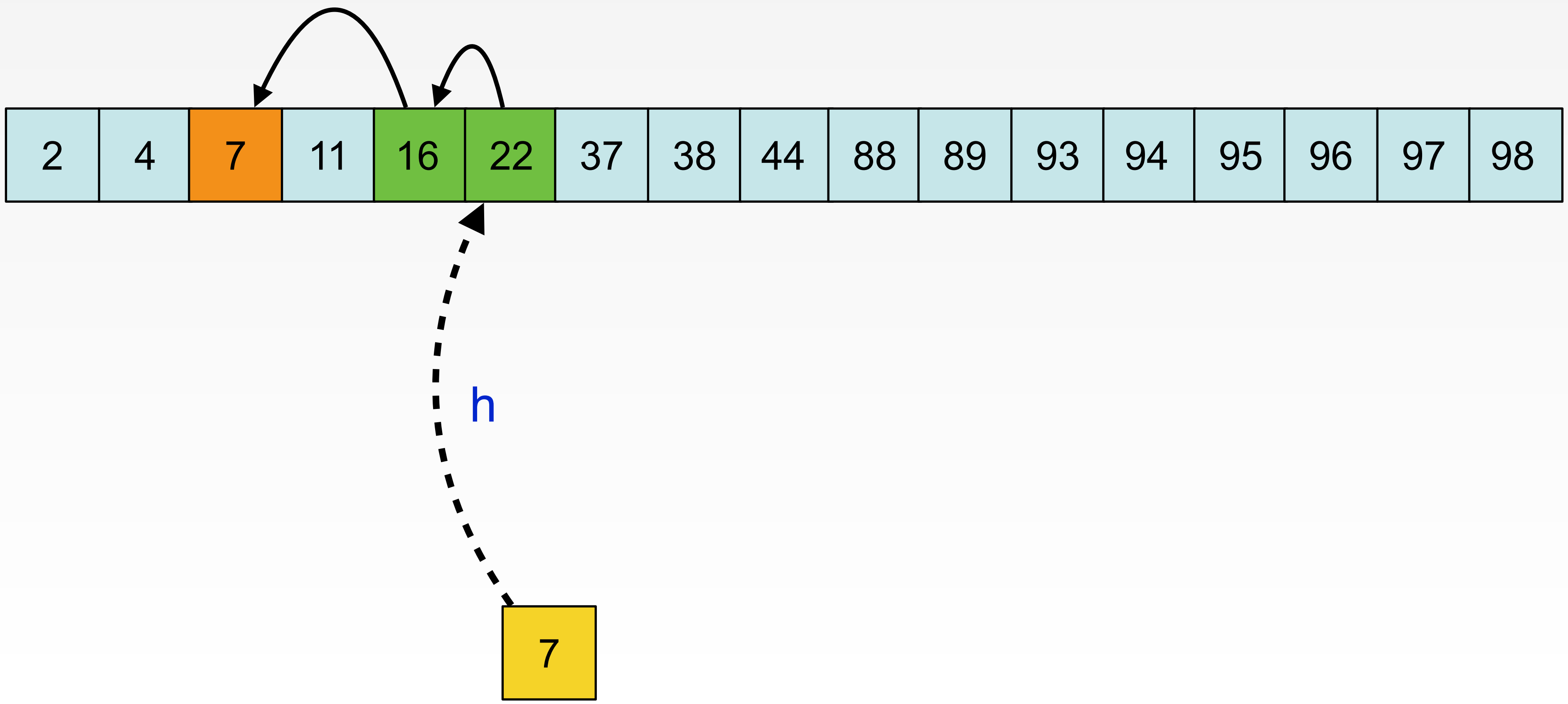
– Train a predictor h to learn where q should appear. [Kraska et al.’18]

– Then proceed via doubling binary search

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



2	4	7	11	16	22	37	38	44	88	89	93	94	95	96	97	98

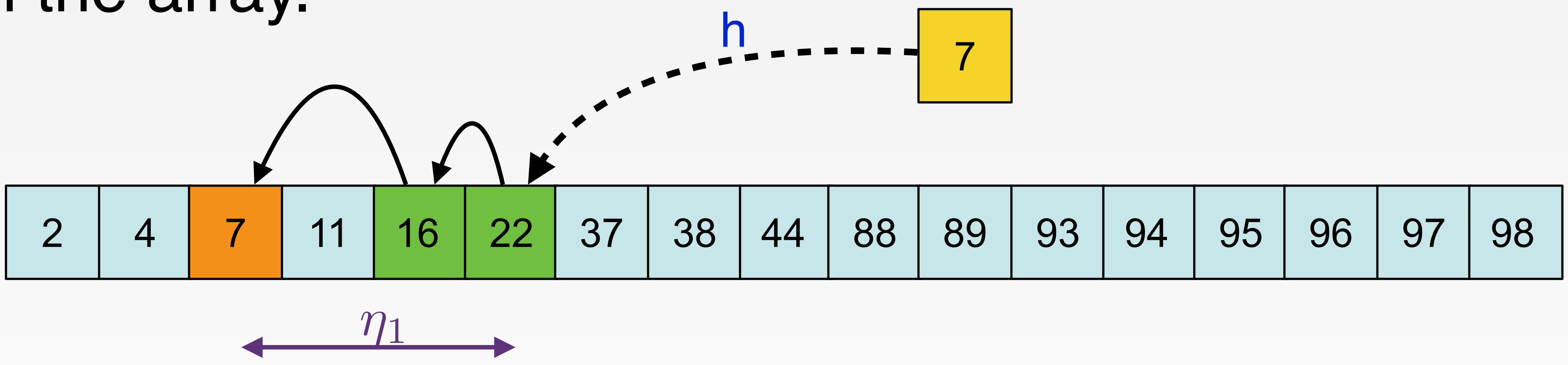
– Train a predictor h to learn where q should appear. [Kraska et al.’18]

– Then proceed via doubling binary search

Slide by Sergei Vassilvitskii

Motivating Example

Given a sorted array of integers A[1…n], and a query q check if q is in the array.



Analysis:

- Let $\eta_1 = |h(q) - \mathrm{OPT}(q)|$ be the absolute error of the predicted position
- Running time: $O(\log \eta_1)$
  - Can be made practical (must worry about speed & accuracy of predictions)

Slide by Sergei Vassilvitskii

More on the analysis

Comparing

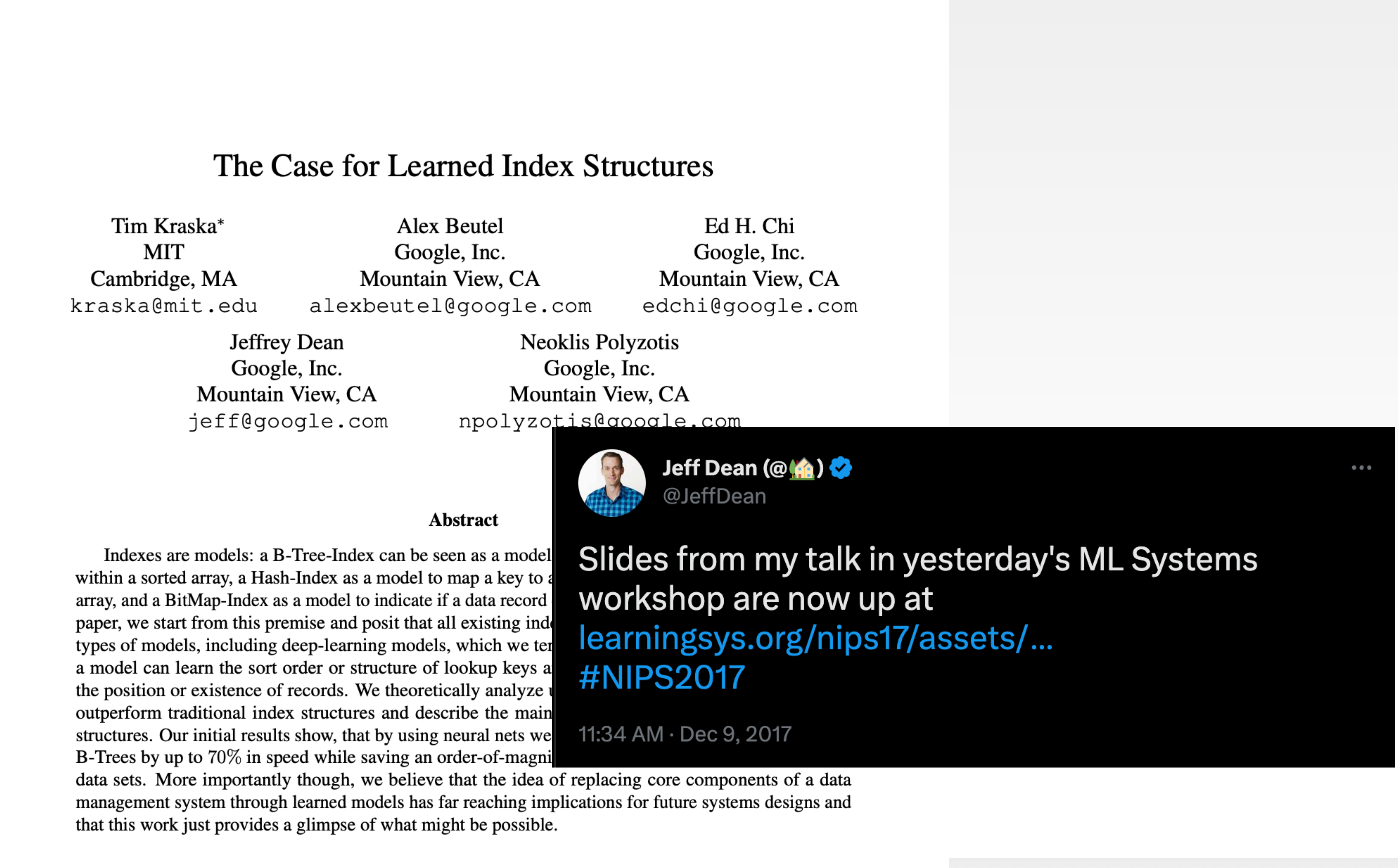
- Classical: $O(\log n)$
- Learning augmented: $O(\log \eta_1)$

Results:

- Consistent: perfect predictions recover optimal (constant) lookup times.
- Robust: even if predictions are bad, not (much) worse than classical

Slide by Sergei Vassilvitskii

How it started…



The Case for Learned Index Structures

Tim Kraska*
MIT
Cambridge, MA
kraska@mit.edu

Alex Beutel
Google, Inc.
Mountain View, CA
alexbeutel@google.com

Ed H. Chi
Google, Inc.
Mountain View, CA
edchi@google.com

Jeffrey Dean
Google, Inc.
Mountain View, CA
jeff@google.com

Neoklis Polyzotis
Google, Inc.
Mountain View, CA
npolyzotis@google.com

Abstract

Indexes are models: a B-Tree-Index can be seen as a model
within a sorted array, a Hash-Index as a model to map a key to a
array, and a BitMap-Index as a model to indicate if a data record
paper, we start from this premise and posit that all existing ind
types of models, including deep-learning models, which we ter
a model can learn the sort order or structure of lookup keys a
the position or existence of records. We theoretically analyze
outperform traditional index structures and describe the main
structures. Our initial results show, that by using neural nets we
B-Trees by up to 70% in speed while saving an order-of-magni
data sets. More importantly though, we believe that the idea of replacing core components of a data management system through learned models has far reaching implications for future systems designs and that this work just provides a glimpse of what might be possible.

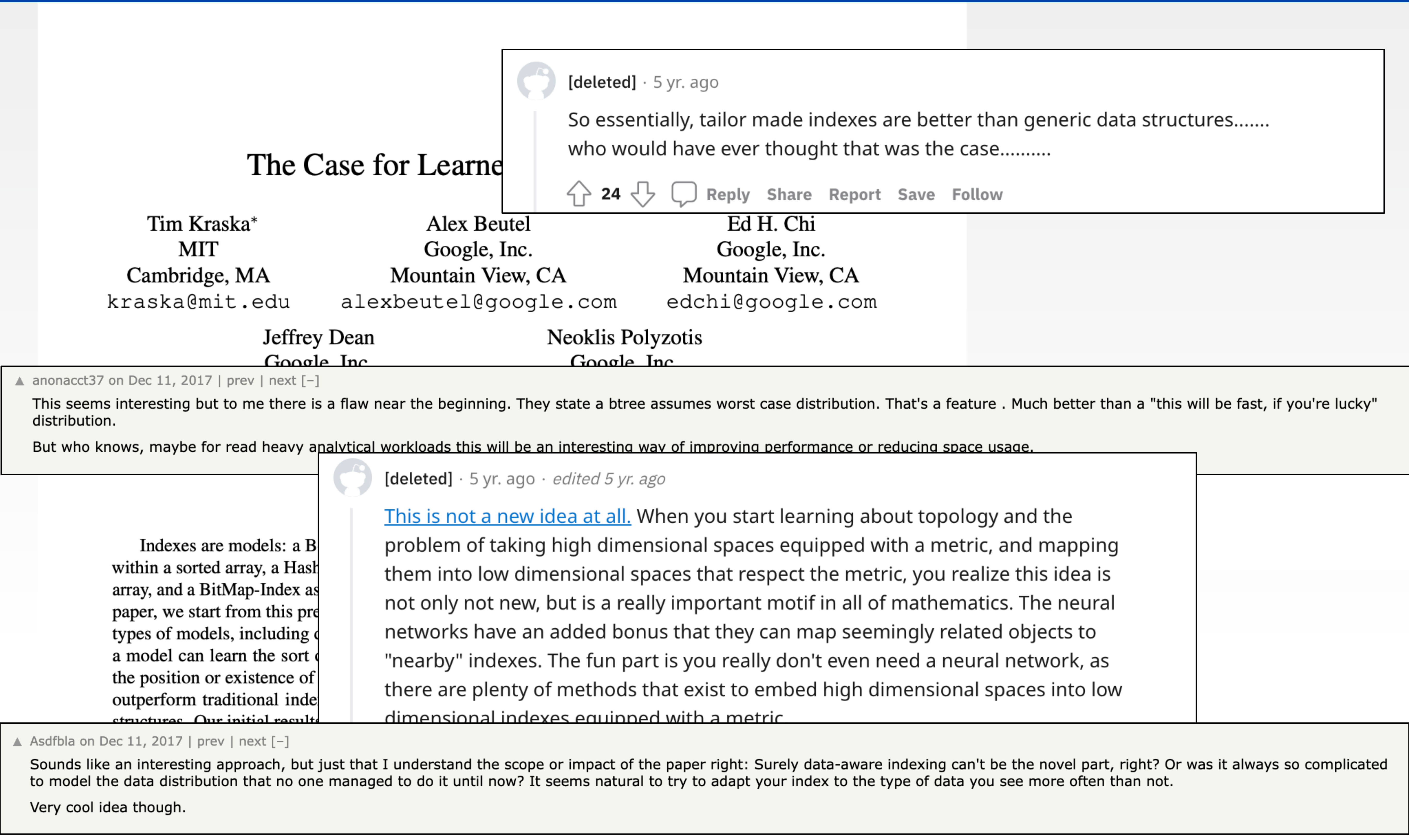
Jeff Dean (@🏡) ✓
@JeffDean

Slides from my talk in yesterday's ML Systems workshop are now up at learningsys.org/nips17/assets/… #NIPS2017

11:34 AM · Dec 9, 2017

Slide by Sergei Vassilvitskii

An inauspicious start..



The Case for Learne

Tim Kraska*
MIT
Cambridge, MA
kraska@mit.edu

Alex Beutel
Google, Inc.
Mountain View, CA
alexbeutel@google.com

Ed H. Chi
Google, Inc.
Mountain View, CA
edchi@google.com

Jeffrey Dean
Google, Inc.

Neoklis Polyzotis
Google, Inc.

Indexes are models: a B
within a sorted array, a Hash
array, and a BitMap-Index as
paper, we start from this pre
types of models, including
a model can learn the sort
the position or existence of
outperform traditional inde
structures. Our initial result

[deleted] · 5 yr. ago

So essentially, tailor made indexes are better than generic data structures.......
who would have ever thought that was the case..........

▲ anonacct37 on Dec 11, 2017 | prev | next [–]

This seems interesting but to me there is a flaw near the beginning. They state a btree assumes worst case distribution. That's a feature . Much better than a "this will be fast, if you're lucky" distribution.

But who knows, maybe for read heavy analytical workloads this will be an interesting way of improving performance or reducing space usage.

[deleted] · 5 yr. ago · *edited 5 yr. ago*

This is not a new idea at all. When you start learning about topology and the problem of taking high dimensional spaces equipped with a metric, and mapping them into low dimensional spaces that respect the metric, you realize this idea is not only not new, but is a really important motif in all of mathematics. The neural networks have an added bonus that they can map seemingly related objects to "nearby" indexes. The fun part is you really don't even need a neural network, as there are plenty of methods that exist to embed high dimensional spaces into low dimensional indexes equipped with a metric.

▲ Asdfbla on Dec 11, 2017 | prev | next [–]

Sounds like an interesting approach, but just that I understand the scope or impact of the paper right: Surely data-aware indexing can't be the novel part, right? Or was it always so complicated to model the data distribution that no one managed to do it until now? It seems natural to try to adapt your index to the type of data you see more often than not.

Very cool idea though.

Slide by Sergei Vassilvitskii

More on the analysis

Comparing

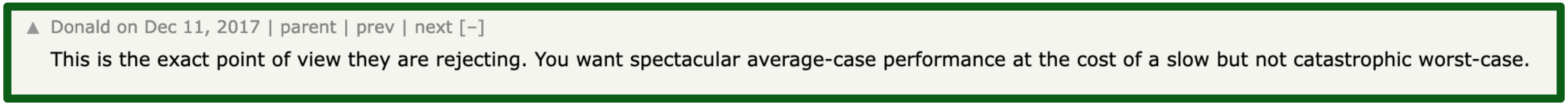
– Classical: $O(\log n)$

– Learning augmented: $O(\log \eta_1)$

Results:

– Consistent: perfect predictions recover optimal (constant) lookup times.

– Robust: even if predictions are bad, not (much) worse than classical

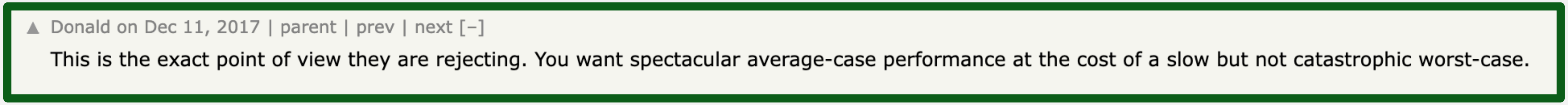


▲ Donald on Dec 11, 2017 | parent | prev | next [–]

This is the exact point of view they are rejecting. You want spectacular average-case performance at the cost of a slow but not catastrophic worst-case.

Slide by Sergei Vassilvitskii

Algorithms with Predictions



▲ Donald on Dec 11, 2017 | parent | prev | next [–]

This is the exact point of view they are rejecting. You want spectacular average-case performance at the cost of a slow but not catastrophic worst-case.

This is the premise of “Algorithms with Predictions”

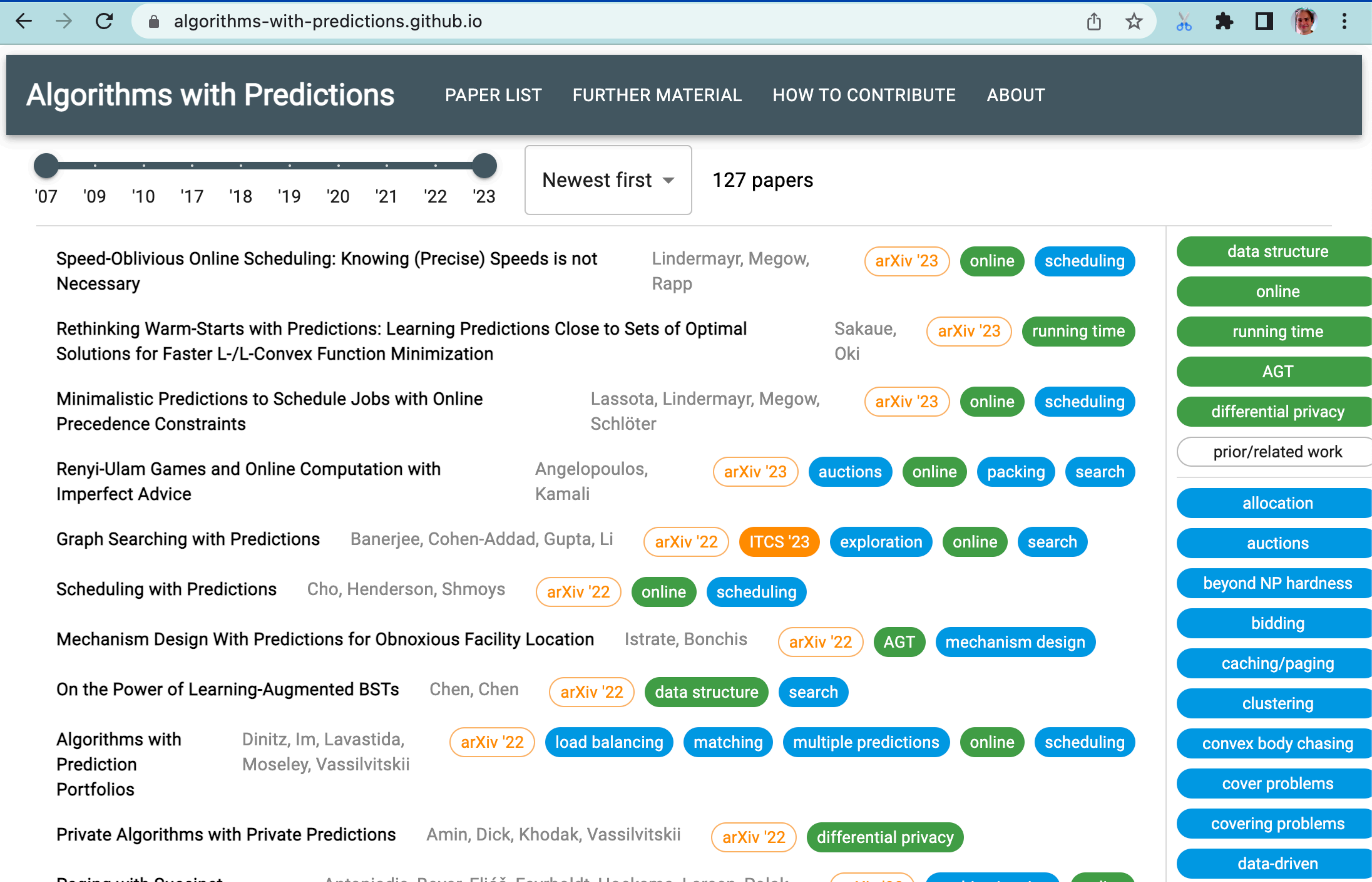
- Aka ‘Learning Augmented Algorithms’

Today:

- Over 100 interesting papers. Hard to keep up!
- See https://algorithms-with-predictions.github.io/
- No way to do justice to all the papers, or all the ideas, or all the authors…

Slide by Sergei Vassilvitskii

How it’s going…



algorithms-with-predictions.github.io

Algorithms with Predictions

PAPER LIST   FURTHER MATERIAL   HOW TO CONTRIBUTE   ABOUT

'07 '09 '10 '17 '18 '19 '20 '21 '22 '23

Newest first ▾

127 papers

Speed-Oblivious Online Scheduling: Knowing (Precise) Speeds is not Necessary	Lindermayr, Megow, Rapp	arXiv '23 online scheduling
Rethinking Warm-Starts with Predictions: Learning Predictions Close to Sets of Optimal Solutions for Faster L-/L-Convex Function Minimization	Sakaue, Oki	arXiv '23 running time
Minimalistic Predictions to Schedule Jobs with Online Precedence Constraints	Lassota, Lindermayr, Megow, Schlöter	arXiv '23 online scheduling
Renyi-Ulam Games and Online Computation with Imperfect Advice	Angelopoulos, Kamali	arXiv '23 auctions online packing search
Graph Searching with Predictions	Banerjee, Cohen-Addad, Gupta, Li	arXiv '22 ITCS '23 exploration online search
Scheduling with Predictions	Cho, Henderson, Shmoys	arXiv '22 online scheduling
Mechanism Design With Predictions for Obnoxious Facility Location	Istrate, Bonchis	arXiv '22 AGT mechanism design
On the Power of Learning-Augmented BSTs	Chen, Chen	arXiv '22 data structure search
Algorithms with Prediction Portfolios	Dinitz, Im, Lavastida, Moseley, Vassilvitskii	arXiv '22 load balancing matching multiple predictions online scheduling
Private Algorithms with Private Predictions	Amin, Dick, Khodak, Vassilvitskii	arXiv '22 differential privacy
Paging with Succinct	Antoniadis, Boyar, Eliáš, Favrholdt, Hoeksma, Larsen, Polak	

Learning-Augmented Online Algorithms and the Primal-Dual Method

Ola Svensson

Joint work with Etienne Bamas and Andreas Maggiori

EPFL

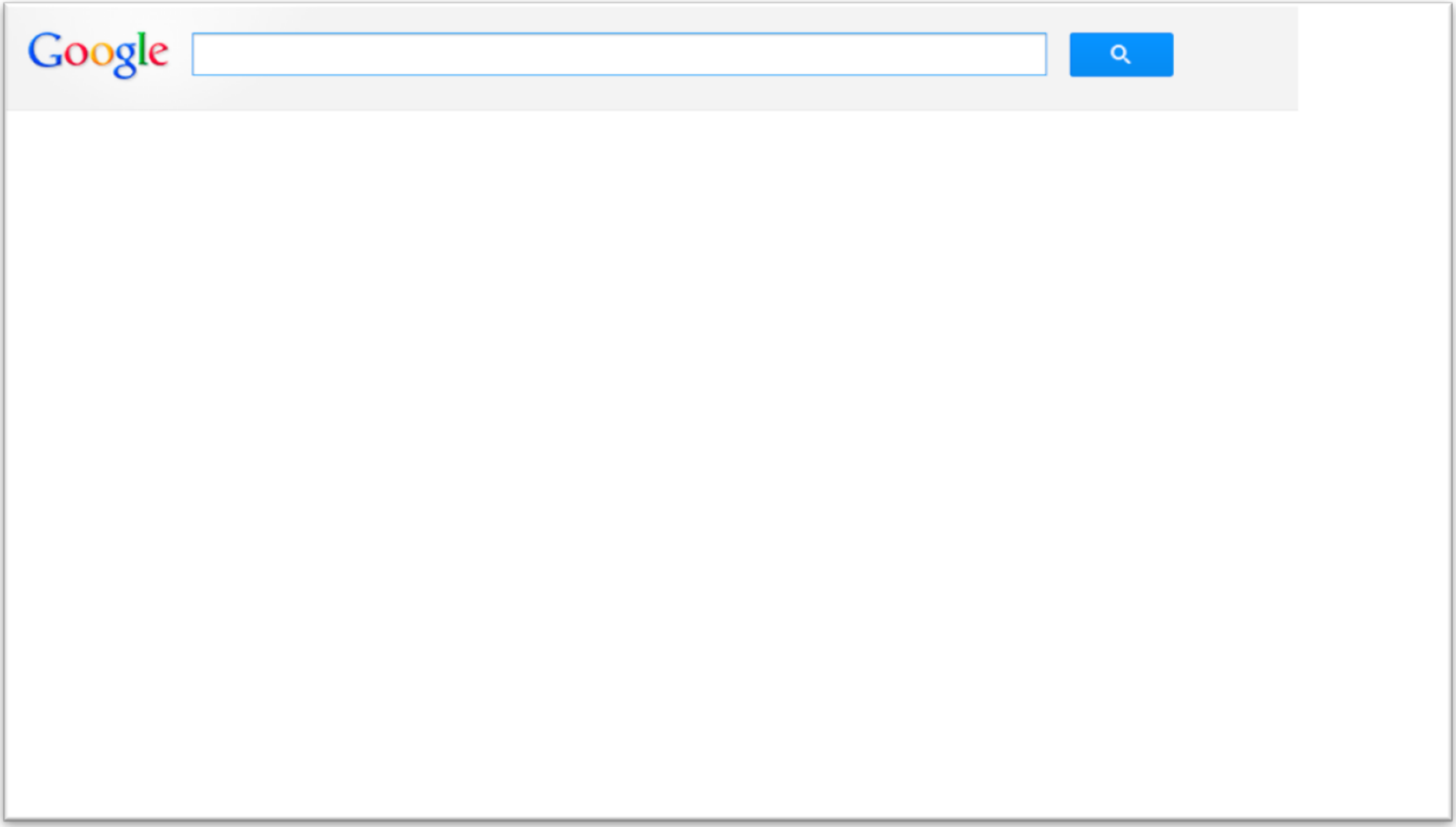
Outline

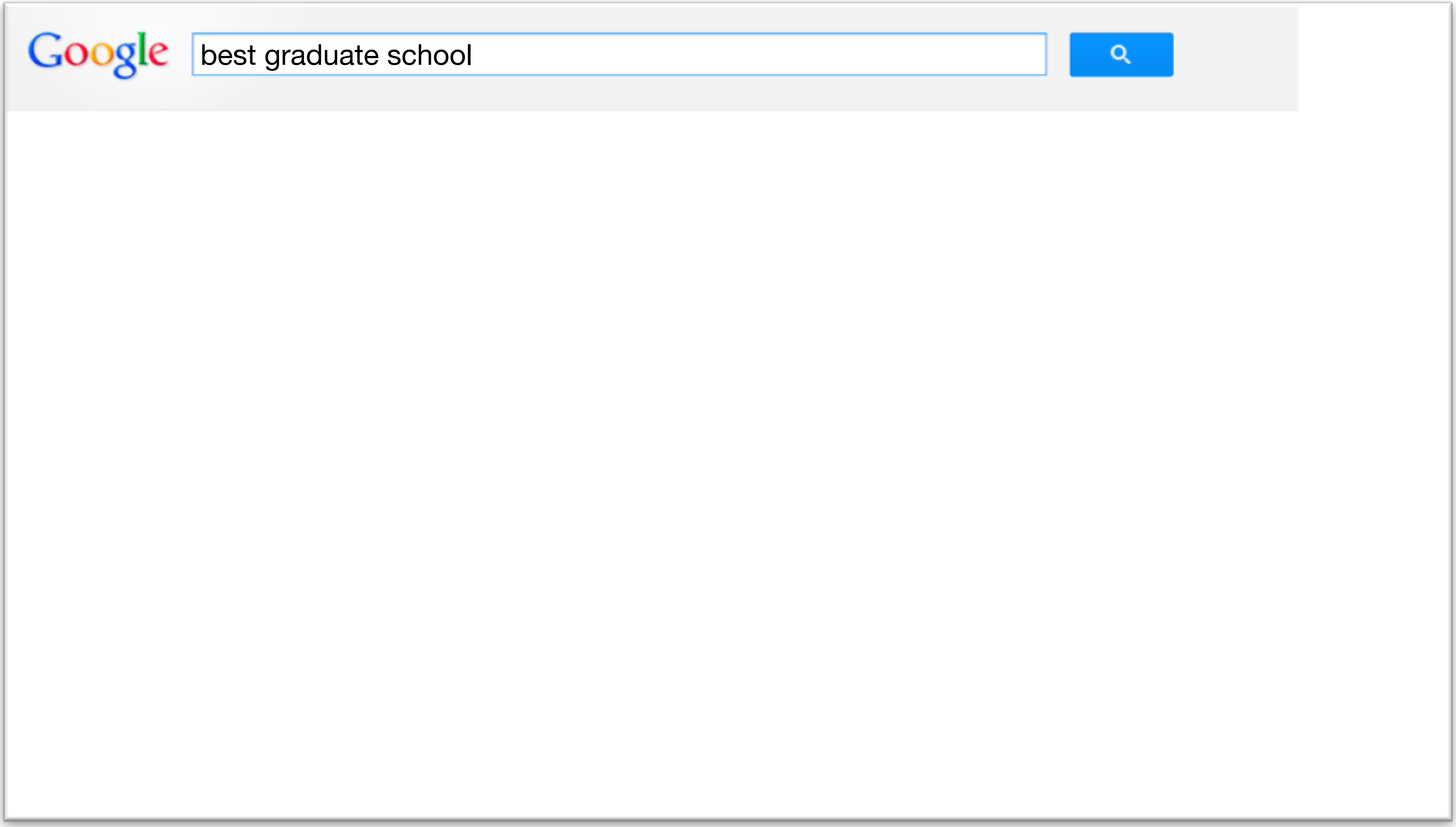
- Learning-augmented online algorithms
- Case study: set cover
- Instantiating PDLA for other problems
- Future directions

Outline

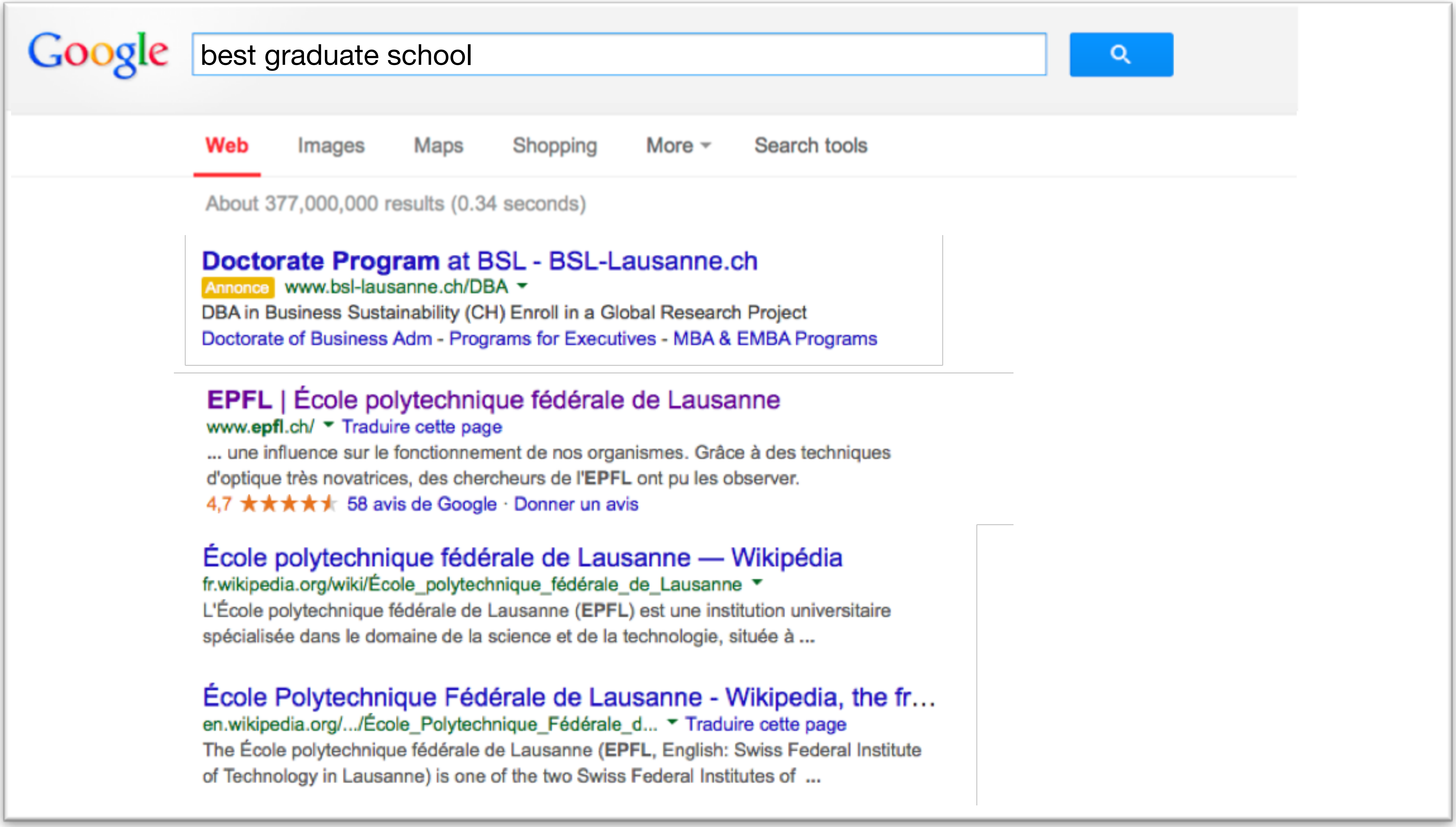
- **Learning-augmented online algorithms**
- Case study: set cover
- Instantiating PDLA for other problems
- Future directions

Online algorithms


Google



best graduate school



best graduate school

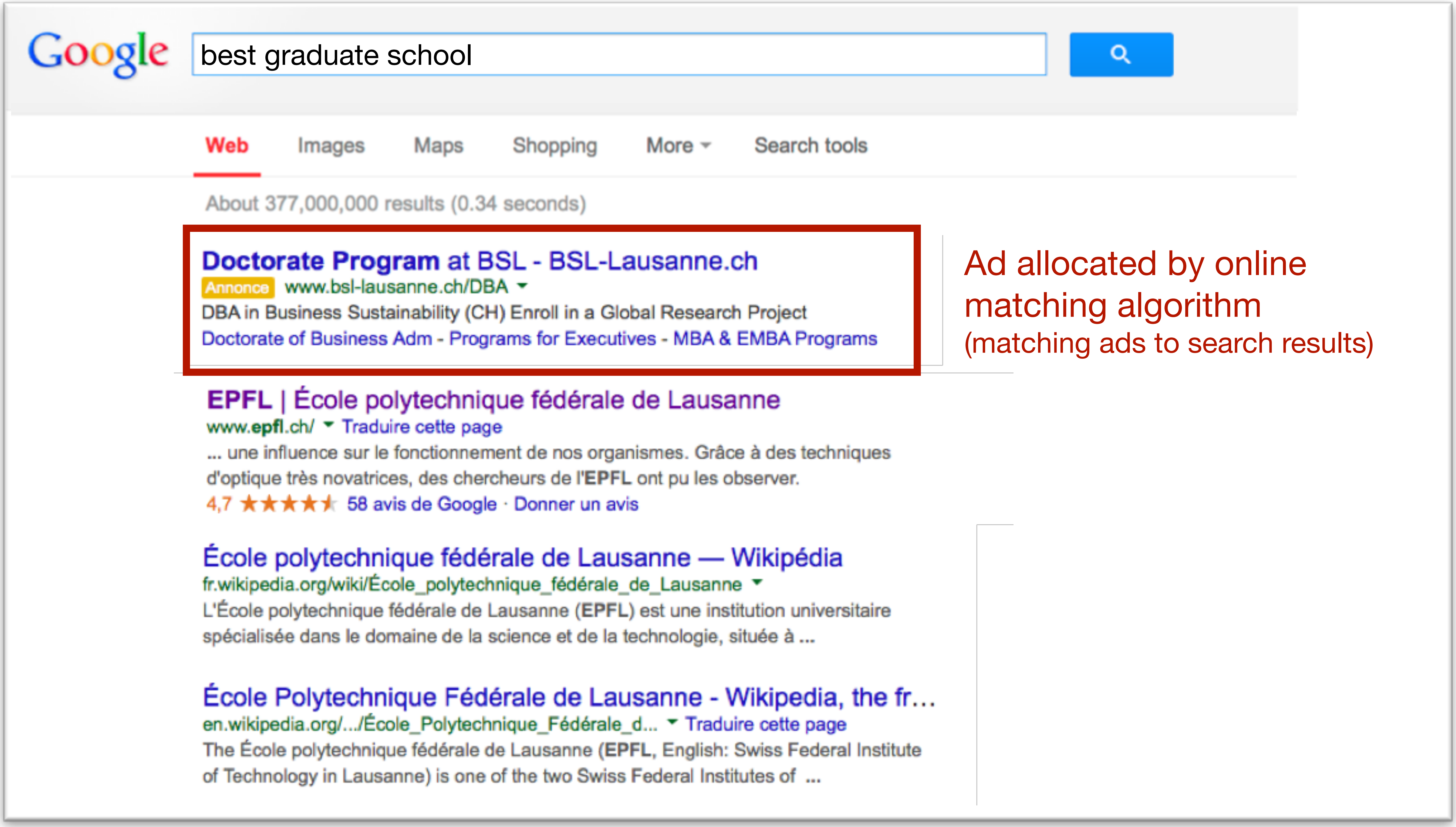
About 377,000,000 results (0.34 seconds)

Doctorate Program at BSL - BSL-Lausanne.ch
Annonce www.bsl-lausanne.ch/DBA
DBA in Business Sustainability (CH) Enroll in a Global Research Project
Doctorate of Business Adm - Programs for Executives - MBA & EMBA Programs

EPFL | École polytechnique fédérale de Lausanne
www.**epfl**.ch/ Traduire cette page
... une influence sur le fonctionnement de nos organismes. Grâce à des techniques d'optique très novatrices, des chercheurs de l'**EPFL** ont pu les observer.
4,7 ★★★★★ 58 avis de Google · Donner un avis

École polytechnique fédérale de Lausanne — Wikipédia
fr.wikipedia.org/wiki/École_polytechnique_fédérale_de_Lausanne
L'École polytechnique fédérale de Lausanne (**EPFL**) est une institution universitaire spécialisée dans le domaine de la science et de la technologie, située à ...

École Polytechnique Fédérale de Lausanne - Wikipedia, the fr...
en.wikipedia.org/.../École_Polytechnique_Fédérale_d... Traduire cette page
The École polytechnique fédérale de Lausanne (**EPFL**, English: Swiss Federal Institute of Technology in Lausanne) is one of the two Swiss Federal Institutes of ...



best graduate school

About 377,000,000 results (0.34 seconds)

Doctorate Program at BSL - BSL-Lausanne.ch
Annonce www.bsl-lausanne.ch/DBA
DBA in Business Sustainability (CH) Enroll in a Global Research Project
Doctorate of Business Adm - Programs for Executives - MBA & EMBA Programs

EPFL | École polytechnique fédérale de Lausanne
www.epfl.ch/ Traduire cette page
... une influence sur le fonctionnement de nos organismes. Grâce à des techniques d'optique très novatrices, des chercheurs de l'**EPFL** ont pu les observer.
4,7 ★★★★★ 58 avis de Google · Donner un avis

École polytechnique fédérale de Lausanne — Wikipédia
fr.wikipedia.org/wiki/École_polytechnique_fédérale_de_Lausanne
L'École polytechnique fédérale de Lausanne (**EPFL**) est une institution universitaire spécialisée dans le domaine de la science et de la technologie, située à ...

École Polytechnique Fédérale de Lausanne - Wikipedia, the fr...
en.wikipedia.org/.../École_Polytechnique_Fédérale_d... Traduire cette page
The École polytechnique fédérale de Lausanne (**EPFL**, English: Swiss Federal Institute of Technology in Lausanne) is one of the two Swiss Federal Institutes of ...

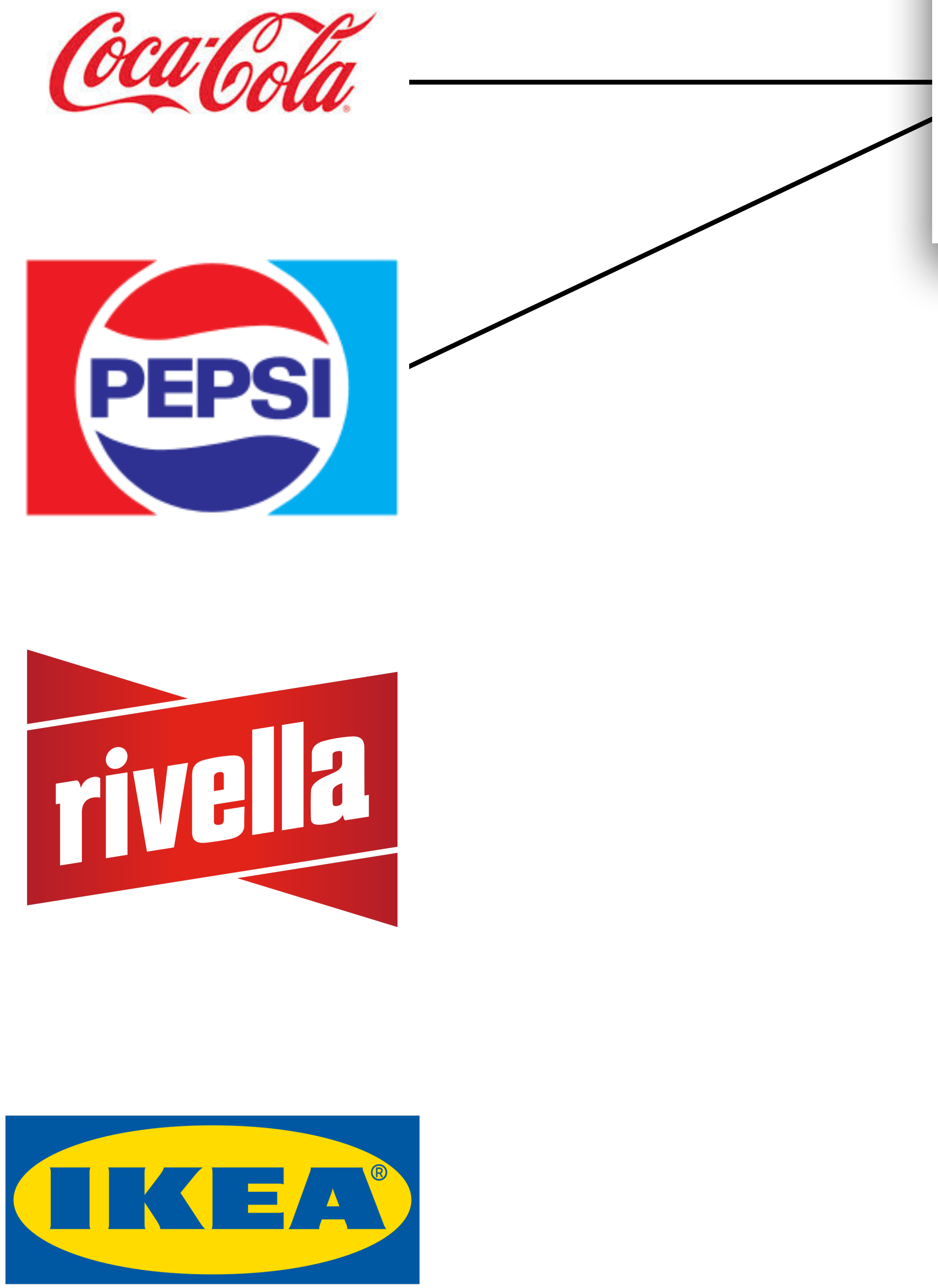
Ad allocated by online matching algorithm
(matching ads to search results)

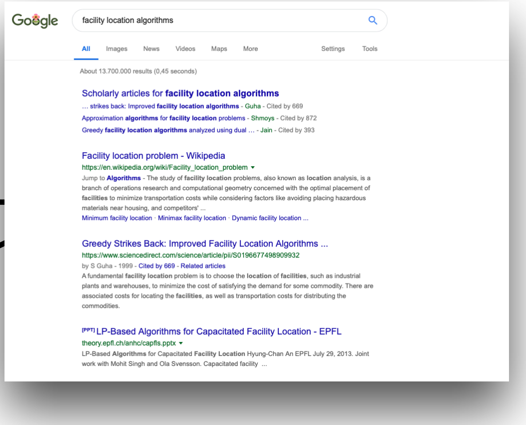
Coca-Cola

PEPSI

rivella

IKEA®





facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

by S Guha - 1999 - Cited by 669 - Related articles

A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

theory.epfl.ch/anhc/capfls.pptx

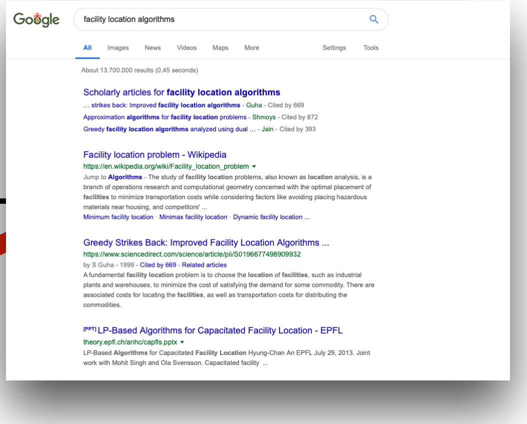
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Coca-Cola

PEPSI

rivella

IKEA®



facility location algorithms

About 13.700.000 results (0,45 seconds)

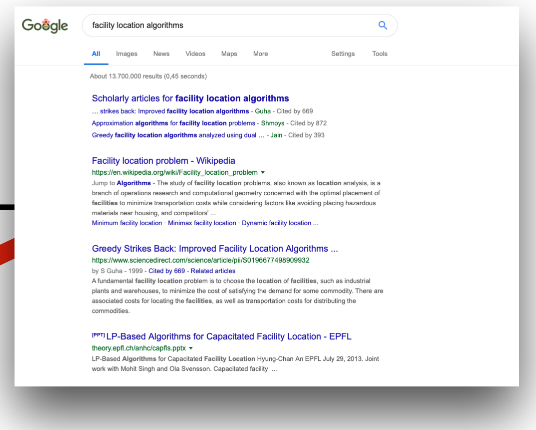
Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Coca-Cola



Google

facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem ▾

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

by S Guha - 1999 - Cited by 669 - Related articles

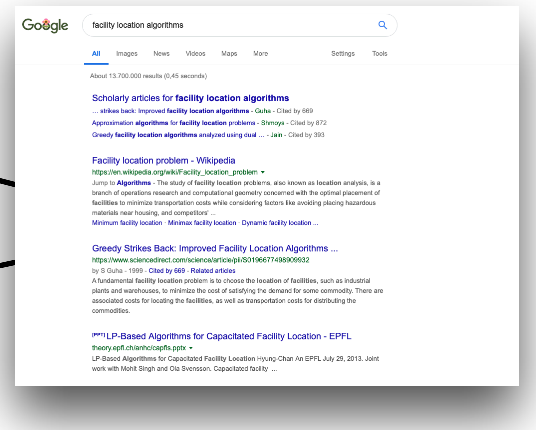
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

theory.epfl.ch/anhc/capfls.pptx ▾

LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

PEPSI



Google

facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem ▾

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

by S Guha - 1999 - Cited by 669 - Related articles

A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

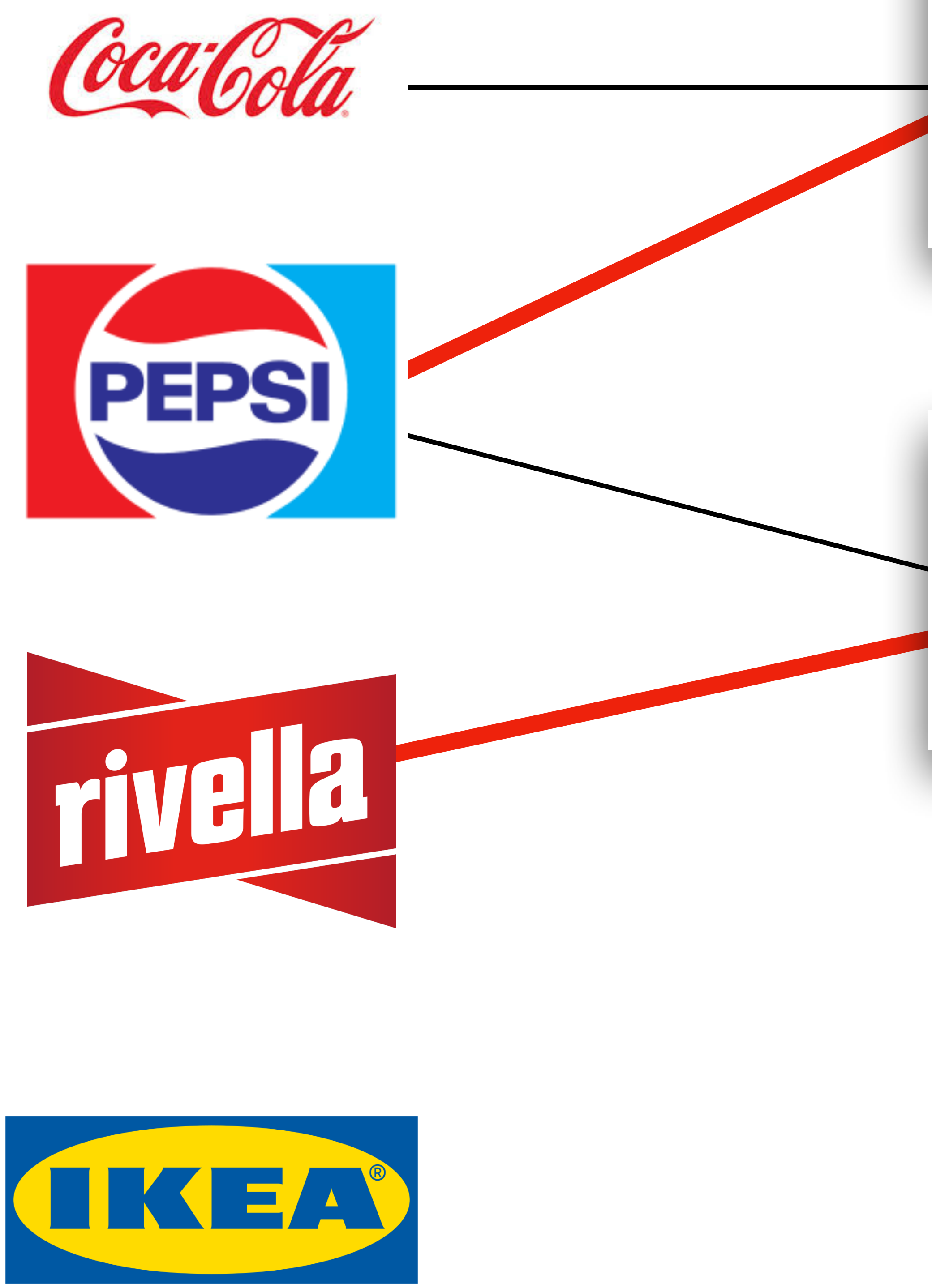
[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

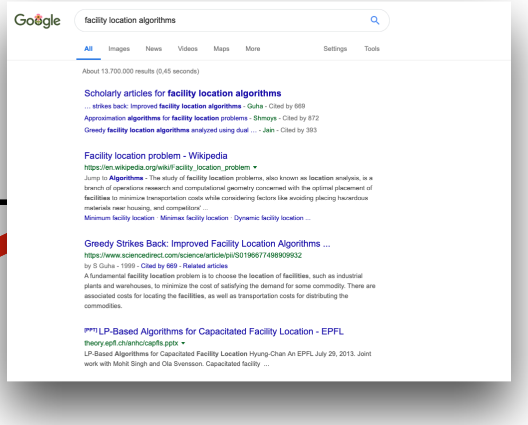
theory.epfl.ch/anhc/capfls.pptx ▾

LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

rivella

IKEA®





facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem ▾

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

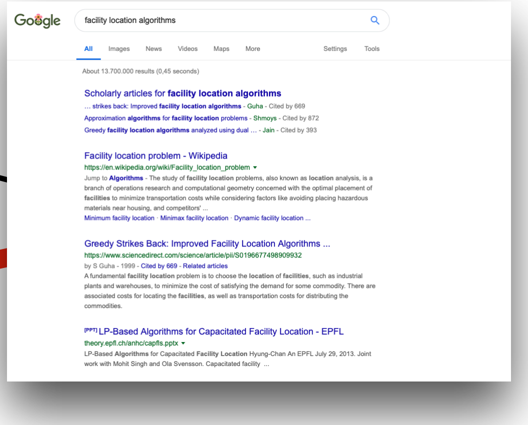
by S Guha - 1999 - Cited by 669 - Related articles

A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

theory.epfl.ch/anhc/capfls.pptx ▾

LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...



facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem ▾

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

by S Guha - 1999 - Cited by 669 - Related articles

A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

theory.epfl.ch/anhc/capfls.pptx ▾

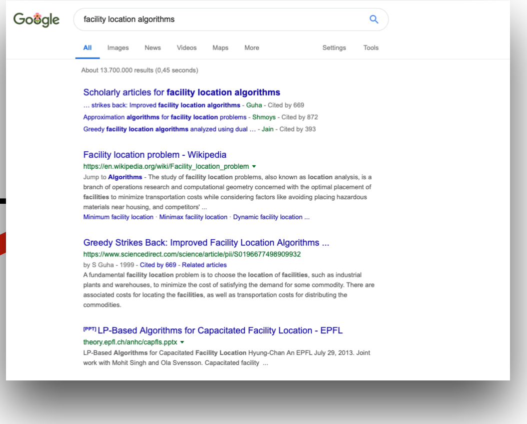
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Coca-Cola

PEPSI

rivella

IKEA



facility location algorithms

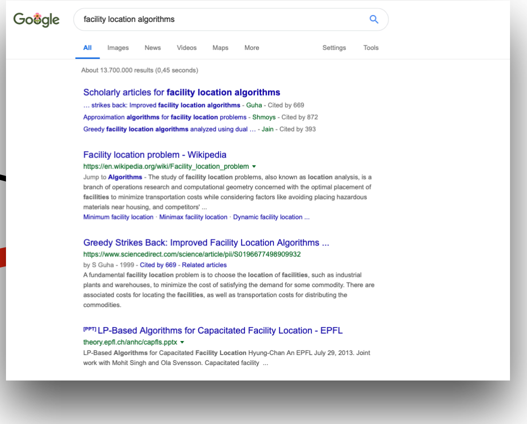
About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...


facility location algorithms

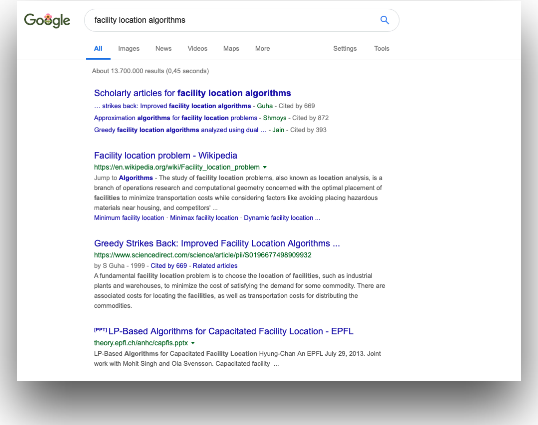
About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...


facility location algorithms

About 13.700.000 results (0,45 seconds)

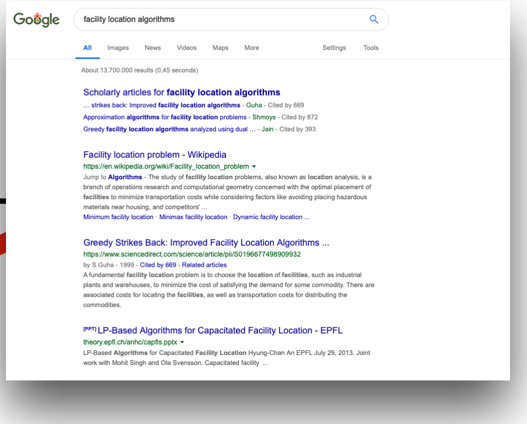
Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Coca-Cola



facility location algorithms

About 13.700.000 results (0,45 seconds)

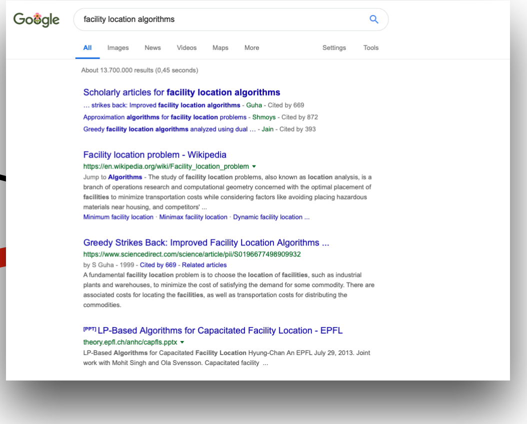
Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

PEPSI



facility location algorithms

About 13.700.000 results (0,45 seconds)

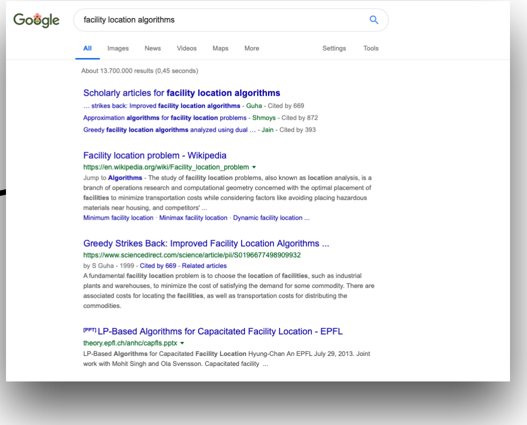
Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

rivella



facility location algorithms

About 13.700.000 results (0,45 seconds)

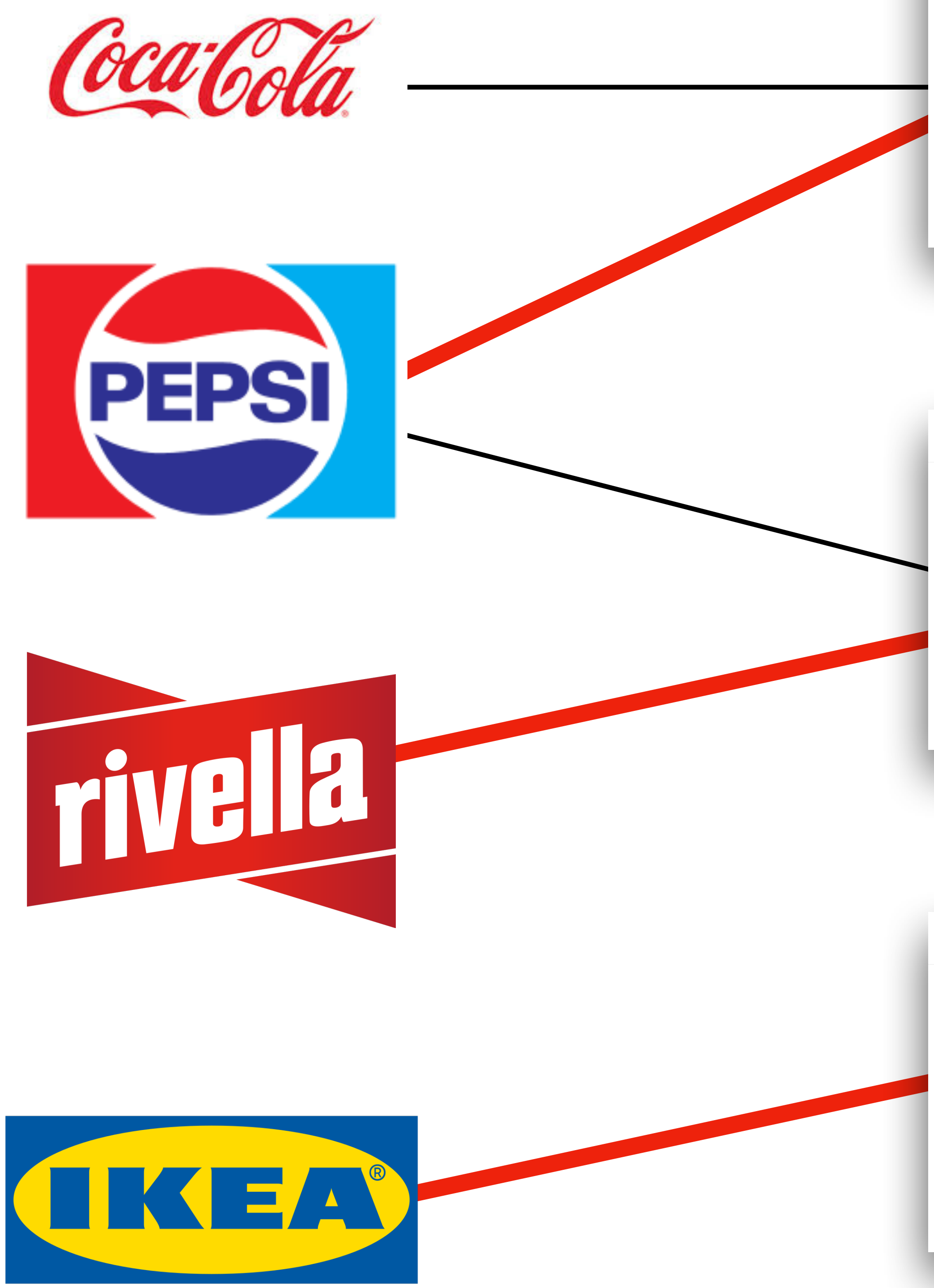
Scholarly articles for **facility location algorithms**
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

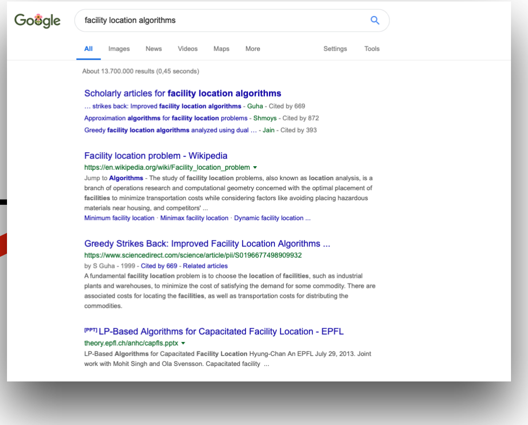
Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

IKEA®




Google

facility location algorithms

All Images News Videos Maps More Settings Tools

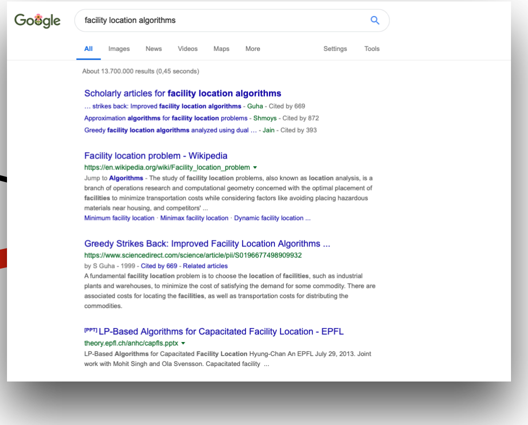
About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...


Google

facility location algorithms

All Images News Videos Maps More Settings Tools

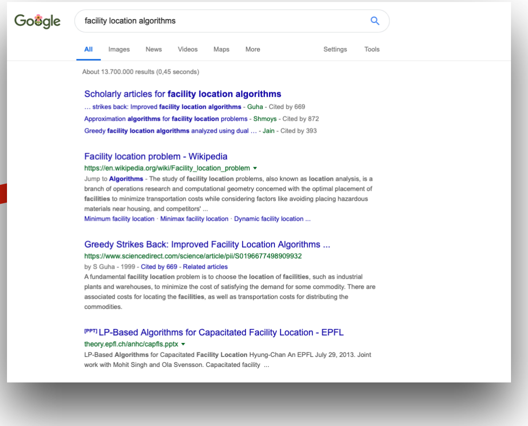
About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...


Google

facility location algorithms

All Images News Videos Maps More Settings Tools

About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Instance
Arrive
Online

Immediate
Decisions

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\text{cost}(\text{solution}) \leq c \cdot \text{OPT} \qquad \text{if minimization}$$

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\le c \cdot \text{OPT} \qquad && \text{if minimization} \\ \text{value(solution)} &\ge c \cdot \text{OPT} \qquad && \text{if maximization} \end{aligned}$$

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost}(\text{solution}) &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value}(\text{solution}) &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**
- The difficulty is that we do not know the total number of days we will be skiing

Strategy:

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost}(\text{solution}) &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value}(\text{solution}) &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

Strategy: Rent for the first B-1 days and buy at the beginning of day B

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\text{cost}(\text{solution}) \leq c \cdot \text{OPT} \qquad \text{if minimization}$$
$$\text{value}(\text{solution}) \geq c \cdot \text{OPT} \qquad \text{if maximization}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**
- The difficulty is that we do not know the total number of days we will be skiing

Strategy: Rent for the first B-1 days and buy at the beginning of day B

- If we ski at most B-1 days, we are optimal

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

Strategy: Rent for the first B-1 days and buy at the beginning of day B

- If we ski at most B-1 days, we are optimal
- If we ski at least B days, we pay 2B-1 whereas OPT pays B

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

Strategy: Rent for the first B-1 days and buy at the beginning of day B

- If we ski at most B-1 days, we are optimal
- If we ski at least B days, we pay 2B-1 whereas OPT pays B
- Strategy is 2-competitive which is optimal for deterministic algorithms. (e/(e-1) is optimal with randomization)

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

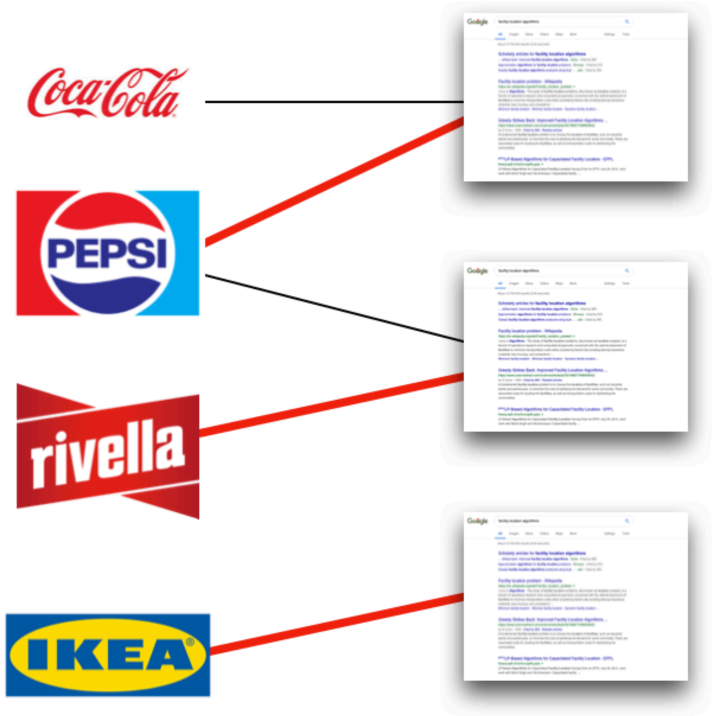
$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} \qquad \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} \qquad \text{if maximization} \end{aligned}$$

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} \qquad && \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} \qquad && \text{if maximization} \end{aligned}$$

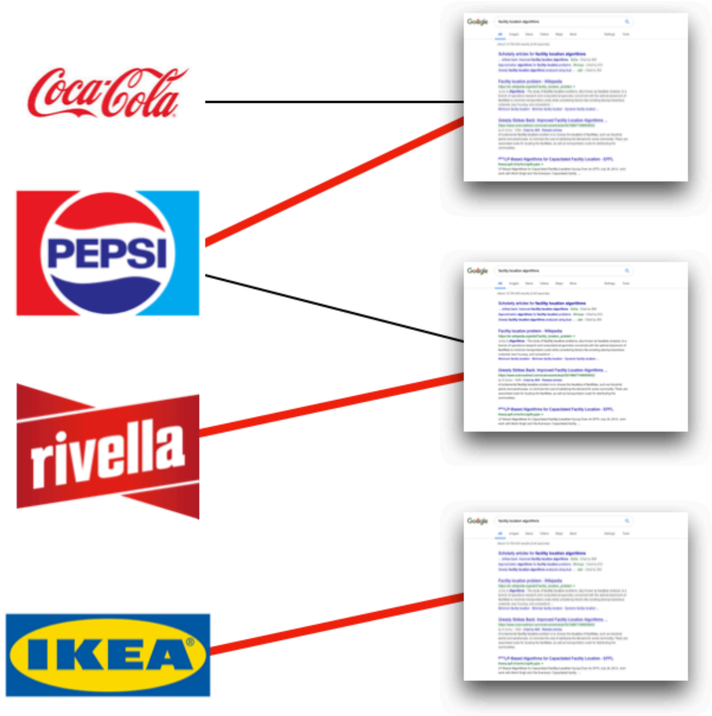


Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost(solution)} &\leq c \cdot \text{OPT} && \text{if minimization} \\ \text{value(solution)} &\geq c \cdot \text{OPT} && \text{if maximization} \end{aligned}$$



Folklore Theorem:

Greedy is 1/2-competitive

This is best possible for *deterministic strategies*

Theorem [KVV'90 + BC08, DJK13…]:

Ranking is (1-1/e)-competitive

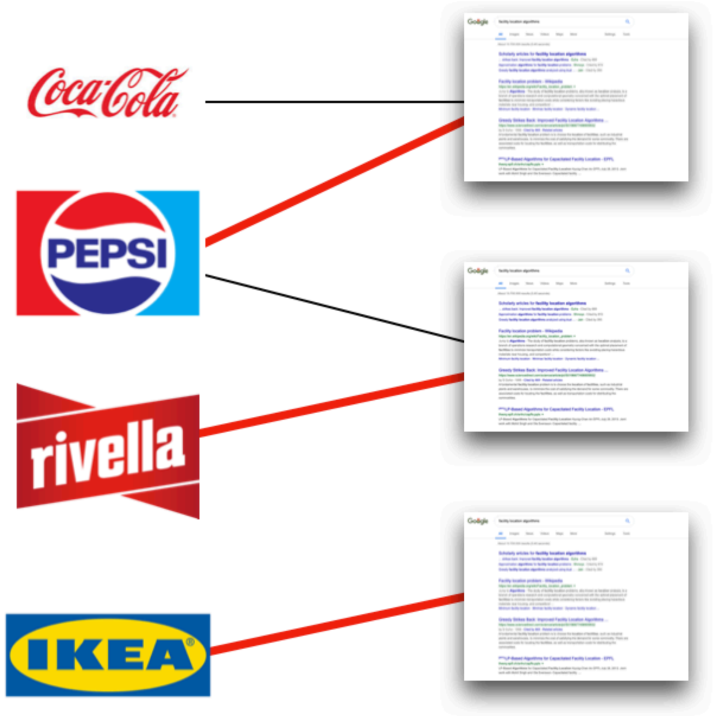
This is best possible for *(randomized) strategies*

Evaluating online algorithms

Competitive ratio

An algorithm is c-competitive if, *for any input sequence*, it finds a solution with

$$\begin{aligned} \text{cost}(\text{solution}) &\leq c \cdot \text{OPT} && \text{if minimization} \\ \text{value}(\text{solution}) &\geq c \cdot \text{OPT} && \text{if maximization} \end{aligned}$$



Folklore Theorem:

Greedy is 1/2-competitive

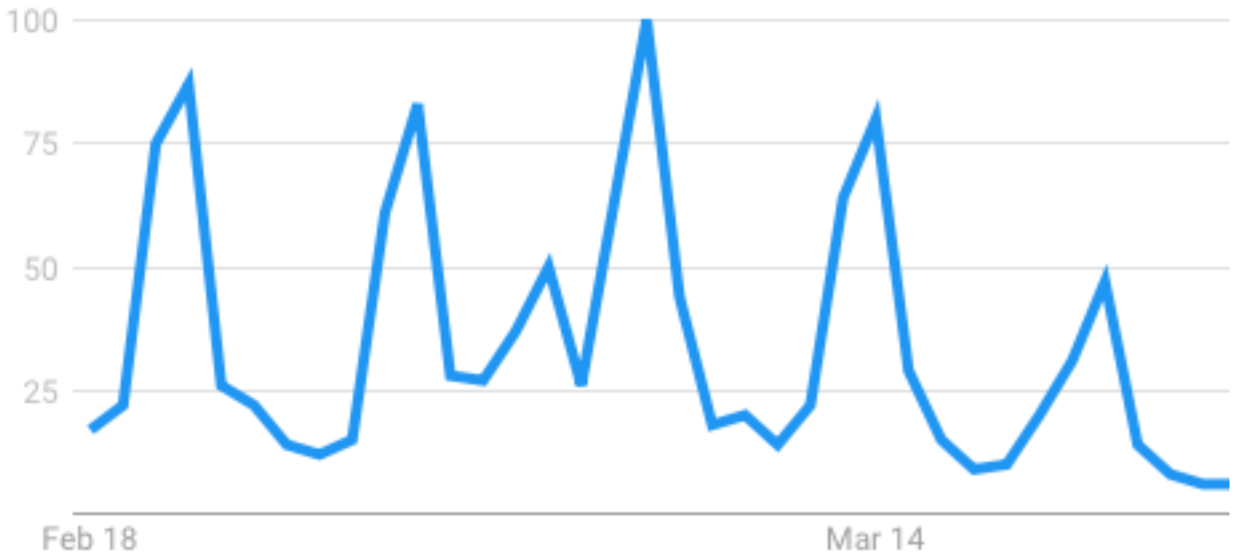
This is best possible for *deterministic strategies*

Theorem [KVV'90 + BC08, DJK13…]:

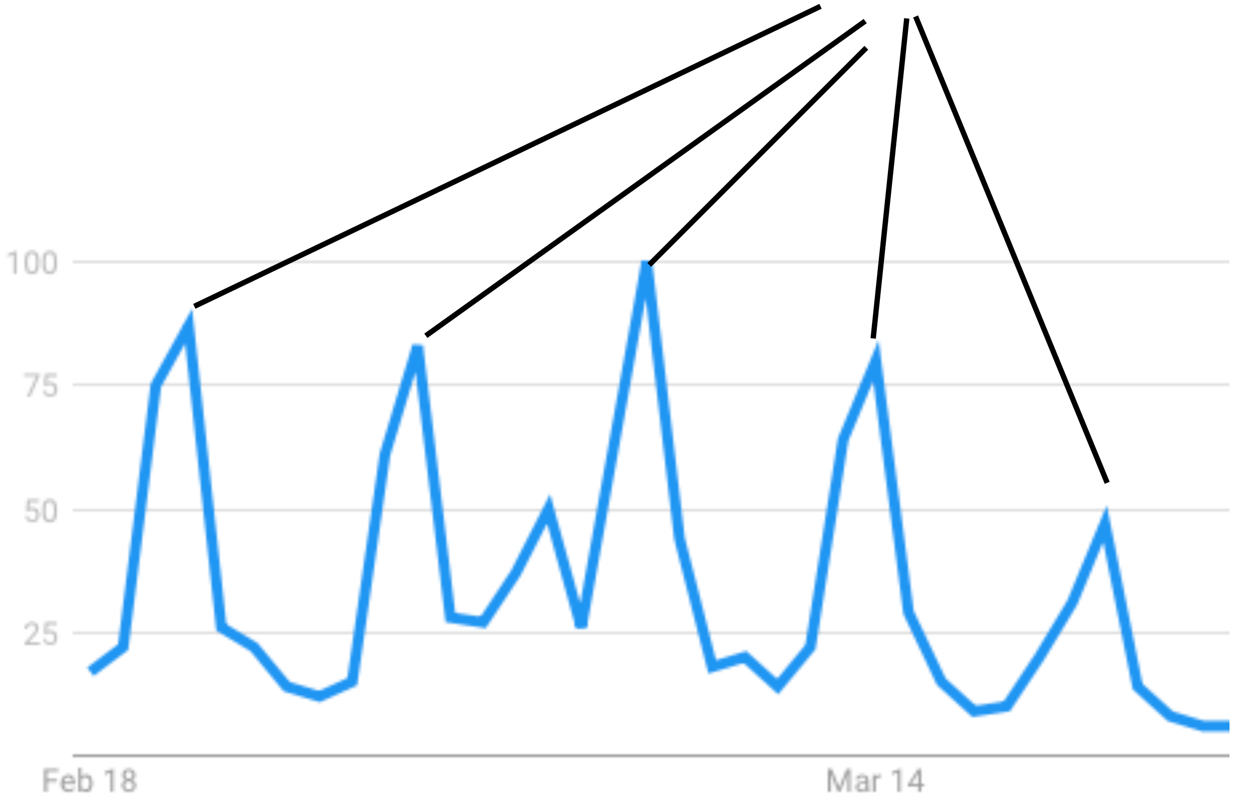
Ranking is (1-1/e)-competitive

This is best possible for *(randomized) strategies*

“Premier league” searches in UK



“Premier league” searches in UK



Sunday

ML Algorithms

Coca-Cola

PEPSI

rivella

IKEA®

ML Algorithm

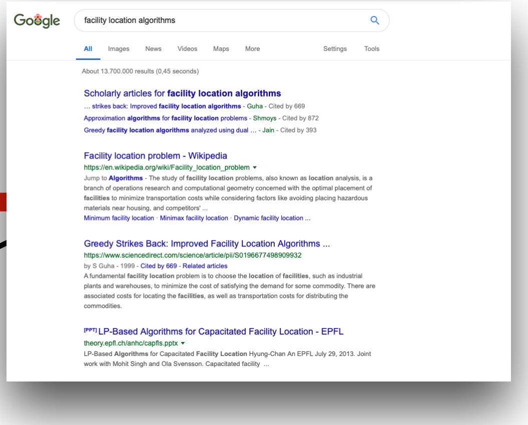
Excellent guarantee normal days

Coca-Cola

PEPSI

rivella

IKEA



facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms

… strikes back: Improved **facility location algorithms** - Guha - Cited by 669

Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872

Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia

https://en.wikipedia.org/wiki/Facility_location_problem ▾

Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...

Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...

https://www.sciencedirect.com/science/article/pii/S0196677498909932

by S Guha - 1999 - Cited by 669 - Related articles

A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL

theory.epfl.ch/anhc/capfls.pptx ▾

LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

ML Algorithm

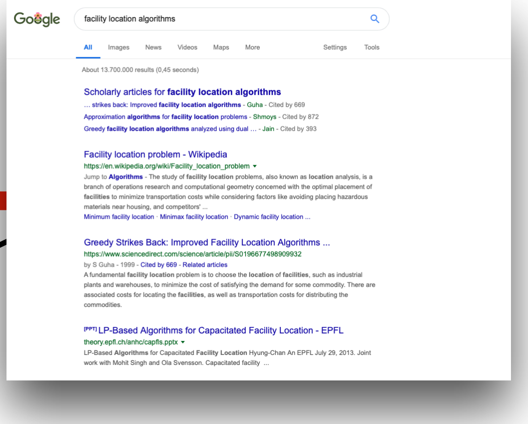
Excellent guarantee normal days

Coca-Cola

PEPSI

rivella

IKEA



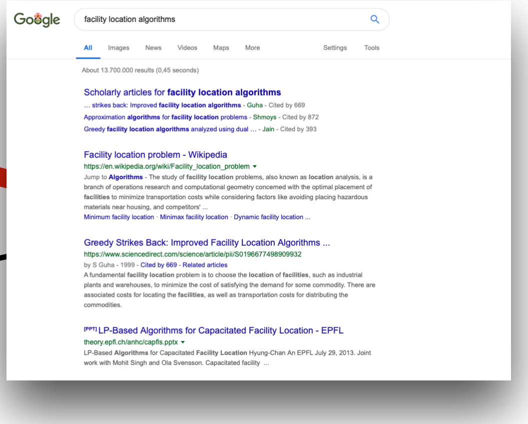
About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...



About 13.700.000 results (0,45 seconds)

Scholarly articles for **facility location algorithms**
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

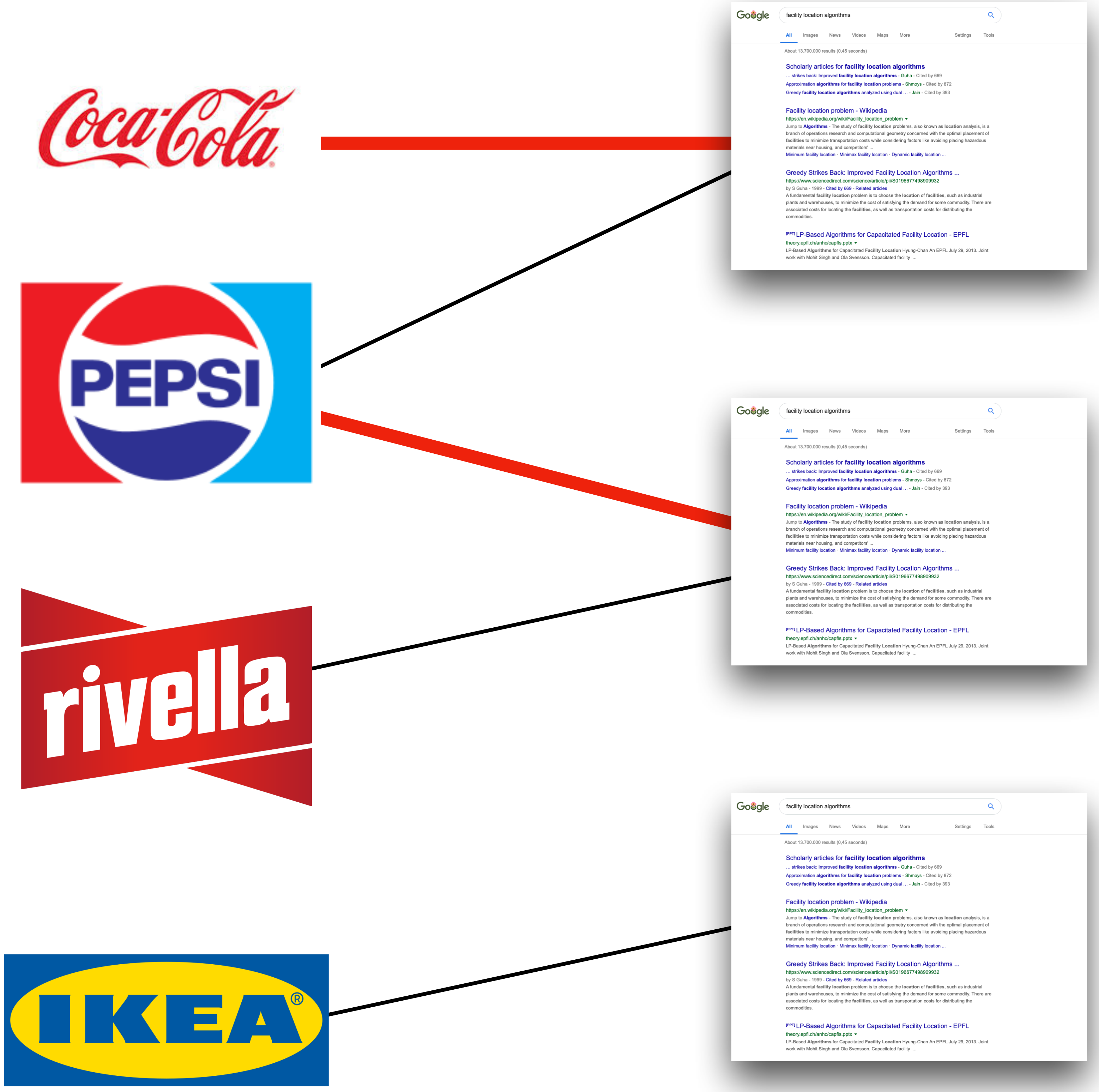
Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

ML Algorithm

Excellent guarantee normal days



Google facility location algorithms

All Images News Videos Maps More Settings Tools

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Google facility location algorithms

All Images News Videos Maps More Settings Tools

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

Google facility location algorithms

All Images News Videos Maps More Settings Tools

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
… strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual … - Jain - Cited by 393

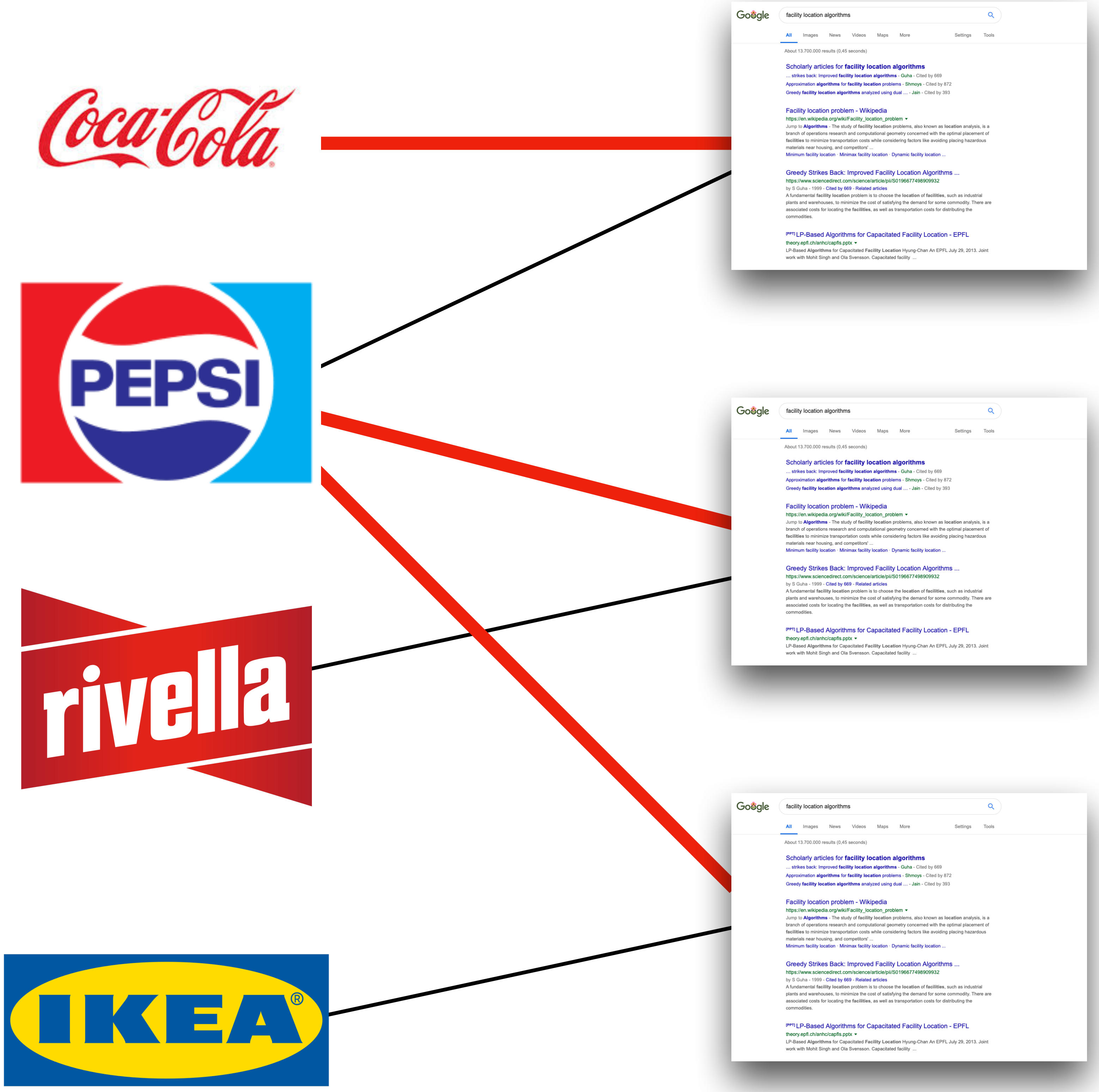
Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem ▾
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx ▾
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

ML Algorithm

**Excellent guarantee
normal days**



facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

facility location algorithms

About 13.700.000 results (0,45 seconds)

Scholarly articles for facility location algorithms
... strikes back: Improved **facility location algorithms** - Guha - Cited by 669
Approximation **algorithms** for **facility location** problems - Shmoys - Cited by 872
Greedy **facility location algorithms** analyzed using dual ... - Jain - Cited by 393

Facility location problem - Wikipedia
https://en.wikipedia.org/wiki/Facility_location_problem
Jump to **Algorithms** - The study of **facility location** problems, also known as **location** analysis, is a branch of operations research and computational geometry concerned with the optimal placement of **facilities** to minimize transportation costs while considering factors like avoiding placing hazardous materials near housing, and competitors' ...
Minimum facility location · Minimax facility location · Dynamic facility location ...

Greedy Strikes Back: Improved Facility Location Algorithms ...
https://www.sciencedirect.com/science/article/pii/S0196677498909932
by S Guha - 1999 - Cited by 669 - Related articles
A fundamental **facility location** problem is to choose the **location** of **facilities**, such as industrial plants and warehouses, to minimize the cost of satisfying the demand for some commodity. There are associated costs for locating the **facilities**, as well as transportation costs for distributing the commodities.

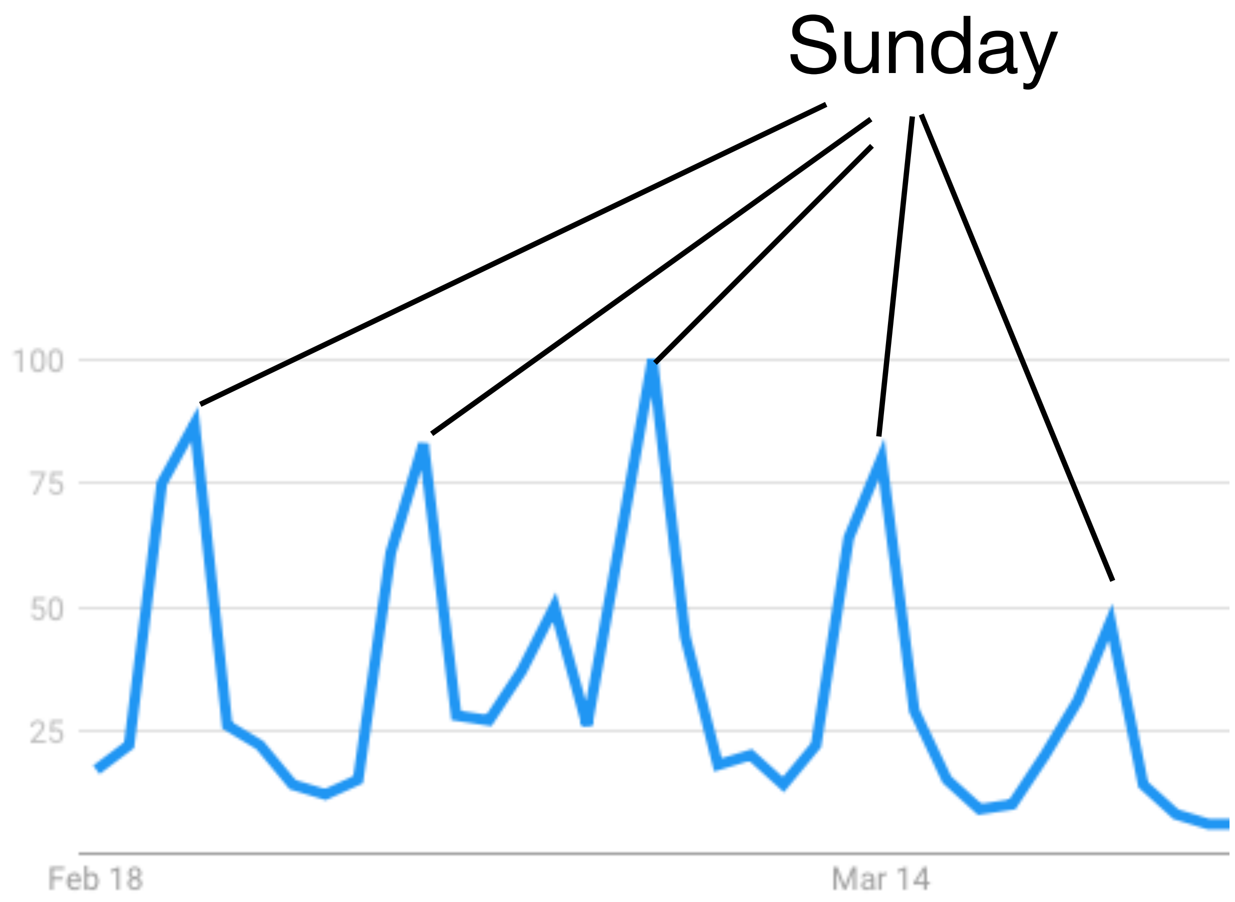
[PPT] LP-Based Algorithms for Capacitated Facility Location - EPFL
theory.epfl.ch/anhc/capfls.pptx
LP-Based **Algorithms** for Capacitated **Facility Location** Hyung-Chan An EPFL July 29, 2013. Joint work with Mohit Singh and Ola Svensson. Capacitated facility ...

ML Algorithm

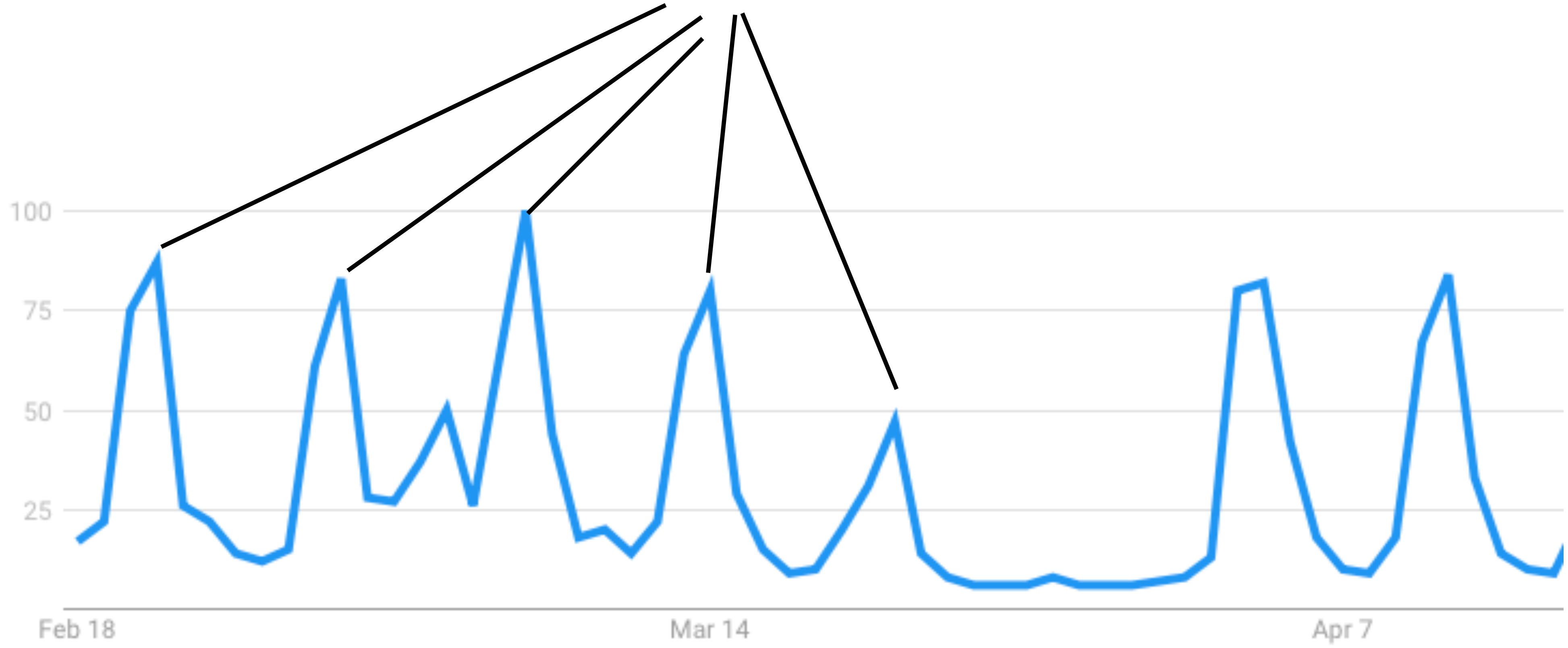
Excellent guarantee normal days

But no worst-case guarantees

“Premier league” searches in UK

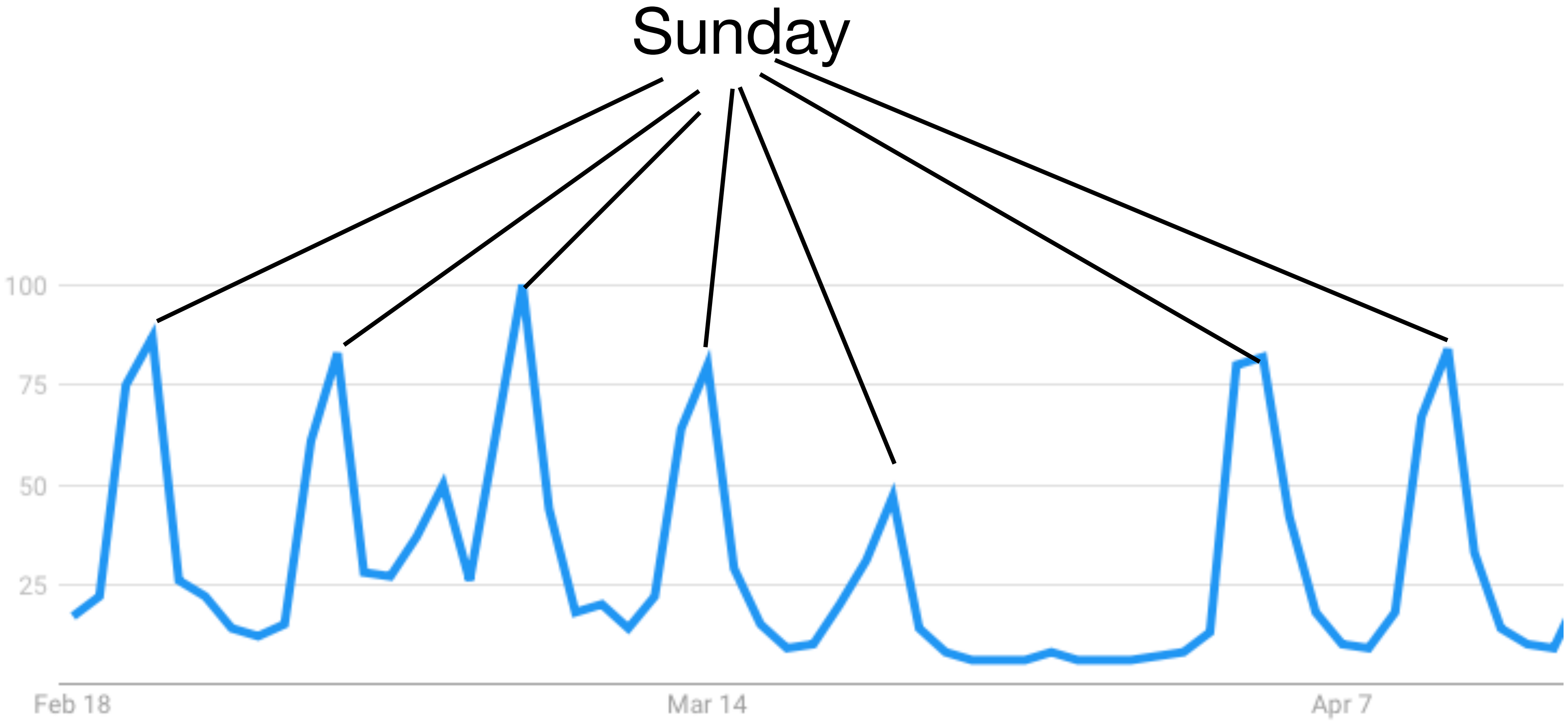


“Premier league” searches in UK

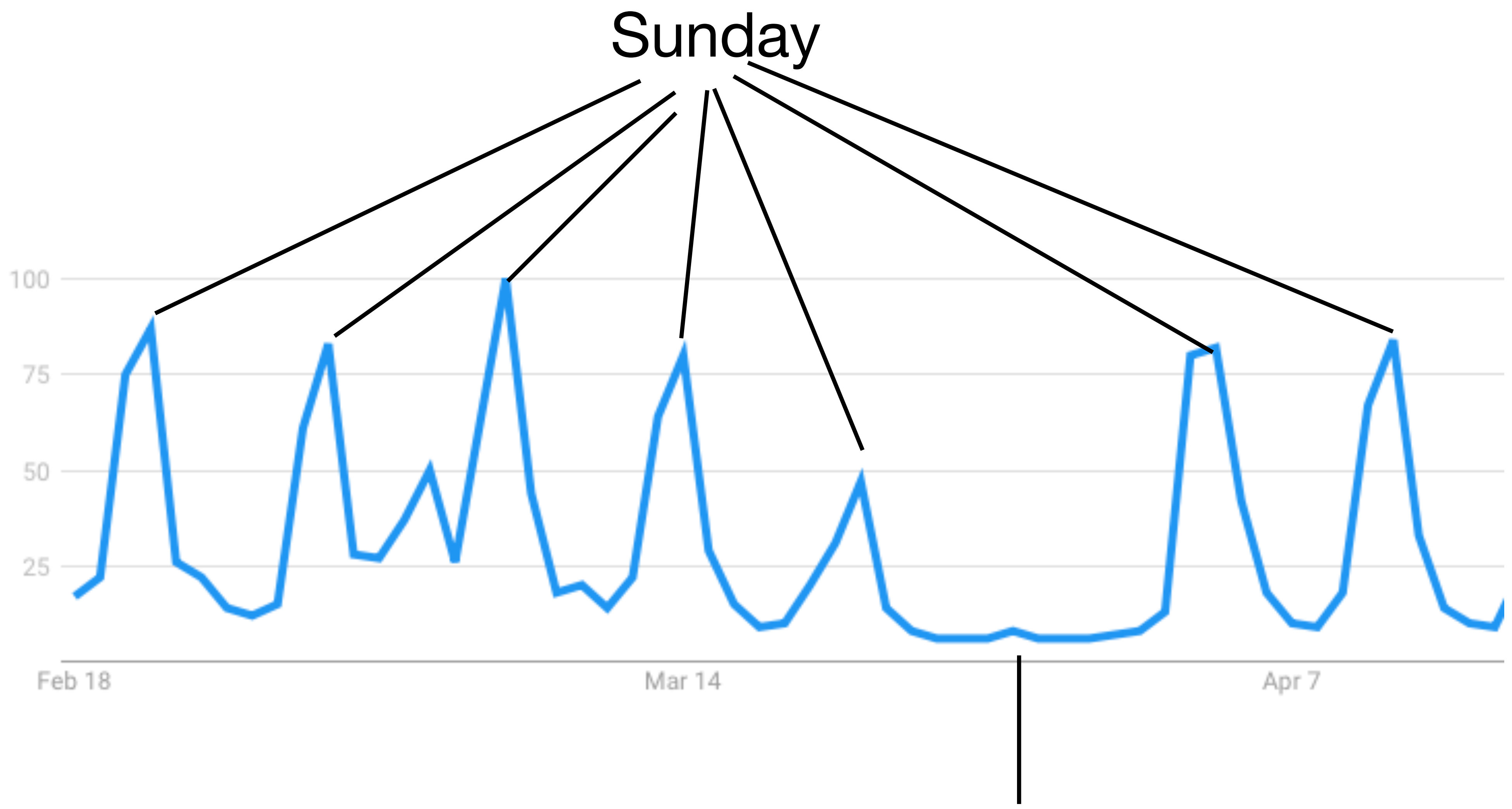


Sunday

“Premier league” searches in UK



“Premier league” searches in UK



Sunday

International fixtures

World-cup qualifiers in Europe

Learning-Augmented Online Algorithms

Online Algorithms $\cap$ ML $=$ Learning Augmented Algorithms

Online Algorithms $\cap$ ML $=$ Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms		
ML		
Learning Augmented Algorithms		

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	
ML		
Learning Augmented Algorithms		

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	😩
ML		
Learning Augmented Algorithms		

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	😩
ML	😩	
Learning Augmented Algorithms		

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	😩
ML	😩	😁
Learning Augmented Algorithms		

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	😩
ML	😩	😁
Learning Augmented Algorithms	😎	

Online Algorithms ∩ ML = Learning Augmented Algorithms

	worst-case guarantees	(often) great performance in real world
Online Algorithms	😁	😩
ML	😩	😁
Learning Augmented Algorithms	😎	😎

Learning-Augmented Online Algorithms

- Online algorithm with access to predictions about the future
- No assumptions on the predictor

Online Algorithm

augmented with predictions

Three Desiderata

Three Desiderata

- **Consistency:** if predictions are correct, algorithm gives close to optimal solution

Three Desiderata

- **Consistency:** if predictions are correct, algorithm gives close to optimal solution
- **Robustness:** Even under adversarial predictions, algorithm maintains a worst-case guarantee (ideally comparable to best known online algorithm)

Three Desiderata

- **Consistency:** if predictions are correct, algorithm gives close to optimal solution
- **Robustness:** Even under adversarial predictions, algorithm maintains a worst-case guarantee (ideally comparable to best known online algorithm)
- **Smoothness:** Performance degrades nicely in the error of the predictor

Consistency vs Robustness

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

- **Prediction P of number of days**

Consistency vs Robustness

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

- **Prediction P of number of days**

No trust

Can't do better than standard online algorithms

Bad consistency

Consistency vs Robustness

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**

- The difficulty is that we do not know the total number of days we will be skiing

- **Prediction P of number of days**

No trust

Can't do better than standard online algorithms

Bad consistency

Complete trust

Excellent consistency but what if Prediction is 10B and reality is 1

Bad robustness

Consistency vs Robustness

Example: Ski rental

- At the beginning of each day, decide whether to **buy skis at a cost of B** or **rent skis for that day at a cost of 1**
- The difficulty is that we do not know the total number of days we will be skiing
- **Prediction P of number of days**

No trust

Can't do better than standard online algorithms

Bad consistency

Complete trust

Excellent consistency but what if Prediction is 10B and reality is 1

Bad robustness

Balanced trust $\lambda \in (0,1)$

Wait λB days to buy if prediction is to buy

Consistency: $(1 + \lambda)$
Robustness: $O(1/\lambda)$

Emerging and quickly growing line of work

Emerging and quickly growing line of work

- Ad allocation by Mahdian, Nazerzadeh, Saberi, EC’07

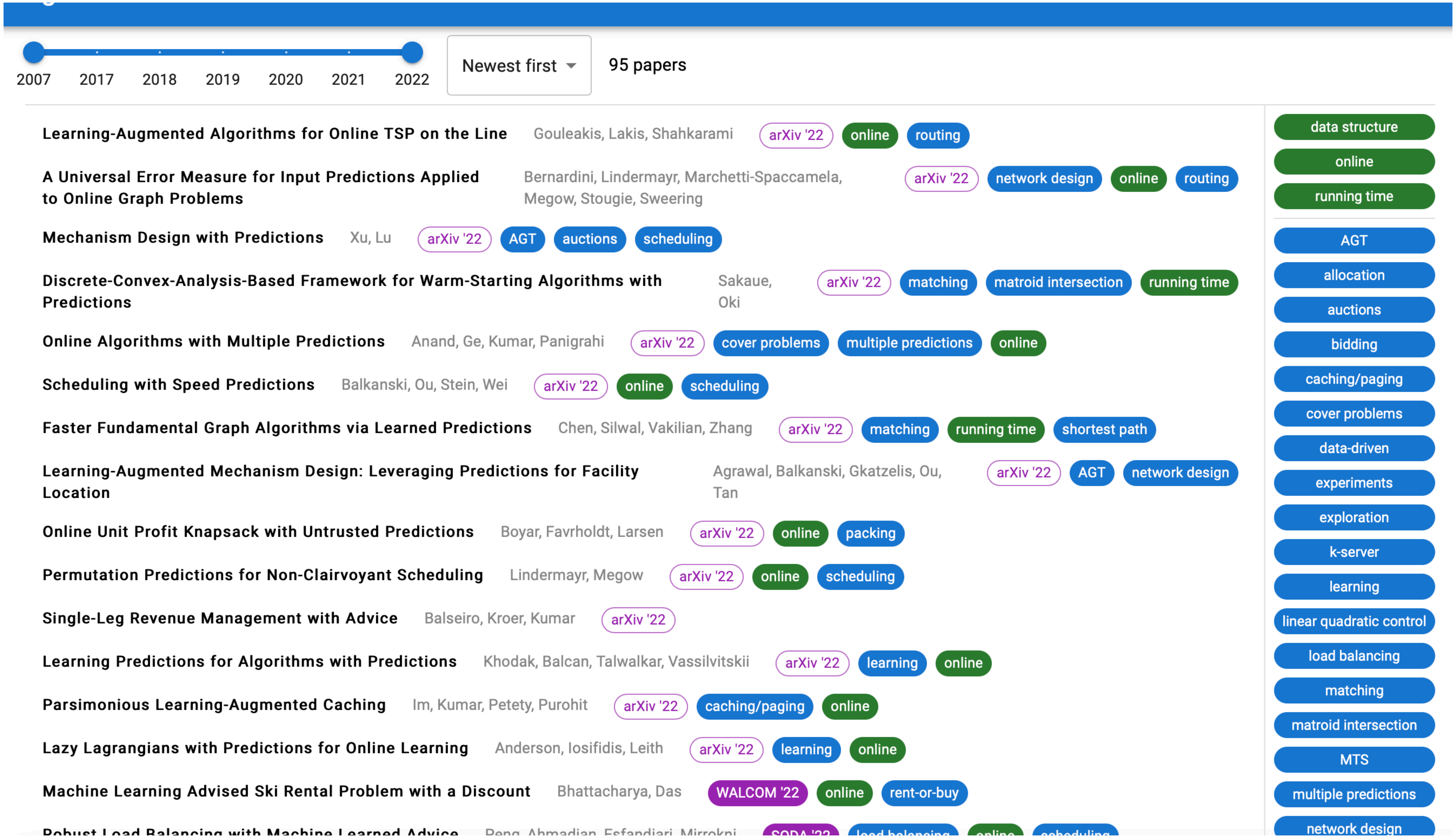
Emerging and quickly growing line of work

- Ad allocation by Mahdian, Nazerzadeh, Saberi, EC’07
- Competitive caching (Lykouris and Vassilvitskii ICML 2018, Rohatgi SODA 2020)

Emerging and quickly growing line of work

- Ad allocation by Mahdian, Nazerzadeh, Saberi, EC’07
- Competitive caching (Lykouris and Vassilvitskii ICML 2018, Rohatgi SODA 2020)
- Ski rental (Kumar et al. NeurIPS 2018, Gollapudi and Panigrahi ICML 2019)
- Bloom filters (Mitzenmacher NeurIPS 2018)
- Metrical task systems (Antoniadis et al. ICML 2020)
- Frequency estimation in data streams (Hsu et al. ICLR 2019)
- Scheduling (Lattanzi et al. SODA 2020, Bamas et al. NeurIPS 2020)
- …
- + courses, workshops…

Emerging and quickly growing line of work



2007 2017 2018 2019 2020 2021 2022

Newest first ▾

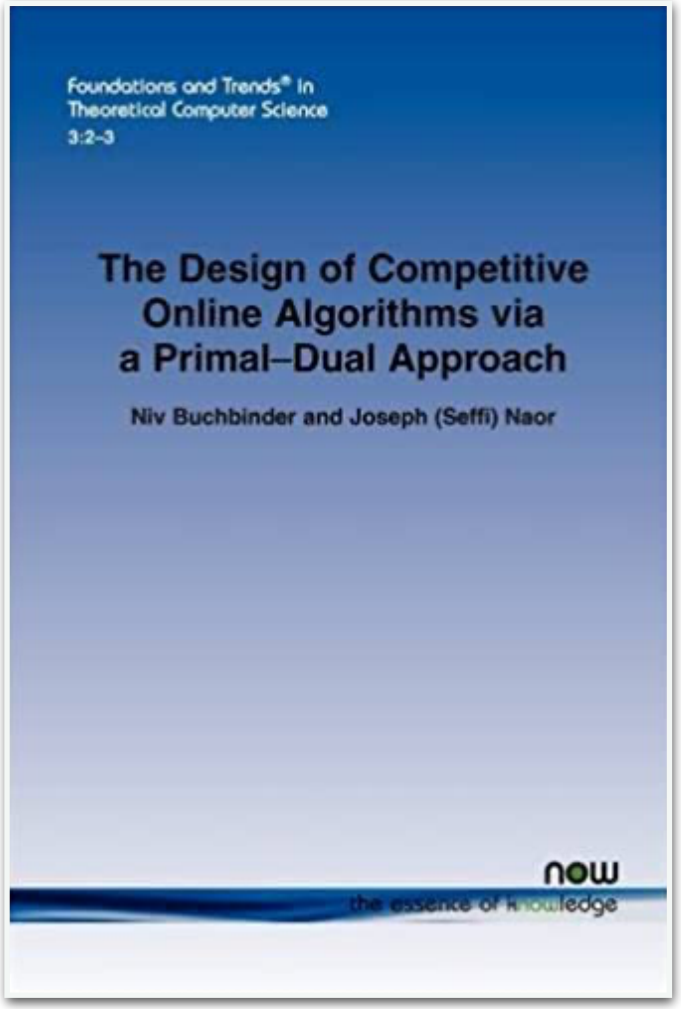
95 papers

Learning-Augmented Algorithms for Online TSP on the Line	Gouleakis, Lakis, Shahkarami	arXiv '22 · online · routing
A Universal Error Measure for Input Predictions Applied to Online Graph Problems	Bernardini, Lindermayr, Marchetti-Spaccamela, Megow, Stougie, Sweering	arXiv '22 · network design · online · routing
Mechanism Design with Predictions	Xu, Lu	arXiv '22 · AGT · auctions · scheduling
Discrete-Convex-Analysis-Based Framework for Warm-Starting Algorithms with Predictions	Sakaue, Oki	arXiv '22 · matching · matroid intersection · running time
Online Algorithms with Multiple Predictions	Anand, Ge, Kumar, Panigrahi	arXiv '22 · cover problems · multiple predictions · online
Scheduling with Speed Predictions	Balkanski, Ou, Stein, Wei	arXiv '22 · online · scheduling
Faster Fundamental Graph Algorithms via Learned Predictions	Chen, Silwal, Vakilian, Zhang	arXiv '22 · matching · running time · shortest path
Learning-Augmented Mechanism Design: Leveraging Predictions for Facility Location	Agrawal, Balkanski, Gkatzelis, Ou, Tan	arXiv '22 · AGT · network design
Online Unit Profit Knapsack with Untrusted Predictions	Boyar, Favrholdt, Larsen	arXiv '22 · online · packing
Permutation Predictions for Non-Clairvoyant Scheduling	Lindermayr, Megow	arXiv '22 · online · scheduling
Single-Leg Revenue Management with Advice	Balseiro, Kroer, Kumar	arXiv '22
Learning Predictions for Algorithms with Predictions	Khodak, Balcan, Talwalkar, Vassilvitskii	arXiv '22 · learning · online
Parsimonious Learning-Augmented Caching	Im, Kumar, Petety, Purohit	arXiv '22 · caching/paging · online
Lazy Lagrangians with Predictions for Online Learning	Anderson, Iosifidis, Leith	arXiv '22 · learning · online
Machine Learning Advised Ski Rental Problem with a Discount	Bhattacharya, Das	WALCOM '22 · online · rent-or-buy
Robust Load Balancing with Machine Learned Advice	Peng, Ahmadian, Esfandiari, Mirrokni	SODA '22 · load balancing · online · scheduling

data structure
online
running time

AGT
allocation
auctions
bidding
caching/paging
cover problems
data-driven
experiments
exploration
k-server
learning
linear quadratic control
load balancing
matching
matroid intersection
MTS
multiple predictions
network design

https://algorithms-with-predictions.github.io



Foundations and Trends® in
Theoretical Computer Science
3:2–3

The Design of Competitive
Online Algorithms via
a Primal–Dual Approach

Niv Buchbinder and Joseph (Seffi) Naor

now
the essence of knowledge

Can we adapt powerful frameworks such as the primal-dual approach to the learning augmented setting?

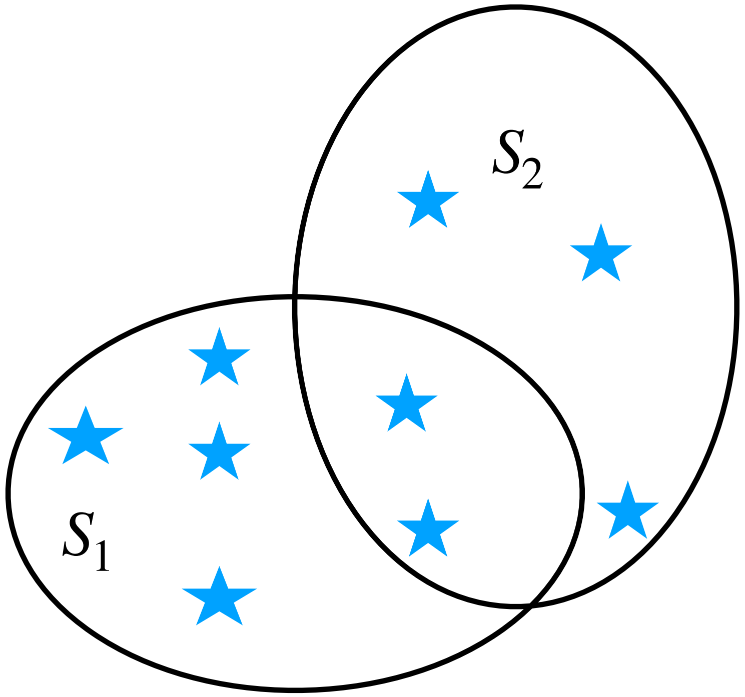
Outline

- Learning-augmented online algorithms
- **Case study: set cover**
- Instantiating PDLA for other problems
- Future directions

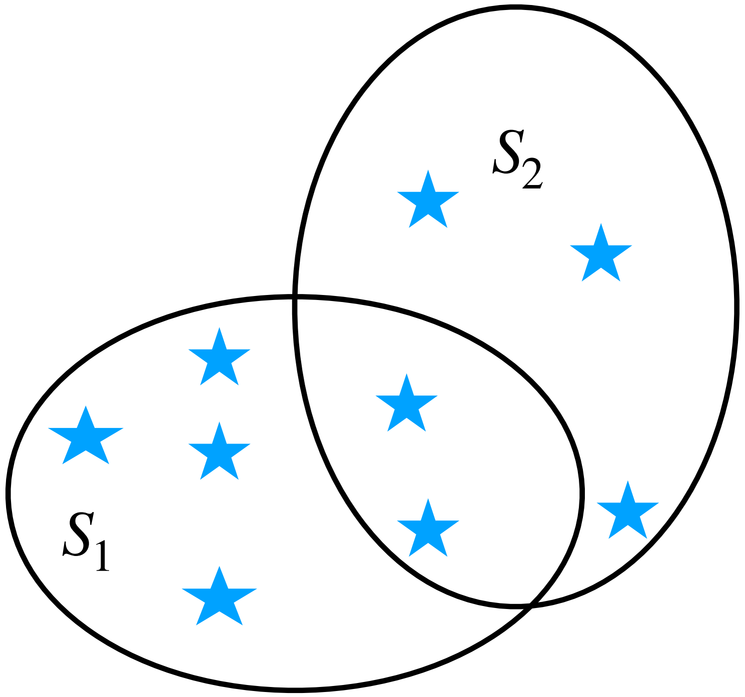
Fractional Online Set Cover

Fractional online set cover problem

Fractional online set cover problem

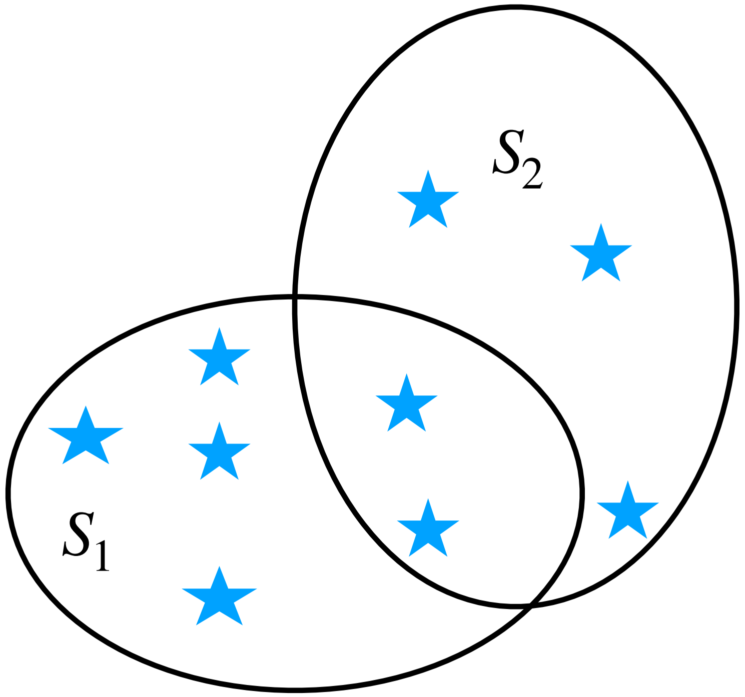


Fractional online set cover problem



Goal:

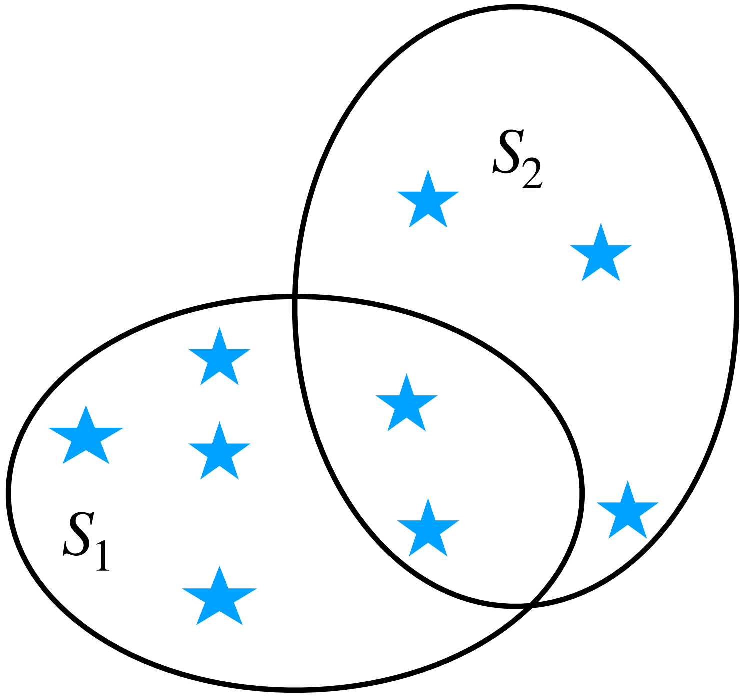
Fractional online set cover problem



Goal:

- cover fractionally every newly arrived element

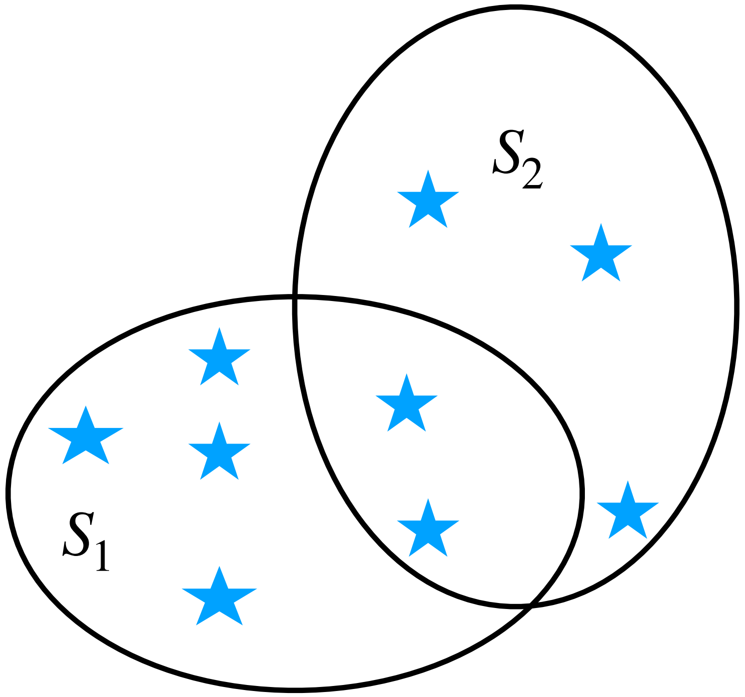
Fractional online set cover problem



Goal:

- cover <u>fractionally</u> every newly arrived element
- decisions are irrevocable = <u>cannot decrease current fractional solution</u>

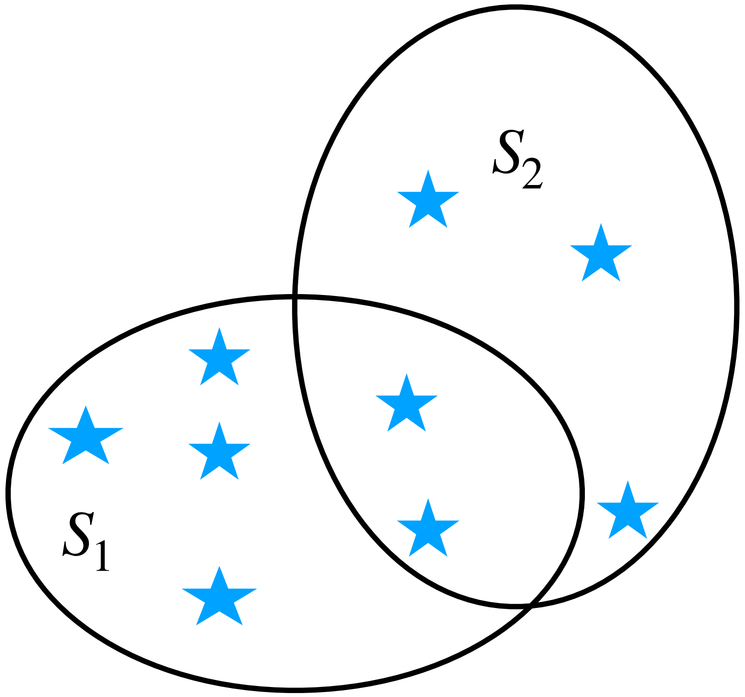
Fractional online set cover problem



Goal:

- cover fractionally every newly arrived element
- decisions are irrevocable = cannot decrease current fractional solution
- minimize the sum of fractionally selected sets

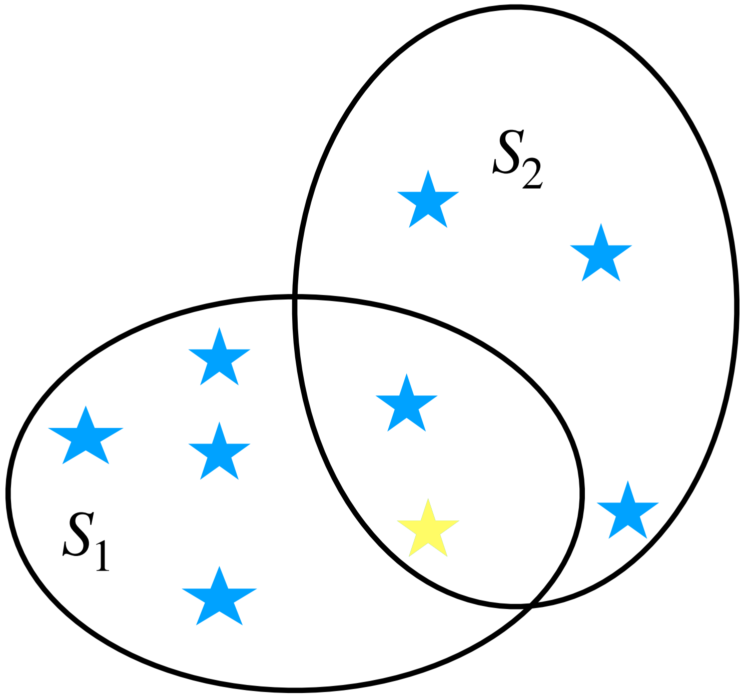
Fractional online set cover problem



LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

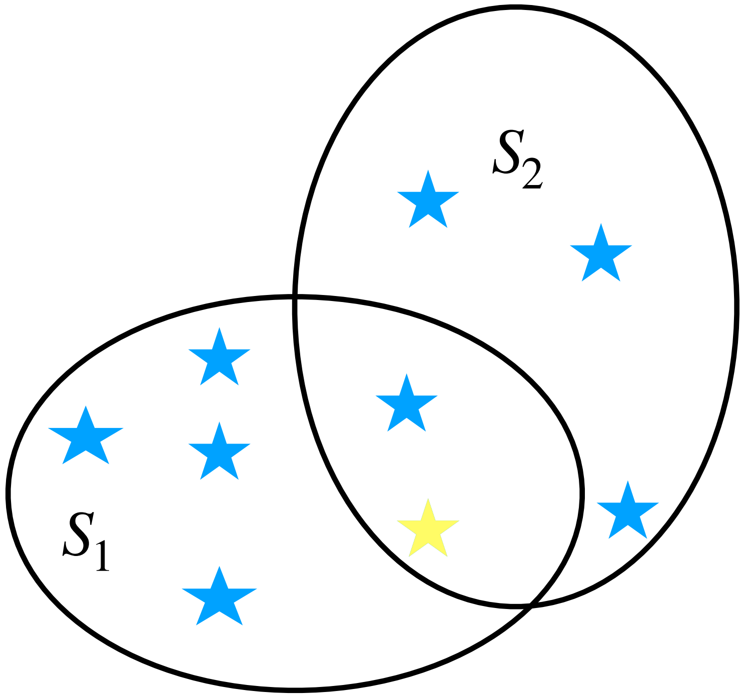
Fractional online set cover problem



LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

Fractional online set cover problem

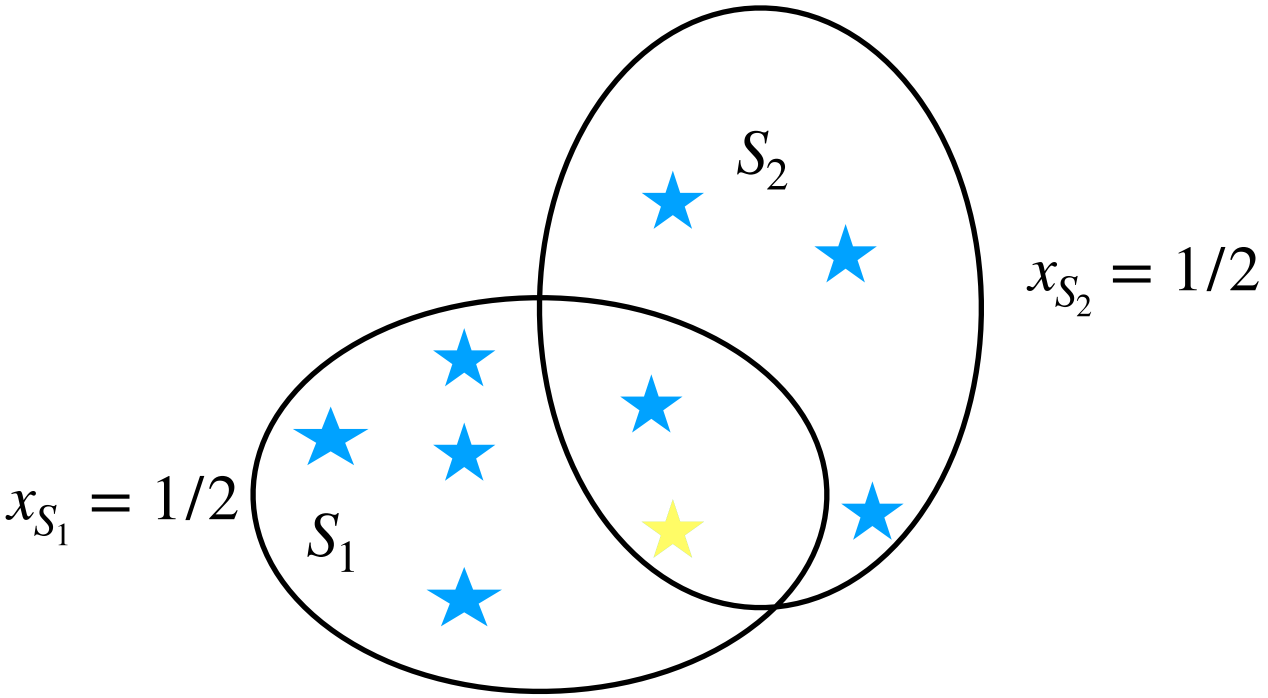


$x_{S_1} + x_{S_2} \geq 1$

LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

Fractional online set cover problem

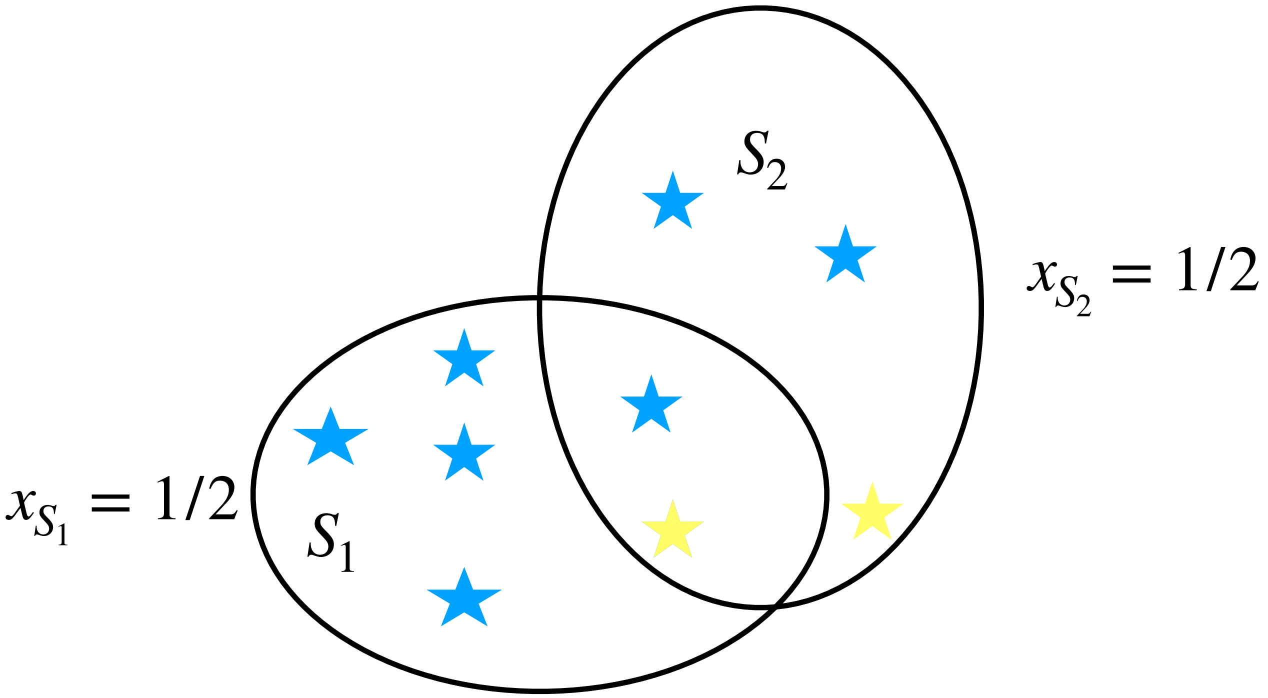


$x_{S_1} + x_{S_2} \geq 1$

LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_{i} x_{S_i}$

Fractional online set cover problem

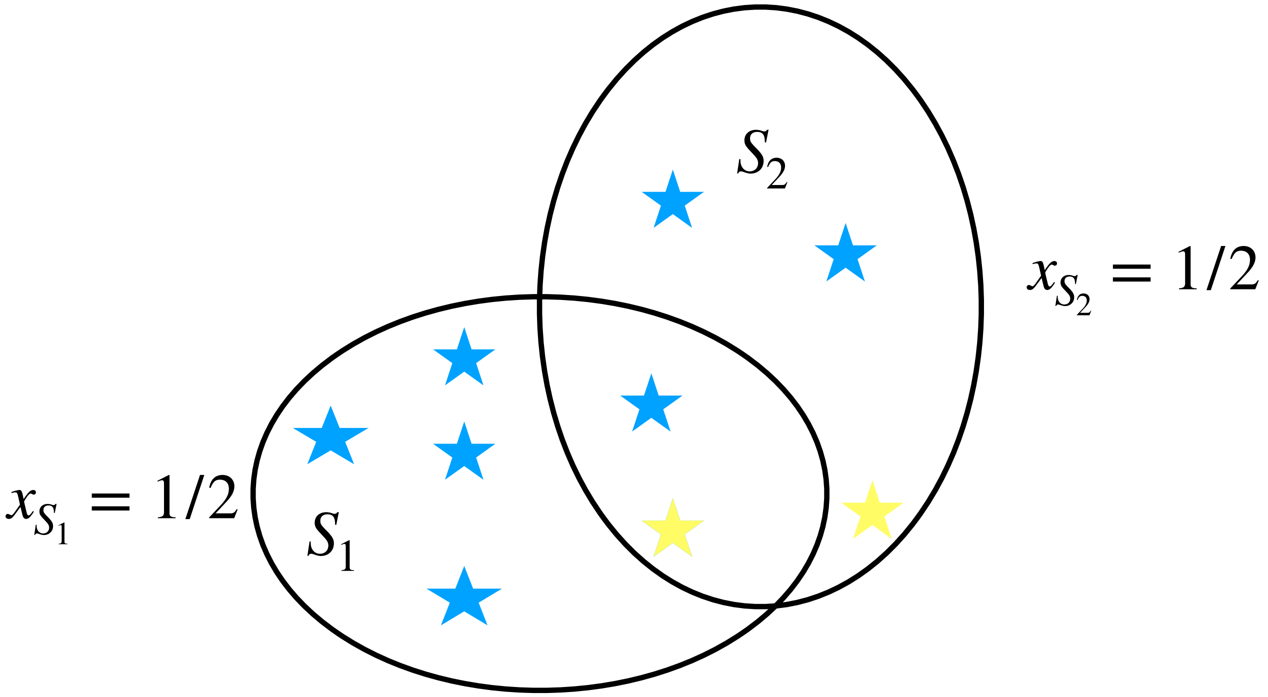


$x_{S_1} + x_{S_2} \geq 1$

LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

Fractional online set cover problem

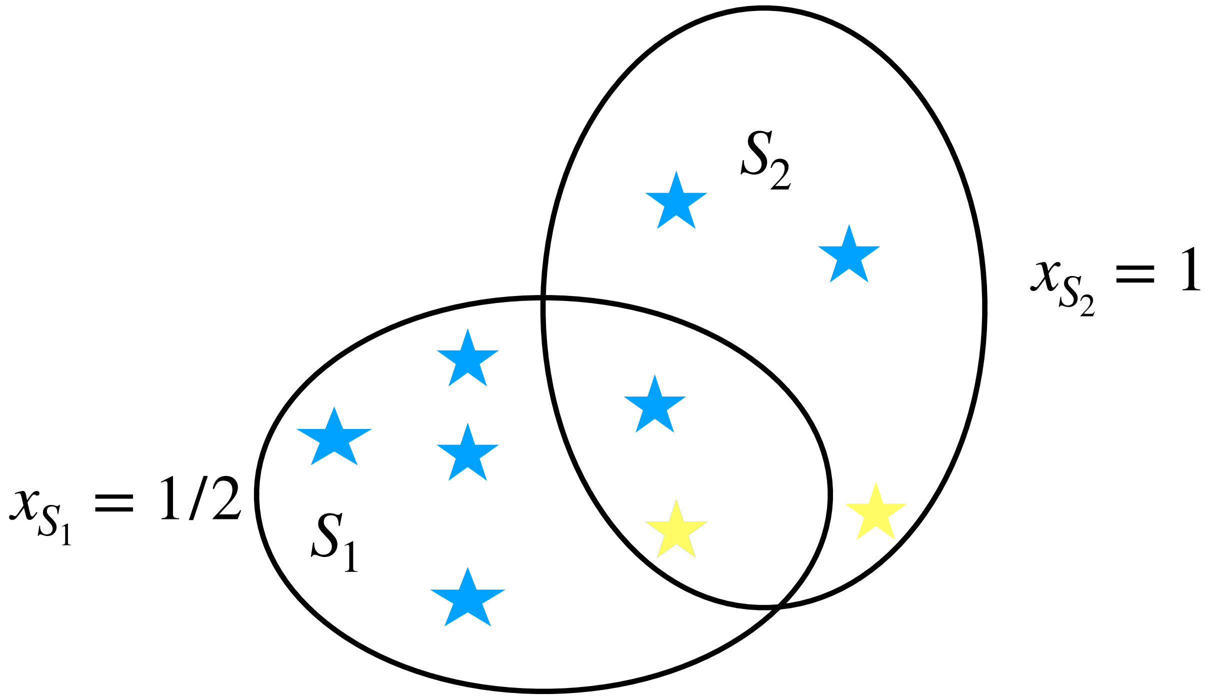


$$x_{S_1} + x_{S_2} \geq 1$$
$$x_{S_2} \geq 1$$

LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

Fractional online set cover problem



$$x_{S_1} + x_{S_2} \geq 1$$
$$x_{S_2} \geq 1$$

LP formulation:

- each set has a corresponding variable
- at every new element e arrival a new constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ needs to be satisfied
- minimize $\sum_i x_{S_i}$

Difficult instance

Difficult instance

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \dots \quad x_{S_m} = 0$

Constraints

Difficult instance

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/m \quad x_{S_3} = 1/m \quad \ldots \quad x_{S_m} = 1/m$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/m \quad x_{S_3} = 1/m \quad \dots \quad x_{S_m} = 1/m$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-1) \quad \ldots \quad x_{S_m} = 1/(m-1)$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-1) \quad \dots \quad x_{S_m} = 1/(m-1)$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \ldots \quad x_{S_m} = 1/(m-2)$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \dots \quad x_{S_m} = 1/(m-2)$

Constraints

$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_3} + \dots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \dots \quad x_{S_m} = 1$

Constraints

$$\begin{aligned} x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\ x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\ x_{S_3} + \dots + x_{S_m} &\geq 1 \\ &\vdots \\ x_{S_m} &\geq 1 \end{aligned}$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \ldots \quad x_{S_m} = 1$

$O(\log m)$

Constraints

$$\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}$$

Difficult instance

$O(\log m)$

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \dots \quad x_{S_m} = 1$ ←

$OPT = 1$

Constraints

$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_3} + \dots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

Difficult instance

Current solution

$x_{S_1} = 1/m \quad x_{S_2} = 1/(m-1) \quad x_{S_3} = 1/(m-2) \quad \dots \quad x_{S_m} = 1$

$O(\log m)$

$OPT = 1$

Constraints

$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_3} + \dots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

Which can be shown to be a lower bound on the performance of any online algorithm

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

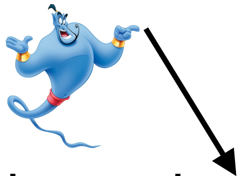
$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \dots \quad x_{S_m} = 0$

Constraints



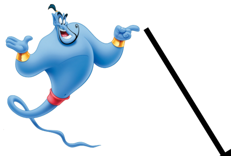
$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 1$

Constraints



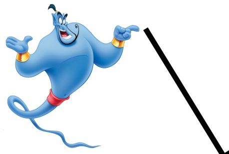
$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 1$

Constraints



$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
x_{S_3} + \ldots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

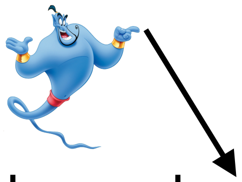
Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 1$

$cost = OPT = 1$

Constraints



$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$\vdots$$

$$x_{S_m} \geq 1$$

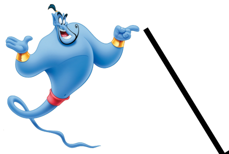
Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 1$

$cost = OPT = 1$

Constraints



$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$\vdots$$

$$x_{S_m} \geq 1$$

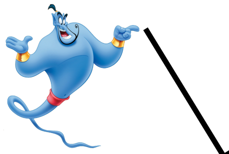
Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \dots \quad x_{S_m} = 1$

$cost = OPT = 1$ 👍

Constraints



$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_3} + \dots + x_{S_m} \geq 1$$

$$\vdots$$

$$x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 0 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 0 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 0 \quad \dots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 0 \quad \ldots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \ldots + x_{S_m} \geq 1$$

$$x_{S_3} + \ldots + x_{S_m} \geq 1$$

Difficult instance with a prediction

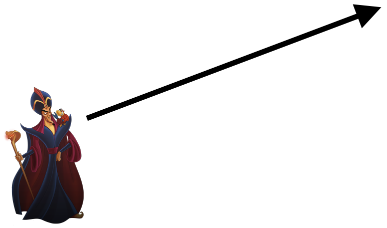
Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 0 \quad \dots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$



$$x_{S_3} + \dots + x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

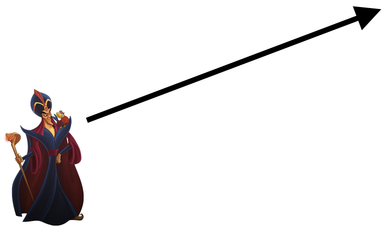
$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 1 \quad \dots \quad x_{S_m} = 0$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_3} + \dots + x_{S_m} \geq 1$$



Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 1 \quad \ldots \quad x_{S_m} = 1$

Constraints

$$\begin{aligned} x_{S_1} + x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\ x_{S_2} + x_{S_3} + \ldots + x_{S_m} &\geq 1 \\ x_{S_3} + \ldots + x_{S_m} &\geq 1 \end{aligned}$$

$$\vdots$$

$$x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 1 \quad \dots \quad x_{S_m} = 1$ ← $cost = m$

Constraints

$$x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_2} + x_{S_3} + \dots + x_{S_m} \geq 1$$

$$x_{S_3} + \dots + x_{S_m} \geq 1$$

$$\vdots$$

$$x_{S_m} \geq 1$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 1 \quad \dots \quad x_{S_m} = 1$

$cost = m$ ☠

Constraints

$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_3} + \dots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

Difficult instance with a prediction

Current solution

$x_{S_1} = 1 \quad x_{S_2} = 1 \quad x_{S_3} = 1 \quad \dots \quad x_{S_m} = 1$ ← $cost = m$ ☠

Constraints

$$
\begin{aligned}
x_{S_1} + x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_2} + x_{S_3} + \dots + x_{S_m} &\geq 1 \\
x_{S_3} + \dots + x_{S_m} &\geq 1 \\
&\vdots \\
x_{S_m} &\geq 1
\end{aligned}
$$

Completely trusting predictor has terrible robustness

Interesting tradeoff between consistency and robustness

The Primal-Dual Approach

Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e \in S_i} x_{S_i} \geq 1$ for every element e

Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e \in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

subject to $\sum_{e \in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

- If $\sum_{i:e \in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \dfrac{1}{\text{|\#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$

Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```

Example

Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e \in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

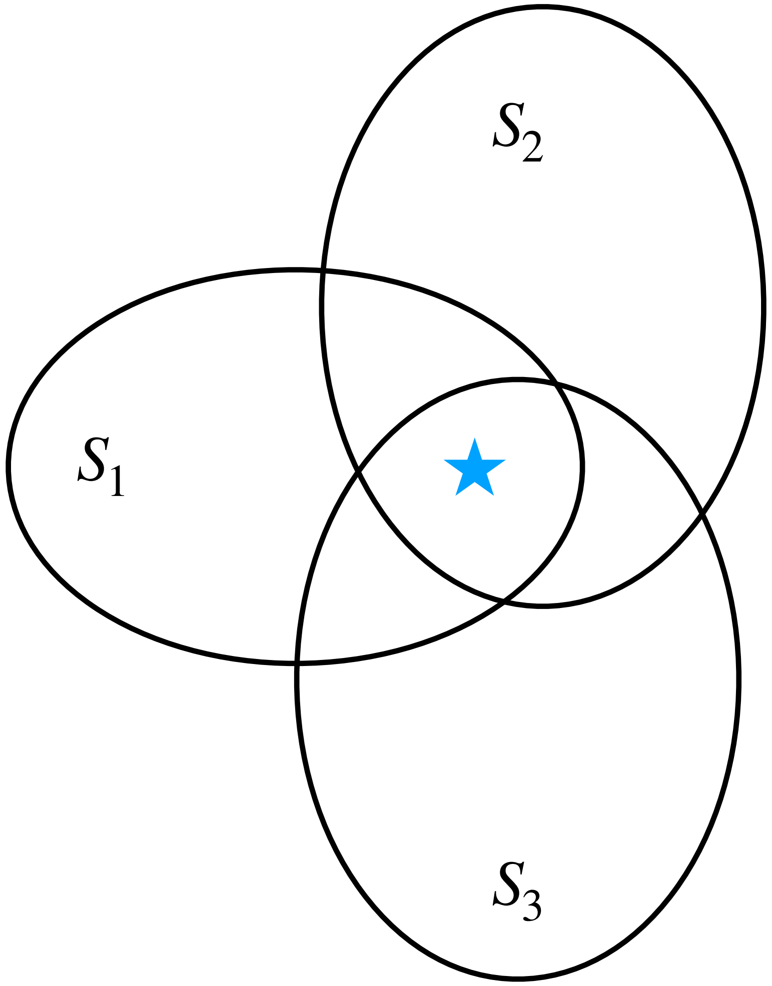
subject to $\sum_{e \in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

- If $\sum_{i:e \in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \dfrac{1}{\text{|\#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$

Example



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

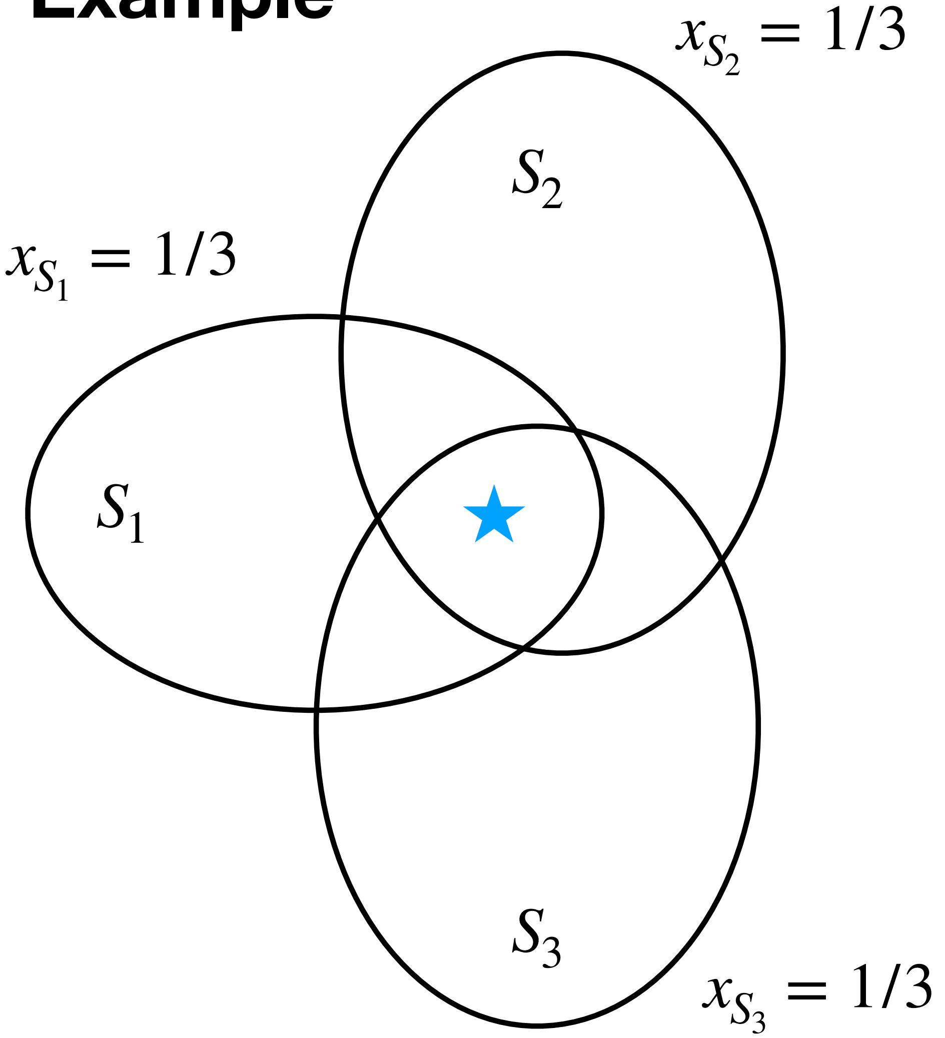
subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|\#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$

Example



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

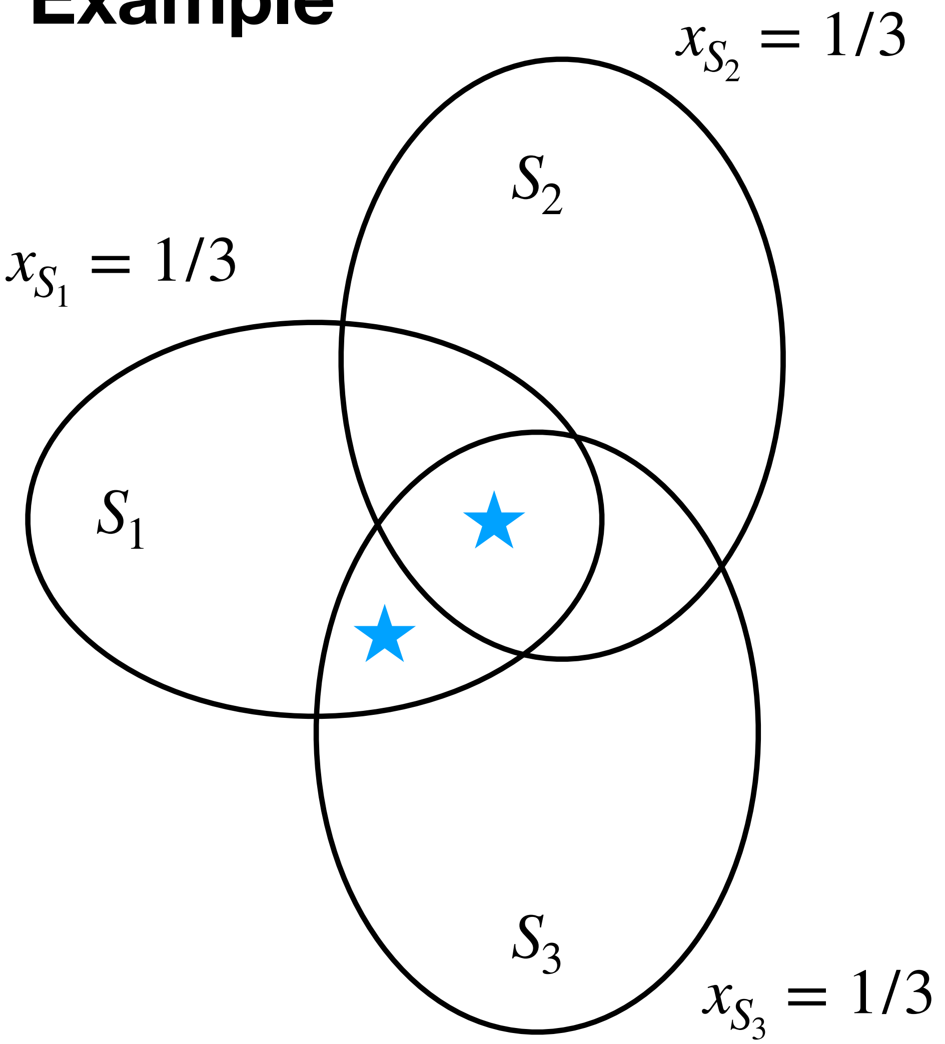
subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \dfrac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$

Example



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

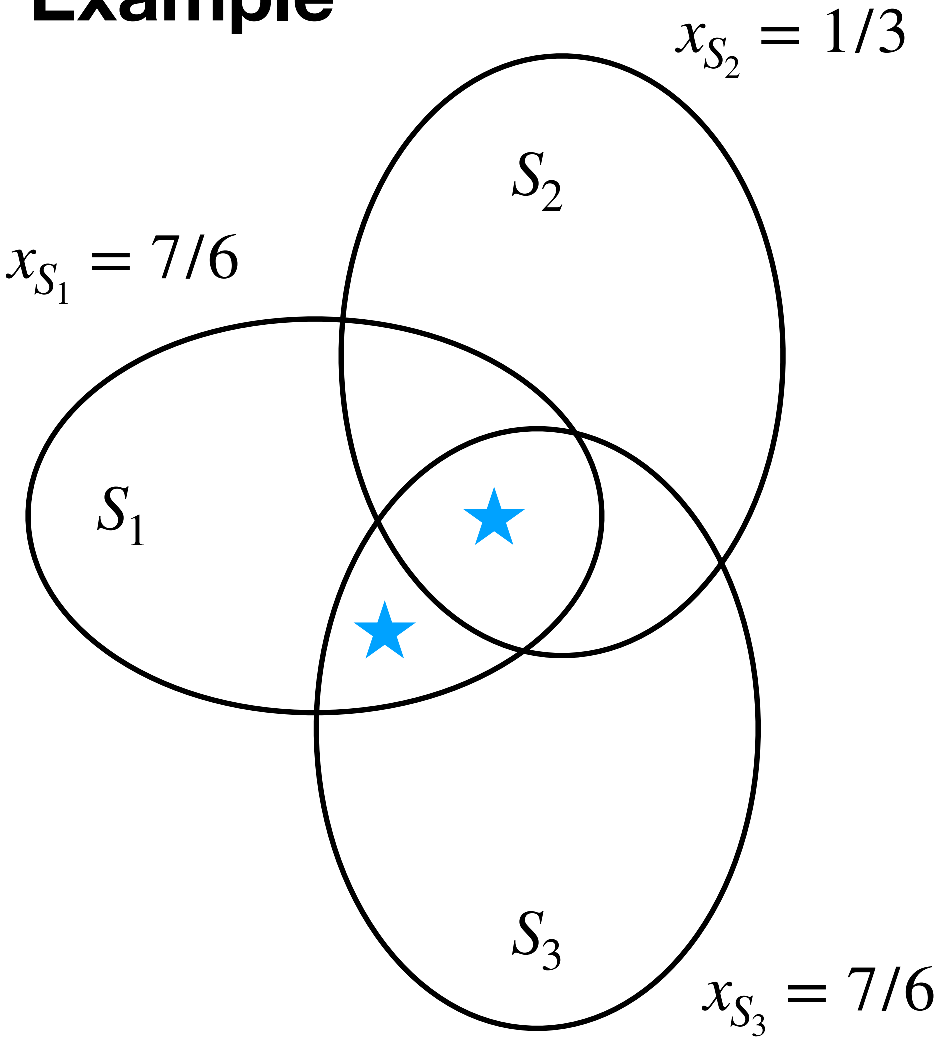
subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```

Example



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e \in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

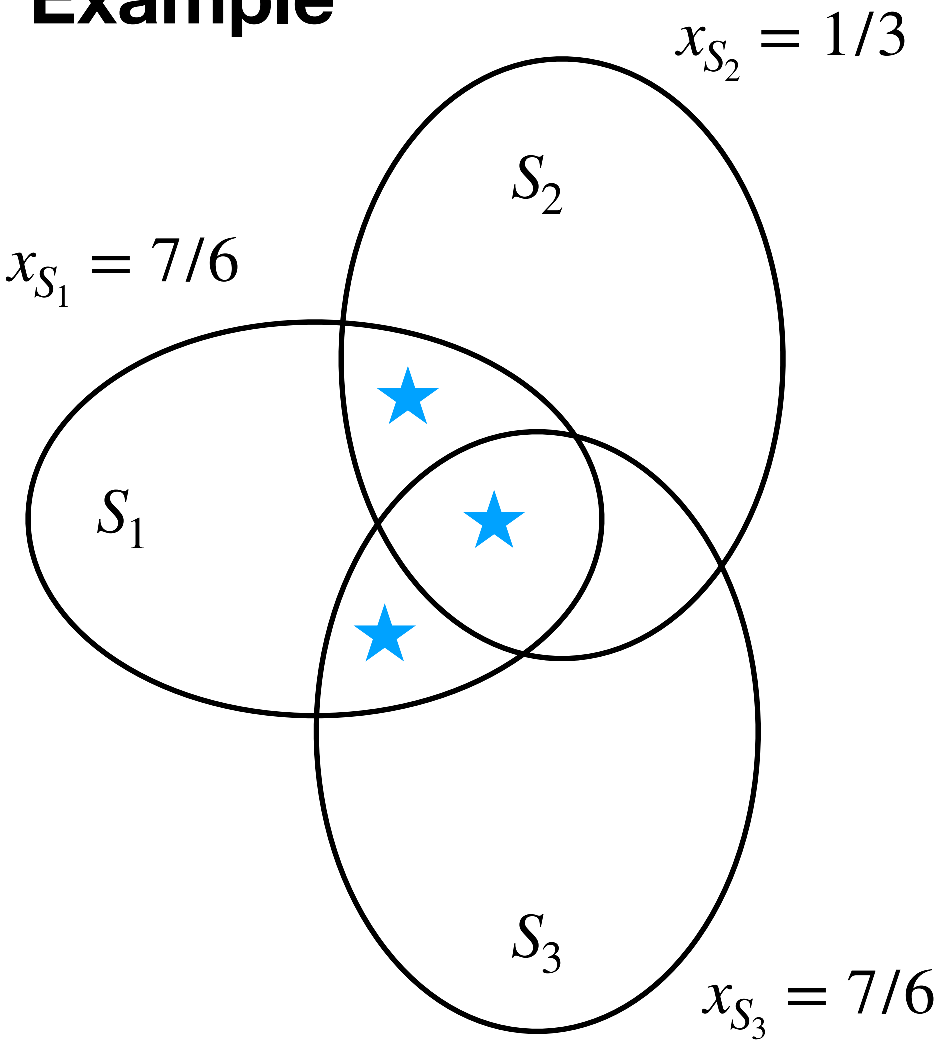
subject to $\sum_{e \in S_i} y_e \leq 1$ for every set S_i

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

- If $\sum_{i:e \in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \dfrac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$

Example



Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```

Primal

minimize $\sum_{i} x_{S_i}$

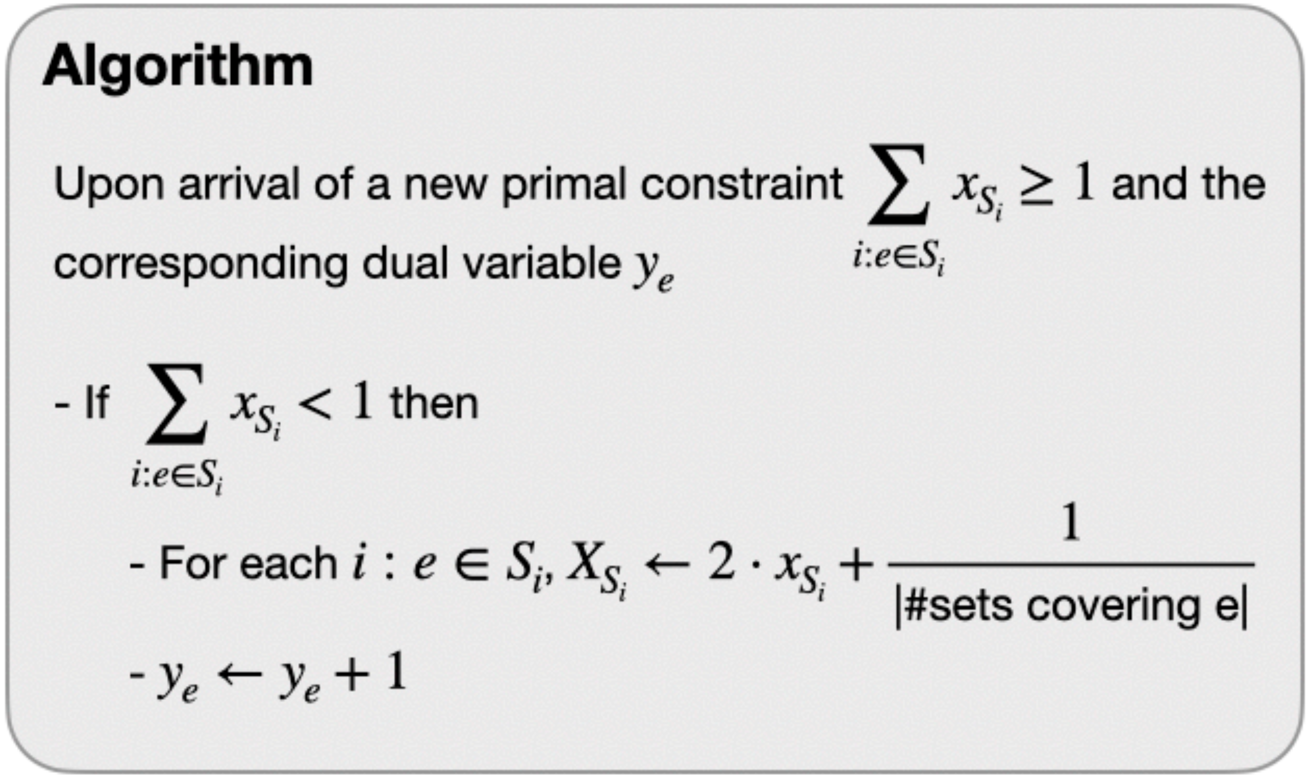
subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

maximize $\sum_{e} y_e$

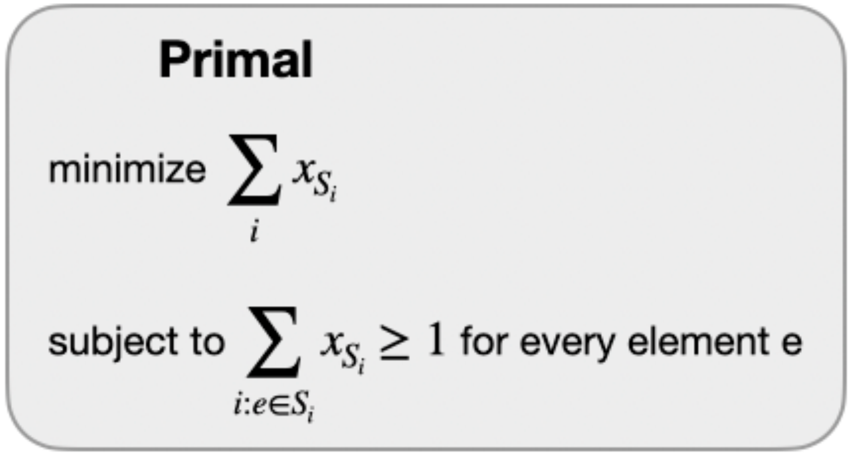
subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis



Algorithm

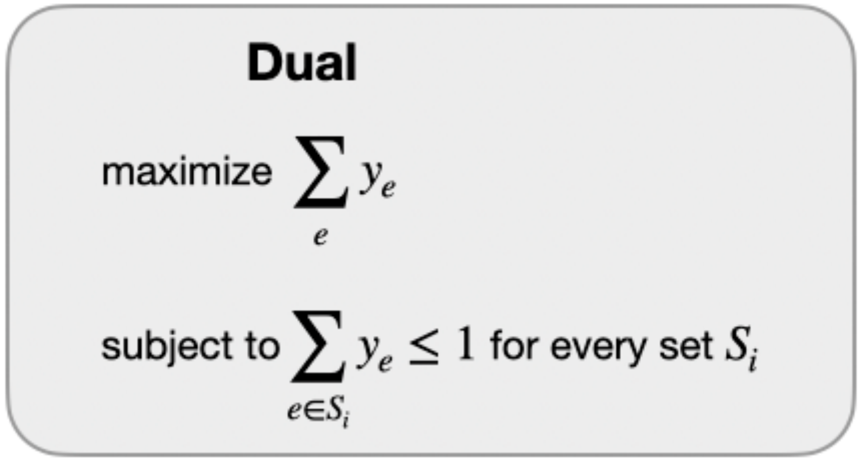
```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the
corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$
```



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e



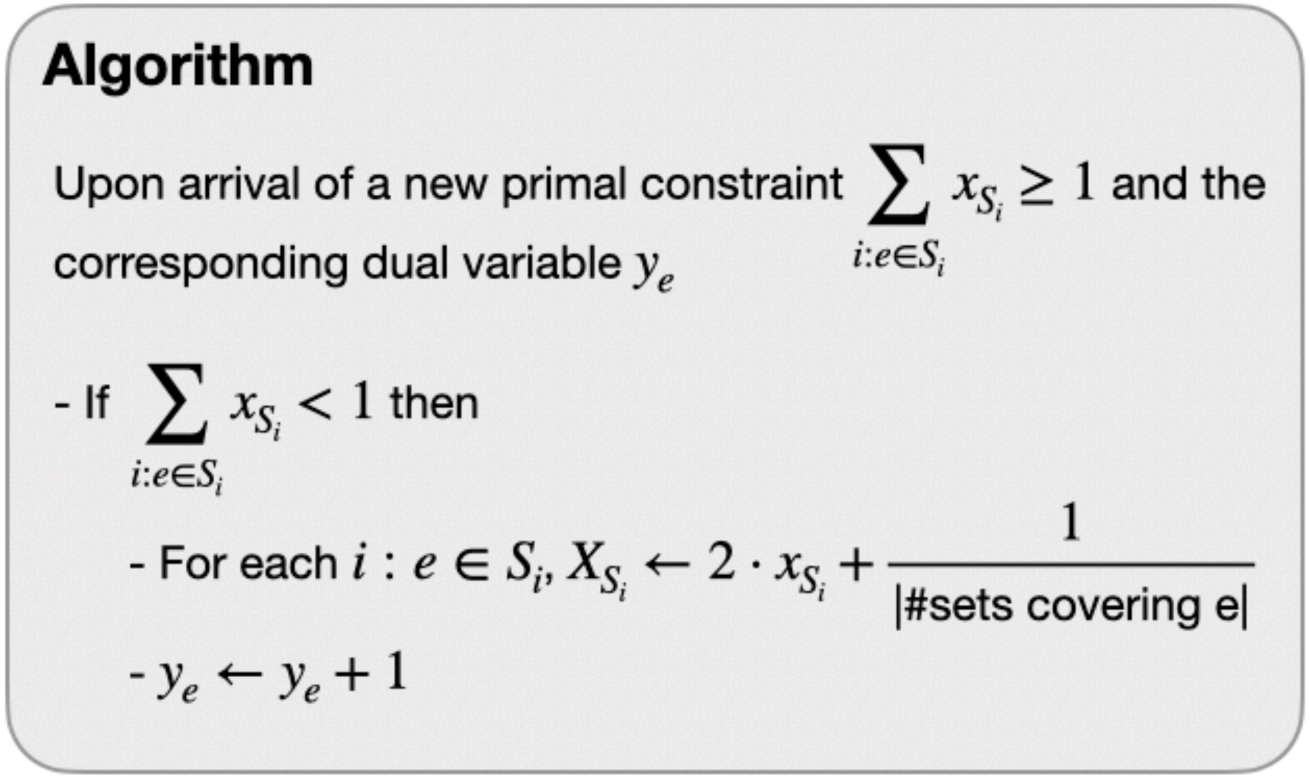
Dual

maximize $\sum_{e} y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis

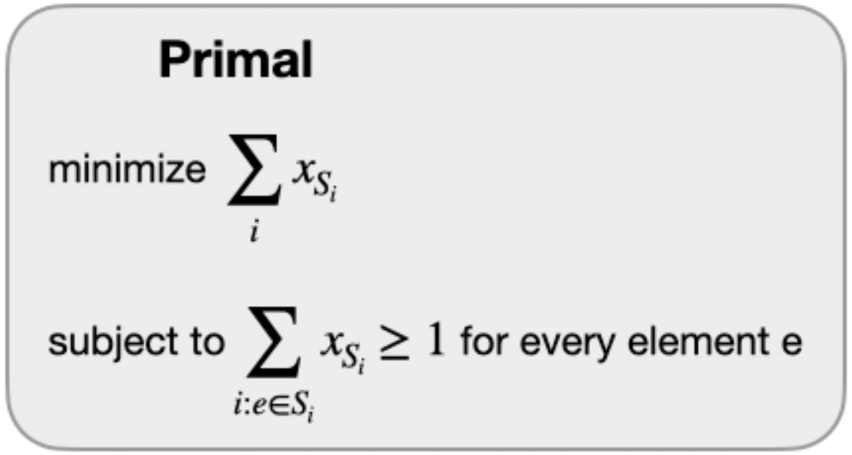
1. At each step the increase of primal is $\sum_{i:e\in S_i}(x_i + 1/|\#\text{sets covering e}|) \leq 2$ whereas increase in dual is 1



Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

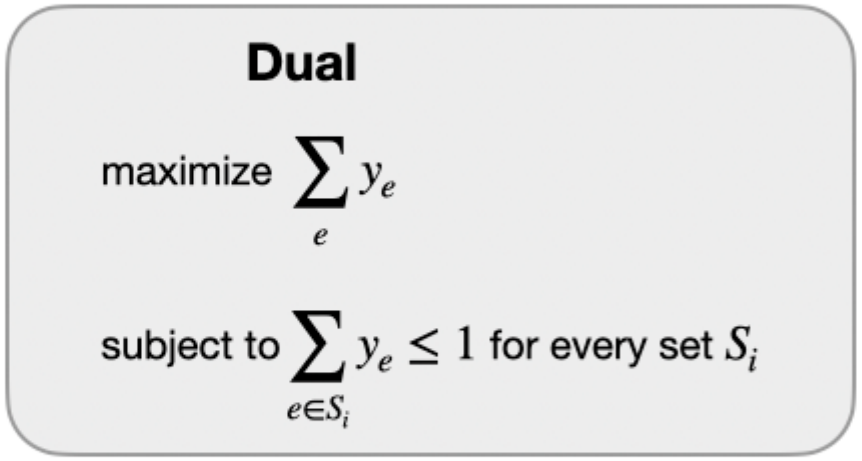
```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$
```



Primal

minimize $\sum_i x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e



Dual

maximize $\sum_e y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis

1. At each step the increase of primal is $\sum_{i:e\in S_i} (x_i + 1/|\#\text{sets covering e}|) \leq 2$ whereas increase in dual is 1

2. $y/\log(m)$ is a feasible dual solution:

Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{|\text{#sets covering e}|}$
    - $y_e \leftarrow y_e + 1$
```

Primal

minimize $\sum_i x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e

Dual

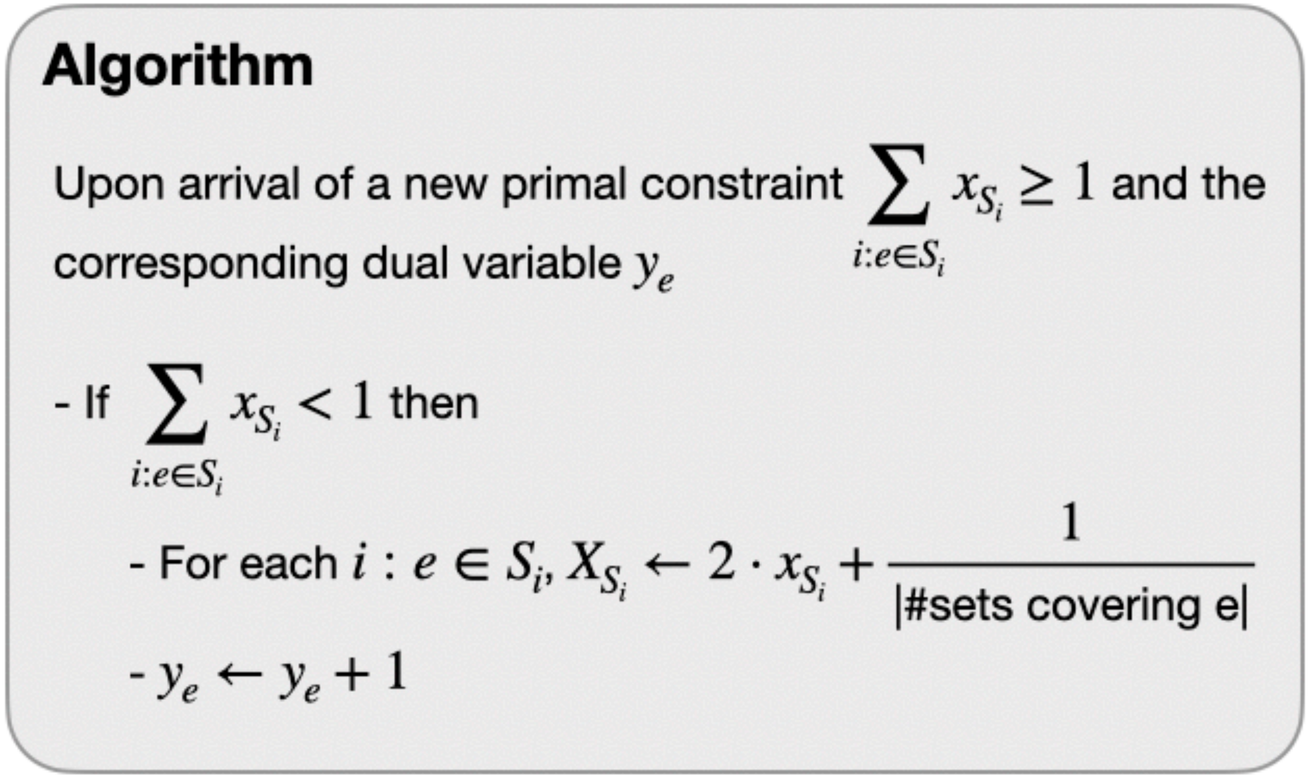
maximize $\sum_e y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis

1. At each step the increase of primal is $\sum_{i:e\in S_i} (x_i + 1/|\text{#sets covering e}|) \leq 2$ whereas increase in dual is 1

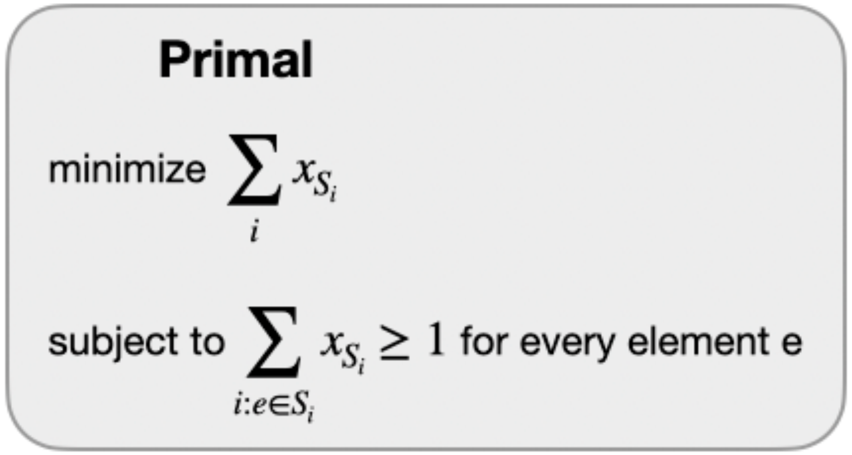
2. $y/\log(m)$ is a feasible dual solution:
   - every time a y_e variable is updated in a constraint $\sum_{e\in S_i} y_e \leq 1$



Algorithm

Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

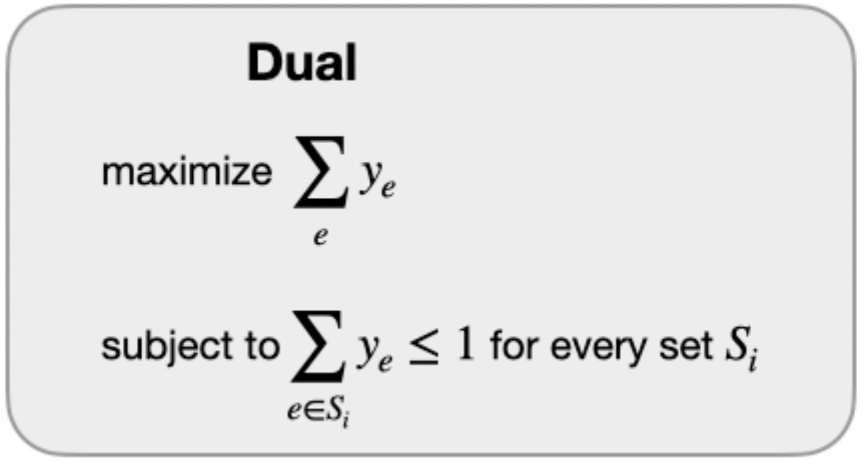
```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$
```



Primal

minimize $\sum_i x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e



Dual

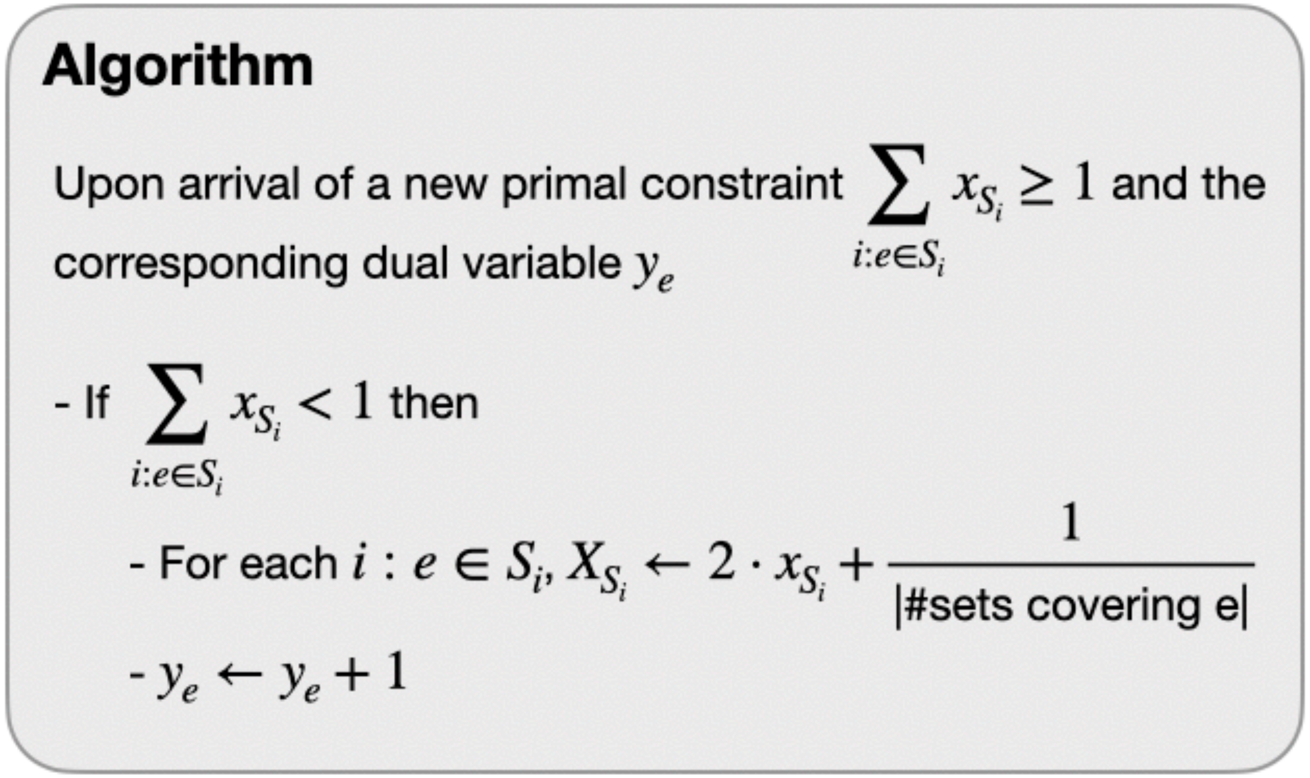
maximize $\sum_e y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis

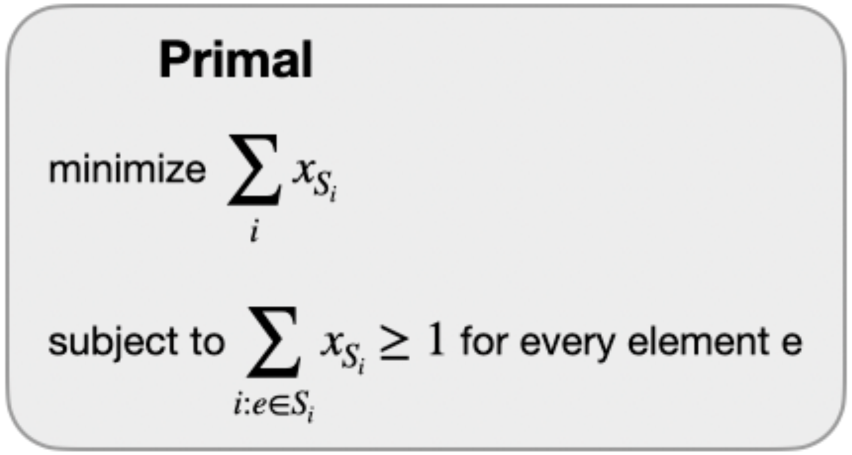
1. At each step the increase of primal is $\sum_{i:e\in S_i}(x_i + 1/|\#\text{sets covering e}|) \leq 2$ whereas increase in dual is 1

2. $y/\log(m)$ is a feasible dual solution:
    - every time a y_e variable is updated in a constraint $\sum_{e\in S_i} y_e \leq 1$
    - The variable x_{S_i} is doubled in primal which can happen at most $\log(m)$ times as its starting value is $1/m$



Algorithm

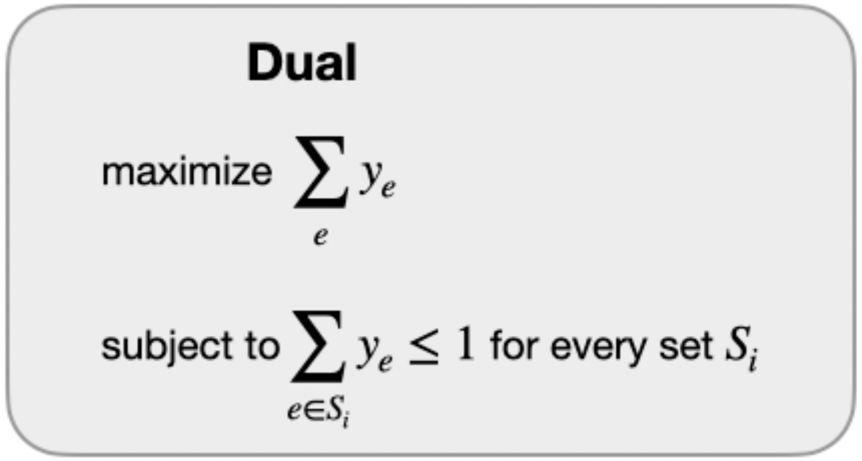
```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the
corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```



Primal

minimize $\sum_{i} x_{S_i}$

subject to $\sum_{i:e\in S_i} x_{S_i} \geq 1$ for every element e



Dual

maximize $\sum_{e} y_e$

subject to $\sum_{e\in S_i} y_e \leq 1$ for every set S_i

Analysis

1. At each step the increase of primal is $\sum_{i:e\in S_i} (x_i + 1/|\text{#sets covering e}|) \leq 2$ whereas increase in dual is 1

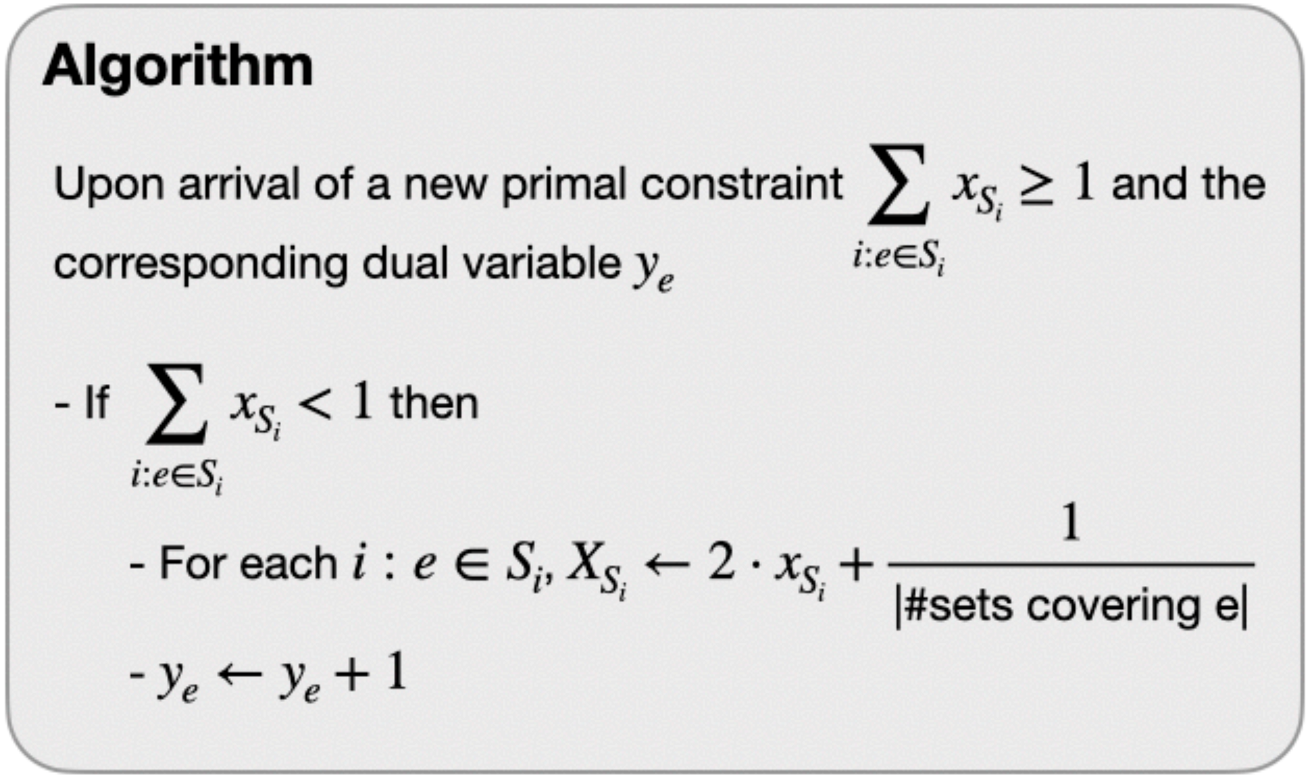
2. $y/\log(m)$ is a feasible dual solution:
    - every time a y_e variable is updated in a constraint $\sum_{e\in S_i} y_e \leq 1$
    - The variable x_{S_i} is doubled in primal which can happen at most $\log(m)$ times as its starting value is $1/m$

1+2 together with LP-duality implies that algorithm is $O(\log m)$-competitive

Making it Learning-Augmented

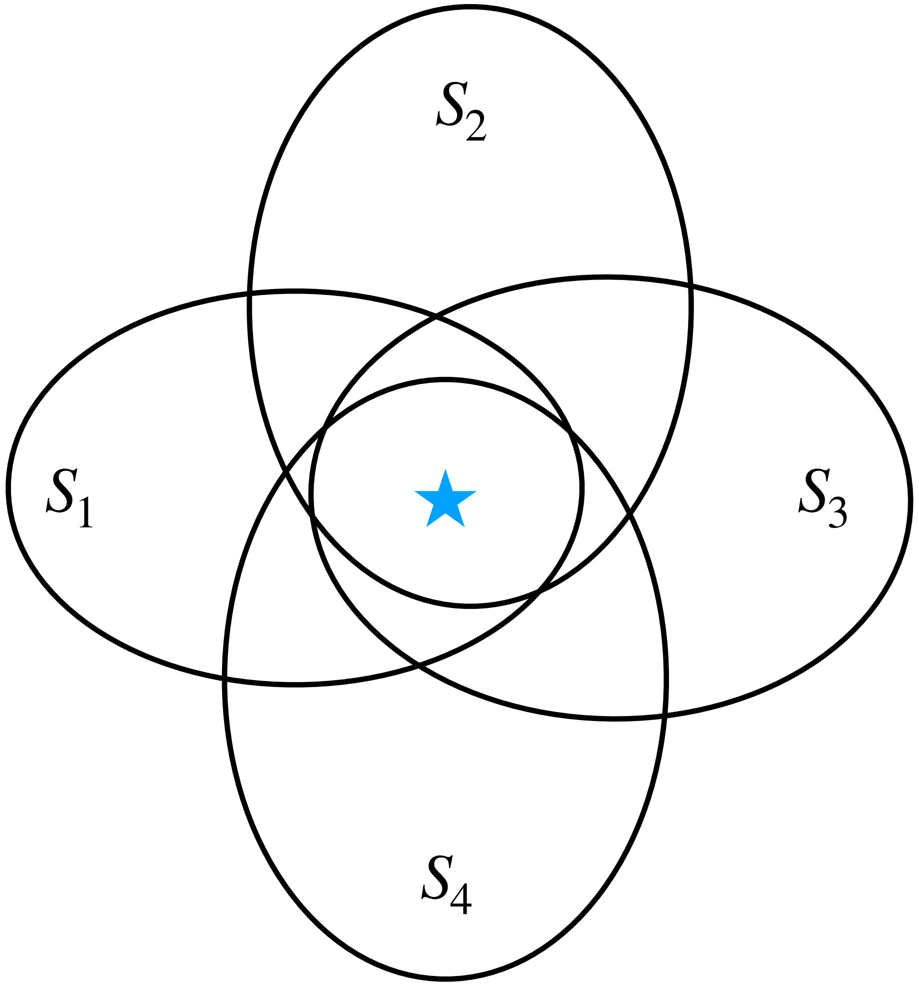
Algorithm

```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i, X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```



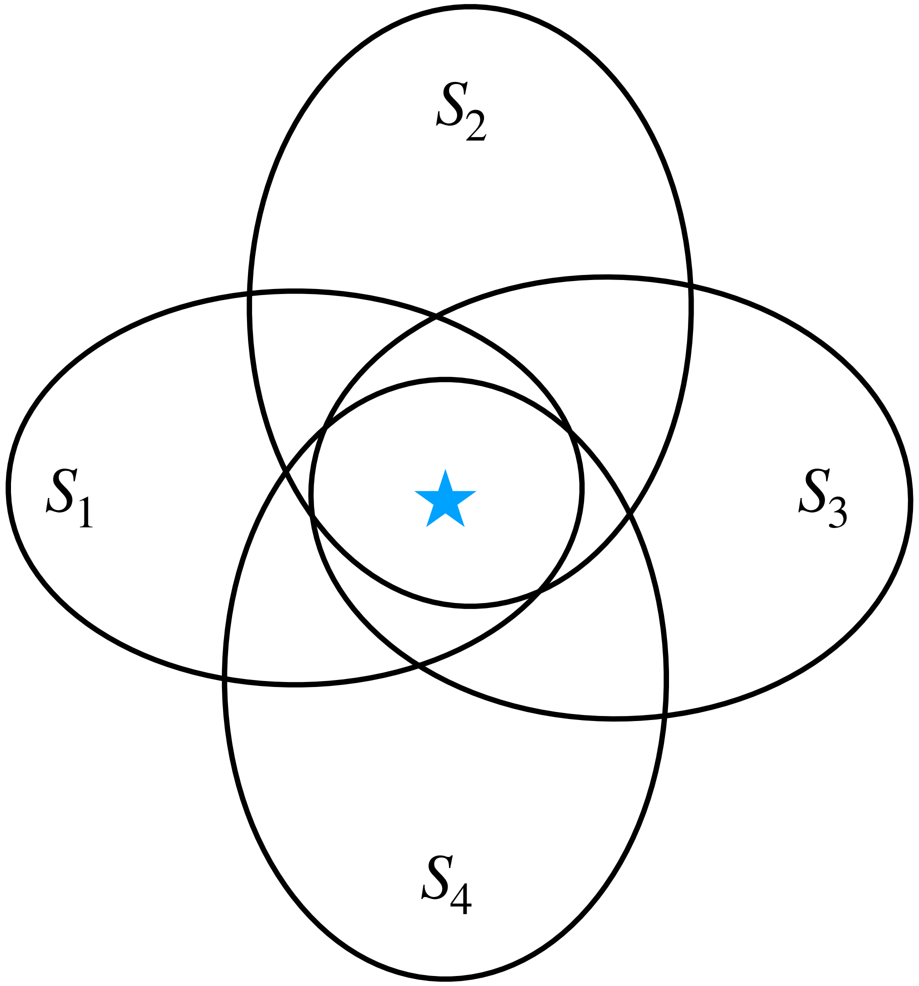
Algorithm

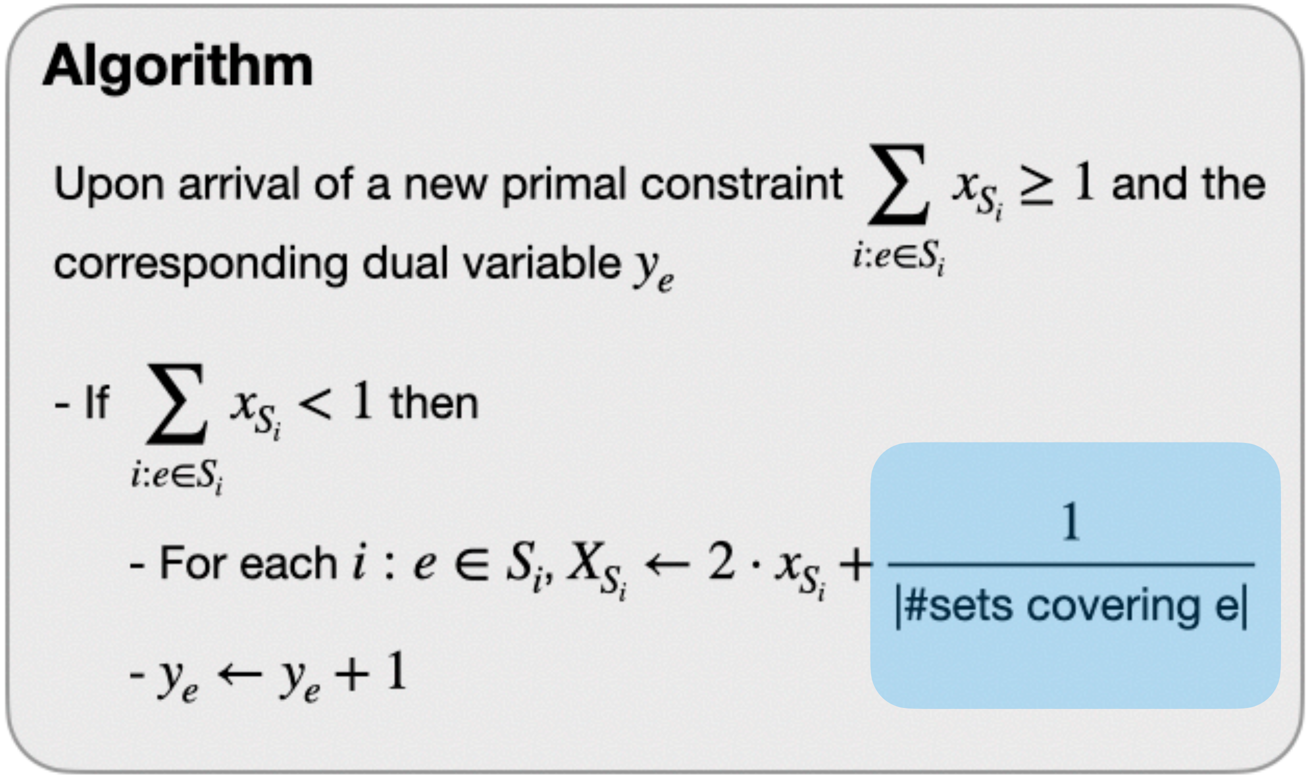
```
Upon arrival of a new primal constraint $\sum_{i:e \in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e \in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```



Algorithm

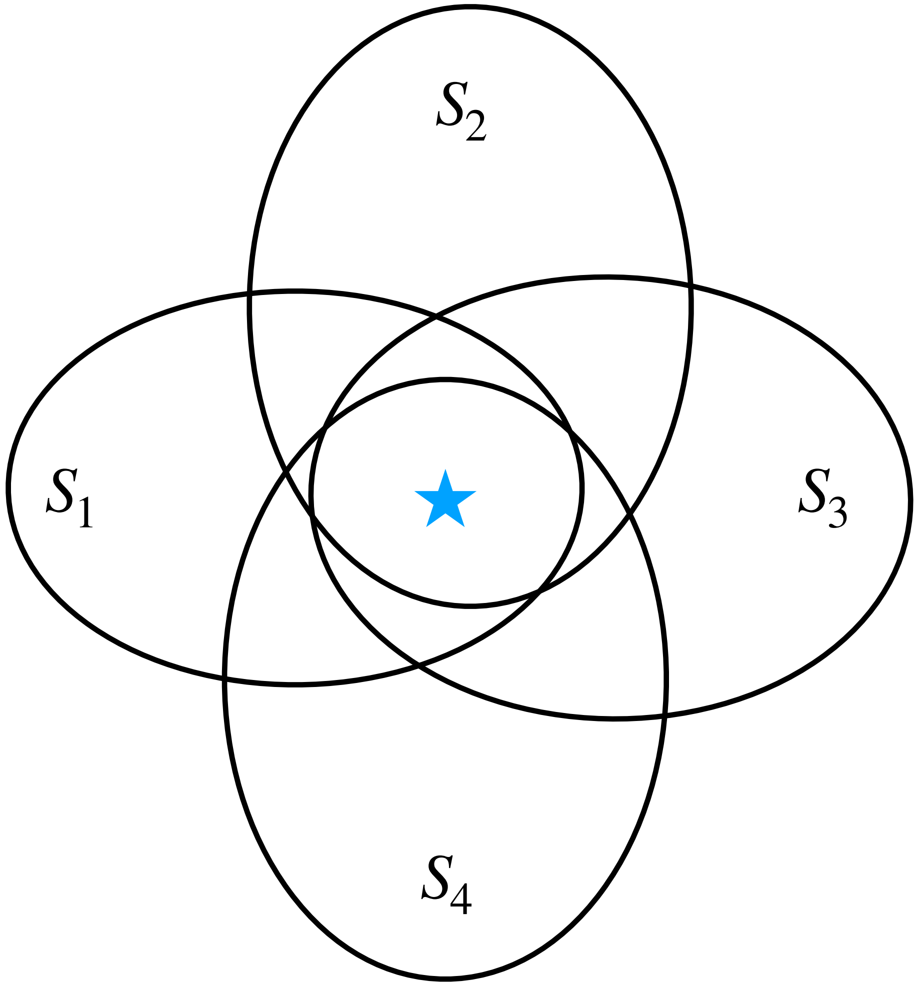
```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```





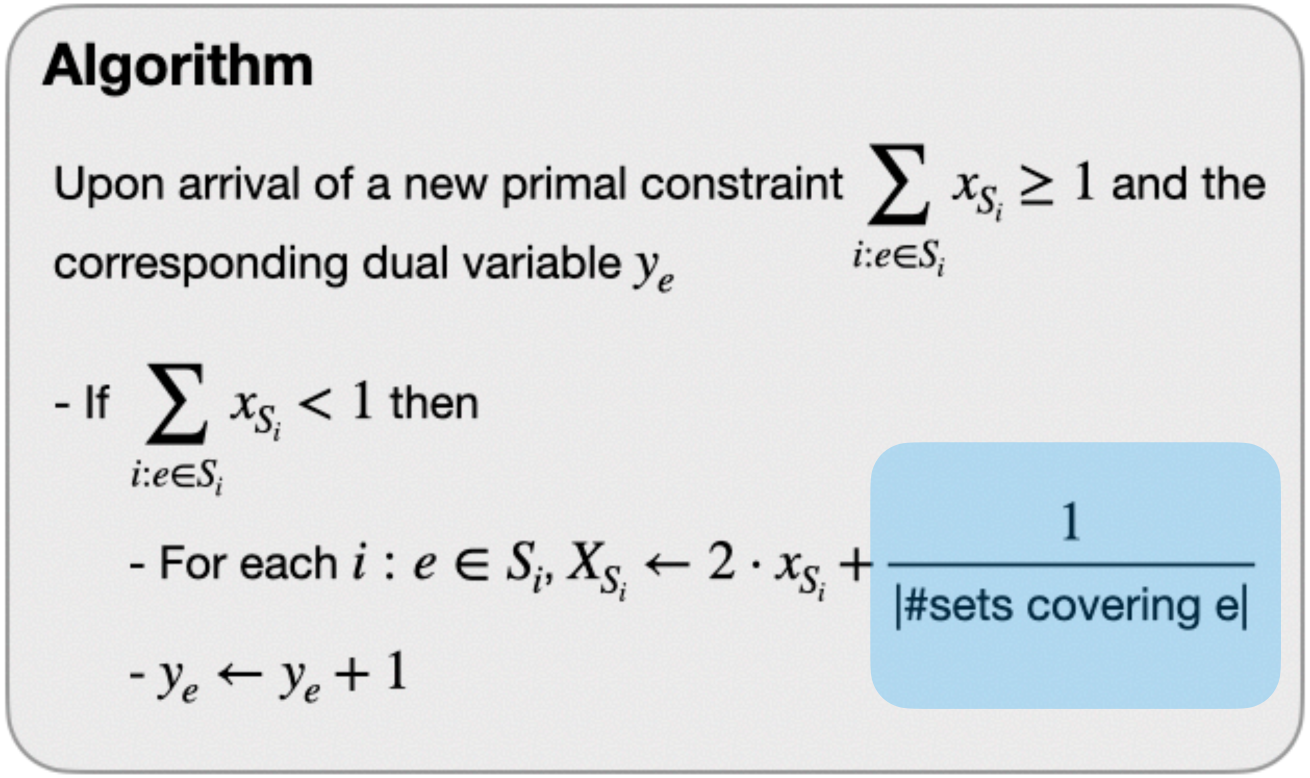
Algorithm

```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{|\#\text{sets covering e}|}$
    - $y_e \leftarrow y_e + 1$
```



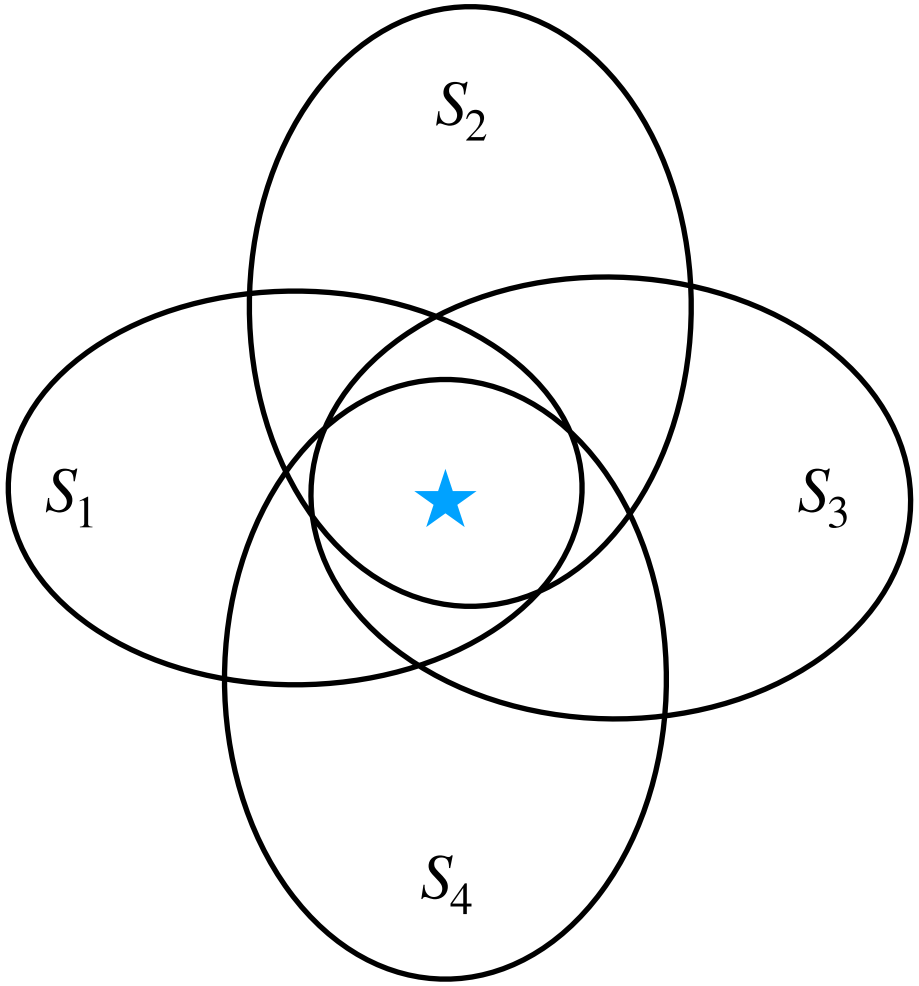
Without prediction all sets are equally likely to be good => hedge uniformly

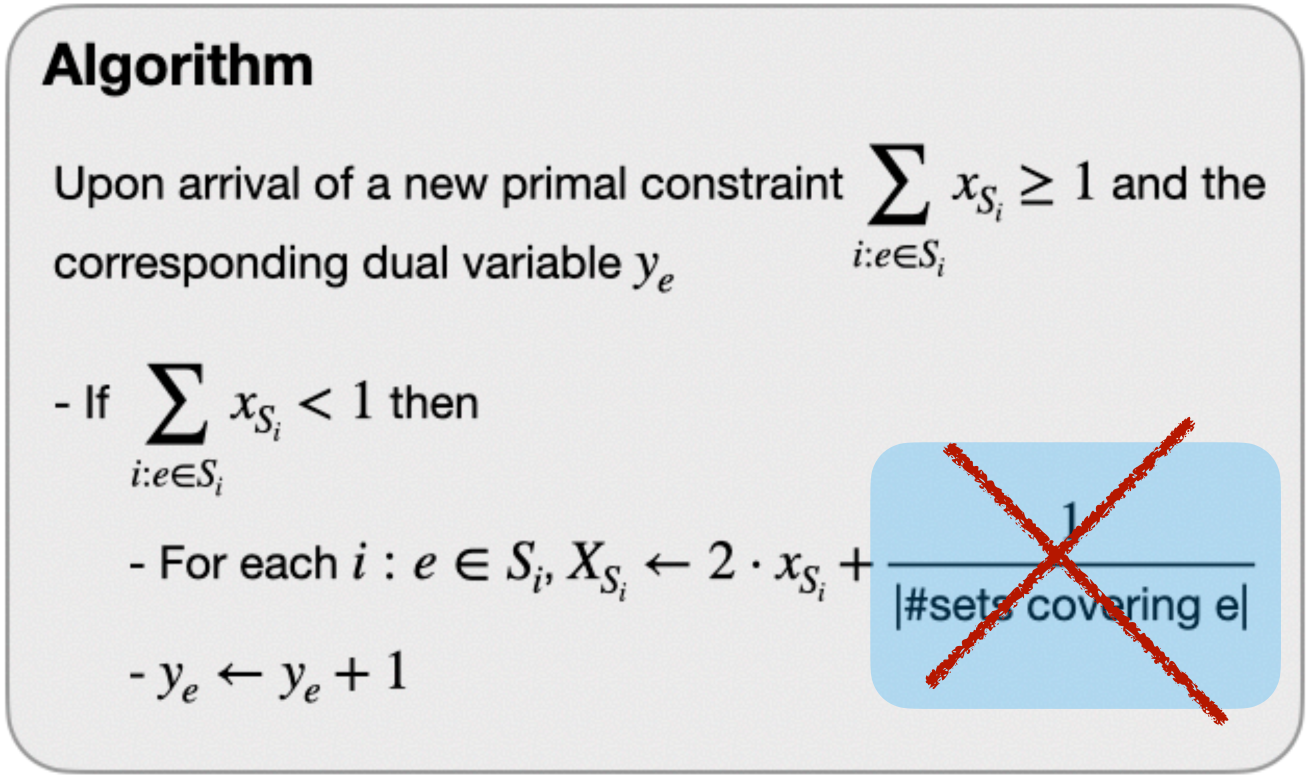
$$x_{S_1} = x_{S_2} = x_{S_3} = x_{S_4} = 1/4$$



Algorithm

```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \frac{1}{\text{|#sets covering e|}}$
    - $y_e \leftarrow y_e + 1$
```





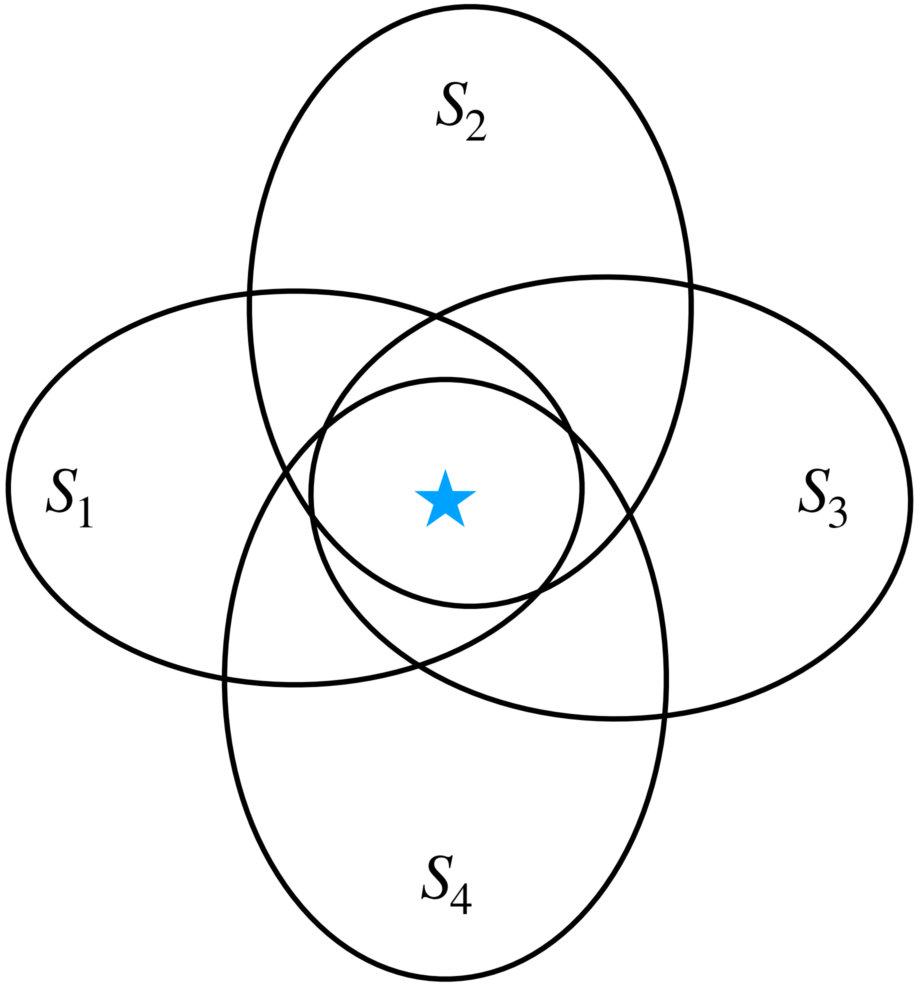
Algorithm

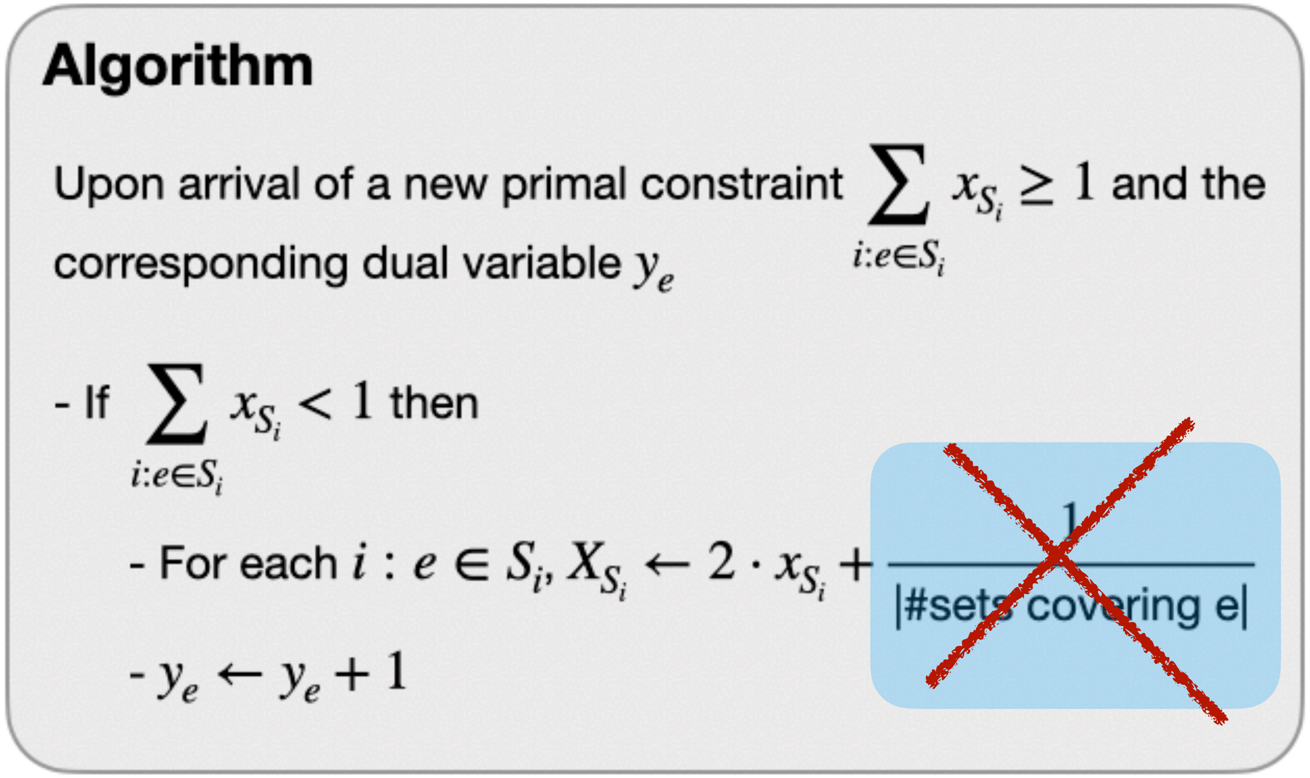
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable y_e

```
- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \cancel{\frac{1}{|\#\text{sets covering e}|}}$
    - $y_e \leftarrow y_e + 1$
```

Learning Augmented

$$\frac{\lambda}{|\#\textbf{sets covering e}|} + \frac{1-\lambda}{|\#\textbf{sets covering e in prediction}|}$$



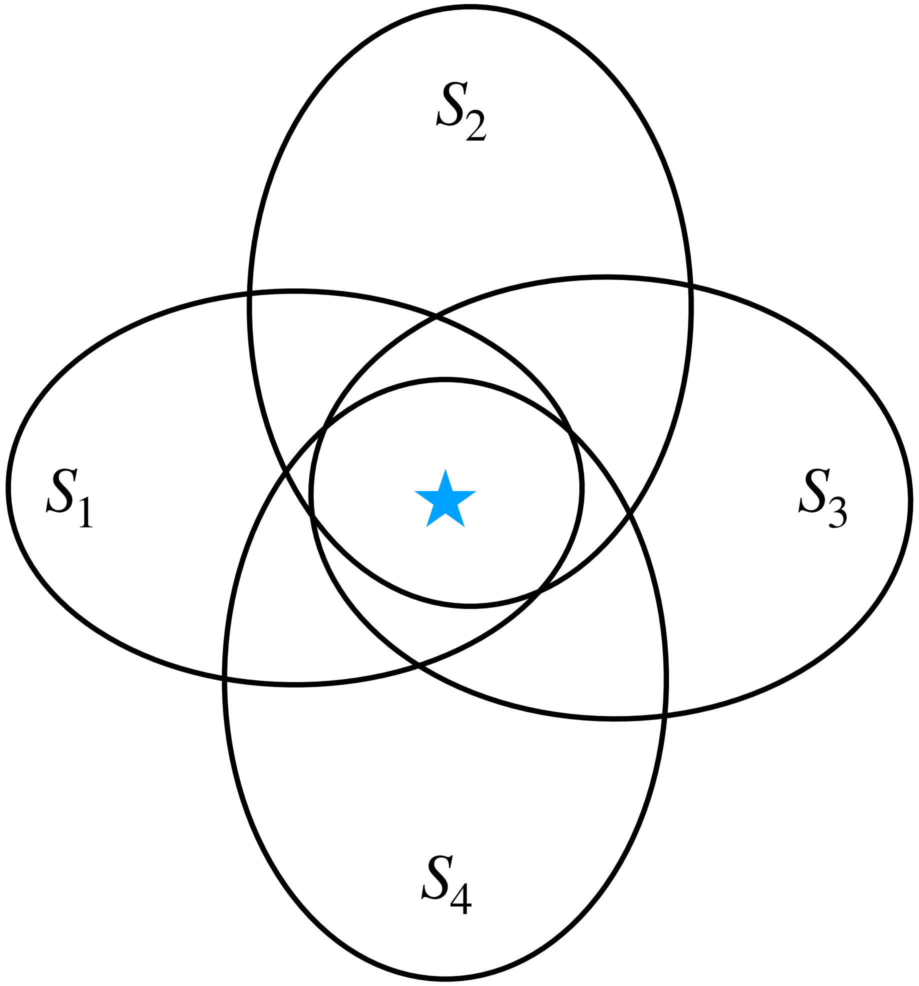


Algorithm

```
Upon arrival of a new primal constraint $\sum_{i:e\in S_i} x_{S_i} \geq 1$ and the corresponding dual variable $y_e$

- If $\sum_{i:e\in S_i} x_{S_i} < 1$ then
    - For each $i : e \in S_i$, $X_{S_i} \leftarrow 2 \cdot x_{S_i} + \cancel{\frac{1}{|\#\text{sets covering e}|}}$
    - $y_e \leftarrow y_e + 1$
```

Learning Augmented

$$\frac{\lambda}{|\#\textbf{sets covering e}|} + \frac{1-\lambda}{|\#\textbf{sets covering e in prediction}|}$$



With prediction, say S_3, should increase that variable more aggressively depending on our trust $\lambda = [0,1]$

$$x_{S_1} = x_{S_2} = x_{S_4} = \lambda/4$$

$$x_{S_3} = \lambda/4 + 1 - \lambda$$

Analysis and guarantees

Analysis and guarantees

Good prediction : $O\left(\frac{1}{1-\lambda}\right)$ competitive

proof via a charging argument +
increase of correct primal variables >> increase of incorrect primal variables

Analysis and guarantees

Good prediction : $O\left(\frac{1}{1-\lambda}\right)$ competitive

proof via a charging argument +
increase of correct primal variables >> increase of incorrect primal variables

Bad prediction : $O\left(\log\frac{m}{\lambda}\right)$ competitive

proof via a primal-dual argument essentially **the same proof** as in the purely online case

Analysis and guarantees

<u>Good prediction</u> : $O\left(\frac{1}{1-\lambda}\right)$ competitive

proof via a charging argument +
increase of correct primal variables >> increase of incorrect primal variables

<u>Bad prediction</u> : $O\left(\log\frac{m}{\lambda}\right)$ competitive

🆚 $O\left(\log m\right)$ competitive
with <u>no prediction</u>

proof via a primal-dual argument essentially **the same proof** as in the purely online case

Analysis and guarantees

Good prediction : $O\left(\frac{1}{1-\lambda}\right)$ competitive

proof via a charging argument +
increase of correct primal variables >> increase of incorrect primal variables

Bad prediction : $O\left(\log\frac{m}{\lambda}\right)$ competitive

🆚 $O\left(\log m\right)$ competitive
with no prediction

proof via a primal-dual argument essentially **the same proof** as in the purely online case

PDLA for Online set cover:

Analysis and guarantees

Good prediction : $O\left(\frac{1}{1-\lambda}\right)$ competitive

proof via a charging argument +
increase of correct primal variables >> increase of incorrect primal variables

Bad prediction : $O\left(\log \frac{m}{\lambda}\right)$ competitive

🆚 $O\left(\log m\right)$ competitive with no prediction

proof via a primal-dual argument essentially **the same proof** as in the purely online case

PDLA for Online set cover: ✅

PDLA

General recipe

PDLA

General recipe

- formulate the LP relaxation of the problem
- solve the problem using the Primal-Dual method
- tweak the rate to which primal variables increase to incorporate predictions

PDLA

General recipe

- formulate the LP relaxation of the problem
- solve the problem using the Primal-Dual method
- tweak the rate to which primal variables increase to incorporate predictions

Simple analysis

Consistency via a charging argument

Robustness mimicking the original PD method proof

PDLA

General recipe

- formulate the LP relaxation of the problem
- solve the problem using the Primal-Dual method
- tweak the rate to which primal variables increase to incorporate predictions

Simple analysis

Consistency via a charging argument

Robustness mimicking the original PD method proof

Widely applicable

Ski rental

Bahncard

TCP-ack

PDLA

General recipe

- formulate the LP relaxation of the problem
- solve the problem using the Primal-Dual method
- tweak the rate to which primal variables increase to incorporate predictions

Simple analysis

Consistency via a charging argument

Robustness mimicking the original PD method proof

Widely applicable

Ski rental

Bahncard

TCP-ack

Easy to implement (TCP-ack)

Good prediction: beat online algorithms

Bad prediction: maintain robustness

Outline

- Learning-augmented online algorithms
- Case study: set cover
- **Instantiating PDLA for other problems**
- Future directions

Ski Rental

Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1) \cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1) \cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

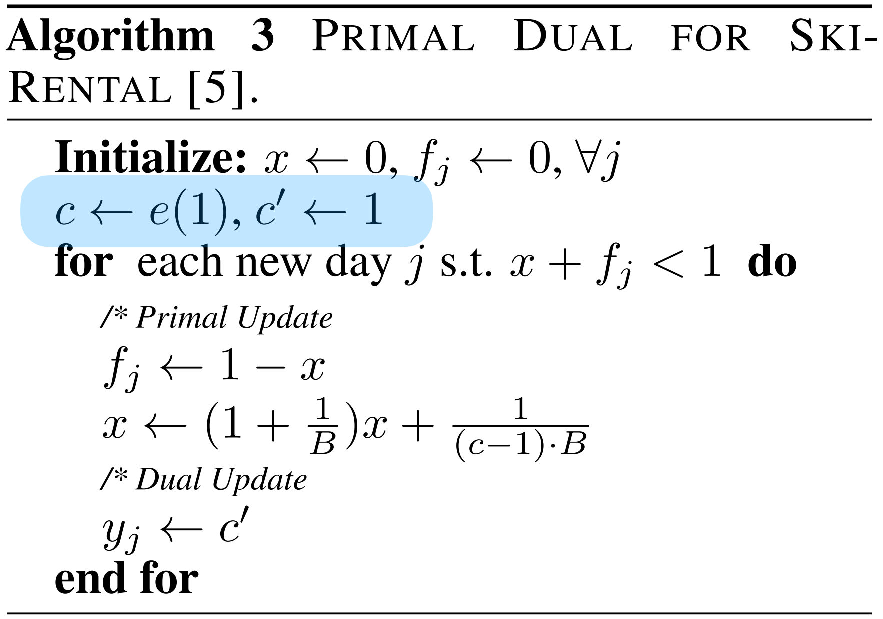
Algorithm 3 Primal Dual for Ski-Rental [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

⇒

Algorithm 4 PDLA for Ski-Rental.

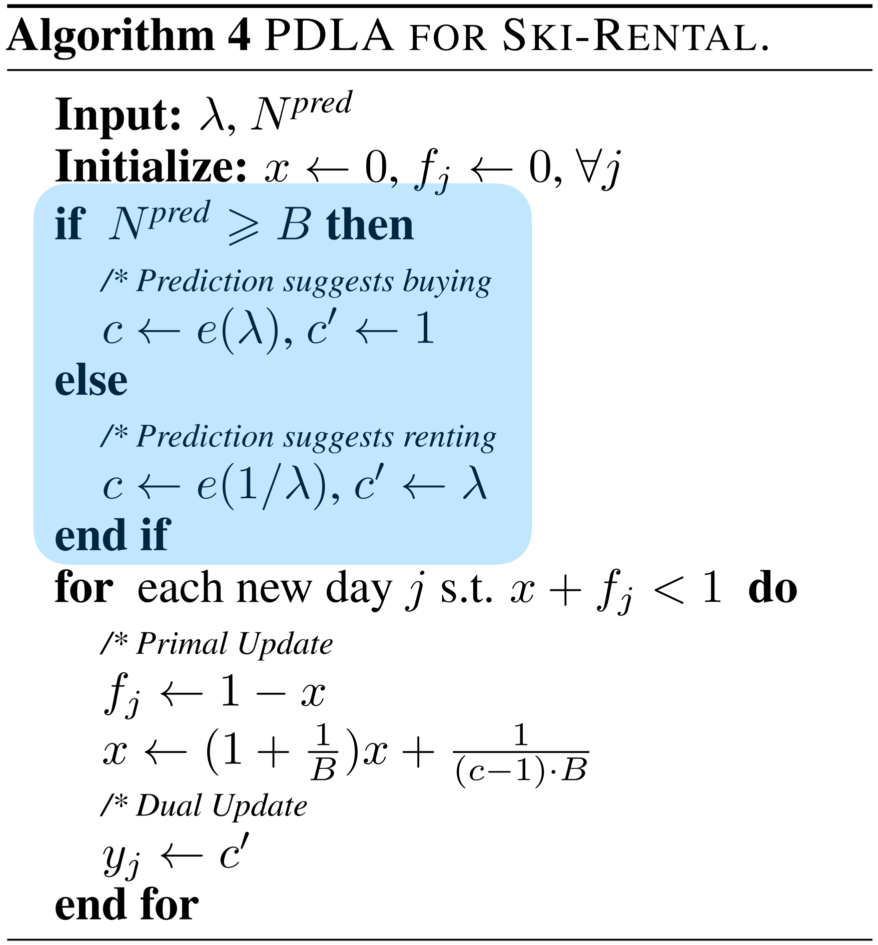
```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda)$, $c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```



Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1) \cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

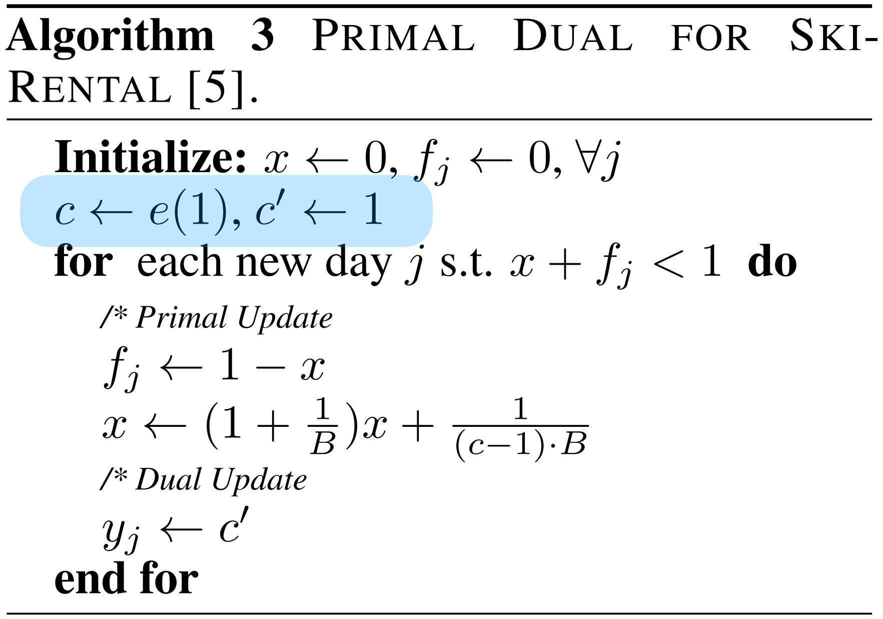
$\Rightarrow$



Algorithm 4 PDLA FOR SKI-RENTAL.

```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda)$, $c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1) \cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

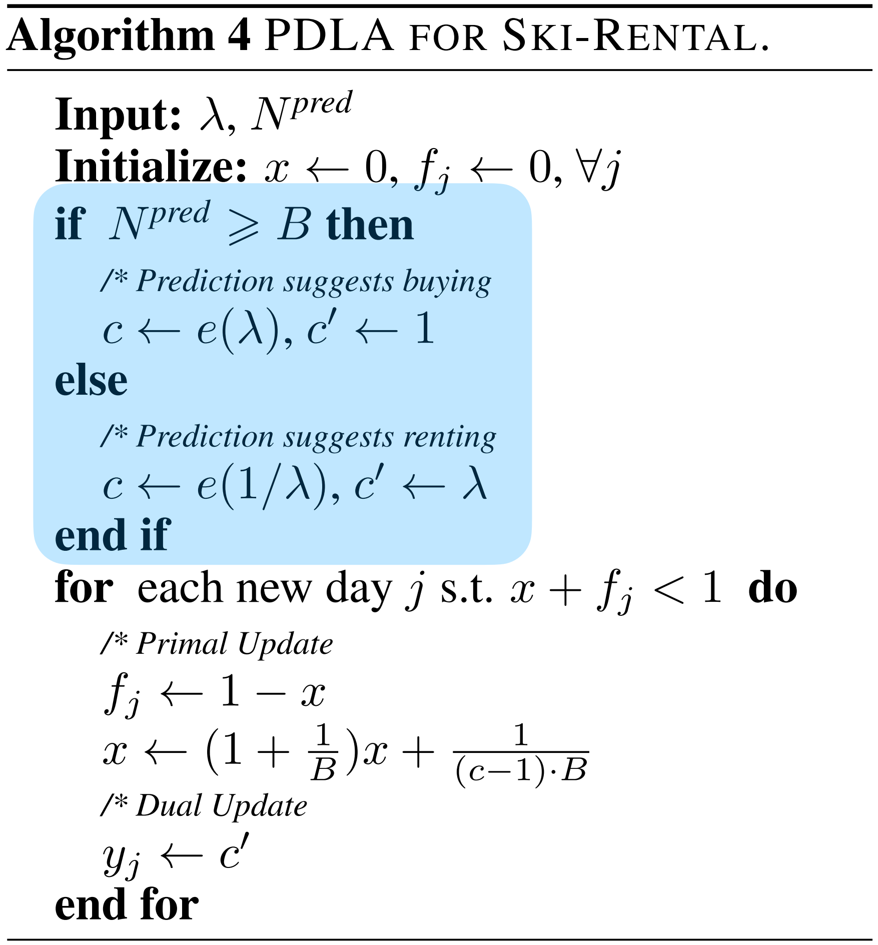
- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$
- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$



Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

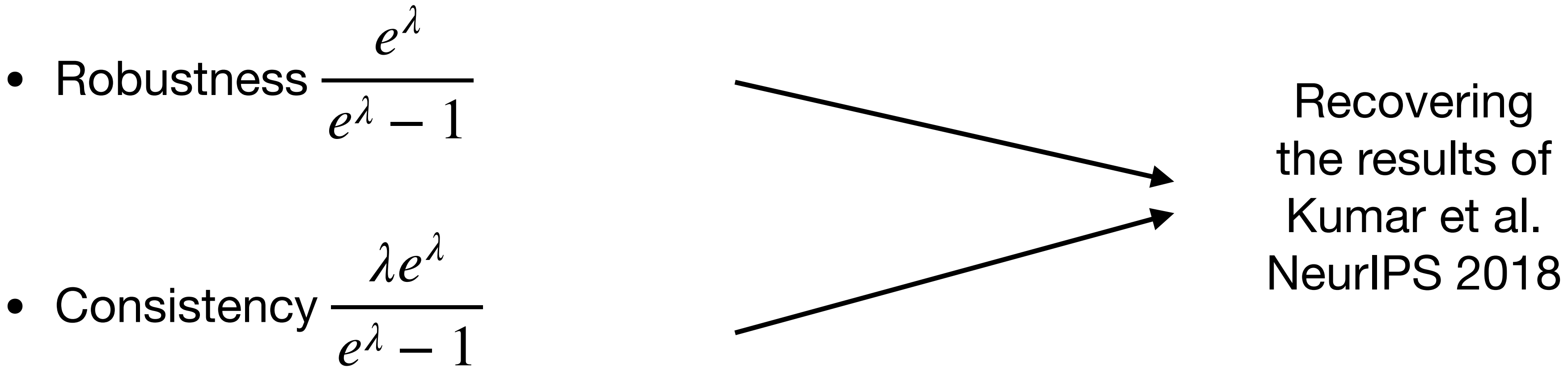
```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

⇒



Algorithm 4 PDLA FOR SKI-RENTAL.

```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda)$, $c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```



- Robustness $\frac{e^{\lambda}}{e^{\lambda} - 1}$
- Consistency $\frac{\lambda e^{\lambda}}{e^{\lambda} - 1}$

Recovering the results of Kumar et al. NeurIPS 2018

Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

```
Initialize: $x \leftarrow 0, f_j \leftarrow 0, \forall j$
$c \leftarrow e(1), c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

⇒

Algorithm 4 PDLA FOR SKI-RENTAL.

```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0, f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda), c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda), c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

- Robustness $\dfrac{e^\lambda}{e^\lambda - 1}$
- Consistency $\dfrac{\lambda e^\lambda}{e^\lambda - 1}$

Recovering the results of Kumar et al. NeurIPS 2018

Best possible robustness-consistency tradeoff

Algorithm 3 Primal Dual for Ski-Rental [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

$\Rightarrow$

Algorithm 4 PDLA for Ski-Rental.

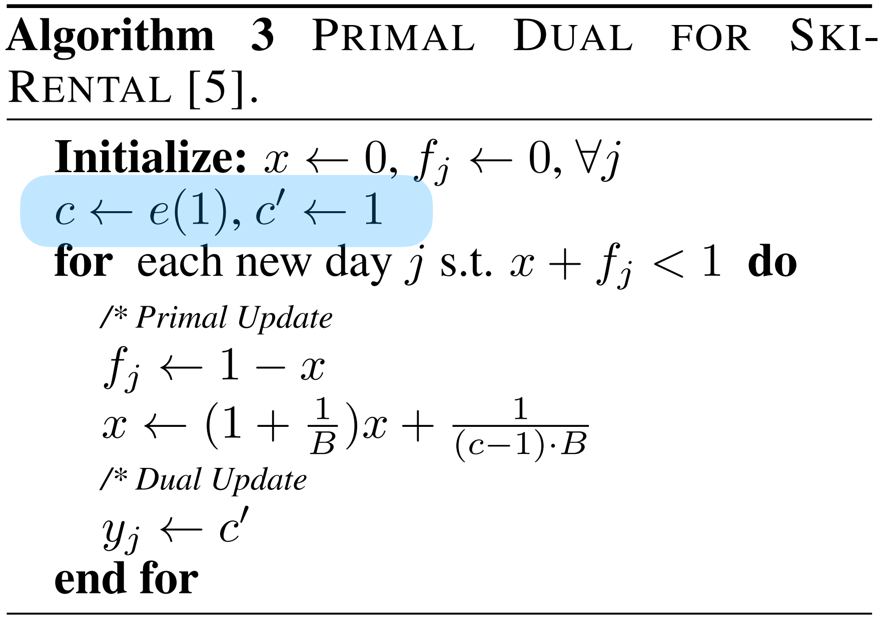
```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda)$, $c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$
- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$

Recovering the results of Kumar et al. NeurIPS 2018

Best possible robustness-consistency tradeoff

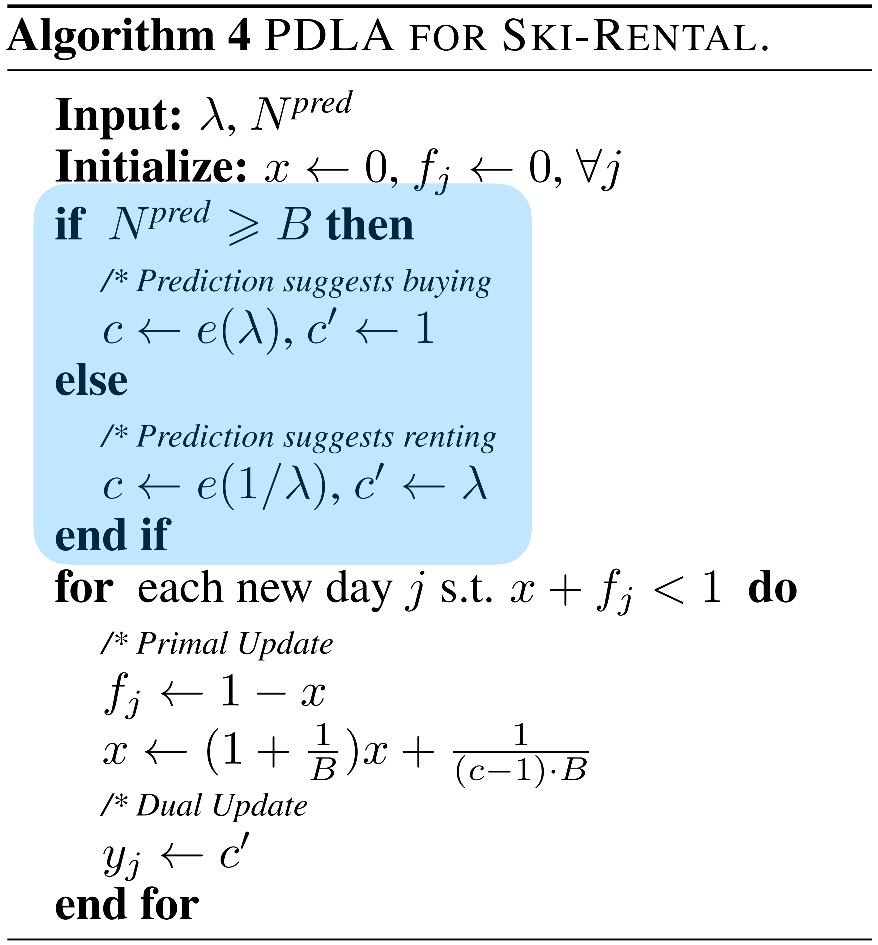
PDLA for Ski rental:



Algorithm 3 PRIMAL DUAL FOR SKI-RENTAL [5].

```
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
$c \leftarrow e(1)$, $c' \leftarrow 1$
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

⇒



Algorithm 4 PDLA FOR SKI-RENTAL.

```
Input: $\lambda$, $N^{pred}$
Initialize: $x \leftarrow 0$, $f_j \leftarrow 0, \forall j$
if $N^{pred} \geqslant B$ then
    /* Prediction suggests buying
    $c \leftarrow e(\lambda)$, $c' \leftarrow 1$
else
    /* Prediction suggests renting
    $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda$
end if
for each new day $j$ s.t. $x + f_j < 1$ do
    /* Primal Update
    $f_j \leftarrow 1 - x$
    $x \leftarrow (1 + \frac{1}{B})x + \frac{1}{(c-1)\cdot B}$
    /* Dual Update
    $y_j \leftarrow c'$
end for
```

- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$

- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$

Recovering the results of Kumar et al. NeurIPS 2018

Best possible robustness-consistency tradeoff

PDLA for Ski rental: ✅

TCP Acknowledgement

TCP-ack problem definition:

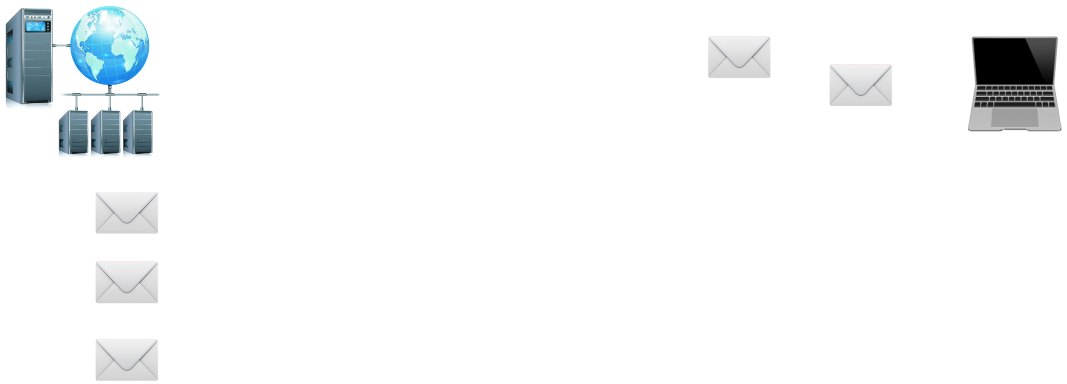
TCP-ack problem definition: A server receives a stream of packets

TCP-ack problem definition: A server receives a stream of packets

TCP-ack problem definition: A server receives a stream of packets

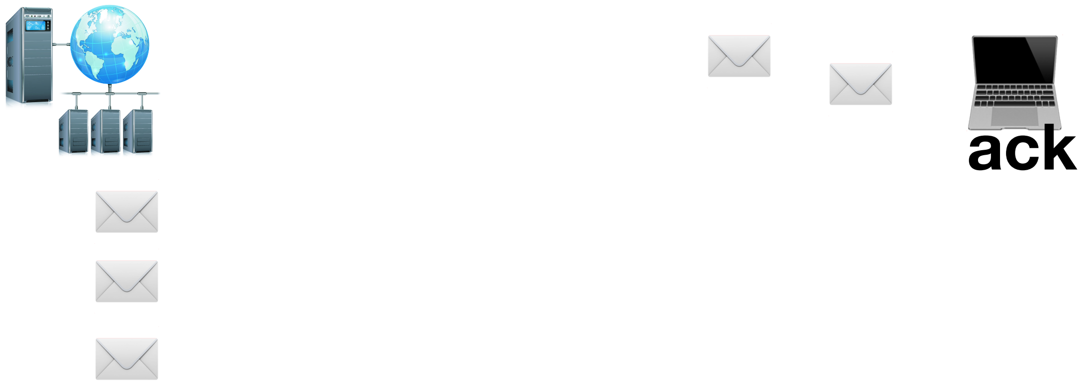
TCP-ack problem definition: A server receives a stream of packets

The server sends an ack to the sender immediately

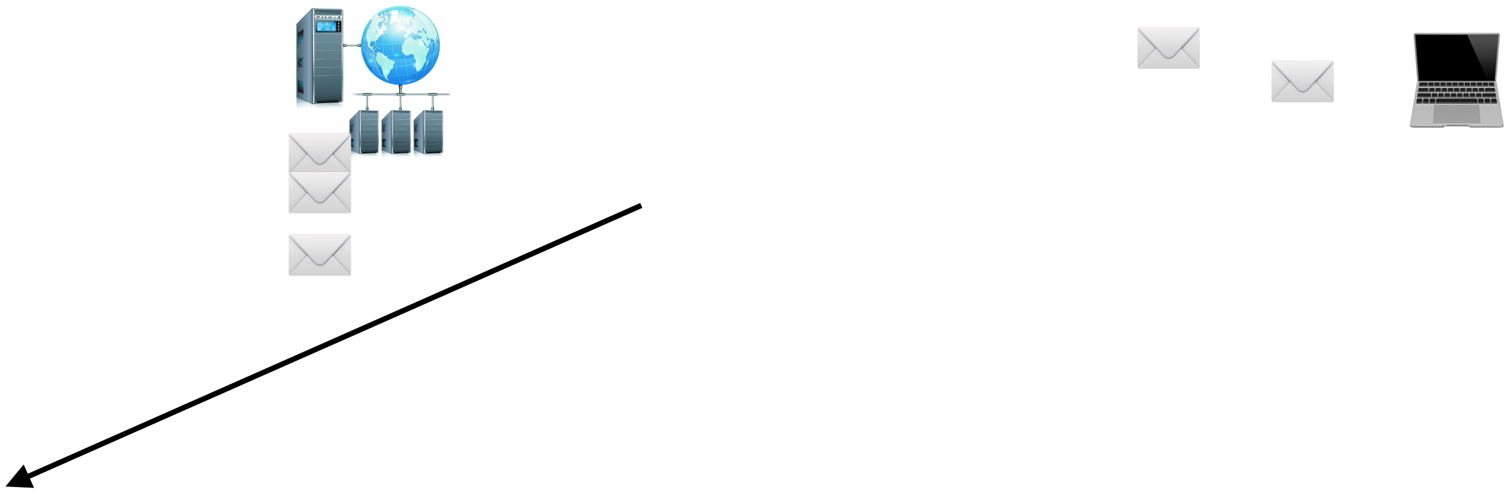


TCP-ack problem definition: A server receives a stream of packets

The server sends an ack to the sender immediately



TCP-ack problem definition: A server receives a stream of packets



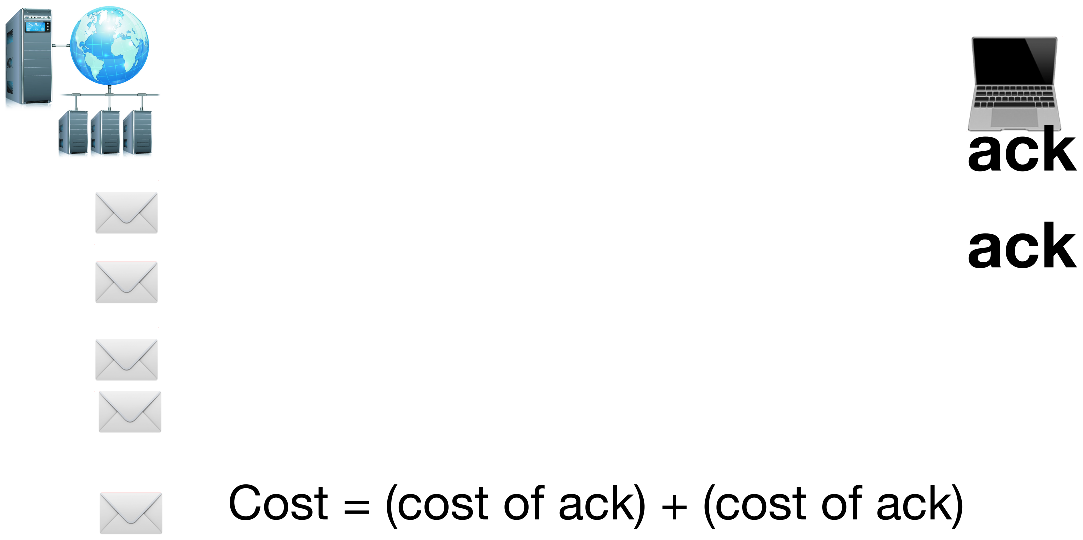
The server sends an ack to the sender immediately

TCP-ack problem definition: A server receives a stream of packets

The server sends an ack to the sender immediately

TCP-ack problem definition: A server receives a stream of packets

The server sends an ack to the sender immediately

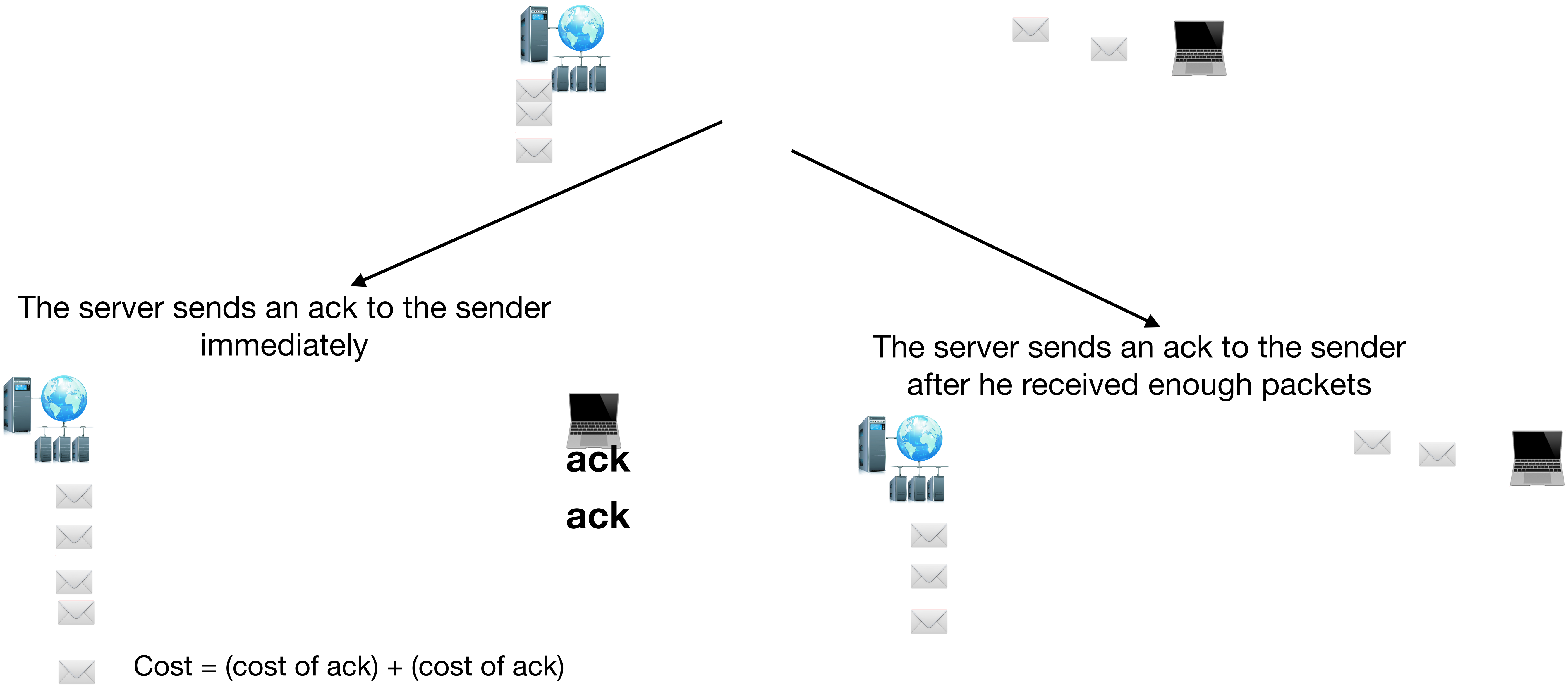


ack

ack

Cost = (cost of ack) + (cost of ack)

TCP-ack problem definition: A server receives a stream of packets



The server sends an ack to the sender immediately

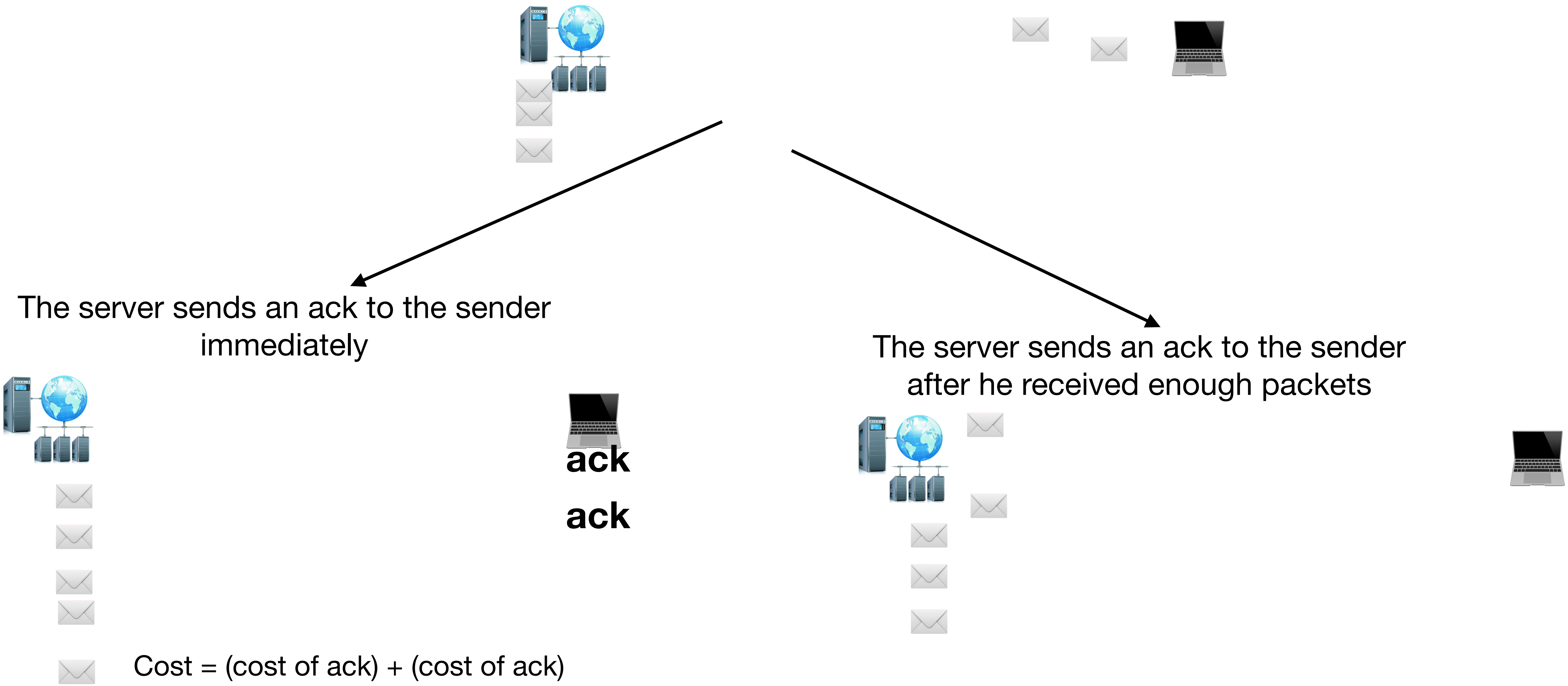
ack

ack

Cost = (cost of ack) + (cost of ack)

The server sends an ack to the sender after he received enough packets

TCP-ack problem definition: A server receives a stream of packets

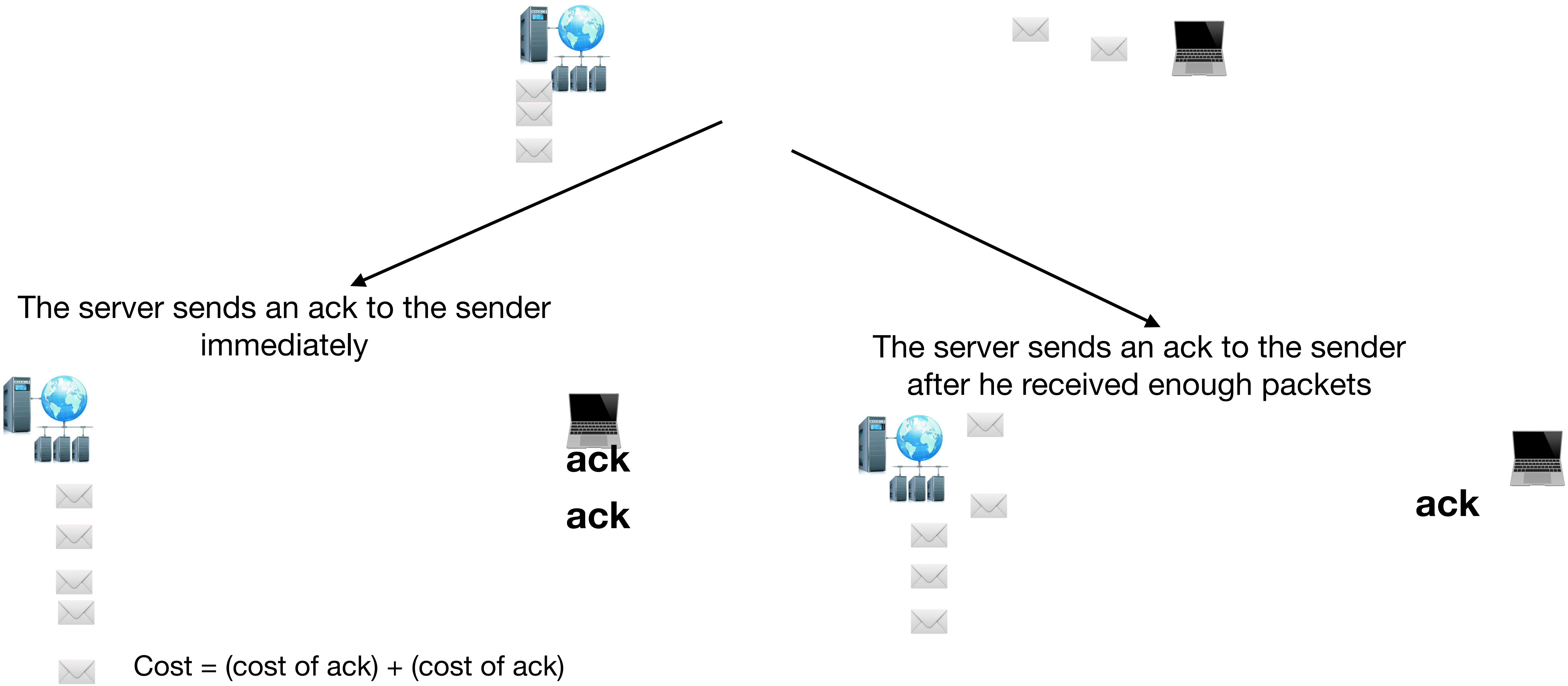


The server sends an ack to the sender immediately

Cost = (cost of ack) + (cost of ack)

The server sends an ack to the sender after he received enough packets

TCP-ack problem definition: A server receives a stream of packets

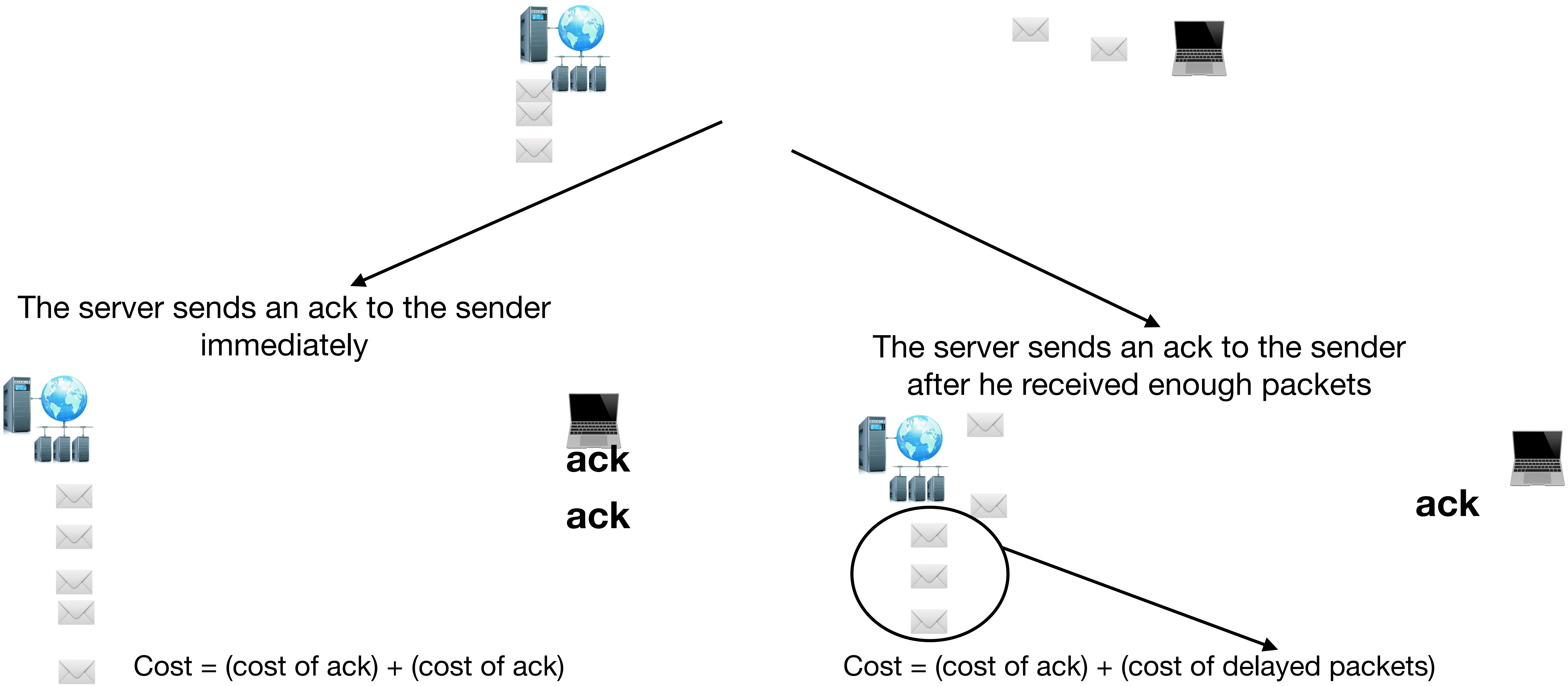


The server sends an ack to the sender immediately

Cost = (cost of ack) + (cost of ack)

The server sends an ack to the sender after he received enough packets

TCP-ack problem definition: A server receives a stream of packets



The server sends an ack to the sender immediately

ack

ack

Cost = (cost of ack) + (cost of ack)

The server sends an ack to the sender after he received enough packets

ack

Cost = (cost of ack) + (cost of delayed packets)

Algorithm 5 Primal Dual method for TCP acknowledgement [5].

```
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        $c \leftarrow e(1), c' \leftarrow 1/d$
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

Algorithm 5 Primal Dual method for TCP acknowledgement [5].

```
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
  for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
    $c \leftarrow e(1), c' \leftarrow 1/d$
    /* Primal Update
    $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
    $x_t \leftarrow x_t + \frac{1}{d} \cdot \left( \sum_{k=t(j)}^{t} x_k + \frac{1}{c-1} \right)$
    /* Dual Update
    $y_{jt} \leftarrow c'$
  end for
end for
```

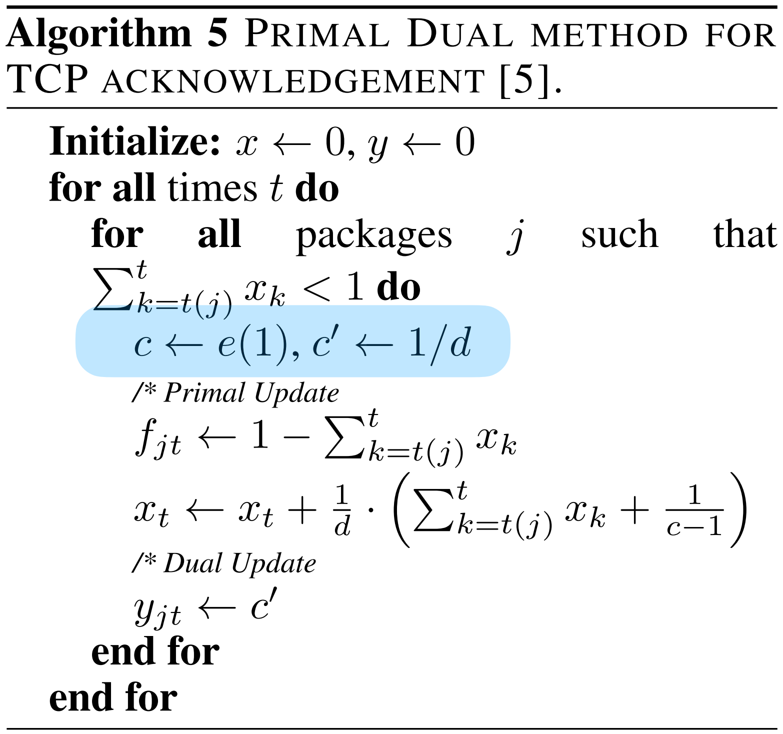
Algorithm 5 PRIMAL DUAL METHOD FOR TCP ACKNOWLEDGEMENT [5].

```
Initialize: $x \leftarrow 0$, $y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        $c \leftarrow e(1)$, $c' \leftarrow 1/d$
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

⇒

Algorithm 6 PDLA FOR TCP ACKNOWLEDGEMENT

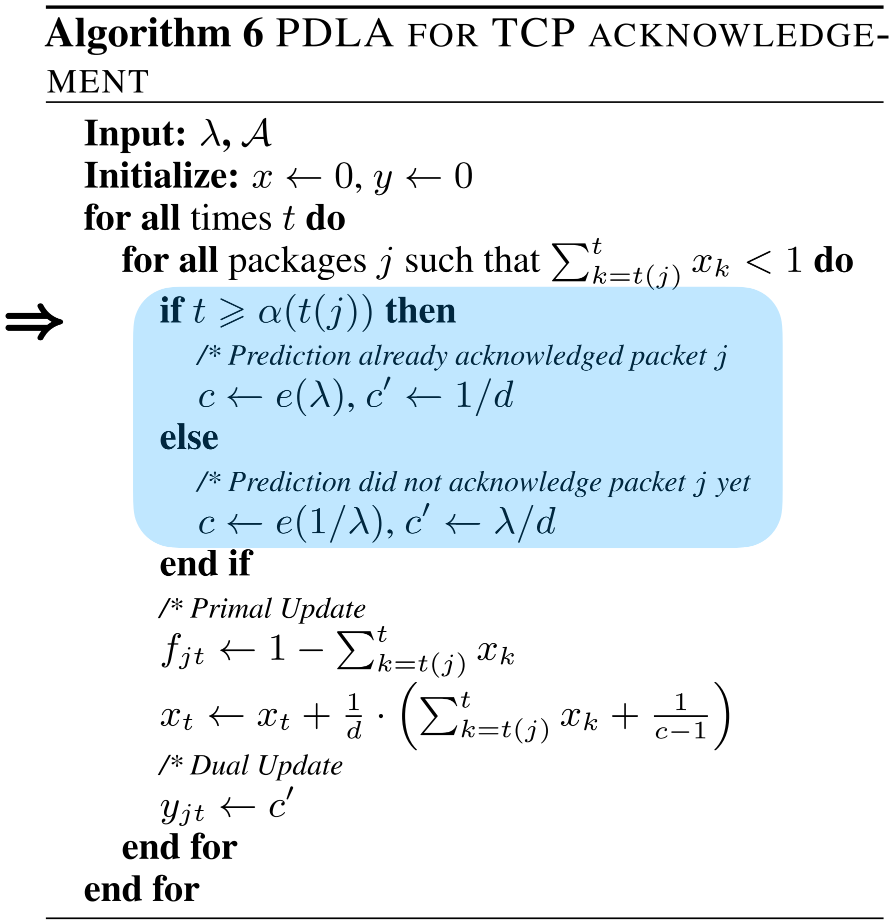
```
Input: $\lambda$, $\mathcal{A}$
Initialize: $x \leftarrow 0$, $y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        if $t \geqslant \alpha(t(j))$ then
            /* Prediction already acknowledged packet $j$
            $c \leftarrow e(\lambda)$, $c' \leftarrow 1/d$
        else
            /* Prediction did not acknowledge packet $j$ yet
            $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda/d$
        end if
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```



Algorithm 5 Primal Dual method for TCP acknowledgement [5].

```
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        $c \leftarrow e(1), c' \leftarrow 1/d$
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

⇒



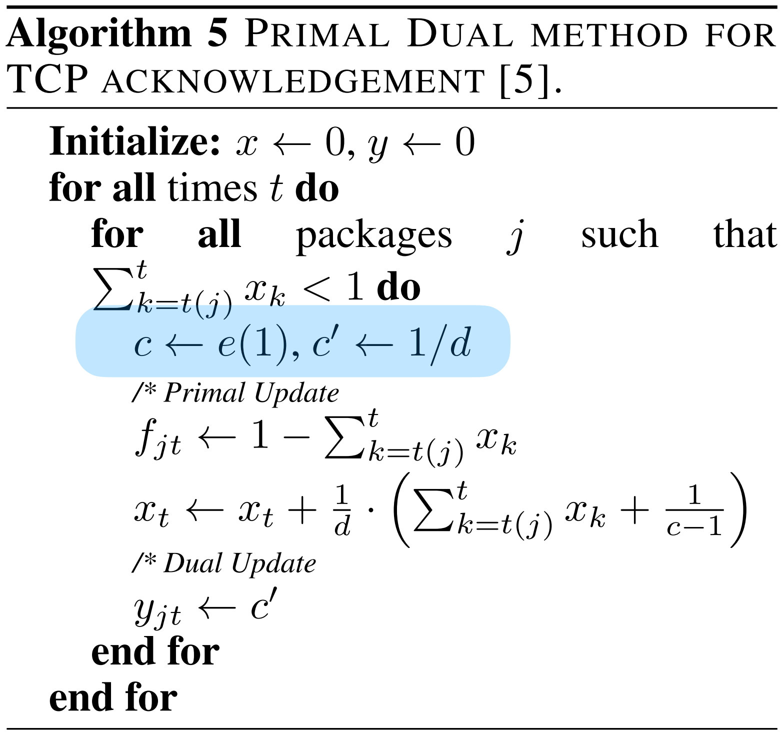
Algorithm 6 PDLA for TCP acknowledgement

```
Input: $\lambda$, $\mathcal{A}$
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        if $t \geqslant \alpha(t(j))$ then
            /* Prediction already acknowledged packet $j$
            $c \leftarrow e(\lambda), c' \leftarrow 1/d$
        else
            /* Prediction did not acknowledge packet $j$ yet
            $c \leftarrow e(1/\lambda), c' \leftarrow \lambda/d$
        end if
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$

- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$

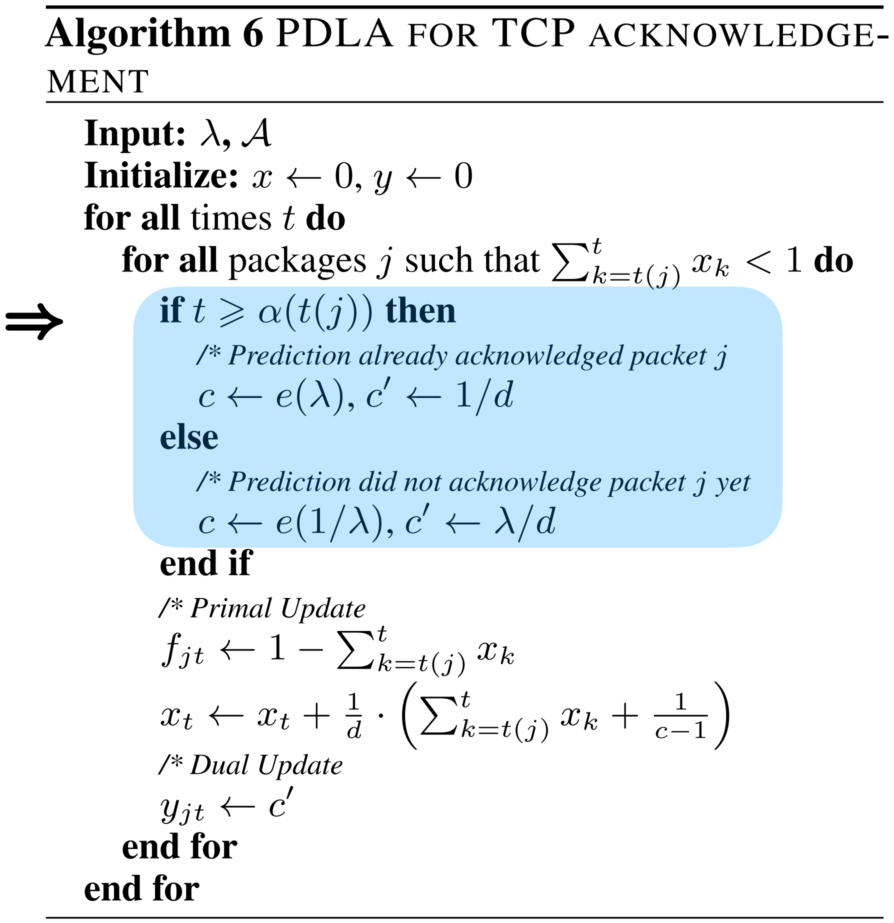
PDLA for TCP Ack:



Algorithm 5 Primal Dual method for TCP acknowledgement [5].

```
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        $c \leftarrow e(1), c' \leftarrow 1/d$
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

⇒

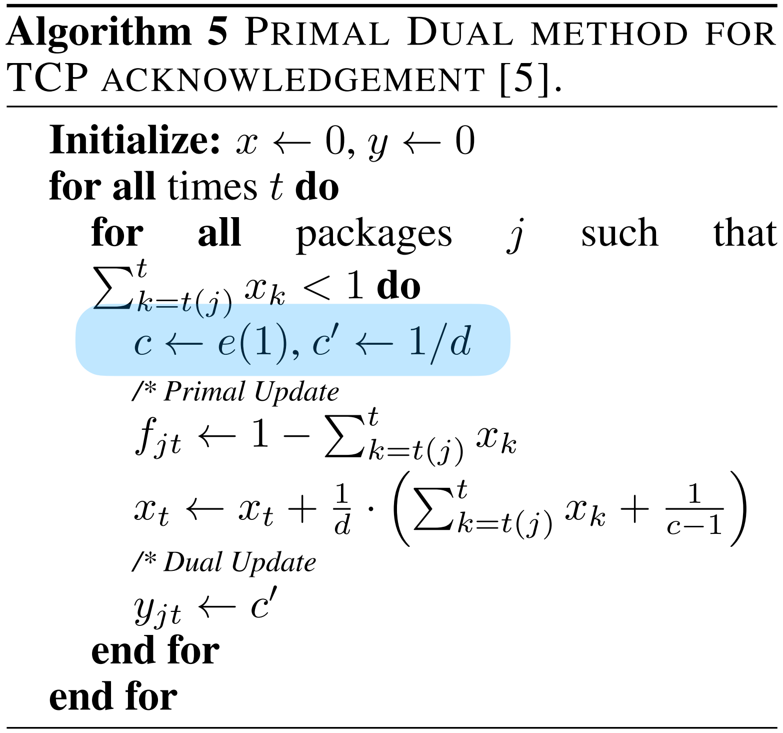


Algorithm 6 PDLA for TCP acknowledgement

```
Input: $\lambda, \mathcal{A}$
Initialize: $x \leftarrow 0, y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        if $t \geqslant \alpha(t(j))$ then
            /* Prediction already acknowledged packet $j$
            $c \leftarrow e(\lambda), c' \leftarrow 1/d$
        else
            /* Prediction did not acknowledge packet $j$ yet
            $c \leftarrow e(1/\lambda), c' \leftarrow \lambda/d$
        end if
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left(\sum_{k=t(j)}^{t} x_k + \frac{1}{c-1}\right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$

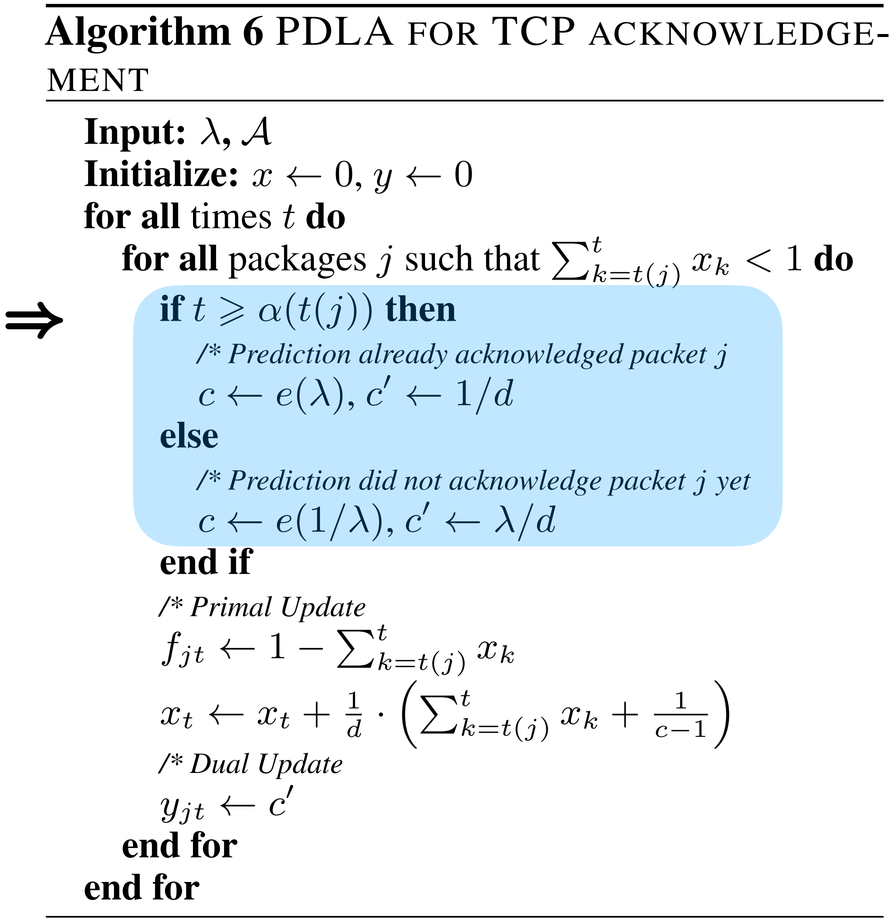
- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$



Algorithm 5 PRIMAL DUAL METHOD FOR TCP ACKNOWLEDGEMENT [5].

```
Initialize: $x \leftarrow 0$, $y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        $c \leftarrow e(1)$, $c' \leftarrow 1/d$
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left( \sum_{k=t(j)}^{t} x_k + \frac{1}{c-1} \right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

$\Rightarrow$



Algorithm 6 PDLA FOR TCP ACKNOWLEDGEMENT

```
Input: $\lambda$, $\mathcal{A}$
Initialize: $x \leftarrow 0$, $y \leftarrow 0$
for all times $t$ do
    for all packages $j$ such that $\sum_{k=t(j)}^{t} x_k < 1$ do
        if $t \geqslant \alpha(t(j))$ then
            /* Prediction already acknowledged packet $j$
            $c \leftarrow e(\lambda)$, $c' \leftarrow 1/d$
        else
            /* Prediction did not acknowledge packet $j$ yet
            $c \leftarrow e(1/\lambda)$, $c' \leftarrow \lambda/d$
        end if
        /* Primal Update
        $f_{jt} \leftarrow 1 - \sum_{k=t(j)}^{t} x_k$
        $x_t \leftarrow x_t + \frac{1}{d} \cdot \left( \sum_{k=t(j)}^{t} x_k + \frac{1}{c-1} \right)$
        /* Dual Update
        $y_{jt} \leftarrow c'$
    end for
end for
```

- Robustness $\dfrac{e^{\lambda}}{e^{\lambda} - 1}$

- Consistency $\dfrac{\lambda e^{\lambda}}{e^{\lambda} - 1}$

PDLA for TCP Ack: ✅

PDLA in Action for TCP Ack

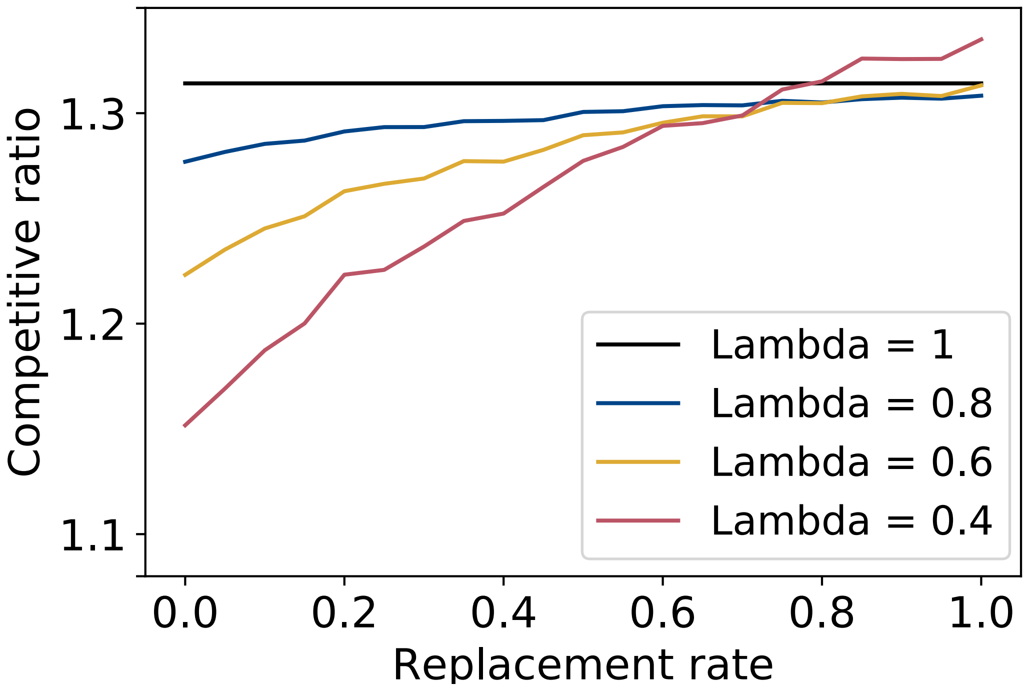
PDLA in Action for TCP Ack

Experimental setting:

PDLA in Action for TCP Ack

Experimental setting:

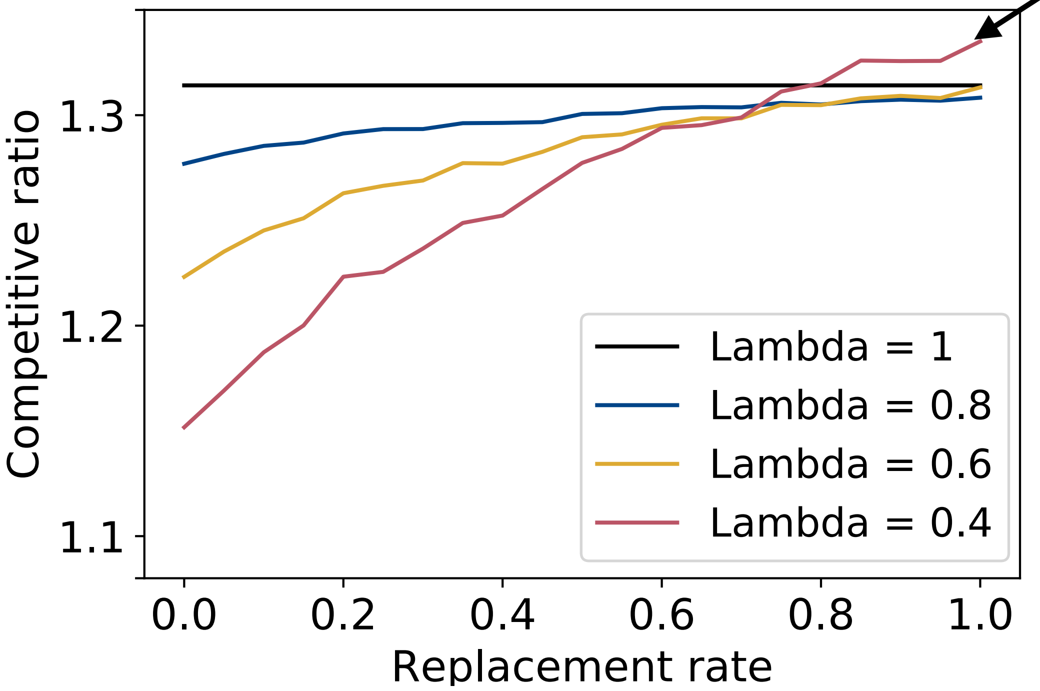
- $I \rightarrow$ number of packets at each time step follows a Lomax distribution
- $I_{pred} \rightarrow$ (perturbed I) at each time step with probability p we delete the packets of the true instance I, and with probability p we add an independent instance



PDLA in Action for TCP Ack

Experimental setting:

- I → number of packets at each time step follows a Lomax distribution
- I_{pred} → (perturbed I) at each time step with probability p we delete the packets of the true instance I, and with probability p we add an independent instance



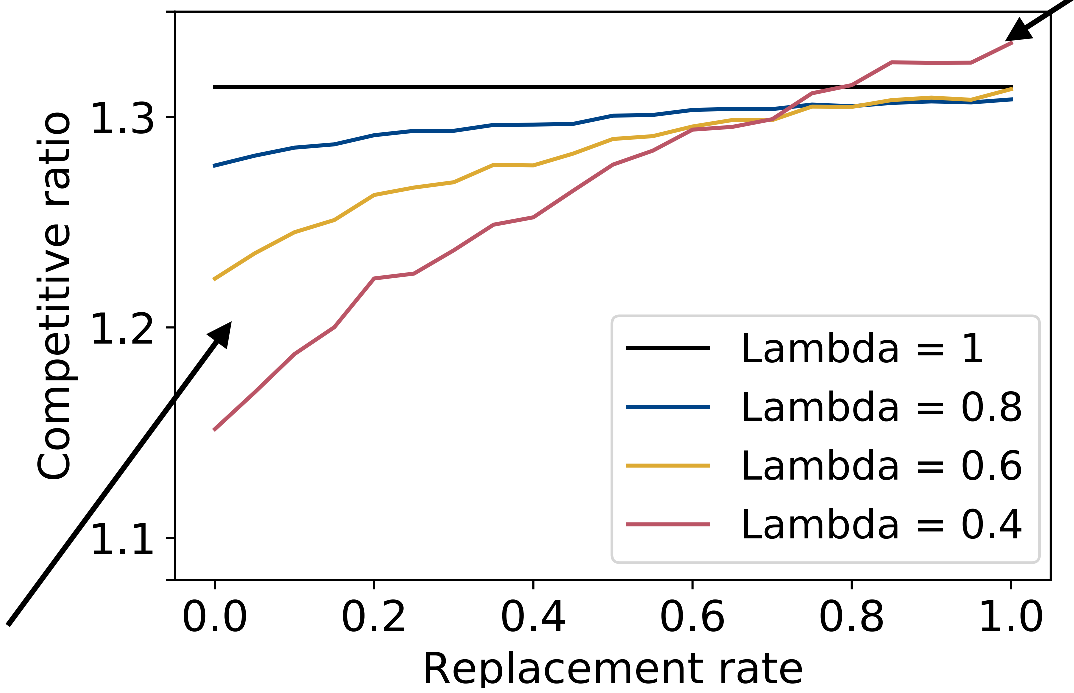
Bad prediction: maintain robustness

PDLA in Action for TCP Ack

Experimental setting:

- $I \rightarrow$ number of packets at each time step follows a Lomax distribution
- $I_{pred} \rightarrow$ (perturbed I) at each time step with probability p we delete the packets of the true instance I, and with probability p we add an independent instance

Bad prediction: maintain robustness



Good prediction: beat online algorithms

Outline

- Learning-augmented online algorithms
- Case study: set cover
- Instantiating PDLA for other problems
- **Future directions**

Summary

- PDLA gives a principled way of extending the primal-dual approach to incorporate new predictions

- Simple proofs (using old analysis)

- Unifies and some new results

Future directions

- Apply PDLA to problems with packing constraints (e.g. revenue maximization in ad-auctions)
- Apply PDLA to problems with covering constraints and non-linear objective functions (e.g. speed scaling for energy minimization scheduling)
- Learning augment and try to get tight consistency/robustness guarantees for many more covering problems (e.g. load balancing, weighted caching etc.)
- Good advice doesn’t come for free

Thank You!