On Generally-Applicable Analysis Techniques in Online Algorithms

Kim S. Larsen

Department of Mathematics and Computer Science
University of Southern Denmark

kslarsen@imada.sdu.dk

Presentation at the University of Toronto
September 8, 2017

Primary collaborators on today's material

Joan Boyar
University of Southern Denmark, Odense, Denmark

Lene M. Favrholdt
University of Southern Denmark, Odense, Denmark

Michal Kotrbčík
The University of Queensland, Brisbane, Australia

But many other contributors, including...

Yossi Azar
Tel-Aviv University

Martin R. Ehmsen
Mobile Industrial Robots

Leah Epstein
University of Haifa

Jens S. Kohrt
Univ. Southern Denmark

Sandy Irani
University of California, Irvine

Asaf Levin
The Technion

Morten N. Nielsen
Danish State Railways

Marie Gabriele Christ, Camelot ITLab
Sushmita Gupta, Univ. Bergen
Abyayananda Maiti, IIT Patna

Online Algorithms: Standard Set-Up

SDU

- Input is given piece-wise.
- Output is committed irrevocably in response to and for each piece.
- Some objective function must be optimized.

Examples: paging, bin covering & packing, scheduling, k-server, ...

The competitive ratio is the most common quality measure.

For minimization problems, $\textsc{Alg}$ is *c-competitive* if

$$\exists\alpha\,\forall I\; \textsc{Alg}(I) \leq c\,\textsc{Opt}(I) + \alpha \quad \left(\text{or, essentially, } \limsup_{|I|\to\infty} \frac{\textsc{Alg}(I)}{\textsc{Opt}(I)} \leq c\right)$$

- $\textsc{Alg}(I)$ is the cost of processing the input sequence I.
- $\textsc{Opt}$ is an assumed optimal offline algorithm.
- The *competitive ratio* is the *infimum* over such c.
- Similar definitions for maximization problems (profit instead of cost).

The Big Picture: The Nature of Online Algorithms

The standard set-up and competitive analysis works well for some problems; poorly for others.

Problems are due to

- strictness wrt. input/ouput requirements,
- competitive analysis being a worst-case measure, and
- a too powerful adversary and optimal algorithm.

Attempts to get better results (prediction of behavior in practice), include

- introducing alternative measures,
- modifying the definition of unrestricted, piece-wise input, and
- modifying the output requirement of irrevocability.

"Alternative performance measures" and "input restrictions" have received significant attention (limiting the power of the adversary).

Performance Measures and Input Restriction (partial)

- competitive ratio
  [Graham, Bell, 1966] [Sleator & Tarjan, CACM, 1985]
  [Karlin et al., Algorithmica, 1988]
- online/online ratio
  [Gyárfás & Lehel, Ars Combinatoria, 1990]
- statistical adversary
  [Raghavan, On-Line Algorithms, 1992]
- loose competitive ratio
  [Young, Algorithmica, 1994]
- max/max ratio
  [Ben-David & Borodin, Algorithmica, 1994]
- access graphs – locality of reference
  [Borodin et al., JCSS, 1995]
- random order ratio
  [Kenyon, SODA, 1996]
- accommodating ratio
  [Boyar & L., Algorithmica, 1999]
- extra resource analysis
  [Kalyanasundaram & Pruhs, JACM, 2000]
- diffuse adversary
  [Koutsoupias & Papadimitriou, SICOMP, 2000]
- accommodating function
  [Boyar, L. & Nielsen, SICOMP, 2001]
- smoothed analysis
  [Spielman & Teng, JACM, 2004]
- working set – locality of reference
  [Albers et al., JCSS, 2005]
- relative worst order analysis
  [Boyar & Favrholdt, TAlg, 2007]
  [Boyar, Favrholdt & L., JCSS, 2007]
- bijective and average analysis
  [Angelopoulos et al., SODA, 2007]
- relative interval analysis
  [Dorrigiv et al., TCS, 2009]
- advice complexity
  [Dobrev et al., SOFSEM, 2008] [Emek el al., WAOA, 2009]
  [Böckenhauer et al., ISAAC, 2009]
- bijective ratio
  [Angelopoulos et al., arXiv, 2016]
- online-bounded analysis
  [Boyar et al., JoS, in press]

Systematic Studies

Most new measures and restrictions are driven by one problematic case, and comparisons are made to competitive analysis only.

Adding techniques to the online algorithms toolbox

[Boyar, L. & Nielsen, SICOMP, 2001]
Systematic studies of *input restrictions*, defining generally-applicable, generalized methods.

[Boyar, Irani & L., WADS, 2009, Boyar, Irani & L., Algorithmica, 2015]
Systematic studies of *performance measures*, comparing many measures on the same problem.

[Boyar et al., WADS, 2017]
Systematic studies of *relaxed irrevocability*.

Bin Packing variant: Dual

[Leung, thesis, 1977]; reported in [Bruno & Downey, Acta Informatica, 1985]

Parameters: n and k.

Integer-sized items.

n bins of size $k \in \mathbb{N}$.

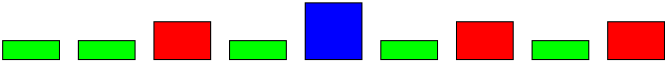
Objective: Pack as many items as possible.

Requirement: *fairness*, i.e., no rejection of items that fit.

Some algorithms are

FirstFit and WorstFit

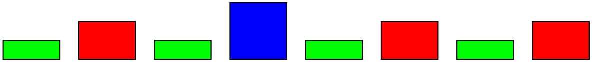
Bin Packing variant: Dual



FIRSTFIT

WORSTFIT

Bin Packing variant: Dual

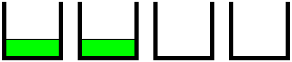


FirstFit

WorstFit

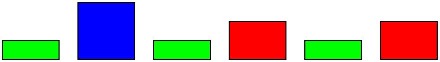
Bin Packing variant: Dual

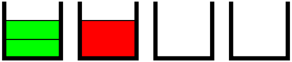
FirstFit



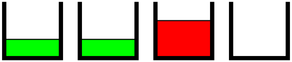
WorstFit

Bin Packing variant: Dual



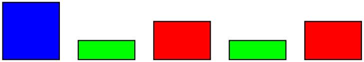


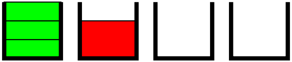
FirstFit



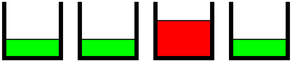
WorstFit

Bin Packing variant: Dual



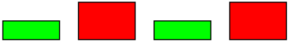


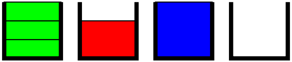
FIRSTFIT



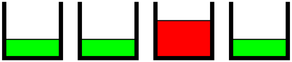
WORSTFIT

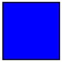
Bin Packing variant: Dual





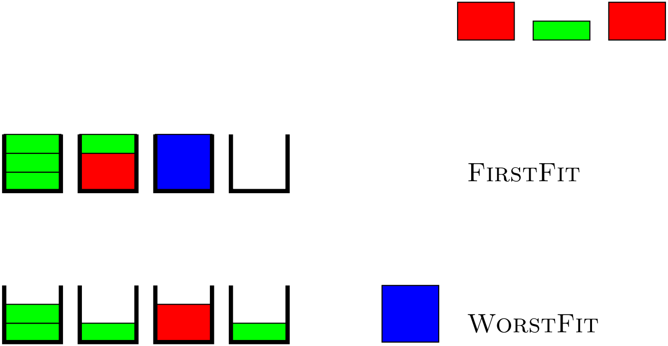
FirstFit



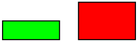


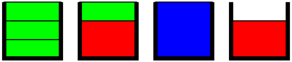
WorstFit

Bin Packing variant: Dual

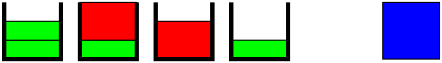


Bin Packing variant: Dual



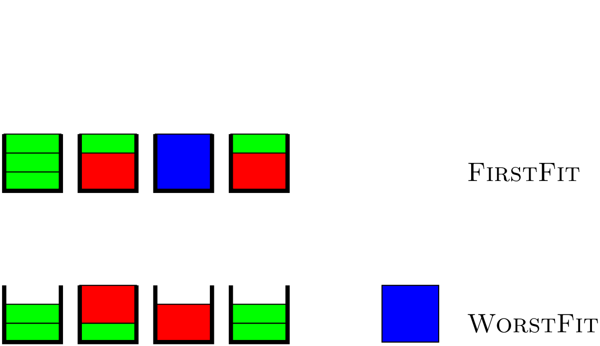


FirstFit

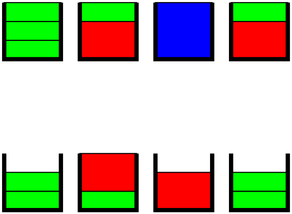


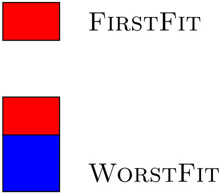
WorstFit

Bin Packing variant: Dual

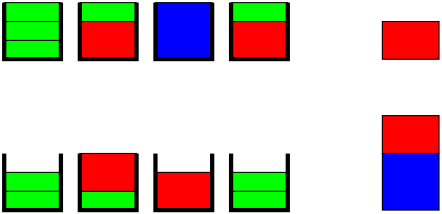


Bin Packing variant: Dual





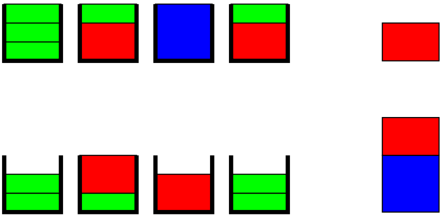
Bin Packing variant: Dual



FIRSTFIT Result: 8

WORSTFIT Result: 7

Bin Packing variant: Dual



FIRSTFIT Result: 8

WORSTFIT Result: 7

Which algorithm would you choose?

Bin Packing variant: Dual

On "real-life" input, FIRSTFIT is consistently better.

Problem?

According to Competitive Analysis,

WORSTFIT is *strictly better* than FIRSTFIT!

[Boyar, L. & Nielsen, SICOMP, 2001]

Accommodating Function

– resource-based input restriction [Boyar, L. & Nielsen, SICOMP, 2001].

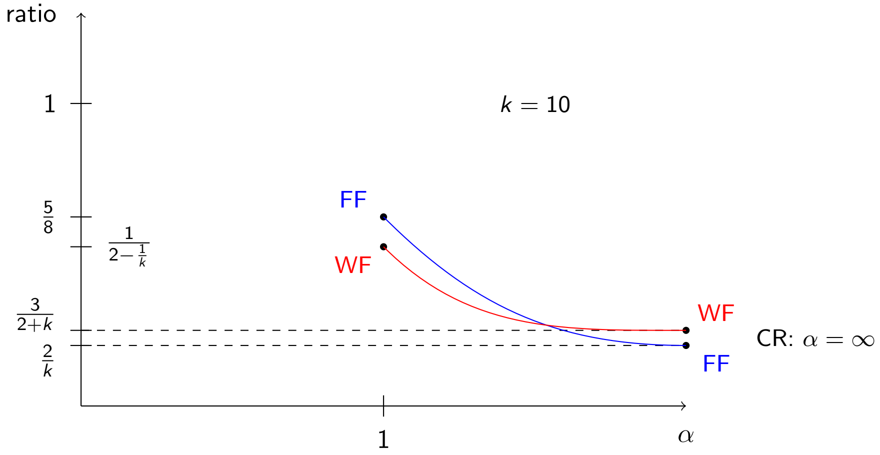
Assume we have some resource of size n.
$\textsc{Alg}_n(I)$ is the cost of running $\textsc{Alg}$ on I with resources n.

I is an *α-sequence* if $\textsc{Opt}_{\alpha n}(I) = \textsc{Opt}_{n'}(I)$ for all $n' \geq \alpha n$.

The value of the *accommodating function* for $\textsc{Alg}$ at the point α is

the competitive ratio of $\textsc{Alg}$ on α-sequences.

Dual Bin Packing and Accommodating Function



[Boyar, L. & Nielsen, SICOMP, 2001]

(Similar results for other bin packing, interval coloring, and other problems.)

Bin Packing: The Torontonian's Problem

Bin Packing: The Torontonian's Problem

Bin Packing variant: Coloring

[Krumke et al., TCS, 2008]

Parameters: B and q.

Unit-sized, colored items.

Bins of size $B \in \mathbb{N}$.

At most q non-empty non-full bins at any time.

Objective: Minimize the maximal number of different colors in any bin.

Some algorithms are

NextFit and GreedyFit

Bin Packing variant: Coloring

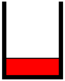
$B = 4$, $q = 3$.

GREEDYFIT

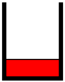
NEXTFIT

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



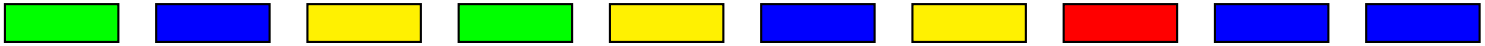
GreedyFit

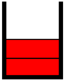


NextFit

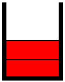
Bin Packing variant: Coloring

$B = 4,\ q = 3.$





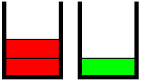
GREEDYFIT



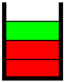
NEXTFIT

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



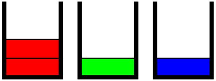
GreedyFit



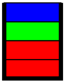
NextFit

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



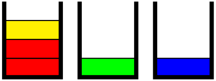
GreedyFit



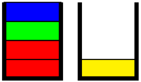
NextFit

Bin Packing variant: Coloring

$B = 4, \; q = 3.$



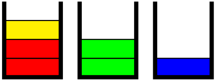
GreedyFit



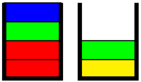
NextFit

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



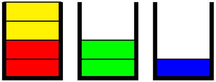
GreedyFit



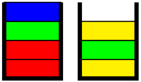
NextFit

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



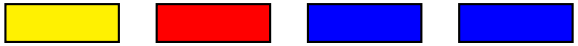
GreedyFit

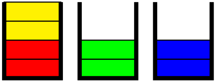


NextFit

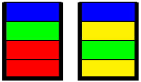
Bin Packing variant: Coloring

$B = 4,\ q = 3.$





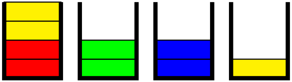
GreedyFit



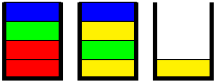
NextFit

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



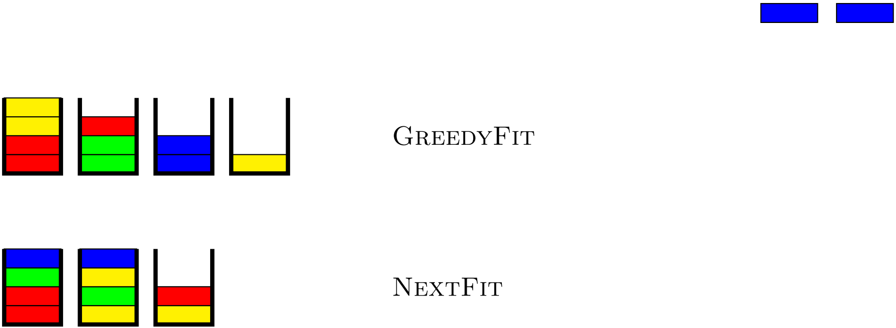
GREEDYFIT



NEXTFIT

Bin Packing variant: Coloring

$B = 4,\ q = 3.$

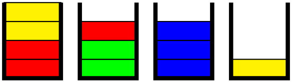


GreedyFit

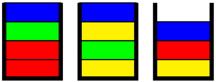
NextFit

Bin Packing variant: Coloring

$B = 4$, $q = 3$.



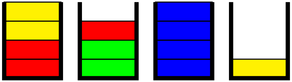
GreedyFit



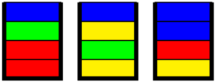
NextFit

Bin Packing variant: Coloring

$B = 4,\ q = 3.$



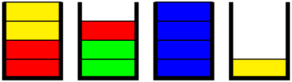
GreedyFit



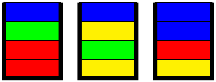
NextFit

Bin Packing variant: Coloring

$B = 4$, $q = 3$.



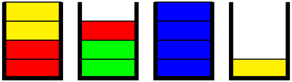
GREEDYFIT — Result: 2



NEXTFIT — Result: 3

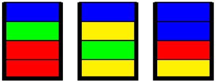
Bin Packing variant: Coloring

$B = 4$, $q = 3$.



GreedyFit

Result: 2



NextFit

Result: 3

Which algorithm would you choose?

Bin Packing variant: Coloring

Problem?

According to (strict) Competitive Analysis,

NEXTFIT is *strictly better* than GREEDYFIT!

[Krumke et al., TCS, 2008]

Relative Worst Order Analysis

[Boyar & Favrholdt, TAlg, 2007], [Boyar, Favrholdt & L., JCSS, 2007]

$\mathbb{A}_{\mathrm{W}}(I)$: worst $\mathbb{A}(\sigma(I))$ for any σ.

For algorithms $\mathbb{A}$ and $\mathbb{B}$,

$$\begin{aligned} c_l(\mathbb{A},\mathbb{B}) &= \sup\{c \mid \exists b\colon \forall I\colon \mathbb{A}_{\mathrm{W}}(I) \geq c\,\mathbb{B}_{\mathrm{W}}(I) - b\} \\ c_u(\mathbb{A},\mathbb{B}) &= \inf\,\{c \mid \exists b\colon \forall I\colon \mathbb{A}_{\mathrm{W}}(I) \leq c\,\mathbb{B}_{\mathrm{W}}(I) + b\} \end{aligned}$$

The *relative worst order ratio*, $\mathrm{WR}_{\mathbb{A},\mathbb{B}}$, of $\mathbb{A}$ to $\mathbb{B}$:

$$\begin{aligned} c_l(\mathbb{A},\mathbb{B}) \geq 1 \quad &\Rightarrow \quad \mathrm{WR}_{\mathbb{A},\mathbb{B}} = c_u(\mathbb{A},\mathbb{B}) \\ c_u(\mathbb{A},\mathbb{B}) \leq 1 \quad &\Rightarrow \quad \mathrm{WR}_{\mathbb{A},\mathbb{B}} = c_l(\mathbb{A},\mathbb{B}) \end{aligned}$$

Intuitively, $\mathrm{WR}_{\mathbb{A},\mathbb{B}}$ is the worst $\frac{\mathbb{A}_{\mathrm{W}}(I)}{\mathbb{B}_{\mathrm{W}}(I)}$ as $I \to \infty$.

Bin Packing variant: Coloring

SDU

Theorem

$\mathrm{WR}_{\textsc{NextFit},\textsc{GreedyFit}} = \min\{q, B\}$ (strict).

GreedyFit is always better (up to the stated factor). Additionally (selected),

- For standard bin packing, Worst-Fit is better than Next-Fit. [Boyar & Favrholdt, TAlg, 2007]
- For dual bin packing, First-Fit is better than Worst-Fit. [Boyar & Favrholdt, TAlg, 2007]
- For paging, LRU is better than FWF and look-ahead helps. [Boyar, Favrholdt & L., JCSS, 2007]
- For scheduling, minimizing makespan on two related machines, a post-greedy algorithm is better than scheduling all jobs on the fast machine. [Epstein et al., J. Comb., 2006]
- For proportional price seat reservation, First-Fit is better than Worst-Fit. [Boyar & Medvedev, TAlg, 2008]

General Techniques: Notable Omissions

- Extra resource analysis [Kalyanasundaram & Pruhs, JACM, 2000].
- Advice complexity – upcoming seminar. :)
- A range of alternative performance measures:
  max/max ratio [Ben-David & Borodin, Algorithmica, 1994]
  random order ratio [Kenyon, SODA, 1996]
  online/online ratio [Gyárfás & Lehel, Ars Combinatoria, 1990]
  statistical adversary [Raghavan, On-Line Algorithms, 1992]
  loose competitive ratio [Young, Algorithmica, 1994]
  diffuse adversary [Koutsoupias & Papadimitriou, SICOMP, 2000]
  smoothed analysis [Spielman & Teng, JACM, 2004]
  bijective and average analysis [Angelopoulos et al., SODA, 2007]
  relative interval analysis [Dorrigiv et al., TCS, 2009]
  online-bounded analysis [Boyar et al., JoS, in press]
- And more problem specific concepts such as look-ahead, fairness, and locality of reference.

Relaxing Irrevocability

Recall the following property of online algorithms:

Output is committed irrevocably in response to and for each piece.

So, why study relaxed irrevocability?

- Application-wise, this is often of interest:

  Problems are not always symmetric, and some decisions may be less irrevocable than others...

- Theoretically of a different flavor than the majority of approaches:

  Extra power to the online algorithm; not limiting the power of the adversary.

- We also want to understand this aspect:

  We add a bit to the understanding of online algorithms.

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model
vertices arrive online with edges incident to earlier vertices.

Edge arrival model
edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model

Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model

Choice of Scene: Online Graph Problems

SDU

Model Details

Vertex arrival model

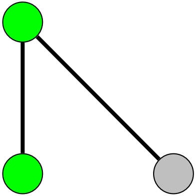
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

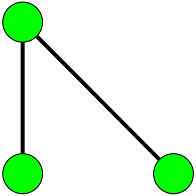
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

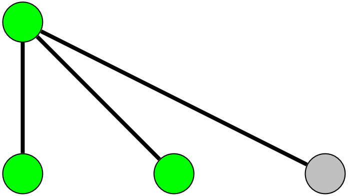
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

SDU

Model Details

Vertex arrival model

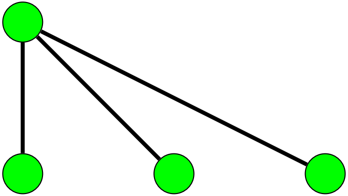
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

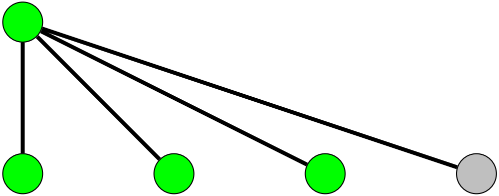
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

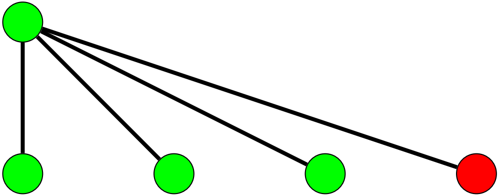
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

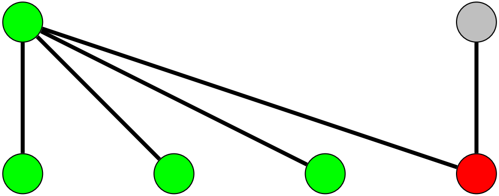
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

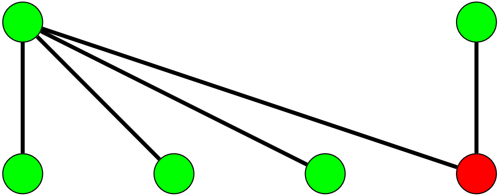
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

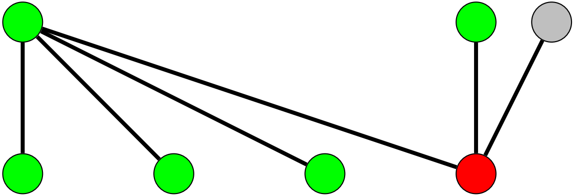
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

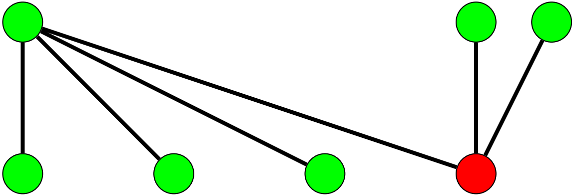
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Model Details

Vertex arrival model

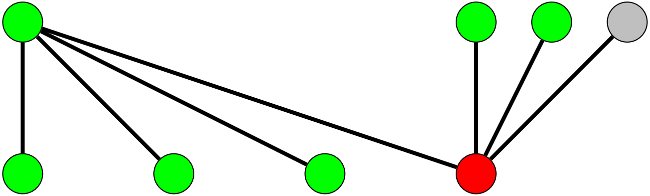
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

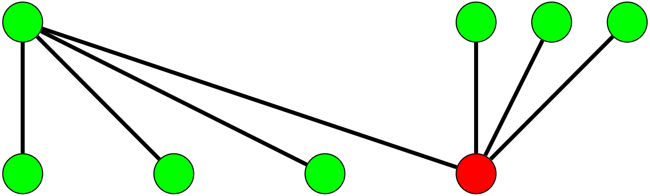
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

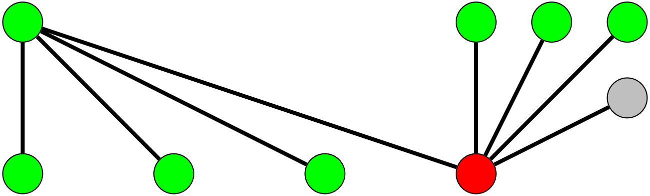
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

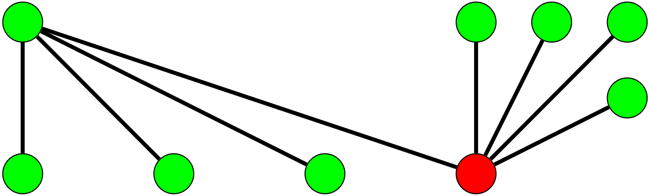
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

SDU

Model Details

Vertex arrival model

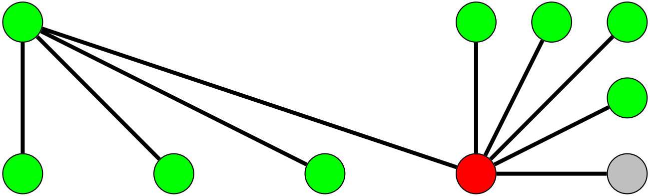
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

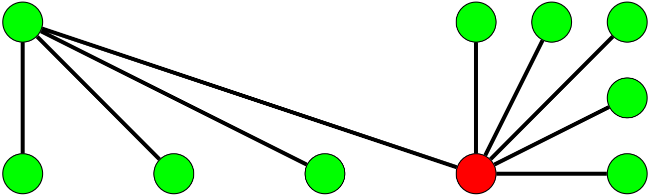
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Choice of Scene: Online Graph Problems

Model Details

Vertex arrival model

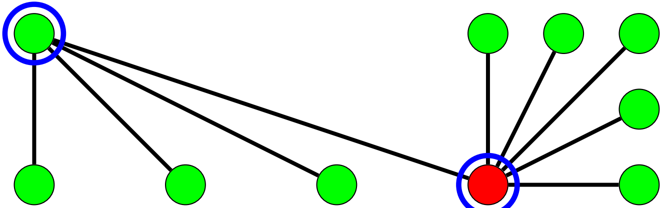
vertices arrive online with edges incident to earlier vertices.

Edge arrival model

edges arrive with their incident vertices
(when natural; similar results in the vertex arrival model)

Choice of graph problems: Set construction, i.e., accept/reject decisions.

Example: Vertex Cover in the Vertex Arrival model



Simple Relaxed Irrevocability Models

To preserve the “online” nature, revocability must be limited:

If decisions are accept/reject, then revoking a decision can be

Late Reject – rejecting an earlier accepted vertex.

Late Accept – accepting an earlier rejected vertex.

We study both separately and also the combination,

Late Accept/Reject – late rejects are irrevocable.

Recall: The input sequence is unknown – we must have a solution at all times.

Related Work

Late Reject has been studied under the names "removable" or "preemption"[1] for

Knapsack
[Iwama & Taketomi, ICALP, 2002] [Han et al., TCS, 2014]
[Han et al., TCS, 2014] [Cygan et al., ToCoSy, 2016] [Han & Makino, TCS, 2016]

Call Control
[Bartal et al., STOC, 1996] [Garay et al., JAlg, 1997]

Maximum Coverage
[Saha & Getoor, SDM, 2009] [Rawitz & Rosén, ESA, 2016]

Weighted Matching
[Epstein et al., JDM, 2011] [Epstein et al., STACS, 2013]

Example The red rule for MST [Tarjan, Book, 1983] can be used as late reject.

[1] Not as in scheduling, where it usually means "postpone".

Related Work

"Solution modification at a cost" (multi-criteria optimization):

Vertex Cover
[Demange & Paschos, TCS, 2005]

Steiner Trees
[Imase & Waxman, JDM, 1991] [Gupta & Kumar, SODA, 2014]
[Gu et al., SICOMP, 2016]

MST/TSP
[Megow et al., SICOMP, 2016] [Jaillet & Lu, Networks, 2014]

Results for the Four Graph Problems

Problem	Standard	Late Accept	Late Reject	Late Accept/Reject
Independent Set	$n-1$	$\frac{n}{\Theta(1)}$	$\lceil \frac{n}{2} \rceil$	$\frac{3\sqrt{3}}{2} \approx 2.598$
Matching	2	2	2	$\frac{3}{2}$
Vertex Cover	$n-1$	2	$n-\Theta(1)$	2
Spanning Forest	W	W	1	1

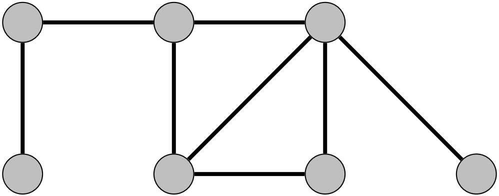
$n = |V|$ in the graph $G = (V, E)$

W is the ratio of the maximum to the minimum edge weight

- All results are tight (matching upper/lower bounds).
- Some summarize known/easy results reformulated in these models.
- The technical highlights are the upper and lower bound proofs of $\frac{3\sqrt{3}}{2}$.

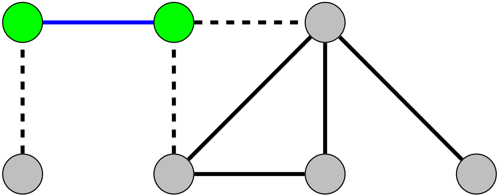
Vertex Cover

The Classic 2-Approximation Algorithm Based on Matching



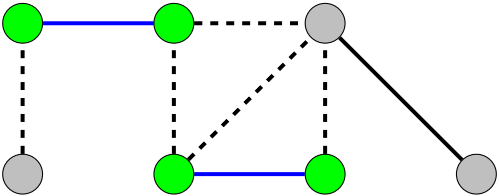
Vertex Cover

The Classic 2-Approximation Algorithm Based on Matching



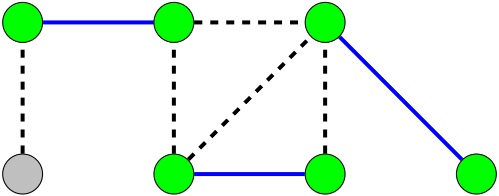
Vertex Cover

The Classic 2-Approximation Algorithm Based on Matching



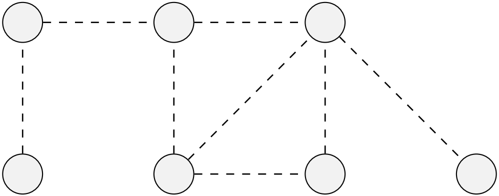
Vertex Cover

The Classic 2-Approximation Algorithm Based on Matching



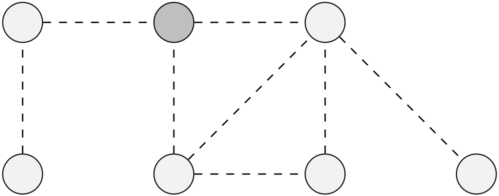
Vertex Cover

Simulate Online with Late Accept



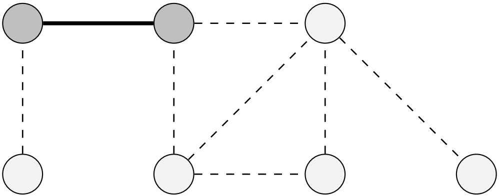
Vertex Cover

Simulate Online with Late Accept



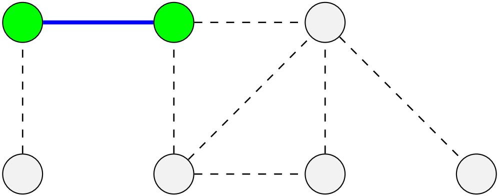
Vertex Cover

Simulate Online with Late Accept



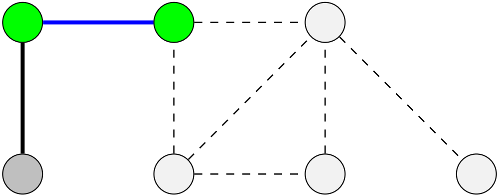
Vertex Cover

Simulate Online with Late Accept



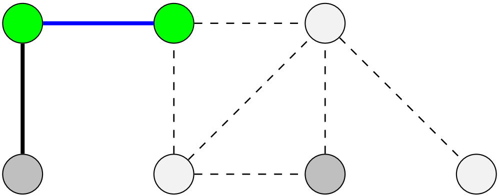
Vertex Cover

Simulate Online with Late Accept



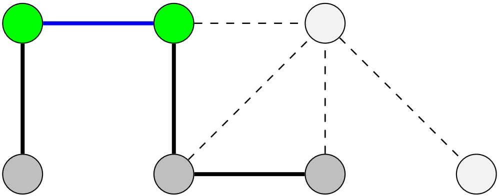
Vertex Cover

Simulate Online with Late Accept



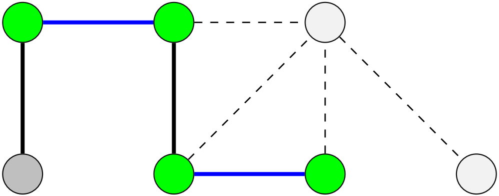
Vertex Cover

Simulate Online with Late Accept



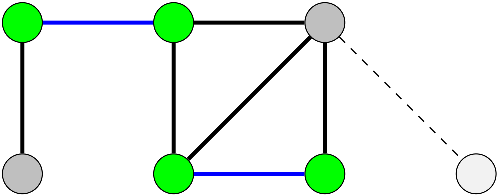
Vertex Cover

Simulate Online with Late Accept



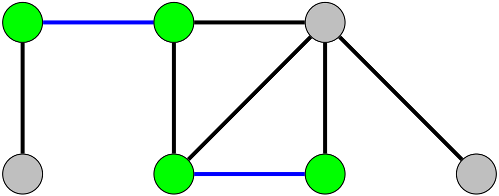
Vertex Cover

Simulate Online with Late Accept



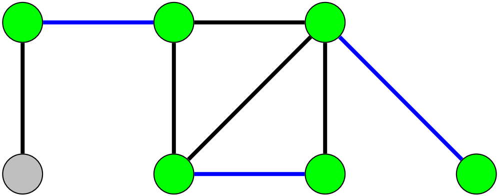
Vertex Cover

Simulate Online with Late Accept



Vertex Cover

Simulate Online with Late Accept

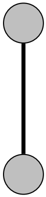


Vertex Cover

Lower Bound

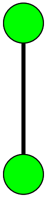
Vertex Cover

Lower Bound



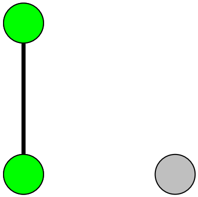
Vertex Cover

Lower Bound



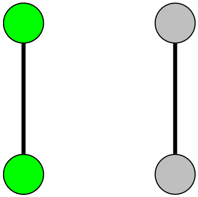
Vertex Cover

Lower Bound



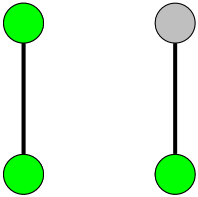
Vertex Cover

Lower Bound



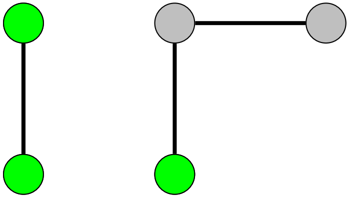
Vertex Cover

Lower Bound



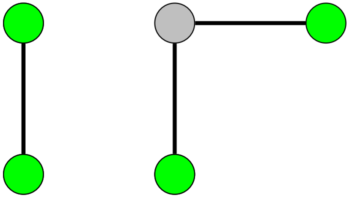
Vertex Cover

Lower Bound



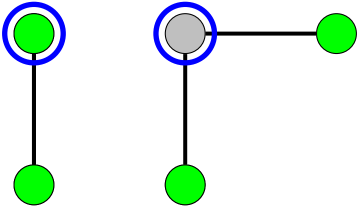
Vertex Cover

Lower Bound



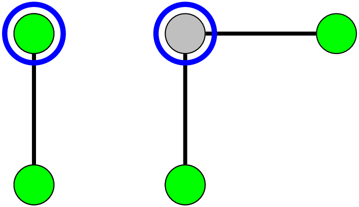
Vertex Cover

Lower Bound



Vertex Cover

Lower Bound



Theorem

For Vertex Cover, in the Late Accept model, the competitive ratio is 2.

Independent Set

Compute an independent set of maximum cardinality.

Independent Set: Upper Bound

T is a *candidate set* if T is an independent set, T contains no late-rejected vertices or vertices in S, and $|T| \geq \sqrt{3}\,|N(T) \cap S|$.

Algorithm: Independent Set in the Late Accept/Reject model.

Result: Independent set S

$S = \emptyset$
while *a vertex* v *is presented* **do**
    if $S \cup \{v\}$ *is independent* **then**
        $S = S \cup \{v\}$
    else
        while *there exists a candidate set* **do**
            Let T be a candidate set minimizing $|S \cap N(T)|$
            $S = S \setminus N(T) \cup T$

The basic algorithmic idea is known, but "rules" and parameters vary.
[Saha & Getoor, SDM, 2009] [Rawitz & Rosén, ESA, 2016]

Independent Set: A Simpler Upper Bound

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

1. $|\mathrm{TheRest} \cap \textsc{Opt}\,| < 2\,|S \setminus \textsc{Opt}\,|$

Independent Set: A Simpler Upper Bound

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

1. $|\mathrm{TheRest} \cap \textsc{Opt}\,| < 2\,|S \setminus \textsc{Opt}\,|$

At termination, why is $\mathrm{TheRest} \cap \textsc{Opt}$ not a candidate set? It's clearly

- independent, and
- another partition than S or late rejects.
- Thus, $|\mathrm{TheRest} \cap \textsc{Opt}\,| < 2\,|N(\mathrm{TheRest} \cap \textsc{Opt}) \cap S|$.

  Since $\textsc{Opt}$ is an independent set,

  $N(\mathrm{TheRest} \cap \textsc{Opt}) \cap S = N(\mathrm{TheRest} \cap \textsc{Opt}) \cap (S \setminus \textsc{Opt}) \subseteq S \setminus \textsc{Opt}$.

Independent Set: A Simpler Upper Bound

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

1. $|\mathrm{TheRest} \cap \mathrm{OPT}\,| < 2\,|S \setminus \mathrm{OPT}\,|$
2. $|S_{\mathrm{exchange}}| \geq |\mathrm{LateRejected}|$

Independent Set: A Simpler Upper Bound

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

1. $|\mathrm{TheRest} \cap \textsc{Opt}\,| < 2\,|S \setminus \textsc{Opt}\,|$
2. $|S_{\mathrm{exchange}}| \geq |\mathrm{LateRejected}|$

We add twice as much to S_{exchange} as we remove from S and therefore add to $\mathrm{LateRejected}$.

Independent Set: A Simpler Upper Bound

SDU

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

1. $|\mathrm{TheRest} \cap \textsc{Opt}\,| < 2\,|S \setminus \textsc{Opt}\,|$
2. $|S_{\mathrm{exchange}}| \geq |\mathrm{LateRejected}|$

Independent Set: A Simpler Upper Bound

Say we exchange only if $|T| \geq 2\,|N(T) \cap S|$.

Partition V into S, $\mathrm{LateRejected}$, $\mathrm{TheRest}$.

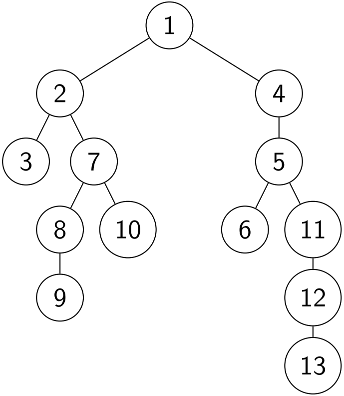
1. $|\mathrm{TheRest} \cap \mathrm{OPT}\,| < 2\,|S \setminus \mathrm{OPT}\,|$
2. $|S_{\mathrm{exchange}}| \geq |\mathrm{LateRejected}|$

Now,

$$\begin{aligned}
|\,\mathrm{OPT}\,| &= |S \cap \mathrm{OPT}\,| + |\mathrm{LateRejected} \cap \mathrm{OPT}\,| + |\mathrm{TheRest} \cap \mathrm{OPT}\,| \\
&< |S \cap \mathrm{OPT}\,| + |\mathrm{LateRejected}| + 2\,|S \setminus \mathrm{OPT}\,| \\
&\leq |S \cap \mathrm{OPT}\,| + |S_{\mathrm{exchange}}| + 2\,|S \setminus \mathrm{OPT}\,| \\
&\leq 3|S|
\end{aligned}$$

Independent Set: Lower Bound

SDU



We give vertices in *bags*.

The algorithm can only hold vertices from one bag (edges to all earlier not late-rejected vertices).

OPT can hold vertices from every second bag on some (long enough) path.

Some pages of calculations are required to sum it all up without loosing any terms.

Independent Set

Compute an independent set of maximum cardinality.

Theorem

For Independent Set, in the Late Accept/Reject model, the competitive ratio is $\frac{3\sqrt{3}}{2}$.

Concluding Remarks Regarding Irrevocability

Is it unfortunate rejects or accepts that are the problem?

Are there different patterns for minimization and maximization problems?

Are there other patterns, depending on more problem-specific characteristics?

Future Work

Investigate other (related) problems in this set-up and draw conclusions.

Consider trade-off results between late operations and solution quality (considering late actions a resource).

Thank you for your attention!

Acknowledging support from ...

The Danish Council for Independent Research | Natural Sciences

The Villum Foundation

References I

S. Albers, L. M. Favrholdt, O. Giel.
On Paging with Locality of Reference.
Journal of Computer and System Sciences, 70(2):145–175, 2005.

S. Angelopoulos, R. Dorrigiv, A. López-Ortiz.
On the Separation and Equivalence of Paging Strategies.
18th ACM-SIAM Symposium on Discrete Algorithms, 229–237, 2007.

S. Angelopoulos, M. P. Renault, P. Schweitzer.
Stochastic Dominance and the Bijective Ratio of Online Algorithms.
arXiv:1607.06132 [cs.DS], 2016.

Y. Bartal, A. Fiat, S. Leonardi.
Lower Bounds for On-line Graph Problems with Application to On-line Circuit and Optical Routing.
28th Annual ACM Symposium on Theory of Computing, 531–540, 1996.

S. Ben-David, A. Borodin.
A New Measure for the Study of On-Line Algorithms.
Algorithmica, 11(1):73–91, 1994.

References II

A. Borodin, S. Irani, P. Raghavan, B. Schieber.
Competitive Paging with Locality of Reference.
Journal of Computer and System Sciences, 50:244–258, 1995.

J. Boyar, L. Epstein, L. M. Favrholdt, K. S. Larsen, A. Levin.
Online-Bounded Analysis.
Journal of Scheduling.
Accepted for Publication.

J. Boyar, L. M. Favrholdt.
The Relative Worst Order Ratio for Online Algorithms.
ACM Transactions on Algorithms, 3(2), no. 22, 2007.

Joan Boyar, Lene M. Favrholdt, Michal Kotrbčík, and Kim S. Larsen.
Relaxing the Irrevocability Requirement for Online Graph Algorithms.
In *15th International Algorithms and Data Structures Symposium (WADS)*, volume 10389 of *Lecture Notes in Computer Science*, pages 217–228. Springer, 2017.

J. Boyar, L. M. Favrholdt, K. S. Larsen.
The Relative Worst Order Ratio Applied to Paging.
Journal of Computer and System Sciences, 73(5):818–843, 2007.

References III

J. Boyar, S. Gupta, K. S. Larsen.
Access Graphs Results for LRU versus FIFO under Relative Worst Order Analysis.
Thirteenth Scandinavian Symposium and Workshops on Algorithm Theory, volume 7357 of Lecture Notes in Computer Science, 328–339, 2012.

J. Boyar, S. Gupta, K. S. Larsen.
Relative Interval Analysis of Paging Algorithms on Access Graphs.
Thirteenth International Algorithms and Data Structures Symposium, volume 8037 of Lecture Notes in Computer Science, 195–206, 2013.

J. Boyar, S. Irani, K. S. Larsen.
A Comparison of Performance Measures for Online Algorithms.
In *11th International Algorithms and Data Structures Symposium (WADS)*, volume 5664 of *Lecture Notes in Computer Science*, pages 119–130. Springer, 2009.

J. Boyar, S. Irani, K. S. Larsen.
A Comparison of Performance Measures for Online Algorithms.
Algorithmica, 72(4): 969–994, 2015.
Preliminary version in WADS 2009.

References IV

J. Boyar, K. S. Larsen.
The Seat Reservation Problem.
Algorithmica, 25(4):403–417, 1999.

J. Boyar, K. S. Larsen, A. Maiti.
The Frequent Items Problem in Online Streaming under Various Performance Measures.
Nineteenth International Symposium on Fundamentals of Computation Theory, volume 8070 of Lecture Notes in Computer Science, 60–71, 2013.

J. Boyar, K. S. Larsen, A. Maiti.
A Comparison of Performance Measures via Online Search.
Theoretical Computer Science, 532:2–13, 2014.

J. Boyar, K. S. Larsen, M. N. Nielsen.
The Accommodating Function: A Generalization of the Competitive Ratio.
SIAM Journal on Computing, 31(1):233–258, 2001.

J. Boyar, P. Medvedev.
The relative worst order ratio applied to seat reservation.
ACM Transactions on Algorithms, 4(4):article 48, 2008.

References V

H.-J. Böckenhauer, D. Komm, R. Královic, R. Královic, Y. Mömke.
On the Advice Complexity of Online Problems.
20th International Symposium on Algorithms and Computation, volume 5878 of Lecture Notes in Computer Science, 331–340, 2009.

J. L. Bruno, P. J. Downey.
Probabilistic Bounds for Dual Bin-Packing.
Acta Informatica, 22:333—345, 1985.

M. G. Christ, L. M. Favrholdt, K. S. Larsen.
Online Bin Covering: Expectations vs. Guarantees.
Theoretical Computer Science, 556: 71–84, 2014.

M. Chrobak, J. Noga.
LRU Is Better than FIFO.
Algorithmica, 23(2): 180–185, 1999.

M. Cygan, Ł. Jeż, J. Sgall.
Online Knapsack Revisited.
Theory of Computing Systems, 58(1):153–190, 2016.

References VI

M. Demange, V. Th. Paschos.
On-Line Vertex-Covering.
Theoretical Computer Science, 332:83–108, 2005.

S. Dobrev, R. Královič, D. Pardubská.
How Much Information about the Future Is Needed?
34th Conference on Current Trends in Theory and Practice of Computer Science, volume 4910 of Lecture Notes in Computer Science, 247–258, 2008.

R. Dorrigiv, A. López-Ortiz.
A Survey of Performance Measures for On-Line Algorithms.
SIGACT News, 36(3):67–81, 2005.

R. Dorrigiv, A. López-Ortiz, J. I. Munro.
On the Relative Dominance of Paging Algorithms.
Theoretical Computer Science, 410:3694–3701, 2009.

M. R. Ehmsen, J. S. Kohrt, K. S. Larsen.
List Factoring and Relative Worst Order Analysis.
Algorithmica, 66(2):287–309, 2013.

References VII

Y. Emek, P. Fraigniaud, A. Korman, A. Rosén.
On the Additive Constant of the k-Server Work Function Algorithm.
7th International Workshop on Approximation and Online Algorithms, volume 5893 of Lecture Notes in Computer Science, 128–134, 2009.

L. Epstein, L. M. Favrholdt, J. S. Kohrt.
Separating scheduling algorithms with the relative worst order ratio.
Journal of Combinatorial Optimization, 12(4):362–385, 2006.

L. Epstein, L. M. Favrholdt, J. S. Kohrt.
Comparing Online Algorithms for Bin Packing Problems.
Journal of Scheduling, 15(1):13–21, 2012.

L. Epstein, A. Levin, J. Mestre, D. Segev.
Improved Approximation Guarantees for Weighted Matching in the Semi-streaming Model.
SIAM Journal of Discrete Mathematics, 25(3):1251–1265, 2011.

L. Epstein, A. Levin, D. Segev, O. Weimann.
Improved Bounds for Online Preemptive Matching.
30th International Symposium on Theoretical Aspects of Computer Science, volume 20 of Leibniz International Proceedings in Informatics, 389–399, 2013.

References VIII

J. A. Garay, I. S. Gopal, S. Kutten, Y. Mansour, M. Yung.
Efficient On-Line Call Control Algorithms.
Journal of Algorithms, 23(1):180–194, 1997.

R. L. Graham.
Bounds for Certain Multiprocessing Anomalies.
Bell Systems Technical Journal, 45:1563–1581, 1966.

A. Gu, A. Gupta, A. Kumar.
The Power of Deferral: maintaining a constant-competitive steiner tree online.
SIAM Journal on Computing, 45(1):1–28, 2016.

A. Gupta, A. Kumar.
Online Steiner Tree with Deletions.
In *25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 455–467, 2014.

A. Gyárfás, J. Lehel.
First-Fit and On-Line Chromatic Number of Families of Graphs.
Ars Combinatoria, 29C:168–176, 1990.

References IX

X. Han, Y. Kawase, K. Makino.
Randomized algorithms for online knapsack problems.
Theoretical Computer Science, 526:395–405, 2015.

X. Han, K. Makino.
Online minimization knapsack problem.
Theoretical Computer Science, 609:185–196, 2016.

X. Han, Y. Kawase, K. Makino, H. Guo.
Online removable knapsack problem under convex function.
Theoretical Computer Science, 540:62–69, 2014.

M. Imase, B. M. Waxman.
Dynamic Steiner Tree Problem.
SIAM Journal of Discrete Mathematics, 4(3):369–384, 1991.

K. Iwama, S. Taketomi.
Removable Online Knapsack Problems.
29th International Colloquium on Automata, Languages and Programming, volume 2380 of Lecture Notes in Computer Science 2380, 293–305, 2002,

References X

P. Jaillet, X. Lu.
Online Traveling Salesman Problems with Rejection Options.
Networks, 64:84–95, 2014.

B. Kalyanasundaram, K. Pruhs.
Speed Is as Powerful as Clairvoyance.
Journal of the ACM, 47(4):617–643, 2000.

A. R. Karlin, M. S. Manasse, L. Rudolph, D. D. Sleator.
Competitive Snoopy Caching.
Algorithmica, 3(1):79–119, 1988.

C. Kenyon.
Best-Fit Bin-Packing with Random Order.
7th ACM-SIAM Symposium on Discrete Algorithms, 359–364, 1996.

E. Koutsoupias, C. H. Papadimitriou.
Beyond Competitive Analysis.
SIAM Journal on Computing, 30(1):300–317, 2000.

References XI

S. O. Krumke, W. E. de Paepe, J. Rambau, L. Stougie.
Bincoloring.
Theoretical Computer Science, 407: 231–241, 2008.

J. Y. Leung.
Fast Algorithms for Packing Problems.
PhD thesis, *Pennsylvania State University*, 1977.

N. Megow, M. Skutella, J. Verschae, A. Wiese.
The Power of Recourse for Online MST and TSP.
SIAM Journal on Computing, 45(3):859–880, 2016.

P. Raghavan.
A Statistical Adversary for On-Line Algorithms.
On-Line Algorithms, Series in Discrete Mathematics and Theoretical Computer Science, 7:79–83, 1992.

D. Rawitz, A. Rosén.
Online Budgeted Maximum Coverage.
24th Annual European Symposium on Algorithms, volume 57 of Leibniz International Proceedings in Informatics, 73:1–73:17, 2016.

References XII

B. Saha, L. Getoor.
On Maximum Coverage in the Streaming Model & Application to Multi-topic Blog-Watch.
9th SIAM International Conference on Data Mining, 697–708, 2009.

D. D. Sleator, R. E. Tarjan.
Amortized Efficiency of List Update and Paging Rules.
Communications of the ACM, 28(2):202–208, 1985.

D. A. Spielman, S.-H. Teng.
Smoothed Analysis of Algorithms: Why the Simplex Algorithm usually Takes Polynomial Time.
Journal of the ACM, 51(3):385–463, 2004.

R. E. Tarjan.
Data Structures and Network Algorithms.
Volume 44 of the Regional Conference Series in Applied Mathematics, SIAM, 1983.

N. Young.
The *k*-Server Dual and Loose Competitiveness for Paging.
Algorithmica, 11:525–541, 1994.