06M1 Lecture
Online Optimization

Martin Grötschel

Beijing Block Course

"Combinatorial Optimization at Work"

September 25 – October 6, 2005

Martin Grötschel
- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum “Mathematik für Schlüsseltechnologien” (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel

Contents

1. Introduction: What does „online“ mean?
2. A practical application: The Yellow Angels of ADAC
3. Other examples
4. Theoretical analysis of online algorithms
5. Bin packing
6. Ski rental
7. Online TSP and online DARP: basic online problems in logistics
8. „Online logistics“, several real-world examples: stacker cranes, elevators, conveyors, integrated transportation systems, Telebus
9. Analysis of online algorithms in practice

Martin Grötschel

Contents

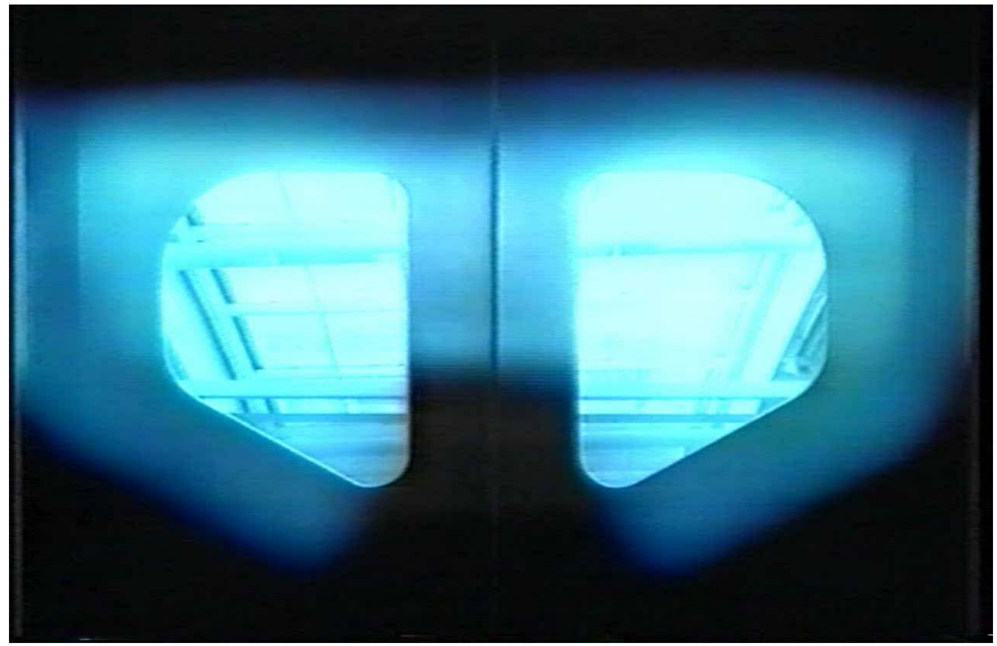
What is „online“?

Mathematical theorems usually start with „Suppose the following … is given, then …“.

And, as a rule, it is required that algorithms start working only when all data is available.

When we deal with online problems, however, we have to make decisions before all information is on hand.

Sometimes „fast decisions “ are necessary; in this case we speak of real-time problems.



aufzug1-dasProblem-1.wmv

Real time

Martin Grötschel

Aim of this Lecture

In my lecture, I will outline several

online problems from practice investigated by the ZIB online optimization research group

(service vehicle planning of the Yellow Angels of ADAC, manufacturing, logistics,...),

and I will dicuss the question:

What is an „appropriate“ action/solution in an online situation.

Again: What is online?

An online algorithm is a method making a decision as soon as new pieces of information become known. Any decision made is irrevocable.

Remarks:

The running time is irrelevant: The key issue is the difficulty arising through the lack of data.

If running time is important: real-time algorithms!

Models of analysis: sequence model, time-stamp model,…

Variants: limited intermediate storage, repacking,…

Martin Grötschel

DFG priority program

- Online optimization of large scale systems
  http://www.zib.de/dfg-echtzeit/

- Book:

Martin Grötschel, Sven O. Krumke, Jörg Rambau (Editors)

Online Optimization of Large Scale Systems

Springer, 2001, 803 pages

Contents

„Unit“ of the service fleet:

Yellow Angel

gelber Engel

Supereinsatz. In Berlin und Brandenburg mussten die Gelben Engel letztes Jahr mehr als 240 000 Mal ausrücken, um Havaristen in der Hauptstadt und auf 1700 Autobahnkilometern wieder flottzumachen – ein Rekordeinsatz. Einen Rückgang von zehn Prozent bei den Pannen registrierten dagegen die Gelben Engel in Mecklenburg-Vorpommern. Bei insgesamt 72 389 Einsätzen schafften sie jedoch auch einen Rekord: In 84 Prozent der Fälle konnten die Autofahrer mit dem Wagen weiterfahren.

Contents

Martin Grötschel

Historically important examples

Online optimization has a long and rich history in continuous applied mathematics (reentry of a space vehicle, control problems,…).

The roots of combinatorial online optimization lie in computer science (~1985, Sleator&Tarjan).

Important examples:

- Paging
- Task scheduling in computing machinery
- Self organizing search (data compression)
- Robotics

It was computer science theory that developed the first (and still the relevant) concepts of analysis of online algorithms.

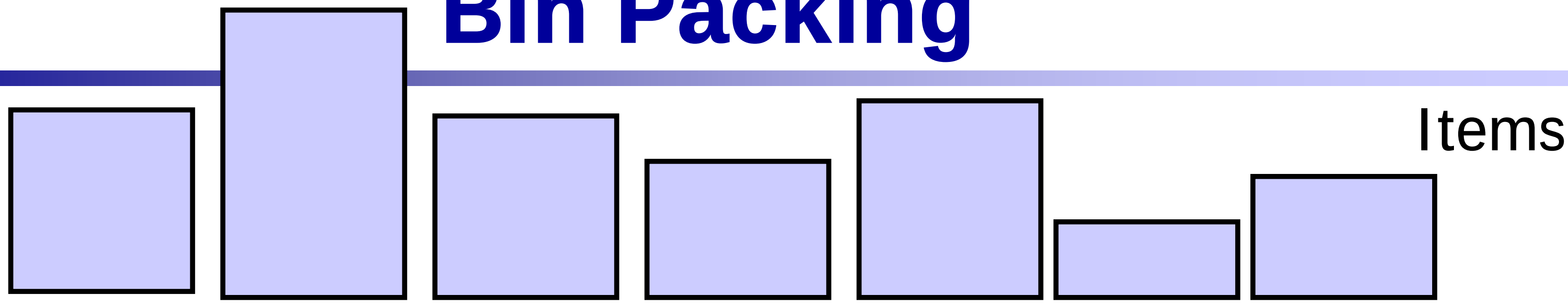
Practical examples of online optimization problems

- The Yellow Angels of ADAC
- Elevator control (Herlitz)
- Warehouse storage control:Stacker Cranes (Siemens, SNI, Herlitz)
- Commissioning (e.g., of greeting cards at Herlitz)
- Control of automatic in-plant logistics systems (SNI, Herlitz)
- The Online Traveling Salesman Problem
- Online Dial-A-Ride Problems (DARP)

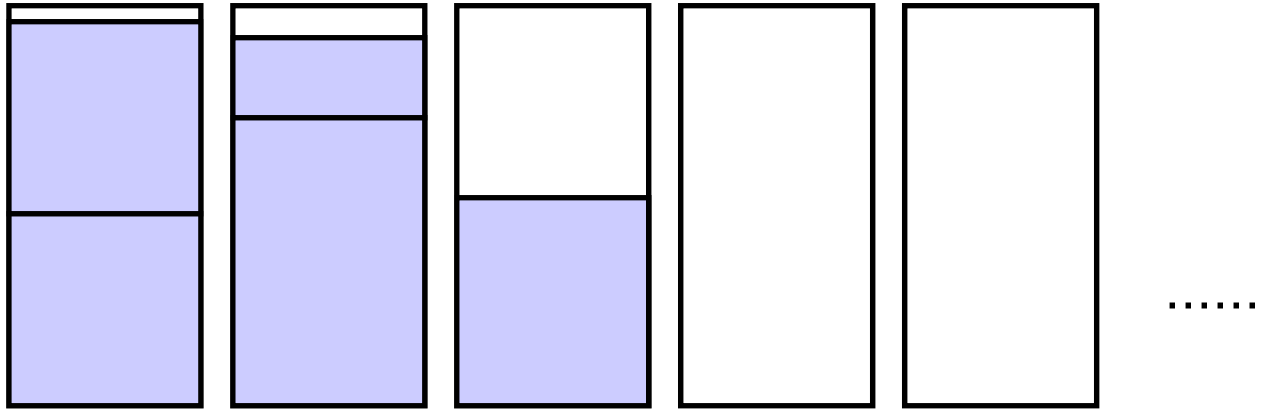
Two didactical examples

- Bin packing
- Ski rental

Bin Packing



(Offline) Bin packing looks trivial, but it is NP-hard in the sense of complexity theory.



Bins (one-dimensional containers)

Goal: Find minimum number of bins (online)

Martin Grötschel

Ski Rental

Imagine someone who goes skiing for the first time in his/her life.

Every morning there is the alternative of

(1) Renting skis for 1 ME a day, or

(2) Buying skis for B ME (B >> 1).

If the person knew for how many days he/she will go skiing, the decision would be simple.

But he/she does not know and has to decide every morning anew.

What shall the skier do?

Contents

A Small Piece of Theory

Competitive Analysis

is, at present, the most important method to theoretically analyse online problems and online algorithms.

Suppose a combinatorial online problem is given.

We have an online algorithm, which has to process a request before the next request arrives.

How to evaluate the goodness or badness (quality) of the online algorithm?

A concept of quality measurement

Compare the solution (more precisely: the value of the solution) with the one an adversary algorithm would have produced with the same data.

What are „reasonable“ adversaries?

One example: the (optimal) offline adversary.

Since he completely knows the entire request sequence, he can compute the optimal solution „offline“.

Martin Grötschel

The offline adversary/competitivity

Let ALG be a (deterministic) online algorithm, and let s be a request sequence. Let ALG(s) denote the value ALG has obtained, and OPT(s) the value of the optimal solution of the offline adversary.

Let $c \geq 1$. We say that ALG is c-competitive, if

(there is a constant b, so that)

$$ALG(s) \leq c\ OPT(s)\ (+\ b)$$

is valid for all request sequences s.

(We suppose that all objective function values are positive.)

Martin Grötschel

History

The term competitive analysis was coined in

Karlin, Manasse, Rudolph, and Sleator (1988).

The foundations of competitiveness were laid in

Sleator and Tarjan (1985).

There are, however, many predecessors in

“worst-case analysis”, e.g., Graham (1966).

Scientific goals

Let **P** be an online problem.

- Given is an online algorithm ALG for **P**, find the best possible competitiveness factor c for ALG.
- Is **P** competitively solvable at all?
- If yes, find an online algorithm for **P** with the best possible competitiveness factor c.
- If **P** is not competitively solvable, what to do then?
- Possibilities:
  - weaker adversary,
  - restricted request sequences,
  - ...

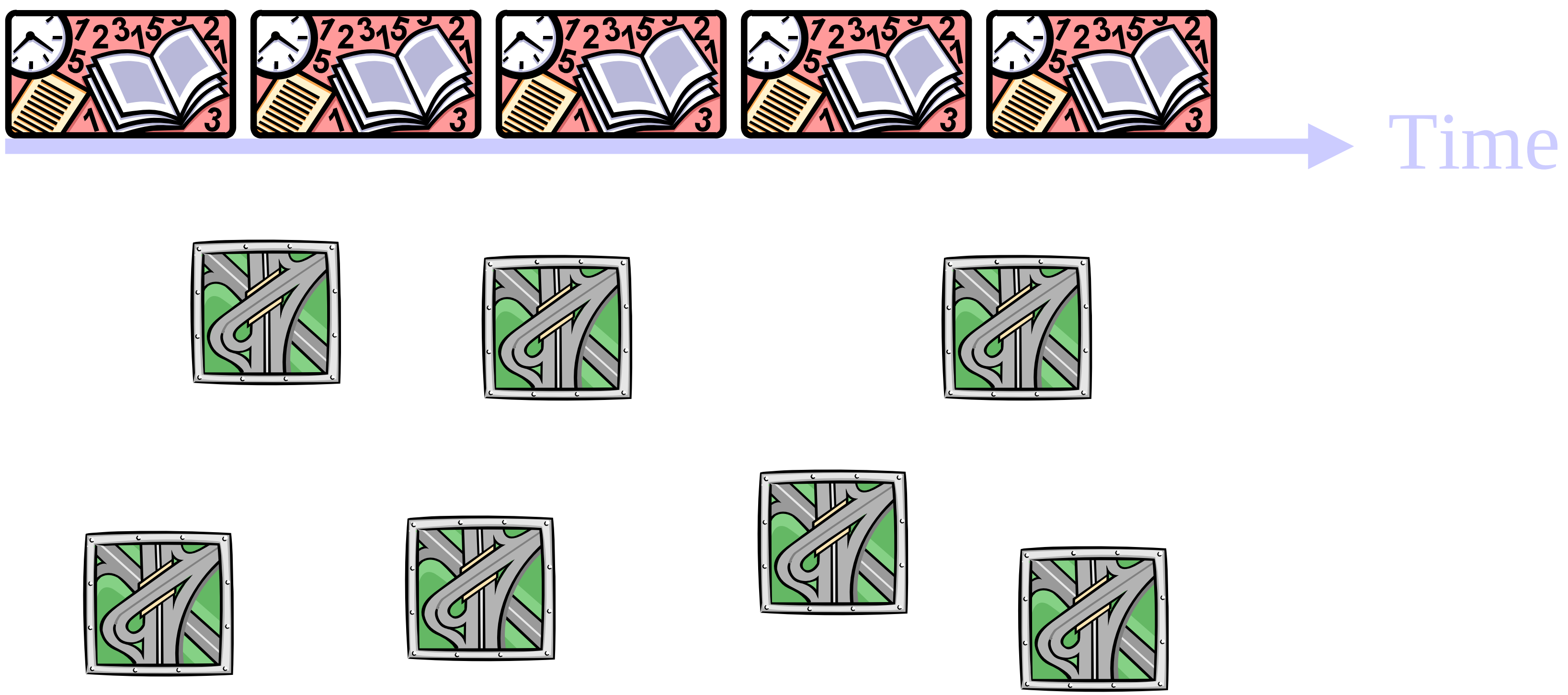
Variations

- Adaptive online adversary
- Adaptive offline adversary
- Stochastic variants
- Randomized algorithms
- Stochastic optimization

- Reasonable load

Online modelling: The time-stamp model

- Request r is not known until time t(r)



Online modeling: The time-stamp model

- Request r is not known until time t(r).
- Request r must be served after time t(r). Waiting is allowed but may incur additional costs.
- Requests need not be scheduled in the order of their appearance.
- Rescheduling of a request is allowed, unless its processing has begun.

Online Modelling and Offline Model

- Sometimes the offline model cannot be stated in a „closed form“
- Offline module is a component of an online optimization algorithm
- Offline module is a possible control instrument

Central Issue of Online Optimization

If we know the competitiveness
of an online algorithm,

or if we even know
the best possible competitiveness factor
for an online problem:

What does it help in practice?

Contents

1. Introduction: What does „online“ mean?
2. A practical application: The Yellow Angels of ADAC
3. Other examples
4. Theoretical analysis of online algorithms
5. Bin packing
6. Ski rental
7. Online TSP and online DARP: basic online problems in logistics
8. „Online logistics“, several real-world examples: stacker cranes, elevators, conveyors, integrated transportation systems, Telebus
9. Analysis of online algorithms in practice

Competitiveness (again)

Let ALG be an online algorithm
(majority decision of the audience).
Let s be a request sequence for the bin packing problem
(generated by me).
ALG(s) denotes the value ALG produced,
whereas OPT(s) denotes the optimal solution of the offline adversary.

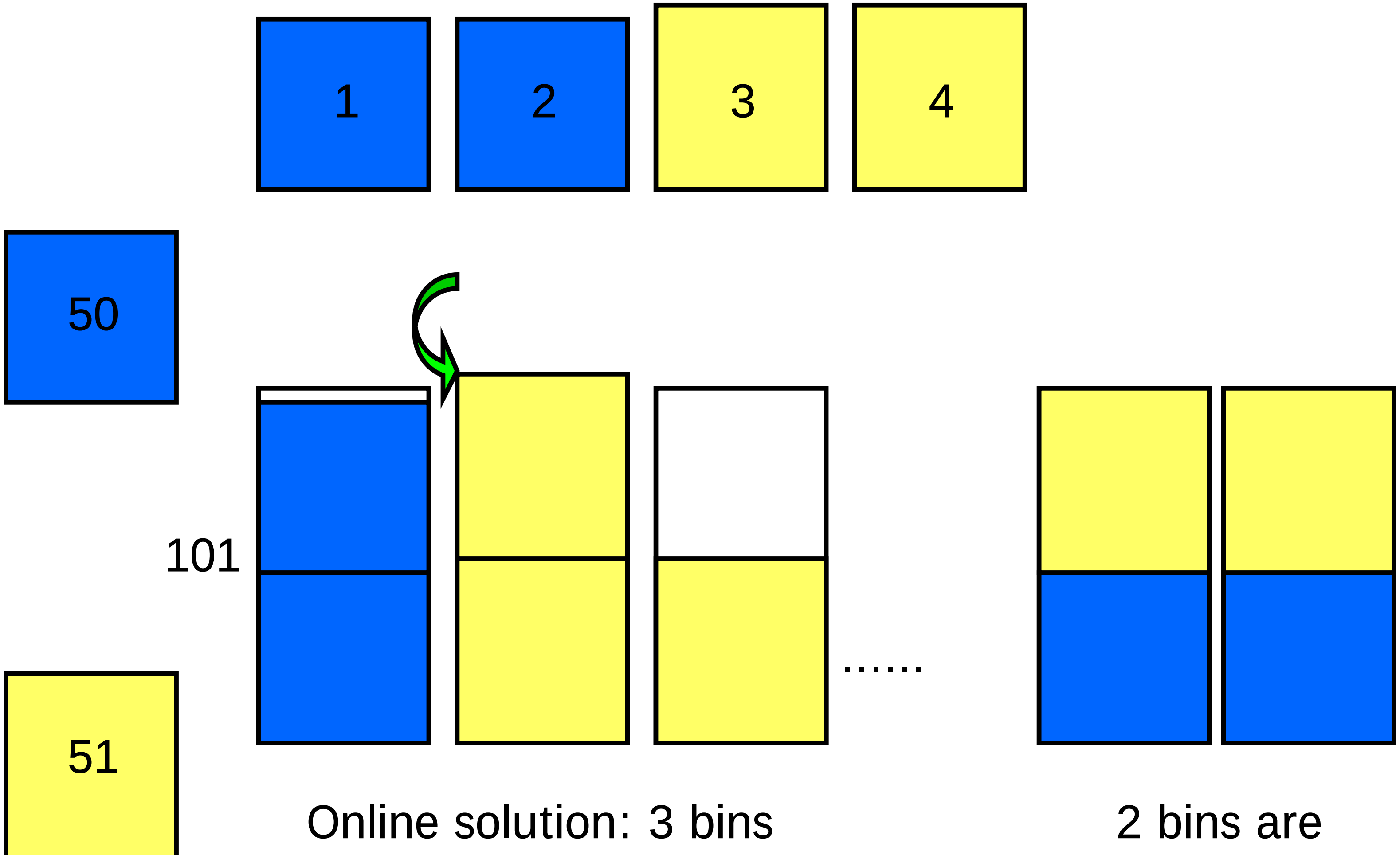
Let $c \geq 1$. We say that ALG is c-competitive, if

$$\mathrm{ALG}(s) \leq c\ \mathrm{OPT}(s)$$

is valid for all request sequences s.

Is there a lower bound on c for bin packing?

Competitiveness of bin packing



Online solution: 3 bins

2 bins are optimal

Martin Grötschel

Competitivity of bin packing

My request sequence yields:

$1.5 \times OPT(s) \leq ALG(s)$

From this follows a general lower bound:

$c \geq 1.5$ for online bin packing.

Best lower bound known:

$c = 1.5401...$ (van Vliet)

Martin Grötschel

Bin packing algorithms

- Next Fit
- First Fit
- First Fit Decreasing (offline)
- Best Fit
- Harmonic Fit (many variants)
- .....

Bin packing algorithms

Simple proofs for:

- Next Fit is 2-competitive.
- First Fit is 2-competitive.

Upper bounds on the bin packing competitiveness

Algorithm	Competitivity
Next Fit	2
Best Fit	1.7
First Fit	1.7
Revised Harmonic Fit	1.69103
Harmonic++	1.58889 (Seiden 01)

Bin packing with repacking

It is permitted

- to keep k containers open and
- to repack among them.

As soon as another container is opened,

- one of the k containers must be closed forever.

Lee & Lee	$c \geq 1.69103$ (for arbitrary k)
Galambos & Wöginger	$c \leq 1.69103$ (for $k = 3$)

One example for „optimal competitiveness“

Contents

Martin Grötschel

Algorithms for ski rental

There are k=1,2,3,… online algorithms ALG(k).

ALG(k): Rent on k-1 days and buy on day k!

Theorem: ALG(B) is c-competitive for c = 2 – 1/B.

Proof: If s is a request sequence for n days, the costs of ALG(B) are equal to

n, if n<B, or

B-1+B, if n>B-1.

The optimal costs are OPT(n) = min (n,B).

This implies the result.

The „unfair“ adversary

Observation: Each competitive algorithm must buy skis at some point in time. (Otherwise c would not be bounded.)

The unfair adversary: Continues to deliver requests, until the algorithm buys. Then he stops.

Conclusion: For all k, ALG(k) is not better than c-competitive for $c = 2 - 1/B$.

Consequence: With respect to the competitiveness factor c, ALG(B) is a best possible deterministic online algorithm for the ski rental problem.

Randomized Online Algorithms

Definition:
The randomized algorithm RALG is c-competitive, if for all request sequences σ:

$$\mathrm{E}[\mathrm{RALG}(\sigma)] \leq c \cdot \mathrm{OPT}(\sigma)$$

Research issue:

- How can one improve by randomization?
- How does one fool the „best randomized algorithm“ in order to find lower bounds?

Improvement through randomization

We define a randomized algorithm for ski renting:

RANDSKI:

1. Set $r := {}^{B}\!/_{(B-1)}$ and $a := {}^{(r-1)}\!/_{(r^B-1)}$

2. Select a random number $k \in \{1,2,\ldots B-1\}$ according to the distribution $P[k=x] = a \bullet r^k$

3. Buy the skis after k days.

Theorem. $E[RANDSKI(s)] = {}^{r^B}\!/_{(r^B-1)}\, OPT(s)$

Conclusion. RANDSKI is c(B)- competitive with c(B) ~ 1.58 for B going to infinity (expected value)

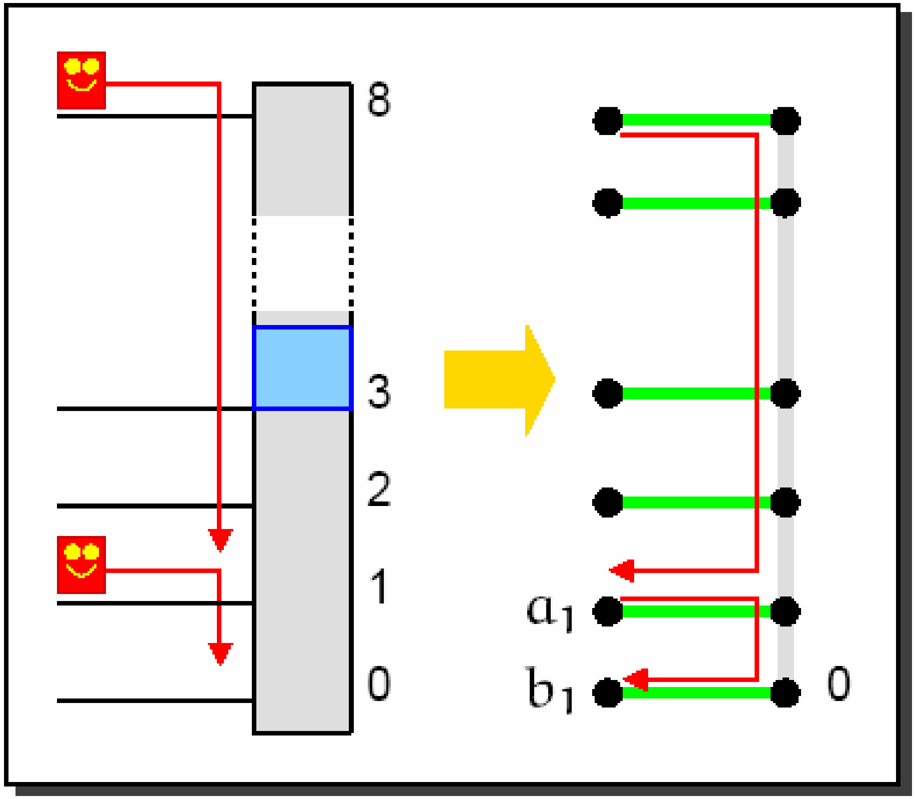
Variations of the ski rental problem

- The rail card problem (BahnCard)
  - The algorithm the German customer service uses is 2-competitive.
  - Not buying is 2-competitive.
  - There is a 3/2-competitive algorithm.
- Purchase of a ski passport
- Purchase of annual or monthly tickets

Contents

The Online Dial-a-Ride Problem

Given:
- Transportation network
- Server with capacity 1.
- request sequence $\rho = r_1, r_2, \ldots, r_m$ with $r_i = (t_i, a_i, b_i)$.

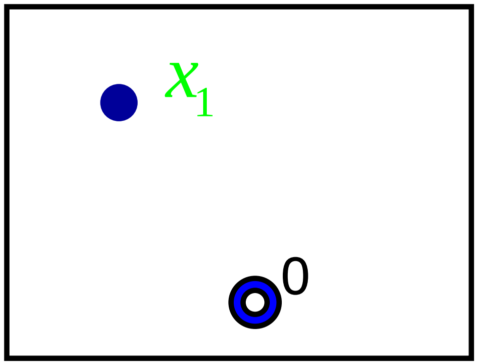


Goal:
- »Best « transporation schedule

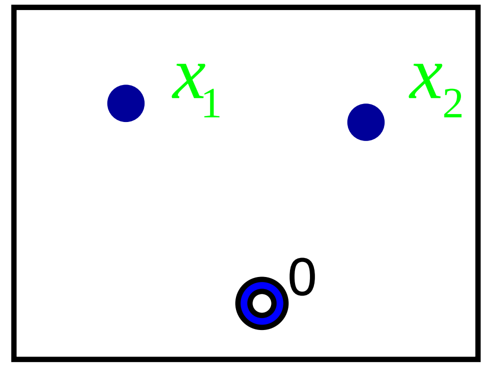
Online TSP (in a metric space)

Instance: $\sigma = r_1, r_2, \ldots, r_n$ where $r_i = (t_i, x_i)$

$t = t_1$



$t = t_2$



• • •

Goal: Find the fastest tour, serving all requests

Algorithm ALG is c-competitive if

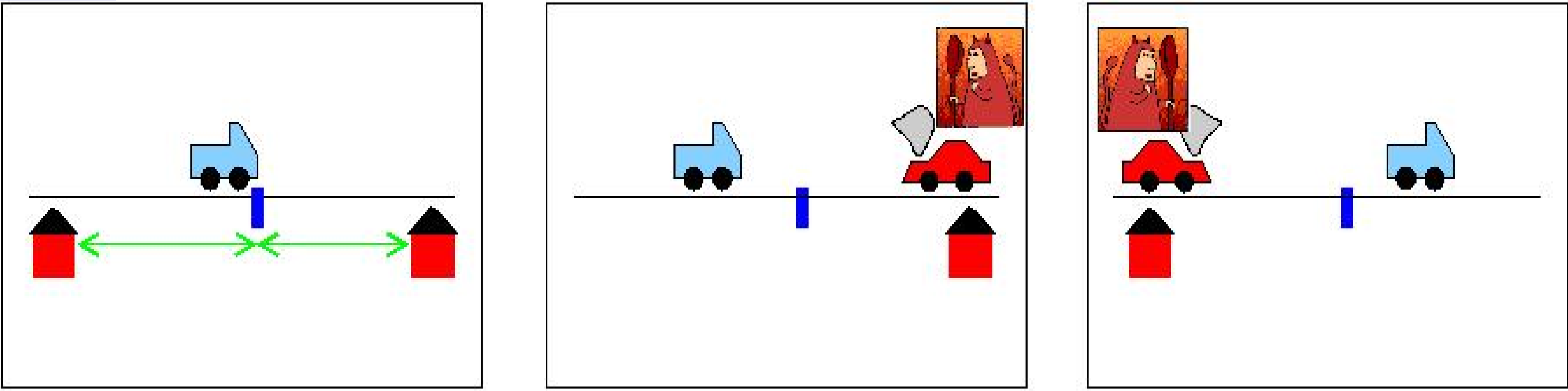
$$\mathrm{ALG}(\sigma) \leq c \cdot \mathrm{OPT}(\sigma)$$

for all request sequences σ

Martin Grötschel

Results (Bad News)

No algorithm can be competitive for the minimization of the maximal flow time.



Time 0 Time 1 Time 1

Online TSP results

	Makespan $C_{\max}$		Latency $\sum C_j$	
	LB	UB	LB	UB
General	2 [AF+94]	2 [AF+94] 2 [AKR00] $\frac{7+\sqrt{13}}{4}$ poly. time	$1+\sqrt{2}$ [FS01]	6 [KPPS01]
Real line	$\frac{9+\sqrt{17}}{8}$ [AF+01]	$\frac{9+\sqrt{17}}{8}$ [L00]		
Half line	$\frac{3}{2}$ [BKPS00]	$\frac{3}{2}$ [BKPS00]		

Online Dial-a-Ride

	Makespan $C_{\max}$		Latency $\sum C_j$	
	LB	UB	LB	UB
General	2 [AF+94]	2 [AKR00] $\frac{\frac{41}{5}+\sqrt{\frac{77}{5}}}{4}$ poly. time	3 [FS01]	6 [KPPS01]
Real line	$\frac{2+\sqrt{2}}{2}$ [AKR00]			
Half line	$\frac{2+\sqrt{2}}{2}$ [AKR00]			

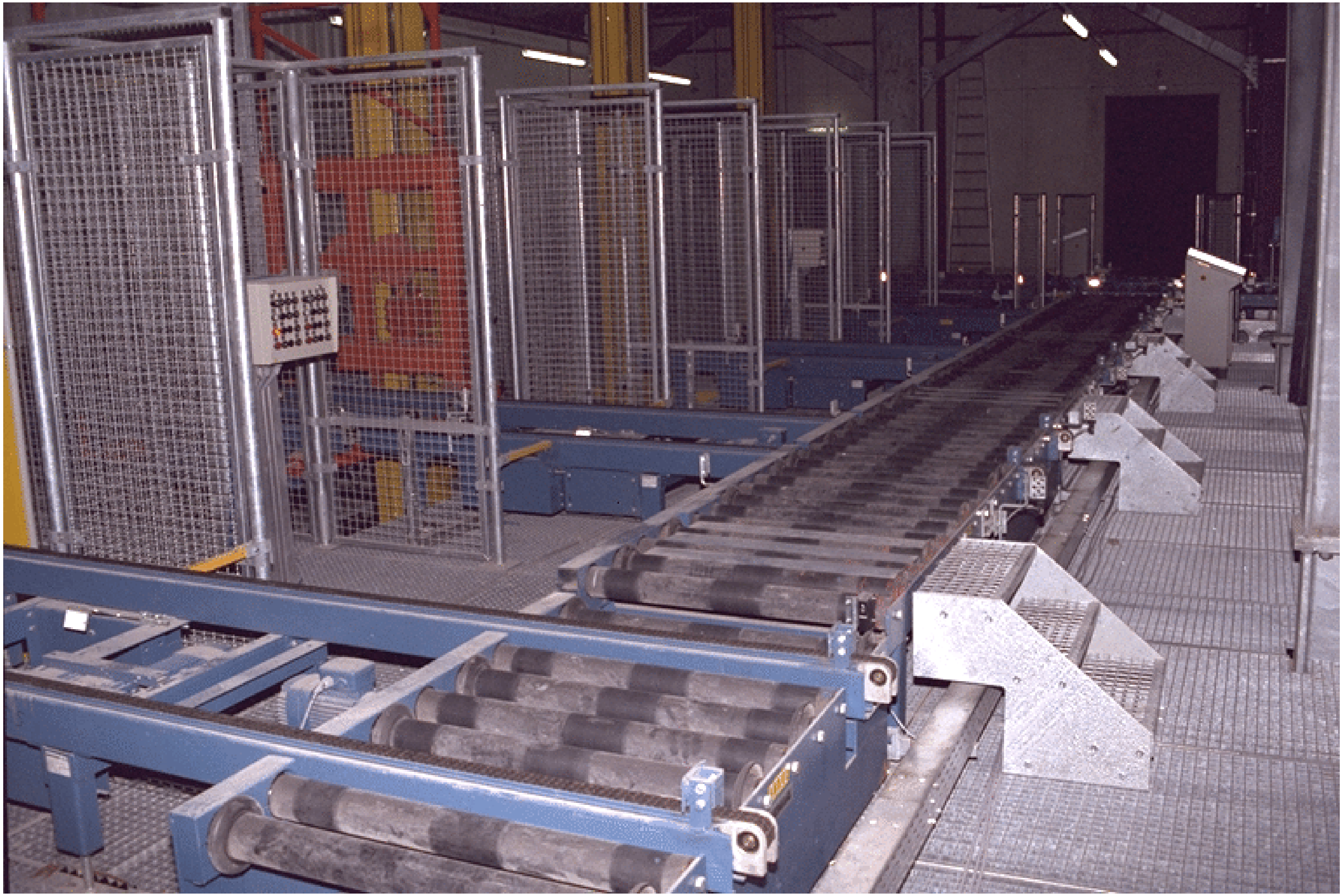
Contents

Herlitz at Falkensee (Berlin)

Martin Grötschel

Example: Control of the stacker cranes in a warehouse

Example: Elevator control



Elevator control

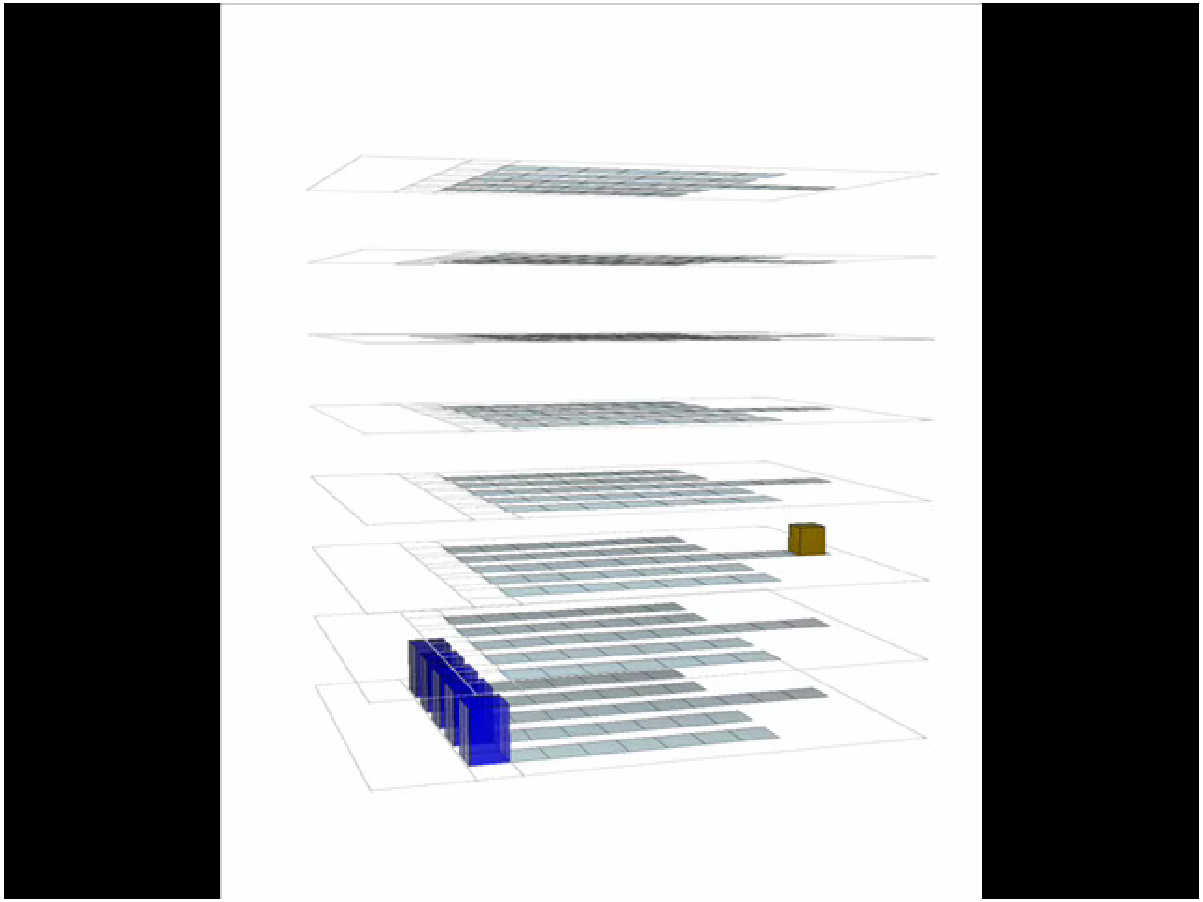
The online algorithms

- Replan
- Ignore
- SmartStart

- Theoretical analysis
- Observations during simulation experiments

aufzug2-OnlineOptimierung-1.wmv

Elevator Simulation



[Aufzug-Simulation-C6-Friese.wmv]

New Approaches

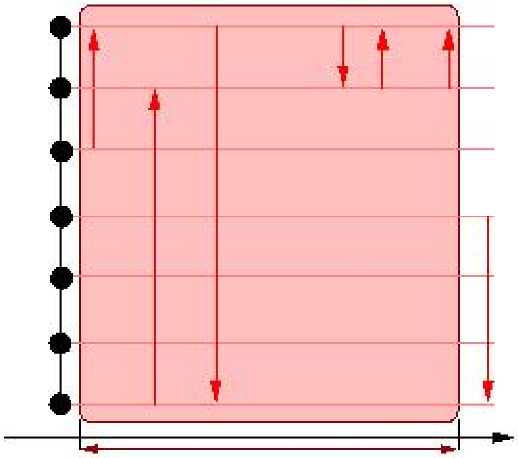
Standard competitive analysis fails for the interesting objectives

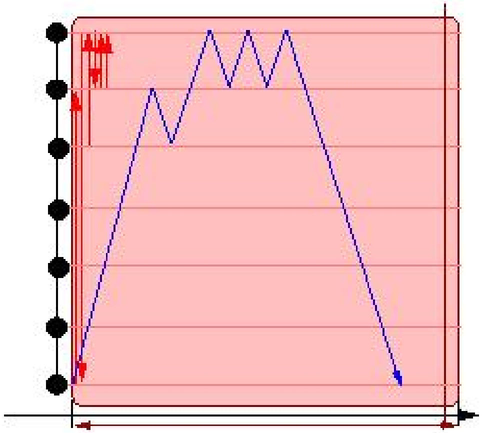
- Find reasonable restrictions for the input sequence
  - Concept of reasonable load

- Find fairness restrictions for the adversary
  - Concept of fair adversary

Reasonable Load

A request sequence is Δ–reasonable,
if offline requests can be served at least as fast as they are released:





and

for all $\delta \leq \Delta$:

$C^{\text{comp}}_{\text{OPT}}\left(R^{offline}\right) \leq \Delta.$

(monotonicity)

if for all $\delta \geq \Delta$

we have…

$\ldots C^{\text{comp}}_{\text{OPT}}\left(R^{offline}\right) \leq \delta,$

Results under Reasonable Load

Theorem (Hauptmeier, Krumke, Rambau, 2000)

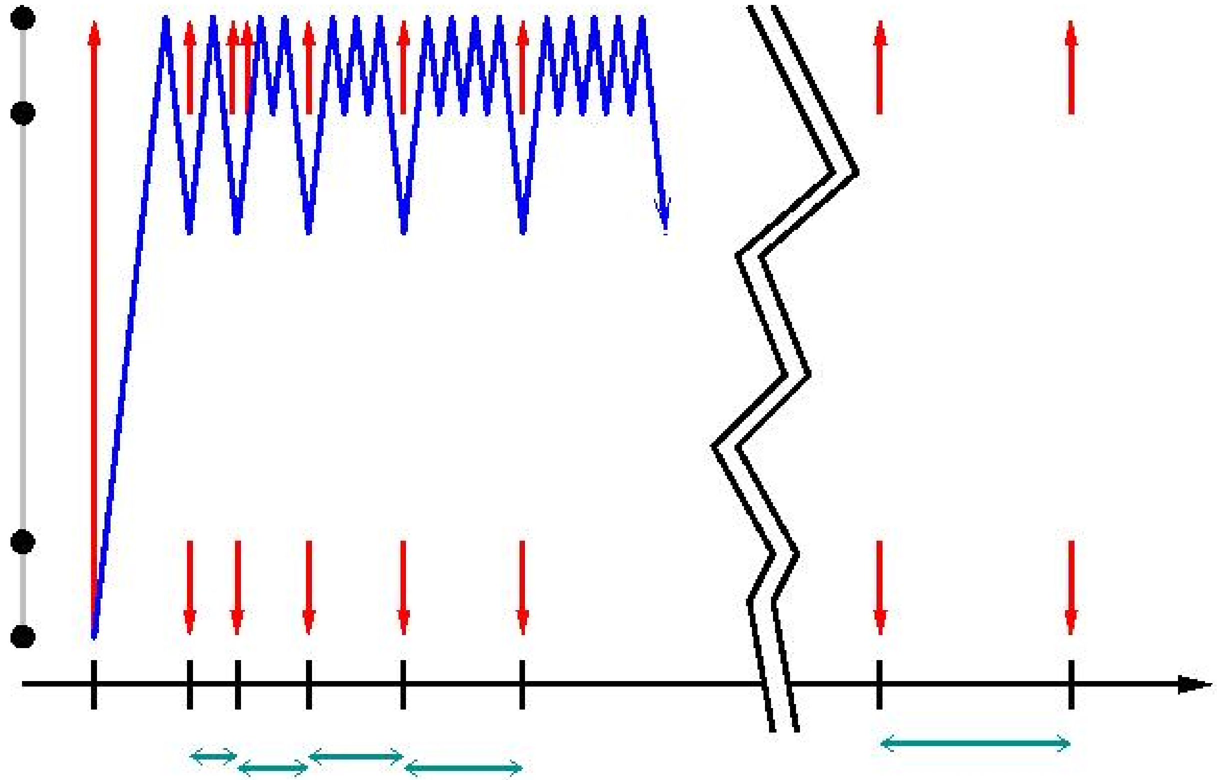
1. *The maximal and average flow time of IGNORE under Δ-reasonable load are bounded from above by 2Δ.*

2. *The above is not true for REPLAN, no matter what your replanning objecive is.*

Why Re-Optimizing Isn‘t Always Good

Objective

$$\frac{1}{n}\sum_{j=1}^{n} F_j$$



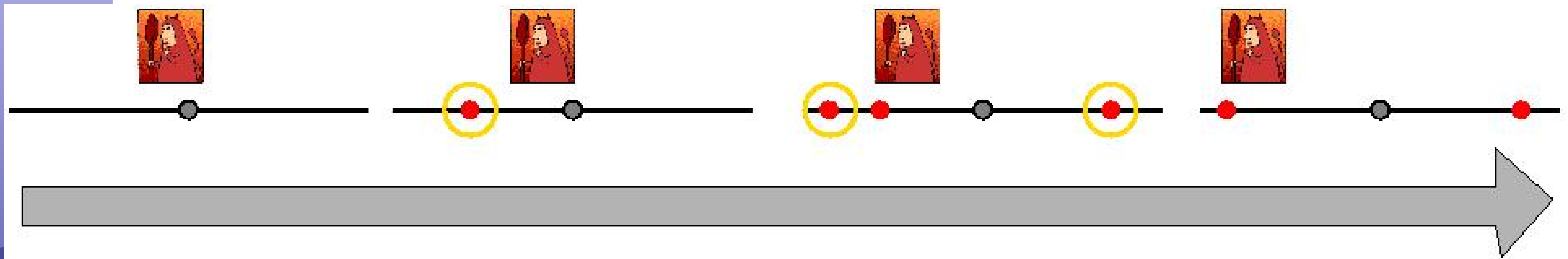
Average flow time of FLOW-REPLAN is unbounded, although one can achieve a bounded flow time even online!

Martin Grötschel

Fair Adversaries

An offline adversary is fair (in R), if …

… he moves his server only towards unserved requests.



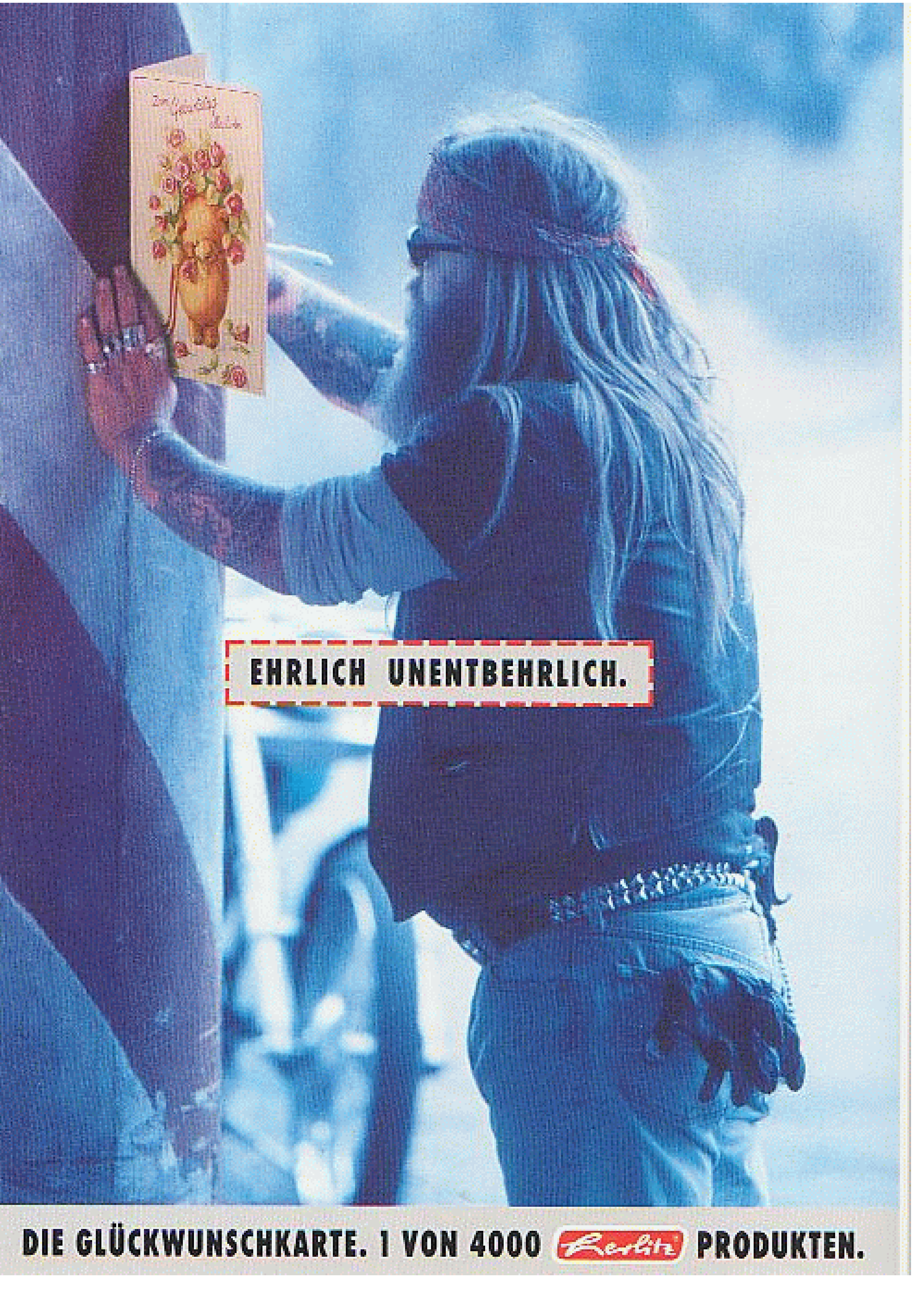
Theorem (Krumke, Laura, de Paepe, Poensgen 02)
There exists a 9/2-competitive algorithm for minimizing the maximal flow time against a fair adversary.

Martin Grötschel

Commissioning of greeting cards

- Theory
- Simulation
- Practice

Greeting Cards



Slides

Order Picking Project:

Norbert Ascheuer, Martin Grötschel, Nicola Kamin, Jörg Rambau

Combinatorial online optimization in practice

OPTIMA, 57 (1998) 1-6

Kamin, Nicola:

On-Line Optimization of Order Picking in an Automated Warehouse

PhD Thesis, TU Berlin, 1998

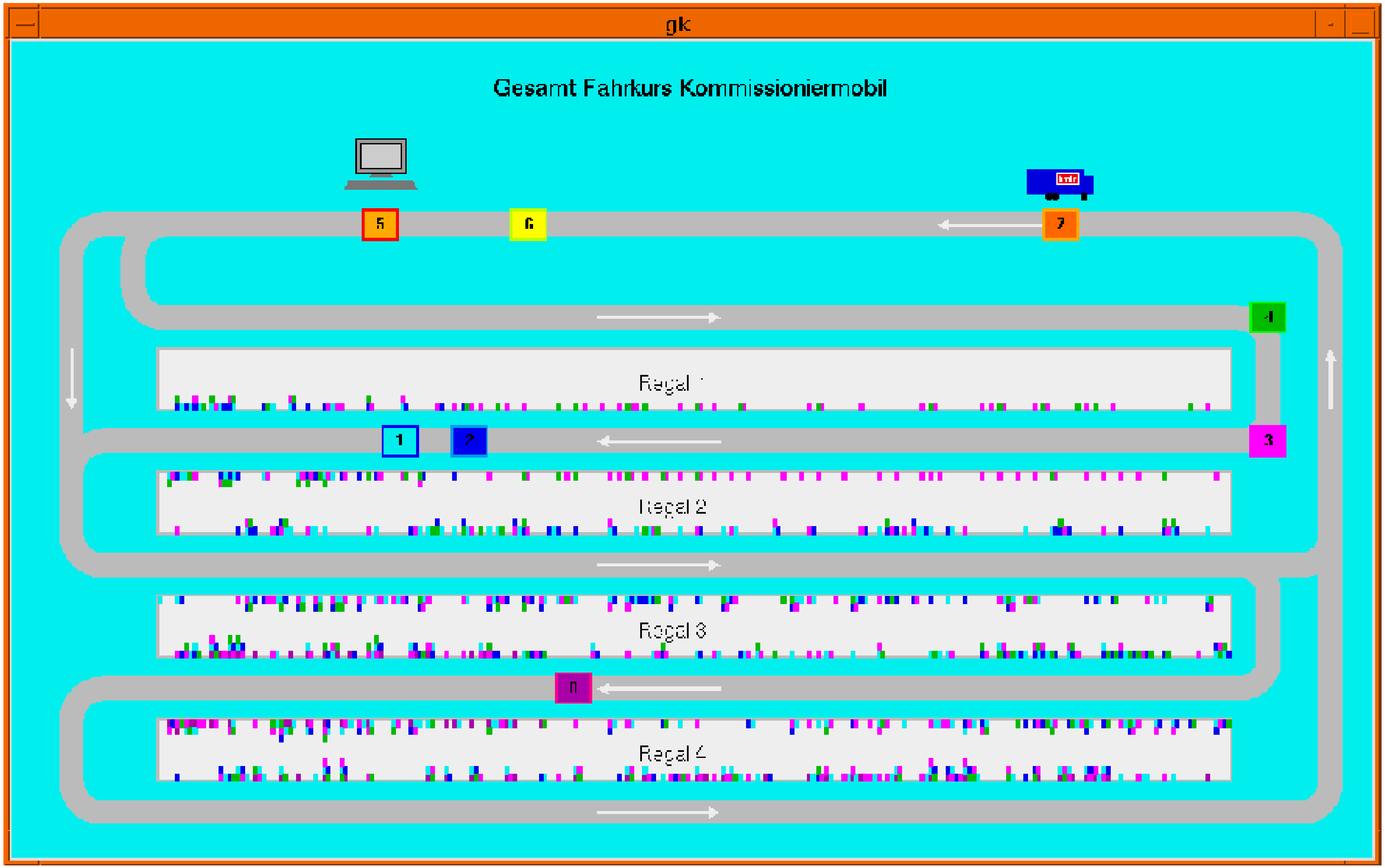


Figure 2 Commissioning area for greeting cards *(screenshot from the simulation program)*

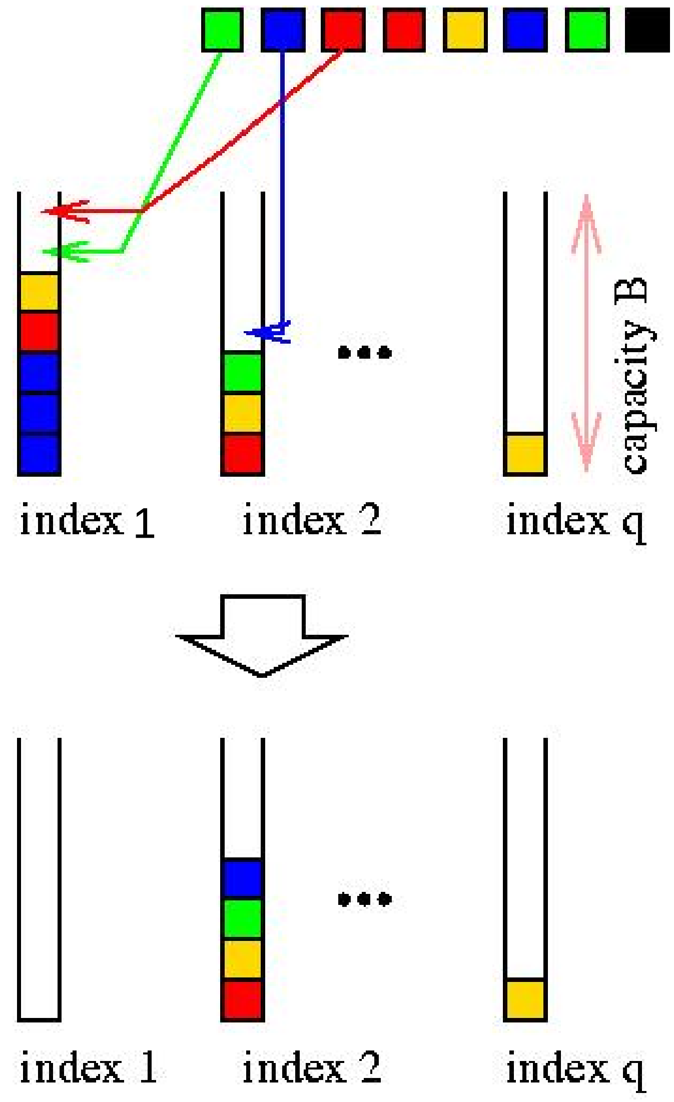
Online Bin Coloring

Given:

- sequence of colored items
- Bins with capacity B

Task:

- Pack items into bins such that there are always at most q open bins and
- Max. #colors in a bin is minimized

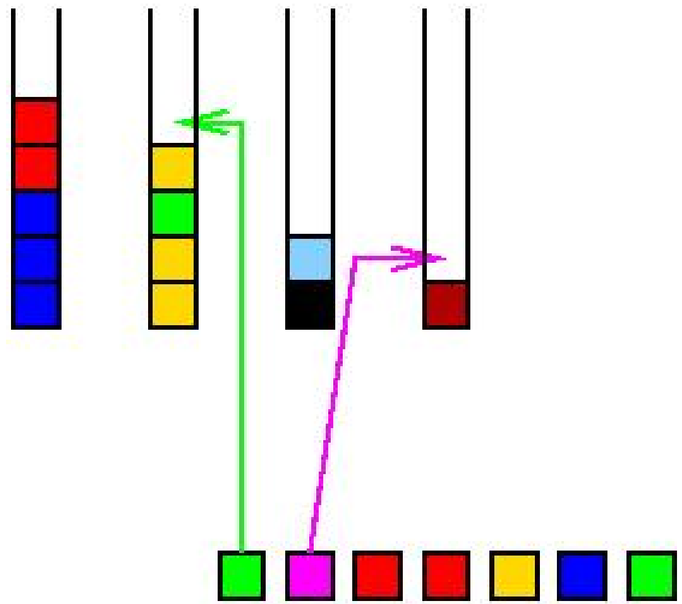


Martin Grötschel

The „Natural Greedy Algorithm“

GREEDYFIT:

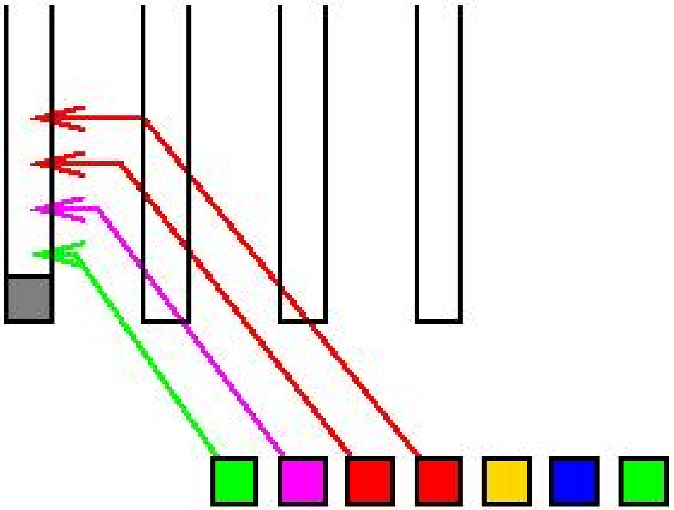
- If the current color is already present in a bin, use this bin.
- Otherwise use the bin with the least number of colors.



The Most Stupid Algorithm one can think of: „Onebin“

Onebin:

- use only the first bin



Theorem: *Onebin is (2q-1)-competitive for the bincoloring problem with q open bins allowed.*

What about GREEDYFIT?

Theorem: *Greedyfit is 3q-competitive, but no better than 2q-competitive.*

Recall: ONEBIN achieves a competitive ratio of (2q-1)

Question: Are there better algorithms?

Sure, there must be …

Martin Grötschel

The Big Disappointment

Theorem: *No deterministic algorithm can achieve a competitive ratio better than q.*

Theorem: *No randomized algorithm can achieve a competitive ratio better than q.*

Even worse: It does not even help to allow the online algorithm to keep q'>q bins open!

Contents

Herlitz Logistik

herlitz-logistik-1.wmv

Alternative analysis instruments 1

Comparative analysis

Alternative analysis instruments 2

Simulation

Some Literature

- M. Grötschel, S. O. Krumke, J. Rambau, Th. Winter, U. Zimmermann: *Combinatorial Online Optimization in Real Time*
- M. Grötschel, S. O. Krumke, J. Rambau: *Online Optimization of Complex Transportation Systems*
- M. Grötschel, S. O. Krumke, J. Rambau: *Online Optimization of Large Systems* Springer, 2001
- S. O. Krumke, J. Rambau: *Online Optimierung* Vorlesungsmanuskript, TU Berlin, U. Bayreuth, U Kaiserslautern, current version April 2005
- A. Borodin, R. El-Yaniv: *Online Computation and Competitive Analysis*, Cambridge Univ. Press, 1998
- A. Fiat, G. J. Woeginger: *Online Algorithms: The State of the Art*, Springer, 1998

06M1 Lecture
Online Optimization

The End

Martin Grötschel

- Institut für Mathematik, Technische Universität Berlin (TUB)
- DFG-Forschungszentrum “Mathematik für Schlüsseltechnologien” (MATHEON)
- Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

groetschel@zib.de http://www.zib.de/groetschel