Deferment Control for Reoptimization

How to Find Fair Reoptimized Dispatches

Jörg Rambau

LS Wirtschaftsmathematik

- Two Real-World Dynamic Optimization Problems
- ILP Reoptimization Policies and Infinite-Deferment
- Why Classical Evaluation Methods Fail
- How to Obtain Performance Guarantees
- Summary and Remarks

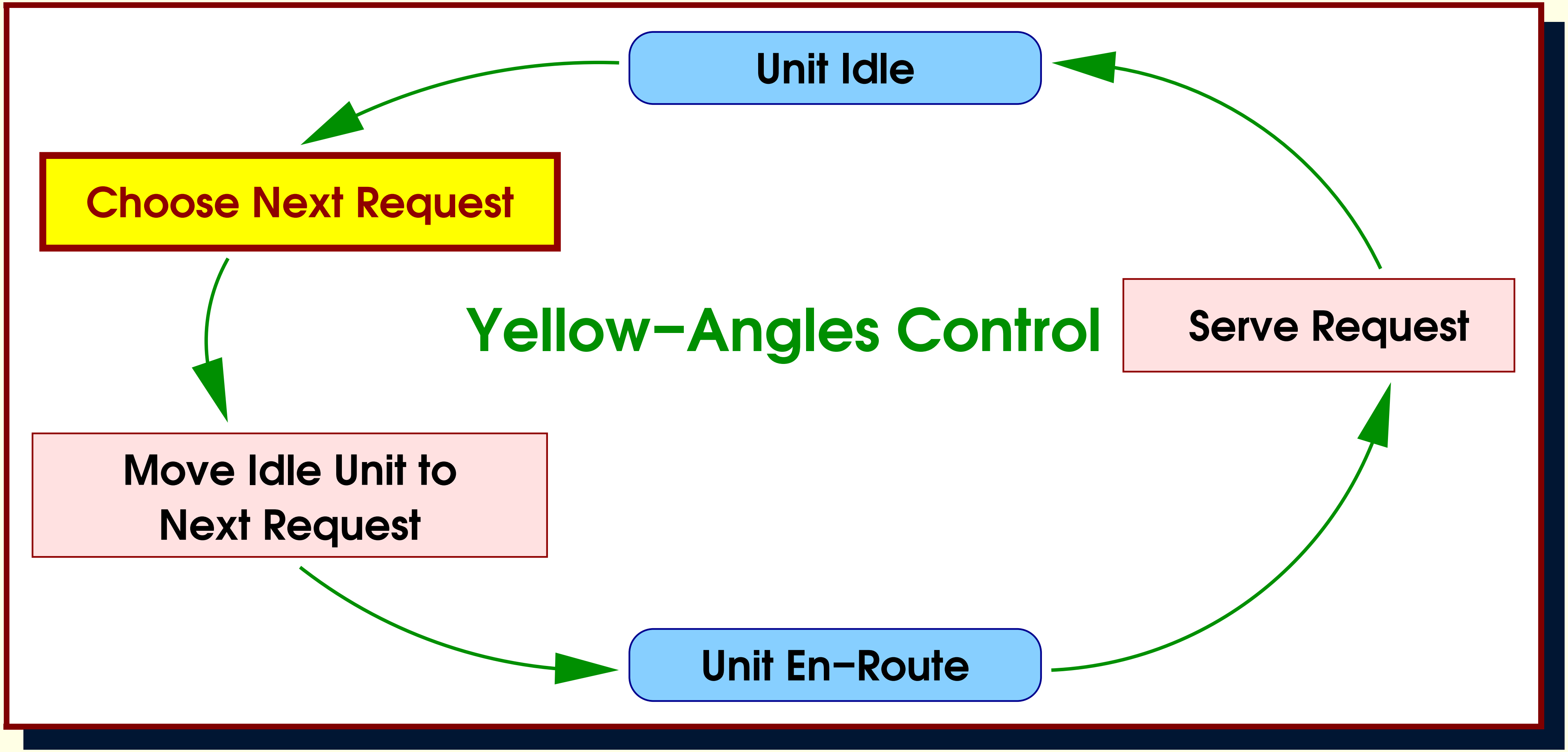
http://www.wm.uni-bayreuth.de

UNIVERSITÄT BAYREUTH

Vehicle Dispatching for ADAC

Team:	Sven O. Krumke, Benjamin Hiller Luis Miguel Torres
System:	~1,700 service units of ADAC ~5,000 service contractors 5 help centers
Task:	requests $\longrightarrow$ units/contractors units $\longrightarrow$ tours
Goals:	productivity & service quality
Before Project:	geographic clustering manual dispatching
Project Goal:	automatic decision support

Yellow Angels Control Cycle



Elevator Group Control for Herlitz

Team: Sven O. Krumke, Philipp Friese

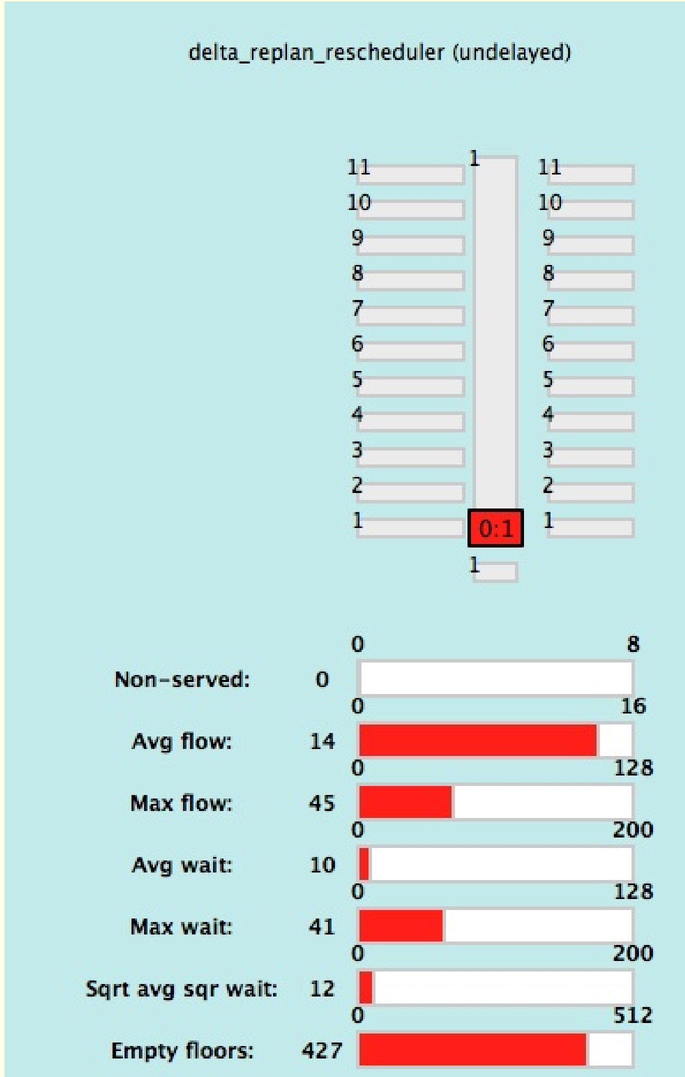
System: pallet transportation Herlitz PBS AG, Falkensee near Berlin

Task: requests $\longrightarrow$ elevators
elevators $\longrightarrow$ schedules

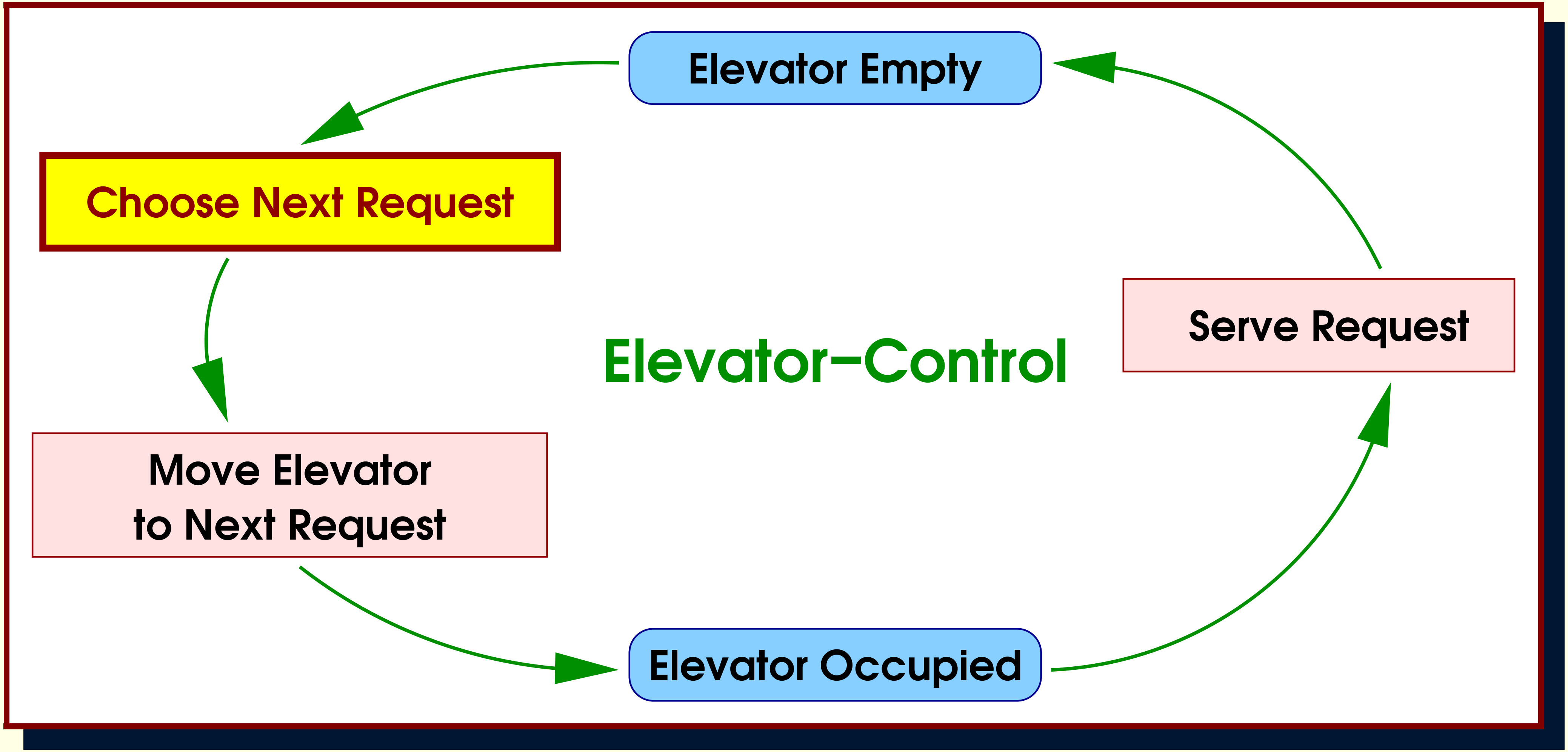
Goals: productivity & service quality

Before Project: choice between:
FIRSTFIT+FIFO or
FIRSTFIT+NEARESTNEIGHBOR

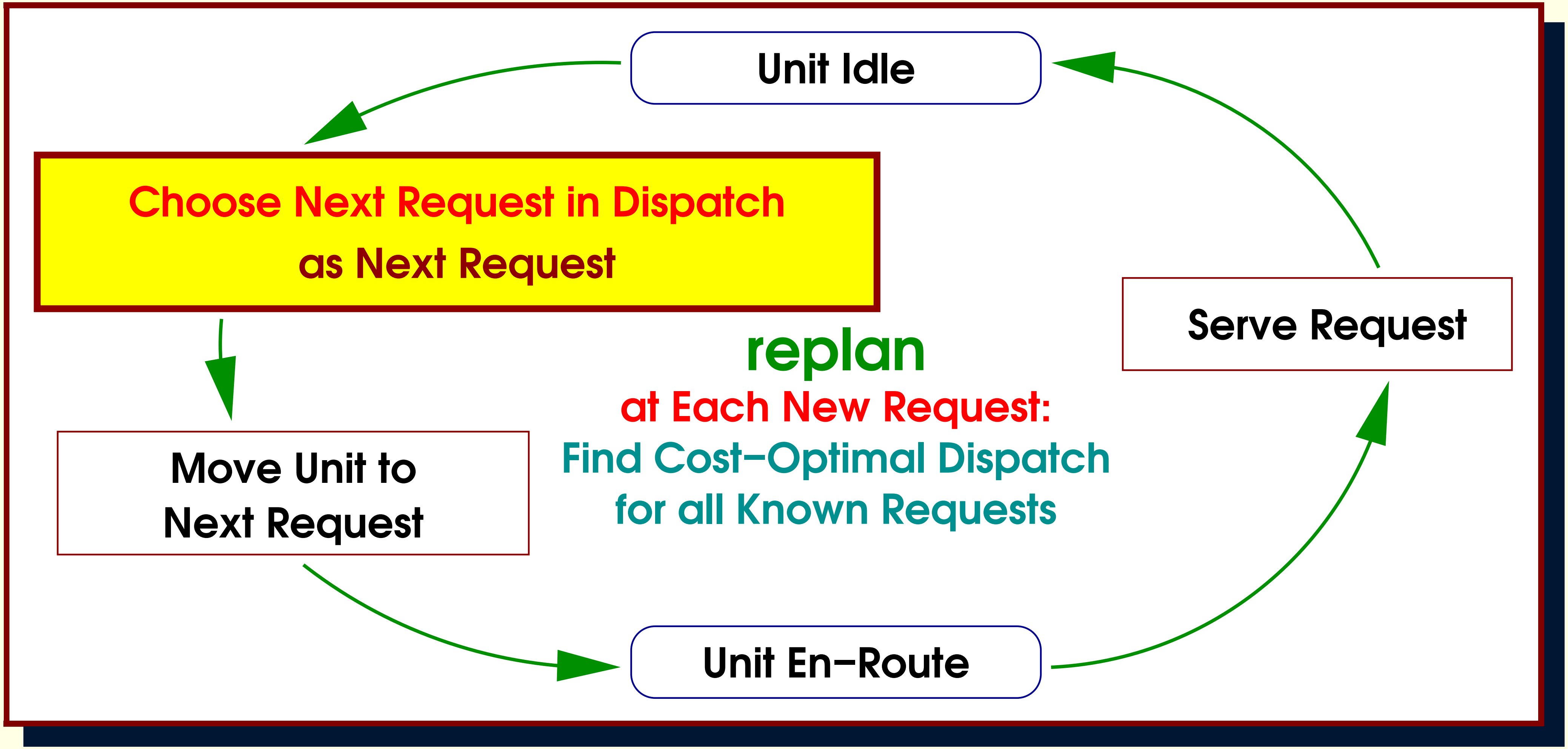
Project Goal: efficient control with manageble deferment



The Elevator Control Cycle

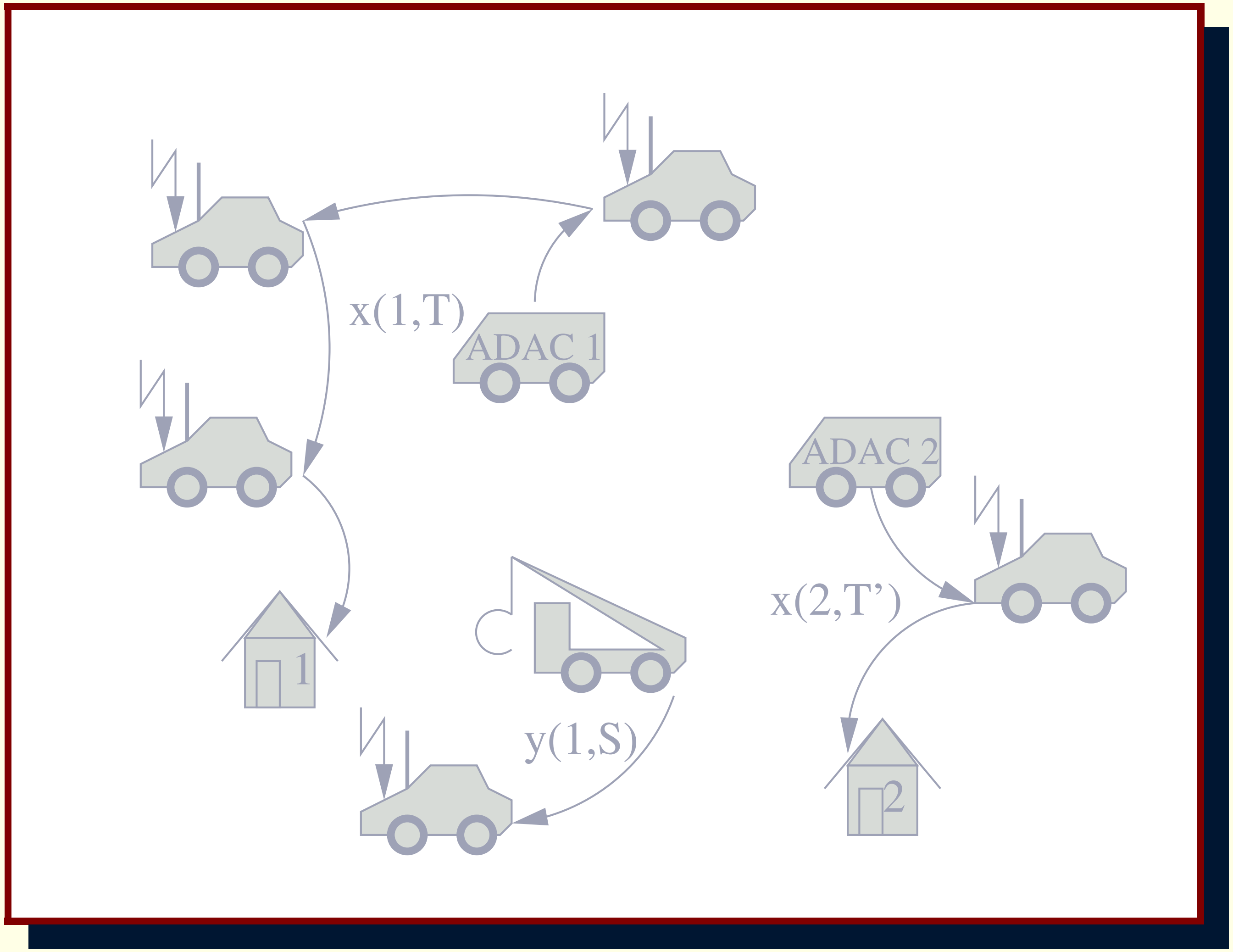


The Reoptimization Policy for the ADAC Problem



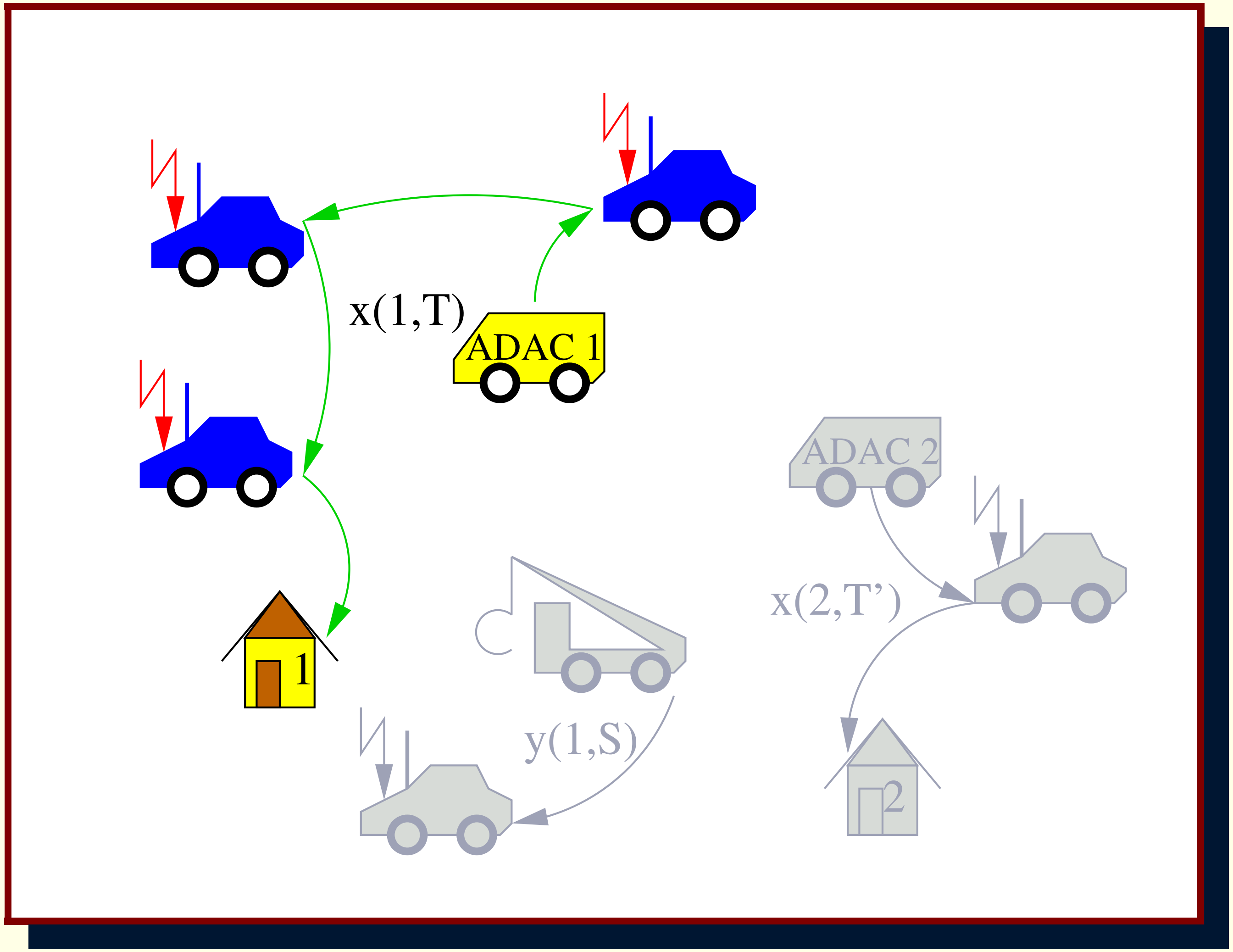
The ILP Reoptimization Model for ADAC

The ILP Reoptimization Model for ADAC



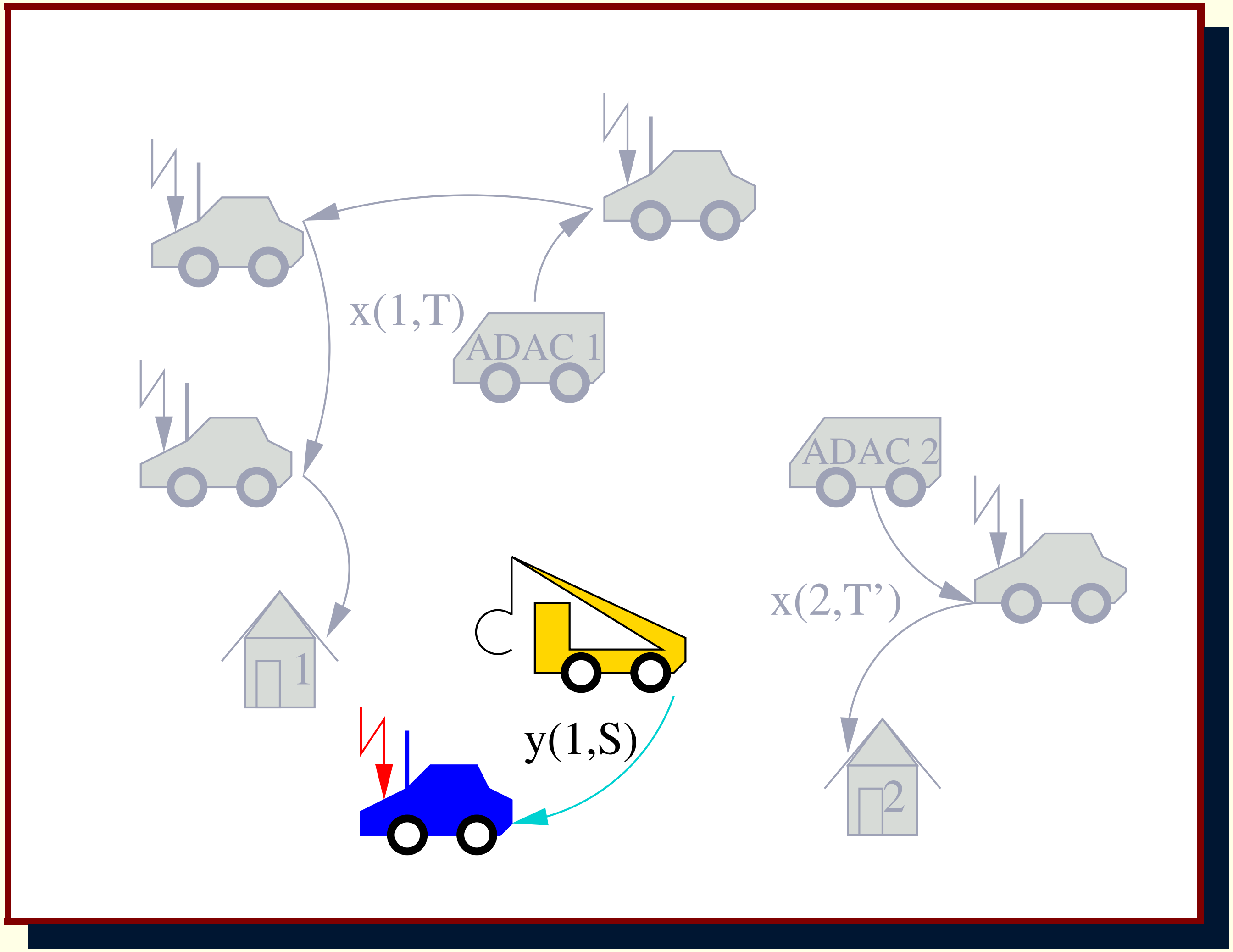
Model with
tour variables
for units and partners.

The ILP Reoptimization Model for ADAC



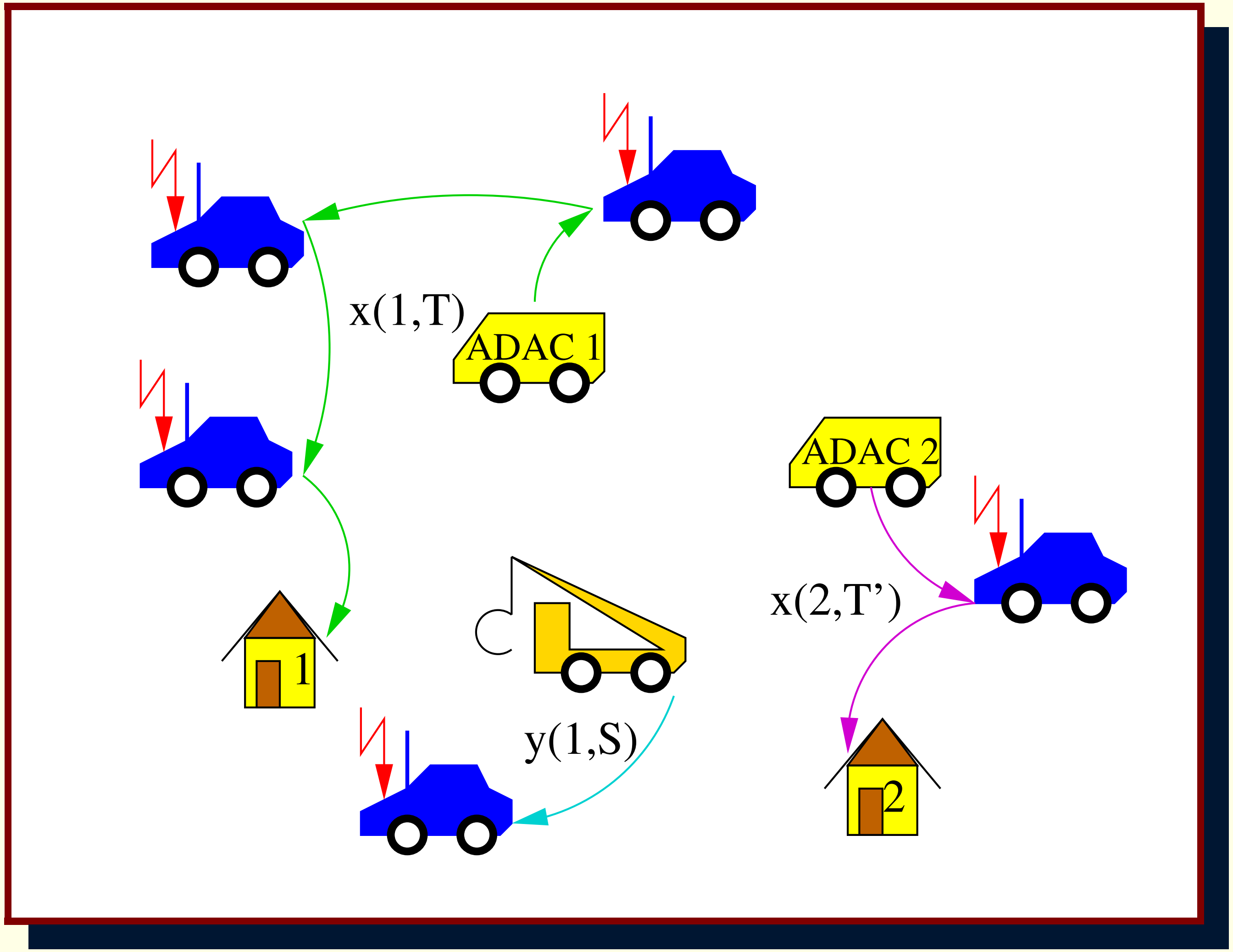
Vehicle 1 goes along Tour T.

The ILP Reoptimization Model for ADAC



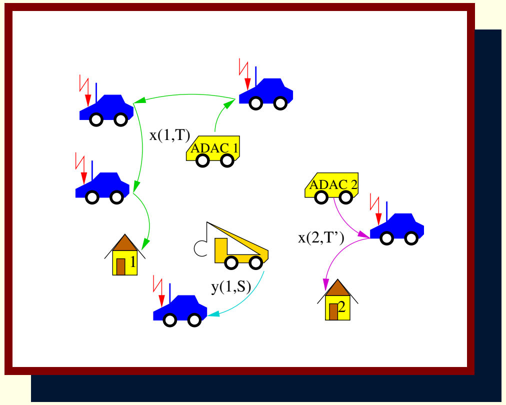
Contractor 1 is assigned Requests S.

The ILP Reoptimization Model for ADAC



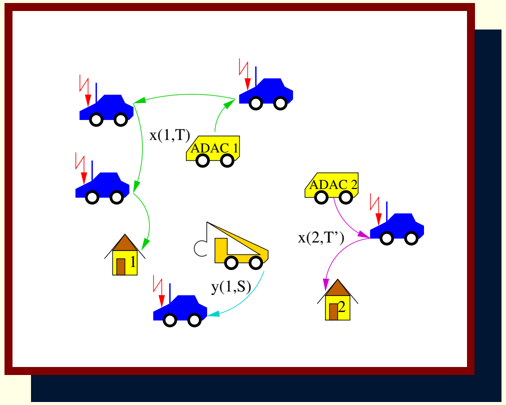
Feasible Solution:
partition of requests
into tours.

The ILP Reoptimization Model for ADAC



$$
\begin{aligned}
\min \sum_{T \in \mathcal{T}} c_T x_T \quad & \text{s.t.} && \\
\sum_{T \in \mathcal{T}} a_{\nu T} x_T = 1 \quad & \forall \text{requests } \nu && \text{(Partitioning Requests)} \\
\sum_{T \in \mathcal{T}_u} x_T = 1 \quad & \forall \text{units } u && \text{(Partitioning Units)} \\
x_T \in \{0, 1\} \quad & \forall T \in \mathcal{T} && \text{(Binary Variables)}
\end{aligned}
$$

The ILP Reoptimization Model for ADAC



$$\min \sum_{T \in \mathcal{T}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \mathcal{T}} a_{vT} x_T = 1 \qquad \forall \text{requests } v \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \mathcal{T}_u} x_T = 1 \qquad \forall \text{units } u \qquad \text{(Partitioning Units)}$$

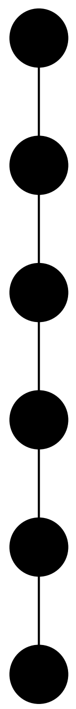
$$x_T \in \{0, 1\} \qquad \forall T \in \mathcal{T} \qquad \text{(Binary Variables)}$$

(No) Problem:

In practice $\sim$100.000.000.000 variables $\Rightarrow$ Dynamic Column Generation

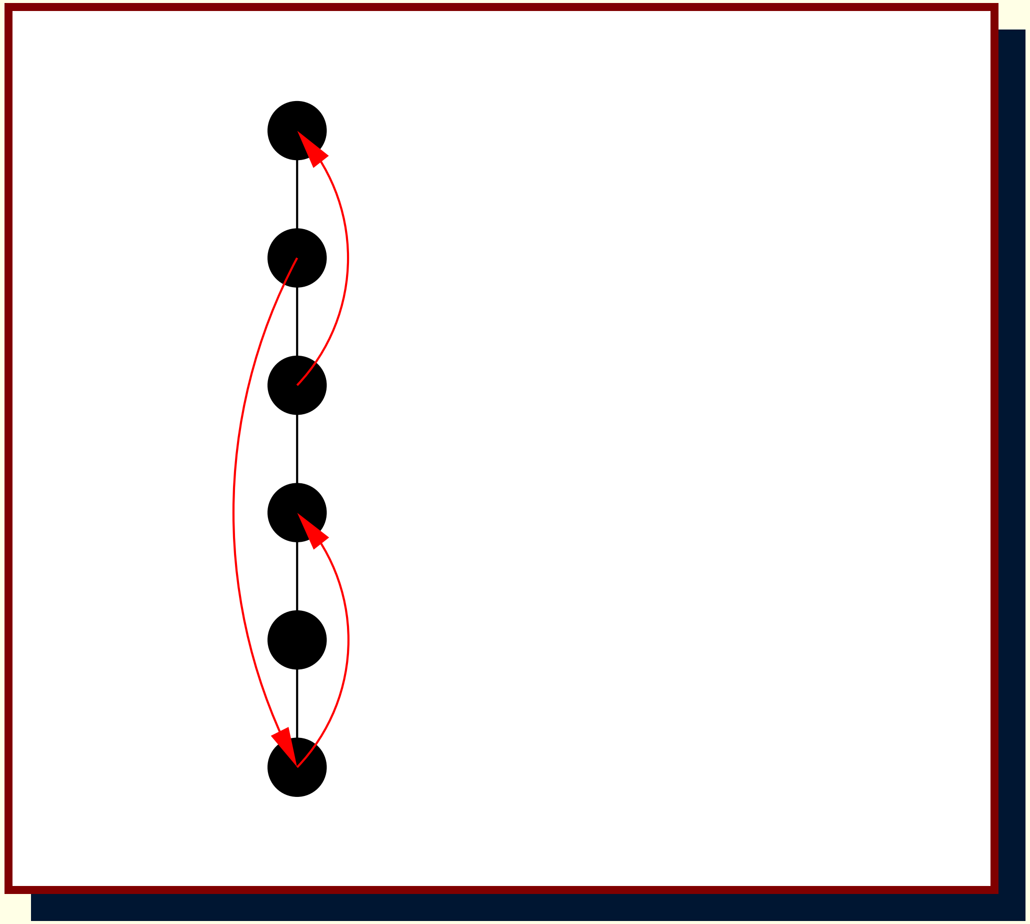
The ILP Reoptimization Model for Elevator Groups

The ILP Reoptimization Model for Elevator Groups



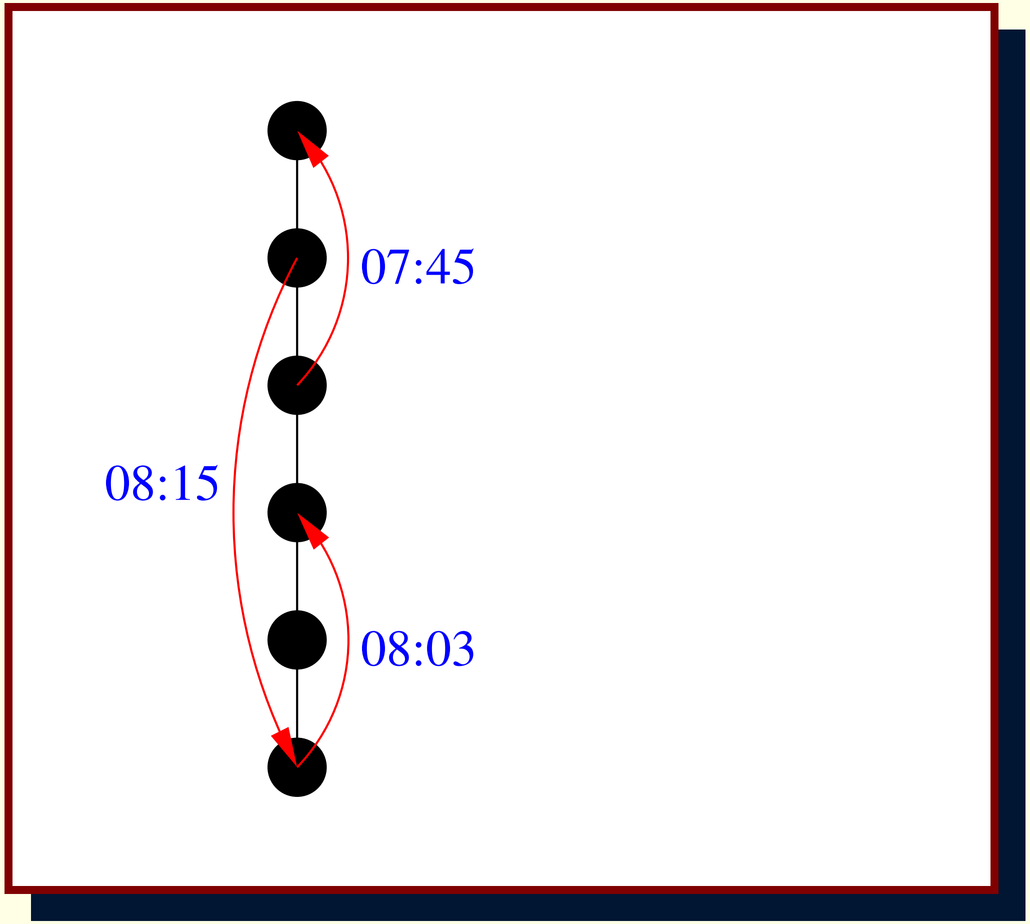
The transport graph for one elevator.

The ILP Reoptimization Model for Elevator Groups



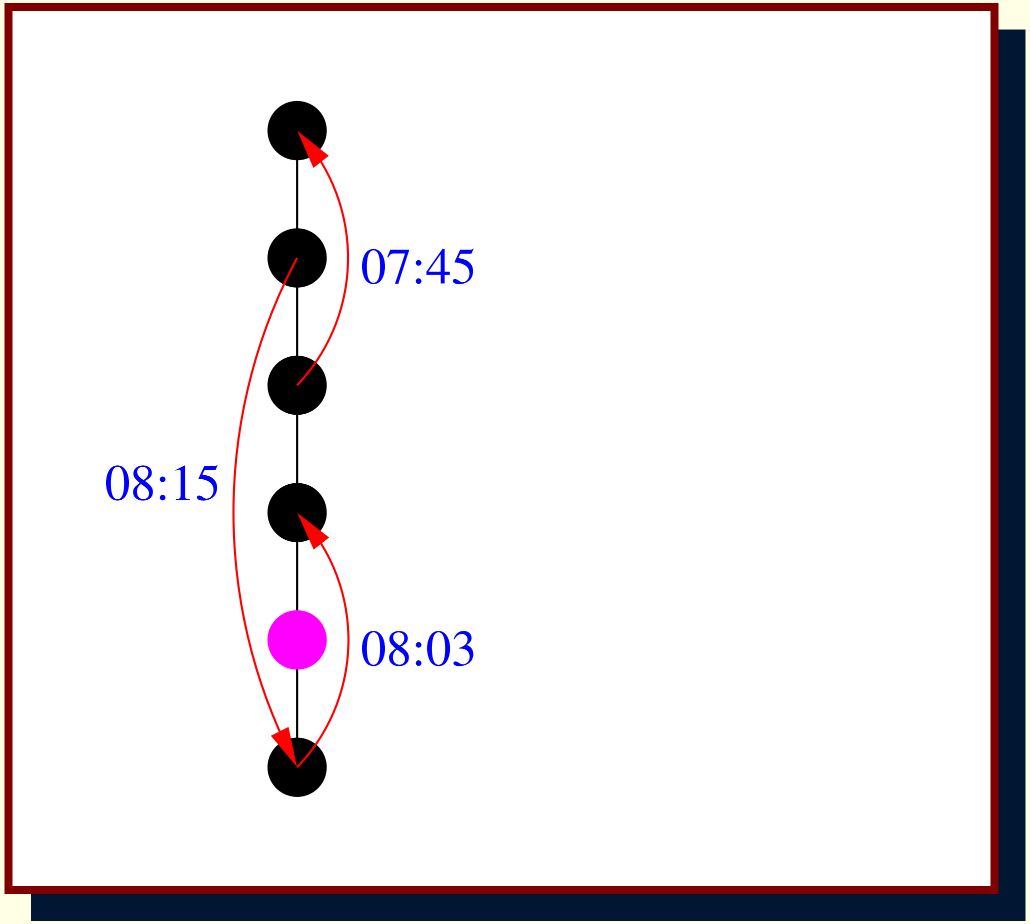
Some requests.

The ILP Reoptimization Model for Elevator Groups



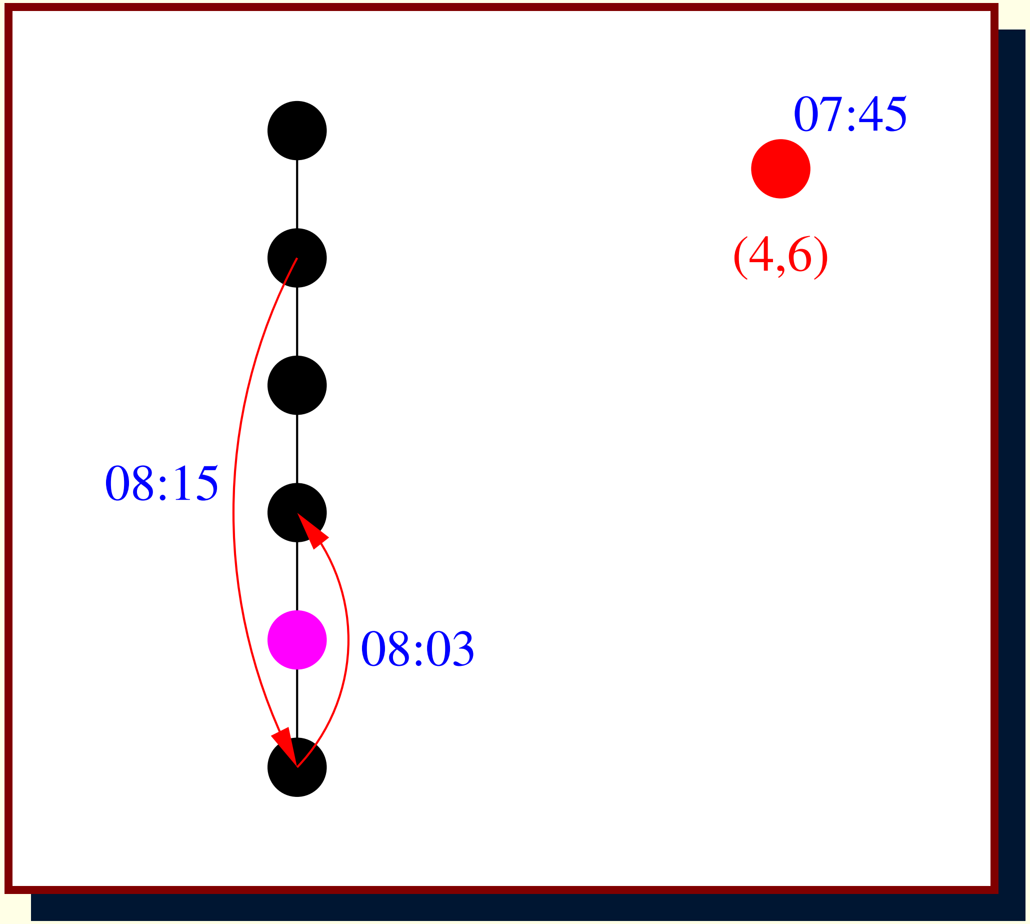
Their time stamps.

The ILP Reoptimization Model for Elevator Groups



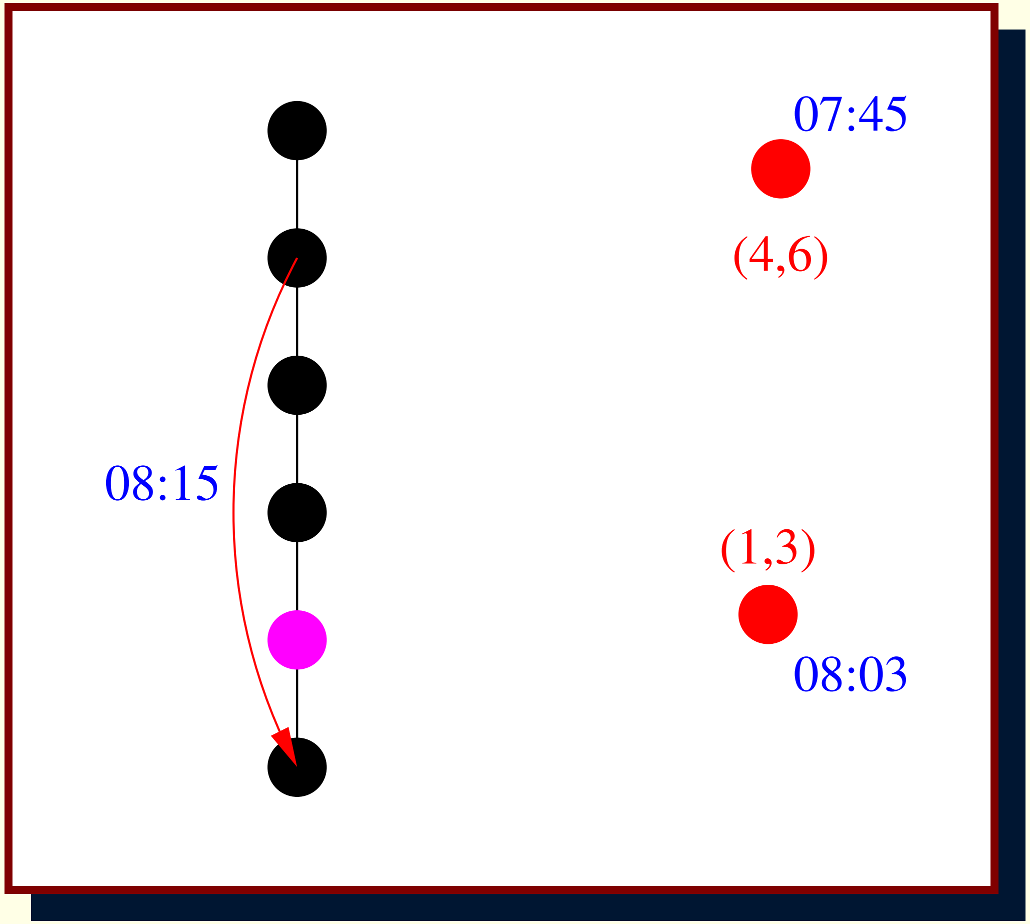
The position of the elevator.

The ILP Reoptimization Model for Elevator Groups



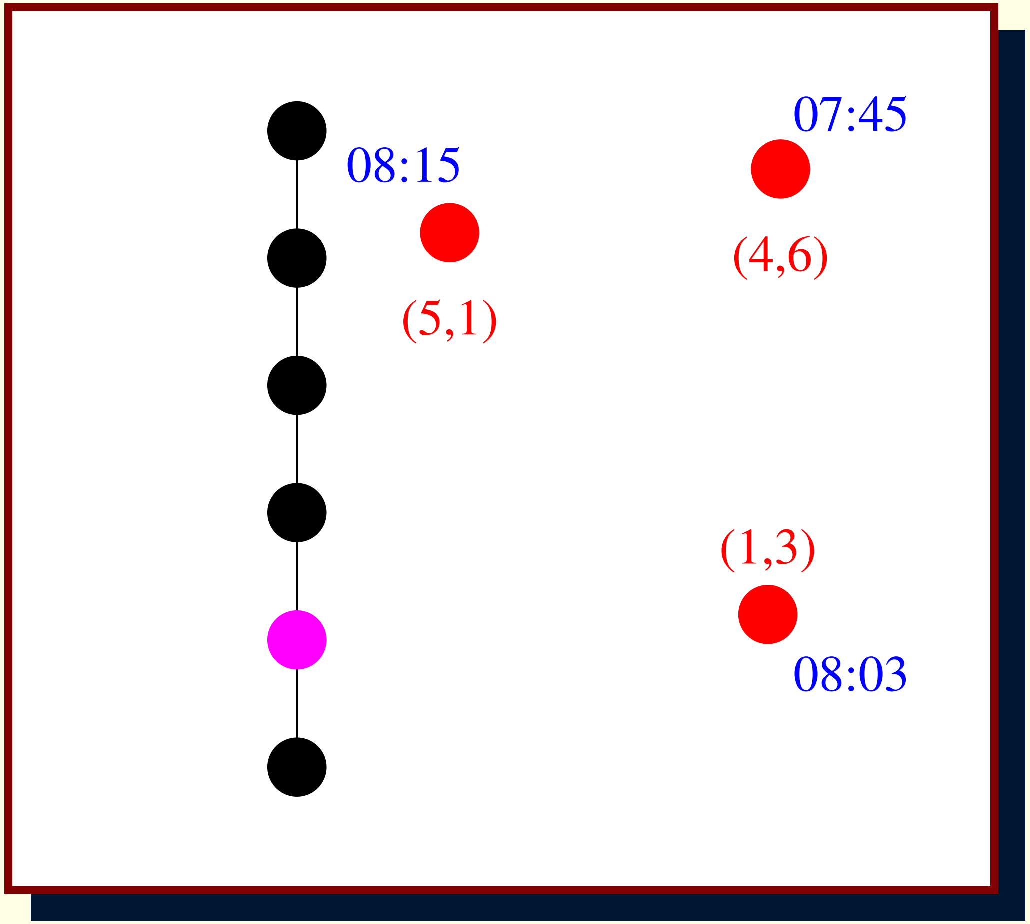
Each request can be seen as a node.

The ILP Reoptimization Model for Elevator Groups



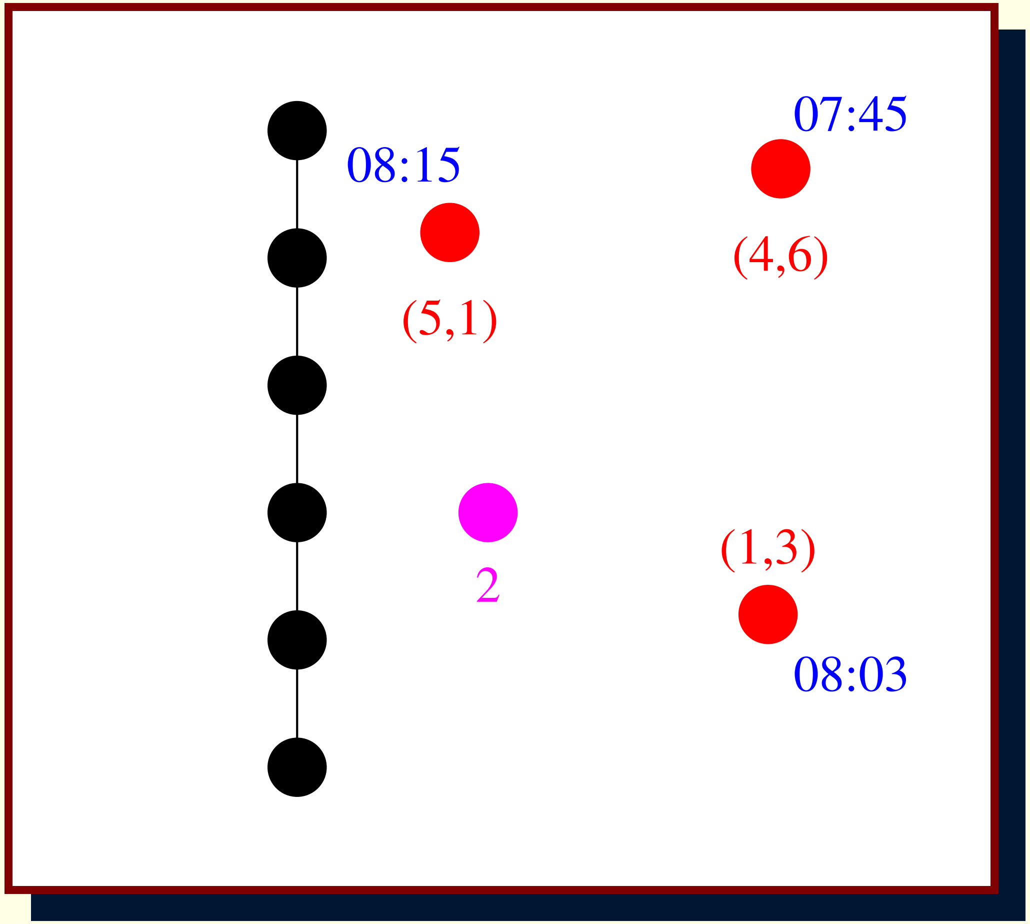
Each request can be seen as a node.

The ILP Reoptimization Model for Elevator Groups



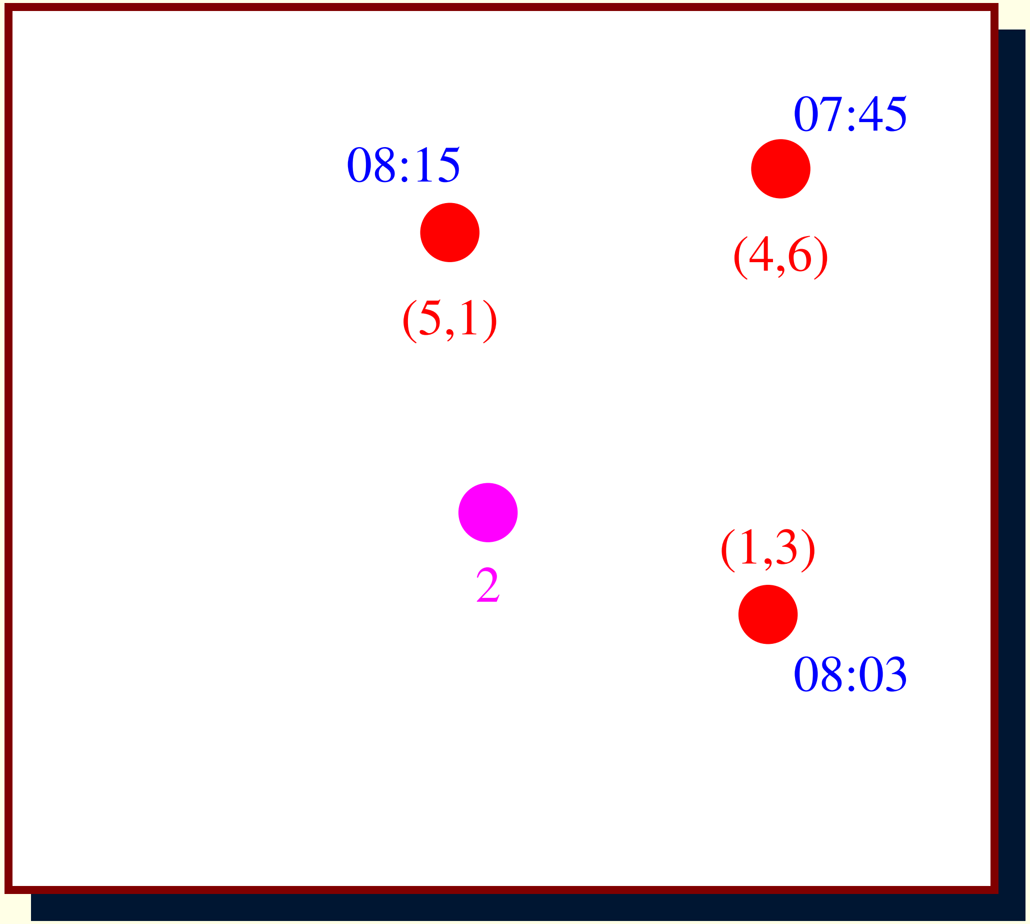
Each request can be seen as a node.

The ILP Reoptimization Model for Elevator Groups



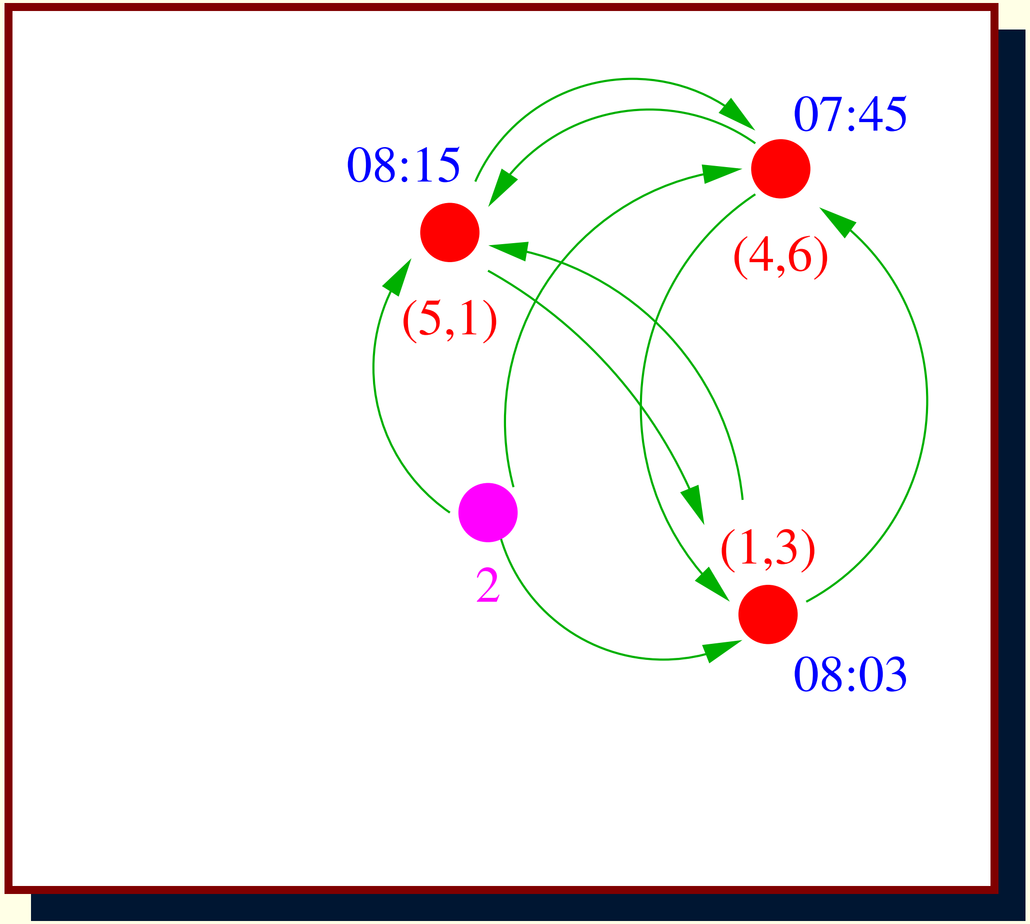
The elevator is a special node.

The ILP Reoptimization Model for Elevator Groups



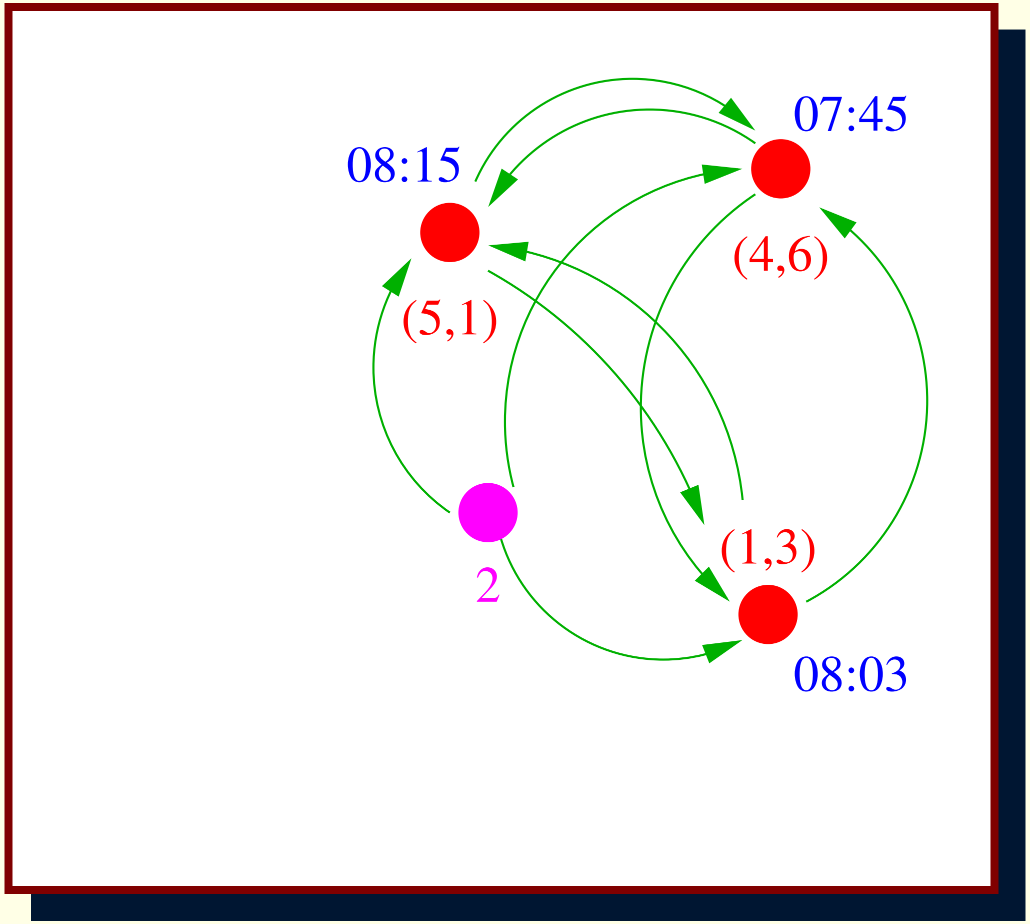
The connecting moves between requests . . .

The ILP Reoptimization Model for Elevator Groups



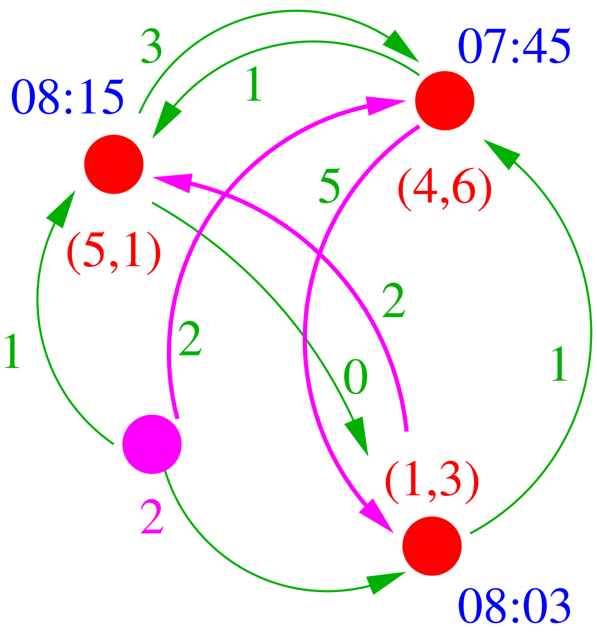
. . . are arcs . . .

The ILP Reoptimization Model for Elevator Groups



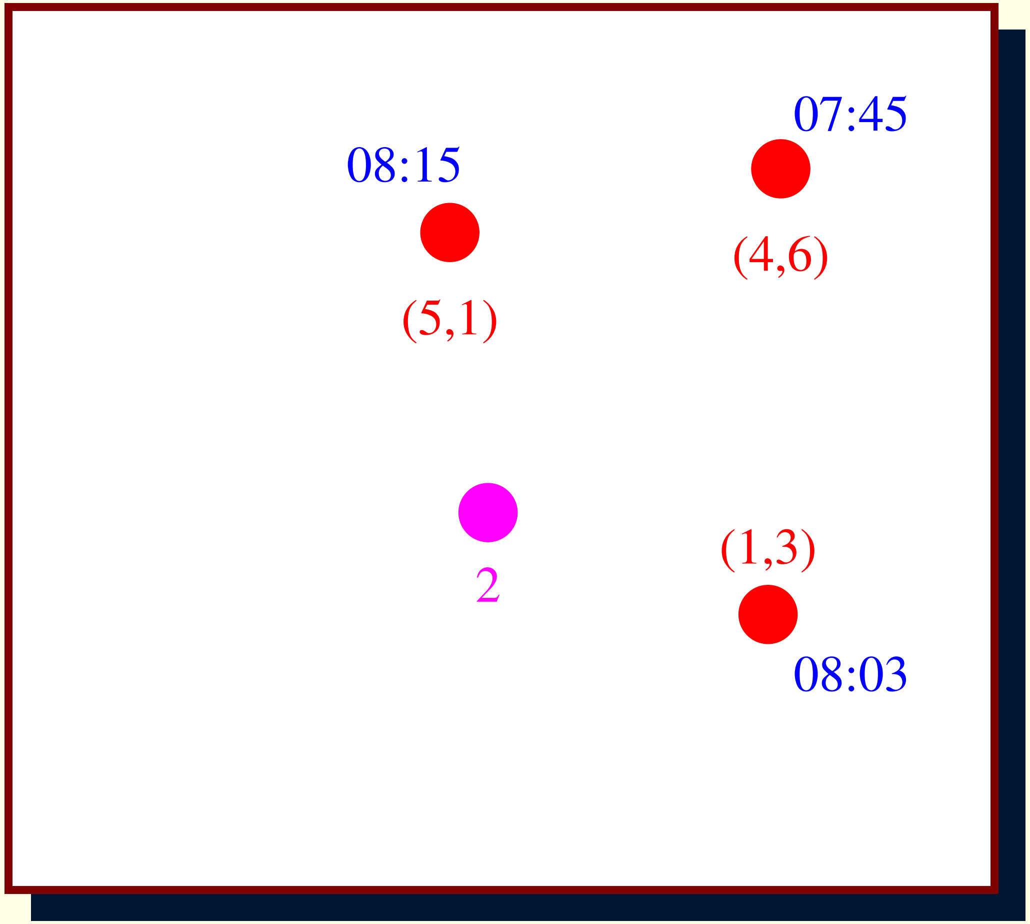
… whose weights are empty moves.

The ILP Reoptimization Model for Elevator Groups



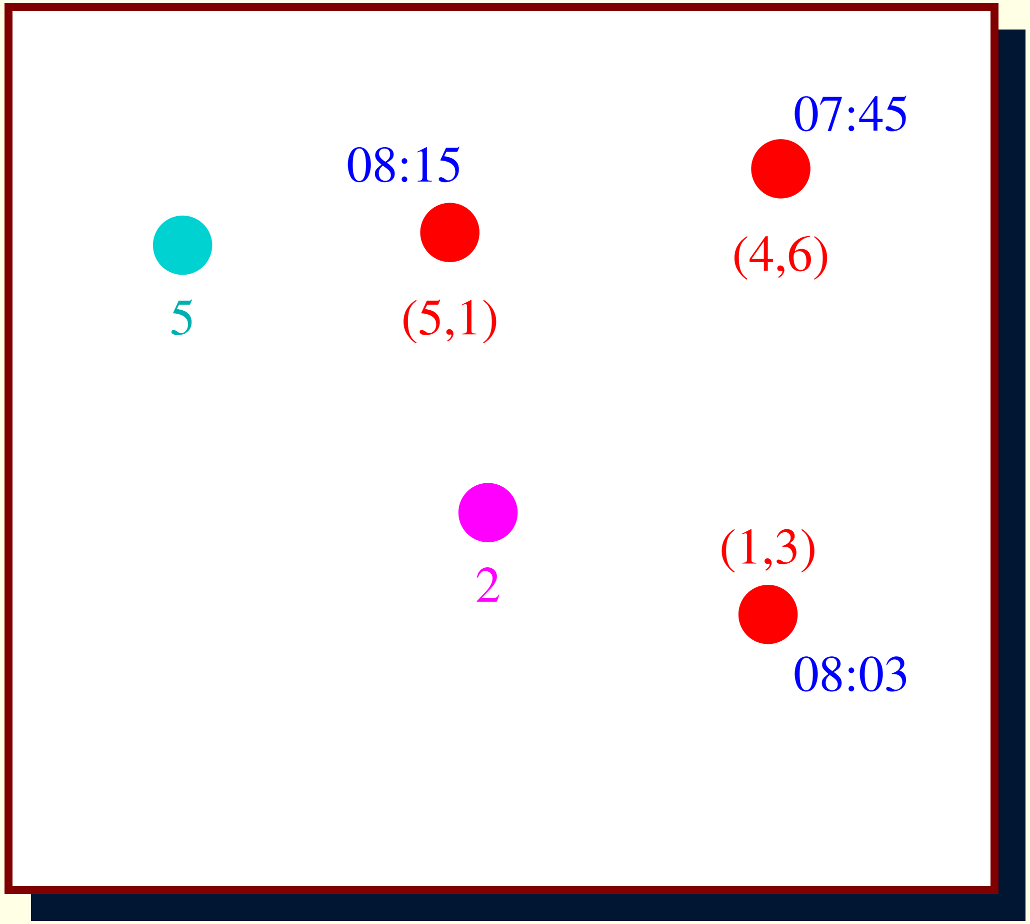
A feasible dispatch
is a tour through all nodes
starting at the elevator's node
with precedence conditions on each floor.

The ILP Reoptimization Model for Elevator Groups



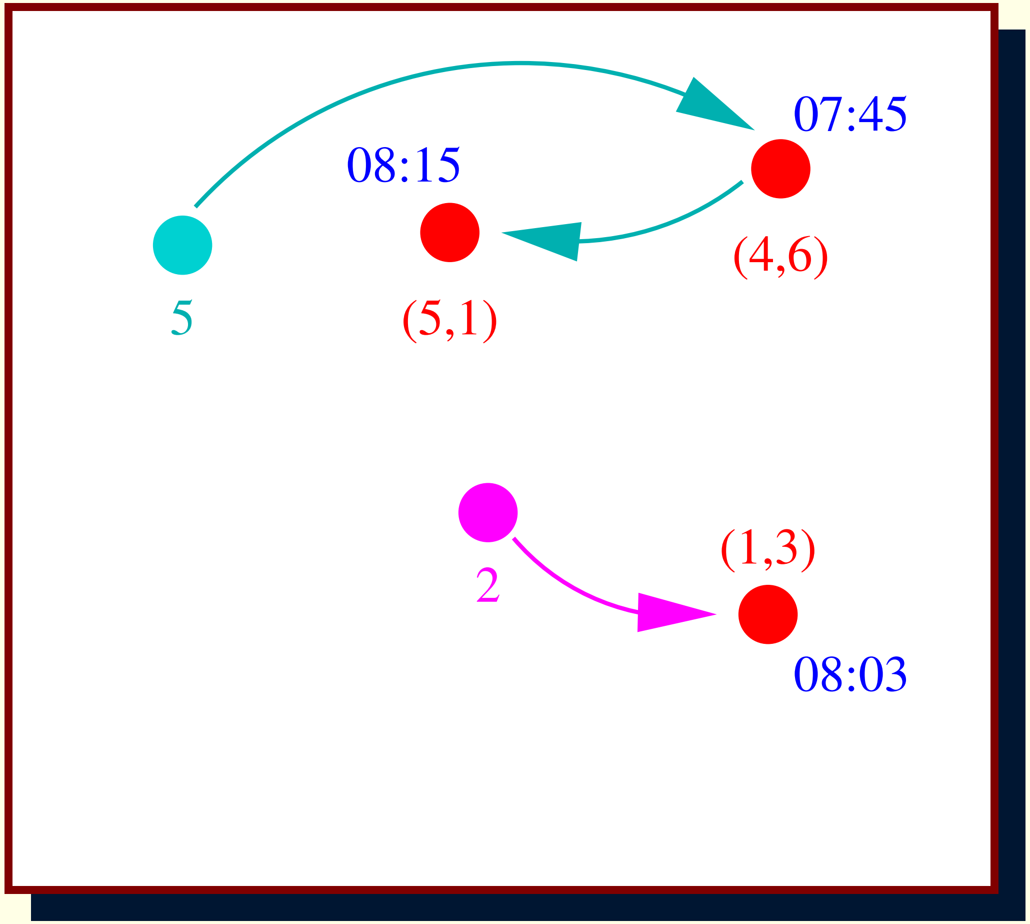
We do not show the arcs anymore since they are implicitly given.

The ILP Reoptimization Model for Elevator Groups



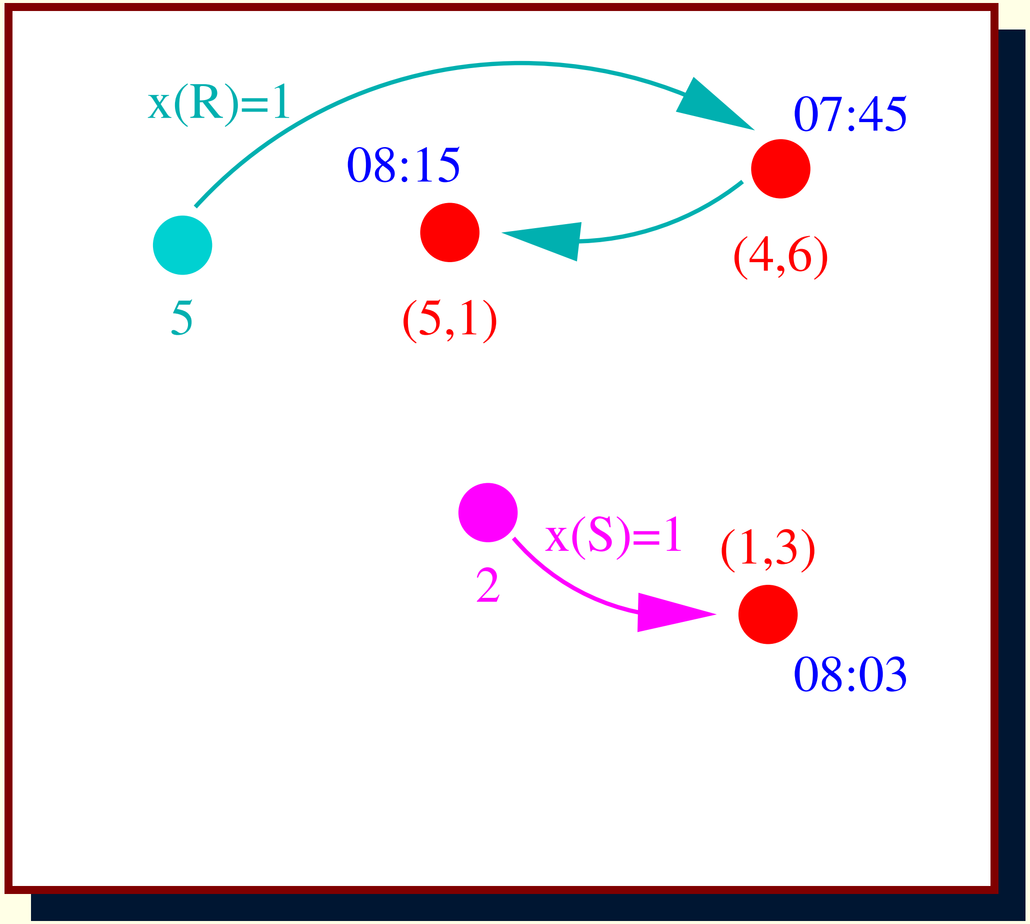
If there is another elevator . . .

The ILP Reoptimization Model for Elevator Groups



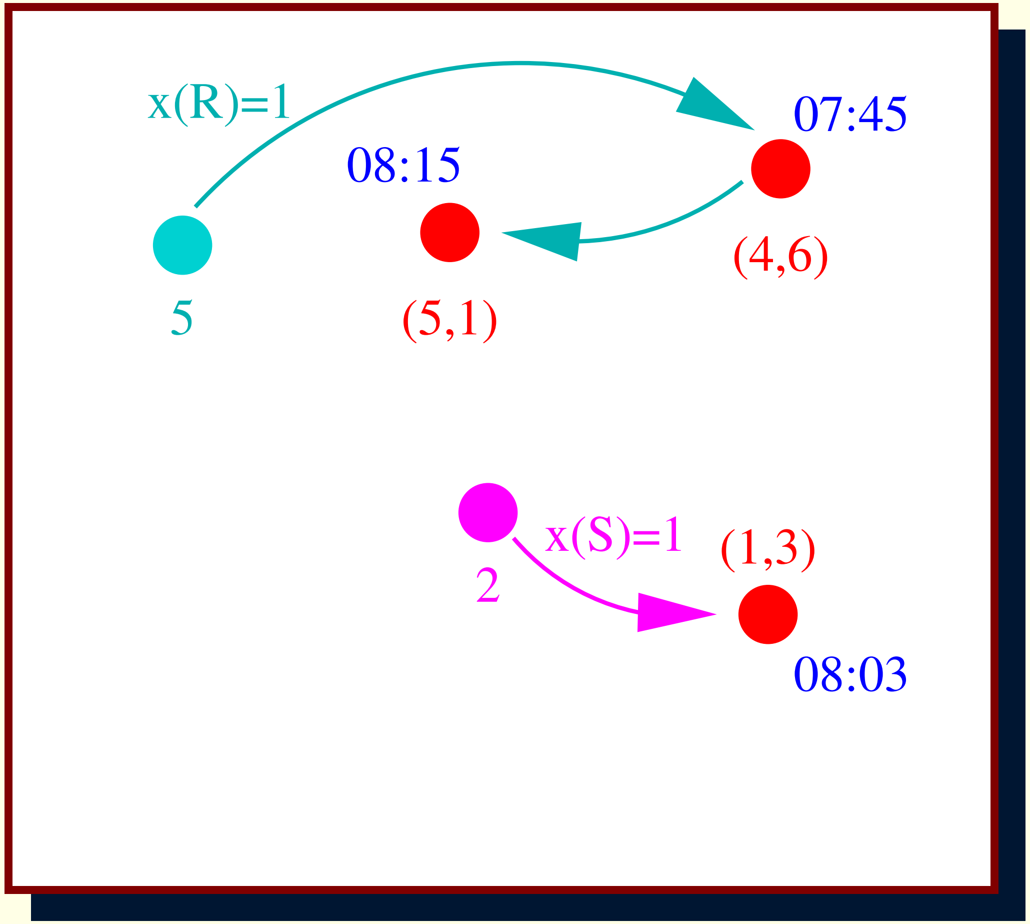
… then a feasible solution is a
partitioning
of requests into tours
with precedence constraints
on each floor.

The ILP Reoptimization Model for Elevator Groups



A tour-variable model contains a variable for each feasible tour of a server.

The ILP Reoptimization Model for Elevator Groups

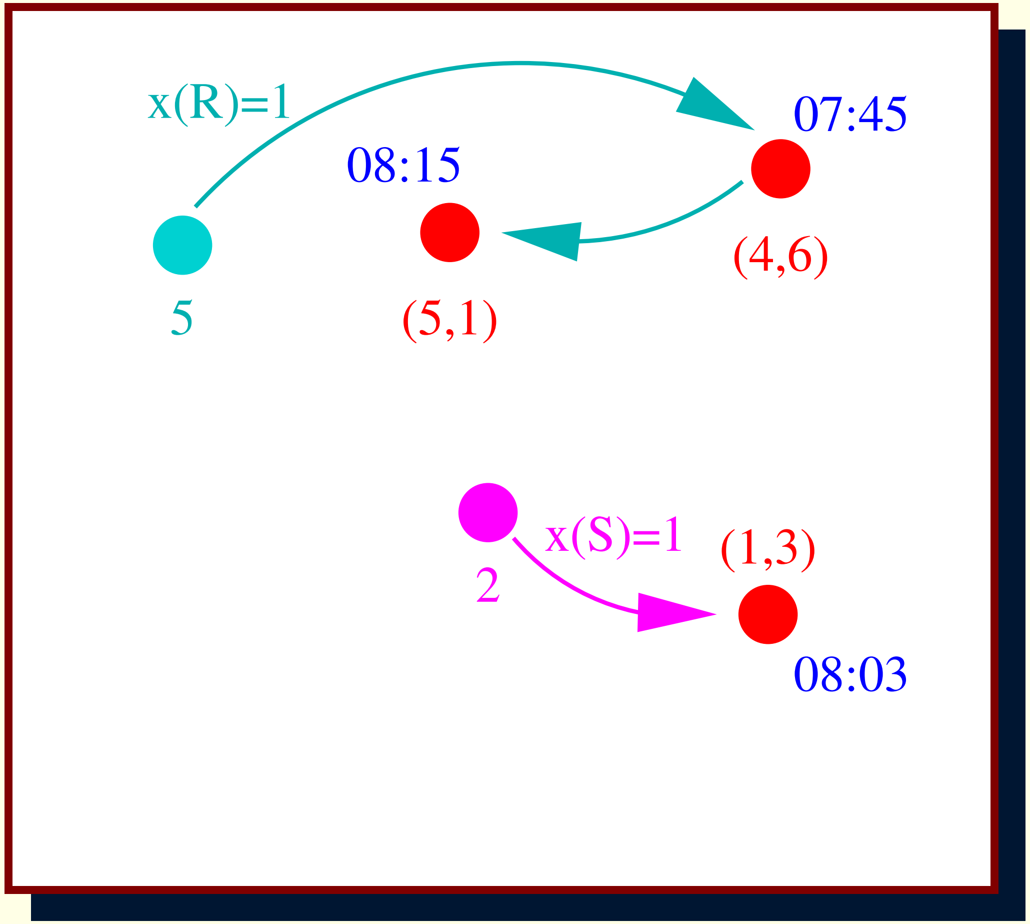


A tour-variable model contains a variable for each feasible tour of a server.

(No) Problem:

astronomic number of variables

The ILP Reoptimization Model for Elevator Groups



A tour-variable model contains a variable for each feasible tour of a server.

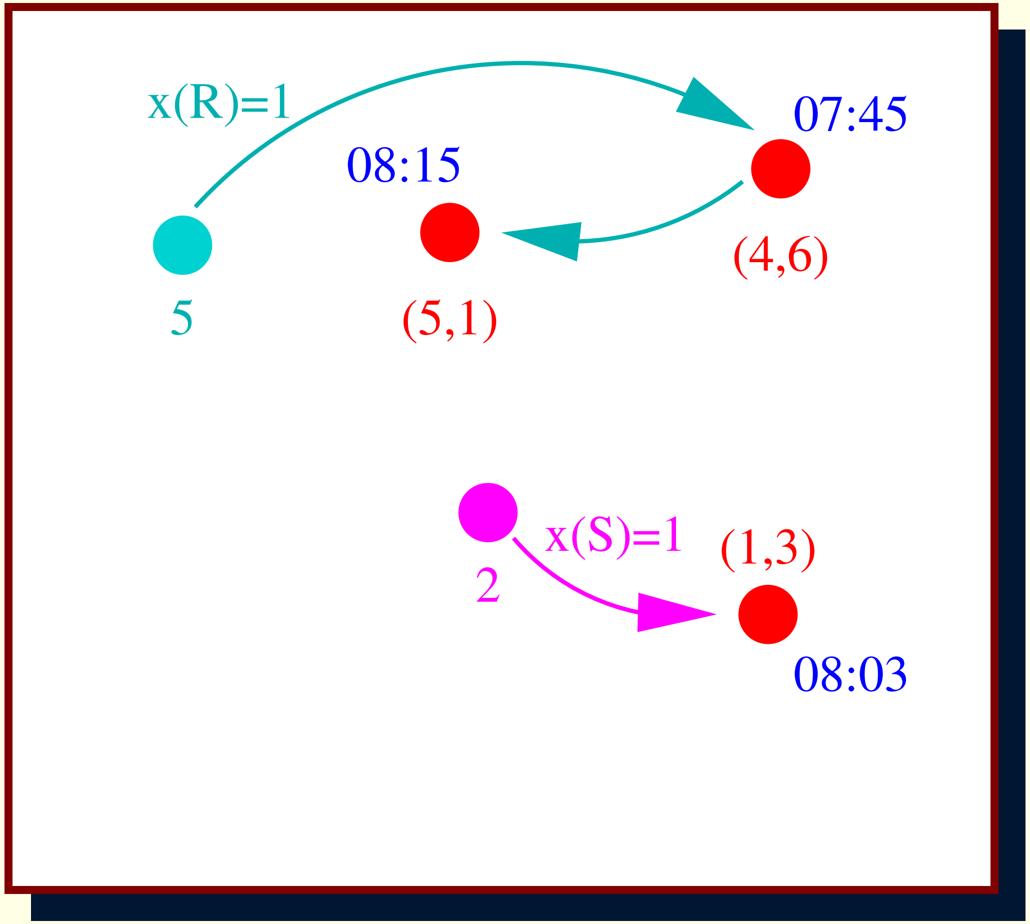
(No) Problem:

astronomic number of variables

Solution:

dynamic column generation

The ILP Reoptimization Model for Elevator Groups



A tour-variable model contains a variable for each feasible tour of a server.

(No) Problem:

astronomic number of variables

Solution:

dynamic column generation

Precedence Constraints:

Good for us!

The Infinite Deferment Problem

The Infinite Deferment Problem

- Depending on the objective, individual requests maybe deferred arbitrarily.
- Infinite deferment unwanted even if original objective does not penalize this.
- Example: Minimize empty moves, minimize total flow time, . . .

The Infinite Deferment Problem

- Depending on the objective, individual requests maybe deferred arbitrarily.
- Infinite deferment unwanted even if original objective does not penalize this.
- Example: Minimize empty moves, minimize total flow time, . . .

Goal:

Minimize (expected) objective function value so that
the maximal flow time of each request is bounded by a constant
(constant may depend on the system load but not on the instance)

Universität Bayreuth

Insecure Information Aspect

Observation:
A currently good-looking decision
– i.p. when applied repeatedly –
may prove bad in the long run because of
insecure or even no information about future requests.

Classical approaches to cope with insecure information about future requests:

With Stochastic Info:	Stochastic (Dynamic) Programming (Expected Performance)
Without Stochastic Info:	Competitive Analysis (Worst-Case Performance)

STOCHASTIC DYNAMIC PROGRAMMING/MARKOV DECISON PROCESSES

Classical computational methods rely on computing
the optimal cost function for all states.

Problem:

- Stochastic information about future requests required.
- $> (m-1)^{mk} m^e$ states for e elevators, m floors, and k slots.

$e = 1$, $m = 8$, $k = 2$: $> 265,863,444,556,808$ states.

$e = 5$, $m = 8$, $k = 1$: $> 188,900,999,168$ states

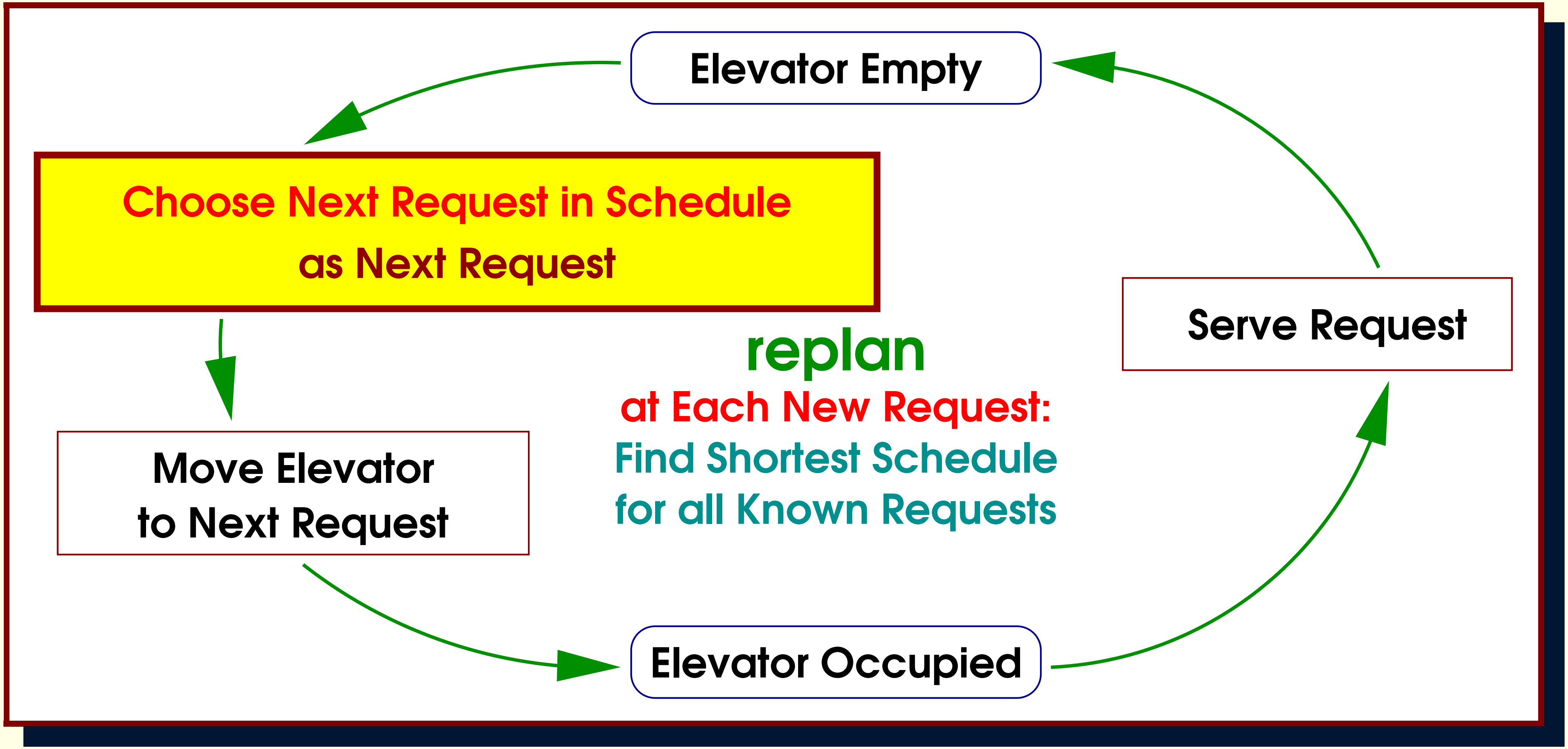
Competitive Analysis: Good News For A Single Elevator

COMPETITIVE ANALYSIS: GOOD NEWS FOR A SINGLE ELEVATOR

Theorem [Ascheuer, Krumke, R. 2000]:

- REPLAN is 2.5-competitive for makespan minimization.
- There is a 2-competitive online-algorithm for makespan minimization.

SINGLE ELEVATOR CONTROL: REPLAN



Competitive Analysis: Bad News

COMPETITIVE ANALYSIS: BAD NEWS

Observations:

Minimizing the long-term makespan for an elevator group/the ADAC fleet is absolutely useless.

COMPETITIVE ANALYSIS: BAD NEWS

Observations:

Minimizing the long-term makespan for an elevator group/the ADAC fleet is absolutely useless.

There is no competitive online-algorithm for max./avg. flow/waiting time minimization

COMPETITIVE ANALYSIS: BAD NEWS

Observations:

Minimizing the long-term makespan for an elevator group/the ADAC fleet is absolutely useless.

There is no competitive online-algorithm
for max./avg. flow/waiting time minimization

Problem: The task doesn't go away!

ALTERNATIVE PERFORMANCE MEASURE: GOOD NEWS

ALTERNATIVE PERFORMANCE MEASURE: GOOD NEWS

requests Δ-reasonable

$:\Longleftrightarrow$

requests presented in time δ
can be served in time at most δ whenever $\delta \geq \Delta$.

ALTERNATIVE PERFORMANCE MEASURE: GOOD NEWS

requests Δ-reasonable

$:\iff$

requests presented in time δ

can be served in time at most δ whenever $\delta \geq \Delta$.

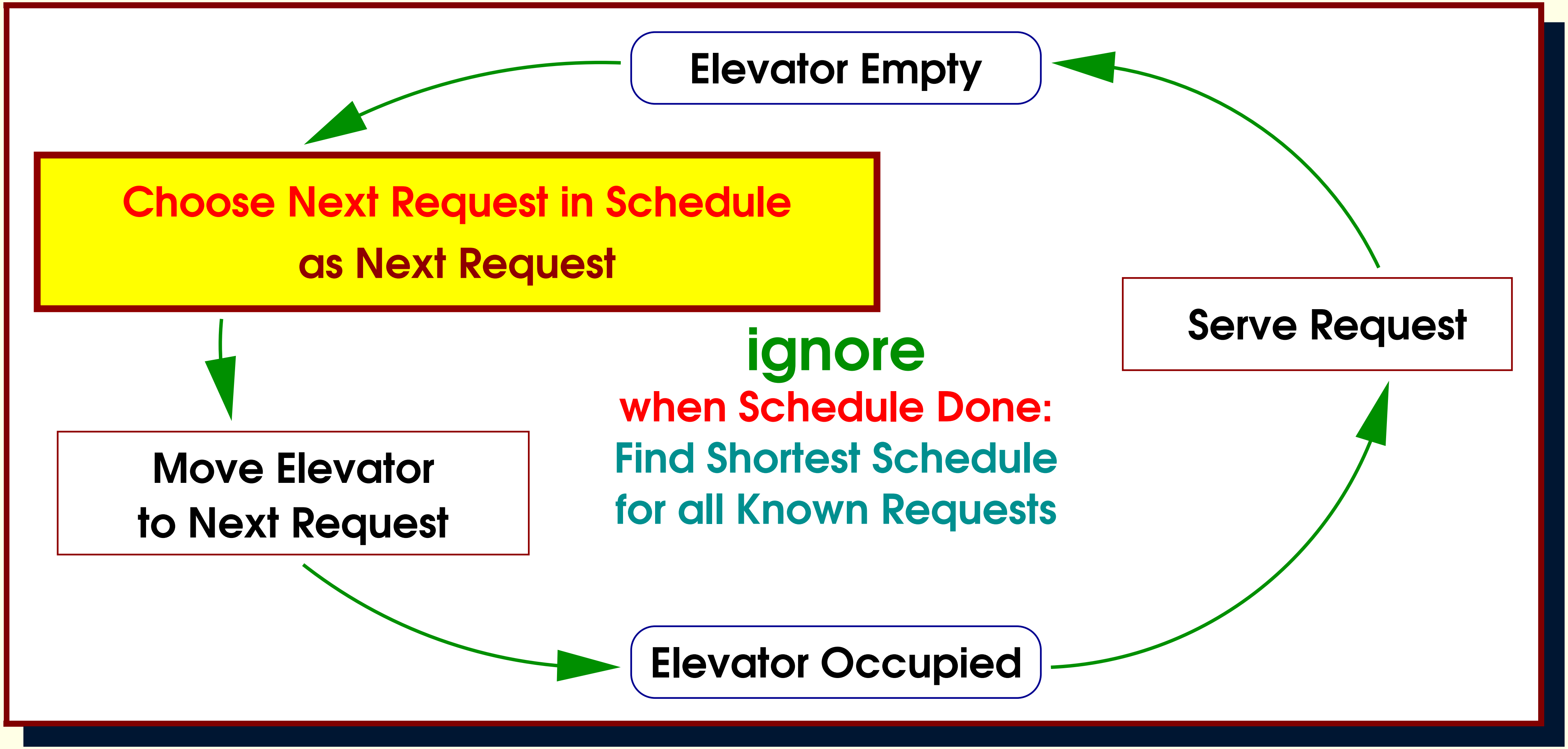
Theorem [Hauptmeier, Krumke, R. 2000]:

Under Δ-reasonable load, the

max./avg. flow time of IGNORE is at most 2Δ;

for REPLAN it is unbounded.

SINGLE ELEVATOR CONTROL: IGNORE



Alternative Performance Measure: Bad News

Alternative Performance Measure: Bad News

Simulation Experiments:

For the more complicated objectives from practice,

IGNORE does not produce good objective function values on average.

ALTERNATIVE PERFORMANCE MEASURE: BAD NEWS

Simulation Experiments:
For the more complicated objectives from practice,
IGNORE does not produce good objective function values on average.

In practice: Reoptimization w.r.t. a tweaked objective function
E.g.: Adding weighted quadratic waiting time penalties works well
However: No theoretical guarantees.

FC-Reoptimization (Flow-Time Constrained)

Assumption: achievable worst-case maximal flow time $\Theta > 0$ known.

FC-Reoptimization (Flow-Time Constrained)

Assumption: achievable worst-case maximal flow time $\Theta > 0$ known.

Original Reoptimization ILP:

$\mathcal{T}$ [$\mathcal{T}_u$]: set of feasible tours [for server u] (according to original model)

$$\min \sum_{T \in \mathcal{T}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \mathcal{T}} a_{vT} x_T = 1 \qquad \forall \text{requests } v \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \mathcal{T}_u} x_T = 1 \qquad \forall \text{servers } u \qquad \text{(Partitioning Servers)}$$

$$x_T \in \{0, 1\} \qquad \forall T \in \mathcal{T} \qquad \text{(Binary Variables)}$$

FC-REOPTIMIZATION (FLOW-TIME CONSTRAINED)

Assumption: achievable worst-case maximal flow time $\Theta > 0$ known.

FC-Reoptimization ILP:

$\tilde{\mathcal{T}}$ $[\tilde{\mathcal{T}}_u]$: set of feasible tours [for server u] with all flow times $\leq \Theta$

$$\min \sum_{T \in \tilde{\mathcal{T}}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \tilde{\mathcal{T}}} a_{\nu T} x_T = 1 \qquad \forall \text{requests } \nu \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \tilde{\mathcal{T}}_u} x_T = 1 \qquad \forall \text{servers } u \qquad \text{(Partitioning Servers)}$$

$$x_T \in \{0, 1\} \qquad \forall T \in \tilde{\mathcal{T}} \qquad \text{(Binary Variables)}$$

FC-Reoptimization (Flow-Time Constrained)

Assumption: achievable worst-case maximal flow time $\Theta > 0$ known.

FC-Reoptimization ILP:

$\tilde{\mathcal{T}}$ [$\tilde{\mathcal{T}}_u$]: set of feasible tours [for server u] with all flow times $\leq \Theta$

$$\min \sum_{T \in \tilde{\mathcal{T}}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \tilde{\mathcal{T}}} a_{vT} x_T = 1 \qquad \forall \text{requests } v \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \tilde{\mathcal{T}}_u} x_T = 1 \qquad \forall \text{servers } u \qquad \text{(Partitioning Servers)}$$

$$x_T \in \{0, 1\} \qquad \forall T \in \tilde{\mathcal{T}} \qquad \text{(Binary Variables)}$$

Is there a feasible solution at all times?

Doubling for FC-Reoptimization

- If at any time there is no feasible solution: $\Theta \leftarrow 2\Theta$.
- Always feasible (whenever the original model is)

Doubling for FC-Reoptimization

- If at any time there is no feasible solution: $\Theta \leftarrow 2\Theta$.
- Always feasible (whenever the original model is)

Open Questions:

Is there a guarantee for the maximal flow time obtained?

Is there a guarantee for the original objective function value?

FMC-Reoptimization (Flow-Time and Makespan Constrained)

Stronger assumption: the request set is Δ-reasonable.

FMC-REOPTIMIZATION (FLOW-TIME AND MAKESPAN CONSTRAINED)

Stronger assumption: the request set is Δ-reasonable.

FMC-Reoptimization ILP:

$\hat{\mathcal{T}}$ [$\hat{\mathcal{T}}_u$]: set of feasible tours [for server u] with all flow times $\leq 2\Delta$ and makespan $\leq \Delta$

$$\min \sum_{T \in \hat{\mathcal{T}}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \hat{\mathcal{T}}} a_{vT} x_T = 1 \qquad \forall \text{requests } v \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \hat{\mathcal{T}}_u} x_T = 1 \qquad \forall \text{servers } u \qquad \text{(Partitioning Servers)}$$

$$x_T \in \{0, 1\} \qquad \forall T \in \hat{\mathcal{T}} \qquad \text{(Binary Variables)}$$

FMC-REOPTIMIZATION (FLOW-TIME AND MAKESPAN CONSTRAINED)

Stronger assumption: the request set is Δ-reasonable.

FMC-Reoptimization ILP:

$\hat{\mathcal{T}}$ $[\hat{\mathcal{T}}_u]$: set of feasible tours [for server u] with all flow times $\leq 2\Delta$ and makespan $\leq \Delta$

$$\min \sum_{T \in \hat{\mathcal{T}}} c_T x_T \quad \text{s.t.}$$

$$\sum_{T \in \hat{\mathcal{T}}} a_{vT} x_T = 1 \qquad \forall \text{requests } v \qquad \text{(Partitioning Requests)}$$

$$\sum_{T \in \hat{\mathcal{T}}_u} x_T = 1 \qquad \forall \text{servers } u \qquad \text{(Partitioning Servers)}$$

$$x_T \in \{0, 1\} \qquad \forall T \in \hat{\mathcal{T}} \qquad \text{(Binary Variables)}$$

Is there a feasible solution at all times?

Reoptimization Admission Control (RAC)

Whenever there is no feasible new solution, . . .

REOPTIMIZATION ADMISSION CONTROL (RAC)

Whenever there is no feasible new solution, . . .

- . . . continue with old dispatch

Reoptimization Admission Control (RAC)

Whenever there is no feasible new solution, . . .

- . . . continue with old dispatch
- . . . buffer the new requests

Reoptimization Admission Control (RAC)

Whenever there is no feasible new solution, …

- … continue with old dispatch
- … buffer the new requests
- … when the current dispatch is finished, compute new dispatch (now feasible!)

REOPTIMIZATION ADMISSION CONTROL (RAC)

Whenever there is no feasible new solution, …

- … continue with old dispatch
- … buffer the new requests
- … when the current dispatch is finished, compute new dispatch (now feasible!)

Theorem:
Under Δ-reasonable load, FMC-Reoptimization with RAC achieves a maximal flow time of 2Δ, no matter what the original reoptimization problem is (and this is best possible).

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

- Unconstrained reoptimization is best w.r.t. original objective.

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

- Unconstrained reoptimization is best w.r.t. original objective.
- Flow time constrained reoptimization is next.

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

- Unconstrained reoptimization is best w.r.t. original objective.
- Flow time constrained reoptimization is next.
- Only slightly worse: FMC-Reoptimization with RAC

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

- Unconstrained reoptimization is best w.r.t. original objective.
- Flow time constrained reoptimization is next.
- Only slightly worse: FMC-Reoptimization with RAC
- Everything else we tested: much worse.

How about the Expected Original Objective Function Value?

Simulation Experiments for Elevator Group Control:

- Unconstrained reoptimization is best w.r.t. original objective.
- Flow time constrained reoptimization is next.
- Only slightly worse: FMC-Reoptimization with RAC
- Everything else we tested: much worse.

Work in Progress:
Theoretical guarantees or computational bounds for original objective
(at least for special cases).

When Δ is unknown

- Δ can be estimated by makespan computations (under-estimation)
  $\longrightarrow$ better flow time guarantee
- Δ can be estimated by doubling (over-estimation)
  $\longrightarrow$ better original objective on average

Summary and Remarks

Summary and Remarks

Summary:

Reoptimization → best observed long-term objective on average

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

SUMMARY AND REMARKS

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

- Dynamic column generation models $\longrightarrow$ FMC easy to implement

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

- Dynamic column generation models $\longrightarrow$ FMC easy to implement
- FMC can be relaxed with $\alpha > 1$:
  flow time $\leq 2\alpha\Delta$ & makespan $\leq \alpha\Delta \Rightarrow$ maximal flow time $\leq 2\alpha\Delta$

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

- Dynamic column generation models $\longrightarrow$ FMC easy to implement
- FMC can be relaxed with $\alpha > 1$:
  flow time $\leq 2\alpha\Delta$ & makespan $\leq \alpha\Delta \Rightarrow$ maximal flow time $\leq 2\alpha\Delta$
- $\Delta \longrightarrow$ capacity planning

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

- Dynamic column generation models $\longrightarrow$ FMC easy to implement
- FMC can be relaxed with $\alpha > 1$:
  flow time $\leq 2\alpha\Delta$ & makespan $\leq \alpha\Delta \Rightarrow$ maximal flow time $\leq 2\alpha\Delta$
- $\Delta \longrightarrow$ capacity planning
- $\Delta \longrightarrow$ system admission control

Summary and Remarks

Summary:

Reoptimization $\longrightarrow$ best observed long-term objective on average

Infinite deferment $\longrightarrow$ bounded flow time required

No guarantees for FC-Reoptimization $\longrightarrow$ FMC-Reoptimization with RAC

- Dynamic column generation models $\longrightarrow$ FMC easy to implement
- FMC can be relaxed with $\alpha > 1$:
  flow time $\leq 2\alpha\Delta$ & makespan $\leq \alpha\Delta \Rightarrow$ maximal flow time $\leq 2\alpha\Delta$
- $\Delta \longrightarrow$ capacity planning
- $\Delta \longrightarrow$ system admission control
- New sparse LP methods for MDP $\longrightarrow$ computational guarantees

Thank you!