

Optimal Rescheduling in Automotive Industry

Markus Kruber Marco Lübbecke Jonas Witt

RWTH Aachen University

[#or2015vienna](#) · Vienna · 09/02/2015

Just-In-Time Manufacturing

Just-In-Time Manufacturing



No Warehousing

Motivation

Just-In-Time Manufacturing



No Warehousing

Long Delivery Routes

Motivation

Just-In-Time Manufacturing



No Warehousing

Long Delivery Routes



Forecast / Scheduling

Motivation

Just-In-Time Manufacturing



No Warehousing

Long Delivery Routes



Forecast / Scheduling

Mass Customization

Motivation

Just-In-Time Manufacturing



No Warehousing

Long Delivery Routes



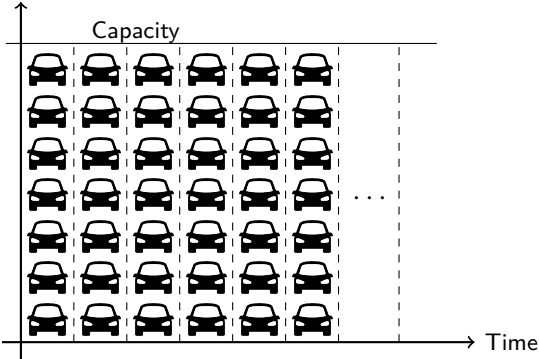
Forecast / Scheduling

Mass Customization

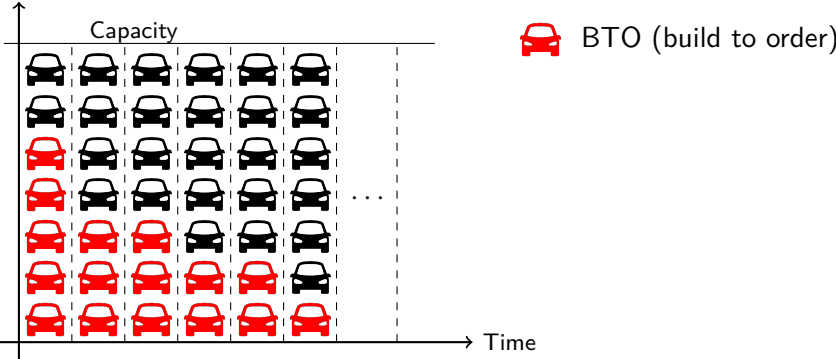


Imperfect Forecast

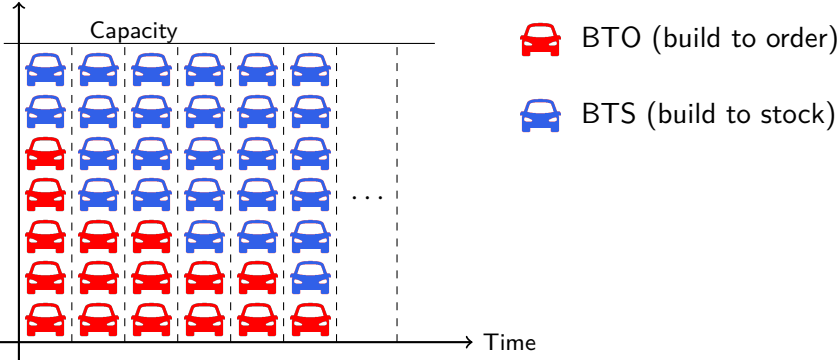
Main Idea - Orderbank



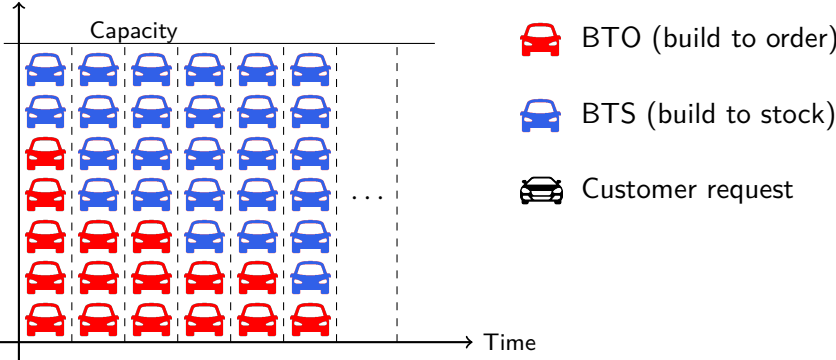
Main Idea - Orderbank



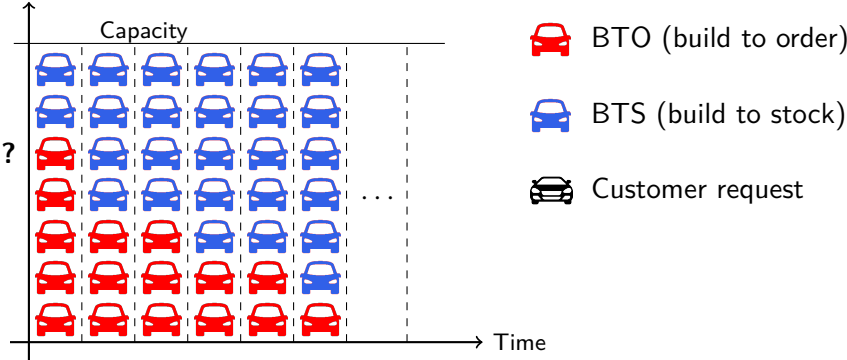
Main Idea - Orderbank



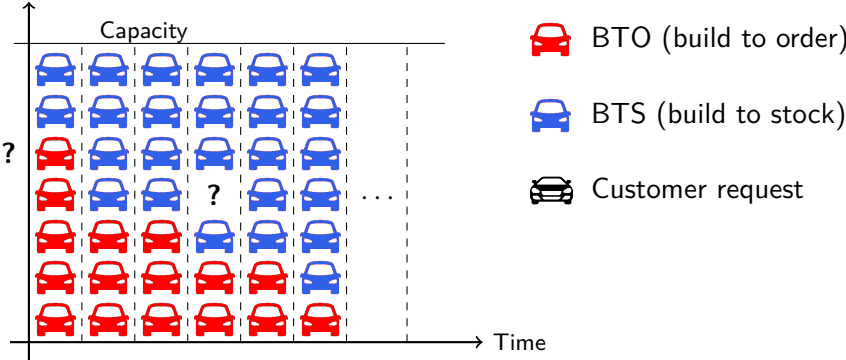
Main Idea - Orderbank



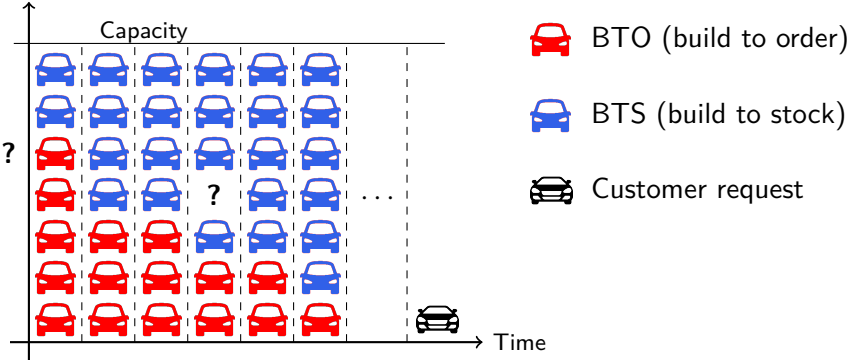
Main Idea - Orderbank



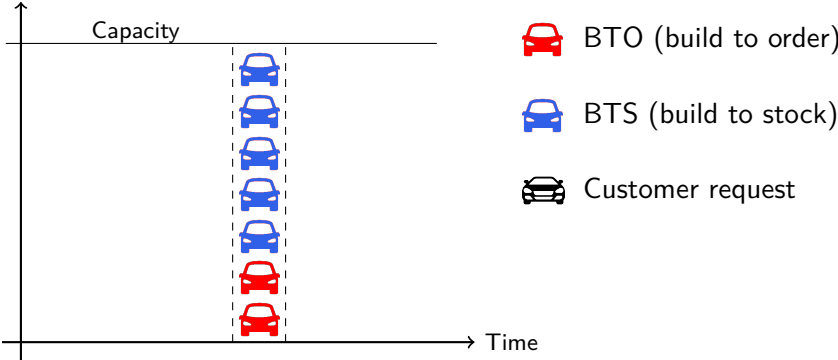
Main Idea - Orderbank



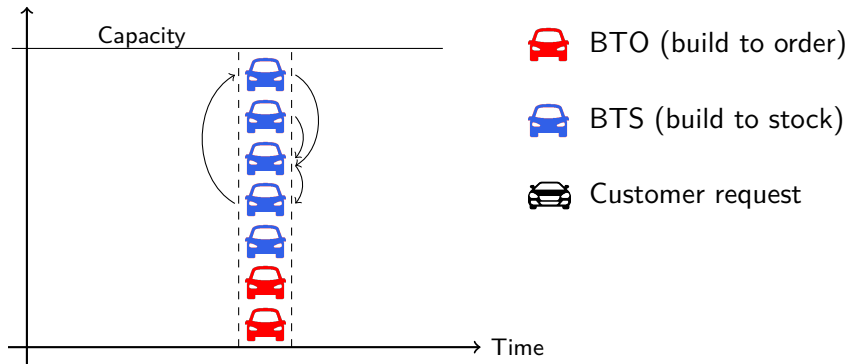
Main Idea - Orderbank



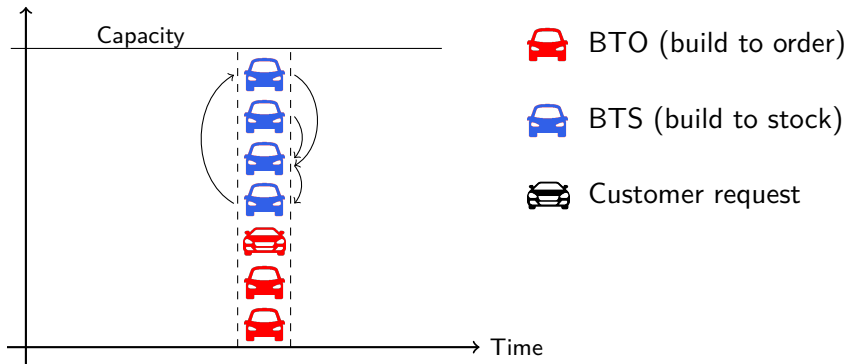
Main Idea - Orderbank



Main Idea - Orderbank

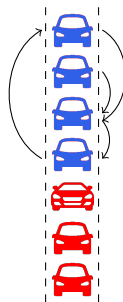


Main Idea - Orderbank



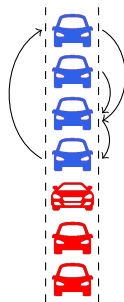
Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only



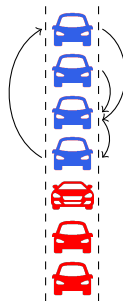
Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only
- ▶ No warehousing
⇒ consider specific day



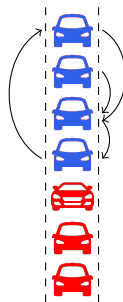
Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only
- ▶ No warehousing
⇒ consider specific day
- ▶ Satisfy customer request



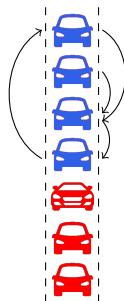
Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only
- ▶ No warehousing
⇒ consider specific day
- ▶ Satisfy customer request
- ▶ Minimize overall modification



Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only
- ▶ No warehousing
⇒ consider specific day
- ▶ Satisfy customer request
- ▶ Minimize overall modification
- ▶ Modified vehicle
 - ▶ Valid
 - ▶ Similar
 - ▶ Saleable



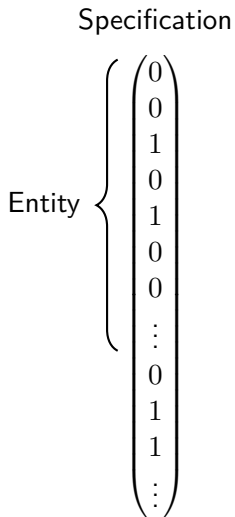
How to describe a car?



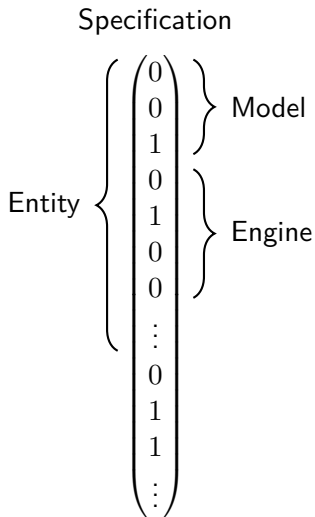
Specification

$$\begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 1 \\ \vdots \end{pmatrix}$$

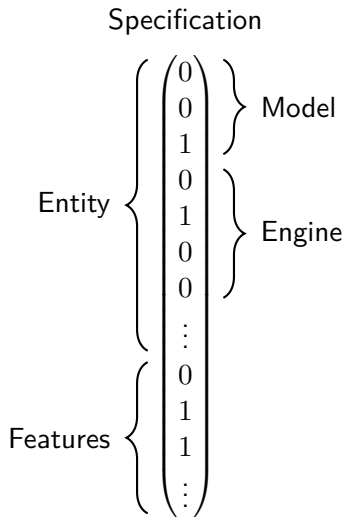
How to describe a car?



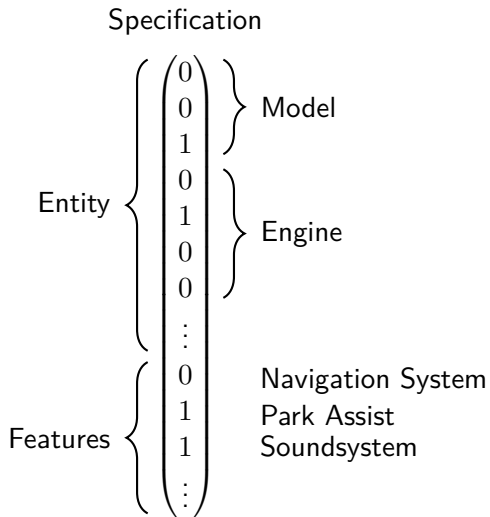
How to describe a car?



How to describe a car?



How to describe a car?



V_{BTS}



Set of nodes V_{BTS}

Each node represents a BTS-Car, which is planned to be build at that day.

Bipartite Graph

V_{BTS}



V



Set of nodes V_{BTS}

Each node represents a BTS-Car, which is planned to be build at that day.

Set of nodes V

The set V contains a node for every valid specification.

V_{BTS}



V



$k = \#Attribute$

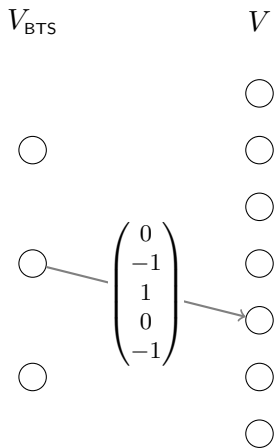
Function **spec** for nodes

Function

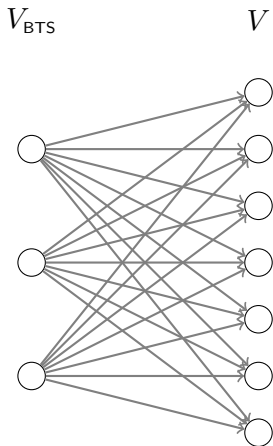
$$\text{spec} : V_{\text{BTS}} \dot{\cup} V \rightarrow \{0, 1\}^k$$

returns for each node the corresponding specification vector.

Bipartite Graph



Bipartite Graph



Function **spec** for edges

The function

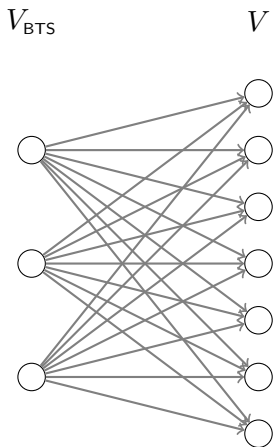
$$\text{spec} : E \rightarrow \{-1, 0, 1\}^k$$

is defined as:

$$\text{spec}(e) = \text{spec}(v_2) - \text{spec}(v_1),$$

for all $(v_1, v_2) = e \in E$.

Bipartite Graph



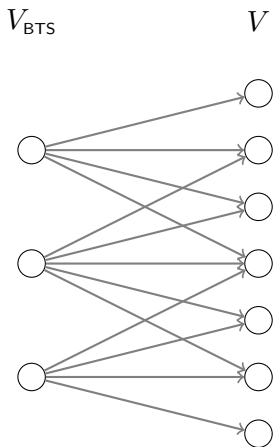
Similarity constraint

For given weights

$w = (w_1, \dots, w_k) \in \mathbb{R}_{\geq 0}^k$ and a
upper bound $\text{maxDiff} \in \mathbb{R}_{\geq 0}$, the set
of edges is defined as:

$$E = \{(v_1, v_2) = e \in V_{BTS} \times V \mid \sum_{i=1}^k w_i \cdot |\text{spec}(e)_i| \leq \text{maxDiff}\}.$$

Bipartite Graph



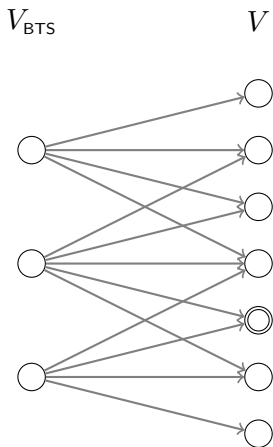
Similarity constraint

For given weights

$w = (w_1, \dots, w_k) \in \mathbb{R}_{\geq 0}^k$ and a
upper bound $\text{maxDiff} \in \mathbb{R}_{\geq 0}$, the set
of edges is defined as:

$$E = \{(v_1, v_2) = e \in V_{BTS} \times V \mid \sum_{i=1}^k w_i \cdot |\text{spec}(e)_i| \leq \text{maxDiff}\}.$$

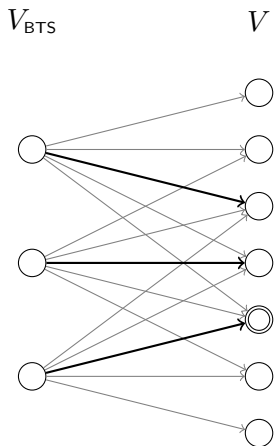
Bipartite Graph



Definition

Let node $v^* \in V$ represent the node corresponding to **customers request**.

Valid Exchange



$$\& \begin{pmatrix} -1 \\ 0 \\ 1 \\ -1 \\ \vdots \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \end{pmatrix} + \dots = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \end{pmatrix}$$

Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only ✓
- ▶ No warehousing
⇒ consider specific day ✓
- ▶ Satisfy customer request ✓
- ▶ Minimize overall modification ✗
- ▶ Modified vehicle
 - ▶ Valid ✓
 - ▶ Similar ✓
 - ▶ Saleable ✗

Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only ✓
- ▶ No warehousing
⇒ consider specific day ✓
- ▶ Satisfy customer request ✓
- ▶ **Minimize overall modification** ✗
- ▶ Modified vehicle
 - ▶ Valid ✓
 - ▶ Similar ✓
 - ▶ Saleable ✗

$$\begin{aligned} \text{s.t.} \quad & \sum_{(v_1, v_2) = e \in E} x_e = 1 && \forall v_1 \in V_{\text{BTS}} \\ & \sum_{e \in E} \text{spec}(e)_i \cdot x_e = 0 && \forall i \in \{1, \dots, k\} \\ & \sum_{(v_1, v^*) = e \in E} x_e \geq 1 \\ & x_e \in \{0, 1\} && \forall e \in E \end{aligned}$$

$$\min \sum_{\substack{e \in E: \\ \text{spec}(e) \neq 0}} x_e$$

$$\text{s.t.} \quad \sum_{(v_1, v_2) = e \in E} x_e = 1$$

$$\forall v_1 \in V_{\text{BTS}}$$

$$\sum_{e \in E} \text{spec}(e)_i \cdot x_e = 0$$

$$\forall i \in \{1, \dots, k\}$$

$$\sum_{(v_1, v^*) = e \in E} x_e \geq 1$$

$$x_e \in \{0, 1\}$$

$$\forall e \in E$$

$$\min \sum_{\substack{e \in E: \\ \text{spec}(e) \neq 0}} x_e + \sum_{e \in E} \|\text{spec}(e)\|_1 \cdot x_e$$

$$\text{s.t.} \quad \sum_{(v_1, v_2) = e \in E} x_e = 1 \quad \forall v_1 \in V_{\text{BTS}}$$

$$\sum_{e \in E} \text{spec}(e)_i \cdot x_e = 0 \quad \forall i \in \{1, \dots, k\}$$

$$\sum_{(v_1, v^*) = e \in E} x_e \geq 1$$

$$x_e \in \{0, 1\} \quad \forall e \in E$$

Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only ✓
- ▶ No warehousing
⇒ consider specific day ✓
- ▶ Satisfy customer request ✓
- ▶ Minimize overall modification ✓
- ▶ Modified vehicle
 - ▶ Valid ✓
 - ▶ Similar ✓
 - ▶ Saleable ✗

Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only ✓
- ▶ No warehousing
⇒ consider specific day ✓
- ▶ Satisfy customer request ✓
- ▶ Minimize overall modification ✓
- ▶ Modified vehicle
 - ▶ Valid ✓
 - ▶ Similar ✓
 - ▶ **Saleable** ✗

- ▶ Invented to find relationships between products in supermarkets,
e.g. {Onions, Potatos} \Rightarrow {Burger}

- ▶ Invented to find relationships between products in supermarkets,
e.g. {Onions, Potatos} \Rightarrow {Burger}
- ▶ Find rules in historical data of the orderbank,
e.g. {Navigation System} \Rightarrow {Voice Control}

Association Rule Learning

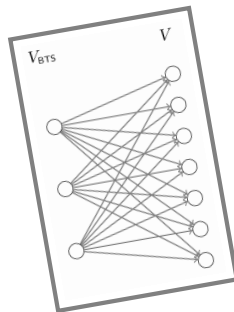
- ▶ Invented to find relationships between products in supermarkets,
e.g. {Onions, Potatos} \Rightarrow {Burger}
- ▶ Find rules in historical data of the orderbank,
e.g. {Navigation System} \Rightarrow {Voice Control}
- ▶ Adjust the graph



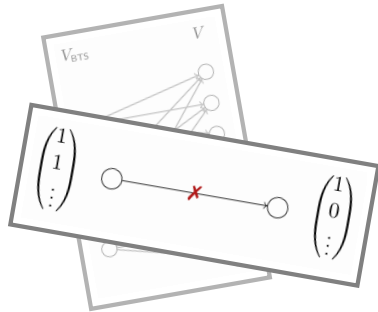
Requirements on an Exchange

- ▶ Rebuild BTS-Vehicles only ✓
- ▶ No warehousing
⇒ consider specific day ✓
- ▶ Satisfy customer request ✓
- ▶ Minimize overall modification ✓
- ▶ Modified vehicle
 - ▶ Valid ✓
 - ▶ Similar ✓
 - ▶ Saleable ✓

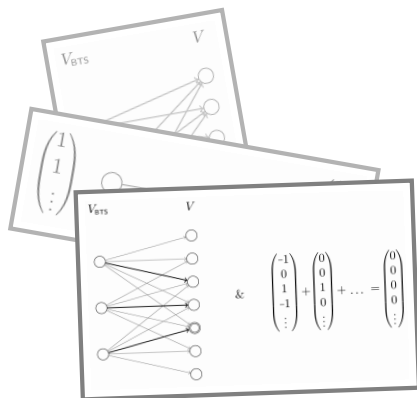
1) Graph construction



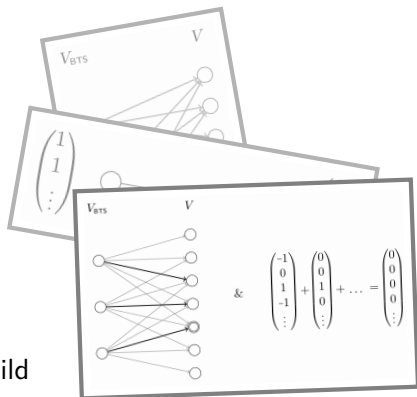
- I) Graph construction
- II) Association analysis, ...



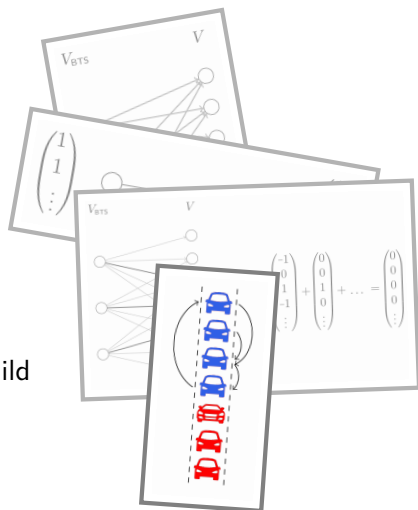
- I) Graph construction
 - II) Association analysis, ...
- 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$



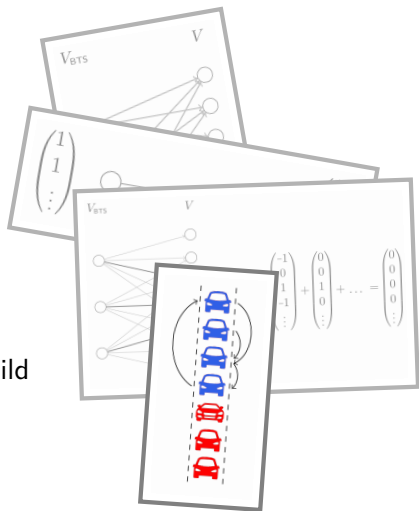
- I) Graph construction
- II) Association analysis, ...
 - 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$
 - 2) Choose earliest / best rebuild



- I) Graph construction
- II) Association analysis, ...
 - 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$
 - 2) Choose earliest / best rebuild
 - 3) Reschedule

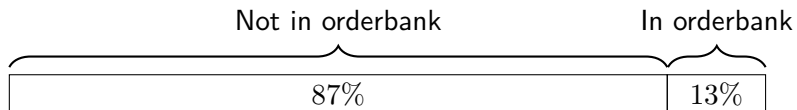


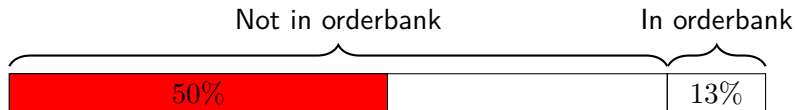
- I) Graph construction
- II) Association analysis, ...
 - 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$
 - 2) Choose earliest / best rebuild
 - 3) Reschedule
 - 4) Adjust graph



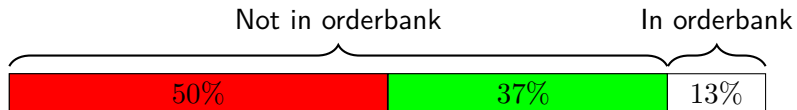
- I) Graph construction
- II) Association analysis, ...
 - 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$
 - 2) Choose earliest / best rebuild
 - 3) Reschedule
 - 4) Adjust graph

- I) Graph construction
- II) Association analysis, ...
- 0) for each valid specification $=: v^*$
 - 1) for day in searching period
solve IP $\sim V_{BTS}, v^*$
 - 2) Choose earliest / best rebuild
 - 3) Reschedule
 - 4) Adjust graph

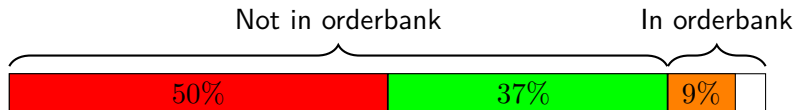




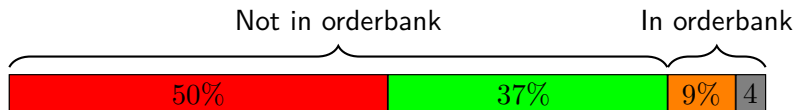
■ No rebuild



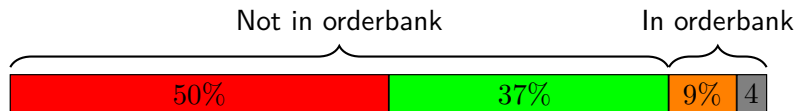
- No rebuild
- Only rebuild (Improvement!)



- No rebuild
- Only rebuild (Improvement!)
- Earlier rebuild



- No rebuild
- Only rebuild (Improvement!)
- Earlier rebuild
- No improvement



- No rebuild
- Only rebuild (Improvement!)
- Earlier rebuild
- No improvement

Thank you for your attention!