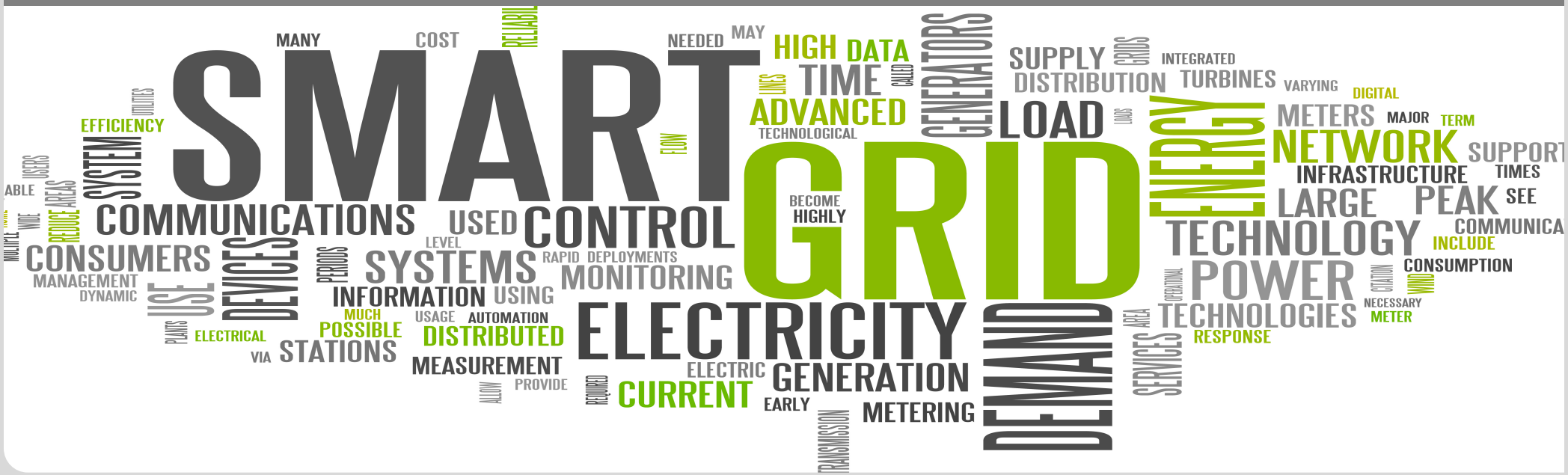
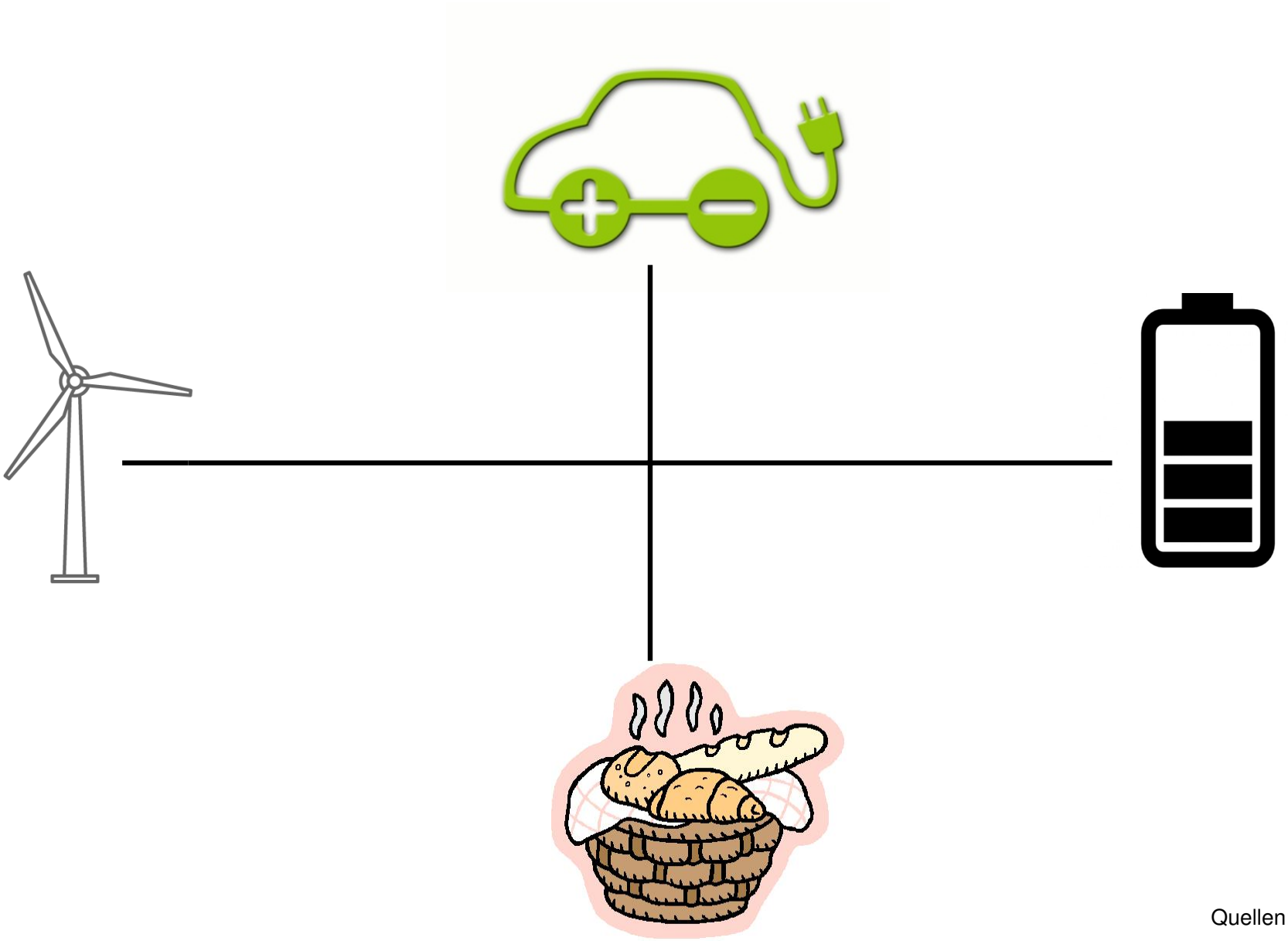
Nachfrageflexibilität in Smart Grids und ihre Nutzen - Modellierung und heuristische Optimierung

Seminar Energieinformatik · December 5, 2017
Lena Winter

INSTITUTE OF THEORETICAL INFORMATICS · ALGORITHMICS GROUP



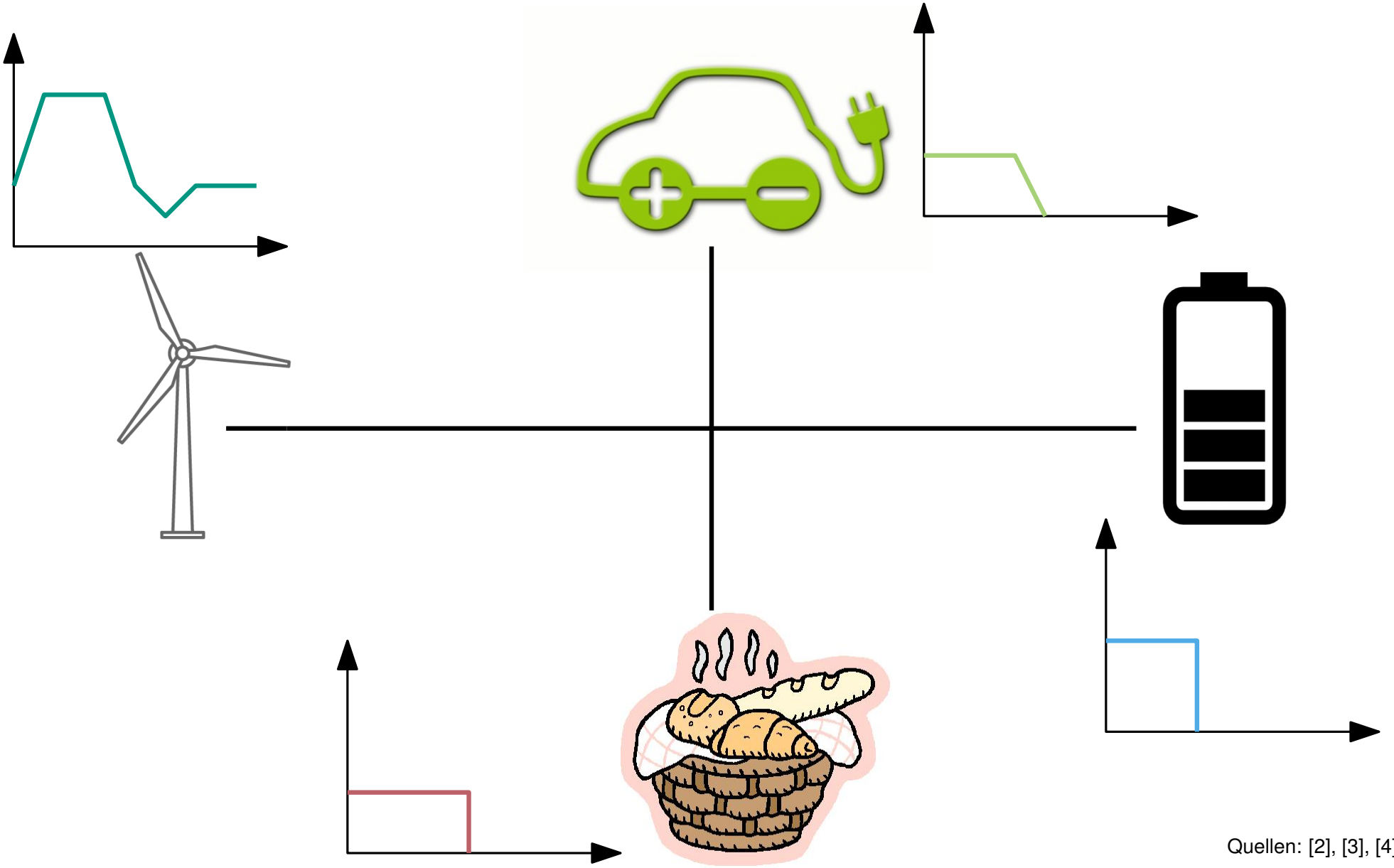
Motivation



Quellen: [2], [3], [4], [5]

Institute of Theoretical Informatics
Algorithmics Group

Motivation



Quellen: [2], [3], [4], [5]

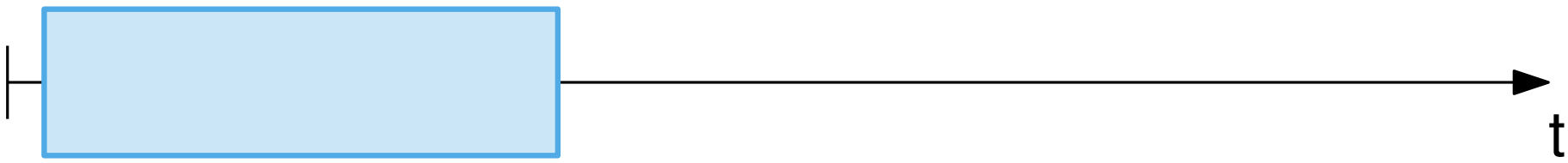
Institute of Theoretical Informatics
Algorithmics Group

Modellierung

Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibiliäten

Unterscheidungspunkte von Flexibilitätsklassen:

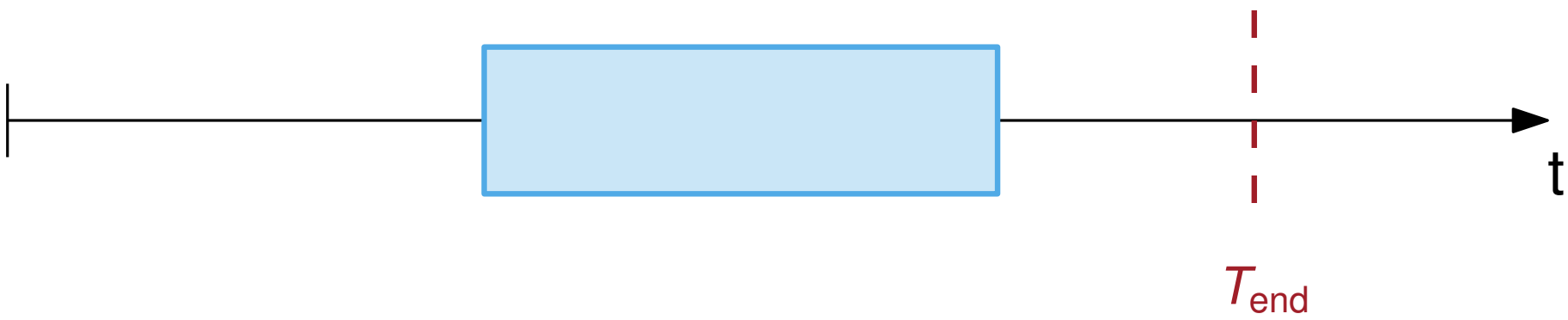


Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibiliäten

Unterscheidungspunkte von Flexibilitätsklassen:

- Gibt es einen festgelegten spätesten Endzeitpunkt?

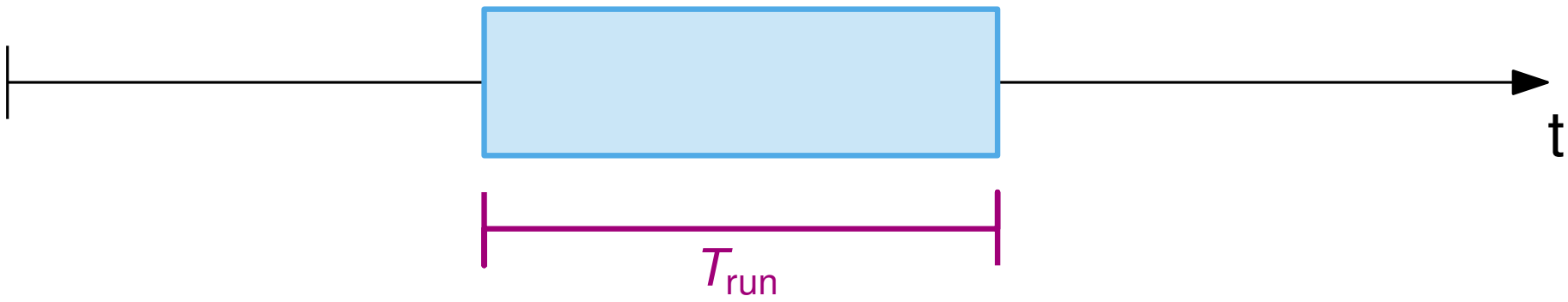


Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibiliäten

Unterscheidungspunkte von Flexibilitätsklassen:

- Gibt es einen festgelegten spätesten Endzeitpunkt?
- Gibt es eine festgelegte Betriebsdauer?

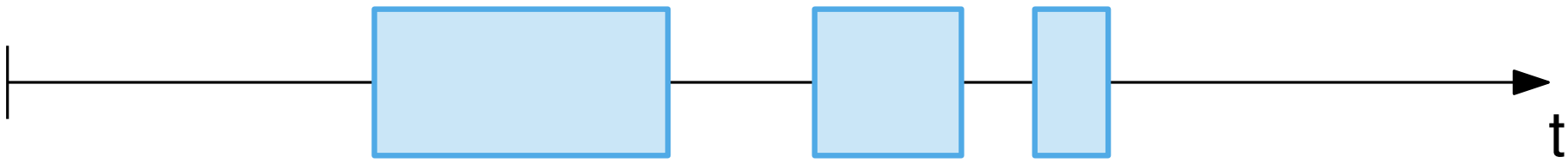


Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibiliäten

Unterscheidungspunkte von Flexibilitätsklassen:

- Gibt es einen festgelegten spätesten Endzeitpunkt?
- Gibt es eine festgelegte Betriebsdauer?
- Ist der Vorgang unterbrechbar?

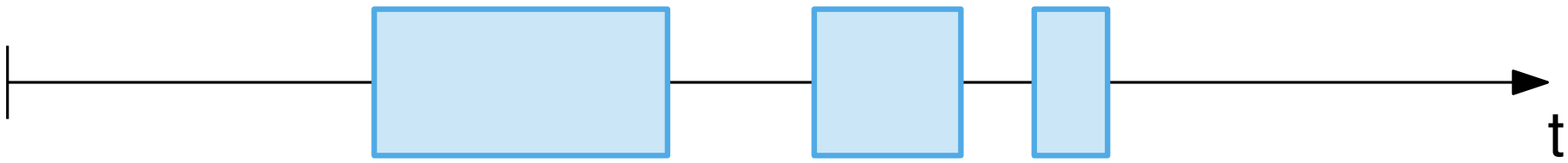


Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibiliäten

Unterscheidungspunkte von Flexibilitätsklassen:

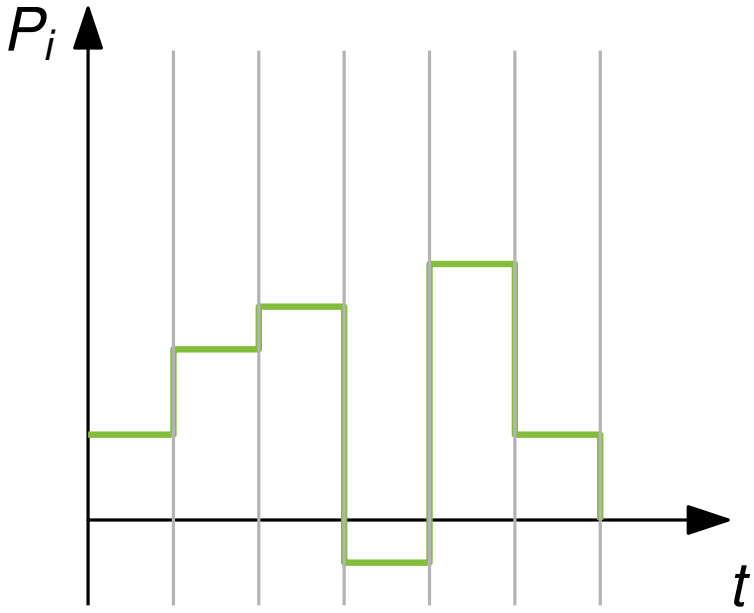
- Gibt es einen festgelegten spätesten Endzeitpunkt?
- Gibt es eine festgelegte Betriebsdauer?
- Ist der Vorgang unterbrechbar?
- Wie viel Energie wird benötigt?



Institute of Theoretical Informatics
Algorithmics Group

Modellierung:

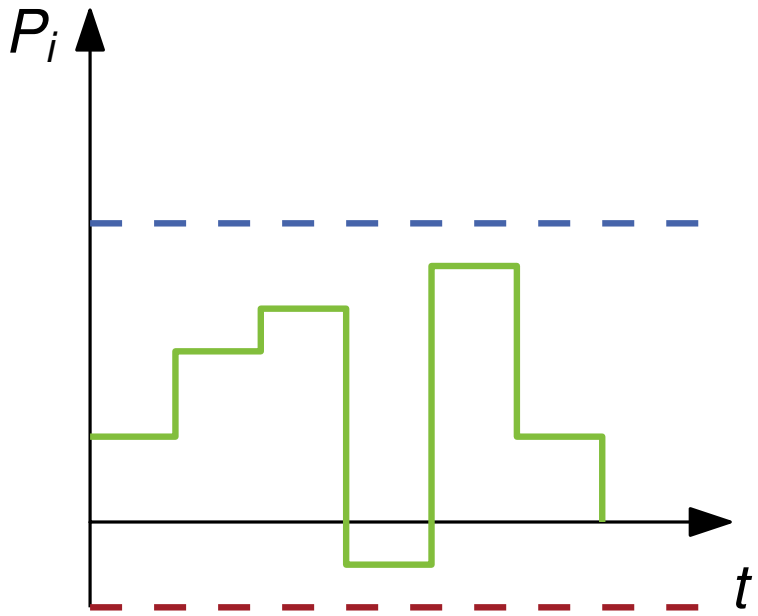
- Größe eines Zeitschrittes: T_s



Institute of Theoretical Informatics
Algorithmics Group

Modellierung:

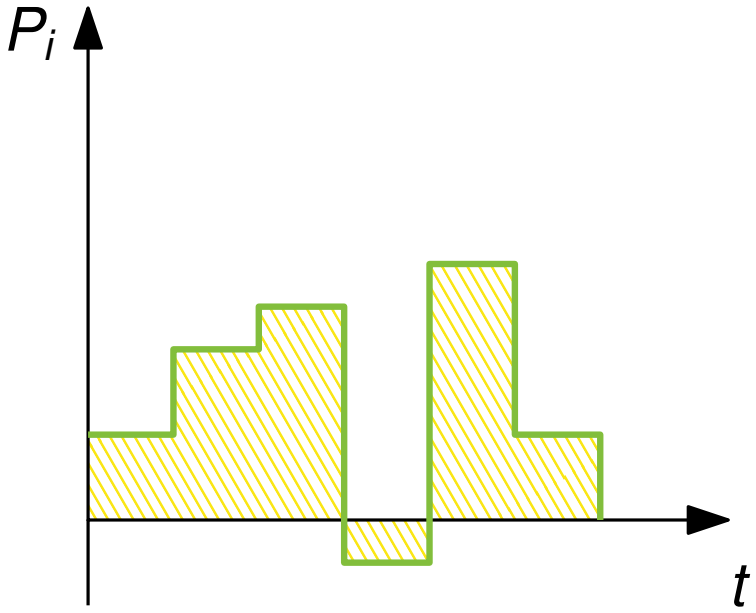
- Größe eines Zeitschrittes: T_s
- untere Grenze der Konsumrate: $\underline{P_i}$
- obere Grenze der Konsumrate: $\overline{P_i}$



Institute of Theoretical Informatics
Algorithmics Group

Modellierung:

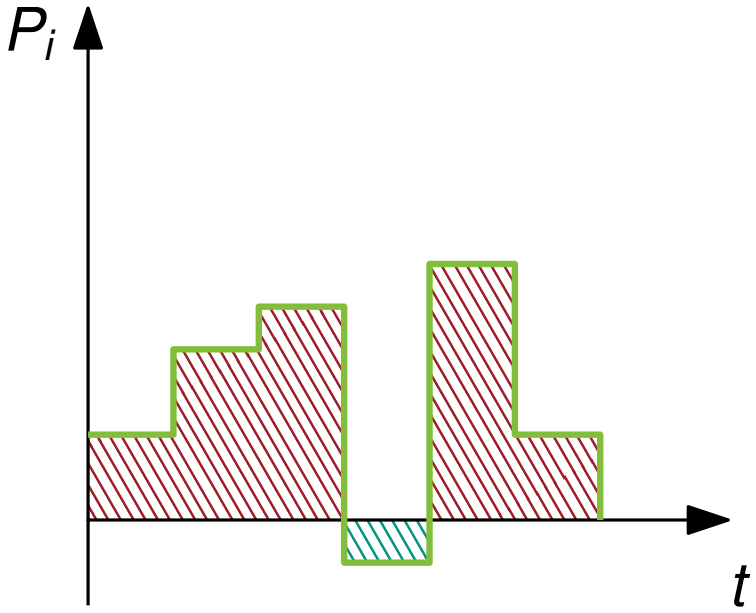
- Größe eines Zeitschrittes: T_s
- untere Grenze der Konsumrate: $\underline{P_i}$
- obere Grenze der Konsumrate: $\overline{P_i}$
- untere Grenze des Ladestandes: $\underline{E_i}$
- obere Grenze des Ladestandes: $\overline{E_i}$



Institute of Theoretical Informatics
Algorithmics Group

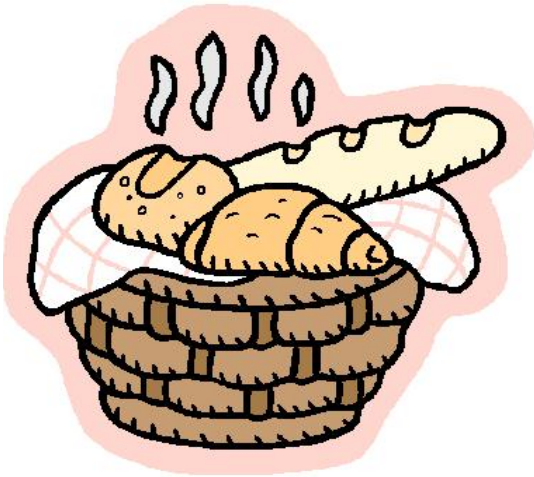
Modellierung:

- Größe eines Zeitschrittes: T_s
- untere Grenze der Konsumrate: $\underline{P_i}$
- obere Grenze der Konsumrate: $\overline{P_i}$
- untere Grenze des Ladestandes: $\underline{E_i}$
- obere Grenze des Ladestandes: $\overline{E_i}$



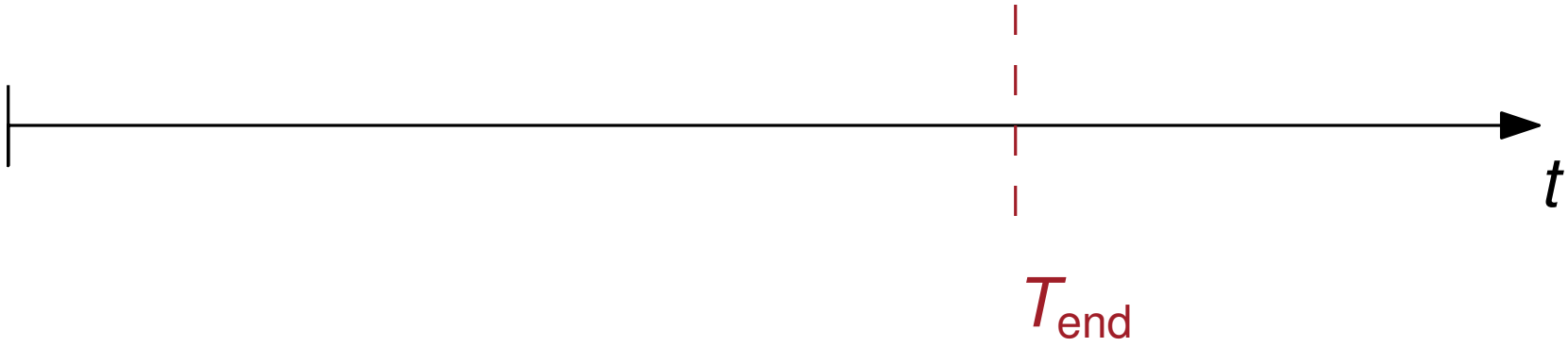
Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen



Eigenschaften:

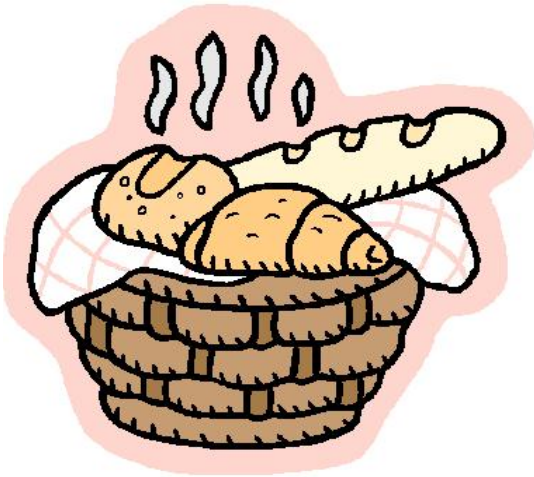
- Festgelegter Endzeitpunkt
- nicht unterbrechbar
- Festgelegte Konsumrate
- Festgelegte Betriebsdauer



Quellen: [2]

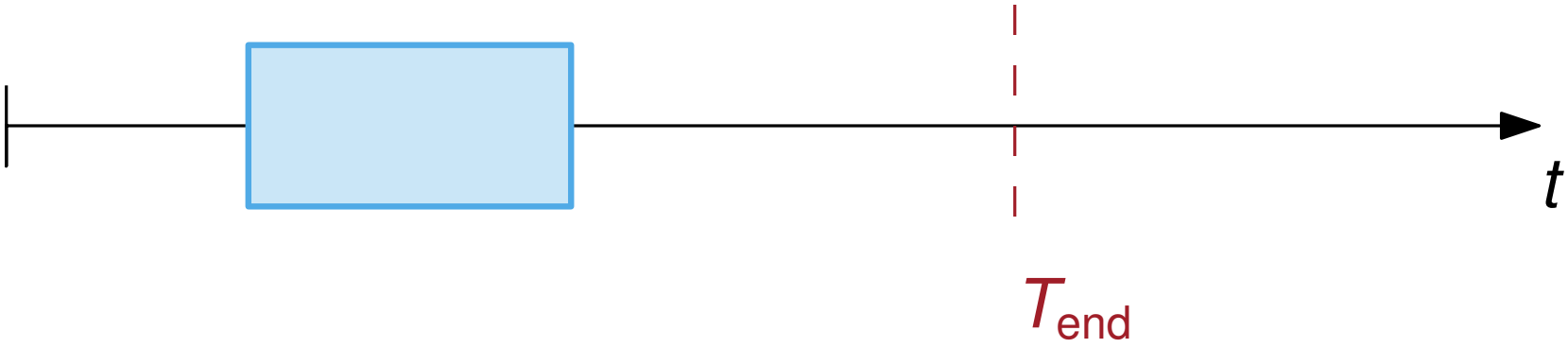
Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen



Eigenschaften:

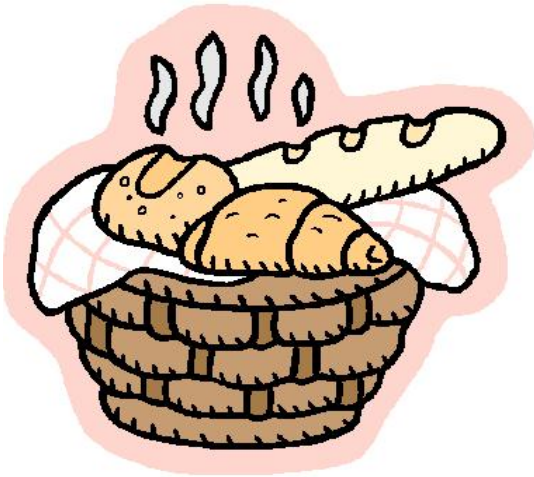
- Festgelegter Endzeitpunkt
- nicht unterbrechbar
- Festgelegte Konsumrate
- Festgelegte Betriebsdauer



Quellen: [2]

Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen



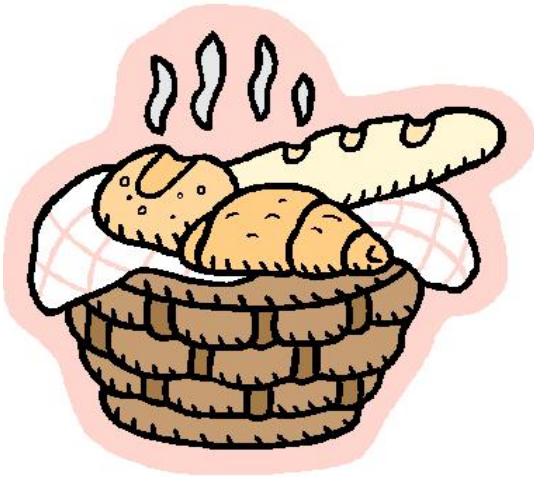
Eigenschaften:

- Festgelegter Endzeitpunkt
- nicht unterbrechbar
- Festgelegte Konsumrate
- Festgelegte Betriebsdauer

Quellen: [2]

Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen



Eigenschaften:

- Festgelegter Endzeitpunkt
- nicht unterbrechbar
- Festgelegte Konsumrate
- Festgelegte Betriebsdauer

Flexibilitätsklasse: Bakery

Beispiele: Gewächshäuser, Produktionsprozesse, ...

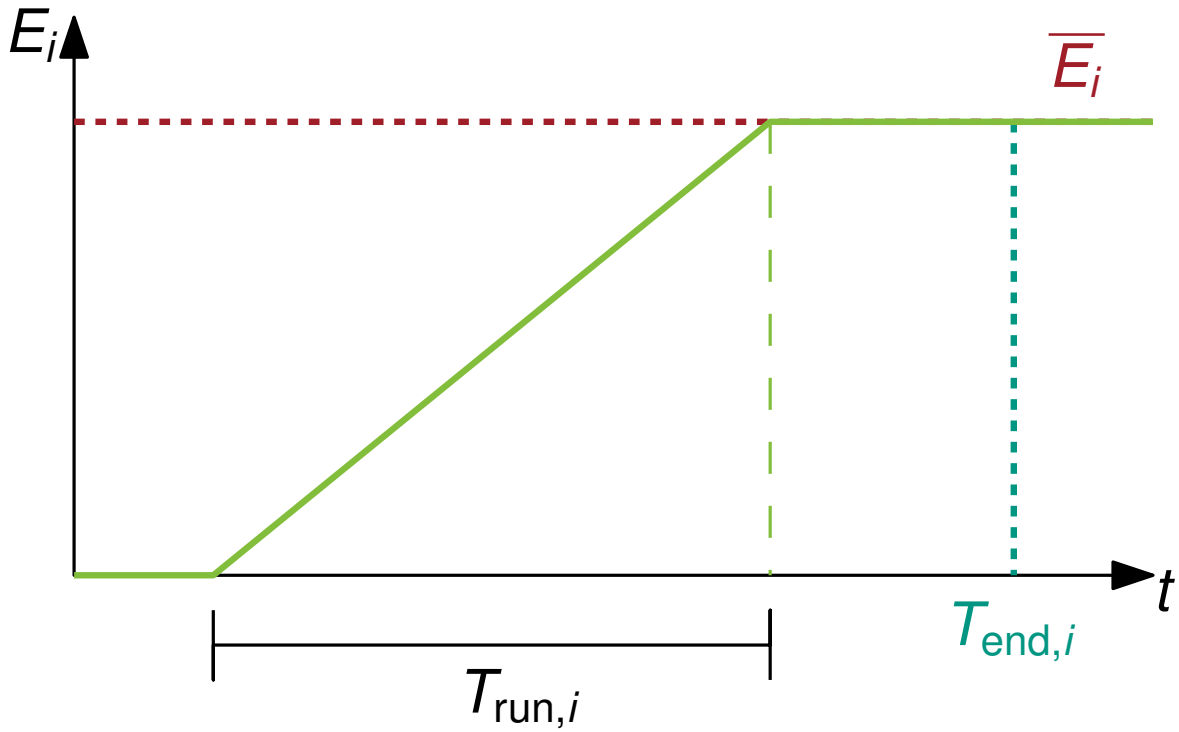
Quellen: [2]

Modellierung: Flexibilitätsklassen

Bakery$_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$



Institute of Theoretical Informatics
Algorithmics Group

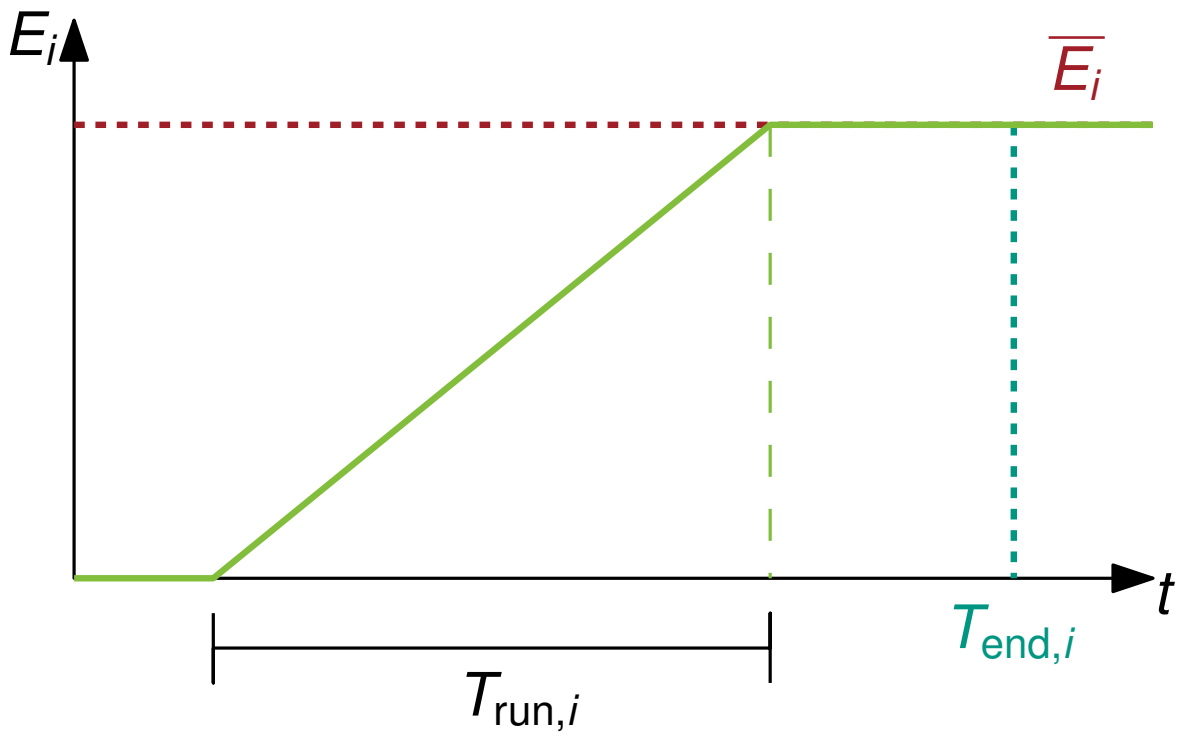
Modellierung: Flexibilitätsklassen

Bakery$_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$

$$P_i(k) = \overline{P_i} v_i(k)$$



Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen

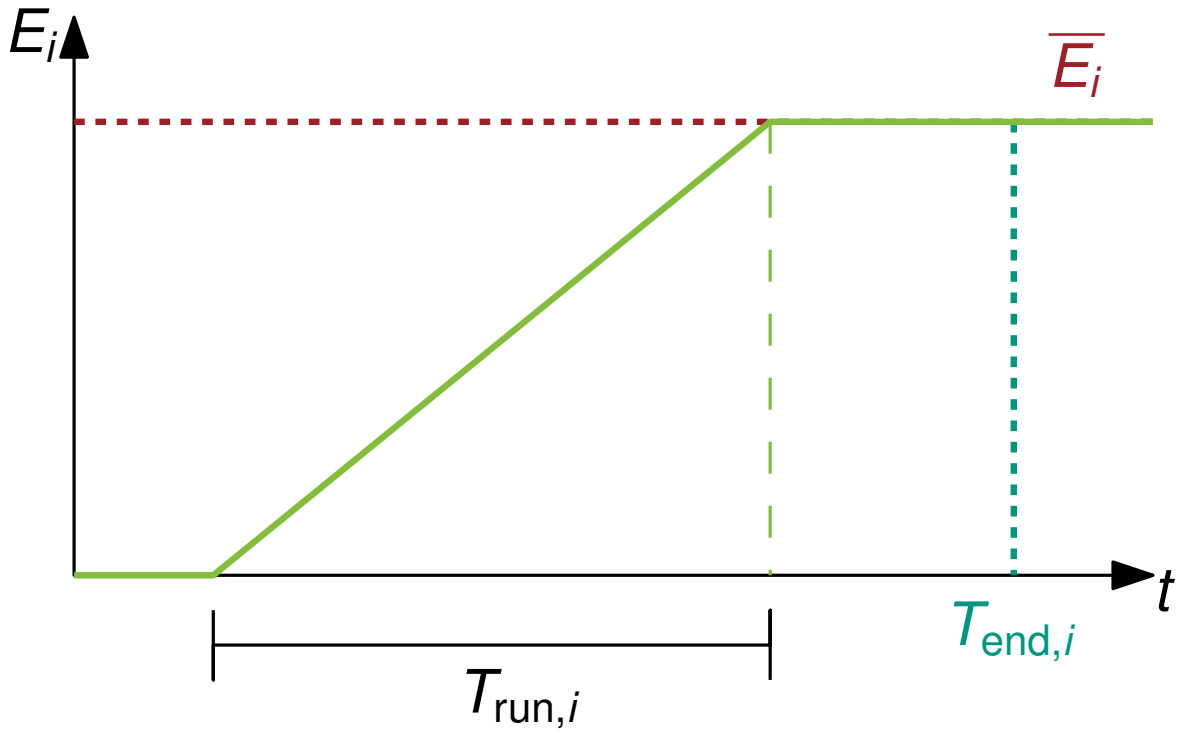
Bakery$_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$

$$P_i(k) = \overline{P_i} v_i(k)$$

$$E_i(k+1) = E_i(k) + T_s P_i(k)$$



Institute of Theoretical Informatics
Algorithmics Group

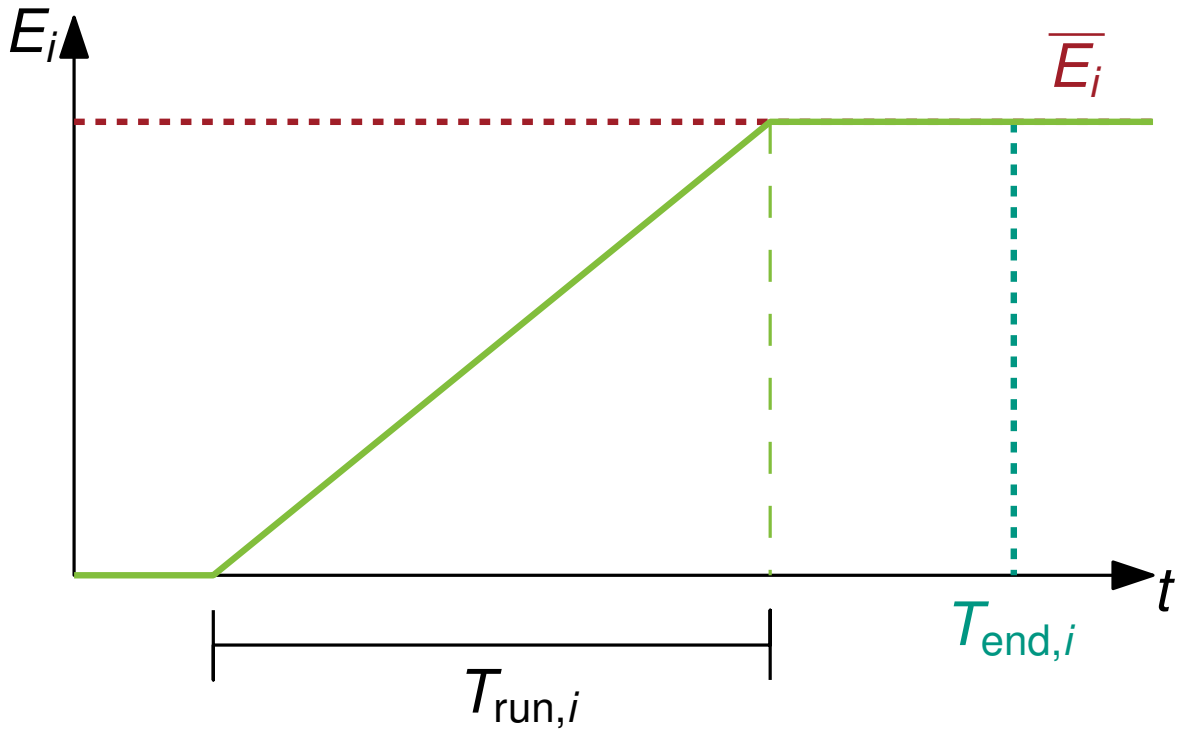
Modellierung: Flexibilitätsklassen

Bakery$_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$

$$P_i(k) = \overline{P_i} v_i(k)$$



$$E_i(k+1) = E_i(k) + T_s P_i(k)$$

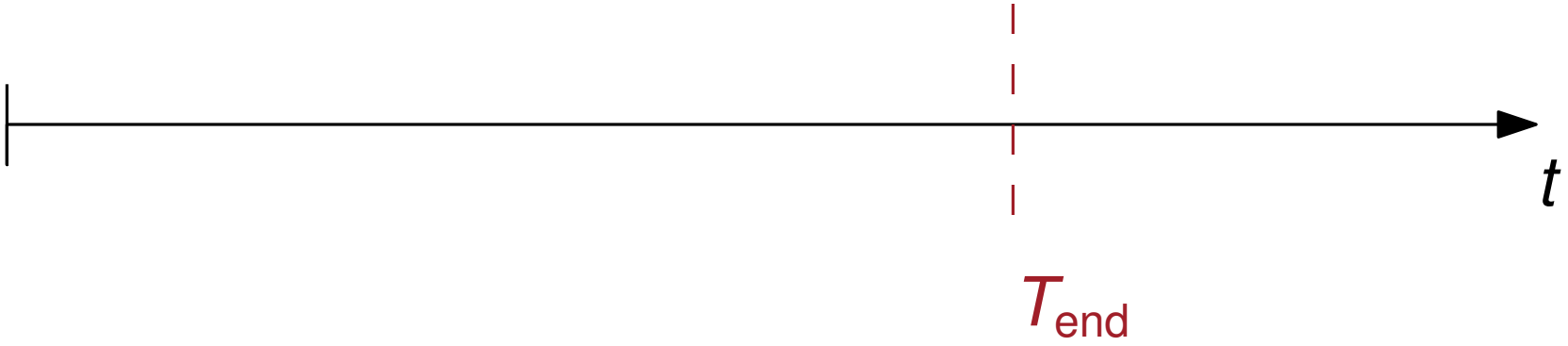
$$0 \leq \sum_{l=k}^{k+T_{\text{run},i}-1} v_i(l) - T_{\text{run},i}(v_i(k) - v_i(k-1))$$

Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen

Eigenschaften:

- Festgelegter Endzeitpunkt
- Unterbrechbar
- Festgelegter Energieverbrauch
- Variable Konsumrate

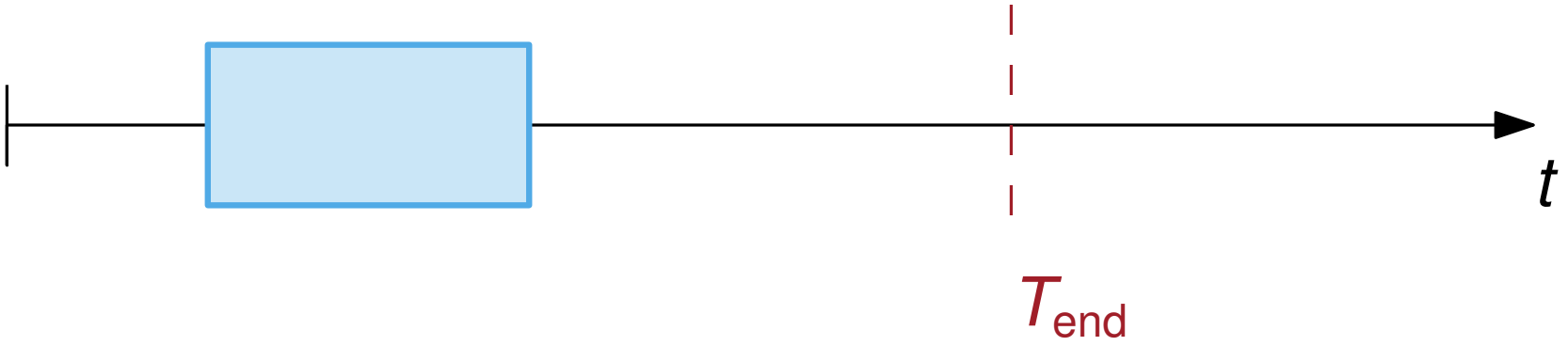


Quellen: [3]

Modellierung: Flexibilitätsklassen

Eigenschaften:

- Festgelegter Endzeitpunkt
- Unterbrechbar
- Festgelegter Energieverbrauch
- Variable Konsumrate

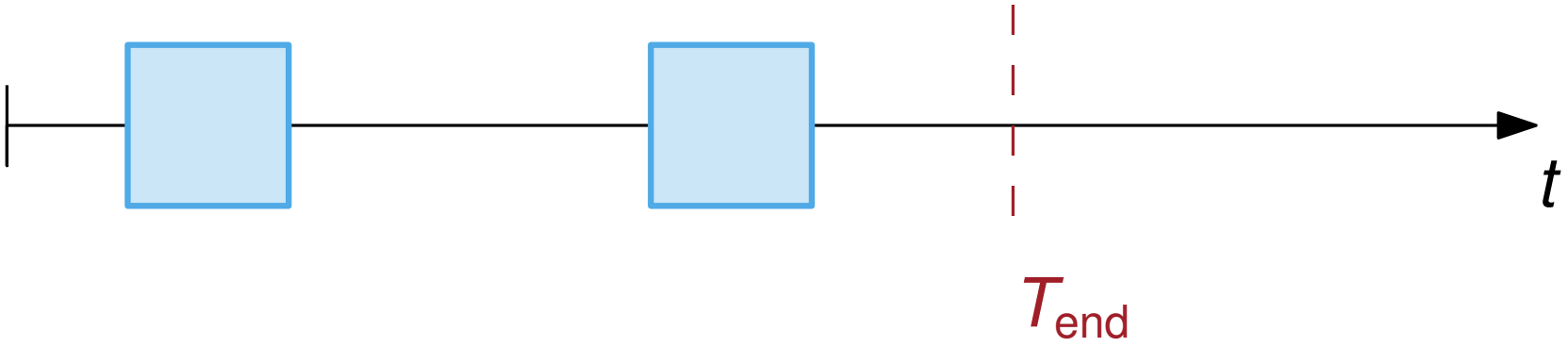


Quellen: [3]

Modellierung: Flexibilitätsklassen

Eigenschaften:

- Festgelegter Endzeitpunkt
- Unterbrechbar
- Festgelegter Energieverbrauch
- Variable Konsumrate

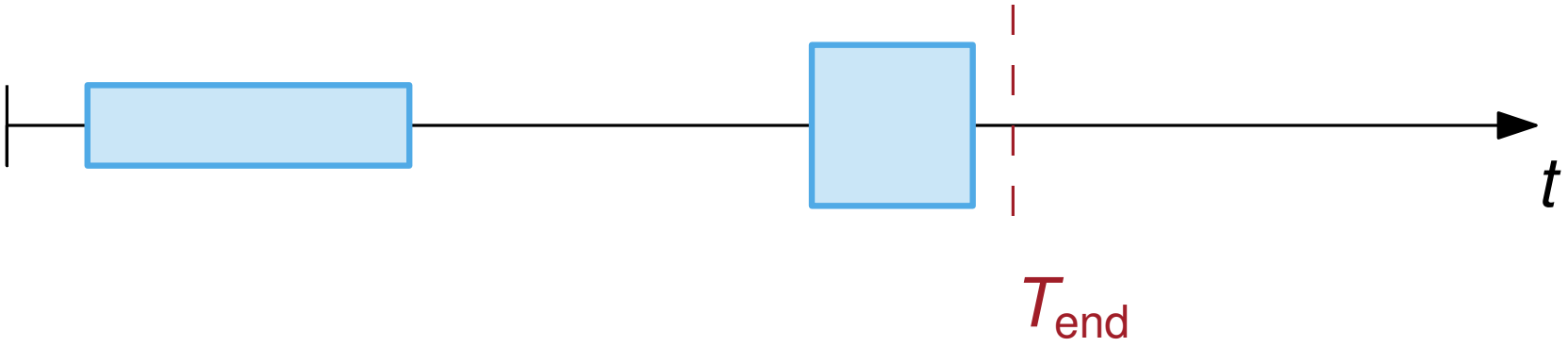


Quellen: [3]

Modellierung: Flexibilitätsklassen

Eigenschaften:

- Festgelegter Endzeitpunkt
- Unterbrechbar
- Festgelegter Energieverbrauch
- Variable Konsumrate



Quellen: [3]

Modellierung: Flexibilitätsklassen

Eigenschaften:

- Festgelegter Endzeitpunkt
- Unterbrechbar
- Festgelegter Energieverbrauch
- Variable Konsumrate

Flexibilitätsklasse: Battery

Beispiele: Laden von Elektrogeräten, Wärmepumpen, ...

Quellen: [3]

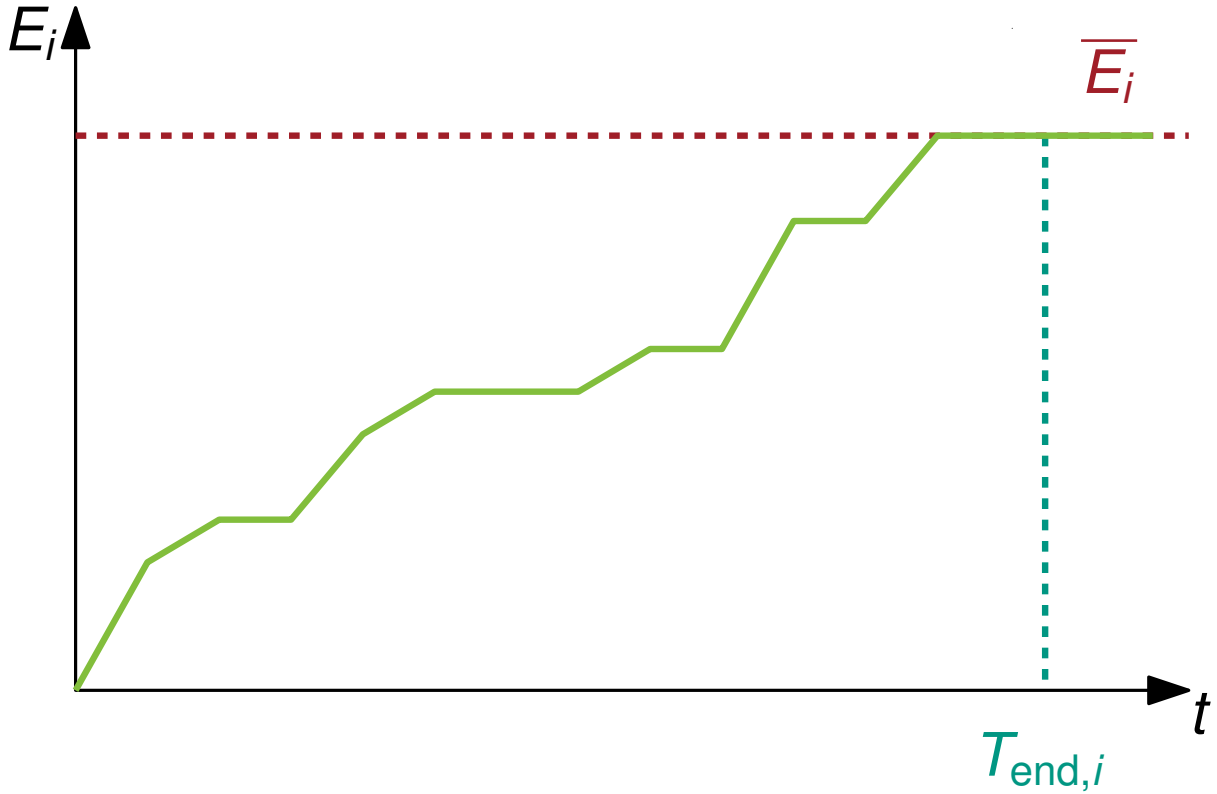
Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen

Battery$_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$



Institute of Theoretical Informatics
Algorithmics Group

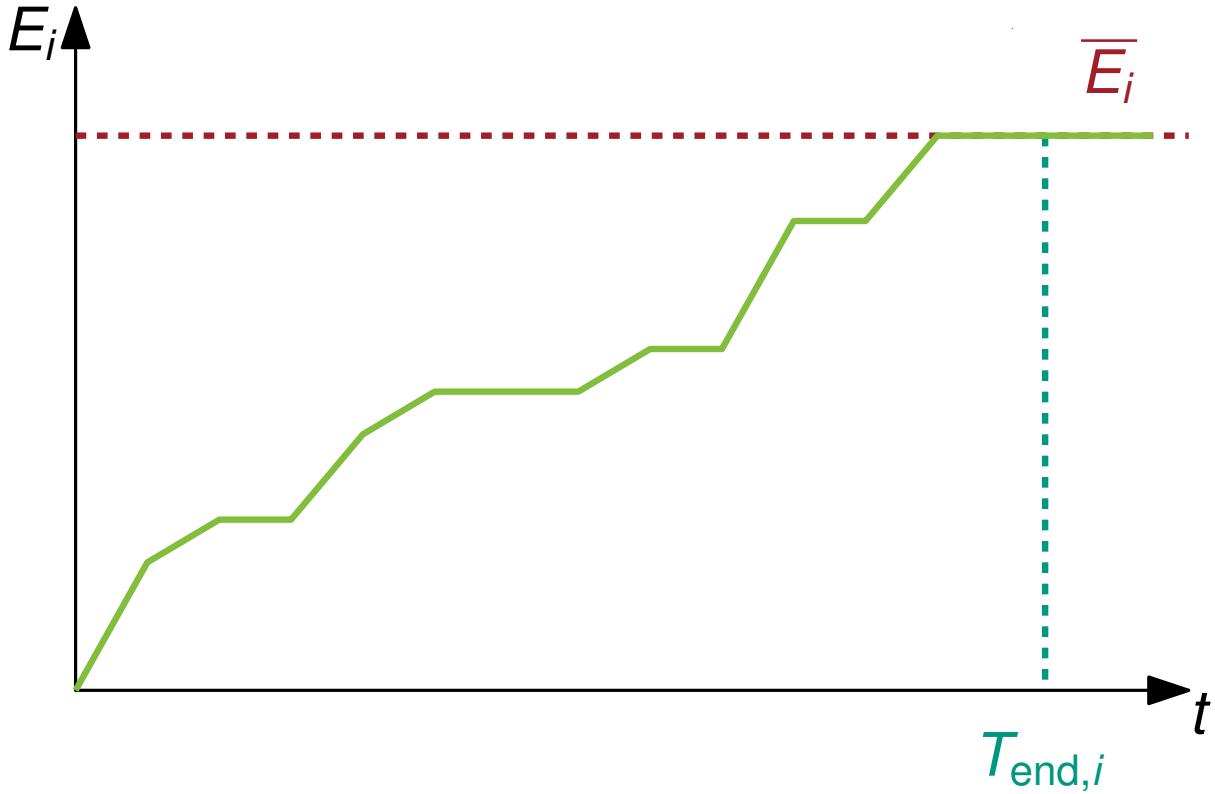
Modellierung: Flexibilitätsklassen

$\text{Battery}_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$

$$0 \leq P_i(k) \leq \overline{P_i}$$



Institute of Theoretical Informatics
Algorithmics Group

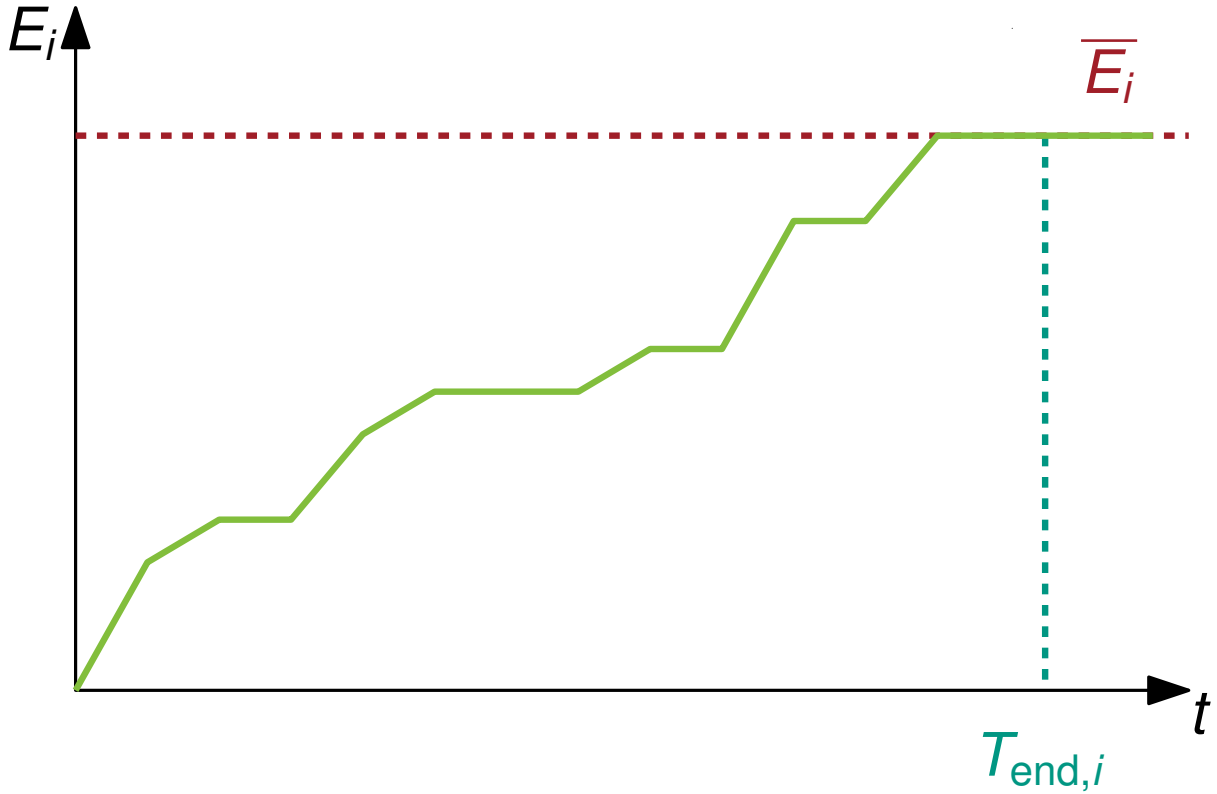
Modellierung: Flexibilitätsklassen

$\text{Battery}_i(k)$:

$$E_i(T_{\text{end},i}) = \overline{E_i}$$

$$0 \leq E_i(k) \leq \overline{E_i}$$

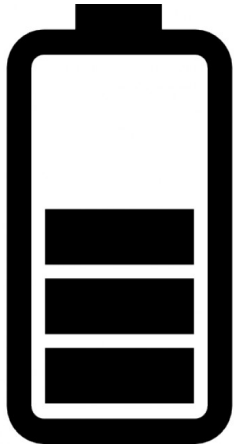
$$0 \leq P_i(k) \leq \overline{P_i}$$



$$E_i(k+1) = E_i(k) + T_s P_i(k)$$

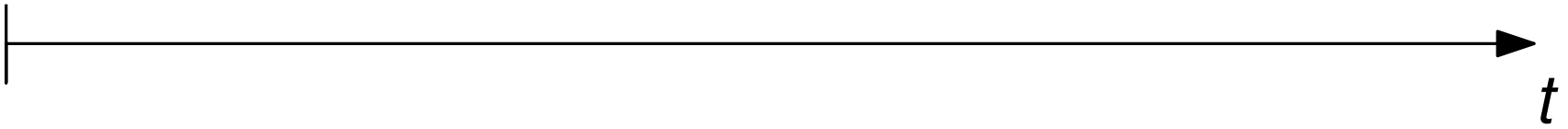
Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen



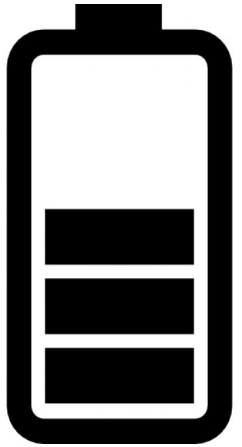
Eigenschaften:

- Variabler Endzeitpunkt
- Unterbrechbar
- Variable Konsum- bzw. Einspeiserate
- Variable Betriebsdauer



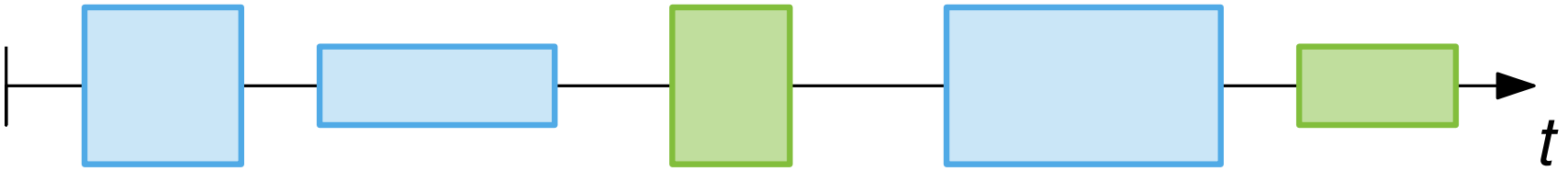
Quellen: [4]

Modellierung: Flexibilitätsklassen



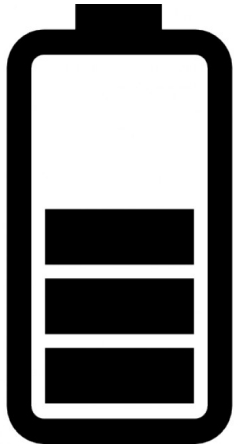
Eigenschaften:

- Variabler Endzeitpunkt
- Unterbrechbar
- Variable Konsum- bzw. Einspeiserate
- Variable Betriebsdauer



Quellen: [4]

Modellierung: Flexibilitätsklassen



Eigenschaften:

- Variabler Endzeitpunkt
- Unterbrechbar
- Variable Konsum- bzw. Einspeiserate
- Variable Betriebsdauer

Flexibilitätsklasse: Bucket

Beispiele: Energiespeicher aller Art

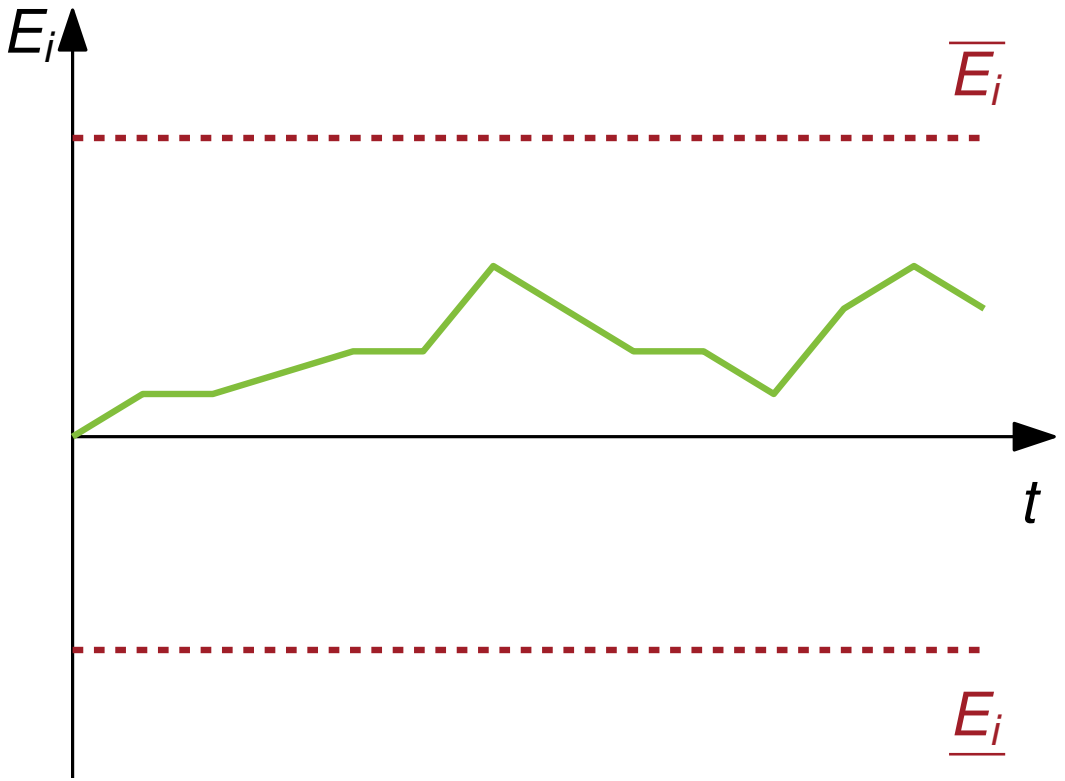
Quellen: [4]

Institute of Theoretical Informatics
Algorithmics Group

Modellierung: Flexibilitätsklassen

Bucket$_i(k)$:

$$\underline{E_i} \leq E_i(k) \leq \overline{E_i}$$

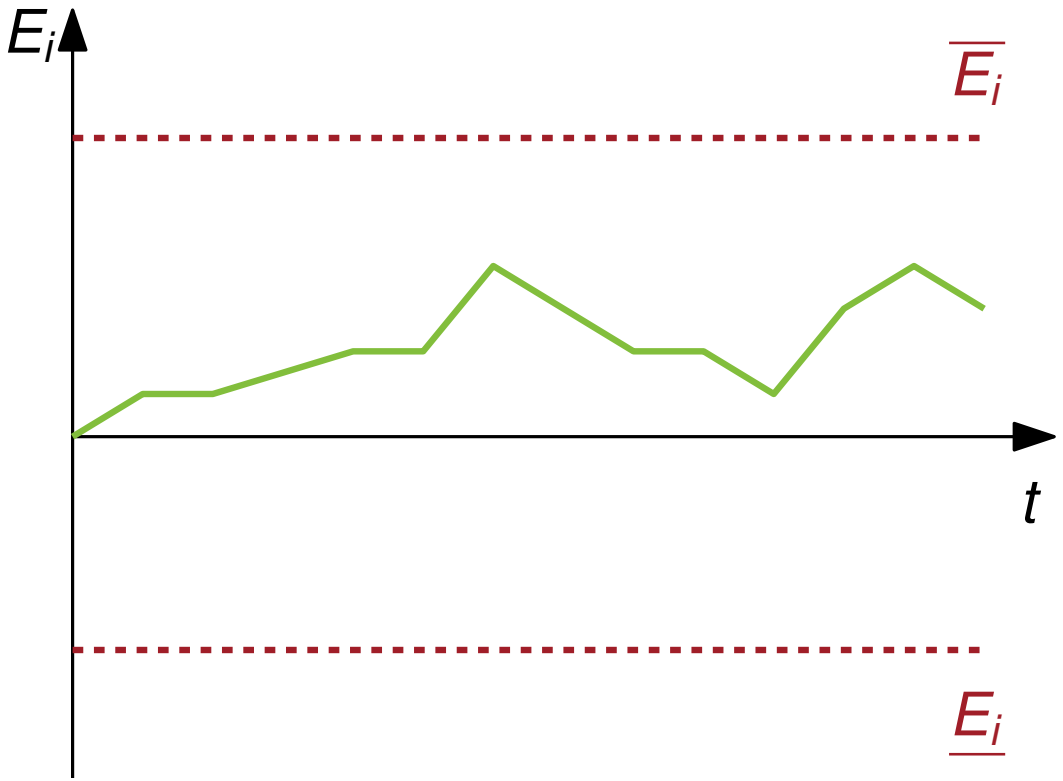


Modellierung: Flexibilitätsklassen

Bucket$_i(k)$:

$$\underline{E_i} \leq E_i(k) \leq \overline{E_i}$$

$$\underline{P_i} \leq P_i(k) \leq \overline{P_i}$$

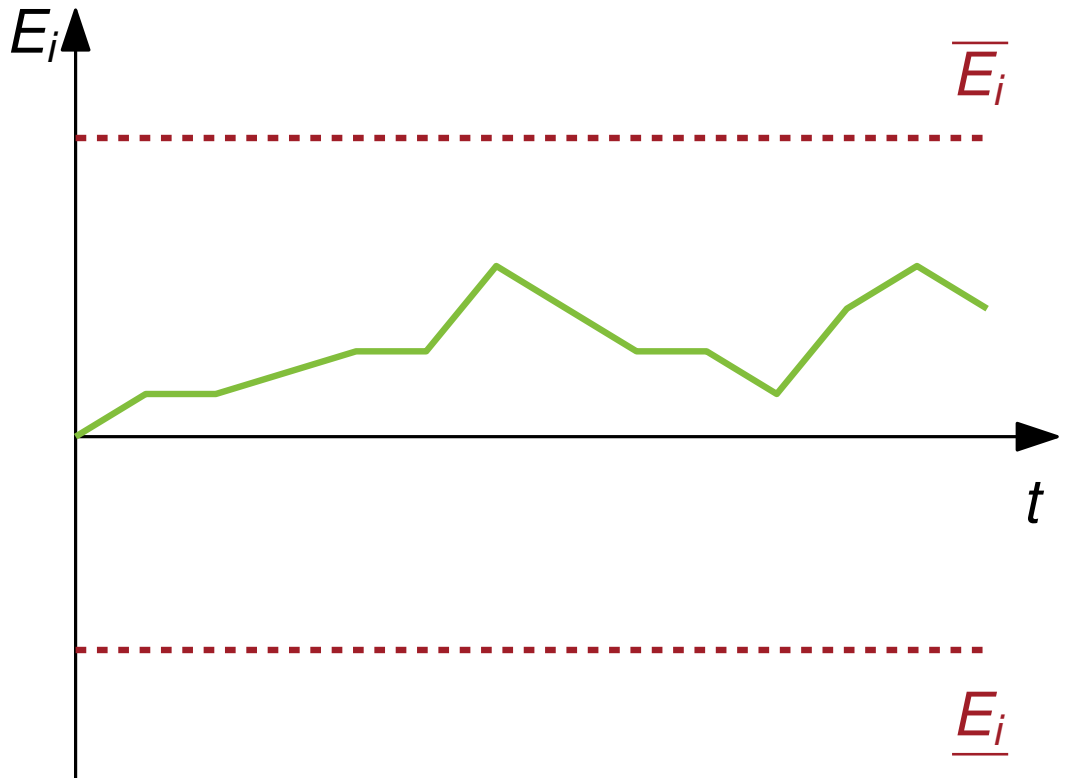


Modellierung: Flexibilitätsklassen

Bucket$_i(k)$:

$$\underline{E_i} \leq E_i(k) \leq \overline{E_i}$$

$$\underline{P_i} \leq P_i(k) \leq \overline{P_i}$$



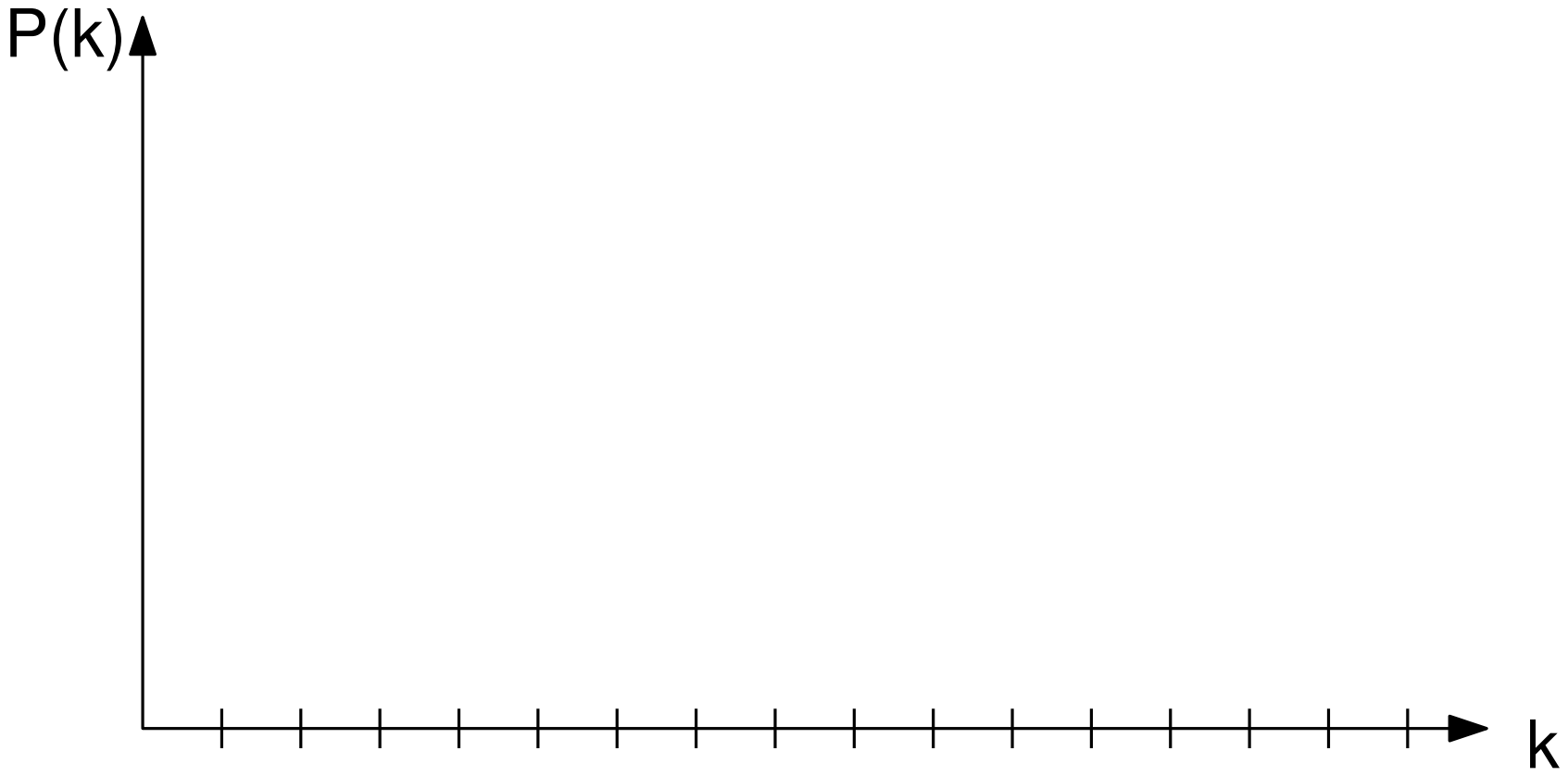
$$E_i(k+1) = E_i(k) + T_s P_i(k)$$

Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

Bestandteile des Systems:

- Vorhersagehorizont: $K \in \mathbb{N}$

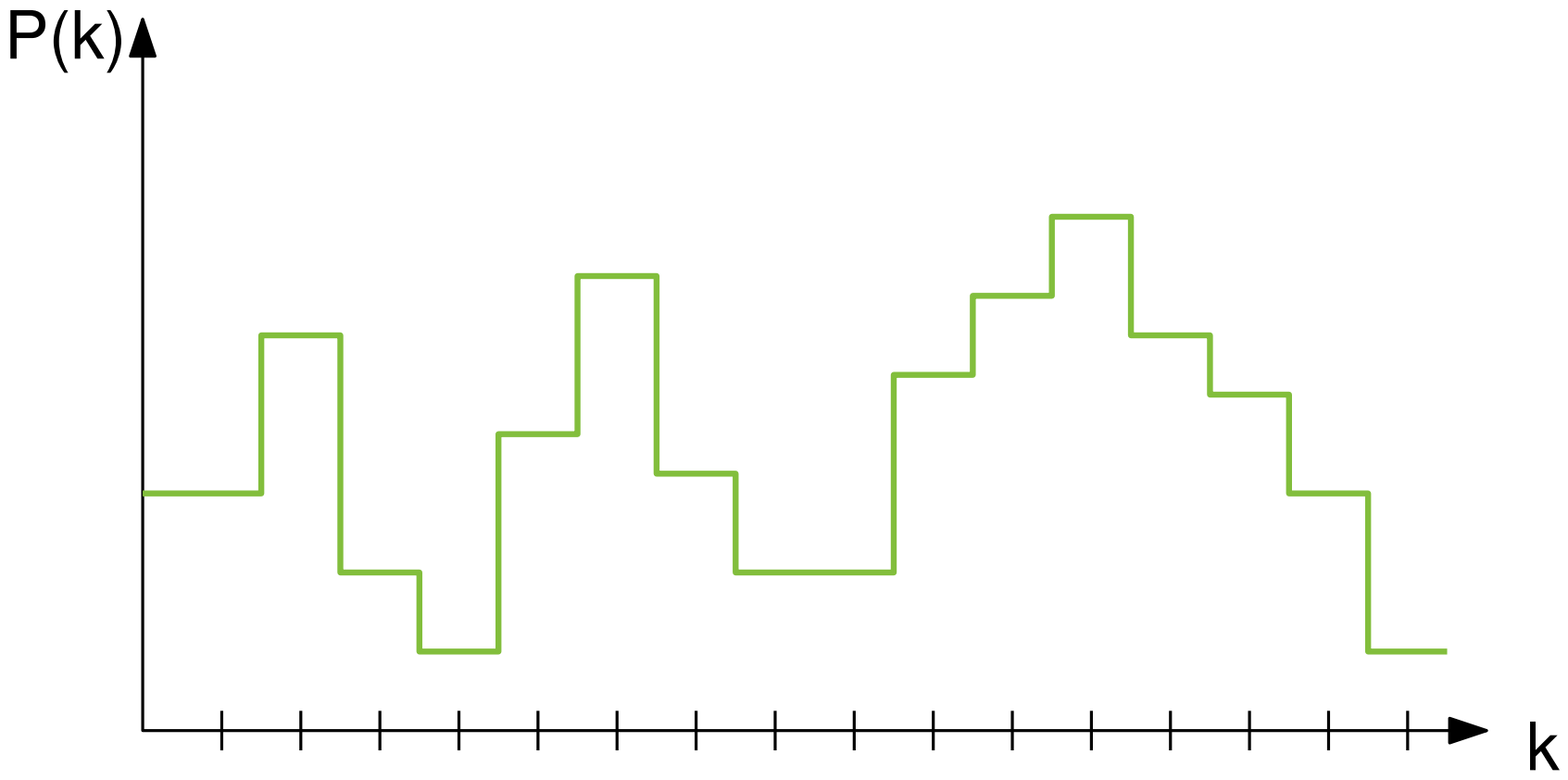


Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

Bestandteile des Systems:

- vituelles Kraftwerk mit fluktuierender Leistung: $P_{Dispatch}(k)$, $k = 0, ..., K$

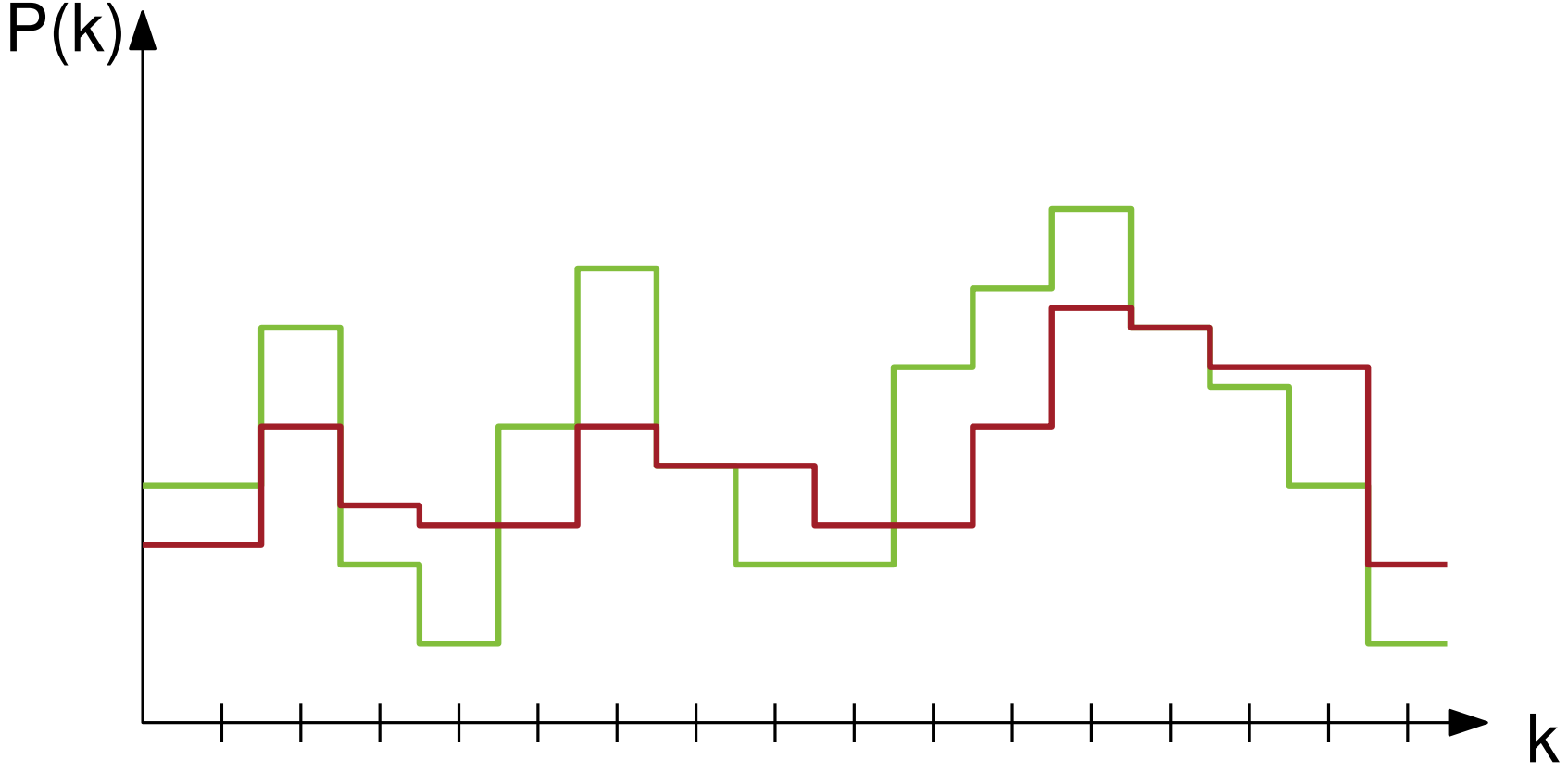


Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

Bestandteile des Systems:

- N lokale Verbraucher: $\{LU_i\}_{i=1,\ldots,N}$
- $P_i(k)$ an der Stelle k an LU_i abgegebene Leistung

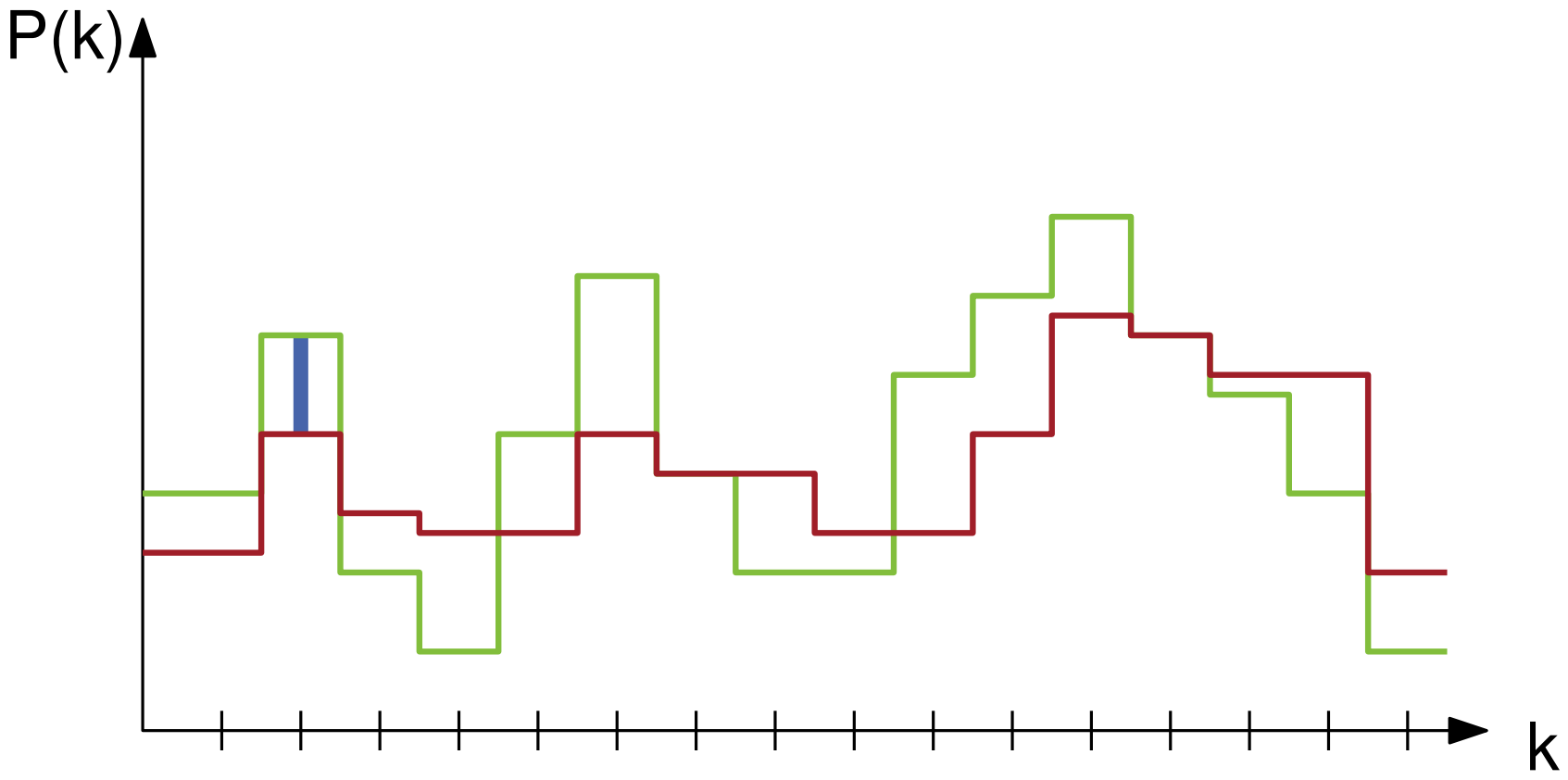


Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

Bestandteile des Systems:

- $S(k)$ Leistung, die an der Stelle k nicht bereitgestellt werden kann

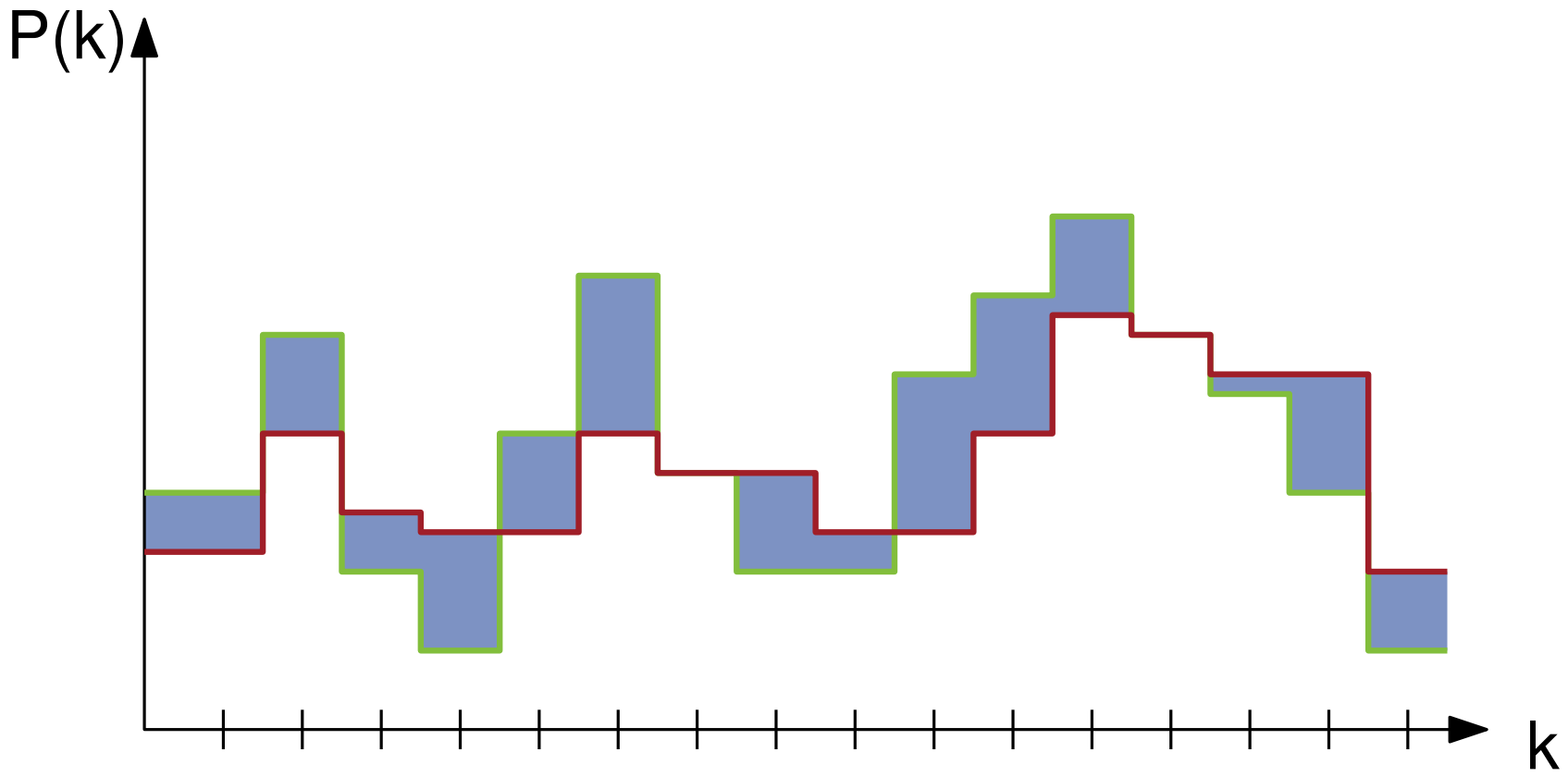


Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

Bestandteile des Systems:

Ziel: Minimieren von $\sum |S|$



Institute of Theoretical Informatics
Algorithmics Group

Problemstellung: Power Plant Dispatch Problem

gesucht:

$$min_{P_i(\cdot)} \sum_{k=0}^{\infty} |S(k)|$$

so, dass:

$$P_{Dispatch}(k) \in \mathbb{R}, k = 0, ..., \infty$$

$$\sum_{i=1}^{N} P_i(k) + S(k) = P_{Dispatch}(k)$$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmen

Anforderungen

- Die Lösung soll schnell gefunden werden
- Der Speicherbedarf soll nicht zu groß sein

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- Die Lösung soll schnell gefunden werden
- Der Speicherbedarf soll nicht zu groß sein

Problem:

- NP-schweres Problem
- große Probleminstanzen

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- Die Lösung soll schnell gefunden werden
- Der Speicherbedarf soll nicht zu groß sein

Problem:

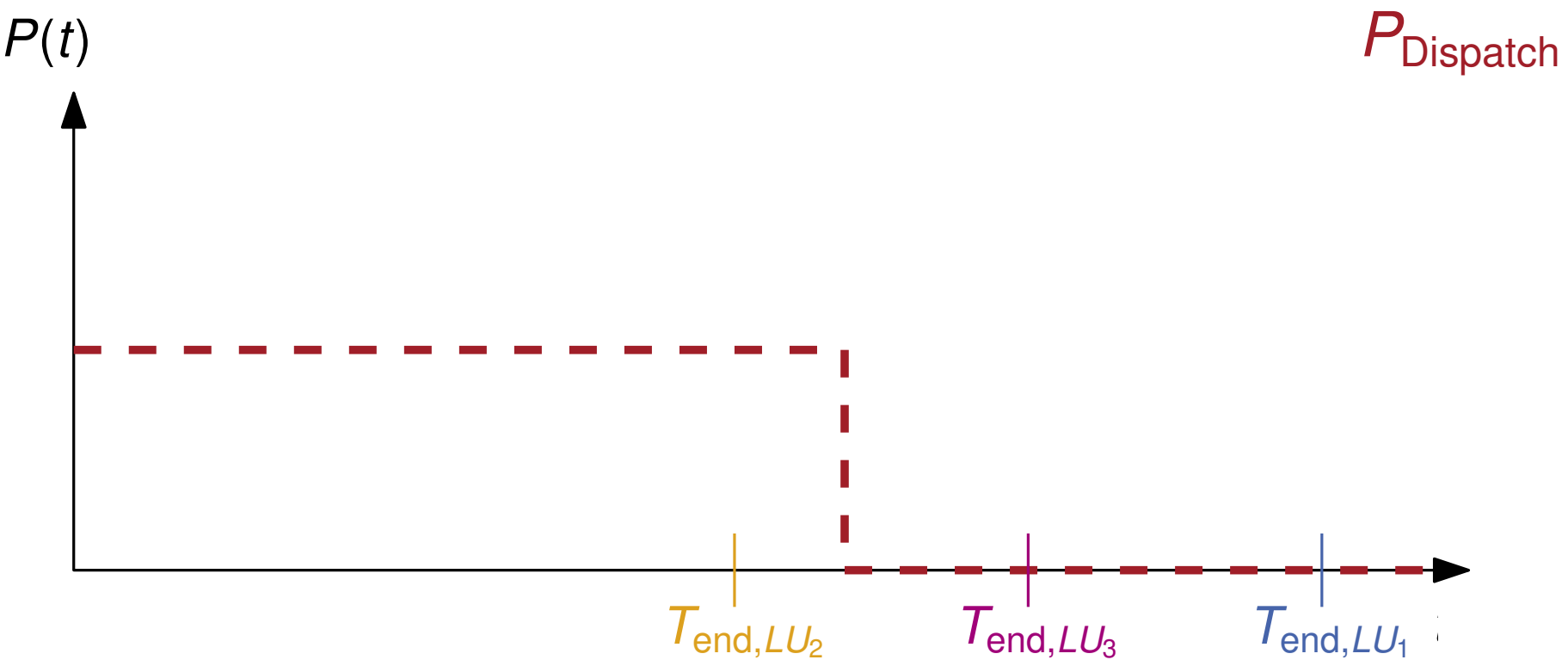
- NP-schweres Problem
- große Probleminstanzen

Verwendung von heuristischen Algorithmen

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

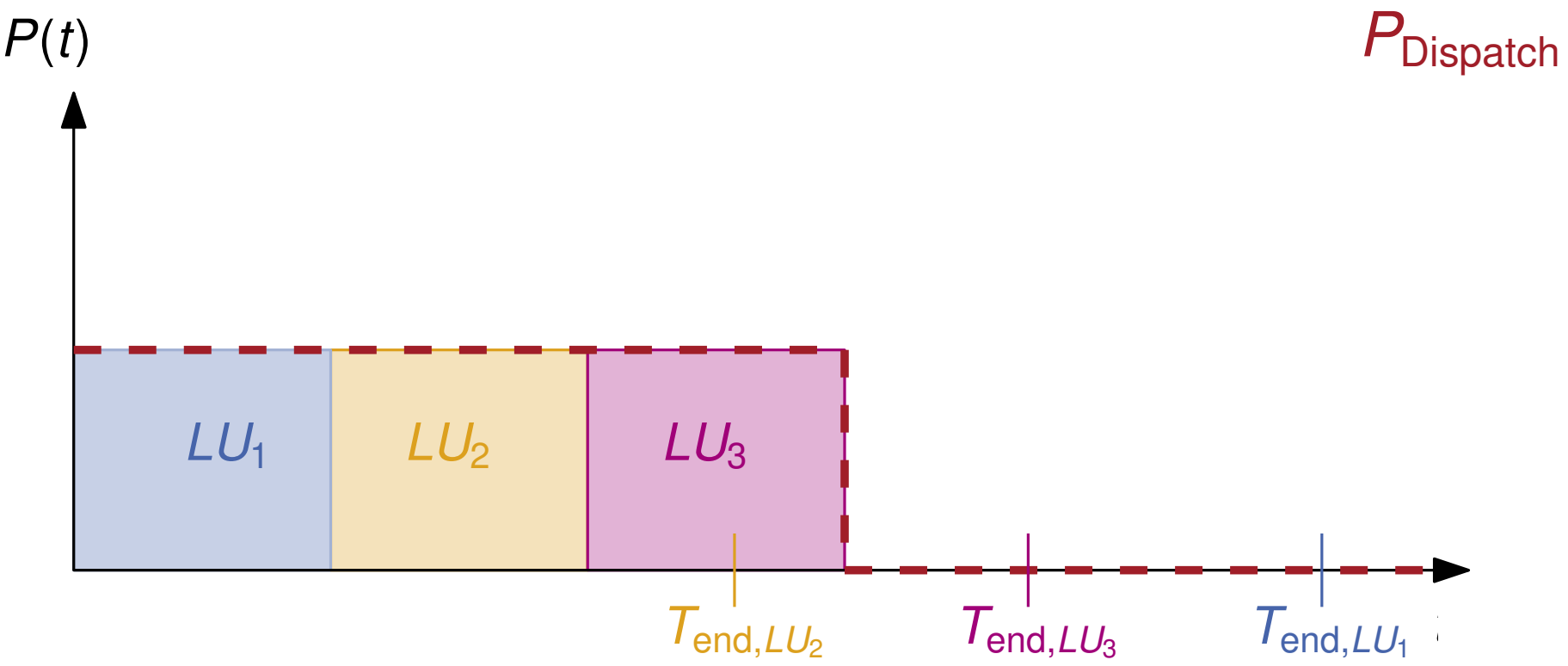
- P_{Dispatch} ist **nur** eine Vorhersage



In Anlehnung an [6]

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage

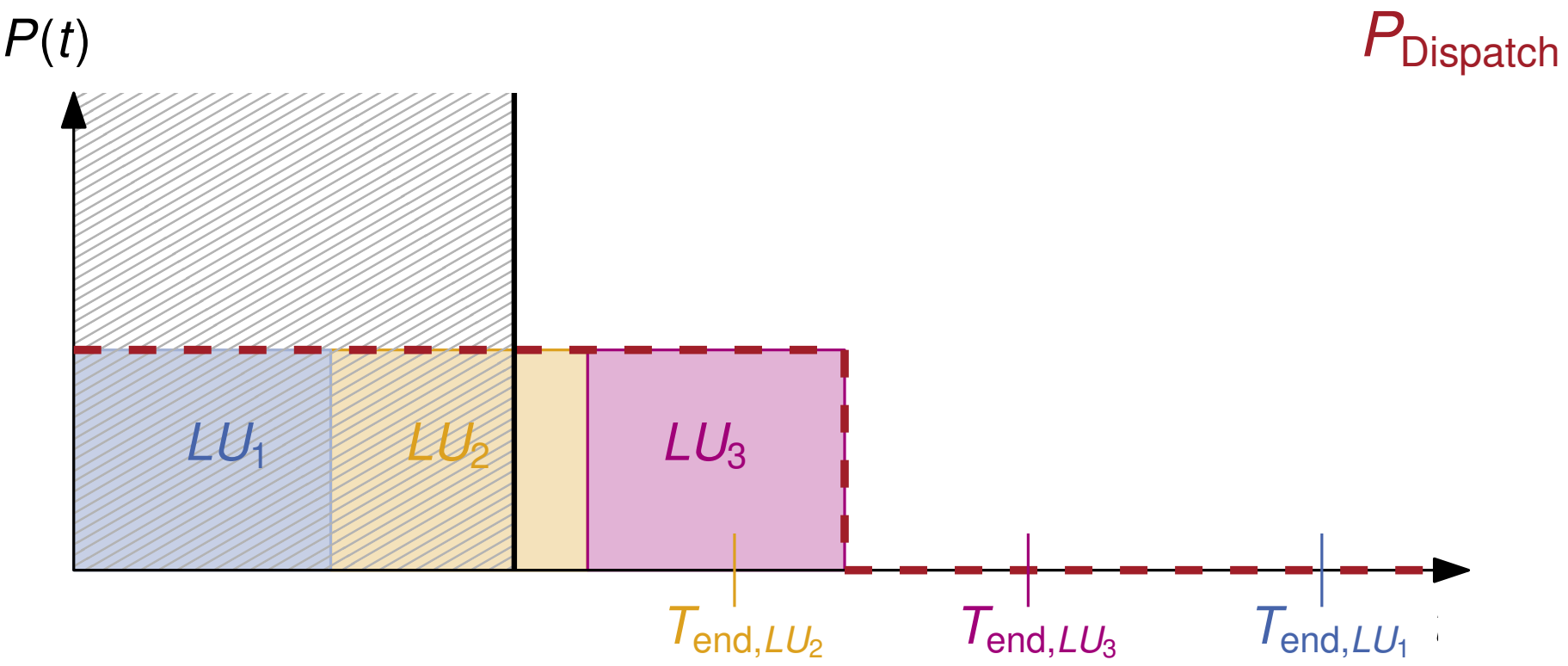


In Anlehnung an [6]

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage

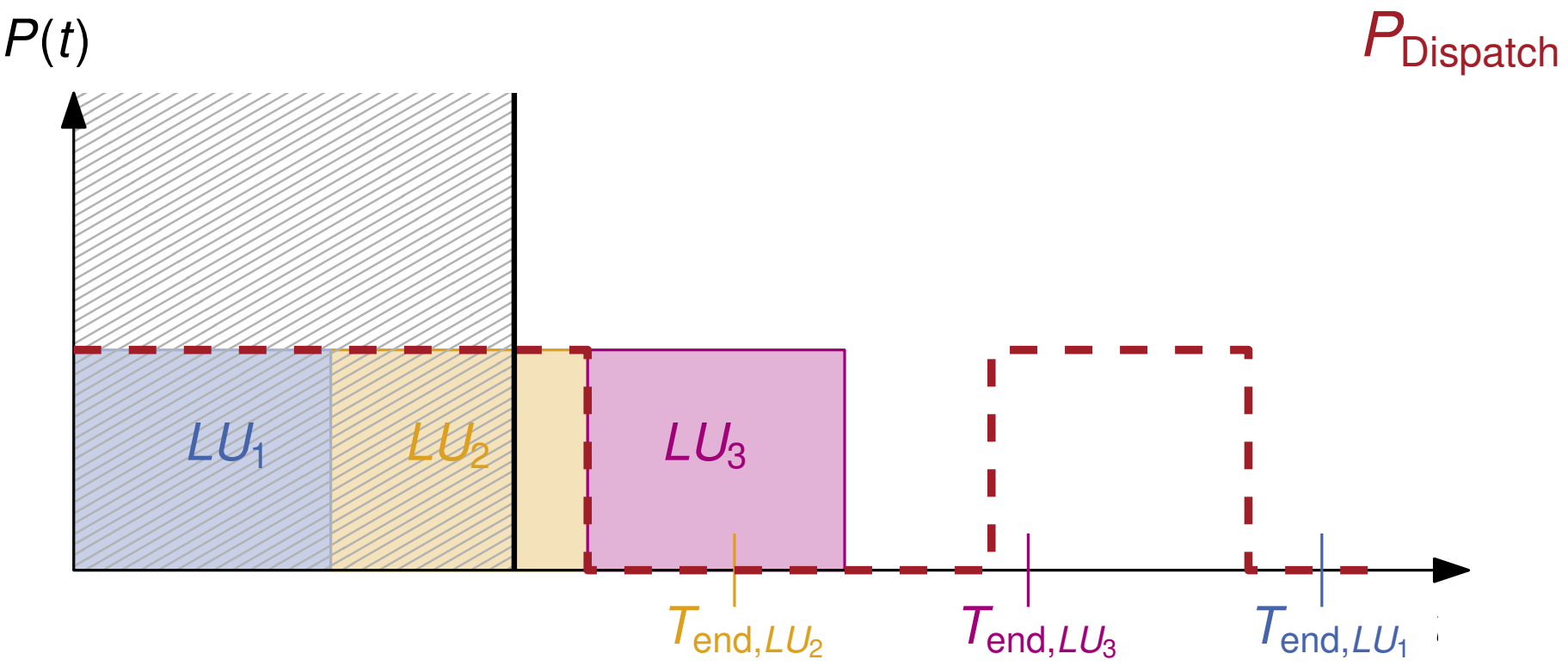


In Anlehnung an [6]

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage

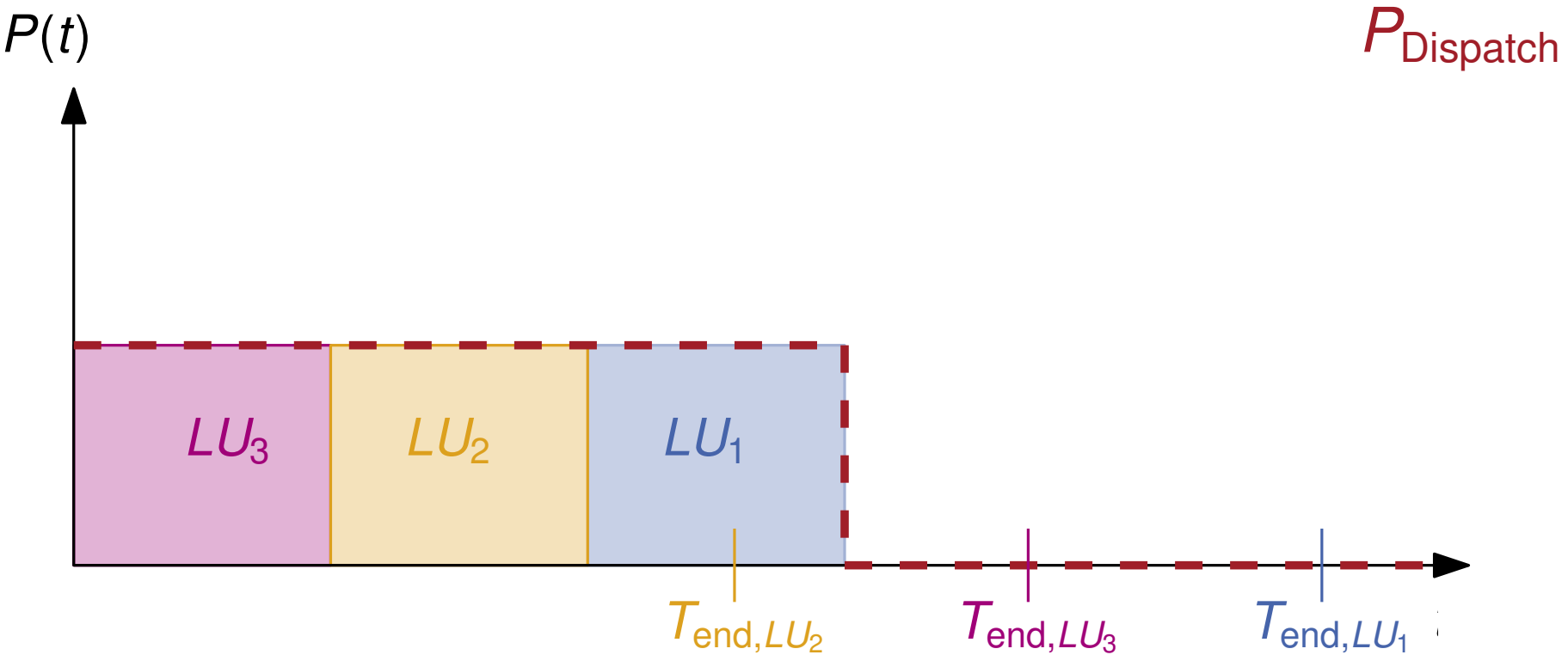


In Anlehnung an [6]

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage

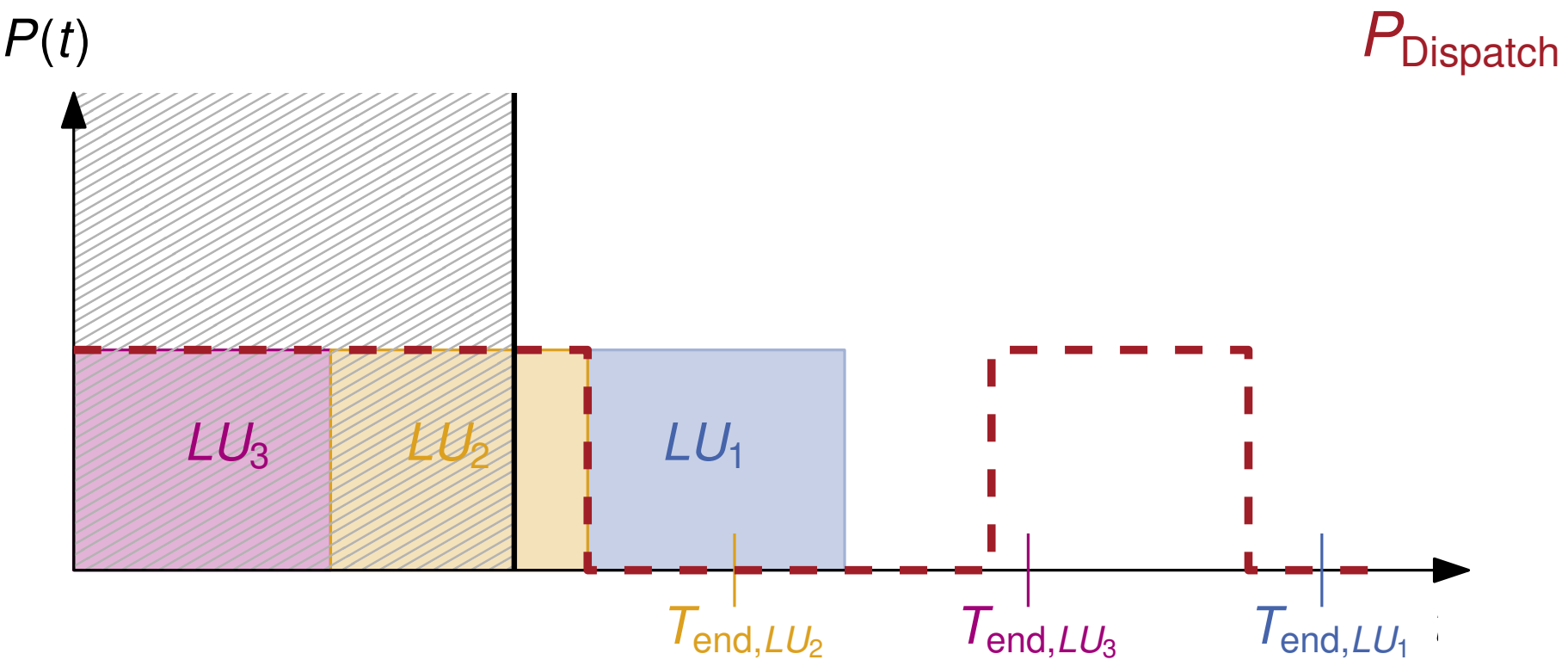


In Anlehnung an [6]

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

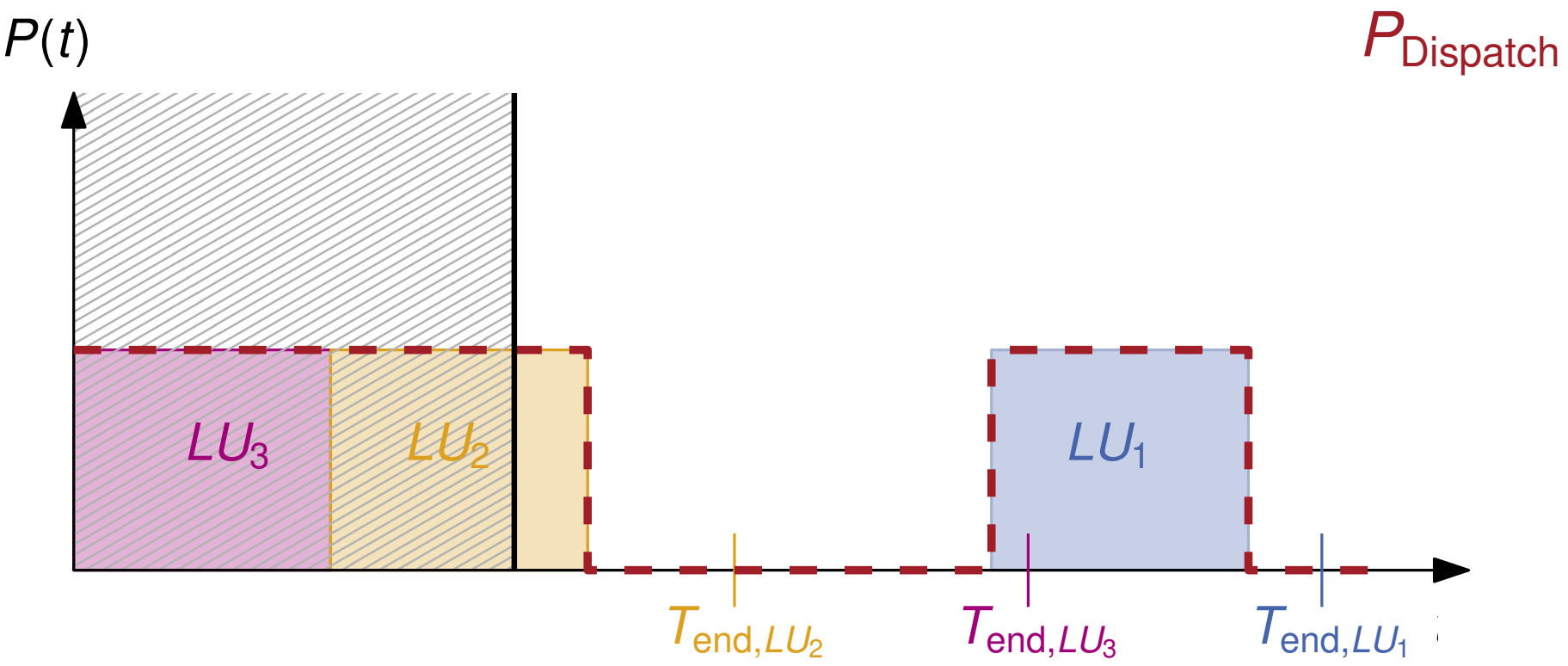
- P_{Dispatch} ist **nur** eine Vorhersage



In Anlehnung an [6]

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage



In Anlehnung an [6]

Institute of Theoretical Informatics
Algorithmics Group

Anforderungen

- P_{Dispatch} ist **nur** eine Vorhersage

P_{Dispatch} soll so genutz werden, dass:

- Die Energie möglichst optimal genutz wird
- die Deadlines der Units eingehalten werden
- möglichst zu späteren Zeitpunkten noch Flexibilität vorhanden ist

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Idee:

- Bakeries und Batteries zuerst in Reihenfolge der Dringlichkeit abarbeiten
- Buckets als Energie-Puffer verwenden

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Idee:

- Bakeries und Batteries zuerst in Reihenfolge der Dringlichkeit abarbeiten
- Buckets als Energie-Puffer verwenden

Wie definiert sich die Dringlichkeit?

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

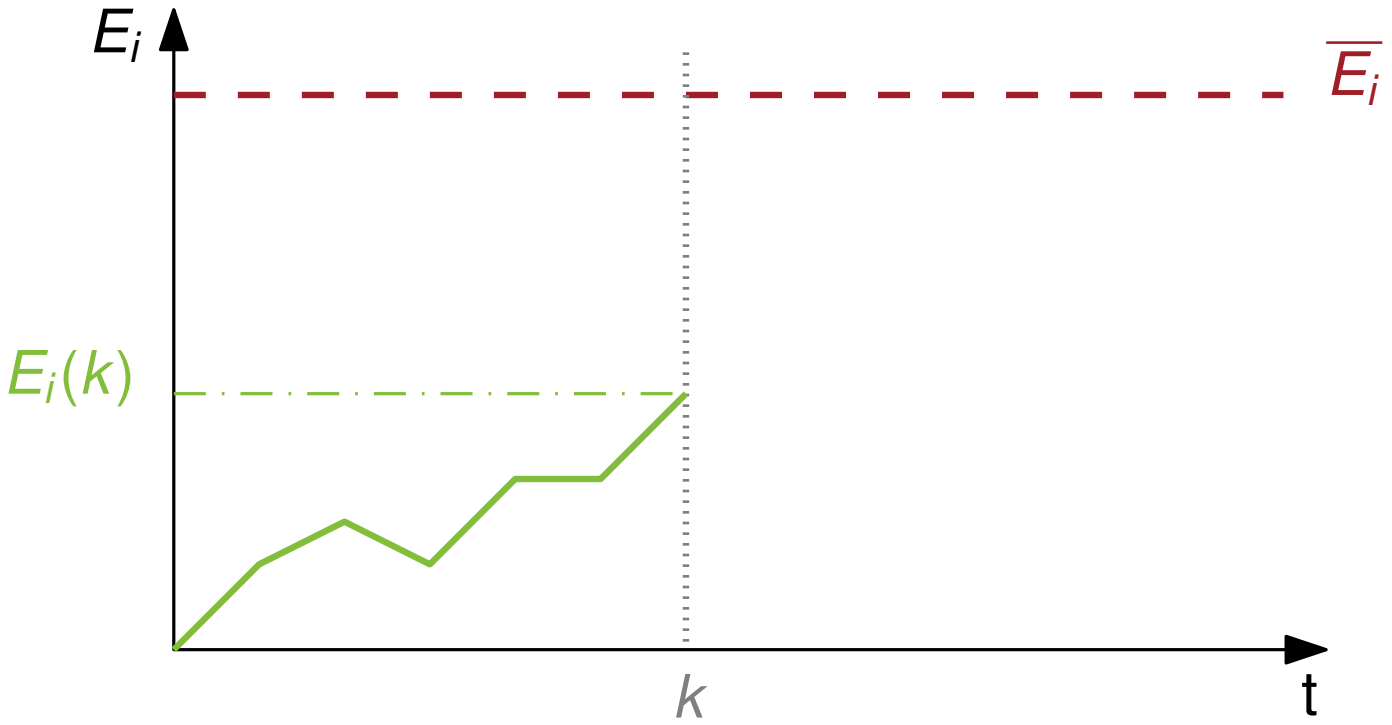
Agilitätsfaktoren:

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bucket$_i(k)$:

$$\mathcal{K}_i^{\text{Bucket}}(k) = \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



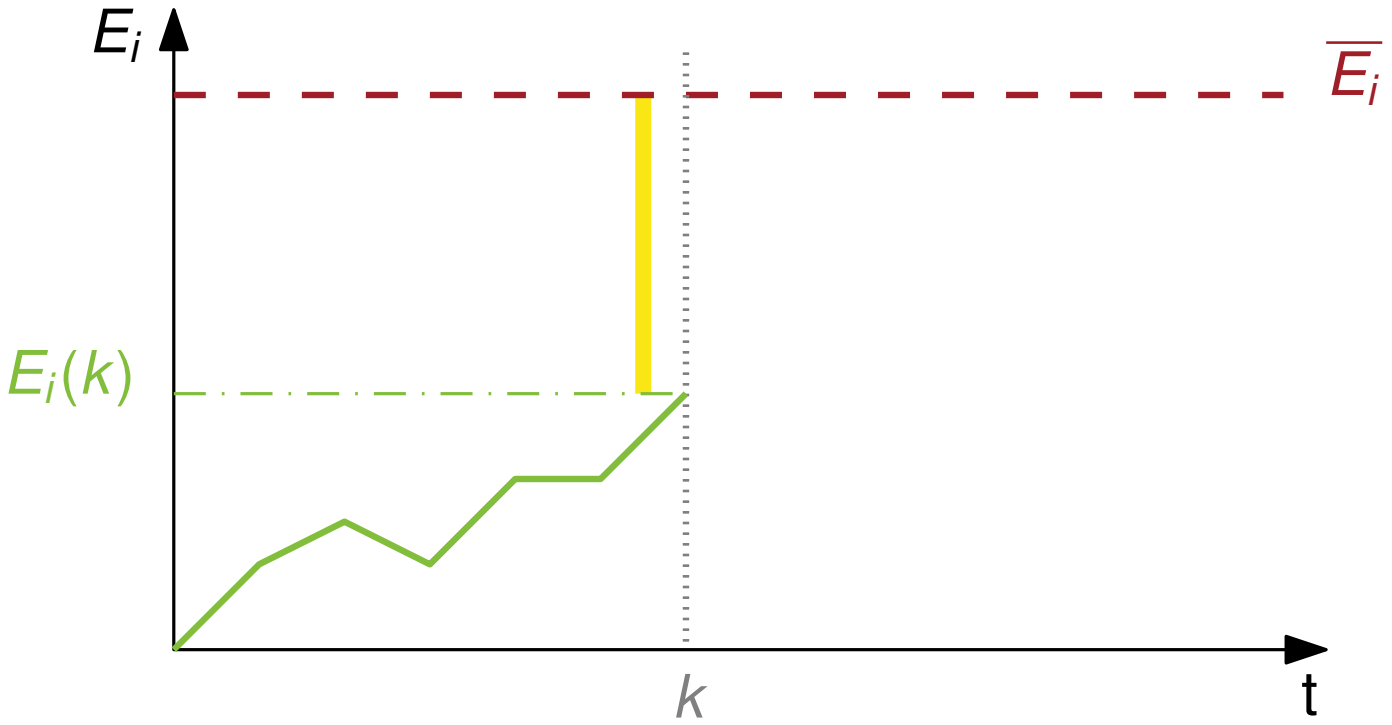
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bucket$_i(k)$:

$$\mathcal{K}_i^{\text{Bucket}}(k) = \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



Institute of Theoretical Informatics
Algorithmics Group

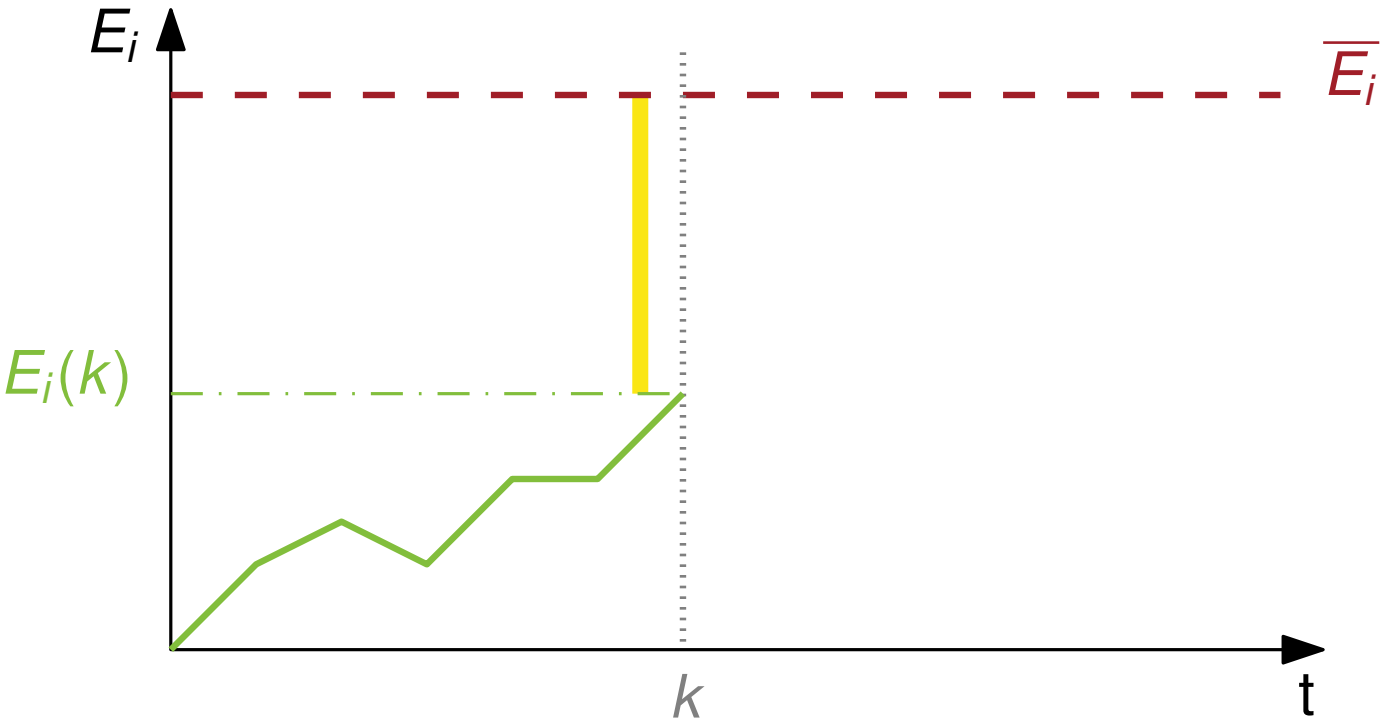
Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bucket$_i(k)$:

$$\mathcal{K}_i^{\text{Bucket}}(k) = \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$

maximaler Energiezugewinn pro Zeitschritt



Institute of Theoretical Informatics
Algorithmics Group

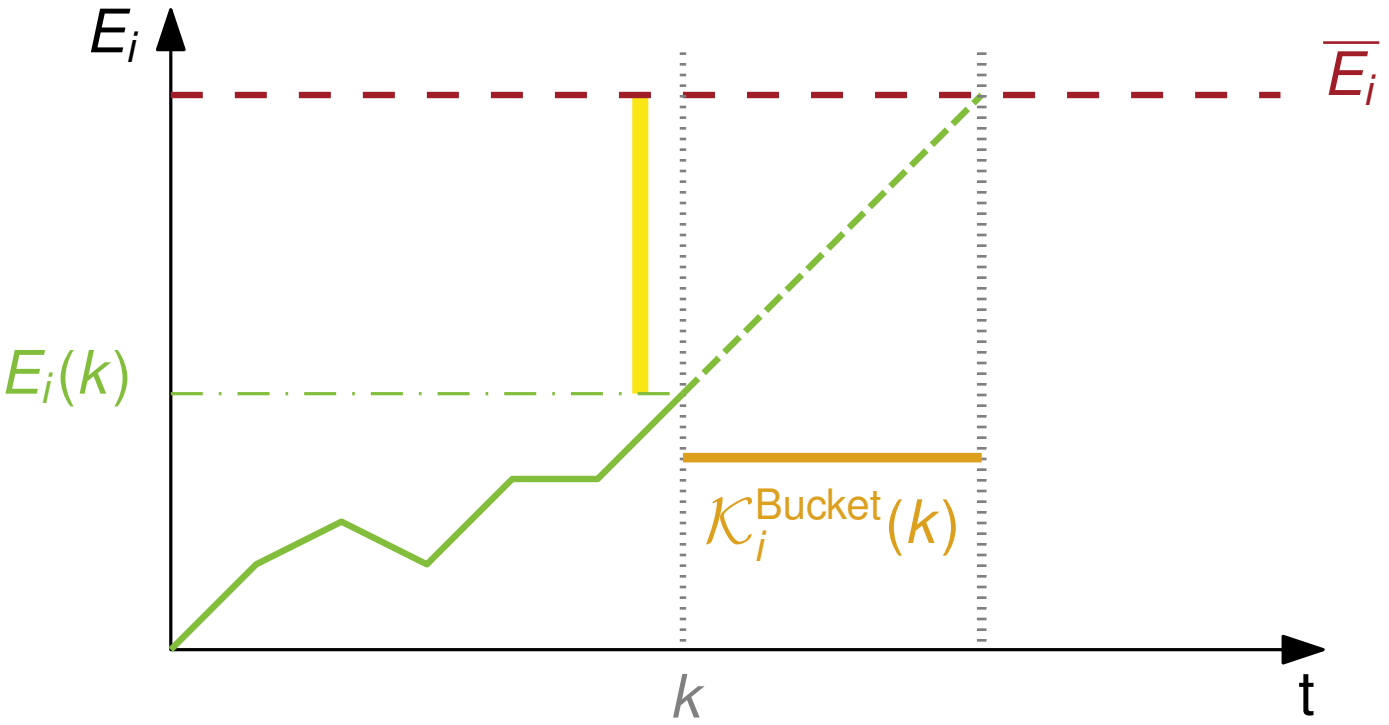
Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bucket$_i(k)$:

$$\mathcal{K}_i^{\text{Bucket}}(k) = \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$

maximaler Energiezugewinn pro Zeitschritt



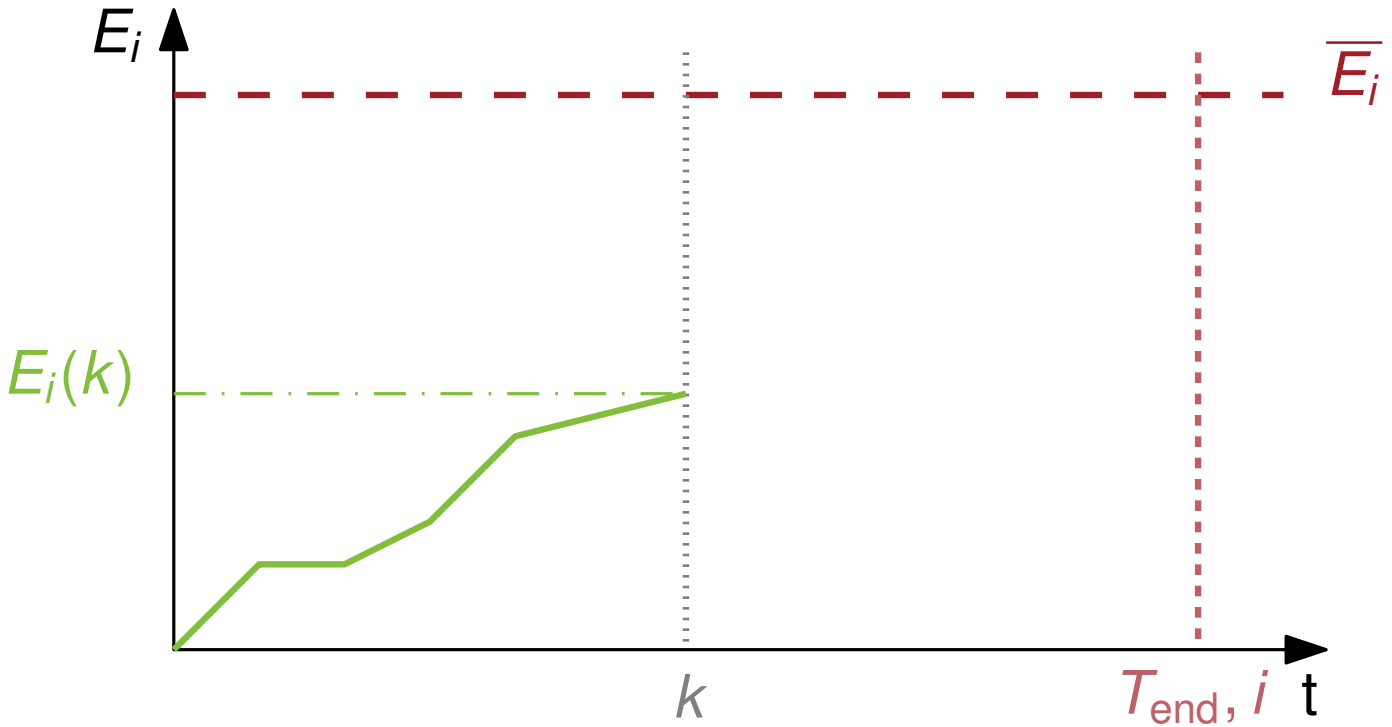
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Battery$_i(k)$:

$$\mathcal{K}_i^{\text{Battery}}(k) = T_{\text{end},i} - k - \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



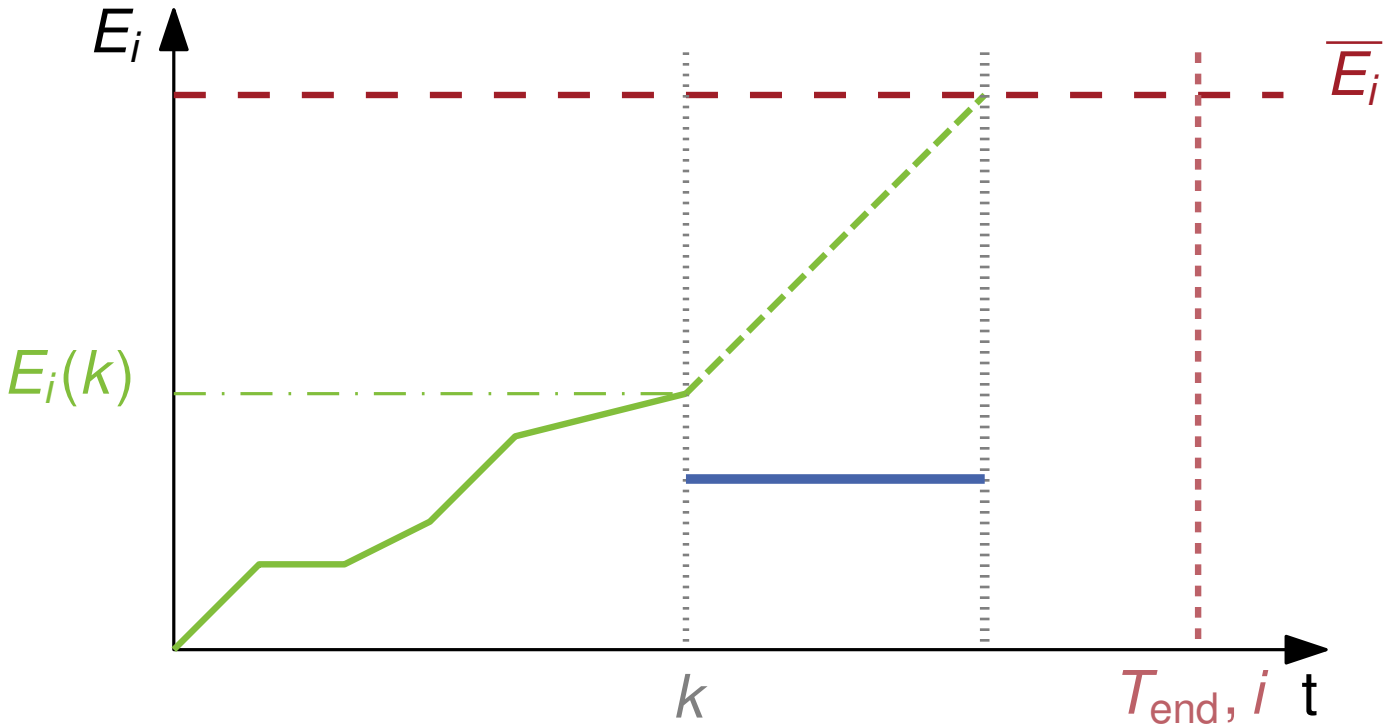
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Battery$_i(k)$:

$$\mathcal{K}_i^{\text{Battery}}(k) = T_{\text{end},i} - k - \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



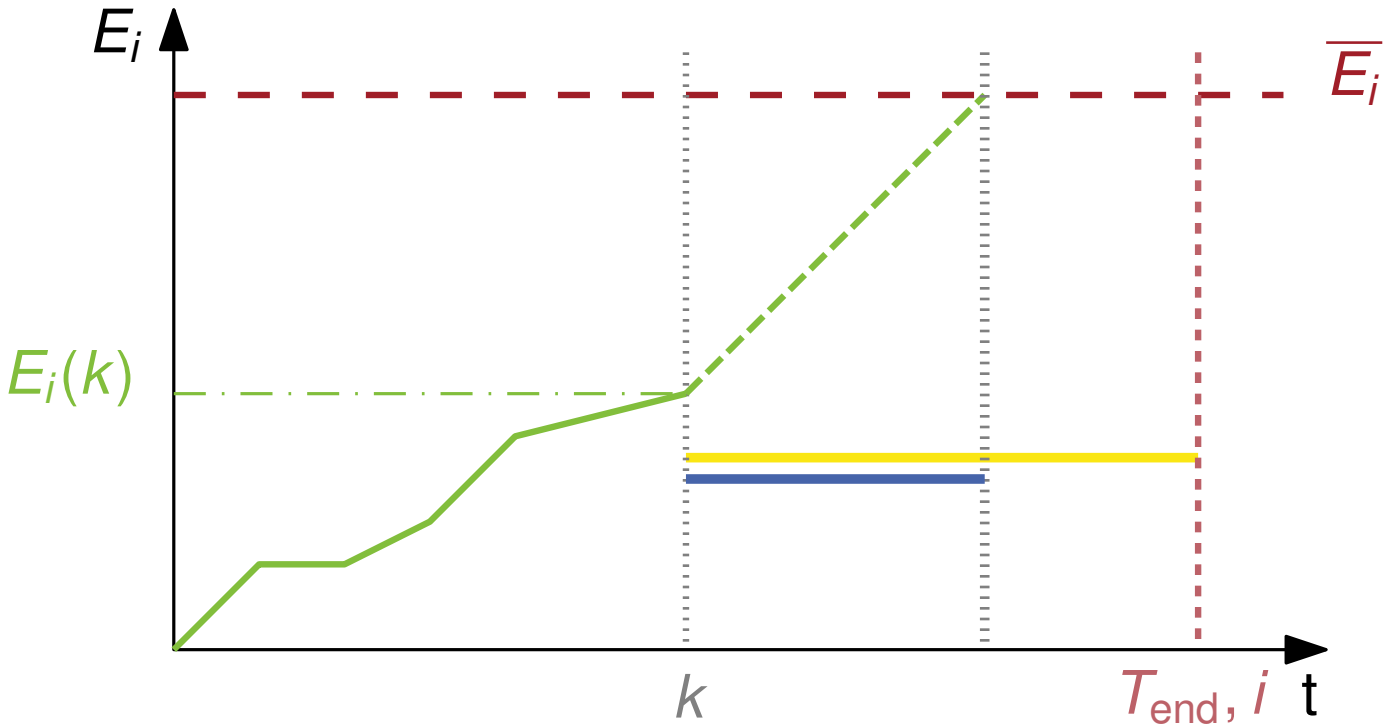
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Battery$_i(k)$:

$$\mathcal{K}_i^{\text{Battery}}(k) = T_{\text{end},i} - k - \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



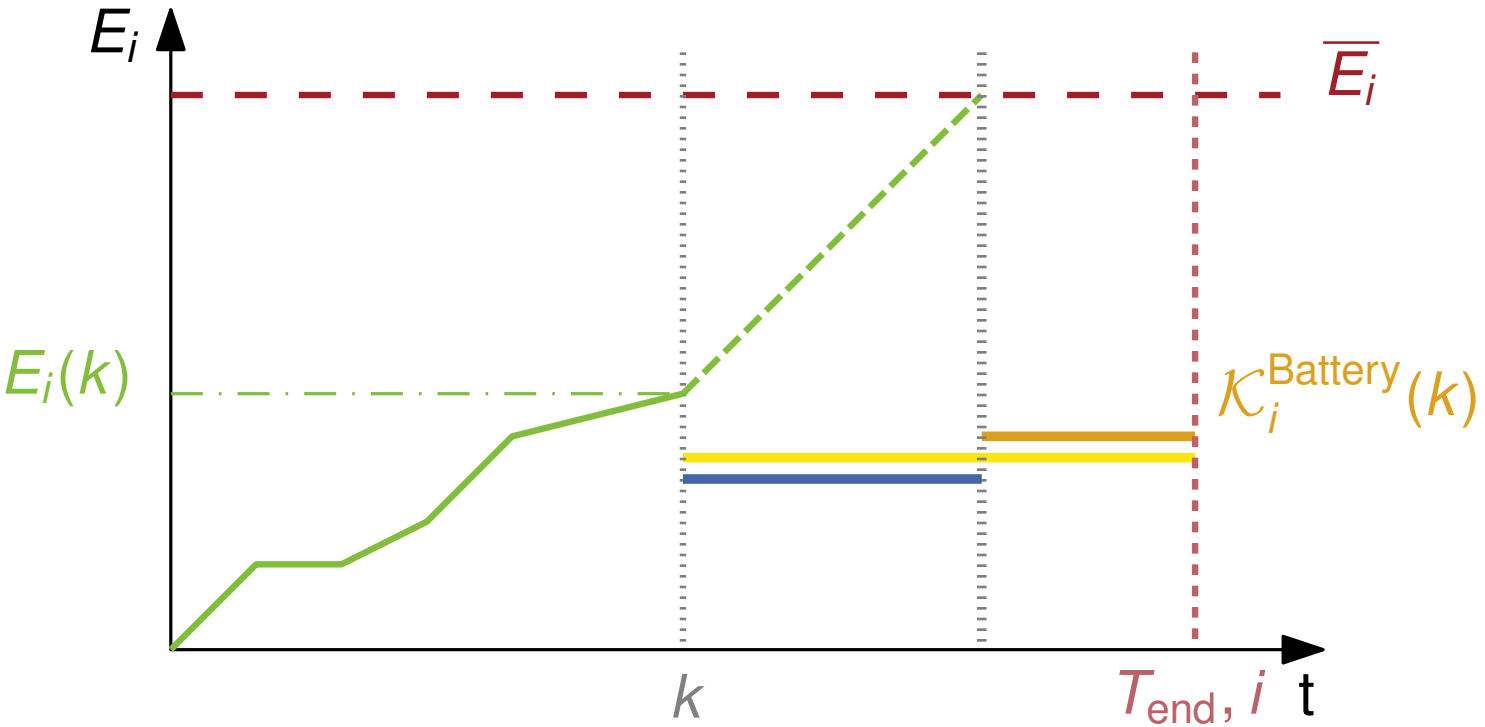
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Battery$_i(k)$:

$$\mathcal{K}_i^{\text{Battery}}(k) = T_{\text{end},i} - k - \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$



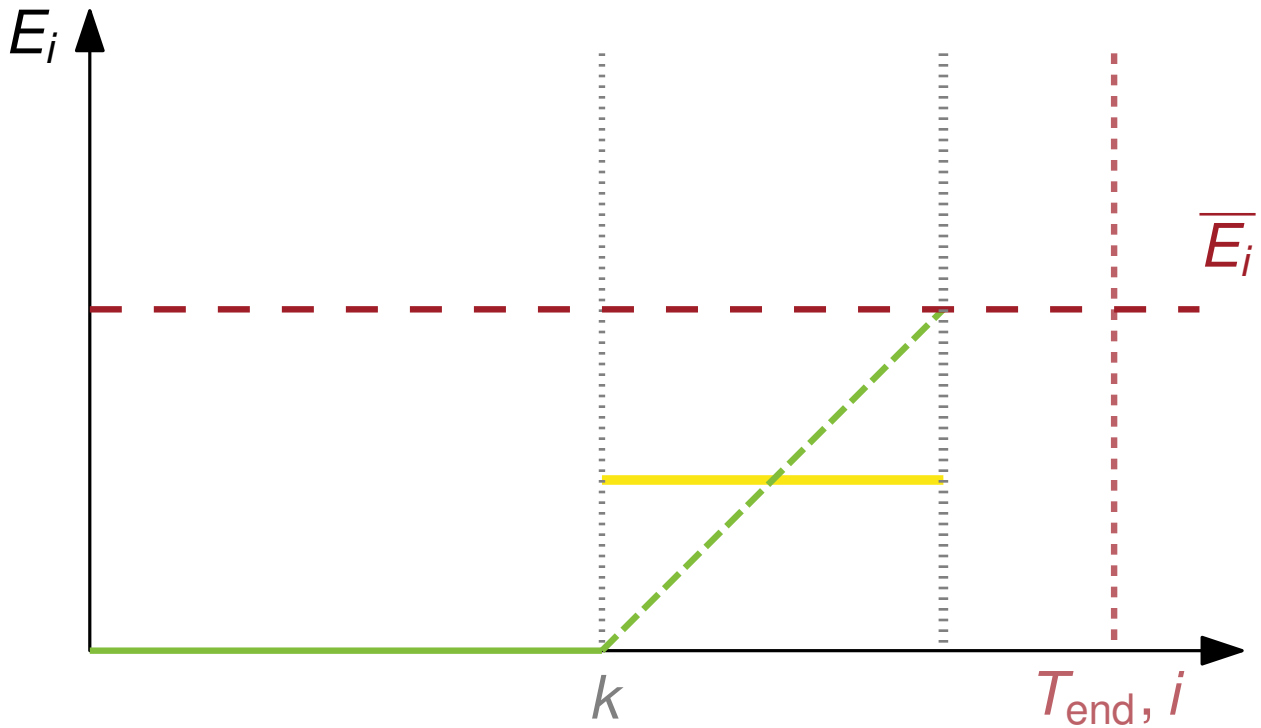
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bakery$_i(k)$:

$$\mathcal{K}_i^{\text{Bakery}}(k) = T_{\text{end},i} - T_{\text{run},i} - k$$



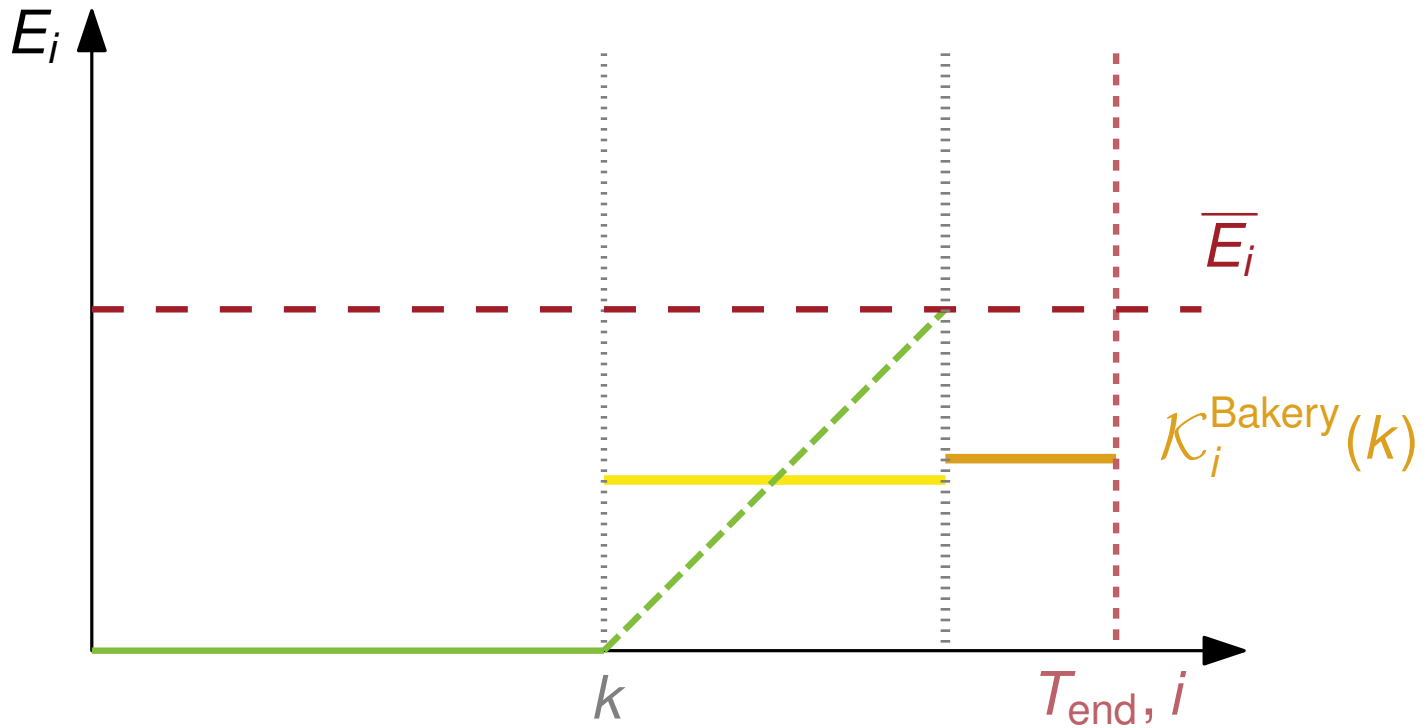
Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Agilitätsfaktoren:

Bakery$_i(k)$:

$$\mathcal{K}_i^{\text{Bakery}}(k) = T_{\text{end},i} - T_{\text{run},i} - k$$



Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Forced consumption:

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Forced consumption:

$\text{Battery}_i(k)$:

$$P^{\text{Battery}}_{\text{Forced},i}(k) = \begin{cases} 0, & \mathcal{K}_i^{\text{Battery}} > 1 \\ \overline{P_i}(1 - \mathcal{K}_i^{\text{Battery}}), & 1 \geq \mathcal{K}_i^{\text{Battery}} > 0 \\ \overline{P_i}, & \mathcal{K}_i^{\text{Battery}} = 0 \end{cases}$$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Forced consumption:

$\text{Battery}_i(k)$:

$$P^{\text{Battery}}_{\text{Forced},i}(k) = \begin{cases} 0, & \mathcal{K}_i^{\text{Battery}} > 1 \\ \overline{P_i}(1 - \mathcal{K}_i^{\text{Battery}}), & 1 \geq \mathcal{K}_i^{\text{Battery}} > 0 \\ \overline{P_i}, & \mathcal{K}_i^{\text{Battery}} = 0 \end{cases}$$

$\text{Bakery}_i(k)$:

$$P^{\text{Bakery}}_{\text{Forced},i}(k) = \begin{cases} 0, & \mathcal{K}_i^{\text{Bakery}} > 1 \\ \overline{P_i}, & \mathcal{K}_i^{\text{Bakery}} = 0 \end{cases}$$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Energiereserve

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Energiereserve

Bucket$_i(k)$:

$$P^{\text{Bucket}}_{\text{Reserve},i}(k) = \min(\overline{P_i}, \frac{\overline{E_i} - E_i(k)}{T_s})$$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

Energiereserve

Bucket$_i(k)$:

$$P^{\text{Bucket}}_{\text{Reserve},i}(k) = \min(\overline{P_i}, \frac{\overline{E_i} - E_i(k)}{T_s})$$

Bei einem Portfolio mit N^{Buckets} bei Zeitschritt k:

$$P^{\text{Bucket}}_{\text{Reserve}}(k) = \sum_{i=1}^{N^{\text{Buckets}}} \min(\overline{P_i}, \frac{\overline{E_i} - E_i(k)}{T_s})$$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing

for $k = 0$ **to** K **do**:

- Berechne $P_{\text{Forced}}(k)$
- **if** $P_{\text{Forced}}(k) > P_{\text{Dispatch}}(k)$**:**
  - Starte alle Forced Units
- **else:**
  - Sortiere Bakeries und Batteries nach aufsteigendem Agilitätsfaktor
  - Starte Units solange der Verbrauch kleiner gleich $P_{\text{Dispatch}}(k)$
- Bestimme ob Leistung übrig ist oder noch benötigt wird
- Buckets dementsprechend nach absteigendem Agilitätsfaktor starten

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel

Buckets

Bucket_1: $\overline{P_1} = 1$, $\underline{P_1} = -1$

$\overline{E_1} = 2$, $\underline{E_1} = 0$

Bucket_2: $\overline{P_2} = 2$, $\underline{P_2} = -2$

$\overline{E_2} = 3$, $\underline{E_2} = -1$

Batteries

Battery_1: $\overline{P_3} = 2$, $\overline{E_3} = 4$

$T_{\text{end},3} = 5$

Battery_2: $\overline{P_4} = 1$, $\overline{E_4} = 3$

$T_{\text{end},4} = 4$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel

Bakeries:

Bakery_1: $\overline{P_5} = 1, \overline{E_5} = 1$

$T_{\text{run},5} = 1, T_{\text{end},5} = 3$

Bakery_2: $\overline{P_6} = 3, \overline{E_6} = 3$

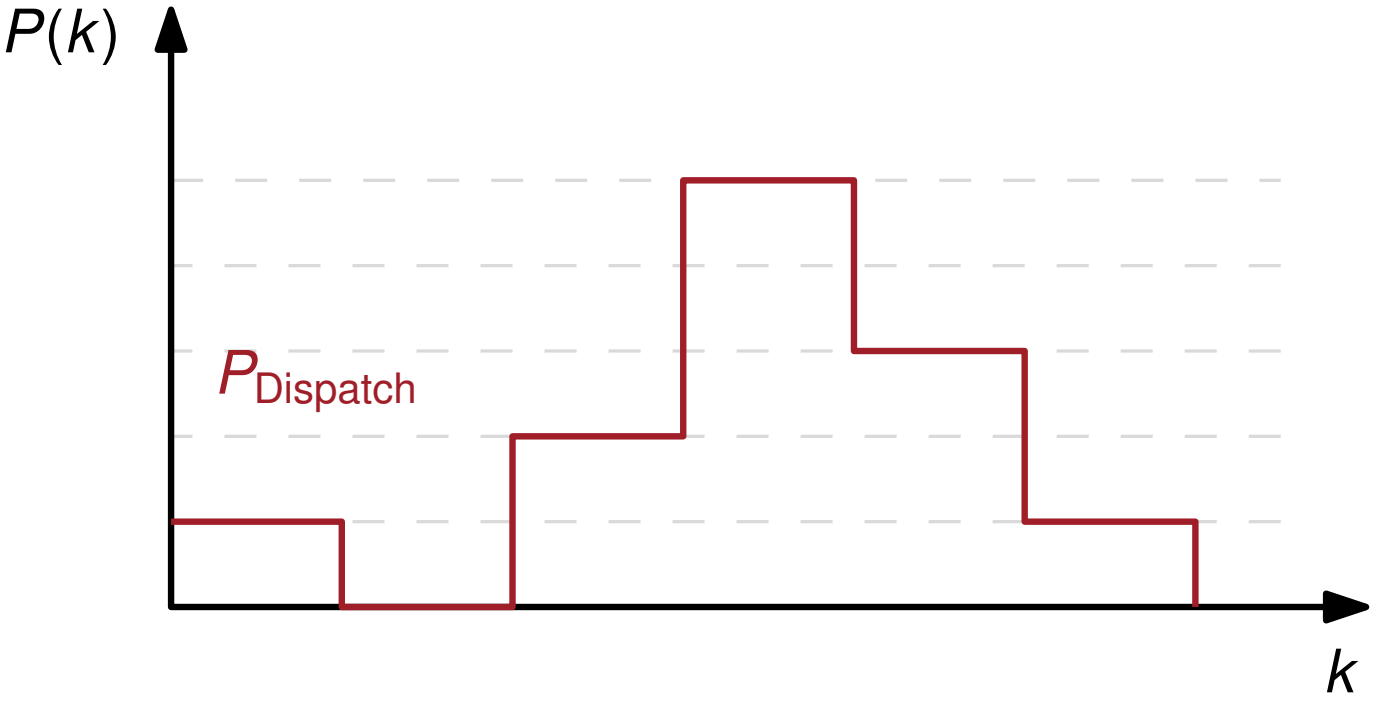
$T_{\text{run},6} = 1, T_{\text{end},6} = 6$

Dispatch-Profil:

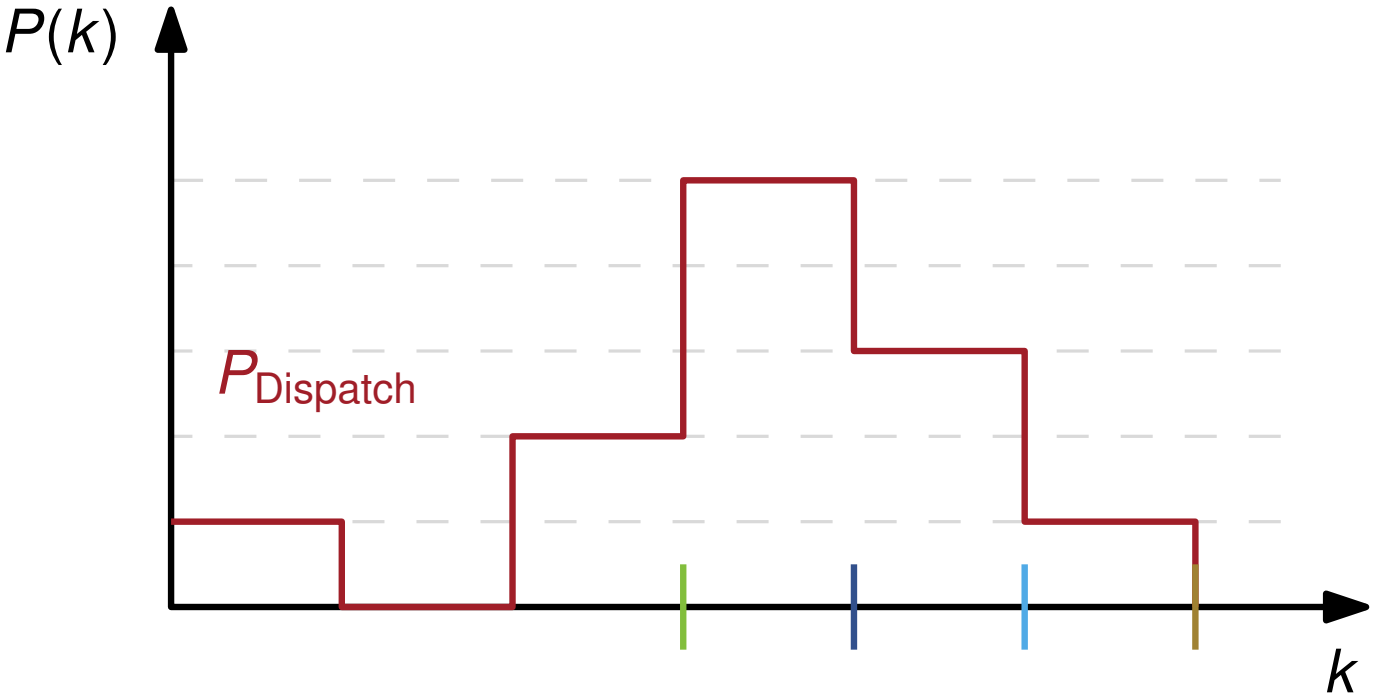
$P_{\text{Dispatch}} = (1, 0, 2, 5, 3, 1)$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Algorithmus: Agile Balancing Beispiel



Battery$_1$

Bakery$_1$

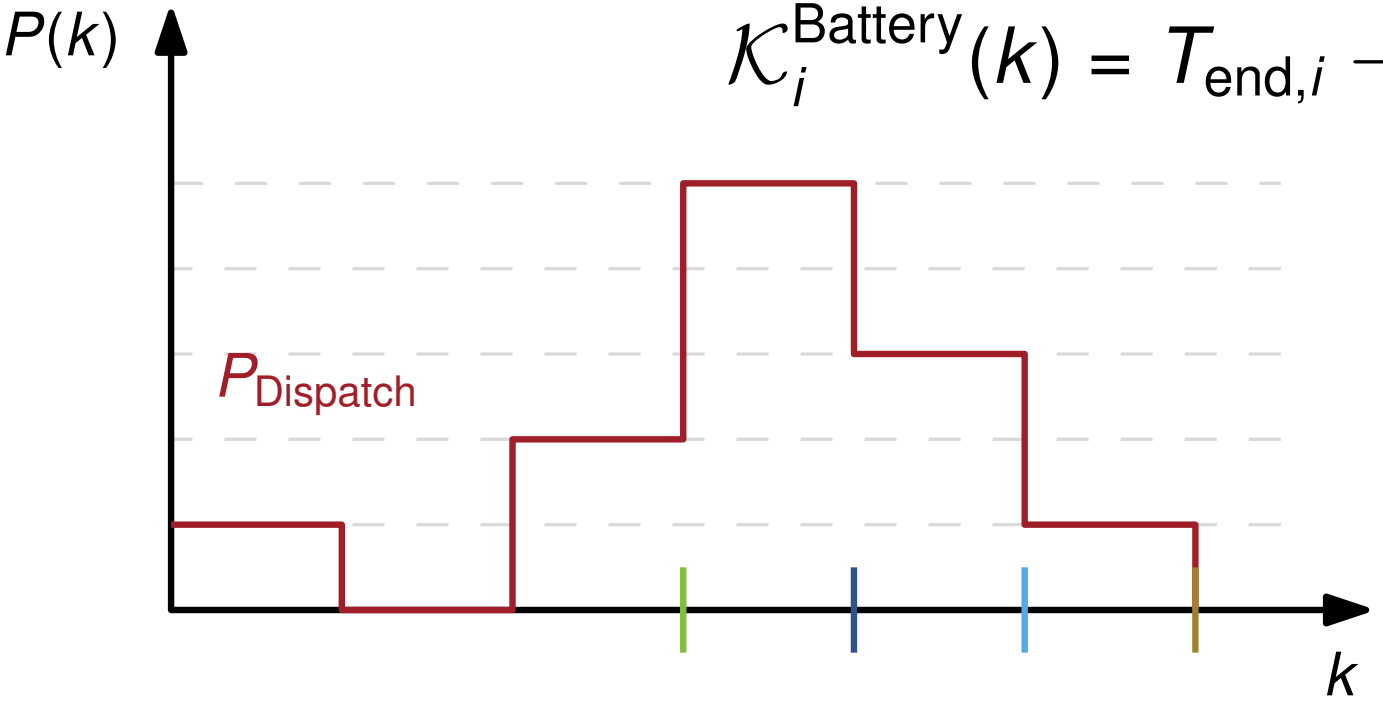
Battery$_2$

Bakery$_2$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel

$$\mathcal{K}_i^{\text{Battery}}(k) = T_{\text{end},i} - k - \frac{\overline{E_i} - E_i(k)}{T_s \overline{P_i}}$$

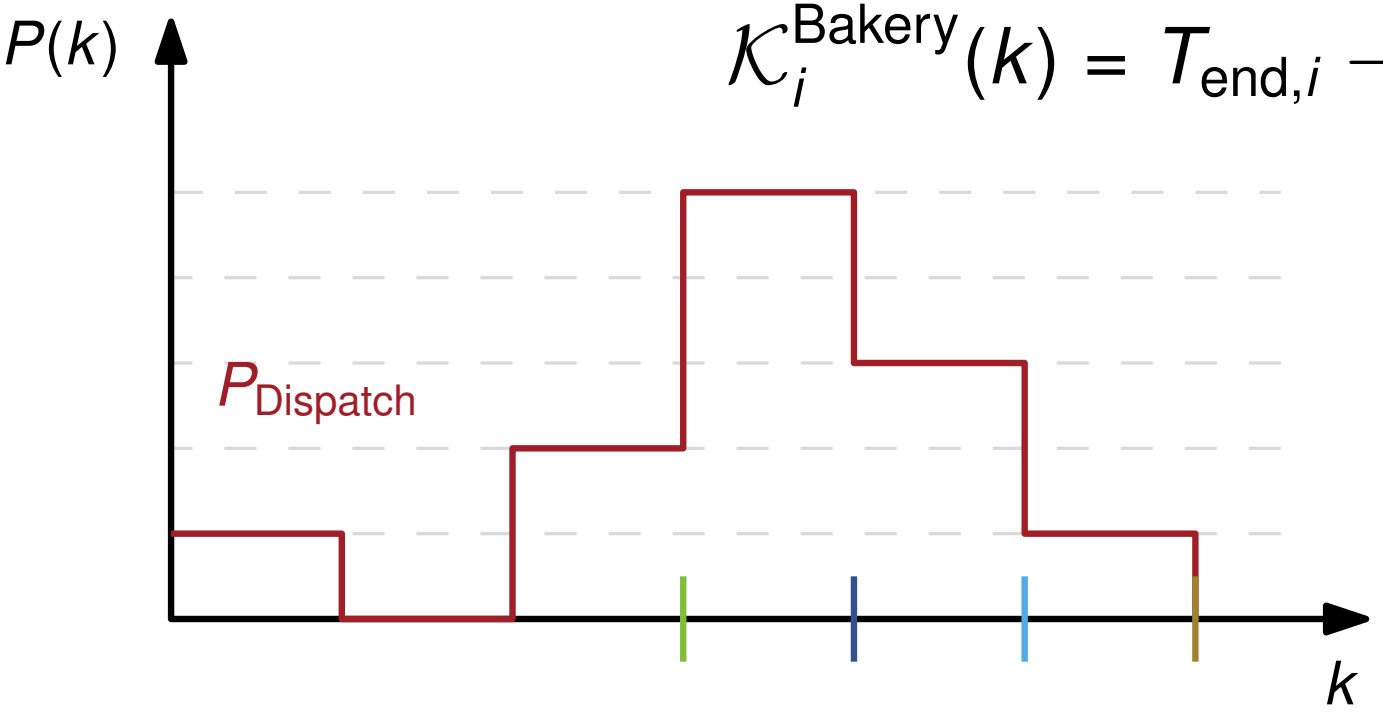


Battery$_1$ $\mathcal{K}_3(0) = 3$ Bakery$_1$

Battery$_2$ $\mathcal{K}_4(0) = 1$ Bakery$_2$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



$$\mathcal{K}_i^{\text{Bakery}}(k) = T_{\text{end},i} - T_{\text{run},i} - k$$

Battery$_1$ $\mathcal{K}_3(0) = 3$

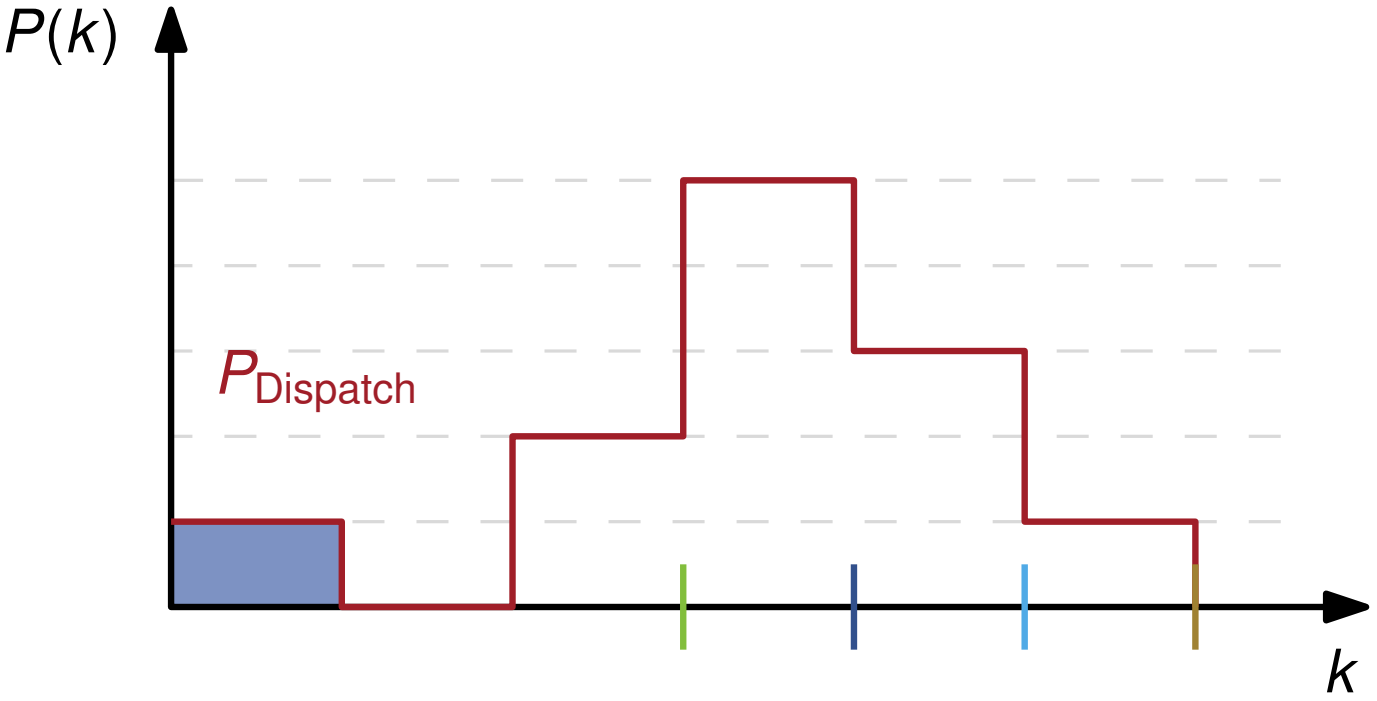
Bakery$_1$ $\mathcal{K}_5(0) = 2$

Battery$_2$ $\mathcal{K}_4(0) = 1$

Bakery$_2$ $\mathcal{K}_6(0) = 5$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Battery$_1$ $\mathcal{K}_3(0) = 3$

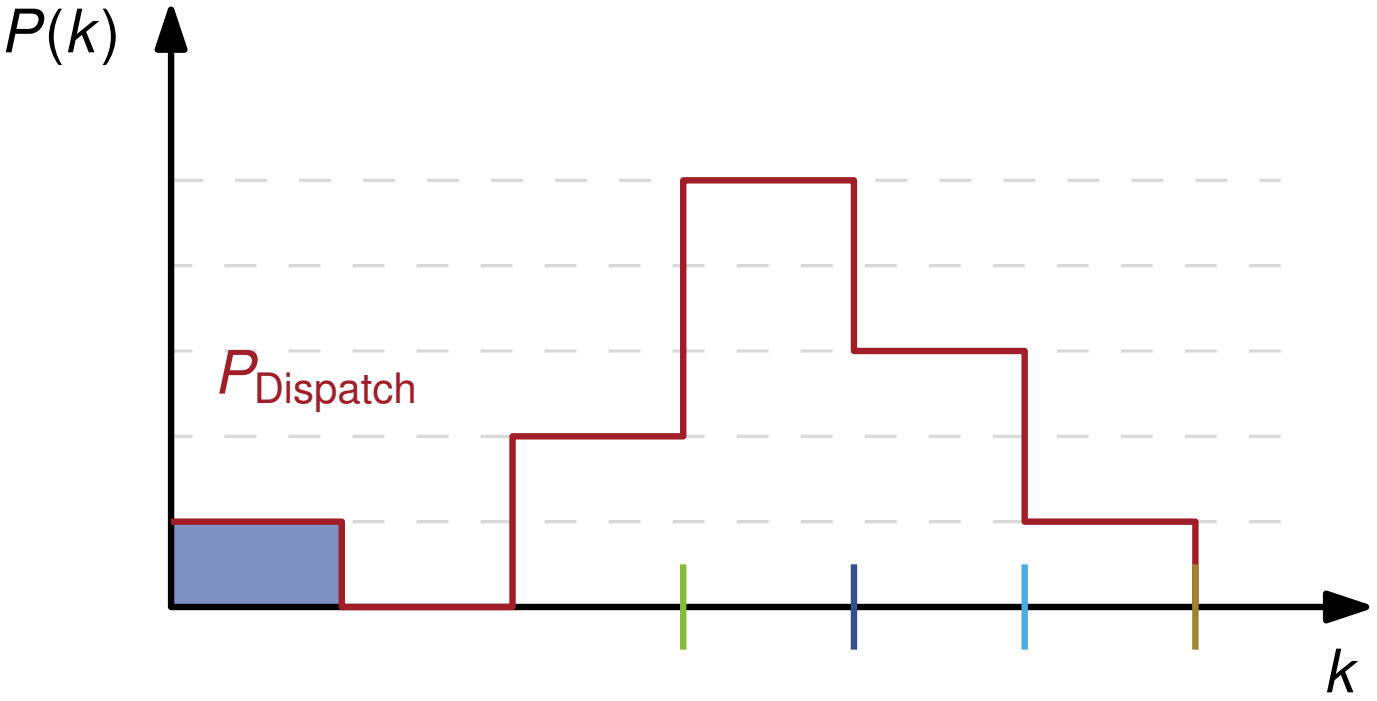
Bakery$_1$ $\mathcal{K}_5(0) = 2$

Battery$_2$ $\mathcal{K}_4(0) = 1$

Bakery$_2$ $\mathcal{K}_6(0) = 5$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Battery$_1$ $\mathcal{K}_3(1) = 2$

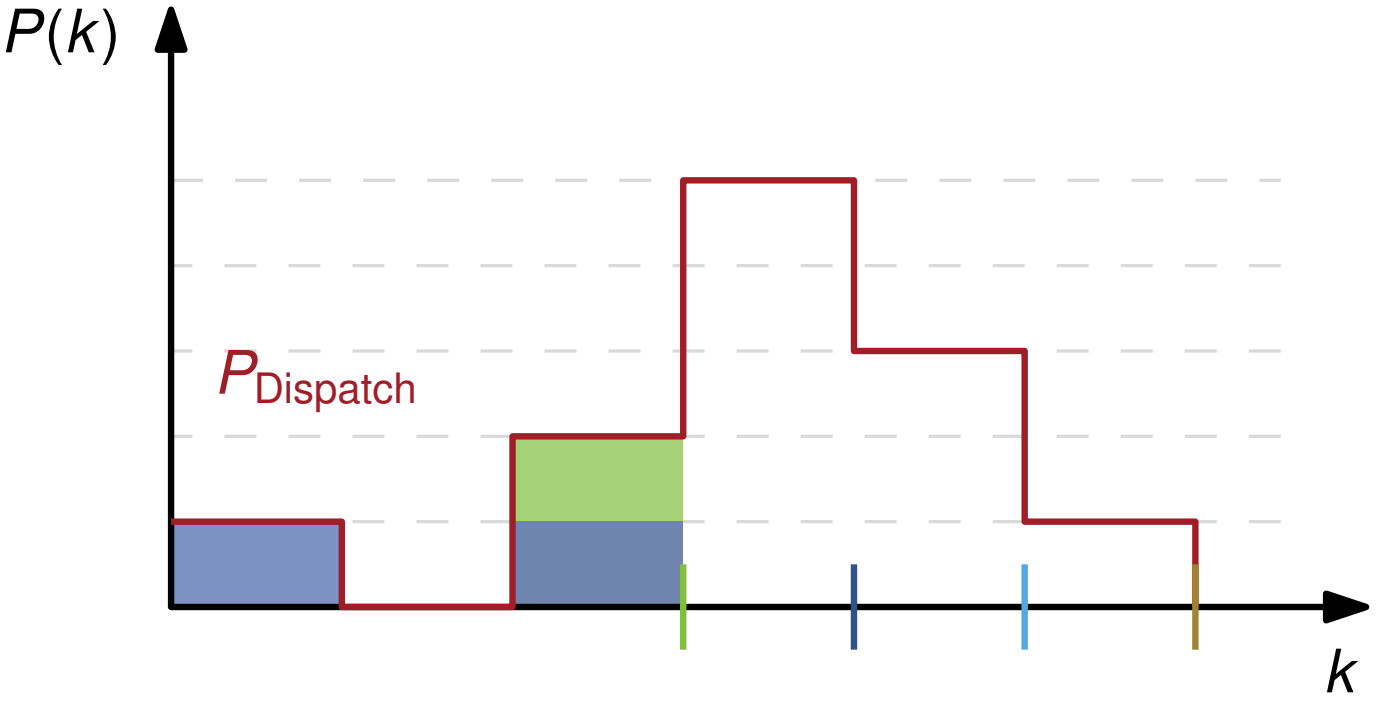
Bakery$_1$ $\mathcal{K}_5(1) = 1$

Battery$_2$ $\mathcal{K}_4(1) = 1$

Bakery$_2$ $\mathcal{K}_6(1) = 4$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Battery$_1$ $\mathcal{K}_3(2) = 1$

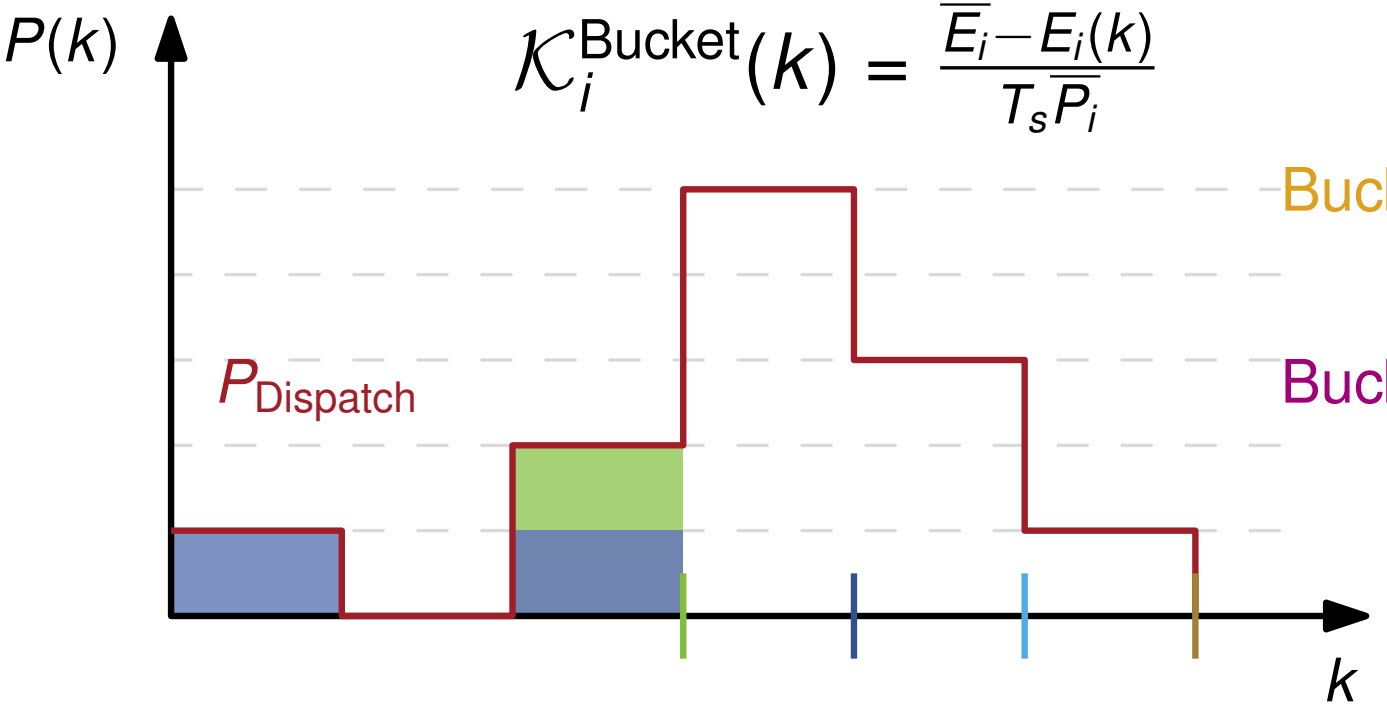
Bakery$_1$ $\mathcal{K}_5(2) = 0$ ✓

Battery$_2$ $\mathcal{K}_4(2) = 0$

Bakery$_2$ $\mathcal{K}_6(2) = 3$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



$P_{\text{Reserve},1}(k) = 1$

$\mathcal{K}_1(3) = 2$

$P_{\text{Reserve},2}(k) = 2$

$\mathcal{K}_2(3) = 1.5$

Battery$_1$ $\mathcal{K}_3(3) = 0$

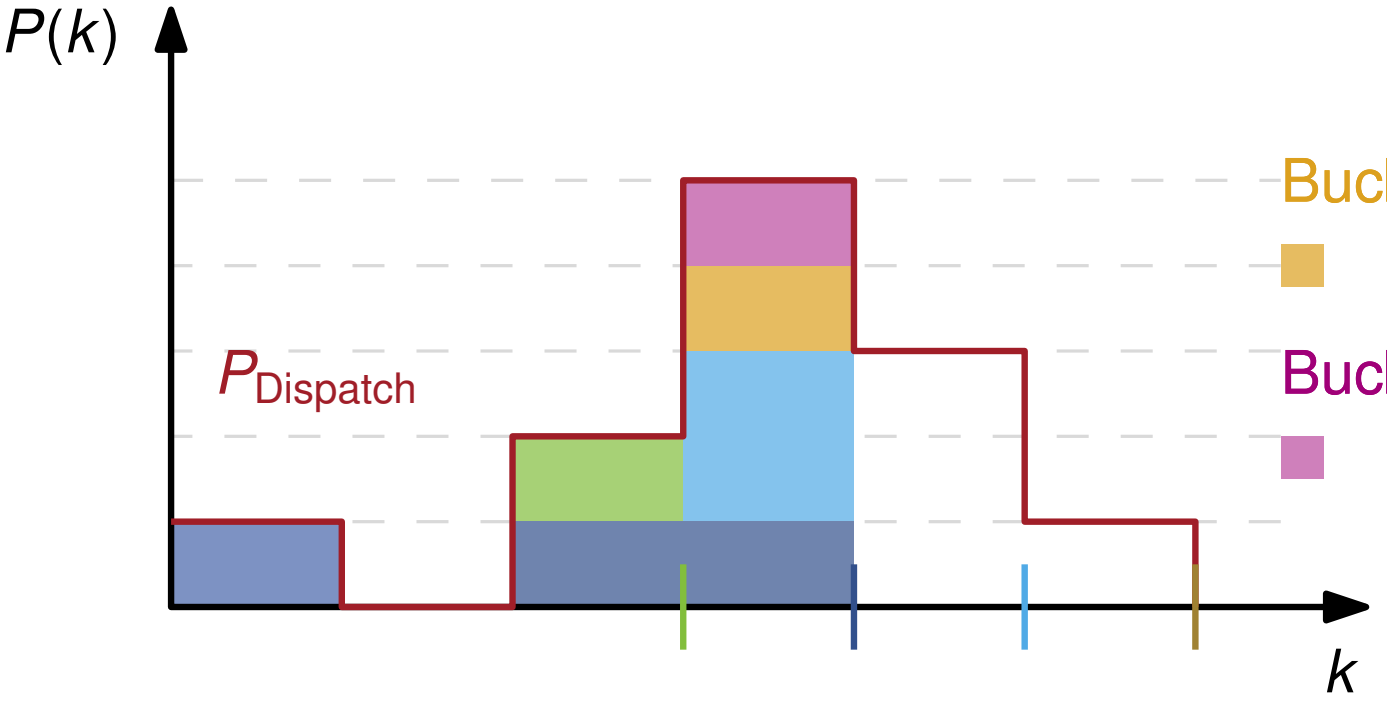
Bakery$_1$ ✓

Battery$_2$ $\mathcal{K}_4(3) = 0$

Bakery$_2$ $\mathcal{K}_6(3) = 2$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Bucket$_1$ $P_{\text{Reserve},1}(k) = 1$

$\mathcal{K}_1(3) = 2$

Bucket$_2$ $P_{\text{Reserve},2}(k) = 2$

$\mathcal{K}_2(3) = 1.5$

Battery$_1$ $\mathcal{K}_3(3) = 0$

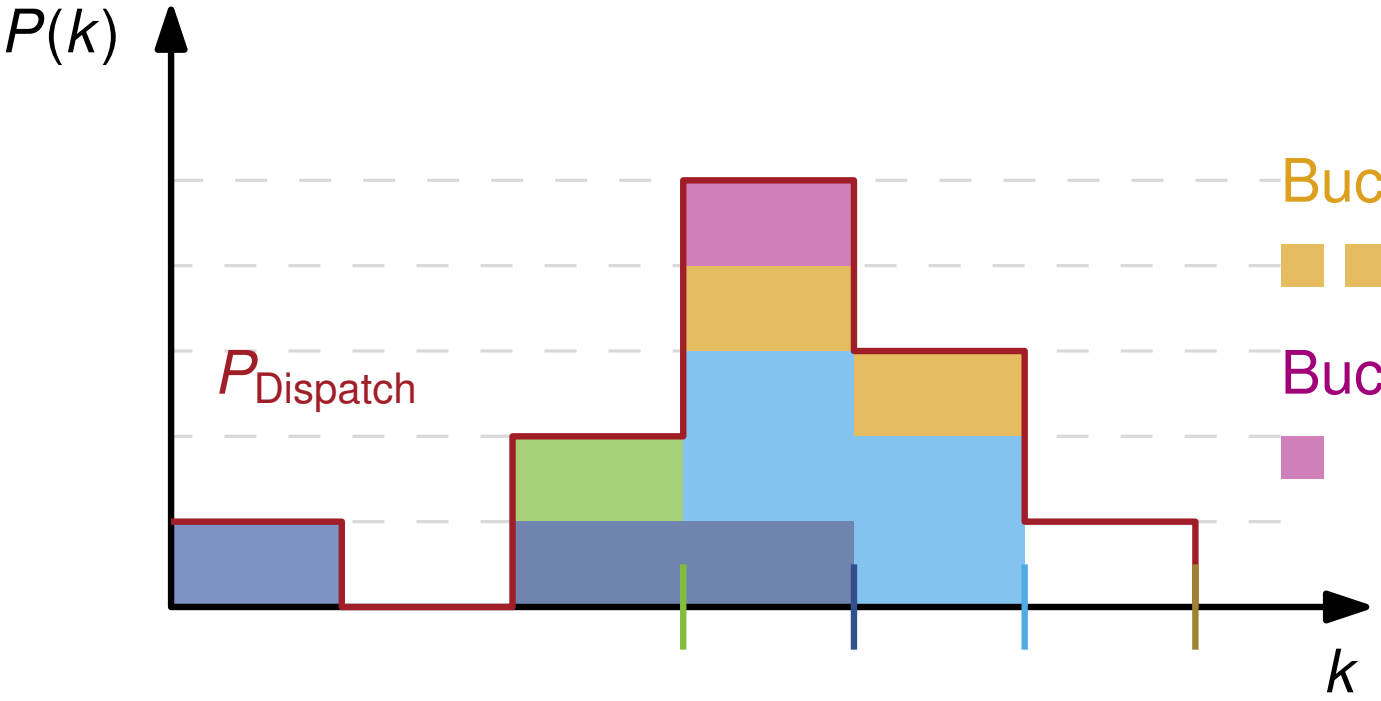
Bakery$_1$ ✓

Battery$_2$ $\mathcal{K}_4(3) = 0$ ✓

Bakery$_2$ $\mathcal{K}_6(3) = 2$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Bucket$_1$ $P_{\text{Reserve},1}(k) = 1$

$\mathcal{K}_1(4) = 1$

Bucket$_2$ $P_{\text{Reserve},2}(k) = 2$

$\mathcal{K}_2(4) = 1$

Battery$_1$ $\mathcal{K}_3(4) = 0$ ✓

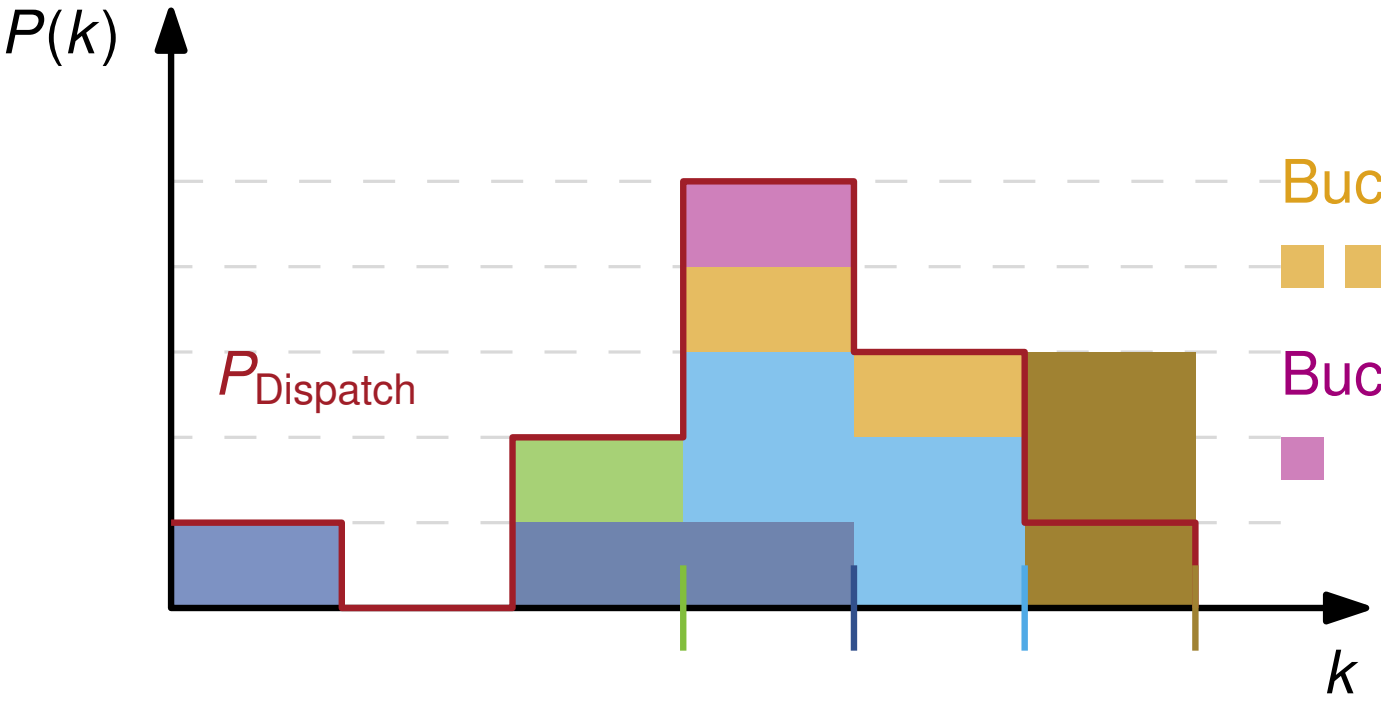
Battery$_2$ ✓

Bakery$_1$ ✓

Bakery$_2$ $\mathcal{K}_6(4) = 1$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Bucket$_1$ $P_{\text{Reserve},1}(k) = 0$

$\mathcal{K}_2(5) = 0$

Bucket$_2$ $P_{\text{Reserve},2}(k) = 2$

$\mathcal{K}_2(5) = 1$

Battery$_1$ ✓

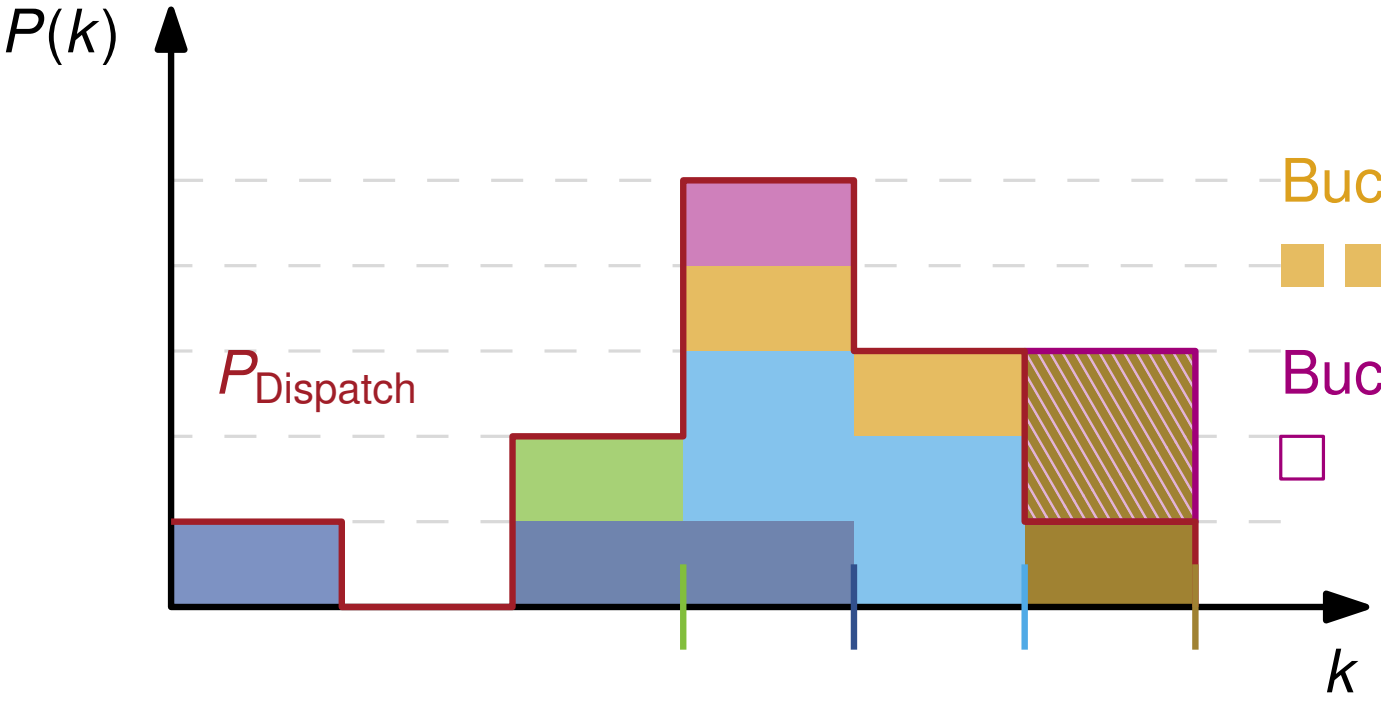
Battery$_2$ ✓

Bakery$_1$ ✓

Bakery$_2$ $\mathcal{K}_6(5) = 0$

Institute of Theoretical Informatics
Algorithmics Group

Algorithmus: Agile Balancing Beispiel



Bucket$_1$ $P_{\text{Reserve},1}(k) = 0$

$\mathcal{K}_2(5) = 0$

Bucket$_2$ $P_{\text{Reserve},2}(k) = 2$

$\mathcal{K}_2(5) = 1$

Battery$_1$ ✓

Battery$_2$ ✓

Bakery$_1$ ✓

Bakery$_2$ $\mathcal{K}_6(5) = 0$ ✓

Institute of Theoretical Informatics
Algorithmics Group

Algoritmus: Agile Balancing

- getested mit 10^6 units
- Rechenzeit ca. 3 min 26 s
- Anteile der Äquivalenzklassen beeinflussen Ergebnis

Institute of Theoretical Informatics
Algorithmics Group

Algoritmus: Agile Balancing

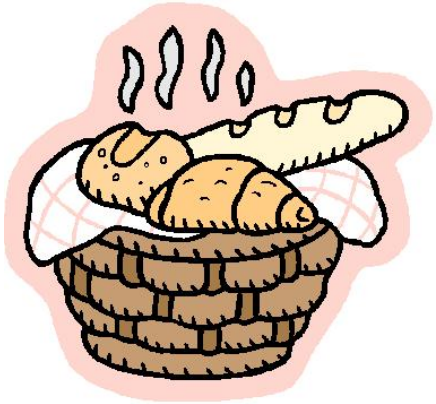
- getested mit 10^6 units
- Rechenzeit ca. 3 min 26 s
- Anteile der Äquivalenzklassen beeinflussen Ergebnis

- Beschleunigung möglich durch vorberechnete Agilitätstabelle
- Bei gleicher Unit Anzahl Rechenzeit ca 1 min 4 s
- Führt bei ungünstiger Aufteilung der Units zu schlechteren Ergebnissen

Institute of Theoretical Informatics
Algorithmics Group

Fazit

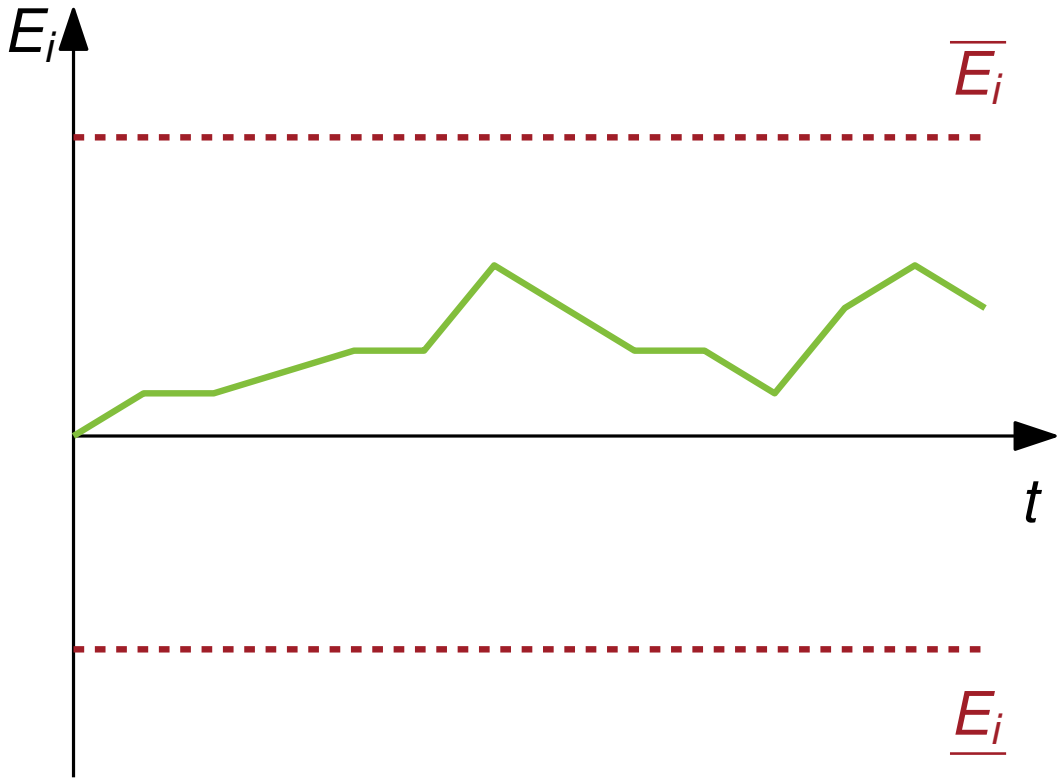
- Nachfrageflexibilität lässt sich in verschiedene Klassen unterteilen



Institute of Theoretical Informatics
Algorithmics Group

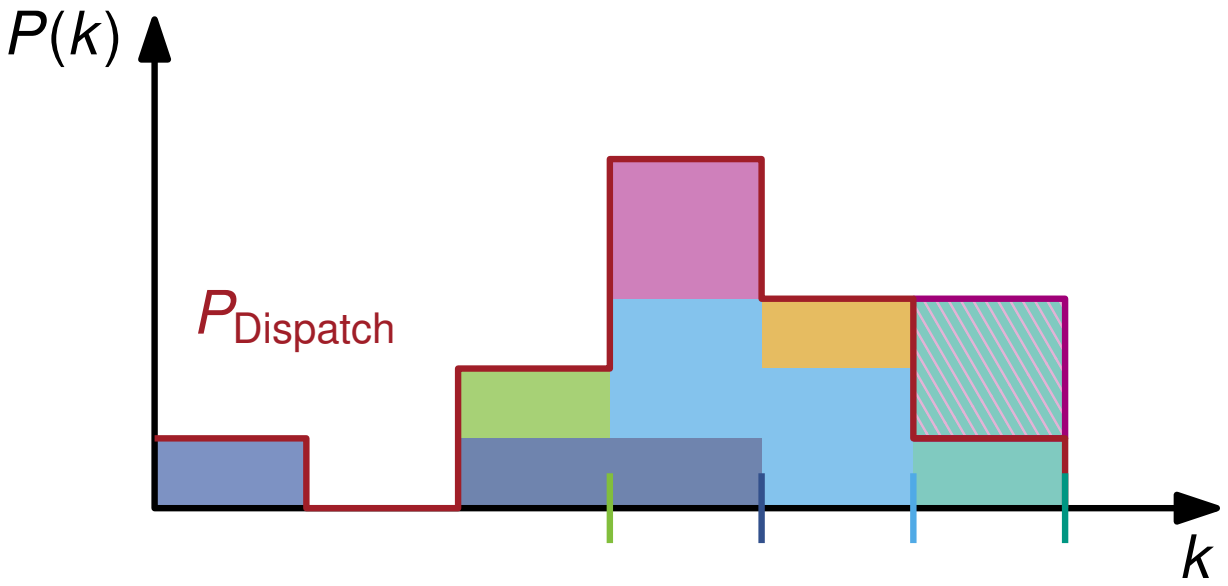
Fazit

- Nachfrageflexibilität lässt sich in verschiedene Klassen unterteilen
- Mathematische Modellierung der Klassen zu Verwendung in Algorithmen



Fazit

- Nachfrageflexibilität lässt sich in verschiedene Klassen unterteilen
- Mathematische Modellierung der Klassen zu Verwendung in Algorithmen
- Heuristische Algorithmen ermöglichen es das Problem schnell zu lösen



Institute of Theoretical Informatics
Algorithmics Group

Fazit

- Nachfrageflexibilität lässt sich in verschiedene Klassen unterteilen
- Mathematische Modellierung der Klassen zu Verwendung in Algorithmen
- Heuristische Algorithmen ermöglichen es das Problem schnell zu lösen
- Die Lösungen der Algorithmen sind nur geringfügig schlechter als die optimale Lösung

Institute of Theoretical Informatics
Algorithmics Group

Quellen

[1] http://www.blog.ze.com/2014/05/big-data-growth-for-utilities-the-smart-grid/

[2] http://cliparting.com/wp-content/uploads/2017/04/Wind-turbine-clip-art.png

[3] https://img.deutsche-handwerks-zeitung.de/files/smthumbnaildata/392x/2/3/5/7/6/7/NeuesBild.jpg

[4] https://image.freepik.com/free-icon/battery-status_318-47374.jpg

[5] http://baeckerei-walzer.de/images/korbmitbrot.gif

[6] Petersen, Mette K., et al. ”Heuristic optimization for the discrete virtual power plant dispatch problem.” IEEE Transactions on Smart Grid 5.6 (2014): 2910-2918.