Grundlagen der Echtzeitplanung

1. Grundlegende Begriffe und Konzepte

2. Planungsverfahren (Scheduling)

   2.1 Planen aperiodischer Tasks
   - Planen durch Suchen
   - Planen nach Fristen
   - Planen nach Spielräumen

   2.2 Planen periodischer Tasks
   - Planen nach monotonen Raten

Prozessorauslastung "U"

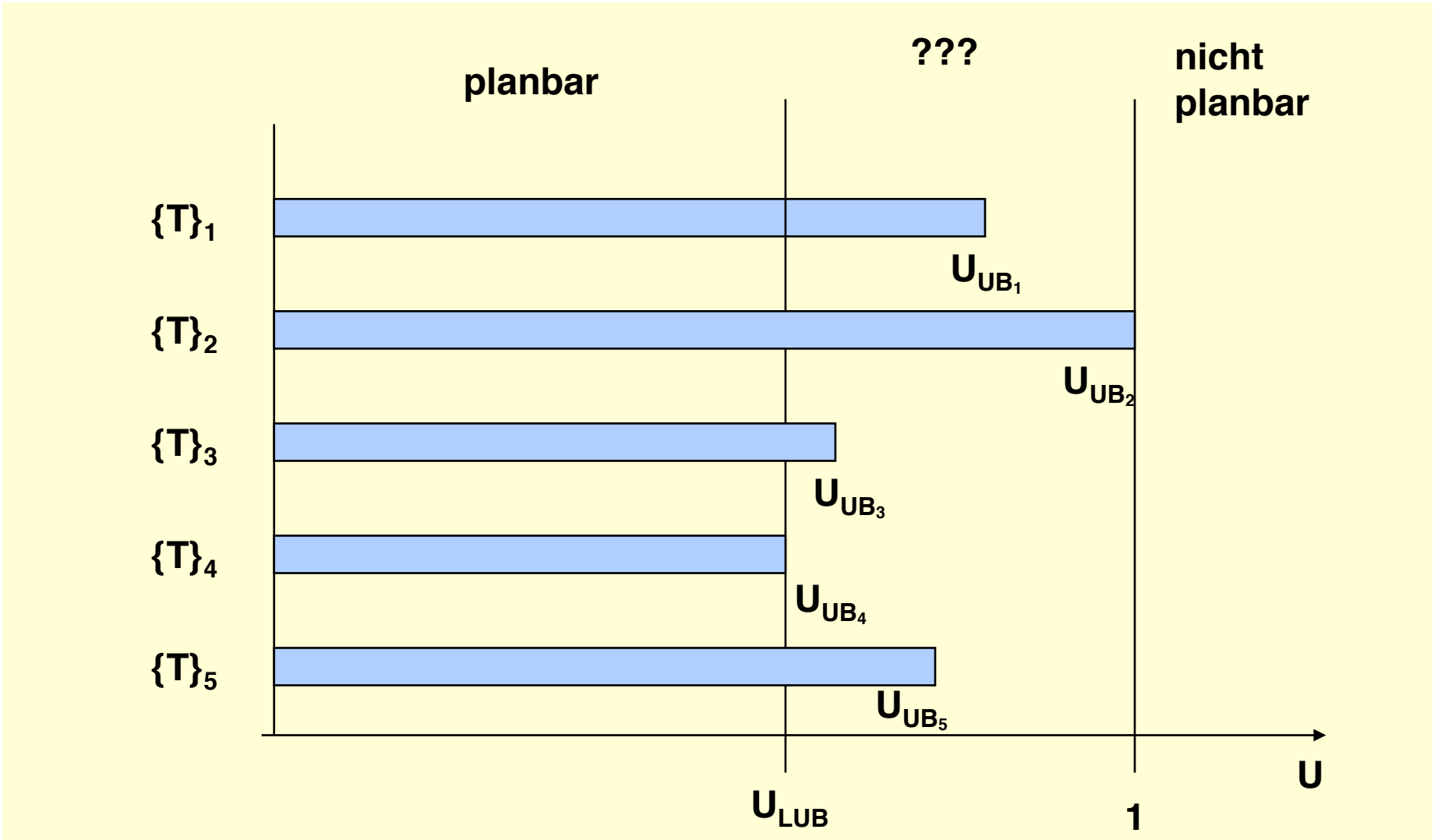
Gegeben sei die Menge periodischer Tasks T_1, T_2, .., T_n ,

$\Delta e_i / \Delta p_i$ ist die Zeit, die Task T_1 in der Periode Δp_i zur Ausführung nutzt.

Der Auslastungsfaktor für n Tasks ist dann gegeben durch:

$$U = \sum_{(i=1,...,n)} (\Delta e_i / \Delta p_i)$$

Sei U_{UB} ({T}, A) die obere Schranke des Auslastungsfaktors für die Menge der Tasks {T}, die von dem Schedulingalgorithmus A eingeplant wird.
Wenn $U = U_{UB}$ ({T},A), dann ist der Prozessor voll ausgelastet.



Die kleinste obere Schranke U_{LUB} (A) für die eine Taskmenge unter einem Schedulingverfahren planbar ist, ist gegeben durch:

$$U_{LUB}\,(A) = \min U_{UB}\,(\{T\}, A) \text{ über alle Taskmengen } \{T\}.$$

Satz: **Wenn der Auslastungsfaktor U >1 ist, kann eine Taskmenge von keinem Planungsalgorithmus eingeplant werden.**

Sei $\Delta P = \Delta p_1 \bullet \Delta p_2 \bullet \Delta p_3 \bullet ..... \bullet \Delta p_n$ (gemeinsames Vielfaches !)

Aus $U > 1$ folgt $U \cdot \Delta P > \Delta P$

$$\sum_{(i=1,...,n)} (\Delta P / \Delta p_i \cdot \Delta e_i) > \Delta P$$

mit $\Delta P / \Delta p_i$: Anzahl der Aktivierungen von T_i im Intervall ΔP
und $\Delta P / \Delta p_i \cdot \Delta e_i$:Ausführungszeit von T_i im Intervall ΔP

$\sum_{(i=1,...,n)} ((\Delta P / \Delta p_i) \cdot \Delta e_i)$ ist die Gesamtrechenzeit aller Tasks, die im Intervall ΔP benötigt wird. Diese kann nicht größer sein als die zur Verfügung stehende Gesamtzeit ΔP. Daher gilt, daß eine Menge von Tasks nur dann planbar ist, wenn $U \leq 1$ gilt.

Prioritätsbasiertes statisches Scheduling

Es wird kein expliziter Plan aufgestellt, der (zeitbasiert) auf Fristen oder Spielräumen beruht, sondern es existiert ein impliziter Plan, der durch eine Prioritätszuordnung repräsentiert wird.

Praxixbezug: Die meisten Echtzeitbetriebssystem(kern)e unterstützen prioritätsbasiertes Scheduling mit unterbrechbaren Prozessen.

Zur Entscheidung, welchem Prozeß dem Prozessor zugeteilt wird, braucht man lediglich die Priorität des gerade laufenden Prozesses mit der Priorität des gerade bereitwerdenden Prozesses zu vergleichen.

Nach welchen Gesichtspunkten wird einem Prozeß seine Priorität zugeordnet ?

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Annahmen:

1. **Eine Task kann zu einem beliebigen Zeitpunkt unterbrochen werden.**
2. **Es wird nur die Ressource "Prozessorzeit" betrachtet.**
3. **Alle Tasks sind unabhängig und es gibt keine Vorrangrelation unter Tasks.**
4. **Es werden ausschließlich unterbrechbare periodische Tasks betrachtet.**
5. **Die relativen Deadlines der Tasks entsprechen ihren Perioden.**

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Def.:
Rate einer periodischen Task = Anzahl der Perioden im Betrachtungszeitraum
= Frequenz (über unbegrenzte Zeitraum)

Prioritätsordnung:

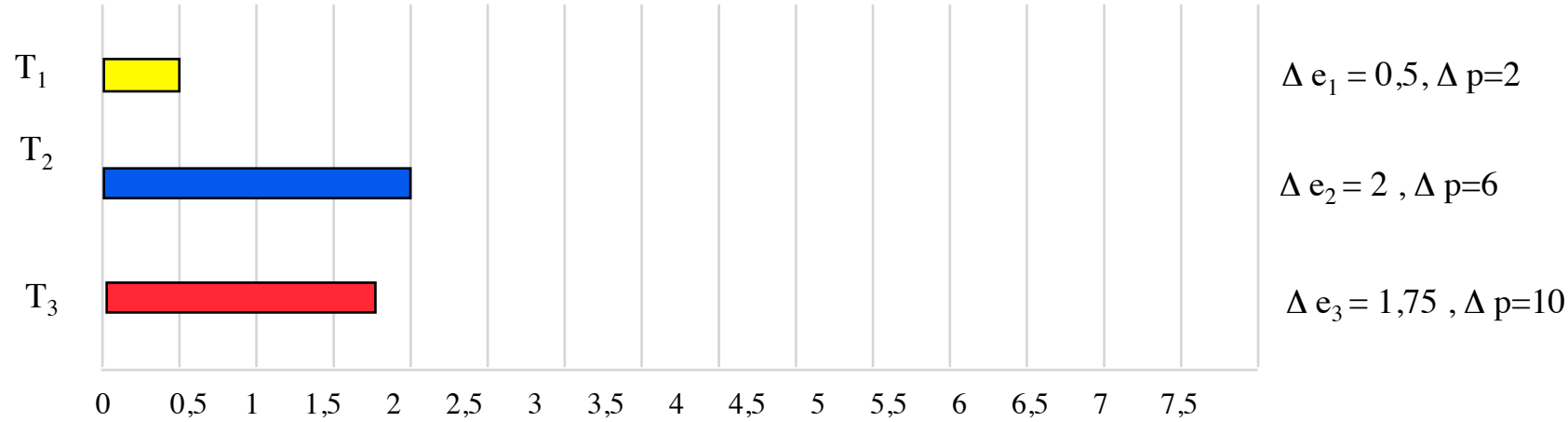
$$rms(i) < rms(j) \Leftrightarrow 1 / \Delta p_i < 1 / \Delta p_j$$

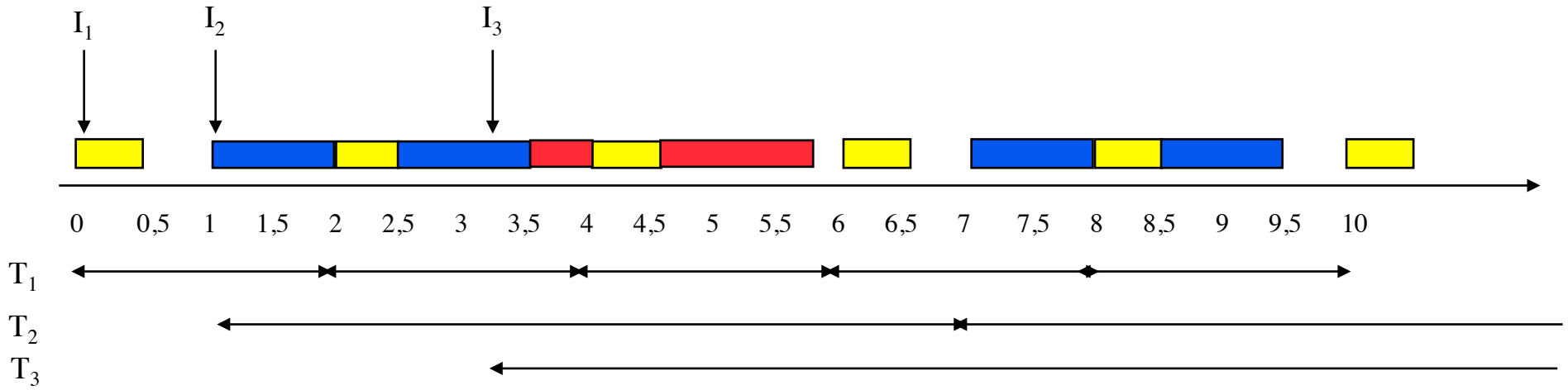
Nummerierung der Tasks gemäß ihrer Priorität:

$$i < j \Leftrightarrow rms(i) < rms(j)$$

Planen nach monotonen Raten : Rate Monotonic Scheduling (RMS)

Beispiel:





Eigenschaften von RMS :

Frage: Ist RMS optimal verglichen mit anderen statischen Planungsverfahren?

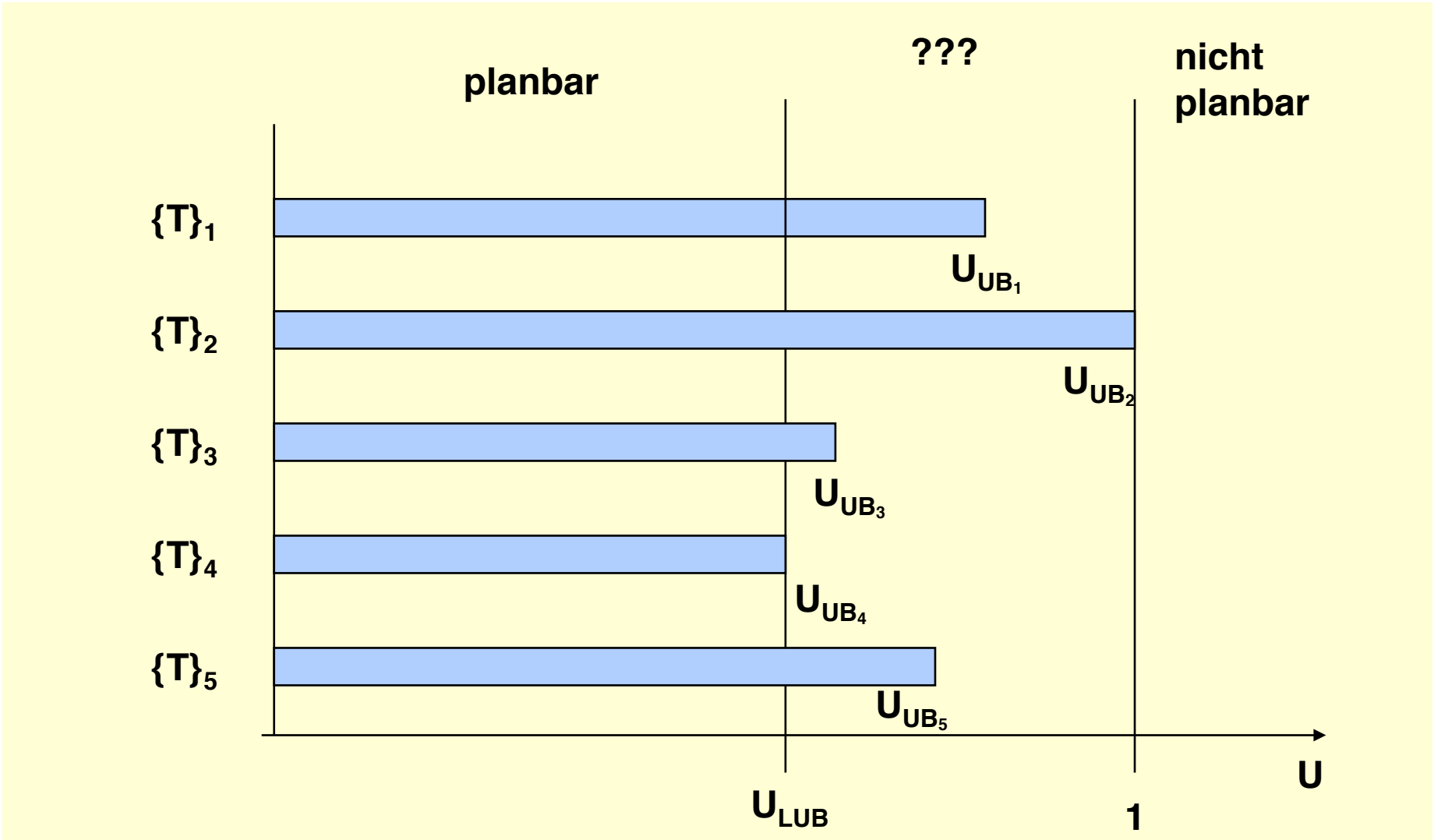
Intuitive Kritik: RMS berücksichtigt keine Deadlines, sondern zum aktuellen Zeitpunkt wird die Task mit der statisch höchsten Priorität bevorzugt.

Frage: Gibt es eine obere Schranke U_{lub} der Prozessorauslastung, für die immer ein Plan nach RMS garantiert werden kann (d.h. ein hinreichendes Kriterium für die Einplanbarkeit) ?

U_{lub} ist die Auslastung, für die RMS optimal ist, d.h. einen Plan findet, wenn überhaupt einer existiert. Es kann natürlich Verfahren geben, die eine bessere Auslastung realisieren.

Für n Tasks gilt: $U_{lub} = n\ (2^{1/n} - 1)$.

Für n = 1 : $U_{lub} = 1$
Für n = 2 : $U_{lub} = 0{,}828$
Für n → ∞ : $\lim U_{lub}(n) = \ln(2) = 0{,}693$



Die kleinste obere Schranke U_{LUB} (A) für die eine Taskmenge unter einem Schedulingverfahren planbar ist, ist gegeben durch:

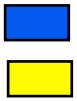
$$U_{LUB}\,(A) = \min U_{UB}\,(\{T\}, A) \text{ über alle Taskmengen } \{T\}.$$

Eigenschaften von RMS :

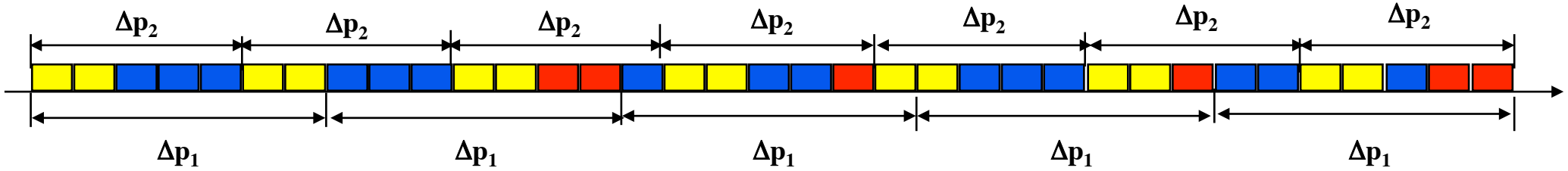
Beispiel für Scheduling nach RMS, in dem die Auslastung über der RMS-Grenze liegt, d.h. RMS keinen Plan findet, obwohl einer existiert.

1. Ausgangssituation: für diese Werte findet RMS einen Plan. Die Auslastung liegt mit 0,828 an der theoretischen Grenze

T_1 : $\Delta e_1 = 3$, $\Delta p_1 = 7$
T_2 : $\Delta e_2 = 2$, $\Delta p_2 = 5$



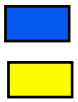
kgV = 35
T_1 : 15 Zeiteinheiten
T_2 : 14 Zeiteinheiten
$29/35 = 0{,}828 \approx n\,(\sqrt[n]{2} - 1)$



Eigenschaften von RMS :

Situation 2: Erhöhung des Rechenbedarfs von T_1 um 1 Einheit ($\Delta e_1 = 4$). RMS kann nicht mehr angewandt werden. Die theoretische Grenze der Auslastung wurde überschritten.

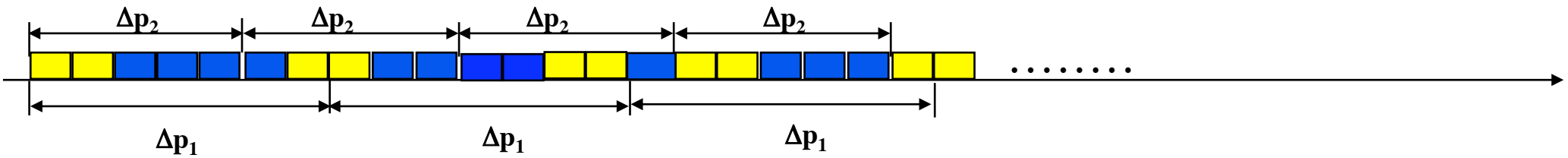
T_1 : $\Delta e_1 = 4$, $\Delta p_1 = 7$
T_2 : $\Delta e_2 = 2$, $\Delta p_2 = 5$



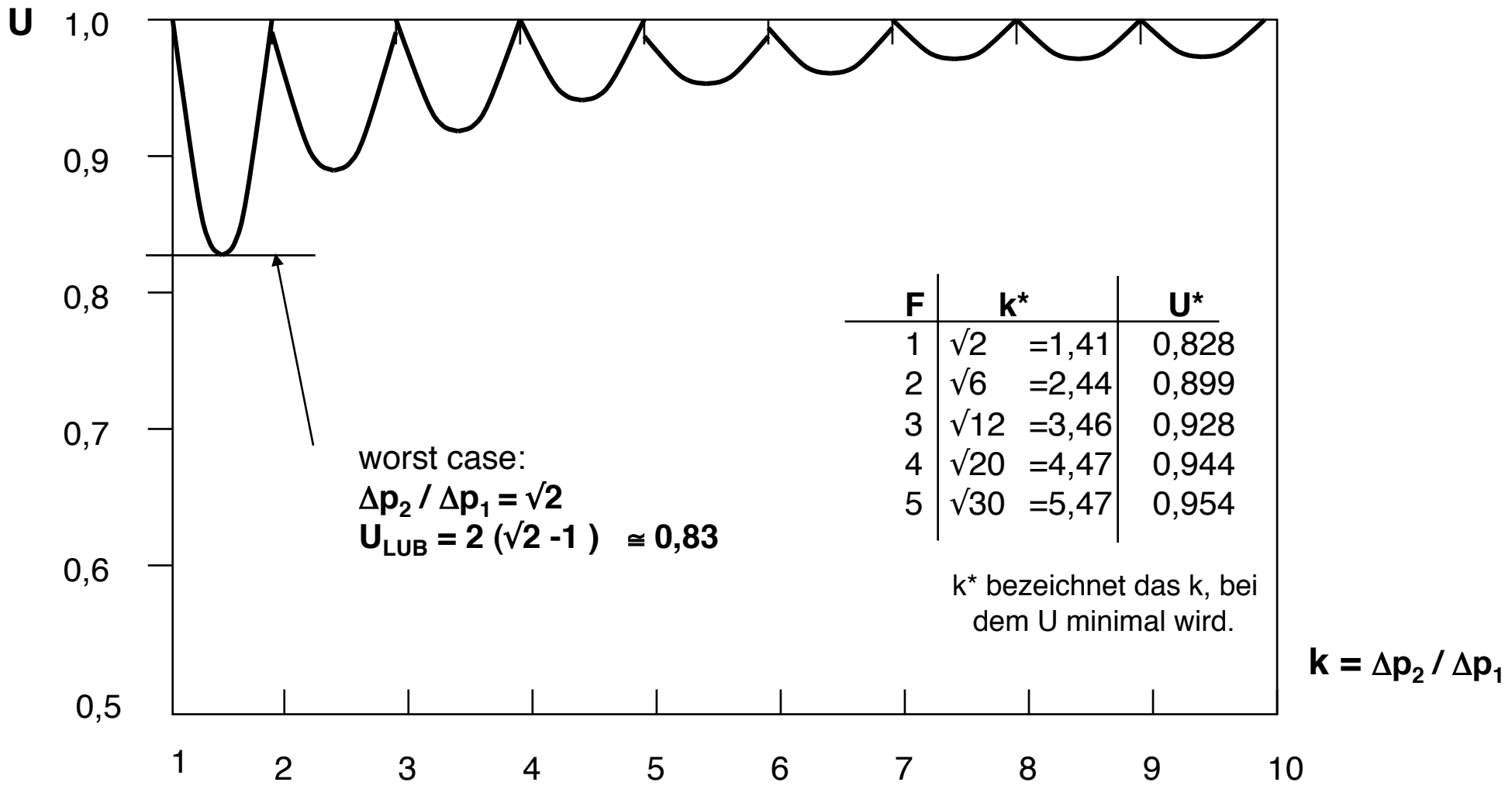
kgV = 35
T_1 : 20 Zeiteinheiten
T_2 : 14 Zeiteinheiten
$34/35 = 0{,}971 > n (\sqrt[n]{2} - 1) = 0{,}828$

T_1 **hat seine Deadline verletzt, da es bis zum Ende seine Intervalls noch keine 4 Zeiteinheiten zur Verfügung hatte.**

Für EDF ist das kein Problem, da das Verfahren den Prozessor bis 1 auslasten kann.



Obere Schranke der Prozessorauslastung in Abhängigkeit des Periodenverhältnisses "k" für 2 Tasks



F	k*	U*
1	√2 =1,41	0,828
2	√6 =2,44	0,899
3	√12 =3,46	0,928
4	√20 =4,47	0,944
5	√30 =5,47	0,954

Optimalität von RMS

Satz:

Sei T eine Taskmenge, für die eine gefundene (statische) Prioritätszuordnung bereits einen brauchbaren Plan liefert. Dann wird auch die Prioritätszuordnung nach monotonen Raten einen brauchbaren Plan liefern.

C.L. Liu, J.W. Layland
Scheduling algorithms for multiprogramming in a hard- real-time environment.
Journal of the ACM 20(1), January 73, pp.46-61