

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
- Deadline Monotonic
- EDF



Folgende Annahmen gelten:

- A1:** Die Instanzen einer Task T_i werden mit einer konstanten Rate in regelmäßigen Abständen aktiviert. Das Intervall zwischen zwei Aktivierungen wird als *Periode* p_i bezeichnet.
- A2:** Alle Instanzen einer periodischen Task T_i haben dieselbe WCET Δe_i
- A3:** Alle Instanzen einer periodischen Task T_i haben dieselbe relative Deadline D_i , die der Periode von T_i entspricht.
- A4:** Alle Tasks sind unabhängig.



Reaktionszeit: Zeit von der Aktivierung bis zum Abschluß

$$R_{i,k} = c_{i,k} - r_{i,k}$$

Kritische Instanz: Der Zeitpunkt an dem die Aktivierung einer Task die größte Reaktionszeit bedingt

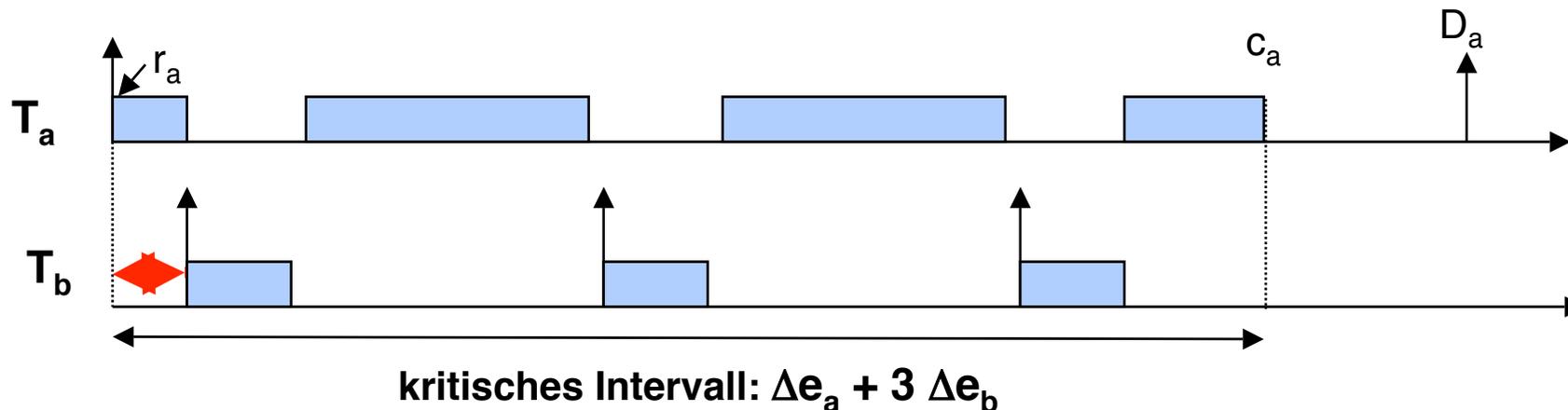
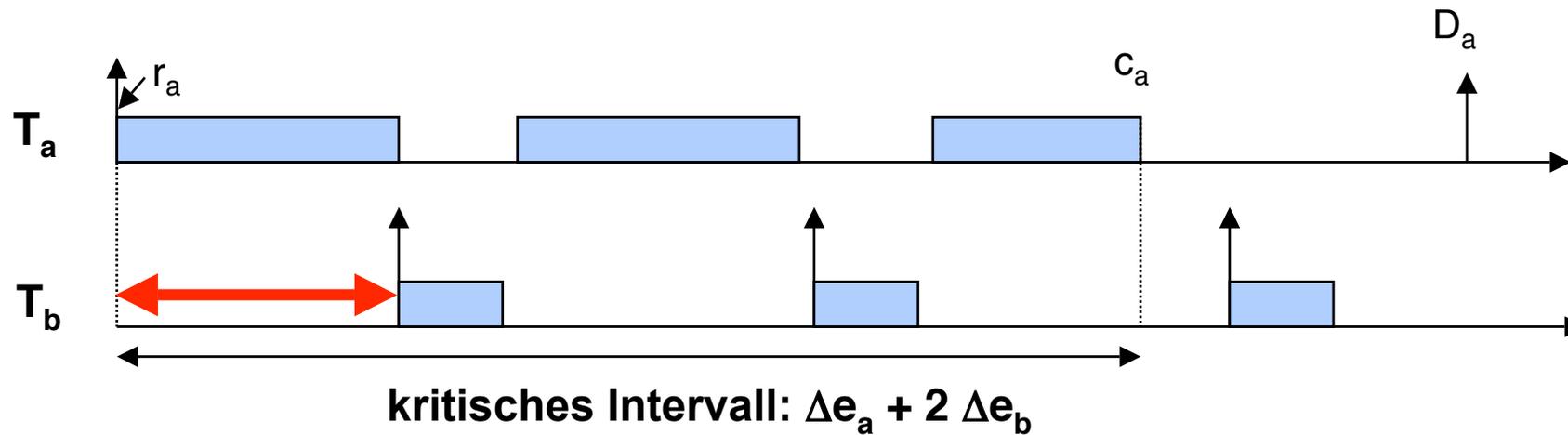
Kritisches Intervall: Das Intervall zwischen der kritische Instanz und der Abschlußzeit der entsprechenden Task

 Eine Task ist planbar, wenn alle ihre Instanzen vor der Deadline fertig werden.

 Eine Taskmenge heißt planbar, wenn alle Tasks planbar sind.



Kritische Instanz und kritisches Intervall



Die Antwortzeit von T_a wird durch die höher priorisierte Task T_b verlängert und zwar je mehr, je häufiger T_b während der Ausführungszeit von T_a aktiviert wird.

Das Maximum der Aktivierungen wird erreicht, wenn T_a und T_b zum selben Zeitpunkt aktiviert werden.

Prozessorauslastung “U”

Gegeben sei die Menge periodischer Tasks T_1, T_2, \dots, T_n ,

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

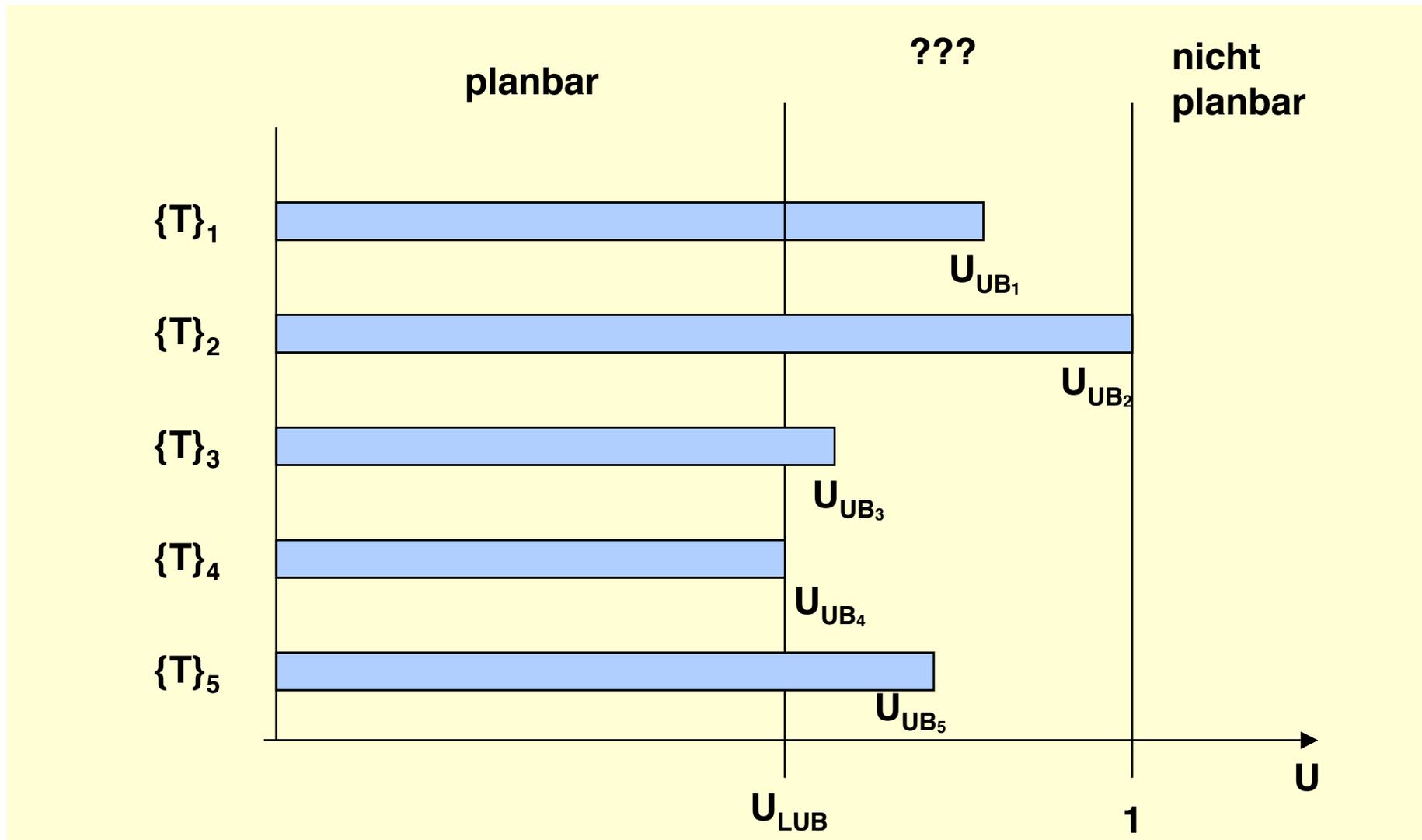
Der Auslastungsfaktor für n Tasks ist dann gegeben durch:

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

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 $\Delta P = \Delta p_1 \cdot \Delta p_2 \cdot \Delta p_3 \cdot \dots \cdot \Delta p_n$ (gemeinsames Vielfaches !)

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

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

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

Daraus folgt, daß $\sum_{(i=1,\dots,n)} ((\Delta P / \Delta p_i) \cdot \Delta e_i)$ die Gesamtrechenzeit aller Tasks ist, 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.

Praxisbezug: 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 bereit werdenden 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:

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

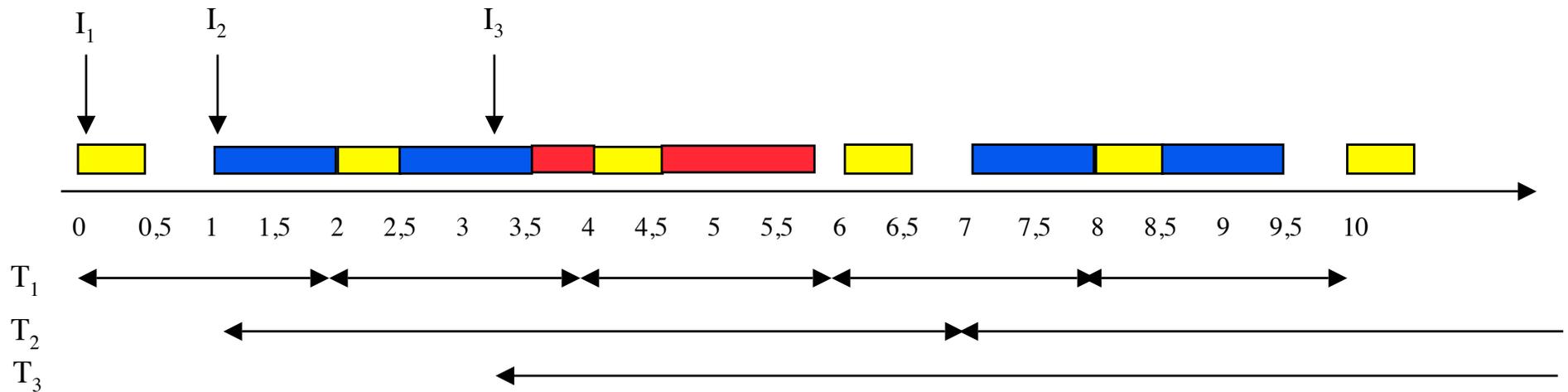
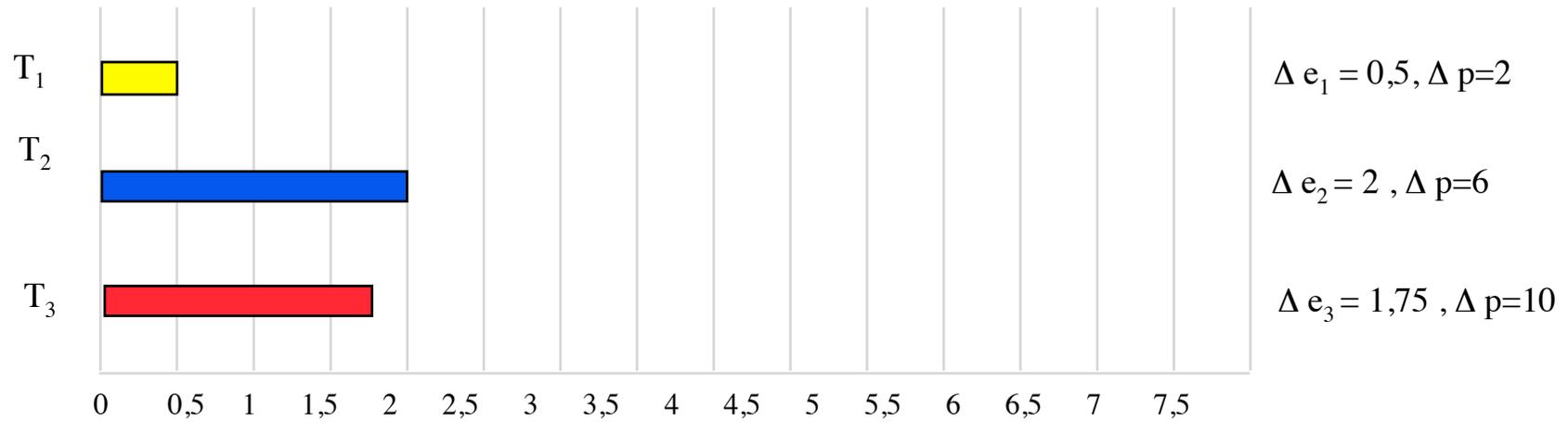
Nummerierung der Tasks gemäß ihrer Priorität:

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



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.

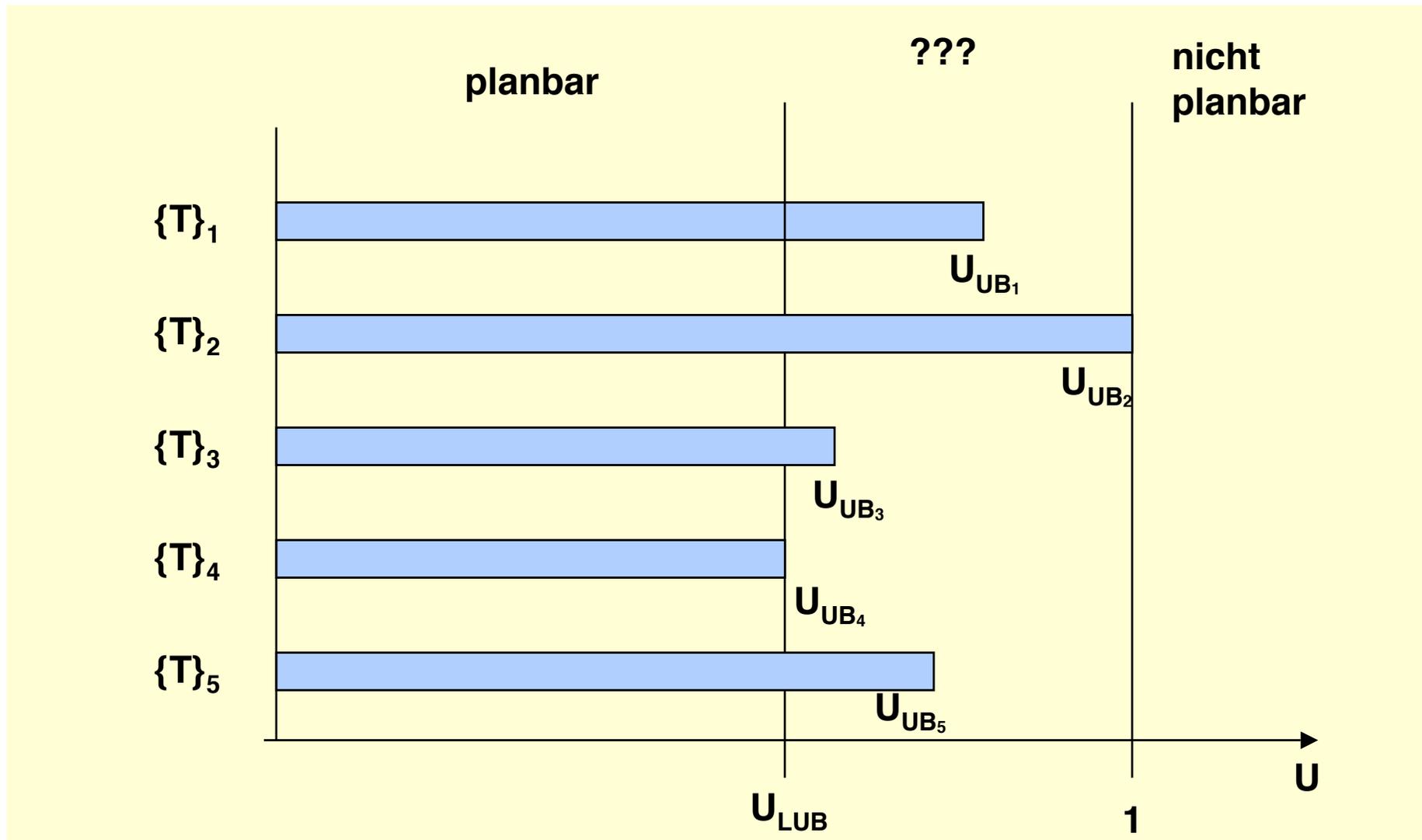
Frage: Gibt es eine obere Schranke U_{lub} der Prozessorauslastung, für die immer ein Plan nach RMS garantiert werden kann (d.h. ein hinreichendes Kriterium für die Einplanbarkeit) ?

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 $\rightarrow \infty$:	$\lim U_{lub} (n) = \ln (2)$	= 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 :

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 : $\Delta e_2 = 2$, $\Delta p_2 = 5$

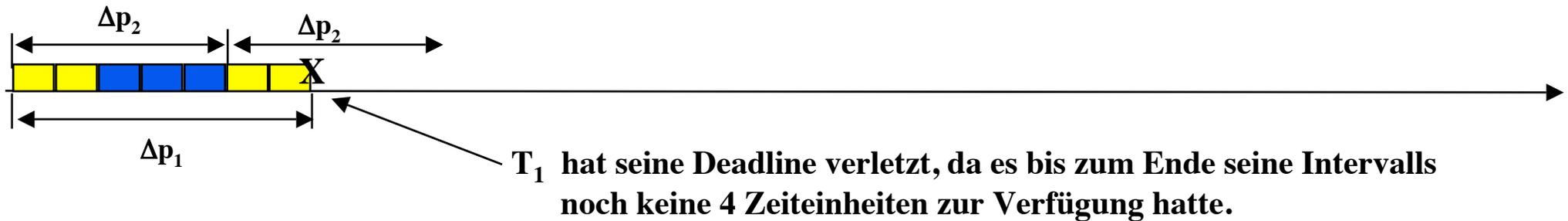


kgV = 35

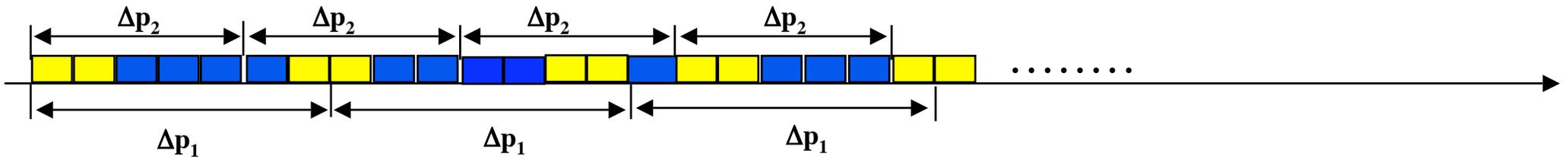
T_1 : 20 Zeiteinheiten

T_2 : 14 Zeiteinheiten

$34/35 = 0,971 > n (\sqrt[n]{2} - 1) = 0,828$



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



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



Bisher:

Hinreichende Bedingung für die Einplanbarkeit durch $U_{LUB} = n (2^{1/n} - 1)$

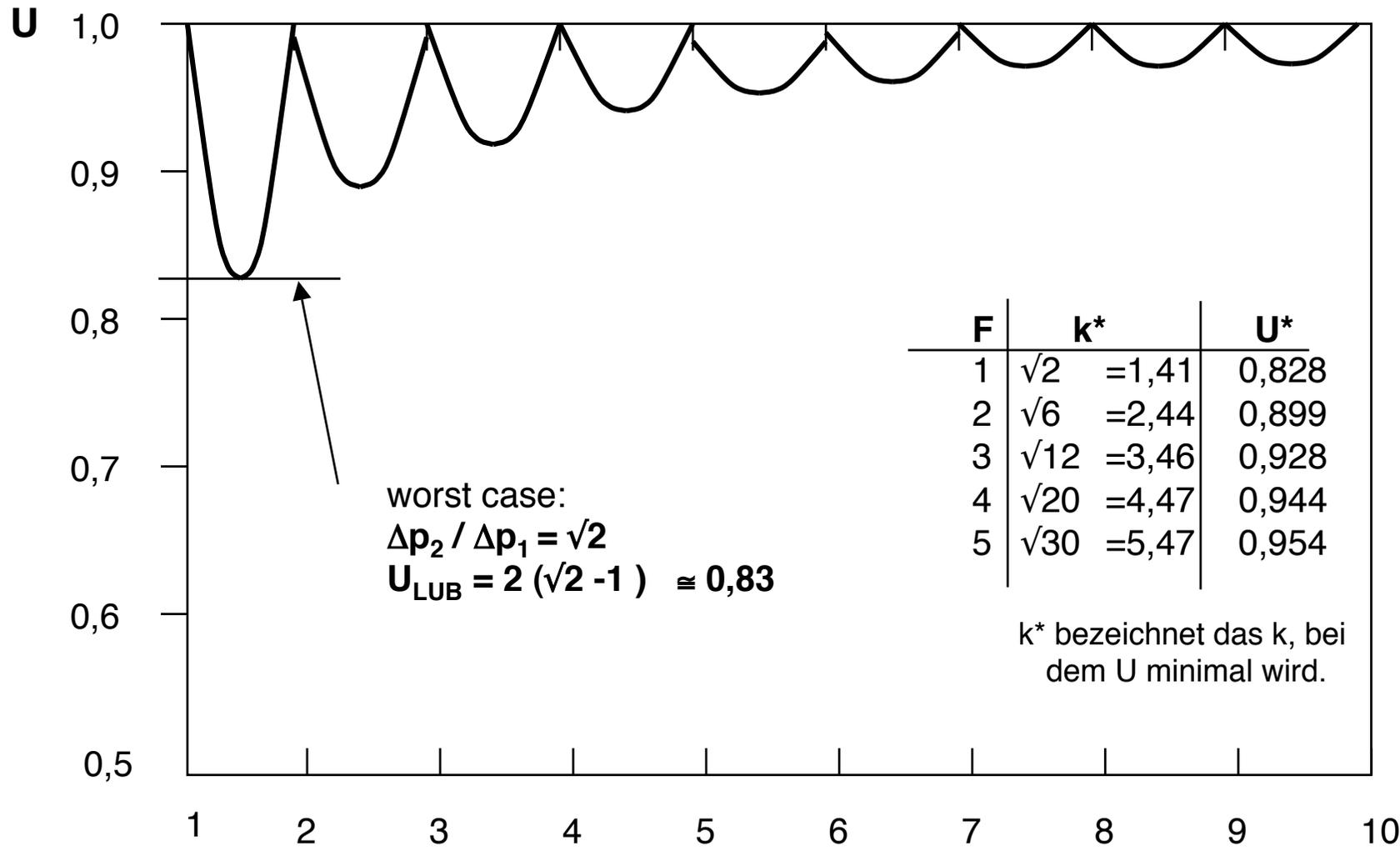
n	U_{LUB}
1	1.00
2	0,83
3	0,78
4	0,76
10	0,72

$$\lim_{n \rightarrow \infty} U_{LUB}(n) = \ln 2 \cong 0,69$$

**Herleitung notwendiger und hinreichender
Bedingungen für die Einplanbarkeit durch RMS**



Obere Schranke der Prozessorauslastung in Abhängigkeit von k (für 2 Tasks)



worst case:
 $\Delta p_2 / \Delta p_1 = \sqrt{2}$
 $U_{LUB} = 2(\sqrt{2} - 1) \approx 0,83$

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

k* bezeichnet das k, bei dem U minimal wird.

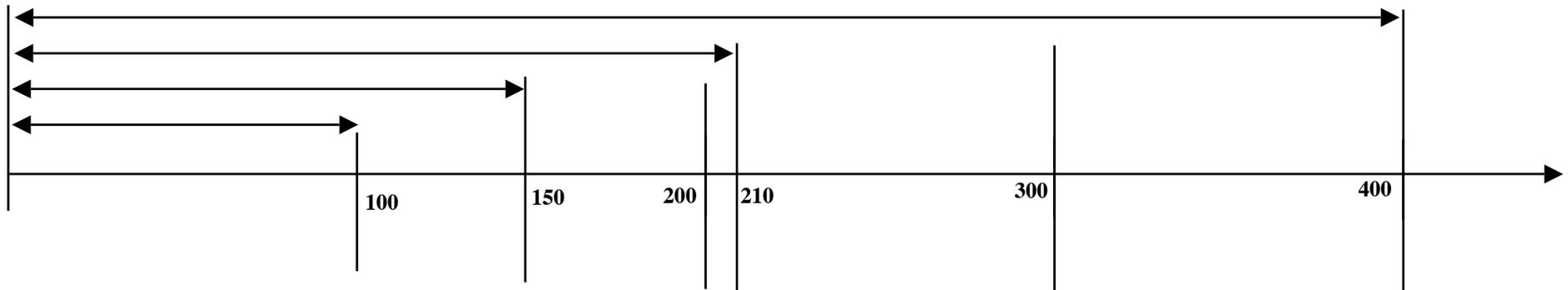
$k = \Delta p_2 / \Delta p_1$



Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS

Beispiel:

T_i	Δe_i	Δp_i
1	20	100
2	30	150
3	80	210
4	80	400



Wann kann T_1 eingeplant werden ?

Da es sich um die Task mit der höchsten Priorität handelt ist die notwendige und hinreichende Bedingung durch: $\Delta e_1 \leq \Delta p_1 = 100$ gegeben.



Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS

Wann können T_2 und T_3 eingeplant werden ?

T_2 kann erfolgreich eingeplant werden, wenn im Intervall $[0, \Delta p_2]$ genügend Prozessorzeit gefunden wird.

Angenommen T_2 wird zum Zeitpunkt t **abgeschlossen**.

Die Zeit, welche die höher priorisierte Task T_1 im Intervall $[0, t]$ benötigt ist:

$$\lceil t / \Delta p_1 \rceil \cdot \Delta e_1 ,$$

so daß bis zum Zeitpunkt t insgesamt: $\lceil t / \Delta p_1 \rceil \cdot \Delta e_1 + \Delta e_2$ benötigt wird.

Analog gilt für T_3 : $\lceil t / \Delta p_1 \rceil \cdot \Delta e_1 + \lceil t / \Delta p_2 \rceil \cdot \Delta e_2 + \Delta e_3$.

Kann man ein t im Intervall $[0, \Delta p_3]$ finden, so daß gilt : $t \geq \lceil t / \Delta p_1 \rceil \cdot \Delta e_1 + \lceil t / \Delta p_2 \rceil \cdot \Delta e_2 + \Delta e_3$
kann RM die Tasks einplanen.



Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS

Man muß nicht alle Zeitpunkte t überprüfen, sondern nur die, an denen neue Tasks aktiviert werden oder eine neue Periode der Task beginnt.

Formal ausgedrückt: Man muß die Gesamtausführungszeiten nur zu den Zeitpunkten
 $\tau_i = \{l \Delta p_j \mid j = 1, \dots, i, l = 1, \dots, \lfloor \Delta p_i / \Delta p_j \rfloor\}$
 berechnen.

Für das Beispiel ergibt sich:

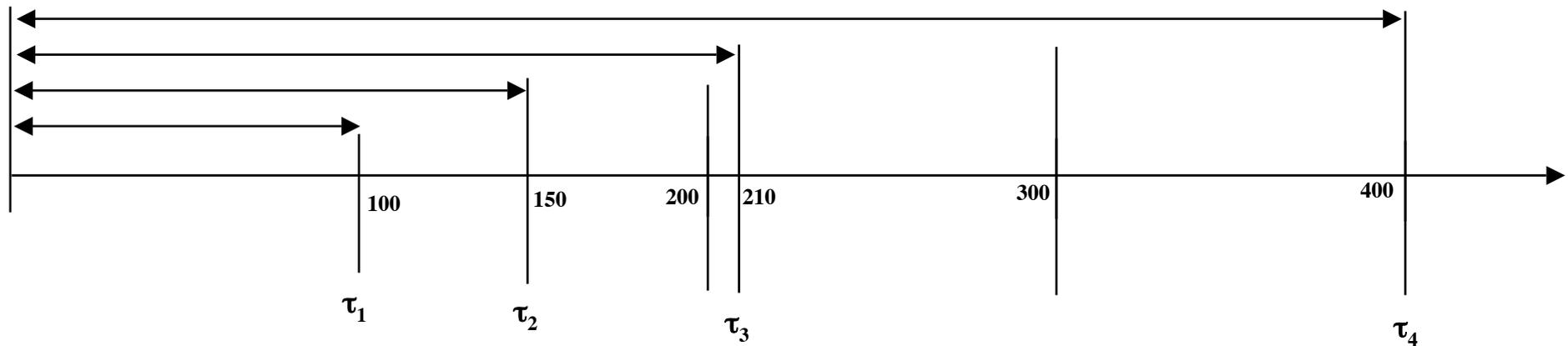
i	$\lfloor \Delta p_i / \Delta p_j \rfloor$	$l \Delta p_j \quad j=1, \dots, i$	τ_i
1	100/100=1	100	{100}
2	j=1: 150/100=1 j=2: 150/150=1	100 150	{100, 150}
3	j=1: 210/100=2 j=2: 210/150=1 j=3: 210/210=1	100, 200 150 210	{100, 150, 200, 210}
4	j=1: 400/100=4 j=2: 400/150=2 j=3: 400/210=1 j=4: 400/400=1	100, 200, 300, 400 150, 300 210 400	{100, 150, 200, 210, 300, 400}



Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS (cont.)

Beispiel:

T_i	Δe_i	P_i
1	20	100
2	30	150
3	80	210
4	80	400



$$\tau_1 = \{100\}$$

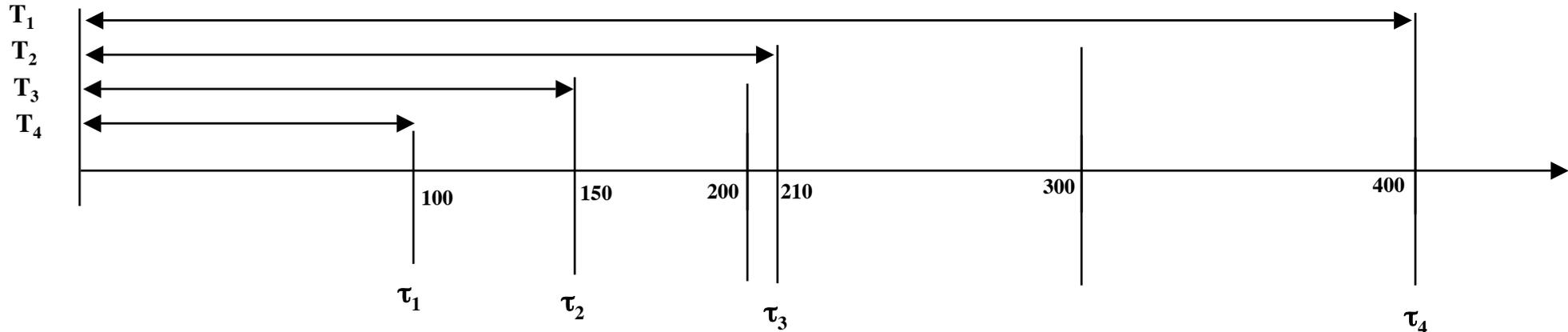
$$\tau_2 = \{100, 150\}$$

$$\tau_3 = \{100, 150, 200, 210\}$$

$$\tau_4 = \{100, 150, 200, 210, 300, 400\}$$



Herleitung notwendiger und hinreichender Bedingungen für die Einplanbarkeit durch RMS (cont.)



Prozeß T_1 kann eingeplant werden, wenn:

$$\Delta e_1 \leq 100$$

Prozeß T_2 kann eingeplant werden, wenn:

oder

$$\Delta e_1 + \Delta e_2 \leq 100$$

$$2 \cdot \Delta e_1 + \Delta e_2 \leq 150$$

Prozeß T_3 kann eingeplant werden, wenn:

oder

oder

oder

$$\Delta e_1 + \Delta e_2 + \Delta e_3 \leq 100$$

$$2 \cdot \Delta e_1 + \Delta e_2 + \Delta e_3 \leq 150$$

$$2 \cdot \Delta e_1 + 2 \cdot \Delta e_2 + \Delta e_3 \leq 200$$

$$3 \cdot \Delta e_1 + 2 \cdot \Delta e_2 + \Delta e_3 \leq 210$$

Prozeß T_4 kann eingeplant werden, wenn:

oder

oder

oder

oder

oder

$$\Delta e_1 + \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 100$$

$$2 \cdot \Delta e_1 + \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 150$$

$$2 \cdot \Delta e_1 + 2 \cdot \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 200$$

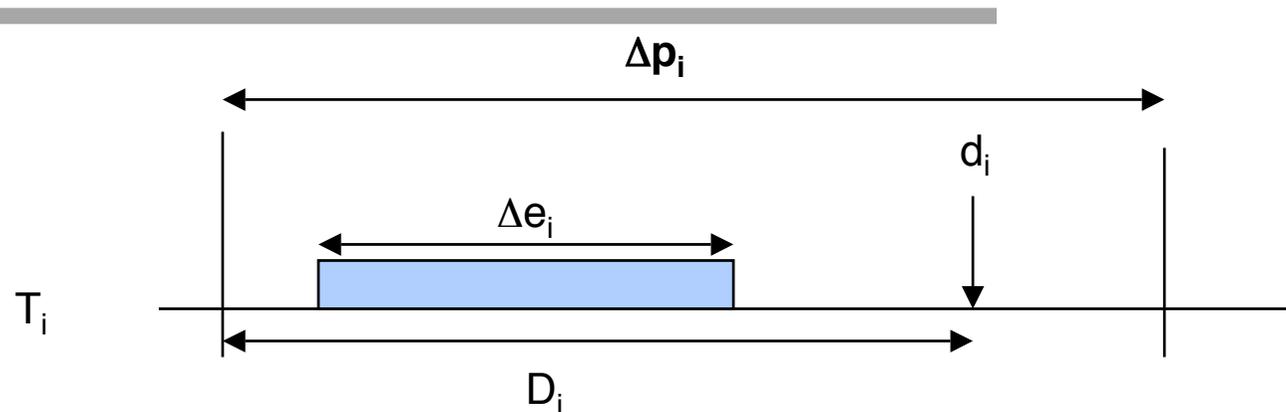
$$3 \cdot \Delta e_1 + 2 \cdot \Delta e_2 + \Delta e_3 + \Delta e_4 \leq 210$$

$$3 \cdot \Delta e_1 + 2 \cdot \Delta e_2 + 2 \cdot \Delta e_3 + \Delta e_4 \leq 300$$

$$4 \cdot \Delta e_1 + 3 \cdot \Delta e_2 + 2 \cdot \Delta e_3 + \Delta e_4 \leq 400$$



Deadline Monotonic



Eine Task T_i wird charakterisiert durch:

- eine Phase Φ_i
- eine Ausführungszeit Δe_i
- eine relative Deadline D_i
- eine Periode Δp_i

Es gelten folgende Beziehungen:

$$\Delta e_i \leq D_i \leq \Delta p_i$$
$$r_{i,k} = \Phi_i + (k-1) \Delta p_i$$
$$d_{i,k} = r_{i,k} + D_i$$

Das Verfahren ordnet einer Task die Prioritäten aufgrund der relative Deadline D_i zu. Da diese relative Deadline konstant ist, handelt es sich um ein statisches Verfahren mit festen Prioritäten. Es gilt:

Deadline Monotonic (DM) ist optimal, d.h. wenn irgendein statisches Verfahren eine Taskmenge mit Deadlines ungleich der Perioden planen kann, findet auch DM einen Plan.



Planbarkeitsanalyse

Hinreichende Bedingungen: Reduzierung der Taskperiode auf die relative Deadline.

1. Pessimistischer Test: $\sum_{i=1}^n \Delta e_i / D_i \leq n (2^{1/n} - 1)$

folgt unmittelbar aus den Bedingungen von DM in Relation zu RM.
Es ist allerdings keine gute Schranke, weil die Belastung des Prozessors überbewertet wird.

2. Weniger pessimistisch:

- Der worst case Prozessorbedarf entsteht, wenn alle Tasks simultan bereit werden.
- Für jede Task muß die Summe der Ausführungszeiten und der Interferenzen (I_i), die durch Unterbrechung (durch höher priorisierte Tasks entstehen) kleiner oder gleich der relativen Deadline D_i sein. Wenn alle Tasks nach Deadlines geordnet sind, d.h. $i < j \Leftrightarrow D_i < D_j$ gilt, ist ein solcher Test gegeben durch:

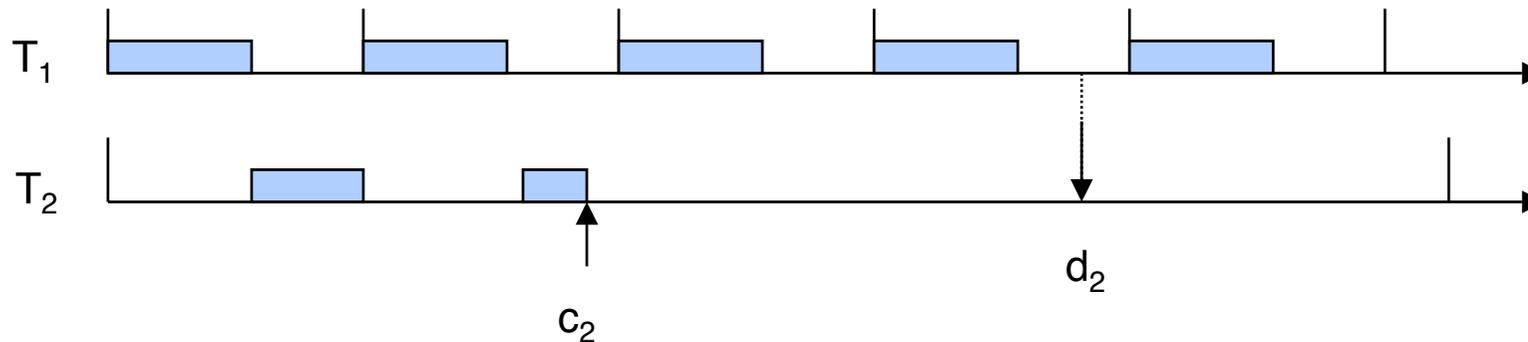
$$\forall i : 1 \leq i \leq n \quad \Delta e_i + I_i \leq D_i$$

mit $I_i = \sum_{j=1}^{i-1} \left[D_i / \Delta p_j \right] \Delta e_j$

Anzahl der Unterbrechnungen durch eine Task höherer Priorität.



Immer noch pessimistisch, wie das Beispiel zeigt:



Bei dem Test wird angenommen, dass jede höher priorisierte Task eine mit niedrigerer Priorität genau $D_i / \Delta p_j$ -mal unterbricht, was nicht der Fall ist, wenn T_i bereits früher beendet ist.

Ein iterativer Algorithmus ähnlich der iterative Ermittlung des Prozessorbedarfs wird von Audsley, Burns, Richardson und Wellings (1991 und 92) angegeben. Er ermittelt iterativ die Antwortzeit der Tasks unter allen Interferenzbedingungen.

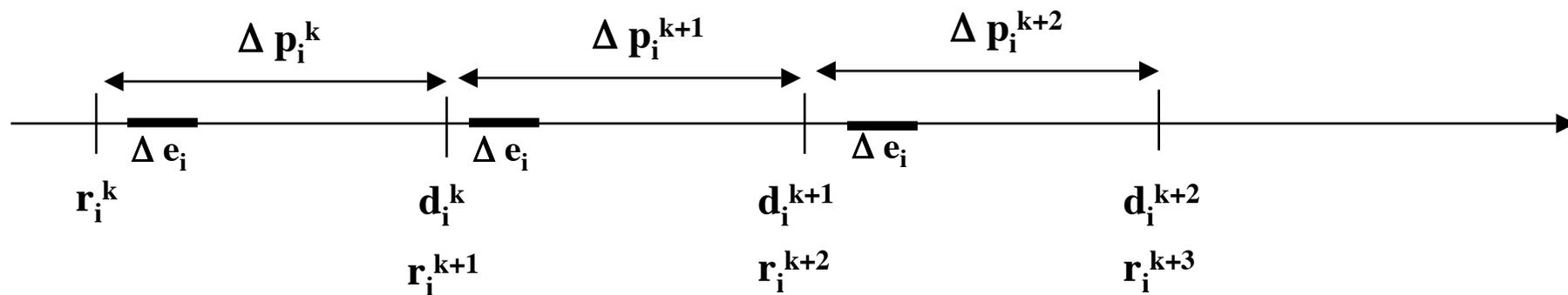
N.C. Audsley, A. Burns, M.F. Richardson and A.J. Wellings: „Hard real-time scheduling: The deadline monotonic approach“
IEEE Workshop on Real-Time Operating Systems, 1992.
Auch diskutiert in Butazzo (siehe Literaturliste)



EDF bei periodischen Prozessen

Annahme: Perioden sind a priori bekannt.

➔ r_i und d_i sind für alle Prozesse bekannt.



Bei periodischen Prozessen ist die Prozessorauslastung U durch die Summe der Anteile gegeben, die die jeweilige Ausführungszeit des Prozesses an seiner Periode hat.

Es gilt:
$$U = \sum_{i \in P} (\Delta e_i / \Delta p_i)$$

Zu beachten ist, dass EDF keinerlei spezielle Annahmen über die Periodizität der Prozesse macht, d.h. zur Planung periodischer und aperiodischer Tasks genutzt werden kann.

Deshalb gilt die Optimalität, die wir bereits gezeigt haben auch für periodische Tasks.

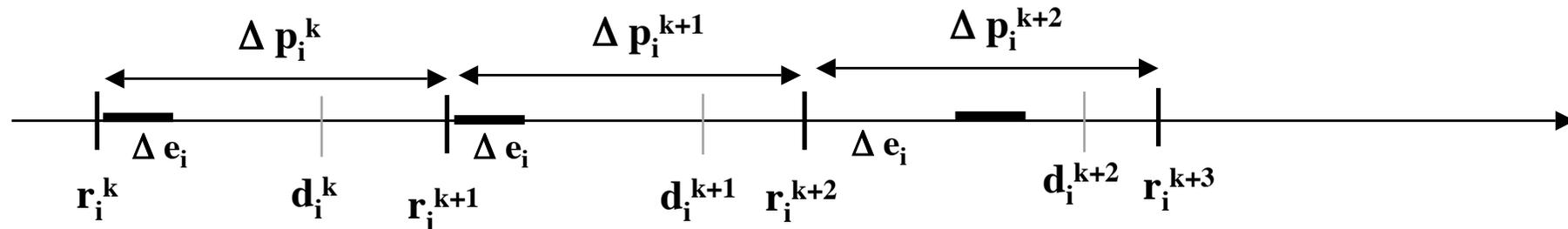


Satz (Liu, Layland 1973) :

Eine Menge periodischer Tasks kann durch EDF eingeplant werden, dann und nur dann, wenn $U = \sum_{i=1,\dots,n} (\Delta e_i / \Delta p_i) \leq 1$



EDF bei periodischen Prozessen, bei denen die Periode und die Deadline nicht zusammenfallen.



Seien: $U = \sum (\Delta e_i / \Delta p_i)$, $d_{\max} = \max_{1 \leq i \leq n} \{d_i\}$, und $P = \text{kgV}(p_1, \dots, p_n)$.

Sei $h_T(t)$ die Summe der Ausführungszeiten aller Prozesse deren absoluten Deadlines vor t liegen.

Eine Prozeßmenge kann **NICHT** erfolgreich nach EDF eingeplant werden, genau dann wenn folgende Bedingungen gelten:

1.) $U > 1$ oder

2.) es gibt ein t_{\min} mit $t_{\min} < \min \{ P + d_{\max}, (U/(1-U)) \max_{1 \leq i \leq n} \{p_i - d_i\} \}$
so daß $h_T(t) > t$



Beispiel 1:

$$\Delta p_1 = 5$$

$$\Delta p_2 = 10$$

$$\Delta e_1 = 3$$

$$\Delta e_2 = 4$$

$$d_1 = 4$$

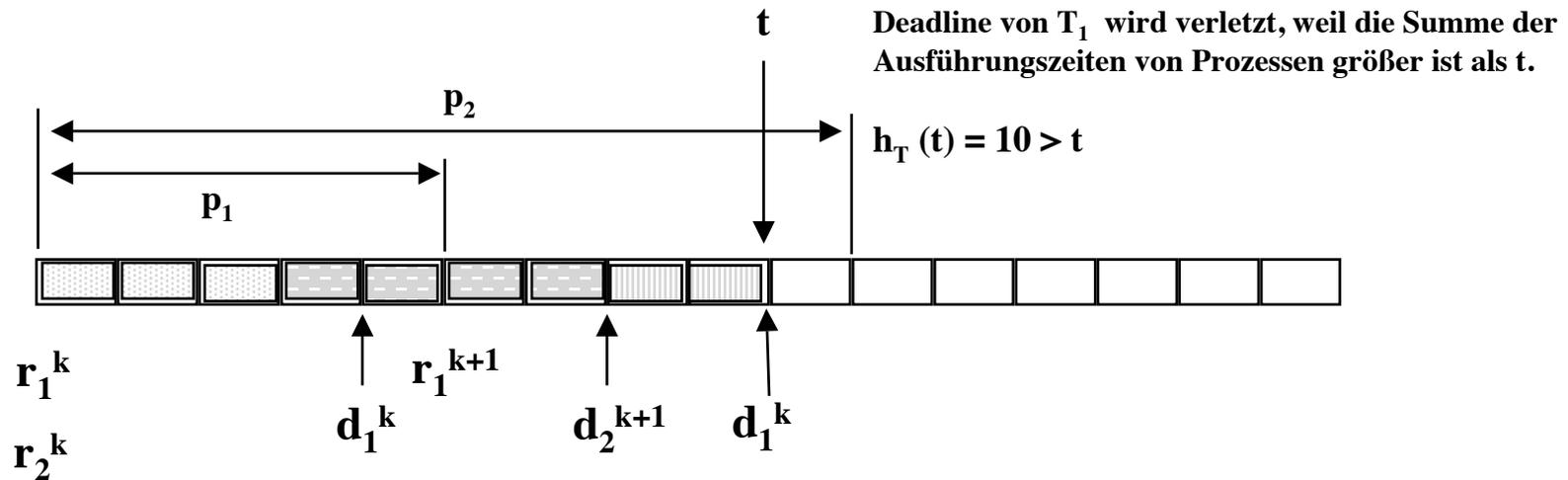
$$d_2 = 7$$

$$U = \sum (\Delta e_i / \Delta p_i) = 3/5 + 4/10 = 1$$

$$P = \text{kgV}(5, 10) = 10$$

$$P + d_{\max} = 10 + 7 = 17 < (U/(1-U)) \max_{1 \leq i \leq n} \{p_i - d_i\}$$

d.h. t kann in dem Intervall von der Länge 17 gefunden werden.



Beispiel 2:

$$\Delta p_1 = 500$$

$$\Delta p_2 = 1000$$

$$\Delta e_1 = 3$$

$$\Delta e_2 = 4$$

$$d_1 = 4$$

$$d_2 = 7$$

$$U = \sum (\Delta e_i / \Delta p_i) = 3/500 + 4/1000 = 0,01$$

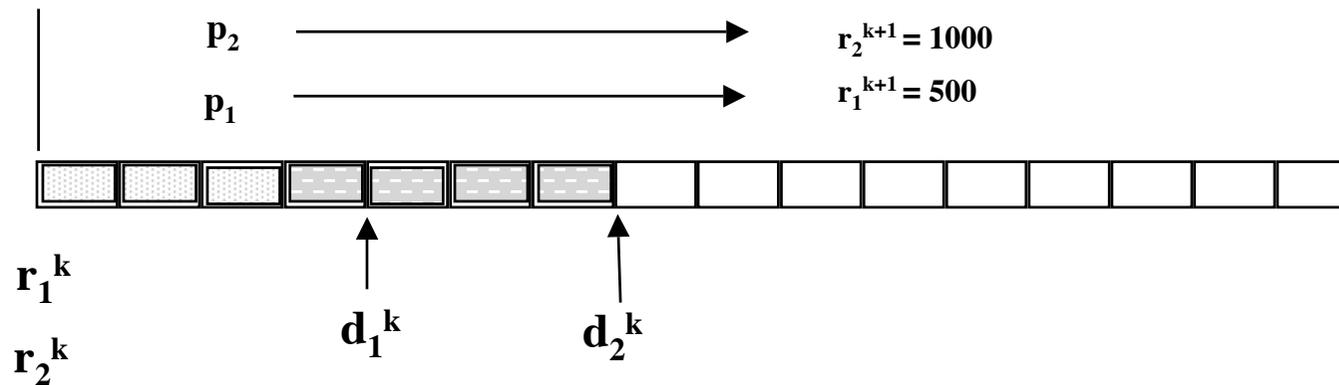
$$P = \text{kgV}(500, 1000) = 1000$$

$$P + d_{\max} = 1000 + 7 = 1007 > (U/(1-U)) \max_{1 \leq i \leq n} \{p_i - d_i\} = 0,01/0,99 * 993 \approx 10$$

d.h. t kann in dem Intervall von der Länge 10 gefunden werden.

Deadline von P_1 wird nicht verletzt, weil kein t gefunden werden kann,
für das gilt:

$$h_T(t) = 10 > t$$



Zusammenfassung:



Allgemeiner Fall: EDF ist optimal, es existieren notwendige und hinreichende Bedingungen für die Einplanbarkeit.



Periodische Prozesse

★ **Deadline = Periode: notwendige und hinreichende Bedingung**
ist: $U = \sum (\Delta e_i / \Delta p_i) = \sum \Delta e_i f_i \leq 1$

★ **Deadline \neq Periode:**

Eine Prozeßmenge kann NICHT erfolgreich nach EDF eingeplant werden, wenn folgende (hinreichenden) Bedingungen gelten:

1.) $u > 1$ oder

2.) es gibt ein t_{\min} mit $t_{\min} < \min \{ P + d_{\max}, (u/1-u) \max_{1 \leq i \leq n} \{p_i - d_i\} \}$
so daß $h_T(t) > t$

