SAPIENZA
UNIVERSITÀ DI ROMA

Algoritmi di scheduling
- Parte 1

Automazione

14/10/2015

Vincenzo Suraci

STRUTTURA DEL NUCLEO TEMATICO

- ALGORITMO RATE MONOTONIC PRIORITY ORDERING (RMPO)
- ALGORITMO EARLIEST DEADLINE FIRST (EDF)

ALGORITMO RATE MONOTONIC PRIORITY ORDERING (RMPO)

ALGORITMO RATE MONOTONIC PRIORITY ORDERING (RMPO)

Siano noti i REQUISITI ed i VINCOLI DI SISTEMA di un problema di scheduling di task periodici:

- REQUISITI DI SISTEMA: $(n, T_1, T_2, \ldots, T_n)$;
- VINCOLI DI SISTEMA: $(C_1, C_2, \ldots, C_n)$;

L'algoritmo RMPO è un algoritmo di scheduling **PREEMPTIVE** che assegna a ciascun task una priorità inversamente proporzionale al periodo di attivazione T_i .

Dato che il periodo di attivazione T_i è fissato per ogni task, l'algoritmo RMPO è **STATICO**.

Dato che al variare del numero e dei periodi di attivazione dei task in ingresso allo scheduler, la configurazione dei task in uscita dallo scheduler può variare, l'algoritmo RMPO è **ON-LINE**.

L'algoritmo RMPO manda in esecuzione per primi quei task che hanno periodo di attivazione più breve. Pertanto è chiamato anche **SHORTEST PERIOD FIRST**.

ESEMPIO

PROBLEMA

Consideriamo un problema di coordinamento di task periodici dove:

- REQUISITI DI SISTEMA: (n = 3 ; T_1 = 8 t.u. ; T_2 = 16 t.u. ; T_3 = 12 t.u.)
- VINCOLI DI SISTEMA: (C_1 = 2 t.u. ; C_2 = 3 t.u. ; C_3 = 5 t.u.)

Mostrare lo scheduling temporale tramite l'algoritmo RATE MONOTONIC PRIORITY ORDERING (RMPO).

SVOLGIMENTO

Per prima cosa verifichiamo che sussista la condizione necessaria di esistenza della soluzione al problema dato, calcolando il **fattore di utilizzazione**:

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{2}{8} + \frac{3}{16} + \frac{5}{12} = \frac{12 + 9 + 20}{48} = \frac{41}{48} \approx 0{,}8542 < 1$$

Dato che U < 1, possiamo escludere che il problema sia inammissibile.

ESEMPIO cont'd

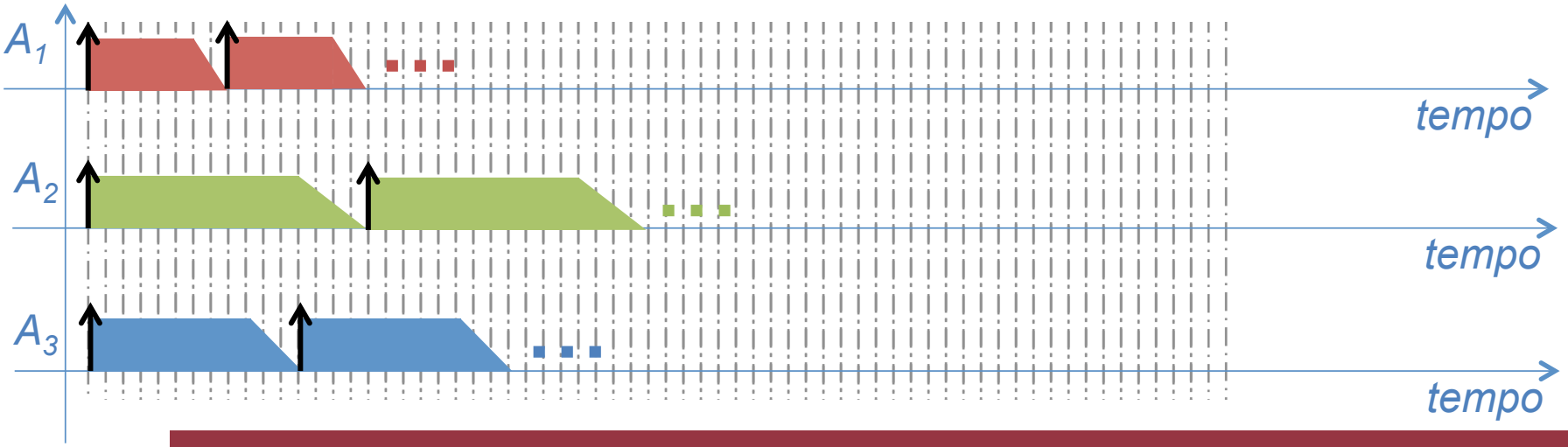
Prima di tracciare passo dopo passo lo scheduling, calcoliamo le priorità statiche dei tre differenti task:

$$A_1 \rightarrow \frac{1}{8} \; ; A_2 \rightarrow \frac{1}{16} \; ; A_3 \rightarrow \frac{1}{12}$$

L'ordine di priorità è pertanto:

$$A_1 = 0{,}125 \; ; A_3 = 0{,}08\overline{3} \; ; A_2 = 0{,}0625$$

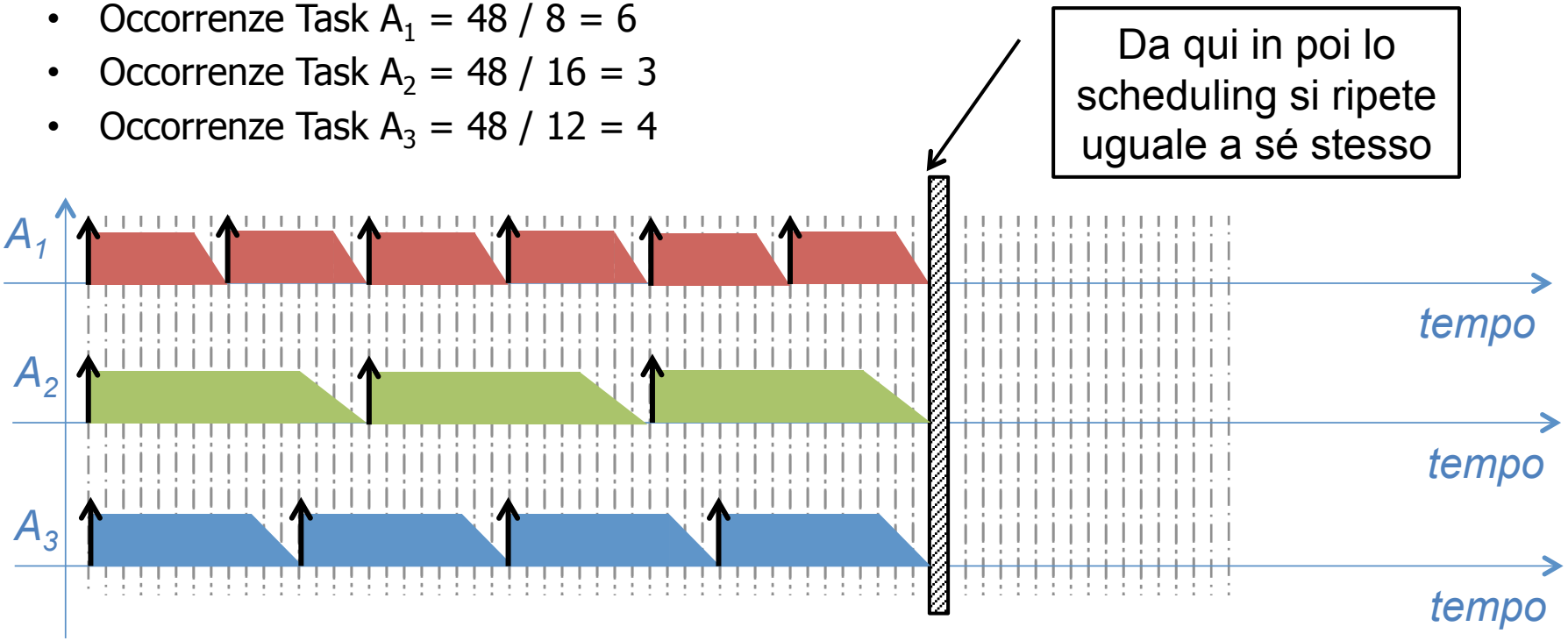
Disponiamo quindi i task in assi temporali isocroni:



ESEMPIO cont'd

Per sapere il numero minimo di occorrenze che bisogna disegnare per il task i-esimo, conviene calcolare il **MINIMO COMUNE MULTIPLO** dei periodi di attivazione di tutti i task e dividere per il periodo di attivazione del task i-esimo.

- Minimo comune multiplo (8 , 16 , 12) = 48 t.u.
- Occorrenze Task A_1 = 48 / 8 = 6
- Occorrenze Task A_2 = 48 / 16 = 3
- Occorrenze Task A_3 = 48 / 12 = 4

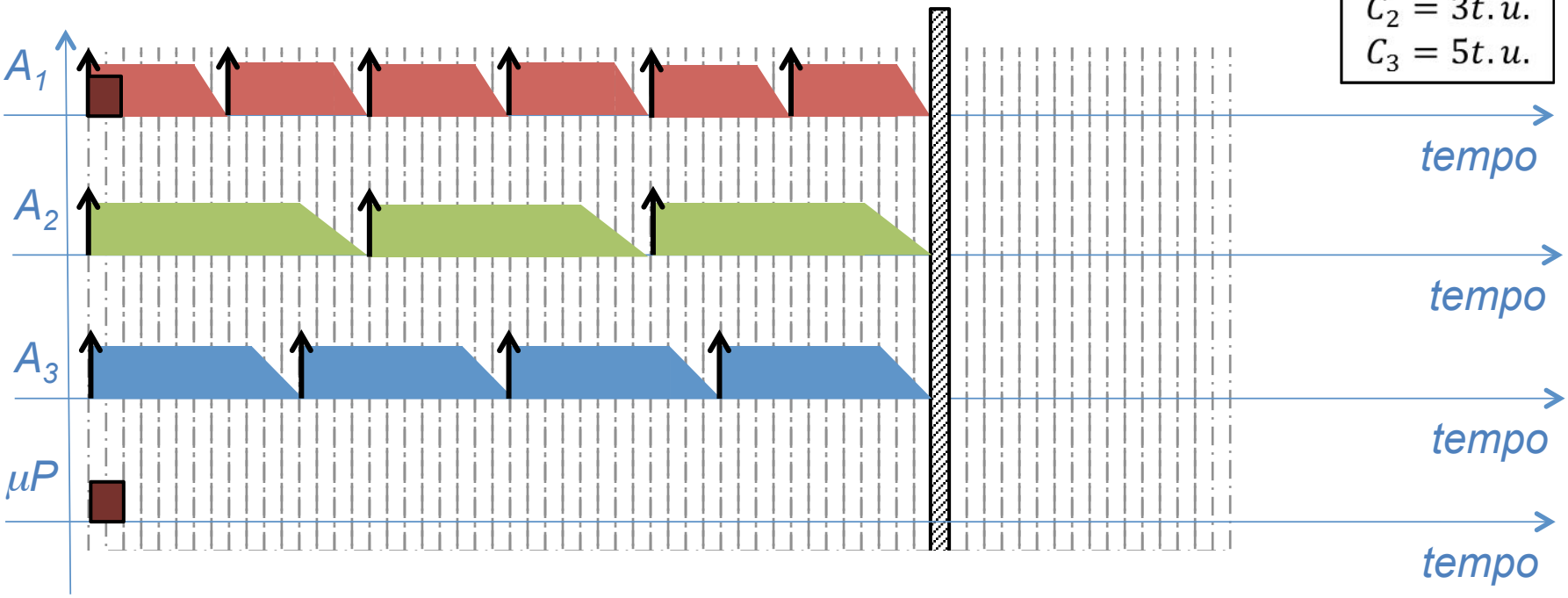


ESEMPIO cont'd

Ricordando le priorità dei task definite dall'algoritmo RMPO (A_1 A_3 A_2), partiamo a scansionare l'asse dei tempi e ad identificare quali task verranno mandati in esecuzione.

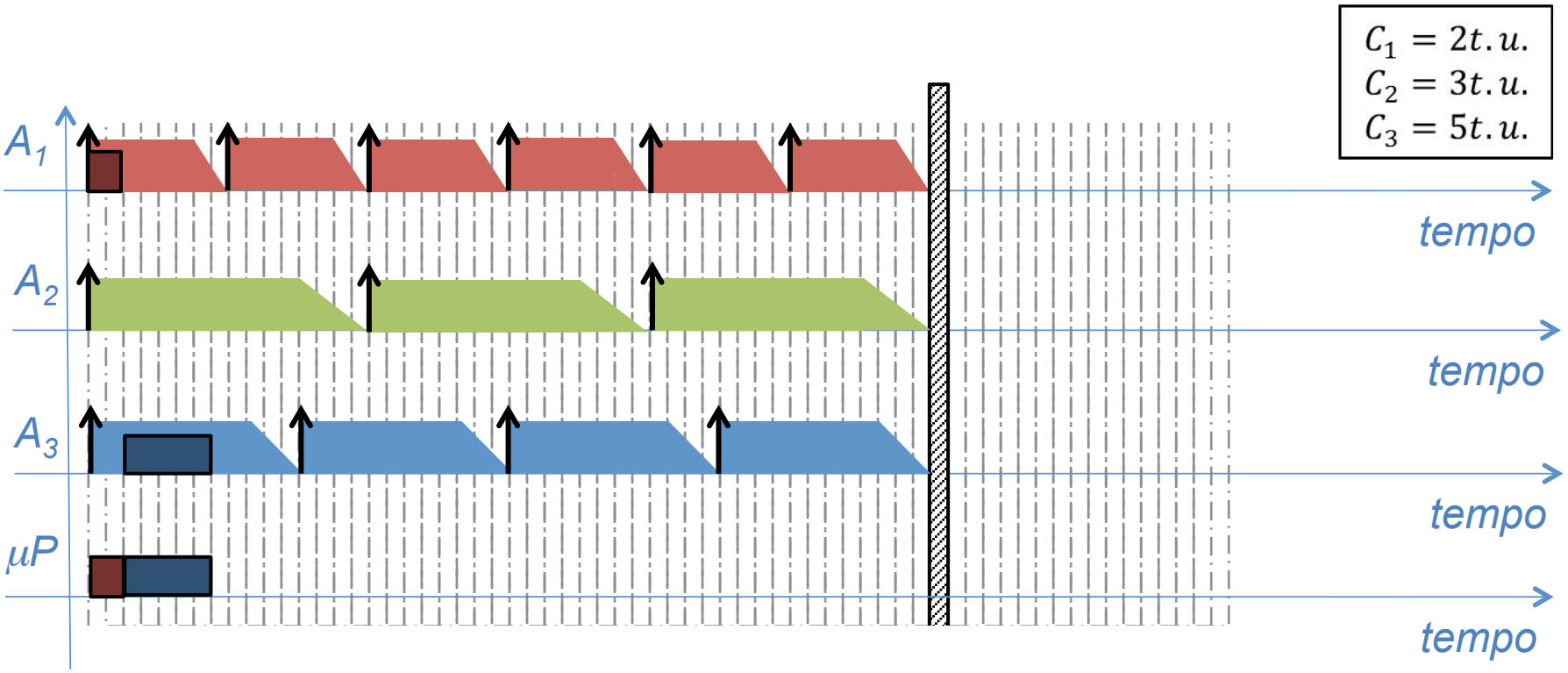
Il task A_1 è il primo ad essere eseguito.

$$C_1 = 2t.u.$$
$$C_2 = 3t.u.$$
$$C_3 = 5t.u.$$



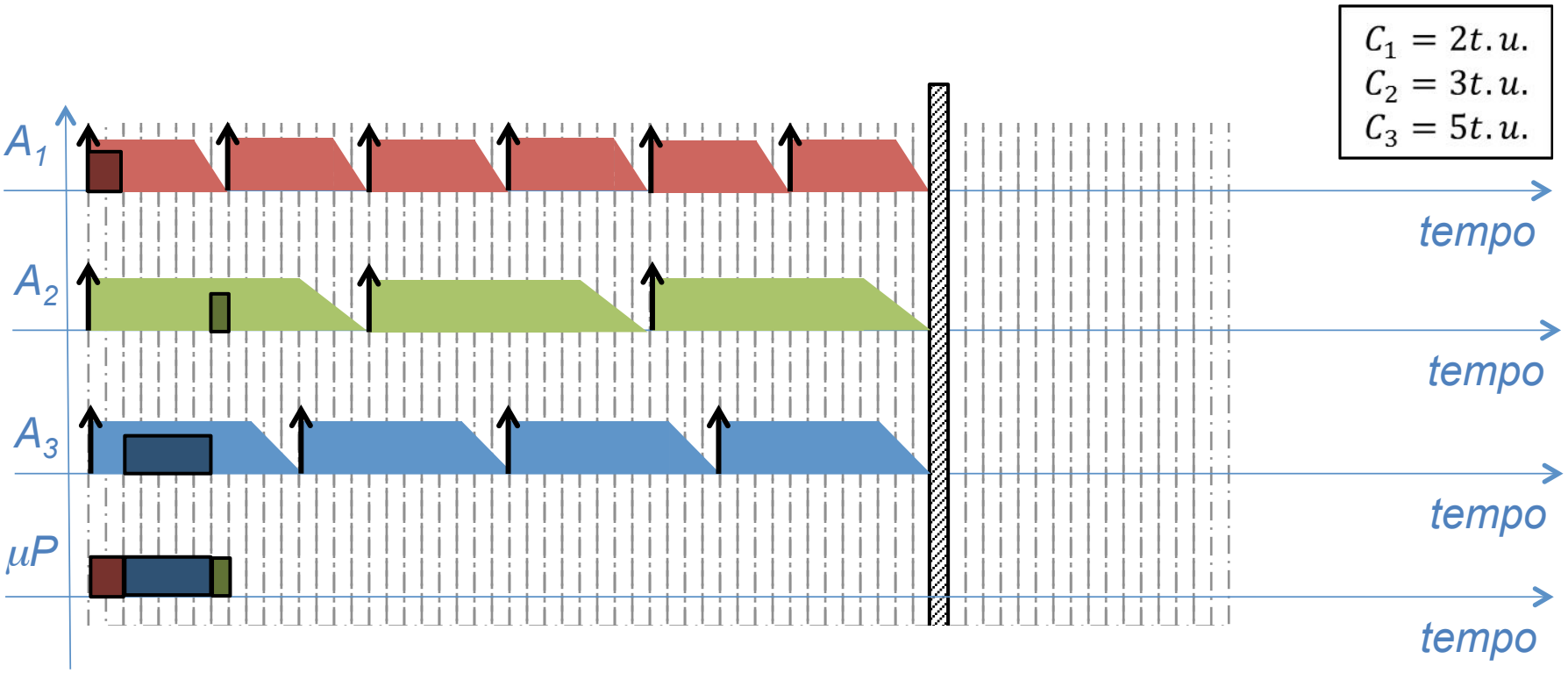
ESEMPIO cont'd

Portato a termine il A_1 (in C_1 = 2 t.u.), non essendo ancora esaurito il tempo di attivazione della prima esecuzione del task A_1, viene mandato in esecuzione il task A_3 perché ha maggiore priorità rispetto al task A_2.



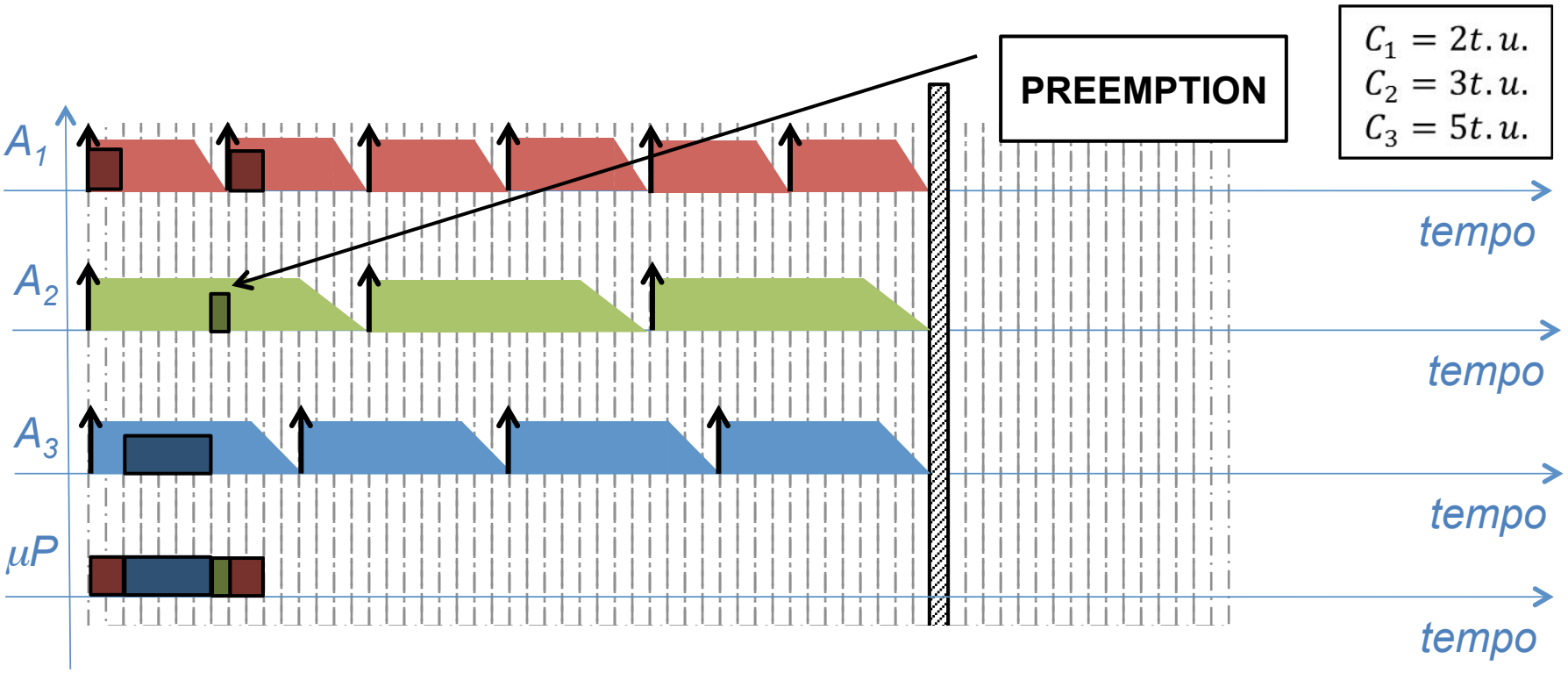
ESEMPIO cont'd

Portato a termine il A_3 (in C_3 = 5 t.u.), non essendo ancora esauriti i tempi di attivazione della prima esecuzione del task A_1 e del task A_3 viene mandato in esecuzione il task A_2.



ESEMPIO cont'd

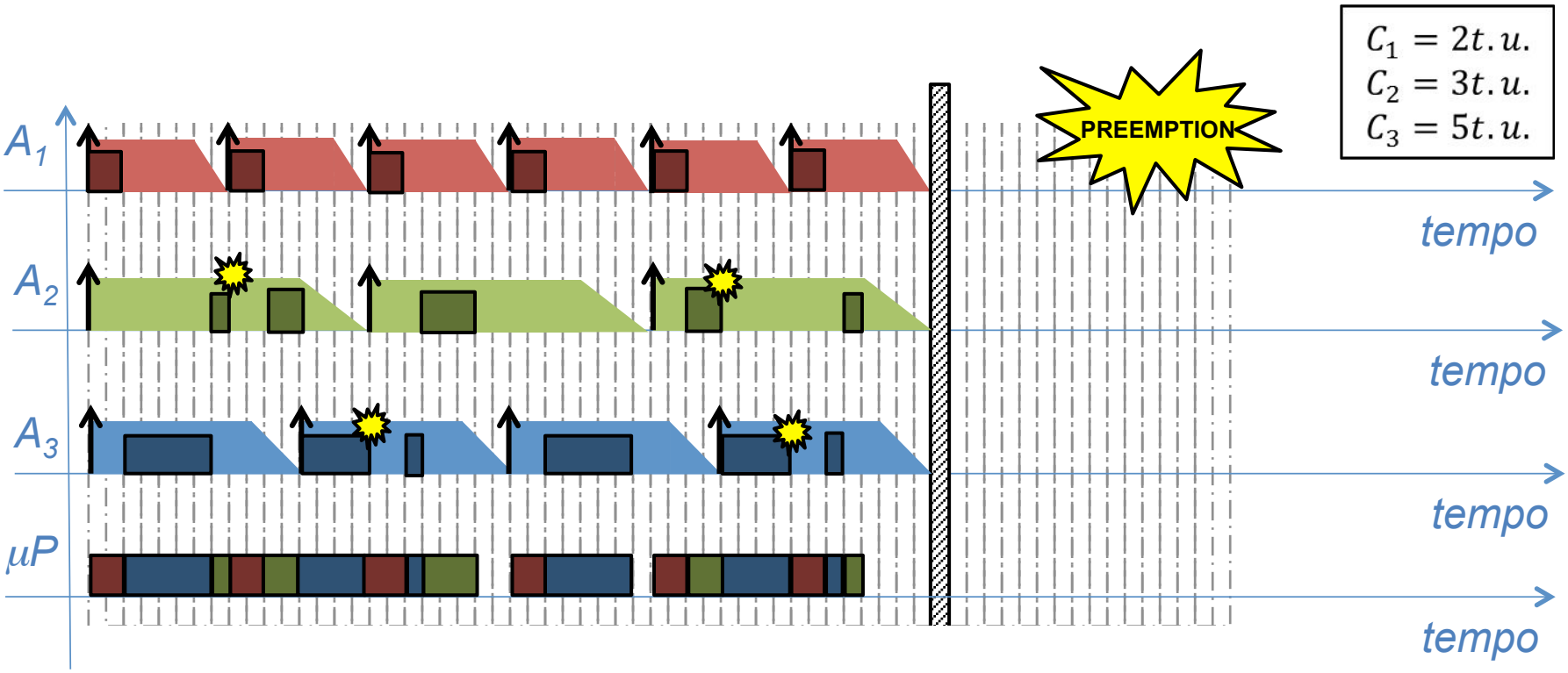
Il task A_2 non può essere terminato, in quanto dopo una t.u. la seconda esecuzione del task A_1, che ha massima priorità, è stata attivata. L'algoritmo di RMPO fa quindi PREEMPTION del task A_2 in favore del task A_1.



ESEMPIO cont'd

Si continua pertanto fino ad ottenere lo scheduling dei task.

- L'insieme di task dato è schedulabile con un algoritmo di RMPO



Proprietà dell'algoritmo RMPO

PROPOSIZIONE 1 (senza dimostrazione)

Se un insieme di task periodici **non risulta schedulabile tramite l'algoritmo RMPO**, allora **non esiste nessun altro algoritmo di scheduling STATICO** che riesca a risolvere lo stesso problema.

PROPOSIZIONE 2 (senza dimostrazione)

Il LIMITE SUPERIORE MINIMO del fattore di utilizzazione dell'algoritmo RMPO, calcolato per un insieme QUALSIASI di *n* task periodici è:

$$U_{lsm}(RMPO) = n\left(2^{1/n} - 1\right)$$

OSSERVAZIONE 1

La prima proposizione evidenzia come **l'algoritmo RMPO sia ottimo** rispetto a tutti gli altri **algoritmi di scheduling** caratterizzati da una **assegnazione statica della priorità** dei task.

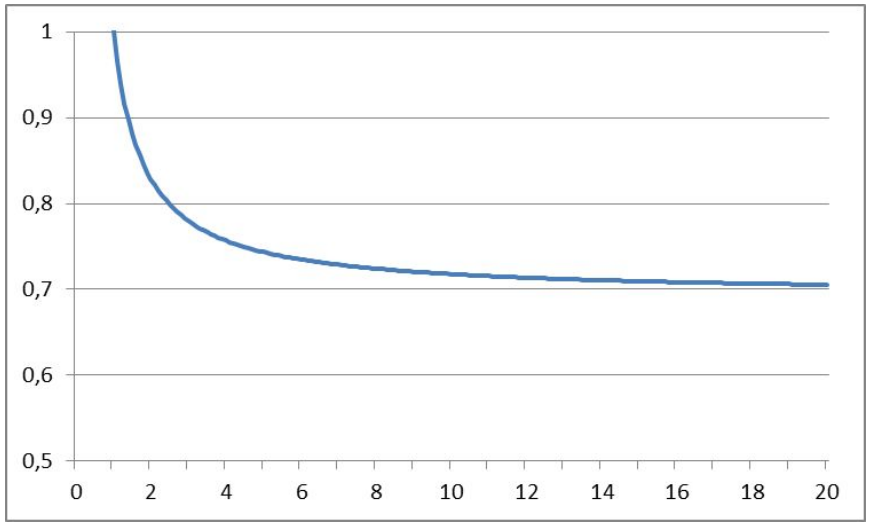
Proprietà dell'algoritmo RMPO

OSSERVAZIONE 2

Sapendo che il limite superiore minimo del fattore di utilizzazione dell'algoritmo RMPO è:

$$U_{lsm}(RMPO) = n\left(2^{1/n} - 1\right)$$

Possiamo tracciarne, al variare di n, il suo valore:



Proprietà dell'algoritmo RMPO

Calcolando il limite per n → ∞ si ottiene una forma indeterminata:

$$\lim_{n\to\infty} n\left(2^{1/n}-1\right) = \infty \cdot 0$$

$$\mathrm{D}\left[f\left(g(x)\right)\right] = f'\left(g(x)\right)\cdot g'(x)$$

$$\mathrm{D}(a^x) = a^x \ln a$$

Applicando la regola di de l'Hôpital abbiamo:

$$\lim_{n\to\infty} n\left(2^{1/n}-1\right) = \lim_{n\to\infty} \frac{2^{1/n}-1}{1/n} = \lim_{n\to\infty} \frac{\frac{d}{dn}\left(2^{1/n}-1\right)}{\frac{d}{dn}(1/n)} =$$

$$\lim_{n\to\infty} \frac{2^{1/n} ln2 (-1/n^2)}{-1/n^2} = \lim_{n\to\infty} 2^{1/n} ln2 = ln2 \approx 0{,}693$$

Proprietà dell'algoritmo RMPO

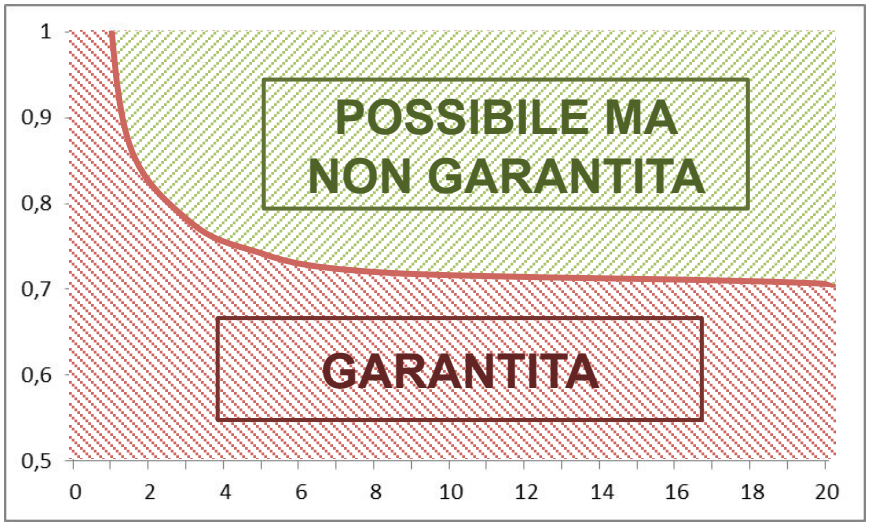
OSSERVAZIONE 3

Dato un QUALSIASI insieme di task periodici, l'algoritmo RMPO GARANTISCE la schedulabilità fino ad un fattore di occupazione pari a 69,3%.

OSSERVAZIONE 4

Dato un insieme di n task periodici, l'algoritmo RMPO GARANTISCE la schedulabilità fino ad un fattore di occupazione pari a $U_{lsm}(RMPO) = n\left(2^{1/n} - 1\right)$, oltre tale valore la schedulabilità potrebbe essere possibile, MA NON È GARANTITA.

$$U \begin{cases} > n\left(2^{1/n} - 1\right) \\ \\ \leq n\left(2^{1/n} - 1\right) \end{cases}$$



ESEMPIO cont'd

Ritornando all'esempio visto precedentemente, essendo n = 3, avremo che il limite superiore minimo del coefficiente di utilizzazione è:

$$U_{lsm}(RMPO) = n\left(2^{1/n} - 1\right) \approx 0{,}78$$

Il coefficiente di utilizzazione dato nell'esempio era:

$$U \approx 0{,}8542 > U_{lsm}(RMPO)$$

Pertanto non era assolutamente garantita la schedulabilità del problema con l'algoritmo RMPO e doveva essere verificata attraverso un diagramma dei tempi.

Proprietà dell'algoritmo RMPO

DEFINIZIONE

Un insieme di n task periodici è **LEGATO DA RELAZIONI ARMONICHE**, se esiste un task i-esimo tale che ogni periodo di esecuzione di un task j-esimo è multiplo del periodo di esecuzione del task i-esimo.

$$\exists\, i \in (1,2,..,n) \mid \{\forall j \in (1,2,..,n)\ \exists\, a_j \in \mathbb{N} \mid\ T_j = a_j T_i\}$$

PROPOSIZIONE 3 (senza dimostrazione)

Dato un QUALSIASI insieme di task periodici legati da relazioni armoniche, esso è schedulabile tramite RMPO a patto che $U \leq 1$.

Proprietà dell'algoritmo RMPO

OSSERVAZIONI

- Abbiamo visto come nella AUTOMAZIONE sia importante che un sistema Real Time sia PREVEDIBILE e quindi schedulabile.

- La schedulazione di *n* task periodici che NON possiedono relazioni armoniche e che sono caratterizzati da un fattore di utilizzazione $n\left(2^{1/n}-1\right) < U \leq 1$, sebbene possibile, NON È GARANTITA DALL'UTILIZZO DELL'ALGORITMO RMPO.

- Essendo RMPO l'ottimo tra gli algoritmi STATICI, dovremo guardare ad algoritmi DINAMICI per ottenere maggiore prevedibilità.

ALGORITMO EARLIEST DEADLINE FIRST (EDF)

ALGORITMO EARLIEST DEADLINE FIRST (EDF)

Siano noti i REQUISITI ed i VINCOLI DI SISTEMA di un problema di scheduling di task periodici:

- REQUISITI DI SISTEMA: $(n, T_1, T_2, \ldots, T_n)$
- VINCOLI DI SISTEMA: $(C_1, C_2, \ldots, C_n)$

L'algoritmo EDF è un algoritmo di scheduling **PREEMPTIVE** che assegna a ciascun task $A_i(k)$ una priorità inversamente proporzionale alla deadline assoluta $d_i(k)$. Quando due task $A_i(k')$ e $A_j(k'')$ hanno la stessa deadline assoluta, viene data priorità al task con il numero di iterazione k più piccolo.

- Dato che al variare del numero e dei periodi di attivazione dei task in ingresso allo scheduler, la configurazione dei task in uscita dallo scheduler può variare, l'algoritmo EDF è **ON-LINE**.

- Dato che ad ogni attivazione di un task $A_i(k)$ la priorità dei task attivi deve essere ricalcolata, l'algoritmo EDF è **DINAMICO**.

ESEMPIO

PROBLEMA

Consideriamo un problema di coordinamento di task periodici dove:

- REQUISITI DI SISTEMA: (n = 3 ; T_1 = 8 t.u.; T_2 = 24 t.u.; T_3 = 12 t.u.)
- VINCOLI DI SISTEMA: (C_1 = 4 t.u.; C_2 = 6 t.u.; C_3 = 3 t.u.)

Mostrare lo scheduling temporale tramite l'algoritmo EARLY DEADLINE FIRST (EDF).

SVOLGIMENTO

Per prima cosa verifichiamo che sussista la condizione necessaria di esistenza della soluzione al problema dato, calcolando il **fattore di utilizzazione**:

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{4}{8} + \frac{6}{24} + \frac{3}{12} = \frac{12 + 6 + 6}{24} = 1$$

Dato che U = 1, possiamo escludere che il problema sia inammissibile.

ESEMPIO cont'd

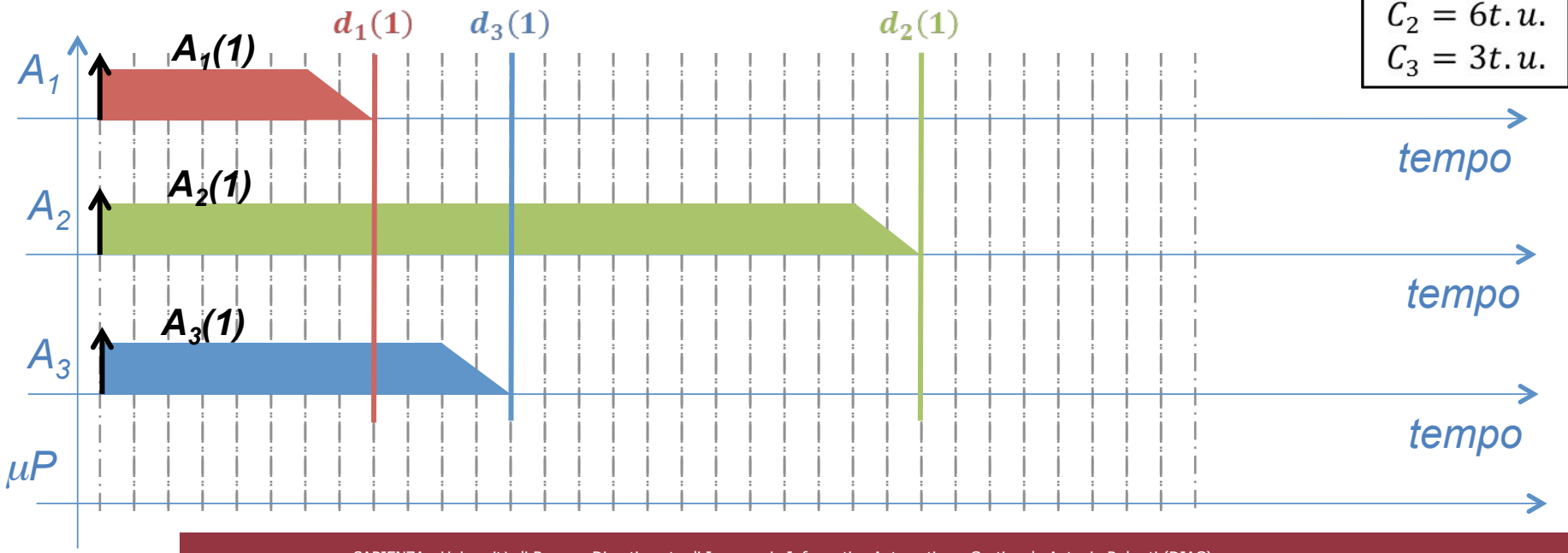
All'inizio si presentano nella READY QUEUE tre task attivi: $A_1(1)$, $A_2(1)$, $A_3(1)$. Le priorità dei tre task dipendono dal reciproco delle loro deadline assolute, pertanto:

$$A_1(1) = 1/d_1(1) = 1/8 = 0{,}125$$

$$A_2(1) = 1/d_2(1) = 1/24 = 0{,}041\overline{6}$$

$$A_3(1) = 1/d_3(1) = 1/12 = 0{,}08\overline{3}$$

$$C_1 = 4t.u.$$
$$C_2 = 6t.u.$$
$$C_3 = 3t.u.$$



ESEMPIO cont'd

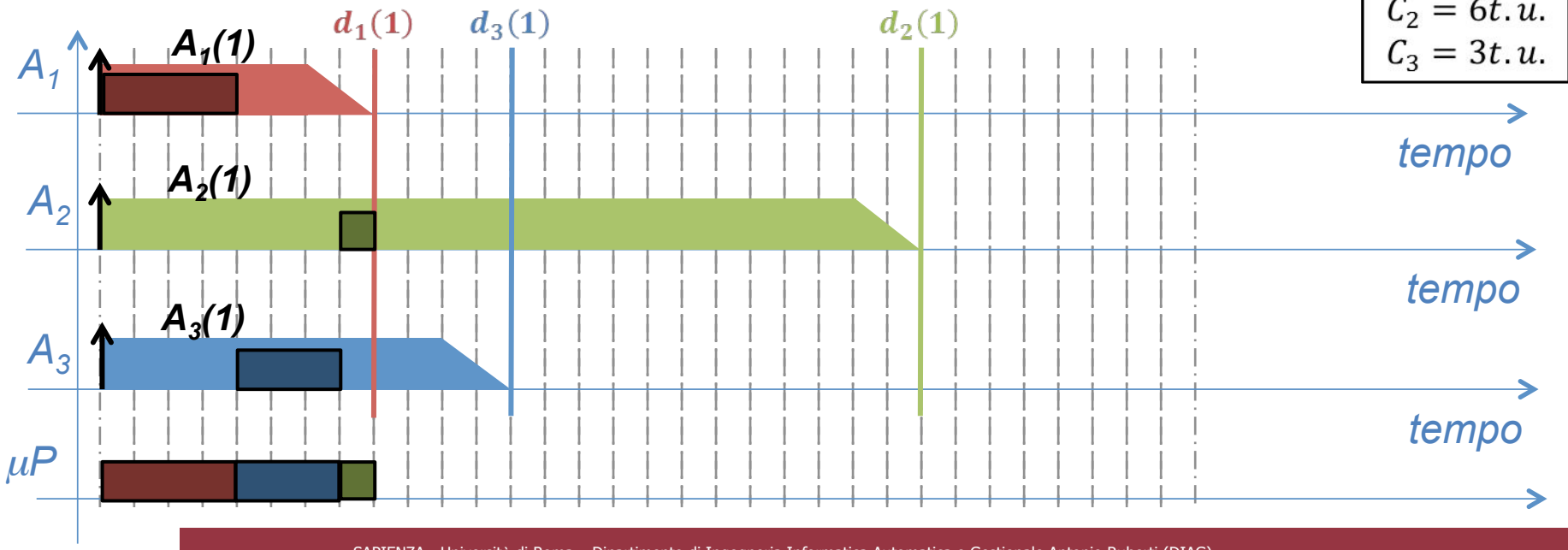
Fino allo scadere della successiva deadline assoluta ($d_1(1)$=8 t.u.) non cambieranno le priorità dei task attivi. Quindi si procede allo scheduling con ordine di priorità:

$$A_1(1) = 1/d_1(1) = 1/8 = 0{,}125$$

$$A_3(1) = 1/d_3(1) = 1/12 = 0{,}08\overline{3}$$

$$A_2(1) = 1/d_2(1) = 1/24 = 0{,}041\overline{6}$$

$$C_1 = 4t.u.$$
$$C_2 = 6t.u.$$
$$C_3 = 3t.u.$$

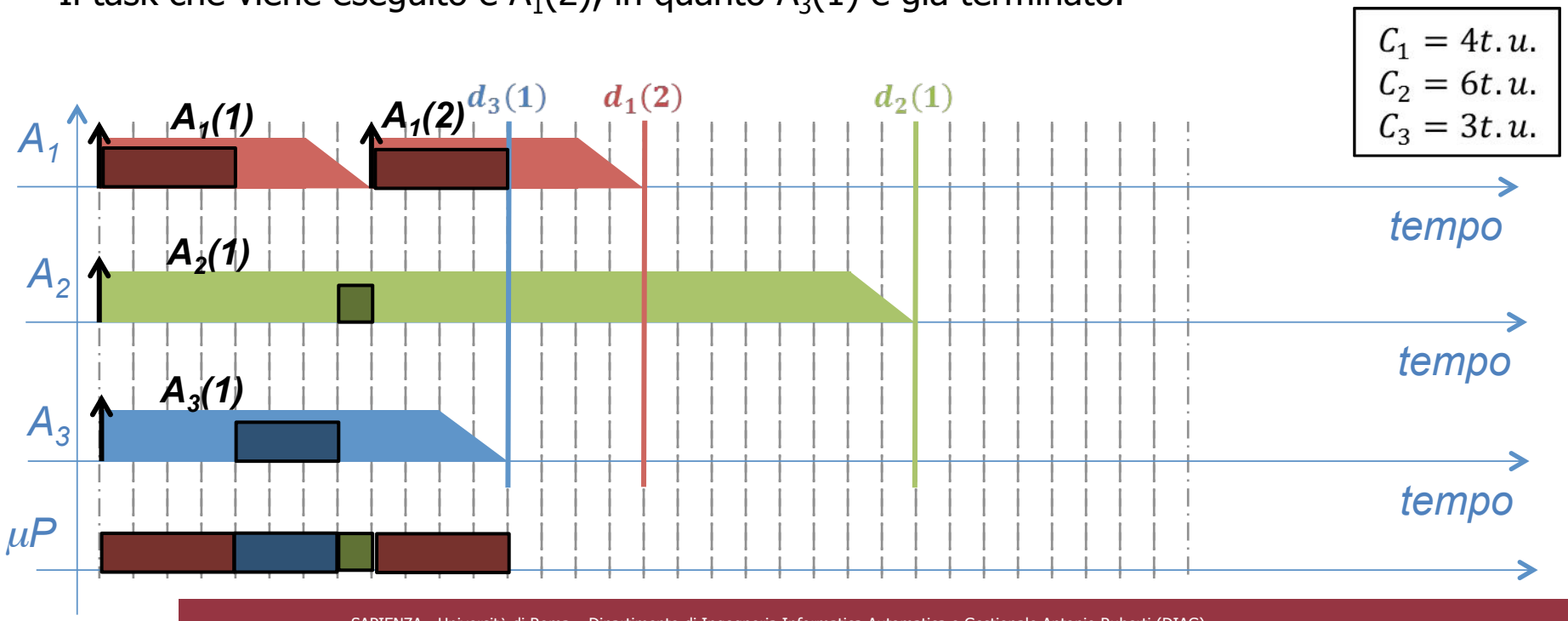


ESEMPIO cont'd

Con l'arrivo nella READY QUEUE del task $A_1(2)$, la priorità dei task attivi deve essere ricalcolata, ottenendo la seguente classifica di priorità:

$$A_3(1) = 0{,}08\overline{3} \qquad A_1(2) = 1/d_1(2) = 1/16 = 0{,}0625 \qquad A_2(1) = 0{,}041\overline{6}$$

Il task che viene eseguito è $A_1(2)$, in quanto $A_3(1)$ è già terminato.

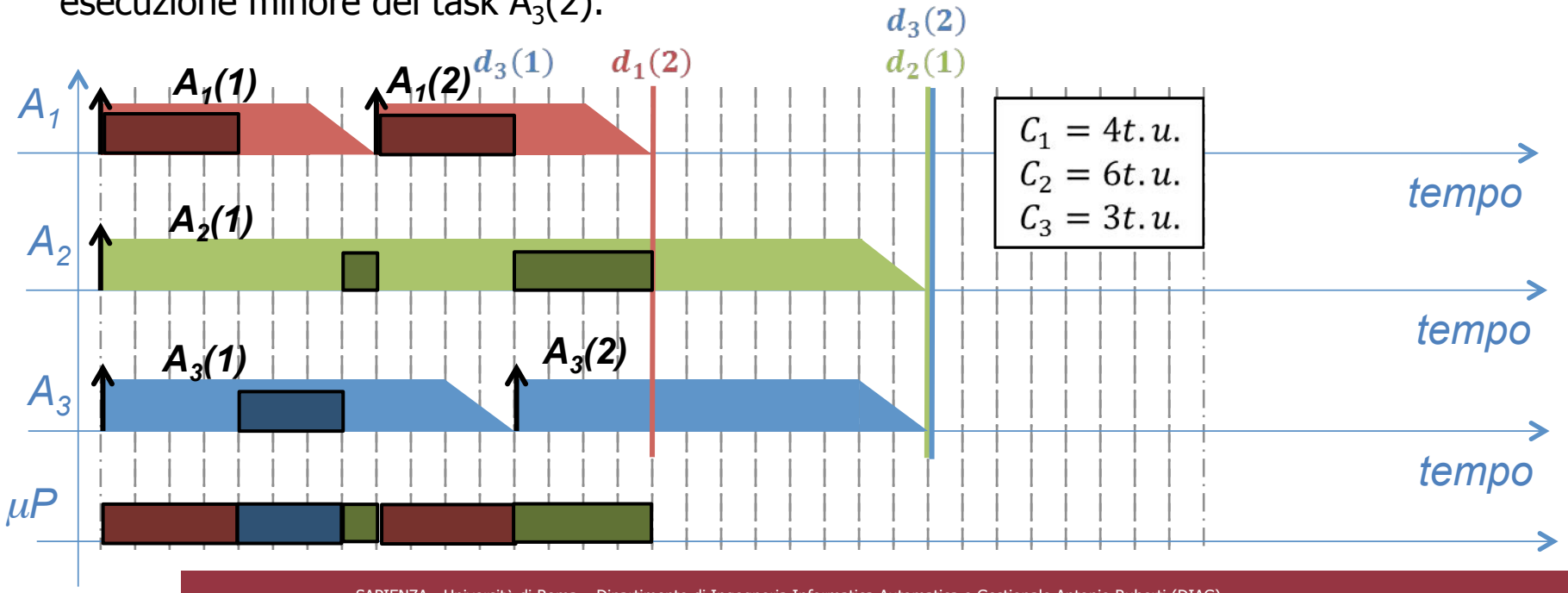


ESEMPIO cont'd

Con l'arrivo nella READY QUEUE del task $A_3(2)$, la priorità dei task attivi deve essere ricalcolata, ottenendo la seguente classifica di priorità:

$$A_1(2) = 1/d_1(2) = 1/16 = 0{,}0625 \qquad A_2(1) = 0{,}041\overline{6} \qquad A_3(2) = 0{,}041\overline{6}$$

Dato che il task $A_1(2)$ è terminato, viene eseguito il $A_2(1)$ in quanto ha numero di esecuzione minore del task $A_3(2)$.

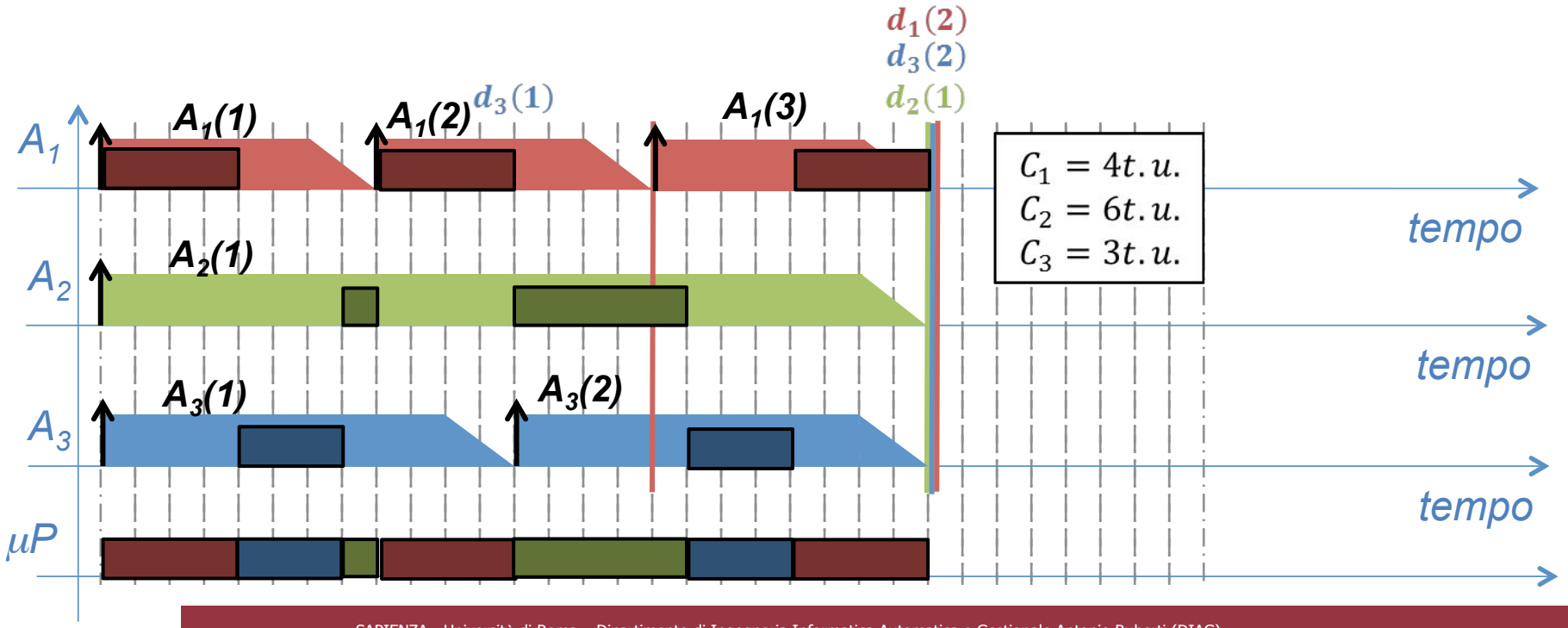


ESEMPIO cont'd

Con l'arrivo nella READY QUEUE del task $A_1(3)$, la priorità viene:

$$A_2(1) = 0{,}041\overline{6} \qquad A_3(2) = 0{,}041\overline{6} \qquad A_1(3) = 0{,}041\overline{6}$$

Quindi si termina il tracciamento dello scheduling.



Proprietà dell'algoritmo EDF

PROPOSIZIONE 1 (senza dimostrazione)

Se un insieme di task periodici **NON è schedulabile tramite EDF**, allora **NON è schedulabile tramite nessun altro algoritmo DINAMICO**.

PROPOSIZIONE 2 (senza dimostrazione)

Un qualsiasi insieme di task periodici è schedulabile tramite un algoritmo EDF se e solo se ha un fattore di utilizzazione $U \leq 1$.

RMPO vs EDF

OSSERVAZIONE 1

Dato un insieme **qualsiasi di task periodici** possiamo affermare che:

- RMPO garantisce la schedulabilità se $U \leq \ln(2) \approx 69{,}3\%$;
- EDF garantisce la schedulabilità se $U \leq 100\%$.

OSSERVAZIONE 2

Dato un insieme di **n task periodici** possiamo affermare che:

- RMPO garantisce la schedulabilità se $U \leq n\left(2^{1/n} - 1\right)$;
- EDF garantisce la schedulabilità se $U \leq 100\%$.

OSSERVAZIONE 3

Dato un insieme di task periodici con relazioni armoniche possiamo affermare che:

- RMPO garantisce la schedulabilità se $U \leq 100\%$;
- EDF garantisce la schedulabilità se $U \leq 100\%$.

ESEMPIO

PROBLEMA

Consideriamo un problema di coordinamento di task periodici dove:

- REQUISITI DI SISTEMA: ($n = 3$; $T_1 = 8$ t.u.; $T_2 = 16$ t.u.; $T_3 = 12$ t.u.)
- VINCOLI DI SISTEMA: ($C_1 = 3$ t.u.; $C_2 = 3$ t.u.; $C_3 = 5$ t.u.)

Verificare la schedulabilità tramite l'algoritmo RMPO. In caso negativo, mostrare lo scheduling dell'algoritmo EDF.

SVOLGIMENTO

Per prima cosa verifichiamo che sussista la condizione necessaria di esistenza della soluzione al problema dato, calcolando il **fattore di utilizzazione**:

$$U = \sum_{i=1}^{3} \frac{C_i}{T_i} = \frac{3}{8} + \frac{3}{16} + \frac{5}{12} = \frac{18 + 9 + 20}{48} = \frac{47}{48} \approx 0{,}979$$

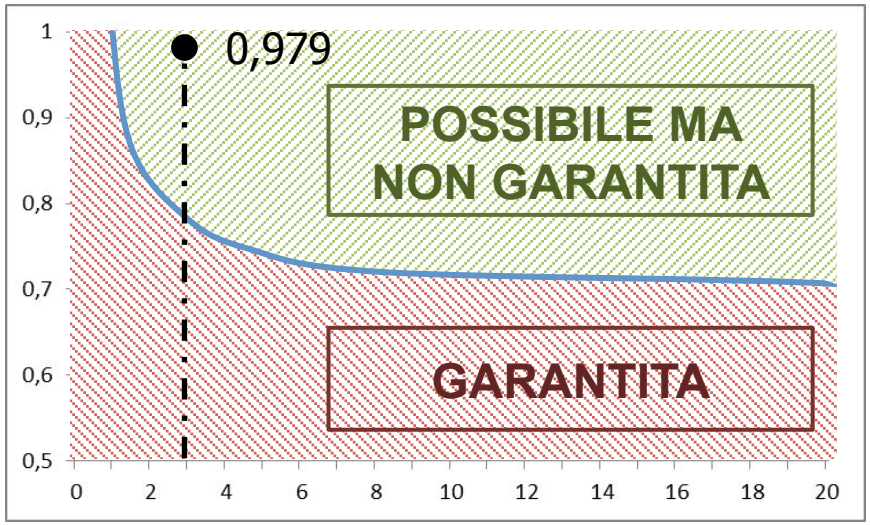
Dato che $U < 1$, possiamo escludere che il problema sia inammissibile e sicuramente è schedulabile tramite EDF.

ESEMPIO cont'd

Il limite superiore minimo del fattore di utilizzazione dell'algoritmo RMPO per un insieme di n = 3 task periodici è:

$$U_{lsm}(RMPO) \approx 0{,}78 < 0{,}979 \approx U$$

Pertanto non è garantita la schedulabilità con RMPO.



ESEMPIO cont'd

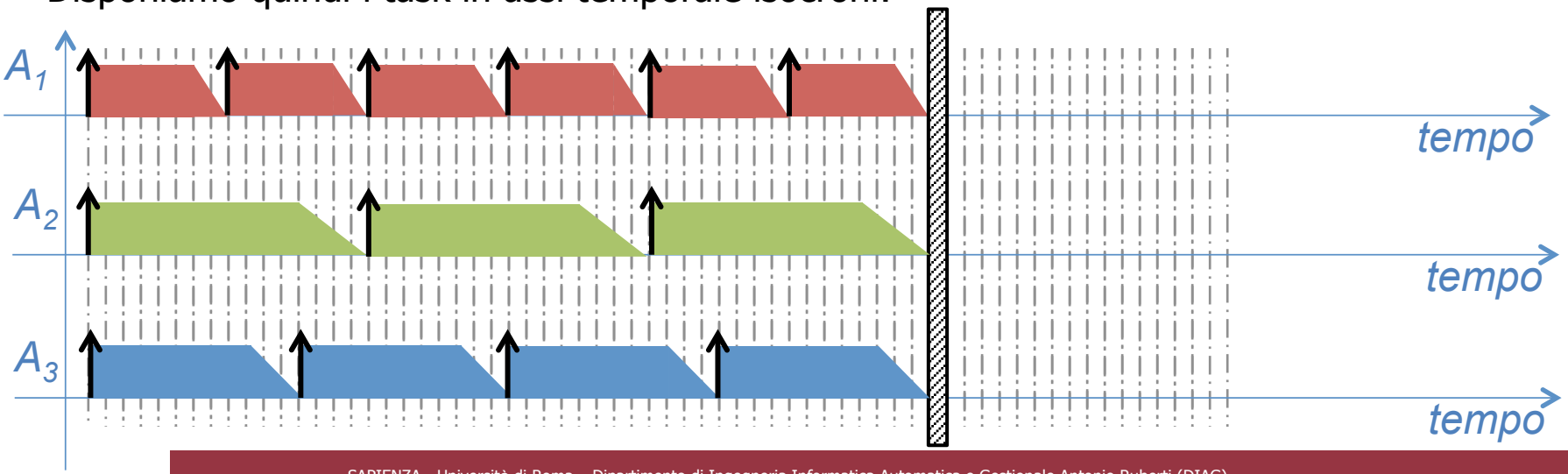
Prima di passare a tracciare passo dopo passo lo scheduling, calcoliamo le priorità statiche dei tre differenti task:

$$A_1 \rightarrow \frac{1}{8} \; ; A_2 \rightarrow \frac{1}{16} \; ; A_3 \rightarrow \frac{1}{12}$$

L'ordine di priorità è pertanto:

$$A_1 = 0{,}125 \; ; A_3 = 0{,}08\overline{3} \; ; A_2 = 0{,}0625$$

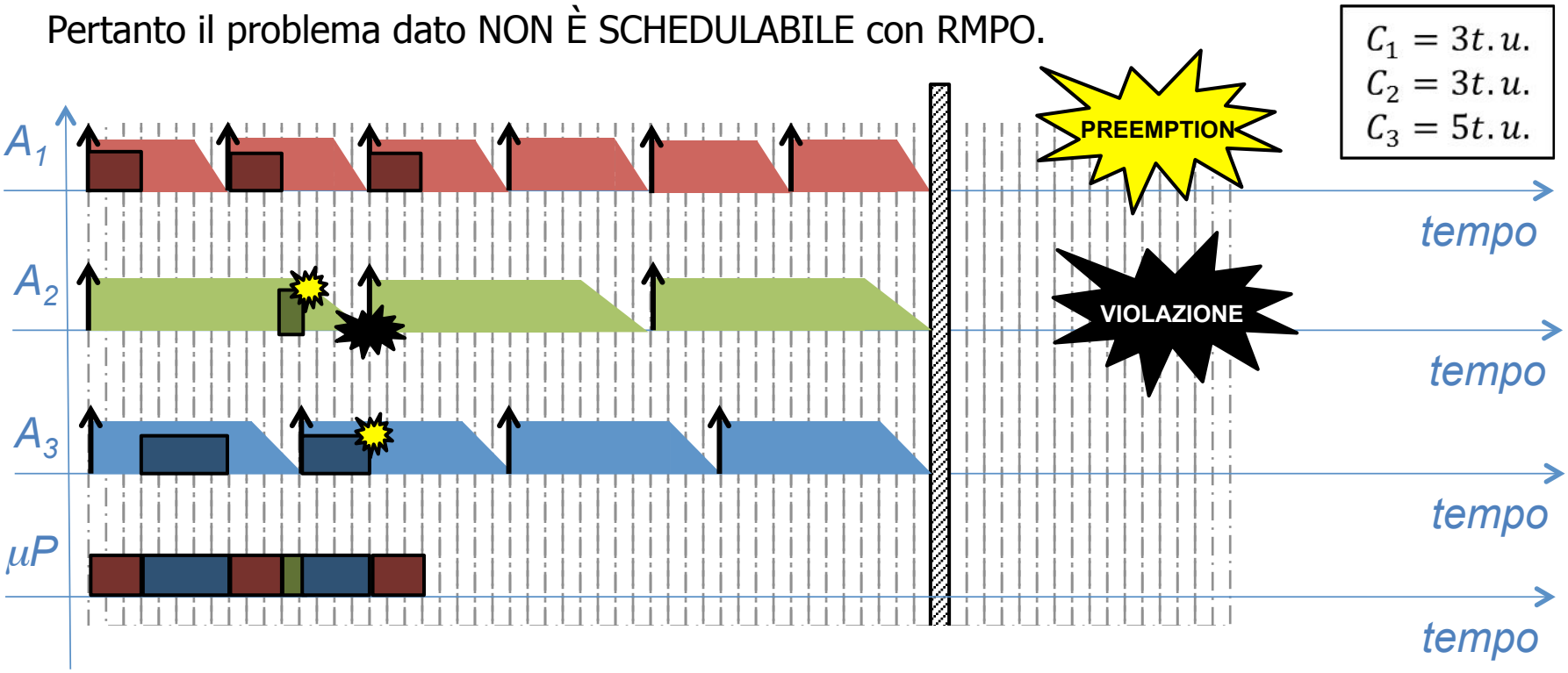
Disponiamo quindi i task in assi temporale isocroni:



ESEMPIO cont'd

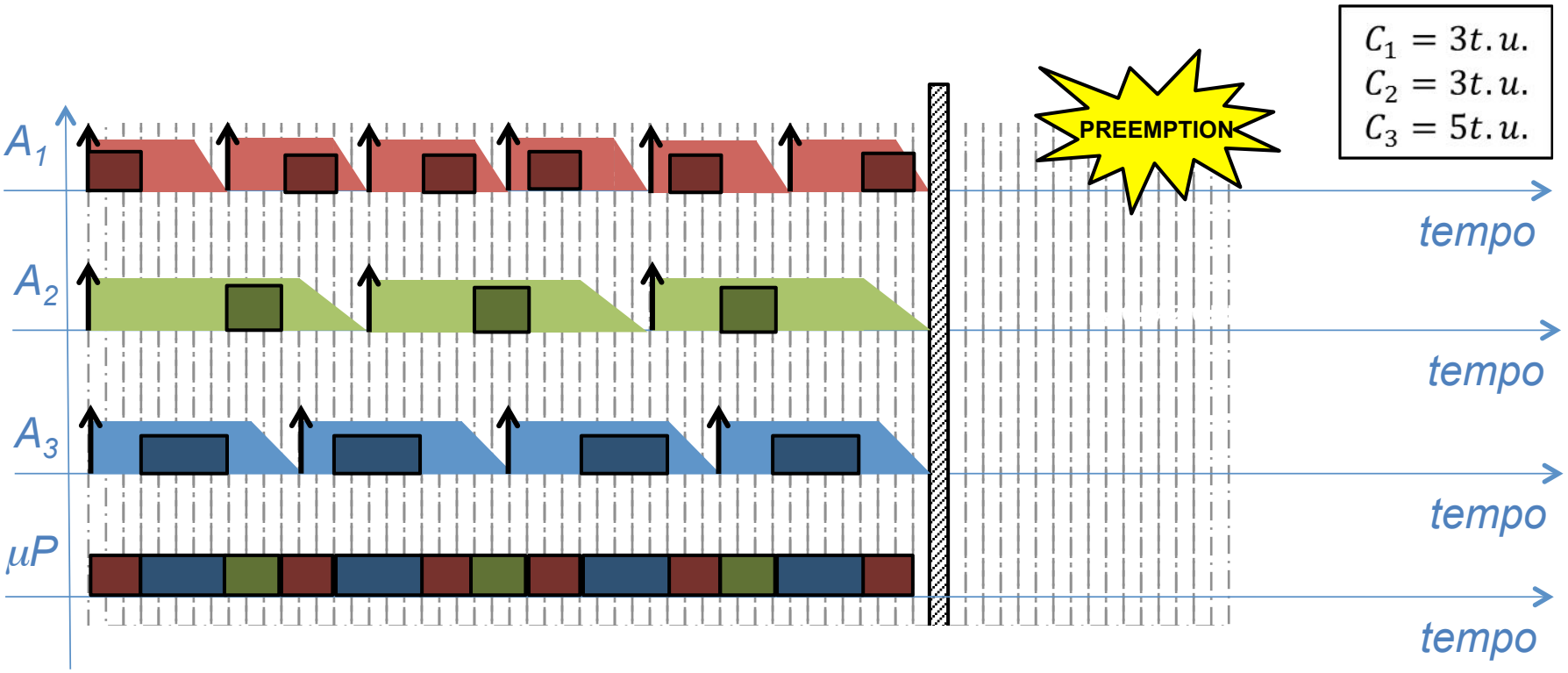
Lo scheduling RMPO risulta nella VIOLAZIONE DEL VINCOLO TEMPORALE della prima esecuzione del task A_2.

Pertanto il problema dato NON È SCHEDULABILE con RMPO.



ESEMPIO cont'd

Lo scheduling EDF risulta come in figura.



RMPO vs EDF

OSSERVAZIONE 1

- Si potrebbe dedurre che EDF garantisce la prevedibilità nei sistemi real time in maniera ottimale rispetto a RMPO.
- Tale prevedibilità però si paga in termini di COMPLESSITÀ COMPUTAZIONALE.
- L'algoritmo **EDF deve ricalcolare le priorità dei task attivi ad ogni deadline**, mentre l'algoritmo **RMPO deve calcolare tale priorità solo una volta**.

OSSERVAZIONE 2

- L'algoritmo **RMPO può essere utilizzato esclusivamente per task periodici**, in cui il periodo di esecuzione è fissato e noto a priori.
- L'algoritmo **EDF può essere utilizzato nello scheduling di task periodici o aperiodici**, in quanto la priorità di scheduling NON dipende dalla ipotesi di periodicità dei task.

BIBLIOGRAFIA

Sezione 2.4 (pagg. 53-56)

TITOLO

Sistemi di automazione industriale

Architetture e controllo

AUTORI

Claudio Bonivento

Luca Gentili

Andrea Paoli

EDITORE

McGraw-Hill