

# Automazione

17 Gennaio 2019

## Esercizio 1

Una linea di produzione deve gestire cinque prodotti  $\{A, B, C, D, E\}$  che necessitano di due lavorazioni complesse in sequenza, la prima effettuata dalla macchina #1 e la seconda dalla macchina #2. In Tab. 1 sono riportati i tempi *nominali* (in ore) necessari per le lavorazioni dei diversi prodotti su ciascuna macchina, con una possibile *incertezza* congiunta sulla tempistica di due lavorazioni. Tale incertezza è descritta dal parametro  $\alpha \in [0, 1.5]$  (anch'esso espresso in ore).

- Sequenziare inizialmente le lavorazioni sulle due macchine in condizioni nominali ( $\alpha = 0$ ), in modo da minimizzare il tempo totale  $T$  di completamento (*makespan*) dei cinque prodotti, descrivendo i singoli passi dell'algoritmo applicato.
- Studiare quindi il problema in condizioni perturbate. Qual è il valore massimo che può assumere  $\alpha$  nel suo intervallo di incertezza in modo che il tempo di completamento  $T$  rimanga *invariato* e il sequenziamento ottimo sia *identico* a quello del caso nominale? Potendo invece stabilire un eventuale *diverso* sequenziamento delle lavorazioni, qual è il valore massimo di  $\alpha$  sempre a parità di  $T$ ?

Riportare tutti i risultati graficamente con i diagrammi di Gantt.

Prodotto	A	B	C	D	E
Macchina #1	4	1	$5 + \alpha$	2	5
Macchina #2	3	$2 - \alpha$	4	3	6

Tabella 1: Tempi di lavorazione nominali, con incertezza sulle durate di  $t_{C1}$  e  $t_{B2}$ .

## Esercizio 2

La matrice di incidenza  $C$  e la marcatura iniziale  $\mathbf{x}_0$  di una rete di Petri  $PN$  sono date da

$$C = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{x}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \quad (1)$$

- A quale particolare classe appartiene la rete  $PN$ ? Studiare le proprietà strutturali di limitatezza, vivezza, reversibilità e conservatività della rete, costruendo l'albero di raggiungibilità e definendo i  $P$ -invarianti e i  $T$ -invarianti.
- Se si associa ai posti  $p_5$  e  $p_6$  il significato di due operazioni  $A$  e  $B$  in esecuzione, definire un opportuno supervisore in modo che l'operazione  $B$  sia sempre eseguita dopo l'operazione  $A$ .

### Esercizio 3

La Fig. 1 mostra l'accesso automatizzato di un ufficio postale (quello di P.zza Dante!). Il cliente che vuole entrare nell'ufficio preme un pulsante sulla porta esterna, mentre chi vuole uscire dall'ufficio si posiziona nelle vicinanze della porta interna (dove è posto un sensore di prossimità). Tali richieste sono servite solo se entrambe le porte sono chiuse e la cabina è libera (altrimenti vengono annullate). Quando una richiesta viene servita, la prima porta si apre per poi richiudersi alle spalle di chi è entrato nella cabina. Si apre quindi la seconda porta, che si chiude quando il cliente sarà uscito dalla cabina. Una volta entrati in cabina, non si può tornare indietro.

Modellare il comportamento di questo sistema di accesso automatizzato con un automa a stati finiti (senza duplicazioni di stati identici), descrivendo il significato e la composizione degli stati e gli eventi che portano alle transizioni di stato.



Figura 1: L'accesso automatizzato di un ufficio postale.

### Esercizio 4

Si consideri un sistema di automazione in cui, a livello di coordinamento, è necessario portare a termine i seguenti task periodici:

1. ogni 6 t.u. una lastra metallica deve essere movimentata impiegando 1 t.u.;
2. ogni 10 t.u. una lastra deve essere pressata impiegando 4 t.u.

A questi task si aggiunge un task aperiodico di verifica di qualità che deve intervenire con un intervallo minimo di 15 t.u. e durare al massimo 3 t.u.

Si ipotizzi che i task suddetti siano indipendenti l'uno dall'altro. I tre suddetti task devono essere gestiti con una modalità di scheduling hard real time.

Il sistema di automazione viene supervisionato da un tecnico. Il task di interfacciamento uomo-macchina è aperiodico e così caratterizzato: l'activation time della prima occorrenza è all'istante  $a_4(1) = 15$  t.u., il computation time è pari a  $C_4(1) = 9$  t.u. e la deadline assoluta è pari a  $d_4(1) = 50$  t.u. Questo task deve essere servito con una modalità di scheduling soft real time.

Rispondere argomentando alle seguenti domande.

1. Se per lo scheduling hard real time si utilizza RMPO e per lo scheduling soft real time si adotta una servizio in BACKGROUND, il task aperiodico riesce a essere eseguito entro la deadline assoluta?
2. Se per lo scheduling hard real time si utilizza RMPO e per lo scheduling soft real time si adotta una politica DEFERRING SERVER (avente  $T_{srv} = 5$  t.u. e  $C_{srv} = 1$  t.u.), il task aperiodico riesce a essere eseguito entro la deadline assoluta?
3. Se per lo scheduling hard real time si utilizza EDF e per lo scheduling soft real time si adotta una politica DEFERRING SERVER (avente  $T_{srv} = 30$  t.u. e  $C_{srv} = 7$  t.u.), il task aperiodico riesce a essere eseguito entro la deadline assoluta?

[180 minuti; libri aperti]

# Soluzioni

17 Gennaio 2019

## Esercizio 1

Si tratta di un problema di flow shop scheduling con  $n = 5$  lavori su  $m = 2$  macchine, per il quale nei vari casi è possibile applicare l'algoritmo di Johnson che minimizza il tempo totale di completamento (*makespan*).

In condizioni nominali ( $\alpha = 0$ ), l'algoritmo processa per primo il prodotto  $i$  ( $i = A, \dots, E$ ) con il minimo tempo di lavorazione  $t_{ij}$ . In questo caso si tratta di  $i = B$  (il minimo tempo nella tabella è  $t_{B1} = 1$ ). Poiché  $t_{B1} = 1 < 2 = t_{B2}$ , il prodotto  $B$  verrà schedato per primo. Rimosso  $B$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{D1} = 2$ . Poiché  $t_{D1} = 2 < 3 = t_{D2}$ , il prodotto  $D$  verrà posto come secondo, ossia immediatamente dopo il primo. Rimosso anche  $D$  dalla lista dei prodotti, il minimo tempo di lavorazione sui restanti è  $t_{A2} = 3$ . Poiché  $t_{A1} = 4 > 3 = t_{A2}$ , il prodotto  $A$  verrà schedato per ultimo. Tra i due rimanenti lavori, quello con il minore tra i tempi di lavorazione è  $C$ , con  $t_{C2} = 4$ . Poiché  $t_{C1} = 5 > 4 = t_{C2}$ , il prodotto  $C$  verrà sequenziato come penultimo, ossia come quarto. Di conseguenza  $E$  sarà in terza posizione. La soluzione ottima fornita dall'algoritmo in condizioni nominale è quindi  $\{B, D, E, C, A\}$ , illustrata nel diagramma di Gantt in Fig. 2.

Il makespan è pari a  $T = 21$  ore. Si noti che questo sequenziamento ottimo è unico (non è stato infatti necessario operare scelte tra tempi di lavorazione uguali). Le somme dei tempi delle lavorazioni sulla macchina #1 e sulla #2 sono pari rispettivamente a 17 ore e 18 ore. La prima macchina rimane quindi a riposo nelle ultime 4 ore. La seconda macchina rimane in attesa per un totale di 3 ore, un'ora all'inizio della sequenza e due ore in attesa del termine della lavorazione del prodotto  $E$  da parte della prima macchina.

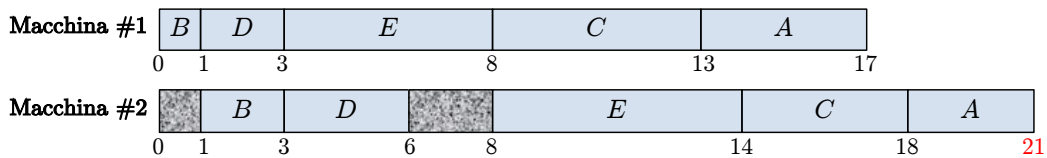


Figura 2: Diagramma di Gantt con la soluzione nominale ( $\alpha = 0$ ) del flow shop scheduling.

Introduciamo ora una perturbazione temporale  $\alpha > 0$  che allunga il tempo della lavorazione  $C$  sulla prima macchina ( $t_{C1} = 5 + \alpha$ ) e riduce quello della lavorazione  $B$  sulla seconda ( $t_{B2} = 2 - \alpha$ ).

Volendo ottenere un makespan uguale a quello del caso nominale, mantenendo anche lo stesso sequenziamento dei lavori, è evidente dalla Fig. fig:gantt che il limite è fornito dalla necessità di non introdurre un ritardo dell'istante di partenza della lavorazione  $C$  sulla seconda macchina, che deve rimanere invariato a  $t = 14$  ore. Non ci sono infatti più intervalli di attesa da lì in poi nel sequenziamento nominale. Il valore massimo tollerabile in queste condizioni è quindi  $\alpha = 1$  ora. Il nuovo diagramma di Gantt dello scheduling è riportato in Fig. 3.

La prima macchina ha ovviamente allungato di un'ora il suo tempo di completamento (essendo diventato più lungo il tempo di lavorazione di  $C$  sulla prima macchina,  $t_{C1} = 5 + \alpha = 6$  ore). La seconda macchina rimane in attesa per un totale di 4 ore, avendo aggiunto un'ora ulteriore in attesa del termine della lavorazione del prodotto  $D$  da parte della prima macchina (la lavorazione  $B$  è diventata più rapida sulla seconda macchina,  $t_{B2} = 2 - \alpha = 1$  ora). Questo però non ha alcun effetto sul makespan che è rimasto pari a quello nominale,  $T = 21$  ore.

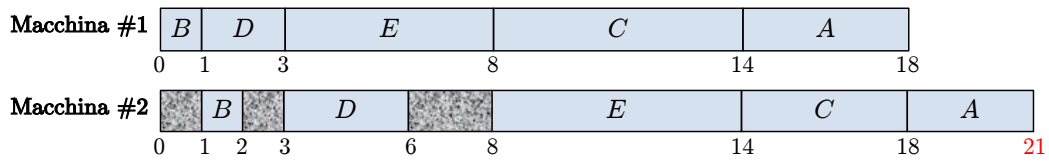


Figura 3: Diagramma di Gantt con la soluzione del problema di flow shop con  $\alpha = 1$ , il valore massimo per il quale l'algoritmo di Johnson fornisce lo stesso makespan ottimo del caso nominale mantenendo anche lo stesso sequenziamento dei lavori.

Portando la durata della lavorazione  $C$  sulla macchina #1 al valore massimo ammissibile nell'intervallo di definizione,  $\alpha = 1.5$  ore, si ottiene  $t_{C1} = 5 + \alpha = 6.5$  ore e quindi  $t_{B2} = 2 - \alpha = 0.5$  ore. La soluzione ottima fornita dall'algoritmo di Johnson è in tal caso  $\{D, E, C, A, B\}$  e il sequenziamento è illustrato in Fig. 4. Il makespan è rimasto comunque invariato a  $T = 21$  ore.

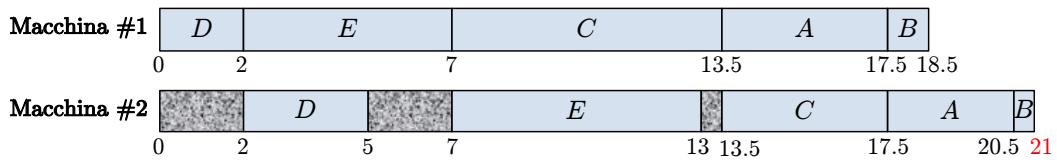


Figura 4: Diagramma di Gantt con la soluzione ottima del problema di flow shop con  $\alpha = 1.5$ , il valore massimo nell'intervallo di definizione. Il sequenziamento dei lavori fornito dall'algoritmo di Johnson è diverso da quello delle Figg. 2 e 3.

### Esercizio 2

A partire dalla matrice di incidenza  $C$  e dalla marcatura iniziale  $\mathbf{x}_0$  in (1), ricaviamo la rete di Petri ordinaria (avente solo archi di peso unitario) con 6 posti e 7 transizioni riportata nella Fig. 5. Per costruzione non ci sono posti contemporaneamente in ingresso e in uscita alla stessa transizione (auto-anelli).

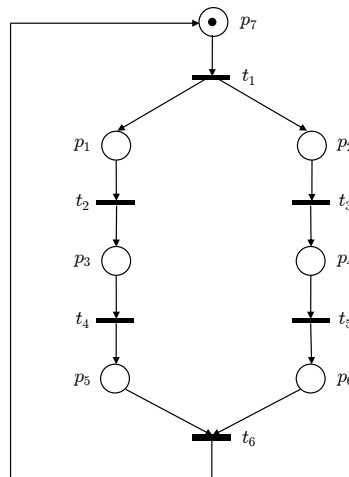


Figura 5: Rete di Petri corrispondente alla matrice di incidenza  $C$  in (1).

- E' facile verificare che si tratta di una rete appartenente alla sotto-classe *Marked Graph* delle reti di Petri ordinarie, poiché ogni posto ha una sola transizione in ingresso e una sola in uscita. La vivezza sussiste quindi se e solo se ogni ciclo nel grafo di Petri contiene almeno un posto marcato. Ciò si verifica immediatamente osservando che il posto (marchato)  $P_7$  fa parte degli unici due cicli presenti nella rete ( $P_1 \rightarrow P_3 \rightarrow P_5 \rightarrow P_7 \rightarrow P_1$  e  $P_2 \rightarrow P_4 \rightarrow P_6 \rightarrow P_7 \rightarrow P_2$ ).
- L'albero di raggiungibilità della rete di Petri è riportato in Fig. 6. L'albero deve essere costruito con una logica 'depth first' e si è considerato l'ordine lessicografico delle etichette nell'espansione delle transizioni abilitate: il ramo che parte dalla transizione  $t_2$  nella marcatura  $x_1 = (1\ 1\ 0\ 0\ 0\ 0\ 0)$  si espande quindi prima di quello di  $t_3$  e così via.

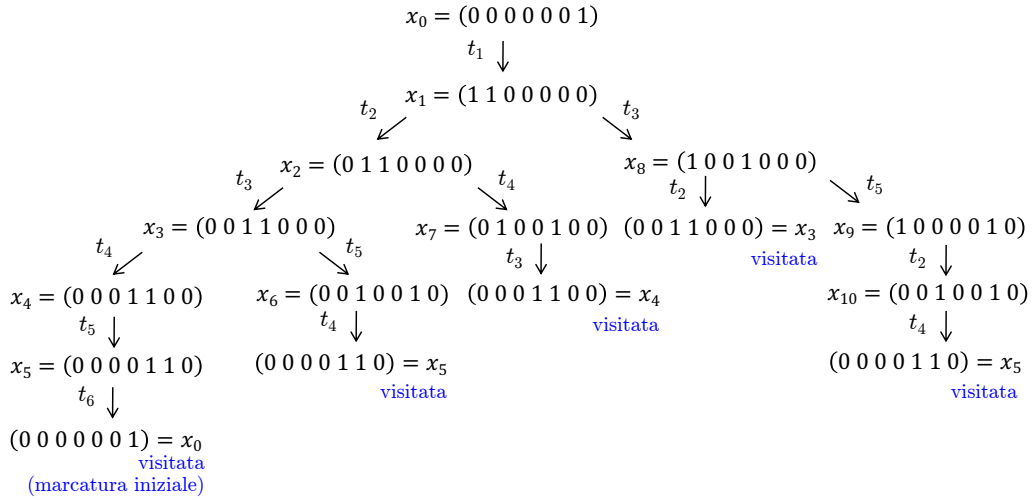


Figura 6: Albero di raggiungibilità della rete di Petri di Fig. 5.

- Analizzando l'albero di raggiungibilità si può verificare che la rete è viva, reversibile e 2-limitata.
- La matrice di incidenza in (1) della rete ha rango  $\rho = 5$ , come si può facilmente verificare: la terza riga è proporzionale alla seconda, la quinta alla quarta, la matrice quadrata ottenuta eliminando terza e quinta riga ha determinante nullo ed esiste (almeno) un minore  $3 \times 3$  non nullo. Lo spazio nullo di  $C^T$  ha quindi dimensione  $|P| - \rho = 6 - 3 = 3$ , mentre lo spazio nullo di  $C$  è mono-dimensionale ( $|T| - \rho = 4 - 3 = 1$ ).
- Il calcolo dei  $P$ -invarianti

$$\gamma^T C = \mathbf{0}^T \Rightarrow \begin{cases} \gamma_1 + \gamma_2 - \gamma_4 = 0 \\ \gamma_3 - \gamma_1 = 0 \\ \gamma_4 - \gamma_2 = 0 \\ \gamma_5 - \gamma_3 = 0 \\ \gamma_6 - \gamma_4 = 0 \\ \gamma_7 - \gamma_6 - \gamma_5 = 0 \end{cases} \Rightarrow \begin{cases} \gamma_1 = \gamma_3 = \gamma_5 \\ \gamma_2 = \gamma_4 = \gamma_6 \\ \gamma_7 = \gamma_5 + \gamma_6 \end{cases}$$

fornisce (avendo scelto rispettivamente  $\{\gamma_5 = 1, \gamma_6 = 0\}$  oppure  $\{\gamma_5 = 0, \gamma_6 = 1\}$ ) le due soluzioni linearmente indipendenti, a supporto minimo e canoniche:

$$\gamma^1 = (1\ 0\ 1\ 0\ 1\ 0\ 1)^T \quad \gamma^2 = (0\ 1\ 0\ 1\ 0\ 1\ 1)^T.$$

Poichè ogni posto compare nel insieme di supporto di almeno un  $P$ -invariante, la rete risulta conservativa e pertanto limitata, a riprova dell'analisi effettuata con l'albero di raggiungibilità.

- Le equazioni di invarianza della rete (della forma  $\gamma^T \mathbf{x} = \gamma^T \mathbf{x}_0$ ) sono

$$\begin{aligned} x(p_1) + x(p_3) + x(p_5) + x(p_7) &= 1 \\ x(p_2) + x(p_4) + x(p_6) + x(p_7) &= 1. \end{aligned}$$

A queste equazioni sono associati *cicli minimi* della rete nei quali si conserva il numero di token complessivi presenti inizialmente. Tali cicli sono quelli già individuati in precedenza e contengono entrambi 1 token negli insiemi dei posti  $C^1 = \{p_1, p_3, p_5, p_7\}$  e  $C^2 = \{p_2, p_4, p_6, p_7\}$ . Si noti che a  $P$ -invarianti non a supporto minimo corrispondono cicli composti.

- Il calcolo dei  $T$ -invarianti fornisce

$$C \boldsymbol{\eta} = \mathbf{0} \Rightarrow \begin{cases} \eta_1 - \eta_2 = 0 \\ \eta_1 - \eta_3 = 0 \\ \eta_2 - \eta_4 = 0 \\ \eta_3 - \eta_5 = 0 \\ \eta_4 - \eta_6 = 0 \\ \eta_5 - \eta_6 = 0 \\ \eta_6 - \eta_1 = 0 \end{cases} \Rightarrow \eta_1 = \eta_2 = \eta_3 = \eta_4 = \eta_5 = \eta_6.$$

Pertanto, come previsto, esiste un solo  $T$ -invariante canonico:

$$\boldsymbol{\eta} = (1 \ 1 \ 1 \ 1 \ 1 \ 1)^T.$$

Come vettore delle occorrenze,  $\boldsymbol{\eta}$  è associato a diverse sequenze ammissibili di scatti, ad esempio a  $\{t_1, t_2, t_4, t_3, t_5, t_6\}$  oppure  $\{t_1, t_3, t_2, t_5, t_4, t_6\}$ . In ogni caso, in ogni sequenza che rende la rete reversibile, tutte le transizioni scattano una e una sola volta (o un numero uguale di volte).

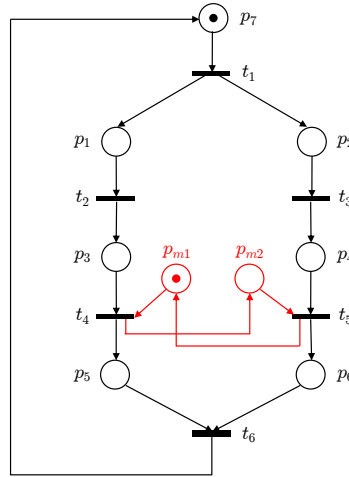


Figura 7: La rete di Petri di Fig. 5 con l'aggiunta di un supervisore con due posti monitor.

Infine, se si vuole che un token raggiunga (in ogni ciclo) il posto  $p_5$  prima del posto  $p_6$ , il che si interpreta con la richiesta che l'operazione  $B$  (in  $p_6$ ) sia sempre eseguita dopo l'operazione  $A$

(in  $p_5$ ) si può definire un supervisore avente due posti monitor,  $p_{m1}$  e  $p_{m2}$ , il primo con un token ( $x(p_{m1}) = 0$ ) e il secondo senza ( $x(p_{m2}) = 0$ ) come in Fig. 7. I due posti monitor implementano la precedenza richiesta, eliminando le marcature che non sono ammissibili per la specifica e senza bloccare la rete.

### Esercizio 3

L'automata che descrive il comportamento dell'accesso automatizzato è mostrato in Fig. 8. Il canale di sinistra rappresenta l'ingresso di un cliente, quello di destra l'uscita.

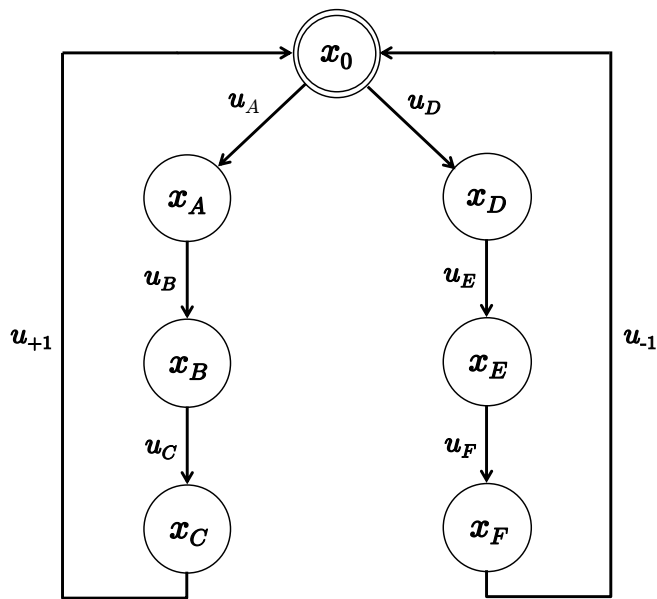


Figura 8: Automa che rappresenta l'accesso automatizzato in Fig. 1.

Per caratterizzare gli stati, utilizziamo quattro variabili, tre binarie e una a tre valori.

- cabina = {occupata, libera}
- porta esterna = {aperta, chiusa}
- porta interna = {aperta, chiusa}
- richiesta in corso = {ingresso, uscita, nessuna}

Le combinazioni di tali valori forniscono  $3 \cdot 2^3 = 24$  possibili combinazioni (stati), delle quali solo 7 sono ammissibili. Questi sono i 7 stati dell'automata:

- $x_0$  = cabina libera, porte esterna e interna chiuse, nessuna richiesta in corso (stato di attesa)
- $x_A$  = cabina libera, porta esterna aperta, porta interna chiusa, richiesta di ingresso in corso
- $x_B$  = cabina occupata, porta esterna chiusa, porta interna chiusa, richiesta di ingresso in corso
- $x_C$  = cabina occupata, porta esterna chiusa, porta interna aperta, richiesta di ingresso in corso
- $x_D$  = cabina libera, porta interna aperta, porta esterna chiusa, richiesta di uscita in corso
- $x_E$  = cabina occupata, porta interna chiusa, porta esterna chiusa, richiesta di uscita in corso
- $x_F$  = cabina occupata, porta interna chiusa, porta esterna aperta, richiesta di uscita in corso

Gli 8 eventi che comportano una transizione di stato sono:

- $u_A$  = richiesta di ingresso di un cliente (premendo il pulsante esterno)

- $u_B$  = cliente entra nella cabina dall'esterno (si rileva un peso sul pavimento della cabina)
- $u_C$  = acknowledgement di ingresso (eventualmente dopo un certo tempo di attesa)
- $u_{+1}$  = cliente esce dalla cabina, entrando nell'ufficio postale (un cliente in più nell'ufficio)
- $u_D$  = richiesta di uscita di un cliente (posizionandosi nell'area del sensore interno)
- $u_E$  = cliente entra nella cabina dall'interno (si rileva un peso sul pavimento della cabina)
- $u_F$  = acknowledgement di uscita (eventualmente dopo un certo tempo di attesa)
- $u_{-1}$  = cliente esce dalla cabina, uscendo dall'ufficio postale (un cliente in meno nell'ufficio)

Qualsiasi altro evento non abilita alcuna transizione di stato. Non sono quindi riportati nell'automa gli eventi non abilitati nei vari stati (eventualmente, sarebbero degli auto-anelli).

#### Esercizio 4

I due task periodici ( $A_1$  e  $A_2$ ) e il task aperiodico di verifica di qualità ( $A_3$ ) devono essere trattati con una politica hard real time, pertanto si genera un problema equivalente di scheduling di task periodici così dato:

$$T_1 = 6, C_1 = 1; \quad T_2 = 10, C_2 = 4; \quad T_3 = 15, C_3 = 3.$$

**Punto 1.** Il coefficiente di utilizzazione  $U$  del problema equivalente di scheduling di task periodici vale  $23/30 = 0.77$  ed è minore di  $U_{lsm}(3) \simeq 0.780$ . Pertanto, RMPO è sicuramente soluzione del problema dato e, infatti, lo scheduling è mostrato in Fig. 9.

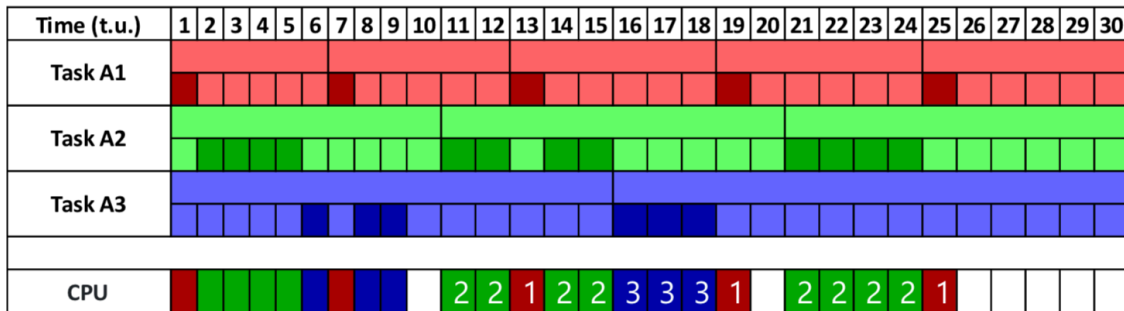


Figura 9: Scheduling RMPO per i tre task da trattare in hard real time.

Il task aperiodico di interfacciamento uomo-macchina soft real time ( $A_4$ ) viene attivato all'istante  $a_4(1) = 15$  t.u. e ha una deadline assoluta pari a  $d_4(1) = 50$  t.u. Con riferimento alla Fig. 10, è facile verificare che il task  $A_4$ , servito in BACKGROUND, riesce ad eseguire solo 8 delle 9 t.u. e quindi non può essere eseguito entro la deadline assoluta.

**Punto 2.** Adottando una politica di DEFERRING SERVER per il trattamento del task aperiodico soft real time  $A_4$ , caratterizzata da un processo server con  $T_{srv} = 5$  t.u. e  $C_{srv} = 1$  t.u., il problema di scheduling diventa

$$T_1 = 6, C_1 = 1; \quad T_2 = 10, C_2 = 4; \quad T_3 = 15, C_3 = 3; \quad T_{srv} = 5, C_{srv} = 1.$$

Il coefficiente di utilizzazione  $U$  vale  $29/30 = 0.97$  ed è maggiore di  $U_{lsm}(4) \simeq 0.757$ . Inoltre, non sussistendo una relazione armonica tra i task, nessuna delle condizioni sufficienti di schedulabilità secondo RMPO è verificata. E' necessario effettuare lo scheduling per sapere se il problema ammette soluzione.



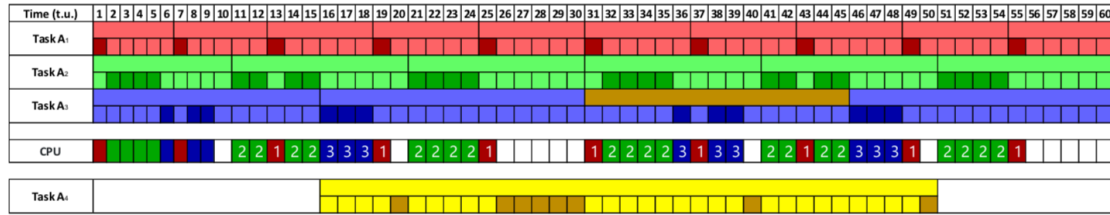


Figura 10: Scheduling RMPO dei task hard real time, con servizio in BACKGROUND del task soft real time.

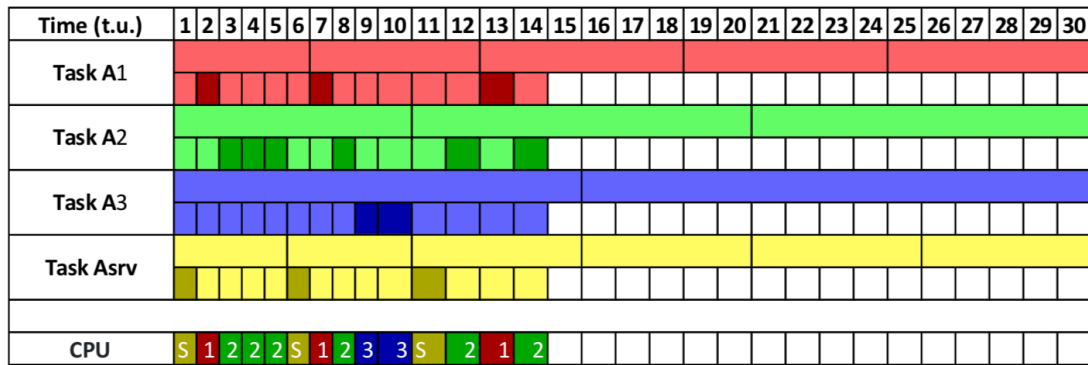


Figura 11: Scheduling RMPO dei task hard real time, con politica DEFERRING SERVER del task soft real time.

Come mostrato in Fig. 11, nel corso della t.u. 15, la prima istanza del task  $A_3$ , che ha minor priorità rispetto al task  $A_2$ , deve cedere il passo alla seconda istanza del task  $A_2$ . Così facendo, supera però la deadline senza aver terminato il task. Lo scheduling RMPO non è quindi in grado di risolvere il problema.

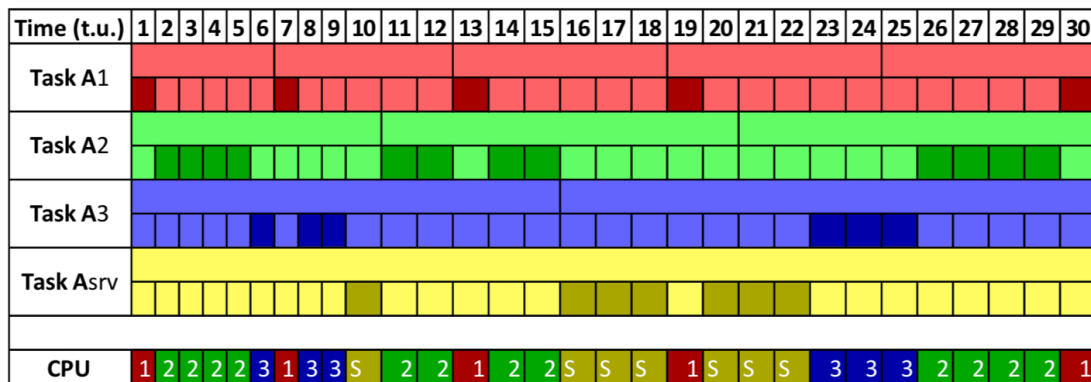


Figura 12: Scheduling EDF.

**Punto 3.** Adottando una politica di DEFERRING SERVER per il trattamento del task aperiodico soft real time  $A_4$ , caratterizzata da un processo server con  $T_{srv} = 30$  t.u. e  $C_{srv} = 7$  t.u., il problema di scheduling diventa

$$T_1 = 6, C_1 = 1; \quad T_2 = 10, C_2 = 4; \quad T_3 = 15, C_3 = 3; \quad T_{srv} = 30, C_{srv} = 7.$$

Il coefficiente di utilizzazione  $U$  vale 1 e quindi sussiste la condizione necessaria e sufficiente per schedulare il problema dato utilizzando EDF. Il risultato dello scheduling è riportato in Fig. 12.

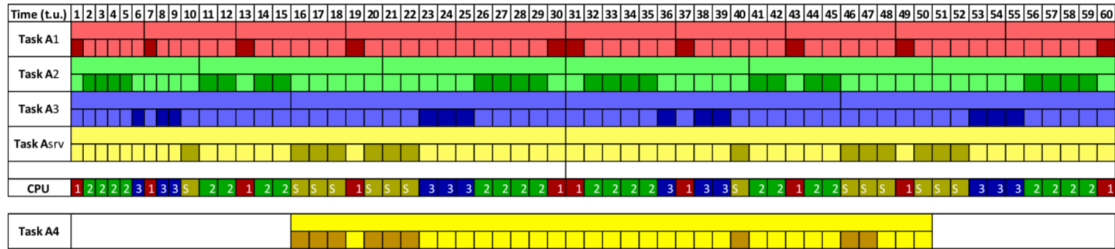


Figura 13: Scheduling EDF dei task hard real time, con politica DEFERRING SERVER per il task soft real time.

E' facile verificare che, in questo caso, il task  $A_4$ , riesce ad essere eseguito entro la deadline assoluta (Fig. 13).

\* \* \* \* \*