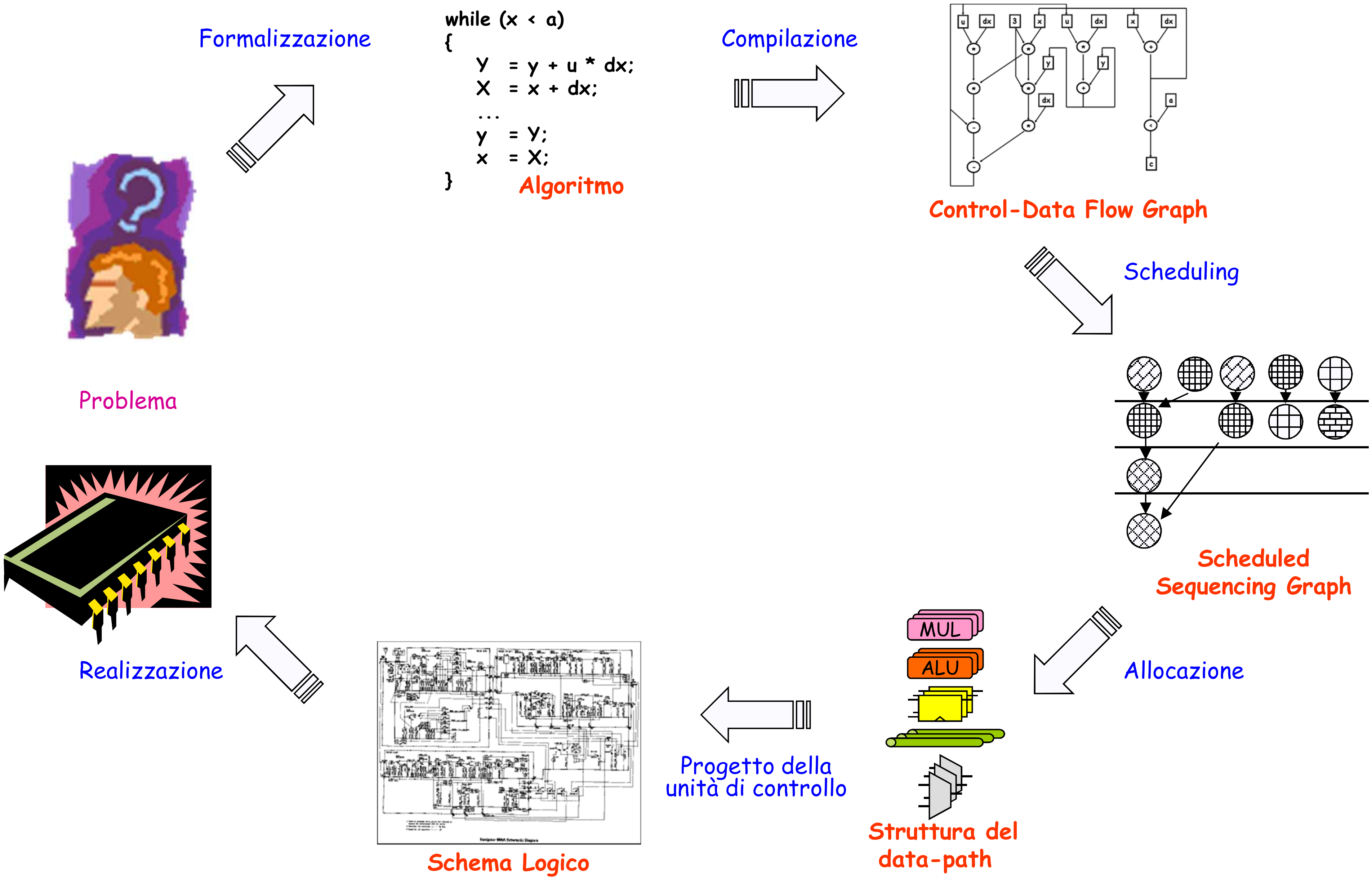
Sintesi a livello architetturale

[1] D. Gajski et al., *High-Level Synthesis: Introduction to Chip and System Design*, Kluwer Academic Press, 1992.
[2] G. De Micheli, *Synthesis and Optimization of Digital Circuits*, McGraw-Hill, 1994.
[3] F. Fummi et al., *Progettazione digitale*, McGraw-Hill, 2002.

Il processo di sintesi



Un classico caso di studio (Paulin & Knight, '89)

Integrazione numerica dell'Equazione Differenziale (IED)

$$y'' + 3\, y'\, x + 3\, y = 0$$

nell'intervallo [0, a] , passo di integrazione dx, valori iniziali x(0), y(0), y'(0)

Descrizione algoritmica del processo di elaborazione ("Forward Euler Method"):

$$y' = u; \quad u' = y'' = -\,3\, u\, x - 3\, y$$

$$dy = u\, dx; \quad du = -\,3\, u\, x\, dx - 3\, y\, dx$$

Dati di ingresso:
a
dx
x(0) = x (< a)
y(0) = y
y'(0) = u

```
do
{
    X  = x + dx;
    U  = u - 3 * u * x * dx - 3 * y * dx;
    Y  = y + u * dx;
    c  = (X < a)
    x  = X;
    u  = U;
    y  = Y;
} while (c)
```

Risultato in uscita:
y

Scomposizione delle operazioni complesse in operazioni elementari

una possibile soluzione:

$$U = u - \underbrace{3 * u}_{t_1} * \underbrace{x * dx}_{t_2} - \underbrace{3 * y}_{t_3} * dx;$$

$$U = u - \underbrace{t_1 * t_2}_{t_4} - \underbrace{t_3 * dx}_{t_5};$$

$$U = \underbrace{u - t_4}_{t_6} - t_5;$$

$$U = t_6 - t_5;$$

$$Y = y + \underbrace{u * dx}_{t_7};$$

$$Y = y + t_7;$$

Descrizione algoritmica del processo di elaborazione in termini di operazioni elementari unarie (ad un operando) e binarie (a due operandi)

```
do
{
    t1 = 3  * u;    /* O1  */
    t2 = x  * dx;   /* O2  */
    t3 = 3  * y;    /* O3  */
    t4 = t1 * t2;   /* O4  */
    t5 = t3 * dx;   /* O5  */
    t6 = u  - t4;   /* O6  */
    t7 = u  * dx;   /* O7  */
    U  = t6 - t5;   /* O8  */
    Y  = y  + t7;   /* O9  */
    X  = x  + dx;   /* O10 */
    c  = (X < a);   /* O11 */
    x  = X;
    u  = U;
    y  = Y;
} while c;
```

potenziale parallelismo di esecuzione di operazioni mutuamente indipendenti:

$\{ O_1 , O_2 , O_3 , O_7 , O_{10} \}$

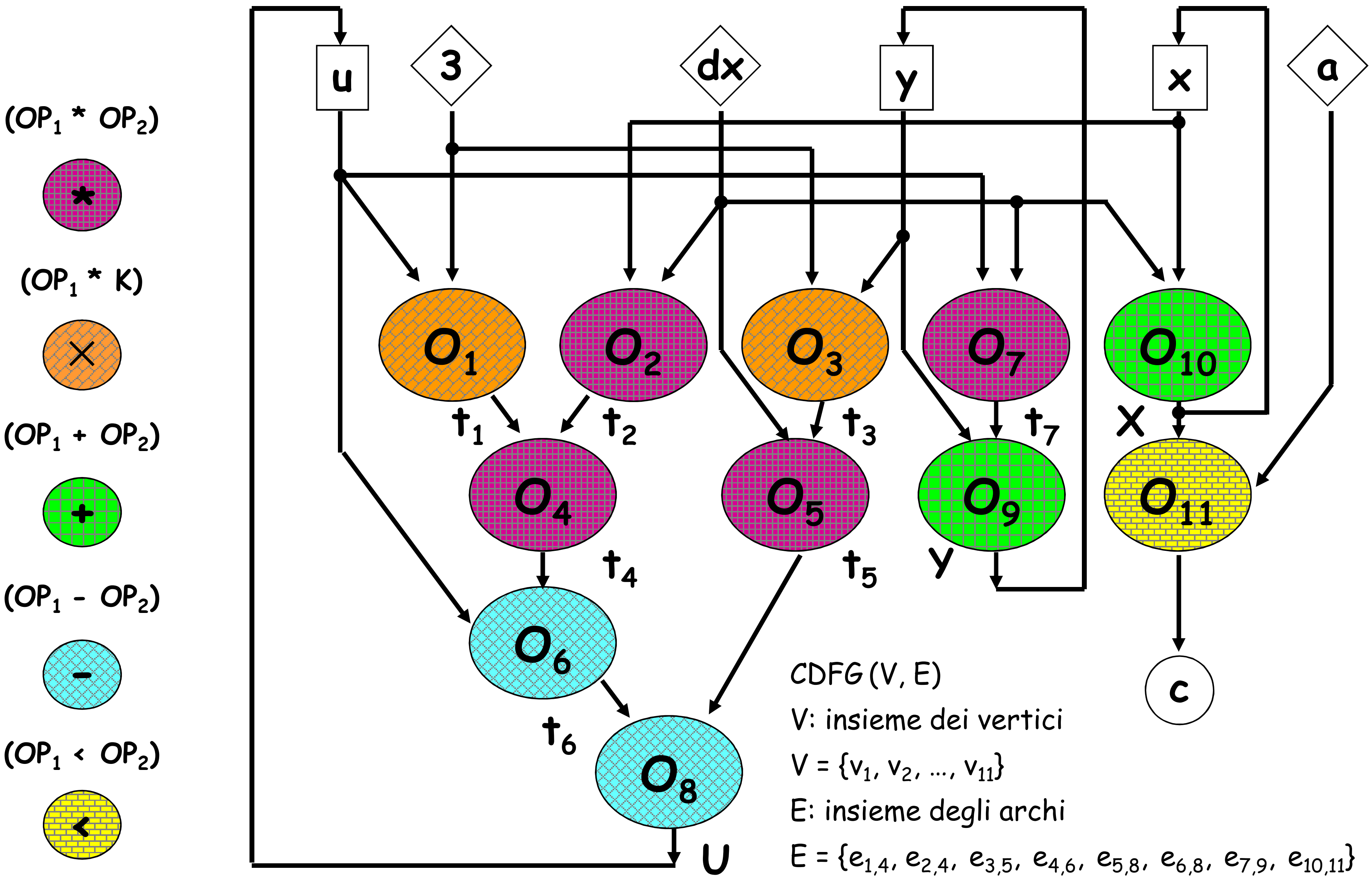
$\{ O_4 , O_5 , O_9 , O_{11} \}$

$\{ O_6 \}$

$\{ O_8 \}$

Rappresentazione in termini di "Control-Data Flow Graph"

Grafo in cui ad ogni operazione corrisponde un vertice
e ad ogni vincolo di precedenza tra operazioni un arco orientato



CDFG (V, E)

V: insieme dei vertici

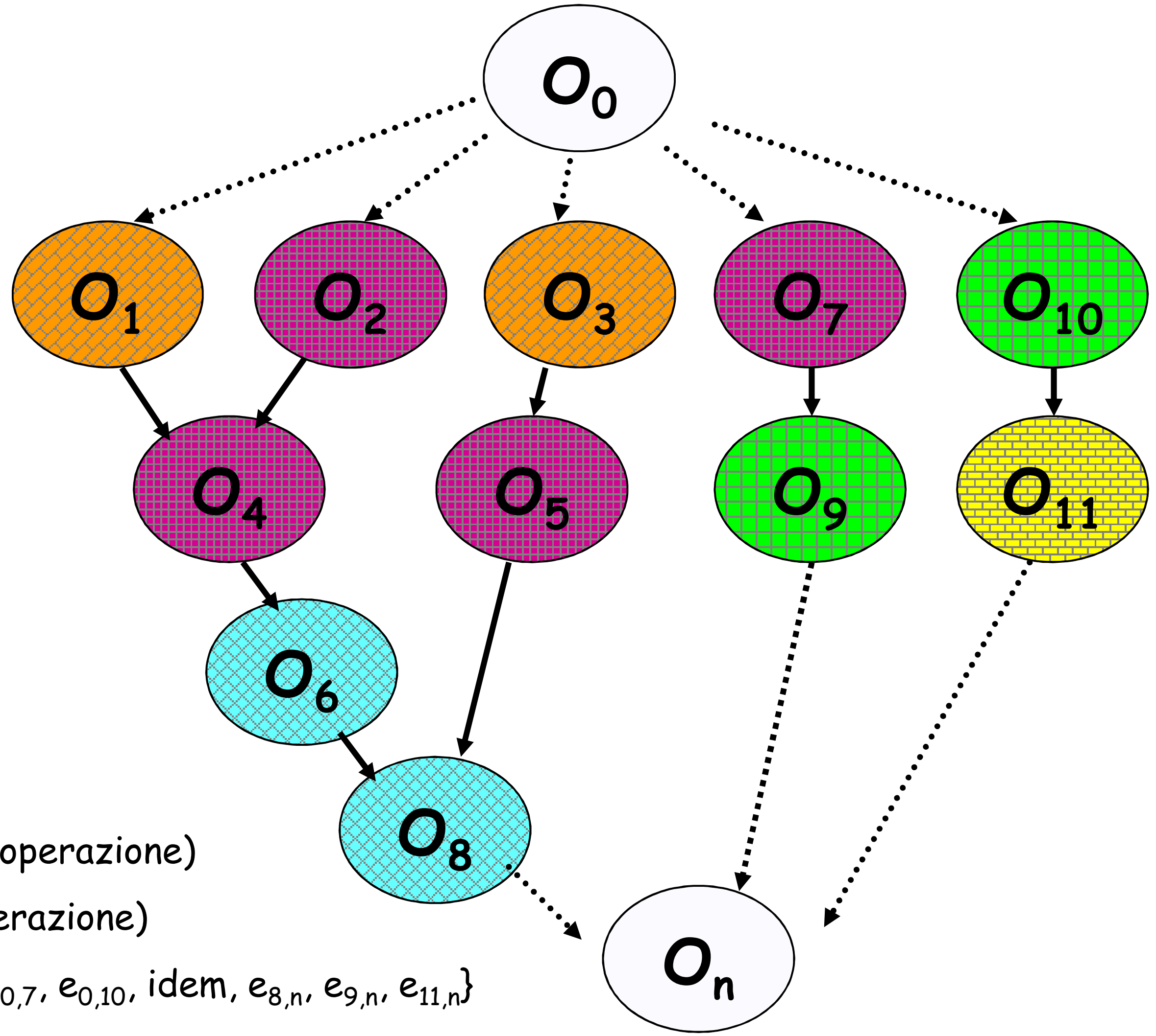
$V = \{v_1, v_2, \ldots, v_{11}\}$

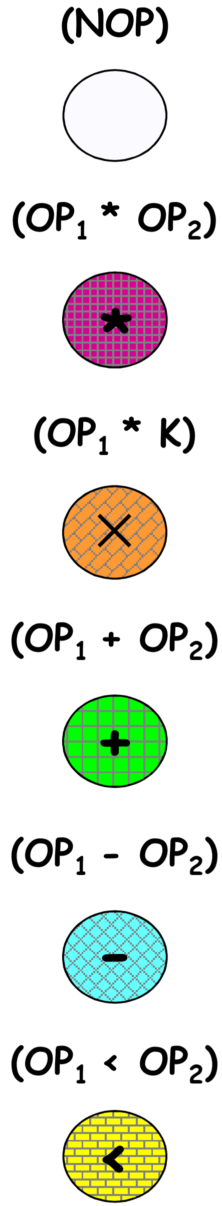
E: insieme degli archi

$E = \{e_{1,4}, e_{2,4}, e_{3,5}, e_{4,6}, e_{5,8}, e_{6,8}, e_{7,9}, e_{10,11}\}$

Rappresentazione in termini di "Sequencing Graph"

Grafo aciclico polare





$G_S(V, E)$

$V = \{v_0, \text{idem}, v_n\}$

v_0: source (pseudo-operazione)

v_n: sink (pseudo-operazione)

$E = \{e_{0,1}, e_{0,2}, e_{0,3}, e_{0,7}, e_{0,10}, \text{idem}, e_{8,n}, e_{9,n}, e_{11,n}\}$

Il compromesso "spazio-tempo"

Dato un insieme di operazioni V contraddistinte da un insieme di vincoli di precedenza E,

definiti i tempi di esecuzione delle operazioni $D = \{d_i;\ i = 0, 1, \dots, n\}$ $(d_0 = d_n = 0)$
una volta stabilita univocamente mediante una funzione $R : V \rightarrow \{1, \dots, N_R\}$
la tipologia di risorsa preposta all'esecuzione di ciascuna di esse,

indicato con $A = \{a_k;\ k = 1, \dots, N_R\}$ il numero di risorse per ciascuna tipologia,
$C = \{c_k;\ k = 1, \dots, N_R\}$ il costo (area) di ogni tipologia di risorse,
$T = \{t_i;\ i = 0, 1, \dots, n\}$ $(t_0 = 1)$ l'insieme degli "scheduling steps"
in cui ha inizio l'esecuzione delle operazioni,

si tratta in generale di individuare

a) il valore minimo della latenza $\lambda = t_n - t_0$ (tempo complessivo di esecuzione delle operazioni), prefissato l'insieme A delle risorse disponibili (resource-constrained scheduling),

b) l'insieme di risorse di costo $\sum_{k=1}^{N_R} c_k a_k$ minimo, prefissato il valore λ della latenza (time-constrained scheduling),

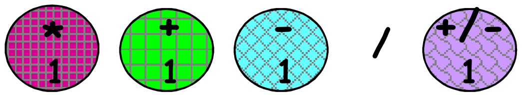
nel rispetto delle seguenti condizioni:

$$t_i \geq t_j + d_j,\ \forall\ i, j : (v_j, v_i) \in E$$

$$|\{v_i: R(v_i) = k \text{ e } t_i \leq l < t_i + d_i\}| \leq a_k,\ \forall\ k = 1, \dots, N_R,\ \forall\ l = 1, \dots, \lambda$$

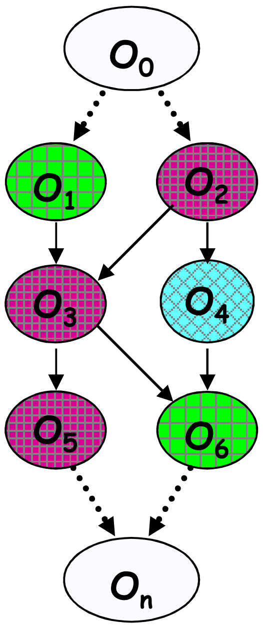
Risorse mono- / multi-funzionali

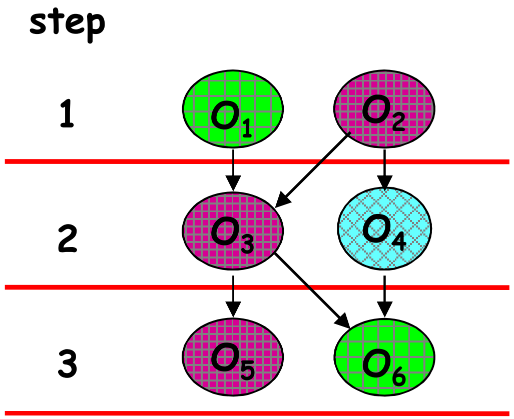
$d_* = d_+ = d_- = d_{+/-} = 1$ t.u. $\longrightarrow$

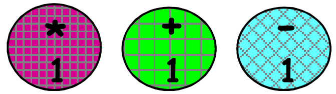


$$\boxed{c_* > c_{+/-} > c_+ = c_-}$$

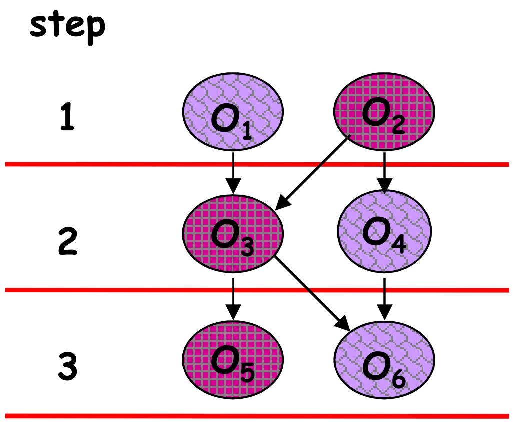
$$\boxed{c_{+/-} < c_+ + c_-}$$

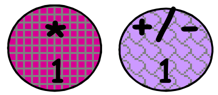






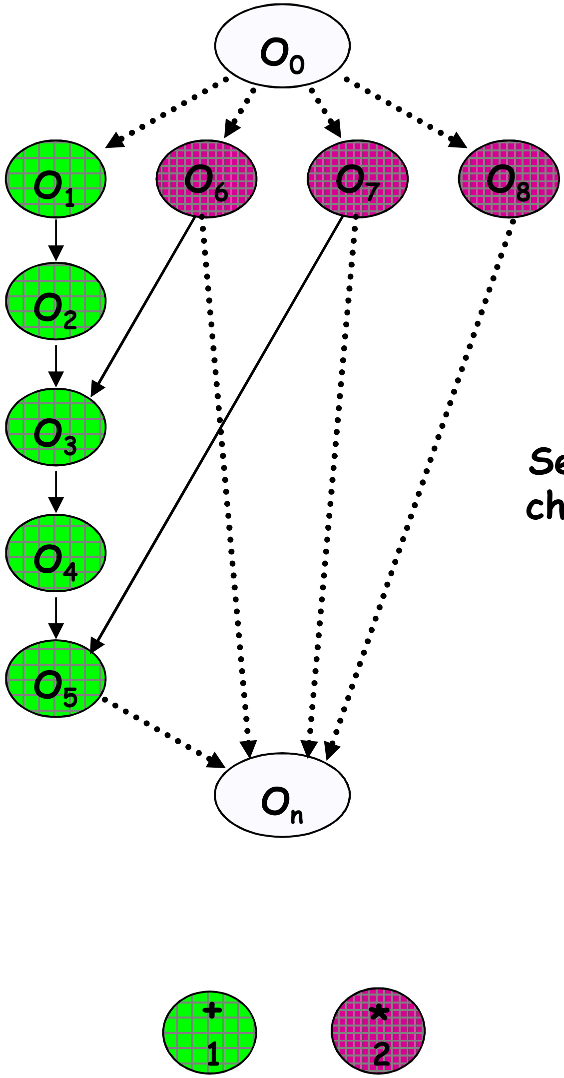
$$\boxed{c_T = c_* + c_+ + c_-}$$



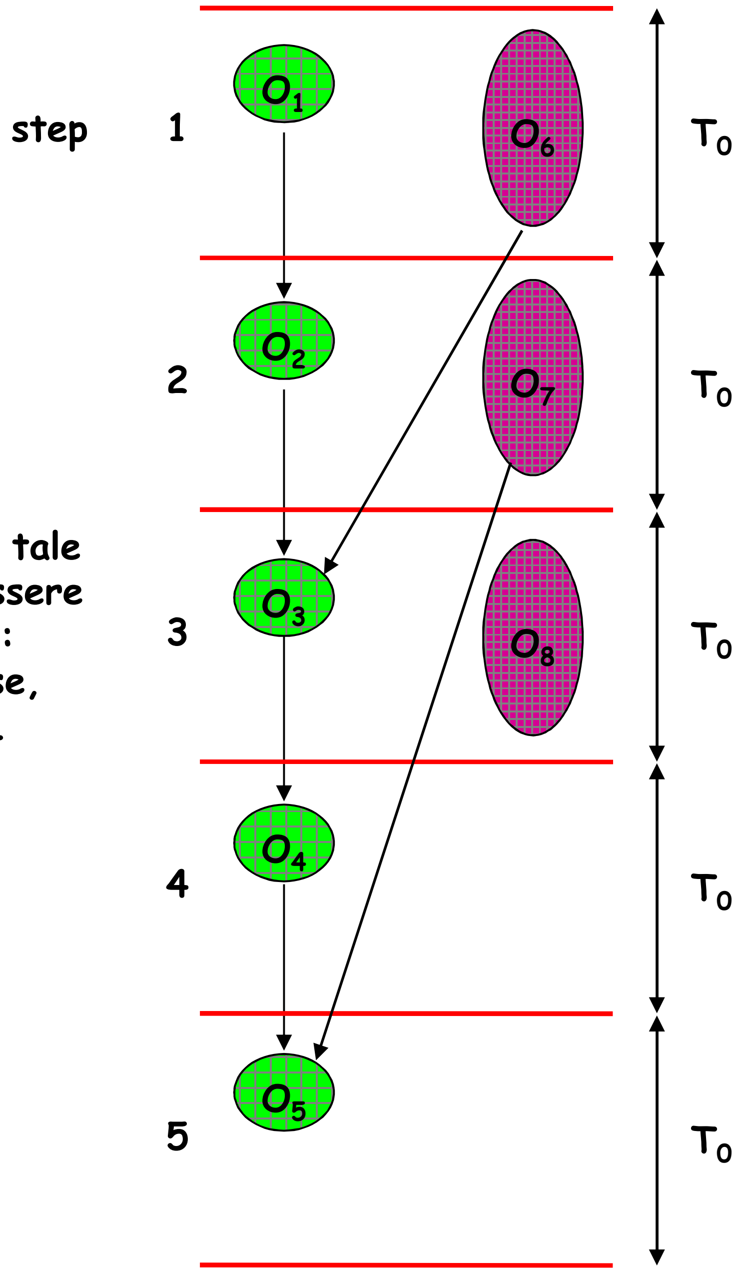


$$\boxed{c_T' = c_* + c_{+/-} < c_T}$$

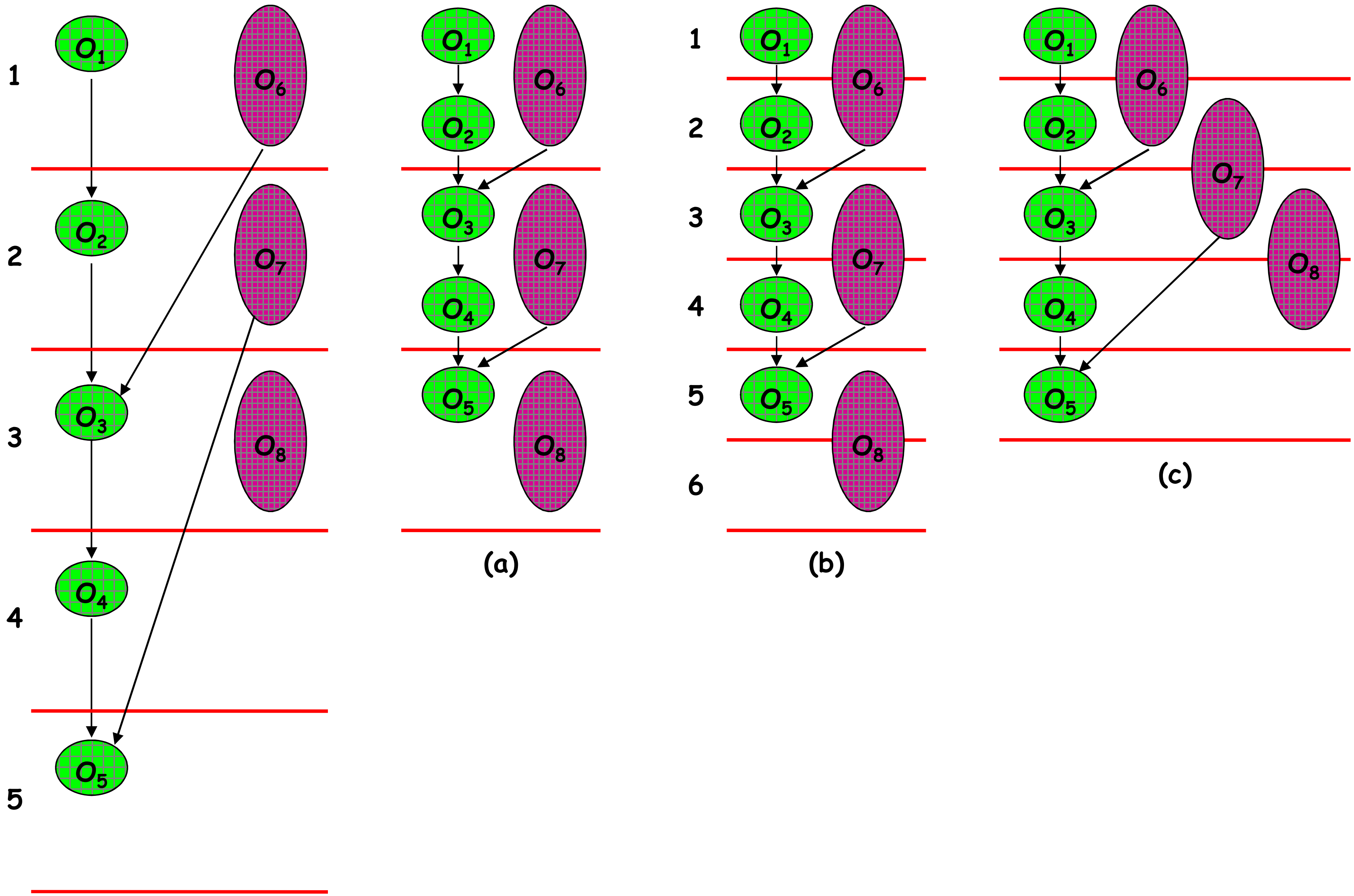
Risorse con tempi di esecuzione differenziati …



Se T_0 è selezionato in modo tale che ogni operazione possa essere eseguita in un unico step: sottoutilizzazione di risorse, estensione della latenza.



... "Chaining" (a), "Multicycling" (b), "Pipelining" (c)



Algoritmi di scheduling “unconstrained”

Ipotesi: risorse illimitate

Obiettivo: latenza minima

Costruzione dello “Scheduled Sequencing Graph” (SSG)

Algoritmo ASAP (As Soon As Possible)

SSG si costruisce a partire dal primo passo, inserendo via via ciascuna operazione allorché tutti i suoi immediati predecessori (operandi) sono disponibili (variabili globali) o già inseriti in passi precedenti (risultati intermedi).

Algoritmo ALAP (As Late As Possible)

SSG si costruisce a partire dall’ultimo passo, inserendo via via ciascuna operazione allorché tutti i suoi immediati successori sono già inseriti in passi successivi.

Gli algoritmi ASAP e ALAP

```
ASAP (G_S (V, E))
{
    schedula v_0 ponendo t^S_0 = 1;
    repeat
    {
        seleziona un vertice v_i i cui predecessori siano già schedulati;
        schedula v_i ponendo t^S_i = max {t^S_j + d_j}, ∀ j : (v_j, v_i) ∈ E;
    } until (v_n è schedulato);
    return (T^S);
}
```

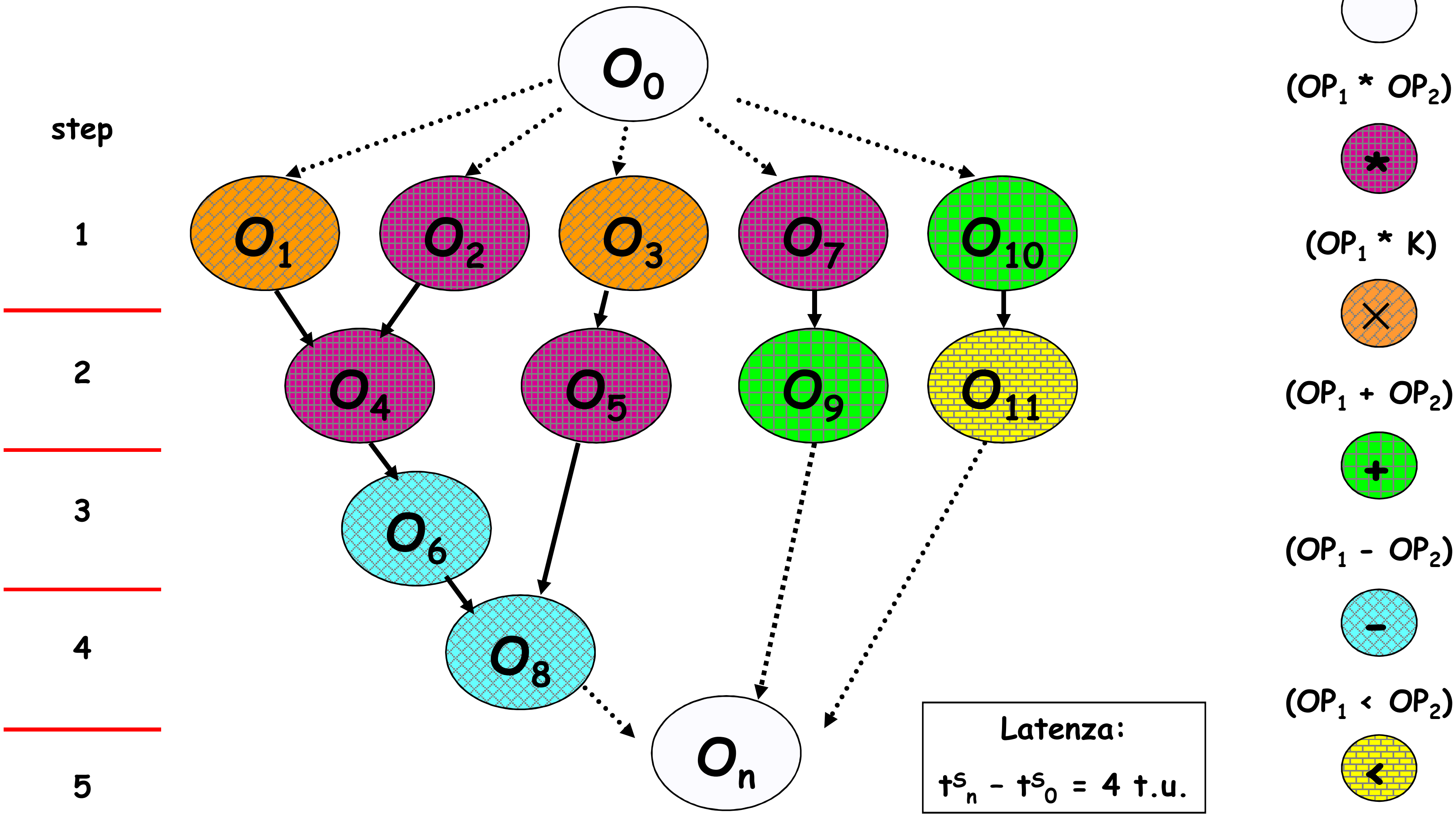
$\lambda_{min} = t^S_n - t^S_0$

```
ALAP (G_S (V, E), λ)
{
    schedula v_n ponendo t^L_n = λ + 1;
    repeat
    {
        seleziona un vertice v_i i cui successori siano già schedulati;
        schedula v_i ponendo t^L_i = min {t^L_j} - d_i, ∀ j : (v_i, v_j) ∈ E;
    } until (v_0 è schedulato);
    return (T^L);
}
```

ASAP: Scheduled Sequencing Graph (SSG)

IED (tempo di esecuzione delle operazioni unitario):

	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_n
t_i^S	1	1	1	1	2	2	3	1	4	2	1	2	5



Latenza:

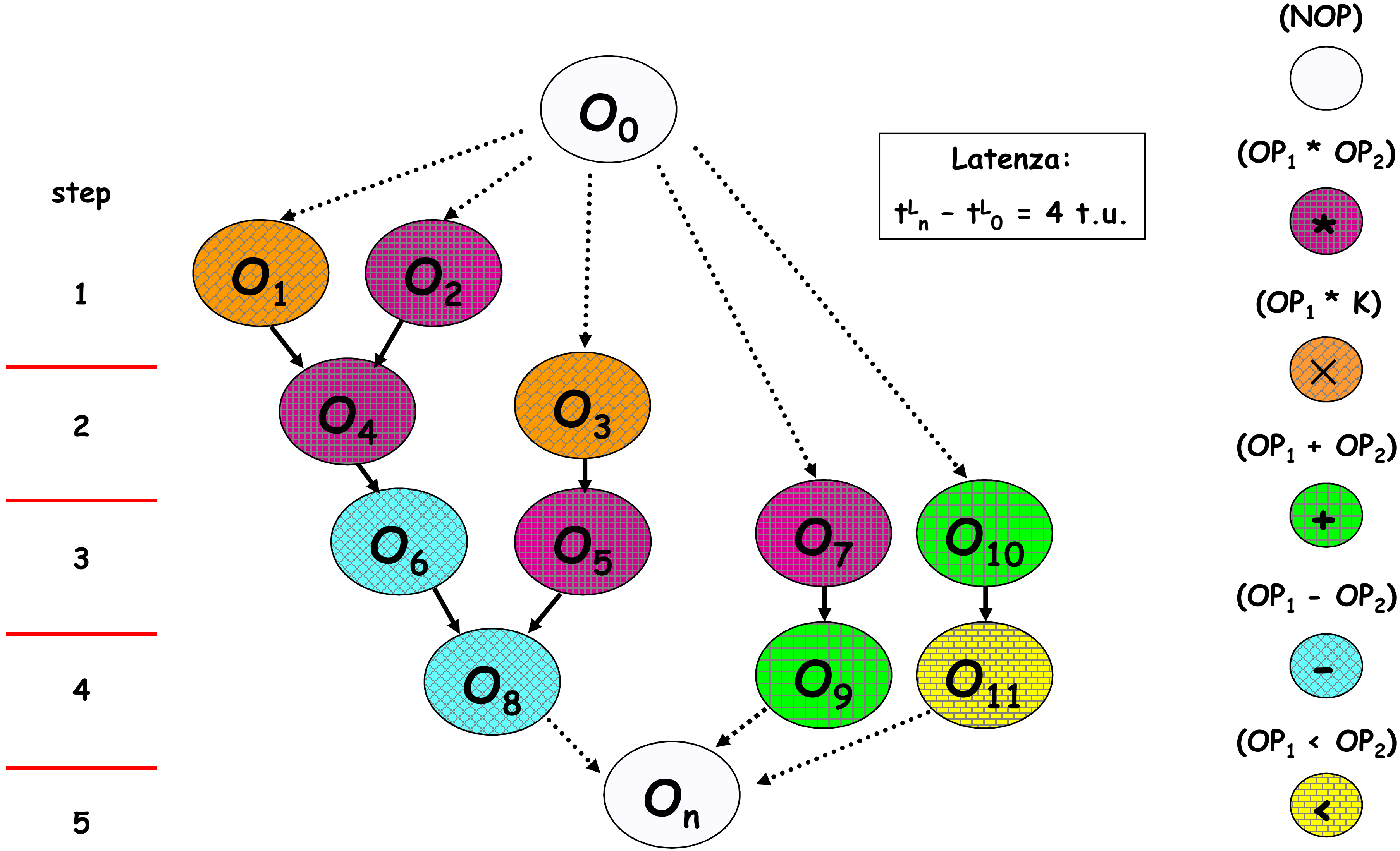
$t^S_n - t^S_0 = 4$ t.u.

ALAP: SSG

	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_n
t_i^L	1	1	1	2	2	3	3	3	4	4	3	4	5

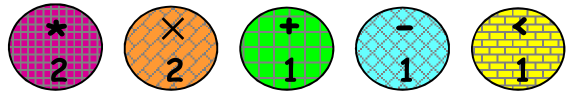
Latenza:

$t^L_n - t^L_0 = 4$ t.u.

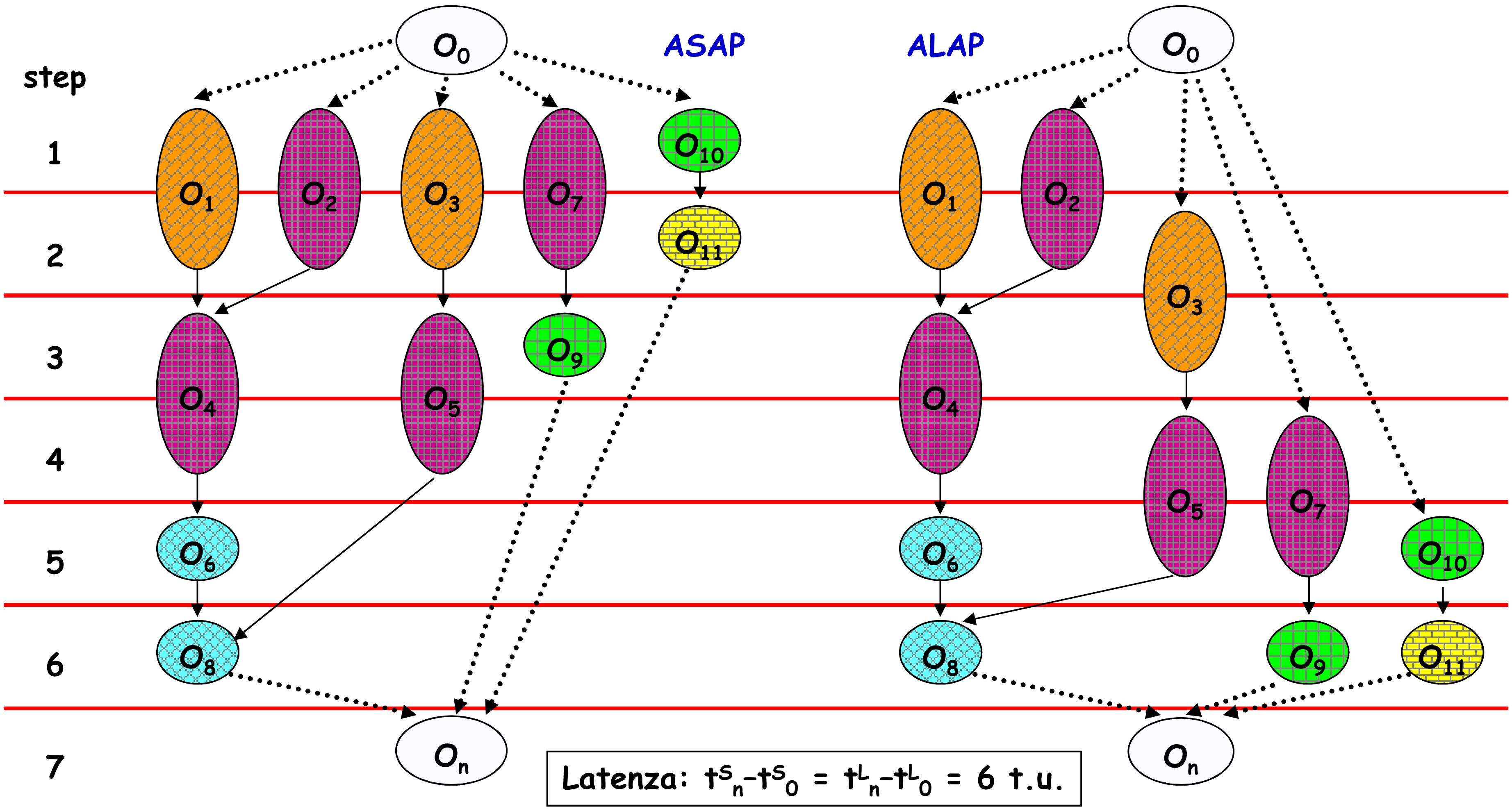


ASAP e ALAP: SSG

Ipotizzando invece i seguenti tempi di esecuzione ([t.u.]), si ottiene:



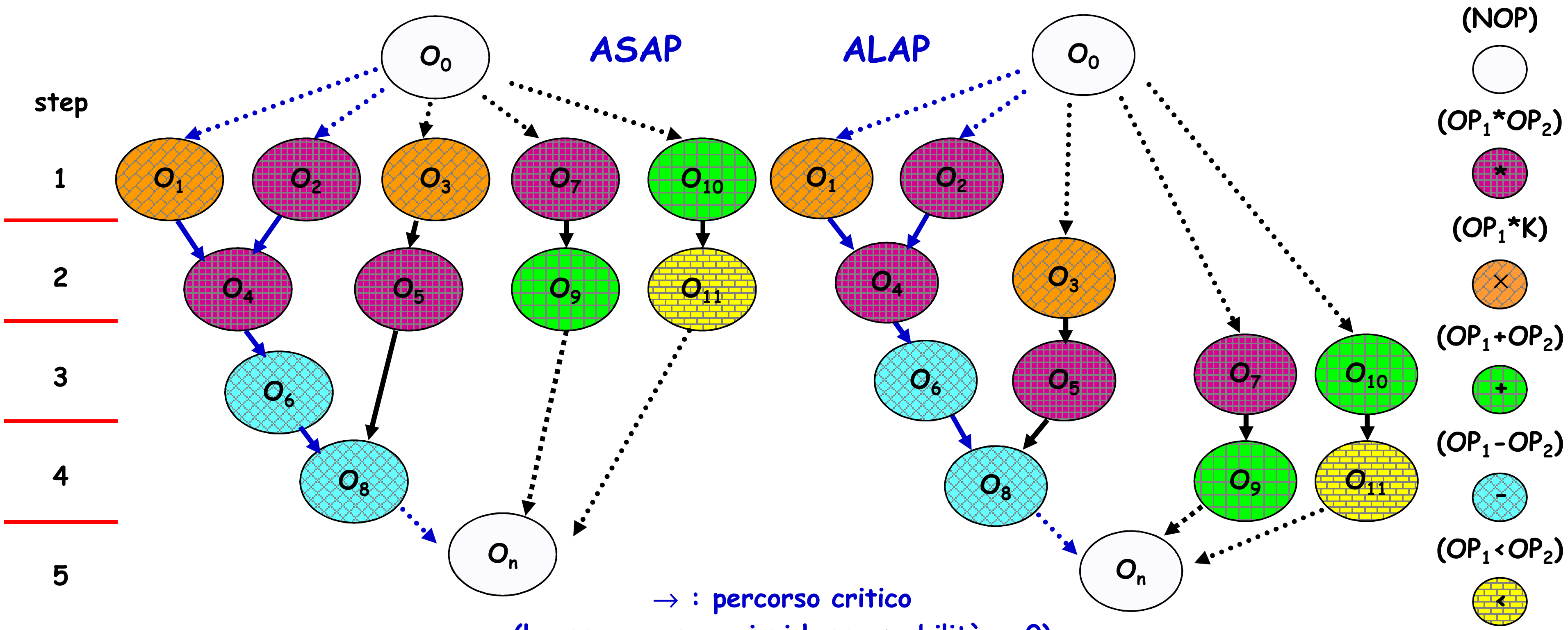
	O_0	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_n
t_i^S	1	1	1	1	3	3	5	1	6	3	1	2	7
t_i^L	1	1	1	2	3	4	5	4	6	6	5	6	7



Latenza: $t^S_n - t^S_0 = t^L_n - t^L_0 = 6$ t.u.

Condivisibilità delle risorse di elaborazione

Un insieme di operazioni, anche funzionalmente diverse, possono condividere una stessa risorsa se sono eseguibili dalla risorsa e non concorrenti.



→ : percorso critico

(le annesse operazioni hanno mobilità μ=0)

Tipologia e numero di risorse

	*	×	+	-	<	*/×	+/-/<
ASAP	2	2	1	1	1	4	2
ALAP	2	1	1	1	1	2	3

In generale gli algoritmi ASAP e ALAP comportano un numero di risorse non ottimale.

Algoritmi di scheduling
"resource-constrained" & "time-constrained"

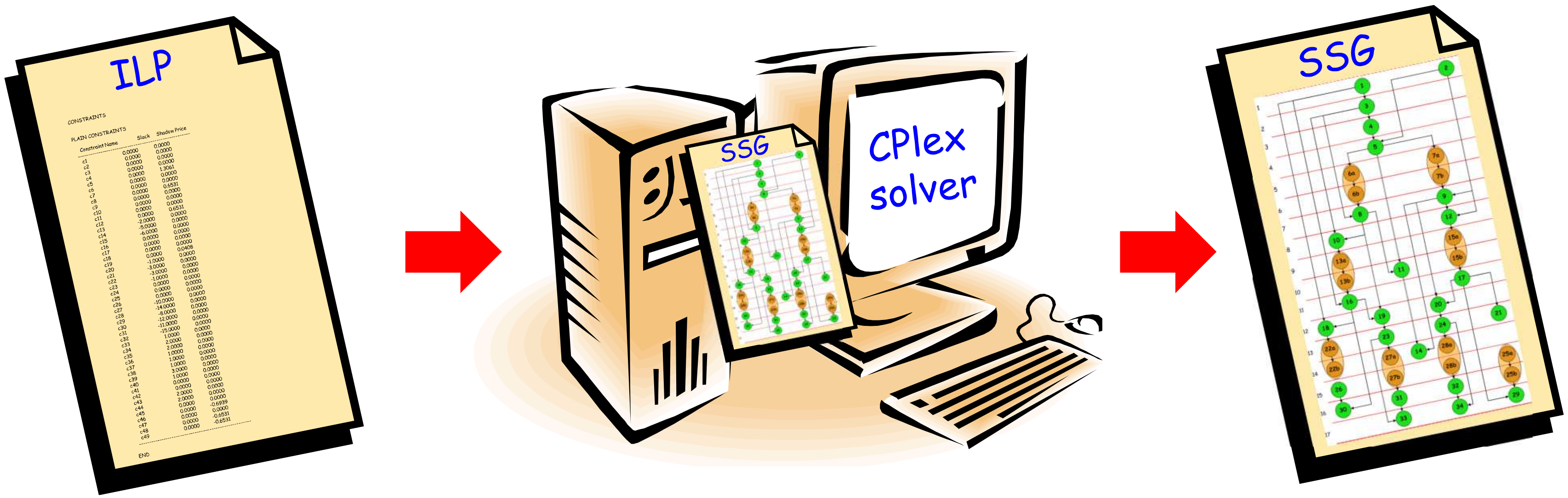
Esistono (?) metodi che forniscono una **soluzione esatta**
per i seguenti problemi di scheduling:

"minimum-latency scheduling problem under resource constraints"

"minimum-resource scheduling problem under latency constraints"

sì

Il modello "Integer Linear Programming" (ILP)



Modello ILP: insieme di vincoli

Il modello si definisce introducendo un insieme di variabili decisionali binarie

$X = \{x_{il} \in \{0, 1\}; i = 0, 1, \ldots, n; l = 1, 2, \ldots, \lambda + 1\}$,

in cui la generica variabile x_{il}, corrispondente all'operazione v_i e allo scheduling step l, vale 1 solo se l'operazione v_i ha inizio nello step l (cioè $t_i = l$),

ed esprimendo i vincoli del problema in termini di tali variabili:

(1) vincoli di unicità dello scheduling step delle operazioni:

$$\Sigma_{1 \leq l \leq \lambda + 1} x_{il} = 1, \quad i = 0, 1, \ldots, n$$

(2) vincoli di precedenza tra operazioni: $t_i \geq t_j + d_j$, $i, j = 0, 1, \ldots, n : (v_j,v_i) \in E$

$$\Sigma_{1 \leq l \leq \lambda + 1} \, l \, x_{il} \geq \Sigma_{1 \leq l \leq \lambda + 1} \, l \, x_{jl} + d_j, \quad i, j = 0, 1, \ldots, n : (v_j,v_i) \in E$$

(3) vincoli sulle risorse: $|\{v_i: R(v_i) = k \text{ e } t_i \leq l < t_i + d_i\}| \leq a_k$, $\forall\, k = 1, \ldots, N_R$, $\forall\, l = 1, \ldots, \lambda$

$$\Sigma_{i \,:\, R(v_i) = k} \Sigma_{l - d_i + 1 \leq m \leq l} \, x_{im} \leq a_k, \quad k = 1, 2, \ldots, N_R, \; l = 1, 2, \ldots, \lambda$$

Modello ILP: funzione obiettivo

Indicato con:
$\underline{A}$ un vettore le cui componenti identificano il numero di risorse per ogni tipologia,
$\underline{C}$ un vettore le cui componenti identificano il costo (area) di ogni tipologia di risorse,
$\underline{T}$ un vettore le cui componenti identificano lo scheduling step di ogni operazione,

la funzione obiettivo è del tipo:

"minimum-latency scheduling problem under resource constraints"

min [0,…,0,1] $\underline{T}$ [1]

ovvero
$\min \Sigma_{1 \leq l \leq \lambda+1}\, l\, x_{nl}$

con il vincolo sulle risorse disponibili espresso da $\underline{A}$

[1] Un "upper bound" per la latenza λ può essere rapidamente identificato con un algoritmo euristico.

"minimum-resource scheduling problem under latency constraints"

min $\underline{C}^T \underline{A}$

ovvero
$\min \Sigma_{1 \leq k \leq N_R}\, c_k\, a_k$

con il vincolo sulla latenza λ espresso da $\Sigma_{1 \leq l \leq \lambda+1}\, l\, x_{nl} \leq \lambda + 1$

Applicazione del modello ILP: λ assegnata $\rightarrow$ risorse minime ...

IED 2 tipologie di risorse: MUL (c_1=5) e ALU (c_2=2), con tempo di esecuzione unitario

$$\lambda = \lambda_{min} = 4$$

1° insieme di vincoli: $\Sigma_{1 \leq l \leq 4}\, x_{il} = 1$, $i = 1, \ldots, 11$

Per ogni operazione i, la sommatoria può convenientemente non essere estesa all'intero range di valori di l, ma limitata al subrange $t_i\,(ASAP) = t_i^S \leq l \leq t_i^L = t_i\,(ALAP)$.

i	t_i^S	t_i^L	μ_i	vincolo	significato
1	1	1	0	$x_{1\,1} = 1$	l'operazione 1 deve iniziare nel passo 1
2	1	1	0	$x_{2\,1} = 1$	l'operazione 2 deve iniziare nel passo 1
3	1	2	1	$x_{3\,1} + x_{3\,2} = 1$	l'operazione 3 deve iniziare nel passo 1 o 2
4	2	2	0	$x_{4\,2} = 1$	l'operazione 4 deve iniziare nel passo 2
5	2	3	1	$x_{5\,2} + x_{5\,3} = 1$	l'operazione 5 deve iniziare nel passo 2 o 3
6	3	3	0	$x_{6\,3} = 1$	l'operazione 6 deve iniziare nel passo 3
7	1	3	2	$x_{7\,1} + x_{7\,2} + x_{7\,3} = 1$	l'operazione 7 deve iniziare nel passo 1 o 2 o 3
8	4	4	0	$x_{8\,4} = 1$	l'operazione 8 deve iniziare nel passo 4
9	2	4	2	$x_{9\,2} + x_{9\,3} + x_{9\,4} = 1$	l'operazione 9 deve iniziare nel passo 2 o 3 o 4
10	1	3	2	$x_{10\,1} + x_{10\,2} + x_{10\,3} = 1$	l'operazione 10 deve iniziare nel passo 1 o 2 o 3
11	2	4	2	$x_{11\,2} + x_{11\,3} + x_{11\,4} = 1$	l'operazione 11 deve iniziare nel passo 2 o 3 o 4

... Applicazione del modello ILP: $\lambda \rightarrow$ risorse minime ...

2° insieme di vincoli: $\sum_{t_i^S \leq l \leq t_i^L} l\, x_{il} \geq \sum_{t_j^S \leq l \leq t_j^L} l\, x_{jl} + d_j,\ \ i, j : (v_j, v_i) \in E$

$d_1 = d_2 = d_3 = d_4 = d_5 = d_7 = d_{MUL} = 1 \qquad d_6 = d_8 = d_9 = d_{10} = d_{11} = d_{ALU} = 1$

vincolo	significato
$2\, x_{5\,2} + 3\, x_{5\,3} - x_{3\,1} - 2\, x_{3\,2} - 1 \geq 0$	$O_3 \prec O_5$
$2\, x_{9\,2} + 3\, x_{9\,3} + 4\, x_{9\,4} - x_{7\,1} - 2\, x_{7\,2} - 3\, x_{7\,3} - 1 \geq 0$	$O_7 \prec O_9$
$2\, x_{11\,2} + 3\, x_{11\,3} + 4\, x_{11\,4} - x_{10\,1} - 2\, x_{10\,2} - 3\, x_{10\,3} - 1 \geq 0$	$O_{10} \prec O_{11}$

notazione:
$O_j \prec O_i$
O_j precede O_i

(solo i vincoli aggiuntivi che coinvolgono operazioni con mobilità non nulla)

$\boxed{O_1 \prec O_4}$ $\boxed{O_2 \prec O_4}$ $O_3 \prec O_5$ $\boxed{O_4 \prec O_6}$ $\boxed{O_5 \prec O_8}$ $\boxed{O_6 \prec O_8}$ $O_7 \prec O_9$ $O_{10} \prec O_{11}$

... Applicazione del modello ILP: $\lambda \rightarrow$ risorse minime ...

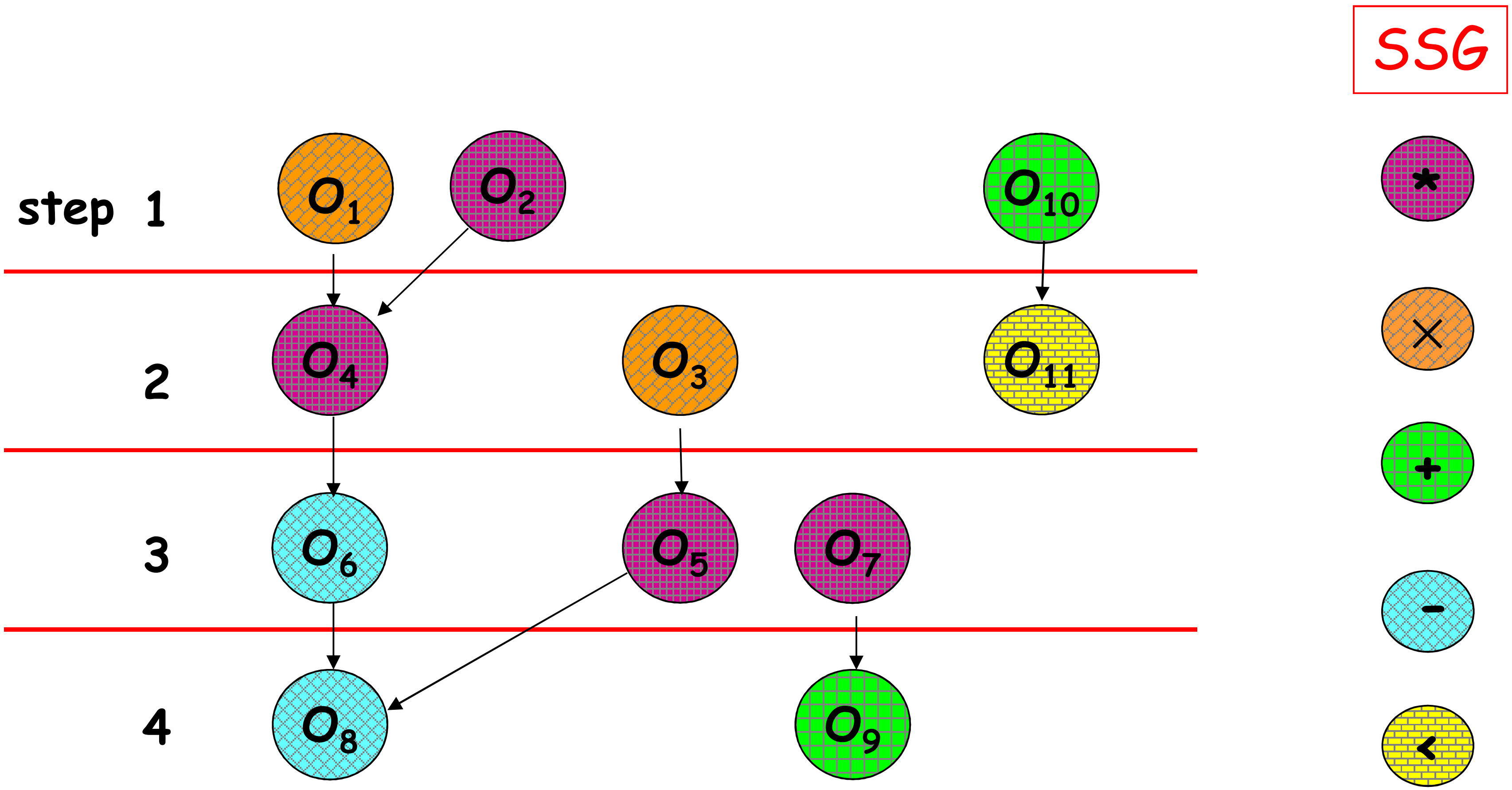
3° insieme di vincoli: $\Sigma_{i\,:\,R(v_i)\,=\,k}\, x_{il} \leq a_k$, $k = 1, 2$, $l = 1, \ldots, 4$

l	vincolo	significato
1	$x_{1\,1} + x_{2\,1} + x_{3\,1} + x_{7\,1} - a_1 \leq 0$	non più di a_1 operazioni del tipo */x in esecuzione nel passo 1
1	$x_{10\,1} - a_2 \leq 0$	non più di a_2 operazioni del tipo +/-/< in esecuzione nel passo 1
2	$x_{3\,2} + x_{4\,2} + x_{5\,2} + x_{7\,2} - a_1 \leq 0$	non più di a_1 operazioni del tipo */x in esecuzione nel passo 2
2	$x_{9\,2} + x_{10\,2} + x_{11\,2} - a_2 \leq 0$	non più di a_2 operazioni del tipo +/-/< in esecuzione nel passo 2
3	$x_{5\,3} + x_{7\,3} - a_1 \leq 0$	non più di a_1 operazioni del tipo */x in esecuzione nel passo 3
3	$x_{6\,3} + x_{9\,3} + x_{10\,3} + x_{11\,3} - a_2 \leq 0$	non più di a_2 operazioni del tipo +/-/< in esecuzione nel passo 3
4	$x_{8\,4} + x_{9\,4} + x_{11\,4} - a_2 \leq 0$	non più di a_2 operazioni del tipo +/-/< in esecuzione nel passo 4

funzione obiettivo:
min $(5\, a_1 + 2\, a_2)$

$$\longrightarrow \left\{ \begin{array}{llllll} x_{3\,1} = 0 & x_{5\,2} = 0 & x_{7\,1} = 0 & x_{9\,2} = 0 & x_{10\,1} = 1 & x_{11\,2} = 1 \\ x_{3\,2} = 1 & x_{5\,3} = 1 & x_{7\,2} = 0 & x_{9\,3} = 0 & x_{10\,2} = 0 & x_{11\,3} = 0 \\ & & x_{7\,3} = 1 & x_{9\,4} = 1 & x_{10\,3} = 0 & x_{11\,4} = 0 \end{array} \right.$$

... Applicazione del modello ILP: $\lambda \rightarrow$ risorse minime



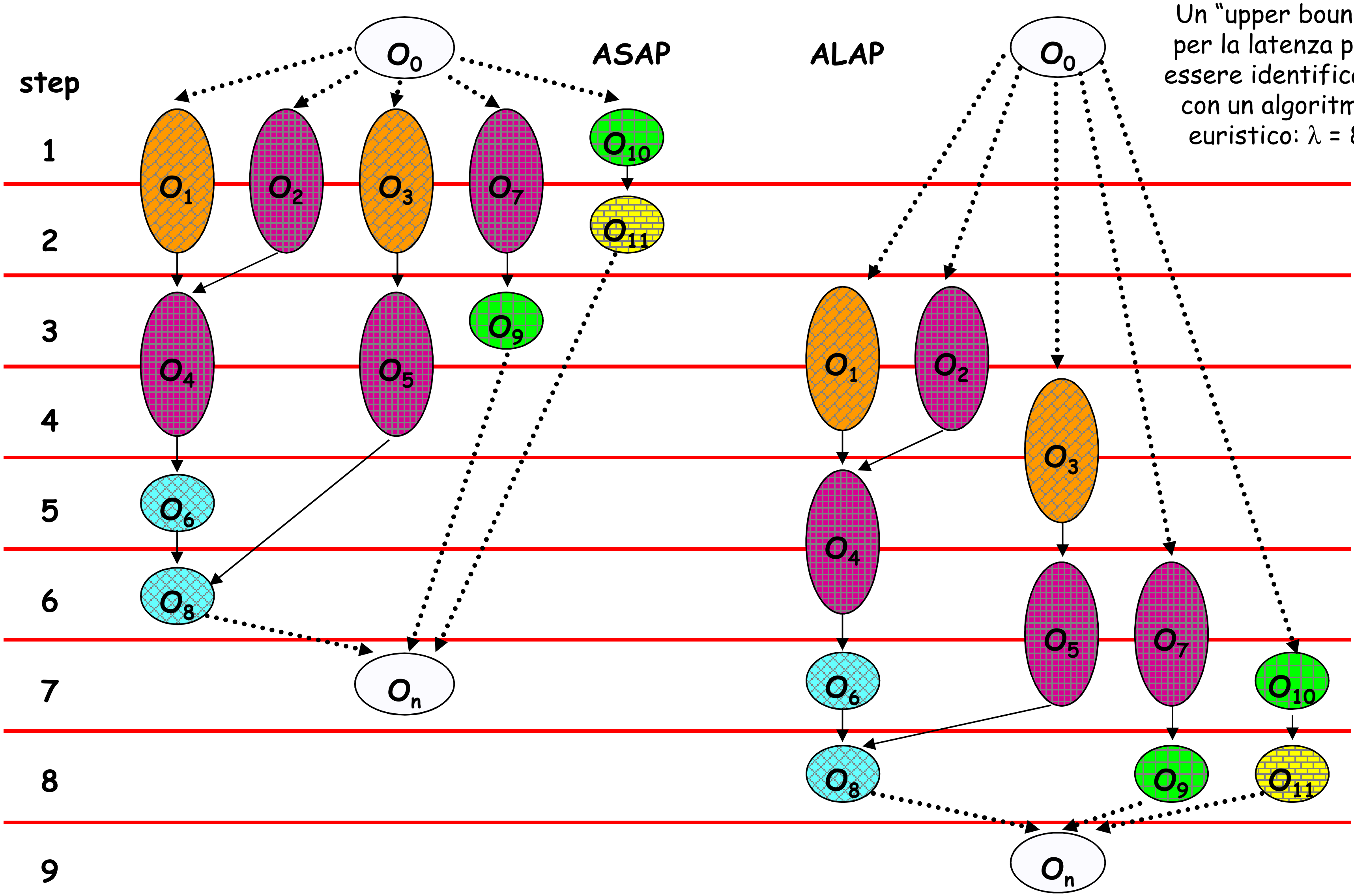
Risorse necessarie:
2 moltiplicatori
2 ALU

Costo complessivo:
14 unità

Latenza:
4 t.u.

Applicazione del modello ILP: risorse predefinite → min λ ...

Risorse disponibili: 2 MUL ($a_1 = 2$) e 1 ALU ($a_2 = 1$), con $d_{MUL} = 2$, $d_{ALU} = 1$



Un "upper bound" per la latenza può essere identificato con un algoritmo euristico: λ = 8

… Applicazione del modello ILP: risorse predefinite → min λ …

1° insieme di vincoli: $\Sigma_{t_i^S \leq l \leq t_i^L} x_{il} = 1$, $i = 1, \ldots, n$

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_n
t_i^L	3	3	4	5	6	7	6	8	8	7	8	9
t_i^S	1	1	1	3	3	5	1	6	3	1	2	7

$x_{1\,1} + x_{1\,2} + x_{1\,3} = 1$

$x_{2\,1} + x_{2\,2} + x_{2\,3} = 1$

$x_{3\,1} + x_{3\,2} + x_{3\,3} + x_{3\,4} = 1$

$x_{4\,3} + x_{4\,4} + x_{4\,5} = 1$

$x_{5\,3} + x_{5\,4} + x_{5\,5} + x_{5\,6} = 1$

$x_{6\,5} + x_{6\,6} + x_{6\,7} = 1$

$x_{7\,1} + x_{7\,2} + x_{7\,3} + x_{7\,4} + x_{7\,5} + x_{7\,6} = 1$

$x_{8\,6} + x_{8\,7} + x_{8\,8} = 1$

$x_{9\,3} + x_{9\,4} + x_{9\,5} + x_{9\,6} + x_{9\,7} + x_{9\,8} = 1$

$x_{10\,1} + x_{10\,2} + x_{10\,3} + x_{10\,4} + x_{10\,5} + x_{10\,6} + x_{10\,7} = 1$

$x_{11\,2} + x_{11\,3} + x_{11\,4} + x_{11\,5} + x_{11\,6} + x_{11\,7} + x_{11\,8} = 1$

$x_{n\,7} + x_{n\,8} + x_{n\,9} = 1$

… Applicazione del modello ILP: risorse predefinite $\rightarrow$ min λ …

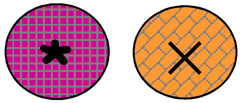
2° insieme di vincoli: $\Sigma_{t_i^S \leq l \leq t_i^L}\ l\, x_{il} \geq \Sigma_{t_j^S \leq l \leq t_j^L}\ l\, x_{jl} + d_j,\ \ i, j : (v_j,v_i) \in E$

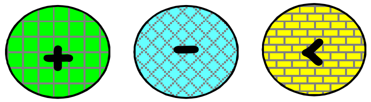
$d_1 = d_2 = d_3 = d_4 = d_5 = d_7 = d_{MUL} = 2$ $\qquad$ $d_6 = d_8 = d_9 = d_{10} = d_{11} = d_{ALU} = 1$

$O_1 \prec O_4$	$3\,x_{4\,3} + 4\,x_{4\,4} + 5\,x_{4\,5} - x_{1\,1} - 2\,x_{1\,2} - 3\,x_{1\,3} - 2 \geq 0$
$O_2 \prec O_4$	$3\,x_{4\,3} + 4\,x_{4\,4} + 5\,x_{4\,5} - x_{2\,1} - 2\,x_{2\,2} - 3\,x_{2\,3} - 2 \geq 0$
$O_4 \prec O_6$	$5\,x_{6\,5} + 6\,x_{6\,6} + 7\,x_{6\,7} - 3\,x_{4\,3} - 4\,x_{4\,4} - 5\,x_{4\,5} - 2 \geq 0$
$O_6 \prec O_8$	$6\,x_{8\,6} + 7\,x_{8\,7} + 8\,x_{8\,8} - 5\,x_{6\,5} - 6\,x_{6\,6} - 7\,x_{6\,7} - 1 \geq 0$
$O_3 \prec O_5$	$3\,x_{5\,3} + 4\,x_{5\,4} + 5\,x_{5\,5} + 6\,x_{5\,6} - x_{3\,1} - 2\,x_{3\,2} - 3\,x_{3\,3} - 4\,x_{3\,4} - 2 \geq 0$
$O_5 \prec O_8$	$6\,x_{8\,6} + 7\,x_{8\,7} + 8\,x_{8\,8} - 3\,x_{5\,3} - 4\,x_{5\,4} - 5\,x_{5\,5} - 6\,x_{5\,6} - 2 \geq 0$
$O_7 \prec O_9$	$3\,x_{9\,3} + 4\,x_{9\,4} + 5\,x_{9\,5} + 6\,x_{9\,6} + 7\,x_{9\,7} + 8\,x_{9\,8} - x_{7\,1} - 2\,x_{7\,2} - 3\,x_{7\,3} - 4\,x_{7\,4} - 5\,x_{7\,5} - 6\,x_{7\,6} - 2 \geq 0$
$O_{10} \prec O_{11}$	$2\,x_{11\,2} + 3\,x_{11\,3} + 4\,x_{11\,4} + 5\,x_{11\,5} + 6\,x_{11\,6} + 7\,x_{11\,7} + 8\,x_{11\,8} - x_{10\,1} - 2\,x_{10\,2} - 3\,x_{10\,3} - 4\,x_{10\,4} - 5\,x_{10\,5} - 6\,x_{10\,6} - 7\,x_{10\,7} - 1 \geq 0$
$O_8 \prec O_n$	$7\,x_{n\,7} + 8\,x_{n\,8} + 9\,x_{n\,9} - 6\,x_{8\,6} - 7\,x_{8\,7} - 8\,x_{8\,8} - 1 \geq 0$
$O_9 \prec O_n$	$7\,x_{n\,7} + 8\,x_{n\,8} + 9\,x_{n\,9} - 3\,x_{9\,3} - 4\,x_{9\,4} - 5\,x_{9\,5} - 6\,x_{9\,6} - 7\,x_{9\,7} - 8\,x_{9\,8} - 1 \geq 0$
$O_{11} \prec O_n$	$7\,x_{n\,7} + 8\,x_{n\,8} + 9\,x_{n\,9} - 2\,x_{11\,2} - 3\,x_{11\,3} - 4\,x_{11\,4} - 5\,x_{11\,5} - 6\,x_{11\,6} - 7\,x_{11\,7} - 8\,x_{11\,8} - 1 \geq 0$

… Applicazione del modello ILP: risorse predefinite $\rightarrow$ min λ …

3° insieme di vincoli: $\sum_{i\,:\,R(v_i)=k} \sum_{l-d_i+1 \le m \le l} x_{im} \le a_k,\ k = 1, 2, \ldots, N_R,\ l = 1, 2, \ldots, \lambda$



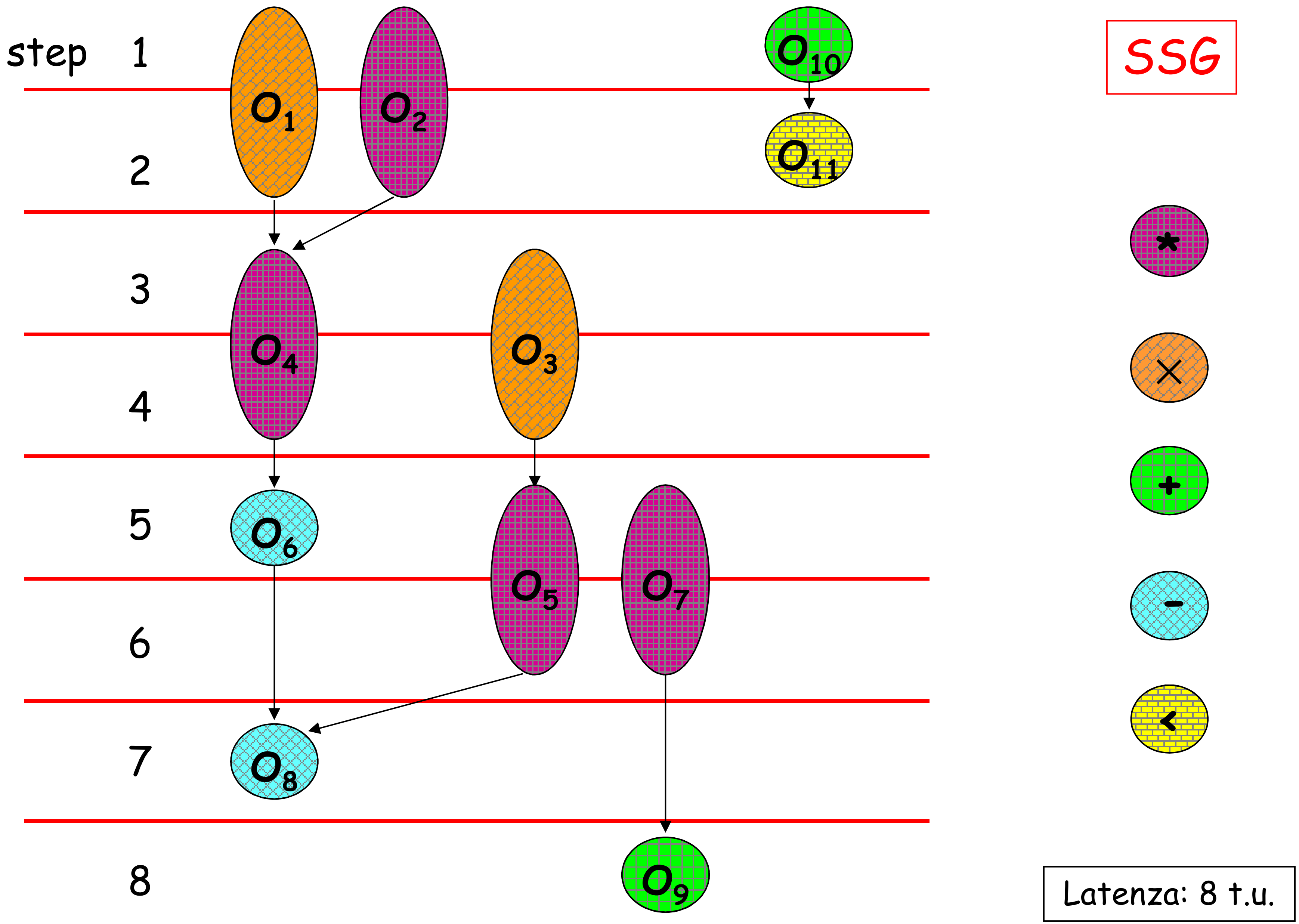


step		
1	$x_{1\,1} + x_{2\,1} + x_{3\,1} + x_{7\,1} - 2 \le 0$	$x_{10\,1} - 1 \le 0$
2	$x_{1\,1} + x_{1\,2} + x_{2\,1} + x_{2\,2} + x_{3\,1} + x_{3\,2} + x_{7\,1} + x_{7\,2} - 2 \le 0$	$x_{10\,2} + x_{11\,2} - 1 \le 0$
3	$x_{1\,2} + x_{1\,3} + x_{2\,2} + x_{2\,3} + x_{3\,2} + x_{3\,3} + x_{4\,3} + x_{5\,3} + x_{7\,2} + x_{7\,3} - 2 \le 0$	$x_{9\,3} + x_{10\,3} + x_{11\,3} - 1 \le 0$
4	$x_{1\,3} + x_{2\,3} + x_{3\,3} + x_{3\,4} + x_{4\,3} + x_{4\,4} + x_{5\,3} + x_{5\,4} + x_{7\,3} + x_{7\,4} - 2 \le 0$	$x_{9\,4} + x_{10\,4} + x_{11\,4} - 1 \le 0$
5	$x_{3\,4} + x_{4\,4} + x_{4\,5} + x_{5\,4} + x_{5\,5} + x_{7\,4} + x_{7\,5} - 2 \le 0$	$x_{6\,5} + x_{9\,5} + x_{10\,5} + x_{11\,5} - 1 \le 0$
6	$x_{4\,5} + x_{5\,5} + x_{5\,6} + x_{7\,5} + x_{7\,6} - 2 \le 0$	$x_{6\,6} + x_{8\,6} + x_{9\,6} + x_{10\,6} + x_{11\,6} - 1 \le 0$
7	$x_{5\,6} + x_{7\,6} - 2 \le 0$	$x_{6\,7} + x_{8\,7} + x_{9\,7} + x_{10\,7} + x_{11\,7} - 1 \le 0$
8		$x_{8\,8} + x_{9\,8} + x_{11\,8} - 1 \le 0$

funzione obiettivo:
min $(7\, x_{n\,7} + 8\, x_{n\,8} + 9\, x_{n\,9})$ $\longrightarrow$

$$\begin{cases} x_{n\,9} = 1 & x_{1\,1} = 1 & x_{2\,1} = 1 & x_{3\,3} = 1 \\ x_{4\,3} = 1 & x_{5\,5} = 1 & x_{6\,5} = 1 & x_{7\,5} = 1 \\ x_{8\,7} = 1 & x_{9\,8} = 1 & x_{10\,1} = 1 & x_{11\,2} = 1 \end{cases}$$

... Applicazione del modello ILP: risorse predefinite → min λ



Pregi e limiti del modello ILP

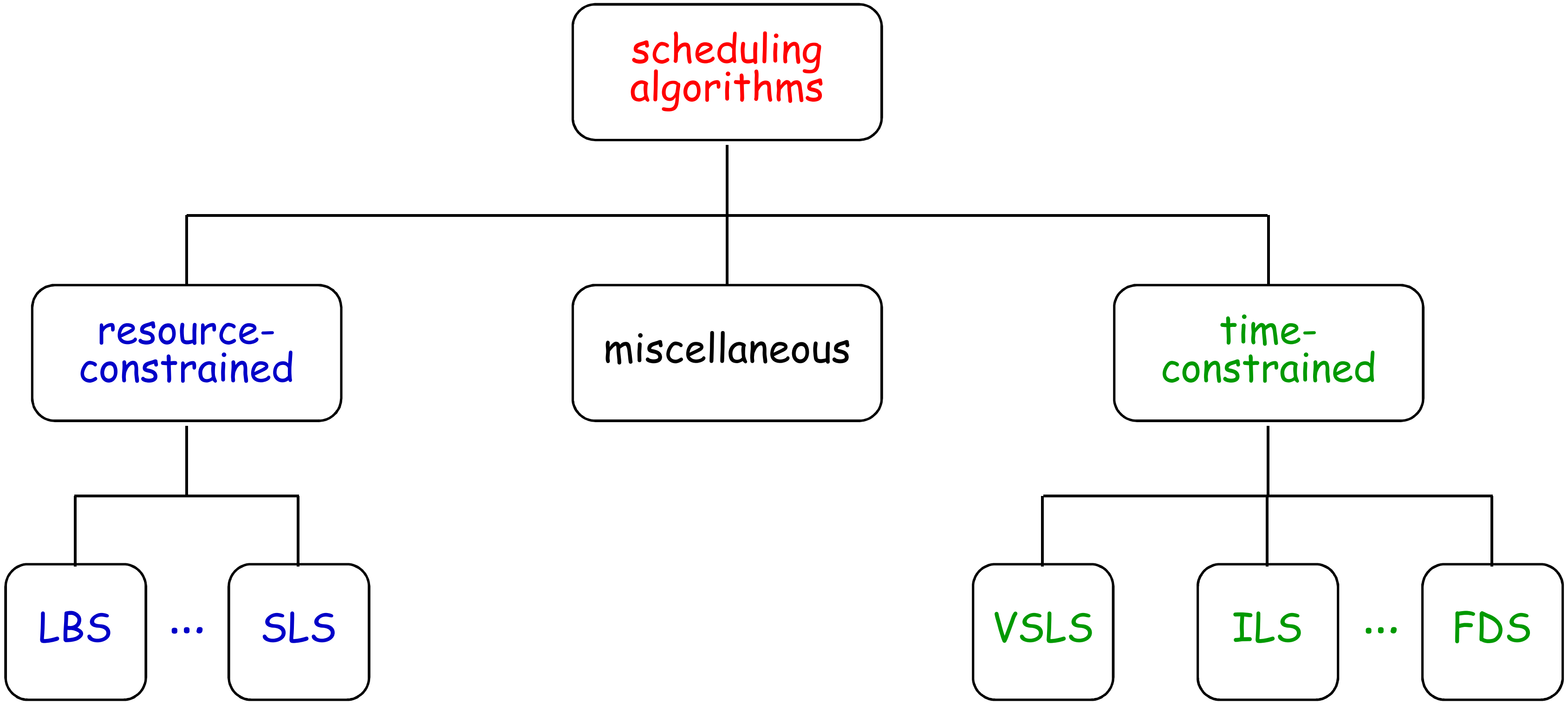
Pro

- Soluzione esatta
  dei problemi di scheduling
- Disponibilità
  di programmi standard
- Estendibilità del modello
  a problemi che coinvolgono
  ulteriori vincoli

Contro

- Complessità computazionale

Algoritmi di scheduling euristici



obiettivo:
latenza minima
in presenza di
risorse predefinite

obiettivo:
numero minimo
di risorse con
latenza predefinita

Static List Scheduling (SLS)

List-Based Scheduling (LSB)

Variante Static List Scheduling (VSLS)

Iterative List Scheduling (ILS)

Force-Directed Scheduling (FDS)

Algoritmi euristici "resource-constrained"

```
AERC (G_S (V, E) , A)
{
    l = 1;
    repeat
    {
        for k = 1, …, N_R
        {
            Determina le operazioni candidate C_lk;
            Determina le operazioni non terminate E_lk;
            Seleziona S_lk ⊆ C_lk : |S_lk| + |E_lk| ≤ a_k;
            Schedula le operazioni in S_lk ponendo t_i = l per ∀ v_i ∈ S_lk;
        }
        l = l + 1;
    } until (v_n è schedulato);
    return (T);
}
```

vincolo sulle risorse disponibili

Le operazioni candidate C_{lk} sono quelle di tipo k schedulabili allo step l, ovvero quelle per cui l'esecuzione di tutti i predecessori è già terminata allo step l.

Le operazioni non terminate E_{lk} sono quelle di tipo k la cui esecuzione non è ancora terminata allo step l. L'insieme E_{lk} è vuoto se il tempo di esecuzione delle operazioni è unitario.

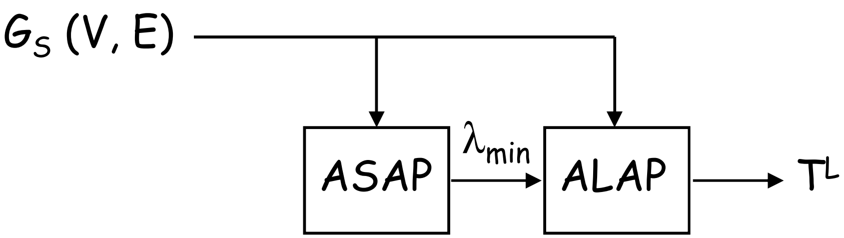
Problema:
come selezionare il sottoinsieme S_{lk} di operazioni tra quelle condidate ?

Definizione di una priority list mediante vari metodi.

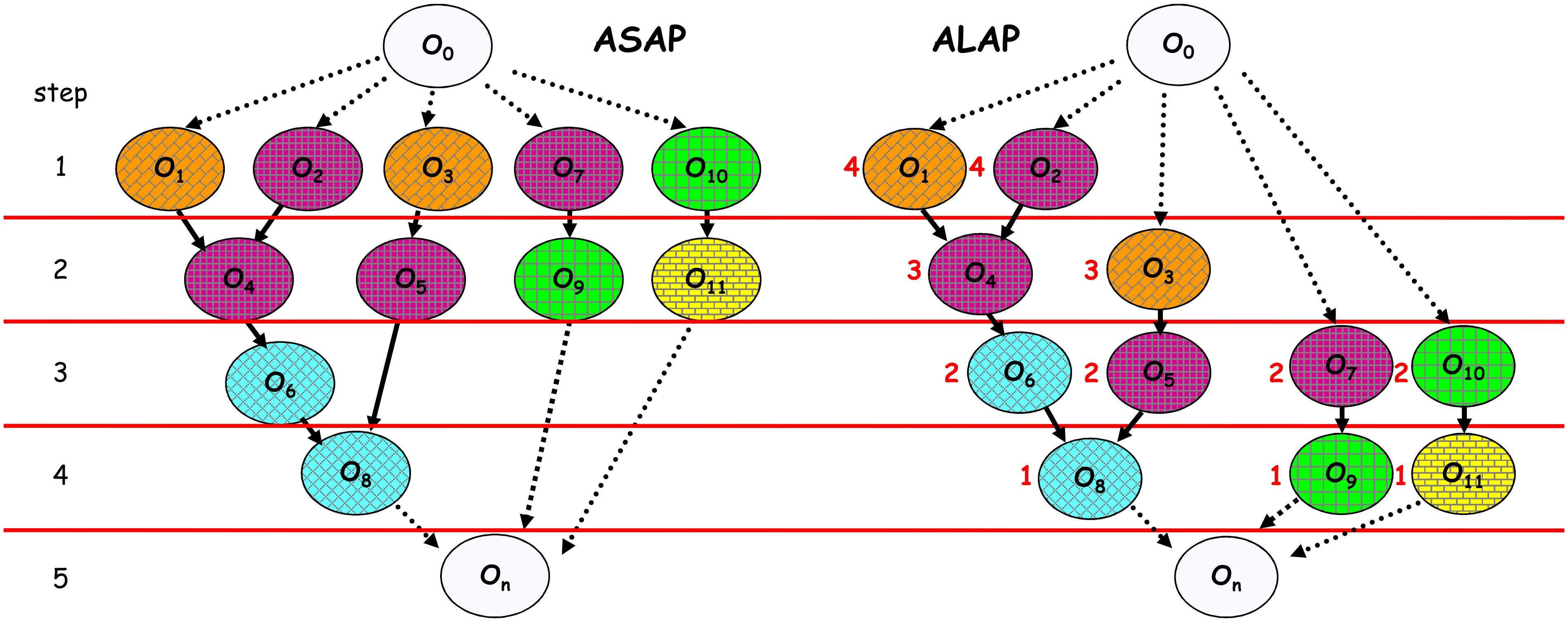
L'algoritmo Static List Scheduling (SLS) ...

IED

Tipologie di risorse disponibili:
moltiplicatori e ALU, entrambe con tempo di esecuzione unitario



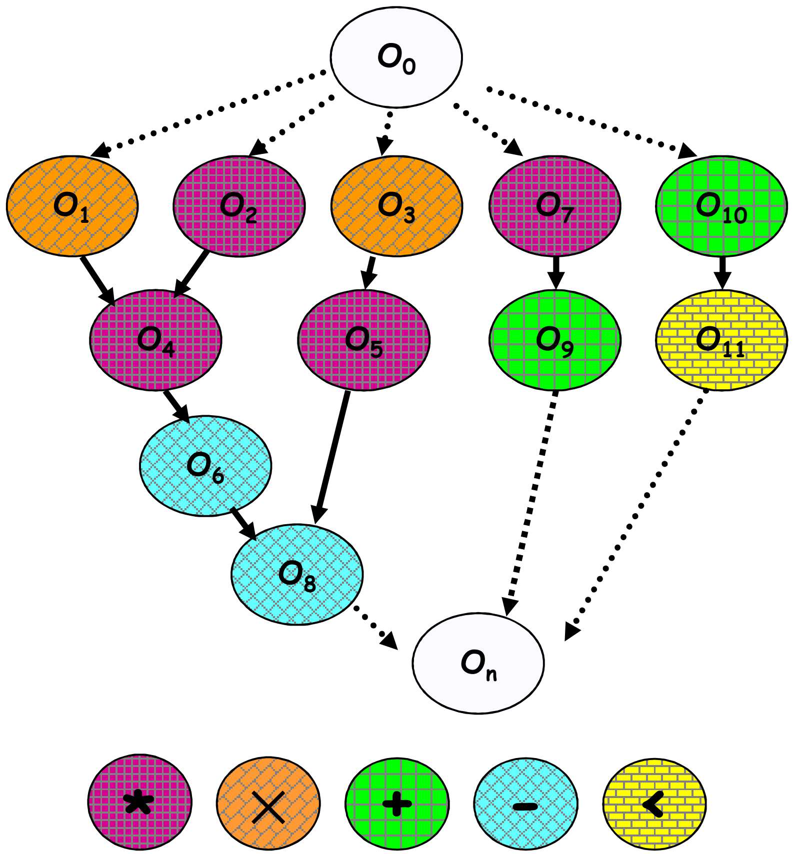
Al vertice v_i ($\forall$ i) è associata staticamente una priorità $p_i = t_n^L - t_i^L$ (quanto più elevata è la "distanza" di v_i da v_n, tanto maggiore è la priorità).



... L'algoritmo Static List Scheduling (SLS) ...

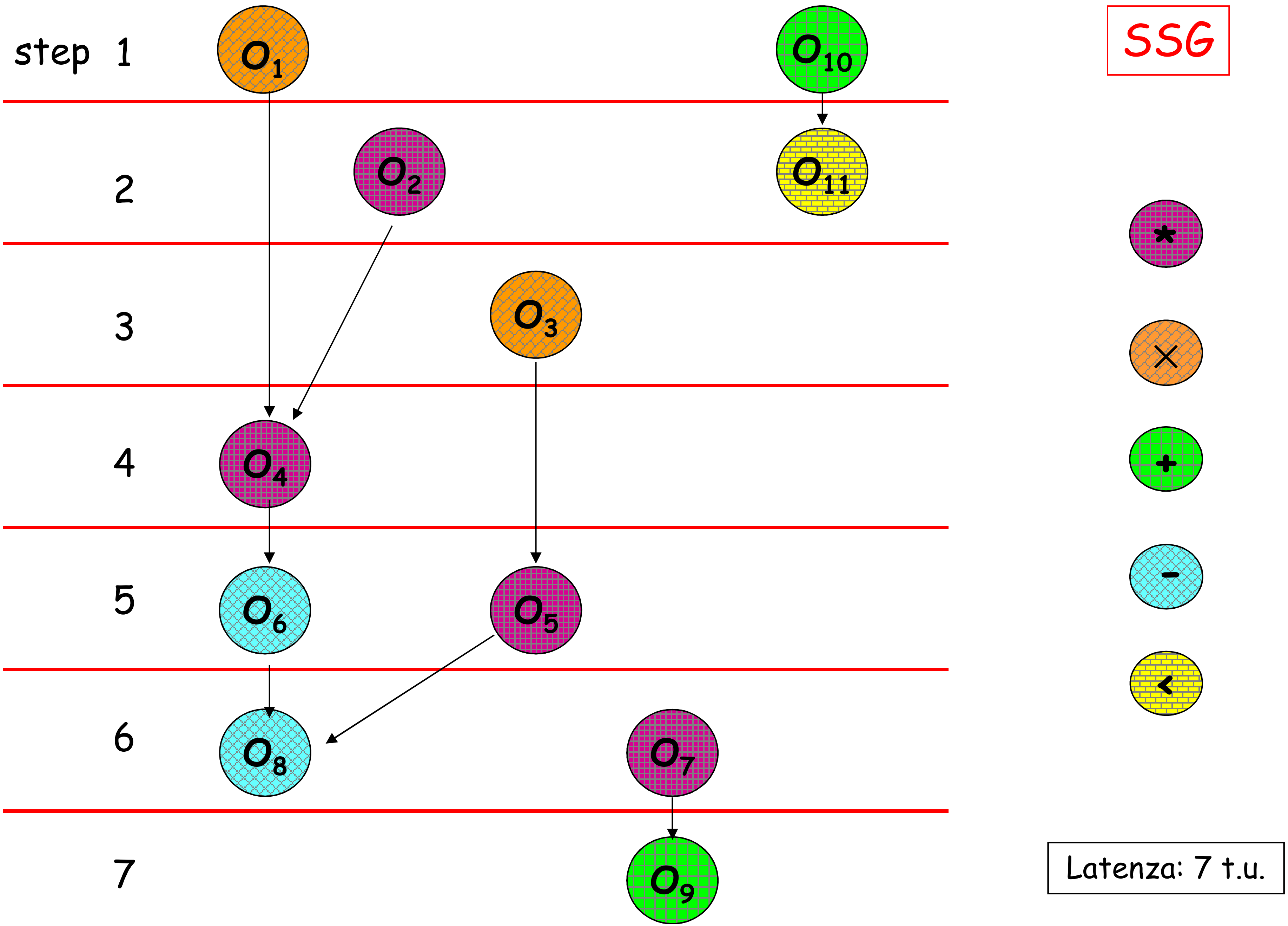
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
p_i	4	4	3	3	2	2	2	1	1	2	1

Ipotizzando che le risorse disponibili siano 1 moltiplicatore e 1 ALU, si ottiene:



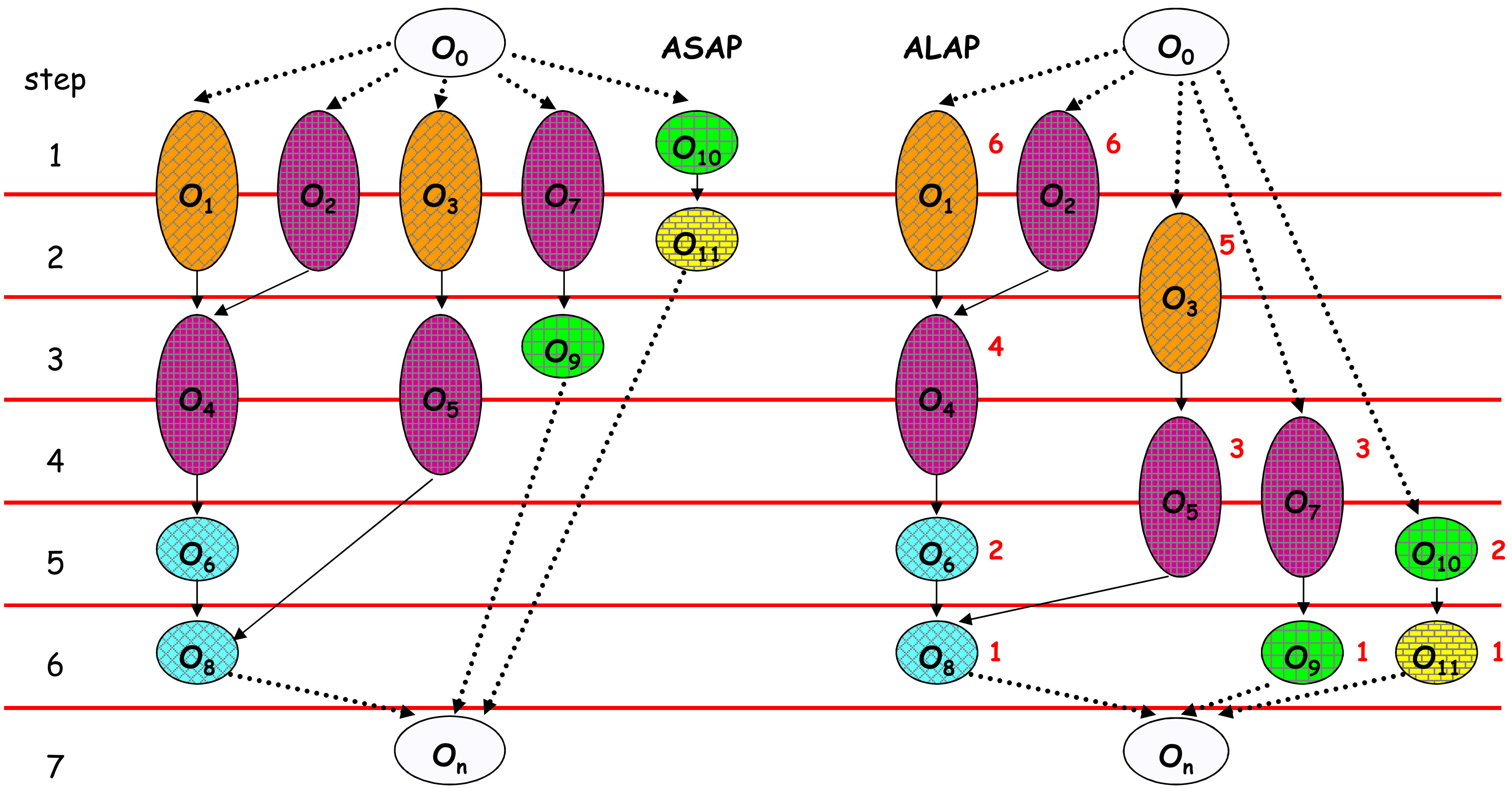
		Operazioni candidate	Operazioni non terminate	Operazioni schedulate
step 1	*/x	O_1,O_2,O_3,O_7	-	O_1
	+/-/<	O_{10}	-	O_{10}
step 2	*/x	O_2,O_3,O_7	-	O_2
	+/-/<	O_{11}	-	O_{11}
step 3	*/x	O_3,O_4,O_7	-	O_3
	+/-/<	-	-	-
step 4	*/x	O_4,O_5,O_7	-	O_4
	+/-/<	-	-	-
step 5	*/x	O_5,O_7	-	O_5
	+/-/<	O_6	-	O_6
step 6	*/x	O_7	-	O_7
	+/-/<	O_8	-	O_8
step 7	*/x	-	-	-
	+/-/<	O_9	-	O_9

… L'algoritmo SLS …



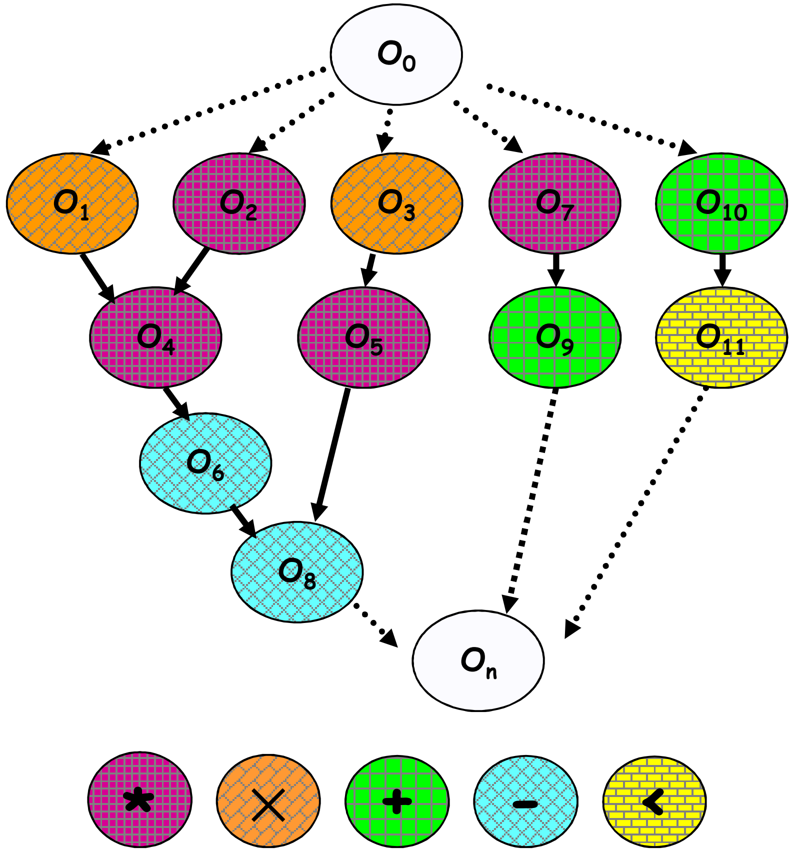
... L'algoritmo SLS ...

Ipotizzando che le risorse disponibili siano 2 moltiplicatori e 1 ALU,
con tempo di esecuzione rispettivamente 2 t.u. e 1 t.u., si ottiene:



… L'algoritmo SLS

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
p_i	6	6	5	4	3	2	3	1	1	2	1



		Operazioni		
		candidate	non terminate	schedulate
step 1	*/x	O_1,O_2,O_3,O_7	-	O_1,O_2
	+/-/<	O_{10}	-	O_{10}
step 2	*/x	O_3,O_7	O_1,O_2	-
	+/-/<	O_{11}	-	O_{11}
step 3	*/x	O_3,O_4,O_7	-	O_3,O_4
	+/-/<	-	-	-
step 4	*/x	O_7	O_3,O_4	-
	+/-/<	-	-	-
step 5	*/x	O_5,O_7	-	O_5,O_7
	+/-/<	O_6	-	O_6
step 6	*/x	-	O_5,O_7	-
	+/-/<	-	-	-
step 7	*/x	-	-	-
	+/-/<	O_8,O_9	-	O_8
step 8	*/x	-	-	-
	+/-/<	O_9	-	O_9

Latenza: 8 t.u.

SSG:
come per ILP

L'algoritmo List-Based Scheduling (LBS) ...

Ad ogni operazione O_i è associata inizialmente una priorità tanto minore quanto maggiore è la corrispondente mobilità μ_i:

$$\mu_i = t_i^L - t_i^S.$$

IED (tempo di esecuzione delle operazioni unitario) :

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
t_i^L	1	1	2	2	3	3	3	4	4	3	4
t_i^S	1	1	1	2	2	3	1	4	2	1	2
μ_i	0	0	1	0	1	0	2	0	2	2	2

Alle operazioni disposte lungo un percorso critico corrisponde mobilità nulla e quindi priorità massima.

Le operazioni eseguibili con ciascuna tipologia di risorse vengono mantenute in due liste distinte, ordinate per priorità decrescente, la prima comprendente le operazioni schedulabili (OS), la seconda quelle non ancora schedulabili (ONS).

In ciascuna iterazione:
vengono schedulate le OS a massima priorità, nei limiti delle risorse disponibili;
vengono quindi aggiornate le liste delle OS rimuovendo le operazioni schedulate e trasferendo in esse dalle liste delle ONS quelle divenute schedulabili;
viene aumentata la priorità delle OS la cui esecuzione è differita,
così come quella delle ONS la cui esecuzione risulterà conseguentemente ritardata.

... L'algoritmo LBS ...

Ipotizzando che le risorse disponibili siano 1 moltiplicatore e 1 ALU, si ottiene:

	Operazioni (priorità)					
	schedulabili		non schedulabili		schedulate	
step	*/x	+/-/<	*/x	+/-/<	*/x	+/-/<
1	$O_1(0)$ $O_2(0)$ $O_3(1)$ $O_7(2)$	$O_{10}(2)$	$O_4(0)$ $O_5(1)$	$O_6(0)$ $O_8(0)$ $O_9(2)$ $O_{11}(2)$	O_1	O_{10}
2	$O_2(-1)$ $O_3(0)$ $O_7(1)$	$O_{11}(2)$	$O_4(-1)$ $O_5(0)$	$O_6(-1)$ $O_8(-1)$ $O_9(1)$	O_2	O_{11}
3	$O_3(-1)$ $O_4(-1)$ $O_7(0)$	-	$O_5(-1)$	$O_6(-1)$ $O_8(-1)$ $O_9(0)$	O_3	-
4	$O_4(-2)$ $O_5(-1)$ $O_7(-1)$	-	-	$O_6(-2)$ $O_8(-2)$ $O_9(-1)$	O_4	-
5	$O_5(-2)$ $O_7(-2)$	$O_6(-2)$	-	$O_8(-2)$ $O_9(-2)$	O_5	O_6
6	$O_7(-3)$	$O_8(-2)$	-	$O_9(-3)$	O_7	O_8
7	-	$O_9(-3)$	-	-	-	O_9

$E = \{e_{1,4}, e_{2,4}, e_{3,5}, e_{4,6}, e_{5,8}, e_{6,8}, e_{7,9}, e_{10,11}\}$

Latenza: 7 t.u.

SSG: come per SLS

... L'algoritmo LBS

Ipotizzando che le risorse disponibili siano 2 moltiplicatori e 1 ALU,
con tempo di esecuzione rispettivamente 2 t.u. e 1 t.u., si ottiene:

$E = \{e_{1,4}, e_{2,4}, e_{3,5}, e_{4,6}, e_{5,8}, e_{6,8}, e_{7,9}, e_{10,11}\}$

O_i	t_i^S	t_i^L	μ_i
O_1	1	1	0
O_2	1	1	0
O_3	1	2	1
O_4	3	3	0
O_5	3	4	1
O_6	5	5	0
O_7	1	4	3
O_8	6	6	0
O_9	3	6	3
O_{10}	1	5	4
O_{11}	2	6	4

Latenza: 8 t.u.

SSG:
come per SLS

	Operazioni (priorità)					
	schedulabili		non schedulabili		schedulate	
step	*/x	+/-/<	*/x	+/-/<	*/x	+/-/<
1	$O_1(0)$ $O_2(0)$ $O_3(1)$ $O_7(3)$	$O_{10}(4)$	$O_4(0)$ $O_5(1)$	$O_6(0)$ $O_8(0)$ $O_9(3)$ $O_{11}(4)$	O_1, O_2	O_{10}
2	$O_3(0)$ $O_7(2)$	$O_{11}(4)$	$O_4(0)$ $O_5(0)$	$O_6(0)$ $O_8(0)$ $O_9(2)$	-	O_{11}
3	$O_3(-1)$ $O_4(0)$ $O_7(1)$	-	$O_5(-1)$	$O_8(-1)$ $O_6(0)$ $O_9(1)$	O_3, O_4	-
4	$O_7(0)$	-	$O_5(-1)$	$O_8(-1)$ $O_6(0)$ $O_9(0)$	-	-
5	$O_5(-1)$ $O_7(-1)$	$O_6(0)$	-	$O_8(-1)$ $O_9(-1)$	O_5, O_7	O_6
6	-	-	-	$O_8(-1)$ $O_9(-1)$	-	-
7	-	$O_8(-1)$ $O_9(-1)$	-	-	-	O_8
8	-	$O_9(-2)$	-	-	-	O_9

Algoritmi euristici "time-constrained"

una Variante dell'algoritmo SLS (VSLS)

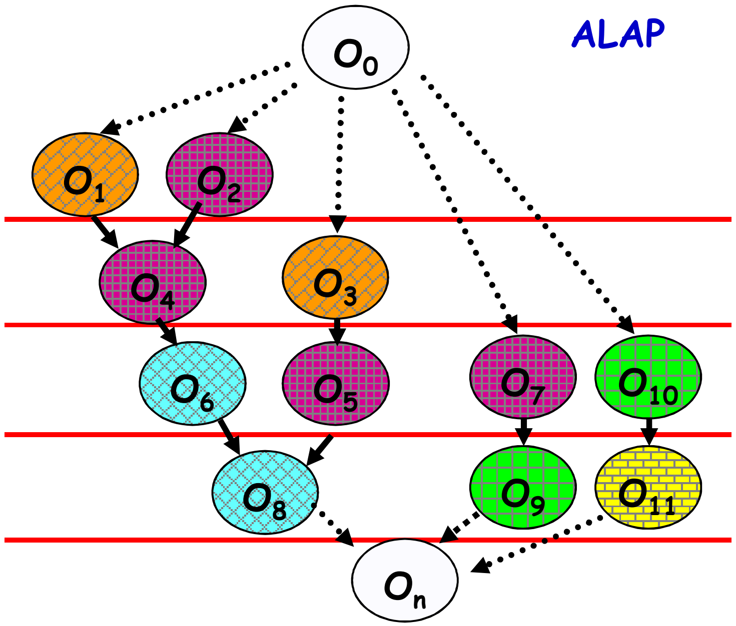
VSLS (G_S (V, E), λ) ← vincolo temporale
{
A = {a_k = 1; k = 1, …, N_R};
Calcola T^L ALAP (G_S (V, E), λ);
l = 1;
repeat
{
for k = 1, …, N_R
{
Determina le operazioni candidate C_{lk};
Calcola lo "slack" $s_i = t_i^L - l$, $\forall$ $v_i \in C_{lk}$;
Identifica l'insieme $U_{lk} \subseteq C_{lk}$ delle operazioni candidate con slack nullo;
Determina le operazioni non terminate E_{lk};
if ($|U_{lk}| + |E_{lk}| > a_k$) Aggiorna $a_k = |U_{lk}| + |E_{lk}|$;
Schedula le operazioni in U_{lk} ed eventualmente altre operazioni candidate D_{lk}, privilegiando quelle con slack inferiore e nei limiti delle risorse disponibili ($|U_{lk}| + |E_{lk}| + |D_{lk}| \leq a_k$), ponendo $t_i = l$ per $\forall$ $v_i \in U_{lk} \cup D_{lk}$;
}
l = l + 1;
} until (v_n è schedulato);
return (T, A);
}

... L'algoritmo VSLS

IED

Ipotizzando che il vincolo sulla latenza sia $\lambda = 4$ e che siano disponibili due tipologie di risorse (moltiplicatore e ALU), entrambe con tempo di esecuzione unitario, posto inizialmente A = [*/x, +/-/<] = [1,1], si ottiene:

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
t_i^L	1	1	2	2	3	3	3	4	4	3	4



		Operazioni: candidate (slack)	Operazioni: schedulate	A
				[1,1]
step 1	*/x	$O_1(0),O_2(0),O_3(1),O_7(2)$	O_1,O_2	[2,1]
	+/-/<	$O_{10}(2)$	O_{10}	
step 2	*/x	$O_3(0),O_4(0),O_7(1)$	O_3,O_4	
	+/-/<	$O_{11}(2)$	O_{11}	
step 3	*/x	$O_5(0),O_7(0)$	O_5,O_7	
	+/-/<	$O_6(0)$	O_6	
step 4	*/x	-	-	
	+/-/<	$O_8(0),O_9(0)$	O_8,O_9	[2,2]

Risorse necessarie:
2 moltiplicatori
2 ALU

SSG:
come per ILP

L'algoritmo "Iterative List Scheduling" (ILS)

ILS differisce da VSLS per il fatto che, ogni qual volta le operazioni candidate non più differibili eccedono il numero di risorse disponibili, il processo, previo incremento delle risorse rivelatesi insufficienti, viene reiterato dall'inizio con l'intendimento di poterle proficuamente utilizzare anche nei precedenti passi di scheduling.

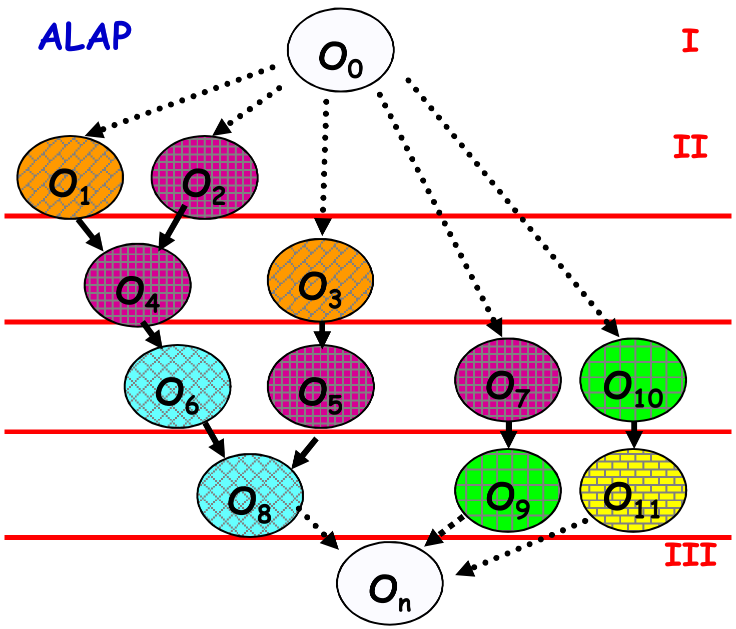
```
ILS (G_S (V, E), λ)  ←  vincolo temporale
{
    A = {a_k = 1; k = 1, …, N_R};
    Calcola T^L ALAP (G_S (V, E), λ);
    l = 1;                                        Back Tracking
    repeat
    {
        Restart = false;
        for k = 1, …, N_R
        {
            … idem …;
            if (|U_lk| + |E_lk| > a_k) Aggiorna a_k = a_k + 1, Restart = true;
        }
        if (Restart)
        else for k = 1, …, N_R
                  … idem …;
        l = l + 1;
    } until (v_n è schedulato);
    return (T, A);
}
```

… L'algoritmo ILS

IED

Ipotizzando che il vincolo sulla latenza sia $\lambda = 4$ e che siano disponibili due tipologie di risorse (moltiplicatore e ALU), entrambe con tempo di esecuzione unitario, posto inizialmente $A = [*/x, +/-/<] = [1,1]$, si ottiene:

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}
t_i^L	1	1	2	2	3	3	3	4	4	3	4



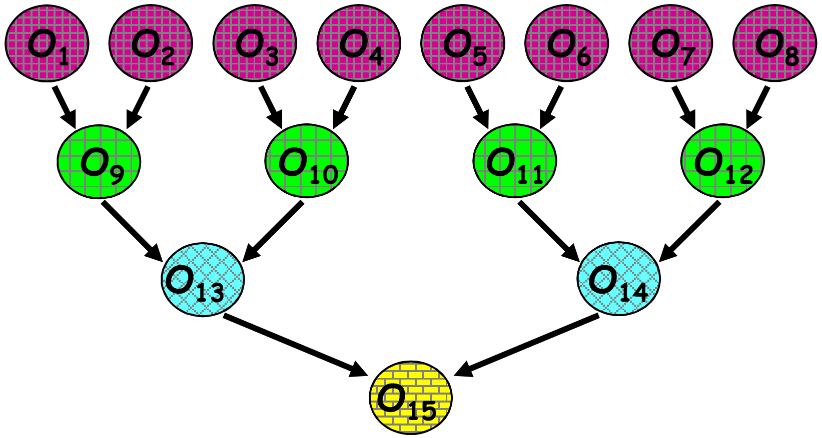
iterazione			Operazioni		A
			candidate (slack)	schedulate	[1,1]
I	step 1	*/x	$O_1(0), O_2(0), O_3(1), O_7(2)$	(restart)	[2,1]
		+/-/<	$O_{10}(2)$		
II	step 1	*/x	$O_1(0), O_2(0), O_3(1), O_7(2)$	O_1, O_2	
		+/-/<	$O_{10}(2)$	O_{10}	
	step 2	*/x	$O_3(0), O_4(0), O_7(1)$	O_3, O_4	
		+/-/<	$O_{11}(2)$	O_{11}	
	step 3	*/x	$O_5(0), O_7(0)$	O_5, O_7	
		+/-/<	$O_6(0)$	O_6	
	step 4	*/x	-		
		+/-/<	$O_8(0), O_9(0)$	restart	[2,2]
III	step 1	…	…	…	
	step 2	…	…	…	
	step 3	…	…	…	
	step 4	…	…	O_8, O_9	

Risorse necessarie: 2 moltiplicatori, 2 ALU

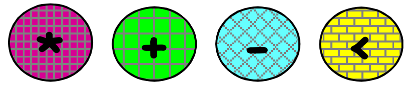
SSG: come per VSLS

Limiti degli algoritmi VSLS e ILS ...

G_S (V, E)



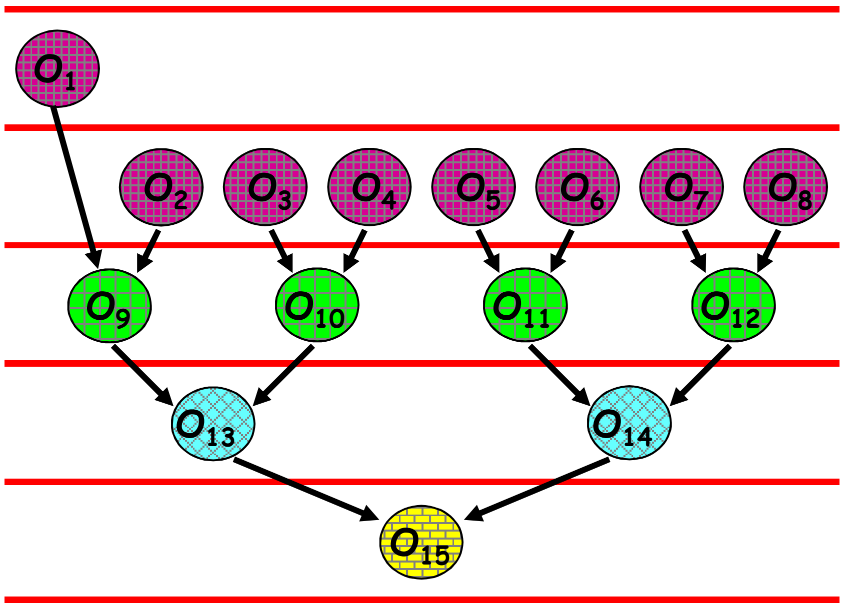
4 tipologie di risorse:



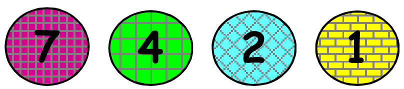
$d_* = d_+ = d_- = d_< = 1$

Vincolo sulla latenza: $\lambda = 5$

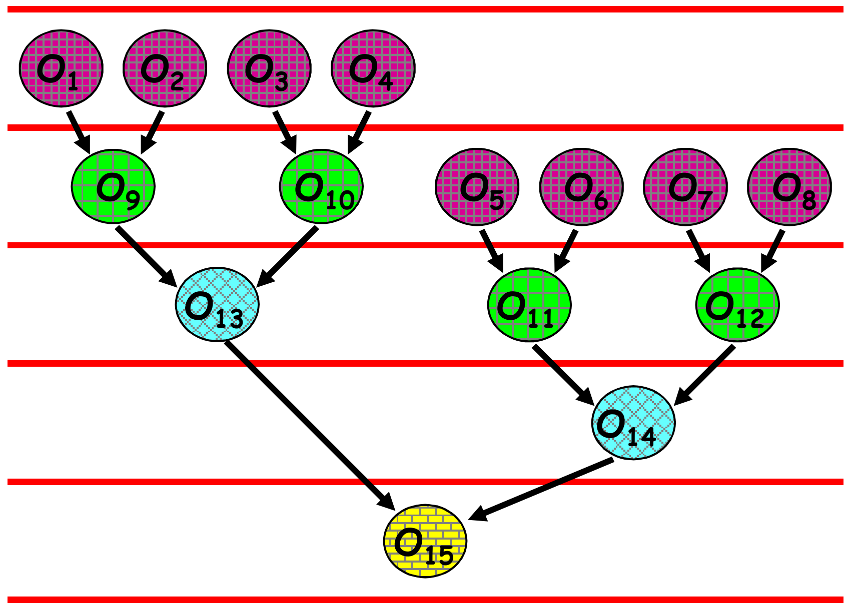
VSLS SSG



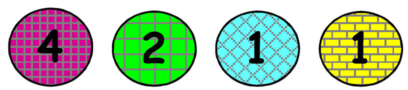
Risorse necessarie:



ILS SSG (5 iterazioni !!!)



Risorse necessarie:



… e possibili estensioni

```
VSLS* / ILS* (G_S (V, E), λ)
{
    A = {a_k = 1; k = 1, …, N_R};
    …
    LB = {LB_k; k = 1, …, N_R};
    l = 1;
    repeat
    {
        for k = 1, …, N_R
        {
            a_k = max {a_k, LB_k};
            …
            idem
            …
            Aggiorna LB_k;
        }
        l = l + 1;
    } until (v_n è schedulato);
    return (T, A);
}
```

Lower Bound

Back Tracking

aggiornamento di LB_k
in ogni step $l \geq ASAP_k$:

$\lambda_k = \lambda_k - 1;$

$N_k = N_k - |E_{lk}|;$

$LB_k = \lceil N_k / \lambda_k \rceil;$

identificazione di LB_k
da assumere come valore iniziale di a_k:

$V_k = \{v_i \in V : R(v_i) = k\};$

(1) $\lambda = \lambda_{min}$:

$Z_k = \{v_i \in V_k : \mu_i = 0 \text{ (ovvero } t_i^L = t_i^S = t_i)\};$

$Z'_{lk} = \{v_i \in Z_k : l - d_k + 1 \leq t_i \leq l\},\ l = 1, \ldots, \lambda;$

$$\nu_k = \max_{1 \leq l \leq \lambda} \{|Z'_{lk}|\};$$

(2) $\forall\ \lambda$:

$$ALAP_k = \max_{v_i \in V_k} \{t_i^L\},\ ASAP_k = \min_{v_i \in V_k} \{t_i^S\};$$

$\lambda_k = ALAP_k - ASAP_k + d_k;$ (°)

$N_k = |V_k| * d_k;$

$n_k = \lceil N_k / (\lfloor \lambda_k / d_k \rfloor * d_k) \rceil;$ (*)

$LB_k = \max \{\nu_k, n_k\}$

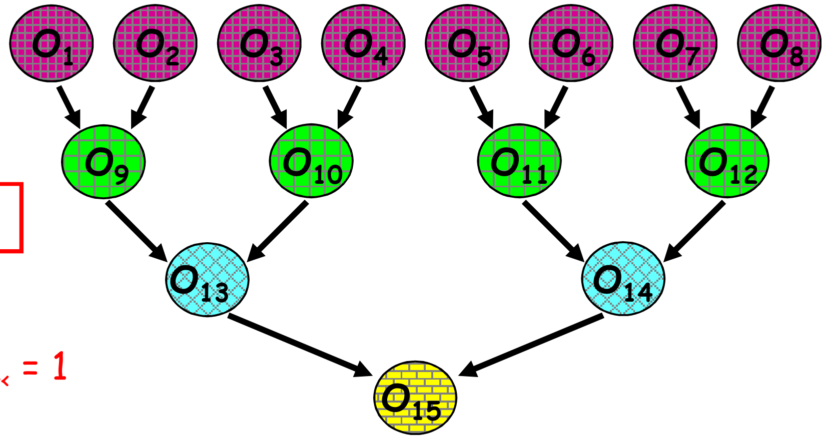
Gli algoritmi VSLS* e ILS* ...

G_S (V, E)

$|V_*| = 8$
$|V_+| = 4$
$|V_-| = 2$
$|V_<| = 1$

(°) $\lambda_k = \lambda$???

$a_* = 2,\ a_+ = a_- = a_< = 1$



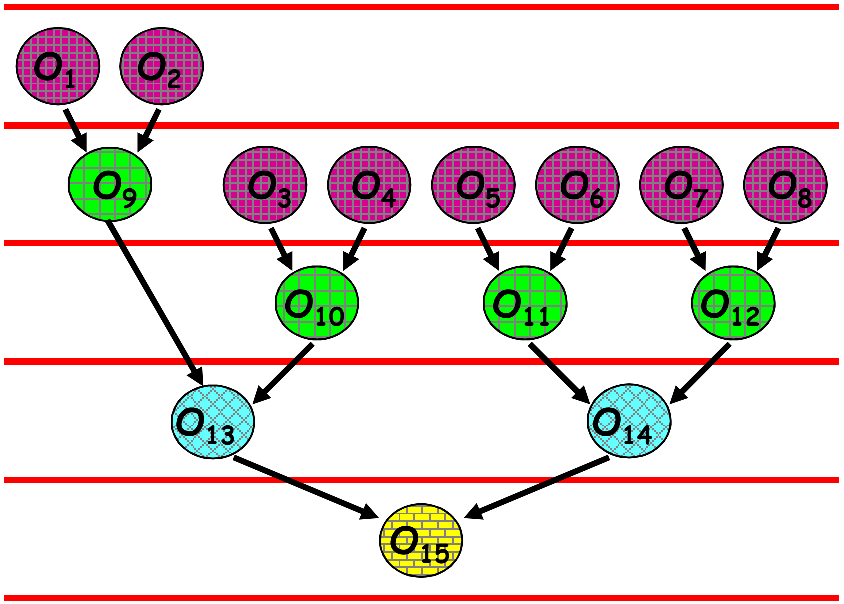
4 tipologie di risorse:

* + – <

$d_* = d_+ = d_- = d_< = 1$

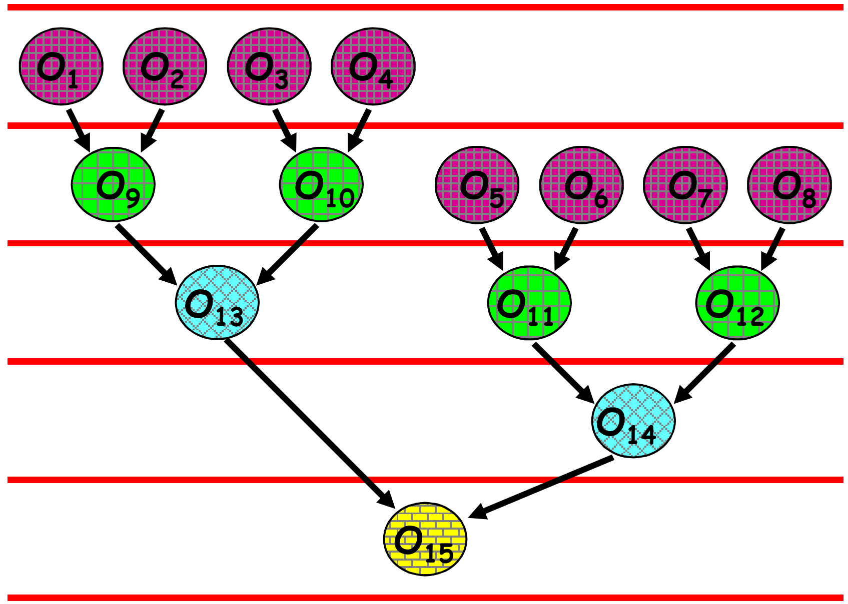
Vincolo sulla latenza: $\lambda = 5$

VSLS* SSG



Risorse necessarie: 6 3 2 1

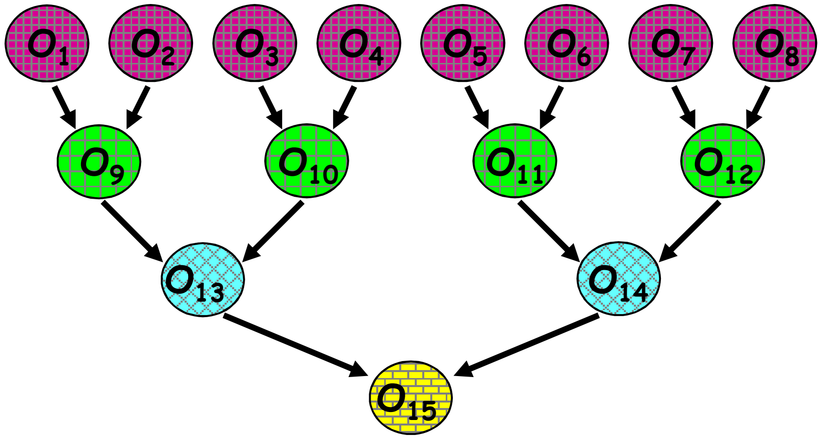
ILS* SSG (4 iterazioni !!!)



Risorse necessarie: 4 2 1 1

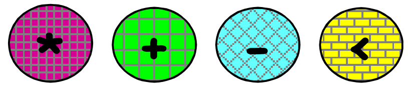
... Gli algoritmi VSLS* e ILS* ...

G_S (V, E)



$|V_*| = 8$
$|V_+| = 4$
$|V_-| = 2$
$|V_<| = 1$

4 tipologie di risorse:



$d_* = d_+ = d_- = d_< = 1$

Vincolo sulla latenza: $\lambda = 5$

$\boxed{\lambda_k = ALAP_k - ASAP_k + d_k}$

$\Downarrow$

$ALAP_* = 2,\ ASAP_* = 1,\ \lambda_* = 2$

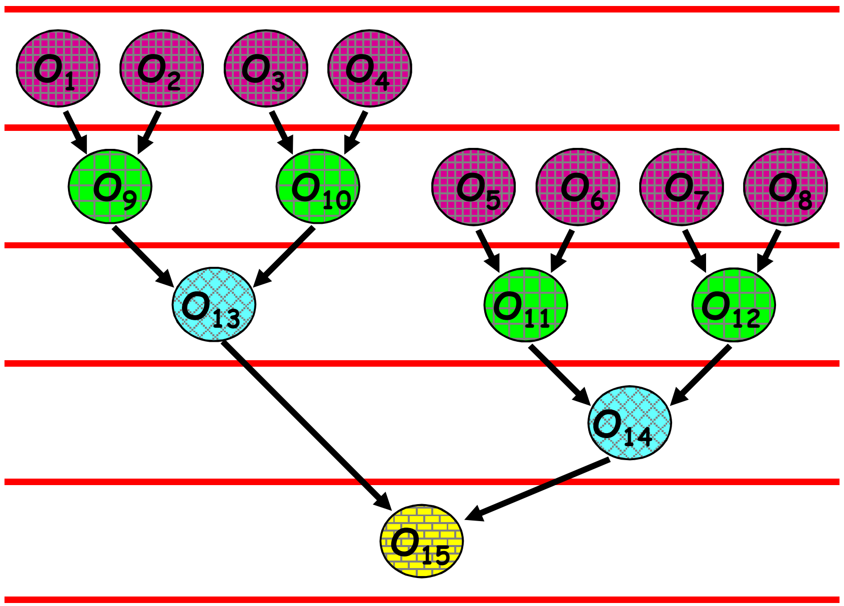
$ALAP_+ = 3,\ ASAP_+ = 2,\ \lambda_+ = 2$

$ALAP_- = 4,\ ASAP_- = 3,\ \lambda_- = 2$

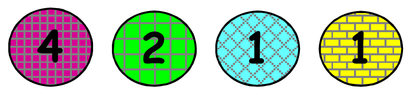
$ALAP_< = 5,\ ASAP_< = 4,\ \lambda_< = 2$

$a_* = 4,\ a_+ = 2,\ a_- = a_< = 1$

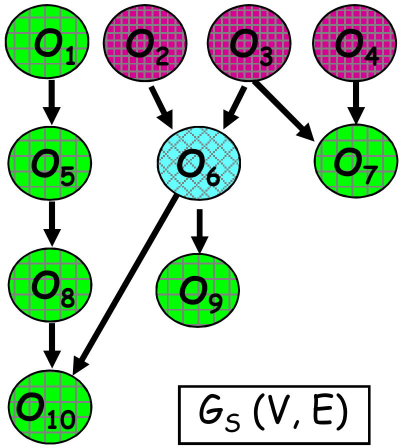
VSLS* / ILS* SSG



Risorse necessarie:

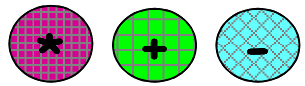


... Gli algoritmi VSLS* e ILS*



G_S (V, E)

3 tipologie di risorse:



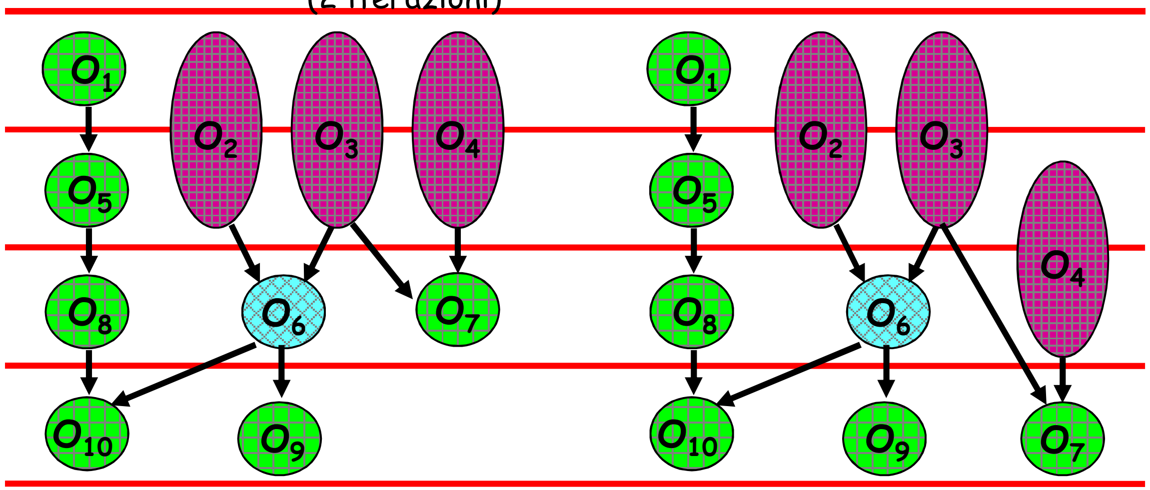
$d_* = 2$, $d_+ = d_- = 1$

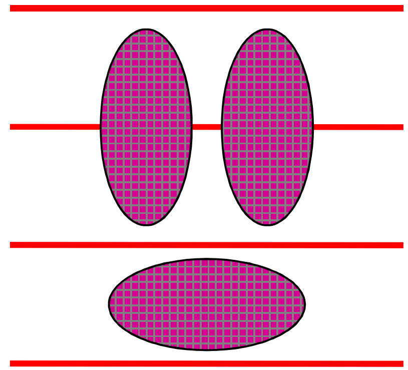
Vincolo sulla latenza: $\lambda = 4$

(*) $n_k = \lceil N_k / \lambda_k \rceil$???

$\lvert V_+ \rvert = 6$	$\lvert V_* \rvert = 3$	$\lvert V_- \rvert = 1$
$\lvert Z_+ \rvert = 5$	$\lvert Z_* \rvert = 2$	$\lvert Z_- \rvert = 1$
$\nu_+ = 2$	$\nu_* = 2$	$\nu_- = 1$
$ALAP_+ = 4$	$ALAP_* = 2$	$ALAP_- = 3$
$ASAP_+ = 1$	$ASAP_* = 1$	$ASAP_- = 3$
$\lambda_+ = 4$	$\lambda_* = 3$	$\lambda_- = 1$
$N_+ = 6$	$N_* = 6$	$N_- = 1$
$n_+ = 2$	$n_* = 2$	$n_- = 1$
$\mathbf{a_+ = 2}$	$\mathbf{a_* = 2}$	$\mathbf{a_- = 1}$

ASAP SSG | ILS* SSG (2 iterazioni) | ALAP SSG | VSLS* SSG





3 2 1 Risorse necessarie 3 3 1

$n_k = \lceil N_k / (\lfloor \lambda_k / d_k \rfloor * d_k) \rceil$

$\mathbf{a_* = 3}$

L'algoritmo Force-Directed Scheduling (FDS) ...

Per time frame di un'operazione si intende l'insieme degli step in cui essa può essere schedulata:

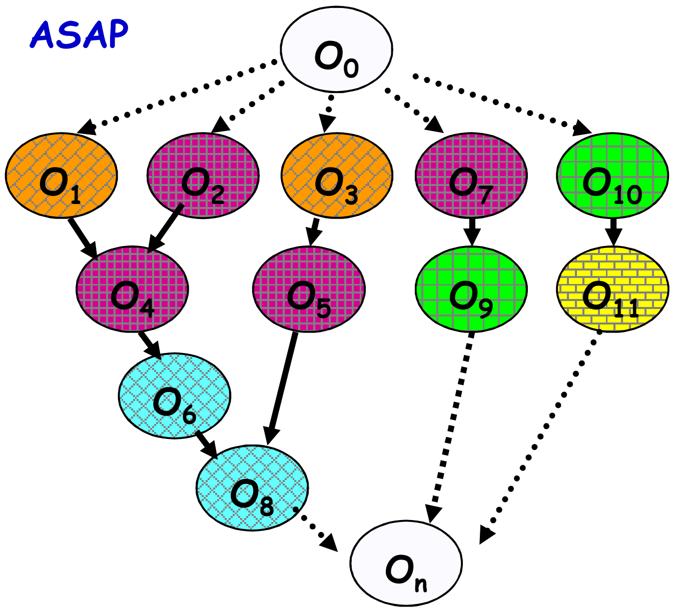
$$TF_i = \{[t_i^S, t_i^L];\ i = 0, 1, \dots, n\}.$$

L'estensione di un time frame dipende dalla mobilità della corrispondente operazione:

$$w_i = \mu_i + 1,\ i = 0, 1, \dots, n.$$

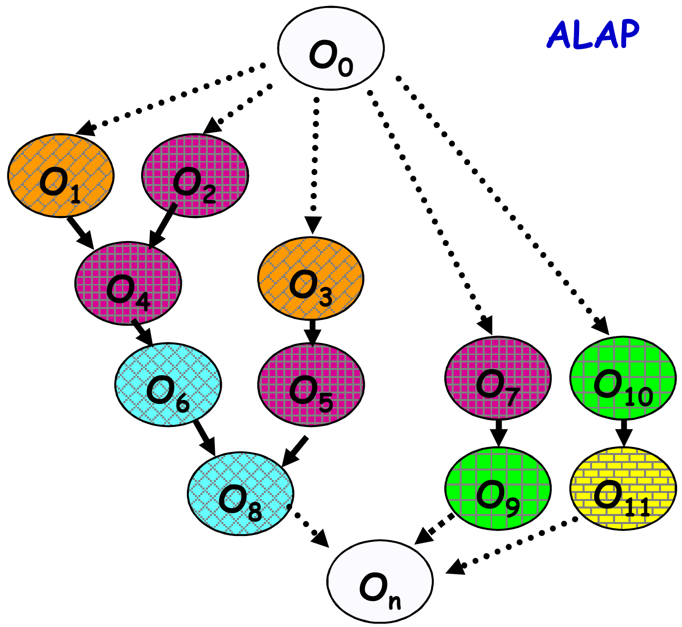
IED: ipotizzando che il vincolo sulla latenza sia $\lambda = 4$
e che il tempo di esecuzione delle operazioni sia unitario, si ottiene:

ASAP



O_i	t_i^S	t_i^L	TF_i	μ_i	w_i
O_1	1	1	[1,1]	0	1
O_2	1	1	[1,1]	0	1
O_3	1	2	[1,2]	1	2
O_4	2	2	[2,2]	0	1
O_5	2	3	[2,3]	1	2
O_6	3	3	[3,3]	0	1
O_7	1	3	[1,3]	2	3
O_8	4	4	[4,4]	0	1
O_9	2	4	[2,4]	2	3
O_{10}	1	3	[1,3]	2	3
O_{11}	2	4	[2,4]	2	3

ALAP



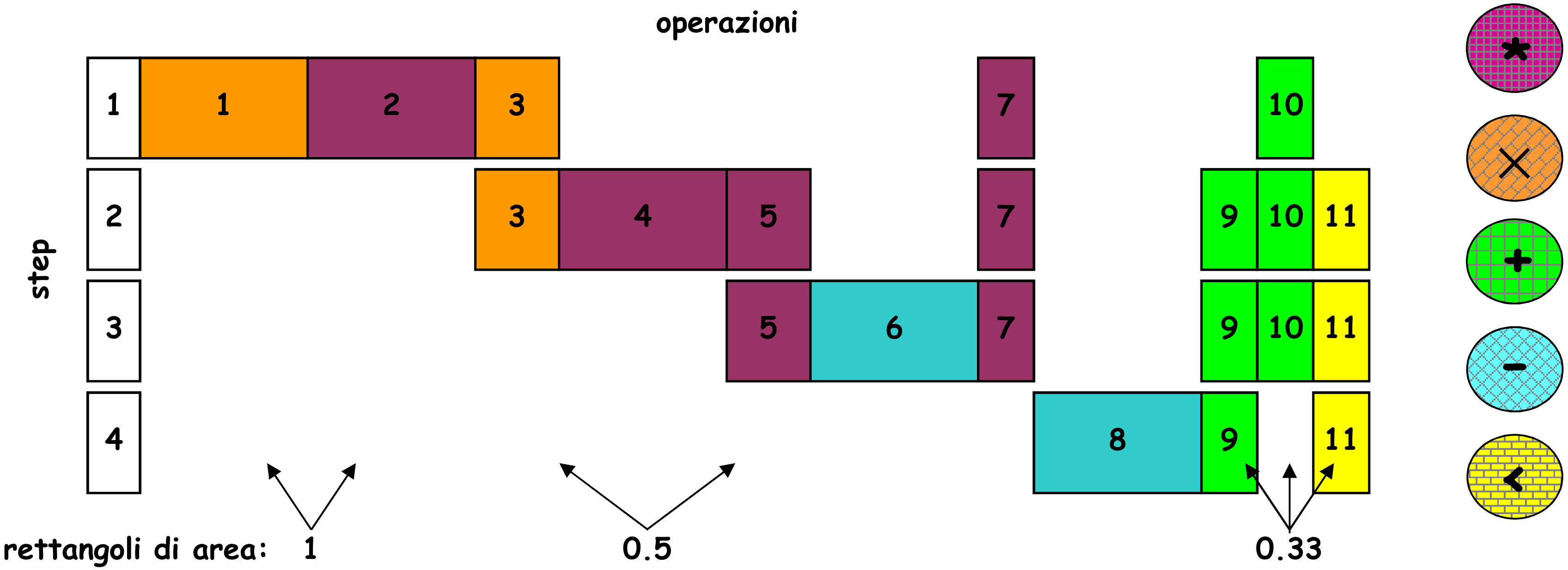
... L'algoritmo FDS ...

Nell'ipotesi che $d_i = 1, \forall v_i \in V$,
per probabilità di un'operazione si intende una funzione che vale
il reciproco dell'estensione del corrispondente time frame all'interno di esso, 0 all'esterno:

$$p_i(s) = \begin{cases} 1/w_i & \text{se } s \in TF_i \\ 0 & \text{se } s \notin TF_i \end{cases} \qquad \forall\ i,\ s = 1, 2, \ldots, \lambda$$

Quanto più esteso è il time frame di un'operazione, tanto minore è la probabilità, per ognuno dei passi che lo compongono, che l'operazione sia schedulata in esso. Un'operazione è vincolata ad essere schedulata in uno specifico passo se il corrispondente time frame ha estensione unitaria.

IED: non sussistono alternative per lo scheduling step delle operazioni
O_1, O_2, O_4, O_6, O_8, aventi tutte time frame di estensione unitaria:



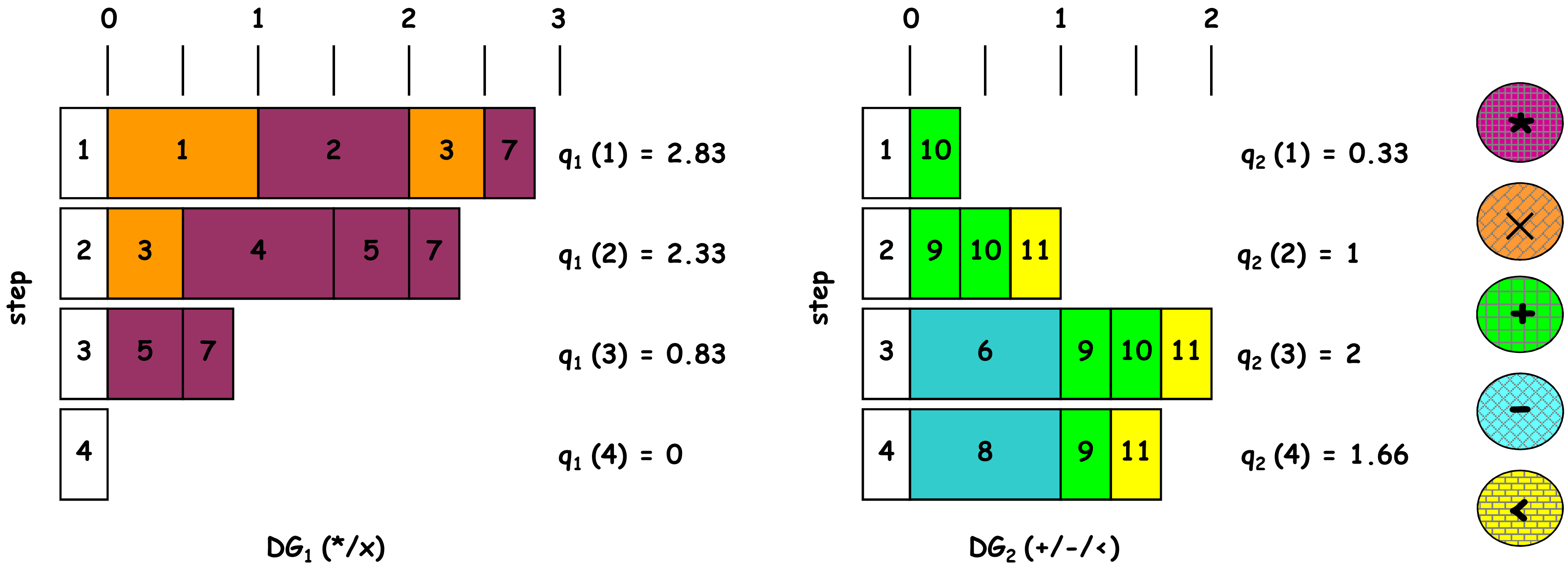
... L'algoritmo FDS ...

Per distribuzione $q_k(s)$ di una tipologia di risorse k nel generico step s si intende la somma delle probabilità delle operazioni eseguibili con la tipologia di risorse k e schedulabili in s:

$$q_k(s) = \Sigma_{i\,:\,R(v_i)\,=\,k}\ p_i(s),\ k = 1, 2, \ldots, N_R,\ s = 1, 2, \ldots, \lambda.$$

Per ogni tipologia di risorse k, i valori di $q_k(s)$ sono congiuntamente rappresentabili in termini di grafo di distribuzione (Distribution Graph) DG_k.

IED: ipotizzando due tipologie di risorse (k = 1: */x, k = 2: +/-/<), si ottiene:



Per ogni tipologia di risorse k, quanto più uniformi sono i valori di $q_k(s)$ in DG_k, tanto meglio saranno sfruttate le risorse nei vari step e, in conseguenza della minore concorrenza tra operazioni, tanto inferiore risulterà il numero delle risorse necessarie.

... L'algoritmo FDS ...

```
FDS (G_S (V, E), λ)   ← vincolo temporale
{
    repeat
    {
        Calcola i time frame e le probabilità delle operazioni;
        Determina il grafo di distribuzione per ogni tipologia di risorse;
        Calcola la self-force e le predecessor/successor (indirect) forces che
        derivano dalla schedulazione di ciascuna operazione in ogni step del
        corrispondente time frame;
        Identifica la coppia operazione-step cui corrisponde il valore minimo
        della forza totale e schedula tale operazione nello step selezionato;
    } until (tutte le operazioni sono schedulate);
    return (T, A);
}
```

La metafora delle molle:
La forza F esercitata da una molla è proporzionale allo scostamento X delle sue estremità (legge di Hook): F = K X, dove K è la costante elastica della molla.

… L'algoritmo FDS …

Per ogni tipologia di risorse k, il relativo DG_k avrà associato un insieme di molle, una per ciascuno step s (l = 1, 2, …, λ), con costanti elastiche di valore, rispettivamente, $q_k(s)$. Le molle esercitano una forza positiva (di repulsione), negativa (di attrazione) o nulla su ciascuna operazione i : $R(v_i) = k$, in dipendenza del fatto che la schedulazione di i in uno specifico step l del suo time frame comporti rispettivamente un aumento, una diminuzione o nessuna variazione di concorrenza. La schedulazione di un'operazione comporta uno scostamento delle molle, imputabile alla variazione delle probabilità sia dell'operazione schedulata, sia, eventualmente, di tutte quelle il cui time frame risulta conseguentemente ridimensionato. Più precisamente, la schedulazione della generica operazione i nello step l $\in TF_i$ comporta un incremento della probabilità dell'operazione i in l uguale a $1 - 1/w_i$, ed un decremento della stessa in ogni altro step $s \neq l$ compreso in TF_i pari a $1/w_i$. La "self-force" SF (i, l) vale pertanto:

$$SF(i, l) = q_k(l)\,(1 - 1 / w_i) - \Sigma_{s\neq l}\, q_k(s) / w_i = q_k(l) - \big(\Sigma_s\, q_k(s)\big) / w_i,\ \forall\, i;\ \forall\, l, s \in TF_i.$$

Il contributo alla "indirect force" IF (i, l), di attrazione o di repulsione, dovuto ad ogni altra operazione $j \neq i$ ($R(v_j) = h$) il cui time frame TF_j, di estensione w_j, risulterebbe ridimensionato in TF_j', con estensione w_j', in conseguenza della schedulazione di i in l, vale:

$$IF_j(i, l) = \big(\Sigma_{s'}\, q_h(s')\big) / w_j' - \big(\Sigma_s\, q_h(s)\big) / w_j,\ \forall\, i, j \neq i;\ \forall\, l \in TF_i;\ \forall\, s \in TF_j;\ \forall\, s' \in TF_j'.$$

La "total force" F (i, l) corrispondente alla schedulazione di i in l vale conseguentemente:

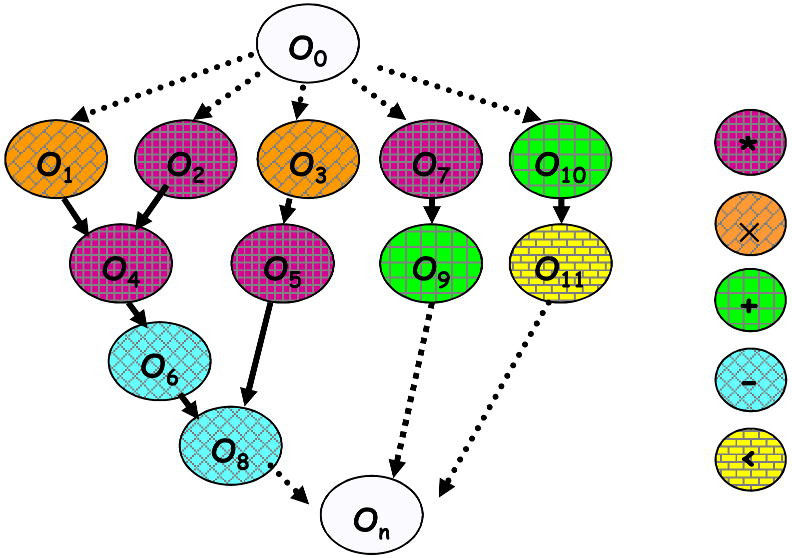
$$F(i, l) = SF(i, l) + IF(i, l) = SF(i, l) + \Sigma_{j\neq i}\, IF_j(i, l),\ \forall\, i;\ \forall\, l \in TF_i.$$

... L'algoritmo FDS ...

IED:
la selezione dello scheduling step per le restanti operazioni $O_3, O_5, O_7, O_9, O_{10}, O_{11}$ procede pertanto come segue:

I iterazione

step	*/x	+/-/<
1	2.83	0.33
2	2.33	1
3	0.83	2
4	0	1.66

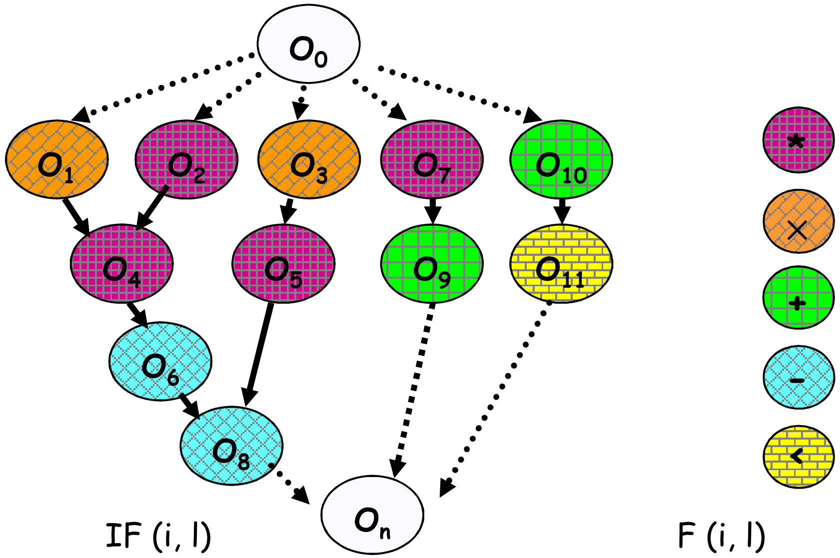


(i, l)	SF (i, l)	IF (i, l)	F (i, l)
(3, 1)	2.83 – (2.83 + 2.33) / 2 = 0.25	0	0.25 + 0 = 0.25
(3, 2)	2.33 – (2.83 + 2.33) / 2 = - 0.25	0.83 – (2.33 + 0.83) / 2 = - 0.75	- 0.25 – 0.75 = - 1
(5, 2)	2.33 – (2.33 + 0.83) / 2 = 0.75	2.83 – (2.83 + 2.33) / 2 = 0.25	0.75 + 0.25 = 1
(5, 3)	0.83 – (2.33 + 0.83) / 2 = - 0.75	0	- 0.75 + 0 = - 0.75
(7, 1)	2.83 – (2.83 + 2.33 + 0.83) / 3 = 0.83	0	0.83 + 0 = 0.83
(7, 2)	2.33 – (2.83 + 2.33 + 0.83) / 3 = 0.33	(2 + 1.66) / 2 – (1 + 2 + 1.66) / 3 = 0.27	0.33 + 0.27 = 0.61
(7, 3)	0.83 – (2.83 + 2.33 + 0.83) / 3 = - 1.16	1.66 – (1 + 2 + 1.66) / 3 = 0.11	- 1.16 + 0.11 = - 1.05

... L'algoritmo FDS ...

I iterazione

step	*/x	+/-/<
1	2.83	0.33
2	2.33	1
3	0.83	2
4	0	1.66

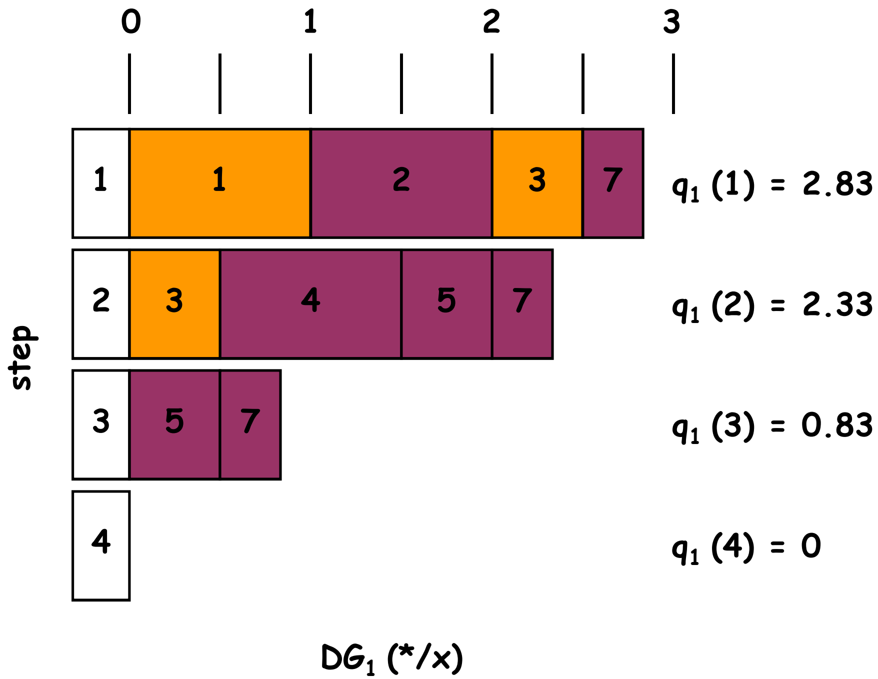


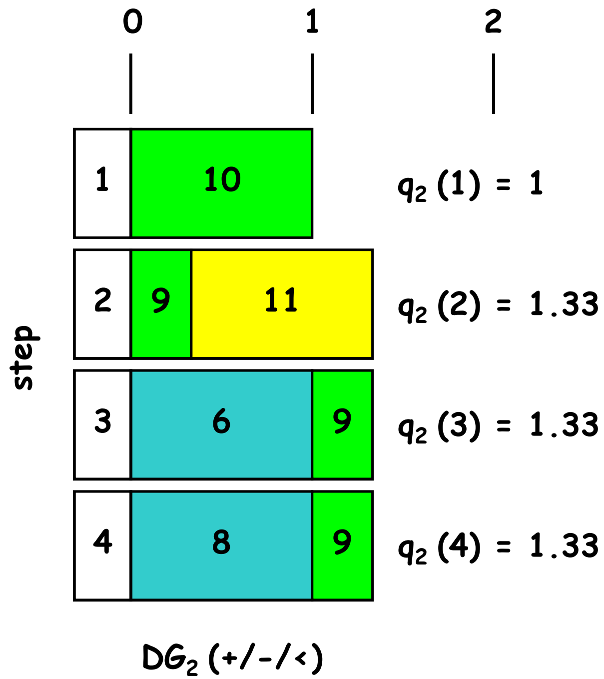
(i, l)	SF (i, l)	IF (i, l)	F (i, l)
(9, 2)	1 – (1 + 2 + 1.66) / 3 = - 0.55	2.83 – (2.83 + 2.33 + 0.83) / 3 = 0.83	- 0.55 + 0.83 = 0.28
(9, 3)	2 – (1 + 2 + 1.66) / 3 = 0.45	(2.83 + 2.33) / 2 – (2.83 + 2.33 + 0.83) / 3 = 0.58	0.45 + 0.58 = 1.03
(9, 4)	1.66 – (1 + 2 + 1.66) / 3 = 0.1	0	0.1 + 0 = 0.1
(10, 1)	0.33 – (0.33 + 1 + 2) / 3 = - 0.78	0	- 0.78 + 0 = - 0.78
(10, 2)	1 – (0.33 + 1 + 2) / 3 = - 0.11	(2 + 1.66) / 2 – (1 + 2 + 1.66) / 3 = 0.28	- 0.11 + 0.28 = 0.17
(10, 3)	2 – (0.33 + 1 + 2) / 3 = 0.89	1.66 – (1 + 2 + 1.66) / 3 = 0.11	0.89 + 0.11 = 1
(11, 2)	1 – (1 + 2 + 1.66) / 3 = - 0.55	0.33 – (0.33 + 1 + 2) / 3 = - 0.78	- 0.55 - 0.78 = - 1.33
(11, 3)	2 – (1 + 2 + 1.66) / 3 = 0.45	(0.33 + 1) / 2 – (0.33 + 1 + 2) / 3 = - 0.45	0.45 – 0.45 = 0
(11, 4)	1.66 – (1 + 2 + 1.66) / 3 = 0.1	0	0.1 + 0 = 0.1

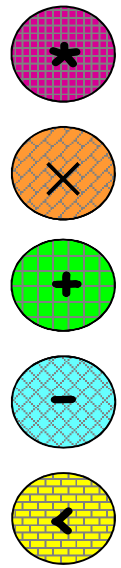
Viene quindi schedulata O_{11} nello step 2 e, conseguentemente, O_{10} nello step 1.

... L'algoritmo FDS ...

II iterazione



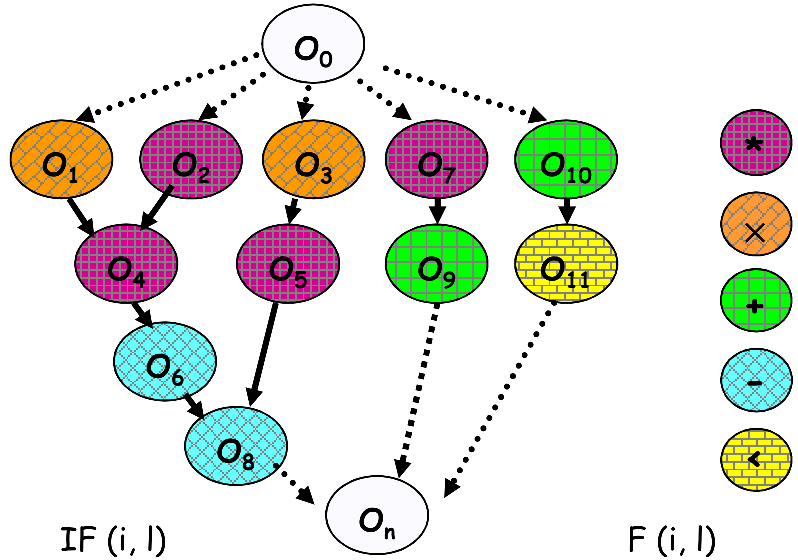




… L'algoritmo FDS …

II iterazione

step	*/x	+/-/<
1	2.83	1
2	2.33	1.33
3	0.83	1.33
4	0	1.33

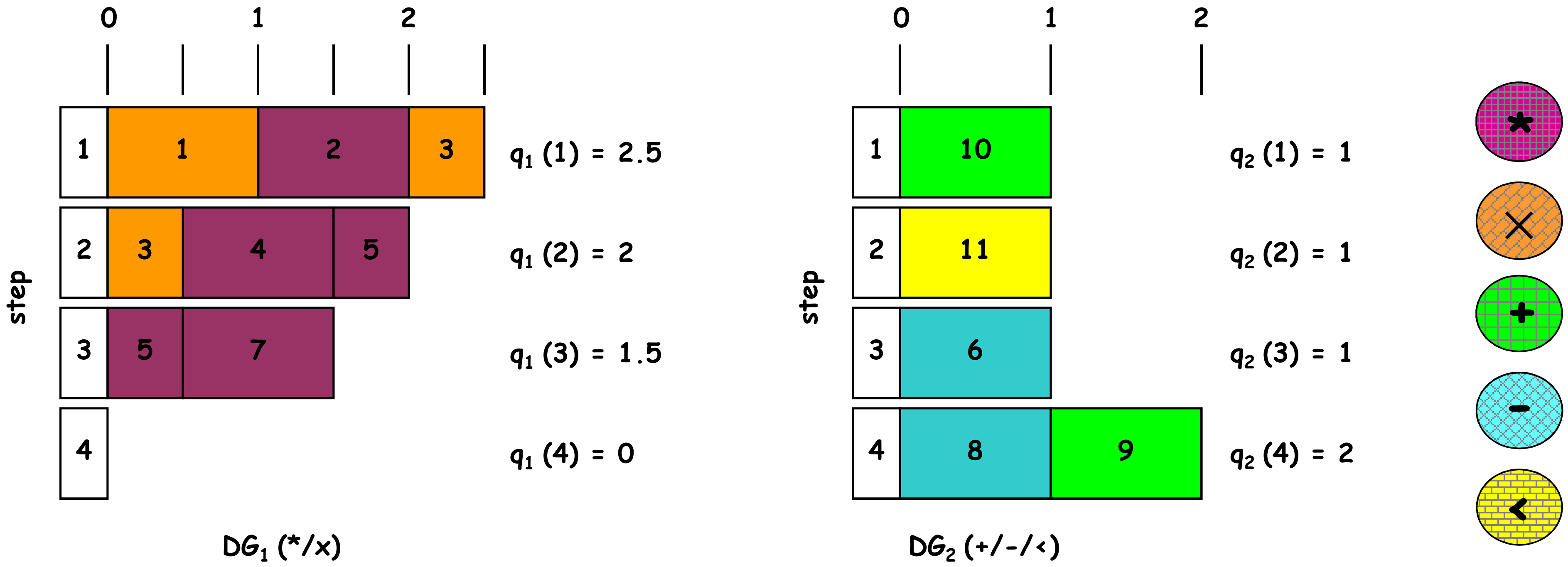


(i, l)	SF (i, l)	IF (i, l)	F (i, l)
(3, 1)	2.83 – (2.83 + 2.33) / 2 = 0.25	0	0.25 + 0 = 0.25
(3, 2)	2.33 – (2.83 + 2.33) / 2 = - 0.25	0.83 – (2.33 + 0.83) / 2 = - 0.75	- 0.25 – 0.75 = - 1
(5, 2)	2.33 – (2.33 + 0.83) / 2 = 0.75	2.83 – (2.83 + 2.33) / 2 = 0.25	0.75 + 0.25 = 1
(5, 3)	0.83 – (2.33 + 0.83) / 2 = - 0.75	0	- 0.75 + 0 = - 0.75
(7, 1)	2.83 – (2.83 + 2.33 + 0.83) / 3 = 0.83	0	0.83 + 0 = 0.83
(7, 2)	2.33 – (2.83 + 2.33 + 0.83) / 3 = 0.33	(1.33 + 1.33) / 2 – (1.33 + 1.33 + 1.33) / 3 = 0	0.33 + 0 = 0.33
(7, 3)	0.83 – (2.83 + 2.33 + 0.83) / 3 = - 1.16	1.33 – (1.33 + 1.33 + 1.33) / 3 = 0	- 1.16 + 0 = - 1.16
(9, 2)	1.33 – (1.33 + 1.33 + 1.33) / 3 = 0	2.83 – (2.83 + 2.33 + 0.83) / 3 = 0.83	0 + 0.83 = 0.83
(9, 3)	1.33 – (1.33 + 1.33 + 1.33) / 3 = 0	(2.83 + 2.33) / 2 – (2.83 + 2.33 + 0.83) / 3 = 0.58	0 + 0.58 = 0.58
(9, 4)	1.33 – (1.33 + 1.33 + 1.33) / 3 = 0	0	0 + 0 = 0

Viene quindi schedulata O_7 nello step 3 e, conseguentemente, O_9 nello step 4.

... L'algoritmo FDS ...

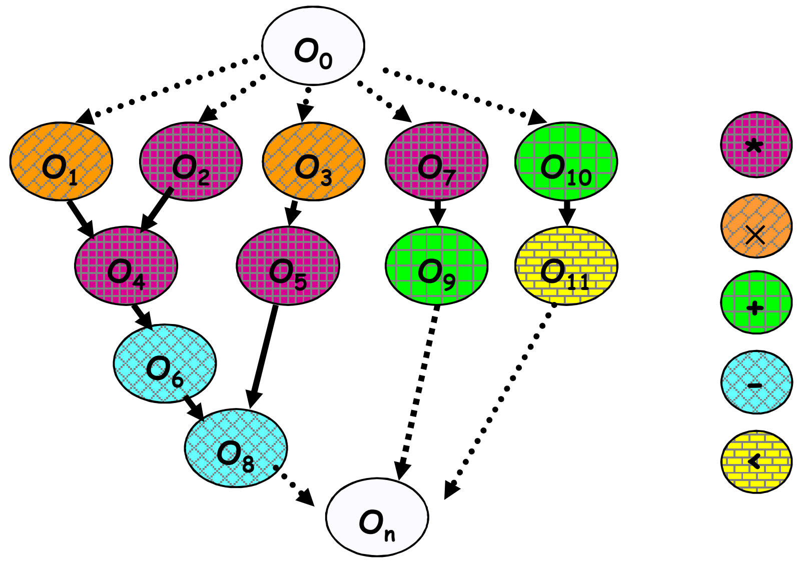
III iterazione



… L'algoritmo FDS …

III iterazione

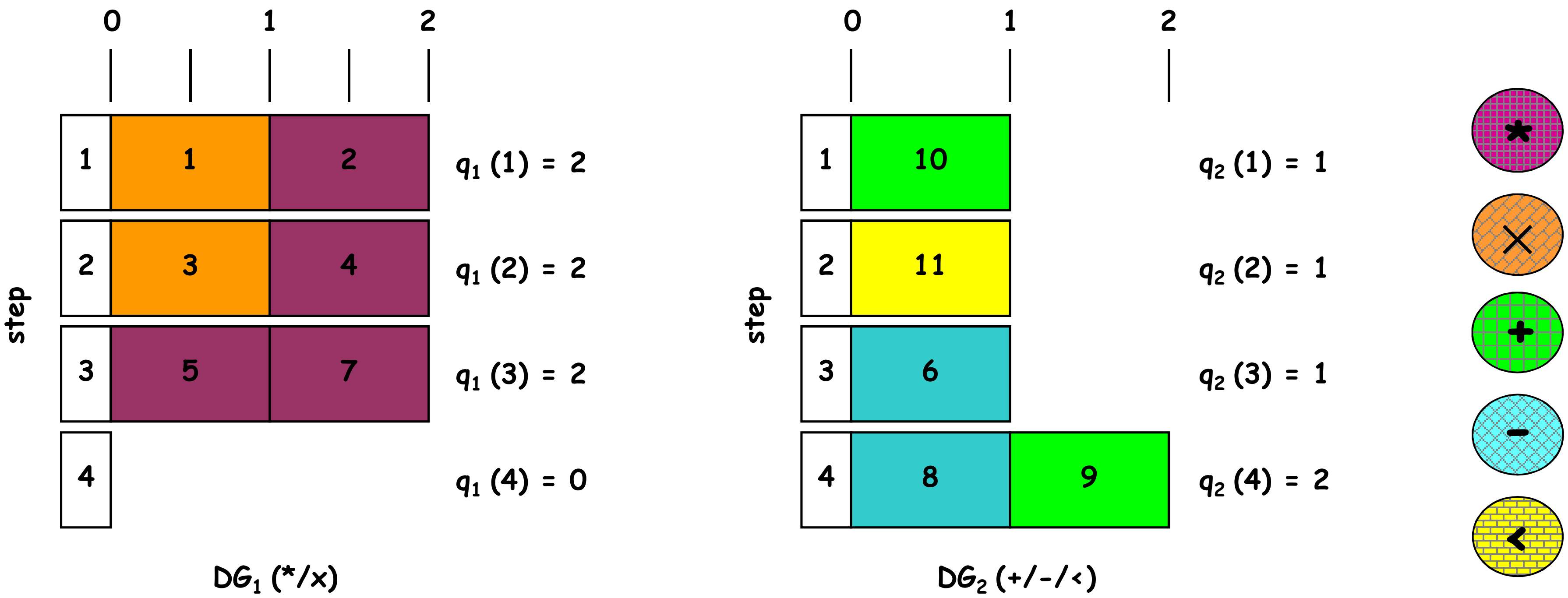
step	*/x	+/-/<
1	2.5	1
2	2	1
3	1.5	1
4	0	2



(i, l)	SF (i, l)	IF (i, l)	F (i, l)
(3, 1)	2.5 - (2.5 + 2) / 2 = 0.25	0	0.25 + 0 = 0.25
(3, 2)	2 - (2.5 + 2) / 2 = - 0.25	1.5 - (2 + 1.5) / 2 = - 0.25	- 0.25 - 0.25 = - 0.5
(5, 2)	2 - (2 + 1.5) / 2 = 0.25	2.5 - (2.5 + 2) / 2 = 0.25	0.25 + 0.25 = 0.5
(5, 3)	1.5 - (2 + 1.5) / 2 = - 0.25	0	- 0.25 + 0 = - 0.25

Viene quindi schedulata O_3 nello step 2 e, conseguentemente, O_5 nello step 3.

... L'algoritmo FDS ...



Risorse necessarie: 2 moltiplicatori, 2 ALU

SSG: come per VSLS e ILS

... L'algoritmo FDS

Calcolo delle forze in presenza di operazioni "multicycle" ($\exists\ v_i \in V : d_i > 1$):

$$U_i(s) = \begin{cases} 1 & \text{se } s \in TF_i \\ 0 & \text{se } s \notin TF_i \end{cases} \qquad \forall\ i,\ s = 1, 2, \ldots, \lambda.$$

$$p_i(s) = \left(\sum_{m=0}^{d_i-1} U_i(s-m)\right) / w_i,\ \forall\ i,\ s = 1, 2, \ldots, \lambda.$$

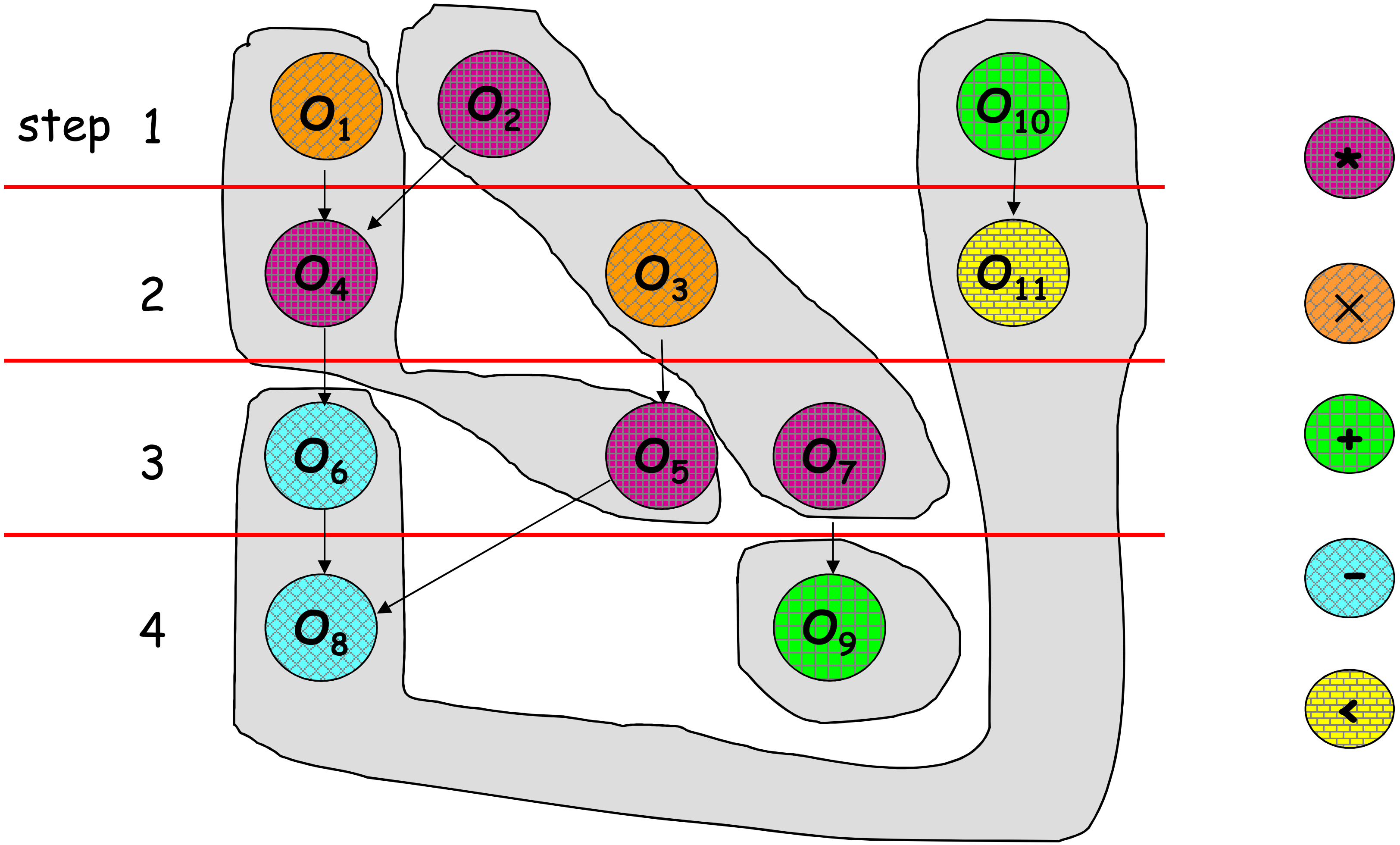
$$q_k(s) = \sum_{i\,:\,R(v_i) = k} p_i(s),\ k = 1, 2, \ldots, N_R,\ s = 1, 2, \ldots, \lambda.$$

$$SF(i, l) = \sum_{m=0}^{d_i-1} \left[q_k(l+m) - \left(\sum_s q_k(s+m)\right) / w_i\right],\ \forall\ i;\ \forall\ l, s \in TF_i.$$

$$IF_j(i, l) = \sum_{m=0}^{d_j-1} \left[\left(\sum_{s'} q_h(s'+m)\right) / w_j' - \left(\sum_s q_h(s+m)\right) / w_j\right],\ \forall\ i, j \neq i;\ \forall\ l \in TF_i;\ \forall\ s \in TF_j;\ \forall\ s' \in TF_j'.$$

$$F(i, l) = SF(i, l) + IF(i, l) = SF(i, l) + \sum_{j \neq i} IF_j(i, l),\ \forall\ i;\ \forall\ l \in TF_i.$$

Allocazione delle risorse di elaborazione



SSG:

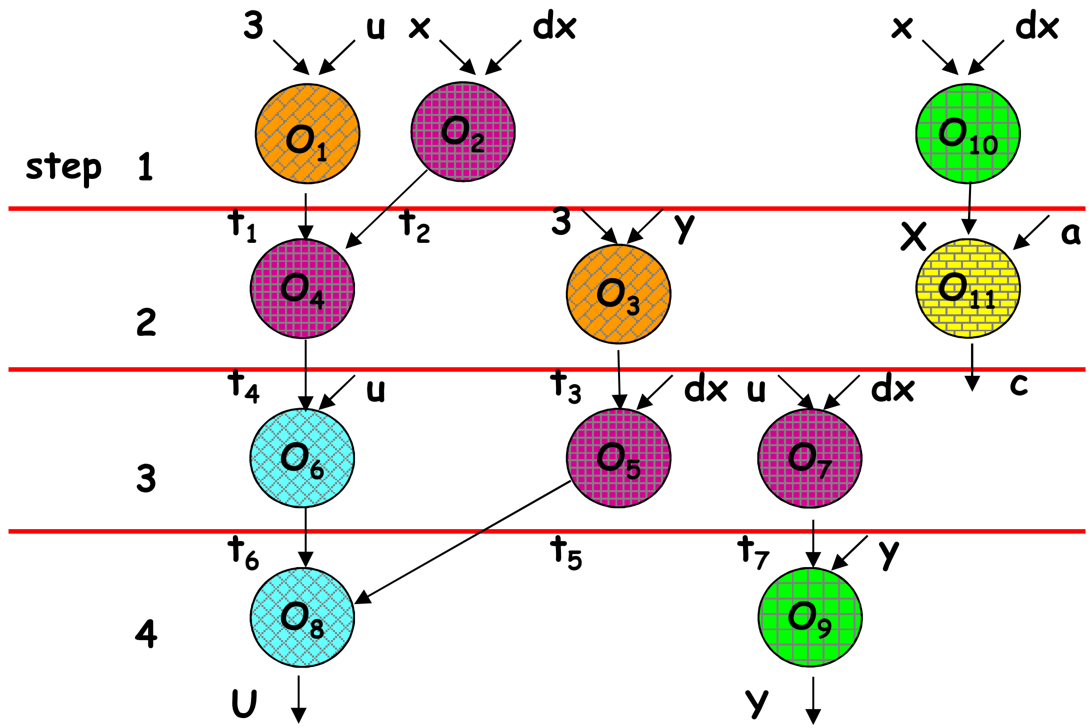
Latenza
4 t.u.

Risorse necessarie
2 moltiplicatori
2 ALU

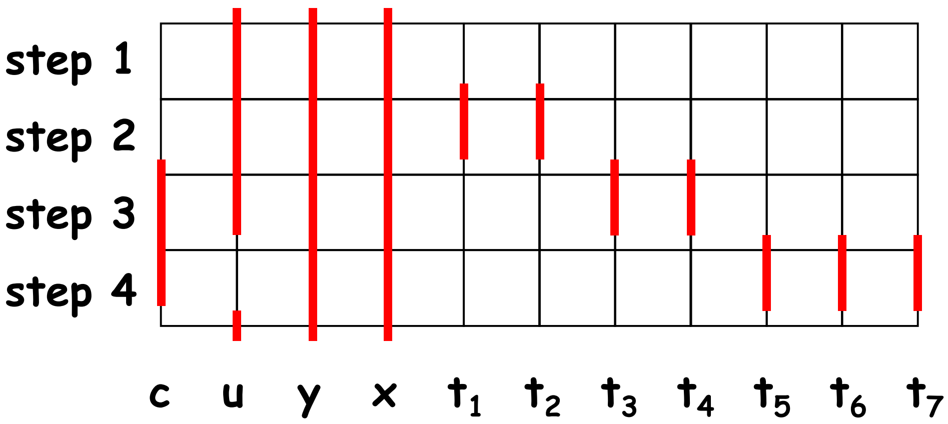
Risorse necessarie
2 moltiplicatori
1 ALU
1 addizionatore

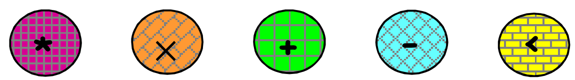
Allocazione dei registri ...

Un insieme di variabili possono condividere uno stesso registro se i relativi tempi di vita non si sovrappongono (variabili compatibili).

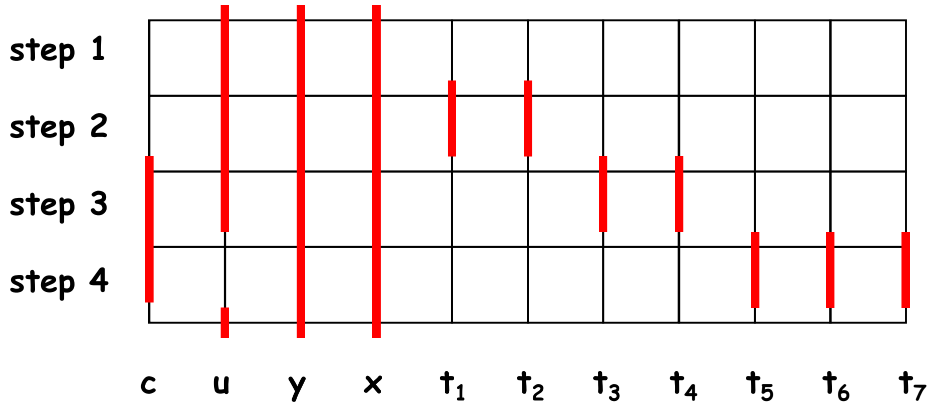


Tempo di vita di una variabile: intervallo fra lo step in cui la variabile è generata e l'ultimo step in cui è utilizzata.





... Allocazione dei registri



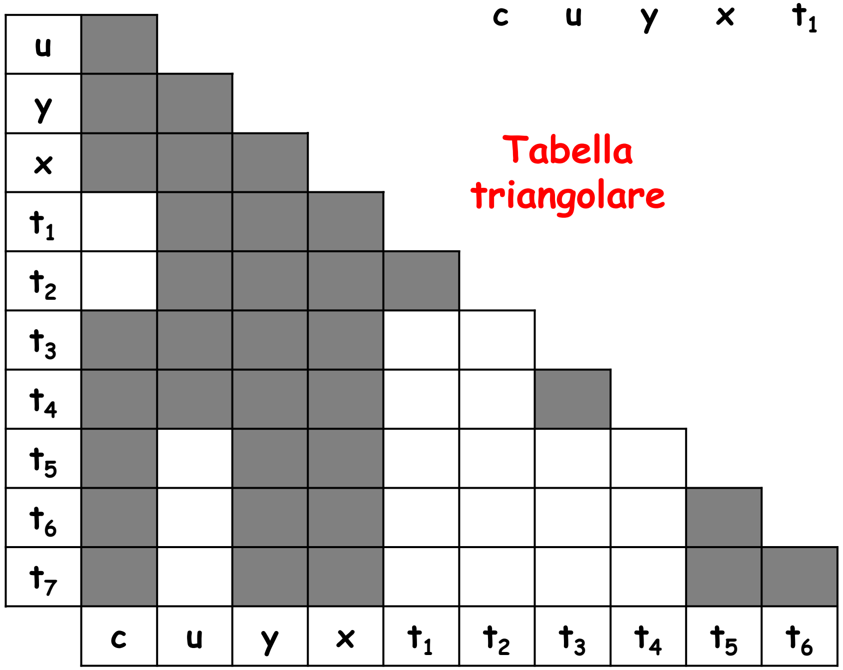


Tabella triangolare

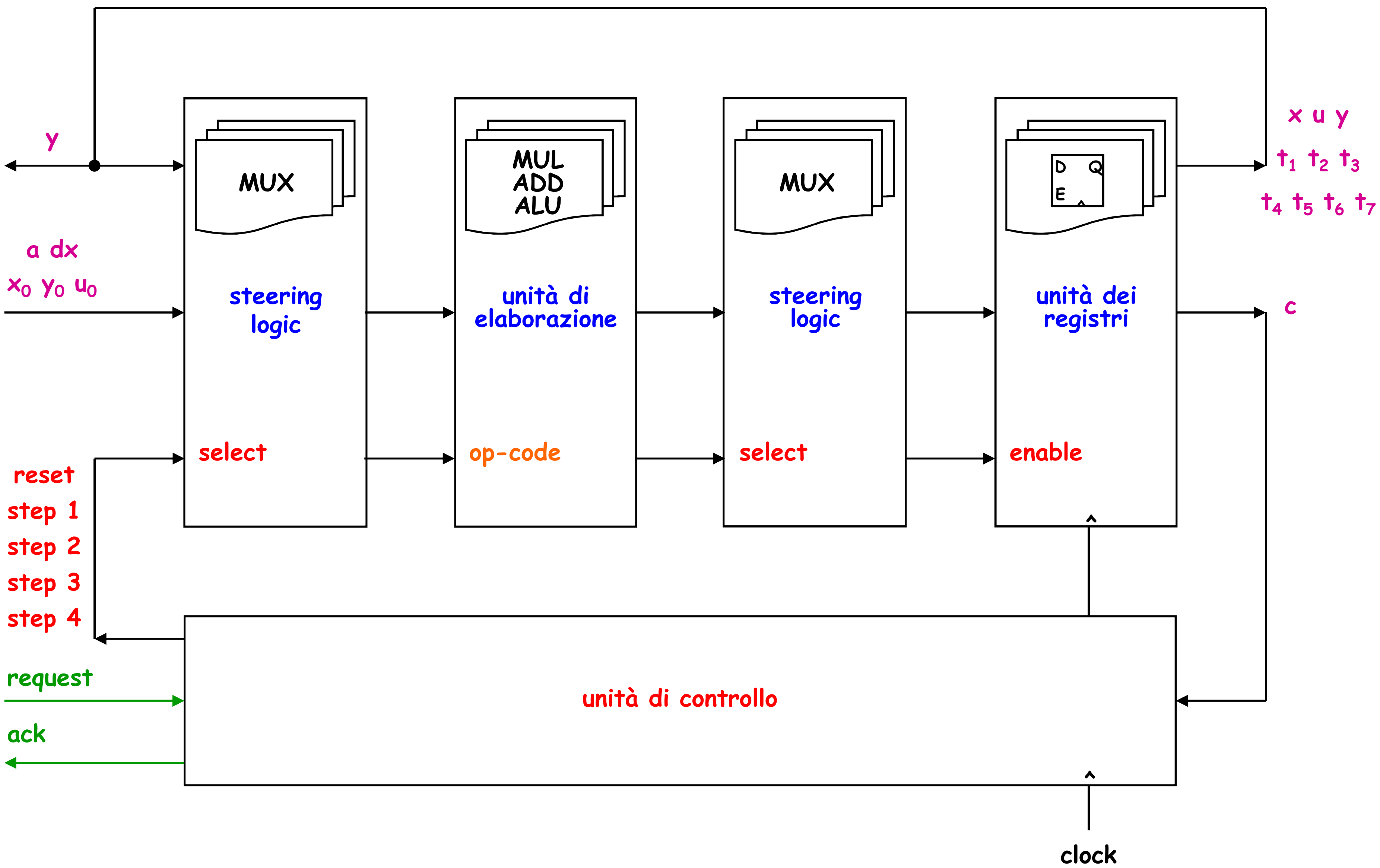
Classi massime di compatibilità

$\{t_2,t_4,t_5\}$, $\{t_2,t_4,t_6\}$, $\{t_2,t_4,t_7\}$,
$\{t_2,t_3,t_5\}$, $\{t_2,t_3,t_6\}$, $\{t_2,t_3,t_7\}$,
$\{t_1,t_4,t_5\}$, $\{t_1,t_4,t_6\}$, $\{t_1,t_4,t_7\}$,
$\{t_1,t_3,t_5\}$, $\{t_1,t_3,t_6\}$, $\{t_1,t_3,t_7\}$,
$\{u,t_5\}$, $\{u,t_6\}$, $\{u,t_7\}$,
$\{c,t_1\}$, $\{c,t_2\}$, $\{y\}$, $\{x\}$

Una possibile collezione minima di insiemi disgiunti di variabili compatibili e relativi registri:

$\{c\}$	$\{u,t_6\}$	$\{y\}$	$\{x\}$	$\{t_1,t_4,t_5\}$	$\{t_2,t_3,t_7\}$
R_c	R_u	R_y	R_x	R_1	R_2

Data-path & control unit



L'unità di controllo

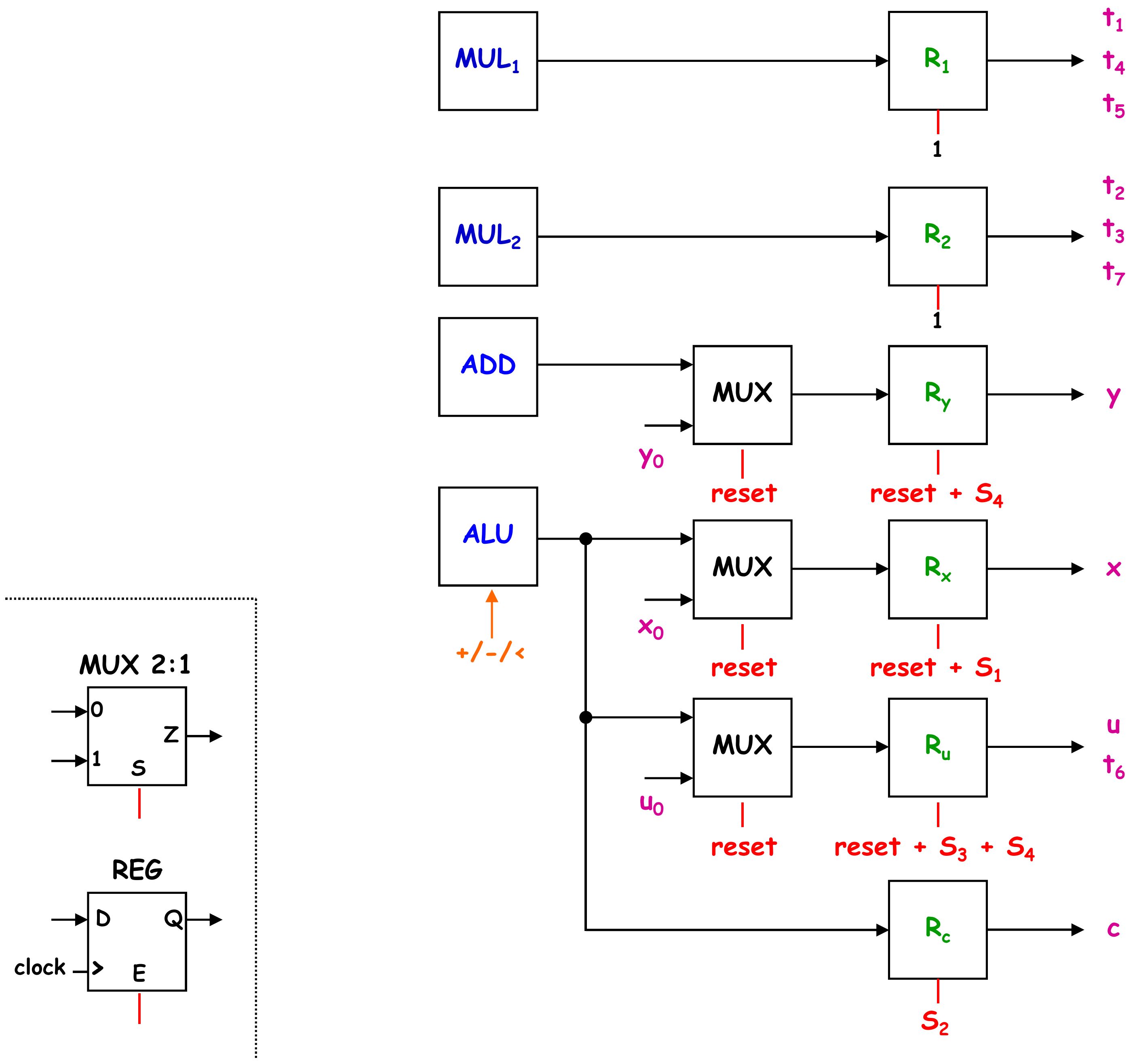
Stato presente	Ingressi		Stato futuro		Uscite					
0	req'	req	0	1	1	0	0	0	0	0
1	-		2		0	1	0	0	0	0
2	-		3		0	0	1	0	0	0
3	-		4		0	0	0	1	0	0
4	c'	c	1	5	0	0	0	0	1	0
5	req'	req	0	5	0	0	0	0	0	1
					Reset	Step 1	Step 2	Step 3	Step 4	ack

L'unità dei registri ...

	Step 1 (S_1)		Step 2 (S_2)		Step 3 (S_3)		Step 4 (S_4)		
	Data	From	Data	From	Data	From	Data	From	Enable
R_c	-	-	c	ALU	c	R_c	-	-	S_2
R_x	X	ALU	x	R_x	x	R_x	x	R_x	S_1
R_y	y	R_y	y	R_y	y	R_y	Y	ADD	S_4
R_u	u	R_u	u	R_u	t_6	ALU	U	ALU	S_3+S_4
R_1	t_1	MUL_1	t_4	MUL_1	t_5	MUL_1	-	-	1
R_2	t_2	MUL_2	t_3	MUL_2	t_7	MUL_2	-	-	1

update	hold	-

... L'unità dei registri



L'unità di elaborazione ...

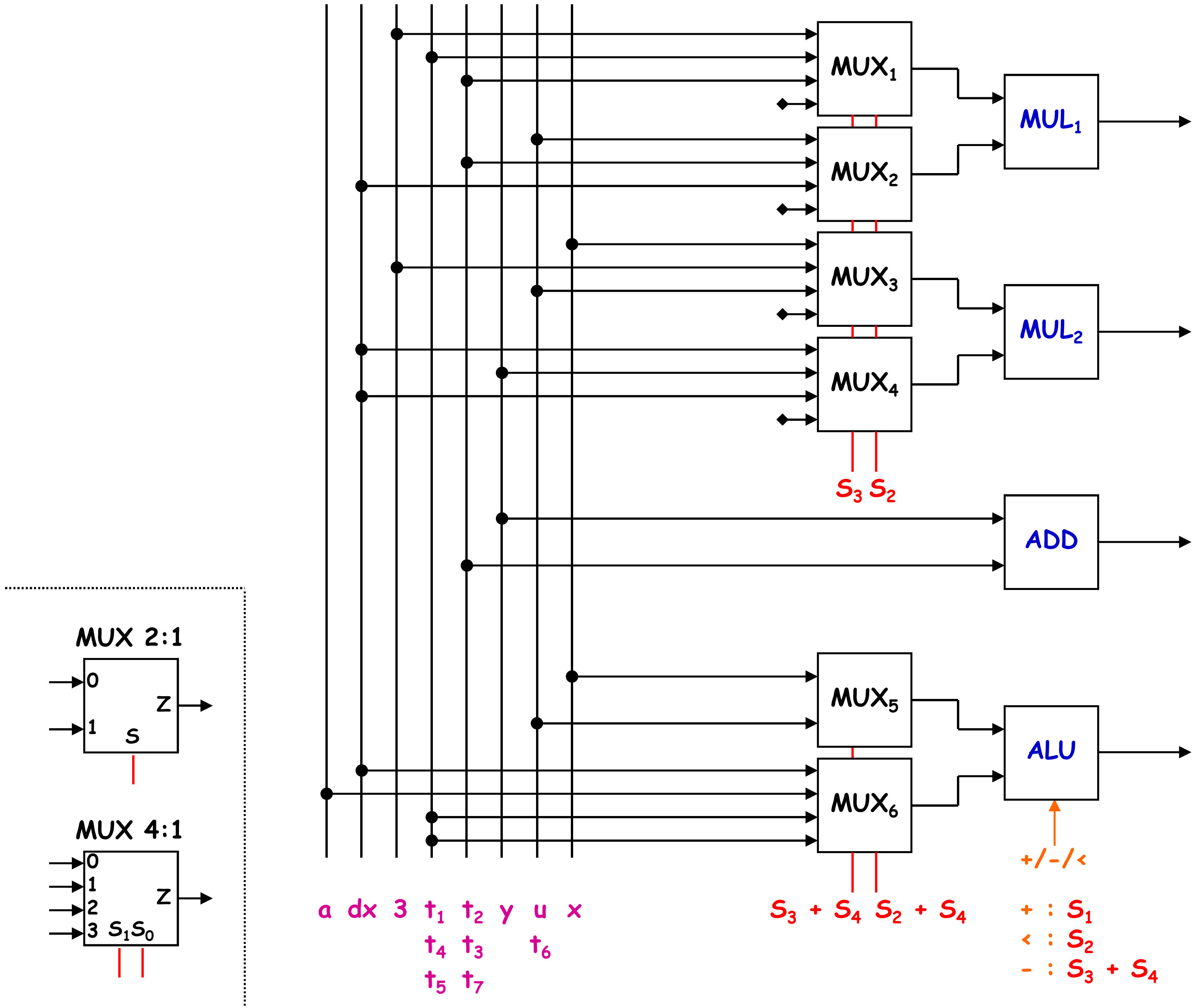
	Step 1		Step 2		Step 3		Step 4		
	OP_1	OP_2	OP_1	OP_2	OP_1	OP_2	OP_1	OP_2	Source
MUL_1	3	u	t_1	t_2	t_3	dx	-	-	$MUX_{1/2}$
MUL_2	x	dx	3	y	u	dx	-	-	$MUX_{3/4}$
ALU	x	dx	X	a	u	t_4	t_6	t_5	$MUX_{5/6}$
ADD	-	-	-	-	-	-	y	t_7	-

I_0	I_1	Select
R_x	R_u	$S_3 + S_4$

MUX_5

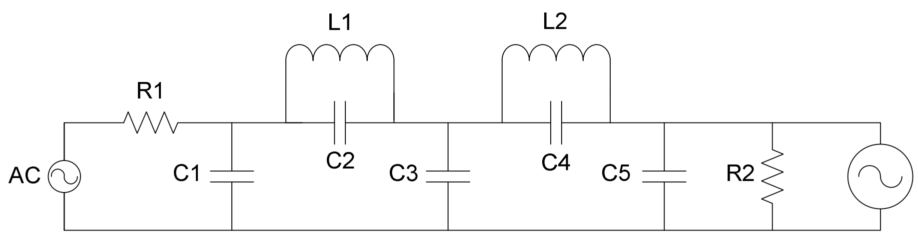
I_0	I_1	I_2	I_3	$Select_1$	$Select_0$
R_x/dx	R_x/a	R_u/R_1	R_u/R_1	$S_3 + S_4$	$S_2 + S_4$
R_x/dx	$3/R_y$	R_u/dx	-/-	S_3	S_2
$3/R_u$	R_1/R_2	R_2/dx	-/-	S_3	S_2

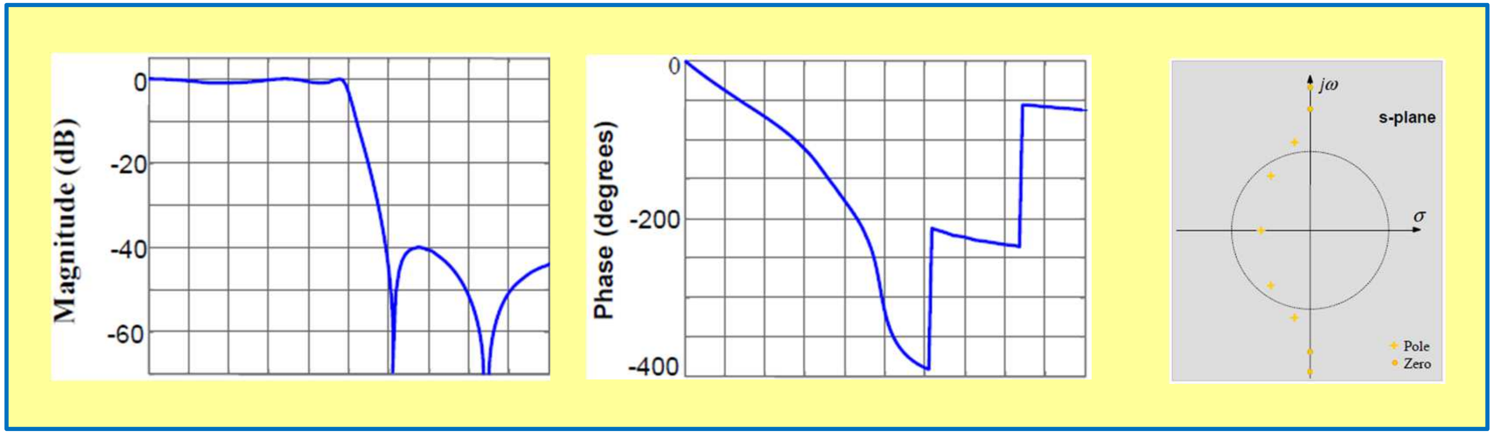
… L'unità di elaborazione



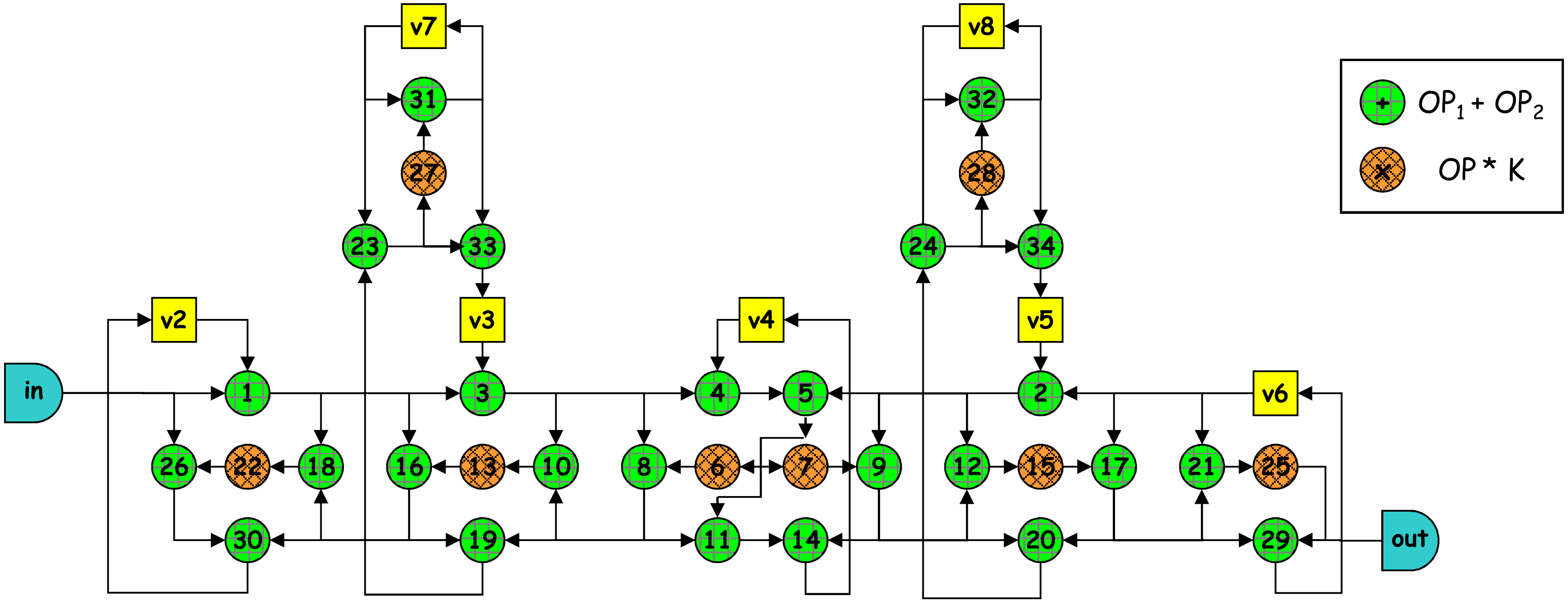
Modello ILP: un altro caso di studio

5th order bilinear elliptic switched-current filter





Elliptic Wave Filter (EWF)



Due tipologie di risorse disponibili (moltiplicatori / addizionatori), con, rispettivamente, tempo di esecuzione 2 / 1 t.u. e costo 5 / 2 unità

Vincolo:
- latenza minima

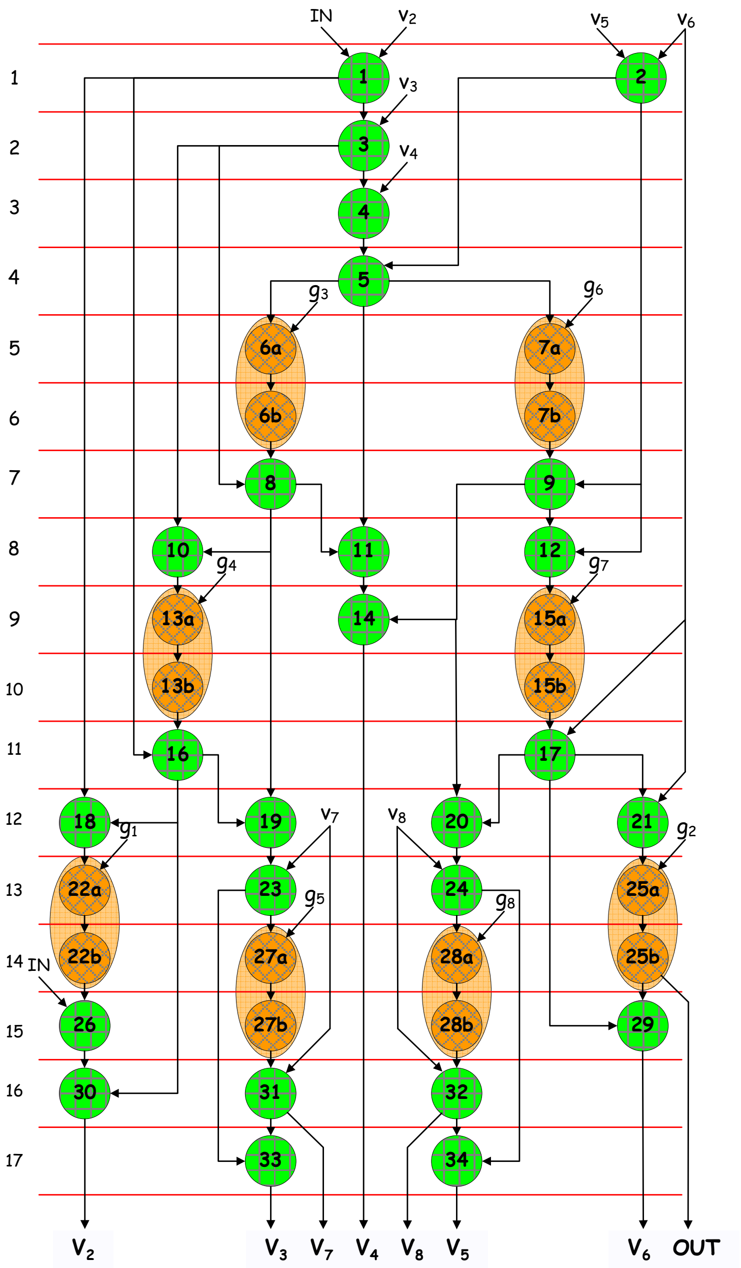
Obiettivo:
- insieme di risorse di costo minimo

Latenza minima → ASAP SSG

Ipotesi:
risorse illimitate

Latenza: 17 t.u.

costo complessivo:
28 unità

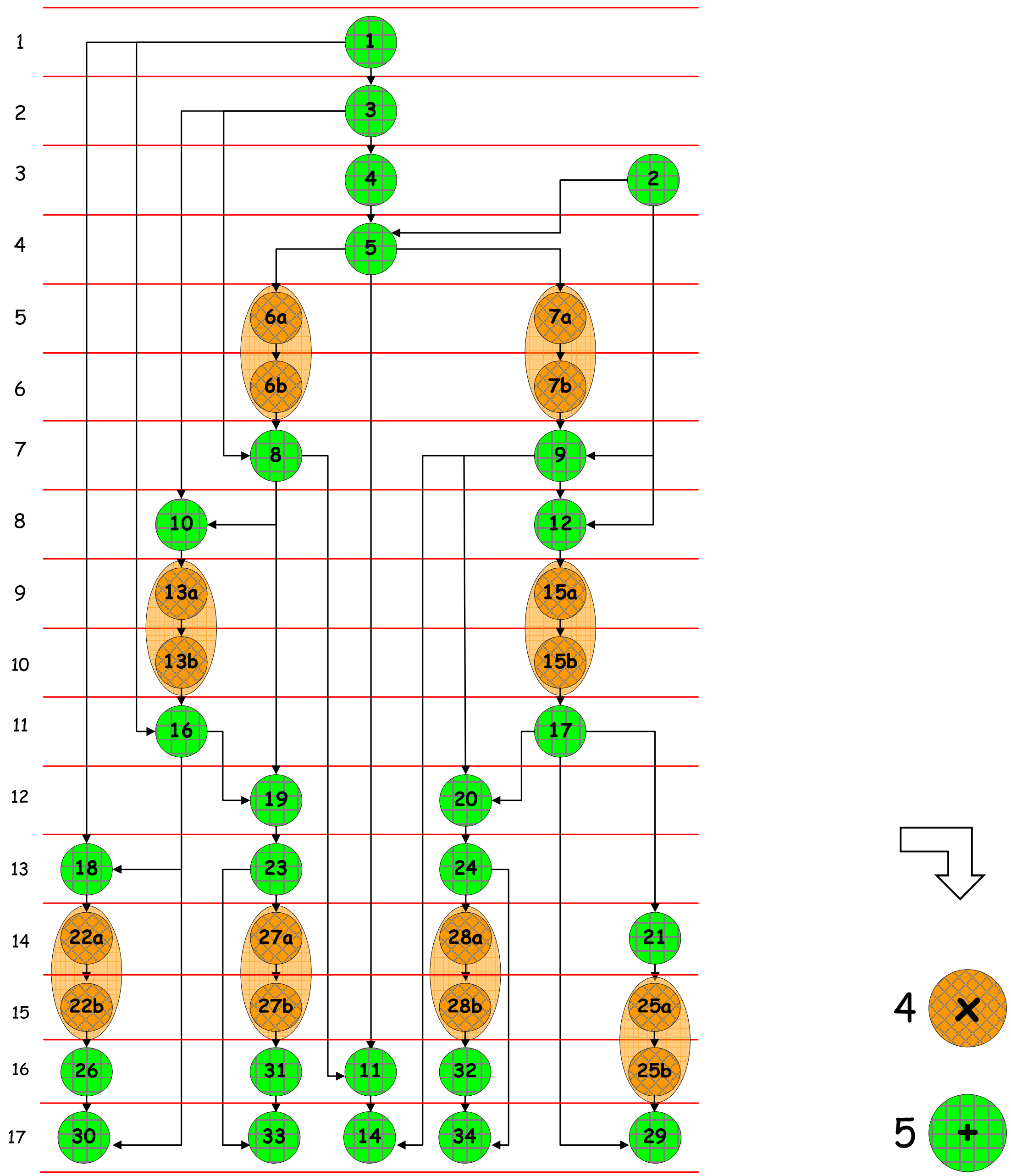


4 +

4 ×

ALAP SSG

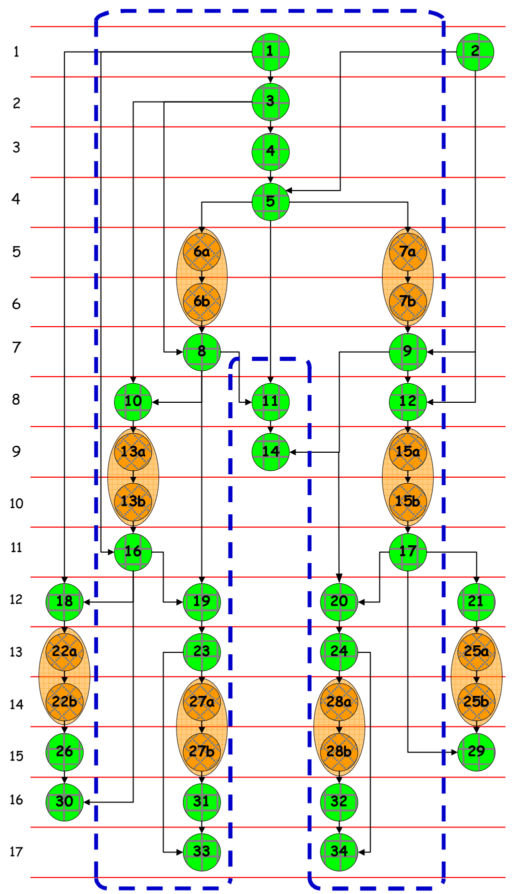
Latenza: 17 t.u.



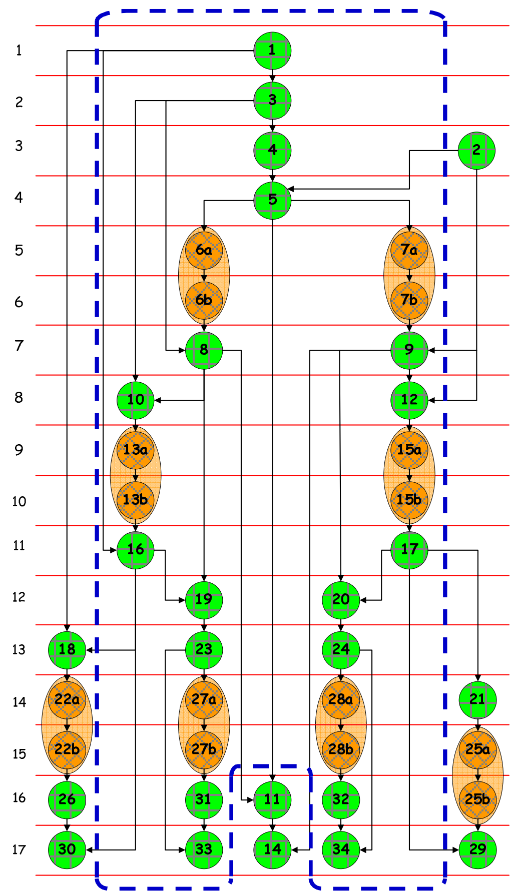
costo complessivo:
30 unità

Modello ILP

ASAP



ALAP



O_i	t_i^S	t_i^L	O_i	t_i^S	t_i^L
O_1	1	1	O_{18}	12	13
O_2	1	3	O_{19}	12	12
O_3	2	2	O_{20}	12	12
O_4	3	3	O_{21}	12	14
O_5	4	4	O_{22}	13	14
O_6	5	5	O_{23}	13	13
O_7	5	5	O_{24}	13	13
O_8	7	7	O_{25}	13	15
O_9	7	7	O_{26}	15	16
O_{10}	8	8	O_{27}	14	14
O_{11}	8	16	O_{28}	14	14
O_{12}	8	8	O_{29}	15	17
O_{13}	9	9	O_{30}	16	17
O_{14}	9	17	O_{31}	16	16
O_{15}	9	9	O_{32}	16	16
O_{16}	11	11	O_{33}	17	17
O_{17}	11	11	O_{34}	17	17

Modello ILP: 1° insieme di vincoli

$$\sum_{t_i^S \le l \le t_i^L} x_{il} = 1 \,, \quad i = 1, 2, \ldots$$

O_i	t_i^S	t_i^L
O_1	1	1
O_2	1	3
O_3	2	2
O_4	3	3
O_5	4	4
O_6	5	5
O_7	5	5
O_8	7	7
O_9	7	7
O_{10}	8	8
O_{11}	8	16
O_{12}	8	8
O_{13}	9	9
O_{14}	9	17
O_{15}	9	9
O_{16}	11	11
O_{17}	11	11
O_{18}	12	13
O_{19}	12	12
O_{20}	12	12
O_{21}	12	14
O_{22}	13	14
O_{23}	13	13
O_{24}	13	13
O_{25}	13	15
O_{26}	15	16
O_{27}	14	14
O_{28}	14	14
O_{29}	15	17
O_{30}	16	17
O_{31}	16	16
O_{32}	16	16
O_{33}	17	17
O_{34}	17	17

$x_{1\,1} = 1$ $\qquad$ $x_{2\,1} + x_{2\,2} + x_{2\,3} = 1$

$x_{3\,2} = 1$ $\qquad$ $x_{4\,3} = 1$

$x_{5\,4} = 1$ $\qquad$ $x_{6\,5} = 1$

$x_{7\,5} = 1$ $\qquad$ $x_{8\,7} = 1$

$x_{9\,7} = 1$ $\qquad$ $x_{10\,8} = 1$

$x_{11\,8} + x_{11\,9} + x_{11\,10} + x_{11\,11} + x_{11\,12} + x_{11\,13} + x_{11\,14} + x_{11\,15} + x_{11\,16} = 1$

$x_{12\,8} = 1$ $\qquad$ $x_{13\,9} = 1$

$x_{14\,9} + x_{14\,10} + x_{14\,11} + x_{14\,12} + x_{14\,13} + x_{14\,14} + x_{14\,15} + x_{14\,16} + x_{14\,17} = 1$

$x_{15\,9} = 1$ $\qquad$ $x_{16\,11} = 1$

$x_{17\,11} = 1$ $\qquad$ $x_{18\,12} + x_{18\,13} = 1$

$x_{19\,12} = 1$ $\qquad$ $x_{20\,12} = 1$

$x_{21\,12} + x_{21\,13} + x_{21\,14} = 1$ $\qquad$ $x_{22\,13} + x_{22\,14} = 1$

$x_{23\,13} = 1$ $\qquad$ $x_{24\,13} = 1$

$x_{25\,13} + x_{25\,14} + x_{25\,15} = 1$ $\qquad$ $x_{26\,15} + x_{26\,16} = 1$

$x_{27\,14} = 1$ $\qquad$ $x_{28\,14} = 1$

$x_{29\,15} + x_{29\,16} + x_{29\,17} = 1$ $\qquad$ $x_{30\,16} + x_{30\,17} = 1$

$x_{31\,16} = 1$ $\qquad$ $x_{32\,16} = 1$

$x_{33\,17} = 1$ $\qquad$ $x_{34\,17} = 1$

Modello ILP: 2° insieme di vincoli

$$\sum_{t_i^S \leq l \leq t_i^L} l\, x_{il} \geq \sum_{t_j^S \leq l \leq t_j^L} l\, x_{jl} + d_j, \quad i, j : (v_j, v_i) \in E$$

$$9\, x_{14\,9} + 10\, x_{14\,10} + 11\, x_{14\,11} + 12\, x_{14\,12} + 13\, x_{14\,13} + 14\, x_{14\,14} + 15\, x_{14\,15} + 16\, x_{14\,16} + 17\, x_{14\,17}$$
$$- 8\, x_{11\,8} - 9\, x_{11\,9} - 10\, x_{11\,10} - 11\, x_{11\,11} - 12\, x_{11\,12} - 13\, x_{11\,13} - 14\, x_{11\,14} - 15\, x_{11\,15} - 16\, x_{11\,16} - 1 \geq 0 \qquad O_{11} \prec O_{14}$$

$$13\, x_{22\,13} + 14\, x_{22\,14} - 12\, x_{18\,12} - 13\, x_{18\,13} - 1 \geq 0 \qquad O_{18} \prec O_{22}$$

$$13\, x_{25\,13} + 14\, x_{25\,14} + 15\, x_{25\,15} - 12\, x_{21\,12} - 13\, x_{21\,13} - 14\, x_{21\,14} - 1 \geq 0 \qquad O_{21} \prec O_{25}$$

$$15\, x_{26\,15} + 16\, x_{26\,16} - 13\, x_{22\,13} - 14\, x_{22\,14} - 2 \geq 0 \qquad O_{22} \prec O_{26}$$

$$15\, x_{29\,15} + 16\, x_{29\,16} + 17\, x_{29\,17} - 13\, x_{25\,13} - 14\, x_{25\,14} - 15\, x_{25\,15} - 2 \geq 0 \qquad O_{25} \prec O_{29}$$

$$16\, x_{30\,16} + 17\, x_{30\,17} - 15\, x_{26\,15} - 16\, x_{26\,16} - 1 \geq 0 \qquad O_{26} \prec O_{30}$$

(solo i vincoli aggiuntivi che coinvolgono operazioni con mobilità non nulla)

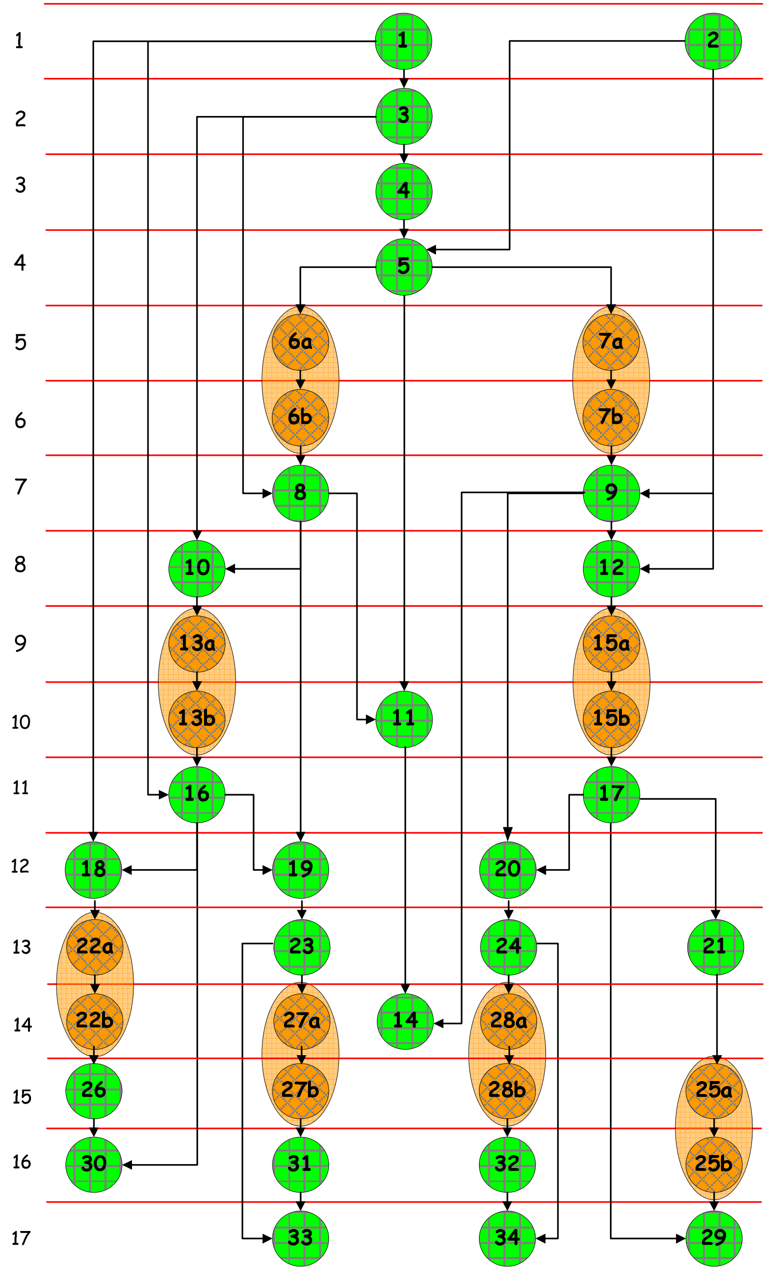
Modello ILP: 3° insieme di vincoli

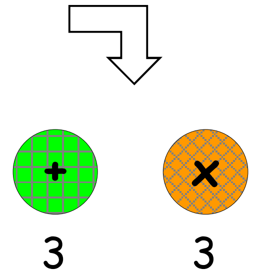
$$\sum_{i\,:\,R(v_i)=k}\ \sum_{l-d_i+1\le m\le l} x_{im} \le a_k,\quad k = 1, 2, \dots, N_R,\ l = 1, 2, \dots, \lambda$$

1	$x_{1\,1} + x_{2\,1} - a_1 \le 0$	
2	$x_{2\,2} + x_{3\,2} - a_1 \le 0$	
3	$x_{2\,3} + x_{4\,3} - a_1 \le 0$	
4	$x_{5\,4} - a_1 \le 0$	
5		$x_{6\,5} + x_{7\,5} - a_2 \le 0$
6		$x_{6\,5} + x_{7\,5} - a_2 \le 0$
7	$x_{8\,7} + x_{9\,7} - a_1 \le 0$	
8	$x_{10\,8} + x_{11\,8} + x_{12\,8} - a_1 \le 0$	
9	$x_{11\,9} + x_{14\,9} - a_1 \le 0$	$x_{13\,9} + x_{15\,9} - a_2 \le 0$
10	$x_{11\,10} + x_{14\,10} - a_1 \le 0$	$x_{13\,9} + x_{15\,9} - a_2 \le 0$
11	$x_{11\,11} + x_{14\,11} + x_{16\,11} + x_{17\,11} - a_1 \le 0$	
12	$x_{11\,12} + x_{14\,12} + x_{18\,12} + x_{19\,12} + x_{20\,12} + x_{21\,12} - a_1 \le 0$	
13	$x_{11\,13} + x_{14\,13} + x_{18\,13} + x_{21\,13} + x_{23\,13} + x_{24\,13} - a_1 \le 0$	$x_{22\,13} + x_{25\,13} - a_2 \le 0$
14	$x_{11\,14} + x_{14\,14} + x_{21\,14} - a_1 \le 0$	$x_{22\,13} + x_{22\,14} + x_{25\,13} + x_{25\,14} + x_{27\,14} + x_{28\,14} - a_2 \le 0$
15	$x_{11\,15} + x_{14\,15} + x_{26\,15} + x_{29\,15} - a_1 \le 0$	$x_{22\,14} + x_{25\,14} + x_{25\,15} + x_{27\,14} + x_{28\,14} - a_2 \le 0$
16	$x_{11\,16} + x_{14\,16} + x_{26\,16} + x_{29\,16} + x_{30\,16} + x_{31\,16} + x_{32\,16} - a_1 \le 0$	$x_{25\,15} - a_2 \le 0$
17	$x_{14\,17} + x_{29\,17} + x_{30\,17} + x_{33\,17} + x_{34\,17} - a_1 \le 0$	

funzione obiettivo: min (c_1 + c_2) ovvero min ($2\,a_1 + 5\,a_2$)

Modello ILP: SSG

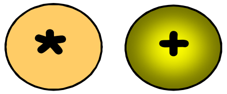


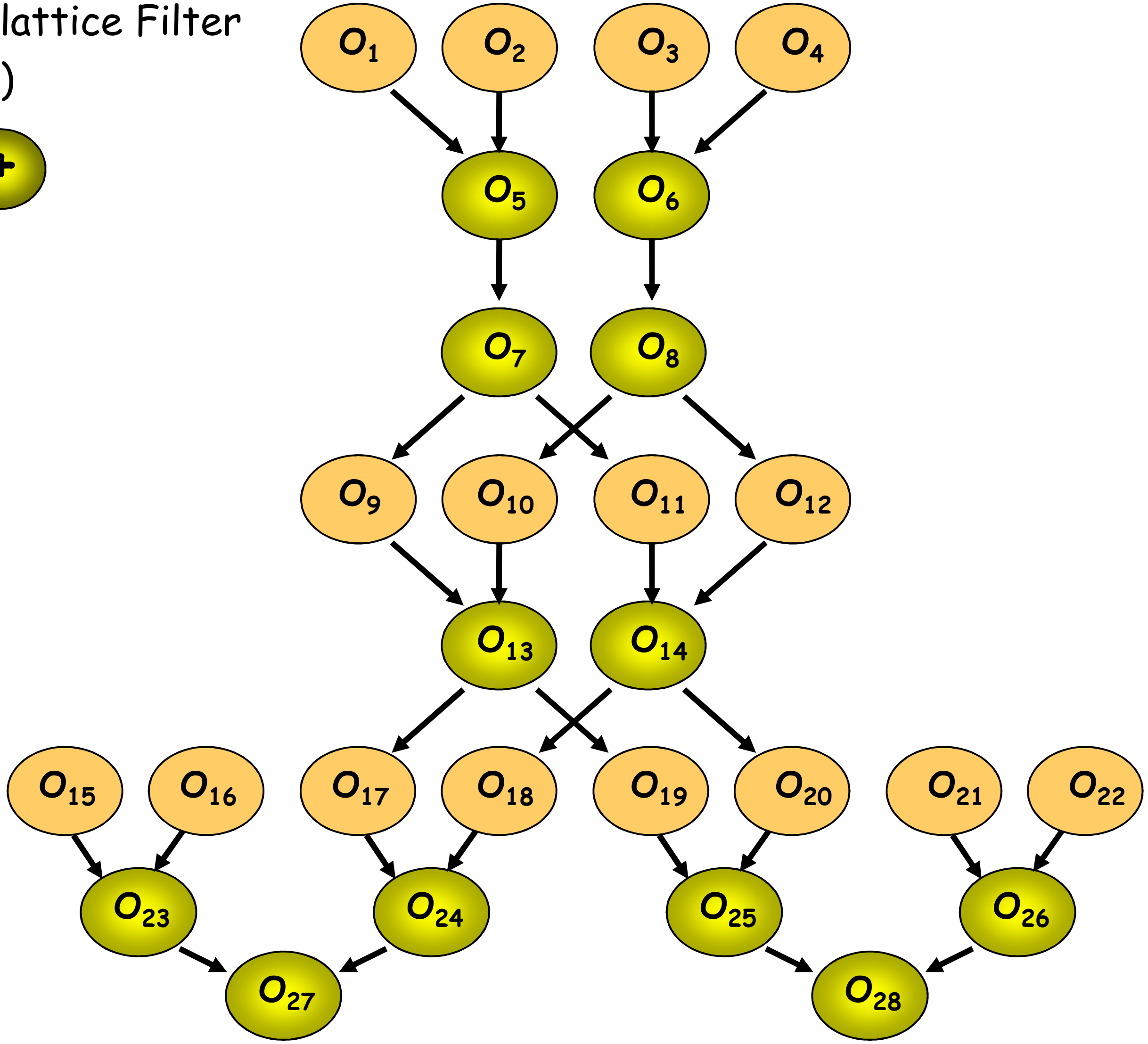


costo complessivo:
21 unità

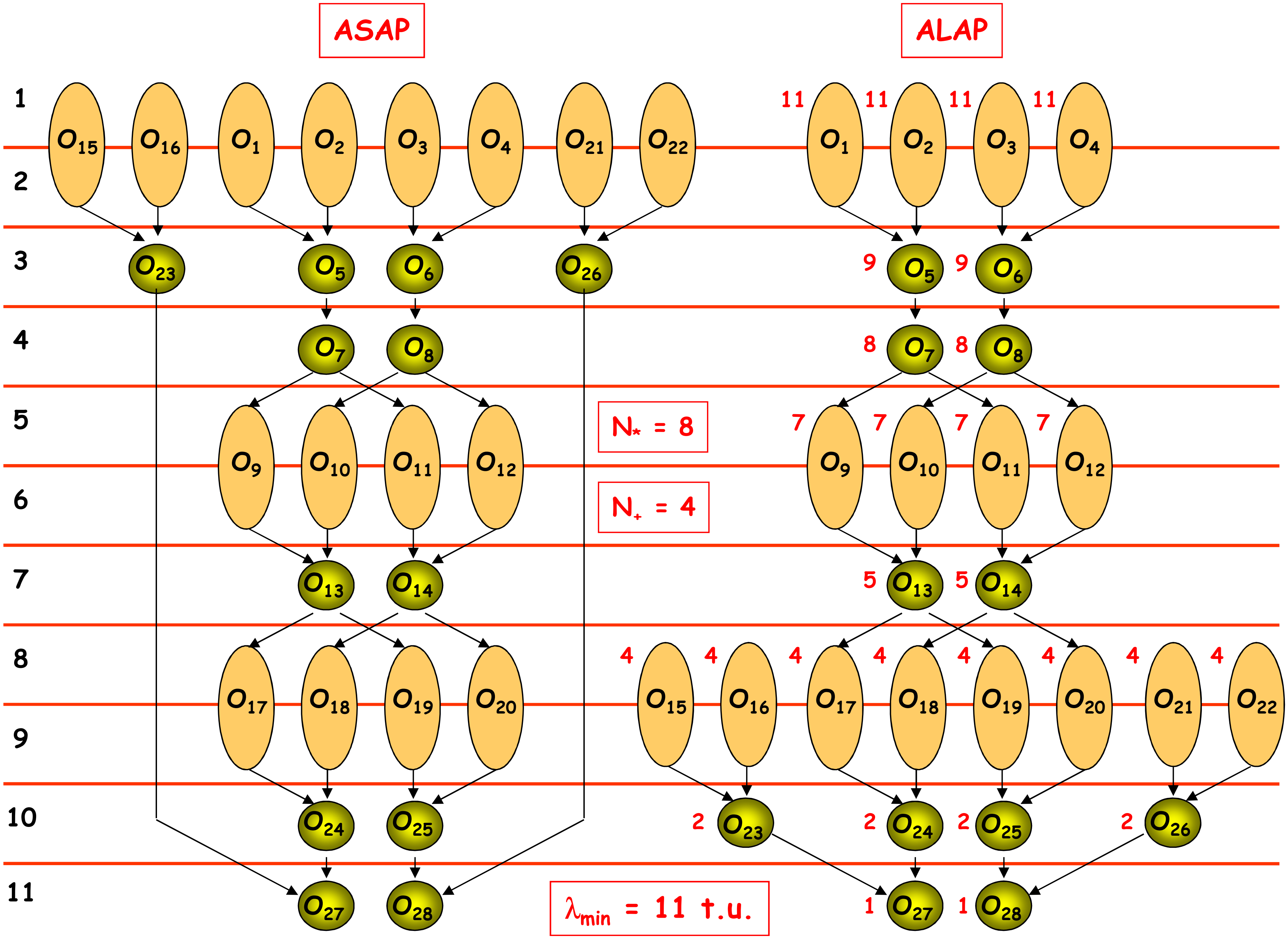
Algoritmi “resource-constrained”: ulteriori casi di studio

Auto-Regressive lattice Filter
(ARF)

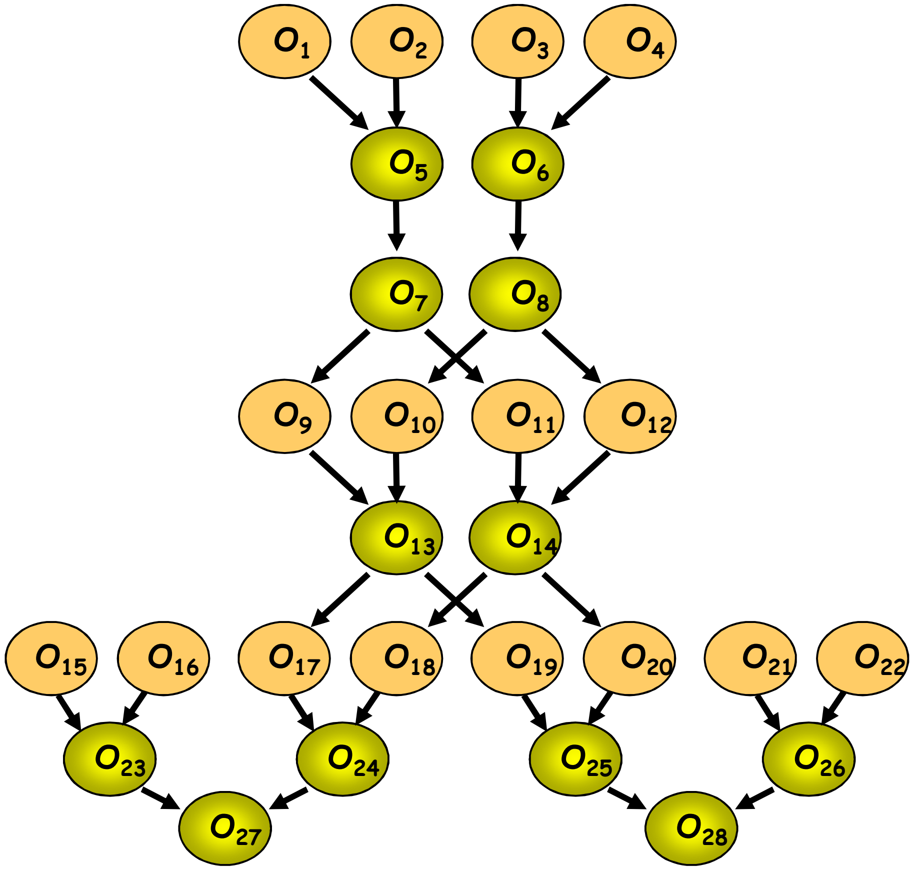




Risorse disponibili: 2 moltiplicatori ($d_* = 2$ t.u.) e 1 addizionatore ($d_+ = 1$ t.u.)


ASAP
ALAP
N* = 8
N+ = 4
λmin = 11 t.u.
1
2
3
4
5
6
7
8
9
10
11
O15 O16 O1 O2 O3 O4 O21 O22
O23 O5 O6 O26
O7 O8
O9 O10 O11 O12
O13 O14
O17 O18 O19 O20
O24 O25
O27 O28
11 O1 11 O2 11 O3 11 O4
9 O5 9 O6
8 O7 8 O8
7 O9 7 O10 7 O11 7 O12
5 O13 5 O14
4 O15 4 O16 4 O17 4 O18 4 O19 4 O20 4 O21 4 O22
2 O23 2 O24 2 O25 2 O26
1 O27 1 O28

Algoritmo SLS



	+	*		
1	-	$O_1,O_2,O_3,O_4,O_{15},O_{16},O_{21},O_{22}$	-	O_1,O_2
2	-	$O_3,O_4,O_{15},O_{16},O_{21},O_{22}$	O_1,O_2	-
3	O_5	$O_3,O_4,O_{15},O_{16},O_{21},O_{22}$	-	O_3,O_4,O_5
4	O_7	$O_{15},O_{16},O_{21},O_{22}$	O_3,O_4	O_7
5	O_6	$O_9,O_{11},O_{15},O_{16},O_{21},O_{22}$	-	O_6,O_9,O_{11}
6	O_8	$O_{15},O_{16},O_{21},O_{22}$	O_9,O_{11}	O_8
7	-	$O_{10},O_{12},O_{15},O_{16},O_{21},O_{22}$	-	O_{10},O_{12}
8	-	$O_{15},O_{16},O_{21},O_{22}$	O_{10},O_{12}	-
9	O_{13},O_{14}	$O_{15},O_{16},O_{21},O_{22}$	-	O_{13},O_{15},O_{16}
10	O_{14}	$O_{17},O_{19},O_{21},O_{22}$	O_{15},O_{16}	O_{14}
11	O_{23}	$O_{17},O_{18},O_{19},O_{20},O_{21},O_{22}$	-	O_{17},O_{18},O_{23}
12	-	$O_{19},O_{20},O_{21},O_{22}$	O_{17},O_{18}	-
13	O_{24}	$O_{19},O_{20},O_{21},O_{22}$	-	O_{19},O_{20},O_{24}
14	O_{27}	O_{21},O_{22}	O_{19},O_{20}	O_{27}
15	O_{25}	O_{21},O_{22}	-	O_{21},O_{22},O_{25}
16	-	-	O_{21},O_{22}	-
17	O_{26}	-	-	O_{26}
18	O_{28}	-	-	O_{28}

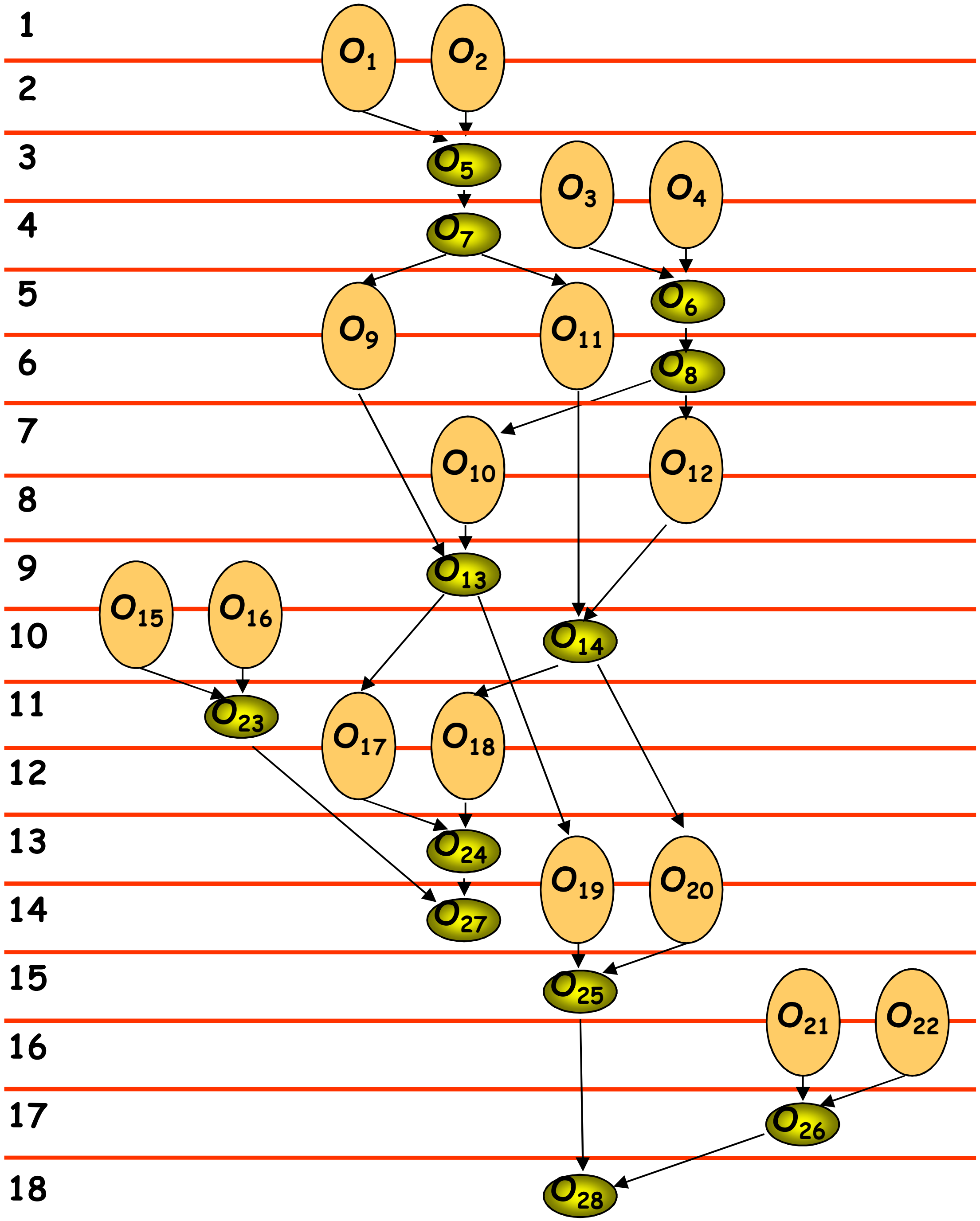
Operazioni

candidate

non terminate

schedulate

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}	O_{21}	O_{22}	O_{23}	O_{24}	O_{25}	O_{26}	O_{27}	O_{28}
p_i	11	11	11	11	9	9	8	8	7	7	7	7	5	5	4	4	4	4	4	4	4	4	2	2	2	2	1	1

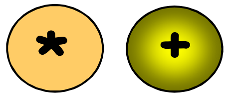


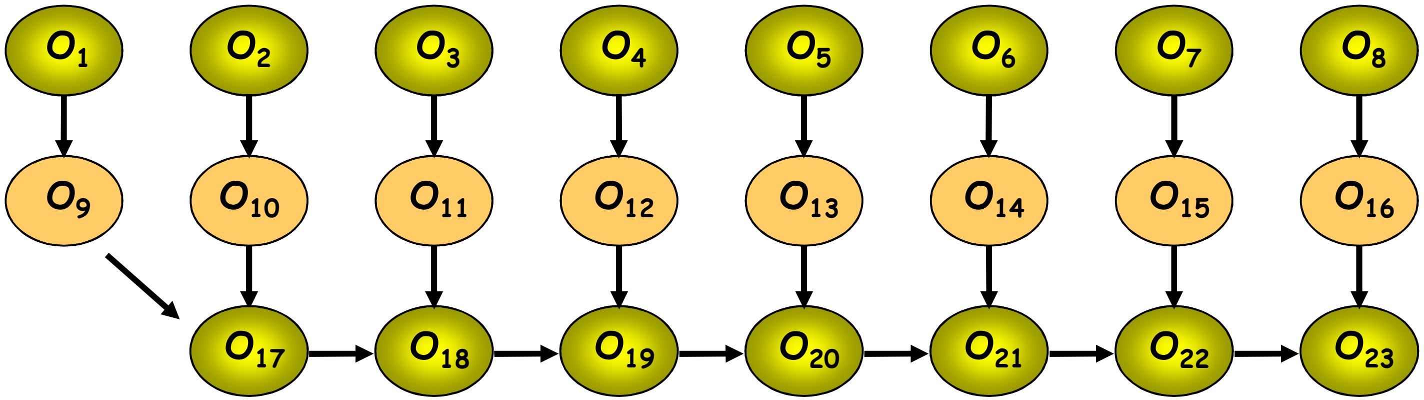
SLS SSG

1	O_1,O_2
2	-
3	O_3,O_4,O_5
4	O_7
5	O_6,O_9,O_{11}
6	O_8
7	O_{10},O_{12}
8	-
9	O_{13},O_{15},O_{16}
10	O_{14}
11	O_{17},O_{18},O_{23}
12	-
13	O_{19},O_{20},O_{24}
14	O_{27}
15	O_{21},O_{22},O_{25}
16	-
17	O_{26}
18	O_{28}

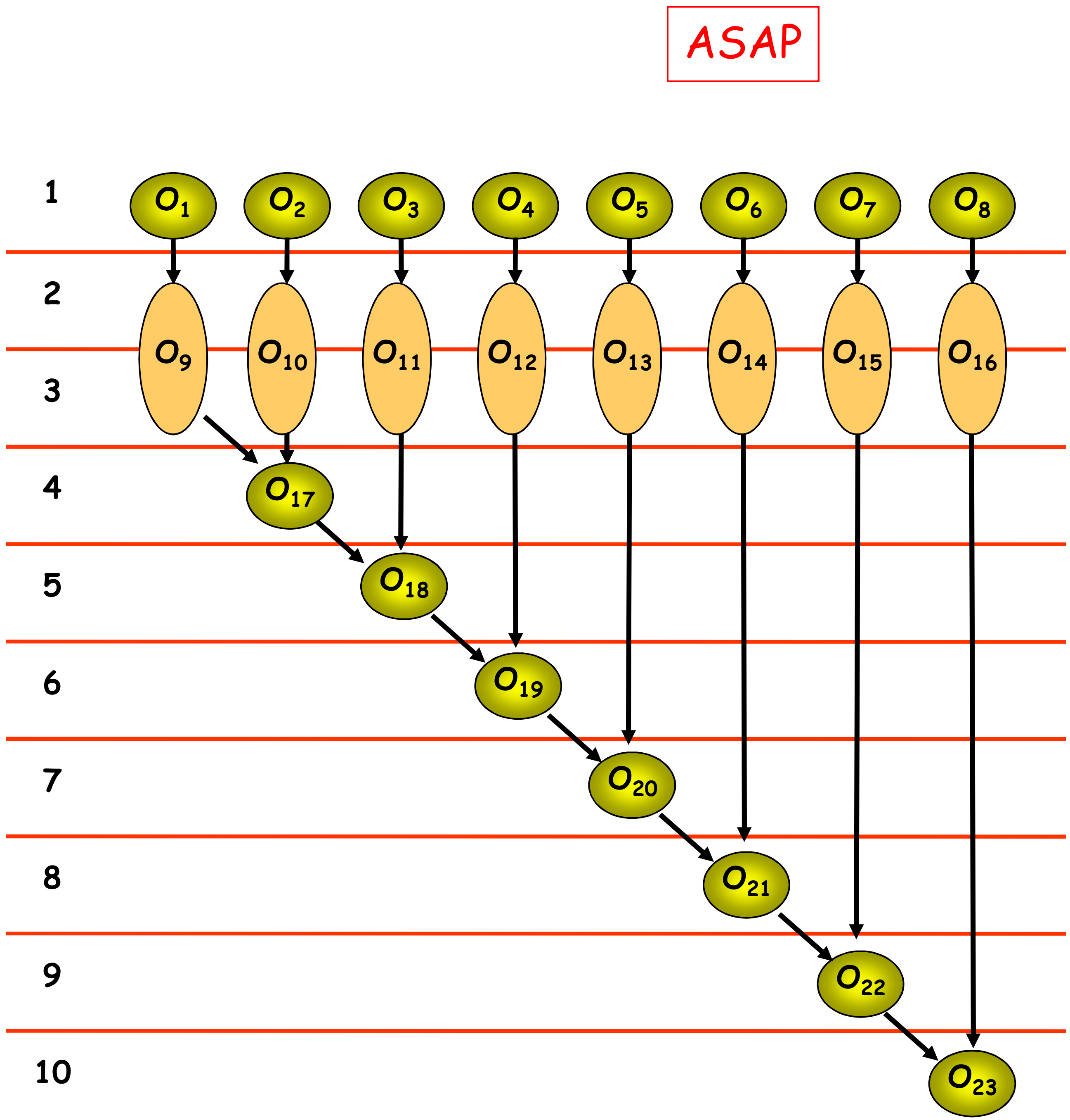
Latenza: 18 t.u.

Finite Impulse Response (FIR) filter





Risorse disponibili:
2 moltiplicatori ($d_* = 2$ t.u.)
2 addizionatori ($d_+ = 1$ t.u.)



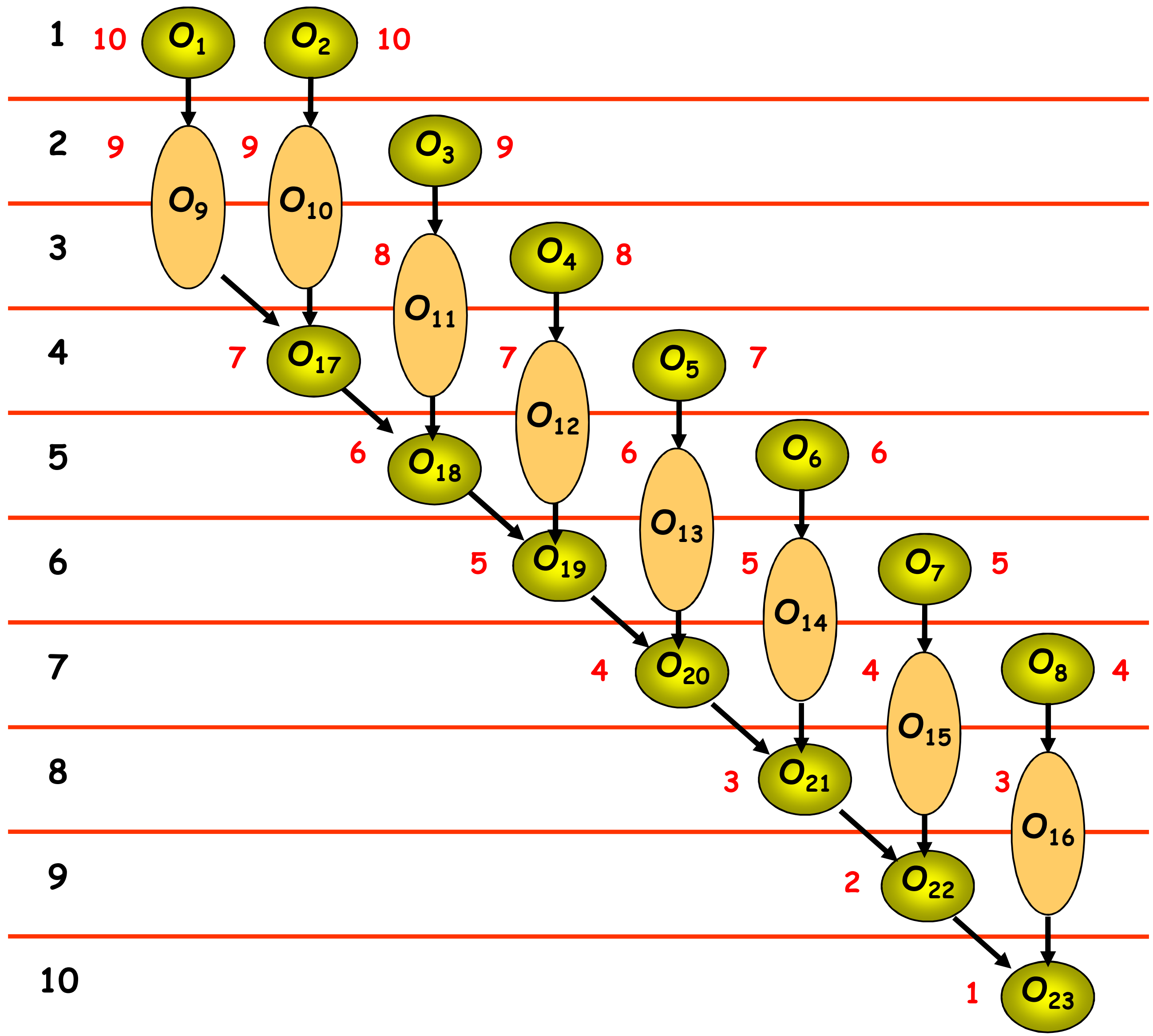
λ_{min} = 10 t.u.

N_* = 8

N_+ = 8

ALAP

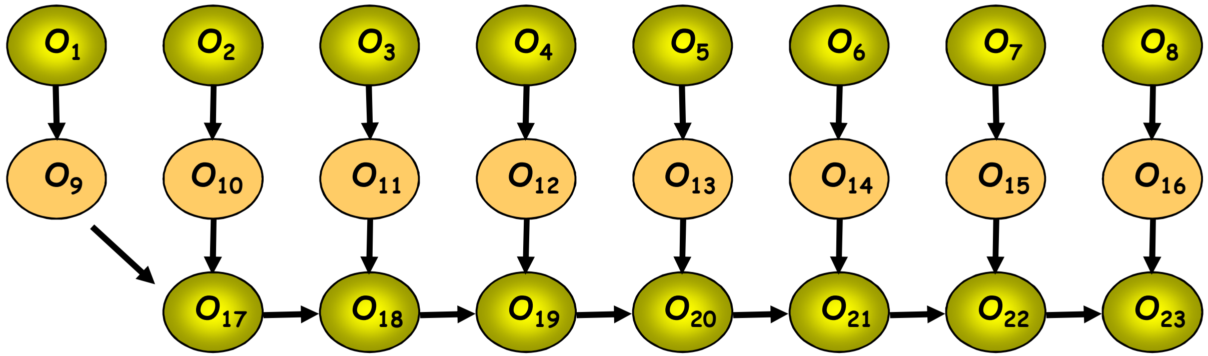
λ : 10 t.u.



$N_* = 3$

$N_+ = 2$

Algoritmo SLS



	+	*		
1	$O_1,O_2,O_3,O_4,O_5,O_6,O_7,O_8$	-	-	O_1,O_2
2	O_3,O_4,O_5,O_6,O_7,O_8	O_9,O_{10}	-	O_3,O_4,O_9,O_{10}
3	O_5,O_6,O_7,O_8	O_{11},O_{12}	O_9,O_{10}	O_5,O_6
4	O_7,O_8,O_{17}	$O_{11},O_{12},O_{13},O_{14}$	-	O_7,O_{11},O_{12},O_{17}
5	O_8	O_{13},O_{14},O_{15}	O_{11},O_{12}	O_8
6	O_{18}	$O_{13},O_{14},O_{15},O_{16}$	-	O_{13},O_{14},O_{18}
7	O_{19}	O_{15},O_{16}	O_{13},O_{14}	O_{19}
8	O_{20}	O_{15},O_{16}	-	O_{15},O_{16},O_{20}
9	O_{21}	-	O_{15},O_{16}	O_{21}
10	O_{22}	-	-	O_{22}
11	O_{23}	-	-	O_{23}

Operazioni

candidate

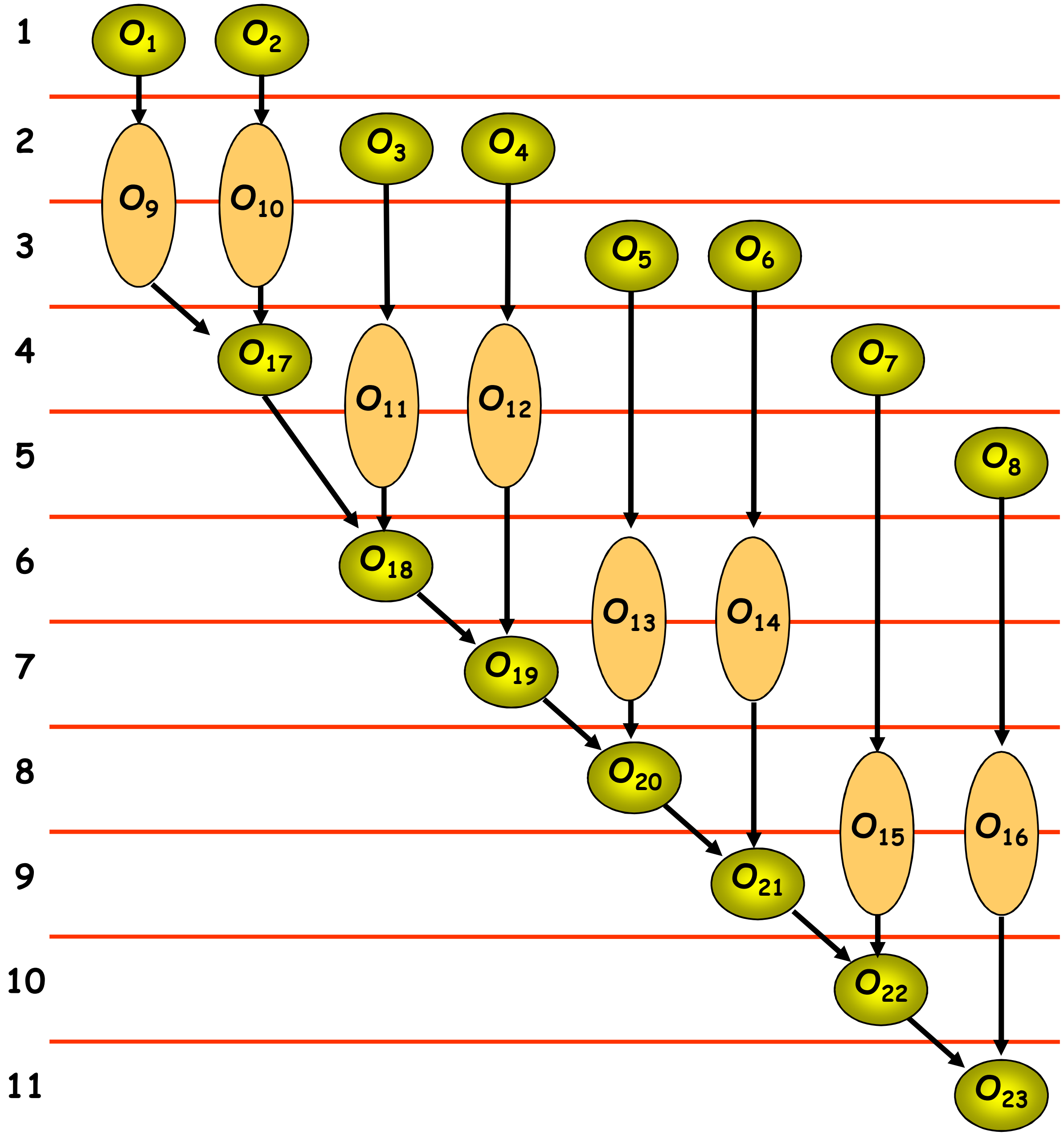
non terminate

schedulate

	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}	O_{18}	O_{19}	O_{20}	O_{21}	O_{22}	O_{23}
p_i	10	10	9	8	7	6	5	4	9	9	8	7	6	5	4	3	7	6	5	4	3	2	1

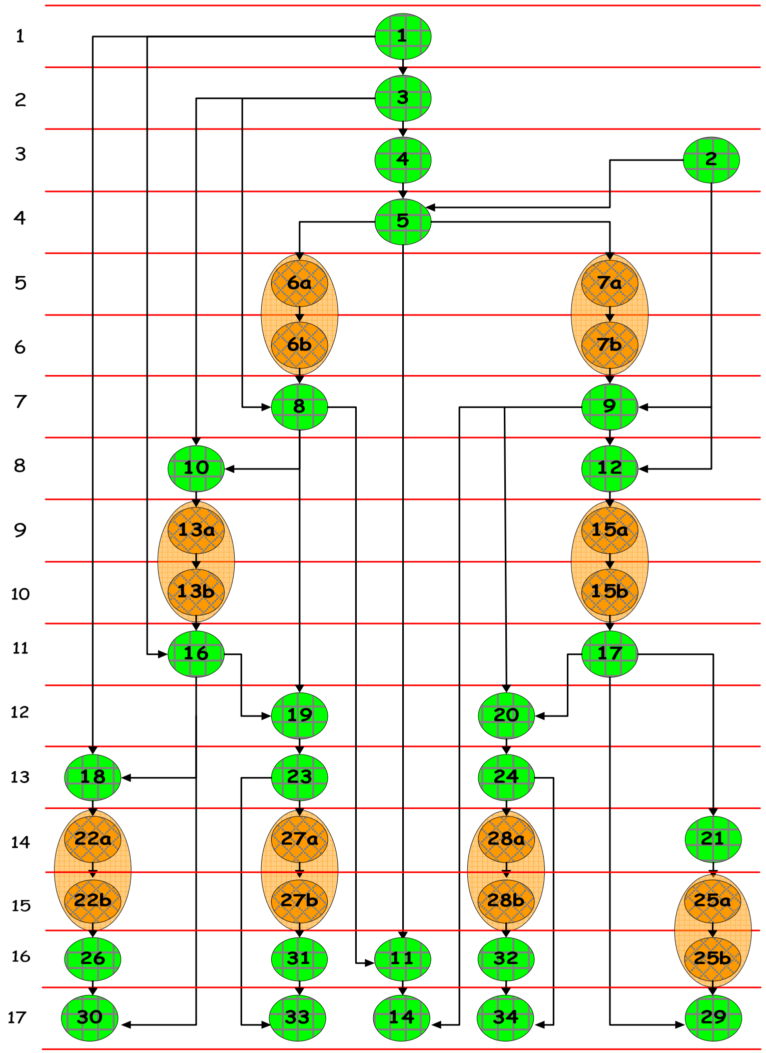
SLS SSG

1	O_1,O_2
2	O_3,O_4,O_9,O_{10}
3	O_5,O_6
4	O_7,O_{11},O_{12},O_{17}
5	O_8
6	O_{13},O_{14},O_{18}
7	O_{19}
8	O_{15},O_{16},O_{20}
9	O_{21}
10	O_{22}
11	O_{23}



Latenza: 11 t.u.

ALAP SSG (EWF_{SG}, $\lambda=\lambda_{min}=17$)



Algoritmo SLS

Risorse disponibili:

2 (+) (d_+ = 1 t.u.)

2 (×) (d_* = 2 t.u.)

	Op. candidate	Op. schedulate
1	O_1, O_2	O_1, O_2
2	O_3	O_3
3	O_4	O_4
4	O_5	O_5
5	O_6, O_7	O_6, O_7
6	-	-
7	O_8, O_9	O_8, O_9
8	O_{10}, O_{12}, O_{11}	O_{10}, O_{12}
9	O_{13}, O_{15}, O_{11}	O_{13}, O_{15}, O_{11}
10	O_{14}	O_{14}
11	O_{16}, O_{17}	O_{16}, O_{17}
12	$O_{19}, O_{20}, O_{18}, O_{21}$	O_{19}, O_{20}
13	$O_{18}, O_{23}, O_{24}, O_{21}$	O_{18}, O_{23}
14	$O_{24}, O_{21}, O_{22}, O_{27}$	$O_{24}, O_{21}, O_{22}, O_{27}$
15	O_{28}, O_{25}	-
16	$O_{28}, O_{25}, O_{26}, O_{31}$	$O_{28}, O_{25}, O_{26}, O_{31}$
17	O_{30}, O_{33}	O_{30}, O_{33}
18	O_{32}, O_{29}	O_{32}, O_{29}
19	O_{34}	O_{34}

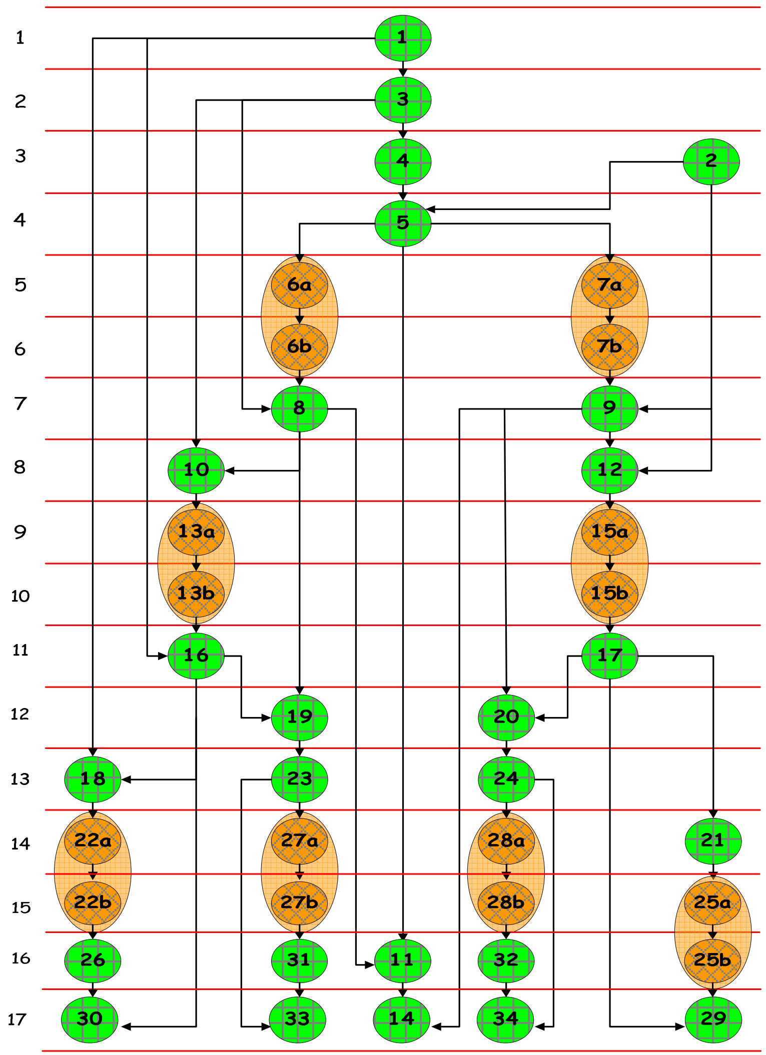
	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_8	O_9	O_{10}	O_{11}	O_{12}	O_{13}	O_{14}	O_{15}	O_{16}	O_{17}
p_i	17	15	16	15	14	13	13	11	11	10	2	10	9	1	9	7	7

	O_{18}	O_{19}	O_{20}	O_{21}	O_{22}	O_{23}	O_{24}	O_{25}	O_{26}	O_{27}	O_{28}	O_{29}	O_{30}	O_{31}	O_{32}	O_{33}	O_{34}
p_i	5	6	6	4	4	5	5	3	2	4	4	1	1	2	2	1	1

Latenza: 19 t.u.

Algoritmo FDLS

ALAP SSG (EWF$_{SG}$, $\lambda=\lambda_{min}=17$)



Risorse disponibili:

2 (+) (d_+ = 1 t.u.)

2 (×) (d_* = 2 t.u.)

"time constraint" iniziale:
λ = 18

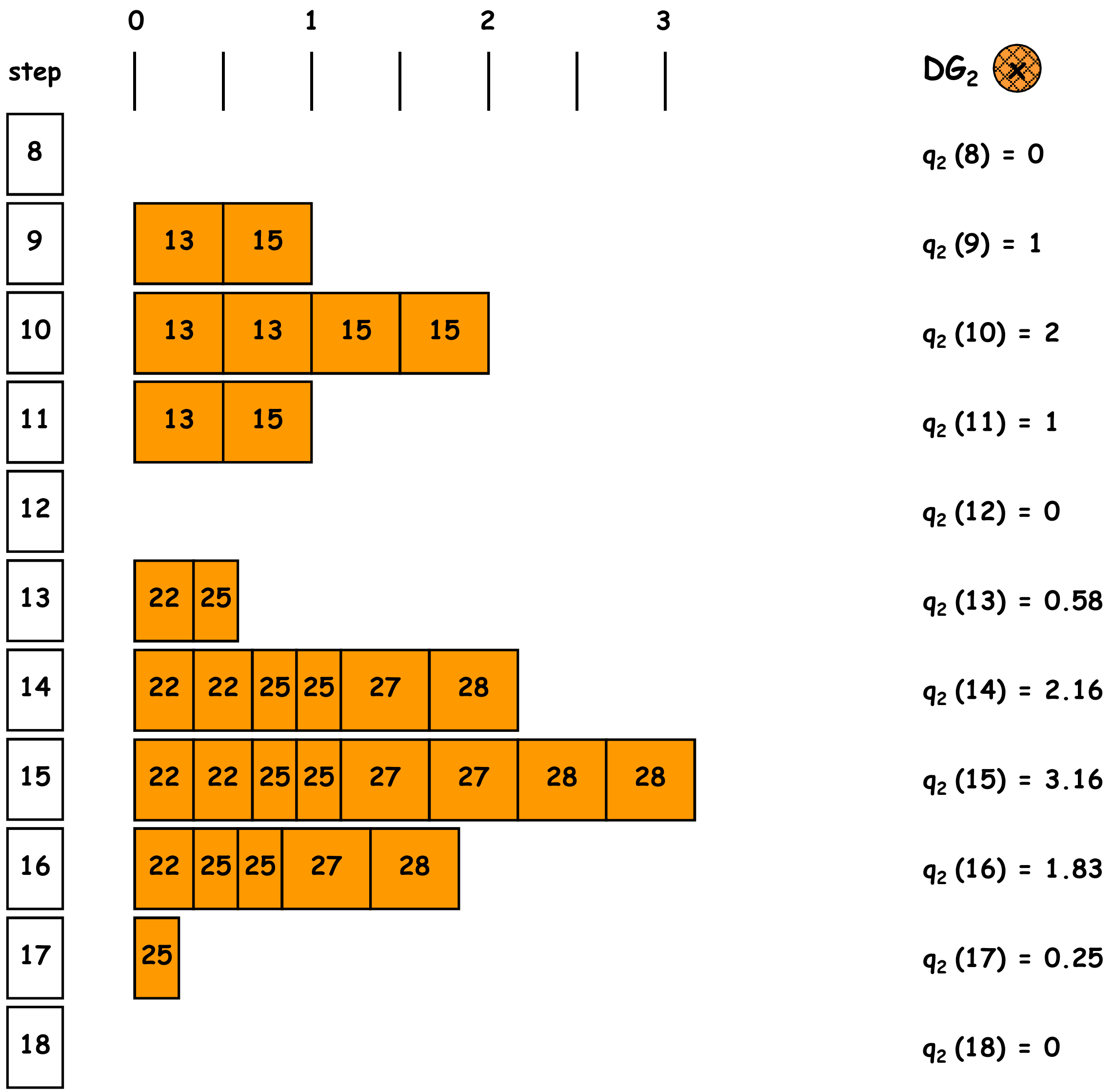
	Op. candidate	Op. schedulate
1	O_1,O_2	O_1,O_2
2	O_3	O_3
3	O_4	O_4
4	O_5	O_5
5	O_6,O_7	O_6,O_7
6	-	-
7	O_8,O_9	O_8,O_9
8	O_{10},O_{12},O_{11}	?,?

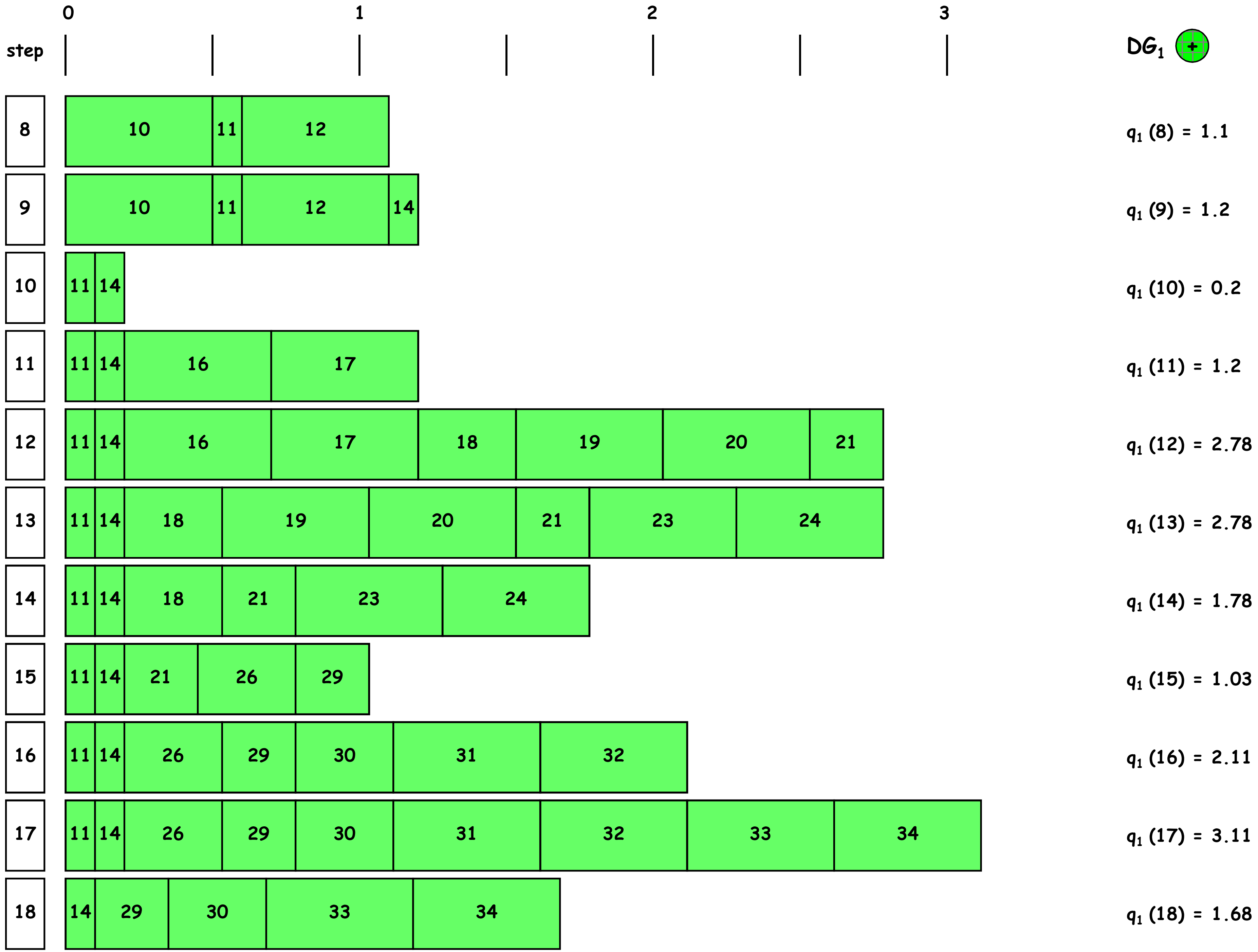
Quale operazione conviene differire ?

O_i	10	11	12	13	14	15	16	17
TF_i	[8,9]	[8,17]	[8,9]	[9,10]	[9,18]	[9,10]	[11,12]	[11,12]

O_i	18	19	20	21	22	23	24	25
TF_i	[12,14]	[12,13]	[12,13]	[12,15]	[13,15]	[13,14]	[13,14]	[13,16]

O_i	26	27	28	29	30	31	32	33	34
TF_i	[15,17]	[14,15]	[14,15]	[15,18]	[16,18]	[16,17]	[16,17]	[17,18]	[17,18]





0
1
2
3
step
DG1
8
10 11 12
9
10 11 12 14
10
11 14
11
11 14 16 17
12
11 14 16 17 18 19 20 21
13
11 14 18 19 20 21 23 24
14
11 14 18 21 23 24
15
11 14 21 26 29
16
11 14 26 29 30 31 32
17
11 14 26 29 30 31 32 33 34
18
14 29 30 33 34
q1 (8) = 1.1
q1 (9) = 1.2
q1 (10) = 0.2
q1 (11) = 1.2
q1 (12) = 2.78
q1 (13) = 2.78
q1 (14) = 1.78
q1 (15) = 1.03
q1 (16) = 2.11
q1 (17) = 3.11
q1 (18) = 1.68

La “Deferral Force” DF_{10} corrispondente al differimento di O_{10},
con conseguente ridimensionamento del TF anche di O_{13}, O_{16}, O_{18}, O_{19}, O_{22}, O_{23}, O_{26}, O_{27}, O_{30}, O_{31} e O_{33}, vale:

DF_{10}	$= 1.2 - (1.1 + 1.2) / 2$	O_{10}
	$+ 2 - (1 + 2) / 2 + 1 - (2 + 1) / 2$	O_{13}
	$+ 2.78 - (1.2 + 2.78) / 2$	O_{16}
	$+ (2.78 + 1.78) / 2 - (2.78 + 2.78 + 1.78) / 3$	O_{18}
	$+ 2.78 - (2.78 + 2.78) / 2$	O_{19}
	$+ (2.16 + 3.16) / 2 - (0.58 + 2.16 + 3.16) / 3 + (3.16 + 1.83) / 2 - (2.16 + 3.16 + 1.83) / 3$	O_{22}
	$+ 1.78 - (2.78 + 1.78) / 2$	O_{23}
	$+ (2.11 + 3.11) / 2 - (1.03 + 2.11 + 3.11) / 3$	O_{26}
	$+ 3.16 - (2.16 + 3.16) / 2 + 1.83 - (3.16 + 1.83) / 2$	O_{27}
	$+ (3.11 + 1.68) / 2 - (2.11 + 3.11 + 1.68) / 3$	O_{30}
	$+ 3.11 - (2.11 + 3.11) / 2$	O_{31}
	$+ 1.68 - (3.11 + 1.68) / 2$	O_{33}
	$= 0.05 + 0 + 0.79 - 0.17 + 0 + 0.8 - 0.5 + 0.53 - 0.17 + 0.1 + 0.5 - 0.72 = 1.21$	

s	q_1 (s)	q_2 (s)
8	1.1	0
9	1.2	1
10	0.2	2
11	1.2	1
12	2.78	0
13	2.78	0.58
14	1.78	2.16
15	1.03	3.16
16	2.11	1.83
17	3.11	0.25
18	1.68	0

O_i	10	13	16	18	19	22	23	26	27	30	31	33	
TF_i	[8,9]	[9,10]	[11,12]	[12,14]	[12,13]	[13,15]	[13,14]	[15,17]	[14,15]	[16,18]	[16,17]	[17,18]	
TF_i'	[9]	[10]	[12]	[13,14]	[13]	[14,15]	[14]	[16,17]	[15]	[17,18]	[17]	[18]	
DF_i	0.05	0	0.79	- 0.17	0	0.8	- 0.5	0.53	- 0.17	0.1	0.5	- 0.72	1.21

DG_1 DG_2

La “Deferral Force” DF_{11} corrispondente al differimento di O_{11},
con conseguente ridimensionamento del TF anche di O_{14}, vale:

DF_{11}	$= (1.2 + 0.2 + 1.2 + 2.78 + 2.78 + 1.78 + 1.03 + 2.11 + 3.11) / 9$	O_{11}
	$- (1.1 + 1.2 + 0.2 + 1.2 + 2.78 + 2.78 + 1.78 + 1.03 + 2.11 + 3.11) / 10$	
	$+ (0.2 + 1.2 + 2.78 + 2.78 + 1.78 + 1.03 + 2.11 + 3.11 + 1.68) / 9$	O_{14}
	$- (1.2 + 0.2 + 1.2 + 2.78 + 2.78 + 1.78 + 1.03 + 2.11 + 3.11 + 1.68) / 10$	
	$= 0.07 + 0.07 = 0.14$	

O_i	11	14	
TF_i	[8,17]	[9,18]	
TF_i'	[9-17]	[10,18]	
DF_i	0.07	0.07	0.14

La "Deferral Force" DF_{12} corrispondente al differimento di O_{12},
con conseguente ridimensionamento del TF anche di O_{15}, O_{17}, O_{20}, O_{21}, O_{24}, O_{25}, O_{28}, O_{29}, O_{32} e O_{34}, vale:

DF_{12}	$= 1.2 - (1.1 + 1.2) / 2$	O_{12}
	$+ 2 - (1 + 2) / 2 + 1 - (2 + 1) / 2$	O_{15}
	$+ 2.78 - (1.2 + 2.78) / 2$	O_{17}
	$+ 2.78 - (2.78 + 2.78) / 2$	O_{20}
	$+ (2.78 + 1.78 + 1.03) / 3 - (2.78 + 2.78 + 1.78 + 1.03) / 4$	O_{21}
	$+ 1.78 - (2.78 + 1.78) / 2$	O_{24}
	$+ (2.16 + 3.16 + 1.83) / 3 - (0.58 + 2.16 + 3.16 + 1.83) / 4$ $+ (3.16 + 1.83 + 0.25) / 3 - (2.16 + 3.16 + 1.83 + 0.25) / 4$	O_{25}
	$+ 3.16 - (2.16 + 3.16) / 2 + 1.83 - (3.16 + 1.83) / 2$	O_{28}
	$+ (2.11 + 3.11 + 1.68) / 3 - (1.03 + 2.11 + 3.11 + 1.68) / 4$	O_{29}
	$+ 3.11 - (2.11 + 3.11) / 2$	O_{32}
	$+ 1.68 - (3.11 + 1.68) / 2$	O_{34}
	$= 0.05 + 0 + 0.79 + 0 - 0.23 - 0.5 + 0.35 -0.17 + 0.32 + 0.5 - 0.72 = 0.84$	

s	q_1 (s)	q_2 (s)
8	1.1	0
9	1.2	1
10	0.2	2
11	1.2	1
12	2.78	0
13	2.78	0.58
14	1.78	2.16
15	1.03	3.16
16	2.11	1.83
17	3.11	0.25
18	1.68	0

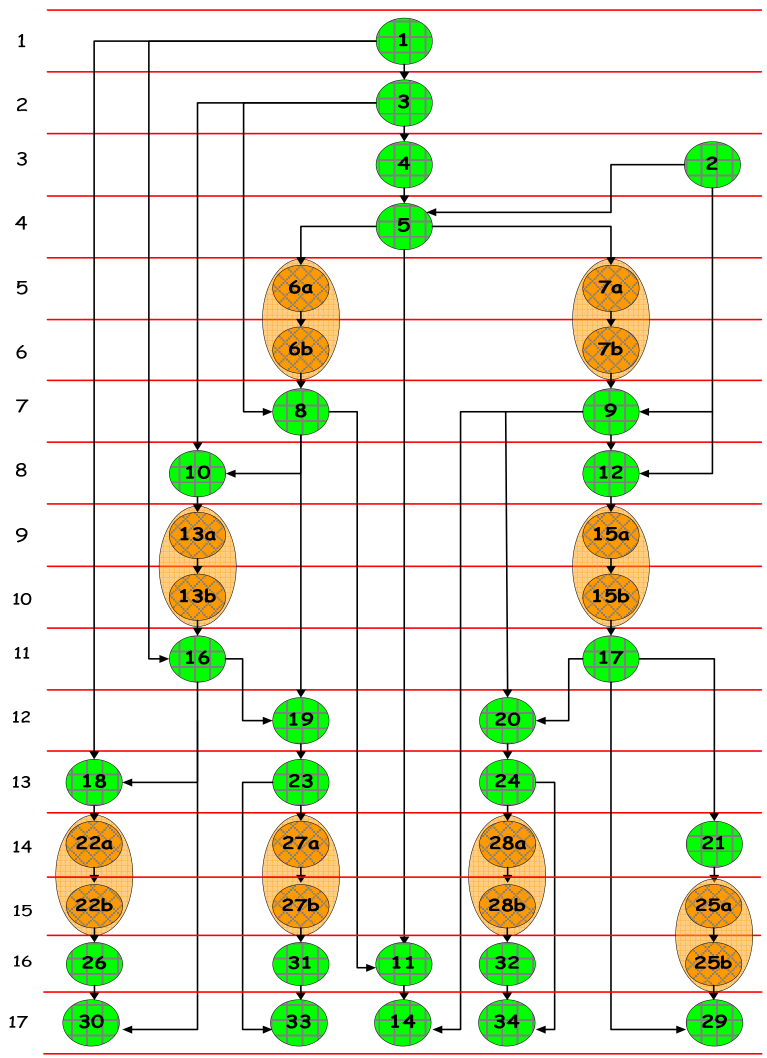
DG_1 DG_2

O_i	12	15	17	20	21	24	25	28	29	32	34	
TF_i	[8,9]	[9,10]	[11,12]	[12,13]	[12,15]	[13,14]	[13,16]	[14,15]	[15,18]	[16,17]	[17,18]	
TF_i'	[9]	[10]	[12]	[13]	[13,15]	[14]	[14,16]	[15]	[16,18]	[17]	[18]	
DF_i	0.05	0	0.79	0	- 0.23	- 0.5	0.35	- 0.17	0.32	0.5	- 0.72	0.84

$DF_{11} < DF_{12} < DF_{10}$

Operazione da differire: O_{11}

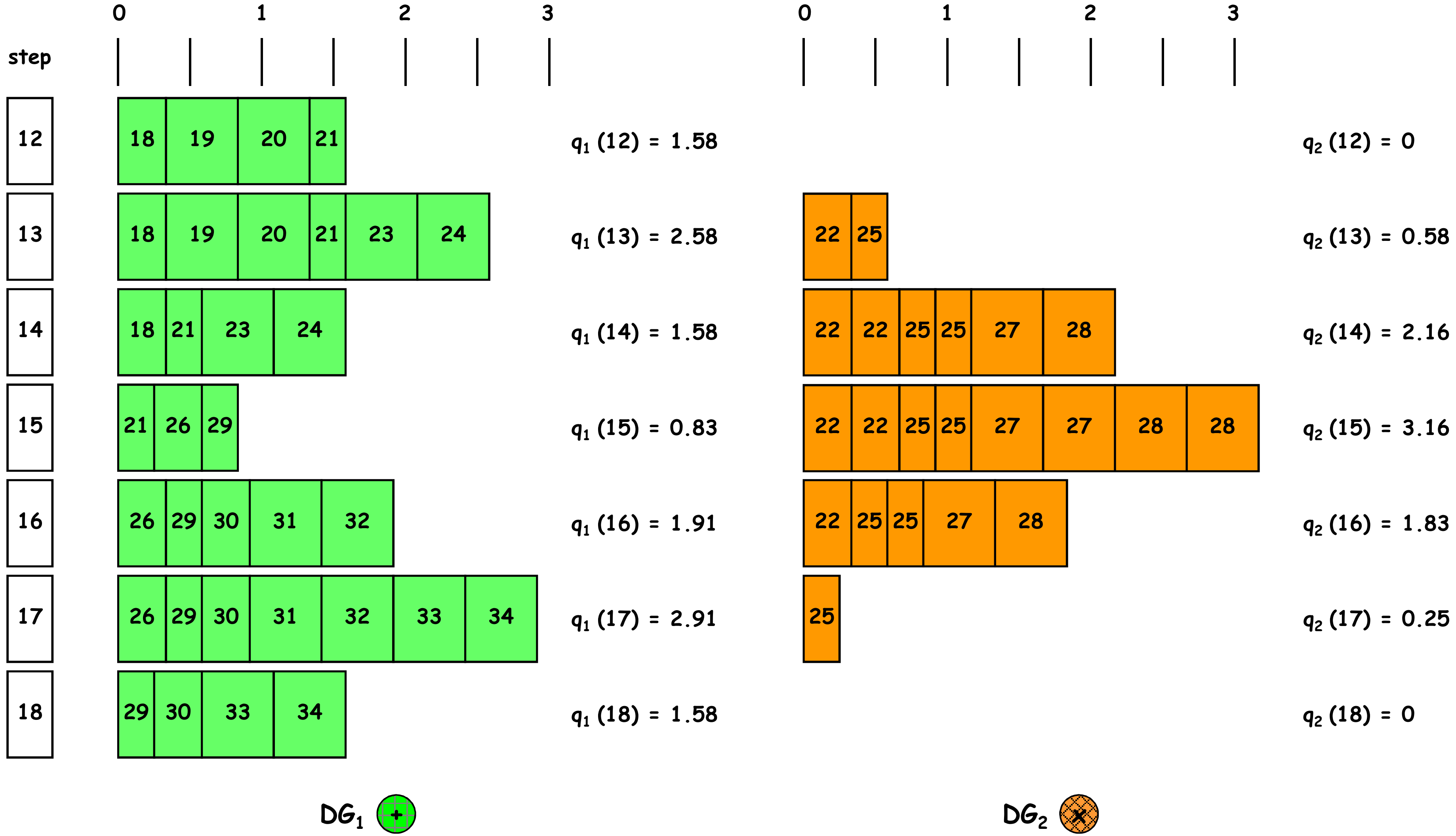
ALAP SSG (EWF$_{SG}$, $\lambda=\lambda_{min}=17$)



	Op. candidate	Op. schedulate
1	O_1, O_2	O_1, O_2
2	O_3	O_3
3	O_4	O_4
4	O_5	O_5
5	O_6, O_7	O_6, O_7
6	-	-
7	O_8, O_9	O_8, O_9
8	O_{10}, O_{12}, O_{11}	O_{10}, O_{12}
9	O_{13}, O_{15}, O_{11}	O_{13}, O_{15}, O_{11}
10	O_{14}	O_{14}
11	O_{16}, O_{17}	O_{16}, O_{17}
12	$O_{19}, O_{20}, O_{18}, O_{21}$	?,?

Quale coppia di operazioni conviene differire ?

O_i	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
TF_i	[12,14]	[12,13]	[12,13]	[12,15]	[13,15]	[13,14]	[13,14]	[13,16]	[15,17]	[14,15]	[14,15]	[15,18]	[16,18]	[16,17]	[16,17]	[17,18]	[17,18]



La "Deferral Force" DF_{18} corrispondente al differimento di O_{18},
con conseguente ridimensionamento del TF anche di O_{22}, O_{26} e O_{30}, vale:

$$
\begin{aligned}
DF_{18} = {} & (2.58 + 1.58) / 2 - (1.58 + 2.58 + 1.58) / 3 + && O_{18} \\
& (2.16 + 3.16) / 2 - (0.58 + 2.16 + 3.16) / 3 + && O_{22} \\
& (3.16 + 1.83) / 2 - (2.16 + 3.16 + 1.83) / 3 + && \\
& (1.91 + 2.91) / 2 - (0.83 + 1.91 + 2.91) / 3 + && O_{26} \\
& (2.91 + 1.58) / 2 - (1.91 + 2.91 + 1.58) / 3 = && O_{30} \\
= {} & 0.17 + 0.8 + 0.53 + 0.11 = 1.61 &&
\end{aligned}
$$

O_i	18	22	26	30	
TF_i	[12,14]	[13,15]	[15,17]	[16,18]	
TF_i'	[13,14]	[14,15]	[16,17]	[17,18]	
DF_i	0.17	0.8	0.53	0.11	1.61

Analogamente si calcolano DF_{19}, DF_{20} e DF_{21}:

s	q_1 (s)	q_2 (s)
12	1.58	0
13	2.58	0.58
14	1.58	2.16
15	0.83	3.16
16	1.91	1.83
17	2.91	0.25
18	1.58	0

DG_1 DG_2

O_i	19	23	27	31	33	
TF_i	[12,13]	[13,14]	[14,15]	[16,17]	[17,18]	
TF_i'	[13]	[14]	[15]	[17]	[18]	
DF_i	0.5	-0.5	-0.16	0.5	-0.66	-0.32

O_i	20	24	28	32	34	
TF_i	[12,13]	[13,14]	[14,15]	[16,17]	[17,18]	
TF_i'	[13]	[14]	[15]	[17]	[18]	
DF_i	0.5	-0.5	-0.16	0.5	-0.66	-0.32

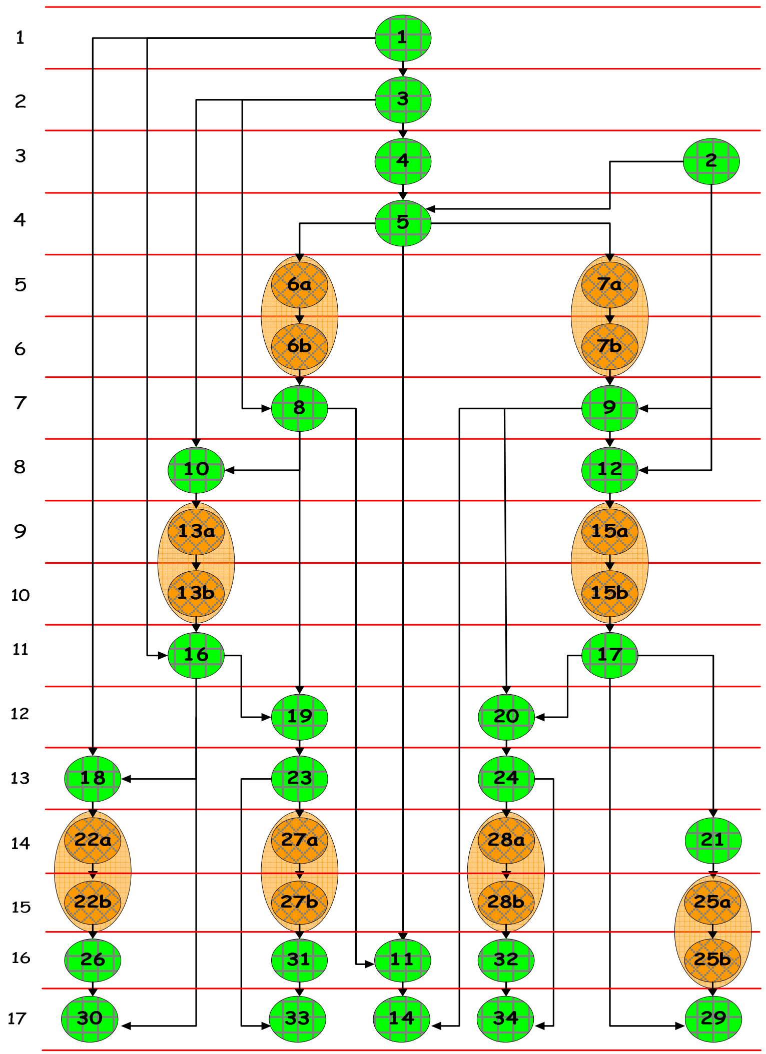
O_i	21	25	29	
TF_i	[12,15]	[13,16]	[15,18]	
TF_i'	[13,15]	[14,16]	[16,18]	
DF_i	0.02	0.35	0.32	0.69

$$DF_{19} = DF_{20} < DF_{21} < DF_{18}$$

Operazioni da differire: O_{19}, O_{20}

Algoritmo FDLS

ALAP SSG (EWF_{SG}, $\lambda = \lambda_{min} = 17$)



Risorse disponibili:

2 + (d_+ = 1 t.u.)

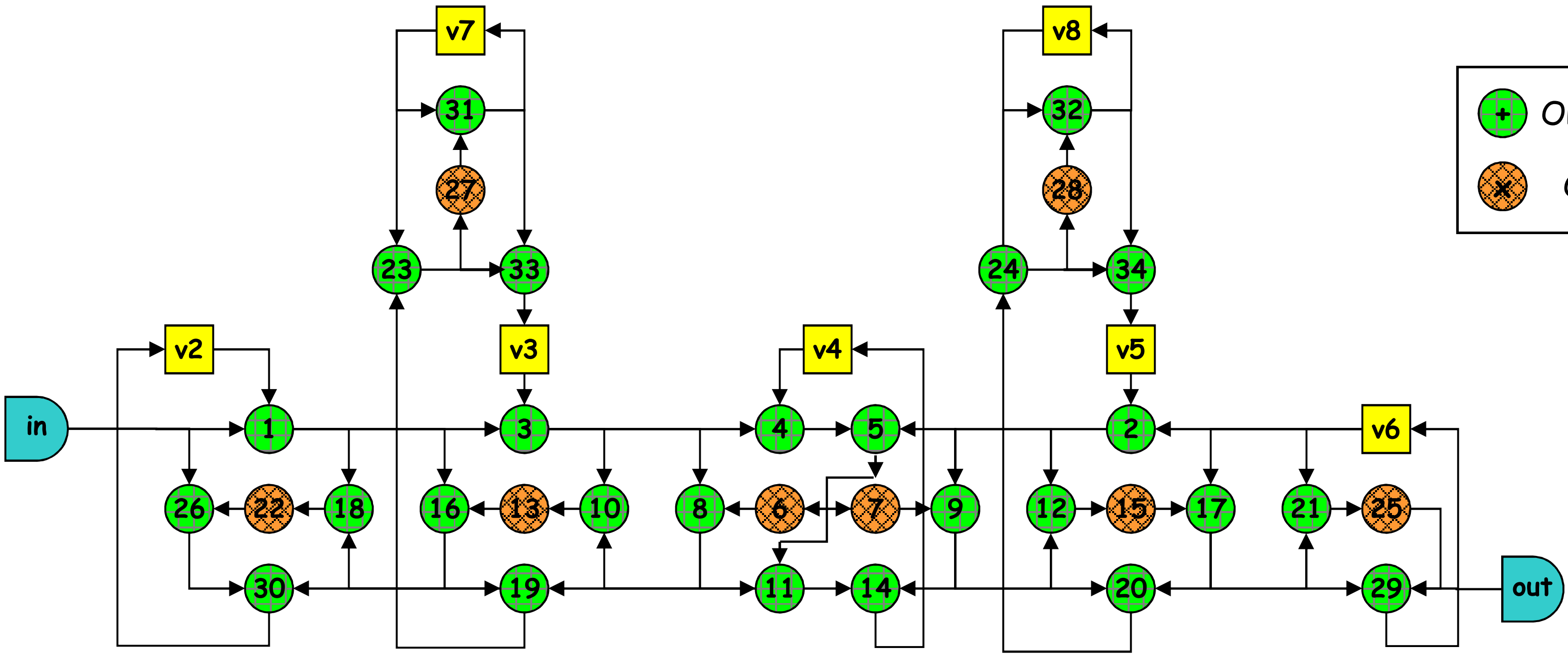
2 × (d_* = 2 t.u.)

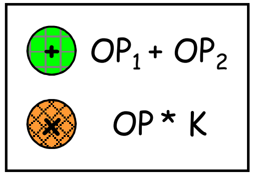
	Op. candidate	Op. schedulate
1	O_1, O_2	O_1, O_2
2	O_3	O_3
3	O_4	O_4
4	O_5	O_5
5	O_6, O_7	O_6, O_7
6	-	-
7	O_8, O_9	O_8, O_9
8	O_{10}, O_{12}	O_{10}, O_{12}
9	O_{13}, O_{15}, O_{11}	O_{13}, O_{15}, O_{11}
10	O_{14}	O_{14}
11	O_{16}, O_{17}	O_{16}, O_{17}
12	$O_{19}, O_{20}, O_{18}, O_{21}$	O_{18}, O_{21}
13	$O_{19}, O_{20}, O_{22}, O_{25}$	$O_{19}, O_{20}, O_{22}, O_{25}$
14	O_{23}, O_{24}	O_{23}, O_{24}
15	$O_{27}, O_{28}, O_{26}, O_{29}$	$O_{27}, O_{28}, O_{26}, O_{29}$
16	O_{30}	O_{30}
17	O_{31}, O_{32}	O_{31}, O_{32}
18	O_{33}, O_{34}	O_{33}, O_{34}

Latenza: 18 t.u.

Algoritmi "time-constrained": un altro caso di studio

5th order Elliptic Wave Filter (EWF)





2 tipologie di risorse disponibili:

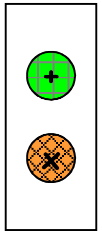
\+ d_+ = 1 t.u., c_+ = 2 u.

* d_* = 2 t.u., c_* = 5 u.

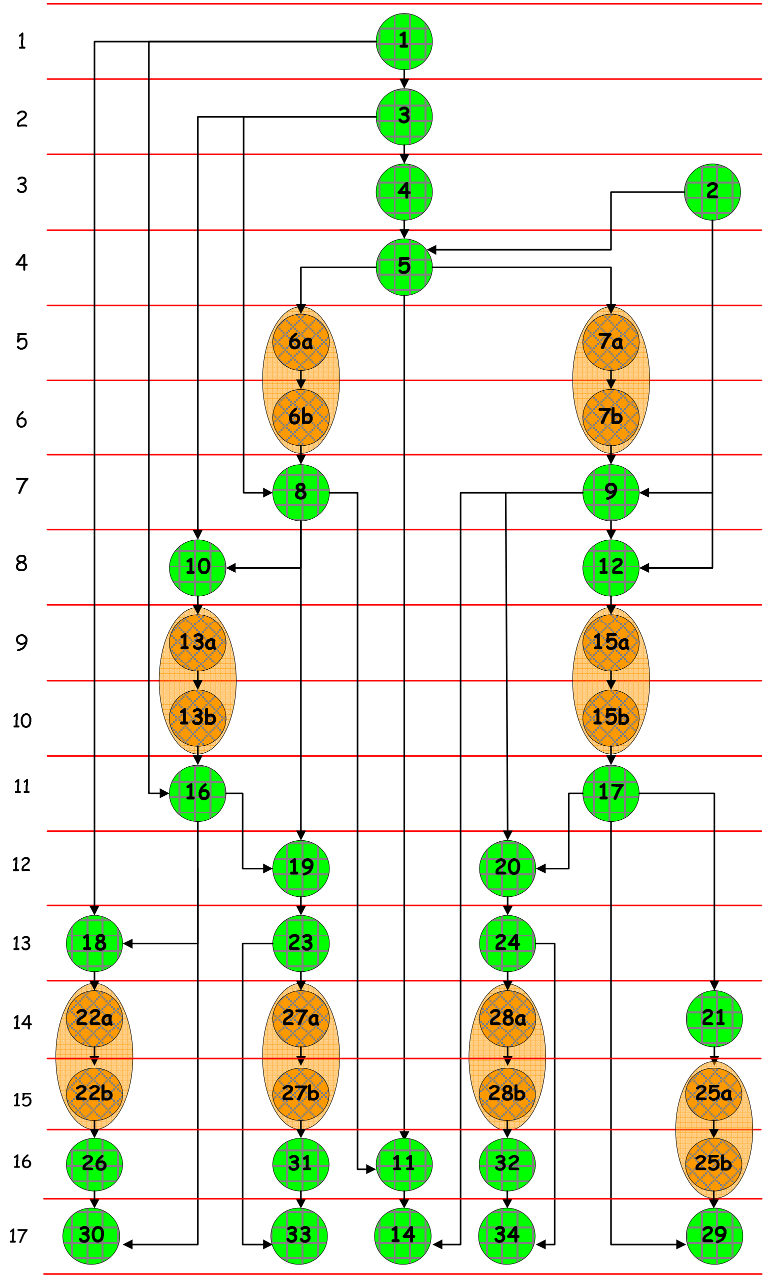
Vincolo:
- latenza minima (λ_{min} = 17 t.u.)

Obiettivo:
- insieme di risorse di costo minimo

Algoritmo VSLS

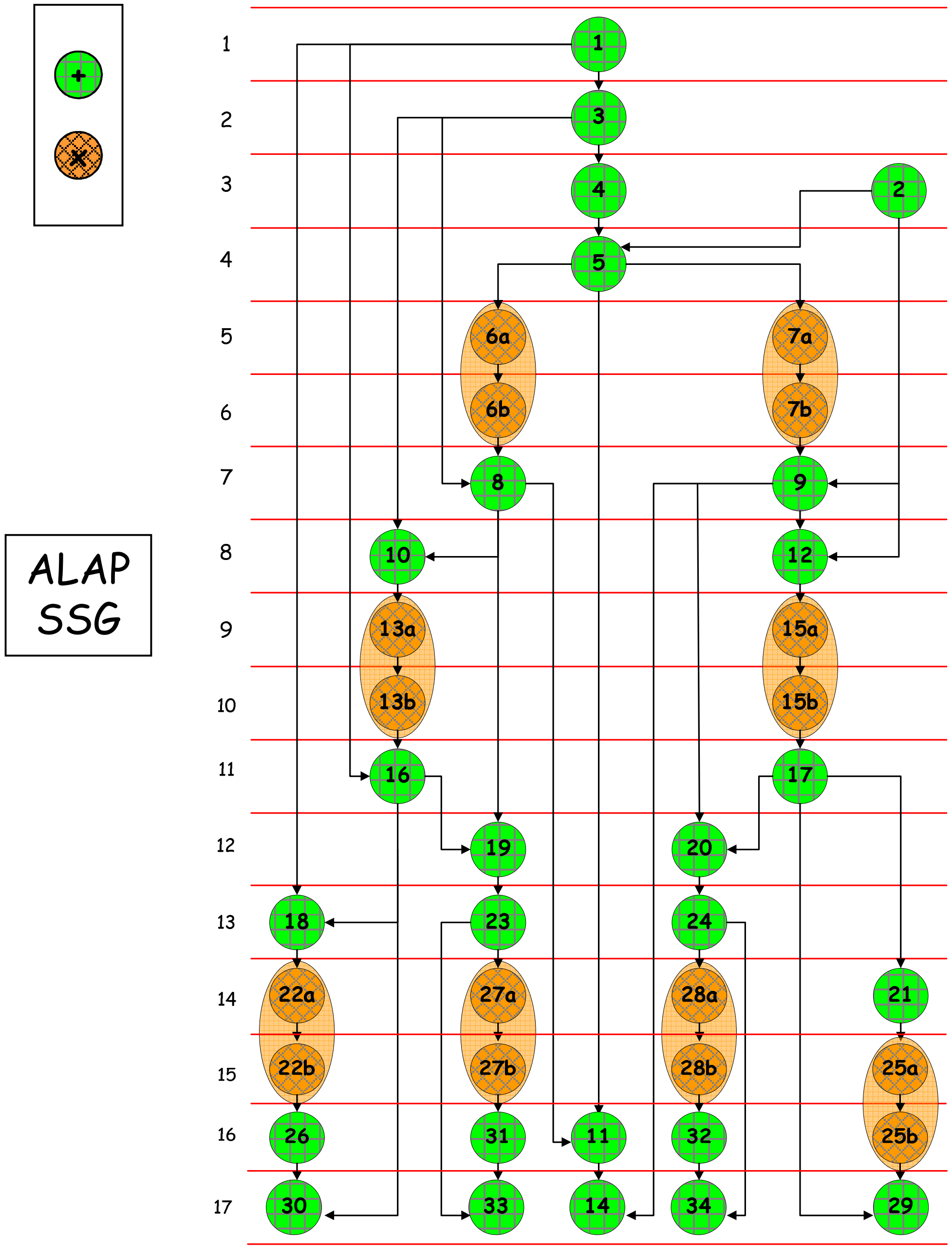


ALAP SSG



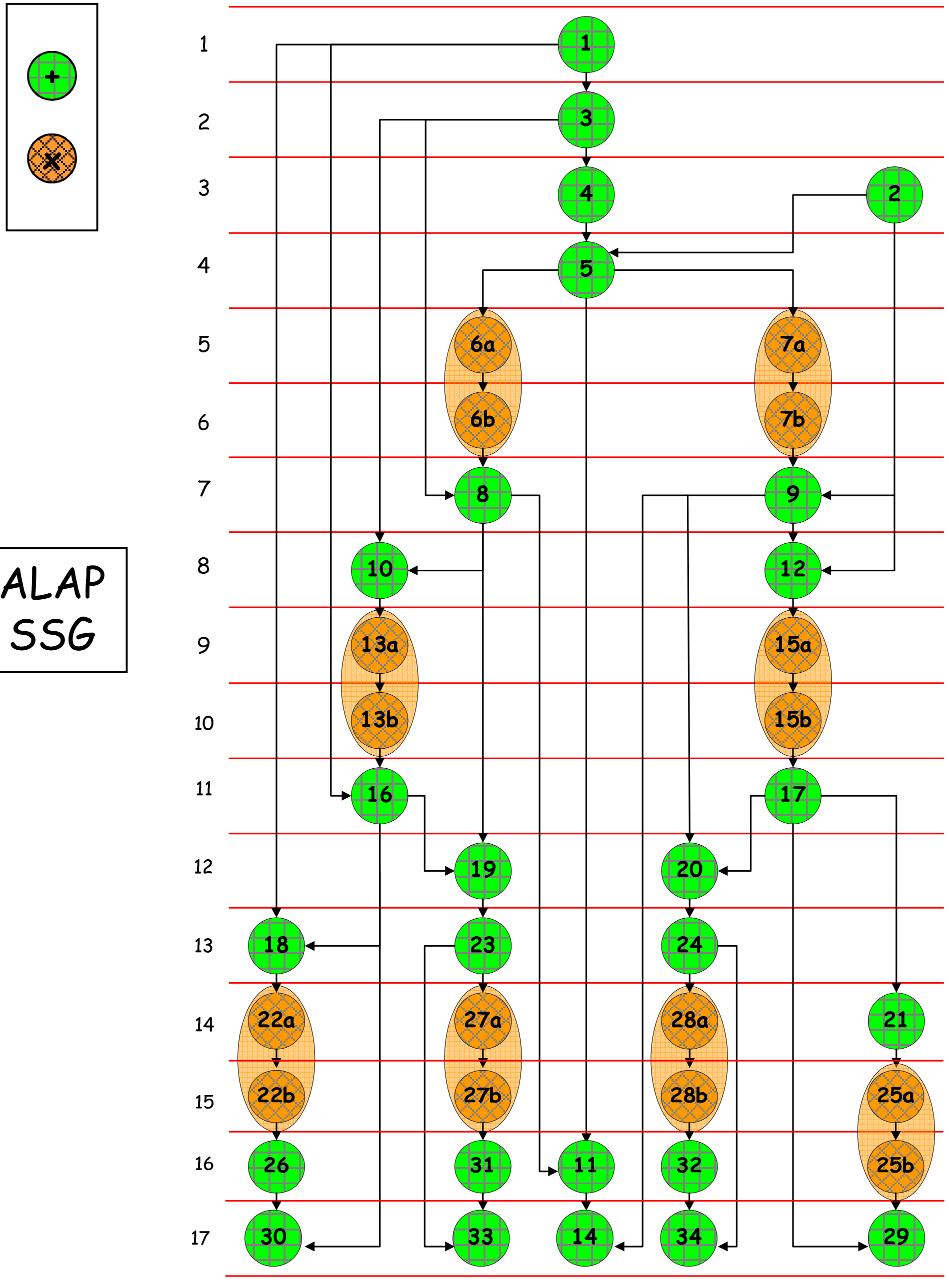
Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0), O_2(2)$	[1,1]	O_1
$O_3(0), O_2(1)$		O_3
$O_2(0), O_4(0)$	[2,1]	O_2, O_4
$O_5(0)$		O_5
$O_6(0), O_7(0)$	[2,2]	O_6, O_7
-		-
$O_8(0), O_9(0)$		O_8, O_9
$O_{10}(0), O_{12}(0), O_{11}(8)$		O_{10}, O_{12}
$O_{13}(0), O_{15}(0), O_{11}(7)$		O_{13}, O_{15}, O_{11}
$O_{14}(7)$		O_{14}
$O_{16}(0), O_{17}(0)$		O_{16}, O_{17}
$O_{19}(0), O_{20}(0), O_{18}(1), O_{21}(2)$		O_{19}, O_{20}
$O_{18}(0), O_{23}(0), O_{24}(0), O_{21}(1)$	[3,2]	O_{18}, O_{23}, O_{24}
$O_{21}(0), O_{22}(0), O_{27}(0), O_{28}(0)$	[3,3]	$O_{21}, O_{22}, O_{27}, O_{28}$
$O_{25}(0)$	[3,4]	O_{25}
$O_{26}(0), O_{31}(0), O_{32}(0)$		O_{26}, O_{31}, O_{32}
$O_{29}(0), O_{30}(0), O_{33}(0), O_{34}(0)$	[4,4]	$O_{29}, O_{30}, O_{33}, O_{34}$

Algoritmo ILS (1ª iterazione)



Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0), O_2(2)$	[1,1]	O_1
$O_3(0), O_2(1)$		O_3
$O_2(0), O_4(0)$	[2,1]	

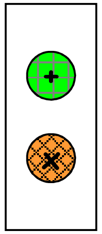
Algoritmo ILS (2ª iterazione)



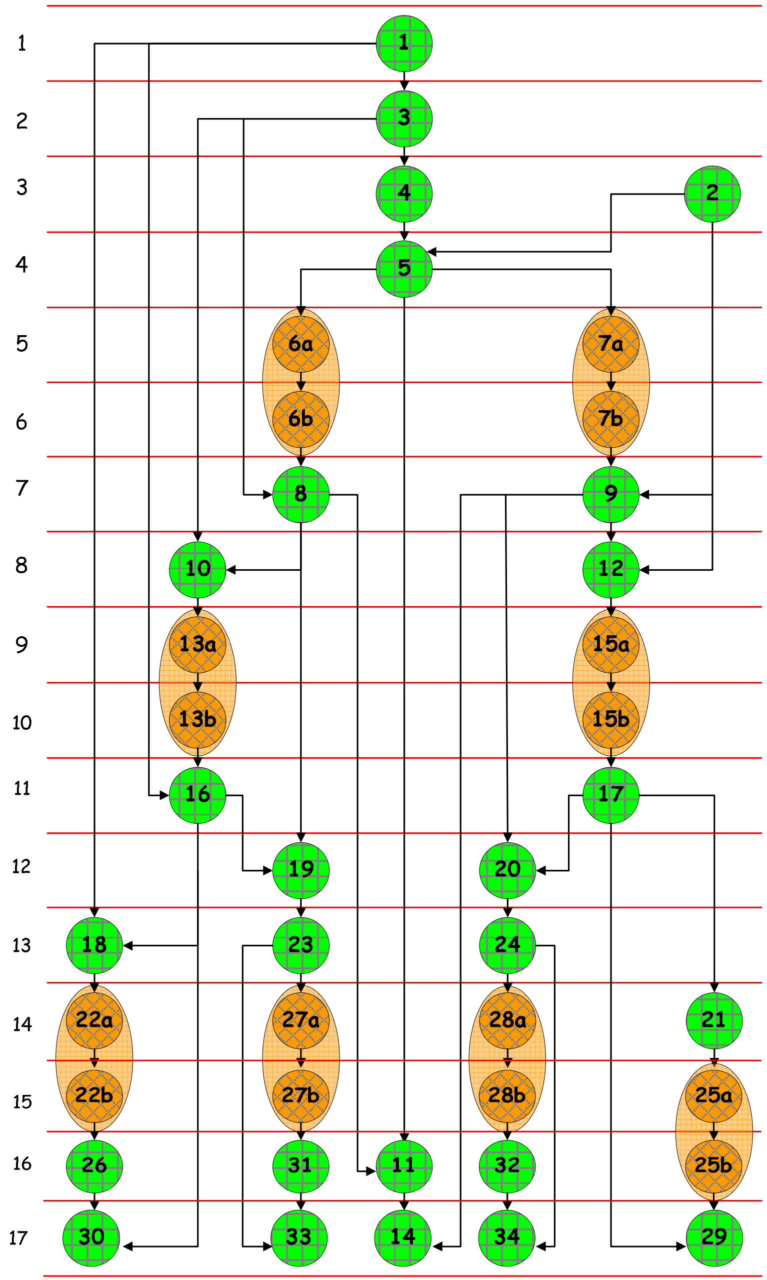
ALAP
SSG

Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0), O_2(2)$	[2,1]	O_1, O_2
$O_3(0)$		O_3
$O_4(0)$		O_4
$O_5(0)$		O_5
$O_6(0), O_7(0)$	[2,2]	

Algoritmo ILS (3ª iterazione)

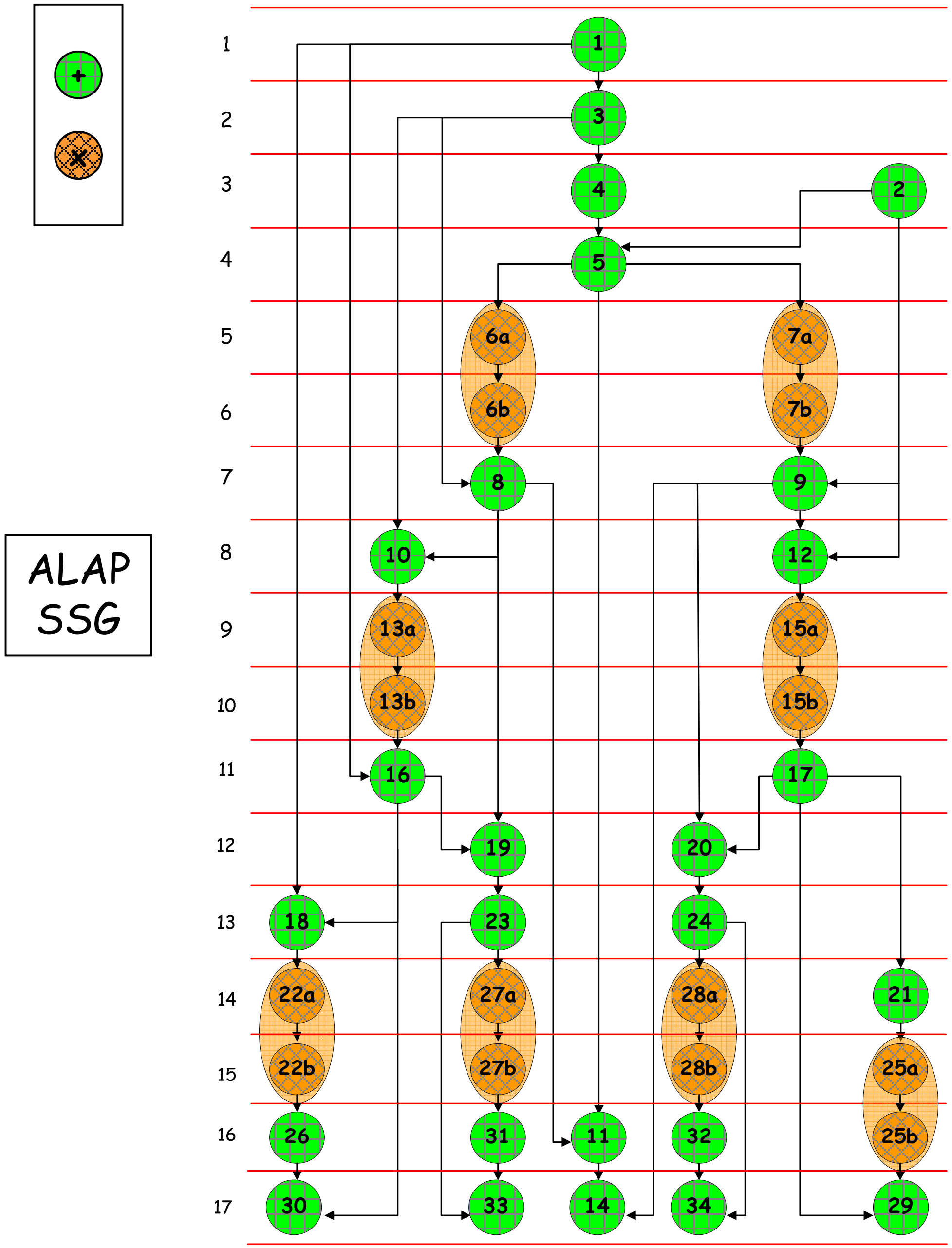


ALAP
SSG



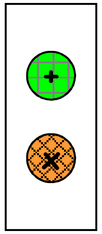
Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0),O_2(2)$	[2,2]	O_1,O_2
$O_3(0)$		O_3
$O_4(0)$		O_4
$O_5(0)$		O_5
$O_6(0),O_7(0)$		O_6,O_7
-		-
$O_8(0),O_9(0)$		O_8,O_9
$O_{10}(0),O_{12}(0),O_{11}(8)$		O_{10},O_{12}
$O_{13}(0),O_{15}(0),O_{11}(7)$		O_{13},O_{15},O_{11}
$O_{14}(7)$		O_{14}
$O_{16}(0),O_{17}(0)$		O_{16},O_{17}
$O_{19}(0),O_{20}(0),O_{18}(1),O_{21}(2)$		O_{19},O_{20}
$O_{18}(0),O_{23}(0),O_{24}(0),O_{21}(1)$	[3,2]	

Algoritmo ILS (4ª iterazione)

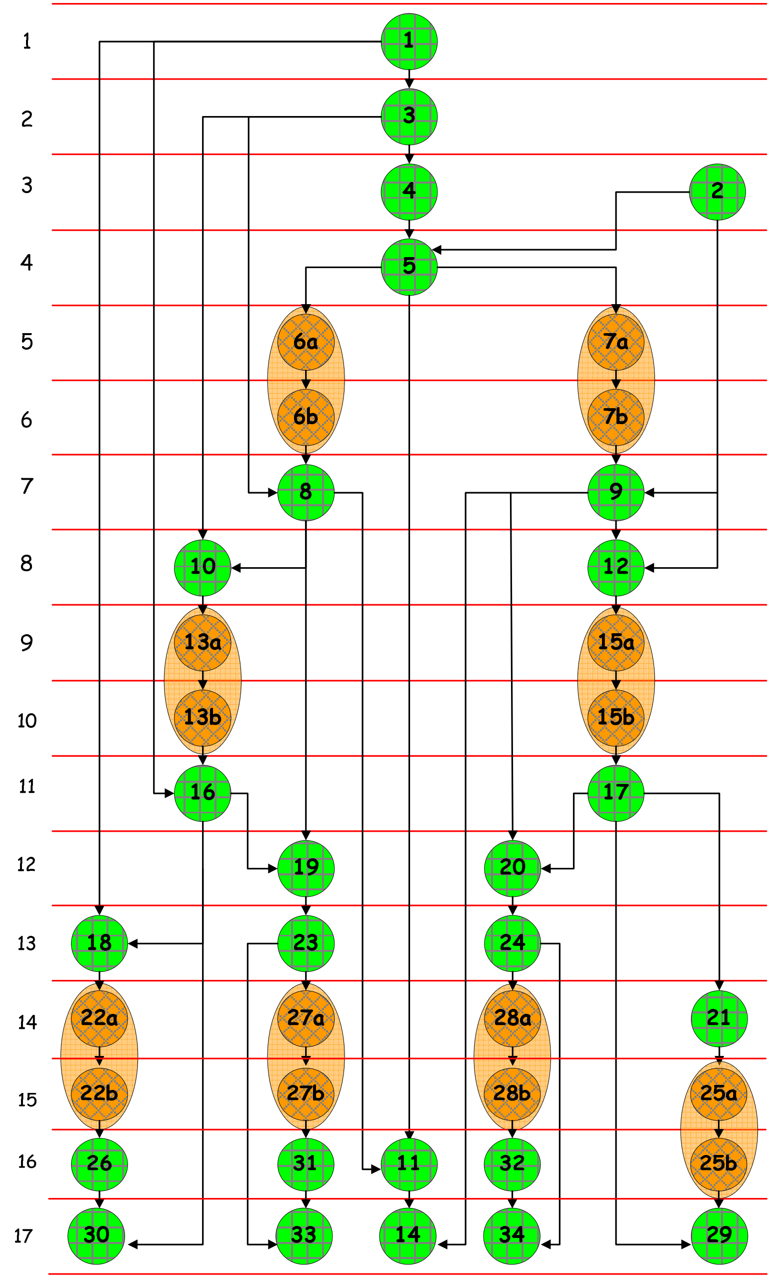


Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0), O_2(2)$	[3,2]	O_1, O_2
$O_3(0)$		O_3
$O_4(0)$		O_4
$O_5(0)$		O_5
$O_6(0), O_7(0)$		O_6, O_7
-		-
$O_8(0), O_9(0)$		O_8, O_9
$O_{10}(0), O_{12}(0), O_{11}(8)$		O_{10}, O_{11}, O_{12}
$O_{13}(0), O_{15}(0), O_{14}(8)$		O_{13}, O_{14}, O_{15}
-		-
$O_{16}(0), O_{17}(0)$		O_{16}, O_{17}
$O_{19}(0), O_{20}(0), O_{18}(1), O_{21}(2)$		O_{18}, O_{19}, O_{20}
$O_{23}(0), O_{24}(0), O_{21}(1), O_{22}(1)$		$O_{21}, O_{22}, O_{23}, O_{24}$
$O_{27}(0), O_{28}(0), O_{25}(1)$	[3,3]	

Algoritmo ILS (5^a iterazione)

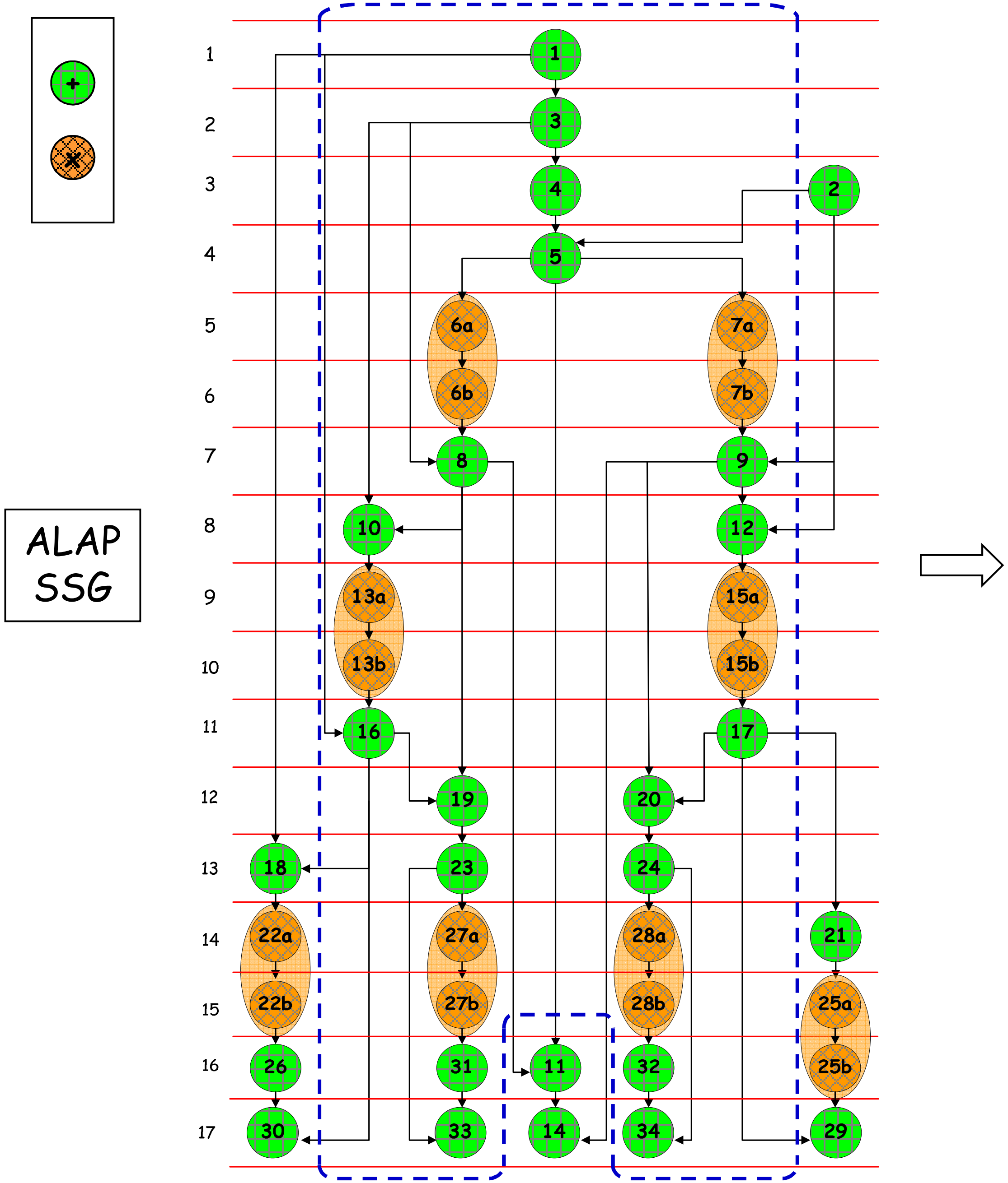


ALAP
SSG



Op.candidate (slack)	[+,x]	Op. schedulate
$O_1(0), O_2(2)$	[3,3]	O_1, O_2
$O_3(0)$		O_3
$O_4(0)$		O_4
$O_5(0)$		O_5
$O_6(0), O_7(0)$		O_6, O_7
-		-
$O_8(0), O_9(0)$		O_8, O_9
$O_{10}(0), O_{12}(0), O_{11}(8)$		O_{10}, O_{11}, O_{12}
$O_{13}(0), O_{15}(0), O_{14}(8)$		O_{13}, O_{14}, O_{15}
-		-
$O_{16}(0), O_{17}(0)$		O_{16}, O_{17}
$O_{19}(0), O_{20}(0), O_{18}(1), O_{21}(2)$		O_{18}, O_{19}, O_{20}
$O_{23}(0), O_{24}(0), O_{21}(1), O_{22}(1)$		$O_{21}, O_{22}, O_{23}, O_{24}$
$O_{27}(0), O_{28}(0), O_{25}(1)$		O_{27}, O_{28}
$O_{25}(0), O_{26}(1)$		O_{25}, O_{26}
$O_{31}(0), O_{32}(0), O_{30}(1)$		O_{30}, O_{31}, O_{32}
$O_{29}(0), O_{33}(0), O_{34}(0)$		O_{29}, O_{33}, O_{34}

Algoritmi VSLS* e ILS* ...

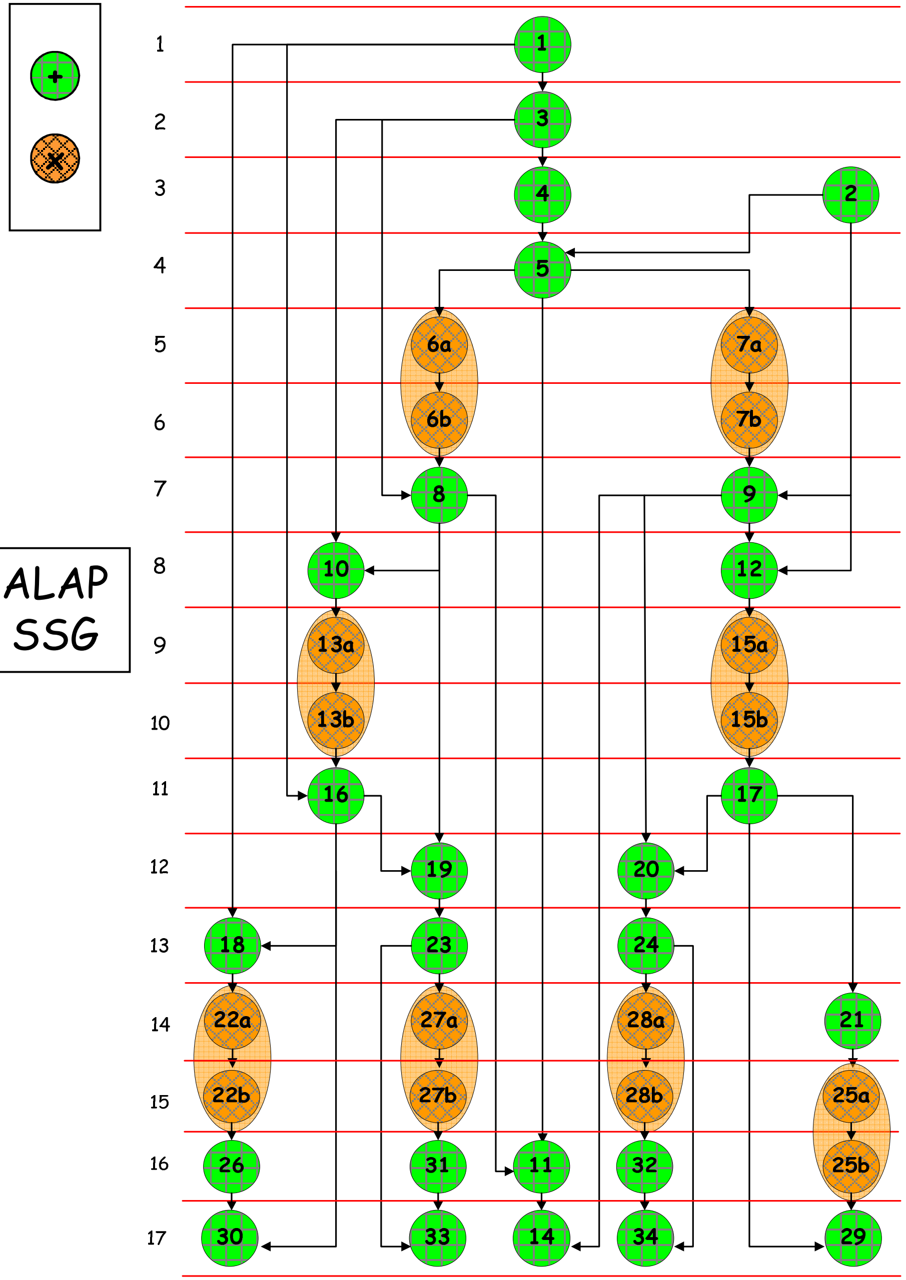


identificazione di LB_k
da assumere come valore iniziale di a_k
$k = 1, ..., N_R$:

$\lvert V_\times \rvert = 8$	$\lvert V_+ \rvert = 26$
$\lvert Z_\times \rvert = 6$	$\lvert Z_+ \rvert = 18$
$\nu_\times = 2$	$\nu_+ = 2$
$ALAP_\times = 15$	$ALAP_+ = 17$
$ASAP_\times = 5$	$ASAP_+ = 1$
$\lambda_\times = 12$	$\lambda_+ = 17$
$N_\times = 16$	$N_+ = 26$
$n_\times = 2$	$n_+ = 2$
$\mathbf{a_\times = 2}$	$\mathbf{a_+ = 2}$

... Algoritmi VSLS* e ILS*

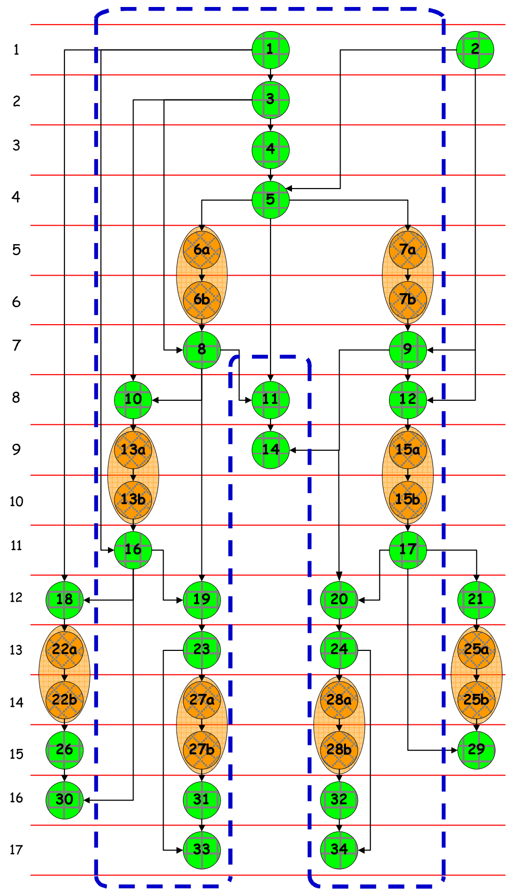
ALAP SSG



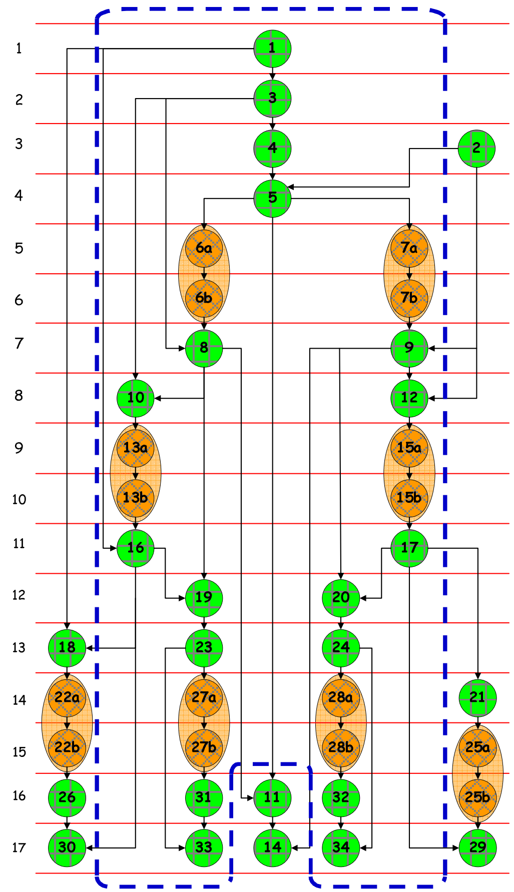
N_+, λ_+	$N_\times, \lambda_\times$	Op.candidate (slack)	[+,x]	Op. schedulate
26,17	16,12	$O_1(0), O_2(2)$	[2,2]	O_1, O_2
24,16	16,12	$O_3(0)$		O_3
23,15	16,12	$O_4(0)$		O_4
22,14	16,12	$O_5(0)$		O_5
21,13	16,12	$O_6(0), O_7(0)$		O_6, O_7
21,12	14,11	-		-
21,11	12,10	$O_8(0), O_9(0)$		O_8, O_9
19,10	12,9	$O_{10}(0), O_{12}(0), O_{11}(8)$		O_{10}, O_{12}
17,9	12,8	$O_{13}(0), O_{15}(0), O_{11}(7)$		O_{13}, O_{15}, O_{11}
16,8	10,7	$O_{14}(7)$		O_{14}
15,7	8,6	$O_{16}(0), O_{17}(0)$	[3,2]	O_{16}, O_{17}
13,6	8,5	$O_{19}(0), O_{20}(0), O_{18}(1), O_{21}(2)$		O_{19}, O_{20}, O_{18}
10,5	8,4	$O_{23}(0), O_{24}(0), O_{21}(1), O_{22}(1)$		$O_{23}, O_{24}, O_{21}, O_{22}$
7,4	7,3	$O_{27}(0), O_{28}(0), O_{25}(1)$	[3,3]	O_{27}, O_{28}
7,3	4,2	$O_{25}(0), O_{26}(1)$		O_{25}, O_{26}
6,2	1,1	$O_{31}(0), O_{32}(0), O_{30}(1)$		O_{31}, O_{32}, O_{30}
3,1		$O_{29}(0), O_{33}(0), O_{34}(0)$		O_{29}, O_{33}, O_{34}

Algoritmo FDS: time frames

ASAP

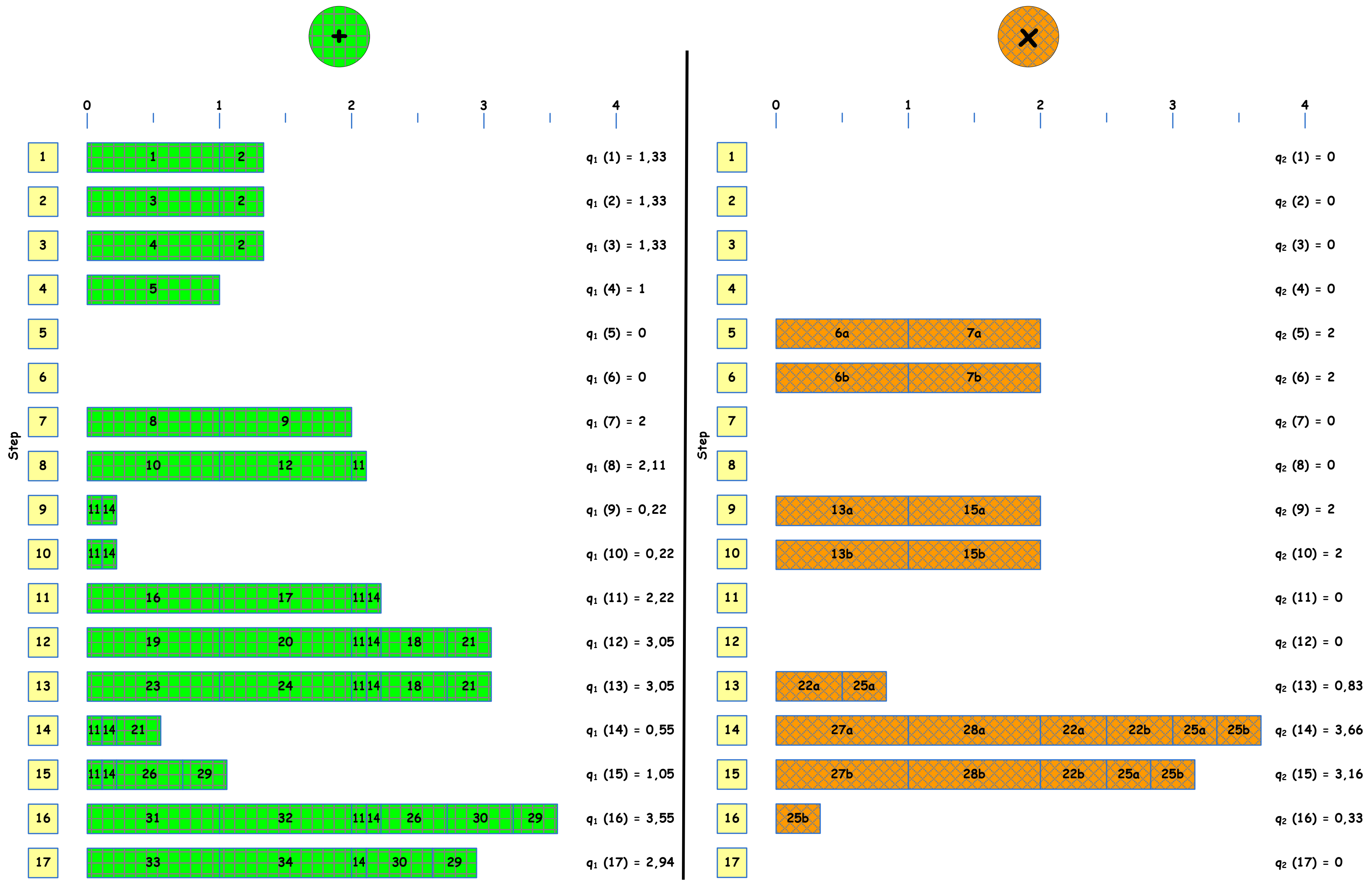


ALAP



O_i	t_i^S	t_i^L	TF_i	μ_i	w_i
O_1	1	1	[1,1]	0	1
O_2	1	3	[1,3]	2	3
O_3	2	2	[2,2]	0	1
O_4	3	3	[3,3]	0	1
O_5	4	4	[4,4]	0	1
O_6	5	5	[5,5]	0	1
O_7	5	5	[5,5]	0	1
O_8	7	7	[7,7]	0	1
O_9	7	7	[7,7]	0	1
O_{10}	8	8	[8,8]	0	1
O_{11}	8	16	[8,16]	8	9
O_{12}	8	8	[8,8]	0	1
O_{13}	9	9	[9,9]	0	1
O_{14}	9	17	[9,17]	8	9
O_{15}	9	9	[9,9]	0	1
O_{16}	11	11	[11,11]	0	1
O_{17}	11	11	[11,11]	0	1
O_{18}	12	13	[12,13]	1	2
O_{19}	12	12	[12,12]	0	1
O_{20}	12	12	[12,12]	0	1
O_{21}	12	14	[12,14]	2	3
O_{22}	13	14	[13,14]	1	2
O_{23}	13	13	[13,13]	0	1
O_{24}	13	13	[13,13]	0	1
O_{25}	13	15	[13,15]	2	3
O_{26}	15	16	[15,16]	1	2
O_{27}	14	14	[14,14]	0	1
O_{28}	14	14	[14,14]	0	1
O_{29}	15	17	[15,17]	2	3
O_{30}	16	17	[16,17]	1	2
O_{31}	16	16	[16,16]	0	1
O_{32}	16	16	[16,16]	0	1
O_{33}	17	17	[17,17]	0	1
O_{34}	17	17	[17,17]	0	1

Algoritmo FDS (1ª iterazione): distribution graphs



Algoritmo FDS (1ª iterazione): self/indirect forces

i,l	SF	IF	F
2, 1	0,00	0,00	0,00
2, 2	0,00	0,00	0,00
2, 3	0,00	0,00	0,00

i,l	SF	IF	F
11, 8	0,33	0,00	0,33
11, 9	-1,56	0,21	-1,35
11, 10	-1,56	0,47	-1,09
11, 11	0,44	0,49	0,93
11, 12	1,27	0,36	1,63
11, 13	1,27	0,15	1,42
11, 14	-1,23	0,64	-0,59
11, 15	-0,73	1,37	0,65
11, 16	1,77	1,07	2,84

i,l	SF	IF	F
14, 9	-1,65	0,33	-1,33
14, 10	-1,65	-0,62	-2,27
14, 11	0,35	-0,93	-0,59
14, 12	1,18	-0,59	0,59
14, 13	1,18	-0,22	0,96
14, 14	-1,32	0.03	-1,29
14, 15	-0,82	-0,15	-0,97
14, 16	1,68	-022	1,46
14, 17	1,07	0,00	1,07

i,l	SF	IF	F
18, 12	0,00	0,00	0,00
18, 13	0,00	2,11	2,11

i,l	SF	IF	F
21, 12	0,83	0,00	0,83
21, 13	0,83	0,95	1,79
21, 14	-1,67	-1,02	-2,69

i,l	SF	IF	F
26, 15	-1,25	-1,17	-2,42
26, 16	1,25	-0,31	0,94

i,l	SF	IF	F
29, 15	-1,46	0,39	-1,07
29, 16	1,04	1,56	2,59
29, 17	0,43	0,00	0,43

i,l	SF	IF	F
30, 16	0,31	-2,42	-2,11
30, 17	-0,31	0,00	-0,31

i,l	SF	IF	F
22, 13	-1,17	0,00	-1,17
22, 14	1,17	0,94	2,11

i,l	SF	IF	F
25, 13	-0,44	0,83	0,39
25, 14	1,89	1,56	3,45
25, 15	-1,44	0,43	-1,02

SF (30, 16) =
$q_+ (16) - [q_+ (16) + q_+ (17)] / 2$
= 0,31

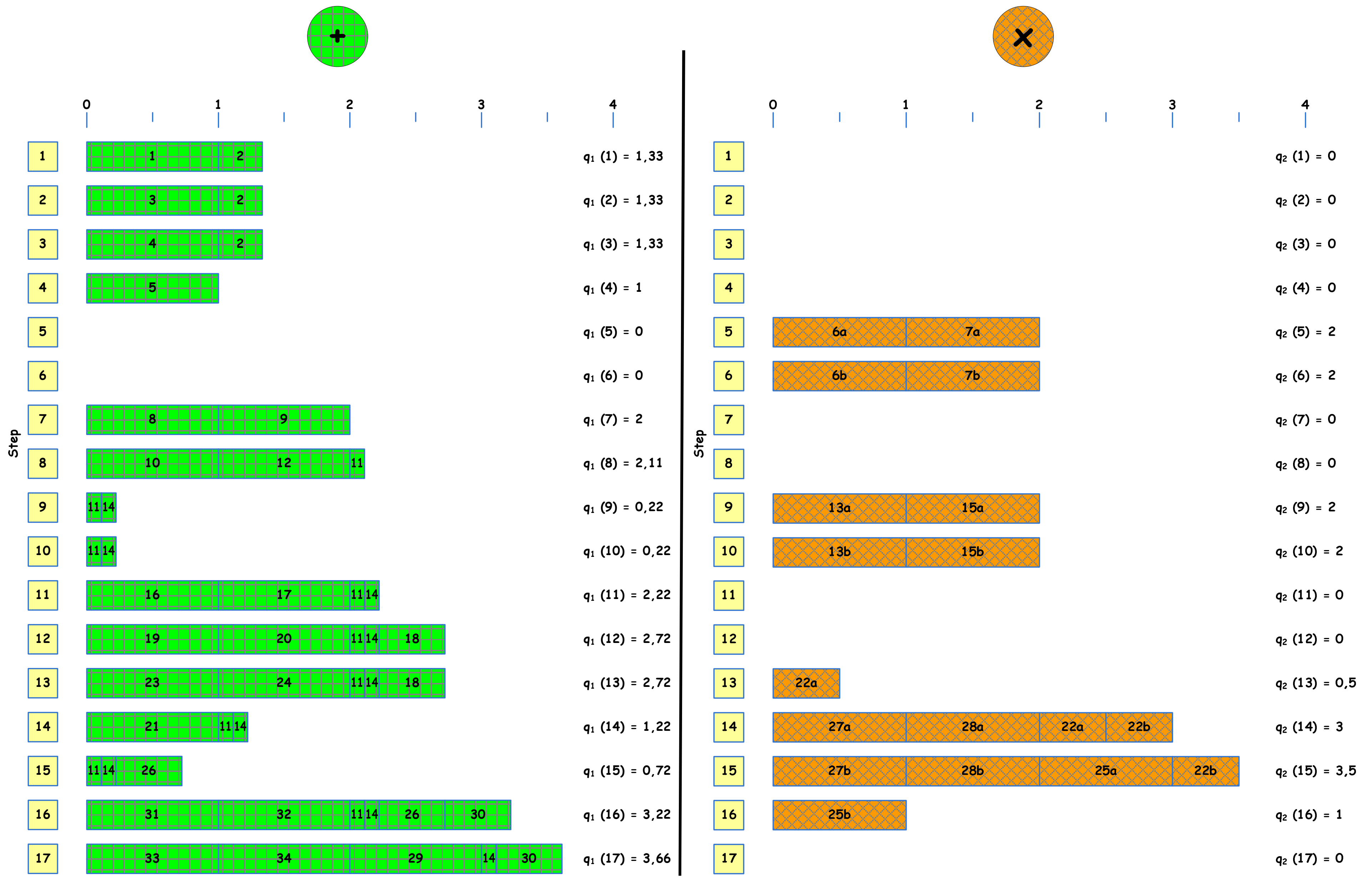
IF_{26} (30, 16) =
$q_+ (15) - [q_+ (15) + q_+ (16)] / 2$
= - 1,25

IF_{22} (30, 16) =
$q_x (13) - [q_x (13) + q_x (14)] / 2 +$
$q_x (14) - [q_x (14) + q_x (15)] / 2$
= - 1,17

IF_{18} (30, 16) =
$q_+ (12) - [q_+ (12) + q_+ (13)] / 2$
= 0

IF (30, 16) =
IF_{26} (30, 16) + IF_{22} (30, 16) + IF_{18} (30, 16)
= -2,42

Algoritmo FDS (2a iterazione): distribution graphs



Algoritmo FDS (2^a iterazione): self/indirect forces

i,l	SF	IF	F
2, 1	0,00	0,00	0,00
2, 2	0,00	0,00	0,00
2, 3	0,00	0,00	0,00

i,l	SF	IF	F
11, 8	0,40	0,00	0,40
11, 9	-1,49	0,21	-1,28
11, 10	-1,49	0,47	-1,01
11, 11	0,51	0,49	1,01
11, 12	1,01	0,42	1,44
11, 13	1,01	0,32	1,33
11, 14	-0,49	0,64	0,15
11, 15	-0,99	1,54	0,55
11, 16	1,51	1,73	3,25

i,l	SF	IF	F
14, 9	-1,65	0,40	-1,25
14, 10	-1,65	-0,54	-2,20
14, 11	0,35	-0,86	-0,51
14, 12	0,85	-0,52	0,33
14, 13	0,85	-0,21	0,64
14, 14	-0,65	-0.01	-0,66
14, 15	-1,15	-0,07	-1,23
14, 16	1,35	-019	1,16
14, 17	1,73	0,00	1,73

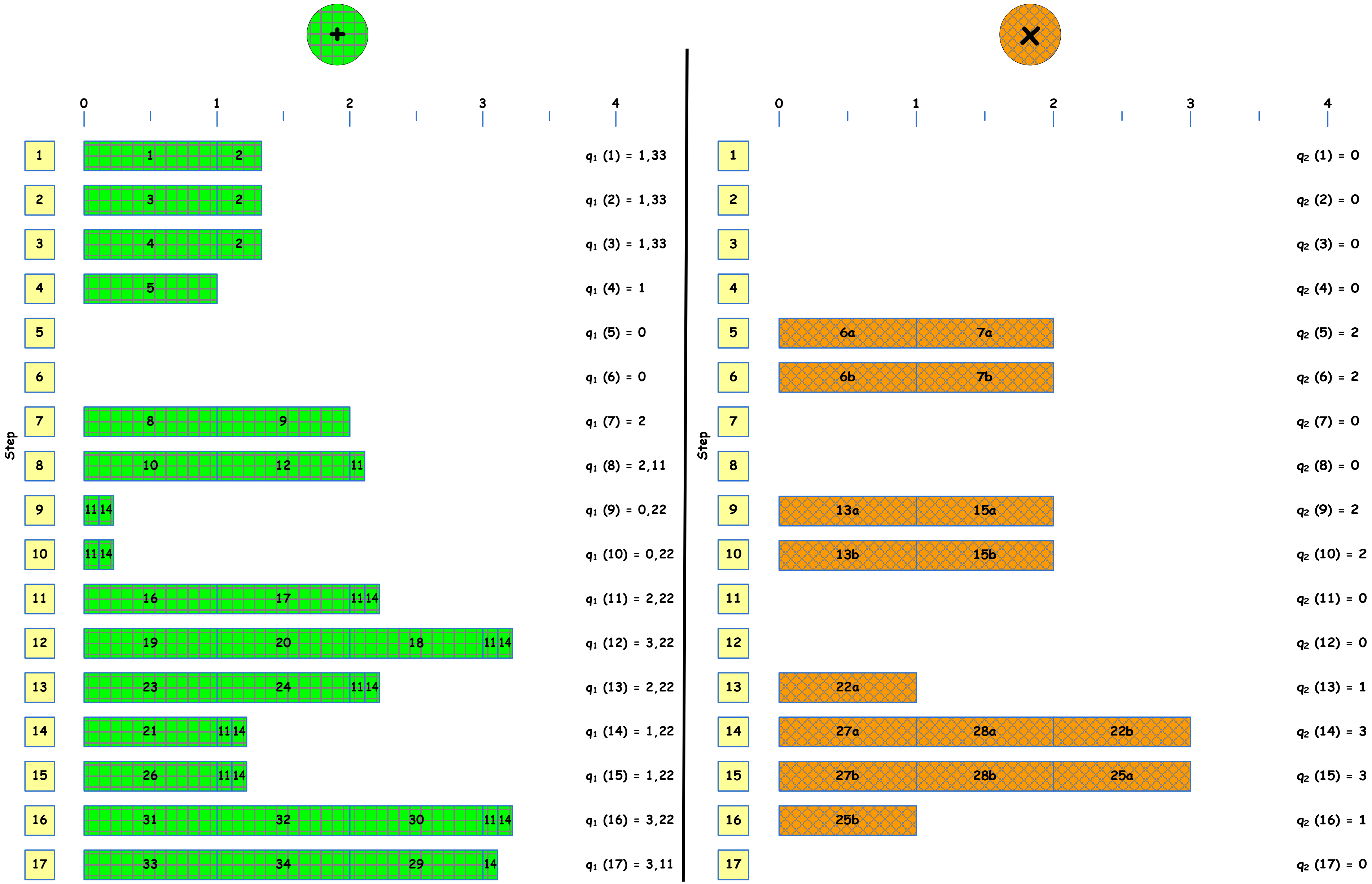
i,l	SF	IF	F
18, 12	0,00	0,00	0,00
18, 13	0,00	2,94	2,94

i,l	SF	IF	F
26, 15	-1,25	-1,50	-2,75
26, 16	1,25	0,19	1,44

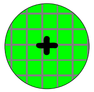
i,l	SF	IF	F
30, 16	-0,19	-2,75	-2,94
30, 17	0,19	0,00	0,19

i,l	SF	IF	F
22, 13	-1,50	0,00	-1,50
22, 14	1,50	1,44	2,94

Algoritmo FDS (3ª iterazione): distribution graphs



Algoritmo FDS (3^a iterazione): self/indirect forces

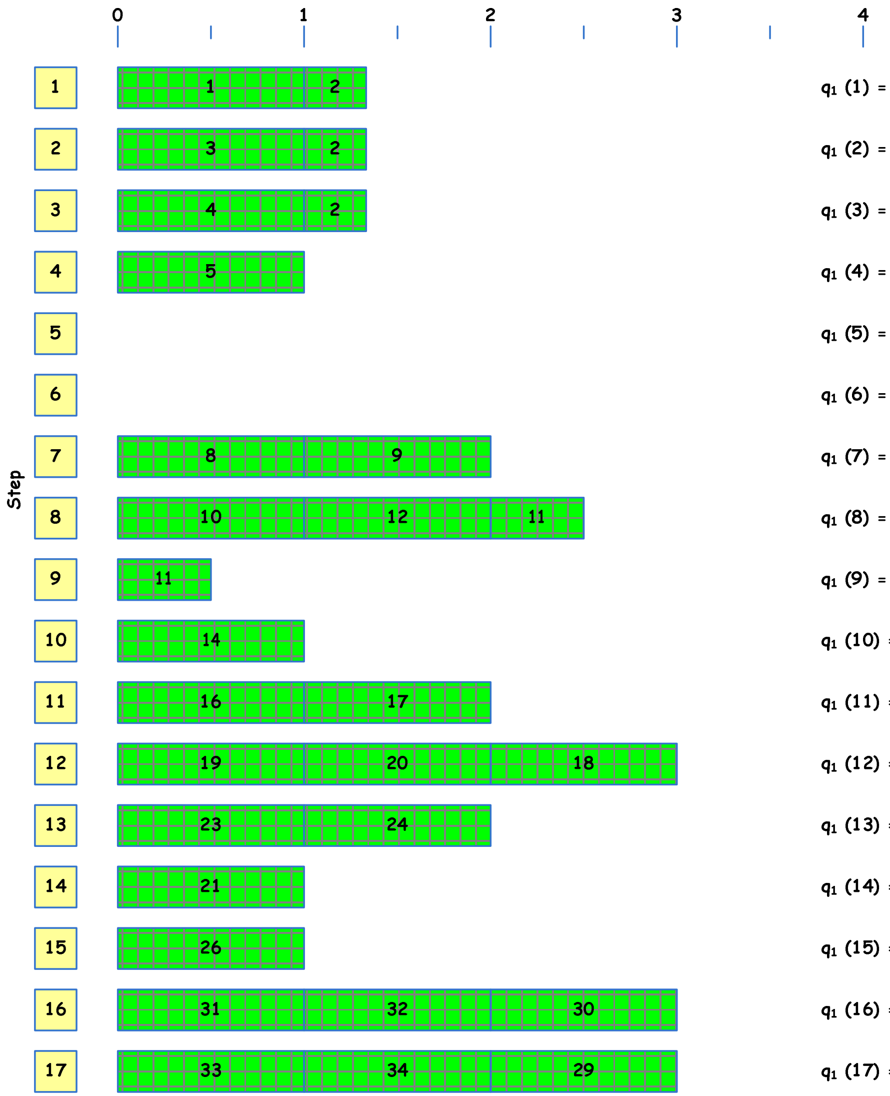


i,l	SF	IF	F
2, 1	0,00	0,00	0,00
2, 2	0,00	0,00	0,00
2, 3	0,00	0,00	0,00

i,l	SF	IF	F
11, 8	0,35	0,00	0,35
11, 9	-1,54	0,21	-1,34
11, 10	-1,54	0,47	-1,07
11, 11	0,46	0,49	0,95
11, 12	1,46	0,32	1,78
11, 13	0,46	0,32	0,77
11, 14	-0,54	0,64	0,10
11, 15	-0,54	1,29	0,75
11, 16	1,46	1,23	2,69

i,l	SF	IF	F
14, 9	-1,65	0,35	-1,31
14, 10	-1,65	-0,60	-2,25
14, 11	0,35	-0,91	-0,57
14, 12	1,35	-0,57	0,77
14, 13	0,35	-0,17	0,18
14, 14	-0,65	-0.06	-0,72
14, 15	-0,65	-0,13	-0,78
14, 16	1,35	-018	1,16
14, 17	1,23	0,00	1,23

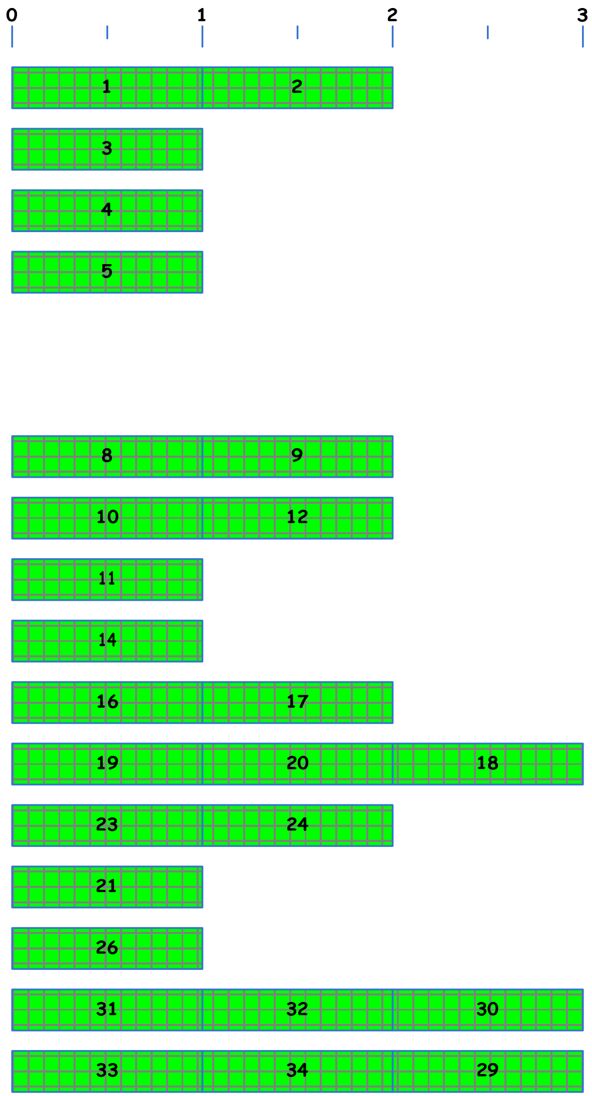
Algoritmo FDS (4ª iterazione)



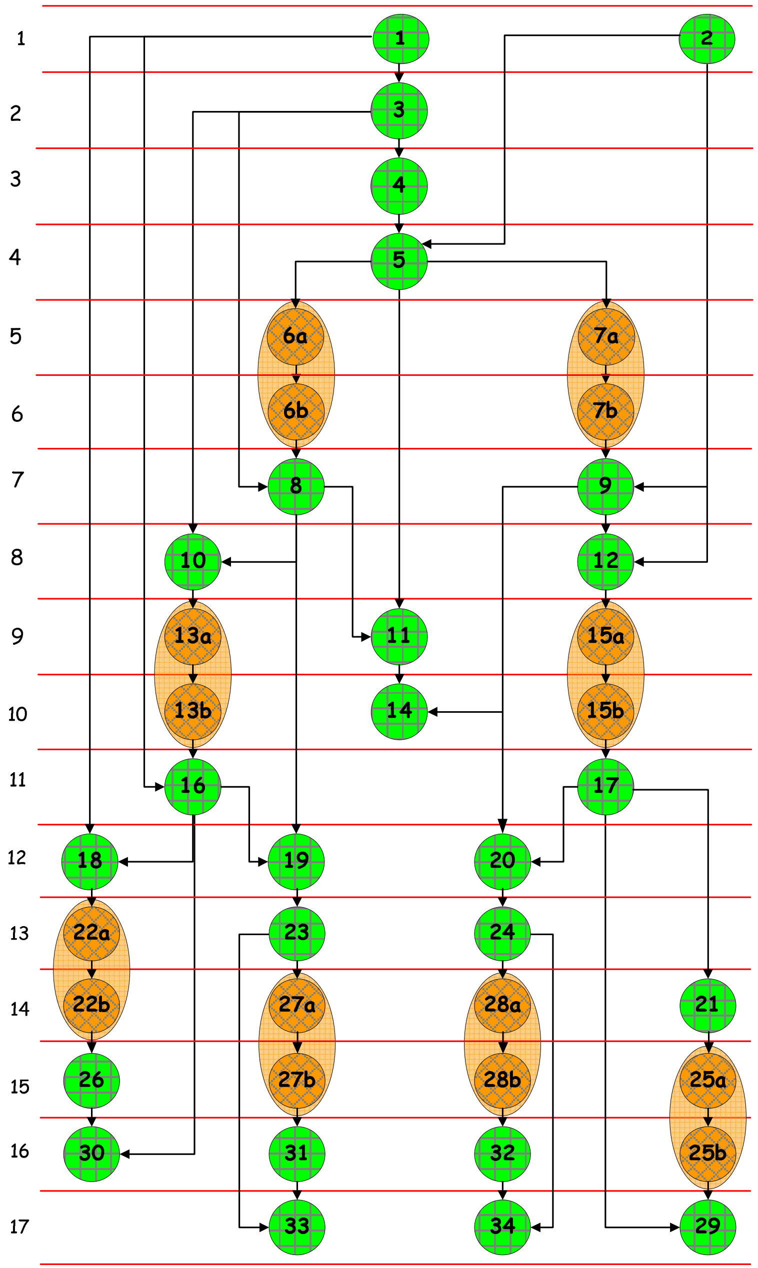
$q_1(1) = 1{,}33$
$q_1(2) = 1{,}33$
$q_1(3) = 1{,}33$
$q_1(4) = 1$
$q_1(5) = 0$
$q_1(6) = 0$
$q_1(7) = 2$
$q_1(8) = 2{,}5$
$q_1(9) = 0{,}5$
$q_1(10) = 1$
$q_1(11) = 2$
$q_1(12) = 3$
$q_1(13) = 2$
$q_1(14) = 1$
$q_1(15) = 1$
$q_1(16) = 3$
$q_1(17) = 3$

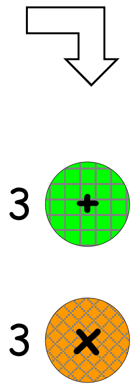
i,l	SF	IF	F
2, 1	0	0	0
2, 2	0	0	0
2, 3	0	0	0

i,l	SF	IF	F
11, 8	1	0	1
11, 9	-1	0	-1



Algoritmo FDS: SSG





VSLS vs ILS vs FDS: valutazione delle prestazioni

con riferimento al 5th order elliptic filter:

		VSLS	ILS	FDS
➢ risorse necessarie:	addizionatori	4	3	3
	moltiplicatori	4	3	3
➢ complessità computazionale:		$O(\lambda)$	$O(n*\lambda)$	$O(n^3)$

con riferimento ad una suite di benchmark disponibili in letteratura: