Per esporre il più importante algoritmo per lo scheduling in sistemi ad instradamento differenziato occorre considerare un'estensione del min L_{max}

Ricerca per ispezione guidata (Branch and Bound) per min L_{max} con $r_j \geq 0$

La ricerca per ispezione riguarda l'albero di tutte le sequenze possibili

BOUND:

Massimo anticipo con interruzione

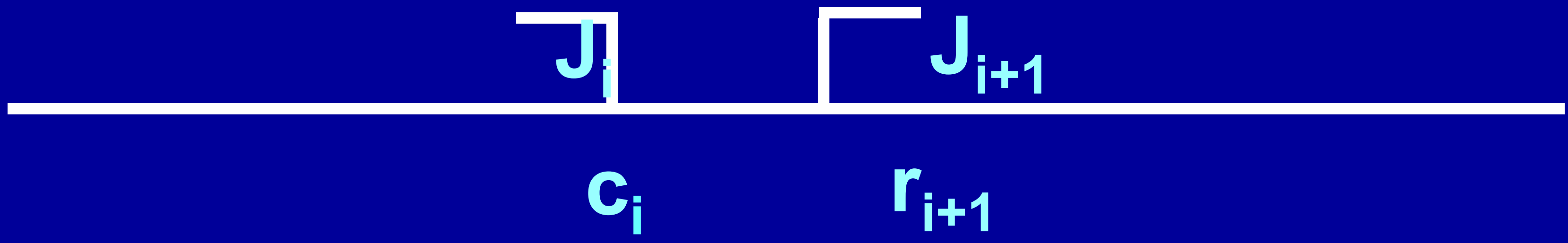
Su una macchina la EDD dà

Min L_{max}

Se i lavori non si possono interrompere e sono disponibili dall'inizio (r_i =0)

Con lavori non disponibili dall'inizio ($r_i \geq 0$), la semplice sequenza EDD ($d_i \leq d_{i+1}$) può avere dei vuoti (nella EDD, $\exists\ c_i < r_{i+1}$)

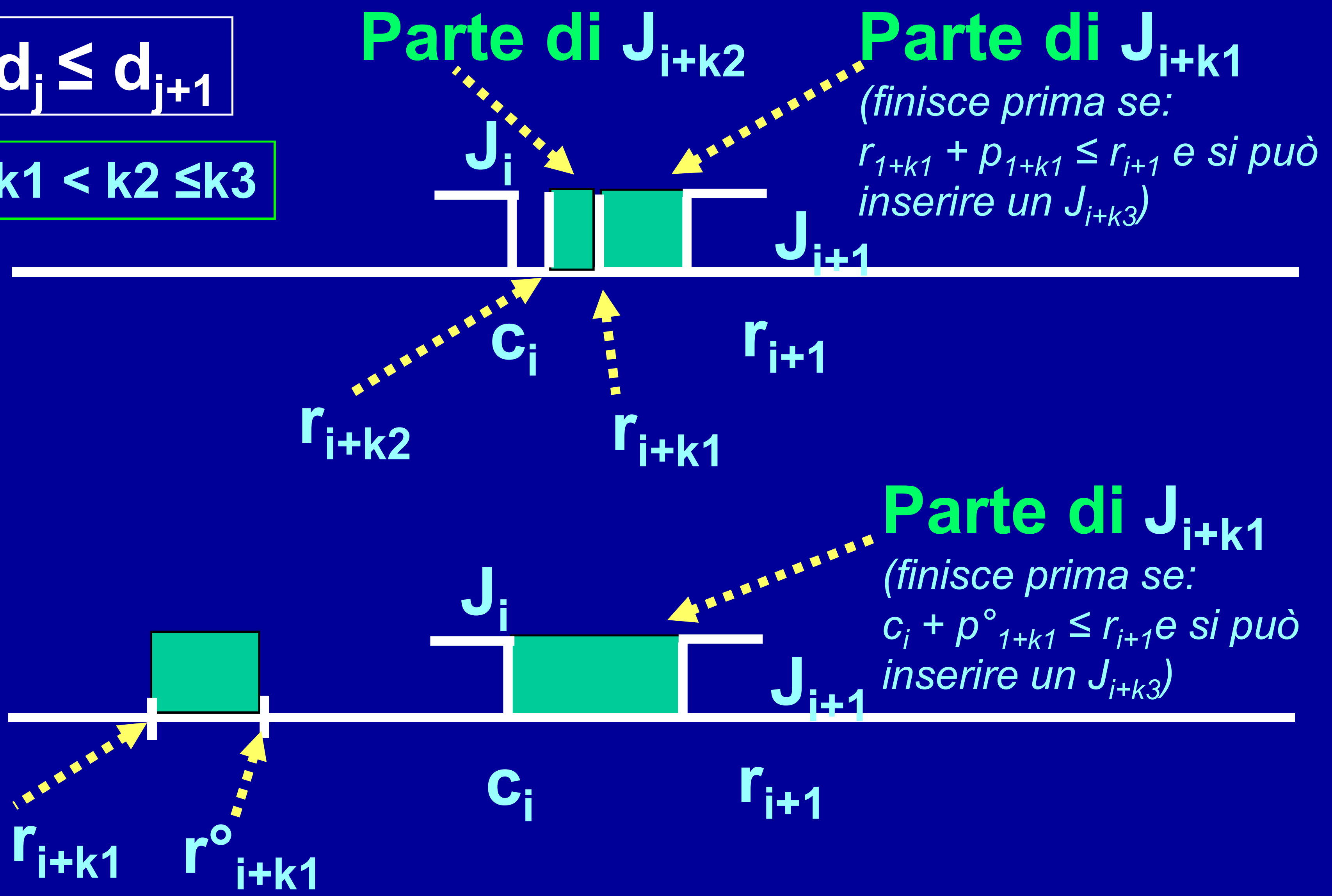
ATT.: se esiste un ordinamento dei rilasci uguale a quello dei tempi dovuti, tale ordinamento (è una EDD) dà la sequenza ottima senza interruzioni



Si tenta di riempire il vuoto anticipando frazioni dei lavori che seguono nella EDD, finchè ci sono. Continuando così il <u>sequezionamento</u> è <u>ottimo</u> (consentendo le interruzioni). =>

$d_j \leq d_{j+1}$

$k1 < k2 \leq k3$



$r°_{i+k1}$ e $p°_{i+k1} = r_{i+k1} + p_{i+k1} - r°_{i+k1}$ sono i tempi aggiornati se J_{i+k1} è **già iniziato in un buco precedente**

Una sequenza EDD con interruzione (e quindi riempimento di vuoti) dà un limite inferiore (Bound) per l'ottimo possibile senza interruzione.

Massimo anticipo con interruzione

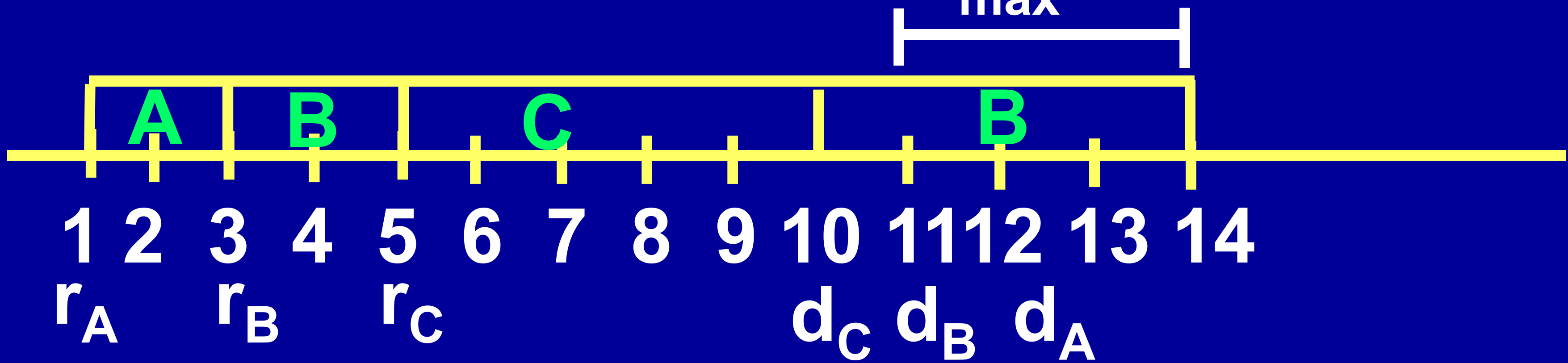
Esempio 1

lavori	A	B	C
p_j tempi proc.	2	6	5
r_j tempi ril.	1	3	5
d_j tempi dov.	12	11	10

EDD : C B A

Ottimo senza int.:
A B C: $L_{max} = 4$

Bound: $L_{max} = 3$



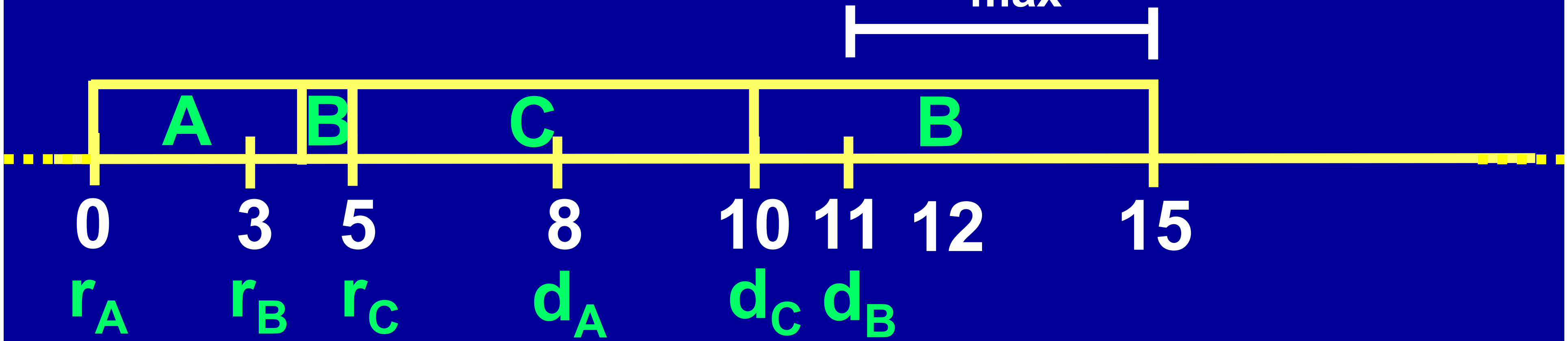
Esempio 2

	A	B	C
p_j	4	6	5
r_j	0	3	5
d_j	8	11	10

EDD : A C B

ottimi senza int.:
A B C; A C B: L_{max}= 5

Bound: L_{max} = 4



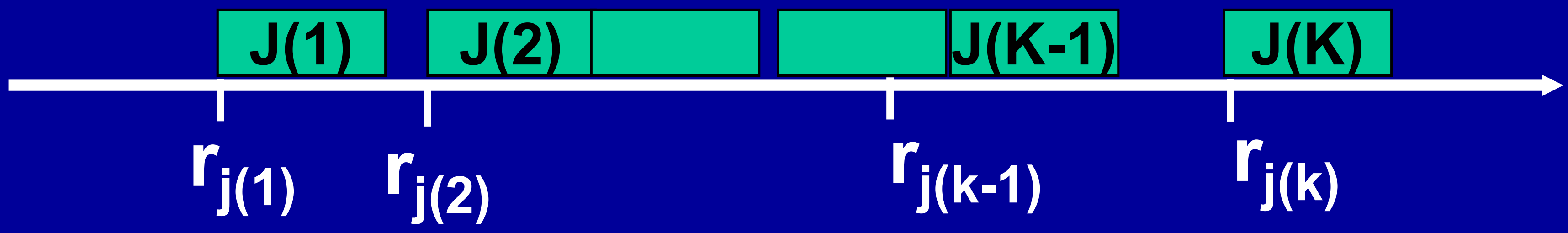
CUT: taglio di un ramo

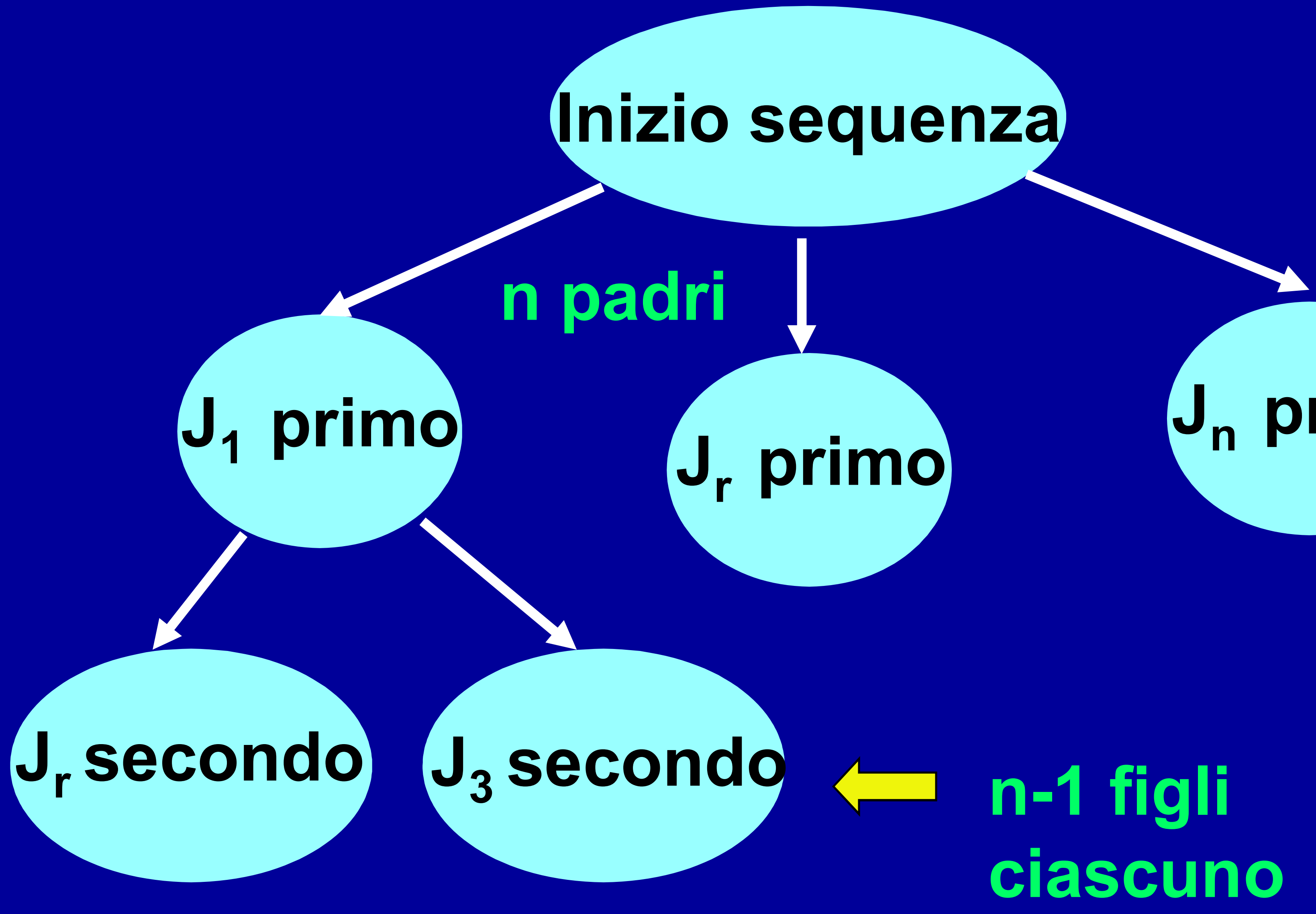
Al livello k-1, sono stati già assegnati k-1 jobs (J(1),J(2),…,J(k-1)), per assegnare il k_esimo e' necessario che questo soddisfi la seguente disuguaglianza

$$r_{j(k)} < \min_{l \in S_k} \left(\max \left(c_{j(k-1)} \,;\, r_l \right) + p_l \right) \qquad c_{j(0)} := 0$$

S_k insieme dei jobs da assegnare <u>dopo</u> J(k)

Se esiste un l per cui non è soddisfatta, c'è spazio per J_l che va inserito, quindi <u>il ramo con radice J(k) va tagliato</u>





Per ogni ramo ci sono un Bound (si sviluppa quello per cui è più basso) o un Cut (non si sviluppa)

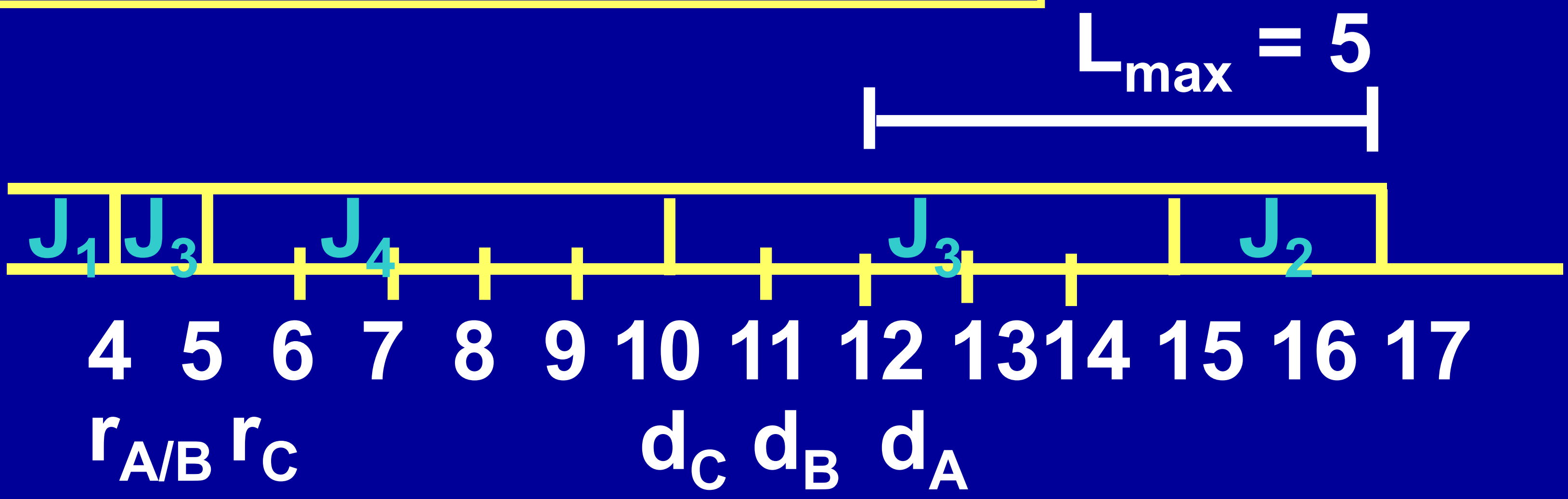
Esempio

lavori	1	2	3	4
p_j tempi proc.	4	2	6	5
r_j tempi rilascio	0	1	3	5
d_j scadenza	8	12	11	10

lavori	J_2	J_3	J_4
p_j tempi proc.	2	6	5
r_j* tempi ril. agg.	4	4	5
d_j scadenza	12	11	10

EDD : J_4 J_3 J_2

*max (c_1, r_j)

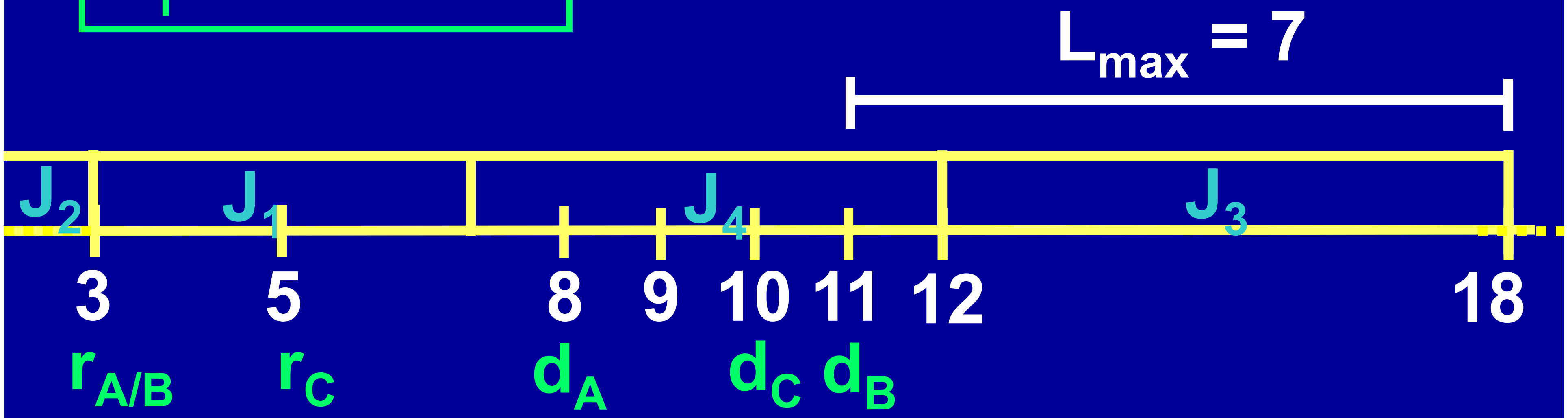


Bound con inizio J_1: 5

	J_1	J_3	J_4
p_j	4	6	5
r_j*	3	3	5
d_j	8	11	10

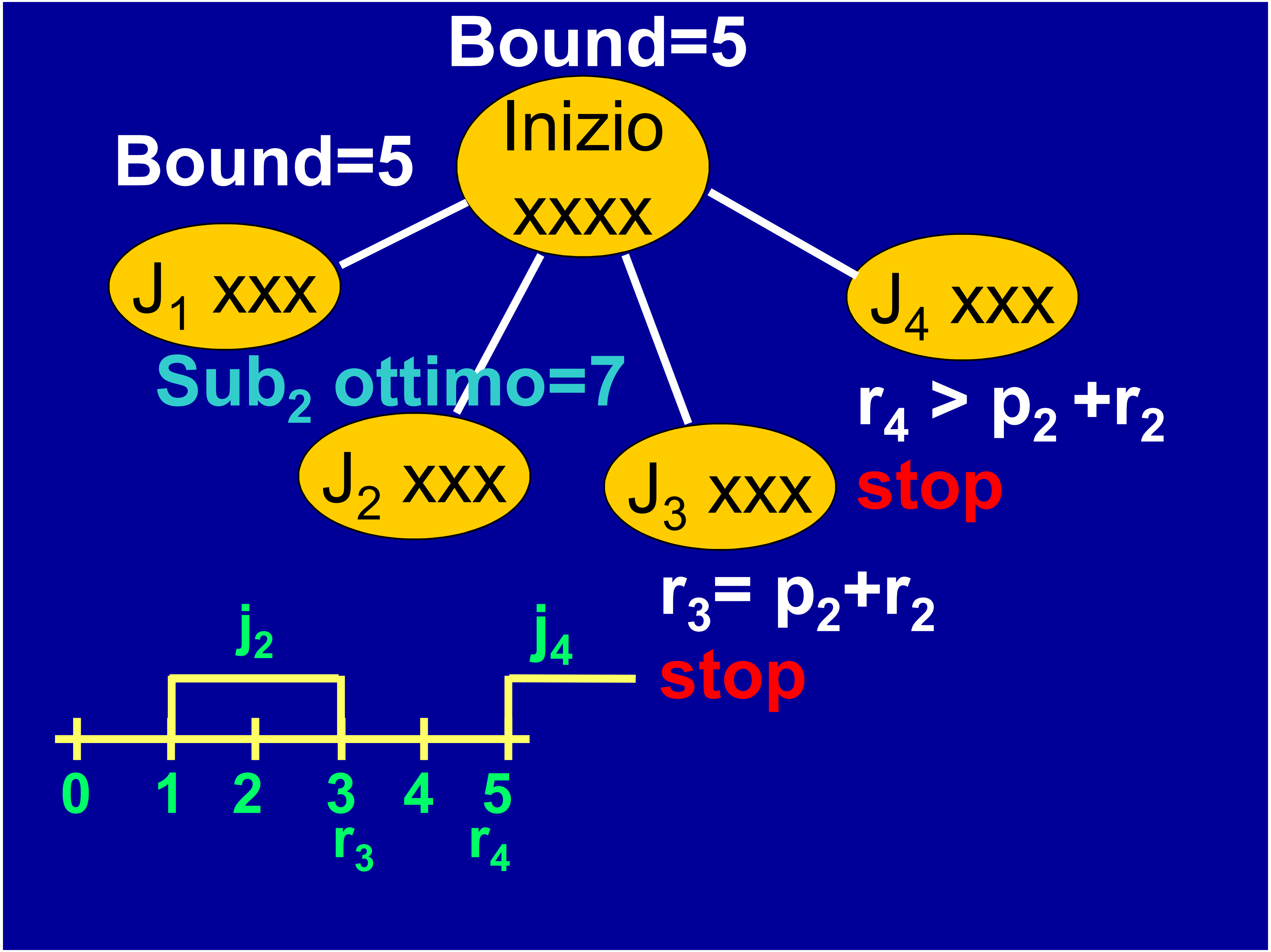
EDD : J_1 J_4 J_3

*max (c_1, r_j)



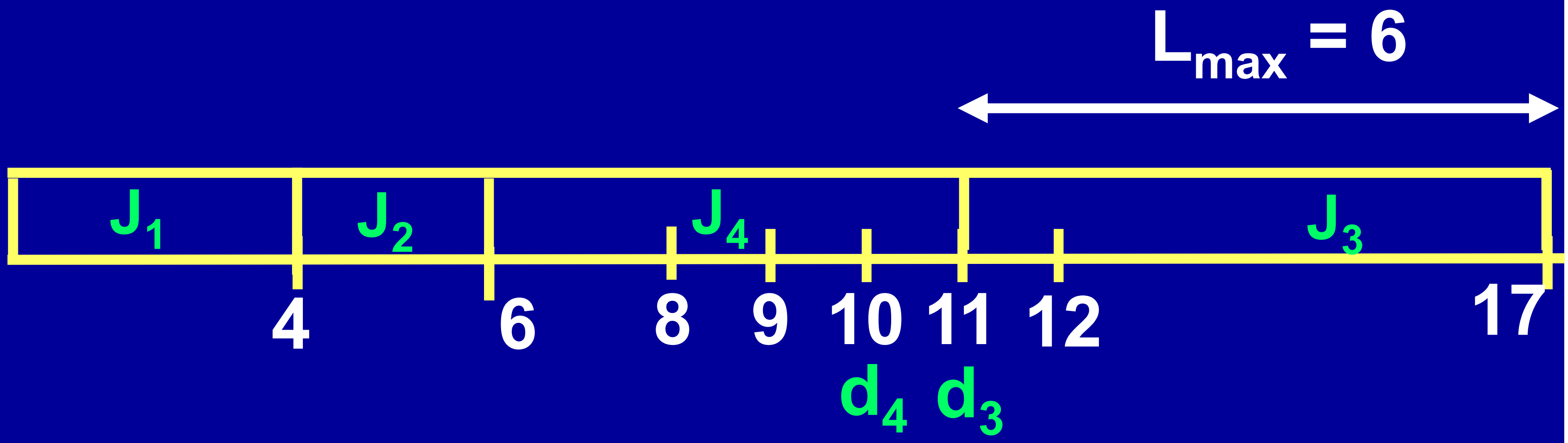
Bound con inizio J_2: 7 att.:OTTIMO

per J_1 J_4 J_3


Bound=5
Inizio
xxxx
Bound=5
J_1 xxx
Sub$_2$ ottimo=7
J_2 xxx
J_3 xxx
J_4 xxx
$r_4 > p_2 + r_2$
stop
$r_3 = p_2 + r_2$
stop
j_2
j_4
0 1 2 3 4 5
r_3
r_4

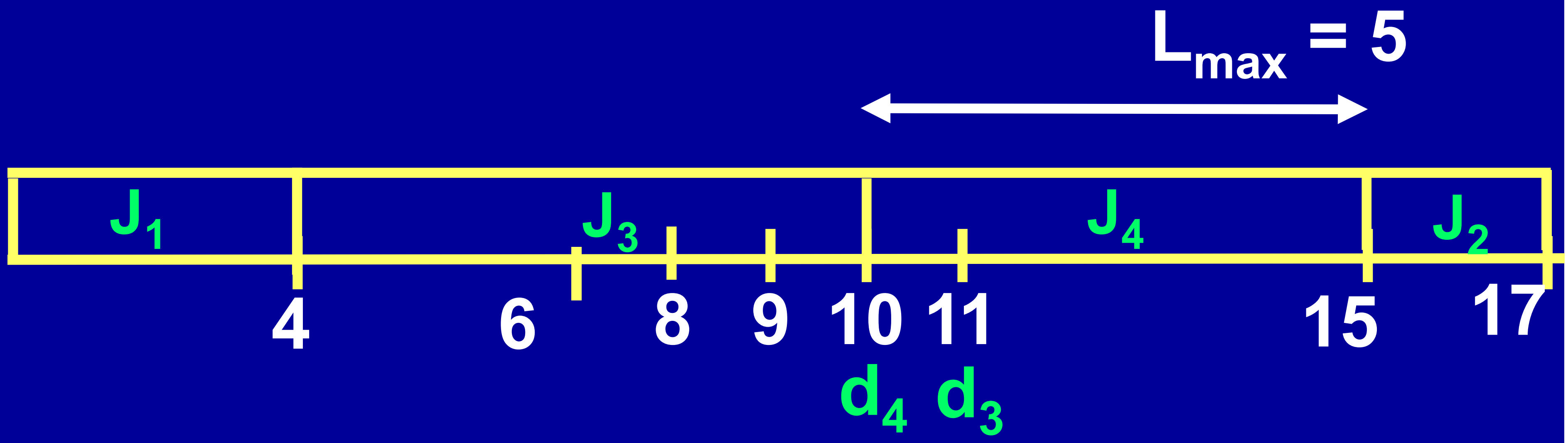
SOL 12

lavori	1	2	3	4
p_j tempi proc.	4	2	6	5
r_j tempi rilascio	0	1	3	5
d_j scadenza	8	12	11	10



SOL 13

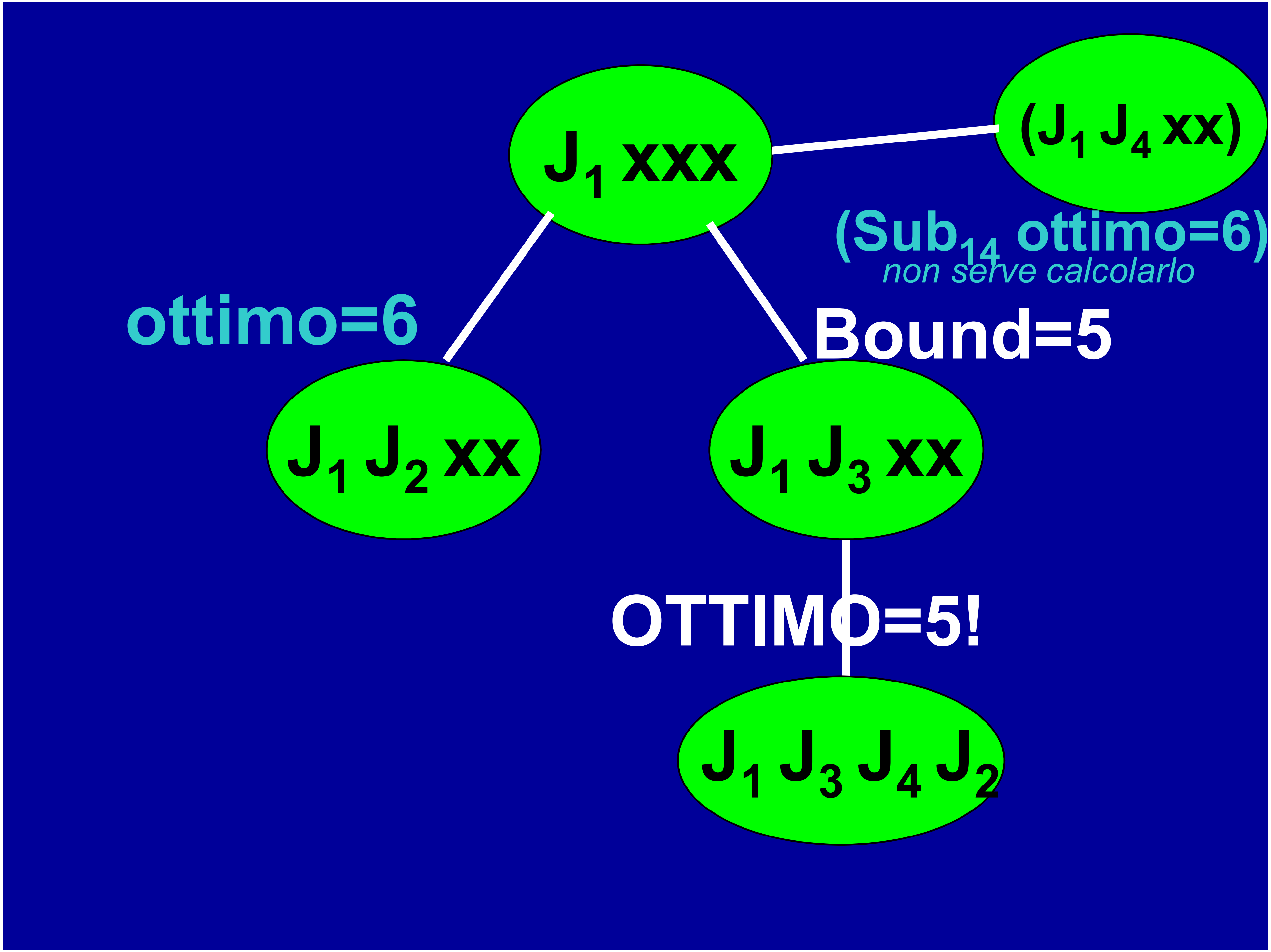
lavori	1	2	3	4
p_j tempi proc.	4	2	6	5
r_j tempi rilascio	0	1	3	5
d_j scadenza	8	12	11	10



SOL OTTIMA

OTTIMO=5!: ci si ferma quindi a J_1J_3xx

La sequenza J_1 J_3 J_4 J_2 è piena (senza attesa della macchina) e senza interruzioni, con L_{max} pari al Bound del nodo da cui discende, e minore di quello di tutti gli altri nodi non sviluppati, quindi è ottima.



J_1 xxx
$(J_1 J_4$ xx)
(Sub_{14} ottimo=6)
non serve calcolarlo
ottimo=6
Bound=5
$J_1 J_2$ xx
$J_1 J_3$ xx
OTTIMO=5!
$J_1 J_3 J_4 J_2$

Sequenziamento nei sistemi integrati con istradamento differenziato

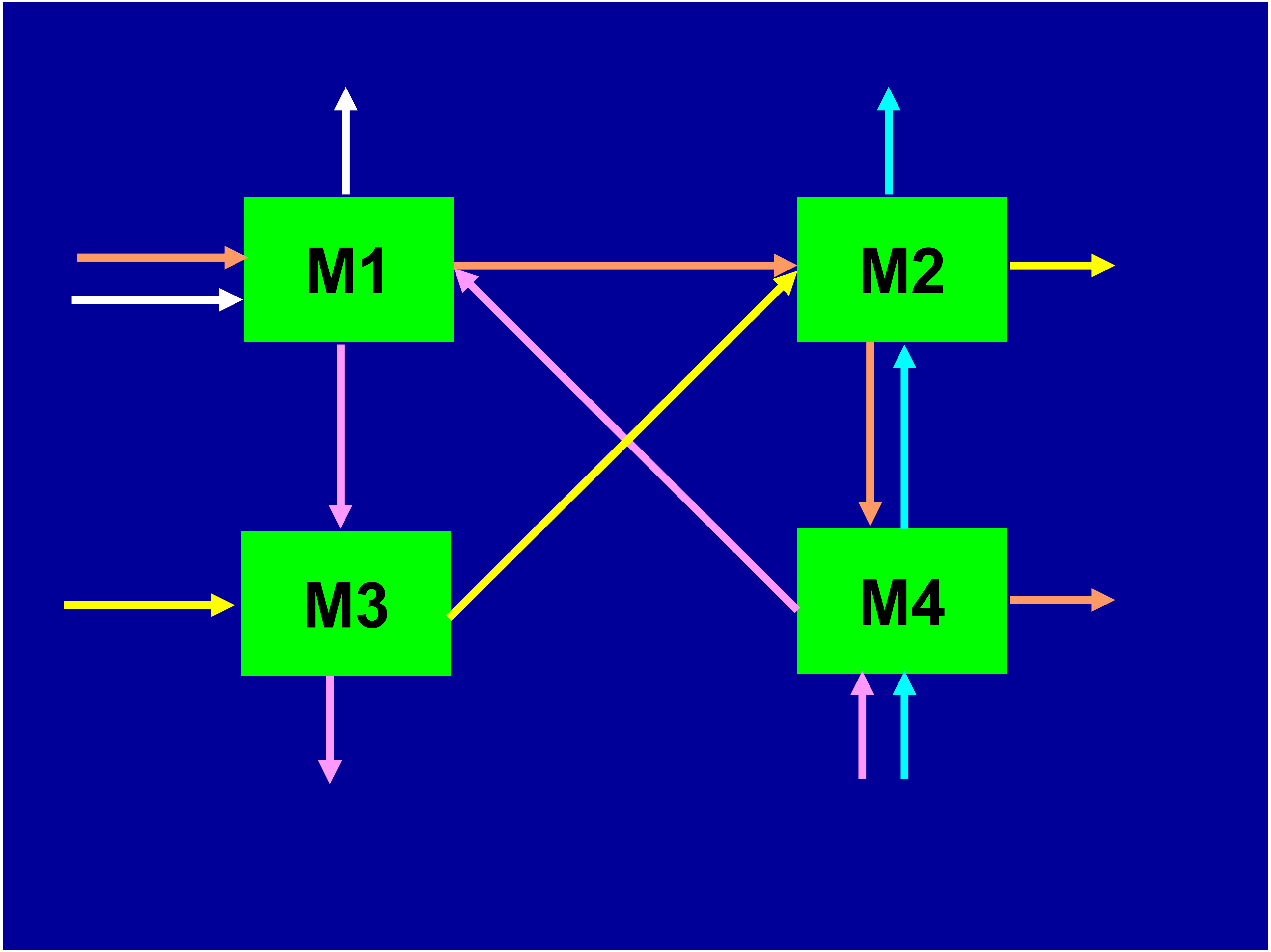
Per il JOB SHOP:

A. Agnetis_scheduling, Dispense ad uso esclusivo degli studenti, disponibili nella cartella 08_09 AU 9 crediti, nella pagina web: http://nicolo.dia.uniroma3.it/

M. Pinedo: "SCHEDULING, Theory, Algorithms, and Systems", Prentice Hall.1995

Per la scelta dell'istradamento:

A. Agnetis: "MODELLI COMBINATORI NELLA PRODUZIONE FLESSIBILE", Dispense ad uso esclusivo degli studenti, disponibili nella cartella 08_09 AU 9 crediti, nella pagina web: http://nicolo.dia.uniroma3.it/

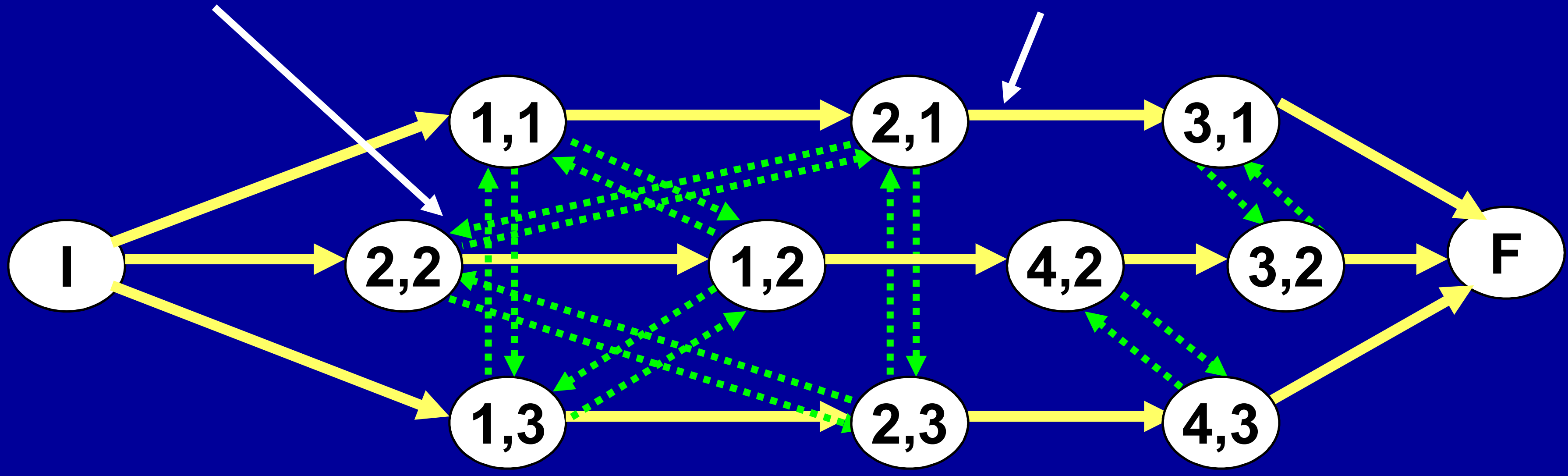

M1
M2
M3
M4

JOB SHOP SENZA RICIRCOLAZIONE

Grafo per $n/m/G/c_{max}$ $r_i=0$

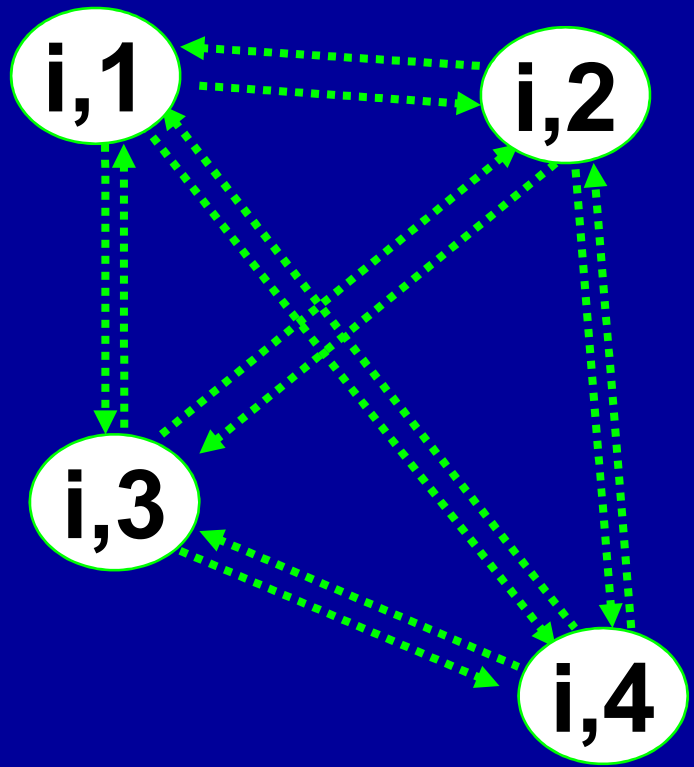
Archi *(precedenze)* disgiuntivi *(macch.)*

Arco *(precedenza)* congiuntivo *(pezzi)*

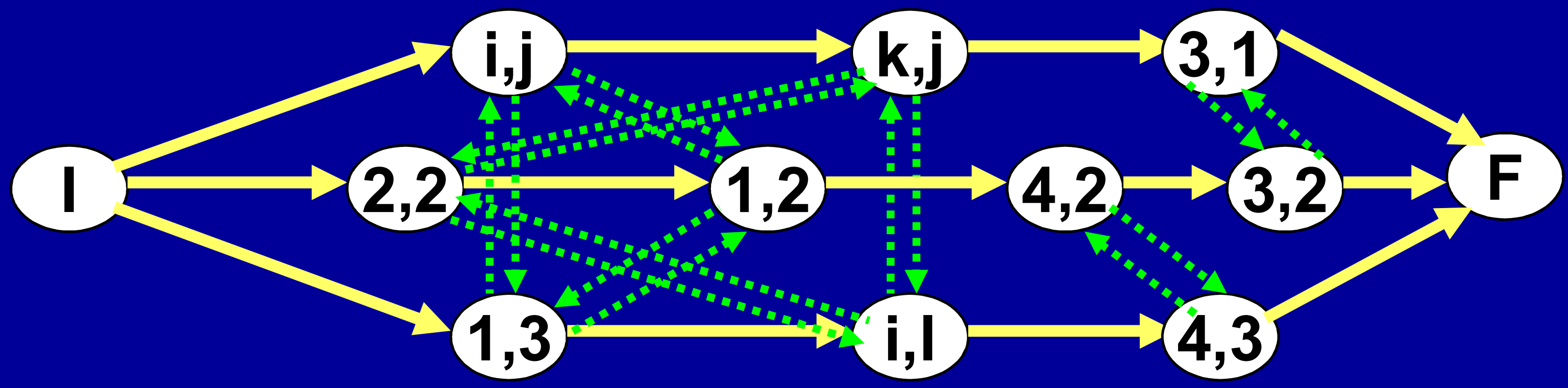


Operazione(i,j): macchina i, lavoro j

Fra i nodi relativi ad una stessa macchina gli <u>archi disgiuntivi</u> sono tutti i possibili



scelta degli archi disgiuntivi che minimizza il cammino critico



Min C_{max}

$y_{ij} - y_{il} \geq p_{il}$ **oppure** $y_{il} - y_{ij} \geq p_{ij}$

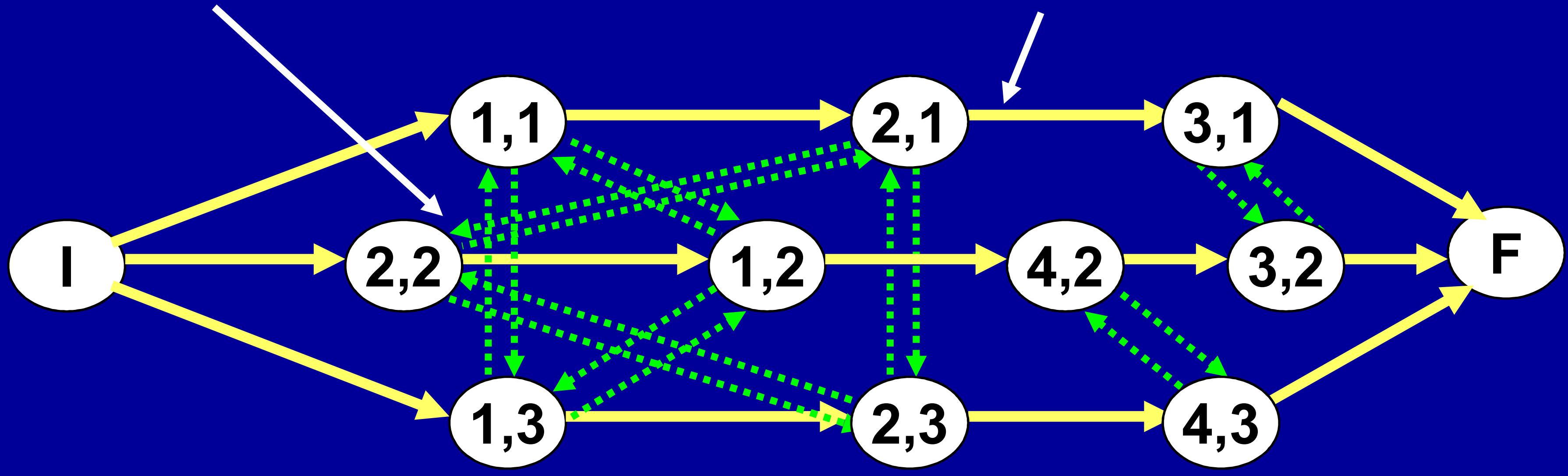
Vincoli disgiuntivi → **problema "difficile"**

JOB SHOP SENZA RICIRCOLAZIONE

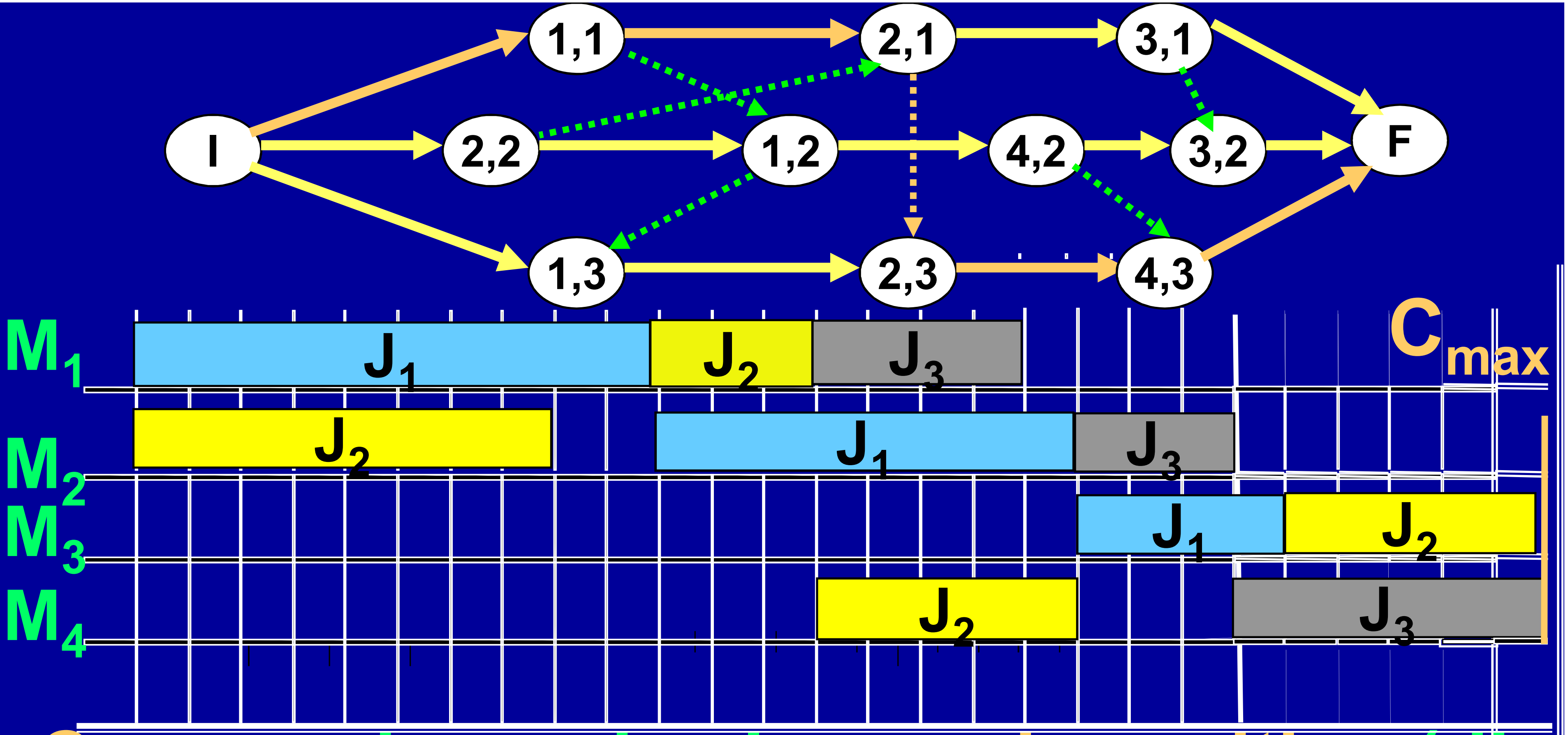
Grafo per $n/m/G/c_{max}$ $r_i=0$

Archi ***(precedenze)*** **disgiuntivi** ***(macch.)***

Arco ***(precedenza)*** **congiuntivo** ***(pezzi)***

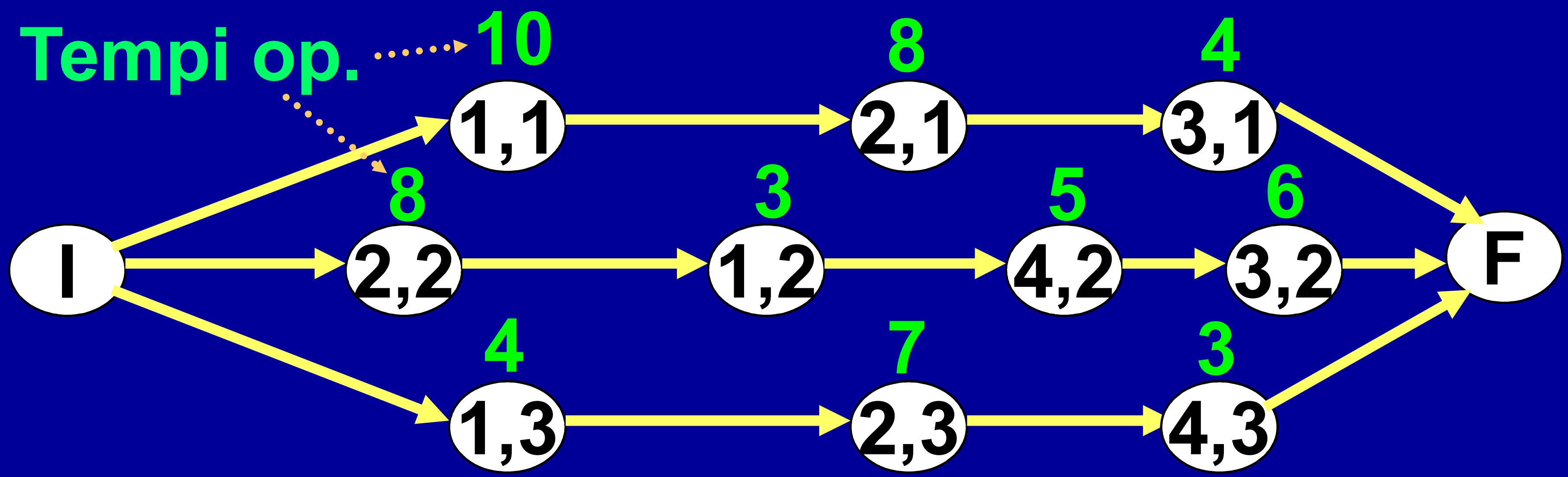


Operazione(i,j): macchina i, lavoro j

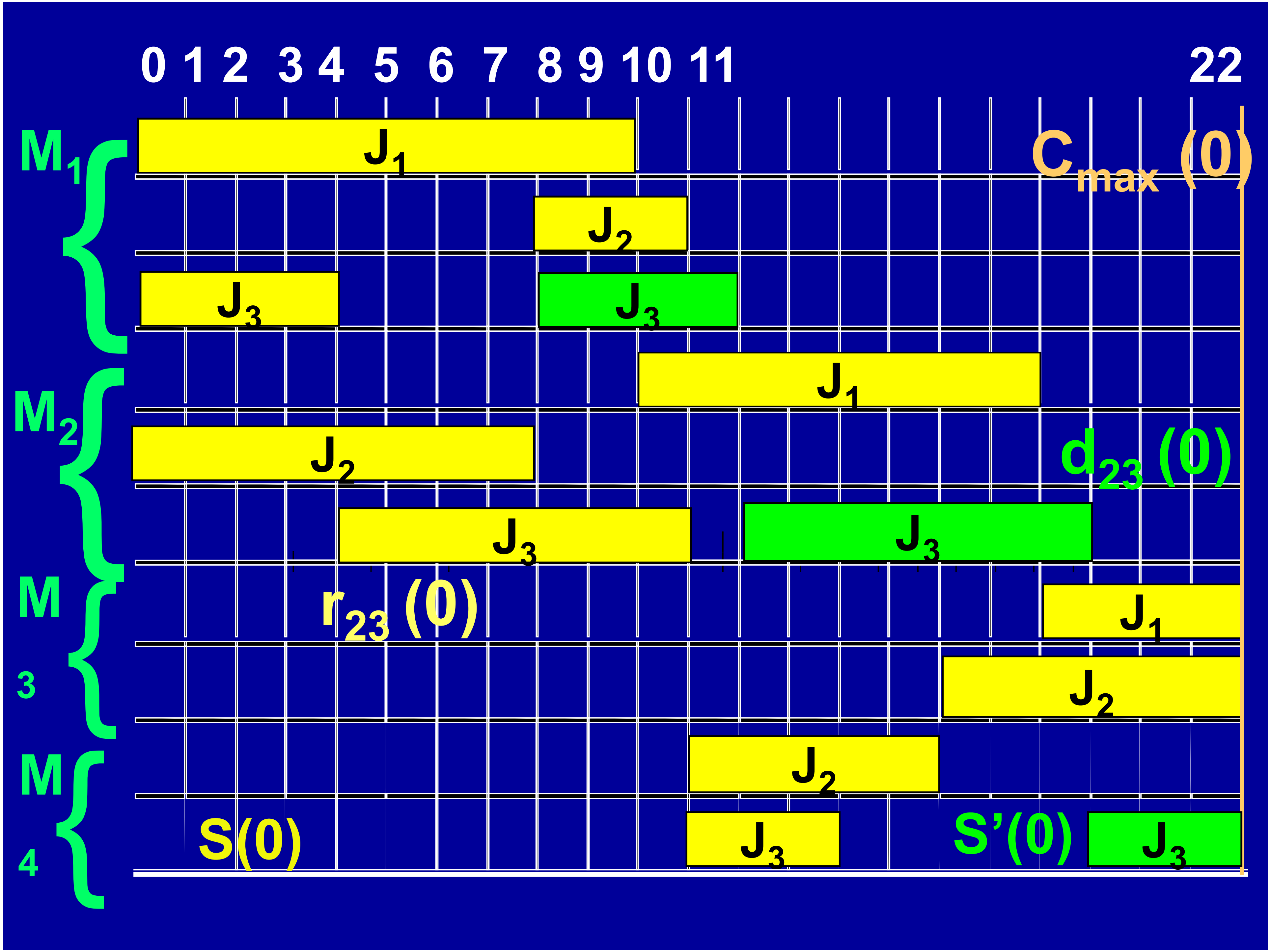


C_{max} corrisponde al cammino critico (di peso massimo) <u>dopo</u> che si è scelto uno solo dei due archi disgiuntivi tra ciascuna coppia di nodi operazione sulla stessa macchina

Euristica per n/m/G/C_{max} (Shifting Bottleneck)



Iterazione zero: solo archi congiuntivi
S(0): prima possibile con tutte le macchine multiplate



0 1 2 3 4 5 6 7 8 9 10 11 22
M1
M2
M3
M4
J1
J2
J3
J1
J2
J3
J3
J1
J2
J3
J2
J3
J3
Cmax (0)
d23 (0)
r23 (0)
S(0)
S'(0)

C_{max} (0) è dato dal massimo tempo di processamento totale in S (0)

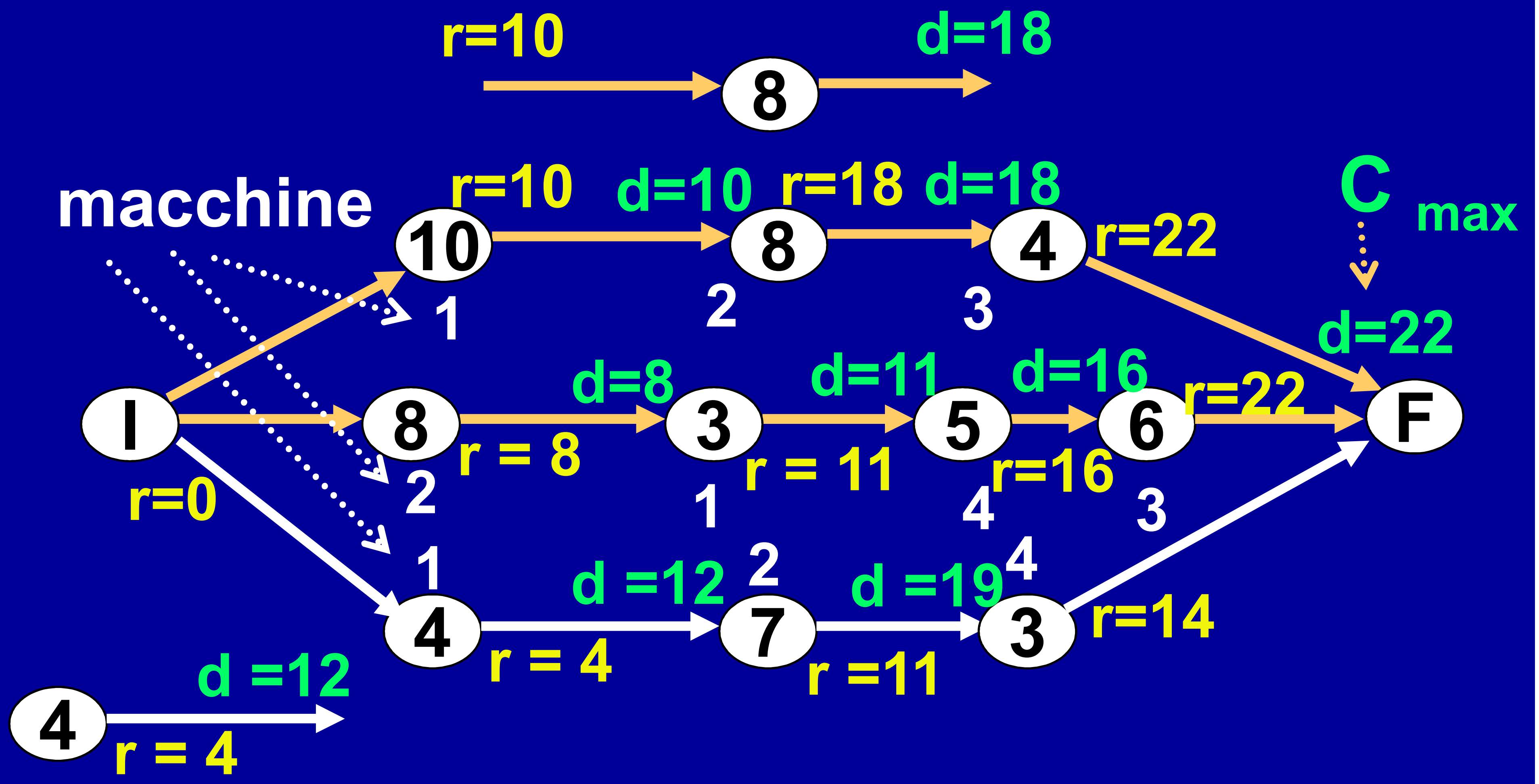
S' (0): ottenuta da S (0) ritardando il più possibile i lavori senza aumentare C_{max}

r_{ij} (0) inizio operazione su **i** del lavoro **j** in **S (0)**
è il tempo in cui è disponibile, il prima possibile, l'op. **ij**

d_{ij} (0) fine operazione su **i** del lavoro **j** in **S' (0)**
è il tempo in cui deve finire, al più tardi, l'op. **ij**

I valori di r_{ij} e d_{ij} possono essere calcolati sul grafo di precedenza
(prossima schermata)

r è il rilascio dell'operazione, indicato sulla coda dell'arco entrante
d è il tempo dovuto per l'operazione, indicato sulla punta dell'arco uscente



Agli estremi dell'arco uscente sono i limiti entro cui può variare la fine dell'oper.

SU UN CAMMINO CRITICO TALI LIMITI COINCIDONO: non c'è margine:
per ogni operazione il dovuto coincide con il rilascio dell'operazione che segue

L’algoritmo prosegue con il compattamento di una macchina ad ogni iterazione successiva

L’idea è di compattare la macchina più critica, dopo aver minimizzato, per ciascuna di esse, il massimo ritardo con i tempi dovuti e con i rilasci di cui all’iterazione precedente.

Per proseguire occorre disporre di un algoritmo per minimizzare il massimo ritardo su una macchina con rilasci differenziati delle singole operazioni (se i rilasci fossero tutti uguali, basterebbe una semplice sequenza EDD, cioè ordinata come i tempi dovuti: si ricordi che tale ordine è unico se e solo se i tempi dovuti sono tutti diversi fra loro)

Nell'esempio in corso le soluzioni di min [$L_{imax} := \max_j(c_{ij} - d_{ij})$] alla prima iterazione, con il Branch & Bound danno:

p_j	10	3	4		p_j	8	8	7		p_j	4	6		p_j	5	3
r_{1j}	0	8	0		r_{2j}	10	0	4		r_{3j}	18	16		r_{4j}	11	11
d_{1j}	10	11	12		d_{2j}	18	8	19		d_{3j}	22	22		d_{4j}	16	22

$L_{1max} = 5$ $L_{2max} = 5$ $L_{3max} = 4$ $L_{4max} = 0$

Si considerano critiche (collo di bottiglia) le macchina con il più alto valore L_{max}

Si compatta un collo di bottiglia perché il compattamento varia i tempi di rilascio e dovuti per le altre non compattate e può peggiorarne l'L_{max}, che però è più piccolo o non maggiore.

Nell'esempio, alla prima iterazione ci sono due "colli di bottiglia" (bottleneck)

Sceltone uno (M_1) si sequenziano i lavori secondo il min L_{max} (per M_1: J_1 J_2 J_3)

In generale all'iterazione k c'è un insieme M^k di macchine con sequenziamento assegnato (nell'esempio M^1={ M_1}), le altre restano multiplate

Dopo il compattamento all'iterazione k, chiamato $L_{MAX}(k-1)$ l'L_{max} della macchina compattata, rispetto ai t. dovuti **all'iterazione k-1, si ha:**

$$C_{max}(k) = C_{max}(k-1) + L_{MAX}(k-1)$$ **=> 27 per k=1, nell'esempio**

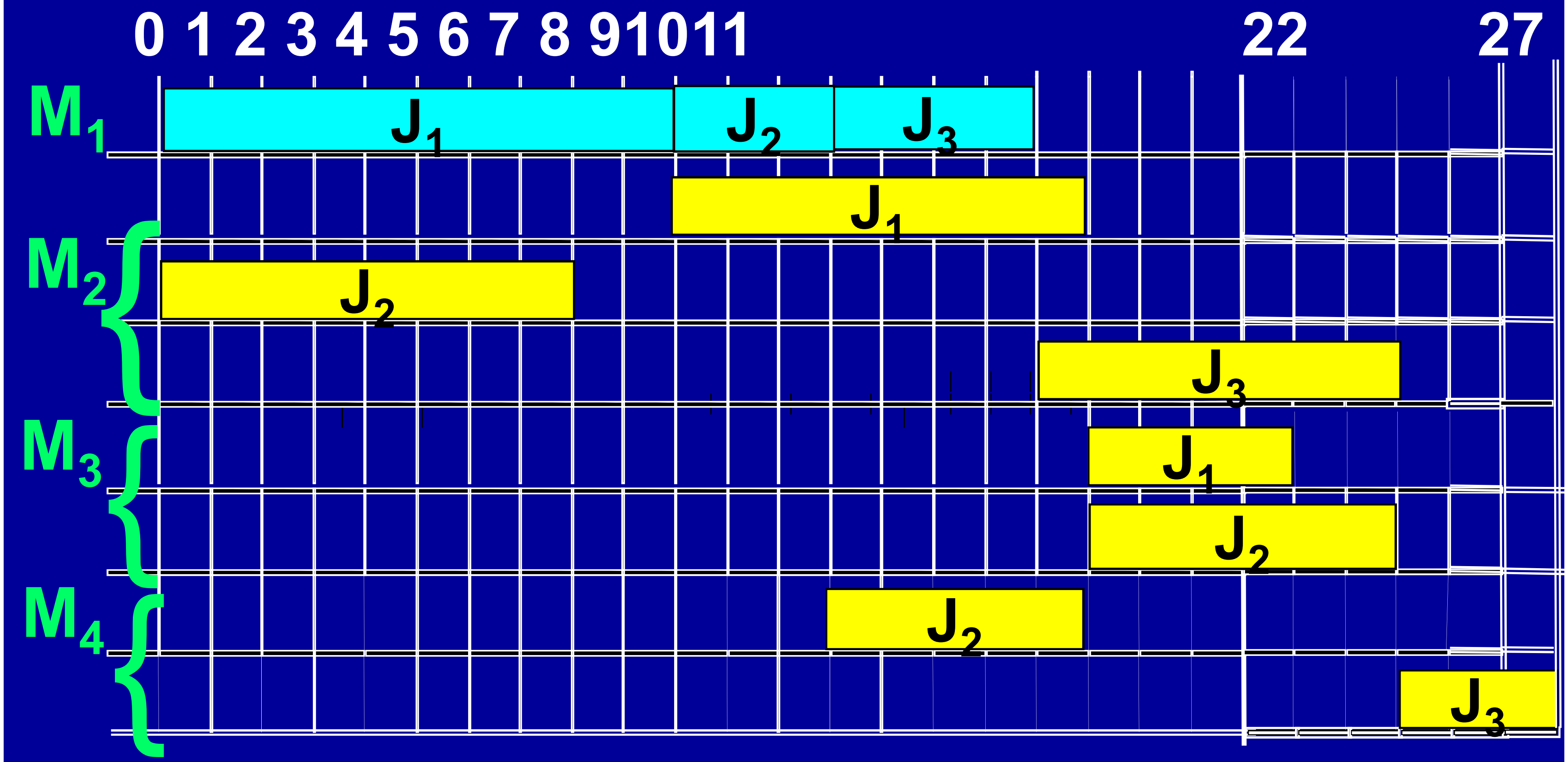
Con le precedenze assegnate sulle macchine di M^k, si calcolano i tempi all'iterazione k:

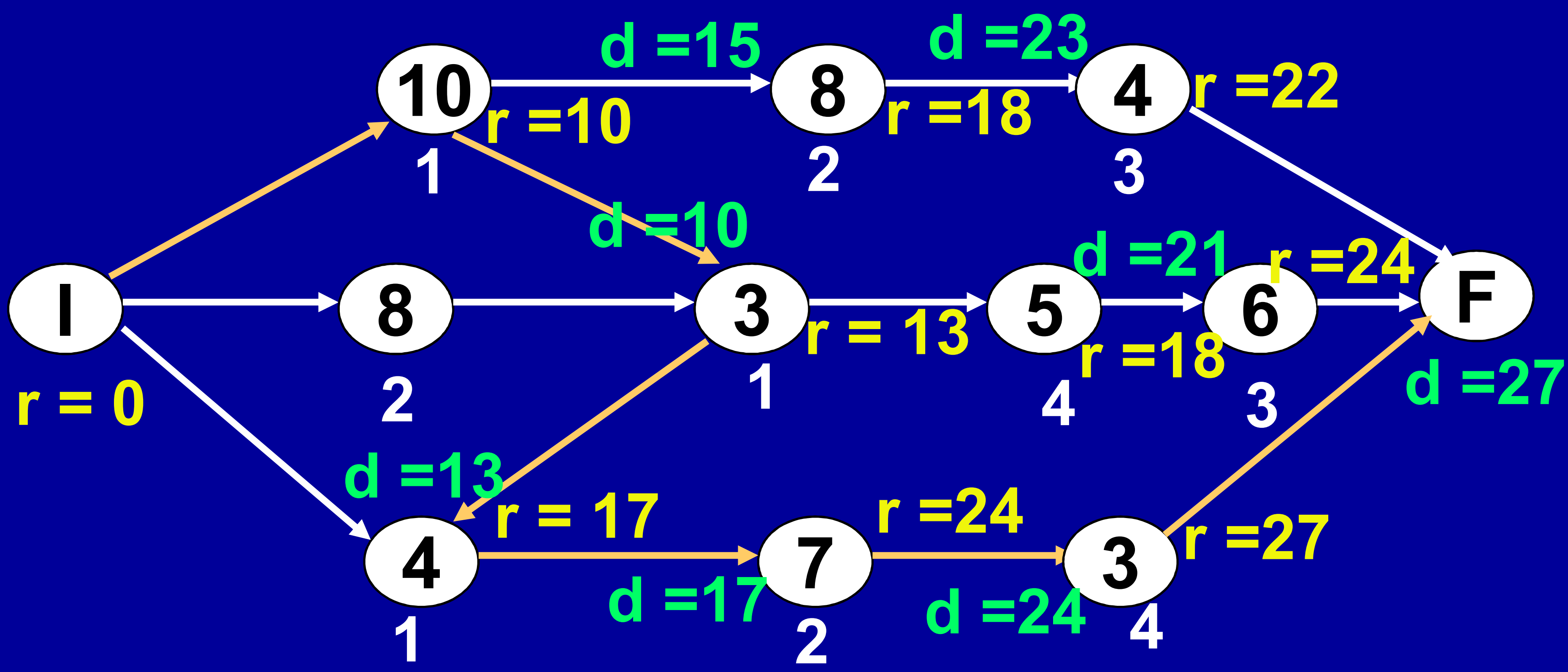
r_{ij} (k) inizio lavori j su i in S(k)

d_{ij} (k) fine lavori j su i in S' (k)

Quindi si calcolano gli L_{max} delle macchine non compattate, si sceglie un nuovo collo di bottiglia e così via fino ad aver compattato tutte le macchine

Sequenziamento S(1)





r va calcolato sul cammino critico a monte
(il più lungo tra l'inizio e il nodo)
d va calcolato sul cammino critico a valle
(il più lungo tra il nodo e la fine)

Att:è uguale il rilascio ai nodi successori, sulle coda degli archi uscenti dallo stesso nodo, come il dovuto dai nodi predecessori sulle punte degli archi entranti nello stesso nodo

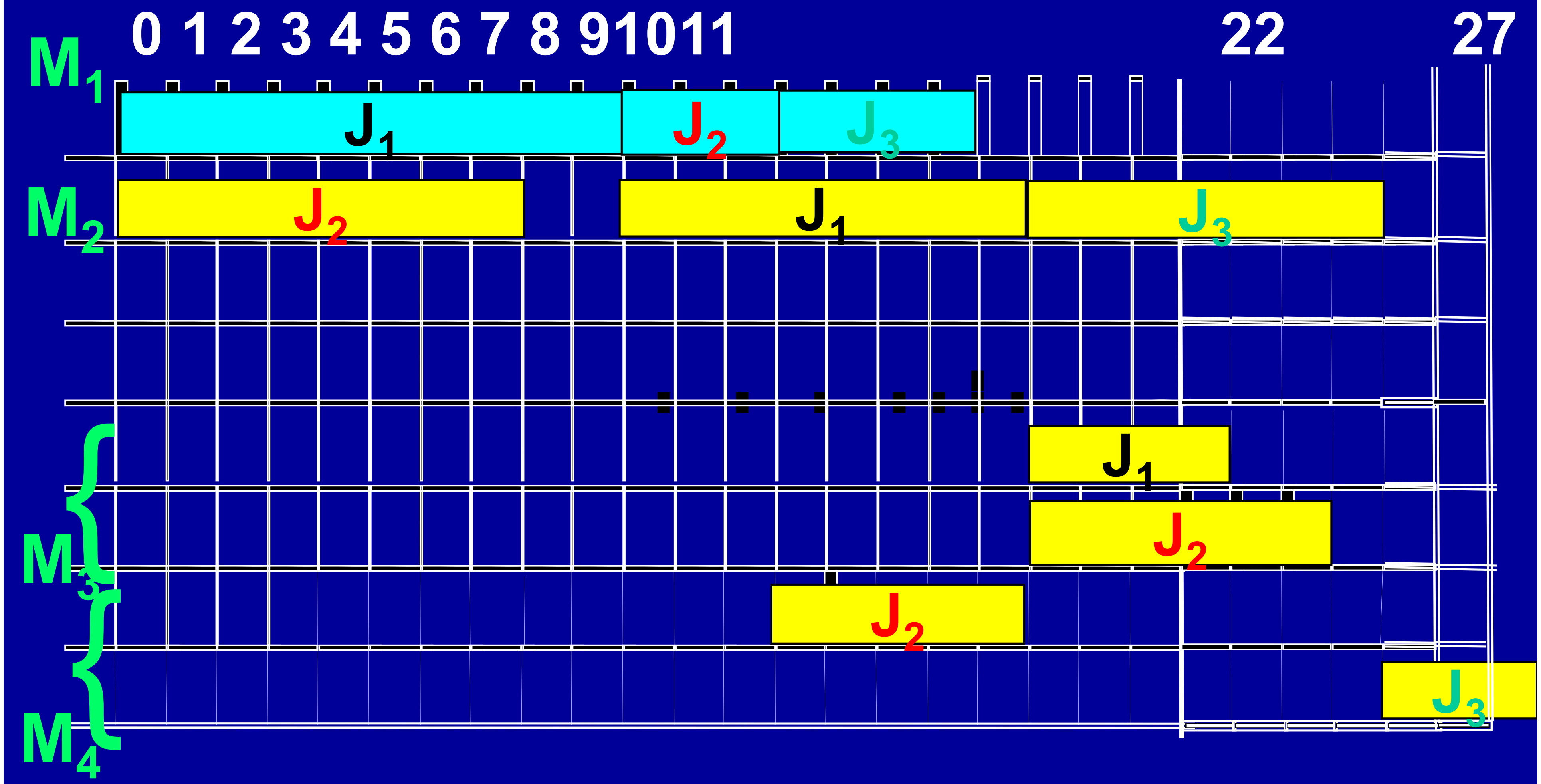
p_j	8	8	7	p_j	4	6		p_j	5	3	
r_{2j}	10	0	17	r_{3j}	18	18		r_{4j}	13	24	
d_{2j}	23	10	24	d_{3j}	27	27		d_{4j}	21	27	

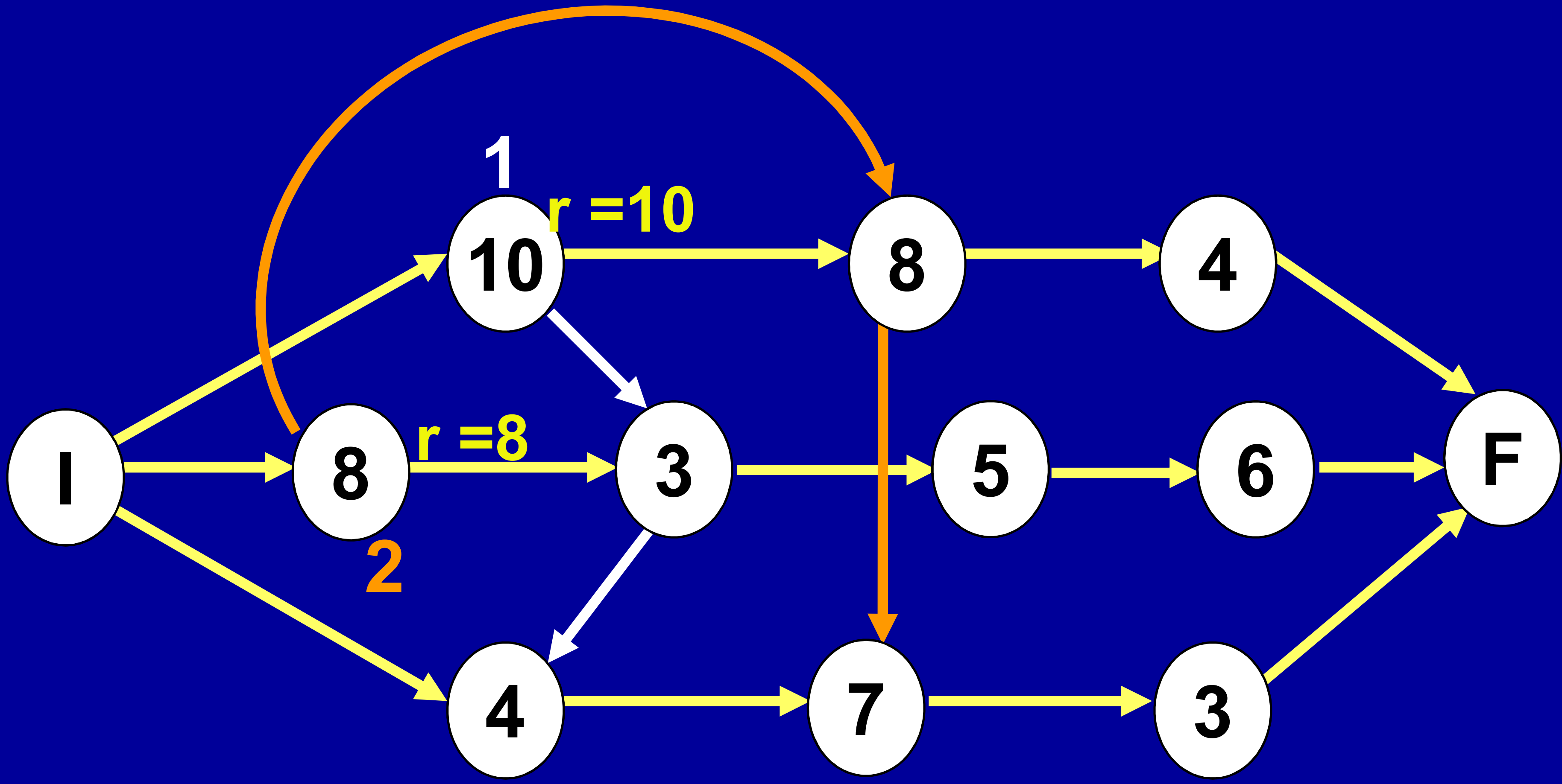
$L_{2max} = 1$ $L_{3max} = 1$ $L_{4max} = 0$

Anche questa volta ci sono due macchine che possono essere inserite in M, scegliamo la 2: $M^2 = \{M_1, M_2\}$

$C_{max}(2) = C_{max}(1) + L_{MAX}(1) = 28$

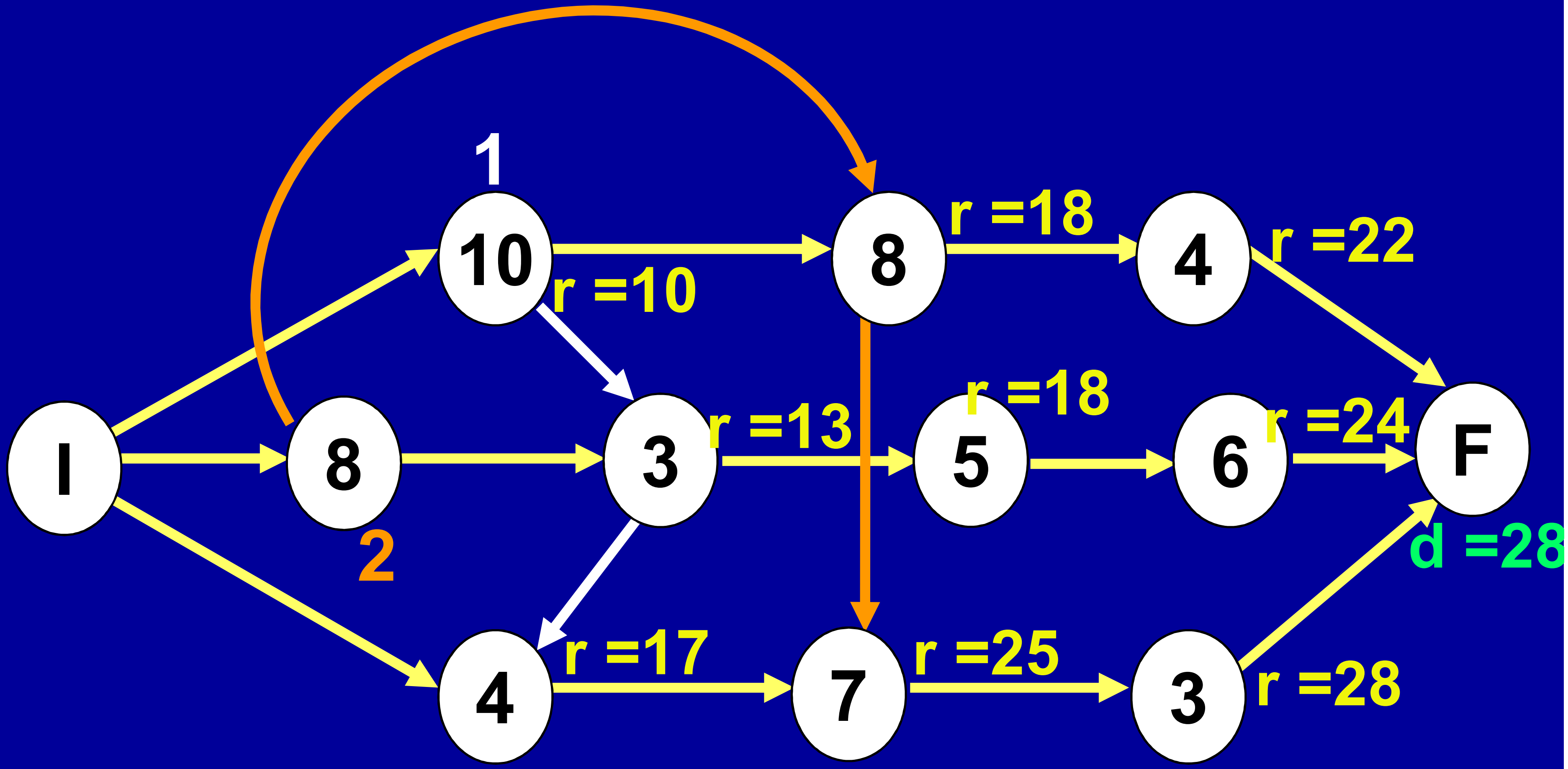
Sequenziamento S(2)





Archi disgiuntivi introdotti al passo 1

Archi disgiuntivi introdotti al passo 2



Archi disgiuntivi introdotti al passo 1

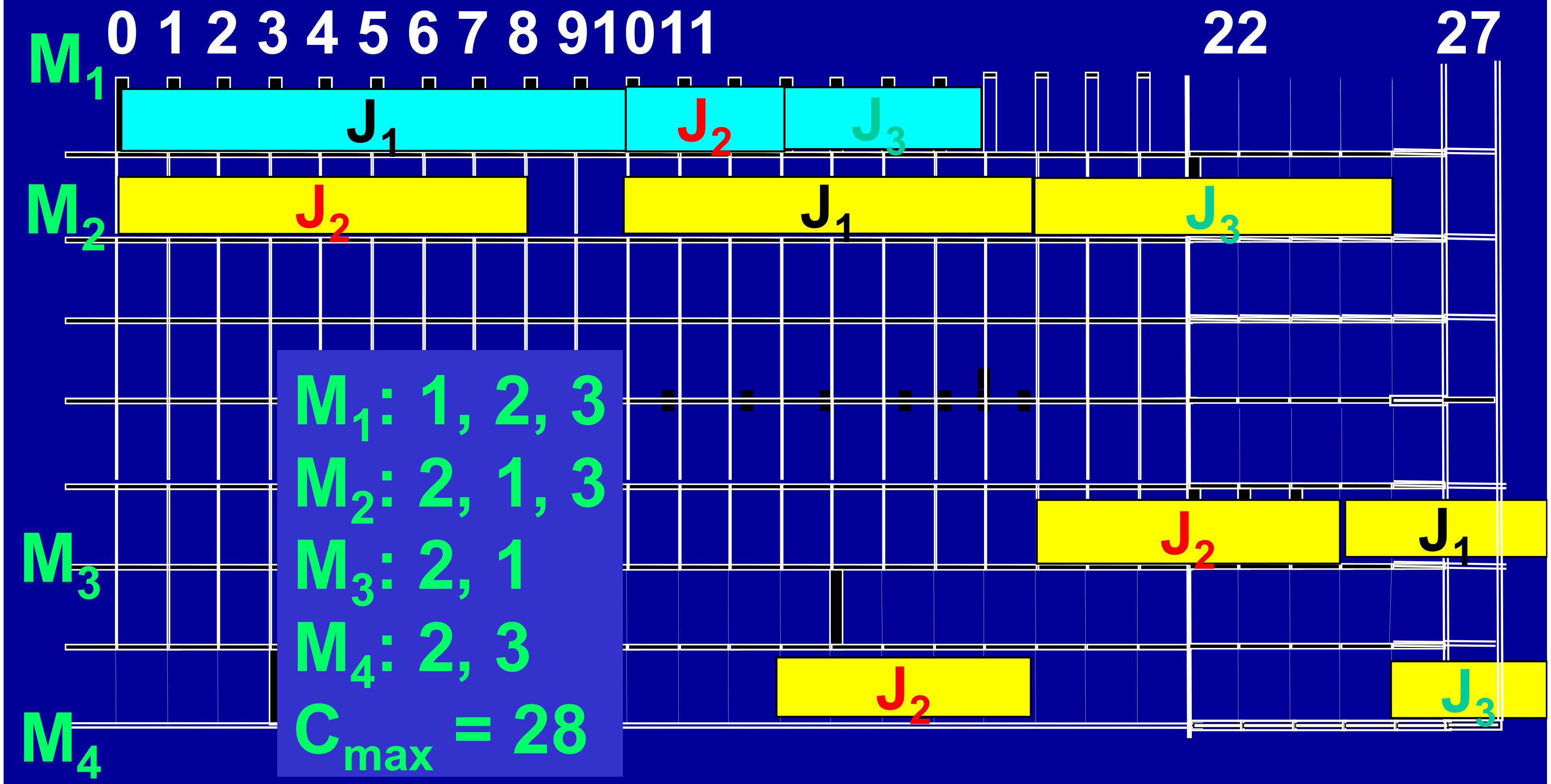
Archi disgiuntivi introdotti al passo 2

Rimane da decidere per le macchine 3 e 4. Calcolando come abbiamo fatto in precedenza le lateness massime, con i nuovi archi introdotti, si ottiene:

$$L_{3max}(3) = L_{4max}(4) = 0$$

nessuna delle due costituisce un collo di bottiglia, quindi si possono compattare entrambe con il sequenziamento minL_{max}, senza aumentare il C_{max}

Sequenziamento



M_1: 1, 2, 3

M_2: 2, 1, 3

M_3: 2, 1

M_4: 2, 3

$C_{max} = 28$

Vediamo ora cosa avviene se si compatta per primo l'altro "collo di bottiglia" M_2 con il relativo primo sequenziamento ottimo $J_2\ J_3\ J_1$

percorso critico

Ricalcoliamo sul grafo di precedenza i valori di r_{ij} e d_{ij}

Ricalcoliamo gli L_{max}

	M_1				M_3			M_4	
J_{EDD}	3	1	2		2	1		2	3
r_{1j}	0	0	3	r_{3j}	16	23	r_{4j}	11	15
p_{1j}	4	10	3	p_{3j}	6	4	p_{4j}	5	3
d_{1j}	8	15	16	d_{3j}	27	27	d_{4j}	21	27
C_{EDD}	4	14	17		22	27		16	19

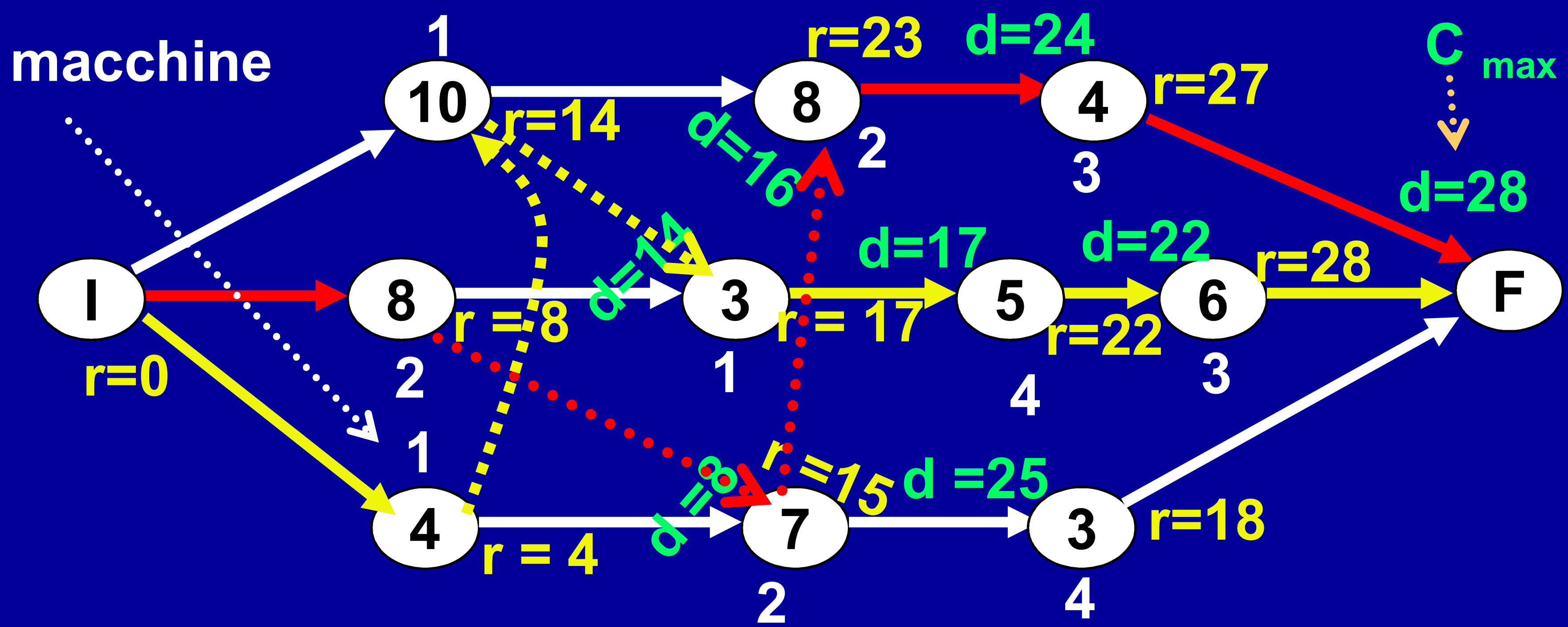
$L_{1max} = 1$ $\quad$ $L_{3max} = 0$ $\quad$ $L_{4max} = -5$

questa volta M_1 è la sola macchina critica:

passo 2: $M^2 = (M_1, M_2)$

$C_{max}(2) = C_{max}(1) + L_{2max}(1) = 28$

quindi si compatta ora M_1 con il relativo sequenziamento ottimo $J_3 J_1 J_2$



Ricalcoliamo sul grafo di precedenza i valori di r_{ij} e d_{ij}

Ricalcoliamo gli L_{max}

	M_3			M_4	
J_{EDD}	2	1		2	3
r_{3j}	22	23	r_{4j}	17	15
p_{3j}	6	4	p_{4j}	5	3
d_{3j}	28	28	d_{4j}	22	28
C_{EDD}	28	32		22	28
	$L_{3max} = 4$			$L_{4max} = -3$	

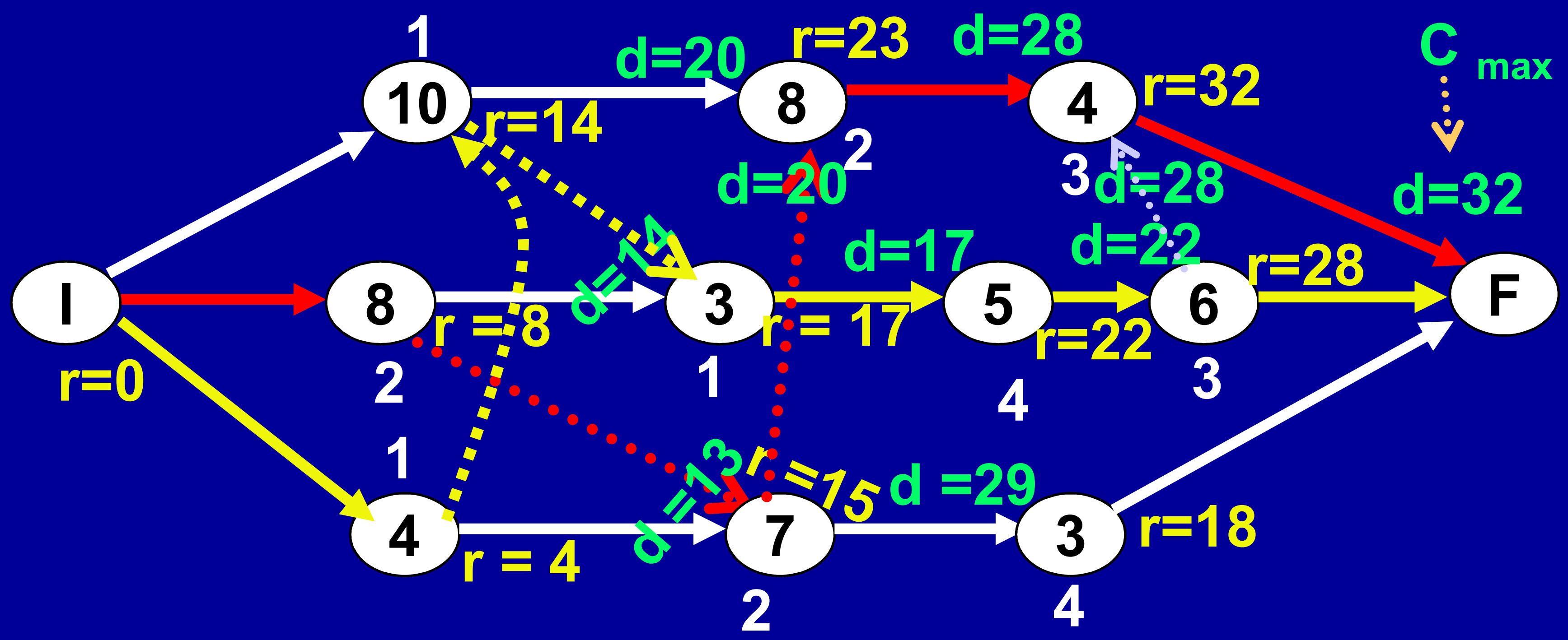
M_3 è la macchina critica:

passo 3: $M = (M_1, M_2, M_3)$

$C_{max}(3) = C_{max}(2) + L_{2max}(2) = 32$

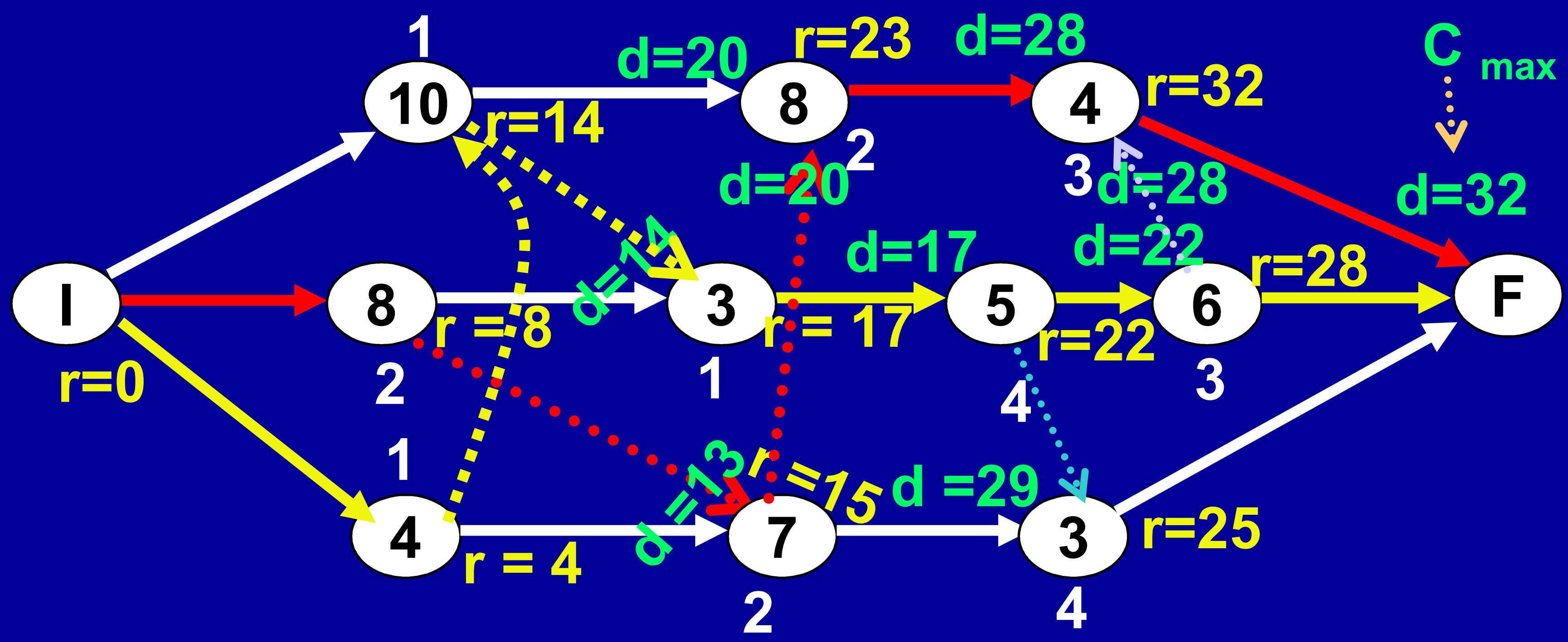
Si compatta M_3 con il relativo sequenziamento ottimo $J_2\ J_1$

Si aggiornano le d e le r per sequenziare M_4 che da' $L_m = -7$ con $J_2\ J_3$



Si compatta M_4 con il relativo sequenziamento ottimo J_2 J_3

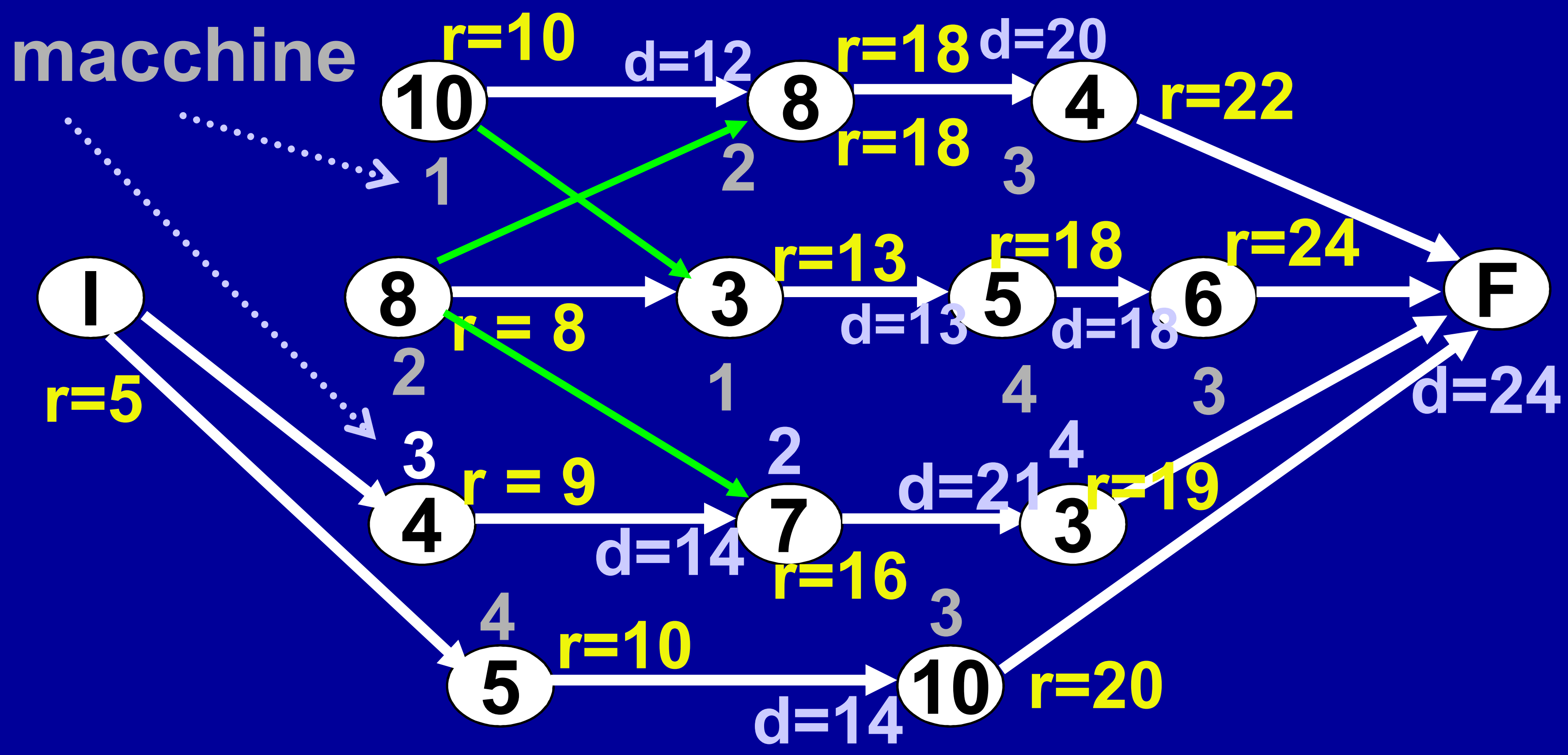
Si riportano le r finali per verificare che M_4 termini a 32 – 7 = 25



La scelta iniziale di M_2 ha portato a un risultato peggiore di quella iniziale di M_1.Questa è l'euristica!

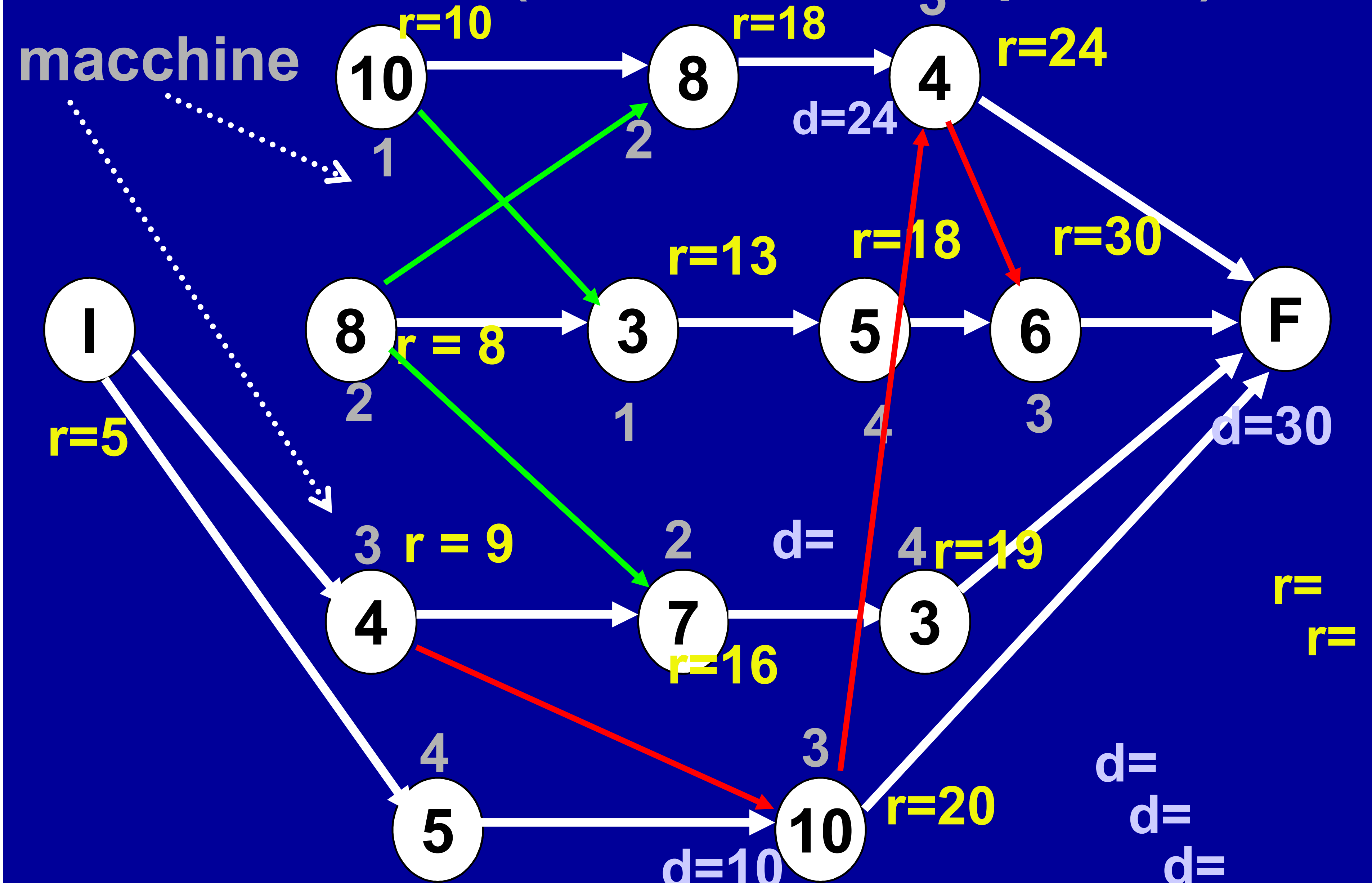
In generale, per cercare di migliorare il risultato è bene sviluppare tutte le alternative che si incontrano nella procedure.

L'algoritmo può essere applicato "in linea" a partire da uno stato qualsiasi, quando intervengono variazioni (**nuovi lavori** e/o **instradamenti**). Ad esempio al tempo 5 si abbia:

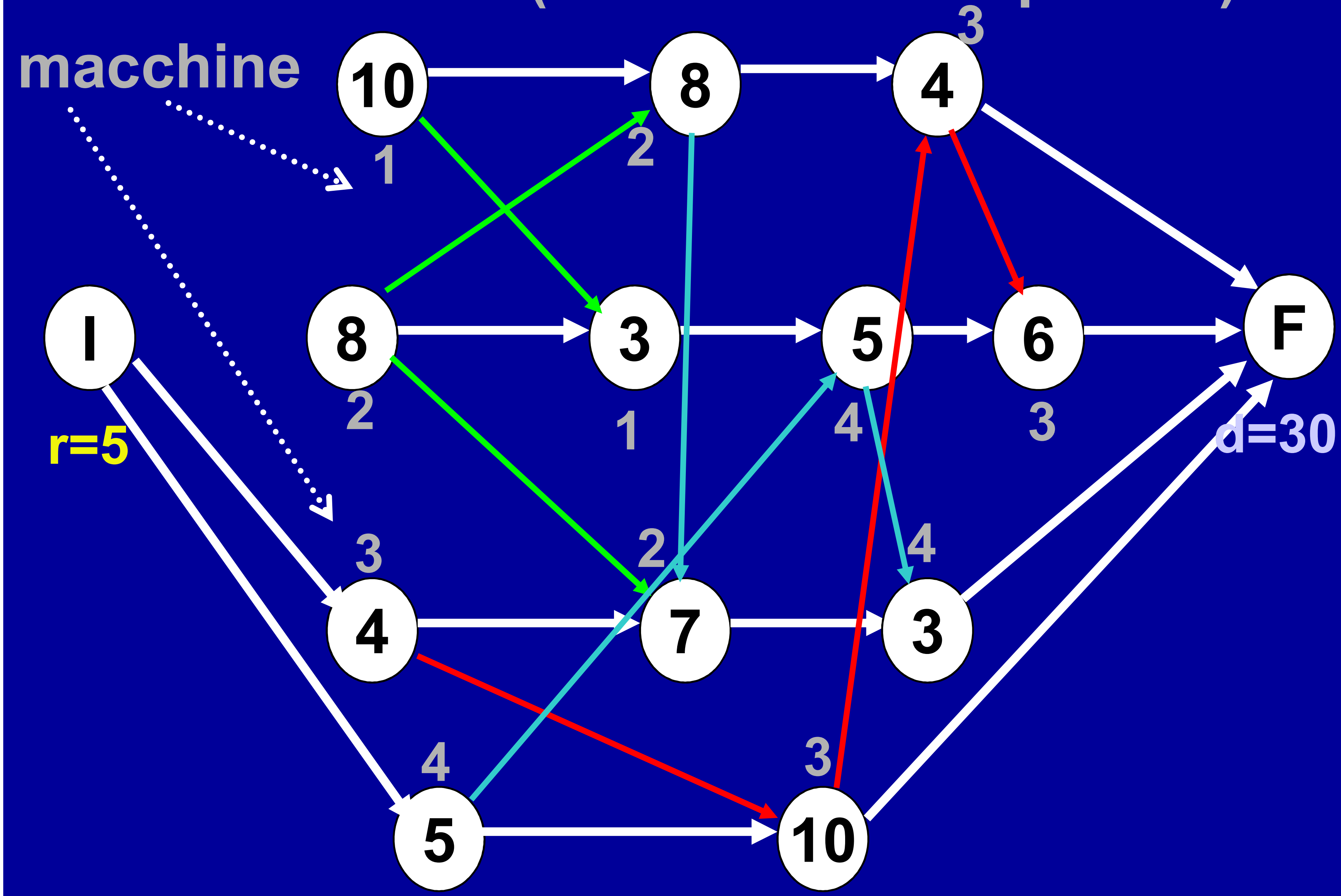


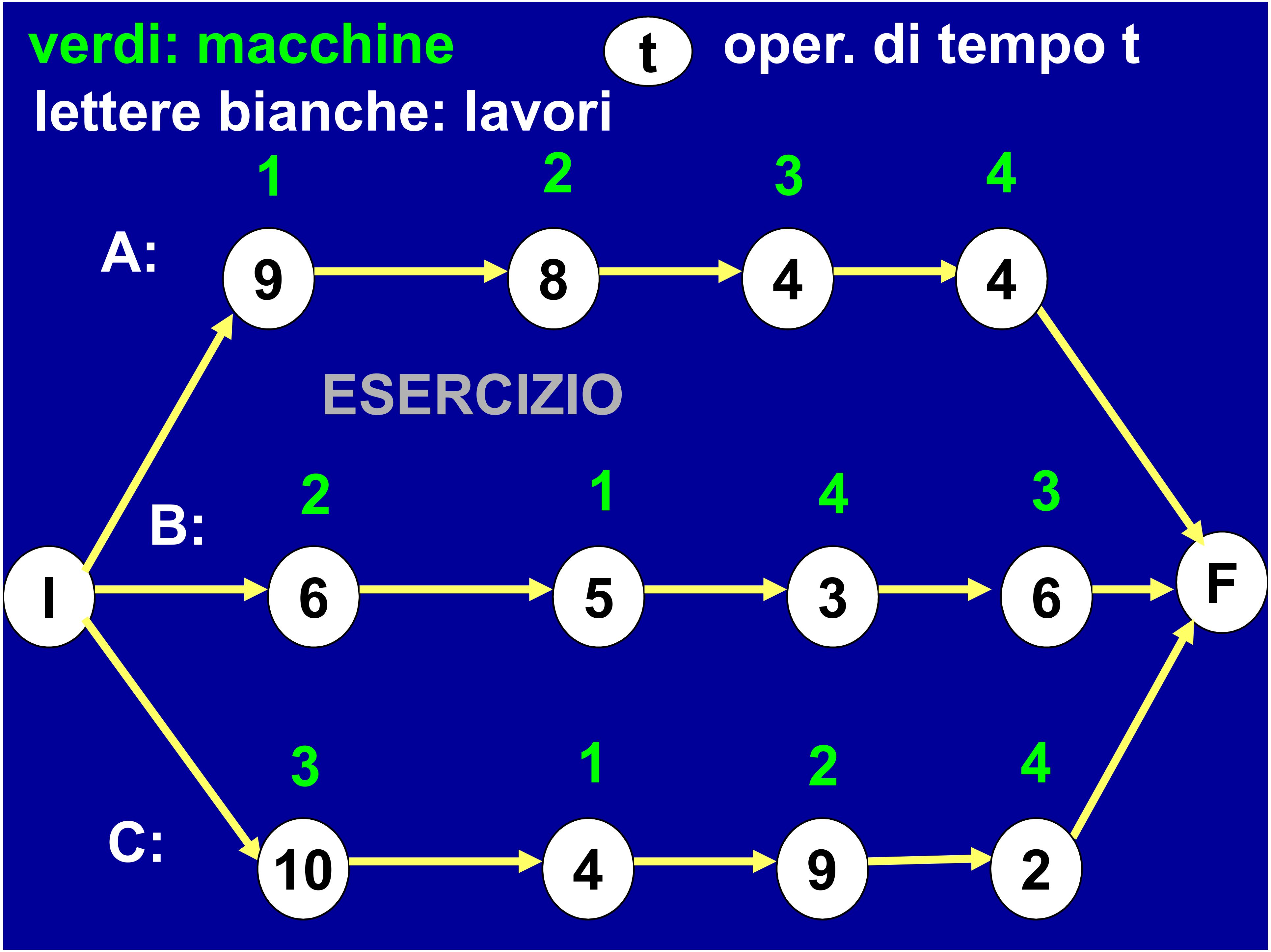
La macchina 1 è completamente sequenziata, la 2 parzialmente: guardando i percorsi critici a monte si calcolano i tempi di rilascio e quindi il dovuto finale (completamento) con le macchine multiplate (M_2 due volte), quindi tutti i dovuti etc……….

Prima iterazione (verificare e completare)



Ultima iterazione (verificare e completare)




verdi: macchine
t
oper. di tempo t
lettere bianche: lavori
ESERCIZIO
A:
1
2
3
4
9
8
4
4
B:
2
1
4
3
6
5
3
6
C:
3
1
2
4
10
4
9
2
I
F

PROBLEMI DI GESTIONE DEGLI FMS

Flexible Manufacturing System: sistemi integrati di produzione flessibile

- mix produttivo
- attrezzaggio
- instradamento
- sequenziamento

controllo