

Χρονοδρομολόγηση/Δρομολόγηση/σχεδιασμός (Scheduling-routing)

$M = \{M_1, M_2, \dots, M_n\}$: n μηχανές (machines, processors)

$J = \{J_1, J_2, \dots, J_m\}$: m εργασίες (Jobs)

$J_j \rightarrow p_j \geq 0$ χρόνοι εκτέλεσης.

schedule(χρονοπρόγραμμα): Μια ανάθεση των εργασιών στις μηχανές.

Περιορισμός: Δεν μπορούν να εκτελούνται 2 εργασίες στην ίδια μηχανή ταυτόχρονα.

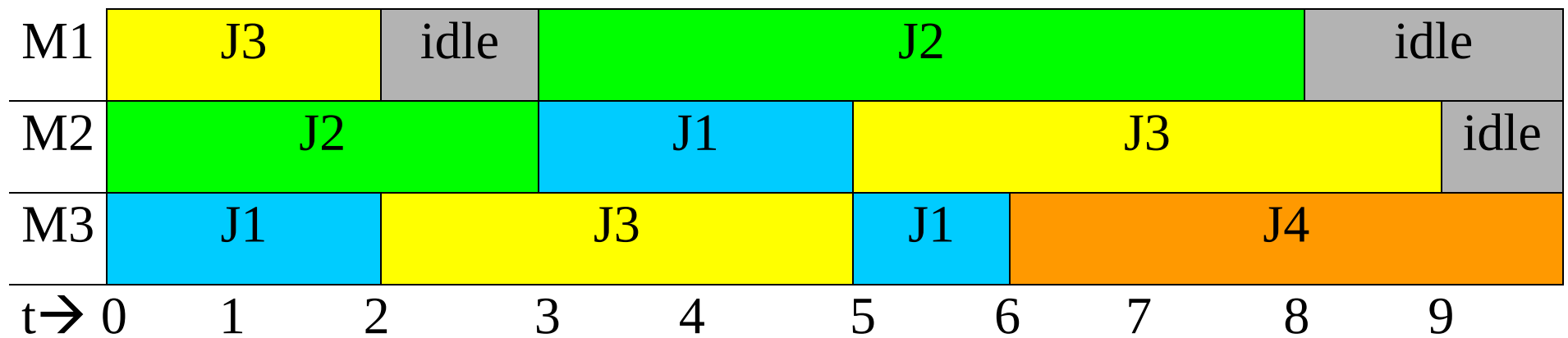
Ζητούμενο: Ένα χρονοπρόγραμμα ώστε να ελαχιστοποιείται κάποια ποσότητα.

εφικτό χρονοπρόγραμμα (feasible schedule): για το Π , πληρεί τους περιορισμούς του Π .

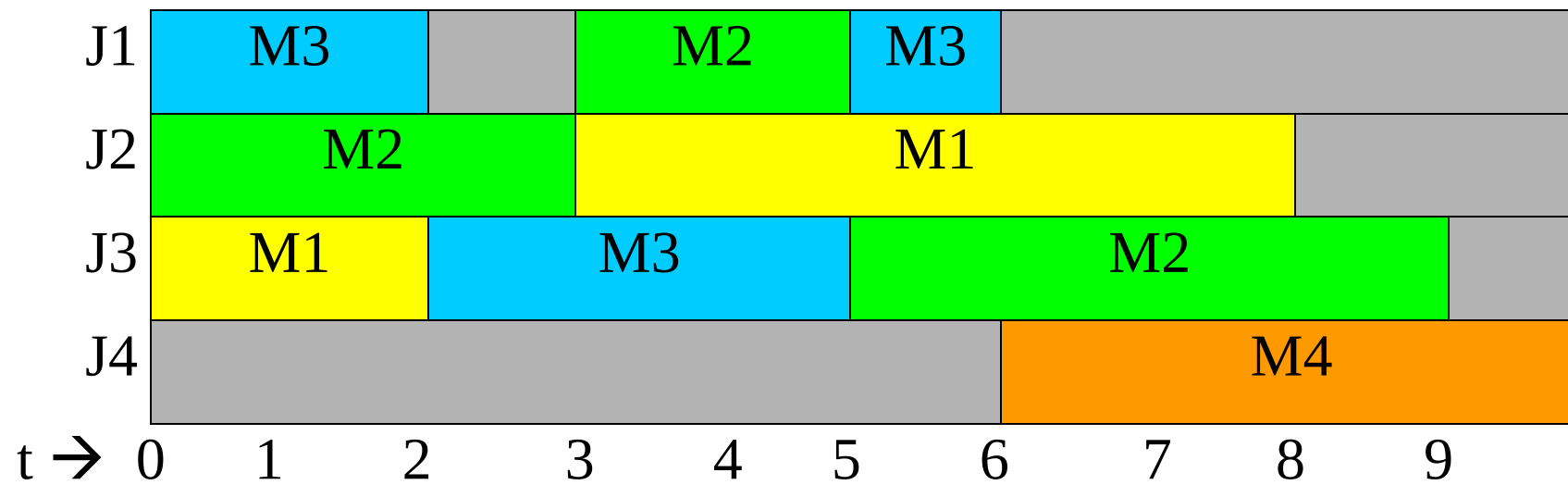
idle (ανενεργή): θα λέγεται μια μηχανή M_i το χρονικό διάστημα $[t_0, t_0+k]$, $k \in \mathbb{N}^*$ αν στο χρονικό αυτό διάστημα δεν εκτελείται καμία εργασία στην M_i .

Gantt Charts

1.Ως προς τις μηχανές (Machine oriented)



2.Ως προς την εργασία (Job Oriented)



Συμβολισμός προβλημάτων χρονοδρομολόγησης: **A** | **B** | **C**

Graham (1979)

A: περιορισμοί/χαρακτηριστικά επεξεργαστών (machine environment)

1: Sequencing (μόνο μια μηχανή)

Pn: n όμοιοι επεξεργαστές, (identical parallel machines).

Qn: n ομοιόμορφοι επεξεργαστές (uniform parallel machines)

$p_{ij} = p_j / s_i$, s_i η ταχύτητα της M_i , p_{ij} : ο χρόνος εκτέλεσης της J_j στην M_i , p_j :
χρόνος επεξεργασίας της J_j

Rn: n το πλήθος ασυσχέτιστες μηχανές (unrelated parallel machines)

p_{ij} : ο χρόνος εκτέλεσης της J_j στην M_i , p_j : χρόνος επεξεργασίας της J_j
τότε $p_{ij} = p_j / s_{ij}$, s_{ij} εξαρτάται από τη μηχανή και την εργασία

On, Jn, Fn: shop problems με n επεξεργαστές (Open, Job, Flow)

B: περιορισμοί/χαρακτηριστικά εργασιών (Job characteristics)

Prec: περιορισμοί προτεραιότητας. (precedence constraints)

Prmn: μπορούμε να διακόψουμε την εκτέλεση μιας εργασίας.
(preemption).

d_j : Η J_j πρέπει να έχει ολοκληρωθεί μέχρι τη χρονική στιγμή d_j . (due dates)

r_j : Η J_j δεν μπορεί να ξεκινήσει πριν από τη χρονική στιγμή r_j . (release dates)

$p_i=a, b, \text{ etc}$: περιορισμοί στους χρόνους εκτέλεσης. (processing requirements)

C: ποσότητα προς ελαχιστοποίηση (optimality criterion)

Συμβολισμός: C_j η χρονική στιγμή ολοκλήρωσης της J_j . (completion time)

$C_{\max}$: Μέγιστη χρονική στιγμή ολοκλήρωσης εργασίας ($\max\{C_j \mid j=1,2,\dots,m\}$).

Συμβολισμός: L_j η καθυστέρηση της J_j

$L_{\max}$: (maximum lateness) μέγιστη καθυστέρηση ($\max\{L_j \mid j=1,2,\dots,m\}$).

$\sum L_j$: συνολική καθυστέρηση (άθροισμα). (total lateness/tardiness)

$\sum C_j$: Sum of completion times

T: πλήθος εργασιών που καθυστέρησαν (number of Tardy Jobs)

Παραδείγματα συμβολισμών:

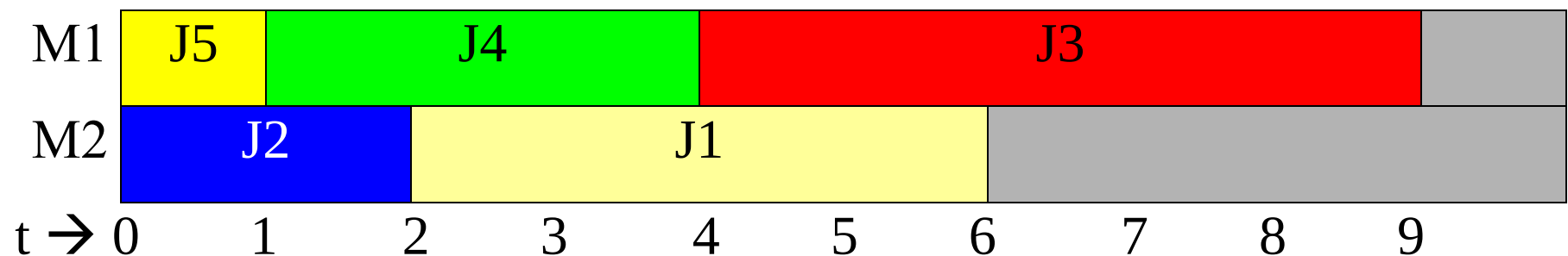
- $Rn \mid \mid \sum C_j$ (Πολυωνυμικό, Horn et al. 1973)
- $P2 \mid \mid C_{max}$ (NP-hard, Garey et al. 1979)
- $1 \mid r_j, d_j \mid L_{max}$ (NP-hard, Lenstra et al. 1977)
- $On \mid p_{ij}=0,1 \mid C_{max}$ (Πολυωνυμικό, Labetoulle et al. 1984)
- $F2 \mid \mid C_{max}$ (Πολυωνυμικό, Johnson et al. 1954)

Παραδείγματα εφικτών δρομολογήσεων:

1. $P2 \mid \mid C_{\max}$ vs $P2 \mid \mid \sum C_j$

	J1	J2	J3	J4	J5
p_j	4	2	5	3	1

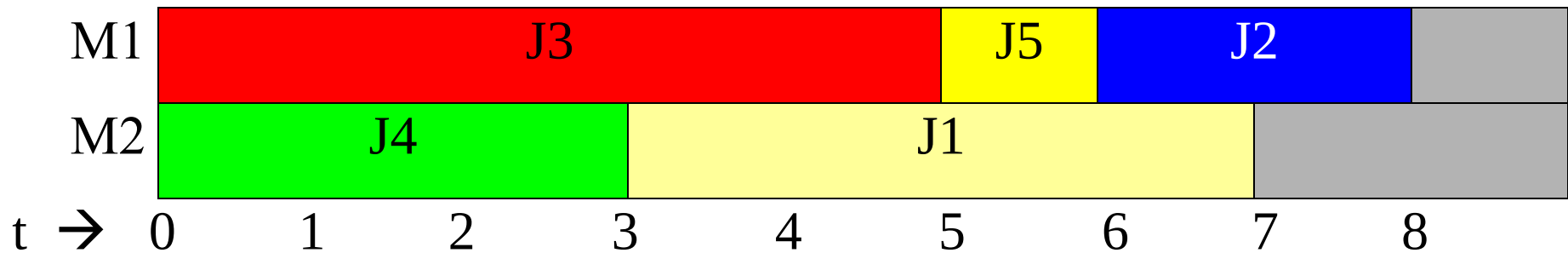
Δρομολόγηση 1:



$$\sum C_j = C_1 + C_2 + C_3 + C_4 + C_5 = 6 + 2 + 9 + 4 + 1 = 22, C_{\max} = 9$$

	J1	J2	J3	J4	J5
p _j	4	2	5	3	1

Δρομολόγηση 2:

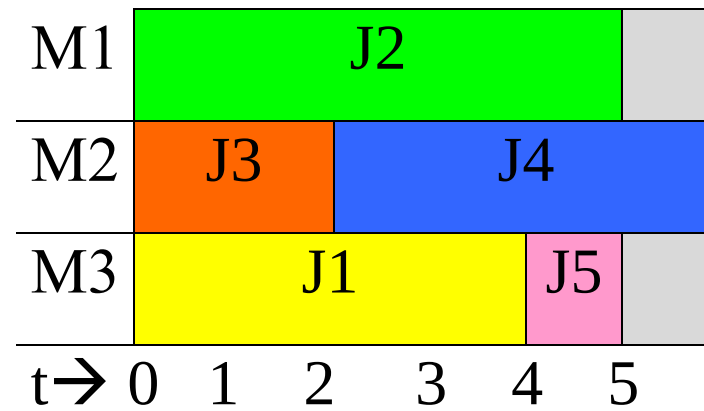


$$\sum C_j = C_1 + C_2 + C_3 + C_4 + C_5 = 7 + 8 + 5 + 3 + 6 = 29$$

$$C_{\max} = 8$$

2.P3 | d_j | $L_{\max}$ or P3 | d_j | T

	J1	J2	J3	J4	J5
p_j	4	5	2	4	1
d_j	3	4	2	4	5



$L_{\max} = 2, T=3$

3. O4 | $p_{ij}=0,1$ | $C_{\max}$

Open shop:

Κάθε εργασία αποτελείται από υποεργασίες (operations) ,

$J_j = \{O_{1j}, O_{2j}, \dots, O_{n_j}\}$, με $p_{ij} \geq 0$, $1 \leq i \leq n$, $1 \leq j \leq m$,

η O_{ij} πρέπει να εκτελεστεί στον i επεξεργαστή,

υποεργασίες της ίδιας εργασίας δεν μπορούν να εκτελούνται ταυτόχρονα.

Ζητούμενο είναι η ελαχιστοποίηση του $C_{\max}$.

Στιγμιότυπο του O4 | $p_{ij} = 0, 1$ | $C_{\max}$

	J1	J2	J3	J4	J5
M1	1	0	0	1	0
M2	0	1	1	1	1
M3	0	1	1	1	0
M4	1	1	0	1	1

Δρομολόγηση MJ:

M1	J1			J4
M2	J4	J2	J5	J3
M3		J4	J3	J2
M4	J2	J1	J4	J5
t → 0	1	2	3	4

Δρομολόγηση JM:

M1			J4	J1	
M2	J4	J3	J2	J5	
M3		J4	J3	J2	
M4	J2	J1	J5	J4	
t →	0	1	2	3	4

$$1 \mid d_j \mid L_{\max}$$

Αλγόριθμος LS (List Scheduling)

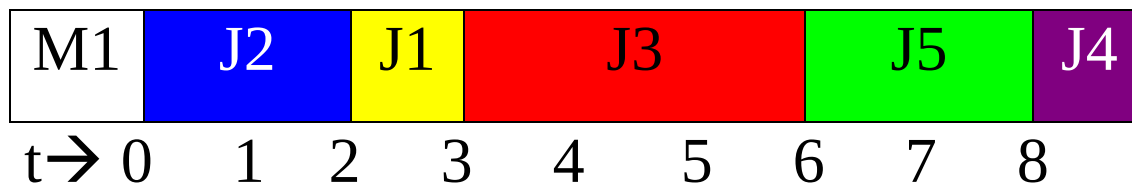
1. Ταξινόμησε τις εργασίες κατά αύξουσα σειρά due date.
2. Ανέθεσε στην μηχανή την εργασία με το μικρότερο due date από αυτά που έχουν απομείνει.

Θεώρημα: Ο LS παράγει βέλτιστο χρονοπρόγραμμα για το $1 \mid d_j \mid L_{\max}$

Παράδειγμα:

	J1	J2	J3	J4	J5
p_i	1	2	3	1	2
d_i	2	1	3	7	4

Ο LS θα παράγει το ακόλουθο χρονοπρόγραμμα:



$$L_1=1, L_2=1, L_3=3, L_4=2, L_5=4 \Rightarrow L_{\max}=4.$$

Απόδειξη (με επιχείρημα ανταλλαγής):

Έστω ένα **τυχαίο χρονοπρόγραμμα** για το πρόβλημα. Θα δείξουμε ότι **αλλάζοντας τη σειρά των εργασιών κατά τέτοιο τρόπο ώστε να ακολουθούν τον LS** η χρονοδρομολόγηση μπορεί μόνο να βελτιωθεί.

Έστω J_i και J_j με χρόνους εκτέλεσης p_i, p_j αντίστοιχα, **οι πρώτες διαδοχικές εργασίες που δεν ακολουθούν τον LS** και έστω ότι προηγείται η J_i με $d_i > d_j$. Έστω επίσης T η χρονική στιγμή που ξεκινάει η επεξεργασία του J_i . Κλπ.

O2 | | C_{max}

Πολυπλοκότητα: Πολυωνυμική

Κανόνας LAPT (Longest alternate processing time)

Οποτεδήποτε μία μηχανή είναι ελεύθερη επίλεξε την εργασία με το μεγαλύτερο χρόνο εκτέλεσης στην άλλη μηχανή.

Παράδειγμα: Στιγμιότυπο Open Shop:

	J1	J2	J3	J4	J5
M 1	2	2	3	1	5
M2	1	3	4	2	3

Gantt Chart σύμφωνα με LAPT :

