

**Σχόλιο:** Αν τα due dates είναι αρκούντως μεγαλύτερα από τους χρόνους εκτέλεσης τότε το  $L_{\max}$  μπορεί να βγει αρνητικό. Προκειμένου να μην συμβεί κάτι τέτοιο μπορούμε να αφαιρέσουμε από κάθε due date μία σταθερά (την ίδια από όλα) ώστε όλα τα  $L_j$  να είναι μη αρνητικά.

## ΑΣΚΗΣΕΙΣ

**13.1:** Έστω ο κανόνας Longest Processing Time (LPT) για το  $O2 \mid \mid C_{\max}$ . Δηλαδή σε όποιον επεξεργαστή μείνει ελεύθερος ανατίθεται το Operation με τον μεγαλύτερο χρόνο εκτέλεσης στον επεξεργαστή αυτό. Εξηγήστε γιατί ο κανόνας αυτός δεν παράγει βέλτιστο schedule.

**13.2:** Δώστε παραδείγματα εφικτών χρονοδρομολογήσεων για τα προβλήματα:

$$(α) P3 \mid \mid C_{\max}, \quad (β) O2 \mid \mid C_{\max}, \quad (γ) 1 \mid d_j \mid L_{\max}.$$

Για τις περιπτώσεις (β) και (γ) χρησιμοποιήστε αλγόριθμο που βρίσκει τη βέλτιστη χρονοδρομολόγηση και πείτε ποιός είναι αυτός. Σχολιάστε την περίπτωση (α).

**13.3:** Δείξτε ότι ο κανόνας LS δεν δίνει βέλτιστο schedule για το πρόβλημα  $1 \mid d_j \mid \sum T_j$ , όπου  $T_j$  είναι η καθυστέρηση της Job  $j$ .

**13.4:** Δώστε παραδείγματα εφικτών χρονοδρομολογήσεων για τα προβλήματα:

$$(α) O3 \mid \mid C_{\max}, \quad (β) 1 \mid r_j, prmt \mid \sum C_j.$$

**13.5:** Έστω το πρόβλημα  $1 \mid r_j, prmt \mid \sum C_j$  και ο κανόνας που αναφέρεται στη βιβλιογραφία ως shortest remaining processing time (SRPT):

*Από τις εργασίες που είναι διαθέσιμες για εκτέλεση χρονοδρομολόγησε αυτήν με το μικρότερο εναπομείναντα χρόνο εκτέλεσης.*

Αποφανθείτε αν ο SRPT δίνει βέλτιστη χρονοδρομολόγηση για το συγκεκριμένο πρόβλημα. Αποδείξτε τον ισχυρισμό σας.

**13.6:** Έστω το πρόβλημα  $1 \mid \mid \sum w_j C_j$ , όπου  $w_j$  είναι το βάρος του Job  $j$ . Έστω επίσης ο κανόνας «weighted shortest processing time first» (WSPT). Σύμφωνα με τον κανόνα αυτό τα jobs ταξινομούνται σε φθίνουσα σειρά του πηλίκου  $w_j/p_j$ , όπου  $p_j$  είναι ο χρόνος εκτέλεσης του job  $j$ . Δείξτε ότι ο WSPT δίνει βέλτιστο schedule για το πρόβλημα.

## 14.1 BUCKET ΓΙΑ ΤΑ

### 14.1.1 Παράδειγμα

Ο Bucket algorithm απαιτεί γραμμή υποθέσεων. Θα μπορούσαμε κάπως να

Ο Bucket algorithm ξένα μεταβλητά όπου  $x = 0, 1$  στα όποιον αλγόριθμο στο διάστημα  $[0, 1]$  λουθούν τα jobs δεν θα εργαζόμαστε έχει περιπτώσεις τώρα απλά τα μημένες  $[0, 1/p]$  και τα jobs που δεν εργαζόμαστε λίστας χρονοδρομολογήσεων