

Προσεγγιστικά Σχήματα για Προβλήματα Χρονοδρομολόγησης

Γιώργος Ζώης

Οικονομικό Πανεπιστήμιο Αθηνών

Τμήμα Πληροφορικής

georzois@aueb.gr

Απρίλιος 2010

Σκιαγράφηση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης

Σκιαγράφιση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης
- 2 Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης σε μοναδική μηχανή
 - Επισκόπηση του προβλήματος
 - Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου

Σκιαγράφιση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης
- 2 Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης σε μοναδική μηχανή
 - Επισκόπηση του προβλήματος
 - Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου
- 3 Χρονοδρομολόγηση με δεδομένες προθεσμίες
 - Συναρτήσεις απώλειας
 - Ένα δυναμικό πρόγραμμα ψευδοπολυωνυμικού χρόνου
 - Εφαρμογές του δυναμικού προγράμματος

Σκιαγράφηση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης
- 2 Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης σε μοναδική μηχανή
 - Επισκόπηση του προβλήματος
 - Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου
- 3 Χρονοδρομολόγηση με δεδομένες προθεσμίες
 - Συναρτήσεις απώλειας
 - Ένα δυναμικό πρόγραμμα ψευδοπολυωνυμικού χρόνου
 - Εφαρμογές του δυναμικού προγράμματος

Ο ρόλος της χρονοδρομολόγησης

Χρονοπρογραμματισμός της εκτέλεσης ενός συνόλου δραστηριοτήτων έχοντας περιορισμένους πόρους.

- Πόροι: μηχανές ενός εργοστασίου, CPUs ενός υπολογιστικού περιβάλλοντος κ.ο.κ
- Δραστηριότητες: λειτουργίες παραγωγής, προγράμματα ενός υπολογιστή κ.ο.κ
- Στόχοι: ελαχιστοποίηση του χρόνου ολοκλήρωσης όλων των δραστηριοτήτων κ.ο.κ

Ο ρόλος της χρονοδρομολόγησης

Χρονοπρογραμματισμός της εκτέλεσης ενός συνόλου δραστηριοτήτων έχοντας περιορισμένους πόρους.

- Πόροι: μηχανές ενός εργοστασίου, CPUs ενός υπολογιστικού περιβάλλοντος κ.ο.κ
- Δραστηριότητες: λειτουργίες παραγωγής, προγράμματα ενός υπολογιστή κ.ο.κ
- Στόχοι: ελαχιστοποίηση του χρόνου ολοκλήρωσης όλων των δραστηριοτήτων κ.ο.κ
- Θεωρία Προβλημάτων Χρονοδρομολόγησης (1950)
 - Οικονομία (Επιχειρησιακή Έρευνα), Πληροφορική (Βάσεις Δεδομένων, Θεωρητική Πληροφορική)

Μοντελοποίηση

Ένα τυπικό πρόβλημα χρονοδρομολόγησης αποτελείται από ένα σύνολο J , n **διεργασιών** και ένα σύνολο M , m **μηχανών**. Κάθε διεργασία $j \in J$ διαθέτει έναν **χρόνο επεξεργασίας** p_j καθώς και κάποιες προαιρετικές ιδιότητες (π.χ. ένα βάρος w_j , έναν χρόνο αποδέσμευσης r_j , μία προθεσμία d_j κ.ο.κ).

Οι μηχανές μπορεί να είναι πανομοιότυπες ενώ η εκτέλεση των διεργασιών σ' αυτές ενδεχομένως να **υπόκειται σε περιορισμούς** (π.χ. preemption, περιορισμοί προτεραιότητας κ.ο.κ).

Ο στόχος είναι να χρονοδρομολογήσουμε τις διεργασίες στις μηχανές έτσι ώστε να βελτιστοποιήσουμε κάποια **αντικειμενική συνάρτηση**.

Περιβάλλον μηχανών

- Μοναδική μηχανή
- Παράλληλες πανομοιότυπες μηχανές
 - Γενικό πλήθος
 - Σταθερό πλήθος $m > 1$
- Παράλληλες σχετιζόμενες μηχανές δεδομένης ταχύτητας
- Παράλληλες μη σχετιζόμενες μηχανές
- Flow shop, Job shop

Ενδεικτικές αντικειμενικές συναρτήσεις

Makespan

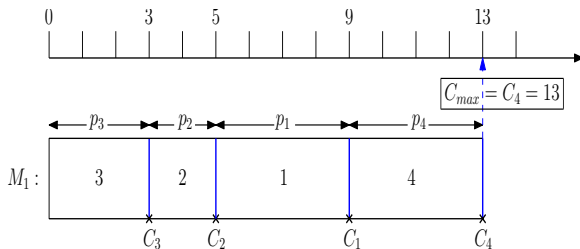
$C_{max} = \max\{C_1, C_2, \dots, C_n\}$, όπου C_j ο χρόνος ολοκλήρωσης της διεργασίας j σε δεδομένη χρονοδρομολόγηση.

Ενδεικτικές αντικειμενικές συναρτήσεις

Makespan

$C_{max} = \max\{C_1, C_2, \dots, C_n\}$, όπου C_j ο χρόνος ολοκλήρωσης της διεργασίας j σε δεδομένη χρονοδρομολόγηση.

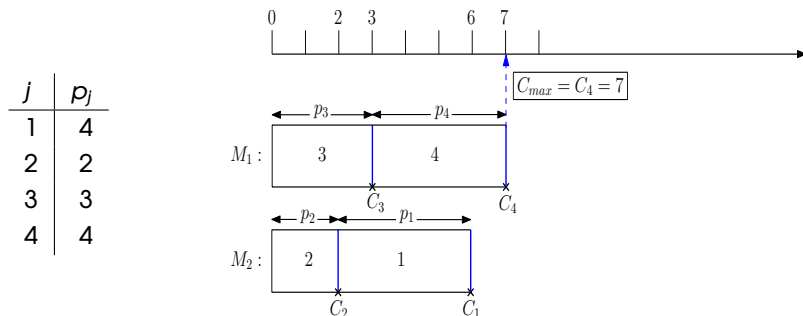
j	p_j
1	4
2	2
3	3
4	4



Ενδεικτικές αντικειμενικές συναρτήσεις

Makespan

$C_{max} = \max\{C_1, C_2, \dots, C_n\}$, όπου C_j ο χρόνος ολοκλήρωσης της διεργασίας j σε δεδομένη χρονοδρομολόγηση.



Ενδεικτικές αντικειμενικές συναρτήσεις

Μέσος βεβαρημένος χρόνος ολοκλήρωσης

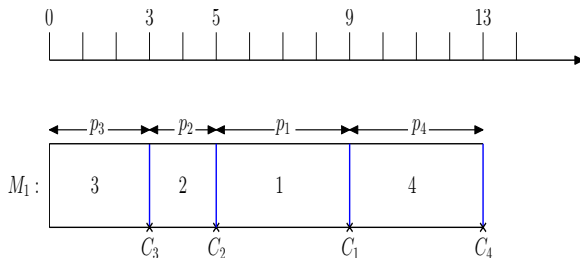
- $\sum_{j=1}^n w_j C_j$
- $\sum_{j=1}^n C_j, w_j = 1$ για κάθε διεργασία j (συνολικός χρόνος ολοκλήρωσης)

Ενδεικτικές αντικειμενικές συναρτήσεις

Μέσος βεβαρημένος χρόνος ολοκλήρωσης

- $\sum_{j=1}^n w_j C_j$
- $\sum_{j=1}^n C_j$, $w_j = 1$ για κάθε διεργασία j (συνολικός χρόνος ολοκλήρωσης)

j	p_j	w_j
1	4	1
2	2	1
3	3	1
4	4	1

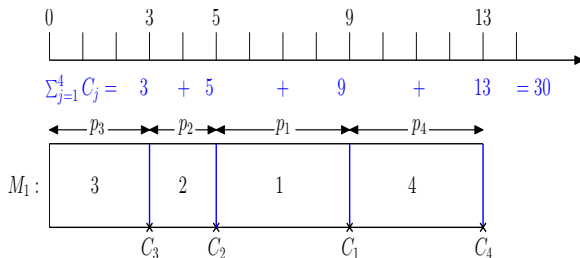


Ενδεικτικές αντικειμενικές συναρτήσεις

Μέσος βεβαρημένος χρόνος ολοκλήρωσης

- $\sum_{j=1}^n w_j C_j$
- $\sum_{j=1}^n C_j$, $w_j = 1$ για κάθε διεργασία j (συνολικός χρόνος ολοκλήρωσης)

j	p_j	w_j
1	4	1
2	2	1
3	3	1
4	4	1

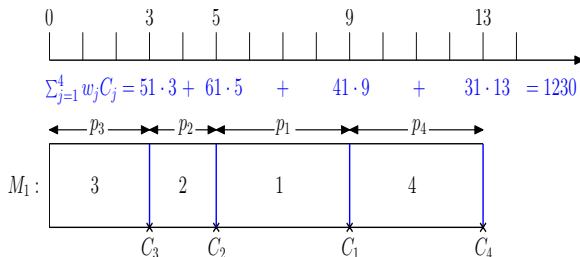


Ενδεικτικές αντικειμενικές συναρτήσεις

Μέσος βεβαρημένος χρόνος ολοκλήρωσης

- $\sum_{j=1}^n w_j C_j$
- $\sum_{j=1}^n C_j$, $w_j = 1$ για κάθε διεργασία j (συνολικός χρόνος ολοκλήρωσης)

j	p_j	w_j
1	4	41
2	2	61
3	3	51
4	4	31



Πολυπλοκότητα των προβλημάτων χρονοδρομολόγησης

- Προβλήματα Συνδυαστικής Βελτιστοποίησης

Πολυπλοκότητα των προβλημάτων χρονοδρομολόγησης

- Προβλήματα Συνδυαστικής Βελτιστοποίησης
- Τα περισσότερα ενδιαφέροντα προβλήματα χρονοδρομολόγησης είναι υπολογιστικά δύσβατα (NP-hard)

Πολυπλοκότητα των προβλημάτων χρονοδρομολόγησης

- Προβλήματα Συνδυαστικής Βελτιστοποίησης
- Τα περισσότερα ενδιαφέροντα προβλήματα χρονοδρομολόγησης είναι υπολογιστικά δύσβατα (NP-hard)
- Προσεγγιστικοί αλγόριθμοι

Προσεγγιστικοί αλγόριθμοι

Ορισμός

Θεωρούμε ένα πρόβλημα βελτιστοποίησης Π και έστω συνάρτηση $\rho : \mathbb{Z}^+ \rightarrow \mathbb{Q}^+$, με $\rho \geq 1$. Ένας πολυωνυμικός αλγόριθμος A καλείται **ρ -προσεγγιστικός αλγόριθμος** για το πρόβλημα Π , αν για κάθε στιγμιότυπο I του Π βρίσκει μία εφικτή λύση f κόστους $c(f)$ τέτοια ώστε,

- $c(f) \leq \rho(|I|) \cdot \text{OPT}(I)$, αν το Π είναι **πρόβλημα ελαχιστοποίησης**
- $c(f) \geq \frac{1}{\rho(|I|)} \cdot \text{OPT}(I)$, αν το Π είναι **πρόβλημα μεγιστοποίησης**

Η τιμή $\rho(|I|)$ καλείται **παράγοντας** (ή λόγος) προσέγγισης του προσεγγιστικού αλγορίθμου A .

Προσεγγιστικοί αλγόριθμοι

Ορισμός

Θεωρούμε ένα πρόβλημα ελαχιστοποίησης (αντίστοιχα μεγιστοποίησης) Π .

- Μία οικογένεια $\mathcal{A}_\epsilon$ από $(1 + \epsilon)$ -προσεγγιστικούς αλγορίθμους για το Π , για κάθε σταθερά $0 < \epsilon < 1$, αποτελεί ένα προσεγγιστικό σχήμα για το πρόβλημα.
- Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου (PTAS) για το πρόβλημα Π είναι ένα προσεγγιστικό σχήμα με πολυπλοκότητα χρόνου πολυωνυμική ως προς το μέγεθος της εισόδου.
- Ένα πλήρες πολυωνυμικό προσεγγιστικό σχήμα (FPTAS) για το πρόβλημα Π είναι ένα προσεγγιστικό σχήμα με πολυπλοκότητα χρόνου πολυωνυμική, τόσο ως προς το μέγεθος της εισόδου, όσο και ως προς το $\frac{1}{\epsilon}$.

Περίληψη της μελέτης

Ελαχιστοποίηση του makespan

- PTAS σε
 - δύο παράλληλες πανομοιότυπες μηχανές [WoegingerS01]
 - γενικό πλήθος παράλληλων πανομοιότυπων μηχανών [HochbaumS87]
 - γενικό πλήθος παράλληλων πανομοιότυπων μηχανών [WoegingerS01]

Περίληψη της μελέτης

Ελαχιστοποίηση του makespan

- PTAS σε
 - δύο παράλληλες πανομοιότυπες μηχανές [WoegingerS01]
 - γενικό πλήθος παράλληλων πανομοιότυπων μηχανών [HochbaumS87]
 - γενικό πλήθος παράλληλων πανομοιότυπων μηχανών [WoegingerS01]

Ελαχιστοποίηση του μέσου βεβαρημένου χρόνου ολοκλήρωσης

- FPTAS σε δύο παράλληλες πανομοιότυπες μηχανές [WoegingerS01]
- PTAS στην περίπτωση που οι διεργασίες διαθέτουν δεδομένους χρόνους αποδέσμευσης σε
 - μοναδική μηχανή (μοναδιαία βάρη) [Afrati et al.99]
 - παράλληλες πανομοιότυπες μηχανές (preemptive, non-preemptive) [Afrati et al.99]

Περίληψη της μελέτης

Χρονοδρομολόγηση με δεδομένες προθεσμίες σε μοναδική μηχανή

- **Ακριβείς ψευδοπολυωνυμικοί αλγόριθμοι** για τα προβλήματα
 - ελαχιστοποίησης του συνολικού βάρους των καθυστερημένων διεργασιών [LawlerM69]
 - μεγιστοποίησης της συνολικής βεβαρημένης πρόωρης ολοκλήρωσης [LawlerM69]
 - ελαχιστοποίησης της συνολικής βεβαρημένης καθυστέρησης, όταν οι διεργασίες διαθέτουν **μία κοινή προθεσμία** [LawlerM69]
 - ελαχιστοποίησης της συνολικής καθυστέρησης (μοναδιαία βάρη) [Lawler77]
- **FPTAS** για τα προβλήματα
 - ελαχιστοποίησης του συνολικού βάρους των καθυστερημένων διεργασιών [WoegingerS01]
 - ελαχιστοποίησης της συνολικής καθυστέρησης (μοναδιαία βάρη) [Lawler82]

Σκιαγράφηση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης
- 2 Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης σε μοναδική μηχανή
 - Επισκόπηση του προβλήματος
 - Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου
- 3 Χρονοδρομολόγηση με δεδομένες προθεσμίες
 - Συναρτήσεις απώλειας
 - Ένα δυναμικό πρόγραμμα ψευδοπολυωνυμικού χρόνου
 - Εφαρμογές του δυναμικού προγράμματος

Γνωστά αποτελέσματα σε μοναδική μηχανή

Αν δεν υπάρχουν περιορισμοί και επιπλέον ιδιότητες για τις διεργασίες

- Βέλτιστη λύση σε πολυωνυμικό χρόνο [Smith56].

Κανόνας SPT (Shortest Processing Time)

- Εκτέλεσε τις διεργασίες σε αύξουσα διάταξη ως προς το χρόνο επεξεργασίας τους:

$$p_1 \leq p_2 \leq \dots \leq p_n.$$

Γνωστά αποτελέσματα σε μοναδική μηχανή

Αν κάθε διεργασία διαθέτει έναν χρόνο αποδέσμευσης

Το πρόβλημα είναι strongly NP-hard [LenstraKB77]

- $e/(e - 1) \simeq 1.58$ -προσεγγιστικός αλγόριθμος [ChekuriMNS97]
- PTAS [Afrati et al.99] ([Afrati, Bampis, Chekuri, [Karger](#), Kenyon, Khanna, Milis, Queyranne, Skutella, [Stein](#), and Sviridenko 1999])

Γνωστά αποτελέσματα σε μοναδική μηχανή

Αν κάθε διεργασία διαθέτει έναν χρόνο αποδέσμευσης

Το πρόβλημα είναι strongly NP-hard [LenstraKB77]

- $e/(e - 1) \simeq 1.58$ -προσεγγιστικός αλγόριθμος [ChekuriMNS97]
- PTAS [Afrati et al.99] ([Afrati, Bampis, Chekuri, [Karger](#), Kenyon, Khanna, Milis, Queyranne, Skutella, [Stein](#), and Sviridenko 1999])

Η [preemptive](#) εκδοχή επιλύεται βέλτιστα σε πολυωνυμικό χρόνο [Baker74]:

Κανόνας SRPT (Shortest Remaining Processing Time)

Εκτελείται όπως ο SPT με τη διαφορά ότι

- μία εκτελούμενη διεργασία διακόπτει την εκτέλεση της όταν μία διεργασία μικρότερου χρόνου επεξεργασίας αποδεσμευθεί.

Η κεντρική ιδέα του προσεγγιστικού σχήματος [Afrati et al.99]

Είσοδος: Ένα σύνολο J , n διεργασιών. Κάθε διεργασία $j \in J$ διαθέτει έναν χρόνο επεξεργασίας p_j και έναν χρόνο αποδέσμευσης r_j .

Αντικειμενική Συνάρτηση: Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης $\sum_{j=1}^n C_j$.

Κεντρική ιδέα

Απλοποιούμε το αρχικό στιγμιότυπο εισόδου εφαρμόζοντας διάφορους μετασχηματισμούς έτσι ώστε να χρονοδρομολογήσουμε το μεγαλύτερο μέρος των διεργασιών βάσει του **κανόνα SPT**. Οι εναπομείναντες διεργασίες θα είναι σταθερές στο πλήθος και επομένως μπορούμε **να απαριθμήσουμε** τις εφικτές χρονοδρομολογήσεις τους σε σταθερό χρόνο.

Μετασχηματισμοί εισόδου

Διακριτοποιούμε τις τιμές των χρόνων αποδέσμευσης και επεξεργασίας.

Λήμμα (Γεωμετρική Στρογγυλοποίηση)

Θεωρούμε μία θετική σταθερά ϵ , $0 < \epsilon < 1$. Με απώλεια $1 + \epsilon$ μπορούμε να υποθέσουμε ότι κάθε χρόνος επεξεργασίας και κάθε χρόνος αποδέσμευσης είναι ακέραια δύναμη του $1 + \epsilon$.

Μετασχηματισμοί εισόδου

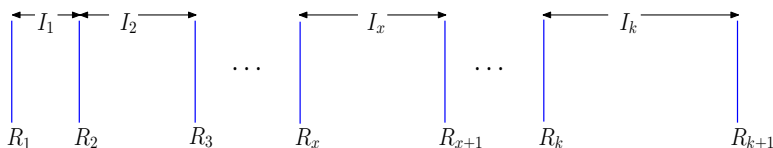
Διακριτοποιούμε τις τιμές των χρόνων αποδέσμευσης και επεξεργασίας.

Λήμμα (Γεωμετρική Στρογγυλοποίηση)

Θεωρούμε μία θετική σταθερά ϵ , $0 < \epsilon < 1$. Με απώλεια $1 + \epsilon$ μπορούμε να υποθέσουμε ότι κάθε χρόνος επεξεργασίας και κάθε χρόνος αποδέσμευσης είναι ακέραια δύναμη του $1 + \epsilon$.

- Κάθε χρόνος αποδέσμευσης θα είναι της μορφής $R_x = (1 + \epsilon)^x$, για κάποιο $x \in \mathbb{Z}_+^*$
- Διαμερίζουμε το χρονικό ορίζοντα $[0, \sum_j p_j + r_{max}]$ σε γεωμετρικά αυξανόμενα διαστήματα

Μετασχηματισμοί εισόδου



- $I_x = [R_x, R_{x+1}), x = 1, \dots, k$, όπου $R_{k+1} \leq \sum_j p_j + r_{max}$
- $R_{x+1} = (1 + \epsilon)R_x$
- $I_x = R_{x+1} - R_x = \epsilon R_x$

Μετασχηματισμοί εισόδου

Θέλουμε οι χρόνοι επεξεργασίας των διεργασιών να μην είναι αυθαίρετα μεγάλοι σε σύγκριση με το μήκος του διαστήματος στο οποίο αποδεσμεύονται.

Λήμμα

Με απώλεια $1 + \epsilon$ μπορούμε να εξασφαλίσουμε ότι $r_j \geq \epsilon p_j$ για κάθε διεργασία j .

Μετασχηματισμοί εισόδου

Θέλουμε οι χρόνοι επεξεργασίας των διεργασιών να μην είναι αυθαίρετα μεγάλοι σε σύγκριση με το μήκος του διαστήματος στο οποίο αποδεσμεύονται.

Λήμμα

Με απώλεια $1 + \epsilon$ μπορούμε να εξασφαλίσουμε ότι $r_j \geq \epsilon p_j$ για κάθε διεργασία j .

Πόρισμα

Κατά την εκτέλεσή της, κάθε διεργασία διασχίζει το πολύ $s = \lceil \log_{1+\epsilon}(1 + \frac{1}{\epsilon}) \rceil$ διαστήματα.

Εφαρμογή του κανόνα SPT

Ομαδοποιούμε τις διεργασίες σε

- μικρές: $p_j \leq \epsilon I_x$ και
- μεγάλες: $p_j > \epsilon I_x$, όπου I_x είναι το διάστημα στο οποίο αποδεσμεύεται η j

Εφαρμογή του κανόνα SPT

Ομαδοποιούμε τις διεργασίες σε

- μικρές: $p_j \leq \epsilon I_x$ και
- μεγάλες: $p_j > \epsilon I_x$, όπου I_x είναι το διάστημα στο οποίο αποδεδμεύεται η j

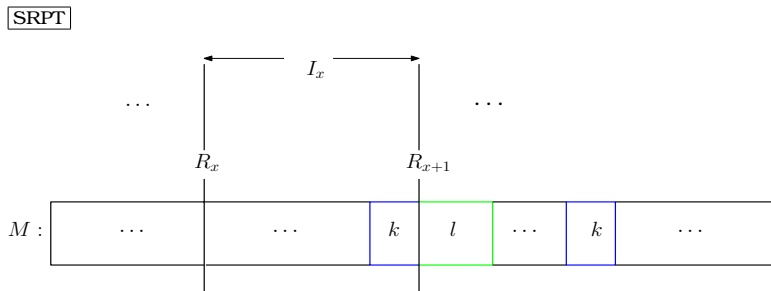
Λήμμα

Αν σε μία βέλτιστη χρονοδρομολόγηση κάθε διεργασία είναι μικρή, τότε ο κανόνας SPT δίνει μία $(1 + \epsilon)$ -προσεγγιστική λύση στο πρόβλημα.

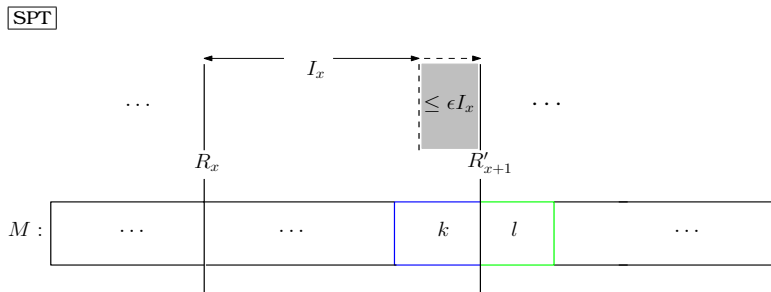
Σκιαγράφηση Απόδειξης.

Θεωρούμε μία βέλτιστη χρονοδρομολόγηση για την preemptive εκδοχή του προβλήματος όπου όλες οι διεργασίες είναι μικρές και με απώλεια $1 + \epsilon$ την μετατρέπουμε σε non-preemptive.

Εφαρμογή του κανόνα SPT



Εφαρμογή του κανόνα SPT



Μετατόπιση των μεγάλων διεργασιών

- OPT : συνολικός χρόνος ολοκλήρωσης της βέλτιστης χρονοδρομολόγησης
- Χρονικό κατώφλι: $t = \epsilon^7 \text{ OPT}$
- **Παρατήρηση:** το πολύ $\frac{1}{\epsilon^7}$ διεργασίες μπορούν να ολοκληρώσουν την εκτέλεση τους μετά το πέρας του χρόνου t

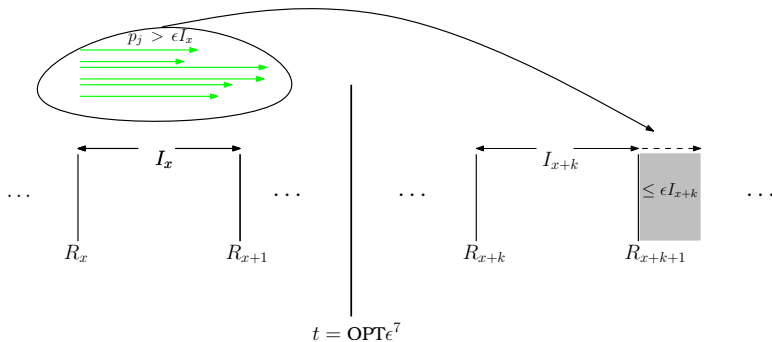
Μετατόπιση των μεγάλων διεργασιών

- OPT : συνολικός χρόνος ολοκλήρωσης της βέλτιστης χρονοδρομολόγησης
- Χρονικό κατώφλι: $t = \epsilon^7 \text{ OPT}$
- **Παρατήρηση:** το πολύ $\frac{1}{\epsilon^7}$ διεργασίες μπορούν να ολοκληρώσουν την εκτέλεση τους μετά το πέρας του χρόνου t

Ιδέα

Όλες οι μεγάλες διεργασίες που εκτελούνται **πριν τη χρονική στιγμή t** σε μία βέλτιστη χρονοδρομολόγηση, μπορούν να καθυστερήσουν την εκτέλεση-αποδέσμευσή τους, με μικρή απώλεια $(1 + 3\epsilon)$, έως ότου γίνουν μικρές.

Μετατόπιση των μεγάλων διεργασιών



Απαρίθμηση εναπομεινασών διεργασιών

- 1 Με απώλεια $1 + 3\epsilon$ μετασχηματίζουμε το στιγμιότυπο εισόδου ώστε οι μεγάλες διεργασίες να αποδεσμεύονται μετά το t
- 2 Με απώλεια $1 + \epsilon$ επαναχρονοδρομολογούμε όλες τις μικρές διεργασίες εφαρμόζοντας τον κανόνα SPT [◀ Λήμμα](#)
- 3 Η συνολική απώλεια μέχρι τη χρονική στιγμή t θα είναι το πολύ $(1 + \epsilon)(1 + 3\epsilon)$

Απαρίθμηση εναπομεινασών διεργασιών

- 1 Με απώλεια $1 + 3\epsilon$ μετασχηματίζουμε το στιγμιότυπο εισόδου ώστε οι μεγάλες διεργασίες να αποδεσμεύονται μετά το t
- 2 Με απώλεια $1 + \epsilon$ επαναχρονοδρομολογούμε όλες τις μικρές διεργασίες εφαρμόζοντας τον κανόνα SPT ◀ Λήμμα
- 3 Η συνολική απώλεια μέχρι τη χρονική στιγμή t θα είναι το πολύ $(1 + \epsilon)(1 + 3\epsilon)$

Πόρισμα

Μετά το t θα έχουν απομείνει να χρονοδρομολογηθούν το πολύ $(1 + \epsilon)(1 + 3\epsilon)^{\frac{1}{\epsilon}} \leq \frac{3}{\epsilon}$ διεργασίες.

Ο Αλγόριθμος [Karger, Stein]

Αλγόριθμος 1 Ένας προσεγγιστικός αλγόριθμος πολυωνυμικού χρόνου.

1. Εκτέλεσε τον κανόνα SPT έως ότου απομείνουν $\frac{3}{\epsilon^7}$ διεργασίες.
 2. Απαρίθμησε όλες τις εφικτές χρονοδρομολογήσεις των εναπομεινασών διεργασιών και επέλεξε την καλύτερη δυνατή.
-

Θεώρημα

Ο Αλγόριθμος 1 βρίσκει μία $(1 + \epsilon)$ -προσεγγιστική λύση στο πρόβλημα ελαχιστοποίησης του συνολικού χρόνου ολοκλήρωσης με δεδομένους χρόνους αποδέσμευσης, σε χρόνο $O(n \log n + (3/\epsilon^7)!).$ *Κατά συνέπεια, αποτελεί ένα PTAS για το πρόβλημα.*

Σκιαγράφηση

- 1 Θεωρία Χρονοδρομολόγησης
 - Προβλήματα χρονοδρομολόγησης
 - Τυποποίηση προβλημάτων
 - Υπολογιστική Πολυπλοκότητα
 - Περίληψη της μελέτης
- 2 Ελαχιστοποίηση του συνολικού χρόνου ολοκλήρωσης σε μοναδική μηχανή
 - Επισκόπηση του προβλήματος
 - Ένα προσεγγιστικό σχήμα πολυωνυμικού χρόνου
- 3 Χρονοδρομολόγηση με δεδομένες προθεσμίες
 - Συναρτήσεις απώλειας
 - Ένα δυναμικό πρόγραμμα ψευδοπολυωνυμικού χρόνου
 - Εφαρμογές του δυναμικού προγράμματος

Εισαγωγή προθεσμιών

Παρατήρηση

Τίθενται περιορισμοί όσον αφορά το χρόνο ολοκλήρωσης κάθε διεργασίας ατομικά.

Εισαγωγή προθεσμιών

Παρατήρηση

Τίθενται περιορισμοί όσον αφορά το χρόνο ολοκλήρωσης κάθε διεργασίας ατομικά.

Συναρτήσεις απώλειας

Κάθε διεργασία j έχει μία προθεσμία d_j και μία συνάρτηση απώλειας που εκφράζει το κόστος από την ολοκλήρωση της j εκτός της προθεσμίας της.

Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη καθυστέρηση

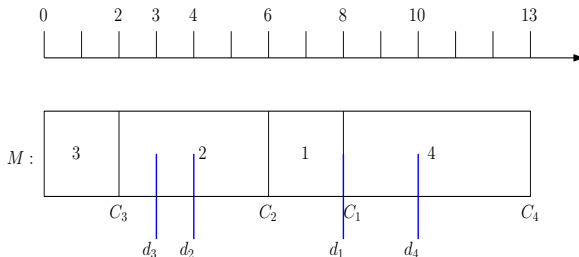
- $\sum_{j=1}^n w_j T_j$, όπου $T_j = \max \{C_j - d_j, 0\}$ η καθυστέρηση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.
- $\sum_{j=1}^n w_j T_j$, $w_j = 1$ για κάθε διεργασία j (συνολική καθυστέρηση).

Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη καθυστέρηση

- $\sum_{j=1}^n w_j T_j$, όπου $T_j = \max \{C_j - d_j, 0\}$ η καθυστέρηση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.
- $\sum_{j=1}^n w_j T_j$, $w_j = 1$ για κάθε διεργασία j (συνολική καθυστέρηση).

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48

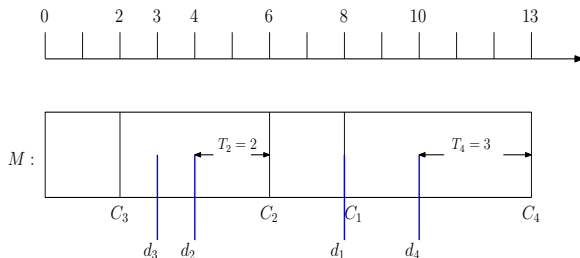


Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη καθυστέρηση

- $\sum_{j=1}^n w_j T_j$, όπου $T_j = \max \{C_j - d_j, 0\}$ η καθυστέρηση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.
- $\sum_{j=1}^n w_j T_j$, $w_j = 1$ για κάθε διεργασία j (συνολική καθυστέρηση).

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48

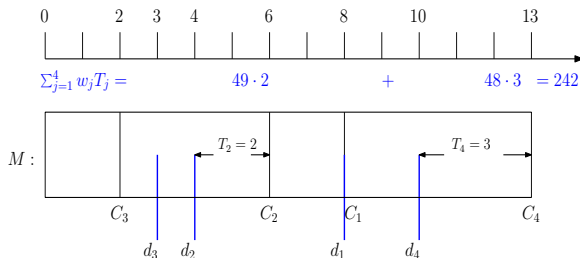


Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη καθυστέρηση

- $\sum_{j=1}^n w_j T_j$, όπου $T_j = \max \{C_j - d_j, 0\}$ η καθυστέρηση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.
- $\sum_{j=1}^n w_j T_j$, $w_j = 1$ για κάθε διεργασία j (συνολική καθυστέρηση).

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48



Ενδεικτικές συναρτήσεις απώλειας

Συνολικό βάρος των καθυστερημένων διεργασιών

$$\sum_{j=1}^n w_j U_j, \text{ όπου, για δεδομένη χρονοδρομολόγηση, } U_j = \begin{cases} 0, & \text{αν } C_j \leq d_j, \\ 1, & \text{αλλιώς.} \end{cases}$$

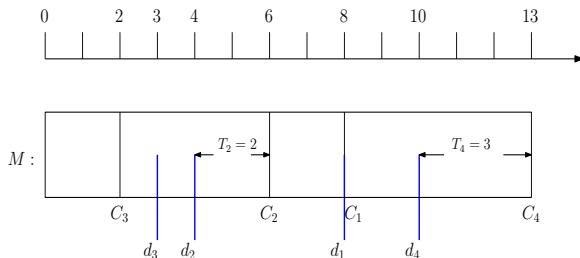
Ενδεικτικές συναρτήσεις απώλειας

Συνολικό βάρος των καθυστερημένων διεργασιών

$$\sum_{j=1}^n w_j U_j, \text{ όπου, για δεδομένη χρονοδρομολόγηση, } U_j = \begin{cases} 0, & \text{αν } C_j \leq d_j, \\ 1, & \text{αλλιώς.} \end{cases}$$

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48

► επιστροφή



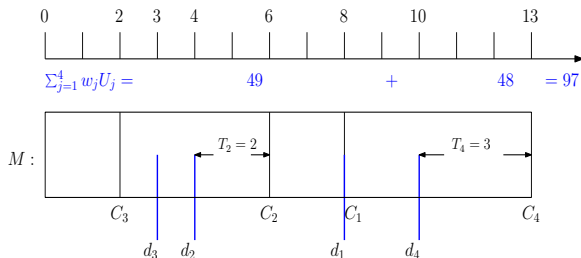
Ενδεικτικές συναρτήσεις απώλειας

Συνολικό βάρος των καθυστερημένων διεργασιών

$$\sum_{j=1}^n w_j U_j, \text{ όπου, για δεδομένη χρονοδρομολόγηση, } U_j = \begin{cases} 0, & \text{αν } C_j \leq d_j, \\ 1, & \text{αλλιώς.} \end{cases}$$

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48

► επιστροφή



Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη πρόωρη ολοκλήρωση

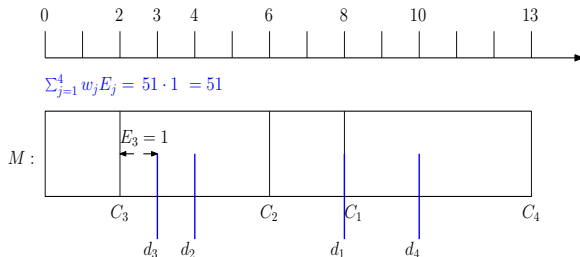
$\sum_{j=1}^n w_j E_j$, όπου $E_j = \max \{d_j - C_j, 0\}$ η πρόωρη ολοκλήρωση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.

Ενδεικτικές συναρτήσεις απώλειας

Συνολική βεβαρημένη πρόωρη ολοκλήρωση

$\sum_{j=1}^n w_j E_j$, όπου $E_j = \max \{d_j - C_j, 0\}$ η πρόωρη ολοκλήρωση μίας διεργασίας j σε δεδομένη χρονοδρομολόγηση.

j	p_j	d_j	w_j
1	2	8	50
2	4	4	49
3	2	3	51
4	5	10	48



Ένα γενικό πρόβλημα ελαχιστοποίησης της συνολικής απώλειας σε μοναδική μηχανή [LawlerM69]

Διαθέτουμε n διεργασίες οι οποίες πρόκειται να χρονοδρομολογηθούν διαδοχικά, **βάσει μίας προκαθορισμένης διάταξης**, $1, 2, \dots, n$. Κάθε διεργασία j μπορεί να χρονοδρομολογηθεί με δύο διαφορετικούς τρόπους:

- 1 Η επεξεργασία της j διαρκεί a_j μονάδες χρόνου, ενώ η απώλεια που δημιουργείται κατά την ολοκλήρωση της j τη χρονική στιγμή t , ισούται με $I_j^{(1)}(t)$
- 2 Η επεξεργασία της j διαρκεί b_j μονάδες χρόνου και η αντίστοιχη απώλεια ισούται με $I_j^{(2)}(t)$

Ποιά ανάθεση διαφορετικών τρόπων πρέπει να κάνουμε στις n διεργασίες έτσι ώστε να ελαχιστοποιήσουμε τη συνολική απώλεια της χρονοδρομολόγησης;

Αλγόριθμος δυναμικού προγραμματισμού

- $f(j, t)$ = η ελάχιστη συνολική απώλεια για τη χρονοδρομολόγηση των πρώτων j διεργασιών, υπό τον περιορισμό ότι η διεργασία j ολοκληρώνεται το πολύ μέχρι τη χρονική στιγμή t

$$\begin{aligned}
 f(0, t) &= 0, & (t \geq 0), \\
 f(j, t) &= +\infty, & (j = 0, 1, \dots, n; t < 0), \\
 f(j, t) &= \min \left\{ \begin{array}{l} f(j, t-1), \\ l_j^{(1)}(t) + f(j-1, t-a_j), \\ l_j^{(2)}(t) + f(j-1, t-b_j) \end{array} \right\}, & (j = 1, \dots, n; t \geq 0).
 \end{aligned}$$

Αλγόριθμος δυναμικού προγραμματισμού

- $f(j, t)$ = η ελάχιστη συνολική απώλεια για τη χρονοδρομολόγηση των πρώτων j διεργασιών, υπό τον περιορισμό ότι η διεργασία j ολοκληρώνεται το πολύ μέχρι τη χρονική στιγμή t

$$\begin{aligned}
 f(0, t) &= 0, & (t \geq 0), \\
 f(j, t) &= +\infty, & (j = 0, 1, \dots, n; t < 0), \\
 f(j, t) &= \min \left\{ \begin{array}{l} f(j, t - 1), \\ l_j^{(1)}(t) + f(j - 1, t - a_j), \\ l_j^{(2)}(t) + f(j - 1, t - b_j) \end{array} \right\}, & (j = 1, \dots, n; t \geq 0).
 \end{aligned}$$

- Πολυπλοκότητα χρόνου: $O(nT)$, όπου T σταθερά ($T = \sum_{j=1}^n \max\{a_j, b_j\}$)

Ελαχιστοποίηση του συνολικού βάρους των καθυστερημένων διεργασιών σε μοναδική μηχανή

Είσοδος:

Ένα σύνολο J , n διεργασιών. Κάθε διεργασία $j \in J$ διαθέτει έναν χρόνο επεξεργασίας p_j , μία προθεσμία d_j και ένα βάρος w_j .

Στόχος:

Η εύρεση μίας διάταξης χρονοδρομολόγησης που ελαχιστοποιεί την ποσότητα $\sum_{j=1}^n w_j U_j$. [◀ βλ. συνάρτηση](#)

Ελαχιστοποίηση του συνολικού βάρους των καθυστερημένων διεργασιών σε μοναδική μηχανή

Είσοδος: Ένα σύνολο J , n διεργασιών. Κάθε διεργασία $j \in J$ διαθέτει έναν χρόνο επεξεργασίας p_j , μία προθεσμία d_j και ένα βάρος w_j .

Στόχος: Η εύρεση μίας διάταξης χρονοδρομολόγησης που ελαχιστοποιεί την ποσότητα $\sum_{j=1}^n w_j U_j$.

NP-hard υπό τη συνήθη έννοια [Karp72]

Συνάφεια με το γενικό πρόβλημα

Διαθέτουμε n διεργασίες οι οποίες πρόκειται να χρονοδρομολογηθούν σε μία μηχανή, **βάσει προκαθορισμένης διάταξης**, με απώλειες

$$\begin{aligned} I_j^{(1)}(t) &= 0, & \text{αν } t \leq d_j, \\ I_j^{(2)}(t) &= w_j, & \text{αλλιώς,} \end{aligned} \quad \text{ή ισοδύναμα} \quad I_j(t) = \begin{cases} 0, & \text{αν } t \leq d_j, \\ w_j, & \text{αλλιώς.} \end{cases}$$

Για δεδομένη χρονοδρομολόγηση κατά την οποία μία διεργασία j ολοκληρώνει την εκτέλεσή της σε χρόνο C_j , θα ισχύει ότι $I_j(C_j) = w_j U_j$

Συνάφεια με το γενικό πρόβλημα

Διαθέτουμε n διεργασίες οι οποίες πρόκειται να χρονοδρομολογηθούν σε μία μηχανή, **βάσει προκαθορισμένης διάταξης**, με απώλειες

$$\begin{aligned} I_j^{(1)}(t) &= 0, & \text{αν } t \leq d_j, \\ I_j^{(2)}(t) &= w_j, & \text{αλλιώς,} \end{aligned} \quad \text{ή ισοδύναμα} \quad I_j(t) = \begin{cases} 0, & \text{αν } t \leq d_j, \\ w_j, & \text{αλλιώς.} \end{cases}$$

Για δεδομένη χρονοδρομολόγηση κατά την οποία μία διεργασία j ολοκληρώνει την εκτέλεσή της σε χρόνο C_j , θα ισχύει ότι $I_j(C_j) = w_j U_j$

Ερώτημα

Με ποιά σειρά πρέπει να χρονοδρομολογήσουμε τις διεργασίες έτσι ώστε να ελαχιστοποιήσουμε τη συνολική απώλεια $\sum_{j=1}^n I_j(C_j)$;

Κριτήριο βελτιστοποίησης

Παρατήρηση [Moore68, Rothkopf64]

Σε μία υποτιθέμενη βέλτιστη διάταξη οι διεργασίες μπορούν να διαχωριστούν σε δύο διαφορετικές ομάδες: η πρώτη περιέχει τις πρόωρες διεργασίες σε EDD διάταξη και η δεύτερη τις καθυστερημένες διεργασίες σε αυθαίρετη διάταξη. Οι διεργασίες της πρώτης ομάδας προηγούνται όλων των διεργασιών της δεύτερης.

Κανόνας EDD (Earliest Due Date) [Jackson55, Smith56]

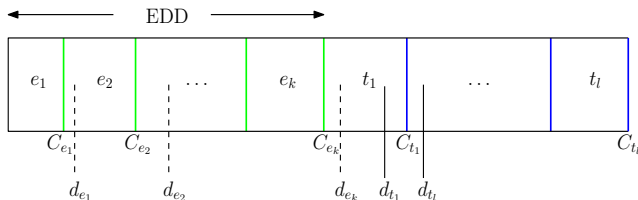
Οι διεργασίες χρονοδρομολογούνται σε αύξουσα διάταξη ως προς τις προθεσμίες τους:

$$d_1 \leq d_2 \leq \dots \leq d_n.$$

Κριτήριο βελτιστοποίησης

Παρατήρηση [Moore68, Rothkopf64]

Σε μία υποτιθέμενη βέλτιστη διάταξη οι διεργασίες μπορούν να διαχωριστούν σε δύο διαφορετικές ομάδες: η πρώτη περιέχει τις πρόωρες διεργασίες σε EDD διάταξη και η δεύτερη τις καθυστερημένες διεργασίες σε αυθαίρετη διάταξη. Οι διεργασίες της πρώτης ομάδας προηγούνται όλων των διεργασιών της δεύτερης.



Εφαρμογή του δυναμικού προγράμματος

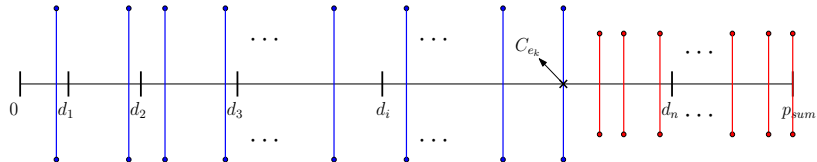
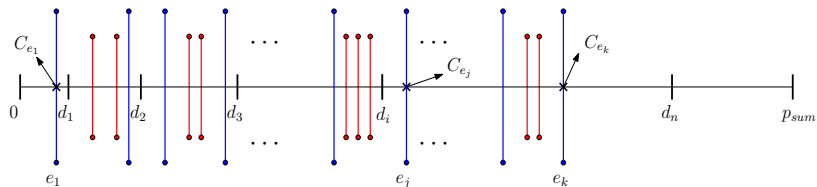
θέτουμε

$$\begin{aligned} a_j &= p_j, & l_j^{(1)}(t) &= 0, \\ b_j &= 0, & l_j^{(2)}(t) &= w_j. \end{aligned}$$

Ισοδύναμη μορφή του δυναμικού προγράμματος

$$\begin{aligned} f(0, t) &= 0, & (t \geq 0), \\ f(j, t) &= +\infty, & (j = 0, 1, \dots, n; t < 0), \\ f(j, t) &= \min \left\{ \begin{array}{l} f(j, t-1), \\ f(j-1, t-p_j), \\ w_j + f(j-1, t) \end{array} \right\}, & (j = 1, \dots, n; t \geq 0). \end{aligned}$$

Πώς εξάγουμε μία βέλτιστη διάταξη χρονοδρομολόγησης;



Επιπλέον εφαρμογές

Εύρεση βέλτιστων ψευδοπολυωνυμικών αλγορίθμων [LawlerM69]

- Ελαχιστοποίηση της συνολικής βεβαρημένης καθυστέρησης σε μοναδική μηχανή, όταν οι διεργασίες διαθέτουν μία κοινή προθεσμία
- Μεγιστοποίηση της συνολικής βεβαρημένης πρόωρης ολοκλήρωσης σε μοναδική μηχανή

Επιπλέον εφαρμογές

Εύρεση βέλτιστων ψευδοπολυωνυμικών αλγορίθμων [LawlerM69]

- Ελαχιστοποίηση της συνολικής βεβαρημένης καθυστέρησης σε μοναδική μηχανή, όταν οι διεργασίες διαθέτουν μία κοινή προθεσμία
- Μεγιστοποίηση της συνολικής βεβαρημένης πρόωρης ολοκλήρωσης σε μοναδική μηχανή

Με μικρές τροποποιήσεις χρησιμοποιούνται στο σχεδιασμό FPTAS.

Ανοιχτά προβλήματα

Ελαχιστοποίηση του makespan σε γενικό αριθμό παράλληλων μη σχετιζόμενων μηχανών

Γνωστά αποτελέσματα: 2-προσεγγιστικός αλγόριθμος [LenstraST90].

Δεν μπορεί να προσεγγιστεί με παράγοντα μικρότερο από $3/2$,
εκτός και αν $P = NP$ [LenstraST90].

Ανοιχτό ερώτημα: Εύρεση προσέγγισης με παράγοντα $\rho \in (3/2, 2)$.

Ανοιχτά προβλήματα

Ελαχιστοποίηση του makespan σε γενικό αριθμό παράλληλων μη σχετιζόμενων μηχανών

Γνωστά αποτελέσματα: 2-προσεγγιστικός αλγόριθμος [LenstraST90].

Δεν μπορεί να προσεγγιστεί με παράγοντα μικρότερο από $3/2$,
εκτός και αν $P = NP$ [LenstraST90].

Ανοιχτό ερώτημα: Εύρεση προσέγγισης με παράγοντα $\rho \in (3/2, 2)$.

Ελαχιστοποίηση του μέσου βεβαρημένου χρόνου ολοκλήρωσης με δεδομένους περιορισμούς προτεραιότητας για τις διεργασίες

Γνωστά αποτελέσματα: 2-προσεγγιστικοί αλγόριθμοι [HallSSW96, ChekuriM99].

Ειδική περίπτωση του προβλήματος Vertex Cover [AmbühlM06].

Ανοιχτό ερώτημα: Βελτίωση του λόγου προσέγγισης, APX-hardness.

Ανοιχτά προβλήματα

Ελαχιστοποίηση του makespan σε γενικό αριθμό παράλληλων μη σχετιζόμενων μηχανών

Γνωστά αποτελέσματα: 2-προσεγγιστικός αλγόριθμος [LenstraST90].

Δεν μπορεί να προσεγγιστεί με παράγοντα μικρότερο από $3/2$,
εκτός και αν $P = NP$ [LenstraST90].

Ανοιχτό ερώτημα: Εύρεση προσέγγισης με παράγοντα $\rho \in (3/2, 2)$.

Ελαχιστοποίηση του μέσου βεβαρημένου χρόνου ολοκλήρωσης με δεδομένους περιορισμούς προτεραιότητας για τις διεργασίες

Γνωστά αποτελέσματα: 2-προσεγγιστικοί αλγόριθμοι [HallSSW96, ChekuriM99].

Ειδική περίπτωση του προβλήματος Vertex Cover [AmbühlM06].

Ανοιχτό ερώτημα: Βελτίωση του λόγου προσέγγισης, APX-hardness.

Ελαχιστοποίηση της συνολικής βεβαρημένης καθυστέρησης

Γνωστά αποτελέσματα: $(n - 1)$ -προσεγγιστικός αλγόριθμος [ChengNYL05].

Ανοιχτό ερώτημα: Βελτίωση του λόγου προσέγγισης.

Ενδεικτικές Αναφορές



F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Millis, M. Queyranne, M. Skutella, C. Stein, and M. Sviridenko.

Approximation schemes for minimizing average weighted completion time with release dates.

In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 32–43, 1999.



D. S. Hochbaum and D. B. Shmoys.

Using dual approximation algorithms for scheduling problems: theoretical and practical results.

Journal of the ACM, 34:144–162, 1987.



E. L. Lawler.

A “pseudopolynomial” algorithm for sequencing jobs to minimize total tardiness.

Annals of Discrete Mathematics, 1:331–342, 1977.



E. L. Lawler.

A fully polynomial approximation scheme for the total tardiness problem.

Operations Research Letters, 1:207–208, 1982.



E. L. Lawler and J. M. Moore.

A functional equation and its application to resource allocation and sequencing problems.

Management Science, 16:77–84, 1969.



P. Schuurman and G. J. Woeginger.

Approximation schemes—a tutorial.

START Project Woe-65, TU Graz, January 2001.

Ευχαριστώ για την προσοχή σας!