

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΟΙΚΟΝΟΜΙΚΗ ΚΑΙ ΔΙΟΙΚΗΣΗ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ
(ΠΜΣ-Ο.ΔΙ.Μ.)

Αλγόριθμοι χρονοπρογραμματισμού σε ντετερμινιστικό περιβάλλον

Κουμή Αναστασία

Επιβλέπων καθηγητής:
Βιδάλης Μιχαήλ

Είμαι συγγραφέας αυτής της Μεταπτυχιακής Διπλωματικής Εργασίας και κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων ή ιδεών, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά για τη συγκεκριμένη μεταπτυχιακή διπλωματική εργασία

Περιεχόμενα

Εισαγωγή.....	5
Κεφάλαιο 1.....	11
Η λειτουργία του χρονοπρογραμματισμού σε μια επιχείρηση.....	11
1.1. Ο Χρονοπρογραμματισμός στη Παραγωγή.....	12
1.2. Χρονοπρογραμματισμός σε υπηρεσίες	14
Κεφάλαιο 2.....	14
Ντετερμινιστικός χρονοπρογραμματισμός.....	14
2.1 Δομή και συμβολισμός.....	15
Κεφάλαιο 3.....	22
Μοντέλο Μεμονωμένης Μηχανής (Ντετερμινιστικό)	22
3.1 Το πρόβλημα του συνολικού σταθμισμένου χρόνου ολοκλήρωσης.....	23
3.1.1 Θεώρημα: Ο κανόνας WSPT είναι βέλτιστος για $1 \parallel wjCj$	23
3.1.2 Αλγόριθμος (Συνολικός σταθμισμένος χρόνος ολοκλήρωσης και αλυσίδες)	23
3.1.3 Παράδειγμα (Συνολικός σταθμισμένος χρόνος ολοκλήρωσης και αλυσίδες)	23
3.2 Το πρόβλημα της μέγιστης καθυστέρησης.....	24
3.2.1 Παράδειγμα (Ελαχιστοποίηση του μέγιστου κόστους)	24
3.3 Το πρόβλημα του αριθμού των καθυστερημένων εργασιών.....	25
3.3.1 Αλγόριθμος: (Ελαχιστοποίηση του αριθμού των καθυστερημένων εργασιών)	26
3.3.2 Θεώρημα: Ο αλγόριθμος δίνει ένα βέλτιστο χρονοδιάγραμμα για $1 \parallel \sum j$.	26
3.3.3 Παράδειγμα (Ελαχιστοποίηση του αριθμού των καθυστερημένων εργασιών)	26
3.4 Το πρόβλημα της συνολικής βραδύτερης περάτωσης - δυναμικός προγραμματισμός	27
3.4.1 Λήμμα.....	27
3.4.2 Λήμμα.....	27
3.4.3 Λήμμα.....	27
3.4.4 Παράδειγμα (ελαχιστοποίηση της συνολικής βραδύτερης περάτωσης) ...	27
3.5 Το πρόβλημα της συνολικής βραδύτερης περάτωσης - ένα σύστημα προσέγγισης.....	28
3.5.1 Παράδειγμα (FPTAS ελαχιστοποίηση της συνολικής καθυστέρησης).....	29
3.6 Το πρόβλημα της συνολικής σταθμισμένης απόκλισης	29

3.6.1	Λήμμα: Αν υπάρχουν δύο εργασίες j και k με $d_j \leq d_k$, $p_j \leq p_k$ και $w_j \geq w_k$, τότε υπάρχει μια βέλτιστη ακολουθία στην οποία η εργασία j εμφανίζεται πριν από την εργασία k .	29
3.6.2	Παράδειγμα (Ελαχιστοποίηση της συνολικής σταθμισμένης απόκλισης).	29
3.7	Σχόλια κεφαλαίου	31
Κεφάλαιο 4		32
	Μοντέλα παράλληλων μηχανών (ντετερμινιστικά)	32
4.1	Το πρόβλημα του Makespan χωρίς προεκχωρήσεις.	33
4.1.1	Θεώρημα: Για $P_m \parallel C_{max}$	34
4.1.2	Παράδειγμα (Παράδειγμα χειρότερης περίπτωσης LPT)	34
4.1.3	Αλγόριθμος: (Ελαχιστοποίηση του makespan ενός project)	35
4.1.4	Παράδειγμα (Ελαχιστοποίηση του makespan ενός project)	36
4.2	Το πρόβλημα του Makespan με προεκχωρήσεις	37
4.2.1	Παράδειγμα (Σχηματισμός LP για ελαχιστοποίηση του Makespan με προεκχωρήσεις)	38
4.3	Το πρόβλημα του συνολικού χρόνου ολοκλήρωσης χωρίς προεκχωρήσεις	39
4.3.1	Θεώρημα: ο κανόνας SPT είναι βέλτιστος για το $P_m \parallel C_j$	39
4.3.2	Παράδειγμα (Εφαρμογή του κανόνα WSPT)	39
4.4	Αντικειμενικές σχετιζόμενες με τις ημερομηνίες παράδοσης	40
4.4.1	Παράδειγμα (Ελαχιστοποίηση της μέγιστης καθυστέρησης με προεκχωρήσεις)	41
4.5	Σχόλια κεφαλαίου	41
Κεφάλαιο 5		42
	Στοχαστικός Χρονοπρογραμματισμός	42
5.1	Δομή και συμβολισμός	42
Κεφάλαιο 6		43
	Μοντέλο Μεμονωμένης Μηχανής (Στοχαστικό)	43
6.1	Το πρόβλημα των αυθαίρετων διανομών χωρίς προεκχωρήσεις	43
6.1.1	Θεώρημα	44
6.1.2	Παράδειγμα (Βέλτιστη πολιτική με τυχαίες μεταβλητές που είναι DCR)	44
6.1.3	Θεώρημα	44
6.1.4	Παράδειγμα (Εφαρμογή του κανόνα EDD)	44
Κεφάλαιο 7		45
	Μοντέλα παράλληλων μηχανών (στοχαστικά)	45
7.1	Το πρόβλημα του makespan και του συνολικού χρόνου ολοκλήρωσης χωρίς προεκχωρήσεις	45
7.1.1	Λήμμα	47
7.1.2	Θεώρημα	47
7.1.3	Θεώρημα	47

7.1.4	Λήμμα.....	47
7.1.5	Παράδειγμα (Αντίθεση προς το βέλτιστο της LEPT με αυθαίρετους χρόνους επεξεργασίας)	47
7.2	Το πρόβλημα της Makespan και του συνολικού χρόνου ολοκλήρωσης με τις προμήθειες.....	48
7.2.1	Θεώρημα	48
7.2.2	Θεώρημα	48
7.2.3	Θεώρημα	48
7.2.4	Παράδειγμα (Αντίκτυπος προς βέλτιστο κανόνα CP με τρεις μηχανές).....	48
	Επίλογος.....	49
	Βιβλιογραφία	49

Εισαγωγή

Ο ρόλος της Διοικητικής Επιστήμης στη νέα επιχειρηματική πραγματικότητα είναι η λήψη και υλοποίηση αποφάσεων σε επιχειρήσεις και οργανισμούς. Δίνει στο μάνατζερ την ικανότητα να αποφασίζει γρήγορα και εμπειριστατωμένα ούτως ώστε να μπορεί να αντιμετωπίσει ένα φάσμα προβλημάτων στην επιχείρηση. Αυτό το πετυχαίνει παρέχοντάς του μεθοδολογίες για την ανάλυση των προβλημάτων, τη λήψη και υλοποίηση των σχετικών αποφάσεων καθώς επίσης και τεχνικές για την επίλυσή τους.

Στο σημερινό περίπλοκο επιχειρηματικό περιβάλλον, η κατάλληλη μεθοδολογία αντιμετώπισης ενός σύνθετου προβλήματος μας βοηθάει να καταλήξουμε σε μια αποτελεσματική, υλοποιήσιμη αλλά και βιώσιμη πρόταση σε ένα συγκεκριμένο χρονικό διάστημα.

Τα στάδια που ακολουθεί η μεθοδολογία της Διοικητικής Επιστήμης είναι τα εξής 5:

1. Εντοπισμός και ανάλυση του προβλήματος (ή της ευκαιρίας).

Ο σωστός και έγκαιρος εντοπισμός ενός προβλήματος ή μίας ευκαιρίας κατέχει σημαντικό ρόλο στη λήψη μιας καλής απόφασης. Οι διαδικασίες πληροφόρησης της επιχείρησης και η εμπειρία των στελεχών αποτελούν τις πηγές από τις οποίες προκύπτει ο έγκαιρος αυτός εντοπισμός. Αυτές οι πηγές έχουν να κάνουν με αυτό που σήμερα αποκαλείται Διαχείριση της Επιχειρησιακής Γνώσης. Η Διαχείριση της Επιχειρησιακής γνώσης αφορά την εξωτερικεύσιμη αλλά και την άρρητη γνώση. Εξωτερικεύσιμη γνώση είναι αυτή που αποκτάται εύκολα μέσα από αρχεία, αναφορές κλπ. Ενώ η άρρητη είναι αυτή που αποκτάται από άυλους παράγοντες όπως η εμπειρία και η διαίσθηση. Η αξιοποίηση αυτών των γνώσεων με στόχο τον εντοπισμό προβλημάτων αποτελεί το πρώτο στάδιο εφαρμογής της Διοικητικής Επιστήμης. Ένας ακόμη παράγοντας που διευκολύνει τον εντοπισμό προβλημάτων ή ευκαιριών είναι το διαδίκτυο μέσω της ενημέρωσης και της επικοινωνίας που παρέχει διευκολύνοντας έτσι τη διάχυση της γνώσης.

Στη συνέχεια ένα πολύ σημαντικό κομμάτι σε αυτό το στάδιο είναι η αποτύπωση και η ανάλυση του συστήματος. Σε αυτή τη φάση προσδιορίζονται οι αιτίες του προβλήματος. Μερικά από τα βήματα που συνήθως χρησιμοποιούνται για την ανάλυση και την καταγραφή ενός συστήματος είναι:

- Συνεντεύξεις,
- Έρευνα μέσω ερωτηματολογίων,
- Καταγραφή της οργάνωσης και των διαδικασιών,
- Συλλογή και ανάλυση στοιχείων.

2. Διατύπωση των στόχων

Στο δεύτερο στάδιο της μεθοδολογίας της Διοικητικής Επιστήμης ανήκει η διατύπωση των στόχων. Οι στόχοι μιας επιχείρησης πρέπει να είναι πολυδιάστατοι, δηλαδή να περιλαμβάνουν περισσότερα κριτήρια του ενός. Η μέθοδος της Ισορροπημένης Στοχοθεσίας περιλαμβάνει 4 κατηγορίες κριτηρίων. Οι κατηγορίες αυτές είναι:

- Χρηματοοικονομικά μεγέθη,
- Πελάτης,
- Εσωτερική οργάνωση και αποτελεσματικότητα διαδικασιών,
- Οργανωσιακή γνώση

3. Σχεδιασμός του συστήματος,

Έχοντας θέσει τους στόχους προχωράμε στο τρίτο στάδιο, το σχεδιασμό νέου συστήματος. Το στάδιο αυτό αποτελεί το κύριο αντικείμενο της Επιχειρησιακής Έρευνας. Σε αυτή τη φάση ο αναλυτής προκειμένου να σχεδιάσει μια αποτελεσματική λύση και να επιλέξει την καλύτερη στρατηγική πρέπει να μπορέσει να μελετήσει και να αναλύσει την επίδραση διάφορων στρατηγικών στους στόχους που τέθηκαν. Επειδή όμως το σημερινό επιχειρηματικό περιβάλλον χαρακτηρίζεται από αβεβαιότητα, πολλές φορές οι αρχικές μας εκτιμήσεις μπορεί να μην ανταποκρίνονται στη μεταβολή των δεδομένων και να χρειάζονται επαναπροσδιορισμό. Θα πρέπει λοιπόν, ο μάνατζερ προτού υλοποιήσει την στρατηγική που έχει επιλέξει, να έχει μελετήσει το μοντέλο που χρησιμοποίησε για την υπόδειξη της σωστής στρατηγικής κάτω από εναλλακτικά σενάρια, έτσι ώστε να γνωρίζει τις επιπτώσεις που θα είχε στη στρατηγική του κάποια αλλαγή στο περιβάλλον.

4. Υλοποίηση της λύσης,

Το στάδιο της υλοποίησης δεν είναι απλή υπόθεση. Μπορεί να προϋποθέτει αλλαγές σε διάφορους τομείς όπως στη στρατηγική του οργανισμού ή τον σχεδιασμό των κατάλληλων συστημάτων που θα στηρίζουν την πρόταση. Αλλαγές στο εξωτερικό περιβάλλον, παραλήψεις ή λάθη του αναλυτή, αντιδράσεις από τα εμπλεκόμενα στελέχη ή τμήματα του οργανισμού είναι μερικές από τις δυσκολίες που μπορεί να προκύψουν στην υλοποίηση των αλλαγών.

5. Παρακολούθηση και βελτίωση της λειτουργίας.

Τέλος, το πέμπτο στάδιο έχει στόχο τον εντοπισμό και τη διόρθωση αδυναμιών στο σύστημα. Αυτό επιτυγχάνεται με την παρακολούθηση της απόδοσης μέσω συστημάτων πληροφορικής του οργανισμού.

Η Διοικητική Επιστήμη χρησιμοποιεί μοντέλα στη λήψη αποφάσεων. Με την έννοια μοντέλο εννοούμε μία απεικόνιση του συστήματος που θα αναλύσουμε. Αυτό μας βοηθάει στη λήψη μιας περίπλοκης απόφασης. Οι αποφάσεις χωρίζονται σε 2 κατηγορίες, στις βασικές αποφάσεις και στις αποφάσεις κλάδου εφαρμογής. Οι βασικές αποφάσεις είναι αποφάσεις που μπορούν να εμφανιστούν σε διάφορα τμήματα της επιχείρησης. Η Διοικητική Επιστήμη εφοδιάζει την επιχείρηση με μοντέλα που μας βοηθούν στη λήψη αυτών των αποφάσεων. Μερικές αποφάσεις και μοντέλα που χρησιμοποιούνται σε μια επιχείρηση είναι:

➤ **Η κατανομή επιχειρησιακών πόρων.**

Η πιο συνηθισμένη μεθοδολογία για την κατανομή επιχειρησιακών πόρων στις δραστηριότητες της επιχείρησης σύμφωνα με την στρατηγική της είναι ο Μαθηματικός Προγραμματισμός. Αυτός αποτελείται από μαθηματικές συναρτήσεις μιας αντικειμενικής συνάρτησης που είναι ο στόχος της επιχείρησης και από περιορισμούς που είναι οι περιορισμοί στο περιβάλλον. Οι κλάδοι του είναι ο γραμμικός προγραμματισμός, όπου οι συναρτήσεις πρέπει να είναι γραμμικές, ο ακέραιος προγραμματισμός, όπου οι μεταβλητές μπορούν να πάρουν μόνο ακέραιες τιμές και ο μη γραμμικός προγραμματισμός, όπου μερικές συναρτήσεις δεν είναι γραμμικές.

➤ **Η επιλογή ή ιεράρχηση εναλλακτικών με πολλαπλά κριτήρια.**

Η επιλογή μιας απόφασης δεν είναι πάντα αποτέλεσμα μιας παραμέτρου, πολλές φορές χρειάζεται να λάβουμε υπόψιν μας πολλά κριτήρια για την βέλτιστη απόφαση. Σε αυτήν την περίπτωση το πιο κατάλληλο μοντέλο είναι αυτό της πολυκριτήριας ανάλυσης. Μέσω της Αναλυτικής Μεθόδου Ιεράρχησης η πολυκριτήρια ανάλυση βοηθάει στην κατανόηση του προβλήματος, στην ιεράρχηση των επιλογών και κατά συνέπεια στην λήψη αποφάσεων. Στην ουσία αυτό που κάνει η Αναλυτική Μέθοδος Ιεράρχησης είναι η σύγκριση ανά δύο εναλλακτικών λύσεων σε σχέση με τα καθένα από τα κριτήρια και η σύγκριση της σπουδαιότητας των κριτηρίων μεταξύ τους.

➤ **Η χάραξη στρατηγικής υπό καθεστώς αβεβαιότητας.**

Η αβεβαιότητα αντιμετωπίζεται με την ανάλυση ευαισθησίας. Δύο είναι τα μοντέλα που ενσωματώνουν την αβεβαιότητα. Αυτά είναι τα Δέντρα Αποφάσεων και η Προσομοίωση. Τα Δέντρα Αποφάσεων είναι η πιο συνηθισμένη τεχνική σε αυτά τα προβλήματα. Είναι χρήσιμα όταν οι αποφάσεις και οι εναλλακτικές είναι αλληλοσυνδεόμενες και βοηθούν στην επιλογή στρατηγικής παρέχοντας προτεινόμενη λύση και ανάλυση ευαισθησίας. Εδώ η αναπαράσταση του προβλήματος γίνεται από γράφημα, όπου κάθε κλαδί αναπαριστά μία απόφαση ή ένα αποτέλεσμα τυχαίου γεγονότος.

➤ **Η διαχείριση επιχειρηματικού κινδύνου.**

Το άλλο μοντέλο που χρησιμοποιείται συχνά και ενσωματώνει την αβεβαιότητα είναι η Προσομοίωση. Η προσομοίωση είναι ένας κώδικας στον Η/Υ. Εδώ δεν υπάρχει πρόταση για μια στρατηγική αλλά ανάλυση και αξιολόγηση αυτής που προτείνει ο αναλυτής. Είναι μία αντιγραφή του πραγματικού συστήματος.

➤ **Η ανάλυση και η βελτίωση διαδικασιών.**

Αυτά τα προβλήματα αντιμετωπίζονται συνήθως με προσομοίωση. Άλλοι πιο σπάνιοι τρόποι είναι ο Ακέραιος προγραμματισμός και οι Γραμμές Αναμονής. Μία απόφαση μπορεί να αποτελεί συνδυασμό των παραπάνω άρα και συνδυασμό μοντέλων για την αντιμετώπιση του προβλήματος.

Οι αποφάσεις κλάδου εφαρμογής συγκεκριμενοποιούν τις βασικές αποφάσεις σε διάφορα τμήματα της επιχείρησης όπως το marketing, τη παραγωγή, τα χρηματοοικονομικά, τους ανθρώπινους πόρους κλπ. Κάθε μοντέλο προϋποθέτει τη χρήση ενός πακέτου στον Η/Υ. Ένα πακέτο μπορεί να βοηθήσει:

- Στην κατανόηση ενός περίπλοκου προβλήματος,
- Στην ανάλυση των δεδομένων,
- Στην ανάλυση ευαισθησίας,
- Στην παραγωγή αναφορών και γραφημάτων.

Η ανθρώπινη κρίση, η εμπειρία και η διαίσθηση δεν θα πρέπει σε καμία περίπτωση να υποκατασταθούν από τα πακέτα. Αυτό που κάνουν τα πακέτα είναι να μας βοηθούν να καταλήξουμε σε μία απόφαση μέσω μεγαλύτερης ανάλυσης και σύνθεσης πληροφοριών.

Η Διοικητική Επιστήμη σε μία σύγχρονη επιχείρηση, μπορεί να συνεισφέρει σε όλους τους επιχειρηματικούς κλάδους, τις λειτουργίες διοίκησης και λήψης αποφάσεων καθώς επίσης και τα περισσότερα οργανωτικά πεδία. Από την εφαρμογή της Διοικητικής Επιστήμης μπορούν να προκύψουν πολλά οφέλη. Μερικά από αυτά είναι:

- Αύξηση των πωλήσεων,
- Αύξηση των εσόδων,
- Μείωση του κόστους,
- Αύξηση της ταχύτητας ή της παραγωγής,
- Κέρδος μεγαλύτερης εκμετάλλευσης του εξοπλισμού,
- Διαχείριση και μείωση του κινδύνου,
- Καλύτερος έλεγχος και διοίκηση αλλαγών,
- Βελτίωση της ποιότητας.

Η παραγωγή, το ανθρώπινο δυναμικό, οι μεταφορές, ο σχεδιασμός εφοδιαστικής αλυσίδας, τα χρηματοοικονομικά, το marketing, οι πωλήσεις και η οργάνωση είναι επιχειρησιακές λειτουργίες όπου εφαρμόζεται η Διοικητική Επιστήμη.

Η Διοικητική Επιστήμη έχει σημαντικό ρόλο στη Κοινωνία της Πληροφορίας. Η συμβολή της αναμένεται να είναι αισθητή σε χώρους της οικονομίας του διαδικτύου όπως τη διοίκηση της εφοδιαστικής αλυσίδας, τη χρηματοοικονομική

διοίκηση, τα δίκτυα και τις μεταφορές αλλά και το marketing. Η Οι λόγοι που ενισχύουν τη σημασία της, αρχικά είναι ότι δημιουργεί αξία από τα συνεχώς αυξανόμενα δεδομένα. Οι οργανισμοί μέσω της Διοικητικής Επιστήμης, της στατιστικής και της τεχνολογίας μπορούν πιο εύκολα να κατανοήσουν τα δεδομένα τους και να τα μετατρέψουν σε γνώση. Ένας ακόμη λόγος είναι ότι μέσω των επιχειρησιακών μοντέλων, της μεθοδολογίας και των εργαλείων που προσφέρει στους οργανισμούς για την ανάλυση διαφόρων προβλημάτων, βοηθάει στην αντιμετώπιση της πολυπλοκότητάς τους. Επιπροσθέτως, διαθέτει εργαλεία για τη λήψη αποφάσεων και τη διαμόρφωση στρατηγικών οι οποίες μας βοηθούν στη διαχείριση του ρίσκου κατά συνέπεια και στην αντιμετώπιση της αβεβαιότητας. Επιπλέον, καθώς η γνώση αναγνωρίζεται όλο και περισσότερο ως κύριο ανταγωνιστικό στοιχείο μιας επιχείρησης και η Διαχείριση της Γνώσης γίνεται μία από τις σημαντικότερες προκλήσεις της, η Διοικητική Επιστήμη μέσω της μοντελοποίησης και της επιστημονικής ανάλυσης γίνεται ένα αξιόπιστο εργαλείο της επιχείρησης, για την ενίσχυση γνώσης και κατανόησης επιχειρηματικών ζητημάτων. Έπειτα, δεν θα μπορούσαμε να παραβλέψουμε ότι σε ένα επιχειρηματικό περιβάλλον που η καινοτομία είναι απαραίτητη, η Διοικητική Επιστήμη βοηθάει στον πειραματισμό καινοτομικών λύσεων χωρίς να ρισκάρει να ζημιωθεί η απόδοση της εταιρείας, το αποτέλεσμα κλπ. Τέλος, ένας από τους πιο σημαντικούς λόγους που ενισχύουν τη σημασία της είναι ότι επιτρέπει τη λήψη αποφάσεων σε γρήγορο χρόνο.

Συνοψίζοντας, η Διοικητική επιστήμη ασχολείται με την εφαρμογή μαθηματικών και μεθόδων ηλεκτρονικών υπολογιστών σε προβλήματα διαχείρισης. Ένας από τους βασικούς της τομείς είναι η διαχείριση λειτουργιών.

Ο τομέας διαχείρισης λειτουργιών εφαρμόζει τις βασικές μεθοδολογίες της επιστήμης της διοίκησης για την αντιμετώπιση των προβλημάτων ανάπτυξης, παραγωγής και παράδοσης αγαθών και υπηρεσιών. Η έμφαση δίνεται στον τρόπο συνδυασμού εννοιών, μοντέλων και δεδομένων για να βοηθήσουν τους διαχειριστές να αναπτύξουν καλύτερα συστήματα και να λάβουν καλύτερες αποφάσεις σχετικά με τις επιχειρήσεις.

Οι βασικοί τομείς έρευνας της διαχείρισης λειτουργιών περιλαμβάνουν:

- Διαχείριση εφοδιαστικής αλυσίδας,
- Συστήματα προσαρμογής, μορφοποίησης και διαμόρφωσης προϊόντων, δηλαδή πώς να παράγουμε προσαρμοσμένα προϊόντα με αποτελεσματικό τρόπο,
- Έλεγχος Παραγωγής και Σχεδιασμού,
- Σχεδιασμός συστημάτων παραγωγής.

Μία λειτουργία που χρησιμοποιείται στα συστήματα παραγωγής, για την επίτευξη των στόχων των επιχειρήσεων στο βέλτιστο χρόνο, είναι ο χρονοπρογραμματισμός.

Ο χρονοπρογραμματισμός είναι μια διαδικασία λήψης αποφάσεων που χρησιμοποιείται σε τακτική βάση σε πολλές βιομηχανίες παραγωγής και

υπηρεσιών. Ασχολείται με την κατανομή πόρων σε εργασίες σε δεδομένες χρονικές περιόδους και έχει ως σκοπό τη βελτιστοποίηση ενός ή περισσότερων στόχων.

Οι πόροι και οι εργασίες σε έναν οργανισμό μπορούν να λάβουν πολλές διαφορετικές μορφές. Οι πόροι μπορεί να είναι μηχανές σε συνεργείο, αεροδιάδρομοι σε αεροδρόμια, πληρώματα σε εργοτάξιο, μονάδες επεξεργασίας σε περιβάλλον πληροφορικής κλπ. Οι εργασίες μπορεί να είναι λειτουργίες σε διαδικασία παραγωγής, απογειώσεις και προσγειώσεις σε αεροδρόμιο, στάδια σε ένα κατασκευαστικό έργο, εκτελέσεις προγραμμάτων ηλεκτρονικών υπολογιστών κλπ. Κάθε εργασία μπορεί να έχει ένα συγκεκριμένο επίπεδο προτεραιότητας, μια συντομότερη δυνατή εκκίνηση και μια ημερομηνία λήξης. Οι στόχοι μπορούν επίσης να έχουν πολλές διαφορετικές μορφές. Ένας στόχος μπορεί να είναι η ελαχιστοποίηση του χρόνου ολοκλήρωσης του τελευταίου έργου και άλλος μπορεί να είναι η ελαχιστοποίηση του αριθμού των εργασιών που ολοκληρώθηκαν μετά τις αντίστοιχες ημερομηνίες λήξης.

Ο χρονοπρογραμματισμός, ως διαδικασία λήψης αποφάσεων, διαδραματίζει σημαντικό ρόλο στα περισσότερα συστήματα παραγωγής καθώς και στα περισσότερα περιβάλλοντα επεξεργασίας πληροφοριών. Είναι επίσης σημαντικό στις ρυθμίσεις μεταφοράς και διανομής και σε άλλους τύπους υπηρεσιών. Το ακόλουθο παράδειγμα απεικονίζει το ρόλο του χρονοπρογραμματισμού σε κεντρική μονάδα επεξεργασίας.

Παράδειγμα (Εργασίες χρονοπρογραμματισμού σε κεντρική μονάδα επεξεργασίας (CPU))

Μία από τις λειτουργίες ενός πολυλειτουργικού συστήματος υπολογιστή είναι ο προγραμματισμός του χρόνου που η CPU αφιερώνει στα διάφορα προγράμματα που πρέπει να εκτελεστούν. Οι ακριβείς χρόνοι επεξεργασίας συνήθως δεν είναι γνωστοί εκ των προτέρων. Ωστόσο, η κατανομή αυτών των τυχαίων χρόνων επεξεργασίας μπορεί να είναι γνωστή εκ των προτέρων, συμπεριλαμβανομένων των μέσων και των διακυμάνσεων τους. Επιπλέον, κάθε εργασία έχει συνήθως ένα συγκεκριμένο επίπεδο προτεραιότητας (το λειτουργικό σύστημα συνήθως επιτρέπει στους χρήστες να καθορίζουν το επίπεδο προτεραιότητας ή το βάρος κάθε εργασίας). Στην περίπτωση αυτή, ο στόχος είναι να ελαχιστοποιηθεί το αναμενόμενο άθροισμα των χρόνων ολοκλήρωσης όλων των εργασιών.

Προκειμένου να αποφευχθεί η κατάσταση όπου σχετικά μικρές εργασίες παραμένουν στο σύστημα για μεγάλο χρονικό διάστημα καθώς περιμένουν τις πολύ μεγαλύτερες εργασίες με υψηλότερη προτεραιότητα, το λειτουργικό σύστημα κόβει κάθε εργασία σε μικρά κομμάτια. Στη συνέχεια περιστρέφει αυτά τα κομμάτια στην CPU έτσι ώστε σε οποιοδήποτε δεδομένο χρονικό διάστημα, η CPU να ξοδεύει κάποιο χρόνο για κάθε εργασία. Με αυτόν τον τρόπο, αν τυχόν ο χρόνος επεξεργασίας μιας εργασίας είναι πολύ σύντομος, η εργασία θα είναι σε θέση να εγκαταλείψει το σύστημα σχετικά γρήγορα.

Μία διακοπή κατά την επεξεργασία μίας εργασίας συχνά ονομάζεται προεκχώρηση. Είναι προφανές ότι η βέλτιστη πολιτική σε τέτοια περιβάλλοντα χρησιμοποιεί σε σημαντικό βαθμό προεκχωρήσεις.

Οι επιπτώσεις που μπορεί να έχει ένα χρονοδιάγραμμα ως προς τους στόχους ενδιαφέροντος μπορεί να μην είναι άμεσα ξεκάθαρες. Στην πράξη, συχνά

αποδεικνύεται ότι η επιλογή του χρονοδιαγράμματος έχει σημαντικό αντίκτυπο στην απόδοση του συστήματος και ότι έχει νόημα να αφιερώσουμε λίγο χρόνο και προσπάθεια για να αναζητήσουμε ένα κατάλληλο χρονοδιάγραμμα. Ο χρονοπρογραμματισμός μπορεί να είναι δύσκολος τόσο από τεχνικής απόψεως όσο και εφαρμογής. Το είδος των δυσκολιών που εμφανίζονται στην τεχνική πλευρά είναι παρόμοιες με αυτές που υπάρχουν σε άλλες μορφές συνδυαστικής βελτιστοποίησης και στοχαστικής μοντελοποίησης. Οι δυσκολίες στην πλευρά εφαρμογής είναι εντελώς διαφορετικές. Μπορεί να εξαρτώνται από την ακρίβεια του μοντέλου που χρησιμοποιείται για την ανάλυση του πραγματικού προβλήματος προγραμματισμού και την αξιοπιστία των απαιτούμενων δεδομένων εισόδου.

Κεφάλαιο 1

Η λειτουργία του χρονοπρογραμματισμού σε μια επιχείρηση.

Η λειτουργία χρονοπρογραμματισμού σε ένα σύστημα παραγωγής ή σε μια υπηρεσία εξυπηρέτησης πρέπει να αλληλεπιδρά με πολλές άλλες λειτουργίες. Αυτές οι αλληλεπιδράσεις εξαρτώνται από το σύστημα και μπορεί να διαφέρουν σημαντικά από την μία κατάσταση στην άλλη. Συχνά λαμβάνουν χώρα σε ένα σύστημα πληροφοριών σε επίπεδο επιχείρησης.

Ένα σύγχρονο εργοστάσιο ή υπηρεσία εξυπηρέτησης έχει συχνά ένα περίπλοκο πληροφοριακό σύστημα που περιλαμβάνει κεντρικό υπολογιστή και βάση δεδομένων. Τα τοπικά δίκτυα προσωπικών υπολογιστών, σταθμών εργασίας και τερματικών εισόδου δεδομένων, τα οποία είναι συνδεδεμένα σε αυτόν τον κεντρικό υπολογιστή, μπορούν να χρησιμοποιηθούν είτε για την ανάκτηση δεδομένων από τη βάση δεδομένων είτε για την εισαγωγή νέων δεδομένων. Το λογισμικό που ελέγχει ένα τόσο περίπλοκο πληροφοριακό σύστημα αναφέρεται τυπικά ως σύστημα ERP (Enterprise Resource Planning). Ορισμένες εταιρείες λογισμικού ειδικεύονται στην ανάπτυξη τέτοιων συστημάτων, συμπεριλαμβανομένων των SAP, J.D. Edwards και PeopleSoft. Ένα τέτοιο σύστημα ERP διαδραματίζει το ρόλο μιας αυτοκινητοδρόμου πληροφόρησης που διασχίζει την επιχείρηση με συνδέσεις με συστήματα υποστήριξης αποφάσεων σε όλα τα οργανωτικά επίπεδα.

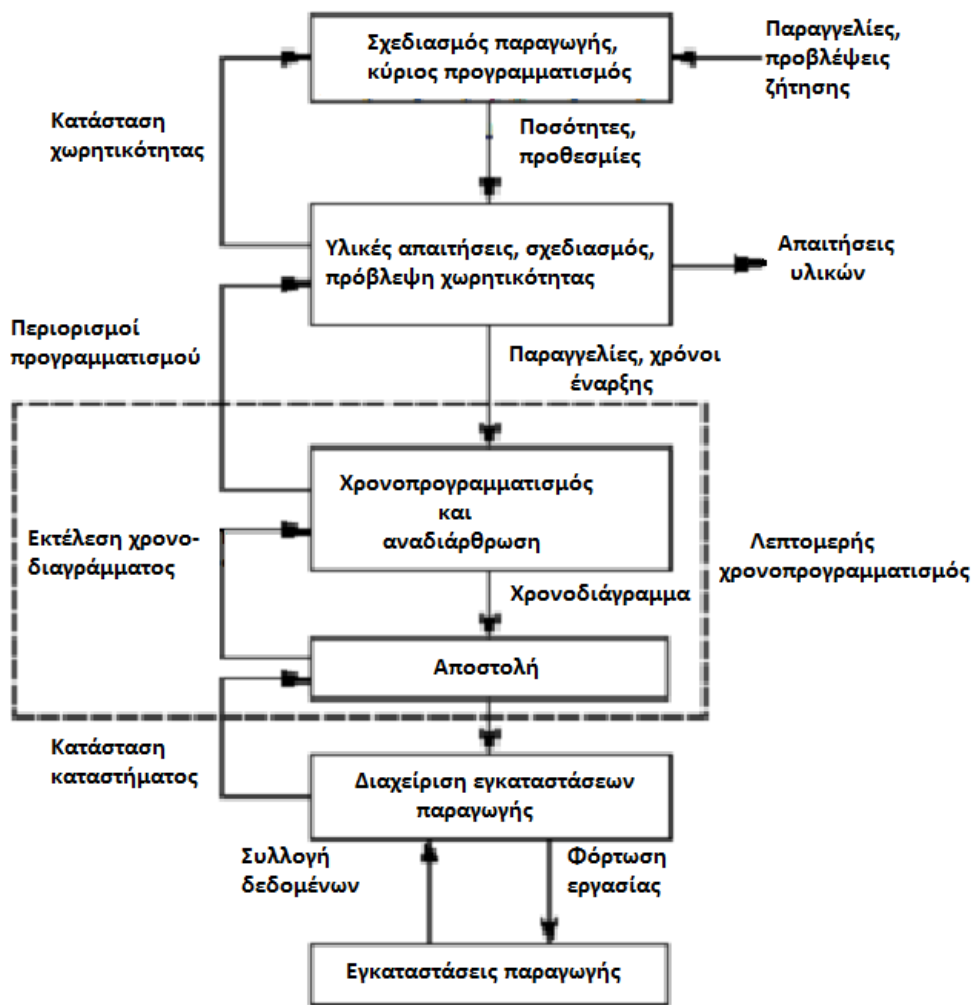
Ο προγραμματισμός γίνεται συχνά διαδραστικά μέσω ενός συστήματος υποστήριξης αποφάσεων που εγκαθίσταται σε έναν προσωπικό υπολογιστή ή σταθμό εργασίας συνδεδεμένο με το σύστημα ERP. Οι τερματικοί σταθμοί σε θέσεις-κλειδιά που είναι συνδεδεμένοι με το σύστημα ERP μπορούν να παρέχουν, στα τμήματα ολόκληρης της επιχείρησης, πρόσβαση σε όλες τις τρέχουσες πληροφορίες χρονοπρογραμματισμού. Αυτά τα τμήματα, με τη σειρά τους, μπορούν να παρέχουν στο σύστημα χρονοπρογραμματισμού ενημερωμένες πληροφορίες σχετικά με τις καταστάσεις εργασιών και μηχανών.

Υπάρχουν, βεβαίως, ακόμα περιβάλλοντα όπου η επικοινωνία μεταξύ της λειτουργίας χρονοπρογραμματισμού και άλλων φορέων λήψης αποφάσεων λαμβάνει χώρα σε συναντήσεις ή μέσω σημειώσεων.

1.1. Ο Χρονοπρογραμματισμός στη Παραγωγή,

Εξετάζει το γενικό περιβάλλον παραγωγής και το ρόλο του χρονοπρογραμματισμού του. Οι εντολές που απελευθερώνονται σε μια εργοστασιακή ρύθμιση πρέπει να μεταφραστούν σε εργασίες σχετιζόμενες με ημερομηνίες λήξης. Αυτές οι εργασίες πρέπει συχνά να υποβάλλονται σε επεξεργασία στις μηχανές σε ένα κέντρο εργασίας με δεδομένη εντολή ή σειρά. Η επεξεργασία των εργασιών μπορεί μερικές φορές να καθυστερήσει, εάν ορισμένες μηχανές είναι απασχολημένες και ενδέχεται να υπάρξουν προεκχωρήσεις όταν οι εργασίες υψηλής προτεραιότητας φτάνουν σε μηχανές που είναι απασχολημένες. Πρέπει επίσης να ληφθούν υπόψη τα απρόβλεπτα συμβάντα στο εργοτάξιο, όπως οι καταστροφές μηχανών ή οι χρόνοι επεξεργασίας που είναι μεγαλύτεροι από τους αναμενόμενους, καθώς ενδέχεται να έχουν σημαντικό αντίκτυπο στα χρονοδιαγράμματα. Σε ένα τέτοιο περιβάλλον, η ανάπτυξη ενός λεπτομερούς προγράμματος εργασιών βοηθά στη διατήρηση της αποτελεσματικότητας και του ελέγχου των λειτουργιών.

Οι εγκαταστάσεις παραγωγής δεν είναι το μόνο μέρος του οργανισμού που επηρεάζει τη διαδικασία του χρονοπρογραμματισμού. Επίσης επηρεάζεται από τη διαδικασία προγραμματισμού παραγωγής που διαχειρίζεται μεσοπρόθεσμο έως μακροπρόθεσμο σχεδιασμό για ολόκληρο τον οργανισμό. Η διαδικασία αυτή προσπαθεί να βελτιστοποιήσει το συνολικό συνδυασμό προϊόντων της επιχείρησης και τη μακροπρόθεσμη κατανομή πόρων βάσει των αποθεμάτων της, των προβλέψεων της ζήτησης και των απαιτούμενων πόρων. Οι αποφάσεις που λαμβάνονται σε αυτό το επίπεδο σχεδιασμού μπορεί να επηρεάσουν άμεσα τη διαδικασία του χρονοπρογραμματισμού. Το σχήμα 1.1 απεικονίζει ένα διάγραμμα ροής πληροφοριών σε ένα σύστημα παραγωγής.



Σχήμα 1.1

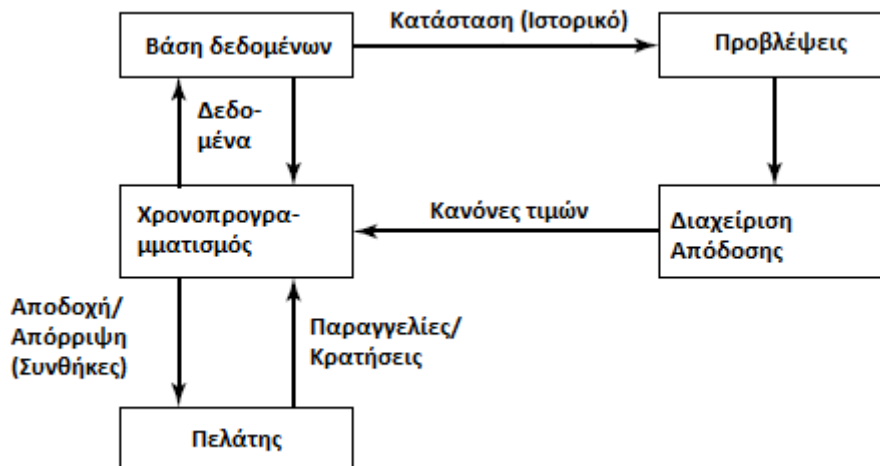
Σε ένα περιβάλλον παραγωγής, η λειτουργία προγραμματισμού πρέπει να αλληλεπιδρά με άλλες λειτουργίες λήψης αποφάσεων. Ένα δημοφιλές σύστημα που χρησιμοποιείται ευρέως είναι το σύστημα προγραμματισμού υλικών απαιτήσεων (Material Requirements Planning)(MRP). Αφού δημιουργηθεί ένα χρονοδιάγραμμα, είναι απαραίτητο να είναι διαθέσιμες όλες οι πρώτες ύλες και οι πόροι στις καθορισμένες ώρες. Οι έτοιμες ημερομηνίες όλων των εργασιών πρέπει να καθοριστούν από κοινού από το σύστημα σχεδιασμού παραγωγής / χρονοπρογραμματισμού και το σύστημα MRP.

Τα συστήματα MRP είναι κατά κανόνα αρκετά περίπλοκα. Κάθε εργασία έχει ένα Bill of Materials (BOM) που αναλύει τα μέρη που απαιτούνται για την παραγωγή. Το σύστημα MRP παρακολουθεί την απογραφή κάθε τμήματος. Επιπλέον, καθορίζει το χρονοδιάγραμμα των αγορών κάθε ενός από τα υλικά. Για να το πετύχει αυτό, χρησιμοποιεί τεχνικές όπως η ταξινόμηση κατά μέγεθος και ο προγραμματισμός παρτίδων που είναι παρόμοιες με αυτές που χρησιμοποιούνται στα συστήματα χρονοπρογραμματισμού. Υπάρχουν πολλά διαθέσιμα εμπορικά πακέτα λογισμικού MRP και, κατά συνέπεια, υπάρχουν πολλές εγκαταστάσεις παραγωγής με συστήματα MRP. Στις περιπτώσεις όπου η εγκατάσταση δεν διαθέτει σύστημα

χρονοπρογραμματισμού, το σύστημα MRP μπορεί να χρησιμοποιηθεί για τον προγραμματισμό της παραγωγής. Ωστόσο, σε σύνθετες ρυθμίσεις δεν είναι εύκολο για ένα σύστημα MRP να κάνει τον λεπτομερή χρονοπρογραμματισμό ικανοποιητικά.

1.2. Χρονοπρογραμματισμός σε υπηρεσίες

Η περιγραφή ενός οργανισμού εξυπηρέτησης και ενός συστήματος χρονοπρογραμματισμού δεν είναι τόσο εύκολη όσο η περιγραφή ενός οργανισμού παραγωγής. Η λειτουργία χρονοπρογραμματισμού σε έναν οργανισμό εξυπηρέτησης ενδέχεται να αντιμετωπίσει διάφορα προβλήματα. Μπορεί να χρειάζεται να αντιμετωπίσει την κράτηση πόρων ή την κράτηση αιθουσών συνεδριάσεων ή άλλων εγκαταστάσεων. Τα μοντέλα που χρησιμοποιούνται είναι μερικές φορές κάπως διαφορετικά από αυτά που χρησιμοποιούνται στη παραγωγή. Ο χρονοπρογραμματισμός σε ένα περιβάλλον υπηρεσίας πρέπει να συντονίζεται με άλλες λειτουργίες λήψης αποφάσεων, συνήθως μέσα σε περίτεχνα πληροφοριακά συστήματα, με τον ίδιο τρόπο όπως η λειτουργία χρονοπρογραμματισμού σε ένα περιβάλλον παραγωγής. Αυτά τα συστήματα πληροφοριών βασίζονται συνήθως σε εκτεταμένες βάσεις δεδομένων που περιέχουν όλες τις σχετικές πληροφορίες όσον αφορά τη διαθεσιμότητα πόρων και (δυνατικών) πελατών. Το σύστημα χρονοπρογραμματισμού συχνά αλληλεπιδρά με τους κλάδους διαχείρισης πρόβλεψης και διαχείρισης απόδοσης. Το σχήμα 1.2 απεικονίζει τη ροή πληροφοριών σε έναν οργανισμό εξυπηρέτησης όπως ένας πρακτορείο ενοικίασης αυτοκινήτων. Σε αντίθεση με το περιβάλλον παραγωγής, στο περιβάλλον υπηρεσιών δεν υπάρχει συνήθως σύστημα MRP.



Σχήμα 1.2

Κεφάλαιο 2

Ντετερμινιστικός χρονοπρογραμματισμός

Στο ντετερμινιστικό προγραμματισμό τα δίκτυα εργασιών είναι συνδεδεμένα μεταξύ τους με εξαρτήσεις οι οποίες περιγράφουν το έργο που πρέπει να εκτελεστεί, τη διάρκεια της εργασίας και τη προγραμματισμένη ολοκλήρωση του έργου. Στα ντετερμινιστικά μοντέλα κάθε εργασία έχει προγραμματισμένη διάρκεια καθώς επίσης μία προκάτοχο και μία διάδοχο εργασία. Το μόνο καθήκον που δεν πρέπει να έχει μία προκάτοχος είναι η έναρξη του έργου και το μόνο καθήκον που έχει όταν δεν έχει διάδοχο είναι η ολοκλήρωση του έργου. Η μεγαλύτερη διαδρομή μέσω του δικτύου είναι η κρίσιμη διαδρομή. Η συνολική διάρκεια του έργου είναι μια σταθερή τιμή και είναι καθοριστική

Τα τελευταία πενήντα χρόνια μια σημαντική ερευνητική προσπάθεια έχει επικεντρωθεί στον ντετερμινιστικό προγραμματισμό. Ο αριθμός και η ποικιλία των μοντέλων που εξετάζονται είναι εκπληκτικός. Κατά τη διάρκεια αυτής της περιόδου έχει αναπτυχθεί ένας συμβολισμός που περιγράφει με ευκολία τη δομή πολλών (αλλά σίγουρα όχι όλων) ντετερμινιστικών μοντέλων που έχουν εξεταστεί στη βιβλιογραφία.

2.1 Δομή και συμβολισμός

Σε όλα τα προβλήματα προγραμματισμού, ο αριθμός εργασιών και ο αριθμός των μηχανών θεωρείται πεπερασμένος. Ο αριθμός των εργασιών υποδηλώνεται με n και ο αριθμός των μηχανών με m . Συνήθως, ο δείκτης j αναφέρεται σε μια εργασία ενώ ο δείκτης i αναφέρεται σε ένα μηχάνημα. Εάν μια εργασία απαιτεί διάφορα βήματα ή λειτουργίες επεξεργασίας, τότε το ζεύγος (i, j) αναφέρεται στο βήμα επεξεργασίας ή στη λειτουργία της εργασίας j στο μηχάνημα i . Τα παρακάτω κομμάτια δεδομένων σχετίζονται με την εργασία j .

➤ Χρόνος επεξεργασίας (p_{ij}) (Processing time)

Το p_{ij} αντιπροσωπεύει το χρόνο επεξεργασίας της εργασίας j στο μηχάνημα i . Ο δείκτης i παραλείπεται εάν ο χρόνος επεξεργασίας της εργασίας j δεν εξαρτάται από το μηχάνημα ή εάν η εργασία j πρέπει να υποβληθεί σε επεξεργασία μόνο σε ένα δεδομένο μηχάνημα.

➤ Χρόνος άφιξης της εργασίας j στο σύστημα (r_j) (Release date)

Η ημερομηνία έναρξης r_j της εργασίας j μπορεί επίσης να αναφέρεται ως έτοιμη ημερομηνία. Είναι η ώρα που φτάνει η εργασία στο σύστημα, δηλ. Ο πρώτος χρόνος κατά τον οποίο η εργασία j μπορεί να ξεκινήσει την επεξεργασία της.

➤ Προθεσμίες, ημερομηνίες παράδοσης (d_j) (Due date)

Η ημερομηνία παράδοσης d_j της εργασίας j αντιπροσωπεύει τη δεσμευμένη αποστολή ή ημερομηνία ολοκλήρωσης (δηλαδή την ημερομηνία υποβολής της εργασίας στον πελάτη). Επιτρέπεται η ολοκλήρωση μιας εργασίας μετά την ημερομηνία παράδοσης, αλλά στη συνέχεια επιβάλλεται ποινή. Όταν μία ημερομηνία παράδοσης πρέπει να τηρηθεί, αναφέρεται ως προθεσμία και δηλώνεται από το \bar{d}_j .

➤ Βαρύτητα (w_j) (Weight)

Η βαρύτητα ή αλλιώς η σπουδαιότητα w_j της εργασίας j είναι βασικά ένας παράγοντας προτεραιότητας, υποδηλώνοντας τη σημασία της εργασίας j σε σχέση με τις άλλες εργασίες στο σύστημα. Για παράδειγμα, αυτή η βαρύτητα μπορεί να αντιπροσωπεύει το πραγματικό κόστος διατήρησης της εργασίας στο σύστημα. Το

κόστος αυτό μπορεί να είναι ένα κόστος εκμετάλλευσης ή αποθέματος. Θα μπορούσε επίσης να αντιπροσωπεύει το ποσό της αξίας που έχει ήδη προστεθεί στην εργασία.

Ένα πρόβλημα προγραμματισμού περιγράφεται από ένα τρίσημο $\alpha | \beta | \gamma$. Το πεδίο α περιγράφει το περιβάλλον του μηχανήματος και περιέχει μόνο μία καταχώρηση. Το πεδίο β παρέχει λεπτομέρειες σχετικά με τα χαρακτηριστικά επεξεργασίας και τους περιορισμούς και μπορεί να μην περιέχει καθόλου είσοδο, μία μόνο καταχώρηση ή πολλαπλές καταχωρίσεις. Το πεδίο γ περιγράφει τον στόχο που πρέπει να ελαχιστοποιηθεί και συχνά περιέχει μια μόνο καταχώρηση. Τα πιθανά περιβάλλοντα μηχανής που ορίζονται στο πεδίο α είναι:

➤ **Ένα μηχάνημα (1) (Single machine)**

Η περίπτωση ενός μηχανήματος είναι το απλούστερο από όλα τα πιθανά περιβάλλοντα μηχανής και είναι μια ειδική περίπτωση όλων των άλλων πολύπλοκων μηχανημάτων.

➤ **Πανομοιότυπες παράλληλες μηχανές(Pm) (Identical machines in parallel)**

Υπάρχουν παράλληλα m πανομοιότυπες μηχανές. Η εργασία j απαιτεί μία μόνο εργασία και μπορεί να υποβληθεί σε επεξεργασία σε οποιαδήποτε από τις μηχανές m ή σε οποιαδήποτε από αυτές ανήκει σε ένα δεδομένο υποσύνολο. Αν η εργασία j δεν μπορεί να επεξεργαστεί μόνο σε κάποιο μηχάνημα, αλλά μόνο σε κάποιο που ανήκει σε ένα συγκεκριμένο υποσύνολο M_j , τότε η καταχώρηση M_j εμφανίζεται στο β πεδίο.

➤ **Μηχανές παράλληλες με διαφορετικές ταχύτητες (Qm)(Machines In parallel with different speeds)**

Υπάρχουν μηχανές m παράλληλα με διαφορετικές ταχύτητες. Η ταχύτητα του μηχανήματος i δηλώνεται από το v_i . Ο χρόνος p_{ij} που η εργασία j περνάει στο μηχάνημα i είναι ίση με p_j/v_i (υποθέτοντας ότι η εργασία j λαμβάνει όλη την επεξεργασία της από το μηχάνημα i). Αυτό το περιβάλλον αναφέρεται ως ομοιόμορφα μηχανήματα. Εάν όλα τα μηχανήματα έχουν την ίδια ταχύτητα, δηλ. $v_i = 1$ για όλα τα i και $p_{ij} = p_j$, τότε το περιβάλλον είναι πανομοιότυπο με το προηγούμενο.

➤ **Μη σχετιζόμενες παράλληλες μηχανές (Rm)(Unrelated machines in parallel)**

Αυτό το περιβάλλον είναι μια περαιτέρω γενίκευση του προηγούμενου. Υπάρχουν παράλληλα m διαφορετικές μηχανές. Το μηχάνημα i μπορεί να επεξεργαστεί την εργασία j σε ταχύτητα v_{ij} . Ο χρόνος p_{ij} όπου η εργασία j περνάει στο μηχάνημα i είναι ίση με p_j/v_{ij} (πάλι υποθέτοντας ότι η εργασία j λαμβάνει όλη την επεξεργασία της από το μηχάνημα i). Αν οι ταχύτητες των μηχανών είναι ανεξάρτητες από τις εργασίες, δηλ. $v_{ij} = v_i$ για όλα τα i και j , τότε το περιβάλλον είναι πανομοιότυπο με το προηγούμενο.

➤ **Σύστημα συνεχούς ροής (Fm) (Flow shop)**

Υπάρχουν μηχανές m σε σειρά. Κάθε εργασία πρέπει να υποβληθεί σε επεξεργασία σε κάθε μία από τις μηχανές m . Όλες οι εργασίες πρέπει να ακολουθούν την ίδια διαδρομή, δηλ. Πρέπει πρώτα να υποβληθούν σε επεξεργασία στη μηχανή 1, στη συνέχεια στη μηχανή 2 κ.ο.κ. Μετά την ολοκλήρωση σε ένα

μηχάνημα, μια εργασία επιστρέφει στην ουρά στο επόμενο μηχάνημα. Συνήθως, όλες οι ουρές αναμένεται να λειτουργούν κάτω από την πειθαρχία First In First Out (FIFO), δηλαδή μια εργασία δεν μπορεί να "περάσει" άλλη ενώ περιμένει σε μια ουρά. Εάν η πειθαρχία FIFO είναι σε ισχύ, το σύστημα συνεχούς ροής αναφέρεται ως μετασχηματισμού και το β πεδίο περιλαμβάνει την καταχώρηση rpm (permutation).

➤ **Ευέλικτο σύστημα συνεχούς ροής (FFc)(Flexible flow shop)**

Ένα ευέλικτο σύστημα συνεχούς ροής είναι μια γενίκευση του συστήματος συνεχούς ροής και των περιβαλλόντων παράλληλων μηχανών. Αντί για μηχανές m σε σειρά υπάρχουν στάδια c σε σειρά και ένας αριθμός πανομοιότυπων παράλληλων μηχανών σε κάθε στάδιο. Κάθε εργασία πρέπει να επεξεργαστεί πρώτα στο στάδιο 1, στη συνέχεια στο στάδιο 2, και ούτω καθεξής. Ένα στάδιο λειτουργεί ως τράπεζα παράλληλων μηχανών. Σε κάθε στάδιο, η εργασία j απαιτεί επεξεργασία σε μία μόνο μηχανή και οποιαδήποτε μηχανή μπορεί να τη κάνει. Οι ουρές μεταξύ των διαφόρων σταδίων μπορεί να λειτουργούν ή να μην λειτουργούν σύμφωνα με την πειθαρχία του First Come First Served (FCFS).

➤ **Σύστημα παραγωγής κατά παραγγελία (Jm) (Job shop)**

Σε ένα σύστημα παραγωγής κατά παραγγελία με μηχανές m κάθε εργασία έχει τη δική της προκαθορισμένη διαδρομή για να ακολουθήσει. Διακρίνονται σε συστήματα παραγωγής κατά παραγγελία στα οποία κάθε εργασία επισκέπτεται κάθε μηχανή το πολύ μία φορά και σε εκείνα στα οποία μια εργασία μπορεί να επισκέπτεται κάθε μηχανή περισσότερες από μία φορές. Στην τελευταία περίπτωση καταχωρείται στο β -πεδίο $rcrc$ για την επανακυκλοφορία (recirculation).

➤ **Ευέλικτο σύστημα παραγωγής κατά παραγγελία (FJc) (Flexible job shop)**

Ένα ευέλικτο σύστημα παραγωγής κατά παραγγελία είναι μια γενίκευση του συστήματος κατά παραγγελία και των περιβαλλόντων παράλληλων μηχανών. Αντί για μηχανές m σε σειρά υπάρχουν c κέντρα εργασίας με μια σειρά παράλληλων πανομοιότυπων μηχανών, σε κάθε κέντρο εργασίας. Κάθε εργασία έχει τη δική της διαδρομή για να ακολουθήσει. Η εργασία j απαιτεί επεξεργασία σε κάθε κέντρο εργασίας σε μία μόνο μηχανή και οποιαδήποτε μηχανή μπορεί να τη κάνει. Αν μια εργασία στη διαδρομή της επισκεφθεί ένα κέντρο εργασίας περισσότερες από μία φορές, τότε το β -πεδίο θα περιέχει την καταχώρηση $rcrc$ για την επανακυκλοφορία (recirculation).

➤ **Open shop (Om)**

Υπάρχουν μηχανές m . Κάθε εργασία πρέπει να υποβληθεί εκ νέου σε επεξεργασία σε κάθε μια από τις μηχανές m . Ωστόσο, ορισμένοι από αυτούς τους χρόνους επεξεργασίας μπορεί να είναι μηδενικοί. Δεν υπάρχουν περιορισμοί όσον αφορά τη δρομολόγηση κάθε εργασίας στο περιβάλλον του μηχανήματος. Ο χρονοπρογραμματιστής επιτρέπεται να καθορίζει μια διαδρομή για κάθε εργασία και διαφορετικές εργασίες μπορεί να έχουν διαφορετικές διαδρομές.

Οι περιορισμοί και τα χαρακτηριστικά επεξεργασίας που καθορίζονται στο πεδίο β μπορεί να περιλαμβάνουν πολλαπλές καταχωρήσεις. Πιθανές καταχωρήσεις στο πεδίο β είναι:

➤ **Χρόνος άφιξης της εργασία j στο σύστημα (r_j) (Release dates)**

Εάν αυτό το σύμβολο εμφανίζεται στο πεδίο β , τότε η εργασία j δεν μπορεί να ξεκινήσει την επεξεργασία της πριν από την ημερομηνία έναρξης r_j . Αν το r_j δεν εμφανίζεται στο πεδίο β , η επεξεργασία της εργασίας j μπορεί να ξεκινήσει ανά πάσα στιγμή. Σε αντίθεση με τις ημερομηνίες έναρξης, οι ημερομηνίες παράδοσης δεν καθορίζονται σε αυτό το πεδίο. Ο τύπος της αντικειμενικής λειτουργίας δίνει επαρκή ένδειξη εάν υπάρχουν ημερομηνίες παράδοσης ή όχι.

➤ **Προεκχωρήσεις (prmp) (preemptions)**

Οι προεκχωρήσεις υποδηλώνουν ότι δεν είναι απαραίτητο να διατηρείται μια εργασία σε ένα μηχάνημα, επειδή έχει αρχίσει, μέχρι την ολοκλήρωσή της. Ο χρονοπρογραμματιστής επιτρέπεται να διακόψει την επεξεργασία μιας εργασίας ανά πάσα στιγμή και να θέτει διαφορετική εργασία στο μηχάνημα. Η ποσότητα της επεξεργασίας που έχει ήδη γίνει στην εργασία που μπαίνει στην αναμονή δεν έχει χαθεί. Όταν μια εργασία επιστρέφει στη συνέχεια από την αναμονή στο μηχάνημα (ή σε άλλο μηχάνημα, στην περίπτωση παράλληλων μηχανημάτων), χρειάζεται το μηχάνημα μόνο για τον υπόλοιπο χρόνο επεξεργασίας. Όταν επιτρέπονται οι προεκχωρήσεις, περιλαμβάνεται το prmp στο β πεδίο, όταν δεν επιτρέπονται, δεν περιλαμβάνεται.

➤ **Περιορισμοί προτεραιότητας (prec) (Precedence constraints)**

Οι περιορισμοί προτεραιότητας ενδέχεται να εμφανίζονται σε ένα μόνο μηχάνημα ή σε περιβάλλον με παράλληλα μηχανήματα, απαιτώντας την ολοκλήρωση μίας ή περισσότερων εργασιών πριν την έναρξη της επεξεργασίας μιας άλλης. Υπάρχουν διάφορες ειδικές μορφές περιορισμών προτεραιότητας: εάν κάθε εργασία έχει το πολύ ένα προκατόχο και το πολύ ένα διάδοχο, οι περιορισμοί αναφέρονται ως αλυσίδες. Εάν κάθε εργασία έχει το πολύ ένα διάδοχο, οι περιορισμοί αναφέρονται ως intree. Εάν κάθε εργασία έχει το πολύ ένα προκατόχο, οι περιορισμοί αναφέρονται ως outtree. Αν δεν εμφανίζεται prec στο β πεδίο, οι εργασίες δεν υπόκεινται σε περιορισμούς προτεραιότητας.

➤ **Ακολουθιακά εξαρτώμενοι χρόνοι προετοιμασίας (s_{jk}) (Sequence dependent setup times)**

Το s_{jk} αντιπροσωπεύει τον χρόνο προετοιμασίας που εξαρτάται από την ακολουθία που προκύπτει μεταξύ της επεξεργασίας των εργασιών j και k . s_{0k} υποδηλώνει το χρόνο ρύθμισης για την εργασία k εάν η εργασία k είναι η πρώτη στην ακολουθία και s_{j0} ο χρόνος καθαρισμού μετά την εργασία j εάν η εργασία j είναι τελευταία στη σειρά (φυσικά, s_{0k} και s_{j0} μπορεί να είναι μηδέν). Αν ο χρόνος προετοιμασίας μεταξύ των εργασιών j και k εξαρτάται από το μηχάνημα, τότε συμπεριλαμβάνεται ο δείκτης i , δηλ. s_{ijk} . Εάν δεν εμφανιστεί κανένα s_{jk} στο β πεδίο, όλοι οι χρόνοι προετοιμασίας θεωρούνται ότι είναι 0 ή ανεξάρτητοι από την ακολουθία, οπότε περιλαμβάνονται απλώς στους χρόνους επεξεργασίας.

➤ **Οικογένειες εργασιών (fmls) (Job families)**

Οι n εργασίες ανήκουν στην περίπτωση αυτή σε F διαφορετικές οικογένειες εργασίας. Οι εργασίες από την ίδια οικογένεια μπορεί να έχουν διαφορετικούς χρόνους επεξεργασίας, αλλά μπορούν να επεξεργαστούν σε μια μηχανή η μία μετά την άλλη χωρίς να απαιτείται κάποια ρύθμιση μεταξύ τους. Ωστόσο, εάν το μηχάνημα μεταβαίνει από μια οικογένεια σε άλλη, ως πούμε από την οικογένεια g στην οικογένεια h , τότε απαιτείται ρύθμιση. Εάν αυτός ο χρόνος ρύθμισης εξαρτάται και από τις δύο οικογένειες g και h και εξαρτάται από ακολουθία, τότε υποδηλώνεται με s_{gh} . Εάν εξαρτάται μόνο από την οικογένεια που πρόκειται να

ξεκινήσει, δηλαδή από την οικογένεια h , τότε υποδηλώνεται με s_h . Αν δεν εξαρτάται από καμία οικογένεια, υποδηλώνεται με s .

➤ **Επεξεργασία παρτίδας (παρτίδα (b)) (Batch processing (batch(b)))**

Μια μηχανή μπορεί να είναι σε θέση να επεξεργαστεί μια σειρά εργασιών, ας πούμε β , ταυτόχρονα. Δηλαδή, μπορεί να επεξεργαστεί ταυτόχρονα μια παρτίδα έως και β εργασίες. Οι χρόνοι επεξεργασίας των εργασιών σε μια παρτίδα μπορεί να μην είναι όλοι ίδιοι και ολόκληρη η παρτίδα τελειώνει μόνο όταν ολοκληρωθεί η τελευταία εργασία της παρτίδας, υποδηλώνοντας ότι ο χρόνος ολοκλήρωσης ολόκληρης της παρτίδας καθορίζεται από την εργασία με το μεγαλύτερο χρόνο επεξεργασίας. Εάν $b = 1$, τότε το πρόβλημα μειώνεται σε ένα συμβατικό περιβάλλον χρονοπρογραμματισμού. Μια άλλη ειδική περίπτωση που έχει ενδιαφέρον είναι $b = \infty$, δηλ. δεν υπάρχει όριο στον αριθμό των εργασιών που μπορεί να χειριστεί το μηχάνημα ανά πάσα στιγμή.

➤ **Βλάβες (brkdwn) (Breakdowns)**

Οι βλάβες υποδηλώνουν ότι μια μηχανή μπορεί να μην είναι συνεχώς διαθέσιμη. Οι περίοδοι κατά τις οποίες ένα μηχάνημα δεν είναι διαθέσιμο θεωρείται ότι είναι σταθερές (π.χ. λόγω μετατοπίσεων ή προγραμματισμένης συντήρησης). Εάν υπάρχει ένας αριθμός πανομοιότυπων μηχανημάτων παράλληλα, ο αριθμός των διαθέσιμων μηχανημάτων ανά πάσα στιγμή είναι συνάρτηση του χρόνου, δηλ. $M(t)$. Οι βλάβες μηχανημάτων αναφέρονται επίσης κατά καιρούς ως περιορισμοί στη διαθεσιμότητα μηχανών.

➤ **Περιορισμοί επιλεξιμότητας μηχανής (M_j) (Machine eligibility restrictions)**

Το σύμβολο M_j μπορεί να εμφανιστεί στο β πεδίο όταν το περιβάλλον μηχανήματος είναι m μηχανές παράλληλα (Pm). Όταν εμφανίζεται το M_j , δεν είναι όλες οι μηχανές m ικανές να επεξεργάζονται την εργασία j . Το M_j υποδηλώνει το σύνολο μηχανών που μπορούν να επεξεργαστούν την εργασία j . Εάν το β πεδίο δεν περιέχει M_j , η εργασία j μπορεί να επεξεργαστεί σε οποιαδήποτε από τις μηχανές m .

➤ **Μετατροπή (prmu) (Permutation)**

Ένας περιορισμός που μπορεί να εμφανιστεί στο περιβάλλον συνεχούς ροής είναι ότι οι ουρές μπροστά σε κάθε μηχανή λειτουργούν σύμφωνα με την πειθαρχία First In First Out (FIFO). Αυτό σημαίνει ότι η εντολή (ή η μεταβολή) στην οποία οι εργασίες περνούν από το πρώτο μηχάνημα διατηρείται σε όλο το σύστημα.

➤ **Αποκλεισμός (Blocking)**

Το μπλοκάρισμα είναι ένα φαινόμενο που μπορεί να συμβεί σε συστήματα συνεχούς ροής. Εάν ένα σύστημα συνεχούς ροής έχει περιορισμένη ενδιάμεση μνήμη μεταξύ δύο διαδοχικών μηχανών, αυτό μπορεί να συμβεί όταν η μνήμη είναι γεμάτη, τότε το ανάντη μηχάνημα δεν επιτρέπεται να απελευθερώσει μια ολοκληρωμένη εργασία. Ο αποκλεισμός σημαίνει ότι η ολοκληρωμένη εργασία πρέπει να παραμείνει στην ανάντη μηχανή εμποδίζοντάς την να λειτουργήσει στην επόμενη εργασία. Η πιο συνηθισμένη εμφάνιση του μπλοκαρίσματος είναι η περίπτωση με μηδενικές μνήμες μεταξύ οποιωνδήποτε δύο διαδοχικών μηχανών. Σε αυτή την περίπτωση, μια εργασία που έχει ολοκληρώσει την επεξεργασία της σε μια δεδομένη μηχανή δεν μπορεί να εγκαταλείψει τη μηχανή, εάν η προηγούμενη εργασία δεν έχει ακόμη ολοκληρώσει την επεξεργασία της στο επόμενο μηχάνημα.

Έτσι, η μπλοκαρισμένη εργασία αποτρέπει επίσης (ή μπλοκάρει) την επόμενη εργασία από το να ξεκινήσει την επεξεργασία της στο συγκεκριμένο μηχάνημα.

➤ **Απαγόρευση Αναμονής (nwt) (No-Wait)**

Η απαίτηση απαγόρευσης αναμονής είναι ένα άλλο φαινόμενο που μπορεί να συμβεί σε καταστήματα ροής. Οι εργασίες δεν επιτρέπεται να περιμένουν μεταξύ δύο διαδοχικών μηχανών. Αυτό σημαίνει ότι ο χρόνος εκκίνησης μιας εργασίας στο πρώτο μηχάνημα πρέπει να καθυστερήσει για να εξασφαλίσει ότι η εργασία μπορεί να περάσει από το σύστημα συνεχούς ροής χωρίς να χρειαστεί να περιμένει για οποιαδήποτε μηχανή. Ένα παράδειγμα μιας τέτοιας λειτουργίας είναι ένα χαλυβουργείο στο οποίο μια πλάκα χάλυβα δεν επιτρέπεται να περιμένει καθώς θα κρυώσει κατά τη διάρκεια μιας αναμονής. Είναι σαφές ότι κάτω από την απαγόρευση αναμονής τα μηχανήματα λειτουργούν επίσης σύμφωνα με την πειθαρχία FIFO.

➤ **Ανακυκλοφορία (rcrc) (Recirculation)**

Η ανακυκλοφορία μπορεί να συμβεί σε ένα σύστημα παραγωγής κατά παραγγελία ή σε ένα ευέλικτο σύστημα παραγωγής κατά παραγγελία όταν μια εργασία μπορεί να επισκεφθεί μια μηχανή ή ένα κέντρο εργασίας περισσότερες από μία φορές.

Κάθε άλλη καταχώρηση που μπορεί να εμφανιστεί στο πεδίο β είναι αυτονόητη. Για παράδειγμα, $p_j = p$ υπονοεί ότι όλοι οι χρόνοι επεξεργασίας είναι ίσοι και $d_j = d$ υποδηλώνει ότι όλες οι ημερομηνίες λήξης είναι ίσες. Όπως αναφέρθηκε προηγουμένως, οι ημερομηνίες λήξης, σε αντίθεση με τις ημερομηνίες έκδοσης, συνήθως δεν καθορίζονται ρητά σε αυτό το πεδίο. Ο τύπος της αντικειμενικής λειτουργίας δίνει επαρκή ένδειξη εάν οι εργασίες έχουν ημερομηνία λήξης ή όχι.

Η αντικειμενική που πρέπει να ελαχιστοποιηθεί είναι πάντα συνάρτηση των χρόνων ολοκλήρωσης των εργασιών, οι οποίες, φυσικά, εξαρτώνται από το χρονοδιάγραμμα. Ο χρόνος ολοκλήρωσης της λειτουργίας της εργασίας j στη μηχανή i υποδηλώνεται από τον C_{ij} . Ο χρόνος της εργασίας j που εξέρχεται από το σύστημα (δηλαδή, ο χρόνος ολοκλήρωσής του στο τελευταίο μηχάνημα στο οποίο χρειάζεται επεξεργασία) δηλώνεται από το C_j . Η αντικειμενική μπορεί επίσης να είναι συνάρτηση των ημερομηνιών λήξης. Η καθυστέρηση της εργασίας j ορίζεται ως

$$L_j = C_j - d_j$$

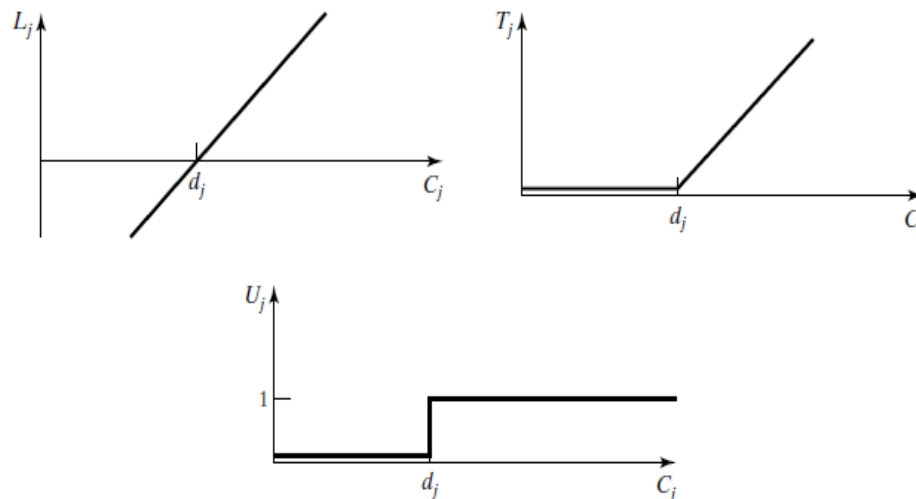
η οποία είναι θετική όταν η εργασία j ολοκληρώνεται αργά και αρνητική όταν ολοκληρωθεί νωρίς. Η βραδύτερη περάτωση της εργασίας j ορίζεται ως

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0).$$

Η διαφορά μεταξύ της βραδύτερης περάτωσης της εργασίας και της καθυστέρησης έγκειται στο γεγονός ότι η βραδύτερη περάτωση δεν είναι ποτέ αρνητική. Η ποινή μονάδας της εργασίας j ορίζεται

$$U_j = \begin{cases} 1, & C_j > d_j \\ 0, & otherwise \end{cases}$$

Η καθυστέρηση, η βραδύτερη περάτωση και η ποινή μονάδας είναι οι τρεις βασικές λειτουργίες ποινής που σχετίζονται με την ημερομηνία λήξης. Το σχήμα αυτών των λειτουργιών απεικονίζεται στο σχήμα 2.1.



Σχήμα 2.1.

Παραδείγματα πιθανών αντικειμενικών λειτουργιών που πρέπει να ελαχιστοποιηθούν είναι:

➤ **Makespan (C_{max}).**

Το makespan, που ορίζεται ως $\max(C_1, \dots, C_n)$, είναι ισοδύναμο με το χρόνο ολοκλήρωσης της τελευταίας εργασίας για έξοδο από το σύστημα. Ένα ελάχιστο makespan συνήθως υποδηλώνει καλή χρήση του μηχανήματος.

➤ **Μέγιστη καθυστέρηση (L_{max}).**

Η μέγιστη καθυστέρηση, L_{max} , ορίζεται ως μέγιστο (L_1, \dots, L_n) . Μετράει τη χειρότερη παραβίαση των προθεσμιών.

➤ **Συνολικός σταθμισμένος χρόνος ολοκλήρωσης ($\sum w_j C_j$).**

Το άθροισμα της βαρύτητας των χρόνων ολοκλήρωσης των εργασιών n δίνει μια ένδειξη για το συνολικό κόστος εκμετάλλευσης ή αποθέματος που προκύπτει από το χρονοδιάγραμμα. Το άθροισμα των χρόνων ολοκλήρωσης στη βιβλιογραφία συχνά αναφέρεται ως χρόνος ροής. Η συνολική βαρύτητα του χρόνου ολοκλήρωσης αναφέρεται τότε ως βαρύτητα του χρόνου ροής.

➤ **Εκπτώτικος συνολικός σταθμισμένος χρόνος ολοκλήρωσης ($\sum w_j (1 - e^{-rC_j})$).**

Αυτή είναι μια πιο γενική συνάρτηση κόστους από την προηγούμενη, όπου το κόστος προεξοφλείται με ρυθμό r , 0

➤ **Συνολική βαρύτητα της βραδύτερης περάτωσης ($\sum w_j T_j$).**

Αυτή είναι επίσης μια γενικότερη λειτουργία κόστους από το συνολικό σταθμισμένο χρόνο ολοκλήρωσης.

➤ **Βαρύτητα του αριθμού των καθυστερημένων εργασιών ($\sum w_j U_j$).**

Η βαρύτητα του αριθμού των εργασιών με βραδύτερη περάτωση δεν είναι μόνο ένα μέτρο ακαδημαϊκού ενδιαφέροντος, είναι συχνά ένας στόχος στην πράξη, καθώς είναι ένα μέτρο που μπορεί να καταγραφεί πολύ εύκολα.

Όλες οι προαναφερόμενες αντικειμενικές λειτουργίες είναι τα αποκαλούμενα κανονικά μέτρα απόδοσης. Ένα κανονικό μέτρο απόδοσης είναι μια λειτουργία που δεν μειώνεται στο C_1, \dots, C_n . Πρόσφατα οι ερευνητές έχουν αρχίσει να μελετούν αντικειμενικές λειτουργίες που δεν είναι κανονικές. Για παράδειγμα, όταν η εργασία j έχει προθεσμία d_j , μπορεί να υποβληθεί σε ποινή νωρίτερης περάτωσης, όπου η νωρίτερη περάτωση της εργασίας j ορίζεται ως

$$E_j = \max(d_j - C_j, 0)$$

Αυτή η ποινή νωρίτερης περάτωσης δεν αυξάνεται στο C_j . Μία αντικειμενική, όπως το σύνολο της νωρίτερης περάτωσης και το σύνολο της βραδύτερης περάτωσης, δηλ.,

$$\sum_{j=1}^n E_j + \sum_{j=1}^n T_j$$

δεν είναι κανονική. Μία γενικότερη αντικειμενική που δεν είναι κανονική είναι η συνολική βαρύτητα της νεότερης περάτωσης και η συνολική βαρύτητα της βραδύτερης περάτωσης, δηλ.,

$$\sum_{j=1}^n w'_j E_j + \sum_{j=1}^n w''_j T_j$$

Η σπουδαιότητα που συνδέεται με την νωρίτερη περάτωση της εργασίας j (w'_j) μπορεί να είναι διαφορετική από τη σπουδαιότητα που συνδέεται με την βραδύτερη περάτωση της εργασίας j (w''_j).

Για παράδειγμα το $1 \mid r_j, pmtp \mid \sum w_j C_j$ δηλώνει ένα σύστημα μίας μηχανής με εργασία j που εισέρχεται στο σύστημα κατά την ημερομηνία έναρξης r_j . Προεκχωρήσεις επιτρέπονται. Η αντικειμενική που πρέπει να ελαχιστοποιηθεί είναι το άθροισμα της βαρύτητας των χρόνων ολοκλήρωσης. Αυτό το μοντέλο μπορεί να χρησιμοποιηθεί για να μελετηθεί το ντετερμινιστικό αντίγραφο του προβλήματος που περιγράφεται στο Παράδειγμα (Εργασίες χρονοπρογραμματισμού σε κεντρική μονάδα επεξεργασίας (CPU)).

Κεφάλαιο 3

Μοντέλο Μεμονωμένης Μηχανής (Ντετερμινιστικό)

Τα μοντέλα μίας μηχανής είναι σημαντικά για διάφορους λόγους. Το περιβάλλον του είναι πολύ απλό και μια ειδική περίπτωση όλων των άλλων περιβαλλόντων. Τα μοντέλα μεμονωμένων μηχανών έχουν συχνά ιδιότητες που ούτε οι παράλληλες μηχανές ούτε οι μηχανές σε σειρά έχουν. Τα αποτελέσματα που μπορούν να ληφθούν για μοντέλα μεμονωμένων μηχανών όχι μόνο παρέχουν πληροφορίες για το περιβάλλον του μεμονωμένου μηχανήματος, αλλά παρέχουν και μια βάση για τα ευρετικά που εφαρμόζονται σε πιο περίπλοκα περιβάλλοντα μηχανών. Στην πράξη, τα προβλήματα χρονοπρογραμματισμού σε πιο περίπλοκα περιβάλλοντα μηχανών συχνά αποσυντίθενται σε υποπροβλήματα που αφορούν μεμονωμένες μηχανές. Για παράδειγμα, ένα πολύπλοκο περιβάλλον μηχανής με ένα μόνο σημείο συμφόρησης μπορεί να οδηγήσει σε ένα μοντέλο μεμονωμένης μηχανής.

Μοντέλα μεμονωμένων μηχανών είναι ο συνολικός χρόνος ολοκλήρωσης, η μέγιστη καθυστέρηση, ο αριθμός των καθυστερημένων εργασιών, η συνολική βραδύτερη περάτωση και η συνολική σταθμισμένη βραδύτερη περάτωση.

3.1 Το πρόβλημα του συνολικού σταθμισμένου χρόνου ολοκλήρωσης

Η πρώτη αντικειμενική που πρέπει να ληφθεί υπόψη είναι ο συνολικός σταθμισμένος χρόνος ολοκλήρωσης, δηλ. $1 \parallel \sum w_j C_j$. Το βάρος w_j της εργασίας j μπορεί να θεωρηθεί ως παράγοντας σπουδαιότητας. Μπορεί να αντιπροσωπεύει είτε ένα κόστος εκμετάλλευσης ανά μονάδα χρόνου είτε την αξία που έχει ήδη προστεθεί στην εργασία j . Αυτό το πρόβλημα δημιουργεί έναν από τους πιο γνωστούς κανόνες στη θεωρία του χρονοπρογραμματισμού που ονομάζεται σταθμισμένος μικρότερος χρόνος κατεργασίας (Weighted Shortest Processing Time) (WSPT). Σύμφωνα με αυτόν τον κανόνα, οι εργασίες διανέμονται κατά φθίνουσα σειρά του w_j/p_j .

3.1.1 Θεώρημα: Ο κανόνας WSPT είναι βέλτιστος για $1 \parallel \sum w_j C_j$.

3.1.2 Αλγόριθμος (Συνολικός σταθμισμένος χρόνος ολοκλήρωσης και αλυσίδες)

Κάθε φορά που το μηχάνημα ελευθερώνεται, επιλέγουμε μεταξύ των υπόλοιπων αλυσίδων τη μία με το το υψηλότερο συντελεστή ρ . Επεξεργαζόμαστε την αλυσίδα χωρίς διακοπή μέχρι και την εργασία που καθορίζει τον συντελεστή ρ της.

Το παρακάτω παράδειγμα απεικονίζει τη χρήση του αλγορίθμου.

3.1.3 Παράδειγμα (Συνολικός σταθμισμένος χρόνος ολοκλήρωσης και αλυσίδες)

Εξετάζουμε τις ακόλουθες δύο αλυσίδες:

1 → 2 → 3 → 4

Και

$$5 \rightarrow 6 \rightarrow 7$$

Όπου οι βαρύτητες των εργασιών και οι χρόνοι επεξεργασίας τους δίδονται στον παρακάτω πίνακα.

jobs	1	2	3	4	5	6	7
w_j	6	18	12	8	8	17	18
p_j	3	6	6	5	4	8	10

Κάθε φορά που το μηχάνημα ελευθερώνεται, επιλέγει μεταξύ των υπόλοιπων αλυσίδων τη μία με το υψηλότερο συντελεστή ρ . Ο συντελεστής ρ της πρώτης αλυσίδας είναι $(6 + 18) / (3 + 6)$ και προσδιορίζεται από την εργασία 2. Ο συντελεστής ρ της δεύτερης αλυσίδας είναι $(8 + 17) / (4 + 8)$ από τη εργασία 6. Καθώς το $24/9$ είναι μεγαλύτερο από το $25/12$ οι εργασίες 1 και 2 επεξεργάζονται πρώτα. Ο συντελεστής ρ του υπόλοιπου τμήματος της πρώτης αλυσίδας είναι $12/6$ και προσδιορίζεται από την εργασία 3. Καθώς το $25/12$ είναι μεγαλύτερο από $12/6$ οι εργασίες 5 και 6 ακολουθούν τις εργασίες 1 και 2. Ο συντελεστής ρ του υπόλοιπου τμήματος της δεύτερης αλυσίδας είναι $18/10$ και καθορίζεται από τη εργασία 7. Οπότε η εργασία 3 ακολουθεί την εργασία 6. Δεδομένου ότι ο λόγος w_j/p_j εργασίας 7 είναι μεγαλύτερος από τον λόγο της εργασίας 4, η εργασία 7 ακολουθεί την εργασία 3 και η εργασία 4 εισέρχεται τελευταία.

3.2 Το πρόβλημα της μέγιστης καθυστέρησης

Η αντικειμενική της μέγιστης καθυστέρησης σχετίζεται με την προθεσμία και είναι μάλλον γενικής φύσης, δηλαδή το πρόβλημα $1 | \text{prec} | h_{max}$, όπου

$$h_{max} = \max (h_1(C_1), \dots, h_n(C_n))$$

με $h_j, j = 1, \dots, n$, οι μη μειούμενες λειτουργίες κόστους. Αυτή η αντικειμενική είναι σαφώς σχετιζόμενη με την προθεσμία αφού οι λειτουργίες h_j μπορεί να λάβουν οποιαδήποτε από τις μορφές που απεικονίζονται στο Σχήμα 2.1.1.

3.2.1 Παράδειγμα (Ελαχιστοποίηση του μέγιστου κόστους)

Θεωρούμε τις ακόλουθες 3 εργασίες.

jobs	1	2	3
p_j	2	3	5
$h_j(C_j)$	$1+C_1$	$1.2C_2$	10

Το makespan $C_{max} = 10$ και $h_3(10) < h_1(10) < h_2(10)$ (αφού $10 < 11 < 12$). Συνεπώς η εργασία 3, είναι προγραμματισμένη τελευταία και πρέπει να αρχίσει την επεξεργασία της στο χρόνο 5. Για να καθορισθεί ποια εργασία πρόκειται να επεξεργαστεί πριν την εργασία 3, το $h_2(5)$ πρέπει να συγκριθεί με το $h_1(5)$. Είτε η εργασία 1 είτε η εργασία 2 μπορούν να υποβληθούν σε επεξεργασία πριν από την εργασία 3 σε ένα βέλτιστο πρόγραμμα αφού $h_1(5) = h_2(5) = 6$. Έτσι δύο προγράμματα είναι βέλτιστα: 1,2,3 και 2,1,3.

Το πρόβλημα $1 \parallel L_{max}$ είναι η πιο γνωστή περίπτωση του $1 \mid \text{prec} \mid h_{max}$. Η συνάρτηση h_j ορίζεται στη συνέχεια ως $C_j - d_j$ και ο αλγόριθμος αποδίδει το χρονοδιάγραμμα που κατατάσσει τις εργασίες με αύξουσα σειρά σύμφωνα με την προθεσμία τους, δηλαδή η εργασία με τη νωρίτερη προθεσμία επεξεργάζεται πρώτη. (Earliest Due Date) (EDD).

Μία γενίκευση του $1 \parallel L_{max}$ είναι το πρόβλημα $1 \mid r_j \mid L_{max}$ με τις εργασίες να ξεκινούν σε διαφορετικές χρονικές στιγμές. Αυτή η γενίκευση, η οποία δεν επιτρέπει την προεκχωρήσεις, είναι σημαντικά πιο δύσκολη από το πρόβλημα με όλες τις εργασίες διαθέσιμες στο χρόνο 0. Το βέλτιστο χρονοδιάγραμμα δεν είναι απαραίτητα ένα χρονοδιάγραμμα χωρίς καθυστέρηση. Μπορεί να είναι πλεονεκτικό να κρατάμε το μηχάνημα σε αδράνεια ακριβώς πριν από την έναρξη μιας νέας εργασίας.

3.3 Το πρόβλημα του αριθμού των καθυστερημένων εργασιών

Μία άλλη αντικειμενική σχετιζόμενη με την ημερομηνία παράδοσης/προθεσμία είναι η $\sum U_j$. Η αντικειμενική αυτή μπορεί αρχικά να φανεί κάπως τεχνητή και να μην έχει πρακτικό ενδιαφέρον. Ωστόσο, στον πραγματικό κόσμο, είναι ένα μέτρο απόδοσης που παρακολουθείται συχνά και σύμφωνα με το οποίο οι διαχειριστές κάνουν μετρήσεις.

Ένα βέλτιστο πρόγραμμα για $1 \parallel \sum U_j$ παίρνει τη μορφή ενός συνόλου εργασιών που θα εκπληρώσουν τις ημερομηνίες παράδοσης/προθεσμίες τους και οι οποίες προγραμματίζονται πρώτες ακολουθούμενες από το σύνολο των υπολειπόμενων εργασιών που δεν θα καλύψουν τις ημερομηνίες παράδοσης/προθεσμίες τους και οι οποίες έχουν προγραμματιστεί τελευταίες. Το πρώτο σύνολο εργασιών πρέπει να προγραμματιστεί σύμφωνα με το EDD, προκειμένου να διασφαλιστεί ότι η L_{max} είναι αρνητική. Η σειρά με την οποία προγραμματίζεται το δεύτερο σύνολο εργασιών είναι άσχετη.

Το πρόβλημα $1 \parallel \sum U_j$ μπορεί να λυθεί εύκολα χρησιμοποιώντας έναν αλγόριθμο προς τα εμπρός. Επανατοποθετούμε τις εργασίες με τέτοιο τρόπο ώστε $d_1 \leq d_2 \leq \dots \leq d_n$. Ο αλγόριθμος περνάει από η επανάληψεις. Στην επανάληψη k του αλγόριθμου οι εργασίες 1,2, ..., k λαμβάνεται υπόψη. Από αυτές τις εργασίες k , το υποσύνολο J αναφέρεται σε εργασίες οι οποίες, σε ένα βέλτιστο χρονοδιάγραμμα, μπορούν να ολοκληρωθούν πριν από τις προθεσμίες τους και το υποσύνολο J^d αναφέρεται σε εργασίες που έχουν ήδη απορριφθεί και δεν θα

εκπληρώσουν τις προθεσμίες τους στο βέλτιστο χρονοδιάγραμμα . Στην επανάληψη k , το σύνολο J^c αναφέρεται στις θέσεις εργασίας $k + 1, k + 2, \dots, n$.

3.3.1 Αλγόριθμος: (Ελαχιστοποίηση του αριθμού των καθυστερημένων εργασιών)

Με λέξεις ο αλγόριθμος μπορεί να περιγραφεί ως εξής. Οι εργασίες προστίθενται στο σύνολο των εργασιών που μπορούν να ολοκληρωθούν πριν την προθεσμία τους κατά αυξανόμενη σειρά των προθεσμιών τους. Εάν η συμπερίληψη της εργασίας k στο σύνολο των προγραμματισμένων εργασιών υποδηλώνει ότι η εργασία k θα ολοκληρωθεί αργά, τότε η προγραμματισμένη εργασία με το μεγαλύτερο χρόνο επεξεργασίας, δηλ. Ας πούμε η εργασία l , σημειώνεται αργή και απορρίπτεται.

Ο αλγόριθμος δημιουργεί στο τελευταίο βήμα του n σύνολα εργασιών J_1, \dots, J_n . Το σύνολο J_k είναι ένα υποσύνολο εργασιών $\{1, \dots, k\}$, που αποτελείται από εκείνες τις εργασίες που είναι υποψήφιες για την εκπλήρωση των προθεσμιών τους στο τελικό βέλτιστο χρονοδιάγραμμα. Το σύνολο J_n αποτελείται από όλες τις εργασίες που πληρούν τις προθεσμίες τους με το βέλτιστο χρονοδιάγραμμα που δημιουργείται.

3.3.2 Θεώρημα: Ο αλγόριθμος δίνει ένα βέλτιστο χρονοδιάγραμμα για $1 \parallel \sum U_j$.

3.3.3 Παράδειγμα (Ελαχιστοποίηση του αριθμού των καθυστερημένων εργασιών)

Εξετάζουμε τις ακόλουθες 5 εργασίες.

jobs	1	2	3	4	5
p_j	7	8	4	6	6
d_j	9	17	18	19	21

Οι εργασίες 1 και 2 μπορούν να τοποθετηθούν πρώτη και δεύτερη στην ακολουθία με τις 2 εργασίες να είναι ολοκληρωμένες εγκαίρως. Η τοποθέτηση της εργασίας 3 στην τρίτη θέση προκαλεί προβλήματα. Ο χρόνος ολοκλήρωσης θα είναι 19, ενώ η προθεσμία της είναι 18. Ο αλγόριθμος 3.3.1 προβλέπει τη διαγραφή της εργασίας με τον μεγαλύτερο χρόνο επεξεργασίας στις τρεις πρώτες εργασίες. Συνεπώς, η εργασία 2 διαγράφεται και οι εργασίες 1 και 3 παραμένουν στις δύο πρώτες θέσεις. Αν τώρα η εργασία 4 ακολουθεί την εργασία 3, ολοκληρώνεται εγκαίρως στο 17. Ωστόσο, εάν η εργασία 5 ακολουθεί την εργασία 4, ολοκληρώνεται αργά. Ο αλγόριθμος τότε διαγράφει την εργασία με το μεγαλύτερο χρόνο επεξεργασίας μεταξύ των ήδη προγραμματισμένων, δηλαδή την εργασία 1. Έτσι το βέλτιστο πρόγραμμα είναι 3, 4, 5, 1, 2 με $\sum U_j = 2$.

3.4 Το πρόβλημα της συνολικής βραδύτερης περάτωσης - δυναμικός προγραμματισμός

Η αντικειμενική $\sum T_j$ είναι αυτή που είναι και σημαντική στην πράξη. Η ελαχιστοποίηση του αριθμού των καθυστερημένων εργασιών, $\sum U_j$, στην πράξη δεν μπορεί να είναι η μόνη αντικειμενική που μετράει αν πληρούνται οι προθεσμίες. Ορισμένες εργασίες ενδέχεται να πρέπει να περιμένουν για απαράδεκτα μεγάλο χρονικό διάστημα, εάν ελαχιστοποιηθεί ο αριθμός των καθυστερημένων εργασιών. Εάν αντίθετα το άθροισμα τις βραδύτερης περάτωσης ελαχιστοποιηθεί είναι λιγότερο πιθανό η αναμονή οποιασδήποτε εργασίας να είναι απαράδεκτα μεγάλη.

Το μοντέλο 1 || $\sum T_j$ έχει λάβει τεράστια προσοχή στη βιβλιογραφία. Για πολλά χρόνια η υπολογιστική του πολυπλοκότητα παρέμεινε ανοιχτή, έως ότου η συνδυαστικότητα καθιερώθηκε το 1990.

3.4.1 Λήμμα: Αν $p_j \leq p_k$ και $d_j \leq d_k$, τότε υπάρχει μια βέλτιστη ακολουθία στην οποία η εργασία j έχει προγραμματιστεί πριν από την εργασία k

3.4.2 Λήμμα: Οποιαδήποτε ακολουθία που είναι βέλτιστη για τη δεύτερη περίπτωση είναι επίσης βέλτιστη για τη πρώτη.

3.4.3 Λήμμα: Υπάρχει ένας ακέραιος δ , $0 \leq \delta \leq n - k$, έτσι ώστε να υπάρχει μια βέλτιστη ακολουθία S στην οποία η εργασία k προηγείται από όλες τις εργασίες j με $j \leq k + \delta$ και ακολουθείται από όλες τις εργασίες j με $j > k + \delta$.

Τα υποσύνολα J που χρησιμοποιούνται σε είναι ειδικού τύπου. Ένα υποσύνολο αποτελείται από όλες τις εργασίες σε ένα σύνολο $\{j, j + 1, \dots, l - 1, l\}$ με χρόνους επεξεργασίας μικρότερους από τον χρόνο επεξεργασίας p_k της εργασίας k . Ένα τέτοιο υποσύνολο συμβολίζεται με $J(j, l, k)$. Έστω ότι το $V(J(j, l, k), t)$ υποδηλώνει τη συνολική καθυστέρηση αυτού του υποσυνόλου υπό τη βέλτιστη ακολουθία, υποθέτοντας ότι αυτό το υποσύνολο ξεκινά στο χρόνο t .

3.4.4 Παράδειγμα (ελαχιστοποίηση της συνολικής βραδύτερης περάτωσης)

Εξετάζουμε τις ακόλουθες 5 εργασίες.

jobs	1	2	3	4	5
p_j	121	79	147	83	130
d_j	260	266	266	336	337

Η εργασία με τον μεγαλύτερο χρόνο επεξεργασίας είναι εργασία 3. Έτσι $0 \leq \delta \leq 2$. Η επαναλαμβανόμενη εξίσωση:

$$V(\{1,2,\dots,5\},0) = \min \begin{cases} V(J(1,3,3),0) + 81 + V(J(4,5,3),347) \\ V(J(1,4,3),0) + 164 + V(J(5,5,3),430) \\ V(J(1,5,3),0) + 294 + V(\emptyset,560) \end{cases}$$

Οι βέλτιστες ακολουθίες των μικρότερων συνόλων μπορούν να προσδιοριστούν εύκολα. Σαφώς, το $V(J(1,3,3),0)$ είναι μηδέν και υπάρχουν δύο αλληλουχίες που δίνουν μηδέν: 1,2 και 2,1. Η αξία του

$$V(J(4,5,3),347) = 94 + 223 = 317$$

και αυτό επιτυγχάνεται με την ακολουθία 4,5. Επίσης

$$V(J(1,4,3),0) = 0.$$

Η τιμή αυτή επιτυγχάνεται με τις ακολουθίες 1,2,4 και 2,1,4.

Η αξία του $V(J(5,5,3),430)$ ισούται με 560 μείον 337 που είναι 223. Τέλος,

$$V(J(1,5,3),0) = 76.$$

Αυτή η τιμή επιτυγχάνεται με τις ακολουθίες 1,2,4,5 και 2,1,4,5.

$$V(\{1,2,\dots,5\},0) = \min \begin{cases} 0 + 81 + 317 \\ 0 + 164 + 223 \\ 0 + 294 + 0 \end{cases}$$

Δύο βέλτιστες ακολουθίες είναι 1,2,4,5,3 και 2,1,4,5,3.

3.5 Το πρόβλημα της συνολικής βραδύτερης περάτωσης - ένα σύστημα προσέγγισης

Αφού το $1 \parallel \sum T_j$ είναι NP-hard, ούτε η μέθοδος branch-and-bound ούτε ο δυναμικός προγραμματισμός δεν μπορούν να δώσουν μια βέλτιστη λύση σε πολυωνυμικό χρόνο. Μπορεί λοιπόν να είναι ενδιαφέρον να έχουμε έναν αλγόριθμο που αποδίδει μια λύση που είναι κοντά στη βέλτιστη. Ένα σχήμα προσέγγισης A καλείται πλήρως πολυωνυμικό εάν η αξία της αντικειμενικής που επιτυγχάνει, ας πούμε $\sum T_j(A)$, ικανοποιεί

$$\sum T_j(A) \leq (1 + \epsilon) \sum T_j(OPT)$$

όπου $\sum T_j(OPT)$ είναι η αξία της αντικειμενικής σε ένα βέλτιστο χρονοδιάγραμμα. Επιπλέον, για να είναι το σύστημα προσέγγισης να είναι πλήρως πολυώνυμο, ο χειρότερος χρόνος λειτουργίας του πρέπει να περιορίζεται από ένα πολυώνυμο σταθερού βαθμού n και $1/\epsilon$. Το παράδειγμα που ακολουθεί δείχνει την επίλυση ενός προβλήματος ελαχιστοποίησης της συνολικής βραδύτερης περάτωσης με την κατασκευή ενός πλήρους συστήματος πολυωνυμικής χρονικής προσέγγισης (Fully Polynomial Time Approximation Scheme) (FPTAS).

3.5.1 Παράδειγμα (FPTAS ελαχιστοποίηση της συνολικής καθυστέρησης)

Θεωρούμε ένα μόνο μηχάνημα και 5 εργασίες.

jobs	1	2	3	4	5
p_j	1210	790	1470	830	1300
d_j	1996	2000	2660	3360	3370

Μπορεί να επαληθευτεί (μέσω δυναμικού προγραμματισμού) ότι η βέλτιστη ακολουθία είναι 1, 2, 4, 5, 3, και ότι η συνολική απόκλιση κάτω από αυτήν την βέλτιστη ακολουθία είναι 3700. Εφαρμόζοντας EDD $T_{max}(EDD) = 2230$. Αν το ϵ έχει επιλεχτεί 0,02, τότε $K = 2,973$. Τα δεδομένα ανακατασκευής είναι:

jobs	1	2	3	4	5
p_j	406	265	494	279	437
d_j	671.38	672.72	894.72	1130.17	1133.54

Η επίλυση αυτού του παραδείγματος χρησιμοποιώντας τη δυναμική διαδικασία προγραμματισμού που περιγράφεται στην Ενότητα 3.4 δίνει δύο βέλτιστες ακολουθίες: 1,2,4,5,3 και 2,1,4,5,3. Εάν η αλληλουχία 2,1,4,5,3 εφαρμόζεται στο αρχικό σύνολο δεδομένων, τότε η συνολική απόκλιση είναι 3704. Είναι σαφές ότι,

$$\sum T_j(2,1,4,5,3) \leq (1.02) \sum T_j(1,2,4,5,3).$$

3.6 Το πρόβλημα της συνολικής σταθμισμένης απόκλισης

Το πρόβλημα $1 \parallel \sum w_j T_j$ είναι μια σημαντική γενίκευση του προβλήματος $1 \parallel \sum T_j$. Δεκάδες ερευνητές έχουν ασχοληθεί με αυτό το πρόβλημα και έχουν πειραματιστεί με πολλές διαφορετικές προσεγγίσεις. Οι προσεγγίσεις κυμαίνονται από πολύ εξελιγμένες τεχνικές ηλεκτρονικών υπολογιστών σε αρκετά ακαθάριστα τεχνικά σχεδιασμένα κυρίως για εφαρμογή στην πράξη.

3.6.1 Λήμμα: Αν υπάρχουν δύο εργασίες j και k με $d_j \leq d_k$, $p_j \leq p_k$ και $w_j \geq w_k$, τότε υπάρχει μια βέλτιστη ακολουθία στην οποία η εργασία j εμφανίζεται πριν από την εργασία k .

3.6.2 Παράδειγμα (Ελαχιστοποίηση της συνολικής σταθμισμένης απόκλισης)

Θεωρούμε τις ακόλουθες 4 εργασίες.

jobs	1	2	3	4
w_j	4	5	3	5
p_j	12	8	15	9
d_j	16	26	25	27

Από το προηγούμενο λήμμα αμέσως ακολουθεί ότι σε μία βέλτιστη ακολουθία η εργασία 4 ακολουθεί την εργασία 2 και η εργασία 3 ακολουθεί την εργασία 1. Μόνο δύο εργασίες πρέπει να θεωρηθούν ως υποψήφιοι για την τελευταία θέση, δηλ. οι εργασίες 3 και 4. Οι κόμβοι του δέντρου branch-and-bound που πρέπει να μελετηθούν απεικονίζονται στο Σχήμα 3.1. Για να επιλεχτεί ένα κλαδί για αναζήτηση, τα όρια καθορίζονται και για τους δύο κόμβους στο επίπεδο 1.

Ένα χαμηλότερο όριο για μια βέλτιστη αλληλουχία μεταξύ των απογόνων του κόμβου (*, *, *, 4) μπορεί να επιτευχθεί εξετάζοντας το πρόβλημα μεταφοράς που περιγράφηκε πριν από την εφαρμογή στις εργασίες 1, 2 και 3. Οι λειτουργίες κόστους επιλέγονται ως εξής

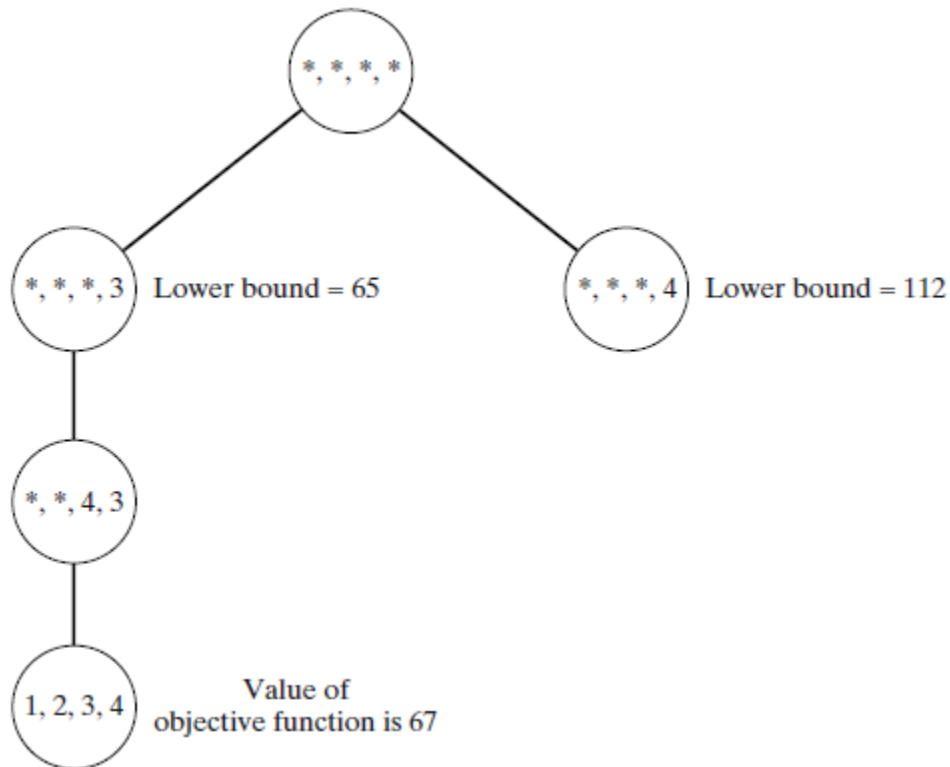
$$\begin{aligned}
c_{1k} &= 0, & k &= 1, \dots, 16 \\
c_{1k} &= 4, & k &= 17, \dots, 35 \\
c_{2k} &= 0, & k &= 1, \dots, 26 \\
c_{2k} &= 5, & k &= 27, \dots, 35 \\
c_{3k} &= 0, & k &= 1, \dots, 35 \\
c_{3k} &= 3, & k &= 26, \dots, 35
\end{aligned}$$

Η βέλτιστη κατανομή των τμημάτων εργασίας σε χρονικά διαστήματα τοποθετεί την εργασία 1 στις πρώτες 12 υποδοχές, την εργασία 2 στις θέσεις 19 έως 26 και την εργασία 3 στις θέσεις 13 έως 18 και 27 έως 35 (αυτή η βέλτιστη λύση μπορεί να βρεθεί με την επίλυση ενός προβλήματος μεταφοράς μπορεί βέβαια να βρεθεί και με δοκιμασία και λάθος). Το κόστος αυτής της κατανομής των τριών εργασιών είναι 3×9 (το κόστος κατανομής της εργασίας 3 στις θέσεις 27 έως 35). Προκειμένου να επιτευχθεί χαμηλότερο όριο για τον κόμβο, πρέπει να προστεθεί η καθυστέρηση της εργασίας 4. αυτό οδηγεί στο κατώτερο όριο $27 + 80$ το οποίο ισούται με 107.

Με παρόμοιο τρόπο μπορεί να ληφθεί ένα κατώτερο όριο για τον κόμβο (*, *, *, 3). Ένα κατώτερο όριο για ένα βέλτιστο χρονοδιάγραμμα για τις εργασίες 1, 2 και 4 αποδίδει 8, ενώ η καθυστέρηση της εργασίας 3 είναι 54 με αποτέλεσμα ένα όριο 62.

Δεδομένου ότι ο κόμβος (*, *, *, 3) φαίνεται να είναι ο πιο ελπιδοφόρος κόμβος, ο απόγονος αυτού του κόμβου θεωρείται πρώτος. Αποδεικνύεται ότι το καλύτερο χρονοδιάγραμμα που είναι εφικτό από αυτόν τον κόμβο είναι 1, 2, 4, 3 με αντικειμενική τιμή 64.

Από το γεγονός ότι το κατώτερο όριο για (*, *, *, 4) είναι 107 προκύπτει ότι 1, 2, 4, 3 είναι το καλύτερο συνολικό χρονοδιάγραμμα.



Σχήμα 3.1

3.7 Σχόλια κεφαλαίου

Όλα τα μοντέλα που εξετάστηκαν σε αυτό το κεφάλαιο έχουν κανονικές αντικειμενικές λειτουργίες. Αυτός είναι ένας από τους λόγους για τους οποίους τα περισσότερα από τα μοντέλα είναι σχετικά εύκολα.

Ορισμένα είναι επιλύσιμα μέσω κανόνων απλής προτεραιότητας (αποστολής), π.χ. WSPT, EDD. Τα περισσότερα από τα μοντέλα που δεν μπορούν να επιλυθούν μέσω απλών κανόνων προτεραιότητας εξακολουθούν να επιλύονται είτε σε πολυωνυμικό χρόνο είτε σε ψευδο-πολυωνυμικό χρόνο. Τα μοντέλα που είναι επιλύσιμα σε πολυωνυμικό χρόνο συνήθως αντιμετωπίζονται με δυναμικό προγραμματισμό, π.χ. $1 \mid \text{prec} \mid h_{max}, 1 \mid \mid \sum T_j$.

Ένα από τα έντονα συνδυαστικά προβλήματα που εξετάστηκαν σε αυτό το κεφάλαιο είναι $1 \mid \mid \sum w_j T_j$. Αυτό το πρόβλημα έχει λάβει μια τεράστια προσοχή στη βιβλιογραφία. Υπάρχουν δύο προσεγγίσεις για την επίτευξη των βέλτιστων λύσεων, δηλαδή του branch-and-bound, και του δυναμικού προγραμματισμού. Η ενότητα 3.6 παρουσιάζει μια προσέγγιση κλάδου και branch-and-bound. Αυτό το κεφάλαιο έχει επίσης δείξει εφαρμογή ενός πλήρους πολυωνυμικού χρονοδιαγράμματος προσέγγισης χρόνου (FPTAS) για ένα πρόβλημα προγραμματισμού μίας μηχανής. Κατά τη διάρκεια της τελευταίας δεκαετίας, τα Σχέδια Προσέγγισης Χρόνου Πολυωνύμου (PTAS) και τα Πλήρως Πολυωνυμικά Χρόνια Προσέγγιση (FPTAS) έλαβαν τεράστια προσοχή.

Αυτό το κεφάλαιο δεν παρουσιάζει όλες τις πιθανές διαδικασίες και τεχνικές που μπορούν να εφαρμοστούν σε προβλήματα προγραμματισμού ενός μηχανήματος. Μια σημαντική κατηγορία διαδικασιών λύσης αναφέρεται συχνά ως προγραμματισμός περιορισμού.

Έχουν αναπτυχθεί πολλές ευρετικές διαδικασίες που μπορούν να εφαρμοστούν σε προβλήματα προγραμματισμού ενός μηχανήματος. Αυτές οι διαδικασίες περιλαμβάνουν τους λεγόμενους σύνθετους κανόνες αποστολής καθώς και τεχνικές τοπικής αναζήτησης.

Κεφάλαιο 4

Μοντέλα παράλληλων μηχανών (ντετερμινιστικά)

Μια τράπεζα παράλληλων μηχανών είναι μια ρύθμιση που είναι σημαντική τόσο από την θεωρητική όσο και από την πρακτική άποψη. Από θεωρητική άποψη, είναι μια γενίκευση της μοναδικής μηχανής και μια ειδική περίπτωση του flexible flow shop. Από πρακτική άποψη, είναι σημαντικό επειδή η παράλληλη εμφάνιση των πόρων είναι κοινή στον πραγματικό κόσμο. Επίσης, οι τεχνικές για παράλληλες μηχανές χρησιμοποιούνται συχνά σε διαδικασίες αποσύνθεσης για συστήματα πολλαπλών σταδίων.

Σε αυτό το κεφάλαιο εξετάζονται διάφορες αντικειμενικές. Οι τρεις κύριες αντικειμενικές είναι η ελαχιστοποίηση του makespan, ο συνολικός χρόνος ολοκλήρωσης και η μέγιστη καθυστέρηση. Με ένα μόνο μηχανήμα η αντικειμενική makespan συνήθως ενδιαφέρει μόνο όταν υπάρχουν χρόνοι εγκατάστασης εξαρτώμενοι από την ακολουθία αλλιώς το makespan ισούται με το άθροισμα των χρόνων επεξεργασίας και είναι ανεξάρτητο από την ακολουθία. Όταν ασχολείται με παράλληλες μηχανές, το makespan γίνεται μία αντικειμενική σημαντικού ενδιαφέροντος. Στην πράξη, κάποιος πρέπει συχνά να αντιμετωπίσει το πρόβλημα της εξισορρόπησης του φορτίου στις παράλληλες μηχανές. Με την ελαχιστοποίηση του makespan ο προγραμματιστής εξασφαλίζει μια καλή ισορροπία.

Κάποιος μπορεί πράγματι να εξετάσει τον προγραμματισμό παράλληλων μηχανών ως διαδικασία δύο σταδίων. Πρώτον, πρέπει να καθορίσει ποιες εργασίες πρέπει να διατεθούν σε ποιες μηχανές. Δεύτερον, πρέπει να προσδιορίσει την ακολουθία των εργασιών που έχουν κατανεμηθεί σε κάθε μηχανή. Με την αντικειμενική του makespan, μόνο η διαδικασία κατανομής είναι σημαντική.

Στις παράλληλες μηχανές, οι προεκχωρήσεις παίζουν σημαντικότερο ρόλο απ' ό,τι με ένα μόνο μηχανήμα. Με ένα μόνο μηχανήμα, οι προεκχωρήσεις συνήθως παίζουν ρόλο μόνο όταν οι εργασίες ξεκινούν σε διαφορετικές χρονικές στιγμές. Αντίθετα, με τις παράλληλες μηχανές, οι προεκχωρήσεις είναι σημαντικές ακόμη και όταν ξεκινούν ταυτόχρονα όλες οι εργασίες.

Για τα περισσότερα μοντέλα που εξετάζονται σε αυτό το κεφάλαιο, υπάρχουν βέλτιστα χρονοδιαγράμματα που δεν είναι καθυστερημένα. Ωστόσο, εάν υπάρχουν μη σχετιζόμενες παράλληλες μηχανές και ο συνολικός χρόνος ολοκλήρωσης πρέπει να ελαχιστοποιηθεί χωρίς προεκχωρήσεις, τότε το βέλτιστο πρόγραμμα ενδέχεται να μην είναι καθυστερημένο.

Τα περισσότερα μοντέλα που εξετάζονται σε αυτό το κεφάλαιο εμπίπτουν στην κατηγορία των αποκαλούμενων προβλημάτων προγραμματισμού offline scheduling problems. Σε ένα offline scheduling problem, όλα τα δεδομένα (π.χ. χρόνοι επεξεργασίας, ημερομηνίες έναρξης, προθεσμίες) είναι γνωστές εκ των προτέρων και μπορούν να ληφθούν υπόψη στη διαδικασία βελτιστοποίησης. Αντίθετα, σε ένα online scheduling problem, τα δεδομένα προβλήματος δεν είναι γνωστά εκ των προτέρων. Ο χρόνος επεξεργασίας μιας εργασίας γίνεται γνωστός μόλις ολοκληρωθεί και μια ημερομηνία έναρξης γίνεται γνωστή μόνο τη στιγμή της έναρξης μιας εργασίας. Σαφώς, οι αλγόριθμοι για τα online scheduling problems τείνουν να είναι αρκετά διαφορετικοί από τους αλγόριθμους για τα offline scheduling problems. Το τελευταίο τμήμα αυτού του κεφαλαίου επικεντρώνεται στον ηλεκτρονικό προγραμματισμό παράλληλων μηχανών.

Τα χαρακτηριστικά επεξεργασίας και οι περιορισμοί που εξετάζονται σε αυτό το κεφάλαιο περιλαμβάνουν τους περιορισμούς προτεραιότητας καθώς και τις ρυθμιστικές λειτουργίες M_j . Σε όλο το κεφάλαιο αυτό υποτίθεται ότι $p_1 \geq \dots \geq p_n$.

4.1 Το πρόβλημα του Makespan χωρίς προεκχωρήσεις.

Το πρώτο πρόβλημα που εξετάζεται είναι το $Pm \parallel C_{max}$. Το πρόβλημα αυτό παρουσιάζει ενδιαφέρον διότι η ελαχιστοποίηση του makespan έχει ως αποτέλεσμα την εξισορρόπηση του φορτίου στις διάφορες μηχανές, πράγμα που αποτελεί σημαντικό στόχο στην πράξη.

Είναι εύκολο να δούμε ότι το $P2 \parallel C_{max}$ είναι NP-hard με τη συνηθισμένη έννοια. Τις τελευταίες δεκαετίες έχουν αναπτυχθεί πολλά ευρετικά για τον $Pm \parallel C_{max}$. Ένα τέτοιο ευρετικό περιγράφεται παρακάτω.

Ο κανόνας του μεγαλύτερου χρόνου επεξεργασίας (Longest Processing Time)(LPT) αναθέτει πρώτες τις μεγαλύτερες εργασίες m στις μηχανές m σε $t=0$. Μετά από αυτό, κάθε φορά που ελευθερώνεται ένα μηχάνημα, τοποθετείται στο μηχάνημα η μακρύτερη εργασία μεταξύ εκείνων που δεν έχουν ακόμη υποστεί επεξεργασία. Αυτός ο κανόνας προσπαθεί να τοποθετήσει τις μικρότερες εργασίες περισσότερο προς το τέλος του χρονοδιαγράμματος, όπου μπορούν να χρησιμοποιηθούν για την εξισορρόπηση των φορτίων.

Στο επόμενο θεώρημα παρουσιάζεται ένα ανώτατο όριο για

$$\frac{C_{max}(LPT)}{C_{max}(OPT)}$$

όπου το $C_{max}(LPT)$ υποδηλώνει το makespan του χρονοδιαγράμματος LPT και το $C_{max}(OPT)$ δηλώνει το makespan του (πιθανώς άγνωστου) βέλτιστου

χρονοδιαγράμματος. Αυτός ο τύπος ανάλυσης της χειρότερης περίπτωσης παρουσιάζει ενδιαφέρον καθώς δίνει μια ένδειξη για το πόσο καλά μπορεί να εκτελεστεί ο κανόνας καθώς και για το είδος των περιπτώσεων όπου ο κανόνας εκτελείται άσχημα.

4.1.1 Θεώρημα: Για $Pm || C_{max}$

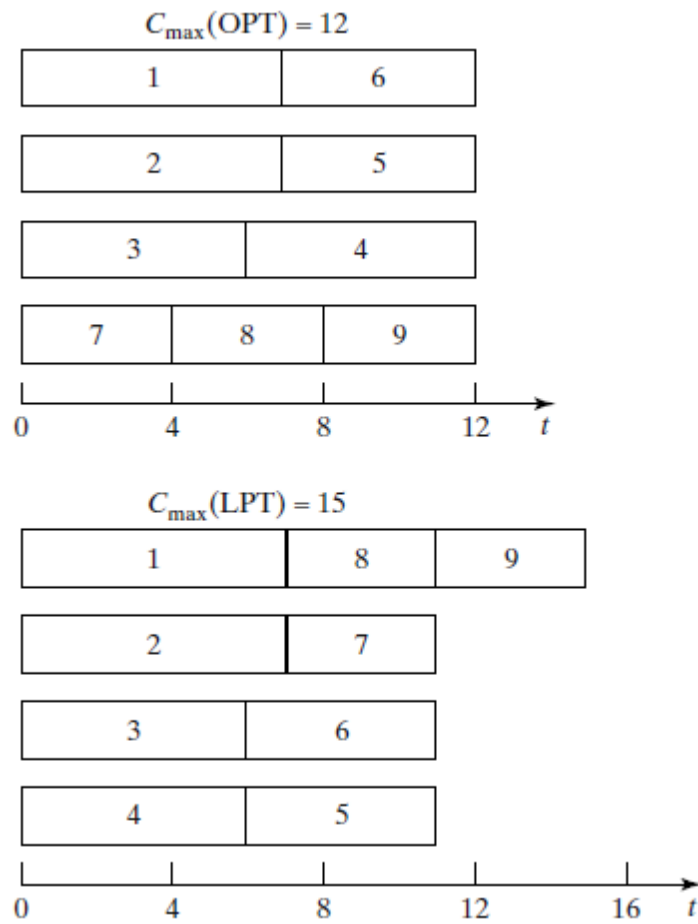
$$\frac{C_{max}(LPT)}{C_{max}(OPT)} \leq \frac{4}{3} - \frac{1}{3m}$$

4.1.2 Παράδειγμα (Παράδειγμα χειρότερης περίπτωσης LPT)

Θεωρούμε 4 παράλληλα μηχανήματα και 9 εργασίες, των οποίων οι χρόνοι επεξεργασίας δίδονται στον παρακάτω πίνακα:

jobs	1	2	3	4	5	6	7	8	9
p_j	7	7	6	6	5	5	4	4	4

Ο χρονοπρογραμματισμός των εργασιών σύμφωνα με το LPT καταλήγει σε $makespan = 15$. Μπορούμε να δείξουμε εύκολα ότι για αυτό το σύνολο των εργασιών μπορεί να βρεθεί ένα πρόγραμμα με ένα $makespan$ 12 (Σχήμα 4.1). Αυτή η συγκεκριμένη περίπτωση είναι επομένως η χειρότερη περίπτωση όταν υπάρχουν 4 παράλληλα μηχανήματα



Σχήμα 4.1 Παράδειγμα της χειρότερης περίπτωσης LPT

Σκεφτείτε τώρα το ίδιο πρόβλημα με τις εργασίες που υπόκεινται σε περιορισμούς προτεραιότητας, δηλ. $P_m \mid prec \mid C_{max}$. Από την άποψη της πολυπλοκότητας, το πρόβλημα αυτό πρέπει να είναι τουλάχιστον τόσο δύσκολο όσο το πρόβλημα χωρίς περιορισμούς προτεραιότητας. Προκειμένου να αποκτηθούν ορισμένες γνώσεις σχετικά με τις συνέπειες των περιορισμών προτεραιότητας, πρέπει να εξεταστούν ορισμένες ειδικές περιπτώσεις. Η ειδική περίπτωση με ένα μόνο μηχάνημα είναι σαφώς τετριμμένη. Αρκεί να κρατάμε το μηχάνημα συνεχώς απασχολημένο και το makespan θα είναι ίσο με το άθροισμα των χρόνων επεξεργασίας. Σκεφτείτε την ειδική περίπτωση όπου υπάρχει ένας απεριόριστος αριθμός παράλληλων μηχανών ή όπου ο αριθμός των μηχανών είναι τουλάχιστον τόσο μεγάλος όσο ο αριθμός των εργασιών, δηλαδή $m \geq n$. Αυτό το πρόβλημα μπορεί να συμβολίζεται με $P^\infty \mid prec \mid C_{max}$. Αυτό είναι ένα κλασικό πρόβλημα στον τομέα του σχεδιασμού έργων (project) και η μελέτη του έχει οδηγήσει στην ανάπτυξη των γνωστών(CPM) μέθοδος της κρίσιμης διαδρομής και (PERT) αξιολόγηση του έργου και ανασκόπηση της τεχνικής. Το βέλτιστο πρόγραμμα και το ελάχιστο makespan καθορίζονται με έναν πολύ απλό αλγόριθμο

4.1.3 Αλγόριθμος: (Ελαχιστοποίηση του makespan ενός project)

Προγραμματίζουμε τις εργασίες μία κάθε φορά ξεκινώντας από το χρόνο μηδέν. Κάθε φορά που έχει ολοκληρωθεί μια εργασία, ξεκινάμε όλες τις εργασίες από τις οποίες έχουν ολοκληρωθεί όλες οι προκαταρκτικές

εργασίες (δηλαδή όλες οι εργασίες που μπορούν να προγραμματιστούν).

Το γεγονός ότι αυτός ο αλγόριθμος οδηγεί σε ένα βέλτιστο πρόγραμμα μπορεί να παρουσιαστεί εύκολα. Η απόδειξη παραμένει ως άσκηση. Αποδεικνύεται ότι στο $P^\infty | \text{prec} | C_{max}$ η έναρξη της επεξεργασίας ορισμένων εργασιών συνήθως μπορεί να αναβληθεί χωρίς να αυξηθεί το makespan. Αυτές οι εργασίες αναφέρονται ως χαλαρές εργασίες. Οι εργασίες που δεν μπορούν να αναβληθούν αναφέρονται ως κρίσιμες εργασίες. Το σύνολο κρίσιμων εργασιών αναφέρεται ως η κρίσιμη διαδρομή. Για να προσδιοριστούν οι κρίσιμες εργασίες, εκτελείται η ίδια διαδικασία που εφαρμόζεται στον αλγόριθμο προς τα πίσω. Ξεκινάμε από το makespan, το οποίο είναι τώρα γνωστό, και εργαζόμαστε προς το μηδέν, ενώ ακολουθούμε τις σχέσεις προτεραιότητας. Με αυτόν τον τρόπο, όλες οι εργασίες ολοκληρώθηκαν στο αργότερο πιθανό χρόνο ολοκλήρωσής τους και συνεπώς ξεκίνησαν και στους τελευταίους χρόνους έναρξης. Οι εργασίες των οποίων οι αρχικοί πιθανοί χρόνοι εκκίνησης είναι ίσοι με τους αργότερους πιθανούς χρόνους έναρξης τους είναι οι κρίσιμες εργασίες.

4.1.4 Παράδειγμα (Ελαχιστοποίηση του makespan ενός project)

Θεωρούμε 9 εργασίες με τους ακόλουθους χρόνους επεξεργασίας.

jobs	1	2	3	4	5	6	7	8	9
p_j	4	9	3	3	6	8	8	12	6

Οι περιορισμοί προτεραιότητας απεικονίζονται στο Σχήμα 4.2.

Ο πρώτος χρόνος ολοκλήρωσης C'_j της εργασίας j μπορεί να υπολογιστεί εύκολα.

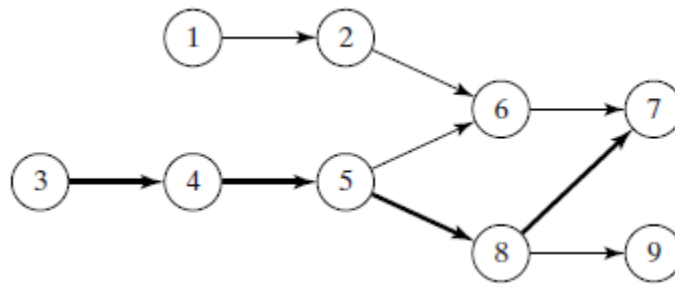
jobs	1	2	3	4	5	6	7	8	9
C'_j	4	13	3	6	12	21	32	24	30

Αυτό σημαίνει ότι το makespan είναι 32. Υποθέτοντας ότι το makespan είναι 32, οι τελευταίοι πιθανοί χρόνοι ολοκλήρωσης C''_j μπορούν να υπολογιστούν.

jobs	1	2	3	4	5	6	7	8	9
C''_j	7	16	3	6	12	24	32	24	32

Οι εργασίες των οποίων οι πιο σύντομοι χρόνοι ολοκλήρωσης είναι ίσοι με τους πιο αργούς πιθανούς χρόνους ολοκλήρωσης λέγεται ότι βρίσκονται στη κρίσιμη διαδρομή. Έτσι, η κρίσιμη διαδρομή είναι η αλυσίδα

$3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 7$



Σχήμα 4.2

4.2. Το πρόβλημα του Makespan με προεκχωρήσεις

Θεωρούμε το ίδιο πρόβλημα με αυτό που συζητήθηκε στην αρχή του προηγούμενου τμήματος, αλλά τώρα επιτρέπονται οι προεκχωρήσεις, δηλ. $P_m \mid pm \mid C_{max}$. Συνήθως, αλλά όχι πάντα, επιτρέποντας προεκχωρήσεις απλοποιείται η ανάλυση ενός προβλήματος. Αυτό συμβαίνει πράγματι για αυτό το πρόβλημα, όπου στην πραγματικότητα αποδεικνύεται ότι πολλά προγράμματα είναι βέλτιστα. Αρχικά, εξετάζουμε την ακόλουθη γραμμική προγραμματική διαμόρφωση του προβλήματος.

ελαχιστοποίηση της C_{max}

Όπου,

$$\sum_{i=1}^m x_{ij} = p_j, \quad j = 1, \dots, n$$

$$\sum_{i=1}^m x_{ij} \leq C_{max}, \quad j = 1, \dots, n$$

$$\sum_{j=1}^n x_{ij} \leq C_{max}, \quad i = 1, \dots, m$$

$$x_{ij} \geq 0 \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Η μεταβλητή x_{ij} αντιπροσωπεύει το συνολικό χρόνο που η εργασία j σπατάλησε στο μηχάνημα i . Το πρώτο σύνολο περιορισμών διασφαλίζει ότι κάθε εργασία λαμβάνει την απαιτούμενη ποσότητα επεξεργασίας. Το δεύτερο σύνολο περιορισμών εξασφαλίζει ότι το συνολικό ποσό επεξεργασίας κάθε εργασίας είναι μικρότερο ή ίσο με το makespan. Το τρίτο σετ εξασφαλίζει ότι το συνολικό ποσό επεξεργασίας σε κάθε μηχανή είναι μικρότερο από το makespan. Δεδομένου ότι η C_{max} είναι βασικά μια μεταβλητή απόφασης και όχι ένα στοιχείο του διανύσματος πόρων του γραμμικού προγράμματος, το δεύτερο και το τρίτο σύνολο περιορισμών μπορούν να ξαναγραφούν ως εξής:

$$C_{max} - \sum_{i=1}^m x_{ij} \geq 0, \quad j = 1, \dots, n$$

$$C_{max} - \sum_{j=1}^n x_{ij} \geq 0, \quad i = 1, \dots, m$$

4.2.1 Παράδειγμα (Σχηματισμός LP για ελαχιστοποίηση του Makespan με προεκχωρήσεις)

Θεωρούμε δύο μηχανές και τρεις εργασίες με $p_1 = 8$, $p_2 = 7$ και $p_3 = 5$. Υπάρχουν λοιπόν 7 μεταβλητές, δηλαδή x_{11} , x_{21} , x_{12} , x_{22} , x_{13} , x_{23} και C_{max} .

Το LP (γραμμικό πρόγραμμα) αναφέρεται σε ένα πρόβλημα βελτιστοποίησης στο οποίο η αντικειμενική και οι περιορισμοί είναι γραμμικές μεταβλητές απόφασης. Μπορούν να διατυπωθούν ως εξής:

Η αντικειμενική

$$\min C_{max}$$

Και οι περιορισμοί

$$x_{11} + x_{21} = p_1 = 8$$

$$x_{12} + x_{22} = p_2 = 7$$

$$x_{13} + x_{23} = p_3 = 5$$

$$C_{max} - x_{11} - x_{21} \geq 0$$

$$C_{max} - x_{12} - x_{22} \geq 0$$

$$C_{max} - x_{13} - x_{23} \geq 0$$

$$C_{max} - x_{11} - x_{12} - x_{13} \geq 0$$

$$C_{max} - x_{21} - x_{22} - x_{23} \geq 0$$

Οι συντελεστές της αντικειμενικής είναι

$$1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0$$

Ο πίνακας A με τους συντελεστές των περιορισμών παίρνει τη μορφή

$$\begin{array}{ccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & -1 \end{array}$$

$$\begin{array}{ccccccc} 1 & -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & 1 \end{array}$$

Η δεξιά πλευρά θα περιέχει

$$8 \ 7 \ 5 \ 0 \ 0 \ 0 \ 0$$

Ο πίνακας A είναι ένας πίνακας 0 και 1. Ο φορέας \bar{c} περιέχει έξι 0 και ένα μονό ένα 1. Ο φορέας \bar{b} περιέχει τους τρεις χρόνους επεξεργασίας και πέντε 0.

Η λύση είναι $C_{max} = 10$

4.3. Το πρόβλημα του συνολικού χρόνου ολοκλήρωσης χωρίς προεκχωρήσεις

Θεωρούμε παράλληλες μηχανές και n εργασίες. Θυμηθείτε ότι $p_1 \geq \dots \geq p_n$. Η αντικειμενική που πρέπει να ελαχιστοποιηθεί είναι ο συνολικός μη σταθμισμένος χρόνος ολοκλήρωσης $\sum C_j$. Από το Θεώρημα στη παράγραφο 3.1 προκύπτει ότι για ένα μόνο μηχάνημα ο κανόνας συντομότερης επεξεργασίας (SPT) ελαχιστοποιεί το συνολικό χρόνο ολοκλήρωσης. Αυτό το απλό αποτέλεσμα μηχανής μπορεί επίσης να παρουσιαστεί με διαφορετικό τρόπο αρκετά εύκολα.

Ας πούμε το $p_{(j)}$ υποδηλώνει τον χρόνο επεξεργασίας της εργασίας στη j θέση στην ακολουθία. Ο συνολικός χρόνος ολοκλήρωσης μπορεί στη συνέχεια να εκφράζεται ως

$$\sum C_j = np_{(1)} + (n-1)p_{(2)} + \dots + 2p_{(n-1)} + p_n.$$

Αυτό σημαίνει ότι υπάρχουν n συντελεστές $n, n-1, \dots, 1$ που θα αντιστοιχούν στους n διαφορετικούς χρόνους επεξεργασίας. Οι χρόνοι επεξεργασίας πρέπει να ταξινομούνται με τέτοιο τρόπο ώστε να ελαχιστοποιείται το άθροισμα των προϊόντων. Από την κοινή λογική προκύπτει ότι ο υψηλότερος συντελεστής, n , προσδιορίζει το μικρότερο χρόνο επεξεργασίας, p_n , ο δεύτερος υψηλότερος συντελεστής, $n-1$, προσδιορίζει το δεύτερο μικρότερο χρόνο επεξεργασίας p_{n-1} , και ούτω καθεξής. Αυτό σημαίνει ότι η SPT είναι η βέλτιστη.

Αυτός ο τύπος επιχειρήματος μπορεί να επεκταθεί και στην παράλληλη ρύθμιση του μηχανήματος.

4.3.1 Θεώρημα: ο κανόνας SPT είναι βέλτιστος για το $Pm \parallel \sum C_j$.

4.3.2 Παράδειγμα (Εφαρμογή του κανόνα WSPT)

Θεωρούμε δύο μηχανές και τρεις εργασίες.

jobs	1	2	3
p_j	1	1	3
w_j	1	1	3

Οι εργασίες 1 και 2 στο χρόνο μηδέν και η εργασία 3 στο χρόνο 1 οδηγεί σε ένα συνολικό σταθμισμένο χρόνο ολοκλήρωσης 14, ενώ η εργασία 3 στο χρόνο μηδέν και οι εργασίες 1 και 2 στην άλλη μηχανή οδηγεί σε συνολικό σταθμισμένο χρόνο ολοκλήρωσης 12. Σαφώς, με αυτό το σύνολο δεδομένων οποιοδήποτε χρονοδιάγραμμα μπορεί να θεωρηθεί ως WSPT. Ωστόσο, κάνοντας τα βάρη των εργασιών 1 και 2 ίσες με $1 - \epsilon$, δείχνει ότι το WSPT δεν δίνει κατ'ανάγκη ένα βέλτιστο χρονοδιάγραμμα.

Έχει αποδειχθεί στη βιβλιογραφία ότι το ευρετικό WSPT είναι ωστόσο πολύ καλό ευρετικό για το συνολικό σταθμισμένο χρόνο ολοκλήρωσης σε παράλληλες μηχανές.

4.4. Αντικειμενικές σχετιζόμενες με τις ημερομηνίες παράδοσης

Τα προβλήματα μεμονωμένων μηχανών με αντικειμενικές σχετιζόμενες με τις ημερομηνίες παράδοσης και οι οποίες είναι επιλύσιμες σε πολυωνυμικό χρόνο έχουν τυπικά τη μέγιστη καθυστέρηση ως αντικειμενική, π.χ. $1 \parallel L_{max}, 1 \parallel rmp \mid L_{max}$ και $1 \mid r_j, rmp \mid L_{max}$. Τα προβλήματα μεμονωμένου μηχανήματος με τη συνολική βραδύτερη περάτωση ή τη συνολική σταθμισμένη καθυστέρηση ως αντικειμενική τείνουν να είναι hard.

Είναι εύκολο να δούμε ότι από άποψη πολυπλοκότητας $Pm \parallel L_{max}$ δεν είναι τόσο εύκολο όσο $1 \parallel L_{max}$. Σκεφτείτε την ειδική περίπτωση όπου όλες οι εργασίες έχουν ημερομηνία παράδοσης 0. Η εύρεση ενός προγράμματος με ελάχιστο L_{max} ισοδυναμεί με $Pm \parallel C_{max}$ και συνεπώς είναι NP-hard.

Σκεφτείτε $Qm \mid rmp \mid L_{max}$. Αυτό το πρόβλημα είναι ένα από τα λίγα προβλήματα προγραμματισμού με παράλληλες μηχανές με αντικειμενική που σχετίζεται με την ημερομηνία παράδοσης και η οποία είναι επιλύσιμη σε πολυωνυμικό χρόνο. Ας υποθέσουμε ότι κάποιος πρέπει να ελέγξει αν υπάρχει εφικτό χρονοδιάγραμμα με $L_{max} = z$. Αυτό σημαίνει ότι για την εργασία j πρέπει ο χρόνος ολοκλήρωσης C_j να είναι μικρότερος ή ίσος με $d_j + z$. Ας υποθέσουμε ότι $d_j + z$ είναι μια σκληρή προθεσμία \bar{d}_j . Βρίσκοντας ένα εφικτό χρονοδιάγραμμα με όλες τις εργασίες που ολοκληρώνουν την επεξεργασία τους πριν από αυτές τις προθεσμίες ισοδυναμεί με την επίλυση του προβλήματος $Qm \mid r_j, rmp \mid C_{max}$. Για να το δούμε αυτό, αντιστρέφουμε την κατεύθυνση του χρόνου στο πρόβλημα της ημερομηνίας παράδοσης. Εφαρμόζουμε τον κανόνα LRPT-FM ξεκινώντας από την τελευταία προθεσμία και εργαζόμαστε προς τα πίσω. Οι προθεσμίες στο αρχικό πρόβλημα παίζουν το ρόλο των ημερομηνιών έναρξης στο αντίστροφο πρόβλημα που είναι τότε ισοδύναμο με $Qm \mid r_j, rmp \mid C_{max}$. Εάν εφαρμόσουμε τον κανόνα LRPT-FM προς τα πίσω, έχουμε ένα εφικτό χρονοδιάγραμμα με όλες τις εργασίες στο αρχικό πρόβλημα ξεκινώντας από χρόνο μεγαλύτερο ή ίσο με μηδέν, τότε υπάρχει ένα χρονοδιάγραμμα για $Qm \mid rmp \mid L_{max}$ με $L_{max} \leq z$. Για να βρεθεί η ελάχιστη L_{max} πρέπει να γίνει απλή αναζήτηση για να καθοριστεί η κατάλληλη ελάχιστη τιμή z .

4.4.1 Παράδειγμα (Ελαχιστοποίηση της μέγιστης καθυστέρησης με προεκχωρήσεις)

Θεωρούμε την ακόλουθη περίπτωση P2 | pmpr | L_{max} με 4 εργασίες. Οι χρόνοι επεξεργασίας και οι ημερομηνίες παράδοσης δίνονται στον παρακάτω πίνακα. Οι προεκχωρήσεις επιτρέπονται σε ακέραια χρονικά σημεία.

jobs	1	2	3	4
d_j	4	5	8	9
p_j	3	3	3	8

Πρώτον, πρέπει να ελεγχθεί εάν υπάρχει εφικτή λύση με $L_{max} = 0$. Τα δεδομένα του παραδείγματος που δημιουργούνται μέσω της αναστροφής του χρόνου καθορίζονται ως εξής. Οι ημερομηνίες έναρξης λαμβάνονται με τον καθορισμό της μέγιστης ημερομηνίας παράδοσης στο αρχικό πρόβλημα που είναι 9 και αντιστοιχεί στην εργασία 4. Η ημερομηνία έναρξης της εργασίας 4 στο νέο πρόβλημα ορίζεται στη συνέχεια ίση με 0. Οι ημερομηνίες έναρξης των υπόλοιπων εργασιών επιτυγχάνονται αφαιρώντας τις αρχικές ημερομηνίες παράδοσης από 9.

jobs	1	2	3	4
r_j	5	4	1	0
p_j	3	3	3	8

Η ερώτηση είναι τώρα: σε αυτό το νέο παράδειγμα μπορεί να δημιουργηθεί ένα πρόγραμμα με ένα makespan μικρότερο από 9; Η εφαρμογή του LRPT παράγει άμεσα ένα εφικτό χρονοδιάγραμμα.

4.5. Σχόλια κεφαλαίου

Αυτό το κεφάλαιο επικεντρώνεται πρωτίστως σε προβλήματα παράλληλων μηχανών που είτε είναι επιλύσιμα σε πολυωνυμικό χρόνο είτε έχουν ορισμένες ιδιότητες που παρουσιάζουν ενδιαφέρον. Αυτό το κεφάλαιο δεν αντιμετωπίζει τα πιο περίπλοκα προβλήματα παράλληλων μηχανών που είναι NP-hard και έχουν μικρή δομή.

Έχει πραγματοποιηθεί σημαντική έρευνα σε προβλήματα χρονοπρογραμματισμού με παράλληλες μηχανές που είναι NP-σκληρά. Έχουν αναπτυχθεί ποικίλες διατυπώσεις ακέραιου προγραμματισμού για $Rm || \sum w_j C_j$ και $Rm || \sum w_j U_j$. Αυτά τα ακέραια προγράμματα μπορούν να λυθούν με τη χρήση μιας ειδικής μορφής branch-and-bound που αποκαλείται branch-and-price και συχνά αναφέρεται ως δημιουργία στήλης. Ωστόσο, υπάρχουν πολλά άλλα προβλήματα παράλληλων μηχανών που είναι πιο περίπλοκα και δεν έχουν ακόμη αντιμετωπιστεί με ακριβείς μεθόδους.

Κεφάλαιο 5

Στοχαστικός Χρονοπρογραμματισμός

Τα περιβάλλοντα παραγωγής στον πραγματικό κόσμο υπόκεινται σε πολλές πηγές αβεβαιότητας ή τυχαιότητας. Πηγές αβεβαιότητας που μπορεί να έχουν σημαντικό αντίκτυπο μπορεί να είναι καταστροφές μηχανών και απροσδόκητες εκκινήσεις εργασιών υψηλής προτεραιότητας. Μια άλλη πηγή αβεβαιότητας έγκειται στους χρόνους επεξεργασίας, οι οποίοι συχνά δεν είναι ακριβώς γνωστοί εκ των προτέρων. Ένα καλό πρότυπο για ένα πρόβλημα προγραμματισμού θα πρέπει να αντιμετωπίσει αυτές τις μορφές αβεβαιότητας.

Υπάρχουν διάφοροι τρόποι με τους οποίους μπορούν να διαμορφωθούν τέτοιες μορφές τυχαιότητας. Για παράδειγμα, θα μπορούσε κανείς να μοντελοποιήσει τη δυνατότητα καταστροφών μηχανών ως αναπόσπαστο μέρος των χρόνων επεξεργασίας. Αυτό μπορεί να γίνει με τροποποίηση της κατανομής των χρόνων επεξεργασίας για να ληφθεί υπόψη η πιθανότητα βλάβης. Εναλλακτικά, μπορεί κανείς να μοντελοποιήσει τις κατανομές ως ξεχωριστή στοχαστική διαδικασία, που καθορίζει πότε είναι διαθέσιμο ένα μηχάνημα και πότε δεν είναι.

5.1. Δομή και συμβολισμός

Εν συνεχεία, θεωρείται ότι οι κατανομές των χρόνων επεξεργασίας, οι ημερομηνίες δέσμευσης και οι ημερομηνίες παράδοσης είναι γνωστές εκ των προτέρων, δηλαδή στο χρόνο μηδέν. Το πραγματικό αποτέλεσμα ή η πραγματοποίηση ενός τυχαίου χρόνου επεξεργασίας γίνεται μόνο μετά την ολοκλήρωση της επεξεργασίας η πραγματοποίηση της ημερομηνίας έναρξης ή της ημερομηνίας παράδοσης γίνεται γνωστή μόνο εκείνη τη χρονική στιγμή κατά την οποία πραγματοποιείται πραγματικά.

Σε αυτό το μέρος χρησιμοποιείται η ακόλουθη συμβολική αναφορά. Οι τυχαίες μεταβλητές κεφαλαιοποιούνται, ενώ οι πραγματικές τιμές είναι μικρές. Η εργασία j έχει τις ακόλουθες ποσότητες ενδιαφέροντος που σχετίζονται με αυτήν.

- X_{ij} = ο χρόνος τυχαίας επεξεργασίας της εργασίας j στο μηχάνημα i . Αν η εργασία j πρόκειται να υποβληθεί σε επεξεργασία μόνο σε ένα μηχάνημα ή αν έχει τους ίδιους χρόνους επεξεργασίας σε κάθε ένα από τα μηχανήματα που μπορεί να επισκεφθεί, ο δείκτης i παραλείπεται
- $1/\lambda_{ij}$ = η μέση ή αναμενόμενη τιμή της τυχαίας μεταβλητής X_{ij} .
- R_j = η τυχαία ημερομηνία έναρξης της εργασίας j .
- D_j = η τυχαία ημερομηνία παράδοσης της εργασίας j .
- w_j = το βάρος (ή ο παράγοντας σπουδαιότητας) της εργασίας j .

Αυτός ο συμβολισμός δεν είναι εντελώς ανάλογος με το συμβολισμό που χρησιμοποιείται για τα ντετερμινιστικά μοντέλα. Ο λόγος για τον οποίο ο X_{ij} χρησιμοποιείται ως χρόνος επεξεργασίας στο στοχαστικό προγραμματισμό οφείλεται στο γεγονός ότι το P αναφέρεται συνήθως σε μια πιθανότητα. Το βάρος w_j , παρόμοιο με αυτό των ντετερμινιστικών μοντέλων, είναι ουσιαστικά ισοδύναμο με το κόστος διατήρησης της εργασίας j στο σύστημα για μία μονάδα χρόνου. Στην τεκμηριωμένη λογική θεωρία, η οποία σχετίζεται στενά με τον στοχαστικό προγραμματισμό, το c_j χρησιμοποιείται συχνά για το βάρος ή το κόστος της εργασίας j . Το c_j και το w_j είναι ισοδύναμα.

Κεφάλαιο 6

Μοντέλο Μεμονωμένης Μηχανής (Στοχαστικό)

Τα στοχαστικά μοντέλα, ειδικά με εκθετικούς χρόνους επεξεργασίας, μπορεί συχνά να περιέχουν περισσότερη δομή από τα ντετερμινιστικά αντίστοιχά τους και μπορεί να οδηγήσουν σε αποτελέσματα τα οποία, κατά την πρώτη όψη, φαίνονται εκπληκτικά.

Σε αυτό το κεφάλαιο εξετάζονται καταρχήν μοντέλα μεμονωμένου μηχανήματος με αυθαίρετους χρόνους επεξεργασίας χωρίς προεκχωρήσεις. Στη συνέχεια, εξετάζονται περιπτώσεις με προεκχωρήσεις, ακολουθούμενες από μοντέλα όπου οι χρόνοι επεξεργασίας είναι δείκτες πιθανότητας.

6.1. Το πρόβλημα των αυθαίρετων διανομών χωρίς προεκχωρήσεις

Για μια σειρά από στοχαστικά προβλήματα, η επίτευξη της βέλτιστης πολιτικής ισοδυναμεί με την επίλυση ενός προκαθορισμένου προβλήματος προγραμματισμού. Συνήθως, όταν υπάρχει μια τέτοια σχέση ισοδυναμίας, το ντετερμινιστικό αντίστοιχο μπορεί να επιτευχθεί αντικαθιστώντας όλες τις τυχαίες μεταβλητές με τα μέσα τους. Το βέλτιστο χρονοδιάγραμμα για το ντετερμινιστικό πρόβλημα στη συνέχεια ελαχιστοποιεί την αντικειμενική της στοχαστικής εκδοχής σε αναμονή.

Μια τέτοια περίπτωση είναι όταν η αντικειμενική στο ντετερμινιστικό αντίστοιχο είναι γραμμικό σε $p_{(j)}$ και $w_{(j)}$, όπου $p_{(j)}$ και $w_{(j)}$ υποδηλώνουν το χρόνο επεξεργασίας και το βάρος της εργασίας στην j -η θέση στην ακολουθία.

Η παρατήρηση αυτή υποδηλώνει ότι είναι εύκολο να βρεθεί το βέλτιστο πρόγραμμα μετάθεσης για το στοχαστικό αντίστοιχο του $1 \parallel \sum w_j C_j$, όταν ο χρόνος επεξεργασίας της εργασίας j είναι X_j , από μια αυθαίρετη κατανομή F_j , και ο αντικειμενική είναι $E(\sum w_j C_j)$. Αυτό το πρόβλημα οδηγεί στην στοχαστική εκδοχή του κανόνα WSPT, η οποία ακολουθεί τις εργασίες σε φθίνουσα σειρά του λόγου $w_j/E(X_j)$ ή του $\lambda_j w_j$. Στη συνέχεια, ο κανόνας αυτός αναφέρεται είτε ως κανόνας σταθμισμένου βραχύτερου αναμενόμενου χρόνου επεξεργασίας (WSEPT) είτε ως κανόνας "λw".

6.1.1 Θεώρημα: Ο κανόνας του WSEPT ελαχιστοποιεί το αναμενόμενο άθροισμα των σταθμισμένων χρόνων ολοκλήρωσης στην κατηγορία των μη διακοπτόμενων πολιτικών στατικής λίστας καθώς και στην κατηγορία μη διακοπτόμενης δυναμικής πολιτικής.

6.1.2 Παράδειγμα (Βέλτιστη πολιτική με τυχαίες μεταβλητές που είναι DCR)

Εξετάζουμε τις εργασίες με τον χρόνο επεξεργασίας X_j κατανομημένες ως εξής. Ο χρόνος επεξεργασίας X_j είναι 0 με πιθανότητα p_j και κατανέμεται σύμφωνα με μια εκθετική με ρυθμό λ_j με πιθανότητα $1 - p_j$. Σαφώς, αυτή η κατανομή είναι DCR καθώς είναι ένα μείγμα δύο εκθετικών με τιμές ∞ και λ_j . Η αντικειμενική που πρέπει να ελαχιστοποιηθεί είναι το αναμενόμενο άθροισμα των σταθμισμένων χρόνων ολοκλήρωσης. Η βέλτιστη διακοπτόμενη δυναμική πολιτική είναι σαφής. Όλες οι n εργασίες n πρέπει να δοκιμαστούν για ένα χωριστό δευτερόλεπτο στο χρόνο μηδέν, προκειμένου να καθοριστεί ποιες εργασίες έχουν μηδενικούς χρόνους επεξεργασίας. Αν μια εργασία δεν έχει μηδενικό χρόνο επεξεργασίας, βγαίνει αμέσως από το μηχάνημα. Αφού έχει καθορίσει με τον τρόπο αυτό ποιες εργασίες έχουν μηδενικούς χρόνους επεξεργασίας, οι υπόλοιπες εργασίες κατατάσσονται κατά φθίνουσα σειρά $\lambda_j w_j$.

6.1.3 Θεώρημα: Ο κανόνας EDD ελαχιστοποιεί την αναμενόμενη μέγιστη καθυστέρηση για τους αυθαίρετα κατανομημένους χρόνους επεξεργασίας και τις ντετερμινιστικές ημερομηνίες παράδοσης στην κατηγορία μη διακοπτόμενων πολιτικών στατικής λίστας, την τάξη των μη διακοπτόμενων δυναμικών πολιτικών και την τάξη των διακοπτόμενων δυναμικών πολιτικών.

6.1.4 Παράδειγμα (Εφαρμογή του κανόνα EDD)

Εξετάζουμε δύο εργασίες με καθορισμένες ημερομηνίες παράδοσης. Οι κατανομές χρόνου επεξεργασίας των εργασιών είναι διακριτές:

$$P(X_1 = 1) = P(X_1 = 2) = P(X_1 = 4) = \frac{1}{3}$$

και

$$P(X_2 = 2) = P(X_2 = 4) = \frac{1}{2}$$

Η ημερομηνία παράδοσης της πρώτης εργασίας είναι $D_1 = 1$ και της δεύτερης εργασίας είναι $D_2 = 4$. Τώρα

$$E(\max(L_1, L_2)) = E(\max(L_1, L_2) | X_1 = 1, X_2 = 2) \times P(X_1 = 1, X_2 = 2)$$

$$\begin{aligned}
& +E(\max(L_1, L_2) | X_1 = 1, X_2 = 4) \times P(X_1 = 1, X_2 = 4) \\
& +E(\max(L_1, L_2) | X_1 = 2, X_2 = 2) \times P(X_1 = 2, X_2 = 2) \\
& +E(\max(L_1, L_2) | X_1 = 2, X_2 = 4) \times P(X_1 = 2, X_2 = 4) \\
& +E(\max(L_1, L_2) | X_1 = 4, X_2 = 2) \times P(X_1 = 4, X_2 = 2) \\
& +E(\max(L_1, L_2) | X_1 = 4, X_2 = 4) \times P(X_1 = 4, X_2 = 4) \\
& = (0 + 1 + 1 + 2 + 3 + 4) \frac{1}{6} = \frac{11}{6}.
\end{aligned}$$

Μπορεί εύκολα να επαληθευθεί ότι η εργασία 2 πρώτα και η εργασία 1 έπειτα οδηγούν σε μεγαλύτερο $E(\max(L_1, L_2))$.

Ωστόσο σε οποιαδήποτε δεδομένη αλληλουχία $E(L_{\max}) = E(\max(L_1, L_2))$ δεν πρέπει απαραίτητα να είναι ίση με το $\max(E(L_1), E(L_2))$. Υπό την ακολουθία 1,2,

$$E(L_1) = 0 \times \frac{1}{3} + 1 \times \frac{1}{3} + 3 \times \frac{1}{3} = \frac{4}{3},$$

Ενώ

$$E(L_2) = \frac{1}{3} \left(\frac{1}{2} \times 0 + \frac{1}{2} \times 1 \right) + \frac{1}{3} \left(\frac{1}{2} \times 0 + \frac{1}{2} \times 2 \right) + \frac{1}{3} \left(\frac{1}{2} \times 2 + \frac{1}{2} \times 4 \right) = \frac{3}{2}.$$

Επομένως $\max(E(L_1), E(L_2)) = 3/2$, το οποίο είναι μικρότερο από $E(\max(L_1, L_2))$.

Κεφάλαιο 7

Μοντέλα παράλληλων μηχανών (στοχαστικά)

Αυτό το κεφάλαιο ασχολείται με μοντέλα παράλληλων μηχανών που είναι στοχαστικά αντίστοιχα μοντέλων που αναλύονται στο Κεφάλαιο 4. Το σώμα γνώσης στην στοχαστική περίπτωση είναι κάπως λιγότερο εκτεταμένο από ό, τι στην ντετερμινιστική περίπτωση.

Τα αποτελέσματα επικεντρώνονται κυρίως στο αναμενόμενο makespan, το συνολικό αναμενόμενο χρόνο ολοκλήρωσης και τον αναμενόμενο αριθμό καθυστερημένων εργασιών. Στη συνέχεια, ο αριθμός των μηχανών περιορίζεται συνήθως σε δύο.

Η πρώτη ενότητα ασχολείται με μη διακοπτόμενα μοντέλα. Τα αποτελέσματα σε αυτή την ενότητα προκύπτουν από επιχειρήματα ανταλλαγής. Το δεύτερο τμήμα επικεντρώνεται σε διακοπτόμενα μοντέλα. Τα αποτελέσματα σε αυτή την ενότητα επιτυγχάνονται μέσω δυναμικών προσεγγίσεων προγραμματισμού.

7.1 Το πρόβλημα του makespan και του συνολικού χρόνου ολοκλήρωσης χωρίς προεκχωρήσεις

Αυτή η ενότητα λαμβάνει υπόψη τις βέλτιστες πολιτικές στις τάξεις μη διακοπτόμενων πολιτικών στατικής λίστας και μη διακοπτόμενων δυναμικής πολιτικής. Δεδομένου ότι δεν επιτρέπονται προεκχωρήσεις, η βασική τεχνική για τον προσδιορισμό βέλτιστων πολιτικών βασίζεται σε επιχειρήματα αντιστοίχισης ανά ζεύγη.

Πρώτον, η εκθετική κατανομή εξετάζεται λεπτομερώς καθώς οι ειδικές ιδιότητές της διευκολύνουν σημαντικά την ανάλυση.

Σκεφτείτε δύο παράλληλες μηχανές και η εργασίες. Ο χρόνος επεξεργασίας της εργασίας j ισούται με την τυχαία μεταβλητή X_j , η οποία είναι εκθετικά κατανομημένη με ρυθμό λ_j . Η αντικειμενική είναι να ελαχιστοποιηθεί το E (C_{max}). Σημειώστε ότι αυτό το πρόβλημα είναι ένα στοχαστικό αντίστοιχο του P2 || C_{max} , η οποία είναι γνωστό ότι είναι NP-hard. Οι διαδικασίες προγραμματισμού με εκθετικά κατανομημένους χρόνους επεξεργασίας συχνά έχουν ιδιότητες που οι εξειδικευμένοι αντισταθμιστές τους δεν έχουν. Αποδεικνύεται ότι αυτό ισχύει και για τις παράλληλες μηχανές.

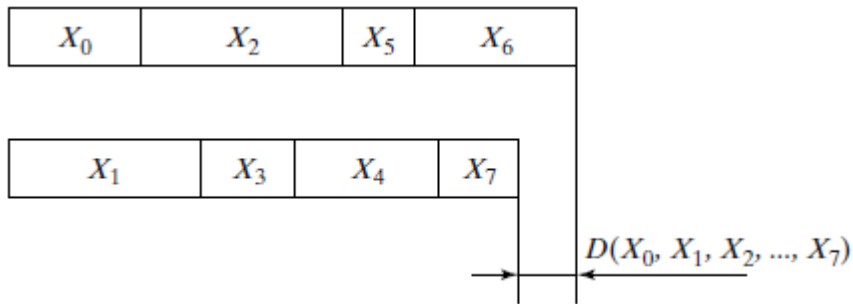
Υιοθετείται μια πολιτική μη διακοπτόμενης στατικής λίστας. Οι εργασίες τίθενται σε λίστα και κατά το χρόνο μηδέν οι δύο εργασίες στην κορυφή της λίστας ξεκινούν την επεξεργασία τους στις δύο μηχανές. Όταν ένα μηχάνημα γίνει ελεύθερο, η επόμενη εργασία στη λίστα τοποθετείται στο μηχάνημα. Δεν έχει καθοριστεί εκ των προτέρων ποια μηχανή θα επεξεργαστεί κάθε εργασία, ούτε είναι γνωστό ποια εργασία θα είναι η τελευταία που θα ολοκληρωθεί.

Το Z_1 δηλώνει την ώρα που ολοκληρώνεται η δεύτερη έως την τελευταία εργασία, δηλ. Η πρώτη φορά που ένα μηχάνημα γίνεται ελεύθερο με τον κατάλογο εργασιών να είναι άδειος. Αυτή τη στιγμή το άλλο μηχάνημα επεξεργάζεται ακόμα την τελευταία του εργασία. Το Z_2 δηλώνει τον χρόνο ολοκλήρωσης της τελευταίας εργασίας στο άλλο μηχάνημα (δηλ., ισούται με το C_{max} του κατασκευαστή). Η διαφορά D είναι ίση με $Z_2 - Z_1$. Είναι σαφές ότι η τυχαία μεταβλητή D εξαρτάται από το χρονοδιάγραμμα. Είναι εύκολο να δούμε ότι η ελαχιστοποίηση του $E(D)$ ισοδυναμεί με την ελαχιστοποίηση του $E(C_{max})$. Αυτό προκύπτει από το γεγονός ότι

$$Z_1 + Z_2 = 2C_{max} - D = \sum_{j=1}^n X_j$$

που είναι μια σταθερή ανεξάρτητη από το χρονοδιάγραμμα.

Στη συνέχεια, εξετάζουμε ένα ελαφρώς γενικότερο πρόβλημα δύο μηχανών για λόγους που θα γίνουν σαφείς αργότερα. Θεωρείται ότι μία από τις μηχανές δεν είναι διαθέσιμη στο χρόνο μηδέν και καθίσταται διαθέσιμη μόνο μετά από τυχαίο χρόνο X_0 , κατανομημένος εκθετικά με ρυθμό λ_0 . Η τυχαία μεταβλητή X_0 μπορεί να θεωρηθεί ως ο χρόνος επεξεργασίας μιας πρόσθετης εργασίας που υπερισχύει και πρέπει να πάει πρώτη. Έστω ότι η D ($X_0, X_1, X_2, \dots, X_n$) υποδηλώνει την τυχαία μεταβλητή D , υπό την προϋπόθεση ότι, στο χρόνο μηδέν, μια εργασία με εναπομένοντα χρόνο επεξεργασίας X_0 επεξεργάζεται σε ένα μηχάνημα και μια εργασία με χρόνο επεξεργασίας X_1 αρχίζει από την άλλη. Όταν μία από τις δύο μηχανές απελευθερώνει μια εργασία με χρόνο επεξεργασίας X_2



Σχήμα 7.1

αρχίζει, και ούτω καθεξής (Σχήμα 7.1). Το επόμενο λήμμα εξετάζει το ζήτημα της D για την αλλαγή ενός προγράμματος με την εναλλαγή των δύο διαδοχικών εργασιών 1 και 2.

7.1.1 **Λήμμα:** Για κάθε λ_0 και για $\lambda_1 = \min(\lambda_1, \lambda_2, \dots, \lambda_n) \leq E(D(X_0, X_1, X_2, \dots, X_n))$

7.1.2 **Θεώρημα:** Αν υπάρχουν δύο μηχανές παράλληλες και οι χρόνοι επεξεργασίας κατανέμονται εκθετικά, τότε ο κανόνας LEPT ελαχιστοποιεί το αναμενόμενο makespan στην κλάση των μη διακοπτόμενων πολιτικών στατικής λίστας.

7.1.3 **Θεώρημα:** Ο κανόνας LEPT ελαχιστοποιεί το αναμενόμενο makespan στην κλάση των μη διακοπτόμενων πολιτικών στατικής λίστας όταν υπάρχουν δύο μηχανές παράλληλες και όταν οι χρόνοι επεξεργασίας κατανέμονται σύμφωνα με ένα μείγμα δύο εκθετικών με τιμές λ_1 και λ_2 .

7.1.4 **Λήμμα:** Για την αυθαίρετη λ_0 , αν $p_{1j} \geq p_{2j}$, δηλ. $E(X_1) \geq E(X_2)$, τότε $E(D(X_0, X_1, X_2, \dots, X_n)) \leq E(D(X_0, X_2, X_1, \dots, X_n))$

7.1.5 **Παράδειγμα (Αντίθεση προς το βέλτιστο της LEPT με αυθαίρετους χρόνους επεξεργασίας)**

Το p_{1j} υποδηλώνει ότι η εργασία j είναι εκθετικά κατανεμημένη με ρυθμό λ_1 και p_{2j} την πιθανότητα που κατανέμεται με τον ρυθμό λ_2 . Υποθέτουμε $\lambda_1 < \lambda_2$. Η πιθανότητα ότι ο χρόνος επεξεργασίας της εργασίας j είναι μηδέν είναι $p_{0j} = 1 - p_{1j} - p_{2j}$. Μέσω παρόμοιων επιχειρημάτων όπως αυτά που χρησιμοποιούνται στο λήμμα 7.1.4 και το Θεώρημα 7.1.3. μπορεί να φανεί ότι για να ελαχιστοποιηθεί το αναμενόμενο makespan οι εργασίες στη βέλτιστη πολιτική μη διακοπτόμενης στατικής λίστας πρέπει να παραγγελθούν σε φθίνουσα σειρά του p_{1j} / p_{2j} . Οι εργασίες με μηδενικούς χρόνους επεξεργασίας πάλι δεν παίζουν κάποιο ρόλο στο πρόγραμμα. Αυτό σημαίνει ότι η βέλτιστη ακολουθία δεν είναι απαραίτητα LEPT.

7.2 Το πρόβλημα της Makespan και του συνολικού χρόνου ολοκλήρωσης με τις προμήθειες

Τα συνολικά επιχειρήματα ανταλλαγής βασικά χρησιμοποιούνται για τον καθορισμό των βέλτιστων πολιτικών στην κατηγορία των μη διακοπτόμενων πολιτικών στατικής λίστας. Αφού προσδιορίσουμε μια βέλτιστη πολιτική μη διακοπτόμενου στατικού καταλόγου, μπορεί συχνά να υποστηριχθεί ότι αυτή η πολιτική είναι επίσης βέλτιστη στην κατηγορία των μη διακοπτόμενων δυναμικών πολιτικών και ενδεχομένως στην κατηγορία των διακοπτόμενων δυναμικών πολιτικών.

Μια δυναμική απόδειξη τύπου προγραμματισμού κατασκευάζεται μέσα στην τάξη των διακοπτόμενων δυναμικών πολιτικών. Αφού αποκτήσουμε το αποτέλεσμα ότι η μη διακοπτόμενη πολιτική LEPT ελαχιστοποιεί το αναμενόμενο makespan στην τάξη των διακοπτόμενων δυναμικών πολιτικών, συμπεραίνεται ότι είναι επίσης βέλτιστη στην κατηγορία των μη διακοπτόμενων δυναμικών πολιτικών καθώς και στην κατηγορία των μη διακοπτόμενων πολιτικών στατικής λίστας.

Αυτή η προσέγγιση μπορεί να χρησιμοποιηθεί για να αποδειχθεί ότι το LEPT ελαχιστοποιεί το makespan για m παράλληλες μηχανές. Θα παρουσιαστεί για δύο παράλληλα μηχανήματα, αφού η σημείωση είναι τότε πολύ απλούστερη.

Υποθέστε $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$. Το $V(J)$ υποδηλώνει την αναμενόμενη τιμή του ελάχιστου απαιτούμενου χρόνου (δηλαδή, με τη βέλτιστη πολιτική) για να ολοκληρώσει όλες τις εργασίες δεδομένου ότι όλες οι εργασίες στο σύνολο $J = j_1, \dots, j_l$ έχουν ήδη ολοκληρωθεί. Αν $J = \emptyset$, τότε $V(J)$ δηλώνεται απλά με V . Το $V^*(J)$ υποδηλώνει την ίδια ποσότητα χρόνου σύμφωνα με την πολιτική LEPT. Παρομοίως, το V^* υποδηλώνει την αναμενόμενη τιμή του υπολειπόμενου χρόνου ολοκλήρωσης στο LEPT όταν δεν έχει ακόμη ολοκληρωθεί καμία δουλειά.

7.2.1 Θεώρημα: Η μη διακοπτόμενη πολιτική LEPT ελαχιστοποιεί το αναμενόμενο makespan στην κατηγορία των διακοπτόμενων δυναμικών πολιτικών

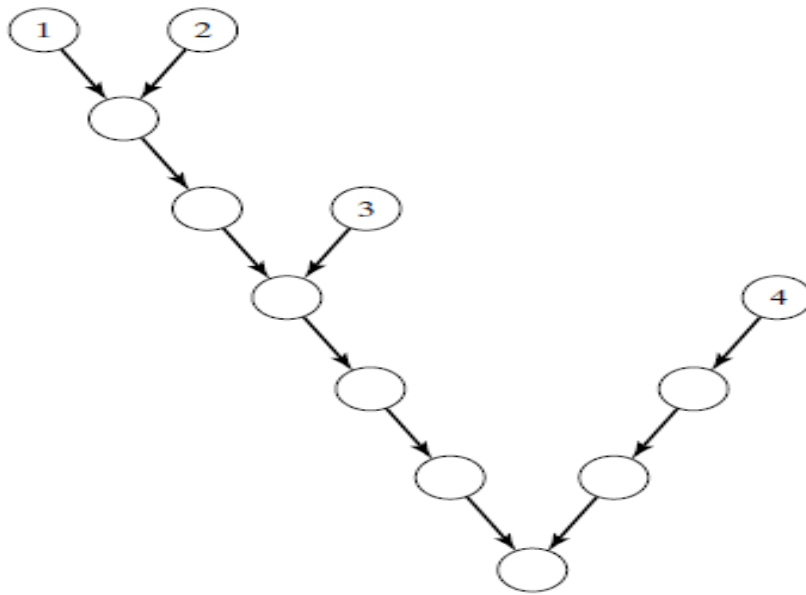
7.2.2 Θεώρημα: Η μη διακοπτόμενη πολιτική SEPT ελαχιστοποιεί το συνολικό αναμενόμενο χρόνο ολοκλήρωσης στην κατηγορία των διακοπτόμενων δυναμικών πολιτικών.

7.2.3 Θεώρημα: Ο μη διακοπτόμενος κανόνας CP ελαχιστοποιεί το αναμενόμενο makespan στην τάξη μη διακοπτόμενης δυναμικής πολιτικής και στην κατηγορία διακοπτόμενων δυναμικών πολιτικών

7.2.4 Παράδειγμα (Αντίκτυπος προς βέλτιστο κανόνα CP με τρεις μηχανές)

Εξετάζουμε τρία μηχανήματα και 12 εργασίες. Οι εργασίες είναι όλες i.i.d. εκθετικές με τη μέση τιμή 1 και υπόκειται στους περιορισμούς προτεραιότητας που περιγράφονται στο σχήμα 7.2. Ο προγραμματισμός σύμφωνα με τον κανόνα CP θα

θέσει τις εργασίες 1, 2 και 3 σε χρόνο μηδέν στις τρεις μηχανές. Ωστόσο, η απλή άλγεβρα δείχνει ότι ξεκινώντας από τις εργασίες 1, 2 και 4 οδηγεί σε ένα μικρότερο αναμενόμενο makespan.



Σχήμα 7.2

Επίλογος

Ο χρονοπρογραμματισμός, ως διαδικασία λήψης αποφάσεων, διαδραματίζει σημαντικό ρόλο στα περισσότερα συστήματα παραγωγής καθώς και στα περισσότερα περιβάλλοντα επεξεργασίας πληροφοριών. Είναι επίσης σημαντικός στις ρυθμίσεις μεταφοράς και διανομής και σε άλλους τύπους υπηρεσιών.

Ο χρονοπρογραμματισμός μπορεί να οριστεί ως "καθορισμός του πότε και που πρέπει να εκτελεσθεί κάθε εργασία που απαιτείται για την κατασκευή του έργου". Ορίζεται επίσης ως "καθορισμός των χρόνων κατά τις οποίες πρέπει να ξεκινήσει και να ολοκληρωθεί κάθε συμβάν ή διαδικασία που περιλαμβάνει μια διαδικασία". Ο βασικός στόχος του χρονοπρογραμματισμού είναι ο προγραμματισμός της σειράς εργασιών έτσι ώστε η παραγωγή να μπορεί να προγραμματιστεί συστηματικά προς το τέλος της ολοκλήρωσης όλων των προϊόντων κατά την προβλεπόμενη ημερομηνία.

Βιβλιογραφία

E.H.L. Aarts and J.K. Lenstra (eds.) (1997) Local Search and Combinatorial

Optimization, J. Wiley, New York.

J.O. Achugbue and F.Y. Chin (1982) "Scheduling the Open Shop to Minimize Mean Flow Time", *SIAM Journal of Computing*, Vol. 11, pp. 709–720.

J. Adams, E. Balas and D. Zawack (1988) "The Shifting Bottleneck Procedure for Job Shop Scheduling", *Management Science*, Vol. 34, pp. 391–401.

H.H. Adelsberger and J.J. Kanet (1991) "The LEITSTAND - A New Tool for Computer-Integrated-Manufacturing", *Production and Inventory Management Journal*, Vol. 32, pp. 43–48.

L. Adler, N.M. Fraiman, E. Kobacker, M. Pinedo, J.C. Plotnicoff and T.-P. Wu (1993) "BPSS: A Scheduling System for the Packaging Industry", *Operations Research*, Vol. 41, pp. 641–648.

A. Agnetis, P. B. Mirchandani, D. Pacciarelli and A. Pacifici (2004) "Scheduling Problems with Two Competing Agents", *Operations Research*, Vol. 52, pp. 229–242.

A.K. Agrawala, E.G. Coffman, Jr., M.R. Garey and S.K. Tripathi (1984) "A Stochastic Optimization Algorithm Minimizing Exponential Flow Times on Uniform Processors", *IEEE Transactions on Computers*, C-33, pp. 351–356.

C. Akkan and S. Karabati (2004) "The Two-Machine Flowshop Total Completion Time Problem: Improved Lower Bounds and a Branch-and-Bound Algorithm", *European Journal of Operational Research*, Vol. 159, pp. 420–429.

R. Akkiraju, P. Keskinocak, S. Murthy, F. Wu (1998) "A New Decision Support System for Paper Manufacturing", in *Proceedings of the Sixth International Workshop on Project Management and Scheduling (1998)*, pp. 147–150, Bogazici University Printing Office, Istanbul, Turkey.

R. Akkiraju, P. Keskinocak, S. Murthy, F. Wu (2001) "An Agent based Approach to Multi Machine Scheduling", *Journal of Applied Intelligence*, Vol. 14, pp. 135–144.

M.S. Akturk and E. Gorgulu (1999) "Match-Up Scheduling under a Machine Breakdown", *European Journal of Operational Research*, Vol. 112, pp. 81–97.

A. Allahverdi, C. Ng, T. Cheng and M. Kovalyov (2008) "A Survey of Scheduling Problems with Setup Times", *European Journal of Operational Research*, Vol. 187, pp. 985–1032.

N. Alon, Y. Azar, G.J. Woeginger and T. Yadid (1998) "Approximation Schemes for Scheduling on Parallel Machines", *Journal of Scheduling*, Vol. 1, pp. 55–66.

D. Applegate and W. Cook (1991) "A Computational Study of the Job-Shop Scheduling Problem", *ORSA Journal on Computing*, Vol. 3, pp. 149–156.

M. Arguello (1994) "Review of Scheduling Software", *Technology Transfer 93091822A-XFR*, SEMATECH, Austin, Texas.

N. Asadathorn (1997) *Scheduling of Assembly Type of Manufacturing Systems: Algorithms and Systems Development*, Ph.D Thesis, Department of Industrial Engineering, New Jersey Institute of Technology, Newark, New Jersey.

H. Atabakhsh (1991) "A Survey of Constraint Based Scheduling Systems Using an Artificial Intelligence Approach", *Artificial Intelligence in Engineering*, Vol. 6, No. 2, pp. 58–73.

H. Aytug, S. Bhattacharyya, G.J. Koehler and J.L. Snowdon (1994) "A Review of Machine Learning in Scheduling", *IEEE Transactions on Engineering*

- Management, Vol. 41, pp. 165–171.
- H. Aytug, M. Khouja and F.E. Vergara (2003) “Use of Genetic Algorithms to Solve Production and Operations Management Problems: a Review”, *International Journal of Production Research*, Vol. 41, pp. 3955–4009.
- T.P. Bagchi (1999) *Multiobjective Scheduling by Genetic Algorithms*, Kluwer Academic Publishers, Boston.
- P.C. Bagga (1970) “n-Job, 2-Machine Sequencing Problem with Stochastic Service”, *Opsearch*, Vol. 7, pp. 184–197.
- K.R. Baker (1974) *Introduction to Sequencing and Scheduling*, John Wiley, NY.
- K.R. Baker (1975) “A Comparative Survey of Flowshop Algorithms”, *Operations Research*, Vol. 23, pp. 62–73.
- K.R. Baker (1995) *Elements of Sequencing and Scheduling*, K. Baker, Amos Tuck School of Business Administration, Dartmouth College, Hanover, NH 03755.
- K.R. Baker and G.D. Scudder (1990) “Sequencing with Earliness and Tardiness Penalties: A Review”, *Operations Research*, Vol. 38, pp. 22–36.
- K.R. Baker and J. C. Smith (2003) “A Multiple-Criterion Model for Machine Scheduling”. *Journal of Scheduling*, Vol. 6, pp. 7–16.
- K.R. Baker and D. Trietsch (2009) *Principles of Sequencing and Scheduling*, John Wiley, New York.