FORTH-ICS / TR-046 Φεβρουάριος 1992

Το πρόβλημα της μέγιστης καθυστέρησης σε μία μηχανή με χρόνους εξάρμωσης

Θαλής Γεωργίου

Περίληψη

Στην εργασία αυτή αναλύουμε το πρόβλημα της ελαχιστοποίησης της μέγιστης καθυστέρησης N εργασιών σε μια μηχανή με χρόνους εξάρμωσης. Οι εργασίες είναι διαθέσιμες την χρονική στιγμή μηδέν και κατανέμονται σε B ομάδες. Χρόνοι εξάρμωσης ανεξάρτητοι ακολουθίας απαιτούνται για την προσαρμογή της μηχανής στις συνθήκες κατεργασίας κάθε ομάδας.

Προβλήματα χρονικού προγραμματισμού όπως το πρόβλημα της μέγιστης καθυστέρησης εμφανίζονται στο προγραμματισμό της παραγωγικής διαδικασίας σε βιομηχανίες. Η παρούσα εργασία εντάσσεται σε μια ευρύτερη προσπάθεια ανάπτυξης διαλογικών συστημάτων λεπτομεριακού προγραμματισμού (κυρίως σε βιομηχανίες πρώτων υλών). Αλγόριθμοι προτείνουν βέλτιστες ή προσεγγιστικές λύσεις ως προς κάποια βασικά κριτήρια. Οι προτάσεις των αλγορίθμων αναθεωρούνται από τον χρήστη του συστήματος που αποφασίζει για το τελικό πρόγραμμα της παραγωγής.

Η επίλυση του προβλήματος που πραγματευόμαστε εδώ βασίζεται στην ιδέα της δημιουργίας παρτίδων (ακολουθία εργασιών από την ίδια ομάδα, που εκτελούνται συνεχόμενες σε κάποια βέλτιστη δίαταξη εργασιών) και στην ανάλυση της κατανομής της καθυστέρησης σε ακολουθίες εργασιών. Το πρόβλημα της μέγιστης καθυστέρησης χωρίς χρόνους εξάρμωσης είναι απλό και λύνεται σε χρόνο O(NlogN). Οταν υπάρχουν και χρόνοι εξάρμωσης τότε είναι NP-hard και είναι αναμενόμενη η δυσκολία εύρεσης βέλτιστης λύσης. Δύο αλγόριθμοι διαμερισμού και φραγής προτείνονται για την εύρεση της βέλτιστης λύσης. Ο πρώτος κατασκευάζει το βέλτιστο πρόγραμμα τοποθετώντας εργασίες από την αρχή προς το τέλος ενώ ο δεύτερος αντίστροφα. Επίσης δίνεται και ένας προσεγγιστικός αλγόριθμος για την επίλυση προβλημάτων μεγάλου μεγέθους που εμφανίζονται συνήθως στην πράξη. Ακολουθούν πειραματικά αποτελέσματα για ένα σύνολο τυχαίων προβλημάτων καθώς και τα συμπεράσματα για το πρόβλημα. Κρίσιμοι

---

† Institute of Computer Science, F.O.R.T.H., Greece
E-mail: thalis@csi.forth.gr

παράγοντες για το πρόβλημα είναι η κατανομή των προθεσμιών, ο αριθμός των ομάδων καθώς και ο αριθμός των εργασιών ανά ομάδα. Παρόλο που το πρόβλημα είναι NP-hard τα πειραματικά αποτελέσματα έδειξαν πως η φύση του προβλήματος είναι τέτοια ώστε να δέχεται καλές προσεγγιστικές λύσεις. Προβλήματα μερικών εκατοντάδων εργασιών μπορούν να λυθούν σε χρόνο ενός λεπτού με μέση απόκλιση από το βέλτιστο μικρότερη του 2%.

Τέλος αναφερόμαστε σε προβλήματα χρονικού προγραμματισμού που η λύση τους σχετίζεται ή ανάγεται στην λύση του προβλήματος που παρουσιάζουμε, καθώς και σε γενικεύσεις των προτεινόμενων αλγορίθμων.

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΜΕΓΙΣΤΗΣ ΚΑΘΥΣΤΕΡΗΣΗΣ ΣΕ ΜΙΑ ΜΗΧΑΝΗ ΜΕ ΧΡΟΝΟΥΣ ΕΞΑΡΜΩΣΗΣ

Θαλής Γεωργίου

Μεταπτυχιακή Εργασία

ΗΡΑΚΛΕΙΟ, ΚΡΗΤΗ
Δεκέμβριος 1991

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟ ΠΡΟΒΛΗΜΑ ΤΗΣ ΜΕΓΙΣΤΗΣ ΚΑΘΥΣΤΕΡΗΣΗΣ ΣΕ ΜΙΑ ΜΗΧΑΝΗ ΜΕ ΧΡΟΝΟΥΣ ΕΞΑΡΜΩΣΗΣ

Εργασία που υποβλήθηκε από τον
ΘΑΛΗ ΚΙΜΩΝΟΣ ΓΕΩΡΓΙΟΥ
ως μερική απαίτηση για την απόκτηση του
ΔΙΠΛΩΜΑΤΟΣ ΜΕΤΑΠΤΥΧΙΑΚΗΣ ΕΞΕΙΔΙΚΕΥΣΗΣ

Ηράκλειο, Δεκέμβριος 1991

Συγγραφέας : . . . .
Τμήμα Επιστ. Υπολογιστών, 12 Δεκεμβρίου 1991

Εισηγητική Επιτροπή

. . . .
Αναπλ. Καθ. Πάνος Κωνσταντόπουλος, Επόπτης

. . . .
Αναπλ. Καθ. Καίτη Χούστη, Μέλος

. . . .
Επικ. Καθ. Κώστας Κουρκουμπέτης, Μέλος

Δεκτή : . . . .
Καθ. Στέλιος Ορφανουδάκης,
Πρόεδρος Επιτρ. Μεταπτυχιακών Σπουδών

Το πρόβλημα της Μέγιστης Καθυστέρησης σε μια Μηχανή με Χρόνους Εξάρμωσης

Θαλής Γεωργίου
Μεταπτυχιακή Εργασια

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Στην εργασία αυτή αναλύουμε το πρόβλημα της ελαχιστοποίησης της μέγιστης καθυστέρησης N εργασιών σε μια μηχανή με χρόνους εξάρμωσης. Οι εργασίες είναι διαθέσιμες την χρονική στιγμή μηδέν και κατανέμονται σε B ομάδες. Χρόνοι εξάρμωσης ανεξάρτητοι ακολουθίας απαιτούνται για την προσαρμογή της μηχανής στις συνθήκες κατεργασίας κάθε ομάδας.

Προβλήματα χρονικού προγραμματισμού όπως το πρόβλημα της μέγιστης καθυστέρησης εμφανίζονται στο προγραμματισμό της παραγωγικής διαδικασίας σε βιομηχανίες. Η παρούσα εργασία εντάσσεται σε μια ευρύτερη προσπάθεια ανάπτυξης διαλογικών συστημάτων λεπτομεριακού προγραμματισμού (κυρίως σε βιομηχανίες πρώτων υλών). Αλγόριθμοι προτείνουν βέλτιστες ή προσεγγιστικές λύσεις ως προς κάποια βασικά κριτήρια. Οι προτάσεις των αλγορίθμων αναθεωρούνται από τον χρήστη του συστήματος που αποφασίζει για το τελικό πρόγραμμα της παραγωγής.

Η επίλυση του προβλήματος που πραγματευόμαστε εδώ βασίζεται στην ιδέα της δημιουργίας παρτίδων (ακολουθία εργασιών από την ίδια ομάδα, που εκτελούνται συνεχόμενες σε κάποια βέλτιστη διάταξη εργασιών) και στην ανάλυση της κατανομής της καθυστέρησης σε ακολουθίες εργασιών. Το πρόβλημα της μέγιστης καθυστέρησης χωρίς χρόνους εξάρμωσης είναι απλό και λύνεται σε χρόνο O(NlogN). Οταν υπάρχουν και χρόνοι εξάρμωσης τότε είναι NP-hard και είναι αναμενόμενη η δυσκολία εύρεσης βέλτιστης λύσης. Δύο αλγόριθμοι διαμερισμού και φραγής προτείνονται για την εύρεση της βέλτιστης λύσης. Ο πρώτος κατασκευάζει το βέλτιστο πρόγραμμα τοποθετώντας εργασίες από την αρχή προς το τέλος ενώ ο δεύτερος αντίστροφα. Επίσης δίνεται και ένας προσεγγιστικός αλγόριθμος για την επίλυση προβλημάτων μεγάλου μεγέθους που εμφανίζονται συνήθως στην πράξη. Ακολουθούν πειραματικά αποτελέσματα για ένα σύνολο τυχαίων προβλημάτων καθώς και τα συμπεράσματα για το πρόβλημα. Κρίσιμοι παράγοντες για το πρόβλημα είναι η κατανομή των προθεσμιών, ο αριθμός των ομάδων καθώς και ο αριθμός των εργασιών ανά ομάδα. Παρόλο που το πρόβλημα είναι NP-hard τα πειραματικά αποτελέσματα έδειξαν πως η φύση του προβλήματος είναι τέτοια ώστε να δέχεται καλές προσεγγιστικές λύσεις. Προβλήματα μερικών εκατοντάδων εργασιών μπορούν να λυθούν σε χρόνο ενός λεπτού με μέση απόκλιση από το βέλτιστο μικρότερη του 2%.

Τέλος αναφερόμαστε σε προβλήματα χρονικού προγραμματισμού που η λύση τους σχετίζεται ή ανάγεται στην λύση του προβλήματος που παρουσιάζουμε, καθώς και σε γενικεύσεις των προτεινόμενων αλγορίθμων.

Επόπτης : Πάνος Κωνσταντόπουλος
Αναπλ. Καθ. Επιστήμης Υπολογιστών, Παν. Κρήτης

The problem of Maximum Tardiness in One Machine
with Set-Up Times

Thalis Georgiou
Master of Science Thesis

Department of Computer Science
University of Crete

Abstract

In this paper we consider the problem of sequencing groups of jobs in which there is a set-up time associated with switching from jobs in one group to another. We consider the case of a single machine and sequence independent set-up times. A cost function (maximum tardiness) evaluates a schedule by assigning it a cost.

Scheduling problems with set-up times appear in the production planning area. This work is a part of a general attempt concerning the development of an interactive information system for detailed scheduling. Algorithms propose optimal or near optimal solutions according to some basic criteria. The proposals of the algorithms might be reconsidered by the user of the system who is the one that makes the decision upon the final production program.

The solution of the maximum tardiness problem is based on the idea of lotts creation (sequence of jobs coming from the same group, which are executed one after the other in an optimal program) and the analysis of tardiness distribution in jobs sequences. The ploblem of maximum tardiness without set-up times is simple and can be solved in $O(Nlog(N))$ time where N is the number of jobs. When we have set-up times it becomes $NP-hard$ and it is expected to be difficult to solve it optimally. Two branch and bound algorithms are proposed in order to find an optimal solution. The first creates the optimal schedule placing jobs from the begining to the end, while the second works in the opposite order. Another approximate algorithm is proposed as well, which solves large-size problems that appear mostly in real time applications. Then we present some experimental results over a set of random problems and our conclusions. Critical factors about the problem is the due-date's distribution, the number of groups and the number of jobs per group. Eventhough the problem is $NP\text{–}hard$ the experimental results showed that the nature of the problem is such that we can apply heuristic algorithms in order to have real good approximate solutions. Problems in the order of hundreds of jobs can be solved in one second CPU-time with mean deviation from the optimal solution less than 2%.

Finally, we refer to scheduling problems whose solution reletes or it can be reduced to the solution of the problem presented here and to generalizations of the algorithms proposed here.

Supervisor : Panos Constantopoulos
Assoc. Professor. of Computer Science, Univ. of Crete

στους γονείς μου

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Πάνο Κωνσταντόπουλο, για τις πολύτιμες συμβουλές του, τη καθοδήγηση και τη συμπαράστασή του.

Επίσης θα ήθελα να ευχαριστήσω τα μέλη της ομάδας Συστημάτων Στήριξης Αποφάσεων για τις συζητήσεις και τις υποδείξεις τους, κατά τις τακτικές εβδομαδιαίες συναντήσεις μας. Ιδιαίτερα ευχαριστώ τον Νίκο Τσατσάκη για τίς ώρες που αφιέρωσε στην ανταλλαγή απόψεων και ιδεών. Επίσης ευχαριστώ τον Κοσμά Χαριτωνίδη για τις χρήσιμες παρατηρήσεις του κατά την διόρθωση της εργασίας.

Θα ήθελα ακόμα να ευχαριστήσω την συνάδελφο Καλλιόπη Χαλκιά για την σημαντική της βοήθεια τόσο στην διαμόρφωση των ιδεών της παρούσας εργασίας όσο και για τις πολύτιμες συμβουλές της κατά την διόρθωσή της.

Τέλος θα ήθελα να ευχαριστήσω το Ινστιτούτο Πληροφορικής του Ι.Τ.Ε. για την οικονομική ενίσχυση και την υλικοτεχνική υποστήριξη που μου παρείχε κατά την διάρκεια των μεταπτυχιακών σπουδών μου.

Περιεχόμενα

1. Εισαγωγή

Τα προβλήματα χρονικού προγραμματισμού αποτελούν μια σημαντική κλάση των συνδυαστικών προβλημάτων βελτιστοποίησης με άμεσο πρακτικό ενδιαφέρον. Στόχος τους είναι η βέλτιστη εκτέλεση ενός συνόλου εργασιών, κάνοντας χρήση ενός συνόλου πόρων (μηχανές, εργαλεία κ.α.) ώστε να τηρούνται συγκεκριμένοι περιορισμοί (προτεραιότητες, προθεσμίες κ.α.). Ο όρος βέλτιστη εκτέλεση αναφέρεται σε σχέση με ένα κριτήριο επίδοσης ή μια συνάρτηση κόστους (συνολικός χρόνος επεξεργασίας, μέσος χρόνος επεξεργασίας, μέγιστη καθυστέρηση, αριθμός καθυστερημένων εργασιών και διάφορα άλλα καθώς και συνδυασμοί αυτών).

Δύο είναι οι βασικοί λόγοι που οδήγησαν πολλούς ερευνητές να ασχοληθούν με τα προβλήματα χρονικού προγραμματισμού. Αφενός παρουσιάζουν μεγάλο πρακτικό ενδιαφέρον διότι εμφανίζονται στο βραχυπρόθεσμο προγραμματισμό παραγωγής και στον προγραμματισμό εργασιών σε υπολογιστικά συστήματα. Αφετέρου τα προβλήματα αυτά αποτελούν πρόκληση μια και στην συντριπτική τους πλειοψηφία είναι NP-hard.

Στην παρούσα εργασία αναλύουμε ένα υποσύνολο προβλημάτων του προβλήματος διατεταγμένων κατηγοριών -ορισμένο από τους Monma και Potts [MoPo-89]- και προτείνουμε ένα αλγόριθμο Διαμερισμού και Φραγής για την επίλυσή τους. Επίσης παρουσιάζεται και ένας προσεγγιστικός αλγόριθμος που με μεγάλη πιθανότητα δίνει βέλτιστη λύση.

1.1. Ταξινόμηση των προβλημάτων χρονικού προγραμματισμού

Ενας από τους συνηθέστερους τρόπους ταξινόμησης των προβλημάτων χρονικού προγραμματισμού ακολουθεί το γενικό σχήμα

$$\alpha / \beta / \gamma / \delta$$

όπου

α: ο αριθμός των εργασιών για προγραμματισμό,

β: ο αριθμός των μηχανών,

γ: περιορισμοί πάνω στα μεγέθη του προβλήματος και

δ: το κριτήριο βελτιστοποίησης.

Για παράδειγμα το $N/1/seq-dep/C_{\max} \mid T_{\max}=0$ είναι το πρόβλημα όπου ζητάμε να βρούμε εφικτή διάταξη ($T_{\max}=0$) που να ελαχιστοποιεί τον χρόνο συμπλήρωσης N εργασιών οι οποίες εκτελούνται πάνω σε μια μηχανή δεδομένου ότι ο χρόνος αλλαγής κατεργασίας εξαρτάται από την εργασία που η μηχανή επεξεργαζόταν την αμέσως προηγούμενη στιγμή. Μερικά από τα γνωστότερα προβλήματα είναι τα $N/1//\sum_{i\in J} C_i \mid T_{\max}=0$, $N/1//L_{\max}$, $N/1//\sum_{i\in J} C_i$ και $N/1//\sum_{i\in J} T_i$. Το πρώτο από τα προβλήματα αναφέρεται στην εύρεση εφικτής διάταξης N εργασιών που να ελαχιστοποιεί τον μέσο χρόνο περάτωσης. Τα επόμενα τρία αναφέρονται στην εύρεση διάταξης που ελαχιστοποιεί την μέγιστη βραδύτητα, τον μέσο χρόνο περάτωσης και την μέση καθυστέρηση αντίστοιχα.

Αν και ο τρόπος της ταξινόμησης δεν παρέχει όλες τις λεπτομέρειες που συνθέτουν το πρόβλημα, δίνει πολύ συνοπτικά τα πλέον βασικά χαρακτηριστικά του.

1.2. Το πρόβλημα $\mathbf{N/1/seq-dep/f(\pi)}$

Το πρόβλημα $N/1/seq-dep/f(\pi)$ έχει ως εξής:

την χρονική στιγμή 0 δίνονται N εργασίες γιά εκτέλεση πάνω σε μια μηχανή. Οι N εργασίες κατανέμονται σε B ομάδες. Με J συμβολίζουμε το σύνολο όλων των εργασιών. Κάθε ομάδα b, με 1≤b≤B έχει N_b το πλήθος εργασίες αυθαίρετα σημειωμένες 1η, 2η, ..., N_b-οστή. Η i-οστή εργασία της ομάδας b απαιτεί χρόνο επεξεργασίας $p_{ib}>0$. Χρόνος εξάρμωσης S_{bc} απαιτείται όταν μια εργασία της ομάδας c πρόκειται να εκτελεστεί ακριβώς μετά από μια εργασία της ομάδας b. Επίσης ένας αρχικός χρόνος εξάρμωσης S_{0b} απαιτείται αν μια εργασία από την ομάδα b εκτελεστεί πρώτη στην μηχανή. Υποθέτουμε ότι για τους χρόνους εξάρμωσης ισχύει η τριγωνική ανισότητα δηλαδή $S_{ac} \le S_{ab} + S_{bc}$, για κάθε $0 \le a \le B$ και $1 \le \{b,c\} \le B$. Οι χρόνοι εξάρμωσης ονομάζονται ανεξάρτητοι ακολουθίας αν ισχύει $S_{bc} = S_c$ για κάθε 0≤b≤B και 1≤c≤B με b≠c. Αν η μηχανή αρχίσει την επεξεργασία της εργασίας i δεν μπορεί να συνεχίσει με άλλη εργασία πριν από την ολοκλήρωση της εργασίας i. Τέλος το πρόβλημα περιλαμβάνει ένα κριτήριο επίδοσης ή μια συνάρτηση κόστους με βάση την οποία αξιολογούμε κάθε διάταξη εργασιών. Αντικειμενικός στόχος είναι η εύρεση μιάς διάταξης π^* που να βελτιστοποιεί το κριτήριο επίδοσης (ή την συνάρτηση κόστους) $f(\pi)$.

Το πρόβλημα διατεταγμένων κατηγοριών είναι μια κατηγορία περιπτώσεων I του $N/1/seq-dep/f\,(\pi)$. Κάθε πρόβλημα στην κατηγορία I έχει τουλάχιστον μια βέλτιστη λύση στην οποία η σχετική θέση των εργασιών που ανήκουν στην ίδια ομάδα είναι καθορισμένη και γνωστή. Στην γενική του μορφή το πρόβλημα των διατεταγμένων κατηγοριών μπορεί να λυθεί σε δύο βήματα. Στο πρώτο βήμα βρίσκουμε τις βέλτιστες ακολουθίες εργασιών μέσα σε κάθε ομάδα (πρόβλημα συνήθως τάξης O(nlogn)) και στο δεύτερο βήμα συγχωνεύουμε βέλτιστες υπακολουθίες ομάδων [MoPo-89].

Στα επόμενα κεφάλαια περιγράφουμε το πρόβλημα $MTMS_j$, δίνουμε τις ήδη γνωστές προτάσεις, και παρουσιαζουμε τις νέες προτάσεις αντιστοιχα. Ακολουθεί η διατύπωση του αλγορίθμου, περιγραφή του περιβάλλοντος εργασίας και του τρόπου υλοποίησης του αλγορίθμου. Αμέσως μετά ορίζουμε τον τρόπο παραγωγής δοκιμαστικών προβλημάτων, δίνουμε αποτελέσματα για τις επιδόσεις του αλγορίθμου και τελικά τα συμπεράσματα μας και τις άμεσες εφαρμογές του σε συγγενή προβλήματα χρονικού προγραμματισμού.

1.3. Το πρόβλημα $MTMS_j$

Το πρόβλημα $n/1/seq-dep/f\,(\pi)$ παρουσιάζει ενδιαφέρον για συναρτήσεις κόστους μέγιστης βραδύτητας, μέγιστης καθυστέρησης, σταθμισμένου αθροίσματος των χρόνων περάτωσης των εργασιών, ελαχίστου αριθμού καθυστερημένων εργασιών και άλλες.

Για τυχόν πρόγραμμα π συμβολίζουμε με C_{ib} τον χρόνο περάτωσης της i-οστής εργασίας της ομάδας b. Επίσης με d_{ib} συμβολίζουμε την προθεσμία της i-οστής εργασίας της ομάδας b. Κάθε εργασία έχει μια συνάρτηση κόστους $f_{ib}(\cdot)$. Ενα αθροιστικό κριτήριο κόστους αναφέρεται σε εύρεση διάταξης που να ελαχιστοποιεί την ποσότητα $\sum_{ib} f_{ib}(C_{ib})$. Ενα κριτήριο κόστους μέγιστης τιμής αναφέρεται σε εύρεση διάταξης που να ελαχιστοποιεί την ποσότητα $\max_{ib} f_{ib}(C_{ib})$.

Στο πρόβλημα $MTMS_j$ έχουμε μια μηχανή και συνάρτηση κόστους την $\max_{ib} T_{ib}(C_{ib})$, με $T_{ib}(C_{ib}) = max(0, C_{ib} - d_{ib})$ και χρόνους εξάρμωσης ανεξάρτητους ακολουθίας. Ο αλγόριθμος που προτείνουμε μπορεί να τροποποιηθεί κατάλληλα ώστε να λύνει το πρόβλημα ακόμα και με γενικά S_{ij} για τα οποία ισχύει η τριγωνική ανισότητα. Επίσης μπορεί να χρησιμοποιηθεί ως υπορουτίνα για την επίλυση συγγενών προβλημάτων σαν και αυτά που αναφέρονται στο κεφάλαιο

8.

Στην συνέχεια της εργασίας με $MTMS_{ij}$, $MTMS_j$, $MTMS$ θα συμβολίζουμε τα προβλήματα μιας μηχανής, με συνάρτηση κόστους την $\max_{ib} T_{ib}(C_{ib})$ και χρόνους εξάρμωσης χωρίς περιορισμούς, ανεξάρτητους ακολουθίας και ίσους μεταξύ τους αντίστοιχα. Οι λόγοι για τους οποίους η συνάρτηση $\max_{ib} T_{ib}(C_{ib})$ είναι από τις βασικές συναρτήσεις κόστους στα προβλήματα χρονικού προγραμματιμού αναπτύσσονται στις επόμενες παραγράφους.

Εστω ότι έχουμε ένα σύστημα εξυπηρέτησης όπου πελάτες υποβάλλουν εργασίες προς εκτέλεση. Βασικό στοιχείο πού συνοδεύει κάθε εργασία i είναι ο χρόνος που απαιτείται για την εκτέλεσή της. Πολλές φορές είναι απαραίτητο η εργασία i να έχει τελειώσει πριν από κάποια συγκεκριμένη χρονική στιγμή. Γι' αυτό και εργασίες με τέτοιες απαιτήσεις συνοδεύονται και από ένα επιπλέον στοιχείο, την προθεσμία παράδοσης τους. Φυσικά κανένας δεν εμποδίζει το σύστημα εξυπηρέτησης να παραβεί τις προθεσμίες των εργασιών. Γι' αυτό υπάρχει ο όρος της ποινής για τις καθυστερημένες εργασίες. Αν κάποια εργασία i ξεπεράσει την προθεσμία της κατά l μονάδες χρόνου το σύστημα εξυπηρέτησης είναι υποχρεωμένο να καταβάλει στον πελάτη πρόστιμο ίσο με $w_i l$. Αν υποθέσουμε ότι $w_i = 1$ για κάθε εργασία i, τότε για να ελαχιστοποιήσουμε την μέγιστη ποινή που το σύστημα πρέπει να καταβάλει θα πρέπει να ελαχιστοποιήσουμε την μέγιστη καθυστέρηση που θα εμφανιστεί στο σύστημα. Βέβαια για να έχουμε ελάχιστη συνολική ποινή θα πρέπει να ελαχιστοποιήσουμε το άθροισμα των καθυστερήσεων. Σίγουρα η τελευταία βελτιστοποίηση είναι και η πιο αποδοτική ως προς το συνολικά καταβαλλόμενο κόστος. Παρ' όλα αύτα η ελαχιστοποιήση της συνολικής ποινής δεν κατανέμει δίκαια τις ποινές και συνεπως δρα ανταγωνιστικά ως προς την αξιοπιστία του συστήματος εξυπηρέτησης. Στην πράξη δημιουργούνται κλάσεις εργασιών που έχουν άνιση μεταχείρηση.

Αν θέσουμε ως κριτήριο δίκαιης μεταχείρησης των πελατών την λογική άποψη του να βρίσκονται όλες οι τιμές των καθυστερήσεων όσο το δυνατόν πιο κοντά μεταξύ τους (στην ιδανική περίπτωση να είναι όλες ίσες) τότε το κριτήριο της ελαχιστοποίησης της μέγιστης καθυστέρησης δίνει βέλτιστη λύση όσο αφορά τη δίκαιη μεταχείριση των πελατών (μια και πάντα υπάρχει τουλάχιστον μια εργασία που δεν καθυστερεί αφού $p_i \le d_i$ για κάθε εργασία i). Από την άλλη πλευρά η έγκαιρη παράδοση των προς εκτέλεση εργασιών αυξάνει την ποιότητα των παρεχόμενων υπηρεσιών και αναβαθμίζει την αξιοπιστία του συστήματος.

Επιπλέον σε ορισμένες περιπτώσεις η παραβίαση προθεσμίας συνεπάγεται ακύρωση της αντίστοιχης εργασίας. Το παρακάτω παράδειγμα προέρχεται από τον χώρο της βιομηχανικής παραγωγής. Τα προιόντα έχουν προθεσμίες που αντιστοιχούν σε ημερομηνίες φόρτωσης τους σε πλοία για την εξαγωγή τους. Αν κάποια προιόντα δεν είναι διαθέσιμα εντός των προθεσμιών η αποστολή τους ματαιώνεται και πιθανόν να είναι εντελώς άχρηστα για τους πελάτες ή στην καλύτερη περίπτωση να έχουμε επιπλέον κόστος αποθήκευσης. Συνεπώς η εύρεση εφικτής διάταξης (κάθε εργασία να πιάνει την προθεσμία της) είναι κρίσιμη και ουσιαστική προυπόθεση σε πολλές εφαρμογές. Προφανές είναι πως υπάρχει εφικτή διάταξη αν και μόνο αν $T_{\max}$=0. Συνεπώς η ελαχιστοποίηση της μέγιστης καθυστέρησης είναι από τα βασικά κριτήρια για την αξιολόγηση ενός προγράμματος. Επιγραμματικά μπορούμε να συνοψίσουμε ότι:

(α) ικανή και αναγκαία συνθήκη ύπαρξης εφικτής λύσης είναι $T_{\max}$=0,

(β) περιστασιακά οδηγεί σε ελαχιστοποίηση και της μέσης καθυστέρησης και

(γ) αποτελεί κριτήριο δικαιοσύνης μεταξύ των πελατών.

1.4. Εφαρμογές των προβλημάτων χρονικού προγραμματισμού στον προγραμματισμό παραγωγής

Η οργάνωση της παραγωγής ενός εργοστασίου είναι μια σύνθετη, ιεραρχικά κατανεμημένη διαδικασία. Για την αντιμετώπιση της αβεβαιότητας των αναμενόμενων γεγονότων αλλά και της πλοκής των αποφάσεων σε σχέση με την έκταση του χρονικού ορίζοντα η όλη παραγωγική διαδικασία έχει χωριστεί σε τρία επίπεδα. Στον στρατηγικό σχεδιασμό, στον μεσοπρόθεσμο προγραμματισμό και στον βραχυπρόθεσμο προγραμματισμό της παραγωγής. Ο ακριβής χρονικός ορίζοντας κάθε επιπέδου εξαρτάται άμεσα από τον τύπο της βιομηχανίας, τον τρόπο λειτουργίας της καθώς και από την οργανωτική της δομή. Αλλαγές και παρεκκλίσεις της παραγωγικής διαδικασίας από την ομαλή λειτουργία, επιφέρουν άμεσες επιδράσεις και σε άλλα τμήματα του εργοστασίου (Τμήμα Πωλήσεων, Αποθήκη, Προσωπικού κ.ά.).

Συνεπώς η ομαλή λειτουργία της παραγωγικής διαδικασίας είναι κρίσιμη για την ομαλή λειτουργία ολόκληρου του βιομηχανικού περιβάλλοντος. Δύο από τα βασικότερα χαρακτηριστικά της παραγωγής είναι :

(α) η ιεραρχική δομή της (κάθε επίπιδο με τις αποφάσεις του περιορίζει και οδηγεί τα χαμηλότερα) και (β) ο μηχανισμός ανάδρασης που περιέχει (αδυναμία άρτιας λειτουργίας των χαμηλών επιπέδων διαβιβάζεται στα υψηλότερα για την αναθεώρηση των στόχων και των ειλημμένων αποφάσεων).

Από την άλλη πλευρά δύο είναι τα κατ' εξοχήν κρίσιμα σημεία της παραγωγικής διαδικασίας. Το επίπεδο του μεσοπρόθεσμου και του βραχυπρόθεσμου προγραμματισμού. Τα σταδια του στρατηγικού και μακροπρόθεσμου προγραμματισμού αντιμετωπίζονται κυρίως με μαθηματικά μοντέλα και μεθόδους επιχειρησιακής έρευνας, ενώ στον βραχυπρόθεσμο προγραμματισμό έχουμε τις απτές εντολές παραγωγής. Στον βραχυπρόθεσμο προγραμματισμό έχουμε την εμφάνιση μη αναμενόμενων γεγονότων (παραγγελίες άμεσης προτεραιότητας, βλάβη μηχανών, απεργίες, διακοπές ρεύματος κ.α.) που θέτουν σε λειτουργία το μηχανισμό ανάδρασης. Ενας βέλτιστος προγραμματισμός σε αυτό το επίπεδο θα απορροφούσε το μέγιστο δυνατό ποσό της ανάδρασης και θα συνέβαλλε καθοριστικά στην ελάττωση των προβλημάτων από μη αναμενόμενα γεγονότα. Συνεπώς η ύπαρξη βελτίστων βραχυπρόθεσμων προγραμμάτων είναι και επιθυμητή και αναγκαία.

Η παρούσα εργασία εντάσσεται σε μια προσπάθεια προγραμματισμού της παραγωγικής διαδικασίας χρησιμοποιώντας ένα πληροφοριακό σύστημα προγραμματισμού και ελέγχου παραγωγής με διαλογικά στοιχεία. Το σύστημα ονομάζεται ΗΦΑΙΣΤΟΣ και αναπτύσσεται από το Ινστιτούτο Πληροφορικής από την ομάδα Συστημάτων Στήριξης Αποφάσεων.

Το ΗΦΑΙΣΤΟΣ είναι κατάλληλο κυρίως για βιομηχανίες επεξεργασίας πρώτων υλών. Στην κατηγορία των βιομηχανιών αυτών ανήκουν οι βιομηχανίες ξύλου, φελλού, πλαστικών κ.α. Το ΗΦΑΙΣΤΟΣ αποτελείται από τα κάτωθι έξι υποσυστήματα:

- υποσύστημα βραχυπρόθεσμου χρονικού προγραμματισμού,
- υποσύστημα λεπτομεριακού προγραμματισμού παραγωγής,
- υποσύστημα υλοποίησης προγραμματισμού,
- υποσύστημα αποθήκης,
- υποσύστημα πωλήσεων και
- υποσύστημα διαχείρισης αρχείων.

Η καρδιά του συστήματος είναι τα υποσυστήματα που αναφέρονται στην παραγωγική διαδικασία. Η συγκεκριμένη εργασία εντάσσεται στο υποσύστημα λεπτομερειακού προγραμματισμού παραγωγής.

Το υποσύστημα λεπτομερειακού προγραμματισμού δίνει στον χρήστη την δυνατότητα να τυποποιεί και να διαμορφώνει ένα ή περισσότερα υποψήφια προγράμματα παραγωγής, να ελέγχει μέσα από συγκεντρωτικά και συνοπτικά στοιχεία την αποτελεσματικότητα τους και να επιλέγει το καταλληλότερο το οποίο και εκδίδει. Επιπλέον αυτή η διαδικασία υποστηρίζεται από ένα σύνολο αλγορίθμων χρονικού προγραμματισμού που του υποδεικνύουν λύσεις ως προς βασικά κριτήρια επίδοσης. Οι προτεινόμενες λύσεις δεν είναι δεσμευτικές προτάσεις. Ο χρήστης μπορεί να τις αποδεχθεί ολικώς, μερικώς, να τις διαμορφώσει ή και να ξανακαλέσει τους αλγορίθμους για να του υποδείξουν νέες λύσεις. Τελικά μέσα από μια διαλογική επεξεργασία ο χρήστης καταλήγει σε ένα τελικό πρόγραμμα παραγωγής το οποίο και εκδίδει. Αξιοσημείωτες είναι οι δυνατότητες επικοινωνίας του ΗΦΑΙΣΤΟΣ και με άλλα συστήματα που τυχόν υπάρχουν στο εργοστασιακό περιβάλλον.

Το πρόβλημα το οποίο καλούμαστε να λύσουμε εμφανίζεται στο επίπεδο του λεπτομεριακού προγραμματισμού της παραγωγικής διαδικασίας σε πολλές βιομηχανίες, κυρίως πρώτων υλών (φαρμάκων, πλαστικών κ.α.). Επίσης εμφανίζεται σε υφαντουργικές βιομηχανίες , σε χαλυβουργεία και γενικότερα σε δραστηριότητες που χρησιμοποιούν μηχανές που παράγουν περισσότερα του ενός προϊόντα με την χρήση διαφόρων εργαλείων. Ορισμένες φορές μπορεί να εμφανιστεί απροσδόκητα και σε άλλους τομείς ακόμα και σε αεροδρόμια όπως αναφέρεται στην βιβλιογραφική ανασκόπηση.

1.5. Ορολογία και συμβολισμοί

Στην παρούσα εργασία εμφανίζονται τέσσερεις οντότητες και πέντε βασικοί τελεστές. Οι τελεστές εφαρμόζονται στις απλές και στις σύνθετες παρτίδες. Σε όποιο σημείο της εργασίας χρησιμοποιείται κάποιος νέος συμβολισμός αυτός ορίζεται ρητά στο συγκεκριμένο σημείο και έχει τοπικό χαρακτήρα.

Οι τέσσερεις βασικές οντότητες είναι :

- εργασίες,
- κρίσιμες εργασίες,
- απλές παρτίδες και

- σύνθετες παρτίδες.

Οι πέντε βασικοί τελεστές είναι :

- p όρισμα: χρόνος επεξεργασίας,
- d όρισμα: προθεσμία,
- O όρισμα: χρονική στιγμή έναρξης επεξεργασίας,
- C όρισμα: χρονική στιγμή περάτωσης επεξεργασίας και
- T όρισμα (t): καθυστέρηση για δεδομένη στιγμή έναρξης επεξεργασίας

Ενας πολύ χρήσιμος τελεστής που επιστρέφει την καθυστέρηση μιας παρτίδας σε κάποιο πρόγραμμα είναι ο T όρισμα και εισάγεται έμμεσα ως απλούστερος συμβολισμός για την ποσότητα T όρισμα (O όρισμα).

Τα σύμβολα που χρησιμοποιούνται και η σημασία τους παρατίθενται στην συνέχεια. Ας σημειώσουμε πως τα σύμβολα μη σταθερών ποσοτήτων αναφέρονται ως προς κάποιο πρόγραμμα π. Ακόμη πολλά από τα σύμβολα μπορούν να εκφραστούν συναρτήσει άλλων, γεγονός που αποφεύγεται για λόγους απλότητας και ευκολίας.

Με τον όρο "διάταξη εργασιών" εννοούμε μια μετάθεση των εργασιών, ενώ με τον όρο "πρόγραμμα" εννοούμε μια μετάθεση των εργασιών τοποθετημένη στον χρόνο. Δηλαδή, επιπλέον γνωρίζουμε τον χρόνο έναρξης της επεξεργασίας της δεδομένης διάταξης.

N : αριθμός εργασιών.
B : αριθμός ομάδων.
$\mathbf{S_{ij}}$: χρόνος εξάρμωσης της μηχανής για την επεξεργασία εργασιών της ομάδας j μετά από εργασίες της ομάδας i.
$\mathbf{N_b}$: αριθμός εργασιών στην ομάδα b, $1 \le b \le B$.
$\mathbf{N^{Crit}}$: αριθμός κρίσιμων εργασιών.
$\mathbf{N_b^{Crit}}$: αριθμός κρίσιμων εργασιών στην ομάδα b, $1 \le b \le B$.

$\mathbf{J_{ib}}$: η i-οστή εργασία της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.
$\mathbf{p_{ib}}$: χρόνος επεξεργασίας της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.

$\mathbf{d_{ib}}$: προθεσμία της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.
$\mathbf{O_{ib}}$: χρόνος έναρξης της επεξεργασίας της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.
$\mathbf{C_{ib}}$: χρόνος περάτωσης της επεξεργασίας της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.
$\mathbf{T_{ib}}$: καθυστέρηση της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$.
$\mathbf{T_{ib}(t)}$: καθυστέρηση της i-οστής εργασίας της ομάδας b, $1 \le b \le B$, $1 \le i \le N_b$ δεδομένου ότι η επεξεργασία της ξεκινά την χρονική στιγμή t.

$\mathbf{J_{ib}^{Crit}}$: κρίσιμη εργασία τάξεως i στην ομάδα b, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$.

$\left[\mathbf{J_{ib}^{Crit}}\right]$: απλή παρτίδα τάξεως i στην ομάδα b, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$.

Με $\mathbf{p}\left[\mathbf{J_{ib}^{Crit}}\right]$, $\mathbf{d}\left[\mathbf{J_{ib}^{Crit}}\right]$, $\mathbf{O}\left[\mathbf{J_{ib}^{Crit}}\right]$, $\mathbf{C}\left[\mathbf{J_{ib}^{Crit}}\right]$ και $\mathbf{T}\left[\mathbf{J_{ib}^{Crit}}\right]$ συμβολίζουμε τον χρόνο επεξεργασίας, την προθεσμία, τον χρόνο έναρξης, τον χρόνο περάτωσης και την καθυστέρηση της i-οστής απλής παρτίδας στην ομάδα b, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$.

Με $\mathbf{T}\left[\mathbf{J_{ib}^{Crit}}\right]\mathbf{(t)}$ συμβολίζουμε την καθυστέρηση της i-οστής απλής παρτίδας στην ομάδα b, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$ δεδομένου ότι η επεξεργασία της ξεκινά την χρονική στιγμή t.

$\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$: σύνθετη παρτίδα που αποτελείται από την απλή παρτίδα τάξεως i έως και την απλή παρτίδα τάξεως j.

Με $\mathbf{p}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$, $\mathbf{d}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$, $\mathbf{O}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$, $\mathbf{C}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$ και $\mathbf{T}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]$, συμβολίζουμε τονχρόνο επεξεργασίας, την προθεσμία, τον χρόνο έναρξης, τον χρόνο περάτωσης και την καθυστέρηση της σύνθετης παρτίδας $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$, $i \le j \le N_b^{Crit}$.

Με $\mathbf{T}\left[\mathbf{J_{ib}^{Crit}},\mathbf{J_{jb}^{Crit}}\right]\mathbf{(t)}$ συμβολίζουμε την καθυστέρηση της σύνθετης παρτίδας $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$, $1 \le b \le B$, $1 \le i \le N_b^{Crit}$, $i \le j \le N_b^{Crit}$ δεδομένου ότι η επεξεργασία της ξεκινά την χρονική στιγμή t.

Η ύπαρξη εκθέτη σε κάποιον από τους τελεστές συνδέει το μέγεθος που ο τελεστής εκφράζει με κάποιο συγκεκριμένο πρόγραμμα. Για παράδειγμα με T_{ib}^2 συμβολίζουμε την καθυστέρηση της i-οστής εργασίας της ομάδας b στο πρόγραμμα π_2.

Ο συμβολισμός $\left[J_{(i)b}^{Crit}\right]$ είναι ταυτόσημος με τον $\left[J_{ib}^{Crit}\right]$. Τον προτιμούμε για ευκρίνεια σε περιπτώσεις όπου στην θέση του δείκτη i υπάρχει ολόκληρη έκφραση.

Για συντομία με $a \leq \{l_1, l_2, \ldots, l_n\} \leq b$ συμβολίζουμε το σύνολο των n περιορισμών $\{\ a \leq l_1 \leq b,\ \ a \leq l_2 \leq b,\ \ldots,\ a \leq l_n \leq b\ \}$.

2. Βιβλιογραφική ανασκόπηση

Οι πρώτες προσεγγίσεις και οι πρώτοι αλγόριθμοι για την αντιμετώπιση του προβήματος διατεταγμένων κατηγοριών εμφανίζονται στην βιβλιογραφία από το 1970 και αργότερα. Οι περισσότερες από αυτές τις προσεγγίσεις μειονεκτούν σε δύο σημεία :

(a) αναφέρονται συνήθως σε εξειδικευμένες περιπτώσεις του προβλήματος και

(b) λύνουν προβλήματα πολύ μικρού μεγέθους

Τα βασικά θεωρήματα που οριοθετούν το πρόβλημα από πλευρά υπολογιστικής πλοκής διατυπώθηκαν από τους Bruno και Downey το 1978 [BrDo-78] και επεκτάθηκαν από τους Monma και Potts το 1988 [MoPo-89].

2.1. Γενική ανασκόπηση

Η πρώτη από τις εργασίες πού έχουν σχέση με το πρόβλημα που εξετάζουμε δημοσιεύθηκε το 1955 από τον J.R. Jackson [Ja-55]. Αναφέρεται στην επίλυση του προβλήματος της ελαχιστοποιήσης της μέγιστης καθυστέρησης n εργασιών σε μια μηχανή ($n/1//T_{\max}$). Η λύση του προβλήματος είναι απλή και δίνεται από τον γνωστό κανόνα EDD (earliest due date). O EDD διατάσσει τις εργασίες κατά μη φθίνουσα τάξη των προθεσμιών του. Συνεπώς το πρόβλημα μπορεί να λυθεί γρήγορα σε χρόνο $O(nlogn)$.

Αμεση γενίκευση της εργασίας του J.R. Jackson αποτελεί η εργασία του E.L. Lawler [La-76] το 1976. Το πρόβλημα το οποίο αντιμετωπίζει είναι και πάλι το $n/1//T_{\max}$ όμως επιπλέον οι διατάξεις των εργασιών πρέπει να ικανοποιούν μια συγκεκριμένη σχέση διαδοχής R. Πιο τυπικά το ακριβές πρόβλημα είναι το $n/1/R/T_{\max}$ όπου R δεδομένη σχέση μερικής διάταξης.

Κάποιες ειδικές περιπτώσεις προβλημάτων που περιλαμβάνουν και χρόνους εξάρμωσης αναπτύσσονται στην συνέχεια.

Το 1980 ο H.N. Psaraftis [Ps-80] παρουσίασε ένα αλγόριθμο δυναμικού προγραμματισμού για την βέλτιστη δίαταξη ενός συνόλου εργασιών J σε μια μηχανή. Οι εργασίες του συνόλου ανήκουν σε B διαφορετικές ομάδες. Οι εργασίες μέσα σε κάθε ομάδα είναι όμοιες και υπάρχουν και περιορισμοί σχετικοί με προτεραιότητες που εκφράζονται ως αποκλίσεις από την αρχική διάταξη (F.C.F.S). Το ζητούμενο είναι να ελαχιστοποιηθεί το συνολικό κόστος επεξεργασίας που ορίζεται ως εξής:

$$\sum_{\text{για κάθε αλλαγη από ομάδα } m \text{ σε ομάδα } n} f(m,n,k_1,k_2,\dots,k_B)$$

όπου $k_i, i=1,..,B$ είναι το πλήθος των αδιάτακτων εργασιών στην ομάδα i. Ο αλγόριθμος που προτείνει εφαρμόζεται στον έλεγχο του αεροδιαδρόμου ενός αεροδρομίου. Οι εργασίες προς εκτέλεση είναι οι προσγειώσεις αεροπλάνων ενώ η μηχανή είναι ο αεροδιάδρομος. Τα αεροπλάνα κατατάσσονται σε κατηγορίες ανάλογα με τον τύπο τους. Επιδόσεις του αλγορίθμου δεν αναφέρονται, όμως για την συγκεκριμένη εφαρμογή θα είναι ικανοποιητικές μια και ο αριθμός των κατηγοριών είναι συνήθως από τρείς εώς πέντε και οι προς εκτέλεση εργασίες δεν ξεπερνούν τις 15.

Οι J.W. Barnes και L.K. Vanston [BaVa-81] δημοσίευσαν το 1981 μια εργασία που απέβλεπε στην βελτιστη διάταξη J εργασιών σε μια μηχανή. Η εργασία συμπεριελάμβανε χρόνους εξάρμωσης S_{ij} και στόχευε στην ελαχιστοποίηση της ποσότητας

$$\sum_{i\in J} W_i O_i + \sum_{i=1}^{N} S_{ii+1} \quad ,$$

όπου W_i και $O_i, i\in J$ το βάρος και ο χρόνος έναρξης της i-οστής εργασίας. Ενας υβριδικός αλγόριθμος διαμερισμού και φραγής προτείνεται για το πρόβλημα. Ο αλγόριθμος εφαρμόζεται σε προβλήματα 10, 15 και 20 εργασιών και δίνει μέσους χρόνους εκτέλεσης 1.77, 9.34 και 32.92 δευτερόλεπτα αντίστοιχα.

Οι A.G. Lockett και A.P. Muhlemann [LoMu-71] το 1971 παρουσίασαν ένα αλγόριθμο διαμερισμού και φραγής για την επίλυση του παρακάτω προβλήματος χρονικού προγραμματισμού.

Εχουμε να προγραμματίσουμε N εργασίες σε μια μηχανή. Κάθε εργασία απαιτεί στην μηχανή την παρουσία ενός συνόλου εργαλείων. Αν τα εργαλεία δεν υπάρχουν τότε ένας χρόνος S απαιτείται για την μεταφορά τους στην μηχανή. Το ζητούμενο είναι να ελαχιστοποιηθεί ο αριθμός των μεταφορών εργαλείων ή ισοδύναμα να ελαχιστοποιηθεί το $C_{\max}$.

Ο αλγόριθμος είναι εφαρμόσιμος σε προβλήματα πολύ μικρού μεγέθους. Δύο είναι οι βασικοί λόγοι γι’ αυτό. Καταρχήν το μικρό πλήθος δυνατών αλλαγων κατεργασίας και δεύτερο η μη ύπαρξη ικανοποιητικού κάτω ορίου. Γι’ αυτό και τελικά παρουσιάζονται αποτελεσματα με βάση ευρηματικές μεθόδους και μέχρι 35 εργασίες.

Η εργασία των E. Uskup και S.B. Smith [UsSm-75] το 1975 δίνει αλγόριθμο για την επίλυση του προβλήματος $N/2/seq-dep/C_{\max} \mid T_{\max}=0$. Δηλαδή βρίσκει εφικτή διάταξη ($C_i < d_i$) τέτοια ώστε να ελαχιστοποιεί τον μέγιστο χρόνο περάτωσης δεδομένου ότι υπάρχουν και χρόνοι εξάρμωσης S_{ij}.

Ο αλγόριθμος είναι βασικά απαριθμητικός και κάνει χρήση κριτηρίων απόρριψης και μηχανισμού φραγμάτων. Για προβλήματα με λιγότερες από 10 εργασίες απαιτεί χρόνο εκτέλεσης 10-20 δευτερόλεπτα. Προβλήματα μέχρι 20 εργασιών απαιτούν χρόνο εκτέλεσης 45-180 δευτερόλεπτα. Επίσης για δύο προβλήματα 30 εργασιών που δοκιμάστηκαν, το μεν πρώτο λύθηκε σε κάτι λιγότερο από 240 δευτερόλεπτα, ενώ το δεύτερο πήρε περίπου 360 δευτερόλεπτα. Τα δεδομένα παρήχθησαν τυχαία και πρέπει να τονίσουμε πως η δυσκολία των προβλημάτων σχετιζόταν άμεσα με την κατανομή των προθεσμιών καθώς και των χρόνων εξάρμωσης. Το ενδιαφέρον συμπέρασμα που συμφωνεί και με το δικό μας είναι πως αν οι χρόνοι εξαρμωσης είναι πολύ κοντά μεταξύ τους ή αν υπάρχουν παρόμοια τμήματα στον πίνακα χρόνων εξάρμωσης (υπάρχουν λίγες διαφορετικές τιμές για τους χρόνους εξάρμωσης) τότε το πρόβλημα είναι πιο δυσκολο.

Μια από τις πρώτες εργασίες στα προβλήματα χρονικού προγραμματισμού με χρόνους εξάρμωσης είναι αυτή του C.R. Glassey [Gl-67]. Το ζητούμενο είναι να βρούμε μια διάταξη εργασιών όπου κάθε εργασία πιάνει την προθεσμία της και ελαχιστοποιεί τον αριθμό των αλλαγών κατεργασίας. Ο χρόνος εξάρμωσης είναι ίσος με μηδέν ενώ το κόστος εξάρμωσης μπορεί να θεώρηθει ίσο με ένα. Ο αλγόριθμος που προτείνεται είναι μια παραλλαγή του μετασχηματισμού του προβλήματος σε πρόβλημα δικτύου με βάση μια πρόταση του Dantzig. Πειραματικά αποτελέσματα δίνονται για μικρά προβλήματα μέχρι 10 εργασιών.

Ολες οι προηγούμενες εργασίες άλλες περισσότερο και άλλες λιγότερο έχουν άμεσο ενδιαφέρον σε σχέση με συγκεκριμένη εφαρμογή. Οι δύο εργασίες που ακολουθούν έχουν περισσότερο θεωρητικό ενδιαφέρον και τα αποτελέσματα τους ισχύουν σε πολύ γενικότερες περιπτώσεις.

2.2. Η εργασία των Bruno και Downey

Από τις πρώτες και πλέον σημαντικές εργασίες στα προβλήματα χρονικού προγραμματισμού με χρόνους εξάρμωσης είναι αυτή των Bruno και Downey το 1978. Ορισαν τα προβλήματα εύρεσης εφικτού προγράμματος και ελαχίστου κόστους και ανάλυσαν την υπολογιστική τους πλοκή. Επίσης για εξειδικευμένες περιπτώσεις αυτών των προβλημάτων παρουσίασαν αλγορίθμους πολυωνυμικούς και ψευδοπολυωνυμικούς. Ακολουθούν μερικά στοιχεία από την εργασία τους που σχετίζονται άμεσα με την δική μας εργασία.

Οι Bruno και Downey όρισαν το πρόβλημα εύρεσης εφικτού προγράμματος ως εξής:

> **Για το πρόβλημα χρονικού προγραμματισμού που περιλαμβάνει κατηγορίες και χρόνους εξάρμωσης υπάρχει ή όχι διάταξη στην οποία κάθε εργασία να τελειώνει πριν από την προθεσμία της;**

Είναι φανερό πως η απάντηση στο πρόβλημα είναι θετική αν και μόνο αν το πρόβλημα $MTMS_{ij}$ έχει βέλτιστη τιμή της αντικειμενικής συνάρτησης ίση με μηδέν. Για το προηγούμενο πρόβλημα εύρεσης εφικτού προγράμματος απέδειξαν πως είναι NP-complete ακόμα και για μοναδιαίους χρόνους εξάρμωσης, με τρεις και μόνο εργασίες σε κάθε ομάδα και με τρεις και μόνο διαφορετικές προθεσμίες. Η απόδειξη έγινε με αναγωγή του προβλήματος του εκδρομικού σάκου (Knapsack problem) στο πρόβλημα εύρεσης εφικτού προγράμματος και συγκεκριμένα, στην κατηγορία των προβλημάτων με ίσους χρόνους εξάρμωσης, με τρεις εργασίες ανά ομάδα και τρείς μόνο διαφορετικές προθεσμίες.

Το πρόβλημα ελαχίστου κόστους ορίστηκε από τους Bruno και Downey ως εξής:

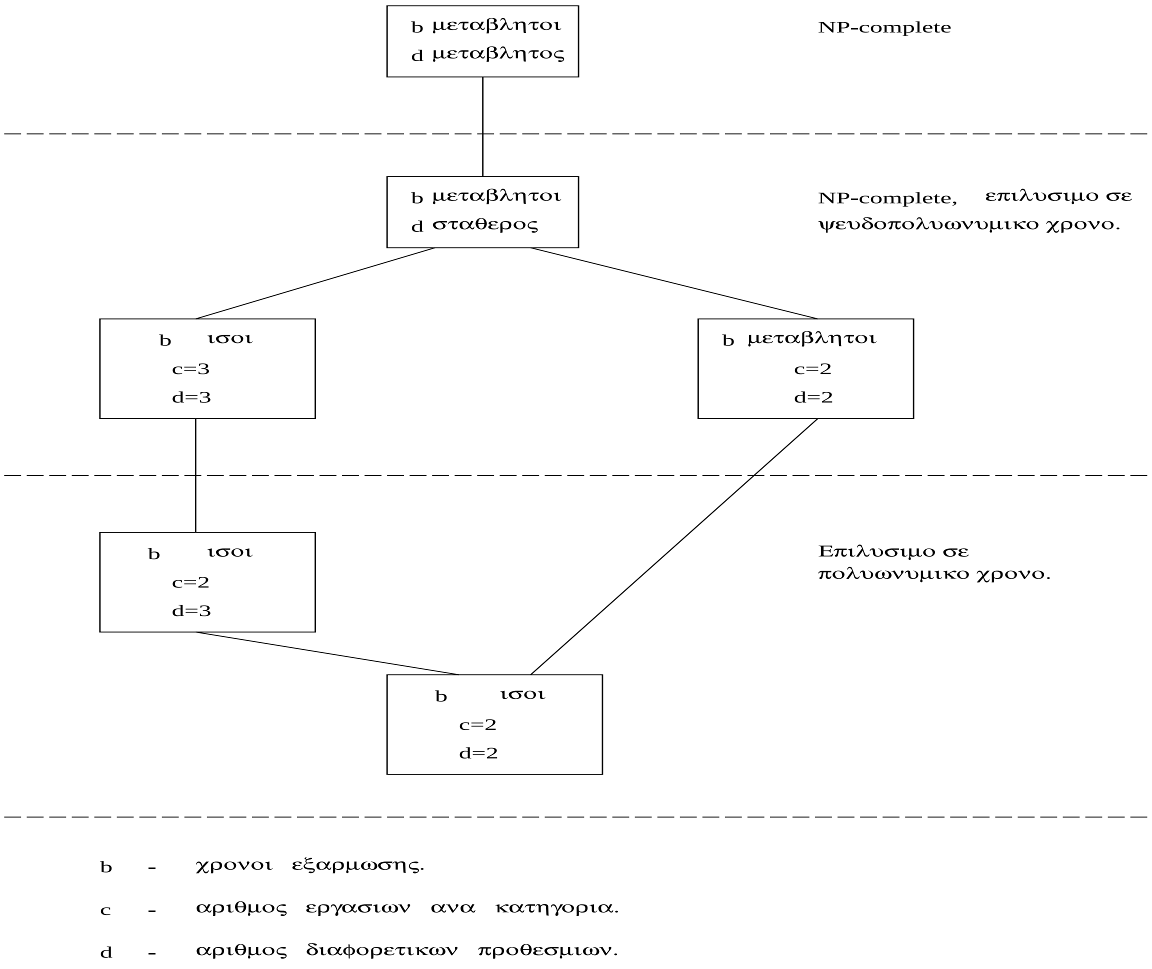
> **Για το πρόβλημα χρονικού προγραμματισμού που περιλαμβάνει κατηγορίες και χρόνους εξάρμωσης υπάρχει ή όχι εφικτή διάταξη που να έχει κόστος μικρότερο ή ίσο ενός μη αρνητικού ακεραίου K;**

Περίπτωση του προβλήματος ελαχίστου κόστους αναφέρεται στο κεφ. 8 και η λύση του βασίζεται στην ανάλυση του προβλήματος $MTMS_{ij}$.

Συνοπτικά το σχήμα 2.1.1. παριστάνει τα πιο σπουδαία αποτελέσματα της εργασίας των Bruno και Downey για το πρόβλημα εύρεσης εφικτού προγράμματος.

2.3. Η εργασία των Monma και Potts

Τα επόμενα σημαντικά θεωρήματα οφείλονται στους Monma και Potts και αποδεικνύουν πως για το πρόβλημα $MLMS_{ij}$ (το πρόβλημα των διατεταγμένων κατηγοριών με αντικειμενική συνάρτηση την $\min_{\pi} \max_{ib} L_{ib}(C_{ib})$ και $L_{ib}(C_{ib}) = C_{ib} - d_{ib}$) καθώς και το MTWC$S_{ij}$ (το πρόβλημα των διατεταγμένων



Σχήμα 2.2.1 : Αποτελέσματα για το πρόβλημα εύρεσης εφικτού προγράμματος.

κατηγοριών με αντικειμενική συνάρτηση την $\min_{\pi} \sum_{ib} w_{ib} C_{ib}$) υπάρχουν βέλτιστες διατάξεις στις οποίες οι εργασίες μέσα σε κάθε ομάδα είναι διατεταγμένες κατά EDD (earliest due date) και SWPT (shortest weighted processing time) αντίστοιχα. Επίσης για το πρόβλημα MWNLS_{ij} (πρόβλημα με αντικειμενική συνάρτηση τον σταθμισμένο αριθμό καθυστερημένων εργασιών) αποδεικνύουν πως υπάρχει βέλτιστη διάταξη στην οποία οι μη καθυστερημένες εργασίες κάθε ομάδας μέσα σε κάθε ομάδα είναι διατεταγμένες κατά EDD. Επίσης παρουσιάζουν και ένα αλγόριθμο δυναμικού προγραμματισμού για την επίλυση του γενικού προβληματος των διατεταγμένων κατηγοριών. Ο αλγόριθμος αυτός έχει θεωρητικό και μόνο ενδιαφέρον εκτός και αν ο αριθμός των ομάδων είναι πολύ μικρος. Αυτό που έχει μεγάλη σημασία είναι πως από την ανάλυση αυτού του αλγορίθμου προκύπτουν συγκεκριμένες εκφράσεις για την υπολογιστική πλοκή των προβληματων $MLMS_{ij}$, MTWCS_{ij} και MWNLS_{ij}. Τα δύο πρώτα μπορούν να λυθούν σε χρόνο O($B^2 N^B \min\{N^V, T\}$), όπου

$$T = \sum_{b=1}^{B}\sum_{i=1}^{N_b} p_{ib} + \sum_{b=1}^{B} N_b \max_{0 \le a \le B}\{S_{ab}\}$$

και V ο αριθμός των διαφορετικών τιμών των χρόνων εξάρμωσης. Το MWNLS_{ij} μπορεί να λυθεί σε χρόνο O($B^2 N^B \min\{W, D, T\}$), όπου

$$W = \sum_{b=1}^{B}\sum_{i=1}^{N_b} w_{ib}$$

$$D = \max_{1 \le b \le B}\{d_{N_b b}\}$$

και T ορισμένο όπως και προηγουμένως.

Οπως φαίνεται εύκολα η υπολογιστική πλοκή του προτεινόμενου αλγορίθμου είναι εκθετική ως προς τον αριθμό των ομάδων. Η υπολογιστική πλοκή για το πρόβλημα $MLMS_j$ μπορεί να γίνει ακόμα καλύτερη αν ο αριθμός των ομάδων B ικανοποιεί την σχέση $B > N/\log_2 N$. Το συμπέρασμα είναι άμεσο από την έκφραση της υπολογιστικής πλοκής για το $MLMS_j$ που δίνεται στο κεφ. 4.1.

3. Ανάλυση του προβλήματος $MTMS_{ij}$

Η όλη προσπάθεια για την εύρεση ενός αλγορίθμου για το $MTMS_{ij}$ βασίζεται στην ιδέα της δημιουργίας παρτίδων. Με τον όρο παρτίδα εννοούμε ακολουθία εργασιών από την ίδια ομάδα, που εκτελούνται συνεχόμενες σε κάποια βέλτιστη διάταξη εργασιών. Οι προτάσεις που ακολουθούν βασικά αποσκοπούν στην αναζήτηση, δημιουργία και χαρακτηρισμό των παρτίδων.

3.1. Υπολογιστική πλοκή του $MTMS_{ij}$

Οπως αναφέρεται και στο κεφ. 2.2 η εύρεση μιας εφικτής διάταξης ($\min_{\pi} \max_{ib} T_{ib}(C_{ib}) = 0$) είναι πρόβλημα NP-complete ακόμα και για $S_{ij}=1$, $N_b \le 3$, και $d_{kb} \in \{d^1, d^2, d^3\}$ για κάθε $0 \le i \le B$, $1 \le j \le B$,$1 \le k \le N_b$ και $1 \le b \le B$. Το πρόβλημα αυτό είναι το πρόβλημα απόφασης για το $MTMS$. Συνεπώς το $MTMS_{ij}$ είναι NP-hard.

Θεώρημα 3.1.1 Τα προβλήματα $MTMS_{ij}$, $MTMS_j$, $MTMS$ είναι NP-hard.

3.2. Ισοδυναμία των $MTMS_{ij}$ και $MLMS_{ij}$

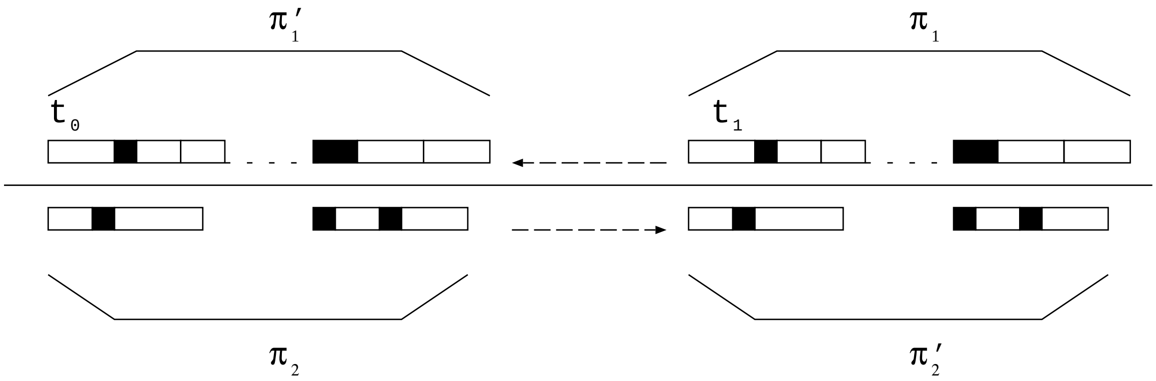
Στο πρόβλημα $MTMS_{ij}$ όπως ορίστηκε στο κεφ. 1.1 θεωρούμε ως στιγμή έναρξης εκτέλεσης των προγραμμάτων την $t=0$. Αν χαλαρώσουμε αυτόν τον περιορισμό έχουμε το πρόβλημα $MTMS_{ij}(t)$ όπου ως στιγμή έναρξης εκτέλεσης των προγραμμάτων θεωρούμε την t. Για το $MTMS_{ij}(t)$ ισχύει το παρακάτω θεώρημα.

Θεώρημα 3.2.1 Αν $t_1 \ge t_0$ τότε το σύνολο λύσεων του προβλήματος $MTMS_{ij}(t_1)$ είναι υποσύνολο του συνόλου λύσεων του προβλήματος $MTMS_{ij}(t_0)$.

Απόδειξη: Χωρίς βλάβη της γενικότητας μπορούμε να υποθέσουμε ότι $t_1 > t_0$ καθώς για $t_1 = t_0$ το θεώρημα προφανώς ισχύει.

Θεωροούμε μια βέλτιστη διάταξη π_1 του $MTMS_{ij}(t_1)$ και έστω T_{1max} η τιμή της αντικειμενικής συνάρτησης.

Αν ολισθήσουμε το πρόγραμμα π_1, $t_1 - t_0$ μονάδες χρόνου αριστερά τότε το νέο πρόγραμμα π_1' θα είναι μια πιθανή λύση για το πρόβλημα $MTMS_{ij}(t_0)$ και με τιμή αντικειμενικής συνάρτησης T'_{1max} (υπενθυμίζουμε οτι ως πρόγραμμα θεωρούμε μια διάταξη μαζί με την χρονική στιγμή που θα ξεκινήσει η εκτέλεση).



Σχήμα 3.2.1 : Τα προγράμματα π_1, π_1', π_2 και π_2'.

Για το T_{1max} έχουμε:

α) Αν $T_{1max} = 0$ τότε προφανώς και $T'_{1max} = 0$ και συνεπώς η διάταξη π_1 είναι βέλτιστη και για το $MTMS_{ij}(t_0)$.

β) Αν $T_{1max} > 0$ τότε $T'_{1max} = \max(0, T_{1max} + t_0 - t_1)$.
Για το T'_{1max} έχουμε:

β1) Αν $T'_{1max} = 0$ τότε προφανώς η διάταξη π_1 είναι βέλτιστη και για το $MTMS_{ij}(t_0)$.

β2) Αν $T'_{1max} > 0$ οπότε και $T'_{1max} = T_{1max} + t_0 - t_1$ και η π_1' δεν είναι βέλτιστη διάταξη στο $MTMS_{ij}(t_0)$ τότε υπάρχει διάταξη π_2 με τιμή της αντικειμενικής συνάρτησης T_{2max} τέτοια ώστε

$$T_{2max} < T'_{1max}$$

Θεωρούμε την διάταξη π_2' που προκύπτει με μεταφορά της π_2, $t_1 - t_0$

μονάδες χρόνου δεξιότερα. Εστω T'_{2max} η τιμή της αντικειμενικής συνάρτησης για το πρόγραμμα π'_2.

Για την T_{2max} έχουμε :

β2.1)Αν $T_{2max}=0$ τότε

$$T'_{2max} \le T_{2max}+t_1-t_0 < T'_{1max}+t_1-t_0 = T_{1max}+t_0-t_1+t_1-t_0 = T_{1max}$$

β2.2)Αν $T_{2max}>0$ τότε

$$T'_{2max} = T_{2max}+t_1-t_0 < T'_{1max}+t_1-t_0 = T_{1max}+t_0-t_1+t_1-t_0 = T_{1max}$$

δηλαδή $T'_{2max}<T_{1max}$, άτοπο αφού η π_1 είναι βέλτιστο πρόγραμμα για το $MTMS_{ij}(t_1)$.

Συνεπώς κάθε βέλτιστη λύση του προβλήματος $MTMS_{ij}(t_1)$ είναι και βέλτιστη λύση του προβλήματος $MTMS_{ij}(t_0)$ και συνεπώς το θεώρημα 3.2.1 ισχύει.□

Αμεση συνέπεια του θεωρήματος 3.2.1 είναι και τα ακόλουθα δύο πορίσματα.

Πόρισμα 3.2.1 Κάθε βέλτιστη λύση του προβλήματος $MTMS_j(t_1)$ είναι και βέλτιστη λύση του προβλήματος $MTMS_j(t_0)$, όπου $t_1 \ge t_0$.

Πόρισμα 3.2.2 Κάθε βέλτιστη λύση του προβλήματος $MTMS(t_1)$ είναι και βέλτιστη λύση του προβλήματος $MTMS(t_0)$, όπου $t_1 \ge t_0$.

Το θεώρημα 3.2.1 έχει πρακτική αξία γιατί μας επιτρέπει να απλοποιήσουμε την αντικειμενική συνάρτηση του προβλήματος $MTMS_{ij}(0)$ (δηλ. του προβλήματος $MTMS_{ij}$). Επειδή κάθε βέλτιστη λύση του $MTMS_{ij}(t)$ για $t \ge 0$, είναι και βέλτιστη λύση του $MTMS_{ij}(0)$ μπόρουμε να λύσουμε το $MTMS_{ij}(d_{\max})$ αντί του $MTMS_{ij}(0)$, όπου $d_{\max}=\max_{ib}\{d_{ib}\}$ για $1 \le i \le N_b$ και $1 \le b \le B$. Το πρόβλημα $MTMS_{ij}(d_{\max})$ έχει πιο απλή αντικειμενική συνάρτηση, την

$$T_{ib}(C_{ib}) = \max(0, C_{ib}-d_{ib}) = C_{ib}-d_{ib}$$

Βέβαια επειδή λύνουμε το πρόβλημα $MTMS_{ij}(d_{\max})$ πιθανό είναι να μην βρούμε ολόκληρο το σύνολο λύσεων του προβλήματος $MTMS_{ij}(0)$ αλλά μόνο ένα υποσύνολο του όπως το θεώρημα 3.2.1 αποδεικνύει.

Στην ουσία μετασχηματίσαμε το προβλημα $MTMS_{ij}$ σε πρόβλημα $MLMS_{ij}$ για να αποφύγουμε την ύπαρξη του μεγίστου στην αντικειμενική συνάρτηση. Εκτός από πρακτικό ενδιαφέρον ο μετασχηματισμός έχει και θεωρητικό ενδιαφέρον μια και απλοποιεί σημαντικά την παραπέρα θεωρητική ανάλυση του προβήματος. Αυτό γιατί σε κάθε απόδειξη μπορούμε να θεωρούμε πως όλες οι καθυστερήσεις είναι θετικές και συνεπώς δεν υπάρχει λόγος να γίνει ανάλυση περιπτώσεων για θετικές ή μηδενικές καθυστερήσεις.

Αμεση συνέπεια του μετασχηματισμού είναι πως το πρόβλημα $MLMS_{ij}$ είναι επίσης NP-hard (αν δεν ήταν τότε και το $MTMS_{ij}$ δεν θα ήταν NP-hard πράγμα άτοπο).

Αντίστοιχο θεώρημα με το 3.2.1 ισχύει και για το $MLMS_{ij}$. Εδώ το θεώρημα είναι πιο ισχυρό ακριβώς λόγω της έλλειψης του μεγίστου στην αντικειμενική συνάρτηση.

Θεώρημα 3.2.2 Τα σύνολα λύσεων των προβλημάτων $MLMS_{ij}(t_0)$ και $MLMS_{ij}(t_1)$ ταυτίζονται.

Απόδειξη: Εστω π_0 βέλτιστο πρόγραμμα για το πρόβλημα $MLMS_{ij}(t_0)$ και L_{0max} η τιμή της αντικειμενικής συναρτησης. Η διάταξη π_0 στο πρόβλημα $MLMS_{ij}(t_1)$ δίνει τιμή της αντικειμενικής συναρτησης $L'_{0max} = L_{0max} + t_1 - t_0$. Αν η π_0 δεν είναι βέλτιστη διάταξη στο $MLMS_{ij}(t_1)$ τότε υπάρχει διάταξη π_1 με τιμή της αντικειμενικής συναρτησης $L_{1max} < L'_{0max}$ Η διάταξη π_1 στο πρόβλημα $MLMS_{ij}(t_0)$ δίνει τιμή της αντικειμενικής συναρτησης $L'_{1max} = L_{1max} - t_1 + t_0 < L_{0max}$, άτοπο αφού η π_0 είναι βέλτιστη λύση.

Τα ίδια ισχύουν και για κάθε βέλτιστη λύση του $MLMS_{ij}(t_1)$ και συνεπώς τα δύο προβλήματα έχουν το ίδιο σύνολο λύσεων.□

Αντίστοιχα πορίσματα με τα 3.2.2 και 3.2.3 ισχύουν και για το πρόβλημα $MLMS_{ij}$.

Πόρισμα 3.2.4 Τα προβλήματα $MLMS_j(t_0)$ και $MLMS_j(t_1)$ έχουν το ίδιο σύνολο λύσεων.

Πόρισμα 3.2.5 Τα προβλήματα $MLMS(t_0)$ και $MLMS(t_1)$ έχουν το ίδιο σύνολο λύσεων.

3.3. Κατανομή της μέγιστης καθυστέρησης

Στο κεφάλαιο αυτό μελετάμε την μεταβολή της κατανομής της μέγιστης καθυστέρησης ως προς την ολίσθηση στο χρόνο και την παρεμβολή. Δίνουμε δύο βασικά λήμματα που θα χρησιμέυσουν αργότερα στην απόδειξη του θεωρήματος για την δημιουργία παρτίδων και του θεωρήματος για το κριτήριο κλαδέματος. Τα λήμματα έχουν ώς εξής :

Λήμμα 3.3.1 Εστω διάταξη J_1J_2 δύο εργασιών J_1 και J_2 με χρόνους και προθεσμίες p_1,d_1 και p_2,d_2 αντίστοιχα. Επίσης, $d_1<d_2$ και για μια δεδομένη χρονική στιγμή $t_0>d_2$ έχουμε :

$$T_1(t_0)\geq T_2(t_0+p_1) \qquad (3.3.1)$$

όπου $T_i(t)$ είναι η καθυστέρηση της εργασίας i δεδομένου ότι η επεξεργασία της ξεκινά την χρονική στιγμή t. Δηλαδή, $T_i(t)=\max(0,t+p_i-d_i)$.
Για τις συναρτήσεις $T_i(t)$, $i=1,2$ ισχύει

$$T_1(t)\geq T_2(t+p_1) \qquad \text{για κάθε } t\in R$$

Απόδειξη: Απο την υπόθεση (3.3.1) έχουμε $T_1(t_0)>T_2(t_0+p_1)\Rightarrow$

$$\max(0,t_0+p_1-d_1) > \max(0,t_0+p_1+p_2-d_2) \qquad (3.3.2)$$

και αφού $t_0 > d_2 > d_1$ η σχέση (3.3.2) γράφεται :

$$t_0+p_1-d_1>t_0+p_1+p_2-d_2 \Rightarrow d_2>p_2+d_1 \qquad (3.3.3)$$

Η προς απόδειξη σχέση γράφεται :

$$\max(0,t+p_1-d_1)\geq\max(0,t+p_1+p_2-d_2)$$

οπότε άν

α) $\max(0,t+p_1-d_1)=0$ έχουμε $0\geq t+p_1-d_1$. Προσθέτοντας κατά μέλη με τήν (3.3.3) έχουμε

$$0>t+p_1+p_2-d_2 \Rightarrow 0\geq\max(0,t+p_1+p_2-d_2)\Rightarrow$$

$$max(0,t+p_1-d_1)\geq max(0,t+p_1+p_2-d_2)$$

και συνεπώς το λήμμα ισχύει.

β) $\max(0,t+p_1-d_1) = t+p_1-d_1$ έχουμε $t+p_1-d_1 \geq 0$. Προσθέτοντας και στα δύο μέλη της (3.3.3) την ποσότητα $t+p_1$ έχουμε $t+p_1-d_1 > t+p_1+p_2-d_2$ οπότε και

$$\left.\begin{array}{l} t+p_1-d_1 \geq 0 \\ t+p_1-d_1 > t+p_1+p_2-d_2 \end{array}\right\} \Rightarrow t+p_1-d_1 \geq max(0,t+p_1+p_2-d_2) \Rightarrow$$

$$max(0,t+p_1-d_1) \geq max(0,t+p_1+p_2-d_2)$$

και συνεπώς το λήμμα ισχύει.□

Με απλά λόγια το λήμμα 3.3.1 δηλώνει πως αν για δύο εργασίες J_1 και J_2 με $d_1 < d_2$ γνωρίζουμε την κατανομή† της μέγιστης καθυστέρησης σε ένα πρόγραμμα που ξεκινά την χρονική στιγμή $t > d_2$ τότε γνωρίζουμε την κατανομή της μέγιστης καθυστέρησης σε κάθε χρονική ολίσθηση του υπο συζητηση προγράμματος. Πιο τυπικά μπορούμε να πούμε πως η κατανομή της μέγιστης καθυστέρησης είναι ανεξάρτητη της ολίσθησης στο χρόνο.

Λήμμα 3.3.2 Εστω διάταξη J_1J_2 δύο εργασιών J_1 και J_2 με χρόνους και προθεσμίες p_1,d_1 και p_2,d_2 αντίστοιχα. Επίσης, $d_1 < d_2$ και για μια δεδομένη χρονική στιγμή $t_0 > d_2$ έχουμε :

$$T_1(t_0) \leq T_2(t_0+p_1)+k \tag{3.3.4}$$

Για τις συναρτήσεις $T_i(t)$, $i=1,2$ ισχύει :

$$T_1(t) \leq T_2(t+p_1+l), \qquad \text{για κάθε } t \in R \text{ και } l \geq k \geq 0$$

Απόδειξη: Απο την υπόθεση (3.3.4) έχουμε :

$$T_1(t_0) \leq T_2(t_0+p_1)+k \Rightarrow$$

$$\max(0,t_0+p_1-d_1) \leq \max(0,t_0+p_1+p_2-d_2) + k \Rightarrow$$

$$t_0+p_1-d_1 \leq t_0+p_1+p_2+k-d_2 \Rightarrow$$

$$d_2 \leq p_2+d_1+k \tag{3.3.5}$$

Η σχέση που θέλω να δείξω γράφεται :

†γνωρίζουμε σε ποιά εργασία θα εμφανιστεί η μέγιστη καθυστέρηση

$$max(0,t+p_1-d_1) \le max(0,t+p_1+p_2+l-d_2)$$

οπότε αν

α) $max(0,t+p_1+p_2+l-d_2) = 0$ έχουμε :

$0 \ge t+p_1+p_2+l-d_2 \overset{l \ge k}{\Rightarrow} 0 \ge t+p_1+p_2+k-d_2$ οπότε

$$\left.\begin{array}{l} 0 \ge t+p_1+p_2+k-d_2 \\ p_2+k+d_1 > d_2 \end{array}\right\} \Rightarrow d_1 > t+p_1 \Rightarrow 0 > t+p_1-d_1 \Rightarrow$$

$$0 \ge max(0,t+p_1-d_1) \Rightarrow max(0,t+p_1+p_2+l-d_2) \ge max(0,t+p_1-d_1)$$

και συνεπώς το λήμμα ισχύει.

β) $max(0,t+p_1+p_2+l-d_2) = t+p_1+p_2+l-d_2$ έχουμε :

$$t+p_1+p_2+l-d_2 \ge 0 \tag{3.3.6}$$

επίσης από την (3.3.5) έχουμε :

$$d_2 \le p_2+k+d_1 \Rightarrow t+p_1+d_2 \le t+p_1+p_2+k+d_1 \overset{l \ge k}{\Rightarrow} t+p_1+d_2 \le t+p_1+p_2+l+d_1 \Rightarrow$$

$$t+p_1-d_1 \le t+p_1+p_2+l-d_2 \tag{3.3.7}$$

Από τις σχέσεις (3.3.6) και (3.3.7) έχουμε

$$max(0,t+p_1+p_2+l-d_2) = t+p_1+p_2+l-d_2 \ge max(0,t+p_1-d_1)$$

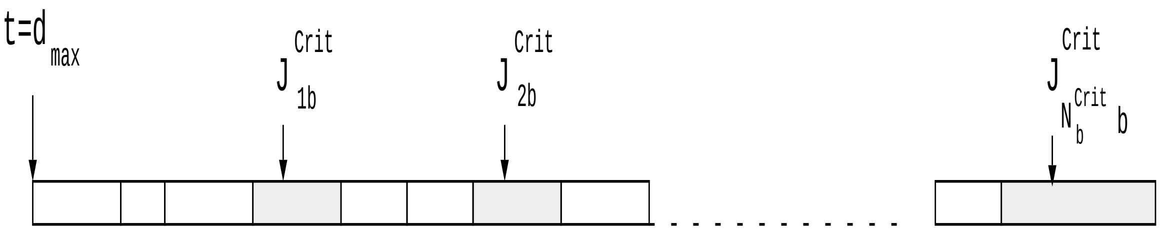
και συνεπώς το λήμμα ισχύει.□

Το λήμμα δηλώνει πως η γνώση της κατανομής της μέγιστης καθυστέρησης για δύο εργασίες J_1, J_2 με $d_1 < d_2$ σε κάποιο πρόγραμμα π που ξεκινά την χρονική στιγμή $t > d_2$ και στο οποίο οι εργασίες J_1και J_2 απέχουν k χρονικές μονάδες συνεπάγεται την γνώση της κατανομής της μεγιστης καθυστέρησης σε κάθε άλλο πρόγραμμα π΄ που ξεκινά την χρονική στιγμή $t' \in R$ και στο οποίο οι εργασίες J_1και J_2 απέχουν τουλάχιστον k χρονικές μονάδες.

3.4. Ορισμός παρτίδων στο $MTMS_{ij}$

Οπως έχουμε ήδη αναφέρει η βασική ιδέα για την επίλυση του $MTMS_{ij}$ είναι η δημιουργία παρτίδων. Υπενθυμίζουμε ότι με τον όρο παρτίδα εννοούμε ακολουθία εργασιών από την ίδια ομάδα, που εκτελούνται συνεχόμενες σε κάποια βέλτιστη διάταξη εργασιών. Το θεώρημα 3.4.1 είναι ένα από τα τρία βασικά θεώρηματα στα οποία βασίζεται ο αλγόριθμος του κεφ. 4.1. Πριν προχωρήσουμε στην διατύπωση και την απόδειξή του θα δώσουμε τον ορισμό του τελεστή ○ (επόμενη εργασία).

Ορισμός 3.4.1 Ο τελεστής ○ που δηλώνει την επόμενη εργασία είναι μια συνάρτηση από το J στο J, με τύπο ○ $J_{ib} = J_{i+1b}$, για κάθε $J_{ib} \in J$, $1 \le b \le B$, $1 \le i < N_b$. Για $i = N_b$ η συνάρτηση δεν ορίζεται.



Σχήμα 3.4.1 : Οι κρίσιμες εργασίες της ομάδας b.

Υποθέτουμε ότι τα S_{ij} ακολουθούν την τριγωνική ανισότητα. Δηλαδή

$$S_{ij} \le S_{ik} + S_{kj}, \qquad \text{για κάθε } 0 \le i \le B \text{ και } 1 \le \{k,j\} \le B$$

Η διαδικασία με την οποία δημιουργούμε τις παρτίδες βασίζεται στον παρακάτω ορισμό ενός συνόλου εργασιών που ονομάζονται κρίσιμες εργασίες. Διατάσσουμε τις εργασίες μέσα σε κάθε ομάδα με βάση τον EDD ξεκινώντας την χρονική στιγμή $t > d_{\max} = \max_{i,b} \{ d_{ib} \}$ όπως φαίνεται sτο σχήμα 3.4.1. Εστω b, $1 \le b \le B$ μια ομάδα και $J_{i_0 b}$ η εργασία της ομάδας b που έχει την μέγιστη καθυστέρηση. Την εργασία $J_{i_0 b}$ την ονομάζουμε κρίσιμη εργασία τάξεως 1 και την συμβολίζουμε με J^{Crit}_{1b}. Αν περισσότερες από μια εργασίες έχουν την ίδια καθυστέρηση, τότε η κρίσιμη εργασία είναι αυτή που έχει τον μεγαλύτερο χρόνο περάτωσης. Η κρίσιμη εργασία τάξεως k, J^{Crit}_{kb} ορίζεται αναδρομικά και είναι η κρίσιμη εργασία τάξεως 1 στο υποσύνολο των εργασιών

$$\{ \bigcirc J^{Crit}_{k-1b}, \bigcirc\bigcirc J^{Crit}_{k-1b}, \ldots, J_{N_b b} \}$$

Φυσικά ο ορισμός κρίσιμων εργασιών τελειώνει όταν για κάποια τάξη k έχουμε

$J_{kb}^{Crit} = J_{N_b b}$. Ο αριθμός των κρίσιμων εργασιών στην ομάδα b συμβολίζεται με N_b^{Crit}.

Η πολύ ενδιαφέρουσα ιδιότητα των κρίσιμων εργασιών είναι ότι ορίζουν παρτίδες όπως αποδεικνύει και το παρακάτω θεώρημα. Η τυχούσα κρίσιμη εργασία $J_{ib}^{Crit},\ i=2,\ldots,N_b^{Crit}$ ορίζει την παρτίδα $\left[J_{ib}^{Crit}\right]$ που περιέχει τις εργασίες ○ J_{i-1b}^{Crit}, ○○ $J_{i-1b}^{Crit},\ldots,J_{ib}^{Crit}$. Η κρίσιμη εργασία J_{1b}^{Crit} ορίζει την παρτίδα $\left[J_{1b}^{Crit}\right]$ που περιέχει τις εργασίες $J_{1b},\ldots,J_{1b}^{Crit}$.

Θεώρημα 3.4.1 Οι ακολουθίες εργασιών πού ορίζονται από τις κρίσιμες εργασίες αποτελούν παρτίδες για το πρόβλημα $MTMS_{ij}$ με χρόνους εξάρμωσης που ικανοποιούν την τριγωνική ανισότητα.

Απόδειξη: Ξεκινάμε με δύο παρατηρήσεις. Λόγω του Θεωρήματος 3.2.1 μπορούμε να θεωρήσουμε ότι ο χρόνος έναρξης κάθε προγράμματος για το πρόβλημα είναι αρκετά μεγάλος ώστε κάθε εργασία να είναι καθυστερημένη. Επίσης από την εργασία των Monma και Potts γνωρίζουμε πως υπάρχει βέλτιστη διάταξη στην οποία οι εργασίες μέσα σε κάθε ομάδα είναι διατεταγμένες κατά EDD (βλέπε κεφ. 2.2).

Εστω π_1 μια βέλτιστη διάταξη εργασίων στην οποία οι εργασίες μέσα σε κάθε ομάδα είναι διατεταγμένες κατά EDD (σχήμα 3.4.2)

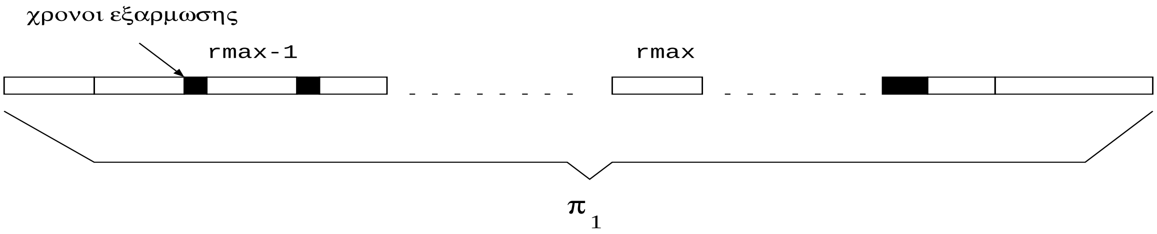
και έστω r τυχούσα παρτίδα ορισμένη όπως προηγουμένως. Εστω i_0 η εργασία της παρτίδας r με τον μεγαλύτερο χρόνο περάτωσης C_{i_0} στο πρόγραμμα π_1. Προφανώς η εργασία i_0 είναι η $rmax$, όπου $rmax$ η εργασία της παρτίδας r που έχει την μεγαλύτερη προθεσμία (σχήμα 3.4.3). Η γειτονιά της εργασίας $rmax-1$ στο πρόγραμμα π_1 θα είναι μια από τις τέσσερεις του σχήματος 3.4.4.

Θεωρούμε το πρόγραμμα π_2 που προέρχεται από το π_1 αν μετακινήσουμε την εργασία $rmax-1$ ακριβώς αριστερά της $rmax$ (σχήμα 3.4.5).

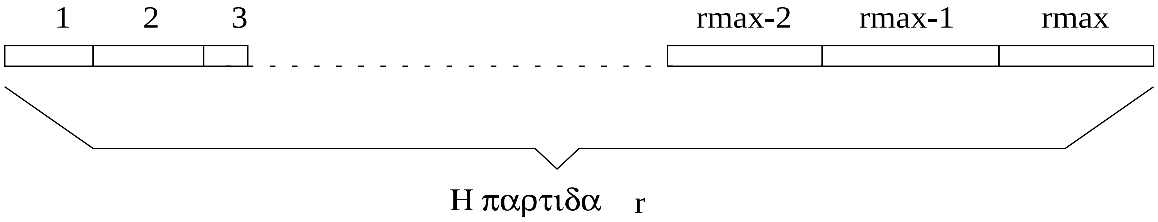
Με T_i^1, και T_i^2 συμβολίζουμε την καθυστέρηση της εργασίας $i \in J$ στα προγράμματα π_1 και π_2 αντίστοιχα. Για το πρόγραμμα π_2 και για κάθε μια από τις τέσσερεις δυνατές μορφές του σχήματος 3.4.4 έχουμε:

α) λόγω της υπόθεσης ότι ισχύει η τριγωνική ανισότητα στους χρόνους εξάρμωσης έχουμε :

$$S_{ik} \le S_{ir} + S_{rk}$$



Σχήμα 3.4.2 : Το πρόγραμμα π_1.



Σχήμα 3.4.3 : Η παρτίδα r.

και συνεπώς

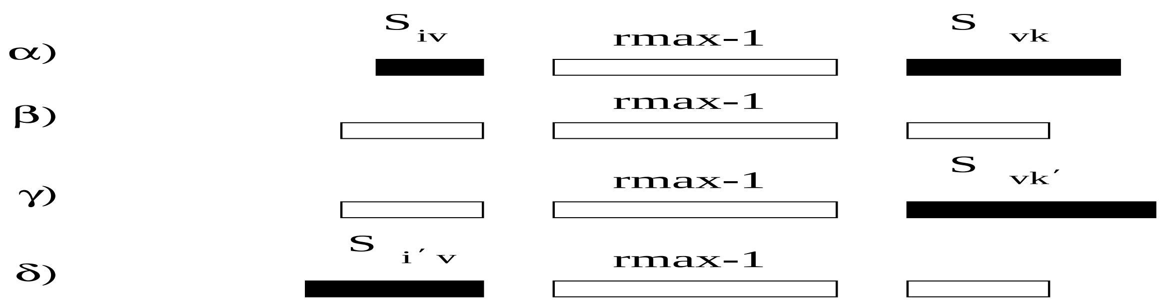
$$T_i^2 \leq T_i^1, \qquad \text{για κάθε } i \in J - \{ rmax-1 \}$$

όμως σύμφωνα με το λήμμα 3.3.1 έχουμε :
$T^2_{rmax-1} \leq T^2_{rmax}$ και από την τελευταία σχέση έχουμε ότι $T^2_{rmax} \leq T^1_{rmax}$ άρα, $T^2_{rmax-1} \leq T^1_{rmax}$

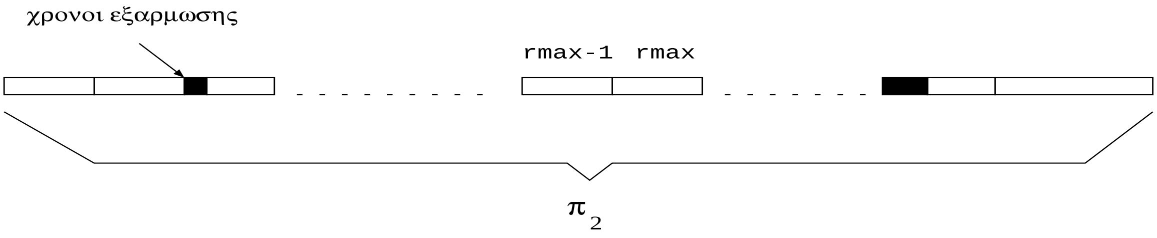
Συνεπώς το πρόγραμμα π_2 είναι επίσης βέλτιστο.

Οι περιπτώσεις β), γ) και δ) του σχήματος 3.4.4 είναι ειδικές μορφές της περίπτωσης α).

Συνεπώς οι εργασίες $rmax-1$ και $rmax$ αποτελούν παρτίδα. Με τον ίδιο συλλογισμό συνεχίζοντας από το πρόγραμμα π_2 δημιουργούμε την ακολουθία βέλτιστων προγραμμάτων $\pi_l, 1 \leq l \leq rmax$. Στο πρόγραμμα π_{rmax} οι εργασίες $1,2,\ldots,rmax-1,rmax$ αποτελούν παρτίδα. Συνεπώς υπάρχει βέλτιστο πρόγραμμα στο οποίο οι εργασίες $1,2,\ldots,rmax-1,rmax$ εκτελούνται η μια μετά την άλλη χωρίς διακοπή από άλλες εργασίες. Με αφετηρία το πρόγραμμα π_r και με τον ίδιο



Σχήμα 3.4.4 : Η γειτονιά της εργασίας $rmax-1$.



Σχήμα 3.4.5 : Το πρόγραμμα π_2.

συλλογισμό εύκολα συνάγεται πως υπάρχει βέλτιστο πρόγραμμα στο οποίο κάθε ακολουθία εργασιών που ορίζεται από κρίσιμες εργασίες με τον τρόπο που περιγράψαμε στο κεφ. 3.4 εκτελείται συνεχώς και συνεπώς αυτές οι ακολουθίες εργασιών αποτελούν παρτίδες.□

Ορισμός 3.4.2 Οι παρτίδες που ορίζονται με βάση τις κρίσιμες εργασίες καλούνται απλές παρτίδες.

Στο παραπάνω θεώρημα αξίζει να σημειώσουμε δύο βασικά χαρακτηριστικά. Πρώτον, έχει γενική ισχύ ως προς τους χρόνους εξάρμωσης αφού απαιτεί μόνο την υπόθεση της ισχύος της τριγωνικής ανισότητας. Δεύτερον, είναι πρόταση σύζευξης. Εστω $I = \{ I_1, I_2, \ldots, I_n \}$ ένα σύνολο προτάσεων για κάποιο πρόβλημα χρονικού προγραμματισμού. Γνωρίζουμε πως υπάρχουν βέλτιστα προγράμματα $\pi_1, \pi_2, \ldots, \pi_n$ που ικανοποιούν αντίστοιχα τις προτάσεις $I_1, I_2, \ldots, I_n$. Η πρόταση I είναι πρόταση διάζευξης. Αν συμβαίνει να υπάρχει βέλτιστο πρόγραμμα π_I^* που να ικανοποιεί κάθε πρόταση από τις $I_1, I_2, \ldots, I_n$, τότε λέμε πως η πρόταση I είναι πρόταση σύζευξης. Ειναι προφανές πως προτάσεις σύζευξης είναι πολύ πιο αποτελεσματικές από προτάσεις διάζευξης. Εύκολα μπορούμε να αντιστοιχίσουμε σε κάθε απλή παρτίδα $\left[J_{ib}^{Crit}\right]$ την πρόταση Π_{ib} που λέει πως σε κάποιο βέλτιστο πρόγραμμα οι εργασίες της απλής παρτίδας $\left[J_{ib}^{Crit}\right]$ εκτελούνται συνεχόμενες. Το σύνολο των προτάσεων $\Pi = \{\Pi_{ib}, b=1,\ldots,B, i=1,\ldots,N_b^{Crit}\}$ αποτελεί πρόταση σύζευξης.

Φυσικά η πρακτική αξία του Θεωρήματος 3.4.1 είναι προφανής μια και ελαττώνει τον αριθμό των εργασίων που προορίζονται για διάταξη. Ομως πέρα από αυτό το γεγονός οι παρτίδες οι οποίες δημιουργούνται ικανοποιούν συγκεκριμένες ιδιότητες που θα οδηγήσουν και στην διατύπωση αλγορίθμου για την επίλυση του $MTMS_{ij}$.

Αμεσες συνέπειες του θεωρήματος 3.4.1 είναι τα ακόλουθα πορίσματα.

Πόρισμα 3.4.1 Οι κρίσιμες εργασίες ορίζουν παρτίδες για το πρόβλημα $MTMS_j$.

Απόδειξη: Αρκεί να δείξω ότι χρόνοι εξάρμωσης ανεξαρτητοι ακολουθίας ικανοποιούν την τριγωνική ανισότητα. Προφανώς έχουμε

$$S_{ij} = S_j \le S_k + S_j = S_{ik} + S_{kj}. \square$$

Πόρισμα 3.4.2 Οι κρίσιμες εργασίες ορίζουν παρτίδες για το πρόβλημα $MTMS$.

Απόδειξη: Αμεση συνέπεια του πορίσματος 3.4.1 αφού το πρόβλημα $MTMS$ είναι ειδική περίπτωση του προβλήματος $MTMS_j$.$\square$

Πόρισμα 3.4.3 Αν στο πρόβλημα $MTMS_j$ κάθε ομάδα b, $1 \le b \le B$ έχει κρίσιμες εργασίες τάξεως 1 και μόνο, τότε ο EDD με βάση τις προθεσμίες $d_{N_b b}$, $1 \le b \le B$ δίνει βέλτιστη λύση.

Απόδειξη: Αφού κάθε ομάδα έχει κρίσιμες εργασίες τάξεως 1 και μόνο υπάρχει λύση στην οποία κάθε ομάδα παραμένει αδιάσπαστη. Συνεπώς το πρόβλημα εκφυλίζεται σέ πρόβλημα $B/1//T_{\max}$ που κατά τα γνωστά λύνεται με βάση τον

EDD. Οι προθεσμίες για τις αδιάσπαστες ομάδες δεν είναι παρά οι προθεσμίες των κρίσιμων εργασιών όπως το θεώρημα 3.7.1 αποδεικνύει.□

Πόρισμα 3.4.4 Αν στο πρόβλημα $MTMS$ κάθε ομάδα $b, 1 \leq b \leq B$ έχει κρίσιμες εργασίες τάξεως 1 και μόνο, τότε ο EDD με βάση τις προθεσμίες $d_{N_b b}$, $1 \leq b \leq B$ δίνει βέλτιστη λύση.

Απόδειξη: Προφανές αφού το πρόβλημα $MTMS$ είναι ειδική περίπτωση του προβλήματος $MTMS_j$.□

3.5. Σχέσεις διαδοχής παρτίδων για το $MTMS_{ij}$

Μετά την δημιουργία παρτίδων ενδιαφέρον παρουσιάζει η πιθανή ύπαρξη σχέσεων διαδοχής μεταξύ των παρτίδων. Το παρακάτω θεώρημα δίνει κάποιες απαντήσεις στο θέμα ορίζοντας μια σχέση μερικής διάταξης στις παρτίδες. Πριν την απόδειξη του θεωρήματος δίνουμε ορισμένους συμβολισμους που θα ισχύουν για την υπόλοιπη εργασία.

Με N_b^{Crit} συμβολίζουμε το πλήθος των παρτίδων (ίσο με το πλήθος των κρίσιμων εργασιών) στην ομάδα b. Με $\left[J_{ib}^{Crit}\right]$ συμβολίζουμε την i-οστή παρτίδα στην ομάδα b. Με $p\left[J_{ib}^{Crit}\right]$, $d\left[J_{ib}^{Crit}\right]$, $C\left[J_{ib}^{Crit}\right]$ και $T\left[J_{ib}^{Crit}\right]$ συμβολίζουμε τον χρόνο επεξεργασίας, την προθεσμία, τον χρόνο περάτωσης και την καθυστέρηση της i-οστής παρτίδας της ομάδας b. Επίσης με $T^0\left[J_{ib}^{Crit}\right]$ συμβολίζουμε την καθυστέρηση της i-οστής παρτίδας μέσα στην ομάδα b με βάση τον EDD και δεδομένου ότι ο χρόνος εναρξης του προγράμματος είναι ίσος με $d_{\max} = \max\limits_{\substack{1 \leq b \leq B \\ 1 \leq i \leq N_b^{Crit}}} \{d\left[J_{ib}^{Crit}\right]\}$.

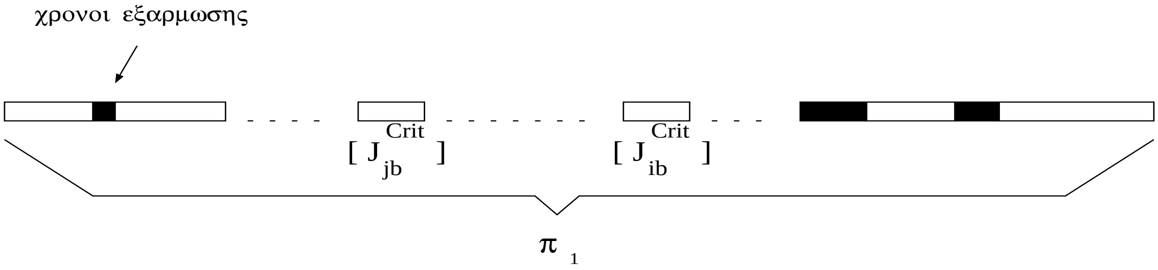
Θεώρημα 3.5.1 Εστω το πρόβλημα $MTMS_{ij}$, $1 \leq i \leq B$, $1 \leq j \leq B$, με χρόνους εξάρμωσης που ικανοποιούν την τριγωνική ανισότητα. Υπάρχει βέλτιστο πρόγραμμα στο οποίο οι παρτίδες κάθε ομάδας εμφανίζονται με την ίδια σειρά με την οποία εμφανίζονται και μέσα στην ομάδα κατά την δημιουργία τους.

Απόδειξη: Εστω δύο παρτίδες $\left[J_{ib}^{Crit}\right]$ και $\left[J_{jb}^{Crit}\right]$ για τις οποίες γνωρίζουμε ότι μέσα στην ομάδα b έχουμε

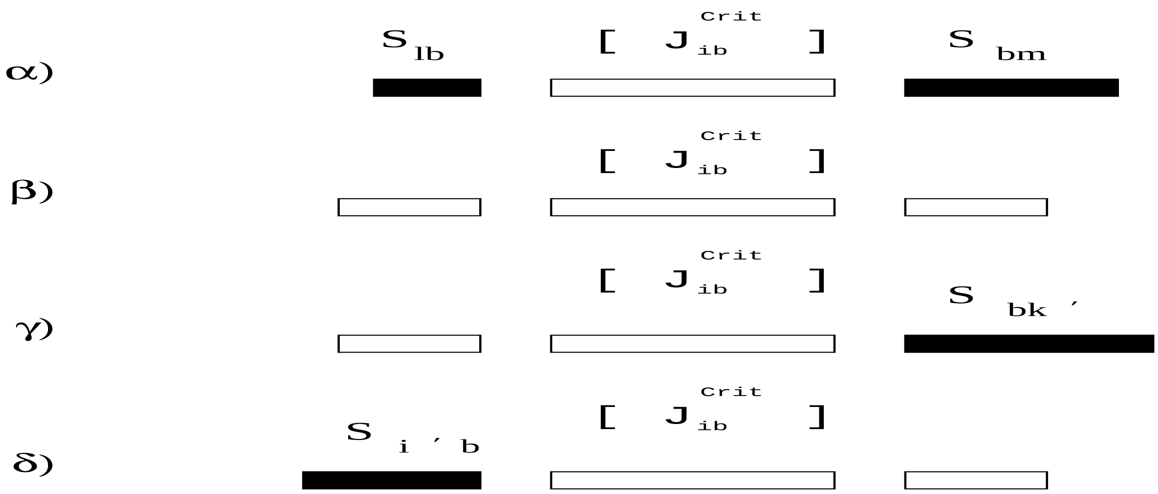
$$\left[J_{ib}^{Crit}\right] \leftarrow \left[J_{jb}^{Crit}\right]$$

ή ισοδύναμα ότι $i < j$. Με $\leftarrow$ συμβολίζουμε την σχέση "προηγείται". Ας θεωρήσουμε ένα βέλτιστο πρόγραμμα π_1 (σχήμα 3.5.1) στο οποίο έχουμε

$$\left[J_{jb}^{Crit}\right] \leftarrow \left[J_{ib}^{Crit}\right]$$



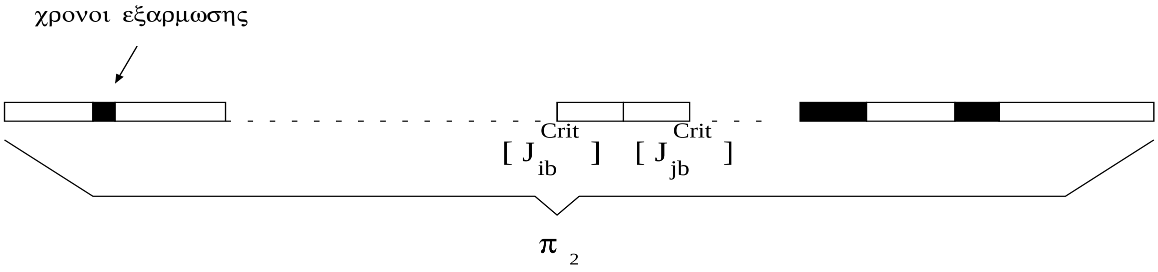
Σχήμα 3.5.1 : Το πρόγραμμα π_1.



Σχήμα 3.5.2 : Η γειτονιά της παρτίδας $\left[J_{ib}^{Crit}\right]$.

Η γειτονιά της παρτίδας $\left[J_{jb}^{Crit}\right]$ θα έχει μια από τις τέσσερεις δυνατές μορφές που απεικονίζονται στο σχήμα 3.5.2

Θεωρούμε το πρόγραμμα π_2 που προέρχεται από το π_1 με την μεταφορά της παρτίδας $\left[J_{jb}^{Crit}\right]$ ακριβώς δεξιά της παρτίδας $\left[J_{ib}^{Crit}\right]$ (σχήμα 3.5.3).



Σχήμα 3.5.3 : Το πρόγραμμα π_2.

Για το πρόγραμμα π_2 και για κάθε μια από τις περιπτώσεις του σχήματος 3.5.2 έχουμε :

α) λόγω της υπόθεσης ότι ισχύει η τριγωνική ανισότητα στους χρόνους εξάρμωσης έχουμε :

$$S_{lm} \leq S_{lb} + S_{bm}$$

και συνεπώς

$$T_i^2 \leq T_i^1$$

για κάθε i $\in$ J - $\left[J_{jb}^{Crit}\right]$ όμως

$$T_i^2 \leq \max_{i \in \left[J_{jb}^{Crit}\right]} T_i^2 \leq \max_{i \in \left[J_{ib}^{Crit}\right]} T_i^2 \leq \max_{i \in \left[J_{ib}^{Crit}\right]} T_i^1$$

για κάθε $i \in \left[J_{jb}^{Crit}\right]$. Συνεπώς το πρόγραμμα π_2 είναι επίσης βέλτιστο.

Οι περιπτώσεις β), γ) και δ) είναι ειδικές μορφές της α). Συνεπώς κάθε πρόγραμμα στο οποίο οι παρτίδες δεν ακολουθούν την διάταξη που έχουν μέσα στην ομάδα μπορεί να μετασχηματιστεί σε ένα νέο στο οποίο οι παρτίδες της ίδιας ομάδας ακολουθούν την διάταξη που υπαγορεύεται από την ομάδα.□

Αμεσες συνέπειες του θεωρήματος 3.5.1 είναι τα παρακάτω πορίσματα.

Πόρισμα 3.5.1 Για το πρόβλημα $MTMS_j$ υπάρχει βέλτιστο πρόγραμμα στο οποίο οι παρτίδες κάθε ομάδας εμφανίζονται με την ίδια σειρά με την οποία εμφανίζονται και μέσα στην ομάδα κατά την διαδικασία της δημιουργίας τους.

Πόρισμα 3.5.2 Για το πρόβλημα $MTMS$ υπάρχει βέλτιστο πρόγραμμα στο οποίο οι παρτίδες κάθε ομάδας εμφανίζονται με την ίδια σειρά με την οποία εμφανίζονται και μέσα στην ομάδα κατά την διαδικασία της δημιουργίας τους.

3.6. Δεμένες παρτίδες

Μέχρι εδώ με τον ορισμό των απλών παρτίδων έχουμε κατορθώσει την ελάττωση των εργασιών του προβλήματος $MTMS_{ij}$. Στην συνέχεια εισάγουμε την έννοια της <u>δεμένης</u> <u>παρτίδας</u> με σκοπό την παραπέρα ελάττωση των εργασιών. Ενα βασικό θεώρημα που αποτελεί ικανή συνθήκη για να χαρακτηριστεί μια παρτίδα δεμένη αποδεικνύεται για το πρόβλημα $MTMS_{ij}$. Αντίστοιχα απλοποιημένα πορίσματα παρέχονται για τα προβλήματα $MTMS_j$ και $MTMS$. Ξεκινάμε το κεφάλαιο αυτό με έναν ορισμό.

Ορισμός 3.6.1 Μια παρτίδα $\left[J_{ib}^{Crit}\right]$ ονομάζεται δεμένη αν υπάρχει βέλτιστο πρόγραμμα (ως προς το $MTMS_{ij}$) στο οποίο η $\left[J_{ib}^{Crit}\right]$ έχει δεξιά της ή (και) αριστερά της παρτίδα από την ίδια ομάδα.

Αν η παρτίδα $\left[J_{ib}^{Crit}\right]$ έχει δεξιά (αριστερά) της παρτίδα από την ίδια ομάδα τότε ονομάζεται δεξιά δεμένη (αριστερά δεμένη). Αν μια παρτίδα είναι και δεξιά και αριστερά δεμένη τότε ονομάζεται ολικά δεμένη. Φυσικά μια δεξιά δεμένη παρτίδα $\left[J_{ib}^{Crit}\right]$ σε κάποιο βέλτιστο πρόγραμμα ακολουθείται αμέσως από την παρτίδα $\left[J_{i+1b}^{Crit}\right]$. Αντίστοιχα μια αριστερά δεμένη παρτίδα $\left[J_{ib}^{Crit}\right]$ σε κάποιο βέλτιστο πρόγραμμα ακολουθεί την παρτίδα $\left[J_{i-1b}^{Crit}\right]$, ενω μια ολικά δεμένη παρτίδα $\left[J_{ib}^{Crit}\right]$ σε κάποιο βέλτιστο πρόγραμμα ευρίσκεται μεταξύ των παρτίδων $\left[J_{i-1b}^{Crit}\right]$ και $\left[J_{i+1b}^{Crit}\right]$. Ας παρατηρήσουμε ότι η παρτίδα $\left[J_{1b}^{Crit}\right]$ μπορεί να είναι μόνο δεξιά δεμένη. Αντίστοιχα η παρτίδα $\left[J_{N_b^{Crit}b}^{Crit}\right]$ μπορεί να είναι μόνο αριστερά δεμένη.

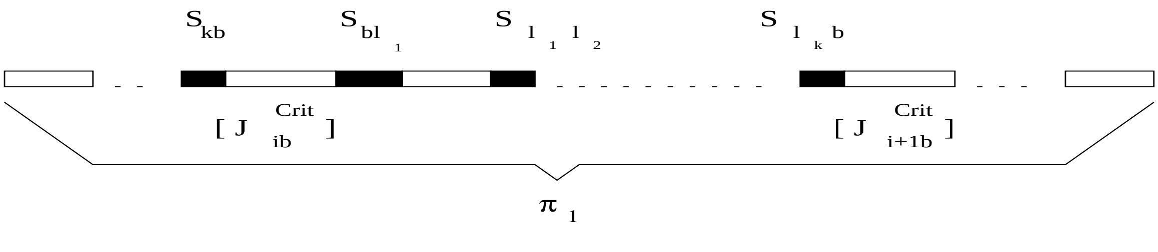
Το παρακάτω θεώρημα δίνει μια ικανή συνθήκη για να είναι η παρτίδα $\left[J_{ib}^{Crit}\right]$ δεμένη.

Θεώρημα 3.6.1 Σε κάθε ομάδα b αντιστοιχίζουμε τον φυσικό αριθμό $\alpha_b = \min\{\beta_b, \gamma_b\}$, όπου $\beta_b = \min_{l}\{S_{bl}+S_{lb}\}$ και $\gamma_b = \min_{k,l}\{S_{kb}+S_{bl}-S_{kl}\}$. Φυσικά για τους χρόνους εξάρμωσης υποθέτουμε ότι ισχύει η τριγωνική ανισότητα.
Αν για δύο διαδοχικές παρτίδες της ίδιας ομάδας b $\left[J_{ib}^{Crit}\right]$, $\left[J_{i+1b}^{Crit}\right]$ έχουμε

$$T^0\left[J_{ib}^{Crit}\right] < T^0\left[J_{i+1b}^{Crit}\right] + \alpha_b,$$

τότε η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι δεμένη. Με $T^0\left[J_{ib}^{Crit}\right]$ συμβολίζουμε την καθυστέρηση της παρτίδας $\left[J_{ib}^{Crit}\right]$ μέσα στην ομάδα b με βάση τον EDD και δεδομένου ότι ο χρόνος εναρξης του προγράμματος είναι ίσος με $d_{\max} = \max\limits_{\substack{1\le b\le B \\ 1\le i\le N_b^{Crit}}}\{d\left[J_{ib}^{Crit}\right]\}$.

Απόδειξη: Σύμφωνα με το θεώρημα 3.5.1 αφού $\left[J_{ib}^{Crit}\right] \leftarrow \left[J_{i+1b}^{Crit}\right]$ μέσα στην ομάδα b υπάρχει βέλτιστο πρόγραμμα π_1 στο οποίο $\left[J_{ib}^{Crit}\right] \leftarrow \left[J_{i+1b}^{Crit}\right]$. Αν συμβαίνει οι $\left[J_{ib}^{Crit}\right]$ και $\left[J_{i+1b}^{Crit}\right]$ να είναι διαδοχικές στο π_1 ή οι παρτίδες $\left[J_{i-1b}^{Crit}\right]$ και $\left[J_{ib}^{Crit}\right]$ να είναι διαδοχικές τότε το θεώρημα ισχύει. Διαφορετικά το πρόγραμμα π_1 θα είναι όπως στο σχήμα 3.6.1 .



Σχήμα 3.6.1 : Το πρόγραμμα π_1.

Εστω t η χρονική στιγμή έναρξης της παρτίδας $\left[J_{i+1b}^{Crit}\right]$ στο πρόγραμμα π_1. Θα δείξουμε οτι ισχύει $t - C\left[J_{ib}^{Crit}\right] > \alpha_b$. Πράγματι

$$t - C\left[J_{ib}^{Crit}\right] > S_{bl_1} + S_{l_1 l_2} + \cdots + S_{l_{k-1} l_k} + S_{l_k b} \geq$$

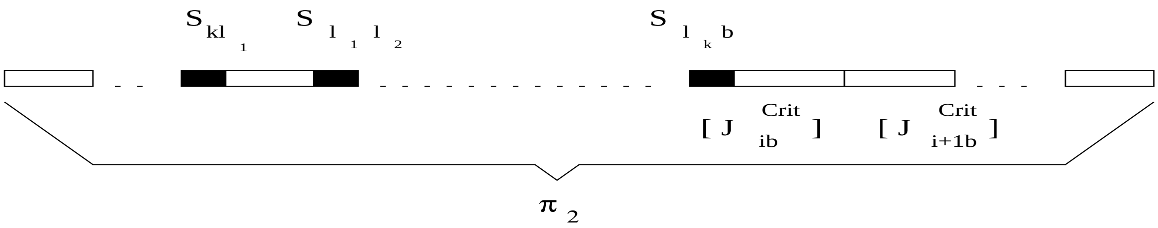
$$S_{bl_2} + S_{l_2 l_3} + \cdots + S_{l_{k-1} l_k} + S_{l_k b} \geq \cdots \geq$$

$$S_{bl_k} + S_{l_k b} \geq \beta_b \geq \alpha_b$$

Συνεπώς με βάση το λήμμα 3.3.2 έχουμε :

$$T^1\left[J_{i+1b}^{Crit}\right] \geq T^1\left[J_{ib}^{Crit}\right].$$

Θεωρούμε το πρόγραμμα π_2 που προκύπτει από το π_1 με την μεταφορά της παρτίδας $\left[J_{ib}^{Crit}\right]$ ακριβώς αριστερά της $\left[J_{i+1b}^{Crit}\right]$ (σχήμα 3.6.2).



Σχήμα 3.6.2 : Το πρόγραμμα π_2.

Για το πρόγραμμα π_2 έχουμε

$$T^2\left[J_{jb}^{Crit}\right] \leq T^1\left[J_{jb}^{Crit}\right]$$

για κάθε $1 \leq b \leq B,\ 1 \leq j \leq N_b^{Crit},\ J \neq i$.

Επίσης για την παρτίδα $\left[J_{i+1b}^{Crit}\right]$ ξέρουμε ότι στο πρόγραμμα π_2 ξεκινά τουλάχιστον α_b μονάδες χρόνου νωρίτερα από τον χρόνο έναρξης στο πρόγραμμα π_1 αφού

$$S_{kb} + S_{bl_1} - S_{kl_1} \geq \gamma_b \geq \alpha_b$$

και συνεπώς $$T^2\left[J_{i+1b}^{Crit}\right] \leq T^1\left[J_{i+1b}^{Crit}\right] - \alpha_b. \qquad (3.6.1)$$

Ομως

$$T^2\left[J_{ib}^{Crit}\right] \overset{*}{\leq} T^2\left[J_{i+1b}^{Crit}\right] + \alpha_b \overset{(3.6.1)}{\leq} T^1\left[J_{i+1b}^{Crit}\right].$$

* Η υπόθεση μεταφερμένη την κατάλληλη στιγμή με την βοήθεια του λήμματος 3.3.1 που λέει ότι η κατανομή της καθυστέρησης είναι ανεξάρτητη της ολίσθησης στον χρόνο. □

Αμεσες συνέπειες του θεωρήματος 3.6.1 είναι τα δύο πορίσματα που ακολουθούν για τις περιπτώσεις που έχουμε χρόνους εξάρμωσης ανεξάρτητους ακολουθίας ή σταθερούς.

Πόρισμα 3.6.1 Εστω το πρόβλημα $MTMS_j$. Αν για δύο διαδοχικές παρτίδες της ίδιας ομάδας $\left[J_{ib}^{Crit}\right]$ και $\left[J_{i+1b}^{Crit}\right]$ έχουμε :

$$T^0\left[J_{ib}^{Crit}\right] < T^0\left[J_{i+1b}^{Crit}\right] + S_b,$$

τότε η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι δεμένη.

Πόρισμα 3.6.2 Εστω το πρόβλημα $MTMS$. Αν για δύο διαδοχικές παρτίδες της ίδιας ομάδας $\left[J_{ib}^{Crit}\right]$ και $\left[J_{i+1b}^{Crit}\right]$ έχουμε :

$$T^0\left[J_{ib}^{Crit}\right] < T^0\left[J_{i+1b}^{Crit}\right] + S_{const},$$

τότε η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι δεμένη.

Οπως θα δούμε αργότερα ο αλγόριθμος που προτείνεται λειτουργεί παίρνοντας διαδοχικές αποφάσεις για το αν μια παρτίδα θα πρέπει να δεθεί ή όχι. Προφανώς για παρτίδες που χαρακτηρίζονται ως δεμένες δεν έχουμε το δίλημμα να τις δέσουμε ή όχι, αφού γνωρίζουμε πως υπάρχει βέλτιστη λύση στην οποία οι παρτίδες αυτές είναι δεμένες. Συνεπώς το θεώρημα 3.6.1 περιορίζει τον χώρο έρευνας για την εύρεση βέλτιστης λύσης. Απο θεωρητική άποψη ο χαρακτηρισμός μιας παρτίδας ως δεμένης ελαττώνει στο μισό τον χώρο έρευνας για βέλτιστη λυση. Απο τα πειράματα που έγιναν για τις επιδόσεις του αλγορίθμου φάνηκε πως στην περίπτωση που οι χρόνοι εξάρμωσης είναι αρκετά μεγάλοι (ως ποσοστό του μέσου χρόνου επεξεργασίας) τότε η ελάττωση του χώρου έρευνας είναι πολύ σημαντική. Δηλαδή, με άλλα λόγια, όταν η διαδικασία του ορθού προγραμματισμού της μηχανής είναι κρίσιμη τότε και η συμπεριφορά του αλγορίθμου είναι καλύτερη. Επίσης για το πρόβλημα $MTMS_j$ όπου οι χρόνοι εξάρμωσης δέν είναι όλοι ίδιοι βλέπουμε μια πολύ καλύτερη συμπεριφορά του αλγορίθμου η οποία οφείλεται στην ελάττωση του χώρου έρευνας από το θεώρημα 3.6.1 και μόνο. Πιο λεπτομερή όμως αξιολόγηση τόσο του θεωρήματος 3.6.1 όσο και των άλλων θα γίνει σε επόμενο κεφάλαιο.

Ενδιαφέρον παρουσιάζει η παρακάτω παρατήρηση

- το θεώρημα 3.6.1 μετά από μερικές πράξεις μπορεί να πάρει την παρακάτω μορφή για το πρόβλημα $MTMS_j$

$$
\begin{aligned}
T^0\left[J_{ib}^{Crit}\right] - T^0\left[J_{i+1b}^{Crit}\right] &< S_b \Leftrightarrow \\
t+p\left[J_{ib}^{Crit}\right]-d\left[J_{ib}^{Crit}\right]-(t+p\left[J_{ib}^{Crit}\right]+p\left[J_{i+1b}^{Crit}\right]-d\left[J_{i+1b}^{Crit}\right]) &< S_b \Leftrightarrow \\
d\left[J_{i+1b}^{Crit}\right]-p\left[J_{i+1b}^{Crit}\right]-d\left[J_{ib}^{Crit}\right] &< S_b \Leftrightarrow \\
d\left[J_{i+1b}^{Crit}\right] &< p\left[J_{i+1b}^{Crit}\right] + d\left[J_{ib}^{Crit}\right] + S_b
\end{aligned} \tag{3.6.2}
$$

Συνεπώς αν δυο διαδοχικές παρτίδες $\left[J_{ib}^{Crit}\right]$ και $\left[J_{i+1b}^{Crit}\right]$ ικανοποιούν την (3.6.1) τότε η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι δεμένη.

3.7. Ιδιότητες των παρτίδων

Οπως έχουμε ήδη αναφέρει με $d\left[J_{ib}^{Crit}\right]$ και $p\left[J_{ib}^{Crit}\right]$ συμβολίζουμε την προθεσμία και τον χρόνο επεξεργασίας της παρτίδας $\left[J_{ib}^{Crit}\right]$. Σ' αυτό το κεφάλαιο θα δώσουμε συγκεκριμένες αριθμητικές εκφράσεις για τις προθεσμίες και για τους χρόνους επεξεργασίας των παρτίδων, καθως και μία βασική ιδιότητά τους που συνδέει την προθεσμία και τον χρόνο επεξεργασίας της παρτίδας τάξεως i+1 με την προθεσμία της παρτίδας τάξεως i (φυσικά εννοούμε παρτίδες από την ίδια ομάδα b).

Το θεώρημα 3.7.1 αποδεικνύει πως η συμπεριφορά μίας παρτίδας $\left[J_{ib}^{Crit}\right]$ ως προς την καθυστέρηση είναι ταυτόσημη με την συμπεριφορά της κρίσιμης εργασίας J_{ib}^{Crit} που ορίζει και την παρτίδα. Συνεπώς η προθεσμία της παρτίδας $\left[J_{ib}^{Crit}\right]$ είναι ίση με την προθεσμία της εργασίας J_{ib}^{Crit}, δηλαδή

$$d\left[J_{ib}^{Crit}\right] = d_{J_{ib}^{Crit}b} \tag{3.7.1}$$

Σχετικά με τον χρόνο επεξεργασίας $p\left[J_{ib}^{Crit}\right]$ της παρτίδας $\left[J_{ib}^{Crit}\right]$ είναι φανερό ότι ισχύει η παρακάτω σχέση

$$p\left[J_{ib}^{Crit}\right] = \sum_{j \in \left[J_{ib}^{Crit}\right]} p_{jb} \tag{3.7.2}$$

Στο παρακάτω θεώρημα με A συμβολίζουμε την ακολουθία των εργασιών $J_1, J_2, \ldots, J_n$. Κάθε εργασία J_i έχει χρόνο επεξεργασίας p_{J_i} και προθεσμία d_{J_i}. Με $T_{J_i}(t)$ συμβολίζουμε την καθυστέρηση της εργασίας J_i αν η εκτέλεση της A ξεκινήσει την χρονική στιγμή t. Με $T_A(t)$ συμβολίζουμε την καθυστέρηση της A δεδομένου ότι η εκτέλεσή της αρχίζει την χρονική στιγμή t. Τέλος με J_{i_0} συμβολίζουμε την εργασία που εμφανίζει μέγιστη καθυστέρηση δεδομένου ότι η εκτέλεση της A ξεκινάει την χρονική στιγμή $t > \max\limits_{1 \le i \le n} d_{J_i}$.

Θεώρημα 3.7.1 Η μέγιστη καθυστέρηση που εμφανίζεται σε μια ακολουθία εργασιών A, είναι ίση με την μέγιστη καθυστέρηση μιας και μόνο εργασίας J με χρόνο επεξεργασίας $P = \sum_{i=1}^{n} p_{J_i}$ και προθεσμία $D = d_{J_{i_0}} + \sum_{i=i_0+1}^{n} p_{J_i}$.

Απόδειξη: Αυτό που θέλουμε να δείξουμε είναι

$$T_A(t) = T_J(t), \qquad \text{για κάθε } t \in \mathrm{R}$$

όπου με $T_J(t)$ συμβολίζουμε την καθυστέρηση της πλασματικής εργασίας J δεδομένου ότι ξεκινά η εκτέλεση της την χρονική στιγμή t.
Πολύ απλά έχουμε:

$$\begin{aligned} T_A(t) &= \max(0, t + \sum_{i=1}^{i_0} p_{J_i} - d_{J_{i_0}}) = \\ &\max(0, t + \sum_{i=1}^{n} p_{J_i} - (d_{J_{i_0}} + \sum_{i=i_0+1}^{n} p_{J_i})) = \\ &= T_J(t) \end{aligned} \tag{3.7.3}$$

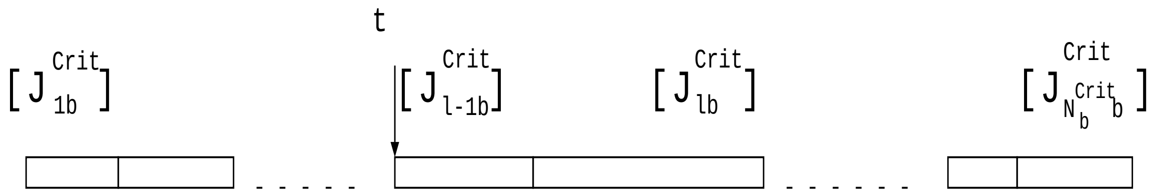
Για την σχέση (3.7.3) χρησιμοποιήθηκε το λήμμα 3.3.1 .□

Το παρακάτω θεώρημα αποτελεί βασική ιδιότητα που ικανοποιούν όλες οι παρτίδες που ανήκουν στην ίδια ομάδα.

Θεώρημα 3.7.2 Εστω δύο παρτίδες $\left[J_{ib}^{Crit}\right]$ και $\left[J_{jb}^{Crit}\right]$ της ομάδας b με $i>j$. Η ακόλουθη σχέση συνδέει τις προθεσμίες των παρτίδων.

$$d\left[J_{ib}^{Crit}\right] > \sum_{l=j+1}^{i} p\left[J_{lb}^{Crit}\right] + d\left[J_{jb}^{Crit}\right]$$

Απόδειξη : Αρχικά θα δείξουμε ότι $d\left[J_{lb}^{Crit}\right] > p\left[J_{lb}^{Crit}\right] + d\left[J_{l-1b}^{Crit}\right]$ για κάθε $1<l\leq N_b^{Crit}$. Με $T\left[J_{lb}^{Crit}\right](t)$ συμβολίζουμε την καθυστέρηση της παρτίδας $\left[J_{lb}^{Crit}\right]$ δεδομένου ότι η εκτέλεση της θα αρχίσει την χρονική στιγμή t.



Σχήμα 3.7.1 : Διάταξη της ομάδας b με βάση τον EDD.

Επειδή η κατανομή των καθυστερήσεων παραμένει αμετάβλητη ως προς την ολίσθηση στον χρόνο, μπορούμε να υποθέσουμε πως λύνουμε το πρόβλημα κάποια χρονική στιγμή $t > \max_b d\left[J_{N_b^{Crit}b}^{Crit}\right]$. Σε συνδυασμό με τον ορισμό των παρτίδων (σχήμα 3.7.1) έχουμε ότι:

$$T\left[J_{l-1b}^{Crit}\right](t) > T\left[J_{lb}^{Crit}\right]\left(t+p\left[J_{l-1b}^{Crit}\right]\right) \Rightarrow$$

$$t+p\left[J_{l-1b}^{Crit}\right]-d\left[J_{l-1b}^{Crit}\right] > t+p\left[J_{l-1b}^{Crit}\right]+p\left[J_{lb}^{Crit}\right]-d\left[J_{lb}^{Crit}\right] \Rightarrow$$

$$d\left[J_{lb}^{Crit}\right] > p\left[J_{lb}^{Crit}\right]+d\left[J_{l-1b}^{Crit}\right] \tag{3.7.4}$$

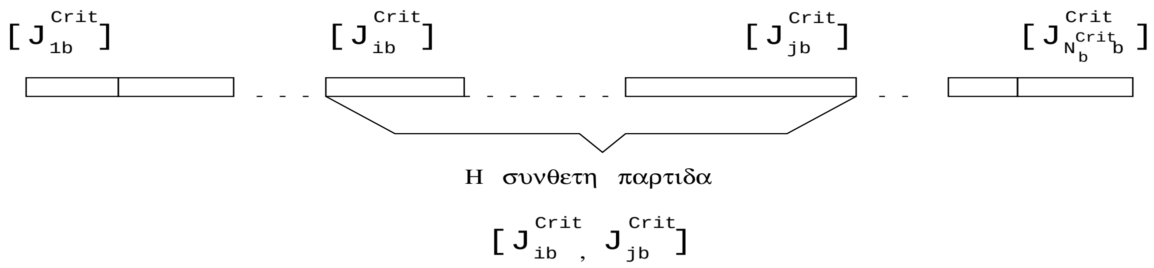
Με συνεχή εφαρμογή της σχέσεως (3.7.4) έχουμε :

$$d\left[J_{ib}^{Crit}\right] > p\left[J_{ib}^{Crit}\right]+d\left[J_{i-1b}^{Crit}\right] > \ldots > \sum_{l=j+1}^{i} p\left[J_{lb}^{Crit}\right] + d\left[J_{jb}^{Crit}\right]$$

και συνεπώς το θεώρημα ισχύει.□

3.8. Σύνθετες παρτίδες και ιδιότητές τους

Η σύνθετη παρτίδα είναι μια ακολουθία απλών παρτίδων από την ίδια ομάδα. Μια σύνθετη παρτίδα θα την συμβολίζουμε με μια δυάδα. Το πρώτο στοιχείο της δυάδας είναι η κρίσιμη εργασία της πρώτης παρτίδας της σύνθετης παρτίδας ενώ το δεύτερο είναι η κρίσιμη εργασία της τελευταίας παρτίδας της σύνθετης παρτίδας. Ετσι η σύνθετη παρτίδα που περιλαμβάνει από την i μέχρι και την j παρτίδα της ομάδας b συμβολίζεται με $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ και απεικονίζεται στο σχήμα 3.8.1



Σχήμα 3.8.1 : Η σύνθετη παρτίδα $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$.

Με $p\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$, $d\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$, $C\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ και $T\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ συμβολίζουμε τον χρόνο επεξεργασίας, την προθεσμία, τον χρόνο περάτωσης και την καθυστέρηση της σύνθετης παρτίδας $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ αντίστοιχα.

Για τον χρόνο επεξεργασίας $p\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ της σύνθετης παρτίδας $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ είναι προφανές ότι ισχύει η παρακάτω σχέση

$$p\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right] = \sum_{l=i}^{j} p\left[J_{lb}^{Crit}\right].$$

Για την προθεσμία $d\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ της σύνθετης παρτίδας $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ ισχύει η σχέση :

$$d\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right] = d\left[J_{ib}^{Crit}\right] + \sum_{l=i+1}^{j} p\left[J_{lb}^{Crit}\right] \tag{3.8.1}$$

Η σχέση (3.8.1) προκύπτει αμέσως από το θεώρημα 3.7.1 αφού το λήμμα 3.3.1 μας εγγυάται πως η μέγιστη καθυστέρηση της σύνθετης παρτίδας $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$ θα εμφανιστεί στην πρώτη παρτίδα $\left[J_{ib}^{Crit}\right]$ και συγκεκριμένα στην κρίσιμη εργασία της, J_{ib}^{Crit}. Τα δύο επόμενα θεωρήματα χαρακτηρίζουν τις προθεσμίες των συνθέτων παρτίδων. Το πρώτο θεώρημα χαρακτηρίζει την σχέση των προθεσμιών δύο συνθέτων παρτίδων που αρχίζουν ή τελειώνουν με την ίδια παρτίδα με βάση την σχέση των ανίσων δεικτών των παρτίδων. Το δεύτερο θεώρημα ορίζει μία περιοχή τιμών για την προθεσμία κάθε σύνθετης παρτίδας.

Θεώρημα 3.8.1 Δυάδες συνθέτων παρτίδων που αρχίζουν (τελειώνουν) με την (στην) ίδια απλή παρτίδα συνδέονται με τις παρακάτω σχέσεις ισοδυναμίας

$$d\left[J_{lb}^{Crit},J_{jb}^{Crit}\right] > d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right] \Leftrightarrow l>i \quad \text{και} \tag{3.8.2}$$

$$d\left[J_{jb}^{Crit},J_{lb}^{Crit}\right] > d\left[J_{jb}^{Crit},J_{ib}^{Crit}\right] \Leftrightarrow l>i \tag{3.8.3}$$

Απόδειξη: Πρώτα θα δείξουμε το αντίστροφο της (3.8.2).

(3.8.2)($\Leftarrow$)

Επειδή γνωρίζουμε ότι $l>i$ έχουμε :

$$d\left[J_{lb}^{Crit},J_{jb}^{Crit}\right] = d\left[J_{lb}^{Crit}\right] + \sum_{m=l+1}^{j} p\left[J_{mb}^{Crit}\right] >$$

$$d\left[J_{l-1b}^{Crit}\right] + p\left[J_{lb}^{Crit}\right] + \sum_{m=l+1}^{j} p\left[J_{mb}^{Crit}\right] >\ldots>$$

$$d\left[J_{ib}^{Crit}\right] + \sum_{m=i+1}^{j} p\left[J_{mb}^{Crit}\right] = d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$$

και συνεπώς ισχύει.

(3.8.2)($\Rightarrow$)

Αρνηση του συμπεράσματος οδηγεί στο συμπέρασμα ότι

$$d\left[J_{lb}^{Crit},J_{jb}^{Crit}\right] < d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right],$$

πράγμα που αντιβαίνει στην υπόθεση.

(3.8.3)($\Leftarrow$)

Επειδή γνωρίζουμε ότι $l>i$ έχουμε

$$d\left[J_{jb}^{Crit},J_{lb}^{Crit}\right] = d\left[J_{jb}^{Crit}\right] + \sum_{m=j+1}^{l} p\left[J_{mb}^{Crit}\right] >$$

$$d\left[J_{jb}^{Crit}\right] + \sum_{m=j+1}^{i} p\left[J_{mb}^{Crit}\right] = d\left[J_{jb}^{Crit},J_{ib}^{Crit}\right]$$

και συνεπώς ισχύει.

(3.8.3)($\Rightarrow$)

Αρνηση του συμπεράσματος οδηγεί σε άτοπο όπως και προηγουμένως.□

Θεώρημα 3.8.2 Η προθεσμία κάθε σύνθετης παρτίδας $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$ ικανοποιεί την σχέση

$$\min_{v}\{\ d\left[J_{1v}^{Crit}\right]\ \} \le d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right] \le \max_{v}\{\ d\left[J_{N_v^{Crit}v}^{Crit}\right]\ \}.$$

Απόδειξη: Κάνοντας χρήση του θεωρήματος 3.8.1 έχουμε :

$$\min_{v}\{\ d\left[J_{1v}^{Crit}\right]\ \} \le d\left[J_{1b}^{Crit}\right] = d\left[J_{1b}^{Crit},J_{1b}^{Crit}\right] \le$$

$$d\left[J_{1b}^{Crit},J_{jb}^{Crit}\right] \le d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right] \le d\left[J_{ib}^{Crit},J_{N_b^{Crit}b}^{Crit}\right] \le$$

$$d\left[J_{N_b^{Crit}b}^{Crit},J_{N_b^{Crit}b}^{Crit}\right] = d\left[J_{N_b^{Crit}b}^{Crit}\right] \le \max_{v}\{\ d\left[J_{N_v^{Crit}v}^{Crit}\right]\ \}.\square$$

Ξέρουμε ότι για το πρόβλημα $MTMS_j$ υπάρχει βέλτιστη λύση π_0 στην οποία οι ακολουθίες εργασιών που ορίζονται με βάση τις κρίσιμες εργασίες αποτελούν παρτίδες και μάλιστα εμφανίζονται στο π_0 με την ίδια σειρά με την οποία εμφανίζονται και μέσα στις ομάδες κατά την διάρκεια της δημιουργίας τους.

Εστω $\pi_i,\ i \in I$ όλα τα βέλτιστα προγράμματα του προβλήματος $MTMS_j$ που έχουν την ίδια ιδιότητα με το π_0. Κάθε πρόγραμμα $\pi_i,\ i \in I$ ορίζει ακολουθίες από παρτίδες της ίδιας ομάδας που οριοθετούνται από χρόνους εξάρμωσης. Αυτές τις ακολουθίες παρτίδων τις ονομάζουμε <u>βέλτιστες σύνθετες παρτίδες.</u>

Οι βέλτιστες σύνθετες παρτίδες σε ένα πρόγραμμα $\pi_{i_0} \in I$ δεν είναι κατ' ανάγκη τοποθετημένες με βάση τον EDD. Ομως αν τις τοποθετούσαμε κατά EDD το νέο πρόγραμμα θα ήταν και πάλι βέλτιστο. Αυτό γιατί όποτε δύο βέλτιστες σύνθετες παρτίδες δεν ακολουθούν τον EDD μπορούμε να τις αναγκάσουμε να συμμορφωθούν σε αυτόν μεταφέροντας την παρτίδα που έχει τον μικρότερο χρόνο έναρξης ακριβώς δεξιά της παρτίδας που έχει τον μεγαλύτερο χρόνο έναρξης και να παραμείνουμε και πάλι βέλτιστοι όπως ακριβώς έγινε και στήν απόδειξη του θεωρήματος 3.5.1 .

Βέβαια δεν γνωρίζουμε ποιες είναι οι βέλτιστες σύνθετες παρτίδες αλλά προσπαθούμε να βρούμε ιδιότητές τους και χαρακτηριστικά τους με τελικό σκοπό να τις κατασκευάσουμε όσο το δυνατόν πιο εύκολα και πιο γρήγορα. Φυσικά, αφού τις κατασκευάσουμε, απλά τις διατάσσουμε κατά EDD και έχουμε μια βέλτιστη λύση για το $MTMS_j$. Τα δύο προηγούμενα θεωρήματα κινήθηκαν προς αυτήν την κατεύθυνση.

3.9. Ειδικές περιπτώσεις του $MTMS_j$

Ηδη στο κεφ. 3.4 αναφέρουμε μία ειδική περίπτωση όπου κάθε ομάδα $b, 1 \le b \le B$ έχει κρίσιμες εργασίες τάξεως 1 και μόνο. Το πόρισμα 3.4.3 έδινε τον χαρακτηρισμό της λύσης σε αυτή την ειδική περίπτωση.

Αλλη μία ειδική περίπτωση αναφέρεται σε αυτό το κεφάλαιο. Πριν δώσουμε το θεώρημα και την απόδειξή του θα δώσουμε την φυσική του ερμηνεία. Με απλά λόγια το θεώρημα 3.9.1 λέει πως αν ο EDD κατά την εφαρμογή του αφήνει τις ομάδες αδιάσπαστες τότε το πρόγραμμα που προκύπτει από αυτόν είναι βέλτιστο.

Θεώρημα 3.9.1 Αν για κάθε ζεύγος ομάδων b, b' η λογική έκφραση

$$(d_{N_b b} < d_{1b'}) \text{ ή } (d_{N'_b b'} < d_{1b}) \tag{3.9.1}$$

είναι αληθής τότε ο EDD ως προς τις προθεσμίες των εργασιών δίνει βέλτιστη λύση.

Απόδειξη: Αρχικά παρατηρούμε πως για οποιαδήποτε ζεύγος ομάδων b, b' μπορεί να ισχύει μόνο η μια από τις δύο υποπροτάσεις που αποτελούν την (3.9.1). Χωρίς βλάβη της γενικότητας υποθέτουμε ότι ισχύει η πρώτη υποπρόταση. Δηλαδή $d_{N_b b} < d_{1b'}$. Για τις τυχαίες παρτίδες $\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right]$, και $\left[J_{i'b'}^{Crit}, J_{j'b'}^{Crit}\right]$ της ομάδας b και b' έχουμε :

$$d\left[J_{ib}^{Crit}, J_{jb}^{Crit}\right] \le d\left[J_{N_b^{Crit} b}^{Crit}, J_{N_b^{Crit} b}^{Crit}\right] = d_{N_b b} < d_{1b'} = d\left[J_{1b'}^{Crit}, J_{1b'}^{Crit}\right] <$$
$$d\left[J_{1b'}^{Crit}, J_{j'b'}^{Crit}\right] \le d\left[J_{i'b'}^{Crit}, J_{j'b'}^{Crit}\right]$$

Συνεπώς κάθε σύνολο βέλτιστων σύνθετων παρτίδων με βάση τον EDD θα δώσει πρόγραμμα που θα αφήνει τις ομάδες αδιάσπαστες και φυσικά θα είναι και βέλτιστο. Αυτό το πρόγραμμα όμως δεν είναι παρά το πρόγραμμα που προκύπτει κατά την εφαρμογή του EDD στις εργασίες.□

Στο σημείο αυτό θα πρέπει να επισημάνουμε πως αντικειμενικός στόχος της εργασίας είναι η επίλυση του προβλήματος $MTMS_j$. Ομως οι αποδείξεις των θεωρημάτων γίνονται (όποτε είναι δυνατο) για το πιο γενικό πρόβλημα $MTMS_{ij}$ και οι αντίστοιχες προτάσεις για το $MTMS_j$ απορρέουν ως πορίσματα. Ο λόγος που περιορίζουμε την εργασία στο πρόβλημα $MTMS_j$ (και κατα συνέπεια και στο $MTMS$) αφορά αφ' ενός την εκτασή της εργασίας και αφ' ετέρου την διαφορετική αντιμετώπιση που απαιτεί το πρόβλημα $MTMS_{ij}$. Η ιδέα της επίλυσης του προβλήματος με την δημιουργία παρτίδων δεν είναι εφαρμόσιμη (άμεσα) μια και η εύρεση ενός συνόλου βέλτιστων παρτίδων δεν οδηγεί σε λύση του προβλήματος αλλά σε ένα νέο NP-hard πρόβλημα. Στην συνέχεια της εργασίας δίνουμε μια παραλλαγή του αλγορίθμου που προτείνουμε για το πρόβλημα $MTMS_j$ που επιλύει και το πρόβλημα $MTMS_{ij}$. Τις προτάσεις που ορίζουν μηχανισμούς κλαδέματος για το πρόβλημα $MTMS_{ij}$ δεν τις παραθέτουμε μια και δεν είναι ο στόχος της παρούσας εργασίας.

4. Απαριθμητικός αλγόριθμος για το $MTMS_j$

Οπως είναι φανερό το πρόβλημα εντοπίζεται πλέον στον προσδιορισμό ενός συνόλου βελτίστων συνθέτων παρτίδων. Μια βέλτιστη διάταξη δίνεται τότε με την τοποθέτηση των παρτίδων κατά EDD.

4.1. Περιγραφή του αλγορίθμου

Μια από τις πιο διαδεδομένες διαδικασίες επίλυσης συνδυαστικών προβλημάτων είναι η ελεγχόμενη απαρίθμηση. Η βασική ιδέα είναι η εξαντλητική εξέταση ενός υποσυνόλου V του συνόλου των εφικτών λύσεων F γνωρίζοντας ότι στο σύνολο V υπάρχει τουλάχιστον μια βέλτιστη λύση. Βέβαια η διαδικασία της εξαντλητικής έρευνας του συνόλου V είναι ελεγχόμενη και γίνεται με τον διαδοχικό διαμερισμό του αρχικού συνόλου V σε υποσύνολα όπου εφαρμόζουμε την ίδια διαδικασία.

Ο έλεγχος και η μείωση του χώρου έρευνας επιτυγχάνεται με την θέσπιση φραγμάτων για την τιμή της αντικειμενικής συνάρτησης για όλες τις λύσεις που ανήκουν στον τρέχοντα χώρο έρευνας. Αν επιπλέον η λύση πρέπει να ικανοποιεί και συγκεκριμένους περιορισμούς τότε ο χώρος έρευνας μπορεί να μειωθεί ακόμα περισσότερο.

Ο παρακάτω αλγόριθμος που προτείνουμε είναι βασικά απαριθμητικός και κάνει χρήση κριτηρίων απόρριψης. Οπως είναι προφανές από την προηγούμενη γενική περιγραφή της διαδικασίας ελεγχόμενης απαρίθμησης κάθε φορά διαμερίζουμε το σύνολο V και στα υποσύνολα εφαρμόζουμε την ίδια διαδικασία. Η διαδικασία που εφαρμόζουμε για την επίλυση του προβλήματος χαρακτηρίζεται από τρία στοιχεία. Πρώτον το σύνολο των διατεταγμένων εργασιών S, δεύτερον το σύνολο των προς διάταξη εργασιών V και τρίτο και τελευταίο έναν περιορισμό Constr που μας λέει πως στο επόμενο βήμα δεν μπορούμε να τοποθετήσουμε εργασία από την ομάδα που ο περιορισμός δηλώνει.

Αρχικά δημιουργούμε τις απλές παρτίδες με τον τρόπο που αναφέρεται στο κεφ. 3.4. Θέτουμε το σύνολο των διατεταγμένων εργασιών $S=\emptyset$. Θέτουμε το σύνολο των προς διάταξη εργασιών

$$V=\{\left[J_{ib}^{Crit}\right],\quad 1\le b\le B,\ 1\le i\le N_b^{Crit}\}.$$

Θέτουμε το περιορισμό $Constr=-1$, δηλαδή στην αρχή δεν έχουμε κανέναν περιορισμό.

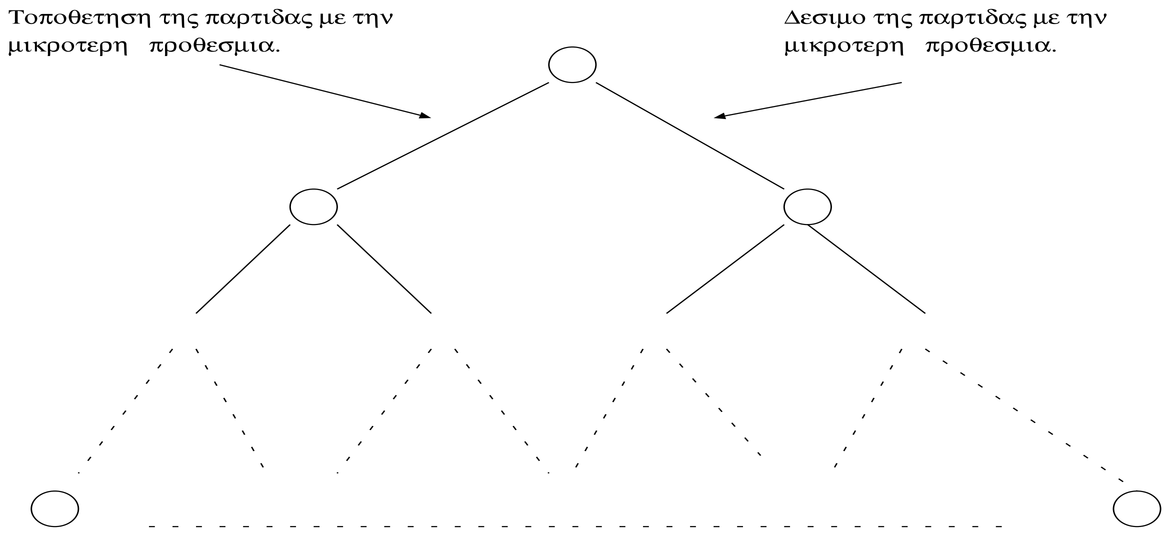
Διαλέγουμε την παρτίδα που έχει την ελάχιστη προθεσμία έστω την $\left[J_{1b_0}^{Crit}\right]$.† Η παρτίδα $\left[J_{1b_0}^{Crit}\right]$ σε κάποιο σύνολο βέλτιστων παρτίδων θα είναι ή μοναχική ή δεμένη.

- Αν είναι μοναχική τότε καμμία σύνθετη παρτίδα δεν μπορεί να έχει μικρότερη προθεσμία και συνεπώς η παρτίδα $\left[J_{1b_0}^{Crit}\right]$ θα πρέπει να τοποθετηθεί πρώτη. Συνεπώς θέτουμε το $S[1]=\left[J_{1b_0}^{Crit}\right]$. Επίσης θέτουμε $V=V-\{\left[J_{1b_0}^{Crit}\right]\}$. Για να είμαστε σίγουροι πως η παρτίδα $\left[J_{1b_0}^{Crit}\right]$ που μόλις τοποθετήθηκε θα παραμείνει μοναχική θέτουμε τον περιορισμό $Constr=b_0$ και ξανακαλούμε την διαδικασία της ελεγχόμενης απαρίθμησης. Αξίζει να παρατηρήσουμε πως το νέο πρόγραμμα είναι όμοιο με το αρχικό αλλά ξεκινά την χρονική στιγμή $S_{b_0}+p\left[J_{1b_0}^{Crit}\right]$ και ότι έχει $N^{Crit}-1$ εργασίες, δηλαδή κατά μία λιγότερες από το αρχικό.
- Αν η παρτίδα $\left[J_{1b_0}^{Crit}\right]$ είναι δεμένη τότε επειδή είναι πρώτη στην ομάδα b_0 δεν μπορεί παρά να είναι δεξιά δεμένη και συνεπώς θέτουμε $S=S$, $V=V-\{\left[J_{1b_0}^{Crit}\right]\}-\{\left[J_{2b_0}^{Crit}\right]\}+\{\left[J_{1b_0}^{Crit},J_{2b_0}^{Crit}\right]\}$, $Constr=Constr$ και ξανακαλούμε την διαδικασία της ελεγχόμενης απαρίθμησης. Και πάλι το νέο πρόβλημα είναι όμοιο με το αρχικό αλλά έχει $N^{Crit}-1$ εργασίες.

Φυσικά αν η παρτίδα με την μικρότερη προθεσμία ανήκει στην ομάδα που ο περιορισμός εξαιρεί τότε η παρτίδα υποχρεωτικά πρέπει να είναι δεμένη (αν αυτό είναι δυνατόν).

Αν αρχικά υποθέσουμε πως οι προθεσμίες των απλών και συνθέτων παρτίδων είναι ανά δύο άνισες τότε η λειτουργία του αλγορίθμου μπορεί να παρασταθεί με το δυαδικό δένδρο του σχήματος 4.1.1. Προφανώς μετά από N^{Crit} επίπεδα τα φύλλα του δυαδικού δένδρου που δημιουργούνται είναι διατάξεις εργασιών και μάλιστα αυτή με το ελάχιστο κόστος είναι η βέλτιστη λύση του $MTMS_j$. Συνεπώς η υπολογιστική πλοκή του προβλήματος είναι $O(2^{N^{Crit}})$.

† Προφανώς η παρτίδα που έχει ελάχιστη προθεσμία είναι η πρώτη παρτίδα κάποιας ομάδας αφού η προθεσμία κάθε παρτίδας είναι η προθεσμία της κρίσιμης εργασίας που ορίζει την παρτίδα και κατά τα γνωστά οι κρίσιμες εργασίες ορίζονται με βάση τον EDD.



Σχήμα 4.1.1 : Λειτουργία του αλγορίθμου.

Τα κριτήρια που χρησιμοποιούμε για την ελάττωση του χώρου έρευνας βασίζονται στα θεωρήματα 3.6.1 και 4.3.1. Το θεώρημα 3.6.1 το εκμεταλλευόμαστε ως εξής. Αν η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι χαρακτηρισμένη ως δεμένη το να αποφασίσουμε να την τοποθετήσουμε είναι περιττό και συνεπώς το αριστερό κλαδί του δένδρου δεν χρειάζεται να ερευνηθεί. Ας παρατηρήσουμε πως μερικά τέτοια κλαδέματα στο ξεκίνημα του αλγορίθμου μειώνουν κατά πολύ το πλήθος των διατάξεων που απαιτείται να ερευνηθούν. Αν για παράδειγμα στο δεύτερο επίπεδο καταφέρουμε και κόψουμε κάποιο από τα παιδιά τότε θεωρητικά γλυτώνουμε $2^N/8$ διατάξεις που σε διαφορετική περίπτωση θα έπρεπε να εξετάσουμε.

Πριν αναφέρουμε τον τρόπο με τον οποίο εκμεταλλευόμαστε το θεώρημα 4.3.1 κρίνουμε σκόπιμη την ακόλουθη σημαντική παρατήρηση. Σε κάθε βήμα του αλγορίθμου από κάθε πρόβλημα δημιουργούμε δύο υποπροβλήματα ίδια με το αρχικό αλλά με μια εργασία λιγότερη . Το σιγουρο λοιπόν είναι πως μετα από N^{Crit} βήματα το πρόβλημα θα έχει λυθεί. Το σημαντικό είναι πως ο αλγόριθμος που προτείναμε θα μπορούσε να δουλέψει και διατάσσοντας εργασίες από το τέλος προς την αρχή. Ετσι μπορούμε να διαλέξουμε την παρτίδα $\left[J^{Crit}_{N^{Crit}_{b_0}b_0}\right]$ που έχει την μεγαλύτερη προθεσμία και να αποφασίσουμε αν σε κάποιο σύνολο βελτίστων συνθέτων παρτίδων είναι δεμένη ή μοναχική. Αν μεν είναι μοναχική τότε υπάρχει βέλτιστο πρόγραμμα στο οποίο η παρτίδα $\left[J^{Crit}_{N^{Crit}_{b_0} b_0}\right]$ είναι τοποθετημένη στην τελευταία θέση. Ενω αν είναι δεμένη επειδή είναι τελευταία στην ομάδα b_0 δεν μπορεί παρά να είναι αριστερά δεμένη. Συνεπώς κατά τον ίδιο τρόπο ρυθμίζουμε τα νέα συνολα S και V καθώς και τον περιορισμό $Constr$ και καλούμε και πάλι την διαδικασία στα νέα υποπροβλήματα. Προφανώς το θεώρημα 3.6.1 μπορούμε να το εκμεταλλευτούμε με τον ίδιο τρόπο όπως και στην περίπτωση που τοποθετουμε εργασίες από την αρχή προς το τέλος.

Σύμφωνα με το θεώρημα 4.3.1, αν δύο ή περισσότερες παρτίδες με τις μεγαλύτερες προθεσμίες ανήκουν στην ίδια ομάδα τότε δεν χρειάζεται να ψάξουμε διατάξεις όπου αυτές οι παρτίδες δεν είναι δεμενες μεταξύ τους. Δηλαδή τα αριστερά κλαδιά του δένδρου έρευνας δεν είναι αναγκαίο να εξεταστούν. Από θεωρητικής πλευράς το θεώρημα 4.3.1 φυσικά ισχύει, όμως στην πράξη τέτοιες ακολουθίες εργασιών που να ανήκουν στην ίδια ομάδα είναι αρκετά σπάνιες. Για παράδειγμα, αν υποθέσουμε ότι έχουμε $B=10$ ομάδες, $N^{Crit}=100$, $N^{Crit}_b=10$, για κάθε $1\leq b\leq 10$ και ότι η κατανομή των προθεσμιών των παρτίδων στις ομάδες είναι ομοιόμορφη, τότε η πιθανότητα να ανήκουν στην ίδια ομάδα οι δύο παρτίδες με την μεγαλύτερη προθεσμία είναι 0.1. Πολύ περισσότερο η πιθανότητα να ανήκουν στην ίδια ομάδα οι τρεις παρτίδες με την μεγαλύτερη προθεσμία είναι ελάχιστη και ίση με 0.01. Ετσι στην ουσία η αξία του θεωρήματος 4.3.1 είναι κυρίως θεωρητική.

Ο τρίτος και τελευταίος μηχανισμός για την ελάττωση της περιοχής έρευνας είναι ο μηχανισμός φραγμάτων και η λειτουργία του βασίζεται στο θεώρημα 4.3.2. Ο κλασικός μηχανισμός φραγμάτων με την ύπαρξη κάτω και άνω φράγματος και απόρριψη της υποπεριοχής έρευνας για την οποία το κάτω φράγμα είναι μεγαλύτερο του πάνω φράγματος κάποιας άλλης περιοχής δεν εφαρμόζεται. Ο κύριος λόγος είναι η φύση του προβλήματος όπου μικρές διακυμάνσεις από την βέλτιστη λύση προκαλούν μικρές μεταβολές στην τιμή της αντικειμενικής συνάρτησης. Συνεπώς πολύ σπάνια συμβαίνει κάποιο κάτω φράγμα υποπεριοχής έρευνας να έχει τιμή μεγαλύτερη από κάποιο άνω φράγμα άλλης υποπεριοχής. Κατά συνέπεια ο κλασικός μηχανισμός φραγμάτων δεν μπορεί να αποδώσει. Ο λόγος για τον οποίο μικρές διακυμάνσεις από την βέλτιστη λύση προκαλούν μικρές μεταβολές στην τιμή της αντικειμενικής συνάρτησης προέρχεται βασικά από το γεγονός πως έχουμε ένα κριτήριο ελαχίστου-μεγίστου. Η τιμή της αντικειμενικής συνάρτησης καθορίζεται από μια και μόνο εργασία και όχι από το σύνολο των εργασιών. Αντίθετα, αν σκεφτούμε το κριτήριο $\sum_{i \in J} C_i$ τότε παρατηρούμε πως κάποια απόκλιση από το βέλτιστο πρόγραμμα στην αρχή του, επιφέρει μεγάλη μεταβολή στην τιμή της αντικειμενικής συνάρτησης καθώς το λάθος αθροίζεται ξανά και ξανά για τις εργασίες που ακολουθούν. Συνεπώς, επειδή οι τιμές της αντικειμενικής συνάρτησης για τα διάφορετικά προγράμματα μαζεύονται όλες σε κάποια πολύ στενή περιοχή, είναι δύσκολο να απορρίψουμε τμήματα του χώρου έρευνας ακόμα και αν έχουμε πάρα πολύ σφικτά κάτω και πάνω φράγματα. Τέτοια όρια δεν είναι γνωστά στην βιβλιογραφία για το $MTMS_j$ αλλά ούτε και εμείς κατορθώσαμε να βρούμε. Αντίθετα για το πρόβλημα $MTMS_{ij}$ υπάρχουν καλύτερα πάνω και κάτω φράγματα ιδίως αν ο αριθμός των εργασιών ανά κατηγορία είναι μικρός και στην καλύτερη περίπτωση ίσος με 1. Από την άλλη πλευρά το γεγονός πως οι τιμές της αντικειμενικής συνάρτησης ανήκουν σε μια στενή περιοχή μας οδηγεί στο συμπέρασμα πως μπορούμε να έχουμε πολύ καλές προσεγγίσεις της βέλτιστης λύσης. Συνεπώς το πρόβλημα είναι από αυτά που επιδέχονται ικανοποιητική λύση με χρήση ευρηματικών μεθόδων. Αυτό θα γίνει φανερό και από τις επιδόσεις του προσεγγιστικού αλγορίθμου που παρουσιάζονται στο επόμενο κεφάλαιο.

Παρά την αδυναμία εφαρμογής των κλασικών μεθόδων φραγμάτων, με το θεώρημα 4.3.2 εισάγουμε την χρήση ενός τέτοιου μηχανισμού με τελείως διαφορετικό τρόπο. Ετσι την εκθετική πλοκή του προβλήματος ως προς τον αριθμό των ομάδων σε ολόκληρο τον χρόνο προγραμματισμού την ανάγουμε σε γραμμική σε υποδιαστήματα του χρόνου προγραμματισμού. Το μέγεθος των υποδιαστημάτων καθώς και η αλληλεπικάλυψή τους (για κάθε ομάδα ορίζεται και διαφορετικό υποδιάστημα που δυναμικά αναπροσαρμόζεται) είναι καθοριστικό για τις επιδόσεις του αλγορίθμου. Πάντως σε κάθε περίπτωση η αύξηση του μεγέθους των προβλημάτων που επιλύονται κάνοντας χρήση αυτού του μηχανισμού είναι εντυπωσιακή. Προβλήματα που χρειάζονταν ως και δέκα ώρες για την επίλυσή τους επιλύονται σε χρόνο ενός λεπτού.

Το θεώρημα 4.3.2 περιγράφει την λειτουργία του μηχανισμού μόνο για την περίπτωση που ο αλγόριθμος τοποθετεί εργασίες από την αρχή προς το τέλος. Μπορεί βέβαια να τροποποιηθεί ώστε να εφαρμόζεται και στην περίπτωση που δημιουργούμε το πρόγραμμα από το τέλος προς την αρχή. Φυσικά σε αυτή την περίπτωση ο μηχανισμός δεν θα είναι τόσο ισχυρός. Αυτό οφείλεται στην άγνοια των χρονικών στιγμών έναρξης των εργασιών (μέχρι την ολοκλήρωση του προγράμματος) που οδηγεί σε εκτίμησή τους με βάση πάνω φράγματα. Ομως και πάλι θα πρέπει να βελτιώνει σημαντικά τις επιδόσεις του αλγορίθμου. Η δραματική αύξηση των επιδόσεων του αλγορίθμου με την χρήση του μηχανισμού φραγμάτων οφείλεται στο γεγονός πως ο μηχανισμός αυτός είναι δραστικός κυρίως στην αρχή του προγράμματος. Κατά συνέπεια τα γρήγορα κλαδέματα στην αρχή περιορίζουν δραστικά τον χώρο έρευνας. Η εντυπωσιακή λειτουργία του μηχανισμού ισχυροποιείται και από το γεγονός πως στην αξιολόγηση του αλγορίθμου το κάτω όριο είναι αρκετά χαλαρό και υπολογίζεται μια και μόνη φορά χωρίς δυναμικό επαναπροσδιορισμό. Παρ' όλα αυτά οι επιδόσεις του αλγορίθμου αυξάνονται εντυπωσιακά.

Ο αλγόριθμος που δημιουργεί το βελτιστο πρόγραμμα από την αρχή προς το τέλος κατά μέσο όρο είναι σημαντικά ταχύτερος και πιο εύκολος ως προς την υλοποίησή του από τον αλγόριθμο που δημιουργεί το βέλτιστο πρόγραμμα από το τέλος πρός την αρχή. Σύνεπως θα μπορούσε κανείς να υποστηρίξει πως ο δεύτερος αλγόριθμος είναι ουσιαστικά άχρηστος. Τα πράγματα όμως δεν είναι έτσι για δύο βασικούς λόγους. Πολλές φορές υπάρχουν προβλήματα στα οποία οι τιμές των προθεσμιών στις παρτίδες με μεγάλες προθεσμίες βρίσκονται πολύ κοντά μεταξύ τους. Αυτά τα προβλήματα είναι πιο εύκολο να τα λύσουμε δημιουργώντας το πρόγραμμα από το τέλος προς την αρχή. Το κριτήριο 3.6.1 θα κλαδέψει πολύ γρήγορα μεγάλες περιοχές του δέντρου έρευνας και τελικά θα οδηγηθούμε σε βέλτιστη λύση. Επίσης πολλές φορές θέλουμε να βρούμε λύση σε δικριτηριακό πρόβλημα. Για παράδειγμα έστω ότι αναζητούμε εφικτή διάταξη $(C_i < d_i)$ που να ελαχιστοποιεί τον μέσο χρόνο αποπεράτωσης $\sum_{i \in J} C_i$. Σε ένα τέτοιο πρόβλημα είναι άστοχο αν όχι αδύνατο να αρχίσει η κατασκευή του βελτίστου προγράμματος από την αρχή πρός το τέλος. Ο λόγος είναι προφανής αφού σε ένα εφικτό πρόγραμμα οι υποψήφιες εργασίες για την πρώτη θέση είναι πολύ περισσότερες (όλες) από τις υποψήφιες εργασίες για την τελευταία θέση. Σύνεπως ο πρώτος αλγόριθμος δέν μπορεί να εφαρμοστεί ενώ ο δεύτερος αν και δεν είναι τόσο γρήγορος θα μπορούσε να αποδώσει.

Η προηγούμενη περιγραφή του αλγοριθμού είναι γενική και παραλείπει ορισμένα θέματα τα οποία είναι καθοριστικά. Για παράδειγμα αναφέρει ότι σε κάθε βήμα από κάθε πρόβλημα δημιουργούνται το πολύ δύο νέα υποπροβλήματα. Ομως παραλείπει το γεγονός πως μπορεί η παρτίδα που έχει την μικρότερη ή μεγαλύτερη προθεσμία να μην είναι μοναδικά ορισμένη. Αν λόγου χάρη τρεις εργασίες έχουν την ίδια ελάχιστη προθεσμία τότε τα νέα υποπροβλήματα δεν θα πρέπει να είναι δύο αλλά έξι. Δηλαδή, από δύο για κάθε εργασία που έχει την ίδια ελάχιστη προθεσμία. Στο επόμενο κεφάλαιο αναπτύσσεται ένας απλός κανόνας διαμερισμού, η εφαρμογή του οποίου εγγυάται την δυαδικότητα του δένδρου καθώς και την μη απόρριψη τουλάχιστον μιας βέλτιστης λύσης. Επίσης εύκολα θα μπορούσαμε να τροποποιήσουμε τον αλγόριθμο ώστε να δουλεύει και για το πρόβλημα $MTMS_{ij}$. Η δυνατότητα αυτή αναπτύσσεται στο κεφ. 8.2.

Για λόγους εποπτείας, την λειτουργία του αλγορίθμου μπορούμε να την παραστήσουμε με ένα δυαδικό δένδρο κάθε κόμβος του οποίου αποτελεί περίπτωση του προβλήματος $MTMS_j$. Το δένδρο έχει βάθος N^{Crit}. Η παράσταση του δένδρου στο σχήμα 4.1.2 χρησιμοποιεί ως ταυτότητα για κάθε πρόβλημα-κόμβο ένα διατεταγμένο ζεύγος. Το πρώτο στοιχείο είναι το επίπεδο στο οποίο ανήκει ενώ το δεύτερο η σειρά του κόμβου από τα αριστερά προς τα δεξιά μέσα στο επίπεδο. Κάθε επίπεδο i έχει συνολίκα 2^i κόμβους-προβλήματα $MTMS_j$. Ας παρατηρήσουμε πως για την αναπαραγωγή ενός προγράμματος που αντιστοιχεί σε κάποιο κόμβο-φύλλο είναι απαραίτητη η γνώση μόνο του δεύτερου στοιχείου του διατεταγμένου ζεύγους. Πράγματι, η γνώση του δεύτερου στοιχείου του διατεταγμένου ζεύγους οδηγεί σε γνώση του μονοπατιού από την ρίζα του δέντρου μέχρι τον κόμβο-φύλλο και κατά συνέπεια και της ακολουθίας των τοποθετήσεων των εργασιών.

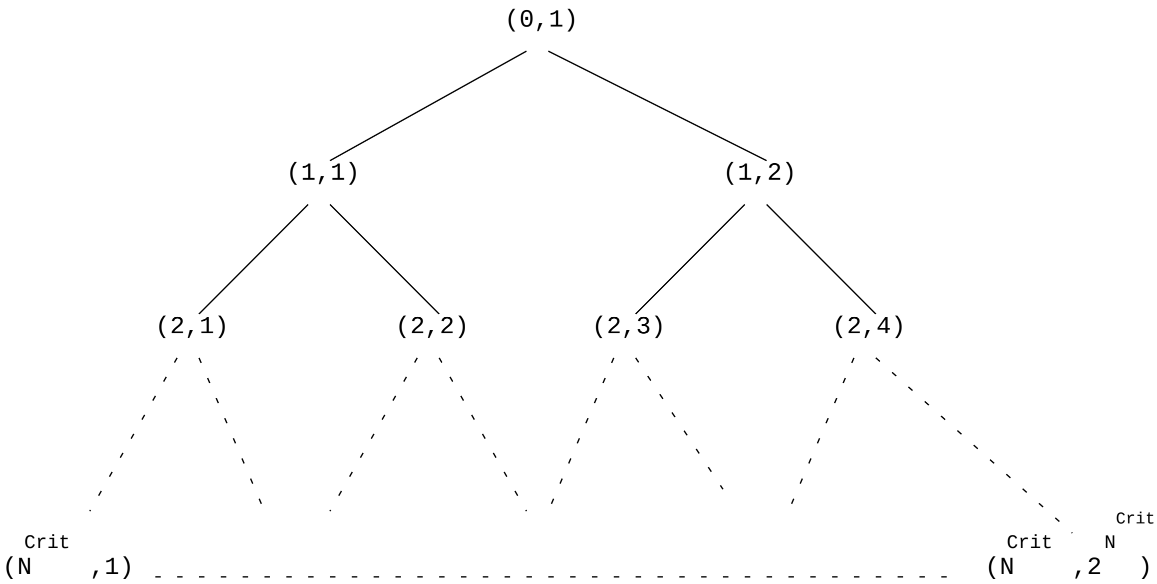
Συνεπώς δεν είναι απαραίτητο κατά την εξέλιξη του αλγορίθμου να κρατάμε την μέχρι τώρα διάταξη αλλά ένα μόνο αριθμό. Επίσης κάθε πρόβλημα (i,j) είναι $MTMS_j$ πρόβλημα με συνολο εργασιών προς διάταξη το $J^{(i,j)}$. Για το συνολο των προς διάταξη εργασιών έχουμε :

$$|J^{(i,j)}| = |J^{(0,1)}| - i$$

Επίσης αν με $S^{(i,j)}$ συμβολίσουμε τις προγραμματισμένες εργασίες για το πρόβλημα (i,j), έχουμε ότι $0 \le |S^{(i,j)}| \le i$. Ακόμα ισχύει :

$$|S^{(i,j)}| = \begin{cases} |S^{(i-1,j/2)}| & \text{αν } j=2k \\ |S^{(i-1,\lfloor j/2 \rfloor+1)}|+1 & \text{αλλιώς.} \end{cases}$$

Φυσικά αν για δύο πρόβληματα (i,j) και (i',j') έχουμε $S^{(i,j)}=S^{(i',j')}$ και έχουν τον ίδιο περιορισμό τότε το δενδρο πρέπει να συνεχίσει να μεγαλώνει μόνο κάτω



Σχήμα 4.1.2 : Σχηματική παράσταση της λειτουργίας του αλγορίθμου.

από το πρόβλημα που έχει το μικρότερο μέχρι στιγμής $T_{\max}$. Φυσικά αυτό ισχύει μόνο όταν η λειτουργία του αλγορίθμου είναι από το τέλος προς την αρχή. Στην αντίθετη περίπτωση δεν ισχύει παρά μόνο εάν το πρόβλημα με το μικρότερο $T_{\max}$ έχει και μικρότερο $C_{\max}$.

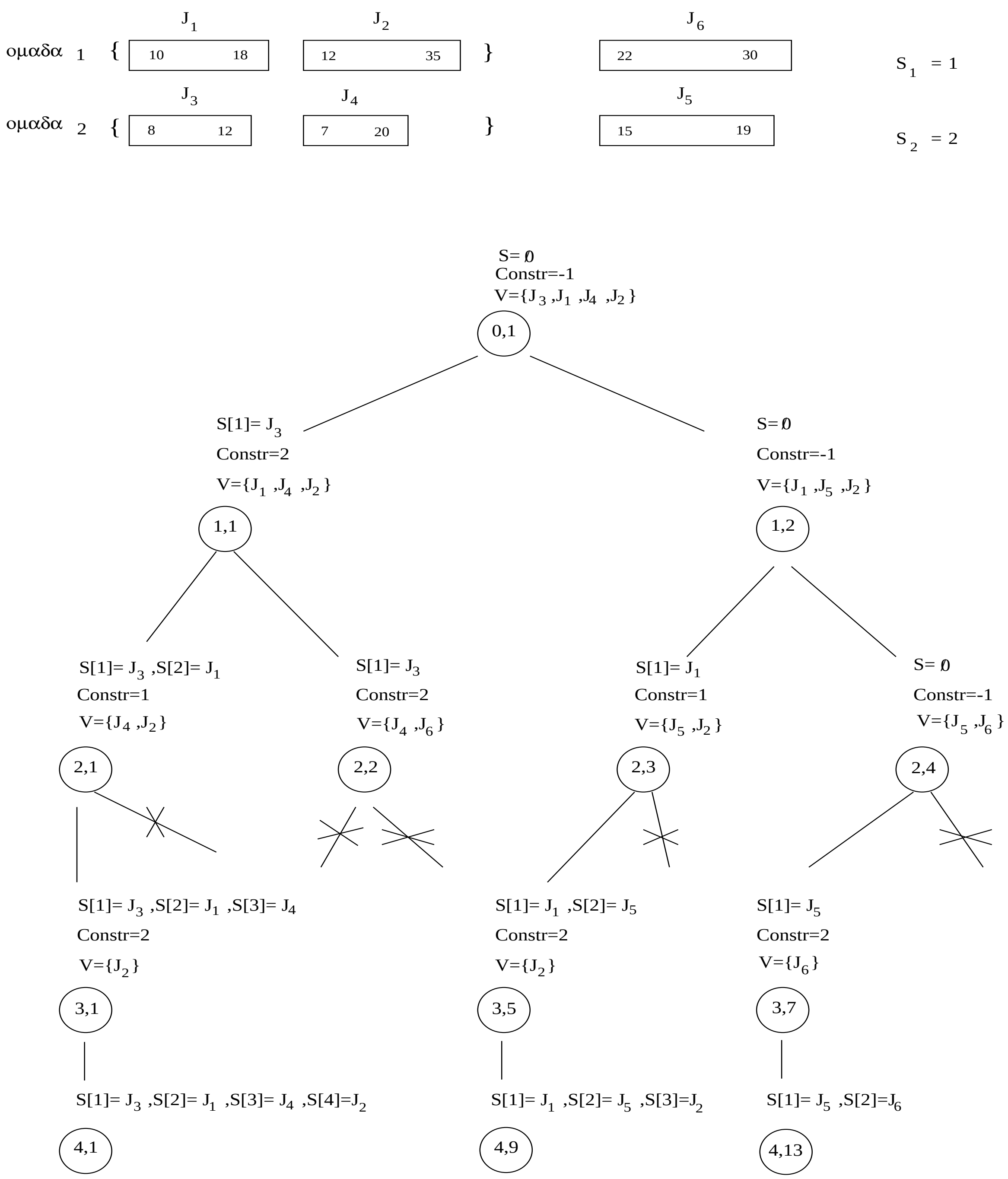
Τέλος παρατηρούμε πως το σύνολο $J^{(i,j)}$ μπορεί να περιγραφεί με την τήρηση B μόνο δεικτών. Σε κάθε ομάδα κρατάμε τον πιο μικρό δείκτη από τις εργασίες που είναι ακόμα αδιάτακτες. Ακολουθεί ένα παράδειγμα του αλγορίθμου σε πρόβλημα τεσσάρων εργασιών που κατανέμονται σε δύο ομάδες.

Παράδειγμα 4.1.1 Εστω ότι έχουμε τέσσερις εργασίες που ανήκουν σε δυο ομάδες. Ο επόμενος πίνακας δίνει τους χρόνους επεξεργασίας και τις προθεσμίες των εργασιών.

	J_{11}	J_{21}	J_{12}	J_{22}
p_{ib}	10	12	8	7
d_{ib}	18	35	12	20

Αρχικά παρατηρούμε πως οι εργασίες J_{ib}, $1 \leq b \leq 2$, $1 \leq i \leq 2$ είναι κρίσιμες εργασίες. Επίσης οι σύνθετες παρτίδες $\left[J_{11}, J_{21}\right]$ και $\left[J_{12}, J_{22}\right]$ έχουν χρόνους κατεργασίας 22 και 15 αντίστοιχα, ενώ οι προθεσμίες τους είναι 30 και 19.

Το σχήμα 4.1.3 δείχνει τον τρόπο λειτουργίας του αλγορίθμου. Στο επίπεδο 4 έχουμε μόνο 3 από τα 16 δυνατά προγράμματα. Η βέλτιστη λύση δίνεται από το φύλλο (4,9) και έχει $T_{\max}$=9.



Σχήμα 4.1.3 : Παράδειγμα λειτουργίας του αλγορίθμου.

4.2. Κανόνας διακλάδωσης

Ο τρόπος με τον οποίο δημιουργούμε τα υποπροβλήματα από κάθε κόμβο-πρόβλημα είναι απλός μεν αλλά έχει κενά. Συγκεκριμένα σιωπηλά υποθέτει πως η παρτίδα που έχει την ελάχιστη (μέγιστη) προθεσμία είναι μονοσήμαντα ορισμένη. Αν αυτή η υπόθεση δεν ισχύει τότε ο τρόπος με τον οποίο δημιουργούμε τα υποπροβλήματα δεν είναι ρητά καθορισμένος. Ο παρακάτω κανόνας διακλάδωσης περιγράφει ένα μηχανισμό δημιουργίας υποπροβλημάτων που δεν παρουσιάζει το προηγούμενο πρόβλημα.

Κανόνας διακλάδωσης:

Από κάθε κόμβο-πρόβλημα (i,j) δημιουργούμε δύο παιδιά-υποπροβλήματα το αριστερό παιδί $(i+1,2j-1)$ και το δεξιό παιδί $(i+1,2j)$.

Στο αριστερό παιδί-υποπρόβλημα η παρτίδα με την μεγαλύτερη (μικρότερη) προθεσμία $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$ διατάσσεται στην τρέχουσα θέση του υπο κατασκευή προγράμματος (αν υπάρχουν περισσότερες από μια, διαλέγουμε αυτήν με τον μικρότερο δείκτη) και ο περιορισμός για το υποπρόβλημα είναι η επόμενη τοποθέτηση να μην γίνει από την ομάδα στην οποία ανήκει η παρτίδα που μόλις τοποθετήθηκε. Αν η παρτίδα $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$ έχει προθεσμία ίση με την προθεσμία της παρτίδας που προηγείται αυτής στο υπό κατασκευή πρόγραμμα τότε ο περιορισμός απενεργοποιείται†. Αυτό σημαίνει πως στο επόμενο βήμα μπορούμε να τοποθετήσουμε εργασία από κάθε ομάδα.

Στο δεξιό παιδί-υποπρόβλημα αποφασίζουμε το δέσιμο της παρτίδας με την μεγαλύτερη (μικρότερη) προθεσμία (αν υπάρχουν περισσότερες από μια τότε διαλέγουμε αυτήν με τον μικρότερο δείκτη).

Θεώρημα 4.2.1. Ο κανόνας διακλάδωσης που μόλις περιγράφηκε διαμερίζει τον χώρο έρευνας.

---

† Ο λόγος για τον οποίο απενεργοποιούμε τον περιορισμό είναι προφανής αφού με ανταλλαγή των εργασιών που έχουν την ίδια προθεσμία έχουμε μερικό πρόγραμμα με το ίδιο $T_{\max}$, το ίδιο ή και μικρότερο $C_{\max}$ και διαφορετικό περιορισμό.

Απόδειξη: Στην περίπτωση που η παρτίδα με την ελάχιστη (μέγιστη) προθεσμία είναι μονοσήμαντα ορισμένη προφανώς ο κανόνας διακλάδωσης λειτουργεί σωστά. Ας υποθέσουμε λοιπόν ότι σε κάποιο βήμα έχουμε k παρτίδες με $d_i = d_{\max}$, $1 \leq i \leq k$. Τότε ένα απλό σχήμα απαρίθμησης που σε κάθε βήμα αποφασίζει αν θα τοποθετήσει ή θα δέσει κάθε μια από τις παρτίδες που έχουν την ίδια προθεσμία, μετά από k επίπεδα θα έχει τοποθετήσει ή δέσει όλες τις k παρτίδες. Ας παρατηρήσουμε πως κάθε κόμβος-πρόβλημα αυτού του σχήματος χαρακτηρίζεται από τρία στοιχεία. Τις διατεταγμένες παρτίδες, τις δεμένες παρτίδες καθώς και από τον περιορισμό. Επειδή όλες οι παρτίδες έχουν την ίδια προθεσμία η διάταξη των παρτίδων και ο περιορισμός δεν χρειάζονται για τον χαρακτηρισμό ενός κόμβου-προβλήμα. Αυτό γιατί αλλάζοντας την θέση δυο παρτίδων με την ίδια προθεσμία δημιουργούμε μια νέα διάταξη με την ίδια μέγιστη καθυστέρηση και διαφορετικό περιορισμό (μια και δεν μπορεί δύο παρτίδες από την ίδια ομάδα να έχουν την ίδια προθεσμία). Συνεπως, το συμπληρωματικό βέλτιστο πρόγραμμα μπορεί να ξεκινά με παρτίδα από οποιαδήποτε ομάδα μια και μπορεί να κολλήσει εναλλακτικά σε οποιαδήποτε από τις δύο ή και περισσότερες διατάξεις που μπορούμε να δημιουργήσουμε. Συνεπώς, κάθε κόμβος-πρόβλημα έχει ώς ταυτότητα το σύνολο των δεμένων παρτίδων και μόνο. Οι δυνατές διαφορετικές τιμές που μπορεί να πάρει είναι 2^k, όσες δηλαδη και ο πληθικός αριθμός του δυναμοσυνόλου των παρτίδων με την ίδια προθεσμία $d_{\max}$. Και συνεπώς από τα $2^k k!$ τελικά προβλήματα † του απλού απαριθμητικού σχήματος μόνο 2^k είναι διαφορετικά μεταξύ τους. Τα υπόλοιπα είναι επαναλήψεις των πρώτων.

Από την άλλη πλευρά ο κανόνας διακλάδωσης μετά από k επίπεδα δημιουργεί 2^k υποπροβλήματα ορίζοντας 2^k διαφορετικά σύνολα ‡ δεμένων εργασιών και συνεπώς εξαντλεί τον χώρο έρευνας χωρίς να εξετάζει κάποιο πρόβλημα περισσότερες από μια φορές.

---

† Στο απλό σχήμα απαρίθμησης λύσεων στο πρώτο επίπεδο έχουμε την δημιουργία $2k$ υποπροβλημάτων. Στο επόμενο επίπεδο από κάθε υποπρόβλημα έχουμε $2(k-1)$ υποπροβλήματα. Τελικά ο συνολικός αριθμός υποπροβλημάτων μετά από k επίπεδα είναι ίσος με $2^k k!$.

‡ Τα 2^k υποπροβλήματα είναι ξένα ανά δύο αφού ανά δύο έχουν κάποιο κοινό πρόγονο και συνεπώς τουλάχιστον μια παρτίδα του ενός είναι δεμένη ενώ στο άλλο οχι.

4.3. Μηχανισμοί ελάττωσης του χώρου έρευνας

Τον προηγούμενο αλγόριθμο μπορούμε να τον βελτιώσουμε με μηχανισμούς που να ελέγχουν την αναγκαιότητα εξέτασης μιας υποπεριοχής. Η περιγραφή της λειτουργίας αυτών των μηχανισμών έχει ήδη γίνει στο προηγούμενο κεφάλαιο. Εδώ δίνουμε τις αποδείξεις των προτάσεων που αιτιολογούν την ορθή λειτουργία αυτών των μηχανισμών. Οπως ήδη έχει περιγραφεί τέσσερις είναι οι μηχανισμοί ελέγχου του διαμερισμού του αρχικού χώρου. Ο πρώτος μηχανισμός βασίζεται στην αρχή του δυναμικού προγραμματισμού, που λέει ότι μερικώς βέλτιστα προγράμματα ολοκληρώνονται με βέλτιστα συμπληρώματα. Ο δεύτερος μηχανισμός βασίζεται στο θεώρημα 4.3.1 που ακολουθεί. Ο τρίτος μηχανισμός βασιζεται στο θεώρημα 3.6.1 και τέλος ο τέταρτος μηχανισμός βασίζεται στο θεώρημα 4.3.2. που ακολουθεί. Οι δύο πρώτοι μηχανισμοί είναι χωρίς ιδιαίτερη πρακτική σημασία. Ο τρίτος δουλεύει αρκετά καλά αν οι χρόνοι εξάρμωσης ανήκουν σε σχετικά ευρύ πεδίο τιμών. Ο τέταρτος και τελευταίος μηχανισμός δουλεύει ικανοποιητικά σε κάθε περίπτωση αν όμως οι χρόνοι εξάρμωσης είναι μικροί ή (και) η μέγιστη καθυστέρηση μεγάλη τότε οι επιδόσεις του είναι εξαιρετικές.

Θεώρημα 4.3.1. Εστω ότι λύνουμε ένα πρόβλημα $MTMS_j$ με τον αλγόριθμο διαμερισμού και φραγής που περιγράφεται στο κεφ. 4.1. Υποθέτουμε ότι

(α) ο αλγόριθμος δημιουργεί το βέλτιστο πρόγραμμα τοποθετώντας εργασίες από το τέλος προς την αρχή και ότι ισχύει

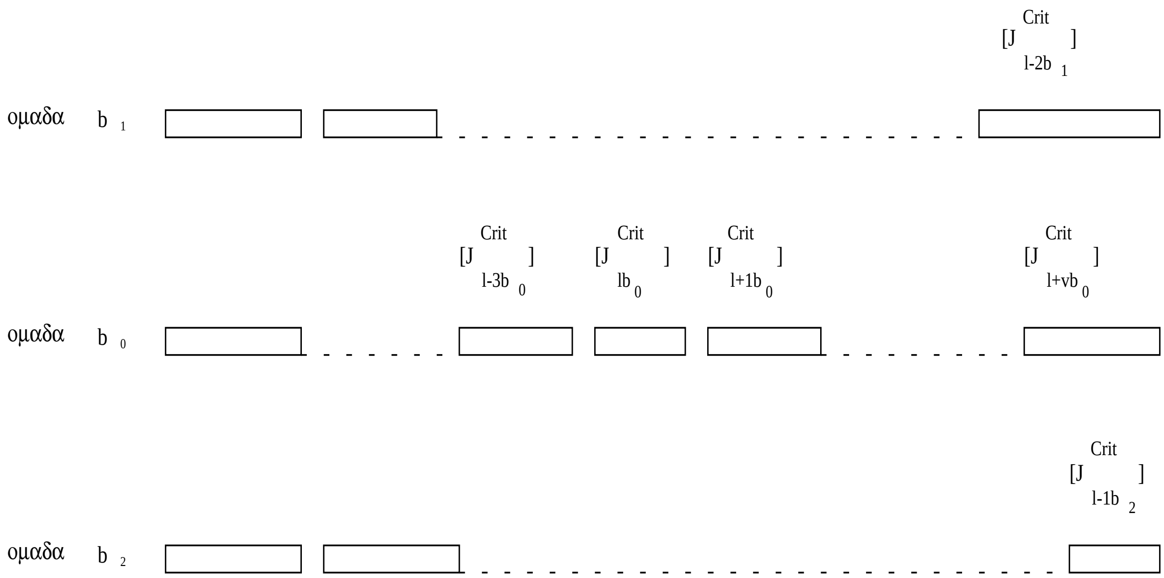
$$p\left[J_{ib}^{Crit}\right] > S_b \qquad \text{, για κάθε } 1 \le b \le B,\ 1 \le i \le N_b^{Crit}.$$

(β) οι απλές παρτίδες που έχουν τις $v+1$ μεγαλύτερες προθεσμίες ανήκουν στην ομάδα b_0

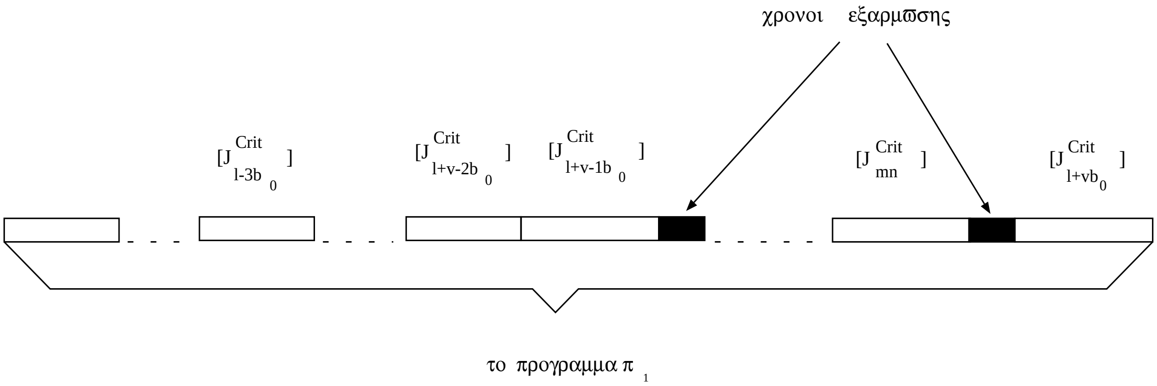
τότε υπάρχει βέλτιστο πρόγραμμα στο οποίο οι τελευταίες v τουλάχιστον παρτίδες αποτελούν βέλτιστη σύνθετη παρτίδα.

Απόδειξη: Για τις προθεσμίες των παρτίδων του σχήματος 4.3.1 με βάση την υπόθεση (β) έχουμε ότι:

$$d\left[J_{1k}\right] \le d\left[J_{ib}\right] \le \cdots \le d\left[J_{(l-3)b_0}\right] \le d\left[J_{(l-2)b_1}\right] \le$$
$$d\left[J_{(l-1)b_2}\right] < d\left[J_{lb_0}\right] < d\left[J_{(l+1)b_0}\right] < \cdots < d\left[J_{(l+v)b_0}\right]$$



Σχήμα 4.3.1 : Διάταξη των παρτίδων με βάση τον EDD.



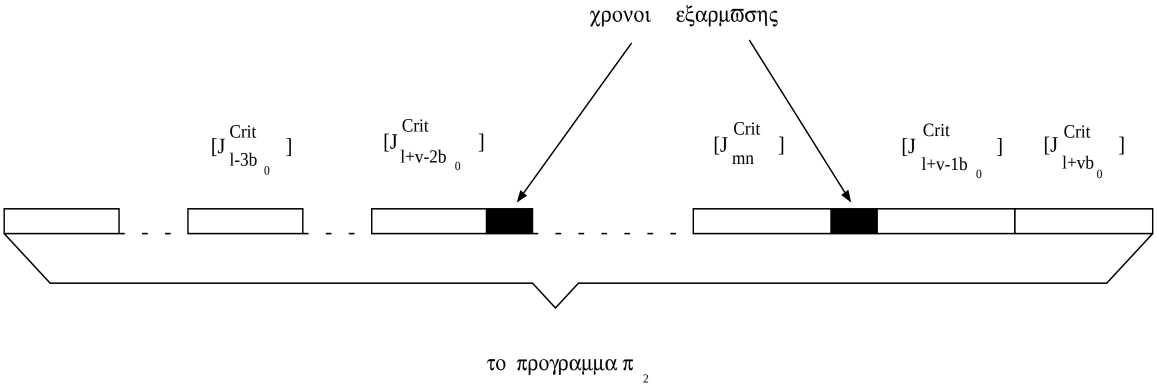
Σχήμα 4.3.2 : Το πρόγραμμα π_1.

Εδώ με $\left[J_{ib}\right]$ συμβολίζουμε την παρτίδα που βρίσκεται στην θέση i αν διατάξουμε τις απλές παρτίδες με βάση τον EDD και όχι την παρτίδα τάξεως i στην ομάδα b. Γι’ αυτό λείπει ο εκθέτης $Crit$. Προφανώς λόγω της υπόθεσης (β) έχουμε ότι $\left[J_{mb_0}\right] \equiv \left[J_{mb_0}^{Crit}\right], m=l,...,l+v$. Αρχικά θα δείξουμε ότι υπάρχει βέλτιστο πρόγραμμα στο οποίο οι παρτίδες $\left[J_{(l+v-1)b_0}\right]$ και $d\left[J_{(l+v)b_0}\right]$ αποτελούν βέλτιστη σύνθετη παρτίδα.

Αν υποθέσουμε ότι ο ισχυρισμός δεν ισχύει τότε κάθε βέλτιστο πρόγραμμα (στην περιοχή έρευνας με βάση τον EDD) έχει την παρτίδα $d\left[J_{(l+v)b_0}\right]$ μοναχική και συνεπώς τελευταία. Εστω ότι το π_1 είναι ένα τέτοιο πρόγραμμα (σχήμα 4.3.2). Εστω $\left[J_{mn}\right]$ η παρτίδα που βρίσκεται ακριβώς αριστερά της παρτίδας $\left[J_{(l+v)b_0}\right]$ στο πρόγραμμα π_1. Από την υπόθεση (α) έχουμε :

$$d\left[J_{mn}\right] < d\left[J_{(l+v-1)b_0}\right].$$

Ας θεωρήσουμε το πρόγραμμα π_2 που προκύπτει από το π_1 με την μεταφορά της παρτίδας $\left[J_{(l+v-1)b_0}\right]$ ακριβώς αριστερά της $\left[J_{(l+v)b_0}\right]$ (σχήμα 4.3.3).



Σχήμα 4.3.3 : Το πρόγραμμα π_2.

Για το πρόγραμμα π_2 έχουμε :

$$T^2\left[J_{ib}\right] \le T^1\left[J_{ib}\right] \quad \text{για κάθε παρτίδα εκτός της } \left[J_{(l+v-1)b_0}\right],$$

όπου με $T^{\kappa}\left[J_{ib}\right]$ συμβολίζουμε την καθυστέρηση της παρτίδας $\left[J_{ib}\right]$ στο πρόγραμμα π_k.

Επίσης για τους χρόνους περάτωσης των προγραμμάτων π_1 και π_2 έχουμε †

† Η παρτίδα $\left[J_{(l+v-1)b_0}\right]$ στο πρόγραμμα π_1 είναι αριστερά δεμένη (αν δεν ήταν αριστερά δεμένη θα έπρεπε να βρίσκεται δεξιότερα της $\left[J_{mn}\right]$ αφού έχει μεγαλύτερη προθεσμία). Συνεπώς τα προγράμματα π_1 και

$$C^2_{\max} = C^1_{\max}.$$

Με $C^k\left[J_{ib}\right]$ συμβολίζουμε τον χρόνο περάτωσης της παρτίδας $\left[J_{ib}\right]$ στο πρόγραμμα π_k. Ισχύει

$$C^1\left[J_{mn}\right] = C^1_{\max} - S_{b_0} - p\left[J_{(l+v)b_0}\right] = C^2_{\max} - p\left[J_{(l+v)b_0}\right] - S_{b_0} =$$
$$C^2\left[J_{(l+v-1)b_0}\right] - S_{b_0}$$

ή ισοδύναμα

$$C^2\left[J_{(l+v-1)b_0}\right] = C^1\left[J_{mn}\right] + S_{b_0}.$$

Ακόμα ισχύει $d\left[J_{(l+v-1)b_0}\right] > p\left[J_{(l+v-1)b_0}\right] + d\left[J_{(l+v-2)b_0}\right] \overset{(\alpha)}{>} d\left[J_{mn}\right] + S_{b_0}$ δηλαδή

$$d\left[J_{(l+v-1)b_0}\right] > d\left[J_{mn}\right] + S_{b_0}$$

και τελικά έχουμε:

$$T^2\left[J_{(l+v-1)b_0}\right] = C^2\left[J_{(l+v-1)b_0}\right] - d\left[J_{(l+v-1)b_0}\right] =$$
$$C^1\left[J_{mn}\right] + S_{b_0} - d\left[J_{(l+v-1)b_0}\right] < C^1\left[J_{mn}\right] + S_{b_0} - d\left[J_{mn}\right] - S_{b_0} = T^1\left[J_{mn}\right].$$

Συνεπώς το πρόγραμμα π_2 είναι επίσης βέλτιστο, πράγμα που αποτελεί άτοπο. Αυτό μας οδηγεί στο συμπέρασμα πως υπάρχει βέλτιστο πρόγραμμα στο οποίο οι παρτίδες $\left[J_{(l+v-1)b_0}\right]$ και $\left[J_{(l+v)b_0}\right]$ αποτελούν σύνθετη παρτίδα. Για τον ίδιο ακριβώς λόγο η σύνθετη παρτίδα $\left[J_{(l+v-1)b_0}, J_{(l+v)b_0}\right]$ μαζί με την παρτίδα $\left[J_{(l+v-2)b_0}\right]$ αποτελούν και πάλι βέλτιστη σύνθετη παρτίδα και τελικά βέλτιστη σύνθετη παρτίδα είναι και η $\left[J_{(l+1)b_0}, J_{(l+v)b_0}\right]$.□

Η απόδειξη δεν μπορεί να επεκταθεί και στην παρτίδα $\left[J_{lb_0}\right]$ γιατί δεν ξέρουμε αν ισχύει η σχέση $d\left[J_{lb_0}\right] > d\left[J_{l-1b_2}\right] + S_{b_0}$. Ο έλεγχος αυτός μπορεί να γίνει κατά την εκτέλεση του αλγορίθμου. Οπότε αν η σχέση ισχύει τότε η παρτίδα $\left[J_{lb_0}\right]$ μαζί με την παρτίδα $\left[J_{l+1b_0}, J_{l+vb_0}\right]$ αποτελούν την τελική βέλτιστη σύνθετη παρτίδα $\left[J_{lb_0}, J_{l+vb_0}\right]$.

π_2 έχουν ακριβώς τους ίδιους χρόνους εξάρμωσης.

Επίσης αν θεωρήσουμε πως η υπόθεση $p\left[J_{ib}^{Crit}\right] > S_b$, $1 \le b \le B$ και $1 \le i \le N_b^{Crit}$ δεν ισχύει τότε και πάλι μπορούμε να εφαρμόσουμε το θεώρημα 4.3.1 ελέγχοντας κατά την εκτέλεση του αλγορίθμου αν η σχέση $d\left[J_{ib_0}\right] > d\left[J_{l-1b_2}\right] + S_{b_0}$ για $l \le i \le l+v-1$ ισχύει.

Αντίστοιχος μηχανισμός μπορεί να δειχθεί και για τον αλγόριθμο που τοποθετεί εργασίες από την αρχή προς το τέλος. Η απόδειξη βασίζεται στο γεγονός πως η παραβίαση του EDD οδηγεί σε γνώση της κατανομής της μέγιστης καθυστέρησης.

Το παρακάτω θεώρημα ορίζει έναν μηχανισμό φραγμάτων για τον απαριθμητικό αλγόριθμο που περιγράφτηκε στο § 4.1.

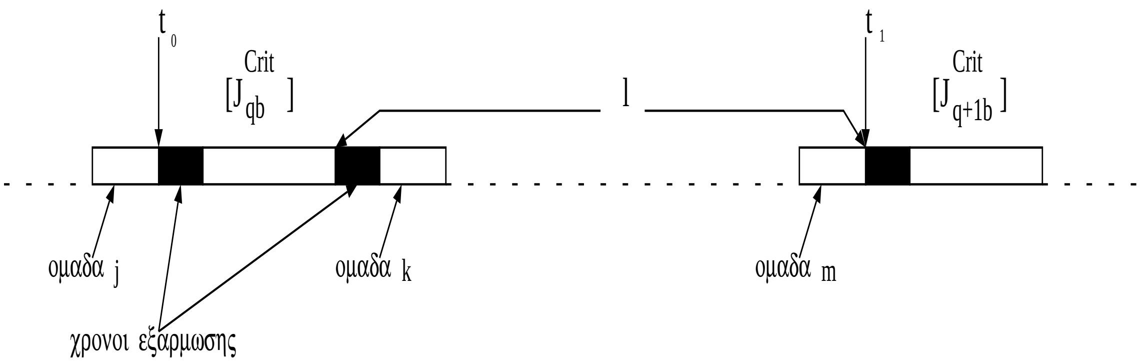
Θεώρημα 4.3.2. Εστω ότι λύνουμε ένα πρόβλημα $MTMS_{ij}$ με τον απαριθμητικό αλγόριθμο που περιγράφεται στο κεφ. 4.1 και ότι το βέλτιστο πρόγραμμα κατασκευάζεται τοποθετώντας εργασίες από την αρχή προς το τέλος. Υποθέτουμε ότι ο κόμβος-πρόβλημα (i,j) του δένδρου έρευνας αντιστοιχεί σε ημιτελές πρόγραμμα π_1 για το οποίο έχουμε $O\left[J_{qb}^{Crit}\right] = t_0$ και $O\left[J_{q+1b}^{Crit}\right] = t_1$ (σχήμα 4.3.4).
Η έρευνα από τον κόμβο (i,j) και μετά θα πρέπει να σταματήσει αν

$$l < LB - C\left[J_{qb}^{Crit}\right] + d\left[J_{qb}^{Crit}\right] - \max_{j,k}\{S_{jk} - S_{jb} - S_{bk}\} - \max_{m}\{S_{mb}\}$$

όπου l η χρονική απόσταση μεταξύ της περάτωσης της παρτίδας $\left[J_{qb}^{Crit}\right]$ και της έναρξης της κατεργασίας της παρτίδας $\left[J_{q+1b}^{Crit}\right]$. Με LB συμβολίζουμε κάποιο κάτω φράγμα της μέγιστης καθυστέρησης.

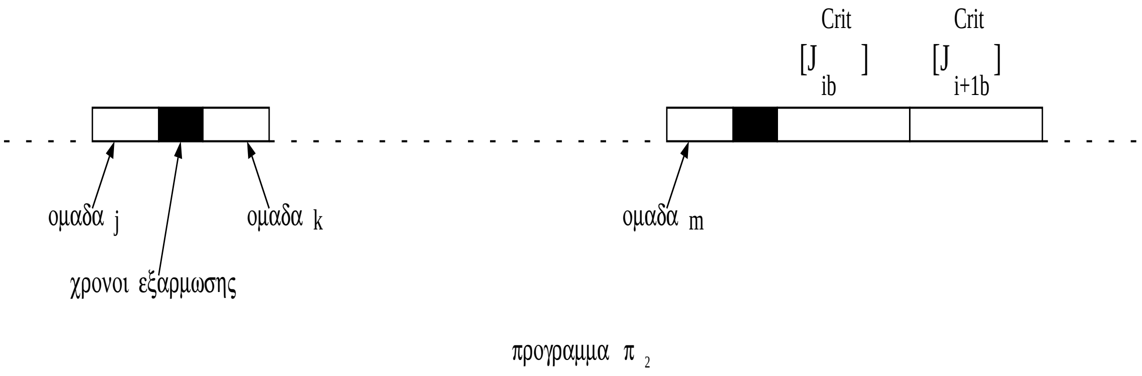
Απόδειξη: Ας θεωρήσουμε το πρόγραμμα π_2 (σχήμα 4.3.5) που προκύπτει από το πρόγραμμα π_1 αν μεταφέρουμε την παρτίδα $\left[J_{qb}^{Crit}\right]$ αριστερά της παρτίδας $\left[J_{q+1b}^{Crit}\right]$. Προφανώς η έρευνα από τον κόμβο (i,j) θα πρέπει να σταματήσει αν

$$LB > \max_{j,k,m} T^2\left[J_{qb}^{Crit}\right] \qquad (4.3.1)$$



Σχήμα 4.3.4 : Το πρόγραμμα π_1.

όπου με $T^2\left[J_{qb}^{Crit}\right]$ συμβολίζουμε την καθυστέρηση της παρτίδας $\left[J_{qb}^{Crit}\right]$ στο πρόγραμμα π_2.



Σχήμα 4.3.5 : Το πρόγραμμα π_2.

Παρατηρούμε ότι

$$T^2\left[J_{q+1b}^{Crit}\right] = T^1\left[J_{q+1b}^{Crit}\right] - S_{jb} - S_{bk} + S_{jk}. \tag{4.3.2}$$

Ακόμα ας ονομάσουμε ct την σταθερή ποσότητα

$$ct = T\left[J_{qb}^{Crit}\right](t) - T\left[J_{q+1b}^{Crit}\right](t+p\left[J_{qb}^{Crit}\right]). \tag{4.3.3}$$

Η ποσότητα ct εκφράζει την διαφορά των καθυστερήσεων των παρτίδων $\left[J_{qb}^{Crit}\right]$ και $\left[J_{q+1b}^{Crit}\right]$ όταν αυτές τοποθετηθούν διαδοχικά σε οποιοδήποτε σημείο του άξονα των χρόνων $t > d_{\max} = \max_{i,j}\{d_{ij}\}$.

Ξεκινώντας από την (4.3.1) και κάνοντας πράξεις έχουμε :

$$LB > \max_{j,k,m} T^2\left[J_{qb}^{Crit}\right] \Rightarrow$$

$$LB > \max_{j,k,m}\{T^2\left[J_{q+1b}^{Crit}\right]\} + ct \Rightarrow$$

$$LB > \max_{j,k,m}\{S_{jk} - S_{jb} - S_{bk} + T^1\left[J_{q+1b}^{Crit}\right]\} + ct \Rightarrow$$

$$LB > \max_{j,k,m}\{S_{jk} - S_{jb} - S_{bk} + S_{mb}\} + t_1 + p\left[J_{q+1b}^{Crit}\right] - d\left[J_{q+1b}^{Crit}\right] + ct \Rightarrow$$

$$LB > \max_{j,k,m}\{S_{jk} - S_{jb} - S_{bk} + S_{mb}\} + C^1\left[J_{qb}^{Crit}\right] + l + p\left[J_{q+1b}^{Crit}\right] - d\left[J_{q+1b}^{Crit}\right] + ct \Rightarrow$$

$$l < LB - C^1\left[J_{qb}^{Crit}\right] - p\left[J_{q+1b}^{Crit}\right] + d\left[J_{q+1b}^{Crit}\right] - ct - \max_{j,k}\{S_{jk} - S_{jb} - S_{bk}\} - \max_{m}\{S_{mb}\} \Rightarrow$$

$$l < LB - C^1\left[J_{qb}^{Crit}\right] - p\left[J_{q+1b}^{Crit}\right] + d\left[J_{q+1b}^{Crit}\right] - d\left[J_{q+1b}^{Crit}\right] + p\left[J_{q+1b}^{Crit}\right] + d\left[J_{qb}^{Crit}\right] -$$

$$- \max_{j,k}\{S_{jk} - S_{jb} - S_{bk}\} - \max_{m}\{S_{mb}\} \Rightarrow$$

$$l < LB + d\left[J_{qb}^{Crit}\right] - C^1\left[J_{qb}^{Crit}\right] - \max_{j,k}\{S_{jk} - S_{jb} - S_{bk}\} - \max_{m}\{S_{mb}\}. \square$$

Στην περίπτωση των προβλημάτων $MTMS_j$ και $MTMS$ το θεώρημα 4.3.2 φυσικά ισχύει και μάλιστα το κριτήριο παύσεως της έρευνας απλουστεύεται σημαντικά.

Πόρισμα 4.3.1 Εστω ότι λύνουμε ένα πρόβλημα $MTMS_j$ ή $MTMS$ με τον απαριθμητικό αλγόριθμο που περιγράφεται στο κεφ. 4.1 και ότι το βέλτιστο πρόγραμμα κατασκευάζεται τοποθετώντας εργασίες από την αρχή προς το τέλος. Υποθέτουμε ότι ο κόμβος-πρόβλημα (i,j) του δένδρου έρευνας αντιστοιχεί σε ημιτελές πρόγραμμα π_1 για το οποίο έχουμε $O\left[J_{qb}^{Crit}\right] = t_0$ και $O\left[J_{q+1b}^{Crit}\right] = t_1$.

Η έρευνα από τον κόμβο (i,j) και μετά θα πρέπει να σταματήσει αν

$$l < LB - C\left[J_{qb}^{Crit}\right] + d\left[J_{qb}^{Crit}\right]$$

όπου l η χρονική απόσταση μεταξύ της περάτωσης της παρτίδας $\left[J_{qb}^{Crit}\right]$ και της έναρξης της κατεργασίας της παρτίδας $\left[J_{q+1b}^{Crit}\right]$. Με LB συμβολίζουμε κάποιο κάτω φράγμα της μέγιστης καθυστέρησης.

Με απλά λόγια το θεώρημα 4.3.2 λέει πως μετά την τοποθέτηση της παρτίδας $\left[J_{qb}^{Crit}\right]$ για χρονικό διάστημα

$$l = LB - C\left[J_{qb}^{Crit}\right] + d\left[J_{qb}^{Crit}\right] - \max_{j,k}\{S_{jk} - S_{jb} - S_{bk}\} - \max_{m}\{S_{mb}\}$$

θα πρέπει από την ομάδα b να πραγματοποιούμε μόνο δεσίματα παρτίδων και όχι τοποθετήσεις. Συνεπώς η εκθετική πλοκή της ομάδας b στον συνολικό ορίζοντα προγραμματισμού μετασχηματίζεται σε γραμμική για συγκεκριμένα υποδιαστήματα του ορίζοντα προγραμματισμού. Το μέγεθος της μέγιστης καθυστέρησης, η ακρίβεια του κάτω φράγματος καθώς και ο τρόπος αλληλεπικάλυψης των υποδιαστημάτων (που καθορίζεται βασικά από την κατανομή των προθεσμιών) είναι καθοριστικοί παράγοντες για τον βαθμό απόδοσης του προηγούμενου μηχανισμού φραγμάτων. Πάντως παρόλο που στην εκτίμηση των επιδόσεων του αλγορίθμου το κάτω φράγμα υπολογίζεται μια και μόνο φορά (πριν ξεκινήσει η διαδικασία έρευνας) και παρόλο που αυτό είναι αρκετά χαλαρό η απόδοση του μηχανισμού είναι σημαντική.

Ο τρόπος υπολογισμού του κάτω φράγματος για το πρόβλημα $MTMS_j$ και $MTMS$ προκύπτει από την παρατήρηση ότι κάθε πρόγραμμα περιέχει τουλάχιστον B χρόνους εξάρμωσης. Αρχικά τοποθετούμε τις παρτίδες με βάση τον EDD σαν να μην ανήκουν σε διαφορετικές ομάδες. Μετά εισάγουμε B χρόνους εξάρμωσης όσο το δυνατόν πιο αργά. Πρώτα εισάγεται ο μικρότερος από τους χρόνους εξάρμωσης, μετά ο δεύτερος μικρότερος και τελευταίος ο πιο μεγάλος από τους χρόνους εξάρμωσης. Ο πρώτος χρόνος εξάρμωσης τοποθετείται πριν από την έναρξη της πρώτης εργασίας. Ο δεύτερος χρόνος εξάρμωσης τοποθετείται O_1 εργασίες αργότερα, όπου O_1 ο αριθμός των εργασιών της ομάδας με τις περισσότερες εργασίες. Ο τρίτος χρόνος εξάρμωσης τοποθετείται O_2 εργασίες αργότερα, όπου O_2 ο αριθμός των εργασιών της δεύτερης μεγαλύτερης ομάδας. Με τον ίδιο τρόπο τοποθετούμε και τους υπόλοιπους χρόνους εξάρμωσης. Η καθυστέρηση του συγκεκριμένου προγράμματος αποτελεί κάτω φράγμα για την καθυστέρηση της βέλτιστης λύσης. Ας παρατηρήσουμε ότι το κάτω φράγμα υπολογίζεται πολύ εύκολα αφού απαιτεί τρεις διατάξεις. Το κάτω φράγμα είναι σχετικά σφικτό αν οι χρόνοι εξάρμωσης είναι μικροί. Αυτό γιατί καταλήγουμε σε κάτω φράγμα παραλείποντας χρόνους εξάρμωσης, οπότε αν αυτοί είναι μικροί κινούμαστε σε κοντινή υπερβέλτιστη τιμή. Αντίθετα αν οι χρόνοι εξάρμωσης είναι μεγάλοι τότε το κάτω φράγμα είναι πολύ χαλαρό όμως η λειτουργία του μηχανισμού που βασίζεται στο θεώρημα 3.6.1 γίνεται πολύ πιο αποδοτική. Η ύπαρξη τέτοιων συμπληρωματικών μηχανισμών κλαδέματος είναι καθοριστική ώστε ο αλγόριθμος να λειτουργεί γρήγορα σε κάθε περίπτωση του προβλήματος.

Ενδιαφέρον παρουσιάζει η παρακάτω παρατήρηση

- αν για μια παρτίδα $\left[J_{ib}^{Crit}\right]$ έχουμε

$$d\left[J_{ib}^{Crit}\right] > \sum_{ib} p_{ib} + \sum_{b=1}^{B} N_b^{Crit} S_b$$

τότε υπάρχει βέλτιστο πρόγραμμα στο οποίο η παρτίδα $\left[J_{ib}^{Crit}\right]$ είναι στην τελευταία θέση και έχει καθυστέρηση ίση με μηδεν. Η απόδειξη είναι προφανής αφού κανένα πρόγραμμα δεν έχει χρόνο περάτωσης μεγαλύτερο της ποσότητας $\sum_{ib} p_{ib} + \sum_{b=1}^{B} N_b^{Crit} S_b$.

4.4. Σύνοψη του απαριθμητικού αλγορίθμου

Για την εύκολη κατανόηση της ροής του αλγορίθμου που αναπτύχθηκε στο κεφάλαιο αυτό, δίνουμε την περιγραφή του σε ψευδογλώσσα.

Με V συμβολίζουμε το σύνολο των εργασιών για διάταξη. Με $crPos$ συμβολίζουμε την θέση στην οποία θα τοποθετήσουμε την παρτίδα με την μικρότερη καθυστέρηση. Ο πίνακας $S[i], i=1,...,N^{Crit}$ κρατάει την παρτίδα που βρίσκεται στην θέση i του υπο κατασκευή προγράμματος. Αν $S[i]=-1$ τότε στην θέση i δέν έχει τοποθετηθεί ακόμα παρτίδα. Ο πίνακας $distance[i], i=1,...,B$ κρατάει τον χρόνο για τον οποίο από την ομάδα i εκτελούμε μόνο δεσίματα. Τέλος η μεταβλητή $Constr$ κρατάει την ομάδα από την οποία έγινε η τελευταία τοποθέτηση. Αν $Constr=-1$ τότε βρισκόμαστε στην αρχή του προγράμματος ή υπάρχουν περισσότερα του ενός προγράμματα που έχουν την ίδια μέγιστη καθυστέρηση τον ίδιο (ή και μικρότερο) χρόνο περάτωσης και τελειώνουν σε διαφορετικές εργασίες.

Αλγόριθμος για το πρόβλημα MTMS$_j$.

Βήμα 1 :

Κατασκεύασε τις απλές παρτίδες (σελ. 27).

Βήμα 2 :

Υπολόγισε ένα κάτω φράγμα LB της μέγιστης καθυστέρησης (σελ. 63).

Βήμα 3 : /* αρχικές συνθήκες */

Θέσε $V = \{ \left[J_{ib}^{Crit}\right] \}, b=1,..,B$, $i=1,...,N_b^{Crit}$; $crPos=1$; $Constr=-1$;

$S[i]=-1$, $i=1,...,N^{Crit}$; $distance[i] = -\infty$, $i=1,...,B$.

Βήμα 4 : /* έξοδος της αναδρομικής διαδικασίας */

Εστω $\left[J_{ib_0}^{Crit}\right]$ η απλή παρτίδα με την μικρότερη προθεσμία.

Αν η $\left[J_{ib_0}^{Crit}\right]$ είναι η τελευταία παρτίδα για διάταξη τοποθέτησε την (αν είναι δυνατόν) και επίστρεψε την μέγιστη καθυστέρηση του προγράμματος καθώς και την ταυτότητα του.

Αν η $\left[J_{ib_0}^{Crit}\right]$ είναι η τελευταία παρτίδα για διάταξη αλλά ο περιορισμός την απορρίπτει, τότε κλάδεψε το κλαδί επιστρέφοντας -1.

Βήμα 5 : /* αναδρομική κλήση για το αριστερό παιδί */

Αν ($(b_0 \neq Constr)$ && $(distance[b_0]<0)$ && ($(\left[J_{1b_0}^{Crit}\right] = \left[J_{N_{b_0}^{Crit}b_0}^{Crit}\right])$ || $(T^0\left[J_{1b_0}^{Crit}\right] - T^0\left[J_{2b_0}^{Crit}\right] > S_{b_0})$)) , τότε τοποθέτησε την απλή παρτίδα $\left[J_{1b_0}^{Crit}\right]$ στην θέση $crPos$.

Θέσε $V = V - \{ \left[J_{1b_0}^{Crit}\right] \}$.

Θέσε $S[crPos] = \left[J_{1b_0}^{Crit}\right]$; $crPos = crPos + 1$;

$distance[b_0] = LB - (O\left[J_{1b_0}^{Crit}\right] + S_{b_0} + p\left[J_{1b_0}^{Crit}\right]) + d\left[J_{1b_0}^{Crit}\right]$.

Αν η προθεσμία της παρτίδας $\left[J_{1b_0}^{Crit}\right]$ είναι διαφορετική της προθεσμίας της αμέσως προηγούμενης παρτίδας, τότε θέσε $Constr=b_0$ άλλως $Constr=-1$.

Εκτέλεσε αναδρομικά από το Βήμα 4.

Βήμα 6 : /* αναδρομική κλήση για το δεξιό παιδί */

Αν $\left[J_{1b_0}^{Crit}\right] \neq \left[J_{N_{b_0}^{Crit}b_0}^{Crit}\right]$, τότε δέσε την απλή παρτίδα $\left[J_{1b_0}^{Crit}\right]$.

Θέσε $V = V - \{ \left[J_{1b_0}^{Crit}\right] \} - \{ \left[J_{2b_0}^{Crit}\right] \} + \{ \left[J_{1b_0}^{Crit}, J_{2b_0}^{Crit}\right] \}$; $Constr=Constr$;

$distance[i] = distance[i]$, $i=1,...,B$; $crPos=crPos$.

Εκτέλεσε αναδρομικά από το Βήμα 4.

Βήμα 7 :

Επίστρεψε την μικρότερη από τις καθυστερήσεις που βρέθηκαν στα βήματα 5 και 6 και την ταυτότητα του αντιστοίχου προγράμματος.

5. Προσεγγιστικός αλγόριθμος

Από το κεφ. 7, όπου περιγράφονται οι επιδόσεις του αλγορίθμου, φαίνεται πως το μέγεθος των προβλημάτων που επιλύονται σε χρόνο ενός λεπτού είναι αρκετά μεγάλο. Πολλά όμως από τα πραγματικά προβλήματα έχουν πολύ μεγάλο αριθμό εργασιών ή, ακόμα χειρότερα, μεγάλο αριθμό ομάδων που καθιστά την εφαρμογή του αλγορίθμου αδύνατη. Εξ άλλου στην πράξη είναι επιθυμητή η εύρεση λύσης, ακόμα και μή βέλτιστης, σε σύντομο χρονικό διάστημα. Στο κεφάλαιο αυτό αναπτύσσουμε μια παραλλαγή του προηγούμενου αλγορίθμου που μπορεί να λύσει μεγάλα προβλήματα γρήγορα με καλή προσέγγιση του βελτίστου.

5.1. Αυθαίρετο δέσιμο παρτίδων

Η βασική ιδέα του γρήγορου προσεγγιστικού αλγορίθμου είναι και πάλι η εφαρμογή του προηγούμενου απαριθμητικού σχήματος αλλά σε κάποιο υποσύνολο της αρχικής περιοχής έρευνας. Η περιοχή έρευνας μικραίνει υλοποιώντας αυθαίρετα δεσίματα περτίδων. Κάθε αυθαίρετο δέσιμο παρτίδων αντιστοιχεί σε σύνολο κλαδεμάτων του δυαδικού δενδρου. Με αυτόν τον τρόπο μεγάλες περιοχές έρευνας απορρίπτονται και ο αλγόριθμος τερματίζει σε σύντομο χρόνο (τάξεως λεπτου). Το ζητούμενο είναι να γίνει σωστή επιλογή των αυθαίρετων δεσιμάτων έτσι ώστε η πιθανότητα να βρούμε βέλτιστη λύση να είναι μεγάλη. Η επιλογή των αυθαίρετων δεσιμάτων γίνεται με την εφαρμογή ενός μυωπικού αλγορίθμου που βασίζεται στο παρακάτω θεώρημα.

Θεώρημα 5.1.1 Εστω πρόβλημα $MTMS_{ij}$ και δύο διαδοχικές παρτίδες $\left[J_{i_0 b_0}^{Crit}\right]$ και $\left[J_{i_0+1 b_0}^{Crit}\right]$ της ομάδας b_0.

Εστω $M\hat{T}MS_{ij}$ το προηγούμενο πρόβλημα που επιπλέον έχει ως περιορισμό οι παρτίδες $\left[J_{i_0 b_0}^{Crit}\right]$ και $\left[J_{i_0+1 b_0}^{Crit}\right]$ να είναι δεμένες μεταξύ τους. Αν με T^* και $\hat{T}$ συμβολίσουμε τις βέλτιστες τιμές της αντικειμενικής συνάρτησης για τα $MTMS_{ij}$ και $M\hat{T}MS_{ij}$ αντίστοιχα τότε έχουμε:

$$T^* \leq \hat{T} \leq T^* + ct$$

όπου $ct = d\left[J_{i_0+1 b_0}^{Crit}\right] - p\left[J_{i_0+1 b_0}^{Crit}\right] - d\left[J_{i_0 b_0}^{Crit}\right]$.

Απόδειξη: Η σχέση $T^* \le \hat{T}$ προφανώς ισχύει. Αν υπάρχει βέλτιστη διάταξη του $MTMS_{ij}$ στην οποία οι παρτίδες $\left[J^{Crit}_{i_0b_0}\right]$ και $\left[J^{Crit}_{i_0+1b_0}\right]$ είναι δεμένες μεταξύ τους τότε προφανώς η διάταξη αυτή είναι εφικτή για το πρόβλημα $M\hat{T}MS_{ij}$ και συνεπώς $\hat{T} = T^*$ και αφού $ct > 0$ η προς απόδειξη σχέση ισχύει. Ας υποθέσουμε λοιπόν πως καμμιά βέλτιστη διάταξη του $MTMS_{ij}$ δεν έχει τις παρτίδες $\left[J^{Crit}_{i_0b_0}\right]$ και $\left[J^{Crit}_{i_0+1b_0}\right]$ δεμένες μεταξύ τους. Εστω π_1 ένα τέτοιο πρόγραμμα (με βάση τον EDD). Ας θεωρήσουμε το πρόγραμμα π_2 που προέρχεται από το π_1 αν μετακινήσουμε την παρτίδα $\left[J^{Crit}_{i_0b_0}\right]$ αριστερά της $\left[J^{Crit}_{i_0+1b_0}\right]$. Για κάθε παρτίδα εκτός της $\left[J^{Crit}_{i_0b_0}\right]$ έχουμε :

$$T^2\left[J^{Crit}_{ib}\right] \le T^1\left[J^{Crit}_{ib}\right].$$

Ομως για την παρτίδα $\left[J^{Crit}_{i_0b_0}\right]$ έχουμε:

$$T^2\left[J^{Crit}_{i_0b_0}\right] = T^2\left[J^{Crit}_{i_0+1b_0}\right] + ct \le T^1\left[J^{Crit}_{i_0+1b_0}\right] + ct$$

και συνεπώς

$$\hat{T} \le \max_{\substack{1\le b\le B \\ 1\le i\le N_b^{Crit}}} \{T^2\left[J^{Crit}_{ib}\right]\} = \max\{ \max_{\substack{1\le b\le B \\ 1\le i\le N_b^{Crit} \\ (i,b)<>(i_0,b_0)}} \{T^2\left[J^{Crit}_{ib}\right]\}, T^2\left[J^{Crit}_{i_0b_0}\right]\} \le$$

$$\max\{T^*, T^1\left[J^{Crit}_{i_0+1b_0}\right] + ct\} \le \max\{T^*, T^* + ct\} = T^* + ct$$

και συνεπώς η προς απόδειξη σχέση ισχύει.□

Πόρισμα 5.1.1 Το θεώρημα 5.1.1 ισχύει και για τα προβλήματα $MTMS_j$ και $MTMS$.

Αυτό που δηλώνει το θεώρημα 5.1.1 είναι πως το αυθαίρετο δέσιμο παρτίδων μπορεί να μεταβάλλει την τιμή της αντικειμενικής συνάρτησης αλλά όχι περισσότερο από ένα γνωστό εύκολα υπολογίσιμο μέγεθος. Προκειμένου λοιπόν να περιορίσουμε την αρχική περιοχή έρευνας εκτελούμε αυθαίρετα διαδοχικά δεσίματα που έχουν ελάχιστη τιμή ct. Μετά από κάθε αυθαίρετο δέσιμο υπολογίζουμε τη νέα ct και συνεχίζουμε την διαδικασία. Η διαδικασία τερματίζεται μετά από συγκεκριμένο αριθμό αυθαίρετων δεσιμάτων που προσδιορίζεται από τον χρήστη. Με τον τρόπο αυτό τελικά βρίσκουμε ένα υποσύνολο του δένδρου έρευνας έχοντας να λύσουμε ένα πρόβλημα με πολύ λιγότερες εργασίες από το αρχικό. Το ερώτημα είναι κατά πόσο αυτός ο τρόπος προσέγγισης της βέλτιστης λύσης οδηγεί σε αποτελέσματα κοντά στο βέλτιστο. Εκτεταμένες δοκιμές δείχνουν ότι με πολύ μεγάλη πιθανότητα (0.7-0.96) ο προσεγγιστικός αλγόριθμος δίνει βέλτιστη λύση. Επίσης στις περιπτώσεις που δεν κατορθώνει να βρεί βέλτιστη λύση η προσεγγιστική λύση βρίσκεται πάρα πολύ κοντα στην βέλτιστη με λάθη μικρότερα του 1%-2% (βλ. κεφ. 7).

5.2. Μετασχηματισμοί των p_{ib}, d_{ib}

Αμεση απόρροια του θεωρήματος 3.7.1 είναι το γεγονός πως το αυθαίρετο δέσιμο μιας ακολουθίας από παρτίδες $\left[J_{ib}^{Crit}\right],..,\left[J_{jb}^{Crit}\right]$ δημιουργεί την σύνθετη παρτίδα $\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]$ με χρόνο επεξεργασίας

$$p\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]=\sum_{l=i}^{j}p\left[J_{lb}^{Crit}\right]$$

και προθεσμία

$$d\left[J_{ib}^{Crit},J_{jb}^{Crit}\right]=d\left[J_{ib}^{Crit}\right]+\sum_{l=i+1}^{j}p\left[J_{lb}^{Crit}\right]$$

Επίσης από την σχέση (3.6.1) είναι προφανές πως στην ομάδα b μετα το αυθαίρετο δέσιμο της παρτίδας $\left[J_{i-1b}^{Crit}\right]$ με την παρτίδα $\left[J_{ib}^{Crit}\right]$ μόνο η παρτίδα $\left[J_{i+1b}^{Crit}\right]$ θα πρέπει να εξεταστεί ξανά για να διαπιστωθεί αν το θεώρημα 3.6.1 την χαρακτηρίζει ως δεμένη ή όχι. Ολες οι υπόλοιπες παρτίδες εξακολουθούν να έχουν τον ίδιο χαρακτηρισμό που είχαν και πριν το αυθαίρετο δέσιμο.

Από το θεώρημα 5.1.1 με επαγωγή συνάγουμε ότι ακολουθία αυθαίρετων δεσιμάτων διαδοχικών παρτίδων δεν οδηγεί σε σφάλμα μεγαλύτερο από το άθροισμα των αντίστοιχων ct. Πολύ μεγαλύτερο ενδιαφέρον παρουσιάζει η περίπτωση ακολουθίας αυθαίρετων δεσιμάτων από διαφορετικές ομάδες. Το σφαλμα φράσσεται όχι από το άθροισμα των αντίστοιχων ct αλλά από το μέγιστο από τα ct. Η παρατήρηση αυτή μπορεί να οδηγήσει και σε μία διαφορετική πολιτική αυθαίρετων δεσιμάτων που αναμένουμε να δίνει με μεγαλύτερη πιθανότητα βέλτιστη λύση.

6. Υλοποίηση

Σ' αυτό το κεφάλαιο αναφέρουμε το περιβάλλον στο οποίο οι αλγόριθμοι υλοποιήθηκαν καθώς και μερικές από τις βασικές επιλογές στις δομές δεδομένων.

6.1. Περιβάλλον υλοποίησης

Για την υλοποίηση των αλγορίθμων χρησιμοποιήθηκαν σταθμοί εργασίας Sun με λειτουργικό σύστημα Unix SunOS 4.1.1 και σύστημα παραθύρων Xwindows έκδοση 11. Ολα τα αποτελέσματα προέρχονται από SPARCstation 2 με χρονισμό 40 MHz, απόδοση της τάξεως των 28 MIPS και μνήμη 48 MBytes. Αν και το μηχάνημα που χρησιμοποιήθηκε είναι από τους πιο γρήγορους σταθμούς εργασίας, ελάχιστη είναι η βελτίωση που παρατηρείται ως προς το μέγεθος των προβλημάτων που επιλύονται σε σχέση με πιο αργά μηχανήματα. Αιτία του φαινομένου είναι ο εκθετικός χαρακτήρας του αλγορίθμου. Αυτό φαίνεται καθαρά στο παρακάτω υποθετικό παράδειγμα.

Ας υποθέσουμε ότι έχουμε κάποιον υπολογιστή που λύνει το πρόβλημα $MTMS_j$ με τον αλγόριθμο που περιγράφεται στο § 4.1.1. Επίσης ας υποθέσουμε ότι σε μία ώρα κατορθώνει και λύνει πρόβλημα n_0 εργασιών. Οπως ήδη έχει δειχθεί η πολυπλοκότητα του αλγορίθμου είναι $O(2^n)$. Η χρήση ενός υπολογιστή με 1000-πλάσια ταχύτητα του πρώτου θα κατορθώσει να λύσει πρόβλημα n_0+10 εργασιών. Παρατηρούμε πως παρόλο που αυξήσαμε την ταχύτητα του υπολογιστή 1000 φορές το μέγεθος των προβλημάτων που επιλύονται στον ίδιο χρόνο αυξήθηκε μόνο κατά 10 εργασίες.

Η γλώσσα που χρησιμοποιήθηκε είναι η C. Η μετάφραση έγινε με την επιλογή βελτιστοποίησης -O3 και χρησιμοποιήθηκε η 4.1.1 εκδοση του μεταφραστού της. Οι βασικοί λόγοι για τους οποίους προτιμήθηκε η C είναι η πληρότητα των προγραμματιστικών δομών της καθώς και η ποιότητα του κώδικα που παράγει σε σχέση με τον μικρό χρόνο μεταγλώττισης που απαιτεί.

6.2. Υλοποίηση αλγορίθμων

Τόσο ο ακριβής όσο και ο προσεγγιστικός αλγόριθμος σε κάθε βήμα του δημιουργεί δύο νέα προβλήματα ίδια με το αρχικό αλλά με λιγότερες εργασίες.

Προφανώς μια αναδρομική διαδικασία μπορεί να υλοποιήσει ένα τέτοιο σχήμα. Η διαδικασία ελέγχει αν πρέπει να δημιουργήσει το αριστερό ή(και) το δεξιό παιδί του δένδρου και μετά καλεί τον εαυτό της για κάθε παιδί. Φυσικά η βασική παράμετρος της διαδικασίας είναι οι εργασίες που πρόκειται να διαταχθούν. Το ενδιαφέρον είναι πως ενώ τα δυνατά προγράμματα είναι στην χειρότερη περίπτωση 2^N οι δυνατές παρτίδες είναι το πολύ $\frac{N(N+1)}{2}$. Προφανώς † ο αριθμός των δυνατών παρτίδων στην ομάδα b είναι $\frac{N_b(N_b+1)}{2}$. Ο συνολικός αριθμός παρτίδων είναι

$$\sum_{b=1}^{B} \frac{N_b(N_b+1)}{2} < \frac{N(N+1)}{2} \tag{6.2.1}$$

δοθέντος ότι $\sum_{b=1}^{B} N_b = N$.

Οτι η σχέση (6.2.1) ισχύει για $B=2$ μπορεί να δειχθεί εύκολα.

$$\sum_{b=1}^{2} \frac{N_b(N_b+1)}{2} = \frac{N_1(N_1+1)}{2} + \frac{N_2(N_2+1)}{2} = \frac{N_1^2+N_2^2+N_1+N_2}{2} <$$

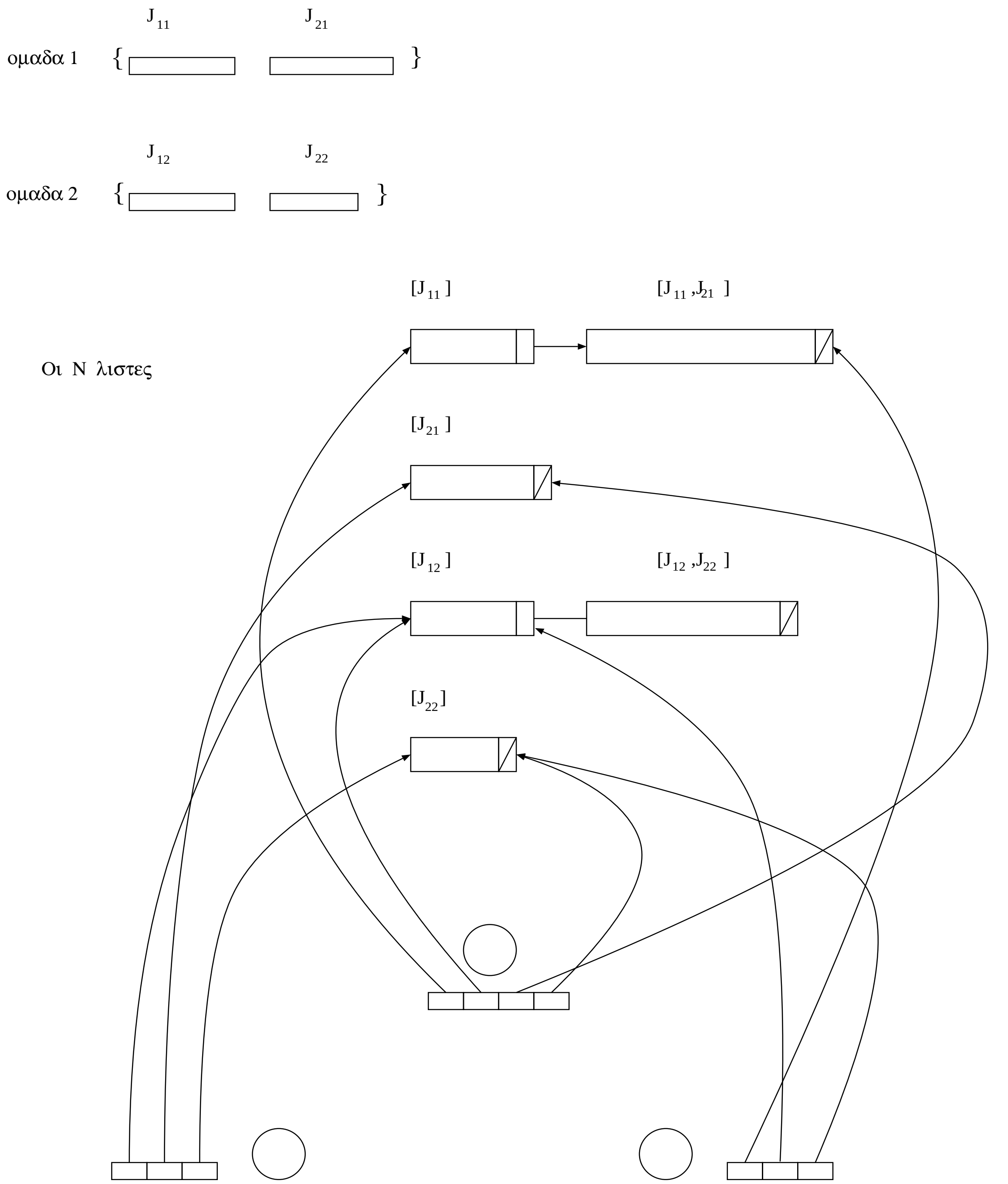
$$< \frac{N_1^2+N_2^2+N_1+N_2+2N_1N_2}{2} = \frac{(N_1+N_2)^2+N_1+N_2}{2} = \frac{N(N+1)}{2} \tag{6.2.2}$$

και σύνεπως η σχέση (6.2.1) για $B=2$ ισχύει. Επαγωγικά γενικεύουμε την σχέση (6.2.1) για κάθε ακέραιο B.

Αφού ο αριθμός των παρτίδων είναι πολυωνυμικά φραγμένος από το μέγεθος του προβλήματος μπορούμε να δημιουργήσουμε όλες τις δυνατές παρτίδες στην αρχή του αλγορίθμου και μετά να χειριζόμαστε δείκτες στις παρτίδες. Επίσης τις παρτίδες τις εισάγουμε σε N λίστες κατά τέτοιο τρόπο ώστε κάθε παρτίδα i να έχει ως επόμενή της την παρτίδα που προκύπτει αν η i δεθεί δεξιά(αριστερά). Ετσι η δημιουργία των εργασίων για το αριστερό υποπρόβλημα είναι άμεση ενώ η δημιουργία του δεξιού υποπρόβληματος γίνεται με την μετάβαση από την παρτίδα i στην επόμενή της στην κατάλληλη λίστα.

† Στην τυχαία ομάδα b μπορούμε να έχουμε μια παρτίδα με N_b εργασίες δύο παρτίδες με N_b-1 εργασίες, και τελικά N_b παρτίδες με μια εργασία. Συνεπώς οι δυνατές παρτίδες είναι $N_b(N_b+1)/2$.

Στο σχήμα 6.2.1 απεικονίζονται οι αρχικές λίστες όλων των δυνατών παρτίδων, το αρχικό σύνολο εργασίων καθώς και τα σύνολα των εργασιών για το αριστερό και δεξιό παιδί του επιπέδου 0 ενός τυχαίου προβλήματος $MTMS_j$ τεσσάρων εργασιών κατανεμημένων σε δύο ομάδες.



Σχήμα 6.2.1 : Παράσταση του συνόλου των εργασιών.

7. Αξιολόγηση επιδόσεων του αλγορίθμου

Στο κεφάλαιο αυτό παραθέτουμε τα κριτήρια αξιολόγησης των αλγορίθμων, τον τρόπο με τον οποίο παράγουμε τα προβλήματα δοκιμών και τελικά τις επιδόσεις των αλγορίθμων.

7.1. Κριτήρια επιδόσεων

Για τους ακριβείς αλγορίθμους δίνουμε το μέγιστο μέγεθος προβλημάτων που επιλύονται σε χρόνο ενός λεπτού. Φυσικά αναφερόμαστε σε χρόνο CPU. Επίσης θεωρούμε πως το πρόβλημα μεγέθους n επιλύεται σε χρόνο ενός λεπτού αν το 80% των προβλημάτων μεγέθους n επιλύονται σε χρόνο λιγότερο του ενός λεπτού. Επίσης δίνουμε τον ελάχιστο, τον μέγιστο και τον μέσο χρόνο που απαιτείται για την επίλυση των προβλημάτων μεγέθους n. Για τους προσεγγιστικούς αλγορίθμους δίνουμε το μέσο απόλυτο σφάλμα, το μέσο σχετικό σφαλμα και την συχνότητα με την οποία ο αλγόριθμος δίνει βέλτιστη λύση. Θα πρέπει να επισημάνουμε ότι τα σφάλματα αναφέρονται σε σχέση με τα προβλήματα που δεν δίνουν βέλτιστη λύση και όχι σε σχέση με το σύνολο των προβλημάτων. Συνεπώς το πραγματικό αναμενόμενο σφάλμα είναι πολύ μικρότερο και δίνεται από την σχέση:

μεσο σφάλμα(σχετικό ή απόλύτο) * (1-πιθανότητα ευρεσης βέλτιστης λύσης)

Επίσης δίνουμε τον μέσο χρόνο εκτέλεσης τόσο του προσεγγιστικού όσο και του ακριβούς αλγορίθμου καθώς και την απόσταση (απόλυτη και σχετική) του κάτω φράγματος από την βέλτιστη λύση για τον έλεγχο της απόδοσης του.

7.2. Προβλήματα δοκιμών

Για την παραγωγή ενός προβλήματος N εργασιών με B ομάδες κάνουμε χρήση των παραμέτρων:

- παράγοντας καθυστέρησης t και
- σχετικό εύρος των προθεσμιών r.

Ο παράγοντας καθυστέρησης t εκφράζει το αναμενόμενο ποσοστό καθυστερημένων εργασιών. Ετσι αν t=0.1 σημαίνει πως κατά μέσο όρο $0.1N$

εργασίες θα καθυστερήσουν.

Το σχετικό εύρος των προθεσμιών r ορίζει την περιοχή $[x_0-P\cdot r/2, x_0+P\cdot r/2]$ απ' όπου προέρχονται οι προθεσμίες. Ο όρος σχετικό αναφέρεται ως προς το συνολικό χρόνο επεξεργασίας P των εργασιών†.

Οι χρόνοι επεξεργασίας προέρχονται από την ομοιόμορφη κατανομή στο διάστημα [1,100]. Αφού παράγουμε όλα τα p_{ib} υπολογίζουμε την ποσότητα $P=\sum_{i,b} p_{ib}$ και παράγουμε τα d_{ib} με ομοιόμορφη κατανομή στο διάστημα $[P(1-t-r/2), P(1-t+r/2)]$. Η κατανομή των εργασιών στις ομάδες είναι επίσης ομοιόμορφη στο διάστημα $[1, B]$.

Οι χρόνοι εξάρμωσης ορίζονται ως ποσοστό του μέσου χρόνου επεξεργασίας και συγκεκριμένα καθορίζονται στο 10%, 50%, 100%, 150% αυτού. Στο $MTMS_j$ οι χρόνοι εξάρμωσης για κάθε ομάδα b προέρχονται από την ομοιόμορφη στο διάστημα [5,50] που αντιστοιχεί σε ποσοστά 5% και 50% του μέγιστου χρόνου επεξεργασίας.

7.3. Επιδόσεις ακριβών αλγορίθμων

Στο κεφάλαιο αυτό θα δώσουμε τις επιδόσεις για τις δύο διαφορετικές υλοποιήσεις του αλγορίθμου διαμερισμού και φραγής που περιγράφτηκε στο κεφ. 4. Η πρώτη υλοποίηση (ευθύς αλγόριθμος) κατασκευάζει το βέλτιστο πρόγραμμα διατάσσοντας εργασίες από την αρχή προς το τέλος, ενώ η δεύτερη ανάποδα (αντίστροφος αλγόριθμος).

7.3.1. Ευθύς αλγόριθμος για το πρόβλημα *MTMS*.

Τα αποτελέσματα αναφέρονται σε προβλήματα *MTMS* με $S=5$. Για κάθε κατηγορία λύθηκαν 10 προβλήματα που δημιουργήθηκαν με τον τρόπο που περιγράφεται στο κεφ. 7.2.

Στους πίνακες 7.3.1 εώς 7.3.4 παραθέτουμε αποτελέσματα μόνο για τέσσερεις κατηγορίες, που πιστεύουμε ότι βρίσκονται κοντά στα προβλήματα που εμφανίζονται στην πράξη στο προγραμματισμό της παραγωγής. Τα αποτελέσματα και για τις 16 κατηγορίες υπάρχουν στο παράρτημα Α της

† Ο τρόπος παραγωγής προβλημάτων με βάση τις δύο παραπάνω παραμέτρους έχει χρησιμοποιηθεί από τους Potts και Wassenhove[PoWa-85] και άλλους ερευνητές.

εργασίας. Η μονάδα του χρόνου είναι το sec.

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	9	8	8	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(29)87	9(30.1)92	2(41.2)128	30(64.2)86	1(3.3)17
Μέσος Αρ. Παρτίδων	44	42.7	39.7	51.5	43.1
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96	79	54.6	37.7	26.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.23	0.19	0.12	0.05	0.01
Μέση Καθυστέρηση	418.1	417.2	447.4	795.1	2527.4

Πίνακας 7.3.1

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	9	8	10	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(8.9)32	1(22.9)80	1(21.9)70	1(8.3)35	1(10.1)37
Μέσος Αρ. Παρτίδων	38.4	37.9	34.7	32	33.5
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96.7	79.3	60.7	43	36.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.8	0.7	0.7	0.6	0.7
Μέση Καθυστέρηση	122.4	106.7	88.7	77.6	58.4

Πίνακας 7.3.2

Η δυσκολία του προβλήματος μεγαλώνει όσο μεγαλώνει το σχετικό εύρος των προθεσμιών. Ο παράγοντας καθυστέρησης δεν επιδρά ουσιαστικά στην δυσκολία του προβλήματος.

Αξίζει να σημειώσουμε πως ο αριθμός των απλών παρτίδων σε κάθε κατηγορία παραμένει ανεξάρτητος από τον παράγοντα καθυστέρησης. Γεγονός που το αναμέναμε αφού κατά την δημιουργία των απλών παρτίδων ολισθαίνουμε τις ομάδες στην χρονική στιγμή $d_{\max} = \max_{i,b}\{ d_{ib} \}$ και συνεπώς

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	8	7	7	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(25.1)54	11(32)91	2(34.6)68	29(64.75)84	1(5.8)25
Μέσος Αρ. Παρτίδων	44.6	42.3	39.6	51.5	45.3
Μέση Απόλ. Απόσταση Κάτω Φράγματος	97.7	78.1	53.8	38.4	26.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.09	0.07	0.04	0.02	0.003
Μέση Καθυστέρηση	1035.1	1061.7	1202.4	2287.2	7499

Πίνακας 7.3.3

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	9	10	9	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(21)76	1(12.2)46	2(28.5)112	1(11.1)35	1(10.5)37
Μέσος Αρ. Παρτίδων	39.4	37	34.4	33	33.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96.1	77.8	61.4	45.5	35.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.16	0.14	0.11	0.08	0.05
Μέση Καθυστέρηση	585.9	565.3	551.2	582.5	719.9

Πίνακας 7.3.4

πάντα μπορούμε να θεωρήσουμε ότι $t=1$.

Το κάτω φράγμα είναι σχετικά ικανοποιητικό μόνο για προβλήματα με μικρό αριθμό ομάδων. Οταν οι ομάδες ξεπεράσουν τις 10 τότε η μέση σχετική απόσταση από την μέγιστη καθυστέρηση φτάνει μέχρι 0.25.

7.3.2. Αντίστροφος αλγόριθμος για το πρόβλημα *MTMS*.

Οι Πίνακες 7.3.5 και 7.3.6 παρουσιάζουν τα αποτελέσματα για δύο μόνο κατηγορίες προβλημάτων *MTMS*. Στο παράρτημα Β δίνουμε το σύνολο των αποτελεσμάτων για παράγοντα καθυστέρησης t=0.2. Τα αποτελέσματα για παράγοντα καθυστέρησης 0.4, 0.6 και 0.8 είναι σχεδόν ίδια. Η ανεξαρτησία από τον παράγοντα καθυστέρησης έχει θεωρητική βάση αφού στον αντίστροφο αλγόριθμο δεν εφαρμόζουμε μηχανισμό φραγμάτων, που προφανώς θα επηρεαζόταν από την τιμή της μέγιστης καθυστέρησης και συνεπώς από τον παράγοντα καθυστέρησης.

Οι πίνακες 7.3.5 και 7.3.6 μας δίνουν τον μέγιστο αριθμό ομάδων, όπου αν κατανείμουμε 35, 50, 75, 100, 125 ή 150 εργασίες η λύση του αντίστοιχου προβλήματος *MTMS* απαιτεί χρόνο CPU τάξεως λεπτού.

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.2			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	8(50)	6(75)	5(100)	5(125)	5(150)
25	20(35)	10(50)	8(75)	6(100)	6(125)	5(150)
50	20(35)	20(50)	10(75)	8(100)	7(125)	6(150)
75	20(35)	20(50)	11(75)	9(100)	7(125)	7(150)

Πίνακας 7.3.5

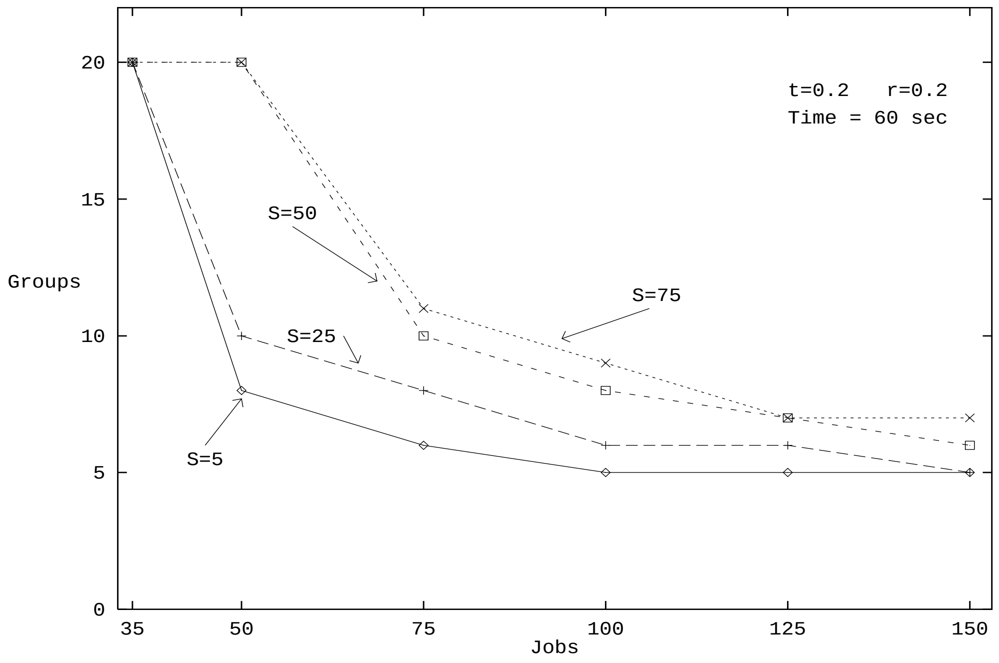
Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.4			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	5(50)	3(75)	3(100)	3(125)	2(150)
25	20(35)	5(50)	3(75)	3(100)	3(125)	3(150)
50	20(35)	6(50)	4(75)	4(100)	3(125)	3(150)
75	20(35)	7(50)	5(75)	4(100)	4(125)	3(150)

Πίνακας 7.3.6

Είναι φανερό πως οι επιδόσεις του αντίστοφου αλγορίθμου είναι σημαντικά χαμηλότερες. Μοναδική διαφορά είναι η χρήση μηχανισμού φραγμάτων από τον ευθύ αλγόριθμο. Ενδεικτικό παράδειγμα των διαφορετικών επιδόσεων του αλγορίθμου είναι το παρακάτω. Ο ευθύς αλγόριθμος, σε χρόνο 30 δευτερολέπτων λύνει το πρόβλημα 60 εργασιών σε 20 ομάδες, της κατηγορίας $t=0.2, r=0.2$ και $S=5$. Ο αντίστροφος αλγόριθμος κατορθώνει να το λύσει μόνο για 35 εργασίες. Από θεωρητική άποψη ο ευθύς αλγόριθμος εμφανίζεται να λειτουργεί 32000000 φορές γρηγορότερα.

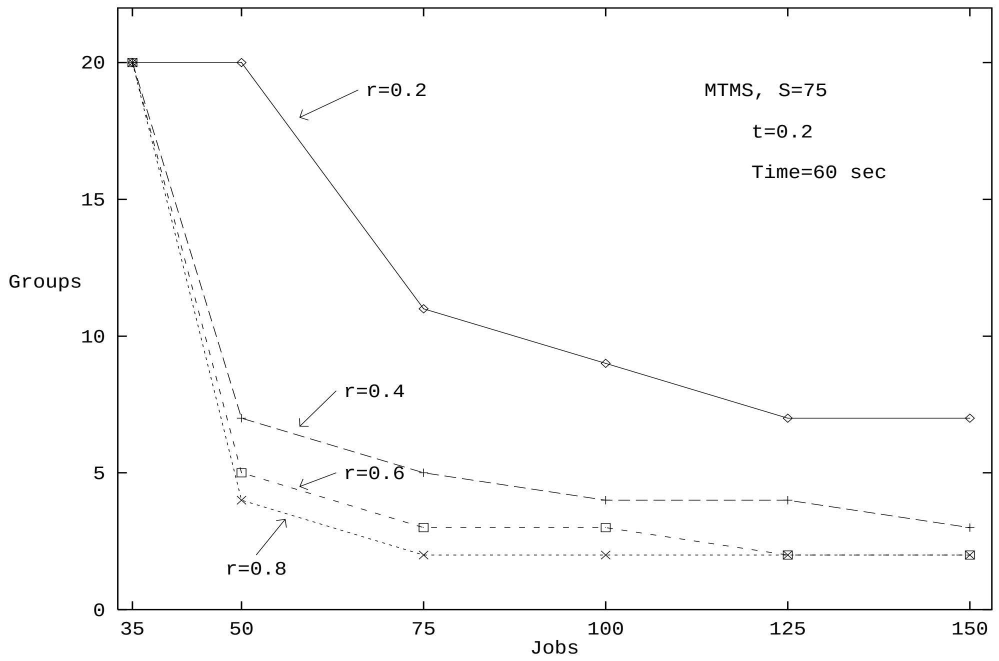
Τα αποτελέσματα του αντίστροφου αλγορίθμου μπορούν να βελτιωθούν σημαντικά με την εφαρμογή μηχανισμού φραγμάτων.

Οπως φαίνεται στο σχήμα 7.3.1 οσο μεγαλύτερη είναι η τιμή του χρόνου εξάρμωσης S τόσο πιό ευκολο είναι το πρόβλημα $MTMS$. Αυτο είναι σύνεπεια της ευρύτερης εφαρμογής του θεωρήματος 3.6.1 για μεγάλες τιμές του S.



Σχήμα 7.3.1 : Επίδραση του χρόνου εξάρμωσης

Επίσης όσο αυξάνει το σχετικό εύρος των προθεσμιών αυξάνει και η δυσκολία του προβλήματος. Στο σχήμα 7.3.2 απεικονίζεται η επίδραση του σχετικού εύρους των προθεσμιών στό πρόβλημα.



Σχήμα 7.3.2 : Επίδραση του σχετικού εύρους προθεσμιών.

7.3.3. Ευθύς αλγόριθμος για το πρόβλημα $MTMS_j$.

Δεν θα δώσουμε πλήρη αποτελέσματα, όπως για την περίπτωση του $MTMS$. Αυτό που θα δείξουμε πειραματικά είναι ότι το πρόβλημα $MTMS_j$ είναι ευκολότερο του προβλήματος $MTMS$ όταν $S = \frac{1}{B}\sum_{j=1}^{B} S_j$. Οι πίνακες 7.3.7 και 7.3.8 δίνουν τις επιδόσεις του αλγορίθμου για το $MTMS$ και $MTMS_j$ αντίστοιχα. Για το $MTMS$ έχουμε διαλέξει τον χρόνο εξάρμωσης ίσο με 50 (όσο και ο μέσος χρόνος επεξεργασίας) ενώ οι χρόνοι εξάρμωσης του $MTMS_j$ παράγονται από την ομοιόμορφη κατανομή στο διάστημα [1,100]. Στο σχήμα 7.3.3 απεικονίζεται η δυσκολία του προβλήματος $MTMS$ σε σχέση με το πρόβλημα $MTMS_j$.

MTMS, S=50						
t	r	Ομάδες (Εργασίες) Λύθηκαν (Χρόνος)				
0.2	0.2	20(47) 10(6.6)	15(44) 10(8.9)	10(45) 10(1.5)	7(47) 10(0.7)	5(50) 10(0.2)
0.2	0.4	20(45) 9(44.7)	15(40) 9(19.4)	10(40) 9(13.5)	7(41) 10(8.1)	5(47) 10(1.1)
0.2	0.6	20(37) 10(11.8)	15(36) 10(20.7)	10(35) 8(17)	7(40) 6(33.1)	5(45) 8(31.1)
0.2	0.8	20(36) 10(34.2)	15(35) 9(29.5)	10(35) 7(37.7)	7(40) 3(24.3)	5(45) 4(88.7)

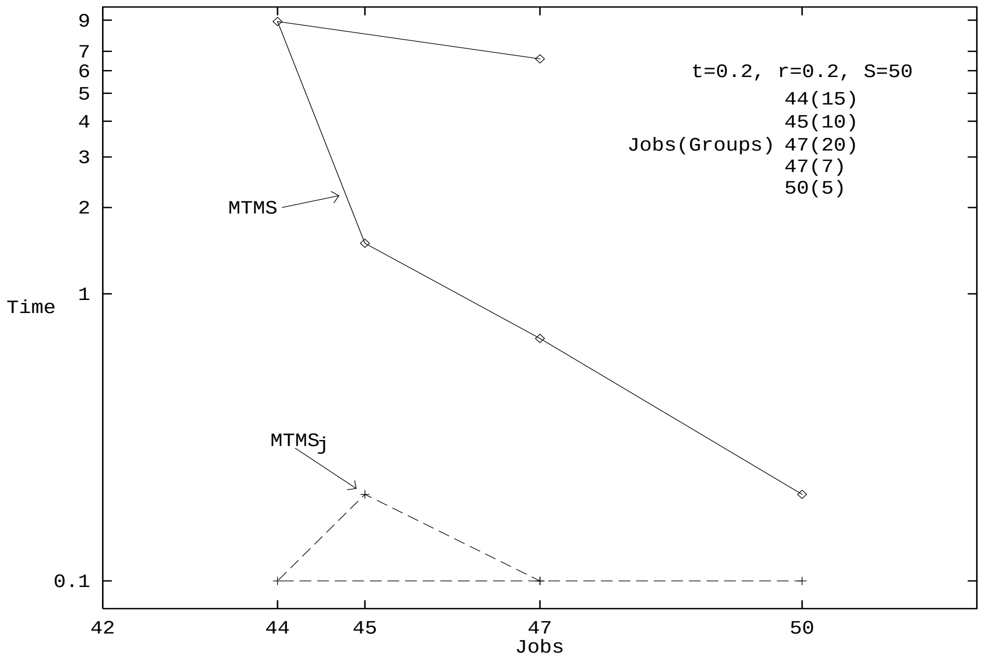
Πίνακας 7.3.7

$MTMS_j,\ S_j\ \tilde{}\ U(1{,}100)$						
t	r	Ομάδες (Εργασίες) Λύθηκαν (Χρόνος)				
0.2	0.2	20(47) 10(0.1)	15(44) 10(0.1)	10(45) 9(0.2)	7(47) 10(0.1)	5(50) 10(0.1)
0.2	0.4	20(45) 10(0.1)	15(40) 10(0.2)	10(40) 10(0.1)	7(41) 10(1.2)	5(47) 9(9)
0.2	0.6	20(37) 10(0.1)	15(36) 10(0.1)	10(35) 10(0.2)	7(40) 8(0.6)	5(45) 8(8.1)
0.2	0.8	20(36) 10(0.1)	15(35) 10(0.1)	10(35) 10(8.9)	7(40) 5(25)	5(45) 4(0.1)

Πίνακας 7.3.8

7.4. Επιδόσεις προσεγγιστικών αλγορίθμων

Οπως έχουμε ήδη αναφέρει μεγάλο πρακτικό ενδιαφέρον παρουσιάζει η εύρεση γρήγορων υποβέλτιστων λύσεων. Στο κεφάλαιο αυτό θα δώσουμε τα αποτελέσματα των επιδόσεων των δύο διαφορετικών υλοποιήσεων του προσεγγιστικού αλγορίθμου διαμερισμού και φραγής.



Σχήμα 7.3.3 : Τα προβλήματα *MTMS* και $MTMS_j$.

7.4.1. Ευθύς προσεγγιστικός αλγόριθμος για το πρόβλημα *MTMS*.

Τα αποτελέσματα αναφέρονται σε προβλήματα *MTMS* με S=5. Για κάθε κατηγορία λύθηκαν 10 προβλήματα που δημιουργήθηκαν με τον τρόπο που περιγράφεται στο κεφ. 7.2. Στο συγκεκριμένο σημείο της εργασίας παραθέτουμε αποτελέσματα μόνο για τέσσερεις κατηγορίες. Τα αποτελέσματα και για τις 16 κατηγορίες υπάρχουν στο παράρτημα Γ της εργασίας.

Κάθε στήλη των πινάκων 7.4.1, 7.4.2, 7.4.3 και 7.4.4 περιγράφει τα αποτελέσματα λύσης 10 προβλημάτων. Σε κάθε πρόβλημα τέθηκε χρονικό όριο εκτέλεσης τεσσάρων ωρών CPU. Η πρώτη συνιστώσα κάθε στήλης δίνει τον αριθμό των ομάδων, τον αριθμό των εργασιών και τον αριθμό των αρχικών παρτίδων για τον προσεγγιστικό αλγόριθμο. Η δεύτερη συνιστώσα δίνει τον αριθμό των προβλημάτων που λύθηκαν. Η τρίτη και τέταρτη συνιστώσα δίνει τον μέσο χρόνο επίλυσης με τον ακριβή και προσεγγιστικό αλγόριθμο αντίστοιχα. Οι τρείς επόμενες συνιστώσες δίνουν την συχνότητα εύρεσης βέλτιστης λύσης, το μέσο απόλυτο σφάλμα και το μέσο σχετικό σφάλμα. Υπενθυμίζουμε ότι τα σφάλματα δεν αναφέρονται στο σύνολο των προβλημάτων που λύθηκαν, αλλά στα προβλήματα που δεν λύθηκαν βέλτιστα.

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)55	5(1000)60
Λύθηκαν	10	10	10	4	5
Μέσος Χρ. CPU Βέλ.	1100	1995	175	113	7
Μέσος Χρ. CPU Πρ.	19	62	35	74	7
Συχνότητα	0.6	0.8	0.7	1	1
Απόλυτο Σφάλμα	7.5	2	6.6	0	0
Σχετικό Σφάλμα	0.015	0.004	0.012	0	0

Πίνακας 7.4.1

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες(Εργ.)Παρτίδες	20(55)40	15(55)35	10(55)35	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	245	752	439	159	16
Μέσος Χρ. CPU Πρ.	23	19	51	36	5
Συχνότητα	0.9	0.6	1	0.8	0.8
Απόλυτο Σφάλμα	21	26.2	0	34	14
Σχετικό Σφάλμα	0.154	0.230	0	0.290	0.246

Πίνακας 7.4.2

Ο προσεγγιστικός αλγόριθμος χαρακτηρίζεται ως ικανοποιητικός αφού

- οι χρόνοι απόκρισης είναι τάξεως λεπτού ,
- με μεγάλη πιθανότητα δίνει βέλτιστη λύση και
- το μέσο απόλυτο και μέσο σχετικό σφάλμα είναι πολύ μικρό.

Ενδιαφέρον παρουσιάζει η μελέτη του προσεγγιστικού αλγορίθμου για μεγάλα προβλήματα (200-500 εργασίες), όμως η τεράστια υπολογιστική πλοκή της εύρεσης βέλτιστης λύσης την καθιστά σχεδόν αδύνατη.

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)55	5(1000)60
Λύθηκαν	10	10	10	4	4
Μέσος Χρ. CPU Βέλ.	843	883	170	97	8
Μέσος Χρ. CPU Πρ.	18	53	37	8	8
Συχνότητα	0.3	0.9	0.7	1	1
Απόλυτο Σφάλμα	11.857	2	5.7	0	0
Σχετικό Σφάλμα	0.009	0.001	0.004	0	0

Πίνακας 7.4.3

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες(Εργ.)Παρτίδες	20(55)40	15(55)35	10(55)35	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	1584	900	220	215	9
Μέσος Χρ. CPU Πρ.	41	16	25	19	5
Συχνότητα	0.8	0.2	0.8	0.7	0.8
Απόλυτο Σφάλμα	3.5	19.2	6	20	5
Σχετικό Σφάλμα	0.005	0.028	0.009	0.026	0.006

Πίνακας 7.4.4

7.4.2. Αντίστροφος προσεγγιστικός αλγόριθμος για το πρόβλημα *MTMS*.

Τα αποτελέσματα του πίνακα 7.4.6 αναφέρονται σε προβλήματα *MTMS* με $S=5$. Για κάθε κατηγορία προβλημάτων έχουν λυθεί 20 ή 30 προβλήματα. Κάθε γραμμή μας δίνει τον αριθμό των προβλημάτων που προσπαθήσαμε να λύσουμε, τον αριθμό των προβλημάτων που λύθηκαν (σε χρονικό όριο τεσσάρων ωρών για κάθε πρόβλημα), τον αριθμό των προβλημάτων που λύθηκαν βέλτιστα, το μέσο απόλυτο σφάλμα, το μέσο σχετικό σφάλμα και την συχνότητα με την οποία ο αλγόριθμος δίνει βέλτιστη λύση. Ο αριθμός των ομάδων και των εργασιών προσδιορίζεται από το σχετικό εύρος των προθεσμιών και δίνονται στον πίνακα 7.4.5. Ο αριθμός των αρχικών παρτίδων για τον αλγόριθμο είναι σταθερός ίσος με 28. Χρόνους απόκρισης δεν δίνουμε. Σε γενικές όμως γραμμές η εύρεση ακριβούς λύσεως ήταν της τάξεως των 2 ώρων, ενώ η εύρεση προσεγγιστικής λύσεως της τάξεως λεπτού. Οι λόγοι που δεν δίνουμε ακριβείς χρόνους είναι οι εξής:

- προβλήματα αυτού του μεγέθους επιλύονται σε μερικά δευτερόλεπτα από τον ευθύ αλγόριθμο και
- οι επιδόσεις του αντίστροφου αλγόριθμου μπορούν να βελτιωθούν σημαντικά με την χρήση μηχανισμού φραγμάτων.

Προβλήματα Δοκιμών Συναρτήσει Σχετικού Εύρους Προθεσμιών	
Σχετικό Εύρος Προθεσμιών	Ομάδες(Εργασίες)Προβλήματα
0.2	10(60)10 7(75)10 7(90)10
0.4	10(45)10 5(65)10
0.6	10(40)10 4(55)10
0.8	5(45)10 10(38)10
0.9	5(45)10 10(38)10
0.98	5(45)10 10(38)10

Πίνακας 7.4.5

Μπορούμε να παρατηρήσουμε ότι

- με μεγάλη πιθανότητα δίνει βέλτιστη λύση και
- το μέσο απόλυτο και μέσο σχετικό σφάλμα είναι πολύ μικρό.

Αξιοσημείωτο είναι πως τα προβλήματα της κατηγορίας t=0.7 και r=0.8 είναι τα πιο δύσκολα με κριτήριο την πιθανότητα εύρεσης βέλτιστης λύσης. Τα προβλήματα της κατηγορίας t=0.6 και r=0.98 είναι τα πιο δύσκολα με κριτήριο το μέσο σχετικό σφάλμα.

Ακόμα αξίζει να παρατηρήσουμε ότι για σταθερό παράγοντα καθυστέρησης, το πρόβλημα γίνεται πιο δύσκολο όσο αυξάνει το σχετικό εύρος των προθεσμιών. Από κάποιο όριο και πέρα η συμπεριφορά αντιστρέφεται. Στις περισσότερες περιπτώσεις το όριο είναι η τιμή r=0.6 σε άλλες όμως το όριο αυτό εμφανίζεται πολύ αργότερα.

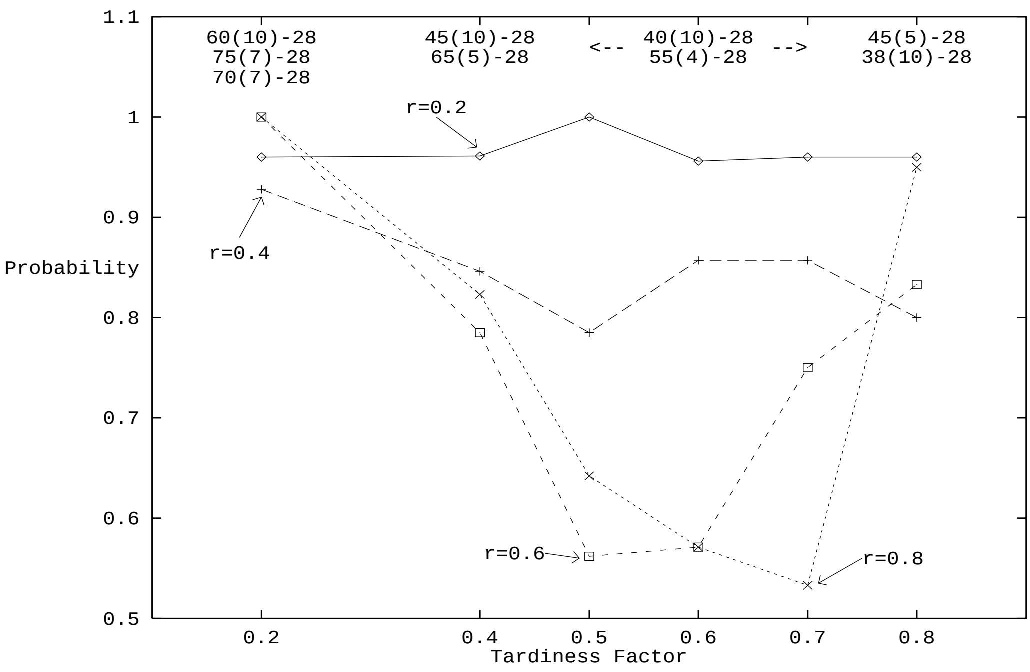
Στο σχήμα 7.4.1 παρατηρούμε οτι ο προσεγγιστικός αλγόριθμος με μεγάλη πιθανότητα δίνει βέλτιστη λύση. Ιδιαίτερα στις περιπτώσεις r=0.2 και r=0.4 που είναι πιό κοντά στα προβλήματα που εμφανίζονται στην πράξη στον

t	r	Προβλ.	Λύθηκαν	Βέλτιστα	Σφάλμα	Σχ. Σφάλμα	Συχνότ.
0.2	0.2	30	25	24	2	0.009	0.960
	0.4	20	14	13	3	0.047	0.928
	0.6	20	14	20	0	0	1.000
	0.8	20	14	20	0	0	1.000
0.4	0.2	30	26	25	2	0.001	0.961
	0.4	20	13	11	4	0.005	0.846
	0.6	20	14	11	15.3	0.044	0.785
	0.8	20	17	14	16.6	0.083	0.823
0.5	0.2	30	26	26	0	0	1.000
	0.4	20	14	11	16.66	0.018	0.785
	0.6	20	16	9	11.14	0.019	0.562
	0.8	20	14	9	13.6	0.041	0.642
0.6	0.2	30	23	22	12	0.007	0.956
	0.4	20	14	12	4	0.003	0.857
	0.6	20	14	10	11.75	0.015	0.714
	0.8	20	14	8	31.66	0.055	0.571
	0.9	20	18	10	29.75	0.058	0.55
	0.98	20	15	9	31.33	0.066	0.600
0.7	0.2	30	25	24	2	0.001	0.960
	0.4	20	14	12	4	0.002	0.857
	0.6	20	16	12	8.5	0.008	0.750
	0.8	20	15	8	21.28	0.028	0.533
0.8	0.2	30	25	24	2	0.0008	0.960
	0.4	20	15	12	16.33	0.009	0.800
	0.6	20	18	15	13	0.010	0.833
	0.8	20	20	19	8	0.007	0.950

Πίνακας 7.4.6

προγραμματισμό παραγωγής.

Πάντως τα προβλήματα που επιλύονται με τον αλγόριθμο είναι σχετικά μικρά. Ενδιαφέρον θα παρουσίαζε η μελέτη μιας βελτιωμένης έκδοσης του αλγορίθμου με μηχανισμό φραγμάτων ανάλογο με εκείνο που περιγράφεται στο κεφ. 4.3.



Σχήμα 7.4.1 : Η πιθανότητα εύρεση βέλτιστης λύσης.

8. Συμπεράσματα, Επεκτάσεις και Εφαρμογή σε συγγενή προβλήματα

Στο κεφάλαιο αυτό δίνουμε τα συμπεράσματά μας καθώς και μερικά στοιχεία για την γενίκευση των προτεινόμενων αλγορίθμων. Επίσης παρουσιάζεται ένα παρεμφερές πρόβλημα χρονικού προγραμματισμού του οποίου η λύση ανάγεται στην λύση του $MTMS_j$.

8.1. Συμπεράσματα

Στην εργασία αυτή παρουσιάζουμε ένα νέο αλγόριθμο διαμερισμού και φραγής για την επίλυση του προβλήματος της μέγιστης καθυστέρησης με χρόνους εξάρμωσης ανεξάρτητους διαδοχής. Ο αλγόριθμος κατορθώνει και λύνει προβλήματα μέχρι και 60 εργασιών σε χρόνο ενός λεπτού. Ενδιαφέρον είναι το γεγονός πως τα προβλήματα που παρουσιάζονται στην πράξη ανήκουν στην κατηγορία με παράγοντα καθυστέρησης 0.1 και σχετικό εύρος προθεσμιών 0.1. Μέσα σε χρόνο ενός λεπτού ο αλγόριθμος κατορθώνει και λύνει προβλήματα αυτής της κατηγορίας που έχουν ως και 200 εργασίες. Ο αλγόριθμος υλοποιεί μια γρήγορη απαρίθμηση ενός μεγάλου συνόλου πιθανών λύσεων. Ο εκθετικός χαρακτήρας του ελέγχεται από ένα απλό μηχανισμό φραγμάτων καθώς και από άλλους μηχανισμούς απόρριψης.

Ο προσεγγιστικός αλγόριθμος για τα προβλήματα που συναντώνται στην πράξη κατορθώνει και δίνει βέλτιστη λύση με πολύ μεγάλη πιθανότητα. Στις ελάχιστες περιπτώσεις (0.1) που δεν δίνει βελτιστη λύση το σχετικό σφάλμα δεν ξεπερνά το 2%. Ακόμα αξίζει να παρατηρήσουμε πως οι χρόνοι επεξεργασίας ανήκουν στο διάστημα [1,100] και όχι στο [1,10] όπως σε πολλές δημοσιεύσεις. Οταν οι χρόνοι επεξεργασίας είναι μικροί τότε ο μηχανισμός φραγμάτων δουλεύει καλύτερα, καθως και τα δεσίματα απλών παρτίδων δημιουργούν σύνθετες παρτίδες με προθεσμίες όχι πολύ διαφορετικές από αυτές των απλών. Τελικό αποτέλεσμα είναι ο σημαντικός περιορισμός του χώρου έρευνας.

Τέλος, ο αλγόριθμος που δημιουργεί το βέλτιστο πρόγραμμα από το τέλος προς την αρχή μπορεί να χρησιμοποιηθεί για την επίλυση δικριτηριακών προβλημάτων χρονικού προγραμματισμού. Φυσικά στην παρούσα εργασία δεν αποδεικνύονται οι μηχανισμοί απόρριψης και γι’ αυτόν τον αλγόριθμο αλλά μπορούν να προκύψουν εύκολα με ανάλογες αποδείξεις.

Ενδιαφέρον παρουσιάζουν μερικές παρατηρήσεις που απορρέουν από τα πειραματικά δεδομένα. Ο ακριβής αλγόριθμος όταν δεν κάνει χρήση του μηχανισμού φραγμάτων παρουσιάζει συμπεριφορά ανεξάρτητη από τον παράγοντα καθυστέρησης. Αυτό έχει και θεωρητική βάση διότι κάθε πρόβλημα το ολισθαίνουμε δεξιά και το επιλύουμε την χρονική στιγμη $d_{\max} = \max_{\substack{1 \le b \le B \\ 1 \le i \le N_b^{Crit}}} \{d\left[J_{ib}^{Crit}\right]\}$. Συνεπώς κάθε πρόβλημα εμφανίζεται σαν να έχει παράγοντα καθυστέρησης 1. Οταν κάνουμε χρήση του μηχανισμού φραγμάτων τότε η παραπάνω παρατήρηση δεν ισχύει μια και η τιμή του κάτω ορίου επηρεάζει σημαντικά την εξέλιξη του αλγορίθμου και φυσικά εξαρτάται από τον παράγοντα καθυστέρησης.

Η επόμενη σημαντική παρατήρηση σχετίζεται με τις τιμές των χρόνων εξάρμωσης. Εστω (Π_1) μια περίπτωση του προβλήματος $MTMS_j$ με $S_0 = \frac{1}{B}\sum_{b=1}^{B} S_b$. Το πρόβλημα ($\Pi_2$) που προκύπτει από το ($\Pi_1$) αν θέσουμε $S_b = S_0$, $1 \le b \le B$ είναι *MTMS* πρόβλημα και εμφανίζεται πιο δύσκολο από το (Π_1). Το πειραματικό αποτέλεσμα φαίνεται αρκετά παράδοξο καθώς το πρόβλημα με ίσους χρόνους εξάρμωσης εμφανίζεται δυσκολότερο από το πρόβλημα με διαφορετικούς χρόνους εξάρμωσης. Παρ’ όλ’ αυτά η παρατήρηση αυτή συμφωνεί και με τις αντίστοιχες διαπιστώσεις των E.Uskup και S.B.Smith [UsSm-75]. Επίσης σύμφωνα με τις παρατηρήσεις τους, το πρόβλημα που επιλύουν (παρόμοιο με το $MTMS_{ij}$ σε δύο μηχανές) δυσκολεύει αν υπάρχουν παρόμοια τμήματα στον πίνακα των χρόνων εξάρμωσης, πράγμα που σημαίνει πως όσο πιο μεγάλος είναι ο αριθμός των εργασιών ανά κατηγορία τόσο πιο δύσκολο είναι και το πρόβλημα. Η παρατήρηση αυτή έχει θεωρητική υπόσταση όπως φαίνεται άμεσα από το σχήμα 2.2.1 που παριστάνει τα θεωρητικά αποτελέσματα των Bruno και Downey [BrDo-78].

Σε γενικές γραμμές το μέγεθος των προβλημάτων που επιλύεται είναι μεγάλο και ο προσεγγιστικός αλγόριθμος μπορεί να λειτουργήσει ικανοποιητικά και για τα προβλήματα που εμφανίζονται στην πράξη (20-200 εργασίες και 3-20 κατηγορίες).

8.2. Επεκτάσεις του αλγορίθμου

Είναι προφανές πως ο αλγόριθμος για το πρόβλημα $MTMS_j$ μπορεί να γενικευθεί ώστε να λύνει το πρόβλημα $MTMS_{ij}$ με χρόνους εξάρμωσης που ικανοποιούν την τριγωνική ανισότητα. Το θεώρημα για την δημιουργία των απλών παρτίδων, ο μηχανισμός φραγμάτων καθώς και το θεώρημα 3.6.1 εξακολουθούν να ισχύουν. Παρ' όλ' αυτά αν κάποιος μας δώσει ένα σύνολο βέλτιστων παρτίδων δεν είναι σίγουρο πως η τοποθέτηση τους με βάση τον EDD θα οδηγήσει σε βέλτιστο πρόγραμμα.

Ετσι αν θεωρήσουμε πως λύνουμε το πρόβλημα $MTMS_{ij}$ αυξάνοντας ημιτελή προγράμματα από την αρχή πρός το τέλος τότε για την πρώτη θέση του βέλτιστου προγράμματος είναι υποψήφιες B εργασίες, από κάθε ομάδα η έχουσα την μικρότερη προθεσμία. Συνεπώς, σε κάθε βήμα, από κάθε πρόβλημα δημιουργούνται $2B$ υποπροβλήματα. Αν ο αριθμός των ομάδων είναι μικρός τότε ο αλγόριθμος θα έχει πρακτική εφαρμογή. Επίσης αντίστοιχος προσεγγιστικός αλγόριθμος μπορεί να προκύψει με τρόπο ανάλογο με αυτόν που περιγράφεται στο κεφ. 5.1. Στις περισσότερες από τις εργασίες που αναφέρονται στην βιβλιογραφία ([UsSm-75], [Gl-67]) λύνεται μια ειδική περίπτωση του $MTMS_{ij}$ με $N_b=1$, $1 \le b \le B$. Η μη ύπαρξη ομοίων τμημάτων στον πίνακα των χρόνων εξάρμωσης λειτουργεί υπέρ της στενότητας του κάτω φράγματος. Ο αλγόριθμος για το $MTMS_{ij}$ μπορεί να λύσει και το πρόβλημα $N/1/seq-dep/C_{\max}|T_{\max}=0$ κατ' αναλογία με την επίλυση του $N/1/seq-indep/C_{\max}|T_{\max}=0$ που περιγράφεται στο επόμενο κεφάλαιο.

8.3. Εφαρμογή στην λύση συγγενών προβλημάτων

Πολύ κοντά στο πρόβλημα $MTMS_{ij}$ βρίσκεται το πρόβλημα της εύρεσης προγράμματος ελάχιστης συνολικής διάρκειας με προθεσμίες. Πάλι έχουμε εργασίες που κατανέμονται σε ομάδες και το ζητούμενο είναι να βρούμε μια διάταξη εργασιών που να είναι εφικτή ($\max\limits_{\substack{1\le b\le B\\ 1\le i\le N_b}}\{T_{ib}\}=0$) και επιπλέον ελαχιστοποιεί την ποσότητα $\max\limits_{\substack{1\le b\le B\\ 1\le i\le N_b}}\{C_{ib}\}$.

Θεωρούμε το αντίστοιχο πρόβλημα απόφασης που έχει ως εξής:

έχουμε N εργασίες που κατανέμονται σε B ομάδες και κάθε εργασία έχει χρόνο επεξεργασίας p_{ib} και προθεσμία d_{ib}. Χρόνος εξάρμωσης S_{ij} απαιτείται για την προσαρμογή της μιας και μόνης μηχανής προκειμένου να εκτελέσει εργασία από την ομάδα j ενώ την προηγούμενη στιγμή εκτελούσε εργασία από την ομάδα i. Επίσης δίνεται ένας μη αρνητικός ακέραιος K. Το θεμα είναι αν υπάρχει εφικτή διάταξη που να έχει χρόνο αποπεράτωσης μικρότερο ή ίσο του K.

Θα δείξουμε ότι η λύση του παραπάνω προβλήματος μπορεί να αναχθεί στην λύση μιας περίπτωσης του προβλήματος εύρεσης εφικτού προγράμματος ή ισοδύναμα στην επίλυση του $MTMS_{ij}$.

Για κάθε εργασία J_{ib} θέτουμε $d'_{ib} = \min(K, d_{ib})$ και λύνω το πρόβλημα εύρεσης εφικτού προγράμματος με τις νέες προθεσμίες. Αν υπάρχει εφικτή διάταξη τότε η διάταξη αυτή είναι και λύση για το πρόβλημα του προγράμματος ελάχιστης συνολικής διάρκειας με προθεσμίες. Αυτό γιατί για την τυχαία εργασία J_{ib} στην εφικτή διάταξη έχουμε:

$$T'_{ib} = 0 \Rightarrow \max(0, C_{ib} - d'_{ib}) = 0 \Rightarrow 0 \geq C_{ib} - d'_{ib} \Rightarrow$$

$$0 \geq C_{ib} - \min(K, d_{ib}) \Rightarrow \min(K, d_{ib}) \geq C_{ib} \Rightarrow K \geq C_{ib}$$

Δηλαδή, κάθε εργασία έχει χρόνο αποπεράτωσης μικρότερο του K. Αντίθετα αν υποθέσουμε ότι δεν υπάρχει εφικτή διάταξη για το πρόβλημα εύρεσης εφικτού προγράμματος τότε αν το πρόβλημα του προγράμματος ελάχιστης συνολικής διάρκειας με προθεσμίες έχει λύση θα έχουμε:

$$T_{ib} = 0 \Rightarrow \max(0, C_{ib} - d_{ib}) = 0 \Rightarrow C_{ib} - d_{ib} \leq 0 \Rightarrow C_{ib} \leq d_{ib}$$

Επίσης έχουμε $C_{ib} \leq K$ οπότε:

$$\left.\begin{matrix} C_{ib} \leq d_{ib} \\ C_{ib} \leq K \end{matrix}\right\} \Rightarrow C_{ib} \leq \min(K, d_{ib}) \Rightarrow C_{ib} - \min(K, d_{ib}) \leq 0 \Rightarrow$$

$$\max(0, C_{ib} - \min(K, d_{ib})) = 0 \Rightarrow T'_{ib} = 0$$

πράγμα άτοπο αφού υποθέσαμε πως το πρόβλημα εύρεσης εφικτού προγράμματος δεν έχει λύση.

Εκτός από το πρόβλημα εύρεσης προγράμματος ελάχιστης συνολικής διάρκειας με προθεσμίες η λύση του $MTMS_{ij}$ επιδρά και σε άλλα προβλήματα όπως το πρόβλημα της ελαχιστοποίησης της μέσης καθυστέρησης, του αριθμού των εργασιών που καθυστερούν, το πρόβλημα εφικτότητας με δύο μηχανές [Sa-70] κ.α.

Το πρόβλημα της ελαχιστοποίησης της μέσης καθυστέρησης $(n/1//\frac{1}{n}\sum_i T_i)$ είναι ισοδύναμο με το πρόβλημα της ελαχιστοποίησης του αθροίσματος των καθυστερήσεων $(n/1//\sum_i T_i)$. Στην περίπτωση που δεν έχουμε χρόνους εξάρμωσης το πρόβλημα είναι NP-hard ([Em-69], [GTW-88], [PoWa-85], [KaKy-88]) και η λύση του κινείται στα πλαίσια της λύσης του προβλήματος της ελαχιστοποίησης της μέγιστης καθυστέρησης $(n/1//T_{\max})$ και του προβλήματος της ελαχιστοποίησης του αθροίσματος των χρόνων περάτωσης ($n/1//\sum_i C_i$). Ενας απλός ευρηματικός κανόνας MDD (Modified due date) [Si] συνδιάζει δυναμικά τις βέλτιστες λύσεις του $n/1//T_{\max}$ και του $n/1//\sum_i C_i$. Αντίστοιχα μπορεί να αντιμετωπισθεί και η περίπτωση που έχουμε χρόνους εξάρμωσης με την μόνη διαφορά ότι τώρα θέλουμε τις λύσεις των προβλημάτων $n/1/seq-indep/T_{\max}$ και $n/1/seq-indep/\sum_i C_i$. Το πρώτο πρόβλημα είναι το $MTMS_j$.

Το πρόβλημα της ελαχιστοποίησης του αριθμού των εργασιών που καθυστερούν όταν δεν έχουμε χρόνους εξάρμωσης λύνεται με την συνεχή επίλυση του προβλήματος εύρεσης προγράμματος ελάχιστης συνολικής διάρκειας με προθεσμίες ($n/1//C_{\max}|T_{\max}$=0). Αντίστοιχα μπορούμε να αντιμετωπίσουμε την περίπτωση που έχουμε χρόνους εξάρμωσης με την μόνη διαφορά ότι τώρα θέλουμε λύση του προβλήματος $n/1/seq-indep/C_{\max}|T_{\max}=0$ που όπως δείξαμε η λύση του ανάγεται στην λύση του $MTMS_{ij}$.

Βιβλιογραφία

[Ba-75]

Baker, K. R., "A Comparative Study of Flow-Shop Algorithms," Opns. Res., vol. 23, pp. 62-73, 1975.

[BaVa-81]

Barnes, J. W. and L. K. Vanston, "Scheduling Jobs with Linear Delay Penalties and Sequence Dependent Setup Costs," Opns. Res., vol. 29, pp. 146-160, 1981.

[BLK-83]

Blazewicz, J., J. K. Lenstra, and A. H. G. Rinnooy Kan, "Scheduling Subject to Resource Constraints: Classification and Complexity," Ann. Discrete Math., vol. 5, pp. 11-24, 1983.

[BrDo-78]

Bruno, J. and P. Downey, "Complexity of Task Sequencing with Deadlines, Set-Up Times and Changeover Costs," SIAM J. COMPUT., vol. 7, pp. 393-404, 1978.

[BCS-74]

Bruno, J., E. G. Coffman Jr., and R. Sethi, "Scheduling Independent Tasks to Reduce Mean Finishing Time," Association for Computing Machinery, vol. 17, pp. 382-387, 1974.

[DiSe-88]

Dileepan, P. and T. Sen, "Bicriterion Static Scheduling Research for a Single Machine," Mgmt. Sci., vol. 16, pp. 53-59, 1988.

[Em-69]

Emmons, H., "One-Machine Sequencing to Minimize Certain Functions of Job Tardiness," Opns. Res., vol. 17, pp. 701-715, 1969.

[FaSc-87]

Faaland, B. and T. Schmitt, "Scheduling Tasks with Due Dates in a Fabrication/Assembly process," Opns. Res., vol. 35, pp. 378-388, 1987.

[FLL-83]

Fisher, M. L., B. J. Lageweg, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Surrogate Duality Relaxation for Job Shop Scheduling," Ann. Discrete Math., vol. 5, pp. 65-75, 1983.

[Fo-78]

Fox, B. L., "Data Structures and Computer Science Techniques in Operations Research," Opns. Res., vol. 26, pp. 686-717, 1978.

[GTW-88]

Garey, M. R., R. E. Tarjan, and G. T. Wilfong, "One-Processor Scheduling with Symmetric Earliness and Tardiness Penalties," Opns. Res., vol. 13, pp. 330-348, 1988.

[GaLe-88]

Gascon, A. and R. C. Leachman, "A Dynamic Programming Solution to the Dynamic, Multi-Item, Single-Machine Scheduling Problem," Opns. Res., vol. 36, pp. 50-56, 1988.

[Gl-67]

Glassey, C. R., Minimum Change-Over Scheduling of Several Products on One Machine, 1967.

[Gr-81]

Graves, S. C., A Review of Production Scheduling, 1981.

[GuBa]

Guida, M. and G. Basaglia, Integrating Operational Research and Artificial Intelligence in a Distributed Approach to Dynamic Scheduling : the B.I.S. Project.

[HaPo-83]

Hariri, A. M. A. and C. N. Potts, "An Algorithm for Single Machine Sequencing with Realease Dates to Minimize Total Weighted Completion Time," Ann. Discrete Math., vol. 5, pp. 99-109, 1983.

[HeKi-89]

Hendry, L. C. and B. G. Kingsman, "Production planning systems and their applicability to make-to-order companies," Operational Research, vol. 40, pp. 1-15, 1989.

[IgSc-65]

Ignall, E. and L. Schrage, "Application of the Branch and Bound Technique to some Flow-Shop Scheduling Problems," Opns. Res., vol. 13, pp. 400-412, 1965.

[Ja-55]

Jackson, J. R., "Scheduling a Production Line to Minimize Maximum Tardiness," Management Sciences Research Report 43, University of California at Los Angeles, 1955.

[Jo]

Johnson, S. M., Optimal Two- and Three-Stage Production Schedules with Setup Times Included, pp. 61-68.

[KLL-74]

Kan, A. H. G. Rinnooy, B. J. Lageweg, and J. K. Lenstra, Minimizing Total Costs in One-Machine Scheduling, 1974.

[Ka-80]

Kao, E. P. C., "A Multiple Objective Decision Theoretic Aproach to One-Machine Scheduling Problems," Opns. Res., vol. 7, pp. 251-259, 1980.

[KaKy-88]

Kawaguchi, T. and S. Kyan, "Deterministic Scheduling in Computer Systems: A Survey," Opns. Res., vol. 31, pp. 190-216, 1988.

[La-76]

Lawler, E. L., "Optimal Sequencing of a Single Machine subject to Precedence Constraints," Mgmt. Sci., vol. 19, pp. 544-546, 1976.

[LePa-81]

Lewis, H. R. and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1981.

[LiBu-88]

Liu, C. Y. and R. L. Bulfin, "Scheduling Open Shops with Unit Execution Times to Minimize Functions of Due Dates," Opns. Res., vol. 36, pp. 553-559, 1988.

[LoMu-71]

Lockett, A. G. and A. P. Muhlemann, "A Scheduling Problem Involving Sequence Dependent Changeover Times," Technical Notes, 1971.

[MSS-87]

Matsuo, H., J. S. Shang, and R. S. Sullivan, Crane Scheduling Problem in a Computer-Integrated Manufacturing Environment, 1987.

[Ma-88]

Matsuo, H., "The Weighted Total Tardiness Problem with Fixed Shipping Times and Overtime Utilization," Opns. Res., vol. 36, pp. 293-307, 1988.

[MIH]

Meyer, W., R. Isenberg, and M. Hubner, "Knowledge-based factory supervision - The CIM shell," Int. J. Computer Integrated Manufacturing, vol. 1, pp. 31-43.

[Mi-68]

Mitten, L. G., "Branch-and-Bound Methods: General Formulation and Properties," Opns. Res., vol. 18, pp. 24-34, 1968.

[MoPo-89]

Monma, C. L. and C. N. Potts, "On the Complexity of Scheduling with Batch Setup Times," Opns. Res., vol. 37, pp. 798-804, 1989.

[Mo-79]

Monma, C. L., Sequencing to Minimize the Maximum Job Cost, 1979.

[MoMa-76]

Morin, T. L. and R. E. Marsten, "Branch-and-Bound Strategies for Dynamic Programming," Opns. Res., vol. 24, pp. 611-627, 1976.

[Pa-77]

Panwalkar, S. S., "A Survey of Scheduling Rules," Opns. Res., vol. 25, pp. 45-61, 1977.

[PaSt-82]

Papadimitriou, C. H. and K. Steiglitz, Combinatorial Optimization : Algorithms and Complexity, Prentice-Hall, 1982.

[PoWa-88]

Potts, C. N. and L. N. Van Wassenhove, "Algorithms for Scheduling a Single Machine to Minimize the Weighted Number of Late Jobs," Management Sci., vol. 34, pp. 843-858, 1988.

[PoWa-85]

Potts, C. N. and L. N. Van Wassenhove, "A Branch and Bound Algorith for the Total Weighted Tardiness Problem," Opns. Res., vol. 33, pp. 363-377, 1985.

[Ps-80]

Psaraftis, H. N., "A Dynamic Programming Approach for Sequencing Groups of Identical Jobs," Opns. Res., vol. 28, pp. 1347-1359, 1980.

[Sa-70]

Sahney, V. K., Single-Server, Two-Machine Sequencing with Switching Time, 1970.

[Sa-76]

Sahni, S. K., "Algorithms for Scheduling Independent Tasks," Association for Computing Machinery, vol. 23, pp. 116-127, 1976.

[ShBu-82]

Shanthikumar, J. G. and J. A. Buzacott, "On the Use of Decomposition Approaches in a Single Machine Scheduling Problem," Opns. Res., vol. 25, pp. 29-47, 1982.

[Si-75]

Sidney, J. B., "Decomposition Algorithms for Single-Machine Sequencing with Precedence Relations and Deferral Costs," Opns. Res., vol. 23, pp. 283-298, 1975.

[Si]

Sidney, J. B., An Extension of Moore’s Due Date Algorithm.

[UsSm-75]

Uskup, E. and S. B. Smith, "A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems," Opns. Res., vol. 23, pp. 118-136, 1975.

[Vi-80]

Vickson, R. G., "Choosing the Job Sequence and Processing Times to Minimize Total Processing Plus Flow Cost on a Single Machine," Opns. Res., vol. 28, pp. 1155-1167, 1980.

Παράρτημα Α.

Αποτελέσματα για το *MTMS* με τον ευθύ αλγόριθμο.

Παράγοντας Καθυστέρησης = 0.2	Σχετικό Εύρος Προθεσμιών = 0.2				
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	9	8	8	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(29)87	9(30.1)92	2(41.2)128	30(64.2)86	1(3.3)17
Μέσος Αρ. Παρτίδων	44	42.7	39.7	51.5	43.1
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96	79	54.6	37.7	26.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.23	0.19	0.12	0.05	0.01
Μέση Καθυστέρηση	418.1	417.2	447.4	795.1	2527.4

Παράγοντας Καθυστέρησης = 0.2	Σχετικό Εύρος Προθεσμιών = 0.4				
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	9	8	10	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(8.9)32	1(22.9)80	1(21.9)70	1(8.3)35	1(10.1)37
Μέσος Αρ. Παρτίδων	38.4	37.9	34.7	32	33.5
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96.7	79.3	60.7	43	36.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.8	0.7	0.7	0.6	0.7
Μέση Καθυστέρηση	122.4	106.7	88.7	77.6	58.4

Παράγοντας Καθυστέρησης = 0.2	Σχετικό Εύρος Προθεσμιών = 0.6				
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(45)	5(45)
Λύθηκαν	8	6	5	9	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	1(31.5)136	3(59.2)151	1(26.4)82	1(36.8)129	1(16.8)77
Μέσος Αρ. Παρτίδων	40.7	39.8	37	33.9	31.1
Μέση Απόλ. Απόσταση Κάτω Φράγματος	-73	-86.2	-115.4	-111	-124.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	1	1	1	1	1
Μέση Καθυστέρηση	-73	-86.2	-115.4	-111	-124.6

Παράγοντας Καθυστέρησης = 0.2		Σχετικό Εύρος Προθεσμιών = 0.8			
Ομάδες (Εργασίες)	20(40)	15(40)	10(40)	7(40)	5(40)
Λύθηκαν	10	8	7	8	9
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	3(18.3)82	3(26.2)96	9(35.3)65	2(20.9)66	1(7.4)21
Μέσος Αρ. Παρτίδων	38.3	36.1	35.8	31.7	31
Μέση Απόλ. Απόσταση Κάτω Φράγματος	-197.1	-230.1	-217	-258.2	-244.2
Μέση Σχετ. Απόσταση Κάτω Φράγματος	1	1	1	1	1
Μέση Καθυστέρηση	-197.1	-230.1	-217	-258.2	-244.2

Παράγοντας Καθυστέρησης = 0.4		Σχετικό Εύρος Προθεσμιών = 0.2			
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	8	7	7	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(25.1)54	11(32)91	2(34.6)68	29(64.75)84	1(5.8)25
Μέσος Αρ. Παρτίδων	44.6	42.3	39.6	51.5	45.3
Μέση Απόλ. Απόσταση Κάτω Φράγματος	97.7	78.1	53.8	38.4	26.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.09	0.07	0.04	0.02	0.003
Μέση Καθυστέρηση	1035.1	1061.7	1202.4	2287.2	7499

Παράγοντας Καθυστέρησης = 0.4		Σχετικό Εύρος Προθεσμιών = 0.4			
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	9	10	9	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(21)76	1(12.2)46	2(28.5)112	1(11.1)35	1(10.5)37
Μέσος Αρ. Παρτίδων	39.4	37	34.4	33	33.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96.1	77.8	61.4	45.5	35.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.16	0.14	0.11	0.08	0.05
Μέση Καθυστέρηση	585.9	565.3	551.2	582.5	719.9

Παράγοντας Καθυστέρησης = 0.4	Σχετικό Εύρος Προθεσμιών = 0.6				
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(45)	5(45)
Λύθηκαν	5	5	4	8	9
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	3(22.2)69	8(97)212	7(28.2)62	2(43.4)120	1(15.5)65
Μέσος Αρ. Παρτίδων	39.8	39	37	32	29.7
Μέση Απόλ. Απόσταση Κάτω Φράγματος	99	85	61.2	50	38.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.25	0.23	0.17	0.16	0.12
Μέση Καθυστέρηση	395	381	358.2	320.5	317.7

Παράγοντας Καθυστέρησης = 0.4	Σχετικό Εύρος Προθεσμιών = 0.8				
Ομάδες (Εργασίες)	20(40)	15(40)	10(40)	7(40)	5(40)
Λύθηκαν	9	6	3	4	8
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	5(36)89	2(62.5)129	22(64)92	9(68.7)159	1(20.2)110
Μέσος Αρ. Παρτίδων	37.4	35.5	32	31.2	26.9
Μέση Απόλ. Απόσταση Κάτω Φράγματος	101.3	89.7	71.3	50	42.2
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.57	0.54	0.6	0.44	0.39
Μέση Καθυστέρηση	186.9	180.2	123.7	135.5	120.5

Παράγοντας Καθυστέρησης = 0.6	Σχετικό Εύρος Προθεσμιών = 0.2				
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	8	7	7	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	5(48.7)108	9(29.1)90	6(41.4)125	29(64.4)84	1(1.4)3
Μέσος Αρ. Παρτίδων	46.1	42.7	40.7	51.5	40.4
Μέση Απόλ. Απόσταση Κάτω Φράγματος	97.7	78.8	52.4	38.4	26
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.06	0.05	0.03	0.01	0.002
Μέση Καθυστέρηση	1668	1716.6	1957.8	3779.6	12476.5

Παράγοντας Καθυστέρησης = 0.6		Σχετικό Εύρος Προθεσμιών = 0.4			
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	9	10	9	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(18.7)76	1(16.7)47	1(30.1)112	1(11.1)35	1(10.5)37
Μέσος Αρ. Παρτίδων	39.3	37.6	34.4	33	33.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	95.2	78.2	60.6	45.6	35.6
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.09	0.08	0.06	0.04	0.03
Μέση Καθυστέρηση	1039	1014.9	1004.4	1095.1	1380.5

Παράγοντας Καθυστέρησης = 0.6		Σχετικό Εύρος Προθεσμιών = 0.6			
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(45)	5(45)
Λύθηκαν	7	5	4	7	8
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	3(44.6)126	8(96.2)208	7(28.2)62	2(40.7)96	1(9.4)29
Μέσος Αρ. Παρτίδων	40.4	39	37	31.6	29
Μέση Απόλ. Απόσταση Κάτω Φράγματος	101.7	85	61.2	51	37.9
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.1	0.1	0.07	0.7	0.05
Μέση Καθυστέρηση	869.4	857.6	842	776.8	784

Παράγοντας Καθυστέρησης = 0.6		Σχετικό Εύρος Προθεσμιών = 0.8			
Ομάδες (Εργασίες)	20(40)	15(40)	10(40)	7(40)	5(40)
Λύθηκαν	8	4	4	5	7
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	11(74.7)157	22(207)396	51(142.7)225	17(118.2)252	1(83.6)249
Μέσος Αρ. Παρτίδων	37.7	35.5	34.5	30.8	29.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	98.5	78	62.7	50.8	42.1
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.2	0.1	0.1	0.1	0.08
Μέση Καθυστέρηση	590.5	594.2	579	534	530.1

Παράγοντας Καθυστέρησης = 0.8		Σχετικό Εύρος Προθεσμιών = 0.2			
Ομάδες (Εργασίες)	20(60)	15(65)	10(75)	7(150)	5(500)
Λύθηκαν	9	8	8	8	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	2(28.2)85	9(29.7)90	2(40.4)125	30(63.2)84	1(3.2)17
Μέσος Αρ. Παρτίδων	44	42.7	39.7	51.5	43.1
Μέση Απόλ. Απόσταση Κάτω Φράγματος	96	79	54.6	37.7	26.4
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.04	0.03	0.02	0.007	0.001
Μέση Καθυστέρηση	2271.7	2373.2	2731.2	5272.2	17449.5

Παράγοντας Καθυστέρησης = 0.8		Σχετικό Εύρος Προθεσμιών = 0.4			
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(50)	5(65)
Λύθηκαν	10	10	10	9	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	1(21.7)77	1(19)124	1(16.2)51	1(11.1)40	1(8.5)37
Μέσος Αρ. Παρτίδων	39.7	37.7	34.3	31.6	32.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	95.2	79.7	58.8	47.2	35
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.06	0.05	0.04	0.03	0.02
Μέση Καθυστέρηση	1487.4	1471.9	1451	1616.1	2051.7

Παράγοντας Καθυστέρησης = 0.8		Σχετικό Εύρος Προθεσμιών = 0.6			
Ομάδες (Εργασίες)	20(45)	15(45)	10(45)	7(45)	5(45)
Λύθηκαν	6	6	5	10	10
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	8(25.8)46	12(92.8)170	12(23.6)36	10(61.3)198	1(5.7)25
Μέσος Αρ. Παρτίδων	39	37.7	33.2	32	27.6
Μέση Απόλ. Απόσταση Κάτω Φράγματος	95	88.8	67	51.1	37.5
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.07	0.07	0.05	0.04	0.03
Μέση Καθυστέρηση	1339.2	1298.3	1261.6	1238.6	1225

Παράγοντας Καθυστέρησης = 0.8			Σχετικό Εύρος Προθεσμιών = 0.8		
Ομάδες (Εργασίες)	20(40)	15(40)	10(40)	7(40)	5(40)
Λύθηκαν	9	6	5	8	9
Χρόνος CPU Ελάχ.(Μέσος)Μέγ.	4(42.4)95	1(56.5)127	2(108)143	6(41.4)134	1(29.9)79
Μέσος Αρ. Παρτίδων	35.9	33	31.2	27.5	25.1
Μέση Απόλ. Απόσταση Κάτω Φράγματος	101.3	81.8	65.2	53.7	40.3
Μέση Σχετ. Απόσταση Κάτω Φράγματος	0.1	0.08	0.06	0.05	0.04
Μέση Καθυστέρηση	1047.4	1056	1032.4	1039.7	982.8

Παράρτημα Β.

Αποτελέσματα για το *MTMS* με τον αντίστροφο αλγόριθμο.

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.2			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	8(50)	6(75)	5(100)	5(125)	5(150)
25	20(35)	10(50)	8(75)	6(100)	6(125)	5(150)
50	20(35)	20(50)	10(75)	8(100)	7(125)	6(150)
75	20(35)	20(50)	11(75)	9(100)	7(125)	7(150)

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.4			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	5(50)	3(75)	3(100)	3(125)	2(150)
25	20(35)	5(50)	3(75)	3(100)	3(125)	3(150)
50	20(35)	6(50)	4(75)	4(100)	3(125)	3(150)
75	20(35)	7(50)	5(75)	4(100)	4(125)	3(150)

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.6			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	2(50)	2(75)	2(100)	2(125)	2(150)
25	20(35)	3(50)	3(75)	2(100)	2(125)	2(150)
50	20(35)	4(50)	3(75)	3(100)	2(125)	2(150)
75	20(35)	5(50)	3(75)	3(100)	2(125)	2(150)

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.8			
Χρόνος Εξάρ.	Ομάδες (Εργασίες)					
5	20(35)	2(50)	2(75)	2(100)	1(125)	1(150)
25	20(35)	3(50)	2(75)	2(100)	1(125)	1(150)
50	20(35)	3(50)	2(75)	2(100)	2(125)	1(150)
75	20(35)	4(50)	2(75)	2(100)	2(125)	2(150)

Παράρτημα Γ.

Αποτελέσματα για το *MTMS* με τον ευθύ προσεγγιστικό αλγόριθμο.

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)55	5(1000)60
Λύθηκαν	10	10	10	4	5
Μέσος Χρ. CPU Βέλ.	1100	1995	175	113	7
Μέσος Χρ. CPU Πρ.	19	62	35	74	7
Συχνότητα	0.6	0.8	0.7	1	1
Απόλυτο Σφάλμα	7.5	2	6.6	0	0
Σχετικό Σφάλμα	0.015	0.004	0.012	0	0

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες(Εργ.)Παρτίδες	20(55)40	15(55)35	10(55)35	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	245	752	439	159	16
Μέσος Χρ. CPU Πρ.	23	19	51	36	5
Συχνότητα	0.9	0.6	1	0.8	0.8
Απόλυτο Σφάλμα	21	26.2	0	34	14
Σχετικό Σφάλμα	0.154	0.230	0	0.290	0.246

Παράγοντας Καθυστέρησης = 0.2			Σχετικό Εύρος Προθεσμιών = 0.6		
Ομάδες(Εργ.)Παρτίδες	20(55)40	15(55)35	10(55)35	7(55)35	5(55)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	2100	3545	1892	2548	39
Μέσος Χρ. CPU Πρ.	88	35	122	67	19
Συχνότητα	1	1	1	1	1
Απόλυτο Σφάλμα	0	0	0	0	0
Σχετικό Σφάλμα	0	0	0	0	0

Παράγοντας Καθυστέρησης = 0.2		Σχετικό Εύρος Προθεσμιών = 0.8			
Ομάδες(Εργ.)Παρτίδες	20(50)40	15(50)35	10(50)32	7(50)32	5(50)35
Λύθηκαν	9	8	7	9	10
Μέσος Χρ. CPU Βέλ.	1095	1292	1515	1323	888
Μέσος Χρ. CPU Πρ.	104	47	57	85	105
Συχνότητα	1	1	1	1	1
Απόλυτο Σφάλμα	0	0	0	0	0
Σχετικό Σφάλμα	0	0	0	0	0

Παράγοντας Καθυστέρησης = 0.4		Σχετικό Εύρος Προθεσμιών = 0.2			
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)55	5(1000)60
Λύθηκαν	10	10	10	4	4
Μέσος Χρ. CPU Βέλ.	843	883	170	97	8
Μέσος Χρ. CPU Πρ.	18	53	37	8	8
Συχνότητα	0.3	0.9	0.7	1	1
Απόλυτο Σφάλμα	11.857	2	5.7	0	0
Σχετικό Σφάλμα	0.009	0.001	0.004	0	0

Παράγοντας Καθυστέρησης = 0.4		Σχετικό Εύρος Προθεσμιών = 0.4			
Ομάδες(Εργ.)Παρτίδες	20(55)40	15(55)35	10(55)35	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	1584	900	220	215	9
Μέσος Χρ. CPU Πρ.	41	16	25	19	5
Συχνότητα	0.8	0.2	0.8	0.7	0.8
Απόλυτο Σφάλμα	3.5	19.2	6	20	5
Σχετικό Σφάλμα	0.005	0.028	0.009	0.026	0.006

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.6		
Ομάδες(Εργ.)Παρτίδες	20(50)40	15(50)35	10(50)32	7(50)32	5(50)31
Λύθηκαν	9	10	10	10	10
Μέσος Χρ. CPU Βέλ.	2058	1930	3772	1369	116
Μέσος Χρ. CPU Πρ.	94	46	54	80	66
Συχνότητα	0.8	0.5	0.6	0.6	1
Απόλυτο Σφάλμα	1.5	15.6	21.5	6.5	0
Σχετικό Σφάλμα	0.003	0.036	0.064	0.020	0

Παράγοντας Καθυστέρησης = 0.4			Σχετικό Εύρος Προθεσμιών = 0.8		
Ομάδες(Εργ.)Παρτίδες	20(45)37	15(45)33	10(45)32	7(45)32	5(45)34
Λύθηκαν	10	10	6	7	10
Μέσος Χρ. CPU Βέλ.	1146	1897	3422	3371	461
Μέσος Χρ. CPU Πρ.	85	55	120	386	374
Συχνότητα	0.6	0.8	0.7	1	1
Απόλυτο Σφάλμα	17.75	28	16	0	0
Σχετικό Σφάλμα	0.125	0.197	0.131	0	0

Παράγοντας Καθυστέρησης = 0.6			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)55	5(1000)60
Λύθηκαν	10	10	10	4	5
Μέσος Χρ. CPU Βέλ.	844	1812	171	137	5
Μέσος Χρ. CPU Πρ.	18	47	32	87	5
Συχνότητα	0.3	0.9	0.8	1	1
Απόλυτο Σφάλμα	11.8	9	5	0	0
Σχετικό Σφάλμα	0.006	0.004	0.002	0	0

Παράγοντας Καθυστέρησης = 0.6			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες(Εργ.)Παρτίδες	20(55)37	15(55)33	10(55)33	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	401	1191	376	216	15
Μέσος Χρ. CPU Πρ.	10	8	16	19	5
Συχνότητα	0.9	0.7	0.8	0.7	0.7
Απόλυτο Σφάλμα	2	23.1	4	20	9.6
Σχετικό Σφάλμα	0.002	0.018	0.003	0.014	0.006

Παράγοντας Καθυστέρησης = 0.6			Σχετικό Εύρος Προθεσμιών = 0.6		
Ομάδες(Εργ.)Παρτίδες	20(50)37	15(50)33	10(50)33	7(50)33	5(50)31
Λύθηκαν	9	10	10	10	10
Μέσος Χρ. CPU Βέλ.	2066	1939	3747	1382	117.9
Μέσος Χρ. CPU Πρ.	28	20	89	114.9	65
Συχνότητα	0.77	0.5	0.7	0.7	1
Απόλυτο Σφάλμα	28	23.4	27	6	0
Σχετικό Σφάλμα	0.028	0.024	0.034	0.007	0

Παράγοντας Καθυστέρησης = 0.6			Σχετικό Εύρος Προθεσμιών = 0.8		
Ομάδες(Εργ.)Παρτίδες	20(45)37	15(45)33	10(45)33	7(45)35	5(45)33
Λύθηκαν	10	7	5	8	10
Μέσος Χρ. CPU Βέλ.	1416	1958	2045	2062	963
Μέσος Χρ. CPU Πρ.	98	32	200	902	790
Συχνότητα	0.7	0.57	0.8	0.875	1
Απόλυτο Σφάλμα	16.3	25.6	25	1	0
Σχετικό Σφάλμα	0.025	0.042	0.044	0.001	0

Παράγοντας Καθυστέρησης = 0.8			Σχετικό Εύρος Προθεσμιών = 0.2		
Ομάδες(Εργ.)Παρτίδες	20(70)40	15(75)40	10(85)40	7(200)60	5(1000)60
Λύθηκαν	10	10	10	4	5
Μέσος Χρ. CPU Βέλ.	1093	1971	176	113	7
Μέσος Χρ. CPU Πρ.	18	62	34	113	7
Συχνότητα	0.6	0.8	0.7	1	1
Απόλυτο Σφάλμα	7.5	2	6.6	0	0
Σχετικό Σφάλμα	0.003	0.001	0.002	0	0

Παράγοντας Καθυστέρησης = 0.8			Σχετικό Εύρος Προθεσμιών = 0.4		
Ομάδες(Εργ.)Παρτίδες	20(55)37	15(55)35	10(55)33	7(60)35	5(75)35
Λύθηκαν	10	10	10	10	10
Μέσος Χρ. CPU Βέλ.	527	768	359	218	18
Μέσος Χρ. CPU Πρ.	13	11	14	19	5
Συχνότητα	0.7	0.1	0.6	0.7	0.7
Απόλυτο Σφάλμα	15	25.3	8.5	20	9.3
Σχετικό Σφάλμα	0.008	0.013	0.005	0.010	0.003

Παράγοντας Καθυστέρησης = 0.8			Σχετικό Εύρος Προθεσμιών = 0.6		
Ομάδες(Εργ.)Παρτίδες	20(50)37	15(50)33	10(50)33	7(50)35	5(50)34
Λύθηκαν	9	9	10	9	10
Μέσος Χρ. CPU Βέλ.	579	892	2440	342	22
Μέσος Χρ. CPU Πρ.	41	26	135	259	22
Συχνότητα	0.77	0.33	0.6	0.88	1
Απόλυτο Σφάλμα	21.5	14.5	15.25	3	0
Σχετικό Σφάλμα	0.014	0.009	0.010	0.002	0

Παράγοντας Καθυστέρησης = 0.8			Σχετικό Εύρος Προθεσμιών = 0.8		
Ομάδες(Εργ.)Παρτίδες	20(45)37	15(45)33	10(45)33	7(45)35	5(45)30
Λύθηκαν	10	10	9	9	10
Μέσος Χρ. CPU Βέλ.	688	1813	1747	1937	112
Μέσος Χρ. CPU Πρ.	106	50	389	853	112
Συχνότητα	0.9	0.6	0.88	0.88	1
Απόλυτο Σφάλμα	30	47.25	2	18	0
Σχετικό Σφάλμα	0.028	0.042	0.002	0.016	0