ΠΟΛΥΤΕΧΝΕΙΟ
ΚΡΗΤΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Επίλυση του προβλήματος του flow-shop με χρήση του αλγορίθμου μουσικής αρμονίας.

ΠαπαπέτροςΠοριώτης Βασίλειος Μάριος

Επιβλέπων Καθηγητής :

Ιωάννης Μαρινάκης

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εργασία προσεγγίζεται το πρόβλημα του χρονικού προγραμματισμού εργασιών παραγωγής συνεχούς ροής (flow-shop system) με την εφαρμογή του αλγορίθμου μουσικής αρμονίας. Τα προβλήματα χρονικού προγραμματισμού παρουσιάζουν ιδιαίτερο ενδιαφέρον καθώς πηγάζουν από τον χώρο της βιομηχανικής παραγωγής ενώ παράλληλα χαρακτηρίζονται ως NP-hard προβλήματα, για τα οποία δεν υπάρχει γνωστός αποδοτικός αλγόριθμος που να δίνει βέλτιστη λύση. Στόχος του χρονικού προγραμματισμού είναι η «τακτοποίηση» συνόλου εργασιών κατά τέτοιο τρόπο ώστε να βελτιστοποιούνται ορισμένα κριτήρια, όπως ο χρόνος. Στην εν λόγω εργασία εξετάζεται ένα σύστημα παραγωγής συνεχούς ροής ενώ μοναδικό κριτήριο βέλτιστου είναι η ελαχιστοποίηση του χρόνου περάτωσης του προγράμματος εργασιών. Για την επίλυση του παραπάνω προβλήματος αναπτύσσεται υπολογιστικό πρόγραμμα βασισμένο στον αλγόριθμο μουσικής αρμονίας με σκοπό την εύρεση υποβελτιστων λύσεων.

ABSTRACT

In this thesis a flow-shop production sequencing problem is being analyzed in an attempt to find an optimal solution by the use of a particular algorithm, titled as harmony search algorithm. Production scheduling problems are of particular interest as derived from the fields of industrial production, while they are classified as NP-hard problems for which there has not been discovered any algorithm that would efficiently give an optimal solution. The objective of a scheduling production is to «regulate» a set of actions in such a way that certain criteria are optimized, such as the criterion of time. In this project a flow-shop scheduling system is examined and a single case is dealt with, the objective of which is to minimize the completion time of a set of actions. For this purpose a programm based on music harmony algorithm is applied in order to search for suboptimal solutions.

ΕΙΣΑΓΩΓΗ

Στα τέλη του 20^{ου} και στις αρχές του 21^{ου} αιώνα, η ανακάλυψη νέων προηγμένων τεχνολογιών βρίσκει εφαρμογή στον χώρο της βιομηχανίας με την χρήση πολλαπλών, τεχνολογικά τελειοποιημένων, μηχανών. Ταυτοχρόνως η συνεχής αύξηση του πληθυσμού με την αντίστοιχη αύξηση της ζήτησης προϊόντων σε αντιπαράθεση με τον ολοένα αυξανόμενο ανταγωνισμό στον χώρο της βιομηχανίας και των υπηρεσιών καθιστά ως επιτακτική ανάγκη τον χρονικό προγραμματισμό ενός συστήματος παραγωγής (scheduling of production processes).

Ο χρονικός προγραμματισμός συστημάτων παραγωγής στοχεύει στην βελτιστοποίηση της αποδοτικότητας ενός συνόλου διεργασιών παραγωγής με σκοπό την αύξηση των οικονομικών εισροών στο σύστημα, είτε λόγω αύξησης της παραγωγής σε δεδομένο χρόνο, είτε λόγω μείωσης της ανθρώπινης εργασίας. Αντίθετα, οι μεγάλες καθυστερήσεις σε ένα συστήματα παραγωγής αποτελούν τροχοπέδη για τα οικονομικά αποτελέσματα. Σε γενικές γραμμές, οι αντικειμενικοί σκοποί του χρονικού προγραμματισμού της παραγωγής είναι η αποτελεσματική χρησιμοποίηση μηχανών και προσωπικού και η ελαχιστοποίηση του χρόνου αναμονής πελατών, αποθήκευσης και χρόνου εκτέλεσης.

Στην παρούσα διπλωματική εργασία εξετάζεται ο χρονοπρογραμματισμός εργασιών παραγωγής συνεχούς ροής. Το πρόβλημα προγραμματισμού παραγωγής συνεχούς ροής (flow-shop scheduling problem) χαρακτηρίζεται ως NP-hard πρόβλημα, για το οποίο δεν υπάρχει γνωστός αποδοτικός αλγόριθμος που να δίνει βέλτιστη λύση και είναι σχεδόν αδύνατον να βρεθεί. Ως εκ τούτου, εφαρμόζονται τεχνικές μεθευρετικών-γενετικών αλγορίθμων για την εύρεση μιας καλής λύσης στο πρόβλημα βελτιστοποίησης καθώς η υπολογιστικής ισχύς είναι περιορισμένη και δεν επαρκεί για την εύρεση της βέλτιστης λύσης. Από το σύνολο των αλγορίθμων βελτιστοποίησης εφαρμόζεται ο αλγόριθμος μουσικής αρμονίας.

Η πρώτη ενότητα αναφέρεται στο θεωρητικό μέρος, όπου αναλύονται οι ορισμοί βασικών εννοιών για την κατανόηση των τεχνικών που χρησιμοποιήθηκαν ενώ παράλληλα παρουσιάζονται οι βιβλιογραφικές αναφορές που υποστηρίζουν το πρόβλημα της παραγωγής συνεχούς ροής. Στην δεύτερη ενότητα παρουσιάζεται αναλυτικά η δομή του προγράμματος καθώς και τα αποτελέσματα που προκύπτουν από την χρήση του.

1η ΕΝΟΤΗΤΑ

ΘΕΩΡΗΤΙΚΟ ΜΕΡΟΣ

ΒΑΣΙΚΕΣ ΕΝΝΟΙΕΣ ΠΟΥ ΑΦΟΡΟΥΝ ΤΟΝ ΤΟΜΕΑ ΤΗΣ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Διαχείριση της εφοδιαστικής αλυσίδας

Η διαχείριση της εφοδιαστικής αλυσίδας ασχολείται με τον σχεδιασμό και την διαχείριση όλων των ενεργειών και δραστηριοτήτων που σχετίζονται με την προμήθεια, παραγωγή και διανομή των προϊόντων. Συνεπώς, η Διαχείριση Εφοδιαστικής Αλυσίδας συντονίζει και ολοκληρώνει το σχεδιασμό, τις προμήθειες, την παραγωγή, την αποθήκευση, τη μεταφορά και τις πωλήσεις τόσο μέσα στις επιχειρήσεις όσο και μεταξύ αυτών. Ο αντικειμενικός λοιπόν σκοπός της Διαχείρισης της Εφοδιαστικής Αλυσίδας είναι η αύξηση των κερδών κατά μήκος της αλυσίδας που συνεπάγεται στην συνολική αύξηση των κερδών όλων των εταίρων της. Αυτό πραγματοποιείται με την κατανόηση και ικανοποίηση των πελατειακών αναγκών στον απαιτούμενο χρόνο, και με την προσφορά ποιοτικών προϊόντων σε ανταγωνιστικές τιμές. Για την επίτευξη των παραπάνω στόχων ,η ευελιξία καθώς και η ταχεία προσαρμογή στις δυναμικά μεταβαλλόμενες συνθήκες, αποτελούν αναγκαία χαρακτηριστικά των εφοδιαστικών αλυσίδων που ανταγωνίζονται μέσα στο σύγχρονο παγκοσμιοποιημένο περιβάλλον [1]

Με τον όρο Logistics εννοείται το τμήμα Διαχείρισης Εφοδιαστικής Αλυσίδας που ως σκοπό έχει την σχεδίαση, την υλοποίηση και τον έλεγχο της ροής και της αποθήκευσης των προϊόντων, των υπηρεσιών και των σχετικών πληροφοριών από το σημείο προέλευσης τους έως το σημείο κατανάλωσης τους, με τέτοιο τρόπο ώστε να ικανοποιούνται οι απαιτήσεις των πελατών.

Τα Logistics αφορούν σε:

Απαιτήσεις

Τα Logistics αποσκοπούν στην ανάλυση, σύνθεση και καθορισμό των πόρων που χρειάζονται ώστε να επιτύχουμε ένα σκοπό.

Σχεδιασμό:

Αυτή η λειτουργία περιλαμβάνει όλο το πλάνο του σχεδιασμού μέσα από λεπτομερή σχεδιασμό των προϊόντων, συστημάτων και υπηρεσιών, συμπεριλαμβανομένων της ανάπτυξης, δοκιμής και αξιολόγησης του σχεδιασμού.

Εφοδιασμό:

Η λειτουργία του εφοδιασμού ασχολείται με τον εφοδιασμό και την διανομή των διαθέσιμων πόρων.(προμήθειες, πρόσληψη και εκπαίδευση Προσωπικού, υποστήριξη παραγωγής, συσκευασία, Διοίκηση Αποθεμάτων, διακίνηση και μεταφορές, διαδικασία παραγγελιών, αποθήκευση, αποσύρσεις, κ.λ.π)

Συντήρηση:
Με τον όρο συντήρηση εννοείται η διατήρηση των εγκαταστάσεων, προϊόντων, ανθρώπινου δυναμικού, συστημάτων και υπηρεσιών των παραγωγών και χρηστών, καθώς και της προστασίας των διατιθέμενων πόρων.

Πόρους:
Πρώτες ύλες (υλικά), εξοπλισμός εγκαταστάσεις, Προσωπικό, συμπεριλαμβανομένων των κεφαλαίων και πληροφοριών. Τα Logistics συχνά συνδέονται με την διοίκηση των υλικών, όμως οι τεχνικές της διοίκησης των υλικών μπορούν επίσης να εφαρμοσθούν στην διοίκηση του

ανθρωπίνου δυναμικού, χρημάτων και πληροφοριών.

Οι δραστηριότητες των Logistics συμπληρώνουν και υποστηρίζουν την στρατηγική και την τακτική. Υποστηρίζουν τους στόχους, τα σχέδια και τις επιχειρησιακές δραστηριότητες των συστημάτων. Τα υποστηριζόμενα συστήματα μπορεί να είναι οργανισμοί ή μεμονωμένα άτομα.[1]

ΕΦΟΔΙΑΣΤΙΚΗ

Με τον όρο εφοδιαστική εννοούμε την διαχείριση υλικών από την πηγή προέλευσης τους μέχρι την παράδοση τους στο σημείο κατανάλωσης.

Με τον όρο διαχείριση της εφοδιαστικής αναφερόμαστε στον προγραμματισμό, την υλοποίηση και τον έλεγχο για μια αποτελεσματική , υλοποιήσιμη ροη και αποθήκευση αγαθών , υπηρεσιών και πληροφοριών από την πηγή στο σημείο κατανάλωσης , έχοντας πάντα ως αντικειμενικό σκοπό την ικανοποίηση των απαιτήσεων των πελατών. Συνεπώς η διαχείριση της εφοδιαστικής έχει ως σκοπό την επίτευξη των εφοδιαστικών στόχων μιας επιχείρησης μέσω της ανάπτυξης και υλοποίησης ενός πλάνου που ικανοποιεί την σωστή παράδοση των προϊόντων, στην σωστή ποσότητα, στον σωστό τόπο και χρόνο και στην σωστή τιμή. Πολλές φορές παρατηρείται σύγκρουση συμφερόντων ανάμεσα σε διαφορετικά τμήματα μιας επιχείρησης για το λόγο αυτό απαιτείται ανάλυση του ολικού εφοδιαστικού κόστους, το όποιο αντιπροσωπεύει τα έξοδα της επιχείρησης για ένα συγκεκριμένο προϊόν από την αρχική φάση έρευνας της ιδέας ως το τέλος του κύκλου ζωής του.[2]

Μια εφοδιαστική αλυσίδα έχει ως αντικειμενικό σκοπό την ικανοποίηση των απαιτήσεων του πελάτη και αποτελείται από κατασκευαστές, προμηθευτές, χώρους αποθήκευσης ,κέντρα διανομών, μεταφορείς, πωλητές, πελάτες καθώς και πρώτες ύλες, αποθέματα και έτοιμα προϊόντα.

Από τα παραπάνω γίνεται σαφές ότι ο χρόνος παράγωγης των προϊόντων αφορά άμεσα των τομέα της εφοδιαστικής αλυσίδας αφού τυχών καθυστερήσεις στην παράγωγη επηρεάζουν όλο το δίκτυο λειτουργιάς της επιχείρησης. Άρα η βελτιστοποίηση της σειράς και του χρόνου πραγματοποίησης των εργασιών αποκτά πολύ σημαντικό ρόλο για την επιχείρηση.

ΜΑΘΗΜΑΤΙΚΕΣ ΜΕΘΟΔΟΙ ΚΑΙ ΓΕΝΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Τα μεγάλα και πολύπλοκα προβλήματα τα οποία έχουν υλοποιηθεί τις τελευταίες κυρίως δεκαετίες, κρίνουν επιτακτική την ανάγκη για χρησιμοποίηση διαφόρων αλγορίθμων βελτιστοποίησης με αντικειμενικό σκοπό την μείωση του συνολικού κόστους. Η χρήση τέτοιου είδους αλγορίθμων παρέχουν στους σχεδιαστές την δυνατότητα εξοικονόμησης χρόνου και χρήματος καθώς επίσης και της επίτευξης αποτελεσματικότερων σχεδιασμών.

Με την πάροδο του χρόνου έχουν χρησιμοποιηθεί διαφόρων ειδών μαθηματικές μέθοδοι κυρίως γραμμικού, μη γραμμικού και δυναμικού προγραμματισμού. Παρόλα αυτά τέτοιου είδους μέθοδοι δεν καταφέρνουν να ικανοποιήσουν της απαιτήσεις των σύγχρονων έργων,
είτε αυτά αφορούν τον τομέα των κατασκευών ,είτε αυτά αφορούν τον τομέα της παράγωγης και βελτιστοποίησης της εφοδιαστικής αλυσίδας
των επιχειρήσεων. Αυτό συμβαίνει διότι τα αποτελέσματα αυτών των μεθόδων εξαρτώνται αποκλειστικά από την αρχική γειτονιά αναζήτησης λύσεων αφού ως βέλτιστη λύση θεωρούν το πρώτο τοπικό ελάχιστο που εντοπίζουν γύρω από ένα αρχικό σημείο αναζήτησης, όποτε σε μεγάλα πολύπλοκα προβλήματα που παρουσιάζουν πολλά τοπικά ελάχιστα , οι λύσεις που μας δίνουν συνήθως δεν καταφέρνουν να προσεγγίσουν αποτελεσματικά το ολικό βέλτιστο. Επιπλέον στα πολύπλοκα έργα που παρουσιάζουν πολλές μεταβλητές στην αντικειμενική τους συνάρτηση ελαχιστοποίησης του κόστους είναι αδύνατο να χρησιμοποιηθούν τέτοιου είδους μαθηματικές μέθοδοι. Όλα αυτά τα υπολογιστικά μειονεκτήματα των μαθηματικών μεθόδων ανάγκασαν τους επιστήμονες να στραφούν σε ευρετικους αλγόριθμους βελτιστοποίησης οι οποίοι συνδυάζουν κανόνες και τυχαιότητα.
Αυτοί οι αλγόριθμοι συνηθίζουν να μιμούνται φυσικά φαινόμενα για την επίλυση δύσκολων και πολύπλοκων προβλημάτων βελτιστοποίησης.
[3][4][5]

ΕΥΡΕΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Ένας από τους πρώτους αλγόριθμους που χρησιμοποιήθηκε ήταν ο αλγόριθμος προσομοιωμένης ανόπτησης . Ο καινοτόμος αυτός αλγόριθμος δημιουργήθηκε το 1983 και η λειτουργία του ήταν παρόμοια με την φυσική λειτουργία της ανόπτησης .[3][5]

Ο αλγόριθμος Tabu Search αποτελεί έναν ακόμη σημαντικό αλγόριθμο συνδυαστικής βελτιστοποίησης ο οποίος είχε ως βασική ιδέα την εξερεύνηση όλου του χώρου των εφικτών λύσεων μέσω μια ακολουθίας κινήσεων . Ο αλγόριθμος κινείται από την μια λύση στην άλλη και αποθηκεύει την καλύτερη . Ωστόσο για να αποφύγει τα τοπικά βέλτιστα και την ανακύκλωση των λύσεων κάποιες κινήσεις ταξινομούνται ως ταμπού ή αλλιώς απαγορευμένες κινήσεις.[3][5]

Τέλος ένας ακόμη σημαντικός αλγόριθμος που αξίζει να αναφερθεί είναι ο αλγόριθμος βελτιστοποίησης σμήνους σωματιδίων (Particle Swarm Optimization) ο όποιος αναπτύχτηκε από τον Kennedy και Eberhart. Ο συγκεκριμένος αλγόριθμος είναι εμπνευσμένος από την κοινωνική συμπεριφορά κάποιων οργανισμών όπως του σμήνους πουλιών ή της ομαδικής κίνησης των ψαριών και εμφανίζει πολύ καλά αποτελέσματα σε μεγάλο αριθμό προβλημάτων.[2][3][5]

Κάποιοι από τους εξελικτικούς αλγόριθμους είναι οι ακόλουθοι:

1. Γενετικοί αλγόριθμοι
2. Εξελικτικές στρατηγικές
3. Εξελικτικός προγραμματισμός
4. Γενετικός προγραμματισμός

Οι γενετικοί αλγόριθμοι είναι αλγόριθμοι αναζήτησης με βάση την φυσική επιλογή και τον μηχανισμό γένεσης ενός πληθυσμού λύσεων.
Η συγκεκριμένη θεωρία αρχικά προτάθηκε από τον Holland και αργότερα αναπτύχθηκε περαιτέρω από τον Goldberg και άλλους επιστήμονες . Στους απλούς γενετικούς αλγόριθμους οι κύριες ενέργειες είναι τρεις

1. Αναπαραγωγή
2. Διασταύρωση
3. Μετάλλαξη

Το κύριο χαρακτηριστικό των γενετικών αλγορίθμων είναι ότι ταυτόχρονα μπορούν να δημιουργούν και να αξιολογούν πολλές διαφορετικές λύσεις, κάτι το οποίο δεν μπορεί να πραγματοποιηθεί από τις μαθηματικές μεθόδους ,ούτε από τις ευρετικές μεθόδους βελτιστοποίησης (π.χ. tabu search). Αυτό αποτελεί ένα σημαντικό πλεονέκτημα των γενετικών αλγορίθμων που τις επιτρέπει την ευρεία αναζήτηση χωρίς τον φόβο σύγκλισης σε ένα τοπικό μέγιστο.

Οι εξελικτικές στρατηγικές αναπτύχτηκαν για να μπορούν να λύσουν προβλήματα πολλαπλών παραμέτρων. Σε κάθε επανάληψη διαλέγουν μια λύση για κάθε παράμετρο και εξετάζουν αν βελτιώνεται το συνολικό αποτέλεσμα της αντικειμενικής συνάρτησης.

Ο εξελικτικός προγραμματισμός χρησιμοποιεί μεταβλητούς πίνακες οι οποίοι αντιπροσωπεύουν σειρές λύσεων . Στην συνέχεια οι λύσεις μεταλλάσσονται τυχαία η' επιλέγονται νέες σειρές λύσεων επίσης τυχαία.

Τέλος ο γενετικός προγραμματισμός ,ο οποίος αναπτύχτηκε από τον Koza, είναι μια επέκταση τον γενετικών αλγορίθμων και υπέδειξε ότι το πρόγραμμα το ίδιο θα πρέπει να εξελίσσει την διαδικασία ανάπτυξης των λύσεων.[3][5]

Οι παραπάνω μέθοδοι έχουν μεγάλες υπολογιστικές ικανότητες και με την χρήση τους ξεπερνιούνται οι πολλές ελλείψεις που παρουσιάζουν οι μαθηματικές μέθοδοι υπολογισμού βέλτιστης στρατηγικής. Ειδικά τις τελευταίες δεκαετίες η χρήση των γενετικών αλγορίθμων έχει χρησιμεύσει στην λύση διαφόρων προβλημάτων βελτιστοποίησης σε επίπεδο μηχανικής Είναι σημαντικό να αναφερθεί πως παρόλο που οι γενετικοί αλγόριθμοι έχουν το πλεονέκτημα να μην έχουν μαθηματικές απαιτήσεις για επίλυση ενός προβλήματος , απαιτούν ωστόσο μεγάλες ταχύτητες υπολογιστών που συνεπάγεται σε υπολογιστές υψηλού κόστους.

ΑΛΓΟΡΙΘΜΟΣ ΑΝΑΖΗΤΗΣΗΣ ΤΗΣ ΜΟΥΣΙΚΗΣ ΑΡΜΟΝΙΑΣ

Ο αλγόριθμος που θα χρησιμοποιηθεί σε αυτήν την εργασία είναι ο αλγόριθμος μουσικής αρμονίας και πιο συγκεκριμένα πάνω στο πρόβλημα του flow-shop το οποίο θα αναλυθεί λεπτομερώς αργότερα.

Ο αλγόριθμος αναζήτησης της μουσικής αρμονίας αποτελεί έναν νέο αλγόριθμο που έχει παρουσιαστεί τα τελευταία χρόνια ο οποίος έχει χρησιμοποιηθεί σε πολλά προβλήματα και έχει δώσει πολύ καλά αποτελέσματα. Ο στόχος του αλγορίθμου είναι να επιτευχθεί η βέλτιστη αρμονία. Ο αλγόριθμος αναζήτησης της μουσικής αρμονίας μιμείται το πώς οι μουσικοί κατά τη διάρκεια της φάσης του αυτοσχεδιασμού εμπλουτίζουν την εμπειρία τους με τη συνεχή εξάσκηση για να πετύχουν καλύτερες αρμονίες. Ο κάθε μουσικός δηλαδή το κάθε μουσικό όργανο, παράγει μια συγκεκριμένη συχνότητα από την κλίμακα αρμονίας. Αν όλες οι συχνότητες (τονικότητες) παράγουν μια καλή αρμονία, τότε αυτή αποθηκεύεται στη μνήμη των μουσικών ώστε την επόμενη φορά η πιθανότητα για μια καλή αρμονία να είναι μεγαλύτερη. Η αντιστοίχιση με τη βελτιστοποίηση γίνεται ως εξής: Ένα μουσικό όργανο στη φάση του αυτοσχεδιασμού αντιστοιχεί σε μία μεταβλητή απόφασης για ένα διάνυσμα λύσης. Η έκταση της τονικότητας του μουσικού οργάνου αντιστοιχεί στο πλήθος των τιμών που μπορεί να πάρει μία μεταβλητή και μία μνήμη αντιστοιχεί σε ένα διάνυσμα λύσης. Στον αλγόριθμο αναζήτησης της μουσικής αρμονίας, ένας μουσικός βρίσκει την καλύτερη αρμονία βασιζόμενος σε τρεις παράγοντες:

1. Τυχαιότητα
2. Εμπειρία
3. Μεταβολή της εμπειρίας

Ο αλγόριθμος αναζήτησης της μουσικής αρμονίας αποτελείται από ένα σύνολο από αρμονίες όπου η κάθε μία αρμονία αποτελεί και μία λύση του προβλήματος. Η κάθε αρμονία συμβολίζεται με x_{ij} όπου $i = 1, \ldots, HMS$ είναι το πλήθος των αρμονιών που μπορούν να παραχθούν και $j = 1, \ldots, n$ είναι οι διαστάσεις του προβλήματος. Αρχικά οι τιμές για τις αρμονίες υπολογίζονται τυχαία στο πεδίο τιμών των λύσεων. Έπειτα δημιουργείται η Μνήμη Αρμονίας (Harmony Memory (HM)) η οποία αποτελείται αρχικά από τις λύσεις που παράχθηκαν με τυχαίο τρόπο.

HM=

X_{11}	X_{12}	……………………	X_{1n}	$f(x_1)$
X_{21}	X_{22}	……………………	X_{2n}	$f(x_2)$
….	…..	…………………..	….	….
$X_{HMS,1}$	$X_{HMS,2}$	……………….. ..	$X_{HMS,n}$	$f(x_{HMS})$

(1)

Το HM είναι η μνήμη και HMS είναι το μέγεθος της μνήμης. Επίσης, η κάθε γραμμή αντιστοιχεί σε μία αρμονία, διαστάσεων n και η τελευταία στήλη αντιστοιχεί στην τιμή της αντικειμενικής συνάρτησης κάθε λύσης. Στη συνέχεια επιλέγεται μία καινούρια αρμονία. Η καινούρια αρμονία προέρχεται από τους τρεις παράγοντες που περιγράψαμε προηγούμενα, δηλαδή τυχαιότητα, εμπειρία και μεταβολή της εμπειρίας.

Ο πρώτος παράγοντας είναι η εμπειρία. Από την εμπειρία μπορεί να προκύψει μία λύση με πιθανότητα HMCR (Harmony Memory Considering Rate). Το HMCR αντιστοιχεί στην πιθανότητα να επιλεγεί ένα σημείο τυχαία από τα μέλη της Αρμονικής Μνήμης. Η καινούρια λύση σε αυτή την περίπτωση δίνεται από την εξίσωση:

$X_{NEW,j} \leftarrow x_{1,j}\ \ x_{2,j}$, …….., $x_{HMS,j}$, με πιθανότητα HMCR (2)

Αξίζει να διευκρινιστεί ότι αυτό το ενδεχόμενο νέας λύσης αποτελείται εξολοκλήρου από λύσεις οι οποίες ήδη υπάρχουν μέσα στον αρχικό τυχαίο πίνακα λύσεων.

Αν χρησιμοποιηθεί μόνο αυτός ο παράγοντας τότε υπάρχει η πιθανότητα να μην μπορεί η λύση να εξερευνήσει κάποια σημεία του χώρου λύσεων λόγω του ότι μπορεί οι αρχικές τυχαίες τιμές να μην είναι τόσο καλές. Αυτός είναι ο λόγος που οι ερευνητές χρησιμοποίησαν και το δεύτερο παράγοντα όπου είναι η τύχη. Η τύχη χρησιμοποιείται για να δημιουργηθεί μία λύση στο πεδίο τιμών των μεταβλητών S με πιθανότητα 1 −HMCR. Η καινούρια λύση σε αυτή την περίπτωση δίνεται από την εξίσωση:

$X_{NEW,j} \leftarrow x_j \in S$, με πιθανότητα 1 –HMCR (3)

Άρα λοιπόν συνολικά μια καινούργια λύση θα δίνεται ως εξής

$$x_{NEW,j} = \begin{cases} x_{1,j}, x_{2,j}, \ldots, x_{HMS,j}, & \text{με πιθανότητα HMCR} \\ x_j \in S, & \text{με πιθανότητα 1-HMCR} \end{cases} \tag{4}$$

Για παράδειγμα αν επιλεγεί το HMCR να είναι ίσο με 0.9 δείχνει ότι το 90% των καινούριων μεταβλητών θα επιλεγεί από την ιστορική μνήμη των λύσεων, δηλαδή τη μνήμη αρμονίας και το υπόλοιπο 10% θα επιλεγεί από τον υπόλοιπο χώρο λύσεων. Αν κάποιος επιλέξει το HMCR να είναι ίσο με τη μονάδα τότε υπάρχει η πιθανότητα να μην μπορούν να βελτιωθούν οι λύσεις αν οι λύσεις που βρίσκονται στη μνήμη δεν έχουν τη δυνατότητα να οδηγήσουν στο βέλτιστο.

Στην περίπτωση που η καινούρια λύση προκύψει από τη μνήμη αρμονίας τότε γίνεται μία διαδικασία τοπικής αναζήτησης η οποία πραγματοποιείται με πιθανότητα PAR (Pitch Adjustment Rate). Η καινούρια λύση σε αυτή την περίπτωση δίνεται από την εξίσωση

$$X_{NEW,j} = \begin{cases} x_j\,(k+m) & \text{με πιθανότητα PAR} \\ x_j\,(k) & \text{με πιθανότητα 1-PAR} \end{cases} \tag{5}$$

όπου το $x_j(k + m)$ είναι πανομοιότυπο με το $x_{NEW,j}$ που αποκτήθηκε στη φάση που εφαρμόστηκε ο παράγοντας εμπειρία. Ο παράγοντας γειτονιάς m είναι ένα στοιχείο του συνόλου $\{\ldots,-2,-1, 1, 2, \ldots\}$ με συνηθέστερες τιμές $m \in \{-1, 1\}$.

Πιο αναλυτικά η διαδικασία προσαρμογής της τονικότητας εφαρμόζεται μόνο όταν μία τιμή έχει επιλεγεί από τη Μνήμη Αρμονίας. Σε ένα μεγάλο ποσοστό των τιμών, σε αυτές όπου ισχύει το $1 - PAR$ δεν θα γίνει τίποτα. Αν τώρα θέσουμε το PAR ίσο με 0.3 σημαίνει ότι στο 30% των σημείων που προήλθαν από τη μνήμη αρμονίας (30%×HMCR) θα εφαρμοστεί μία διαδικασία τοπικής αναζήτησης. Έτσι για το σημείο i του διανύσματος η τιμή του υπολογίζεται από την εξίσωση:

$$X_{NEW,j} = X_{NEW,j} + a \tag{6}$$

όπου το α = bw×rand1, και το rand1 είναι μία τιμή στο διάστημα (-1,1). Έτσι για κάθε στοιχείο της καινούριας μνήμης αρχικά ελέγχουμε αν θα είναι από τη μνήμη ή όχι. Αυτός ο έλεγχος πραγματοποιείται από το αν μία τυχαία τιμή βγει μικρότερη από το HMCR. Αν βγει μεγαλύτερη τότε υπολογίζουμε έναν τυχαίο αριθμό μέσα στο πεδίο τιμών. Αν βγει μικρότερη σημαίνει ότι η τιμή θα επιλεγεί τυχαία από μία από τις αρμονίες που υπάρχουν μέσα στη μνήμη αρμονίας HM. Στη συνέχεια ελέγχουμε τη διαδικασία προσαρμογής της τονικότητας για τη συγκεκριμένη τιμή. Λειτουργούμε ως εξής:

Ελέγχουμε αν ένας τυχαίος αριθμός είναι μικρότερος από το PAR. Αν ναι τότε εφαρμόζουμε τη διαδικασία τοπικής αναζήτησης για το συγκεκριμένο στοιχείο, αν όχι η τιμή του στοιχείου είναι αυτή που επιλέχθηκε από τη μνήμη αρμονίας.

ΧΡΟΝΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΑΡΑΓΩΓΗΣ

Τα συστήματα παραγωγής μπορούν να ταξινομηθούν σε τρεις βασικές κατηγορίες: τα συστήματα συνεχούς ροής (flow-shop), τα συστήματα παραγωγής κατά παραγγελία (job-shop) και τα συστήματα κατασκευής έργων (projects). Στην συγκεκριμένη εργασία θα ασχοληθούμε με τα συστήματα συνεχούς ροής (flow-shop). Ας αναφέρουμε όμως πρώτα λίγα λόγια για τον χρονικό προγραμματισμό της παράγωγης και την χρησιμότητα του.

Ο προγραμματισμός της γραμμής παραγωγής επηρεάζει σημαντικά την ροη των οικονομικών εισροών σε οποιοδήποτε σύστημα. Η σωστή διαμόρφωση του χρονικού προγραμματισμού σε ένα σύστημα παραγωγής θεωρείται ιδιαίτερα σημαντικός παράγοντας αύξησης των οικονομικών εισροών στο σύστημα. Αντιθέτως, οι μεγάλες καθυστερήσεις εμφανίζουν αρνητικές επιπτώσεις στα οικονομικά αποτελέσματα του συστήματος. Οι σημαντικότεροι από τους αντικειμενικούς σκοπούς του χρονικού προγραμματισμού της παραγωγής είναι :

1. η αποτελεσματική χρήση των μηχανών και του προσωπικού
2. η ελαχιστοποίηση του χρόνου αναμονής πελατών, αποθήκευσης και εκτέλεσης των εργασιών.

Η ανάγκη για ορθή λήψη ανεξάρτητων στοιχειωδών αποφάσεων , ο συνδυασμός των οποίων θα μας οδηγήσει στην βέλτιστη λύση στο πρόβλημα του προγραμματισμού, αποτελεί κοινό χαρακτηριστικό πολλών προβλημάτων προγραμματισμού παραγωγής (όπως για παράδειγμα η προετοιμασία μηνιαίου πλάνου παραγωγής μιας αυτοκινητοβιομηχανίας, η διαχείριση πρώτων υλών μιας εταιρείας τροφίμων ή ο προγραμματισμός του συνόλου των πτήσεων μιας μεγάλης αεροπορικής εταιρείας).

Για να γίνει ευκολότερη η λήψη σωστών αποφάσεων σε περιπτώσεις όπου υπάρχει μεγάλη υπολογιστική πολυπλοκότητα , αναπτύχτηκαν διάφορες τεχνικές βελτιστοποίησης . Η δημιουργία και εφαρμογή τεχνικών βελτιστοποίησης για κάθε πρόβλημα χρονικού προγραμματισμού απαιτεί την χρήση μιας σειρά παραμέτρων, όπως η διαθέσιμη δυναμικότητα του συστήματος, οι απαιτήσεις για παραγωγή που εξαρτώνται από τη ζήτηση των προϊόντων καθώς και διάφοροι τεχνολογικοί και άλλοι περιορισμοί. Λαμβάνοντας υπόψη αυτές τις παραμέτρους ζητείται η καλύτερη δυνατή τιμή μια αντικειμενικής συνάρτησης κόστους ή οφέλους , δηλαδή η τιμή που αντιστοιχεί στην βέλτιστη επιλογή μεταβλητών απόφασης . Οι μεταβλητές απόφασης συχνά εκφράζουν το μέγεθος μιας παρτίδας παραγωγής , τη σειρά εκτέλεσης των παραγγελιών , την ανάθεση συγκεκριμένων παραγγελιών σε συγκεκριμένες μηχανές κλπ. Έτσι από το σύνολο των εφικτών προγραμμάτων αναζητάμε το καλύτερο δυνατόν, όμως ο καθορισμός του βέλτιστου πλάνου, ειδικά σε προβλήματα με υψηλές απαιτήσεις συχνά είναι μη εφικτός . Οπότε το ζητούμενο είναι η εύρεση ενός προγράμματος το οποίο από την μια να ικανοποιεί όλους τους περιορισμούς του προβλήματος και από την άλλη να πληροί τα ελάχιστα απαιτούμενα κριτήρια που έχει θέσει ο χρήστης.[6]

ΑΠΑΙΤΗΣΕΙΣ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Για την επίλυση ενός προβλήματος χρονικού προγραμματισμού είναι απαραίτητη η ύπαρξη κατάλληλων πληροφοριών για της απαιτήσεις της παραγωγής των προϊόντων . Αυτές προκύπτουν από τις παραγγελίες των πελατών που είναι ήδη γνωστές ή από τις προβλέψεις για τις μελλοντικές παραγγελίες (BillOfResources – BOR). Με αυτόν τον τρόπο γίνεται ο προσδιορισμός των χαρακτηριστικών και του αριθμού των μηχανημάτων που απαιτούνται , των διεργασιών και της σειράς με την οποία πρόκειται να πραγματοποιηθούν , των χρόνων επεξεργασίας των μηχανημάτων καθώς και των προθεσμιών παράδοσης των παραγγελιών. Στο πρόβλημα του χρονικού προγραμματισμού παραγωγής ,οι περιορισμοί και οι κανόνες του συστήματος, αποτελούν σημαντικούς παράγοντες και αφορούν την δυναμικότητα του διαθέσιμου παραγωγικού εξοπλισμού, τις απαιτήσεις για τη συντήρηση και το στήσιμο των μηχανών, την εκπλήρωση των απαιτήσεων που ορίζει η εκάστοτε τεχνολογία των μηχανημάτων και τα δεδομένα του προγράμματος παραγωγής για το συνολικό επίπεδο της παραγωγής, του ανθρώπινου δυναμικού και των αποθεμάτων. Τέλος, η συνάρτηση κόστους/ οφέλους, όπως αναφέρθηκε και προηγουμένως, αναφέρεται στην ικανοποίηση συγκεκριμένων κριτηρίων που μπορεί να αφορούν την γρηγορότερη εξυπηρέτηση των πελατών, το συνολικό κόστος παραγωγής, την αποτελεσματικότερη αξιοποίηση της διαθέσιμης δυναμικότητας κλπ. Έτσι, ένα πρόγραμμα παραγωγής είναι καλύτερο από ένα άλλο αν το πρώτο ικανοποιεί σε μεγαλύτερο βαθμό τα κριτήρια που έχουν τεθεί (π.χ. ικανοποιούνται ταχύτερα οι παραγγελίες), σύμφωνα με την τιμή που παίρνει αντίστοιχα η συνάρτηση κόστους/ οφέλους.[6]

Συνοψίζοντας οι απαιτήσεις του χρονικού προγραμματισμού αφορούν:

1. Πληροφορίες σχετικά με τις απαιτήσεις για παράγωγη προϊόντων.
2. Περιορισμοί του συστήματος (ακολουθία εργασιών, συντήρησης μηχανών , ανθρώπινο δυναμικό κτλ.)
3. Βελτιστοποίηση συνάρτησης κόστους/οφελους (αντικειμενική συνάρτηση).

ΚΑΤΗΓΟΡΙΕΣ ΕΡΓΑΣΙΩΝ

Οι εργασίες σε ένα σύστημα μπορούν να διακριθούν στις παρακάτω κατηγορίες:

1. Συνεχείς / Διακριτές Με τον όρο συνεχείς εννοούμε τις εργασίες οι οποίες μπορούν να μετρηθούν σε συνεχή κλίμακα όπως η παραγωγή πετρελαίου, αντίθετα με τον όρο διακριτές εννοούμε τις εργασίες που για να ολοκληρωθούν απαιτούν μια διαδικασία όπως η παραγωγή ενός αυτοκινήτου ή ενός τηλεφώνου,.

2. Χρονικά Εξαρτημένες / Ανεξάρτητες Η περισσότερες εργασίες είναι χρονικά εξαρτημένες. Μερικές έχουν σταθερό ωράριο, όπως για παράδειγμα το δρομολόγιο ενός σχολικού λεωφορείου. Άλλες έχουν συγκεκριμένο χρόνο εκτέλεσης, όπως το ωράριο ενός δημοσίου υπαλλήλου ή το ωράριο λειτουργίας ενός καταστήματος. Επίσης υπάρχουν εργασίες που είναι χρονικά εξαρτημένες και για να υλοποιηθούν πρέπει υποχρεωτικά να έχουν ολοκληρωθεί κάποιες άλλες εργασίες πριν από αυτές. Αντίθετα χρονικά ανεξάρτητες λέγονται οι εργασίες που δεν έχουν χρονικούς περιορισμούς όπως η βελτιστοποίηση του χρονοδιαγράμματος της παραγωγής .

3. Χωρικά Εξαρτημένες / Ανεξάρτητες Χωρικά εξαρτημένες λέγονται οι εργασίες που πρέπει να πραγματοποιηθούν σε συγκεκριμένο χώρο ή αφορούν μεταφορικές εργασίες όπου οι διαδρομές που πρέπει να

ακολουθούσουν είναι προκαθορισμένες. Χωρικά ανεξάρτητες είναι οι εργασίες οι οποίες δεν επηρεάζονται από τον τόπο εκτέλεσης τους όπως η δημιουργία ενός προγράμματος.

4. Διακοπτόμενες / Μη-διακοπτόμενες Διακοπτόμενες λέγονται οι εργασίες που έχουν την δυνατότητα διακοπής κατά την διάρκεια εκτέλεσης τους χωρίς να έχουν ολοκληρωθεί και συνεχείας κάποια άλλη χρονική στιγμή. Παραδείγματα διακοπτόμενων εργασιών αποτελούν η επισκευή ενός μηχανήματος ή δημιουργία ενός προγράμματος. Μη διακοπτόμενες λέγονται οι εργασίες που είναι αδύνατο να διακοπούν κατά την διάρκεια εκτέλεσης τους. Χαρακτηριστικό παράδειγμα μη- διακοπτόμενης εργασίας είναι η ενοικίαση ενός σπιτιού ή η ενοικίαση αυτοκινήτων σε πελάτες.

5.Εφάπαξ / Επαναλαμβανόμενες Εφάπαξ ονομάζονται οι εργασίες, οι οποίες εκτελούνται μία μόνο φορά κατά την επίλυση ενός προβλήματος. Χαρακτηριστικό παράδειγμα είναι ο σχεδιασμός ενός κτιρίου, πριν την κατασκευή του, από ένα πολιτικό μηχανικό ή η επιλογή της κατάλληλες τοποθεσίας ανέγερσης ενός εργοστάσιου. Επαναλαμβανόμενες λέγονται οι εργασίες που εκτελούνται κυκλικά ανά τακτά χρονικά διαστήματα και συνήθως απαιτείται ο βέλτιστος σχεδιασμός του κύκλου, σε συνάρτηση με τις προηγούμενες και τις μελλοντικές εκτελέσεις. Βασικό χαρακτηριστικό των εργασιών αυτών είναι πως απαιτείται επαναφορά στην αρχική κατάσταση (αρχή κύκλου). Για παράδειγμα επαναλαμβανόμενη εργασία θα μπορούσε να θεωρηθεί η συντήρηση ενός ανελκυστήρα.[6]

ΚΑΤΗΓΟΡΙΕΣ ΠΟΡΩΝ

1.Σταθεροί/ Εναλλάξιμοι

Σταθεροί είναι οι πόροι που χρησιμοποιούνται για την εκτέλεση μιας εργασίας και δεν υπάρχει περίπτωση μεταβολής ή αντικατάστασης τους. Ως παράδειγμα αναφέρεται η περίπτωση προβλημάτων job-shop όπου κάθε εργασία εκτελείται από συγκεκριμένο τύπο πόρων. Αντίθετα, με τον όρο εναλλάξιμοι πόροι εννοούμε τους πόρους που μπορούν να αντικατασταθούν από άλλους χωρίς να εμποδίζουν την εκτέλεση μιας εργασίας . Για παράδειγμα η χρήση μιας αίθουσας διαλέξεων ή η χρήση των μέσων μεταφοράς όπου δεν περιοριζόμαστε σε μια συγκεκριμένη επιλογή για την ικανοποίηση των αναγκών μας.

2. Ανανεώσιμοι/ Αναλώσιμοι/ Μεταβλητής κατάστασης

Ανανεώσιμοι ονομάζονται οι πόροι οι οποίοι είναι διαρκώς διαθέσιμη για την εκτέλεση μιας εργασίας, όπως για παράδειγμα οι μηχανές ενός flow-shop προβλήματος. Αναλώσιμοι είναι οι πόροι που καταναλώνονται κατά τη διάρκεια εκτέλεσης μιας εργασίας όπως τα καύσιμα και οι πρώτες ύλες, και δεν γίνεται να ξαναχρησιμοποιηθούν. Τέλος, μεταβλητής κατάστασης ονομάζονται οι πόροι οι οποίοι ύστερα από την εκτέλεση της κάθε εργασίας αλλάζουν κατάσταση. Αυτοί οι πόροι χρησιμοποιούνται κυρίως σε μοντελοποίηση προβλημάτων χρονικού προγραμματισμού όπου η αρχική κατάσταση του πόρου επηρεάζει σημαντικά τη μετέπειτα ανάθεση του σε εργασίες.

3. Απλοί/ Πολλαπλοί

Απλοί πόροι λέγονται οι πόροι, οι οποίοι είναι δυνατόν να μοντελοποιηθούν διακριτά και ξεχωριστά όπως για παράδειγμα οι πόροι που απαιτούνται για την λειτουργία ενός μηχανήματος. Αντίθετα, πολλαπλοί πόροι είναι οι πόροι, οι οποίοι είναι απαραίτητο να μοντελοποιηθούν μαζί, όπως για παράδειγμα η κατάστρωση ενός προγράμματος μαθημάτων, όπου η επιλογή ώρας, καθηγητή και αίθουσας σχετίζονται άμεσα.[6]

ΣΥΜΒΟΛΟΓΡΑΦΙΑ ΠΡΟΒΛΗΜΑΤΩΝ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

α/β/γ

όπου:

α= αριθμός μηχανημάτων / δομή προβλήματος .

β= χαρακτηριστικά των εργασιών και περιορισμοί του προβλήματος (τρόπος επεξεργασίας των εργασιών)

γ= κριτήριο βέλτιστου (αντικειμενική συνάρτηση).[7][9]

ΛΕΙΤΟΥΡΓΙΕΣ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Για ένα σύστημα παραγωγής στο οποίο πρόκειται να εφαρμοστεί χρονικός προγραμματισμός για τις διάφορες διεργασίες και πόρους που περικλείει. Οι λειτουργίες που πρέπει να εκτελεστούν για την εφαρμογή του προγραμματισμού μπορούν να συνοψισθούν στα παρακάτω:

1. Ανάθεση των παραγγελιών, του εξοπλισμού και των ανθρωπίνων πόρων στο κέντρο εργασίας, από το οποίο θα πραγματοποιηθεί η εκτέλεση των εργασιών
2. Η δρομολόγηση της σειράς εκτέλεσης των εργασιών κατατάσσοντας τις εργασίες κατά προτεραιότητα στο σύστημα.
3. Κατανομή της εκτέλεσης των προγραμματισμένων εργασιών (dispatching).
4. Έλεγχος της παραγωγικής διαδικασίας (shop-floorcontrol), η οποία περιλαμβάνει:

   Α) Ανάλυση της κατάστασης και έλεγχος της εξέλιξης των εργασιών κατά τη διάρκεια εκτέλεσης τους.

   Β) Γρηγορότερη πραγματοποίηση καθυστερημένων και κρίσιμων εργασιών.

Ο υπεύθυνος για τον χρονικό προγραμματισμό του συστήματος ενός κέντρου εργασίας είναι αυτός που επιλέγει και ταξινομεί τις διαθέσιμες εργασίες στις διάφορες θέσεις εργασίας του κέντρου. Για την λήψη αυτών των αποφάσεων ο υπεύθυνος λαμβάνει υπόψη τις λειτουργίες και τις απαιτήσεις δρομολόγησης της κάθε εργασίας, την κατάσταση των υπαρχόντων εργασιών σε κάθε θέση εργασίας, το χρόνο αναμονής της κάθε θέσης, τις προτεραιότητες μιας εργασίας έναντι μιας άλλης, την διαθεσιμότητα των υλικών, πιθανές εισόδους νέων εργασιών στο σύστημα και φυσικά τους διαθέσιμους πόρους του κέντρου εργασίας σε εξοπλισμό και σε ανθρώπινο δυναμικό.[6]

ΣΤΟΧΟΙ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Οι στόχοι του χρονικού προγραμματισμού σε ένα κέντρο εργασίας συνοψίζονται στα παρακάτω:

- Σεβασμός προθεσμιών για τις ημερομηνίες παράδοσης .
- Ελαχιστοποίηση του μέσου χρόνου παραγωγής μιας παρτίδας ή μιας παραγγελίας .
- Ελαχιστοποίηση του χρόνου ή κόστους προετοιμασίας και ρύθμισης του εξοπλισμού του κέντρου.
- Ελαχιστοποίηση των εκκρεμούντων εργασιών στο σύστημα .
- Μεγιστοποίηση χρησιμοποίησης εξοπλισμού ή ανθρώπινου δυναμικού.[6]

ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ ΧΡΟΝΙΚΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Με τον όρο δρομολόγηση των εργασιών εννοείται η διαδικασία κατά την οποία γίνετε ο καθορισμός της σειράς με την οποία θα εκτελεστούν οι εργασίες στα αντίστοιχα κέντρα εργασίας. Η σειρά με την οποία πραγματοποιούνται οι εργασίες καθορίζεται βάση κάποιων κανόνων προτεραιότητας. Οι κανόνες προτεραιότητας μπορεί να είναι απλοί και το μόνο που χρειάζεται να είναι η σύγκριση ενός μόνο χαρακτηριστικού των εργασιών για παράδειγμα η απαιτούμενη ημερομηνία παράδοσής τους, πιο σύνθετοι και να απαιτούν την πραγματοποίηση πράξεων μεταξύ κάποιων χαρακτηριστικών των εργασιών, ή τέλος περίπλοκοι και να απαιτείται η χρησιμοποίηση υπολογιστικών μεθόδων. Οι σημαντικότεροι από αυτούς τους κανόνες περιγράφονται στις παρακάτω παραγράφους. Η αξιολόγηση των κανόνων προτεραιότητας των εργασιών πραγματοποιείται με την χρήση διάφορων κριτήριων απόδοσης, τα οποία εξαρτώνται από τον επιθυμητό στόχο του συστήματος. Σε κάποιες περιπτώσεις χρησιμοποιούνται παράλληλα περισσότερα από ένα κριτήρια. Σε γενικές γραμμές στον προγραμματισμό ενός παραγωγικού συστήματος ο σημαντικότερος παράγοντας που λαμβάνεται υπόψη είναι η όσο το δυνατόν καλύτερη εξυπηρέτηση των πελατών και η βέλτιστη εκμετάλλευση των παραγωγικών πόρων. Τα πιο συνηθισμένα κριτήρια απόδοσης που χρησιμοποιούνται για τη δρομολόγηση των εργασιών στο σύστημα είναι τα εξής:

1. Μέσος χρόνος ροής. Με το κριτήριο αυτό υπολογίζεται ο μέσος χρόνος που δαπανάται κατά την πραγματοποίηση μιας εργασίας στο σύστημα. Το κριτήριο αυτό χρησιμοποιείται όταν ο κύριος σκοπός μας είναι η γρηγορότερη εκτέλεση των εργασιών και τήρηση χαμηλών αποθεμάτων

2. Μέση βραδύτερη περάτωση. Αυτό το κριτήριο μετράει το μέσο χρόνο των καθυστερήσεων που πρόκειται να προκύψουν κατά την εκτέλεση των εργασιών σε σχέση με τους απαιτούμενους χρόνους παράδοσης του πελάτη. Σκοπός του κριτηρίου είναι η ελαχιστοποίηση της συνολικής επιβάρυνσης του συστήματος από την υπέρβαση των χρόνων παράδοσης.

.

3. Μέσος χρόνος αναμονής. Το κριτήριο αυτό μετράει το μέσο χρόνο αναμονής που δαπανά μια εργασία στο σύστημα μέχρι να αρχίσει η πραγματοποίηση της. Αυτό το κριτήριο χρησιμοποιείται όταν ο στόχος μας είναι η ελαχιστοποίηση του χρόνου αναμονής των εργασιών και των αποθεμάτων πρώτων υλών.

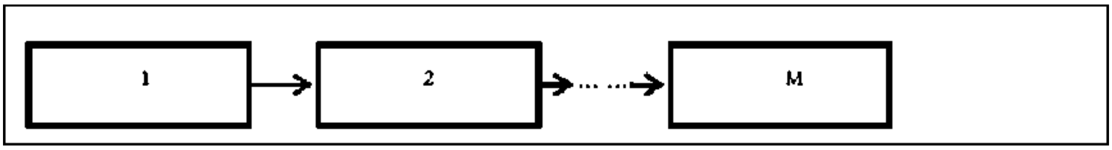
4. Μέγιστη βραδύτερη περάτωση. Το κριτήριο αυτό λαμβάνει υπόψη τη μέγιστη καθυστέρηση που προκύπτει κατά την εκτέλεση των εργασιών. Το συγκεκριμένο κριτήριο αποκτά σημαντικό ρόλο όταν η ποινή για κάθε χρονική μονάδα καθυστέρησης αυξάνει με το χρόνο καθυστέρησης.

5. Αριθμός αργοπορημένων εργασιών. Με το κριτήριο αυτό υπολογίζεται ο αριθμός των εργασιών που ολοκληρώνονται μετά από την επιθυμητή ημερομηνία παράδοσης τους και έχει ιδιαίτερη σημασία όταν στόχος είναι η ελαχιστοποίηση του αριθμού των δυσαρεστημένων πελατών λόγω καθυστερήσεων.[6]

ΣΥΣΤΗΜΑΤΑ ΠΑΡΑΓΩΓΗΣ FLOW-SHOP

Το πρόβλημα χρονικού προγραμματισμού συστημάτων παραγωγής συνεχούς ροής συνιστά ένα σημαντικό πρόβλημα χρονοπρογραμματισμού και έχει ερευνηθεί εκτενώς από τότε που προτάθηκε πρώτα από τον Johnson το 1954.

Τα συστήματα παραγωγής συνεχούς ροής (flow-shop) χρησιμοποιούνται για μαζική παραγωγή περιορισμένης ποικιλίας τυποποιημένων προϊόντων, όπως για παράδειγμα τα προϊόντα διατροφής. Συνήθως τα προϊόντα αυτά παράγονται σε γραμμές παράγωγης και ο τρόπος με τον όποιο παράγονται είναι σε μεγάλο βαθμό αυτοματοποιημένος. Ένα σύστημα flow-shop αποτελείται από m μηχανές (επεξεργαστές), όπου η εκτέλεση κάθε παραγγελίας n (έργου) περιλαμβάνει μέχρι m κατεργασίες (επεξεργασίες), μία σε κάθε μηχανή (επεξεργαστή).[8]



Υπάρχουν δύο κατηγορίες συστημάτων flow-shop. Η πρώτη ονομάζεται «καθαρό» σύστημα flow-shop, όπου όλες οι παρτίδες παραγωγής ενός προϊόντος για να εκτελεστούν περνούν από όλες τις μηχανές (επεξεργαστές), και η δεύτερη ονομάζεται «γενικό» flow-shop, όπου κάθε παρτίδα (έργο), αν και ακολουθεί την ίδια κατεύθυνση μέσα στο σύστημα, δεν περνάει υποχρεωτικά από όλες τις μηχανές (επεξεργαστές). Στην τελευταία περίπτωση όπου κάποιες εργασίες δεν απαιτείται να περάσουν από όλες τις μηχανές πάλι υπάρχει συνεχής ροή εφόσον η σειρά με την οποία επισκέπτονται τις υπόλοιπες μηχανές είναι η καθορισμένη.

Παράδειγμα καθαρού συστήματος flow-shop είναι μια γραμμή συναρμολόγησης, όπου η παραγωγή πραγματοποιείται από φάση σε φάση στην ίδια πάντα κατεύθυνση και περνώντας από όλους τους σταθμούς παραγωγής. Σε άλλες περιπτώσεις, όπως στην περίπτωση παραγωγής ενδυμάτων, η ίδια διαδοχή κατεργασιών (επεξεργασιών) απαιτείται για ένα μεγάλο αριθμό κομματιών μιας παραγγελίας, ενώ η διαδοχή αυτή μπορεί να αλλάζει από παραγγελία σε παραγγελία. Ένα τέτοιο σύστημα θεωρείται γενικό σύστημα flow-shop.

Όπως αναφέρθηκε και προηγούμενος για την κατηγοριοποίηση του προβλήματος παραγωγής συνεχούς ροής χρησιμοποιούμε τον παρακάτω συμβολισμό, όπως χρησιμοποιήθηκε από τους Graham et al. το 1979[7][9], κατά τον οποίο το πρόβλημα χωρίζεται σε τρία πεδία:

α/β/γ   όπου

α: αριθμός μηχανημάτων / δομή προβλήματος

β: χαρακτηριστικά των εργασιών και περιορισμοί του προβλήματος (τρόπος επεξεργασίας των εργασιών)

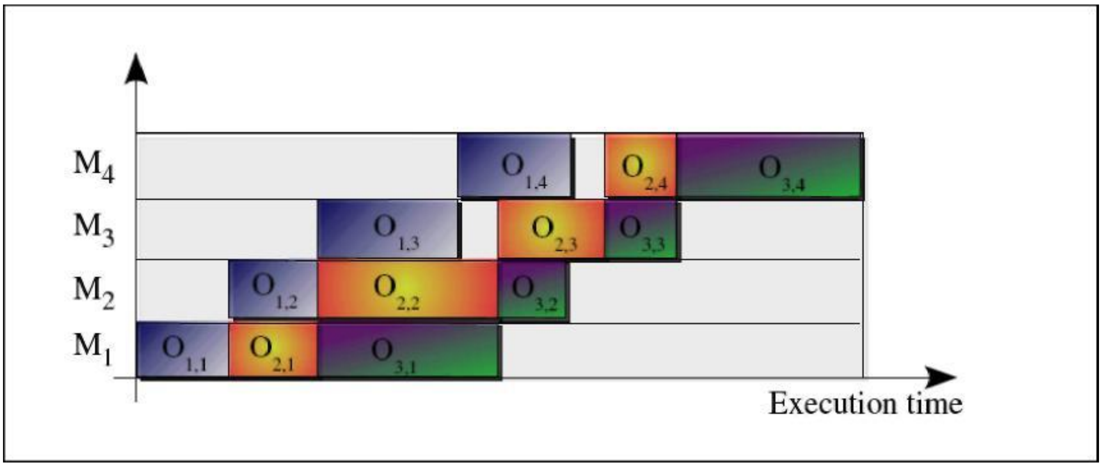
γ: κριτήριο βέλτιστου (αντικειμενική συνάρτηση), π.χ. χρόνος περάτωσης

Στην παρούσα εργασία θα εξεταστεί ένα πρόβλημα αντιμετάθεσης χρονικού προγραμματισμού συστήματος συνεχούς ροής (Permutation Flow-Shop Scheduling Problem - PFSSP) με σκοπό την εφαρμογή ενός βέλτιστου χρονοπρογραμματισμού για N εργασίες σε M μηχανές.

Αναλυτικότερα, το σύστημα παραγωγής συνεχούς ροής αποτελείται από n εργασίες-jobs (i=1,2,….,n) και από m μηχανές-machines σε σειρά (j=1,2,….,m). Επιπλέον κάθε i εργασία αποτελείται από m κατεργασίες-operations και η $j^{οστή}$ κατεργασία πρέπει να υλοποιηθεί στην j μηχανή. Έτσι μία εργασία μπορεί να ξεκινήσει στη μηχανή j εφόσον έχει ολοκληρωθεί προηγουμένως στην j-1 μηχανή και η μηχανή j είναι ελεύθερη. Ο χρόνος επεξεργασίας της κάθε κατεργασίας (operation) σε κάθε μηχανή ορίζεται ίσος με p i,j . [10]

Μπορούμε να συμβολίσουμε τις παραπάνω προϋποθέσεις ως εξής:

- $J_i=(O_{i,1}, O_{i,2},\ldots\ldots O_{i,m})$, όπου η $O_{i,j}$ αντιπροσωπεύει την $j^{οστή}$ κατεργασία της εργασίας J_i
- Η $O_{i,j}$ κατεργασία πρέπει να υποστεί επεξεργασία στην M_j μηχανή
- Για κάθε κατεργασία υπάρχει ο αντίστοιχος χρόνος $p_{i,j}$.

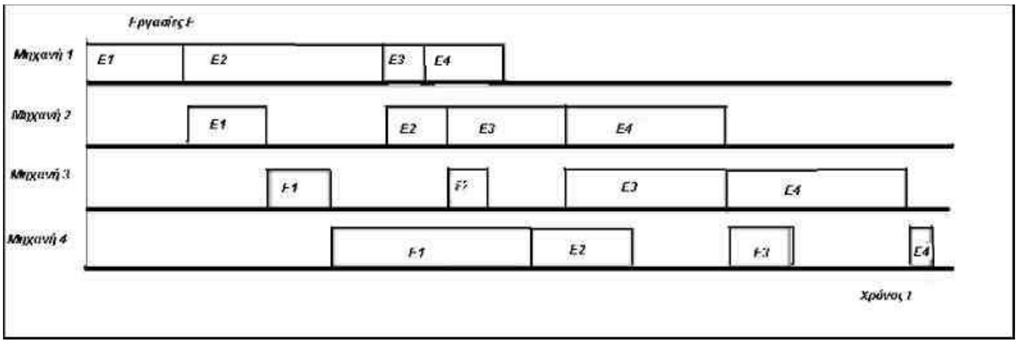


Σε ένα πρόβλημα αντιμετάθεσης παραγωγής συνεχούς ροής (Permutation Flow-Shop Scheduling Problem - PFSSP) η αλληλουχία των εργασιών προς επεξεργασία είναι η ίδια σε κάθε μηχανή. Αν η μία εργασία βρίσκεται στην $i^{οστή}$ θέση στη μηχανή 1, τότε αυτή η εργασία θα βρίσκεται σε αυτή τη θέση σε όλες τις μηχανές.[10]

Τα παραπάνω μπορούν να εξηγηθούν στα δύο διαγράμματα που ακολουθούν που συνιστούν σχηματική απεικόνιση του προβλήματος που περιγράφεται στον πίνακα 1.

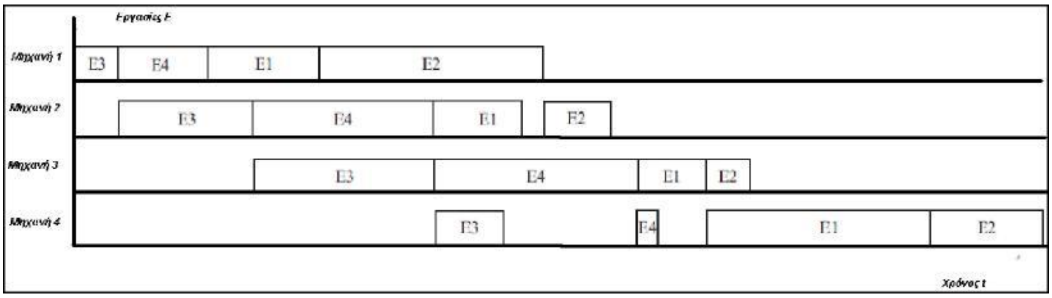
Εργασίες / Μηχανές	Ε1	Ε2	Ε3	Ε4
Μ1	$O_{1,1}$	$O_{2,1}$	$O_{3,1}$	$O_{4,1}$
Μ2	$O_{1,2}$	$O_{2,2}$	$O_{3,2}$	$O_{4,2}$
Μ3	$O_{1,3}$	$O_{2,3}$	$O_{3,3}$	$O_{4,3}$
Μ4	$O_{1,4}$	$O_{2,4}$	$O_{3,4}$	$O_{4,4}$

Οι μηχανές έχουν μία καθορισμένη σειρά: Μ1, Μ2, Μ3 και Μ4. Όταν εκτελεσθεί και η τελευταία διεργασία τότε ολοκληρώνεται η εκάστοτε εργασία. Δηλαδή όταν περάσουν όλες οι εργασίες και από την τελευταία μηχανή, την Μ4, θα έχουν ολοκληρωθεί όλες οι εργασίες.



Εικόνα 3: Διάγραμμα Gantt για σειρά εκτέλεσης εργασιών Ε1,Ε2,Ε3, Ε4

Στο επόμενο διάγραμμα απεικονίζονται πάλι οι χρονικές διάρκειες των διεργασιών, όμως αυτή τη φορά η σειρά πραγματοποίησης των εργασιών είναι διαφορετική: Ε3, Ε4, Ε1, Ε2.



Εικόνα 4: Διάγραμμα Gantt για σειρά εκτέλεσης εργασιών Ε3,Ε4,Ε1,Ε2

Στην παρούσα εργασία του κριτήριο του βέλτιστου είναι η ελαχιστοποίηση του χρόνου περάτωσης της τελευταίας εργασίας (makespan). Στόχος του προβλήματος είναι η αναζήτηση της βέλτιστης μετάθεσης (permutation), δηλαδή αυτής της αλληλουχίας των εργασιών που θα μειώσει σε έναν καλό βαθμό τον χρόνο περάτωσης των εργασιών .[10]

Έστω Π={Π1,Π2,…Πn} είναι η αλληλουχία των εργασιών. Η ολοκλήρωση μιας εργασίας J_i στη μηχανή M_j συμβολίζεται με $C(\pi_i,j)$ και ορίζεται ως ακολούθως:

$C(\pi_i,j)= \max \{C(\pi_{i-1},j), C(\pi_i,j-1)\}+p_{\pi i,j}$

Με i=1,2,….,n και j=1,2,……m

Ο χρόνος λήξης του προγράμματος μιας συγκεκριμένης αντιμετάθεσης ορίζεται ως $C_{max}(\pi)=C(\pi_n,m)$, δηλαδή ο χρόνος ολοκλήρωσης των εργασιών είναι ο χρόνος περάτωσης της τελευταίας εργασίας $J\pi_n$ στην τελευταία μηχανή M_m.[10]

Το πρόβλημα παραγωγής συνεχούς ροής που μελετάται στην εργασία αυτή είναι της μορφής (βάσει του συμβολισμού των Graham)[9]:

N/M/F,Permu/Cmax

Το N/M εκφράζει τη δομή του προβλήματος-αριθμο μηχανημάτων όπως μπορούμε να δούμε και από το πρώτο επίπεδο διαχωρισμού του προβλήματος, όπου N εργασίες εκτελούνται σε M μηχανές . Στο δεύτερο επίπεδο το F μας δηλώνει ότι το πρόβλημα ανήκει στην κατηγορία των προβλημάτων συνεχούς ροής και το Permu ότι οι λύσεις αναφέρονται σε προβλήματα μετάθεσης .Τέλος το Cmax μας δηλώνει το κριτήριο βέλτιστου δηλαδή την ελαχιστοποίηση του χρόνου περάτωσης των εργασιών.

Για την επίλυση του παραπάνω προβλήματος γίνονται οι εξής υποθέσεις: [10][11]

- Όλες οι εργασίες Ν είναι ανεξάρτητες και διαθέσιμες για επεξεργασία τη χρονική στιγμή 0.
- Η κάθε εργασία απαιτεί Μ διεργασίες και η κάθε διεργασία απαιτεί διαφορετική μηχανή.
- Μια εργασία είναι διαθέσιμη σε μία μηχανή, μόνο αν η επεξεργασία της έχει ολοκληρωθεί στην αμέσως προηγούμενη. Όλες οι εργασίες είναι διαθέσιμες στην πρώτη μηχανή του συστήματος.
- Κάθε μηχανή μπορεί να επεξεργάζεται το πολύ μία εργασία σε δεδομένη χρονική στιγμή.
- Η σειρά με την οποία εκτελούνται οι εργασίες είναι ίδια για όλες τις μηχανές.
- . Οι χρόνοι εξάρμωσης των μηχανών συμπεριλαμβάνονται στους χρόνους επεξεργασίας.
- Οι χρόνοι επεξεργασίας των διεργασιών από τις μηχανές είναι γνωστοί από την αρχή.
- Όλες οι μηχανές είναι πάντα διαθέσιμες.
- Όταν ξεκινήσει η διεργασία συνεχίζει χωρίς καμία διακοπή.

Ο χρονικός προγραμματισμός σε παραγωγικά συστήματα flow-shop αφορά την εύρεση της βέλτιστης μεθόδου δρομολόγησης των εργασιών, δηλαδή τέτοιας που να ικανοποιεί στο μέγιστο βαθμό τα επιλεγμένα κριτήρια απόδοσης, λαμβάνοντας πάντα υπόψη τους υπάρχοντες περιορισμούς. Όμως, ακόμα και για σχετικά απλά προβλήματα δεν είναι εύκολο να βρεθούν βέλτιστες λύσεις. Για αυτόν τον λόγο για την επίλυση του συγκεκριμένου προβλήματος χρησιμοποιήθηκε ο αλγόριθμος μουσικής αρμονίας και το αποτελέσματα τα όποια πρόεκυψαν εμφανίζουν πολύ μικρές αποκλίσεις σε σχέση με τα ήδη υπάρχοντα αποτελέσματα της βιβλιογραφίας.[7]

2η ΕΝΟΤΗΤΑ

ΔΟΜΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Όπως έχει ήδη αναφερθεί ο σκοπός του προγράμματος μας είναι η επίλυση του προβλήματος του flow-shop δηλαδή να τοποθετήσει όλες τις εργασίες στην βέλτιστη σειρά ώστε να ελαχιστοποιηθεί ο συνολικός χρόνος παράγωγης. Αυτό επιτυγχάνεται με την βοήθεια του αλγορίθμου αναζήτησης της μουσικής αρμονίας ο όποιος γεννά πληθυσμούς λύσεων οι οποίοι ελέγχονται από τον υπολογιστή κατά το κριτήριο βέλτιστου σε σχέση με τους προηγούμενους πληθυσμούς λύσεων με αποτέλεσμα ύστερα από έναν συγκεκριμένο αριθμό επαναλήψεων να έχουμε συγκεντρώσει τους βέλτιστους συνδυασμούς λύσεων.

Αρχικά παραθέτω ολόκληρο το πρόγραμμα και στην συνέχεια θα αναλυθεί κομμάτι κομμάτι κάθε μια από τις λειτουργίες του.

```
A=…;
%to M tha einai o arithmos twn mhxanhmatwn
%to N tha einai o arithmos twn ergasiwn
%zhtaw apo ton xrhsth na mou ta dwsei. edw ta dinw monos mou
%analoga me ton pinaka pou diavazw allazw kai auta
M=….;
N=…..;
%o pinakas pou antiproswpeuei tis tuxaies luseis mou exei megethos 15
%kai ton onomazw T
MEGETHOS=15;
%dinw poses epanalhpseis thelw na kanei o algorithos mou
Για i από 1 μέχρι MEGETHOS

   Για j από 1μεχρι N
      DIANYSMA[i,j]=11;
   Τέλος_επανάληψης
 Τέλος_επανάληψης
Για i από 1μέχρι MEGETHOS

   APOTEL[i]=1
Τέλος_επανάληψης
```

```
 Για i από 1μεχρι MEGETHOS
     D=randperm(N)
     Για j από 1 μέχρι N
        T[i,j]=D[j]
     Τέλος_επανάληψης
Τέλος_επανάληψης
HMCR=0.9;
PAR=0.3;
Για n από 1 μέχρι 4000
   Για i από 1μέχρι MEGETHOS
   kritirio=rand
   Αν kritirio>HMCR τότε
      D=randperm(N)
      Για j από 1 μέχρι N
         NEW[i,j]=D[j]
      Τέλος_επανάληψης
   Αλλιώς
      Για j από 1 μέχρι N
         random=randi(MEGETHOS)
         NEW[i,j]=T[random,j]
      Τέλος_επανάληψης
      Για k από 1 μέχρι N
         plithos=0
         Για j από 1 μέχρι N
            Αν NEW[i,j]=k τότε
               plithos=plithos+1
            Τέλος_αν
         Τέλος_επανάληψης
         DIANYSMA[i,k]=plithos
      Τέλος_αν
      Για j από 1 μέχρι N
         Όσο DIANYSMA[i,j]>1 επανέλαβε
            metritis1=0
            Για l από 1 μέχρι N
               Αν metritis1<1 τότε
                  Αν DIANYSMA[i,l]=0 τότε
                     DIANYSMA[i,l]=1
                     DIANYSMA[i,j]=DIANYSMA[i,j]-1
                     metritis1=metritis1+1
                     metritis2=0;
                     Για k=1:N
                        Αν metritis2==0 τότε
                           Αν NEW[i,k]==j
```

```
                        NEW[i,k]=l
                        metritis2=metritis2+1
                      Τέλος_αν
                    Τέλος_αν
                  Τέλος_επανάληψης
                Τέλος_αν
              Τέλος_αν
            Τέλος_επανάληψης
          Τέλος_επανάληψης
        Τέλος_επανάληψης
      Τέλος_αν
      Τέλος_επανάληψης
      Για i από 1μεχρι MEGETHOS
        Για j από 1μέχρι N
          NEW1[i,j]=NEW[i,j]
        Τέλος_επανάληψης
      Τέλος_επανάληψης
      Για i από 1 μέχρι MEGETHOS
        kritirio2=rand;
        Αν kritirio2<PAR τότε
          random=randi(N-1);
          NEW1[i,random]=NEW[i,random+1]
          NEW1[i,random+1]=NEW[i,random]
        Τέλος_αν
      Τέλος_επανάληψης
%edw exw ena arxiko tuxaio pinaka T kai ena neo ton NEW1
 %thelw na sugkrinw ta apotelesmata twn lusewn tous
 %an tou NEW einai kapoio apotelesma kalutero tha to
%antikatastisw ston T
   test1=ones[M,N]
   test2=ones[M,N]
   Για k από 1 μέχρι MEGETHOS
      Για κάθε εργασία j από 1 μέχρι N
        Για i από 1 μέχρι M
          test1[i,j]=A[i,T[k,j]];
          test2(i,j)=A[i,NEW1[k,j]];
        Τέλος_επανάληψης
      Τέλος_επανάληψης
      testA=ones[M,N]
      testB=ones[M,N]
      testA[1,1]=test1[1,1]
      testB[1,1]=test2[1,1]
      Για i από 2 μέχρι M
```

```
            testA[i,1]=testA[i-1,1]+test1[i,1]
            testB(i,1)=testB[i-1,1]+test2[i,1]
        Τέλος_επανάληψης
        Για j από 2 μέχρι N
            testA[1,j]=testA[1,j-1]+test1[1,j]
            testB[1,j]=testB[1,j-1]+test2[1,j]
        Τέλος_επανάληψης
        Για j από 2 μέχρι N
            Για i από 2 μέχρι M
                TIMH1=test1[i,j]+testA[i,j-1]
                TIMH2=test1[i,j]+testA[i-1,j]
                TIMH3=test2[i,j]+testB[i,j-1]
                TIMH4=test2[i,j]+testB[i-1,j]
                Αν TIMH1>TIMH2 τότε
                    testA[i,j]=TIMH1
                Αλλιώς
                    testA[i,j]=TIMH2
                Τέλος_αν
                Αν TIMH3>TIMH4 τότε
                    testB[i,j]=TIMH3
                Αλλιώς
                    testB[i,j]=TIMH4
                Τέλος_αν
            Τέλος_επανάληψης
        Τέλος_επανάληψης
        ApotARX=testA[M,N]
        ApotNEW1=testB[M,N]
        Αν ApotNEW1<ApotARX τότε
            Για z από 1μεχρι N
                T[k,z]=NEW1[k,z]
            Τέλος_επανάληψης
            APOTEL[k]=ApotNEW1;
        Αλλιώς
            APOTEL[k]=ApotARX;
        Τέλος_αν
    Τέλος_επανάληψης
Τέλος_επανάληψης
beltisthlush=APOTEL[1];
Για  t από 2 μέχρι MEGETHOS
    Αν APOTEL[t]<beltisthlush
        beltisthlush=APOTEL[t];
    Τέλος_αν
Τέλος_επανάληψης
```

Εμφάνισε beltisthlush

Ως Μ το πρόγραμμα δέχεται τον αριθμό των μηχανημάτων.
Ως Ν το πρόγραμμα δέχεται τον αριθμό των εργασιών.

Ως Ρ το πρόγραμμα δέχεται έναν πινάκα ο όποιος κατά αντιστοιχία γραμμών και στηλών μας δίνει τον χρόνο που απαιτεί η ολοκλήρωση της κάθε εργασίας στην αντίστοιχη μηχανή. Αυτό υλοποιείται με την βοήθεια του Excel.

Το μεγεθος του πινακα λυσεων εχει ορισθει 15.

Σε αυτό το σημείο παραθέτω κάθε κομμάτι του αλγορίθμου ξεχωριστά και αναλύω τη λειτουργία του πάνω στο πρόβλημα μας.

```
Για i από 1 μέχρι MEGETHOS

   Για j από 1μεχρι N
      DIANYSMA[i,j]=11;
   Τέλος_επανάληψης
 Τέλος_επανάληψης
Για i από 1μεχρι MEGETHOS

   APOTEL[i]=1
Τέλος_επανάληψης
```

Αρχικά δίνω κάποιες αρχικές τιμές σε δύο διανύσματα τα οποία θα μας χρησιμεύσουν κατά την επίλυση του αλγορίθμου.

```
Για i από 1μεχρι MEGETHOS
      D=randperm(N)
      Για j από 1 μέχρι N
         T[i,j]=D[j]
      Τέλος_επανάληψης
Τέλος_επανάληψης
```

Εδώ γεμίζω τον πινάκα λύσεων με τυχαίες λύσεις που αντιστοιχούν σε τυχαίες σειρές με τις οποίες οι εργασίες θα περάσουν από τα μηχανήματα.

Ο πίνακας με τις τυχαίες αρχικές λύσεις είναι ο T, έχει μέγεθος 15 και συμπληρώνεται με την βοήθεια ένα βοηθητικού διανύσματος και της εντολής randperm η οποία δίνει στο διάνυσμα τυχαίες τιμές από το 1 έως το N που είναι ο αριθμός των εργασιών.

HMCR=0.9
PAR=0.3

Το HMCR και το PAR αντιπροσωπεύουν 2 μεταβλητές απόφασης όπως έχουν αναλυθεί και στο πρώτο μέρος τις εργασίας.

Ο αριθμός επαναλήψεων N έχει ορισθει τέσσερεις χιλιάδες.

N=4000

```
Για i από 1μεχρι MEGETHOS
kritirio=rand
Αν kritirio>HMCR τότε
   D=randperm(N)
   Για j από 1 μέχρι N
      NEW[i,j]=D[j]
   Τέλος_επανάληψης
Αλλιώς
   Για j από 1 μέχρι N
      random=randi(MEGETHOS)
      NEW[i,j]=T[random,j]
   Τέλος_επανάληψης
   Για k από 1 μέχρι N
      plithos=0
      Για j από 1 μέχρι N
         Αν NEW[i,j]=k τότε
            plithos=plithos+1
         Τέλος_αν
      Τέλος_επανάληψης
      DIANYSMA[i,k]=plithos
   Τέλος_αν
   Για j από 1 μέχρι N
      Όσο DIANYSMA[i,j]>1 επανέλαβε
         metritis1=0
         Για l από 1 μέχρι N
            Αν metritis1<1 τότε
               Αν DIANYSMA[i,l]=0 τότε
                  DIANYSMA[i,l]=1
```

```
                        DIANYSMA[i,j]=DIANYSMA[i,j]-1
                        metritis1=metritis1+1
                        metritis2=0;
                        Για k=1:N
                           Αν metritis2==0 τότε
                              Αν NEW[i,k]==j
                                 NEW[i,k]=l
                                 metritis2=metritis2+1
                              Τέλος_αν
                           Τέλος_αν
                        Τέλος_επανάληψης
                     Τέλος_αν
                  Τέλος_αν
               Τέλος_επανάληψης
            Τέλος_επανάληψης
         Τέλος_επανάληψης
      Τέλος_αν
      Τέλος_επανάληψης
```

Σε αυτό το σημείο για κάθε μια τυχαία λύση του πίνακα T γίνεται ένας έλεγχος . Μια τυχαία μεταβλητή kritirio παίρνει τυχαία τιμές από το 0 έως το 1 και εάν η τιμή της μεταβλητής kritirio είναι μεγαλύτερη από την τιμή που έχει δοθεί στο HMCR τότε στον νέο πίνακα NEW δημιουργείται μια νέα εντελώς τυχαία λύση. Εάν η τιμή της μεταβλητής είναι μικρότερη του HMCR τότε το πρόγραμμα μας δημιουργεί μια λύση με βάση την εμπειρία δηλαδή των ήδη υπάρχοντων λύσεων που βρίσκονται στον πίνακα μας . Αυτό γίνεται με τον εξής τρόπο :

Το κάθε στοιχείο της νέας λύσης θα αποτελείται από τα στοιχεία που υπάρχουν στην αντίστοιχη στήλη του ήδη υπάρχοντος πίνακα λύσεων.
Δηλαδή για να γίνει κατανοητό η εργασία που θα μπορεί να μπει για παράδειγμα στην θέση 1 του διανύσματος της νέας λύσης θα πρέπει να υπάρχει ήδη στην 1η στήλη του πίνακα λύσεων και η επιλογή της γίνεται εντελώς τυχαία. Όμως λόγο της τυχαίας επιλογής μπορεί στο διάνυσμα της νέας λύσης μας να εμφανιστεί η ίδια εργασία 2 η περισσότερες φορές κάτι που στην πραγματικότητα είναι αδύνατο αφού αυτή η εργασία έχει ήδη περάσει από τα μηχανήματα και έχει ολοκληρωθεί.

Για να αντιμετωπίσουμε αυτό το πρόβλημα δημιουργούμε ένα βοηθητικό διάνυσμα με όνομα DIANYSMA το οποίο καταγράφει πόσες φορές εμφανίζεται η κάθε εργασία . Σκοπός μας είναι να φτάσουμε στο σημείο ώστε το διάνυσμα να αναγράφει ότι όλες οι εργασίες εμφανίζονται μια και μόνο φορά.

```
Για j από 1 μέχρι N
      Όσο DIANYSMA[i,j]>1 επανέλαβε
         metritis1=0
         Για l από 1 μέχρι N
            Αν metritis1<1 τότε
               Αν DIANYSMA[i,l]=0 τότε
                  DIANYSMA[i,l]=1
                  DIANYSMA[i,j]=DIANYSMA[i,j]-1
                  metritis1=metritis1+1
                  metritis2=0;
                  Για k=1:N
                     Αν metritis2==0 τότε
                        Αν NEW[i,k]==j
                           NEW[i,k]=l
                           metritis2=metritis2+1
                        Τέλος_αν
                     Τέλος_αν
                  Τέλος_επανάληψης
               Τέλος_αν
            Τέλος_αν
         Τέλος_επανάληψης
      Τέλος_επανάληψης
   Τέλος_επανάληψης
```

Σε αυτό το κομμάτι του κώδικα επιτυγχάνεται ακριβώς αυτό το πράγμα δηλαδή φέρνουμε την νέα λύση που πρόεκυψε βάση της εμπειρίας σε μια αποδεκτή μορφή όπου η κάθε εργασία υπάρχει μόνο μια φορά.

Αυτό γίνεται με τον ακόλουθο τρόπο:

Όσο το διάνυσμα δείχνει πως μια εργασία υπάρχει πάνω από μια φορά το πρόγραμμα πάει και βρίσκει την πρώτη εργασία η οποία έχει τιμή 0 στο διάνυσμα και της δίνει τιμή 1. Ύστερα υλοποιείται ένας ακόμη έλεγχος και στην πρώτη θέση της νέας λύσης όπου εμφανίζεται η εργασία j η οποία είναι η εργασία που εμφανίζεται παραπάνω από μια φορά, αντικαθίσταται με την εργασία l η οποία είναι η πρώτη κατά σειρά εργασία που δεν έχει εμφανιστεί καμία φορά και με αυτήν την επαναληπτική διαδικασία συμπληρώνεται το επιθυμητό διάνυσμα της νέας λύσης μας.

```
Για i από 1μεχρι MEGETHOS
    Για j από 1μέχρι N
      NEW1[i,j]=NEW[i,j]
    Τέλος_επανάληψης
  Τέλος_επανάληψης
  Για i από 1 μέχρι MEGETHOS
    kritirio2=rand;
    Αν kritirio2<PAR τότε
      random=randi(N-1);
      NEW1[i,random]=NEW[i,random+1]
      NEW1[i,random+1]=NEW[i,random]
    Τέλος_αν
  Τέλος_επανάληψης
```

Εδώ γίνεται ένας έλεγχος σε μια μεταβλητή απόφασης . Η μεταβλητή kritirio2 παίρνει τυχαίες τιμές από το 0 έως το 1. Αν η μεταβλητή kritirio2 έχει μικρότερη τιμή από το PAR τότε πραγματοποιείται τοπική αναζήτηση στην λύση που έχει προκύψει βάση της εμπειρίας και γίνεται αντιμετάθεση της θέσης μιας τυχαίας εργασίας με την ακριβώς επόμενη της .

```
   test1=ones[M,N]
 test2=ones[M,N]
 Για k από 1 μέχρι MEGETHOS
    Για κάθε εργασία j από 1 μέχρι N
      Για i από 1 μέχρι M
        test1[i,j]=A[i,T[k,j]];
        test2(i,j)=A[i,NEW1[k,j]];
      Τέλος_επανάληψης
    Τέλος_επανάληψης
    testA=ones[M,N]
    testB=ones[M,N]
    testA[1,1]=test1[1,1]
```

```
        testB[1,1]=test2[1,1]
        Για i από 2 μέχρι M
           testA[i,1]=testA[i-1,1]+test1[i,1]
           testB(i,1)=testB[i-1,1]+test2[i,1]
        Τέλος_επανάληψης
        Για j από 2 μέχρι N
           testA[1,j]=testA[1,j-1]+test1[1,j]
           testB[1,j]=testB[1,j-1]+test2[1,j]
        Τέλος_επανάληψης
        Για j από 2 μέχρι N
           Για i από 2 μέχρι M
              TIMH1=test1[i,j]+testA[i,j-1]
              TIMH2=test1[i,j]+testA[i-1,j]
              TIMH3=test2[i,j]+testB[i,j-1]
              TIMH4=test2[i,j]+testB[i-1,j]
              Αν TIMH1>TIMH2 τότε
                 testA[i,j]=TIMH1
              Αλλιώς
                 testA[i,j]=TIMH2
              Τέλος_αν
              Αν TIMH3>TIMH4 τότε
                 testB[i,j]=TIMH3
              Αλλιώς
                 testB[i,j]=TIMH4
              Τέλος_αν
           Τέλος_επανάληψης
        Τέλος_επανάληψης
        ApotARX=testA[M,N]
        ApotNEW1=testB[M,N]
        Αν ApotNEW1<ApotARX τότε
           Για z από 1μεχρι N
              T[k,z]=NEW1[k,z]
           Τέλος_επανάληψης
           APOTEL[k]=ApotNEW1;
        Αλλιώς
           APOTEL[k]=ApotARX;
        Τέλος_αν
     Τέλος_επανάληψης
Τέλος_επανάληψης
```

Στο παραπάνω κομμάτι γίνεται η σύγκριση των λύσεων του παλαιού πίνακα T και του νέου πίνακα NEW ως προς το κριτήριο βέλτιστου που στην συγκεκριμένη περίπτωση είναι ο πινάκας makespan. Οπότε για κάθε νέα λύση ελέγχεται αν το τελευταίο στοιχειό του πίνακα makespan είναι μικρότερο από αυτό της αντίστοιχης παλαιάς λύσης , αν είναι μικρότερο τότε οι παλαιά λύση του πίνακα T αντικαθίσταται με την νέα λύση του πίνακα NEW και κατά αυτόν τον τρόπο ο πίνακας T όλο βελτιώνεται ως προς τα αποτελέσματα των λύσεων του σύμφωνα με το κριτήριο βέλτιστου.

```
beltisthlush=APOTEL[1];
Για  t από 2 μέχρι MEGETHOS
   Αν APOTEL[t]<beltisthlush
      beltisthlush=APOTEL[t];
   Τέλος_αν
Τέλος_επανάληψης
Εμφάνισε beltisthlush
```

Στο τελευταίο κομμάτι του κώδικα γίνεται έλεγχος για το ποια λύση του τελικού πίνακα είναι η βέλτιστη και ύστερα μας εμφανίζει τον συνολικό βέλτιστο χρόνο από τον πίνακα makespan που έχει βρεθεί.

ΑΞΙΟΛΟΓΗΣΗ ΚΡΙΤΗΡΙΟΥ ΒΕΛΤΙΣΤΟΥ (ΠΙΝΑΚΑΣ MAKESPAN)

Για την καλύτερη κατανόηση του πίνακα makespan παρακάτω εξηγείται αναλυτικά ο τρόπος με τον οποίο συμπληρώνεται.

Για παράδειγμα έχουμε τον παρακάτω αρχικό πίνακα Α:

5	10	2	4
4	3	6	8
3	2	8	9
10	5	3	1

Στον παραπάνω πίνακα οι γραμμές i αναφέρονται στα μηχανήματα ενώ οι στήλες j αναφέρονται στις εργασίες . Τα στοιχειά του πίνακα A(i,j) μας δηλώνουν τον χρόνο που χρειάζεται το μηχάνημα i ώστε να πραγματοποιήσει την εργασία j.

Το κριτήριο προς ελαχιστοποίηση είναι το makespan, δηλαδή ο χρόνος περάτωσης της τελευταίας εργασίας. Η αντικειμενική συνάρτηση του προβλήματος είναι:

min(Cmax) = min(max(CiM)), i = 1,2,…,n

Όπου Cmax ο χρόνος λήξης του προγράμματος (makespan) και CiM ο χρόνος περάτωσης κάθε εργασίας στην τελευταία μηχανή M.

Υπολογίζω τον χρόνο περάτωσης της κάθε εργασίας και δημιουργώ τον πίνακα κόστους C του οποίου οι σειρές συμβολίζουν τις μηχανές και οι στήλες τις εργασίες. Στην πρώτη μηχανή οι εργασίες εκτελούνται σειριακά χωρίς να περιμένει η μηχανή για την επόμενη εργασία. Επίσης, η πρώτη εργασία εκτελείται χωρίς να περιμένει, σε όλες τις μηχανές. Για όλες τις υπόλοιπες περιπτώσεις όμως θα πρέπει να υπολογιστεί αν η μηχανή είναι διαθέσιμη και περιμένει την εργασία ή αν η εργασία είναι αναγκασμένη να περιμένει τη μηχανή: ci,j = max(ci-1,j,ci,j-1) + pi,j (όπου pi,j οι χρόνοι της εργασίας j στην μηχανή i).

Για να γίνει πιο κατανοητό, η διαδικασία με την οποία συμπληρώνεται ο πίνακας C είναι η παρακάτω:

Η πρώτη γραμμή και στήλη συμπληρώνεται με την απλή πρόσθεση κατά σειρά των στοιχείων (1,j) και (i,1) αντίστοιχα . Αφού συμπληρωθούν η πρώτη σειρά και η πρώτη στήλη του πίνακα C, η διαδικασία συμπλήρωσης του πίνακα αλλάζει . Από εδώ και πέρα η διαδικασία με την οποία γίνεται η συμπλήρωση κάθε στοιχείου C(i,j) είναι η εξής: Αρχικά προσθέτουμε το A(i,j) με το με το C(i-1,j) έπειτα προσθέτουμε το A(i,j) με το C(i,j-1) από τα δυο αθροίσματα κρατάμε το μεγαλύτερο και συμπληρώνουμε το στοιχείο C(i,j) του πίνακα C.

Ο πίνακας κόστους οπότε του πίνακα A με σειρά εργασιών E1, E2, E3, E4 θα είναι:

C=

5	15	17	21
9	18	24	32
12	20	32	41
22	27	35	42

Το στοιχείο προς ελαχιστοποίηση, δηλαδή το makespan, είναι το στοιχείο της τελευταίας σειράς και της τελευταίας γραμμής. Δηλαδή το makespan του παραδείγματος είναι Cmax = 42.

Έτσι λοιπόν η αξιολόγηση της αντικειμενικής συνάρτησης γίνεται με βάση τον πίνακα C.

ΕΦΑΡΜΟΓΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΑΝΑΖΗΤΗΣΗΣ ΤΗΣ ΜΟΥΣΙΚΗΣ ΑΡΜΟΝΙΑΣ

Στην συγκεκριμένη ενότητα παραθέτω τους πίνακες με τα αποτελέσματα του αλγορίθμου μουσικής αρμονίας πάνω στους ήδη υπάρχοντες πίνακες προβλημάτων flowshop . Οι πίνακες αυτοί είναι, οι πίνακες που δέχεται το πρόγραμμα και αντιπροσωπεύουν την ώρα που χρειάζεται το κάθε μηχάνημα για την επίλυση της κάθε εργασίας . Τα προβλήματα πάνω στα οποία έχει χρησιμοποιηθεί ο αλγόριθμος είναι τα προβλήματα του Taillard και στους πίνακες αποτελεσμάτων αναγράφονται τα αποτελέσματα του αλγορίθμου αναζήτησης της μουσικής αρμονίας καθώς και τα αποτελέσματα Taillard πάνω στα ίδια ακριβώς προβλήματα flowshop. Εξετάζουμε 10 διαφορετικά προβλήματα για 12 διαφορετικές κατηγορίες προβλημάτων . Οι κατηγορίες των εργασιών- μηχανημάτων είναι 20x5, 20x10, 20x20, 50x5, 50x10, 50x20, 100x5, 100x10,100x20, 200x10, 200x20, 500x20. Όσο πιο κοντά στο ανώτερο όριο του Taillard είναι η λύση που βρίσκει ο αλγόριθμος τόσο καλύτερη είναι η απόδοση του. Αυτό αποτελεί το βασικό κριτήριο για την αξιολόγηση της αποτελεσματικότητας του.

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
20x5	Tai01	1278	1297	9-15-1-3-6-16-2-8-17-14-5-13-10-4-11-7-19-18-12-20
	Tai02	1359	1360	6-19-10-7-18-15-12-3-17-20-13-16-14-9-11-1-5-4-8-2
	Tai03	1081	1114	3-16-19-14-20-12-10-18-11-5-1-7-9-2-4-8-15-17-6-13
	Tai04	1293	1356	13-11-7-17-18-19-16-20-9-2-5-1-12-3-15-6-10-4-8-14
	Tai05	1236	1266	5-3-4-12-9-10-17-16-13-19-2-15-14-11-8-6-7-18-1-20
	Tai06	1195	1272	14-16-13-11-6-17-8-12-18-1-7-2-10-5-4-9-3-20-15-19
	Tai07	1239	1259	5-15-20-11-2-13-14-4-3-1-16-6-17-9-7-10-8-12-19-18
	Tai08	1206	1234	17-2-6-12-16-9-10-1-14-19-18-15-8-3-5-4-20-7-13-11
	Tai09	1230	1258	4-7-20-1-8-18-17-10-12-6-11-3-5-16-13-2-9-19-15-14
	Tai10	1108	1127	5-11-16-7-6-14-8-10-13-18-12-19-20-17-1-2-3-4-15-9
20x10	Tai01	1582	1622	18-5-4-2-15-12-9-10-17-14-8-20-11-19-6-3-13-7-1-16
	Tai02	1659	1683	19-15-12-13-17-9-2-7-14-11-20-10-1-5-16-3-8-4-6-18
	Tai03	1496	1537	4-7-13-5-1-10-9-2-6-11-16-17-14-15-20-12-18-19-3-8
	Tai04	1378	1442	3-12-9-7-6-10-4-1-2-11-15-13-16-19-20-18-8-14-5-17

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
20x10	Tai05	1419	1478	16-8-4-18-17-13-1-3-2-14-15-20-12-11-19-5-6-9-10-7
	Tai06	1397	1444	19-6-4-3-19-7-8-20-11-13-10-9-16-17-14-12-5-2-15-1
	Tai07	1484	1696	10-17-4-7-9-18-19-20-1-3-15-8-5-11-13-6-12-16-14-2
	Tai08	1538	1645	12-20-14-17-8-2-1-4-13-15-18-16-19-7-5-3-10-11-6-9
	Tai09	1593	1692	5-12-13-4-19-7-17-6-8-1-2-10-18-3-15-20-14-11-16-9
	Tai10	1591	1688	5-2-13-8-20-4-19-15-17—6-16-3-1-18-7-10-9-12-11-14
20x20	Tai01	2297	2362	8-18-15-9-16-14-7-11-5-10-2-12-13-1-17-20-6-4-3-19
	Tai02	2100	2162	3-11-6-4-10-20-19-12-13-5-16-14-15-8-18-1-7-17-9-2
	Tai03	2326	2411	5-19-2-6-14-16-20-1-13-11-15-3-17-9-4-8-7-12-18-10
	Tai04	2223	2272	14-3-4-8-13-16-5-6-12-11-20-2-18-7-15-1-17-19-10-9
	Tai05	2291	2348	10-15-2-3-13-18-9-1-19-20-16-17-12-14-4-7-5-11-6-8
	Tai06	2226	2319	6-13-18-16-15-14-2-8-12-10-9-5-20-11-4-17-3-1-7-19
	Tai07	2273	2374	10-12-5-18-16-6-2-11-14-15-1-4-3-13-20-9-7-8-19-17
	Tai08	2200	2276	4-16-10-7-2-17-5-1-8-11-9-20-6-12-13-3-19-14-18-15

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
20x20	Tai09	2237	2352	1-16-17-4-6-14-7-3-2-20-10-13-11-18-15-8-5-9-12-19
	Tai10	2178	2243	3-7-15-17-19-12-14-6-8-2-10-4-13-20-5-11-18-1-16-9
50x5	Tai01	2724	2766	31-40-41-3-10-15-39-27-11-32-26-13-24-28-1-21-6-44-47-25-29-17-12-50-2-34-5-36-4-45-18-38-7-22-43-33-37-30-46-35-9-8-42-19-14-20-48-16-49-23
	Tai02	2834	2896	42-5-7-37-22-10-32-46-19-26-17-15-33-12-6-23-2-25-49-50-39-24-45-47-16-41-27-29-38-40-48-35-13-44-28-20-9-8-4-30-31-21-14-36-3-43-1-18-11-34
	Tai03	2621	2678	4-2-45-23-3-8-1-6-34-37-24-31-25-15-38-28-22-18-27-29-40-43-49-48-47-11-16-36-9-19-5-41-32-35-33-42-26-20-39-12-10-44-46-17-7-13-14-21-50-30
	Tai04	2751	2844	2-14-5-3-20-24-19-50-18-12-9-13-7-44-36-25-32-26-49-4-47-10-8-28-48-11-22-30-42-33-6-39-23-17-46-27-45-16-29-31-21-15-34-40-35-41-43-38-37-1
	Tai05	2863	2908	48-1-8-43-31-3-50-28-5-46-7-20-12-36-19-27-22-37-17-4-40-21-2-39-24-34-30-14-35-41-15-23-32-44-45-47-9-10-42-33-38-29-11-49-25-26-6-13-16-18
	Tai06	2829	2867	41-30-47-29-46-9-35-10-5-1-39-28-31-8-44-21-19-13-42-38-33-24-4-40-50-11-49-16-25-22-15-2-23-20-18-48-7-6-17-32-14-36-45-12-26-34-3-37-43-27

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
50x5	Tai07	2725	2776	37-19-49-34-45-39-27-43-47-30-18-20-33-21-8-9-15-31-22-12-48-3-6-25-38-5-40-24-50-16-41-46-35-11-26-32-17-36-1-2-4-10-7-29-42-44-13-23-14-28
	Tai08	2683	2717	47-8-2-34-24-36-19-1-49-30-35-5-9-26-7-14-10-17-46-31-18-44-3-25-43-20-28-27-39-37-13-40-50-42-29-45-41-33-23-21-48-22-15-6-16-12-4-38-11-32
	Tai09	2552	2569	46-44-3-7-47-13-36-10-1-48-49-15-2-5-23-28-45-39-50-27-40-43-24-9-33-38-25-11-34-21-22-42-29-16-17-41-32-26-19-6-20-35-18-8-37-30-12-31-14-4
	Tai10	2782	2789	6-2-11-1-16-28-24-21-27-35-33-19-17-32-10-36-48-8-5-42-46-47-49-15-30-3-40-14-43-41-50-37-26-38-31-7-34-39-20-44-18-25-4-9-23-29-45-13-22-12
50x10	Tai01	3025	3192	20-22-6-44-26-25-4-34-33-49-40-15-18-46-36-14-31-3-47-43-42-30-9-38-37-16-29-23-35-32-39-21-17-10-13-41-19-27-12-28-11-5-2-7-24-48-45-50-8-1
	Tai02	2892	3083	49-35-3-38-22-12-47-20-28-50-17-23-33-42-32-39-34-37-21-6-18-15-31-43-16-24-9-40-13-14-27-25-30-10-8-19-4-46-36-41-29-48-26-2-45-11-5-1-44-7

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
50x10	Tai03	2864	3026	10-28-46-4-45-25-3-2-24-31-11-38-33-47-39-26-1-29-32-19-9-50-41-42-13-27-7-18-40-30-49-16-22-35-36-14-5-8-43-44-37-34-20-15-17-21-23-48-6-12
	Tai04	3064	3152	5-31-20-36-19-22-14-1-2-25-11-34-17-39-33-41-13-40-3-8-32-21-6-48-23-24-26-44-46-37-43-38-35-49-9-10-16-4-28-47-18-15-29-12-7-45-50-27-42-30
	Tai05	2986	3193	6-14-1-16-5-4-7-10-35-42-48-3-40-11-2-8-31-24-25-45-12-49-15-47-43-17-41-38-29-50-46-44-37-36-39-33-21-30-28-13-18-34-22-27-19-26-20-9-32-23
	Tai06	3006	3173	16-46-40-1-3-29-28-25-45-30-42-27-5-11-9-10-39-49-33-13-14-23-47-26-48-17-22-15-38-2-41-24-32-20-34-36-35-19-12-43-50-18-31-6-7-8-37-21-4-44
	Tai07	3107	3271	32-9-1-17-2-40-42-7-49-4-6-37-13-10-16-26-28-20-12-35-15-19-45-14-21-41-46-50-39-8-43-30-23-3-24-44-48-38-22-11-27-47-5-31-36-18-29-33-34-25
	Tai08	3039	3137	6-18-9-26-11-47-1-17-34-7-5-21-12-13-15-42-29-36-35-20-8-19-22-31-37-33-2-49-45-28-44-41-46-16-32-50-48-25-27-4-10-38-39-3-14-30-43-24-23-40

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
50x10	Tai09	2902	3079	42-27-44-45-48-4-23-22-21-16-41-13-18-46-11-3-34-50-39-37-33-5-25-2-36-30-40-31-17-38-35-29-20-24-10-8-47-15-9-19-6-28-14-49-1-32-43-12-26-7
	Tai10	3091	3241	49-11-22-6-1-42-28-26-20-5-9-34-4-27-31-7-3-48-24-44-36-40-8-19-47-35-50-12-17-32-18-39-25-45-43-14-16-29-15-10-23-41-46-33-21-13-38-37-30-2
50x20	Tai01	3875	4100	17-20-39-35-31-37-34-5-3-11-29-30-47-49-45-8-28-42-23-1-44-27-43-6-15-24-13-2-32-16-33-36-10-19-46-26-4-9-38-7-14-41-18-12-22-40-50-25-21-48
	Tai02	3715	3962	33-37-32-20-41-43-49-21-35-5-1-4-36-16-39-28-45-3-11-23-46-42-47-50-38-31-2-18-40-14-34-15-7-6-29-25-12-9-17-10-12-27-44-30-48-24-22-8-19-26
	Tai03	3668	3881	24-4-10-22-5-28-19-46-39-37-8-21-20-16-25-15-12-2-27-3-32-9-18-30-31-11-43-48-26-1-50-45-14-29-33-23-6-47-49-35-34-44-41-7-42-40-13-38-17-36
	Tai04	3752	3968	20-14-3-13-38-31-49-37-7-4-24-12-11-30-10-33-39-44-46-48-50-22-43-32-8-40-35-23-26-45-6-1-15-2-17-36-5-19-21-9-41-47-29-34-18-27-16-28-42-25

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας	
			Makespan	Σειρά εργασιών
50x20	Tai05	3635	3899	48-40-31-32-2-41-34-35-23-4-3-19-21-43-9-18-42-33-27-25-45-49-8-16-15-28-50-26-1-22-20-47-17-30-36-38-5-44-10-46-13-14-6-24-12-37-39-29-7-11
	Tai06	3698	3943	37-11-5-43-42-50-6-14-28-45-20-32-3-21-4-16-9-47-49-33-8-15-29-35-36-22-34-26-1-40-46-18-24-13-10-31-17-48-41-30-38-25-39-27-23-2-7-19-44-12
	Tai07	3716	3945	4-23-12-2-17-10-47-21-11-1-19-45-20-13-14-49-15-28-3-43-38-22-29-32-24-5-50-31-39-34-36-48-40-18-7-27-6-35-9-8-33-41-37-42-46-26-25-30-44-16
	Tai08	3709	3910	20-30-29-4-18-26-48-32-24-10-12-22-35-7-3-15-27-1-5-16-36-21-8-2-19-6-41-28-37-38-33-47-31-17-11-40-50-49-45-9-13-34-42-43-25-14-23-39-46-44
	Tai09	3765	4007	37-3-14-26-8-28-30-32-12-31-20-7-50-34-48-27-38-23-19-9-4-17-16-11-13-1-22-42-47-24-25-2-46-41-49-39-40-10-29-43-15-36-5-21-6-33-18-35-45-44
	Tai10	3777	3903	33-12-32-19-1-46-23-31-3-2-36-10-8-37-50-29-49-42-43-25-22-7-44-40-6-38-14-47-9-17-30-39-5-11-16-26-41-27-20-35-28-24-18-21-15-48-45-13-4-34

Από εδώ και πέρα σταματαω να γράφω αναλυτικά την σειρά εργασιών που βρίσκω, λόγο του μεγάλου αριθμού των εργασιών, και γράφω μόνο το βέλτιστο χρόνο.

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας
			Makespan
100x5	Tai01	5493	5617
	Tai02	5268	5290
	Tai03	5175	5303
	Tai04	5014	5044
	Tai05	5250	5267
	Tai06	5135	5161
	Tai07	5246	5297
	Tai08	5106	5137
	Tai09	5454	5485
	Tai10	5328	5389
100x10	Tai01	5770	5986
	Tai02	5349	5486
	Tai03	5677	5911
	Tai04	5791	5995
	Tai05	5468	5627
	Tai06	5303	5533
	Tai07	5599	5787
	Tai08	5623	5903

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας
			Makespan
100x10	Tai09	5875	5979
	Tai10	5845	5943
100x20	Tai01	6286	6699
	Tai02	6241	6668
	Tai03	6329	6636
	Tai04	6306	6703
	Tai05	6377	6784
	Tai06	6437	6689
	Tai07	6346	6662
	Tai08	6481	6826
	Tai09	6358	6678
	Tai10	6465	6814
200x10	Tai01	10868	11204
	Tai02	10494	10882
	Tai03	10922	11176
	Tai04	10889	11021
	Tai05	10524	10924
	Tai06	10331	10698
	Tai07	10857	11142
	Tai08	10731	11214
	Tai09	10438	10757

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Αποτελέσματα εφαρμογής αλγορίθμου μουσικής αρμονίας
			Makespan
200x10	Tai10	10676	10967
200x20	Tai01	11294	12019
	Tai02	11420	11939
	Tai03	11446	12196
	Tai04	11347	12255
	Tai05	11311	11875
	Tai06	11282	11860
	Tai07	11456	12270
	Tai08	11415	12128
	Tai09	11343	12009
	Tai10	11422	12147
500x20	Tai01	26189	27647
	Tai02	26629	28352
	Tai03	26458	27921
	Tai04	26549	27933
	Tai05	26404	27940
	Tai06	26581	27908
	Tai07	26461	27641
	Tai08	26615	28137
	Tai09	26083	27484
	Tai10	26527	28007

Αφού έχουμε βρει τα αποτελέσματα , βρίσκουμε τις αποκλίσεις που εχουνε σε σχέση με αυτά του Taillard χρησιμοποιώντας τον τύπο

Απόκλιση=(Makespan-Ανώτερο Όριο Taillard) / Ανώτερο Όριο Taillard

ΠΙΝΑΚΑΣ ΑΠΟΚΛΙΣΕΩΝ

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
20x5	Tai01	1278	1297	0,01486
	Tai02	1359	1360	0,00073
	Tai03	1081	1114	0,03052
	Tai04	1293	1356	0,04872
	Tai05	1236	1266	0,02427
	Tai06	1195	1272	0,06443
	Tai07	1239	1259	0,01614
	Tai08	1206	1234	0,02321
	Tai09	1230	1258	0,02276
	Tai10	1108	1127	0,01714
20x10	Tai01	1582	1622	0,02528
	Tai02	1659	1683	0,01446
	Tai03	1496	1537	0,02740
	Tai04	1378	1442	0,04644
	Tai05	1419	1478	0,04157
	Tai06	1397	1444	0,03364
	Tai07	1484	1696	0,14285
	Tai08	1538	1643	0,06827
	Tai09	1593	1692	0,06214
	Tai10	1591	1688	0,06096

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
20x20	Tai01	2297	2362	0,02829
	Tai02	2100	2162	0,02952
	Tai03	2326	2411	0,03654
	Tai04	2223	2272	0,02204
	Tai05	2291	2348	0,02487
	Tai06	2226	2319	0,04177
	Tai07	2273	2374	0,04443
	Tai08	2200	2276	0,03454
	Tai09	2237	2352	0,05140
	Tai10	2178	2243	0,02984
50x5	Tai01	2724	2766	0,01541
	Tai02	2834	2896	0,02187
	Tai03	2621	2678	0,02174
	Tai04	2751	2844	0,03380
	Tai05	2863	2908	0,01571
	Tai06	2829	2867	0,01343
	Tai07	2725	2776	0,01871
	Tai08	2683	2717	0,01267
	Tai09	2552	2569	0,00666
	Tai10	2782	2789	0,00251

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
50x10	Tai01	3025	3192	0,05515
	Tai02	2892	3083	0,06604
	Tai03	2864	3026	0,05656
	Tai04	3064	3152	0,02872
	Tai05	2986	3193	0,06932
	Tai06	3006	3173	0,05555
	Tai07	3107	3271	0,05278
	Tai08	3039	3137	0,03224
	Tai09	2902	3079	0,06099
	Tai10	3091	3241	0,04852
50x20	Tai01	3875	4100	0,05806
	Tai02	3715	3962	0,06648
	Tai03	3668	3881	0,05806
	Tai04	3752	3968	0,05756
	Tai05	3635	3899	0,07262
	Tai06	3698	3943	0,06625
	Tai07	3716	3945	0,06162
	Tai08	3709	3910	0,05419
	Tai09	3765	4007	0,06427
	Tai10	3777	3903	0,03335

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
100x5	Tai01	5493	5617	0,02257
	Tai02	5268	5290	0,00417
	Tai03	5175	5303	0,02473
	Tai04	5014	5044	0,00598
	Tai05	5250	5267	0,00323
	Tai06	5135	5161	0,00506
	Tai07	5246	5297	0,00972
	Tai08	5106	5137	0,00607
	Tai09	5454	5485	0,00568
	Tai10	5328	5389	0,01144
100x10	Tai01	5770	5986	0,03743
	Tai02	5349	5486	0,02561
	Tai03	5677	5911	0,04121
	Tai04	5791	5995	0,03522
	Tai05	5468	5627	0,02907
	Tai06	5303	5533	0,04337
	Tai07	5599	5787	0,03357
	Tai08	5623	5903	0,04979
	Tai09	5875	5979	0,01770
	Tai10	5845	5943	0,01676

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
100x20	Tai01	6286	6699	0,06570
	Tai02	6241	6668	0,06841
	Tai03	6329	6636	0,04850
	Tai04	6306	6703	0,06295
	Tai05	6377	6784	0,06382
	Tai06	6437	6689	0,03914
	Tai07	6346	6662	0,04979
	Tai08	6481	6826	0,05323
	Tai09	6358	6678	0,05033
	Tai10	6465	6814	0,05398
200x10	Tai01	10868	11204	0,03091
	Tai02	10494	10882	0,03697
	Tai03	10922	11176	0,02325
	Tai04	10889	11021	0,01212
	Tai05	10524	10924	0,03800
	Tai06	10331	10698	0,03552
	Tai07	10857	11142	0,02625
	Tai08	10731	11214	0,04500
	Tai09	10438	10757	0,03056
	Tai10	10676	10967	0,02725

Είδος Προβλήματος	Πρόβλημα	Ανώτερο Όριο Taillard	Makespan	Απόκλιση Αποτελέσματος
200x20	Tai01	11294	12019	0,06419
	Tai02	11420	11939	0,04544
	Tai03	11446	12196	0,06552
	Tai04	11347	12255	0,08002
	Tai05	11311	11875	0,04986
	Tai06	11282	11860	0,05123
	Tai07	11456	12270	0,07105
	Tai08	11415	12128	0,06246
	Tai09	11343	12009	0,05871
	Tai10	11422	12147	0,06347
500x20	Tai01	26189	27647	0,05567
	Tai02	26629	28352	0,06470
	Tai03	26458	27921	0,05529
	Tai04	26549	27933	0,05213
	Tai05	26404	27940	0,05817
	Tai06	26581	27908	0,04992
	Tai07	26461	27641	0,04459
	Tai08	26615	28137	0,05718
	Tai09	26083	27484	0,05371
	Tai10	26527	28007	0,05579

ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΠΑΡΑΜΕΤΡΩΝ

Ο προσδιορισμός των παραμέτρων γίνεται με τυχαίο πειραματικό τρόπο έχοντας πάντα ως γνώμονα το εύρος των τιμών που πρέπει να κινηθούμε και να εξετάσουμε. Στην συγκεκριμενη εργασια ο αριθμος των επαναλήψεων είναι 4000. Δίνοντας τιμές στο HMCR και στο μέγεθος του πίνακα λύσεων παρατηρώ ότι οι διαφορές στο τελικό αποτέλεσμα είναι πολύ μικρές και βασίζονται κυρίως στην τύχη και όχι τοσο στην μεταβολή των παραμέτρων, δηλαδή για συγκεκριμένες τιμές HMCR και MEGETHOS τρέχοντας 2 φορές τον αλγόριθμο μπορεί την μια φορά να μας δώσει καλύτερα αποτελέσματα σε σχέση με άλλες συγκεκριμένες τιμές για HMCR και MEGETHOS και αν το ξανατρέξουμε να μας δινει μετα χειρότερες τιμές . Αυτό είναι λογικό αν σκεφτεί κανείς ότι ο συγκεκριμένος αλγόριθμος βασίζεται καθαρά σε τυχαίες επιλογές λύσεων.

Εξετάζοντας τα ενδεχόμενα για HMCR =0,8και HMCR=0,9 καθώς και MEGETHOS=10 η MEGETHOS=15 έχω τα εξής αποτελέσματα.

Για τον Πινακα 1. 5x20

Για MEGETHOS=10 και HMCR=0,8→ Βέλτιστος χρόνος =1297

Για MEGETHOS=15 και HMCR=0,8→ Βέλτιστος χρόνος =1323

Για MEGETHOS=10 και HMCR=0,9→ Βέλτιστος χρόνος =1297

Για MEGETHOS=15 και HMCR=0,9→ Βέλτιστος χρόνος =1297

Για τον Πίνακα 2. 5x20

Για MEGETHOS=10 και HMCR=0,8→ Βέλτιστος χρόνος =1367

Για MEGETHOS=15 και HMCR=0,8→ Βέλτιστος χρόνος =1367

Για MEGETHOS=10 και HMCR=0,9→ Βέλτιστος χρόνος =1366

Για MEGETHOS=15 και HMCR=0,9→ Βέλτιστος χρόνος =1366

Για τον Πίνακα 11. 10x20

Για MEGETHOS=10 και HMCR=0,8→ Βέλτιστος χρόνος =1636

Για MEGETHOS=15 και HMCR=0,8→ Βέλτιστος χρόνος =1641

Για MEGETHOS=10 και HMCR=0,9→ Βέλτιστος χρόνος =1674

Για MEGETHOS=15 και HMCR=0,9→ Βέλτιστος χρόνος =1622

Παρατηρώντας τα αποτελέσματα βλέπω πως το MEGETHOS=15 δίνει λίγο καλύτερα αποτελέσματα ειδικά σε μεγαλύτερους πίνακες . Το ίδιο ισχύει και για HMCR=0,9 και για αυτό τον λόγο επέλεξα αυτές τις τιμες .

Η παράμετρος PAR=0,3 έχει επιλεχτεί βασιζομένη σε βιβλιογραφικά δεδομένα λόγο του ότι το 0,3 θεωρείται καλή τιμή για την συγκεκριμένη παράμετρο.

ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Ξεκινώντας με τα προβλήματα 5 μηχανών και 20 εργασιών παρατηρούμε ότι τα αποτελέσματα είναι παρά πολύ κοντά στα βέλτιστα .

Οι αποκλίσεις είναι πολύ μικρές με μέγιστη απόκλιση 6,4% και ελάχιστη απόκλιση 0,07 %. Η μέση απόκλιση των αποτελεσμάτων είναι περίπου 2,6%.

Στα προβλήματα 10 μηχανών και 20 εργασιών αυξάνεται λίγο η απόκλιση σε σχέση με τα όρια Taillard. Έχουμε ως μέγιστη απόκλιση 14% η όποια είναι και η μεγαλύτερη απόκλιση όλων των μετρήσεων και ίσως να οφείλεται σε πολύ άστοχη πρώτη προσέγγιση λύσεων και ελάχιστη απόκλιση 1,4%. Η μέση απόκλιση της κατηγορίας είναι περίπου 3,9%.

Συνεχίζοντας στην κατηγορία προβλημάτων 20 μηχανών ,20 εργασιών

παρατηρούμε μικρές αποκλίσεις, μικρότερες από την κατηγορία 10x20 με μέγιστη απόκλιση 5,1% και ελάχιστη απόκλιση 2,2%. Η μέση απόκλιση είναι 3,4% , καλύτερη δηλαδή σε σχέση με την προηγούμενη που ήταν 3,9%.

Στην κατηγορία 50x5 οι αποκλίσεις είναι μικρές , με μέγιστη απόκλιση 3,3%, μια πολύ καλή τιμή , και ελάχιστη απόκλιση 0,02%. Η μέση τιμή των αποκλίσεων είναι 1,6% δηλαδή τα αποτελέσματα μας σχεδόν ταυτίζονται με αυτά του Taillard.

Αυξάνοντας τον αριθμό των μηχανημάτων από 5 σε 10 με τις εργασίες να παραμένουν ίδιες (50) παρατηρούμε μια σημαντική αύξηση στις αποκλίσεις . Η μέγιστη απόκλιση αγγίζει το 7% και η ελάχιστη είναι περίπου 2,8%. Σε αυτήν την κατηγορία προβλημάτων έχουμε μέση απόκλιση 5,2% εμφανώς αυξημένη σε σχέση με την προηγούμενη.

Με την περεταίρω αύξηση του αριθμού των μηχανημάτων σε 20 ερχόμαστε στην κατηγορία προβλημάτων 50x20 . Οι αποκλίσεις σε αυτήν την κατηγορία δεν αυξάνονται ιδιαίτερα . Η μέγιστη απόκλιση είναι 7,2% και η ελάχιστη 3,3%. Τέλος η μέση απόκλιση της κατηγορίας είναι 5,9% ελαφρός μεγαλύτερα σε σχέση με πριν.

Στην κατηγορία προβλημάτων 100 εργασιών και 5 μηχανημάτων οι αποκλίσεις είναι πολύ μικρές και τα αποτελέσματα πολύ ικανοποιητικά. Η μέγιστη απόκλιση είναι 3,7% και η ελάχιστη 0,03%. Μέση απόκλιση έχουμε 1% δηλαδή καλύτερη ακόμη και από την κατηγορία των 20 εργασιών , 5 μηχανημάτων.

Προχωρώντας στα προβλήματα 100 εργασιών και 10 μηχανημάτων παρατηρούμε από τη μια αύξηση στις αποκλίσεις αλλά από την άλλη βλέπουμε ότι τα αποτελέσματα μας εξακολουθούν να προσεγγίζουν καλά τα βέλτιστα και οι αποκλίσεις να κυμαίνονται σε χαμηλά επίπεδα. Μέγιστη απόκλιση έχουμε σχεδόν 5% και ελάχιστη 1,6%. Η μέση απόκλιση είναι περίπου 3,2%.

Στην κατηγορία προβλημάτων 100 εργασιών , 20 μηχανημάτων έχουμε μια εμφανή αύξηση των αποκλίσεων με μέγιστη απόκλιση 6,8% και ελάχιστη 3,9%. Παρατηρούμε ότι οι αποκλίσεις μας έχουν μικρή διακύμανση στις τιμές τους όσο αυξάνονται τα μεγέθη των προβλημάτων. Η μέση απόκλιση είναι 5,5%.

Στα προβλήματα με 200 εργασίες σε 10 μηχανήματα παρατηρούμε πάλι μικρότερες αποκλίσεις σε σχέση με τα προβλήματα 100x20 αφού ο αριθμός των μηχανημάτων επηρεάζει τις λύσεις μας περισσότερο από τον αριθμό των εργασιών. Η μέγιστη απόκλιση είναι 4,5% και η ελάχιστη 1,2%. Η μέση απόκλιση είναι 3% εμφανώς βελτιωμένη σε σχέση με την κατηγορία προβλημάτων 100 εργασιών , 20 μηχανημάτων.

Αυξάνοντας τον αριθμό των εργασιών σε 20 με τις εργασίες να παραμένουν στον ίδιο αριθμό παρατηρούμε αύξηση στις αποκλίσεις. Η μέγιστη απόκλιση πλέον είναι 7,1 % και η ελάχιστη 4,5%. Η μέση απόκλιση είναι 6,1 %.

Η τελευταία κατηγορία προβλημάτων είναι αυτή των 500^{ων} εργασιών και 20 μηχανημάτων. Οι αποκλίσεις βλέπουμε ότι κυμαίνονται στα ίδια επίπεδα με αυτές της κατηγορίας προβλημάτων 200 εργασιών και 20 μηχανημάτων . Η ελάχιστη απόκλιση σε αυτήν την κατηγορία είναι 4,4% και η μέγιστη 6,4%. Η μέση απόκλιση είναι 5,4%.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Από την ανάλυση των αποτελεσμάτων προκύπτουν οι εξεις σημαντικές παρατηρήσεις :

- Όσο αυξάνεται ο αριθμός των μηχανημάτων οι λύσεις μας εμφανίζουν μεγαλύτερες αποκλίσεις . Άρα ο αριθμός των μηχανημάτων παίζει καθοριστικό ρολό στα αποτελέσματα που μας δίνει ο αλγόριθμος σε σχέση με τα βέλτιστα κατά Taillard.

- Ο αριθμός των εργασιών δεν επηρεάζει σε μεγάλο βαθμό τις λύσεις μας. Δηλαδή ο αλγόριθμος τείνει να ταυτιστεί με τα βέλτιστα αποτελέσματα σε μεγάλους αριθμούς εργασιών αρκεί να είναι μικρός ο αριθμός των μηχανημάτων.

- Σε μεγάλα προβλήματα με πολλές εργασίες και πολλά μηχανήματα παρατηρώ ότι η αποκλίσεις των αποτελεσμάτων παρουσιάζουν μικρή διακύμανση και εμφανίζουν πολύ κοντινές τιμές ανεξαρτήτως του προβλήματος . Αντιθέτως σε μικρά προβλήματα οι αποκλίσεις διαφέρουν μεταξύ τους κάτι που είναι λογικό αν σκεφτεί κάνεις ότι τα αποτελέσματα βασίζονται σε ένα βαθμό στην τύχη και σε μικρά προβλήματα μικρές διαφορές στο αποτέλεσμα μπορεί να εμφανίζει μεγάλες ποσοστιαίες αποκλίσεις.

- Συμπεραίνουμε ότι ο αλγόριθμος μας είναι αποτελεσματικός σε όλες τις κατηγορίες προβλημάτων αφού σε καμία κατηγορία προβλημάτων η μέση απόκλιση δεν ξεπερνά το 6,1% που είναι και η μέγιστη μέση απόκλιση .

- Ο αλγόριθμος μουσικής αρμονίας μας δίνει πολύ καλά αποτελέσματα μέσα σε μικρό χρονικό διάστημα . Είναι σημαντικό να επισημανθεί ότι ο χρόνος εκτέλεσης του προγράμματος ξεκιναει με διάρκεια λιγότερη των 30 δευτερολέπτων για προβλήματα 5x20 και φτάνει σε μέγιστη διάρκεια περίπου 8-9 λεπτά για τα πιο σύνθετα προβλήματα 20x500

- Η πιο εύστοχη επιλογή παραμέτρων HMCR, PAR και μεγέθους του πίνακα λύσεων θα μπορούσε να επηρεάσει θετικά τα αποτελέσματα μας . Το ίδιο ακριβώς ισχύει και για τον αριθμό των επαναλήψεων.

[1] Ελληνική Εταιρεία Logistics Παράρτημα Θεσσαλονίκης

[2] Μαρινάκης Ι., Μυγδαλάς Α.,« Σχεδιασμός και βελτιστοποίηση της εφοδιαστικής αλυσίδας» Θεσσαλονίκη, Εκδόσεις Σοφία (2008)

[3] Kang Seok Lee « Standard Harmony Search Algorithm for Structural Design Optimization »

[4]« Harmony search based algorithms for bandwidth-delay-constrained least-cost» multicast routing R. Forsati a,*, A.T. Haghighat a, M. Mahdavi

[5] A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice
Kang Seok Lee a,*,1, Zong Woo Geem b

[6] «Χρονοπρογραμματισμός και έλεγχος παραγωγής» Σημειώσεις μαθήματος Διοίκησης Παραγωγής και Συστημάτων Υπηρεσιών ,Σχολή ΗΜΜΥ του ΕΜΠ.

[7] «Χρονικός Προγραμματισμός » Σημειώσεις από το μάθημα Προγραμματισμός και Έλεγχος Παραγωγής, Σχολή Βιομηχανικής Διοίκησης του Πανεπιστημίου Πειραιά.

[8] «Οργάνωση και διοίκηση παραγωγής» Σημειώσεις μαθήματος Διοίκησης Παραγωγής και Συστημάτων Υπηρεσιών ,Σχολή ΗΜΜΥ του ΕΜΠ.

[9]Graham R.L., Lawler E.L., Lenstra J.K., and Rinnoy Kan. A.H.G. «Optimization and approximation in deterministic sequencing and scheduling: a survey.» In Annals of Discrete Mathematics, (1979) 287-326

[10] «Local search methods for the flow-shop scheduling problem with flow time minimization» Quan-Ke Pana , Rubén Ruizb

[11]«Αλγόριθμοι τοπικής αναζήτησης με απαγόρευση κινήσεων και παραλληλοποίηση της αναζήτησης για την επίλυση του προβλήματος παραγωγής συνεχούς ροής» Παπαδάκης Δ. Τμήμα Επιστήμης Υπολογιστών Πανεπιστήμιο Κρήτης (1995)