

**ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΕΛΕΓΧΟΣ ΑΠΟΔΟΧΗΣ ΠΑΡΑΓΓΕΛΙΩΝ ΣΕ ΓΡΑΜΜΕΣ  
ΠΑΡΑΓΩΓΗΣ ΤΥΠΟΥ ΚΑΝΒΑΝ**

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΛΑΓΟΥΔΑΚΗΣ ΜΙΧΑΛΗΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΕΥΣΤΡΑΤΙΟΣ Θ. ΙΩΑΝΝΙΔΗΣ

Χανιά, 2004

## Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον Καθηγητή Βασίλειο Κουϊκόγλου για τις πολύτιμες υποδείξεις του για την ολοκλήρωση και συγγραφή της διπλωματικής μου εργασίας. Ακόμη οφείλω ένα μεγάλο ευχαριστώ στους γονείς μου και τον αδερφό μου για την αμέριστη υποστήριξη τους καθ'όλη την διάρκεια των σπουδών μου. Επίση θέλω να ευχαριστήσω τους φίλους μου οι οποίοι με ενθάρρυναν να συνεχίζω την προσπάθεια μου κάθε φορά που παρουσιάζονταν κάποια δυσκολία. Πάνω απ' όλα όμως, είμαι ευγνώμων στον επιβλέποντα της εργασίας, Δρ. Ευστράτιο Ιωαννίδη τόσο για την πολύτιμη βοήθεια και καθοδήγηση κατά τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας, όσο και για την ηθική υποστήριξη που μου πρόσφερε καθόλη τη διάρκεια εκπόνησης αυτής.

## ΠΕΡΙΕΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ .....	1
1.1	Αντικείμενο της διπλωματικής εργασίας.....	1
1.2	Βιβλιογραφική ανασκόπηση.....	3
1.3	Δομή της διπλωματικής, μεθοδολογία, αποτελέσματα.....	5
2	ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ.....	7
2.1	Εισαγωγή.....	7
2.2	Περιγραφή του συστήματος παραγωγής .....	7
2.3	Περιγραφή Αλγορίθμου Προσομοίωσης .....	10
2.4	Αλγόριθμος Βελτιστοποίησης .....	16
3	ΑΡΙΘΜΗΤΙΚΟ ΠΑΡΑΔΕΙΓΜΑ .....	21
3.1	Εισαγωγή.....	21
3.2	Αριθμητικά αποτελέσματα.....	21
4	ΣΥΜΠΕΡΑΣΜΑΤΑ.....	30
	ΒΙΒΛΙΟΓΡΑΦΙΑ.....	32
	ΠΑΡΑΡΤΗΜΑ (ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ).....	33

# ΚΕΦΑΛΑΙΟ 1

## Εισαγωγή

### 1.1 Αντικείμενο της διπλωματικής εργασίας

Στις μέρες μας κάθε επιχείρηση προσπαθεί να αναπτύσσεται και να εκσυγχρονίζεται μέρα με τη μέρα ολοένα και περισσότερο για να είναι ικανή να επιβιώσει στο σύγχρονο ανταγωνιστικό περιβάλλον. Καθοριστικός παράγοντας για την ανταγωνιστικότητα και τη βιωσιμότητα κάθε βιομηχανίας αποτελεί ο αποτελεσματικός έλεγχος των συστημάτων παραγωγής. Τα συστήματα παραγωγής απαρτίζονται από μηχανές και ενδιάμεσες αποθήκες όπου γίνονται οι απαιτούμενες διαδικασίες επεξεργασίας των πρώτων υλών όπως μορφοποίηση, συναρμολόγηση, επιθεώρηση και έλεγχος ποιότητας τους με σκοπό την παραγωγή έτοιμων προϊόντων, για την παράδοση σε πελάτες. Κάθε μηχανή του συστήματος λειτουργεί ανάλογα με το χρονικό προγραμματισμό που έχει γίνει και ο οποίος αποτελεί μέρος του ελέγχου συστημάτων παραγωγής. Με άλλα λόγια πότε ξεκινά η παραγωγή, με τι ρυθμό πραγματοποιείται και τέλος πότε διακόπτεται. Επίσης στον έλεγχο αυτό περιλαμβάνεται και ο έλεγχος των πωλήσεων δηλαδή το πότε γίνεται δεκτή μία αφικνούμενη παραγγελία, πότε απορρίπτεται ή ανατίθεται σε υπεργολάβους. Αυτός ο έλεγχος της παραγωγής μπορεί να θεωρηθεί σαν ένα πρόβλημα βελτιστοποίησης κάποιου μέτρου απόδοσης όπως για παράδειγμα το καθαρό κέρδος, το κόστος λειτουργίας, το μέσο απόθεμα, το επίπεδο εξυπηρέτησης πελατών κ.α.

Ένας τρόπος να αντιμετωπιστεί αυτό το πρόβλημα ελέγχου των συστημάτων παραγωγής είναι να το μοντελοποιήσουμε σαν ένα πρόβλημα βελτιστοποίησης και κατόπιν να προσπαθήσουμε να βρούμε τη βέλτιστη πολιτική. Βέβαια αυτή η προσέγγιση ενδείκνυται μόνο για πολύ απλά προβλήματα. Παρόλα αυτά οι πληροφορίες που αποκομίζουμε από τέτοιου

είδους απλές εφαρμογές συμβάλλουν στην κατανόηση και την επιτυχή λειτουργία πολύπλοκων συστημάτων.

Μία πιο πρακτική προσέγγιση για την αντιμετώπιση του προβλήματος ελέγχου συστημάτων παραγωγής είναι να περιορίσουμε την αναζήτηση για μια καλή πολιτική ελέγχου, σε μία απλή υποβέλτιστη πολιτική έτσι ώστε να μειώσουμε τον αριθμό των παραμέτρων και άρα την δυσκολία επίλυσης του προβλήματος. Ένα αρκετά μεγάλο μέρος της ερευνητικής προσπάθειας σε αυτόν τον τομέα έχει επικεντρωθεί στην ανάπτυξη και αξιολόγηση απλών πολιτικών ελέγχου παραγωγής, οι οποίες εξαρτώνται από μικρό αριθμό παραμέτρων και έχουν συχνά προκύψει, από πραγματική βιομηχανική εμπειρία. Σε πολλές από αυτές τις πολιτικές, η παραγωγή εξαρτάται άμεσα και ενεργοποιείται από τη ζήτηση για έτοιμα προϊόντα. Τέτοιες πολιτικές συχνά αναφέρονται σαν πολιτικές *έλξης* (*pull production control policies* π.χ. πολιτική ελέγχου KANBAN).

Στην παρούσα διπλωματική εξετάζουμε το πρόβλημα ελέγχου αποδοχής παραγγελιών σε γραμμές παραγωγής όπου γίνεται έλεγχος των αποθεμάτων με την πολιτική KANBAN. Το KANBAN είναι μία τεχνική ελέγχου συστημάτων παραγωγής στην οποία η παραγωγή σχετίζεται απόλυτα με τη ζήτηση. Έτσι όταν μία μονάδα τελικού προϊόντος αφαιρείται λόγω πώλησης από την αποθήκη ετοιμών, η τελευταία μηχανή ( η τελευταία βαθμίδα της γραμμής παραγωγής) παίρνει έγκριση να ξεκινήσει την παραγωγή και νέου κομματιού για την αναπλήρωση του προηγούμενου. Αυτή με τη σειρά της δίνει έγκριση στην προηγούμενη της μηχανή να ξεκινήσει την παραγωγή νέου. Αυτή η διαδικασία συνεχίζεται μέχρις ότου φτάσει η έγκριση στην πρώτη μηχανή του συστήματος παραγωγής έτσι ώστε να ξεκινήσει η παραγωγή νέου προϊόντος. Η διαδικασία ελέγχεται σε σχέση με τις στάθμες των αποθηκών κάθε μηχανής του συστήματος οι οποίες δεν έχουν άπειρη χωρητικότητα αλλά περιορίζονται από συγκεκριμένα κατώφλια

Στην διπλωματική αυτή για τον έλεγχο αποδοχής παραγγελιών προτείνουμε την πολιτική μερικής αποδοχής παραγγελιών (PLS, partly lost sales). Στην πολιτική αυτή σε περιόδους που δεν υπάρχει απόθεμα ετοιμών

προϊόντων δεχόμαστε παραγγελίες μέχρι κάποιο μέγιστο όριο, όταν το πλήθος εκκρεμών παραγγελιών είναι ίσο με το όριο αυτό απορρίπτουμε όλες τις επιπλέον παραγγελίες. Η πολιτική αυτή συγκρίνεται την με την πολιτική *πλήρους αποδοχής* (CB, complete backordering) στην οποία έχουμε αποδοχή όλων των παραγγελιών και την πολιτική *πλήρους απόρριψης* (LS, lost sales) στην οποία έχουμε απόρριψη όλων των παραγγελιών όταν δεν είναι δυνατόν να ικανοποιηθούν άμεσα. Σκοπός του ελέγχου είναι ο καθορισμός των παραμέτρων ελέγχου του συστήματος, που εδώ είναι οι χωρητικότητες των αποθηκών και το κατώφλι των εκκρεμών παραγγελιών έτσι ώστε να βελτιστοποιείται ένα μέτρο απόδοσης του συστήματος. Ως μέτρο απόδοσης χρησιμοποιούμε το αναμενόμενο κέρδος ανά μονάδα χρόνου, το οποίο προκύπτει από τα έσοδα των πωλήσεων μείον το κόστος αποθεμάτων και το κόστος εκκρεμών παραγγελιών.

## 1.2 Βιβλιογραφική ανασκόπηση

Το πρόβλημα του ελέγχου συστημάτων παραγωγής μπορεί να αντιμετωπιστεί μοντελοποιώντας το σαν ένα πρόβλημα βελτιστοποίησης όπως προτείνει ο S.B.Gershwin [1]. Ο τρόπος αυτός όμως δίνει ικανοποιητικά αποτελέσματα μόνο για πολύ απλά συστήματα. Για πολύπλοκα συστήματα ένα πρακτικός τρόπος αντιμετώπισης του προβλήματος του ελέγχου παραγωγής είναι η αναζήτηση της βέλτιστης πολιτικής μέσα σε μία οικογένεια απλών πολιτικών, που υλοποιούνται εύκολα και εξαρτώνται από ένα μικρό αριθμό παραμέτρων. Σε αυτήν την περίπτωση η γνώση των δομικών ιδιοτήτων της βέλτιστης πολιτικής μπορεί να βοηθήσει στην εύρεση τέτοιων απλών πολιτικών.

Υπάρχουν αρκετές κατηγορίες απλών πολιτικών ελέγχου αποθεμάτων σε συστήματα παραγωγής : οι πολιτικές αποθέματος βάσης (base stock), οι πολιτικές KANBAN, CONWIP καθώς και συνδυασμοί των (γενικευμένες πολιτικές KANBAN, οι εκτεταμένες πολιτικές KANBAN, κ.α.). Οι παραπάνω

οικογένειες πολιτικών παρουσιάζονται εκτενέστερα στις εργασίες Liberopoulos and Dallery [2] καθώς και Buzacott and Shanthikumar[3].

Σε προβλήματα συστημάτων παραγωγής και αποθεματικών συστημάτων είναι πολύ σημαντική και η πολιτική αποδοχής παραγγελιών που θα υιοθετηθεί. Συνήθως ακολουθούνται απλές πολιτικές αποδοχής σύμφωνα με τις οποίες, όταν δεν υπάρχει έτοιμο προϊόν, οι παραγγελίες είτε απορρίπτονται όλες είτε γίνονται όλες αποδεκτές (βλ., π.χ., Hadley and Whitin [4], Smith [5], Zipkin [6]). Η αποδοχή όλων των παραγγελιών σε περιόδους που δεν υπάρχει απόθεμα καλείται πολιτική *πλήρους αποδοχής* (CB, complete backordering), ενώ η απόρριψη όλων των παραγγελιών όταν δεν είναι δυνατόν να ικανοποιηθούν άμεσα ονομάζεται πολιτική *πλήρους απόρριψης* (LS, lost sales). Όταν ο ρυθμός παραγωγής της μονάδας παραγωγής είναι μικρότερος της ζήτησης η πολιτική πλήρους αποδοχής καθίσταται ζημιογόνος για μία επιχείρηση αφού το πλήθος των παραγγελιών σε αναμονή θα τείνει να μεγαλώνει χωρίς φραγμό. Μία εναλλακτική πολιτική ελέγχου όταν δεν υπάρχει απόθεμα είναι η αποδοχή ή απόρριψη παραγγελιών με τυχαίο τρόπο (π.χ. πραγματοποιώντας ένα πείραμα Bernoulli) ανεξαρτήτως από το πλήθος των παραγγελιών που ήδη εκκρεμούν (βλ. Moinzadeh [7]). Αυτή η πολιτική ονομάζεται πολιτική τυχαίας αποδοχής (RAC, randomized admission control) και μέχρι τώρα χρησιμοποιείται μόνο σε συστήματα μίας μηχανής.

Οι πολιτικές CB και LS είναι ακριβώς αντίθετες. Με την πλήρη αποδοχή της μη άμεσα ικανοποιημένης ζήτησης το πλήθος των καθυστερημένων παραγγελιών αυξάνεται χωρίς όριο, ενώ με την πολιτική πλήρους απόρριψης το πλήθος των εκκρεμών παραγγελιών είναι μηδέν. Μία ενδιάμεση πολιτική ελέγχου των παραγγελιών είναι η *πολιτική μερικής αποδοχής της μη ικανοποιημένης ζήτησης* (PLS, partly lost sales). Η πολιτική αυτή είναι τύπου κατωφλίου και απορρίπτει όλες τις παραγγελίες όταν το πλήθος των μη ικανοποιημένων γίνει ίσο με ένα κατώφλι που ονομάζεται *έλλειμμα βάσης*. Η συνολική πολιτική του συστήματος καθορίζεται πλήρως από δύο μη αρνητικούς αριθμούς, το απόθεμα βάσης και το έλλειμμα βάσης. Η πολιτική μερικής αποδοχής παραγγελιών είναι πιο γενική από τις CB και LS αφού για

τις ακραίες τιμές  $\infty$  και 0 του ελλείμματος βάσης εκφυλίζεται σε μία από αυτές. Η πολιτική αυτή εξετάζεται για πρώτη φορά στις εργασίες Caldentey [8] και Kouikoglou and Phillis [9]. Σε αυτές τις εργασίες μελετάται η εφαρμογή της πολιτικής PLS σε συστήματα μίας μηχανής. Οι Ioannidis, Kouikoglou and Phillis [10], εξετάζουν το πρόβλημα της εφαρμογής της προτεινόμενης πολιτικής σε δίκτυα παραγωγής ελεύθερης γεωμετρίας, όταν ο έλεγχος των αποθεμάτων γίνεται με μια απλή παραλλαγή της πολιτικής αποθέματος βάσης.

### **1.3 Δομή της διπλωματικής, μεθοδολογία, αποτελέσματα**

Στην παρούσα εργασία εξετάζουμε γραμμές παραγωγής όπου εφαρμόζεται η πολιτική ελέγχου αποθεμάτων KANBAN. Το σύστημα τροφοδοτείται από αρχική αποθήκη πρώτων υλών άπειρης χωρητικότητας. Οι χρόνοι μεταξύ διαδοχικών αφίξεων πελατών είναι τυχαίοι. Οι μηχανές παράγουν σε τυχαίους χρόνους, μπορεί να υποστούν βλάβες με τυχαίο τρόπο και επισκευάζονται σε τυχαίους χρόνους. Οι παραγγελίες των πελατών που δεν μπορούν να ικανοποιηθούν άμεσα καταχωρούνται σε μία λίστα αναμονής και ικανοποιούνται όταν υπάρχουν έτοιμα προϊόντα. Σκοπός της εργασίας είναι η ανάλυση της πολιτικής μερικής αποδοχής σε περιόδους που δεν υπάρχει απόθεμα στην οποία δεχόμαστε παραγγελίες μέχρι κάποιο μέγιστο όριο (πολιτική PLS), η εύρεση του βέλτιστου κέρδους σε συνάρτηση με τις χωρητικότητες των αποθηκών και τη χωρητικότητα της λίστας αναμονής και η σύγκριση των αποτελεσμάτων αυτών σε σχέση με τις ευρέως εφαρμοζόμενες πολιτικές πλήρους αποδοχής (CB) και πλήρους απόρριψης (LS).

Η εκτίμηση των βέλτιστων τιμών των παραμέτρων ελέγχου του συστήματος έγινε με τη χρήση ενός αλγόριθμου μη γραμμικής βελτιστοποίησης. Ο αλγόριθμος αυτός βασίστηκε στο επαναληπτικό σχήμα μέγιστης ανόδου και χρησιμοποιεί ένα προσομοιωτή γραμμών παραγωγής για την εκτίμηση διαφόρων μέτρων απόδοσης του συστήματος.

Η παρούσα εργασία αποτελείται από τρία επιπλέον κεφάλαια:



Στο κεφάλαιο 2 παρουσιάζονται εκτενέστερα τόσο τα υπό μελέτη συστήματα παραγωγής όσο και τα εργαλεία που χρησιμοποιήθηκαν για την ανάλυση και τη σύγκριση των διαφόρων πολιτικών ελέγχου.

Στο κεφάλαιο 3 παρουσιάζονται τα αποτελέσματα από τη σύγκριση των διαφόρων πολιτικών ελέγχου αποδοχής παραγγελιών σε ένα συγκεκριμένο αριθμητικό παράδειγμα. Εξάγονται συμπεράσματα για τα πλεονεκτήματα και μειονεκτήματα της πολιτικής που προτείνουμε σε αντιπαράθεση με τις δύο άλλες πολιτικές.

Τέλος στο κεφάλαιο 4 παρουσιάζονται τα συμπεράσματα και κατευθύνσεις για μελλοντική έρευνα.

## ΚΕΦΑΛΑΙΟ 2

# Περιγραφή Συστήματος Παραγωγής και Εργαλεία που χρησιμοποιήθηκαν

### 2.1 Εισαγωγή

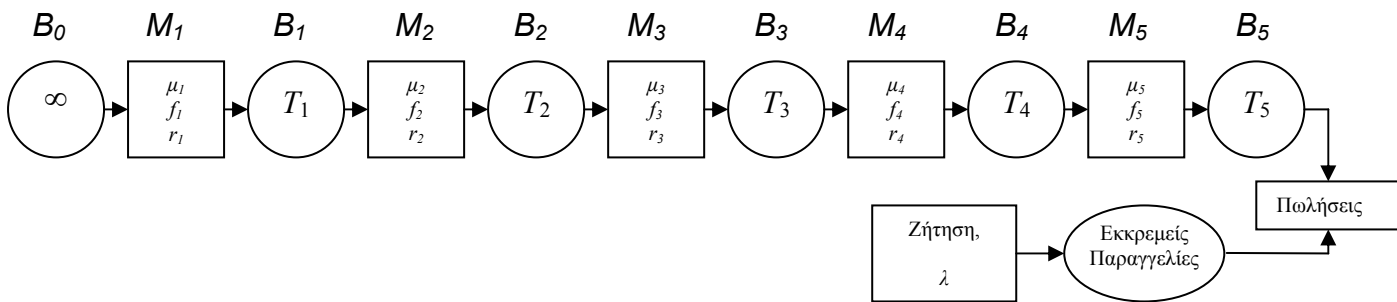
Σε αυτό το κεφάλαιο εξετάζουμε πιο αναλυτικά το πρόβλημα ελέγχου αποδοχής παραγγελιών σε γραμμές παραγωγής όπου γίνεται έλεγχος των αποθεμάτων με την πολιτική KANBAN. Ακόμη παρουσιάζουμε αναλυτικά τα εργαλεία που χρησιμοποιήθηκαν για την ανάλυση και βελτιστοποίηση των υπό μελέτη συστημάτων.

Το κεφάλαιο είναι διαρθρωμένο ως εξής: Στην Παράγραφο 2.2 γίνεται η περιγραφή του συστήματος παραγωγής που εξετάζουμε και αναπτύσσονται οι λόγοι που μας οδήγησαν να χρησιμοποιήσουμε εργαλεία από το χώρο της προσομοίωσης για τη μελέτη της συμπεριφοράς του. Στην Παράγραφο 2.3 περιγράφονται οι βασικές αρχές λειτουργίας του αλγορίθμου προσομοίωσης που χρησιμοποιήθηκε και τέλος στην Παράγραφο 2.4 παρουσιάζεται ο αλγόριθμος μεγιστοποίησης του αναμενόμενου κέρδους.

### 2.2 Περιγραφή του συστήματος παραγωγής

Εξετάζουμε γραμμές παραγωγής, που αποτελούνται από  $N$  μηχανές, ενδιάμεσες αποθήκες πεπερασμένης χωρητικότητας (μία για κάθε μηχανή), μία αποθήκη πρώτων υλών άπειρης χωρητικότητας και τέλος μία λίστα αναμονής εκκρεμών παραγγελιών. Οι πελάτες φθάνουν στο σύστημα σε τυχαίους χρόνους και ο καθένας ζητάει μία μονάδα προϊόντος. Οι χρόνοι μεταξύ διαδοχικών αφίξεων των πελατών είναι ανεξάρτητες τυχαίες μεταβλητές εκθετικά κατανομημένες με μέσο  $1/\lambda$ . Οι χρόνοι παραγωγής κάθε μηχανής  $i$  ( $i = 1, 2, \dots, N$ ) είναι επίσης ανεξάρτητες τυχαίες μεταβλητές εκθετικά κατανομημένες με μέσες τιμές  $1/\mu_i$ .

Οι μηχανές υφίστανται βλάβες που συμβαίνουν και επισκευάζονται σε τυχαίους χρόνους. Οι χρόνοι μεταξύ βλαβών και επισκευών κάθε μηχανής  $i$  είναι επίσης εκθετικά κατανομημένοι, με μέσες τιμές  $1/f_i$  και  $1/r_i$  αντίστοιχα. Μετά την ολοκλήρωση της κατεργασίας ενός προϊόντος από την μηχανής  $i$  προωθείται στην αντίστοιχη ενδιάμεση αποθήκη, η οποία τροφοδοτεί την επόμενη μηχανή. Οι μηχανές υπόκεινται σε φαινόμενα αποστέρησης και αποκλεισμού, δηλαδή μία μηχανή διακόπτει τη λειτουργία της όταν η προηγούμενη αποθήκη είναι άδεια ή όταν το απόθεμα της επόμενης αποθήκης είναι ίσο με ένα συγκεκριμένο κατώφλι. Τα έτοιμα προϊόντα αποθηκεύονται σε μία αποθήκη ετοιμών προϊόντων η οποία είναι η αποθήκη της τελευταίας μηχανής. Μία αφικνούμενη παραγγελία που βρίσκει άδεια την αποθήκη τοποθετείται στη λίστα αναμονής. Αντίθετα εάν υπάρχει έτοιμο προϊόν, ο πελάτης εξυπηρετείται αμέσως. Η ζήτηση μπορεί να μοντελοποιηθεί ως μία μηχανή που δεν υφίστανται βλάβες οπότε οι εξεταζόμενες γραμμές παραγωγής, μπορούν να θεωρηθούν σαν συστήματα συναρμολόγησης όπως φαίνεται στο Σχήμα 2.1.



Σχήμα 2.1. Παράδειγμα γραμμής παραγωγής με πέντε μηχανές

Η λειτουργία του συστήματος σχετίζεται με τρία οικονομικά μεγέθη:

- $\rho$  το κέρδος από την πώληση μίας μονάδας προϊόντος (τιμή πώλησης μείον κόστος πρώτων υλών και επεξεργασίας),

- $h_i$  το μοναδιαίο κόστος αποθέματος, που είναι το κόστος διατήρησης αποθέματος μίας μονάδας προϊόντος στην αποθήκη της μηχανής  $i$  για μία χρονική μονάδα,
- $b$  το μοναδιαίο κόστος μη ικανοποιημένης ζήτησης, που είναι το κόστος καθυστέρησης της ικανοποίησης μίας εκκρεμούσας παραγγελίας στη μονάδα του χρόνου.

Το κόστος αποθέματος  $h_i$  της αποθήκης  $i$  εμπεριέχει δύο συνιστώσες κόστους. Η μία είναι το χρηματοοικονομικό κόστος, που προκύπτει από τη δέσμευση κεφαλαίου για την αγορά πρώτων υλών. Η άλλη συνιστώσα περιλαμβάνει όλα τα είδη κόστους που σχετίζονται με τη φυσική διαδικασία συντήρησης αποθεμάτων, όπως το κόστος συντήρησης των αποθηκευτικών χώρων, το κόστος λειτουργίας ενός συστήματος διαχείρισης υλικών το κόστος κατάψυξης για κάποια είδη προϊόντων κλπ. Το κόστος μη ικανοποιημένης ζήτησης  $b$ , μπορεί να περιλαμβάνει επίσης δύο είδη κόστους. Το ένα κόστος είναι χρηματοοικονομικό και σχετίζεται με την απώλεια της ευκαιρίας επένδυσης του κέρδους από την πώληση ενός προϊόντος για το χρονικό διάστημα που μια παραγγελία μένει ανικανοποίητη. Η δεύτερη συνιστώσα του  $b$  απαρτίζεται από το τυχόν κόστος δυσφήμισης και τις ρήτρες καθυστέρησης, για την περίπτωση που ο πελάτης δεν ικανοποιηθεί αμέσως αλλά υποχρεωθεί να περιμένει.

Στο πρόβλημα μας υπάρχει συνεχώς έλεγχος αποθέματος σε κάθε αποθήκη κάθε μηχανής. Όταν λοιπόν τα κομμάτια που περιέχονται σε μία αποθήκη είναι ίσα με τη χωρητικότητα της, σταματάει η παραγωγή της προηγούμενης(αντίστοιχης) μηχανής έως ότου ελευθερωθεί μία θέση στην αποθήκη της. Το ίδιο ισχύει και για τη λίστα αναμονής πελατών στην οποία εισέρχονται πελάτες μέχρι να συμπληρωθούν όλες οι θέσεις αναμονής. Όταν συμβεί αυτό όλες οι επόμενες παραγγελίες απορρίπτονται.

Το αναμενόμενο κέρδος του συστήματος είναι

$$J = pTH - \sum_{i=1}^N h_i \bar{H}_i - b\bar{B}$$

όπου  $p$  η τιμή πώλησης μίας μονάδας προϊόντος,  $TH$  είναι ο αναμενόμενος ρυθμός παραγωγής του συστήματος,  $\bar{H}_i$  το αναμενόμενο απόθεμα της αποθήκης  $i$ , το κόστος αποθεματοποίησης της αποθήκης  $i$ ,  $b$  το κόστος μη ικανοποιημένης ζήτησης και τέλος  $\bar{B}$ , το αναμενόμενο πλήθος εκκρεμών παραγγελιών. Το αναμενόμενο κέρδος του συστήματος είναι συνάρτηση των παραμέτρων ελέγχου του συστήματος, δηλαδή των κατωφλίων ελέγχου των αποθηκών και της λίστας αναμονής των παραγγελιών.

Επειδή τα συστήματα της μορφής που εξετάζουμε είναι ιδιαίτερα σύνθετα, είναι πάρα πολύ δύσκολο να μελετηθούν αναλυτικά. Τα συστήματα αυτά μπορούν να περιγραφούν με τη βοήθεια αλυσίδων Markov αφού οι χρόνοι όλων των φαινομένων που παρατηρούνται είναι εκθετικοί. Για γραμμές παραγωγής με πολλές μηχανές, το πλήθος των δυνατών καταστάσεων είναι τόσο μεγάλο ώστε είναι πρακτικά αδύνατη η ανάλυση του συστήματος. Για το λόγο αυτό χρησιμοποιήσαμε εργαλεία από το χώρο της προσομοίωσης συστημάτων, για τη μελέτη της συμπεριφοράς των εξεταζόμενων συστημάτων.

### 2.3 Περιγραφή αλγορίθμου προσομοίωσης

Προσομοίωση είναι η μίμηση της λειτουργίας συστημάτων ή της εξέλιξης διαδικασιών μέσα στο χρόνο με τη βοήθεια υπολογιστή. Τα συστήματα παραγωγής είναι δυναμικά συστήματα, διακριτών γεγονότων, στα οποία εισέρχονται πρώτες ύλες και αφού υποστούν κάποια επεξεργασία από τις μηχανές του συστήματος, εξέρχονται ως έτοιμα προϊόντα. Ένα δυναμικό σύστημα είναι μία συλλογή οντοτήτων που εξελίσσονται στο χρόνο σύμφωνα με ορισμένους κανόνες. Οι μεταβλητές που περιγράφουν το σύστημα αποτελούν την κατάσταση του συστήματος και ονομάζονται μεταβλητές κατάστασης. Ως εξέλιξη ενός συστήματος μπορούμε να θεωρήσουμε μία ακολουθία καταστάσεων, τις οποίες επισκέπτεται στο σύστημα κατά τη διάρκεια μιας περιόδου παρατήρησης. Το μοντέλο ενός δυναμικού συστήματος είναι η συμβολική περιγραφή των κανόνων που διέπουν τη λειτουργία του και συνδέουν την τρέχουσα κατάσταση με τις προηγούμενες

καταστάσεις του συστήματος. Ένα σύστημα διακριτών γεγονότων είναι ένα σύστημα που μετακινείται με τυχαίο τρόπο μεταξύ διαφόρων καταστάσεων σε διακριτές χρονικές στιγμές, ενώ η κατάσταση του συστήματος παραμένει σταθερή μεταξύ των διαφορετικών χρονικών στιγμών. Οι αλλαγές καταστάσεων προκαλούνται από ένα σύνολο διακριτών γεγονότων που "συναγωνίζονται" για το ποιο θα συμβεί ενωρίτερα. Στη γενική του μορφή ένας αλγόριθμος προσομοίωσης συστημάτων διακριτών περιέχει τις ακόλουθες μεταβλητές

*Ρολόι προσομοίωσης:* Μια μεταβλητή που δίνει τον τρέχοντα χρόνο της προσομοίωσης  $t$ .

*Κατάσταση του συστήματος:* Συλλογή μεταβλητών για την περιγραφή του συστήματος.

*Εξισώσεις καταστάσεων:* Δίδουν τη μεταβολή των μεταβλητών καταστάσεως με την πάροδο του χρόνου.

*Χρόνος προηγούμενου γεγονότος:* Η αμέσως προηγούμενη στιγμή κατά την οποία σημειώθηκε βηματική μεταβολή στο διάνυσμα κατάστασης. Αυτή η μεταβλητή είναι απαραίτητη για την ενημέρωση των μεταβλητών κατάστασης όπως θα δούμε στη συνέχεια.

*Λίστα επόμενων γεγονότων:* Αν το σύστημα αποτελείται από πολλές συνιστώσες τότε σε κάθε μία από αυτές μπορεί να συμβεί μια βηματική μεταβολή στο μέλλον. Κάθε στοιχείο της λίστας αναφέρεται σε μία συνιστώσα του συστήματος. Είναι ένα ζευγάρι που αποτελείται από το είδος και το χρόνο του επόμενου γεγονότος που πρόκειται να συμβεί στη συγκεκριμένη συνιστώσα.

Σε μία γραμμή παραγωγής η κατάσταση του συστήματος περιγράφεται από τις ακόλουθες μεταβλητές κατάστασης:

- Την κατάσταση της μηχανής  $i$ ,

$$s_i(t) = \begin{cases} 0, & \text{αν η } i \text{ είναι υπό επισκευή τη χρονική στιγμή } t \\ 1, & \text{αν η } i \text{ είναι λειτουργική τη χρονική στιγμή } t \end{cases}$$

- Τη στάθμη της αποθήκης  $i$ ,  $H_i(t)$
- Το πλήθος των εκκρεμών παραγγελιών,  $B_i(t)$ .

Τα γεγονότα που μπορεί να συμβούν είναι:

- Παραγωγή στη μηχανή  $i$ ,
- Βλάβη στη μηχανή  $i$ ,
- Επισκευή της μηχανής  $i$ ,
- Άφιξη νέας παραγγελίας.

Η δυνατότητα υλοποίησης των γεγονότων εξαρτάται από την κατάσταση του συστήματος. Όταν η μηχανή  $i$  έχει υποστεί βλάβη προφανώς δεν είναι εφικτή η παραγωγή από τη συγκεκριμένη μηχανή. Το ίδιο ισχύει όταν  $H_{i-1} = 0$  ή  $H_i = T_i$ . Επίσης όταν  $B = T_a$  οι αφιχθείσες παραγγελίες απορρίπτονται. Τέλος όταν η  $i$  είναι λειτουργική δεν είναι εφικτή η επισκευή της.

Κάθε φορά που εκτελείται ένα γεγονός ενημερώνονται οι μεταβλητές κατάστασης της συνιστώσας του συστήματος, που επηρεάζονται από την εκτέλεση του γεγονότος, σύμφωνα με τις εξισώσεις κατάστασης του συστήματος. Οι εξισώσεις καταστάσεων των εξεταζόμενων γραμμών παραγωγής διαμορφώνονται ως εξής:

**Παραγωγή στη μηχανή  $i$ :**

$$H_{i-1}(t_k) = H_{i-1}(t_{k-1}) - 1$$

$$H_i(t_k) = \begin{cases} H_i(t_{k-1}) + 1, & \text{αν } i < N \text{ ή } i = N \text{ και } B(t_{k-1}) = 0 \\ 0, & \text{αν } i = N \text{ και } B(t_{k-1}) > 0 \end{cases}$$

$$B(t_k) = \max\{0, B(t_{k-1}) - 1\}$$

$$P_i(t_k) = P_i(t_{k-1}) + 1$$

**Βλάβη στη μηχανή  $i$ :**

$$s_i(t_k) = 0$$

**Επισκευή της μηχανής  $i$ :**

$$s_i(t_k) = 1$$

### Αφιξη νέας παραγγελίας:

$$H_N(t_k) = \max\{0, H_N(t_{k-1}) - 1\}$$

$$B(t_k) = \begin{cases} B(t_{k-1}) + 1, & \text{αν } H_N(t_{k-1}) = 0 \\ 0, & \text{αν } H_N(t_{k-1}) > 0 \end{cases}$$

όπου  $t_k$  είναι η χρονική στιγμή εκτέλεσης του γεγονότος  $k$ ,  $t_{k-1}$  είναι η χρονική στιγμή που εκτελείται το προηγούμενο γεγονός και  $P_i(t)$  είναι η αθροιστική παραγωγή της μηχανής  $i$  τη χρονική στιγμή  $t$ .

Εκτός από τις μεταβλητές κατάστασης ιδιαίτερα σημαντική είναι η παρακολούθηση και εκτίμηση κάποιων μεταβλητών απόδοσης του συστήματος. Στην περίπτωση μας θέλουμε να έχουμε μία εκτίμηση του αναμενόμενου κέρδους του συστήματος. Όπως έχουμε δει, το αναμενόμενο κέρδος του συστήματος  $J$  είναι γραμμική συνάρτηση των μέσων αποθεμάτων των αποθηκών  $\bar{H}_i$ , του μέσου πλήθους εκκρεμών παραγγελιών  $\bar{B}$  και του αναμενόμενου ρυθμού παραγωγής του συστήματος TH. Τα αποθέματα και οι εκκρεμείς παραγγελίες είναι κατά τμήματα συνεχείς συναρτήσεις της κατάστασης του συστήματος και γι' αυτό οι εκτιμήσεις των μέσων τιμών τους προκύπτουν όπως φαίνεται στη συνέχεια

$$\bar{H}_i = \frac{1}{T_{sim}} \int_0^{T_{sim}} H_i(t) dt = \frac{1}{T_{sim}} \sum_{k=1}^K (t_k - t_{k-1}) H_i(t_{k-1})$$

$$\bar{B} = \frac{1}{T_{sim}} \int_0^{T_{sim}} B(t) dt = \frac{1}{T_{sim}} \sum_{k=1}^K (t_k - t_{k-1}) B(t_{k-1})$$

$$TH = \frac{P_N(T_{sim})}{T_{sim}}$$

όπου  $T_{sim}$  είναι η διάρκεια της προσομοίωσης.

Αφού όλοι οι χρόνοι είναι εκθετικοί μπορεί να θεωρηθεί ότι η διαδικασία "γέννησης" γεγονότων είναι μια διαδικασία Poisson με μέσο ρυθμό

$$R_e = \lambda + \sum_{i=1}^N \mu_i + \sum_{i=1}^N f_i + \sum_{i=1}^N r_i$$

Αυτό σημαίνει ότι οι χρόνοι μεταξύ δύο διαδοχικών γεγονότων είναι εκθετικοί με ρυθμό  $R_e$ . Συνεπώς αν το γεγονός  $k-1$  πραγματοποιήθηκε τη χρονική στιγμή  $t_{k-1}$  το γεγονός  $k$  πραγματοποιείται τη χρονική στιγμή  $t_k = t_{k-1} - \ln(u)/R_e$ . Ο όρος  $-\ln(u)/R_e$  εκφράζει ένα ψευδοτυχαίο αριθμό που ακολουθεί την



εκθετική κατανομή με ρυθμό  $R_e$ , ενώ το  $u$  είναι ένας ψευδοτυχαίος αριθμός που ακολουθεί την ομοιόμορφη κατανομή  $U(0,1)$  και προκύπτει με τη χρήση μιας γεννήτριας τυχαίων αριθμών.

Το είδος του επόμενου γεγονότος καθορίζεται επίσης με "τυχαίο" τρόπο σύμφωνα με τις ακόλουθες πιθανότητες

- $P\{\text{επόμενο γεγονός παραγωγή στη μηχανή } i\} = \mu_i/R_e$
- $P\{\text{επόμενο γεγονός βλάβη στη μηχανή } i\} = f_i/R_e$
- $P\{\text{επόμενο γεγονός επισκευή της μηχανή } i\} = r_i/R_e$
- $P\{\text{επόμενο γεγονός άφιξη νέας παραγγελίας}\} = \lambda/R_e$

Αν κάποιο γεγονός δεν είναι δυνατόν να εκτελεστεί, όπως για παράδειγμα η άφιξη νέας παραγγελίας ενώ οι εκκρεμείς παραγγελίες είναι  $B = T_a$ , παραλείπεται και προχωράμε στην υλοποίηση του επόμενου γεγονότος.

Με τη χρήση ενός αλγόριθμου, προσομοίωσης όπως αυτός που περιγράψαμε παραπάνω, μπορεί να γίνει ανάλυση δυναμικών συστημάτων, τα οποία δε μπορούν να περιγραφούν ή να μελετηθούν αναλυτικά. Στόχος μας σε αυτή την εργασία είναι η εκτίμηση των τιμών των παραμέτρων ελέγχου για τις οποίες βελτιστοποιείται το αναμενόμενο κέρδος του συστήματος. Για την αντιμετώπιση του παραπάνω προβλήματος χρησιμοποιείται ένας αλγόριθμος βελτιστοποίησης, που απαιτεί την εκτίμηση των μερικών παραγώγων του κέρδους ως προς τις παραμέτρους ελέγχου. Για το σκοπό αυτό χρησιμοποιείται μία μεθοδολογία ανάλυσης ευαισθησίας, που είναι γνωστή ως τεχνική ωρολογίου αναφοράς. Επειδή οι μεταβλητές ελέγχου είναι διακριτές πρέπει να εκτιμηθούν οι διαφορές της συνάρτησης κέρδους ως προς τις μεταβλητές ελέγχου, που ορίζονται ως εξής

$$\Delta J_i(T_1, \dots, T_i, \dots, T_N, T_a) = J(T_1, \dots, T_i + 1, \dots, T_N, T_a) - J(T_1, \dots, T_i, \dots, T_N, T_a)$$

Για κάθε διαφορά ζητάμε την μεταβολή της απόδοσης του συστήματος, αν μεταβληθεί η τιμή της αντίστοιχης μεταβλητής ελέγχου κατά μία μονάδα. Αφού έχουμε  $N + 1$  παραμέτρους ελέγχου χρειαζόμαστε τις τιμές της συνάρτησης κέρδους για  $N + 2$  συστήματα. Για λόγους μείωσης του υπολογιστικού φόρτου χρησιμοποιήσαμε την τεχνική σταθερού ωρολογίου αναφοράς. Η τεχνική

ωρολογίου αναφοράς συνίσταται στην ταυτόχρονη προσομοίωση του αρχικού συστήματος και όλων των διαταραγμένων συστημάτων, που προκύπτουν όταν μεταβάλλουμε μία παράμετρο κάθε φορά κατά βηματικό τρόπο, χρησιμοποιώντας το ίδιο σύνολο τυχαίων αριθμών. Με την χρήση κοινών τυχαίων αριθμών επιτυγχάνεται μείωση του υπολογιστικού φόρτου αλλά και καλύτερη εκτίμηση της μεταβολής της απόδοσης, αφού όλα τα συστήματα συγκρίνονται με βάση την απόδοσή τους στο ίδιο "σενάριο" γεγονότων.

Η προσομοίωση είναι ένα στατιστικό εργαλείο ανάλυσης δυναμικών συστημάτων. Η ακρίβεια των αποτελεσμάτων είναι συνάρτηση της διάρκειας και του πλήθους των προσομοιώσεων. Επειδή η χρήση της προσομοίωσης είναι ιδιαίτερα χρονοβόρα διαδικασία προσπαθούμε να υπολογίσουμε την ελάχιστη διάρκεια της προσομοίωσης  $T_{sim}$  για την οποία επιτυγχάνουμε ικανοποιητική ακρίβεια στις εκτιμήσεις των μέτρων απόδοσης που μας ενδιαφέρουν. Για την ακρίβεια ξεκινώντας από μία αρχική τιμή της διάρκειας των προσομοιώσεων  $T_{sim,0}$  και χρησιμοποιώντας τη στατιστική θεωρία των διαστημάτων εμπιστοσύνης εκτιμούμε το ελάχιστο πλήθος  $n$  των προσομοιώσεων, που απαιτούνται ώστε το μέγιστο σχετικό σφάλμα στην εκτίμηση των διαφορών του αναμενόμενου κέρδους  $\Delta J_i$ , να είναι μικρότερο από μία επιθυμητή τιμή  $e_d$  με πιθανότητα  $1 - \alpha$ . Στην περίπτωση μας, η επιθυμητή τιμή του σχετικού σφάλματος  $e_d$  είναι 0.01 με πιθανότητα  $1 - \alpha = 0.95$ .

Από την θεωρία των διαστημάτων εμπιστοσύνης γνωρίζουμε, ότι για πιθανότητα  $1 - \alpha$ , η μέγιστη απόκλιση της εκτίμησης της διαφοράς  $\Delta J_i$  από την πραγματική της τιμή δίδεται από τον τύπο :

$$e = z_{\alpha/2} \sqrt{\frac{S_{(n)}^2(\Delta J_i)}{n}}$$

όπου  $z_{\alpha/2}$  είναι το σημείο της κανονικής κατανομής  $N(0,1)$ , για το οποίο  $P(Z_n \geq z) = \alpha/2$ ,  $S_{(n)}^2(\Delta J_i)$  είναι η δειγματική διασπορά της  $\Delta J_i$  μετά από  $n$  προσομοιώσεις και υπολογίζεται από τον τύπο :

$$S_{(n)}^2(\Delta J_i) = \frac{\sum_{k=1}^n (\Delta J_i^{(k)} - \overline{\Delta J_i^{(n)}})^2}{n-1}$$

όπου  $\Delta J_i^{(k)}$  είναι η εκτίμηση της  $\Delta J_i$  από την  $k$  προσομοίωση και  $\overline{\Delta J_i^{(n)}}$  ο δειγματικός μέσος όρος της  $\Delta J_i$  μετά από  $n$  προσομοιώσεις. Αποδεικνύεται ότι

εάν ικανοποιείται η ανισότητα  $e < \frac{e_d |\overline{\Delta J_i^{(n)}}|}{1+e_d}$  (π.χ. βλ. Κουϊκόγλου [11]) τότε το

σχετικό σφάλμα είναι μικρότερο της επιθυμητής τιμής  $e_d$ . Το μικρότερο  $n$ , για το οποίο ικανοποιείται η παραπάνω σχέση για όλες τις διαφορές  $\Delta J_i$ , χρησιμοποιείται για την τελική επιλογή της διάρκειας των προσομοιώσεων, η οποία είναι ίση με  $T_{sim} = nT_{sim,0}$ .

## 2.4 Αλγόριθμος Βελτιστοποίησης

Όπως αναφέραμε προηγουμένως στόχος μας είναι η εκτίμηση των τιμών των μεταβλητών ελέγχου, δηλαδή των κατωφλίων ελέγχου των αποθεμάτων και των εκκρεμών παραγγελιών, που μεγιστοποιούν το αναμενόμενο κέρδος του συστήματος. Το αναμενόμενο κέρδος του συστήματος είναι μια μη γραμμική συνάρτηση των μεταβλητών ελέγχου. Για το λόγο αυτό η μέθοδος που χρησιμοποιήσαμε προέρχεται από το χώρο της μη γραμμικής βελτιστοποίησης. Συγκεκριμένα χρησιμοποιήσαμε μία παραλλαγή της μεθόδου μεγίστης ανόδου. Η μέθοδος αυτή εγγυάται την εύρεση της βέλτιστης λύσης σε προβλήματα, που η αντικειμενική συνάρτηση είναι κοίλη ή σχεδόν κοίλη. Δυστυχώς δεν υπάρχουν εργασίες, στις οποίες να μελετώνται οι δομικές ιδιότητες της συνάρτησης κέρδους σε συστήματα παραγωγής αντίστοιχα με αυτό που εξετάζεται εδώ. Υπάρχουν αρκετές εργασίες που εξετάζουν τις ιδιότητες του μέσου ρυθμού παραγωγής και του μέσου συνολικού αποθέματος συναρτήσει των χωρητικοτήτων των αποθηκών σε μία σειρά από συστήματα παραγωγής. Δυστυχώς όμως δεν έχουν μελετηθεί επαρκώς οι δομικές ιδιότητες του μέσου πλήθους εκκρεμών παραγγελιών, που αποτελεί μια από τις συνιστώσες κόστους του υπό μελέτη συστήματος. Πάντως από τα μέχρι τώρα εμπειρικά δεδομένα αλλά και από τα

δικά μας αποτελέσματα δεν υπάρχουν ενδείξεις ότι η συνάρτηση κέρδους δεν είναι σχεδόν κοίλη. Υποθέτοντας λοιπόν ότι είναι κοίλη, εφαρμόσαμε τη μέθοδο μεγίστης ανόδου.

Η μέθοδος μεγίστης ανόδου είναι μία αριθμητική μέθοδος διαδοχικών προσεγγίσεων της λύσης. Αρχικά επιλέγονται συνήθως τυχαία κάποιες τιμές των μεταβλητών απόφασης. Μεταβάλλουμε τις τιμές αυτές προς τη διεύθυνση του διανύσματος κλίσης. Επαναλαμβάνουμε την παραπάνω διαδικασία μέχρι να προκύψει μία ικανοποιητική προσέγγιση της βέλτιστης λύσης. Η μέθοδος αυτή εφαρμόζεται συνήθως σε προβλήματα όπου οι μεταβλητές απόφασης είναι συνεχείς. Εδώ όμως οι μεταβλητές ελέγχου είναι διακριτές. Αυτό μας υποχρέωσε να προχωρήσουμε σε κάποιες αλλαγές σε σχέση με την κλασσική μορφή της μεθόδου. Καταρχήν αντί των μερικών παραγώγων χρησιμοποιούμε τις διαφορές  $\Delta J_i$ . Επίσης είναι απαραίτητη η ακεραιοποίηση των τιμών των μεταβλητών απόφασης που υπολογίζονται σε κάθε βήμα της μεθόδου. Ακόμη χρειάστηκε να τροποποιήσουμε το κριτήριο σύγκλισης της μεθόδου. Όταν έχουμε να κάνουμε με συνεχής και διαφορίσιμες συναρτήσεις, στο σημείο που υπάρχει τοπικό βέλτιστο οι μερικές παράγωγοι είναι μηδέν. Στην περίπτωση που εξετάζουμε οι διαφορές δεν είναι πάντα μηδέν. Γι' αυτό το λόγο στον αλγόριθμο που χρησιμοποιήσαμε η μέθοδος της μεγίστης ανόδου τερματίζεται όταν δε μπορούμε να βελτιώσουμε άλλο το κέρδος με τη χρήση της συγκεκριμένης μεθόδου. Για τον καθορισμό του βέλτιστου βήματος μεταβολής των μεταβλητών απόφασης, προς την διεύθυνση του διανύσματος κλίσης, χρησιμοποιήσαμε μια παραλλαγή της μεθόδου της χρυσής τομής. Λόγω της διακριτοποίησης, η μέθοδος μεγίστης ανόδου μπορεί να καταλήγει σε μία λύση που προσεγγίζει αλλά δεν είναι η βέλτιστη. Για το λόγο αυτό χρησιμοποιήσαμε συμπληρωματικά έναν ευρετικό αλγόριθμο τοπικής έρευνας, για να βελτιώσουμε αν είναι δυνατόν τα αποτελέσματα που μας δίνει η μέθοδος μεγίστης ανόδου.

Ο αλγόριθμος μεγίστης ανόδου εκτελεί τα ακόλουθα βήματα:

Βήμα 0 (Έναρξη)

α. Αρχικές τιμές κατωφλίων  $T_i^{(0)}$ , θέσε δείκτη επανάληψης  $k = 0$

β. Καλείται ο προσομοιωτής και γίνεται εκτίμηση των  $J^{(0)}$ ,  $\Delta J_i^{(0)}$ .

### Βήμα 1 (Υπολογισμός βέλτιστου βήματος)

Καλείται ο αλγόριθμος προσομοίωσης και εκτιμώνται οι τιμές της συνάρτησης κέρδους για  $m + 1$  διαφορετικές τιμές βήματος. Επιλέγεται το βήμα  $\gamma_j^{(k)}$  που επιτυγχάνει τη μεγαλύτερη βελτίωση της συνάρτησης κέρδους.

α. Για κάθε  $j = 1, \dots, m + 1$ , υπολογίζεται το βήμα από τον τύπο  $\gamma_j^{(k)} = 0.618^j c$ , όπου  $c$  είναι η μέγιστη τιμή του βήματος.

β. Οι τιμές των παραμέτρων ελέγχου μεταβάλλονται ως εξής

$$T_{i,j}^{(k+1)} = \begin{cases} \max(1, \lfloor T_i^{(k)} + \gamma_j^{(k)} \Delta J_{i,j}^{(k)} + 0.5 \rfloor), & i \neq a \\ \max(0, \lfloor T_i^{(k)} + \gamma_j^{(k)} \Delta J_{i,j}^{(k)} + 0.5 \rfloor), & i = a \end{cases} \quad \text{για κάθε } j = 1, \dots, m + 1.$$

γ. Επιλέγεται το βήμα  $\gamma_j^{(k)}$  και οι αντίστοιχες τιμές των παραμέτρων ελέγχου που δίδουν τη μέγιστη τιμή της συνάρτησης κέρδους.

δ. Αν  $J^{(k+1)} < J^{(k)}$  πηγαίνει στο Βήμα 1α και θέσει  $c \leftarrow 0.618^m c$

### Βήμα 2 (Κριτήριο σύγκλισης)

Αν  $J^{(k+1)} = J^{(k)}$  θέσει  $k \leftarrow k + 1$  και πηγαίνει στο Βήμα 3 αλλιώς

Αν  $J^{(k+1)} > J^{(k)}$  θέσει  $k \leftarrow k + 1$  και πηγαίνει στο Βήμα 1

### Βήμα 3 (Αλγόριθμος τοπικής αναζήτησης)

Προσάρμοσε τις μεταβλητές απόφασης ως εξής

$$T_{i,j}^{(k+1)} = \begin{cases} T_i^{(k)}, & i \neq j \\ T_i^{(k)} + 1, & i = j \text{ και } \Delta J_{i,j}^{(k)} > 0 \\ T_i^{(k)} - 1, & i = j \text{ και } \Delta J_{i,j}^{(k)} \leq 0 \end{cases} \quad \text{για κάθε } i, j = 1, \dots, N + 1.$$

Κάλεσε τον αλγόριθμο προσομοίωσης. Θέσει  $J^{(k+1)} = \max_j [J_j^{(k)}]$  και

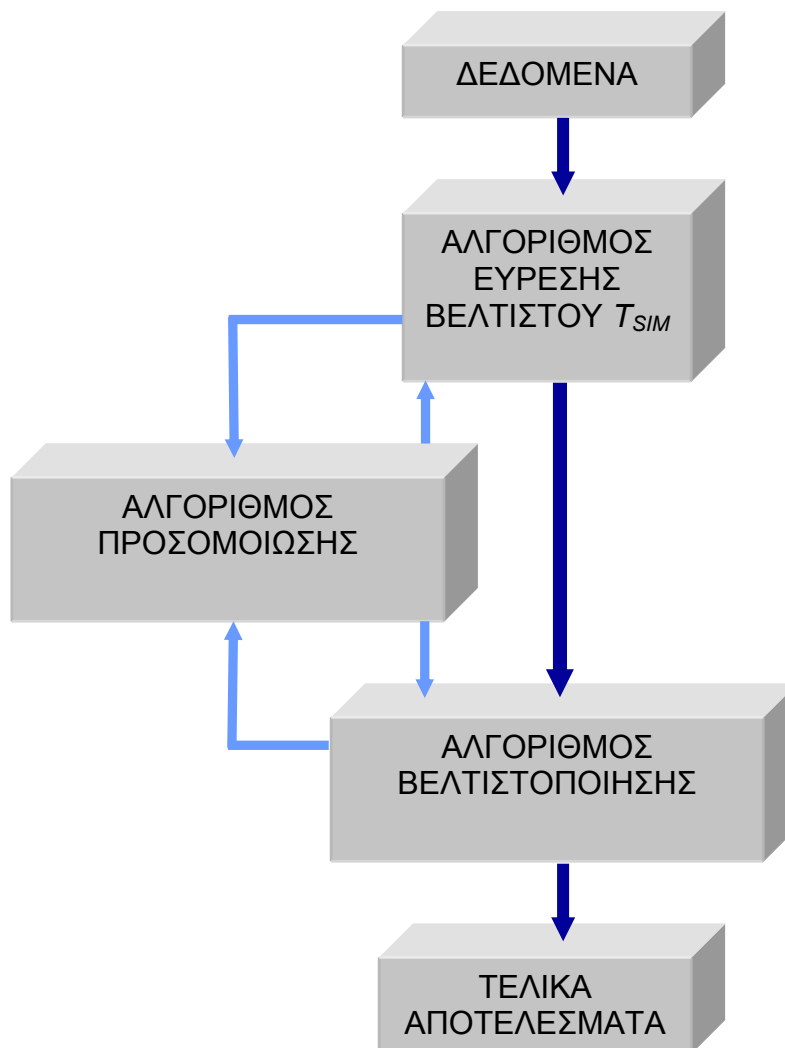
επέλεξε τις αντίστοιχες τιμές των παραμέτρων ελέγχου.

### Βήμα 4 (Κριτήριο σύγκλισης αλγόριθμου τοπικής αναζήτησης)

Αν  $J^{(k+1)} > J^{(k)}$  θέσε  $k \leftarrow k + 1$  και πήγαινε στο Βήμα 3 αλλιώς

Αν  $J^{(k+1)} = J^{(k)}$  τέλος.

Το διάγραμμα ροής, που απεικονίζει σχηματικά τη δομή και τη ροή του συνολικού αλγορίθμου και την αλληλεπίδραση του κάθε βασικού δομικού στοιχείου με τα υπόλοιπα παρουσιάζεται στο Σχήμα 2.3.



Σχήμα 2.3. Διάγραμμα ροής

Τα δεδομένα που απαιτούνται για την εκκίνηση του αλγορίθμου είναι η αρχική διάρκεια προσομοίωσης  $T_{sim,0}$ , ο ρυθμός αφίξεων παραγγελιών  $\lambda$ , οι ρυθμός παραγωγής  $\mu_i$ , βλαβών  $f_i$  και επισκευής  $r_i$  κάθε μηχανής  $i$ , οι αρχικές τιμές των μεταβλητών ελέγχου  $T_i$ , και  $T_a$ , η τιμή πώλησης των προϊόντων  $p$ , το

μοναδιαίο κόστος εκκρεμών παραγγελιών  $b$  και τα μοναδιαία κόστη αποθεματοποίησης  $h_i$  κάθε αποθήκης  $i$  του συστήματος.

## ΚΕΦΑΛΑΙΟ 3

### Αριθμητικό Παράδειγμα

#### 3.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζονται τα αποτελέσματα των αριθμητικών πειραμάτων που πραγματοποιήθηκαν, για τη σύγκριση της προτεινόμενης πολιτικής ελέγχου αποδοχής παραγγελιών, PLS, με τις πολιτικές CB και LS, οι οποίες χρησιμοποιούνται ευρύτατα. Τα αριθμητικά αποτελέσματα επιβεβαίωσαν την εκτίμηση μας, ότι η πολιτική μερικής αποδοχής παραγγελιών υπερέρχει των ευρύτατα διαδεδομένων πολιτικών πλήρους απόρριψης και πλήρους αποδοχής.

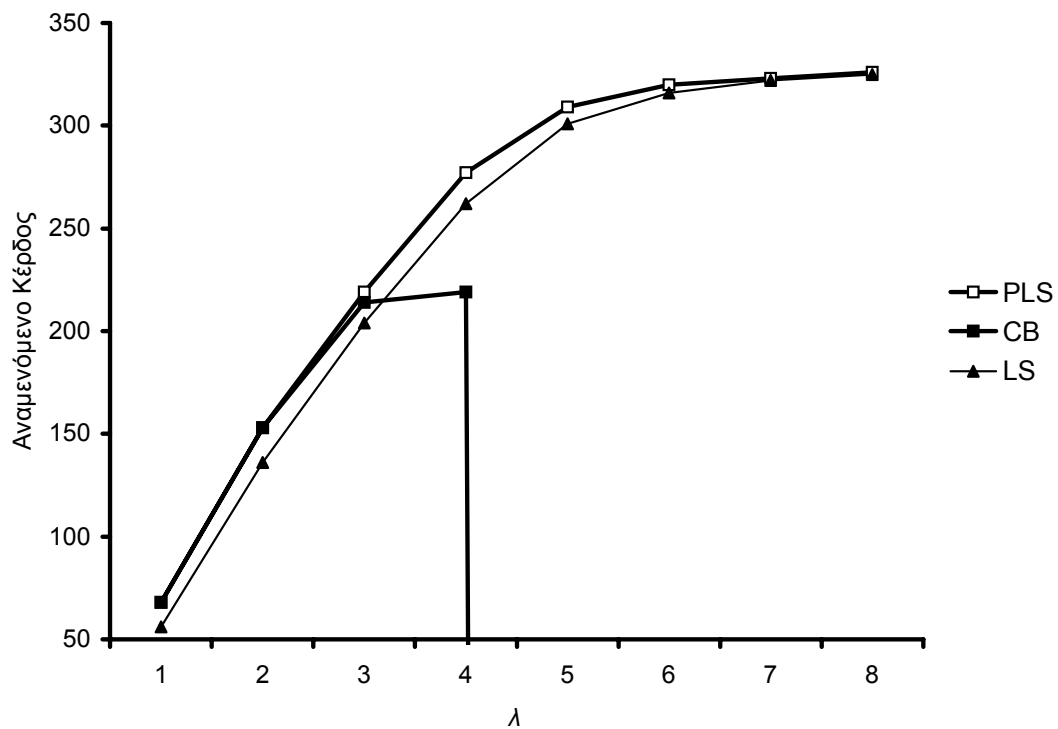
#### 3.2 Αριθμητικά Αποτελέσματα

Το σύστημα που χρησιμοποιήθηκε για τη σύγκριση είναι μία γραμμή παραγωγής με πέντε μηχανές  $M_i$ ,  $i = 1, \dots, 5$ . Οι αρχικές τιμές των παραμέτρων του συστήματος έχουν ως εξής,  $\lambda = 4.0$ ,  $\mu_1 = 6.0$ ,  $\mu_2 = 7.0$ ,  $\mu_3 = 5.0$ ,  $\mu_4 = 5.5$ ,  $\mu_5 = 6.5$ ,  $r_1 = \dots = r_5 = 5.0$ ,  $f_1 = \dots = f_5 = 0.2$ ,  $p = 100$ ,  $h_1 = \dots = h_5 = 8.0$  και  $b = 8.0$ . Διερευνήσαμε την επίδραση της μεταβολής διαφόρων παραμέτρων του συστήματος, όπως η ζήτηση  $\lambda$ , τα μοναδιαία κόστη αποθέματος  $h_i$ , το μοναδιαίο κόστος μη ικανοποιημένης ζήτησης  $b$ , την τιμή πώλησης ανά μονάδα προϊόντος  $p$ , τους ρυθμούς επισκευής  $r_i$ , και τους ρυθμούς βλαβών  $f_i$ , στο αναμενόμενο κέρδος με την προϋπόθεση ότι όταν μεταβάλλεται κάποιο μέγεθος, όλα τα υπόλοιπα μεγέθη παραμένουν σταθερά σύμφωνα με τις παραπάνω τιμές τους. Στα Σχήματα που ακολουθούν και στον Πίνακα 3.1 παρουσιάζονται τα αριθμητικά αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν.

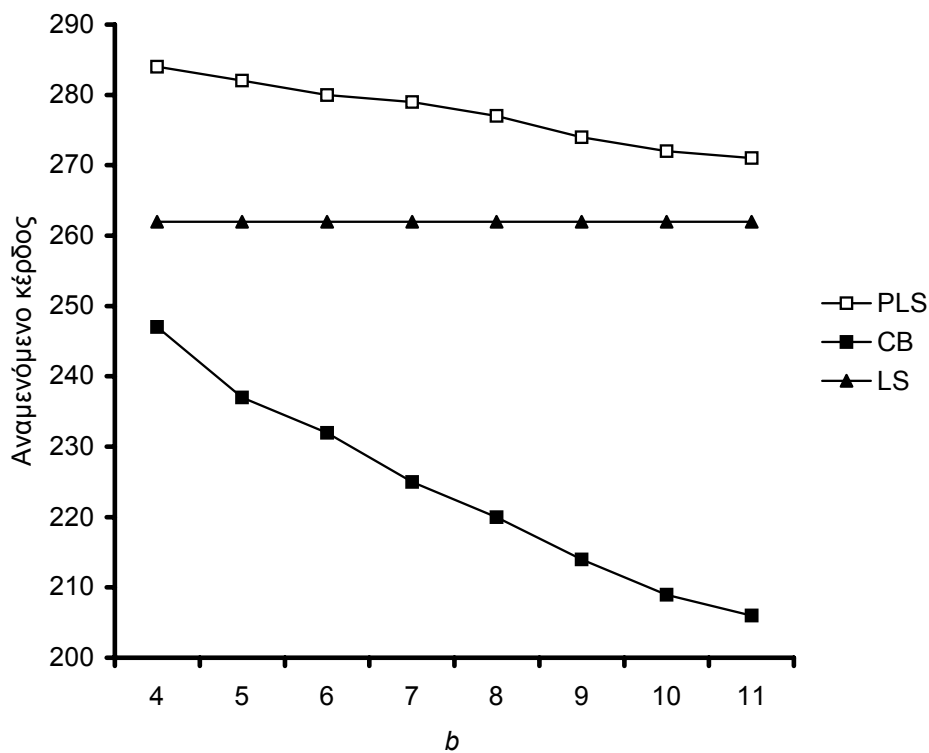


**Πίνακας 3.1.** Αναμενόμενο κέρδος και τιμές των παραμέτρων ελέγχου για διάφορες τιμές του ρυθμού ζήτησης

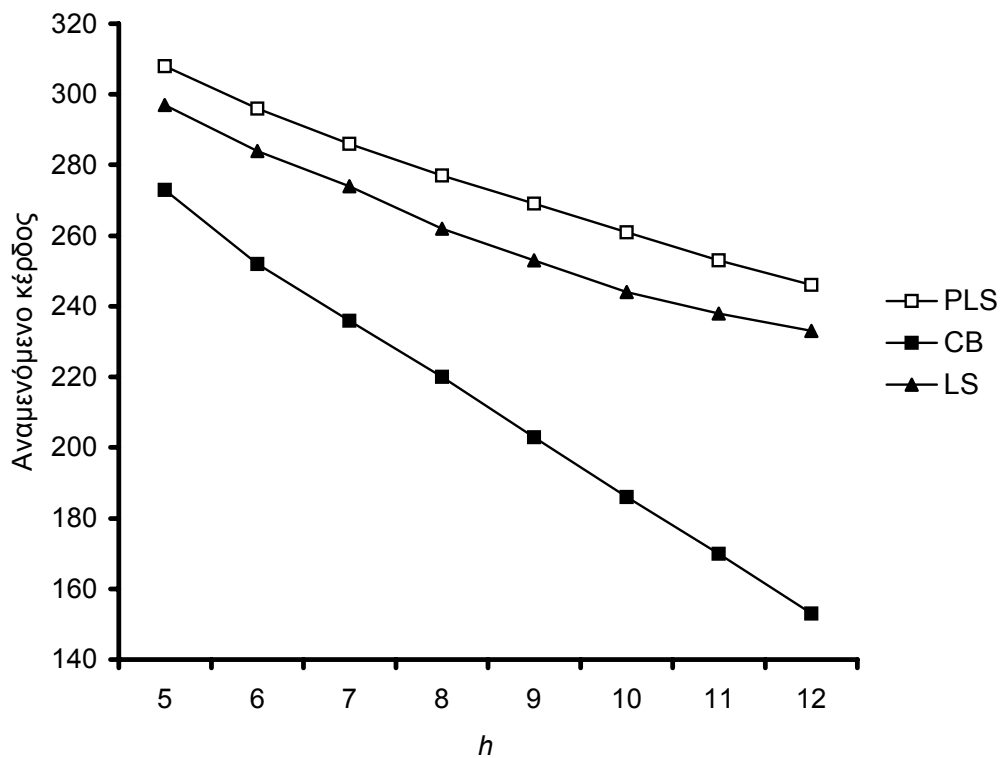
Πολιτική	Ζήτηση								
	$\lambda=2$			$\lambda=2$			$\lambda=2$		
	PLS	CB	LS	PLS	CB	LS	PLS	CB	LS
$T_1$	1	1	1	2	5	2	3	-	3
$T_2$	2	2	1	4	6	3	6	-	5
$T_3$	2	2	2	6	10	4	11	-	11
$T_4$	2	2	2	6	8	5	11	-	10
$T_5$	1	1	4	6	3	6	10	-	10
$T_a$	9	$\infty$	0	4	$\infty$	0	1	$\infty$	0
$J$	153.55	153.08	136.22	277.13	218.79	262.04	323.31	$-\infty$	321.8



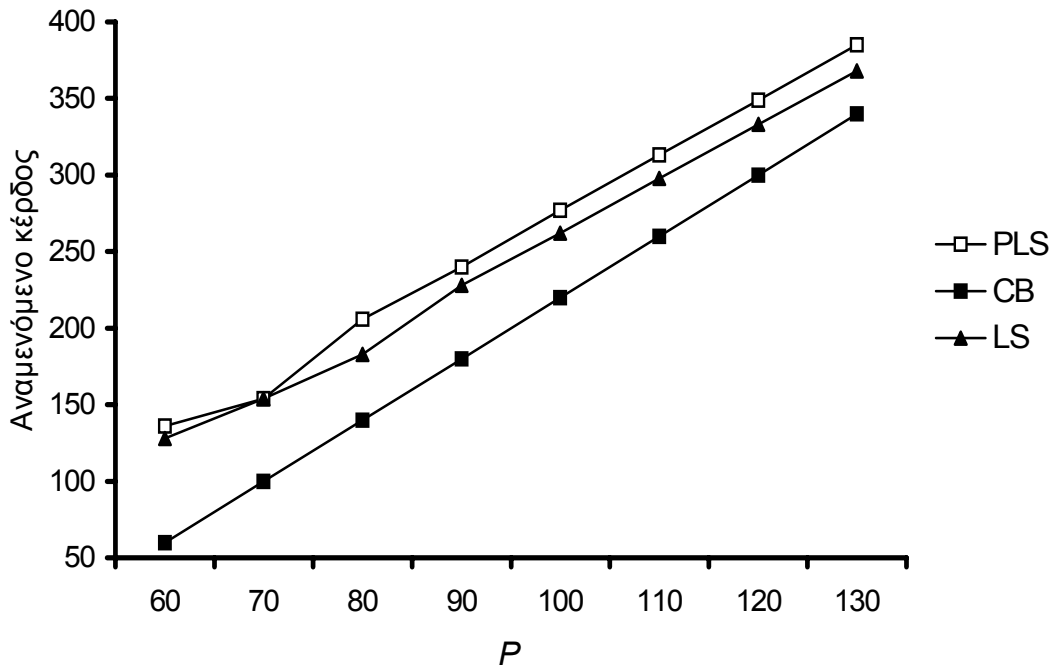
Σχήμα 3.1. Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς το  $\lambda$



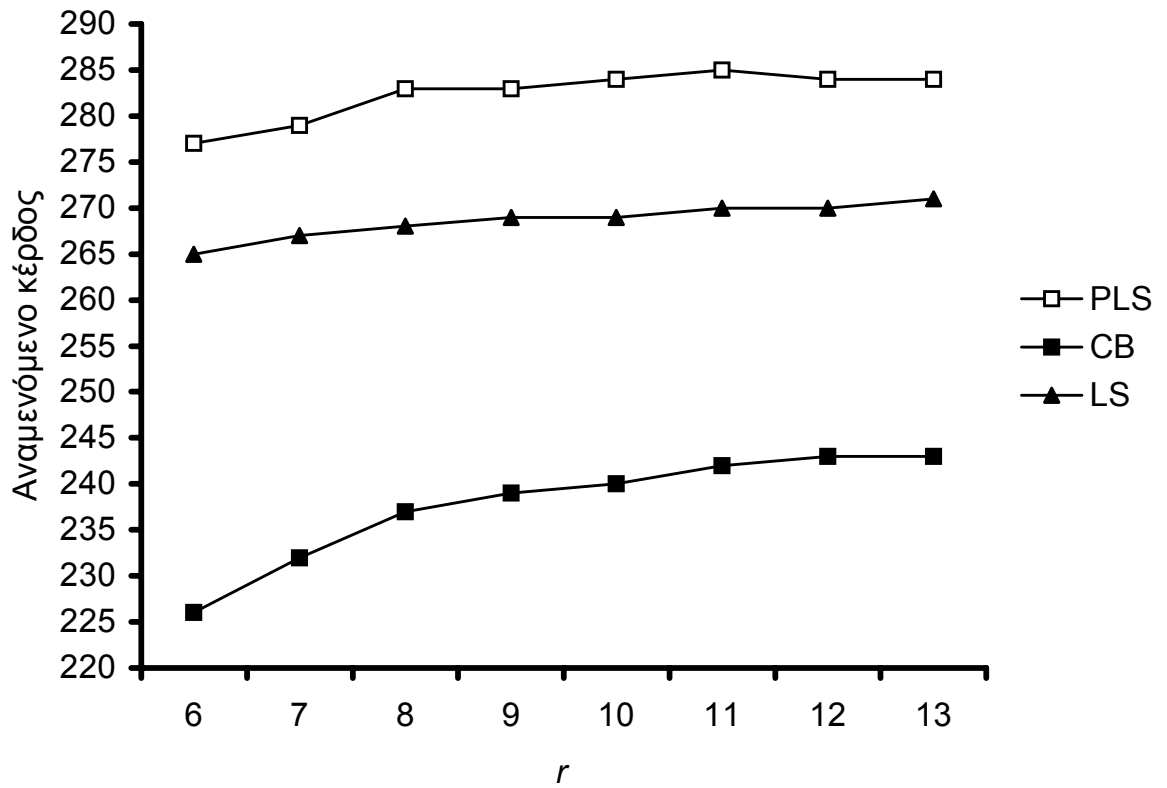
Σχήμα 3.2. Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $b$



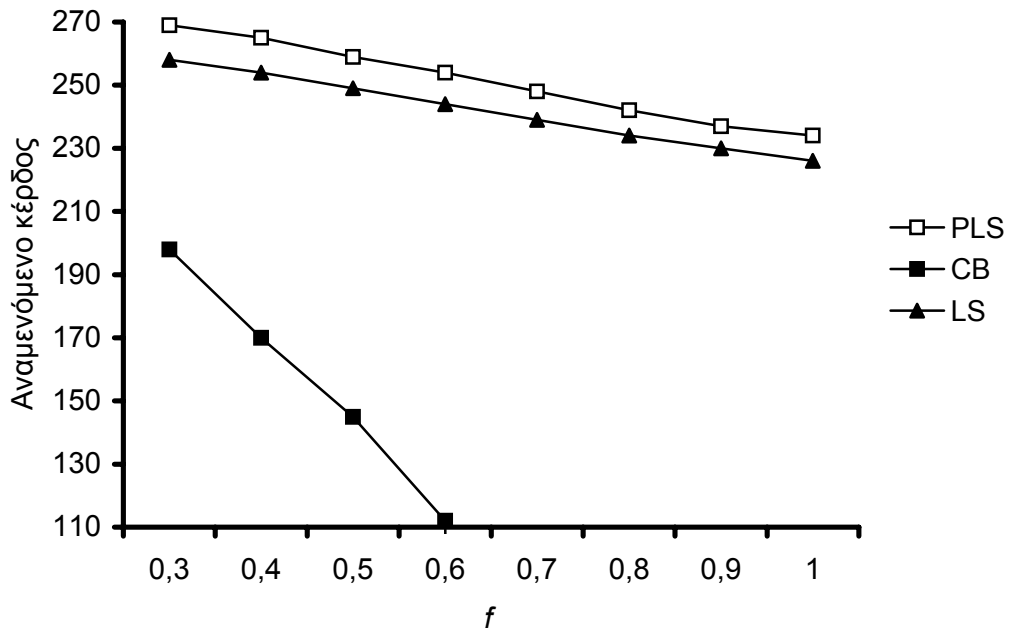
Σχήμα 3.3. Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $h$



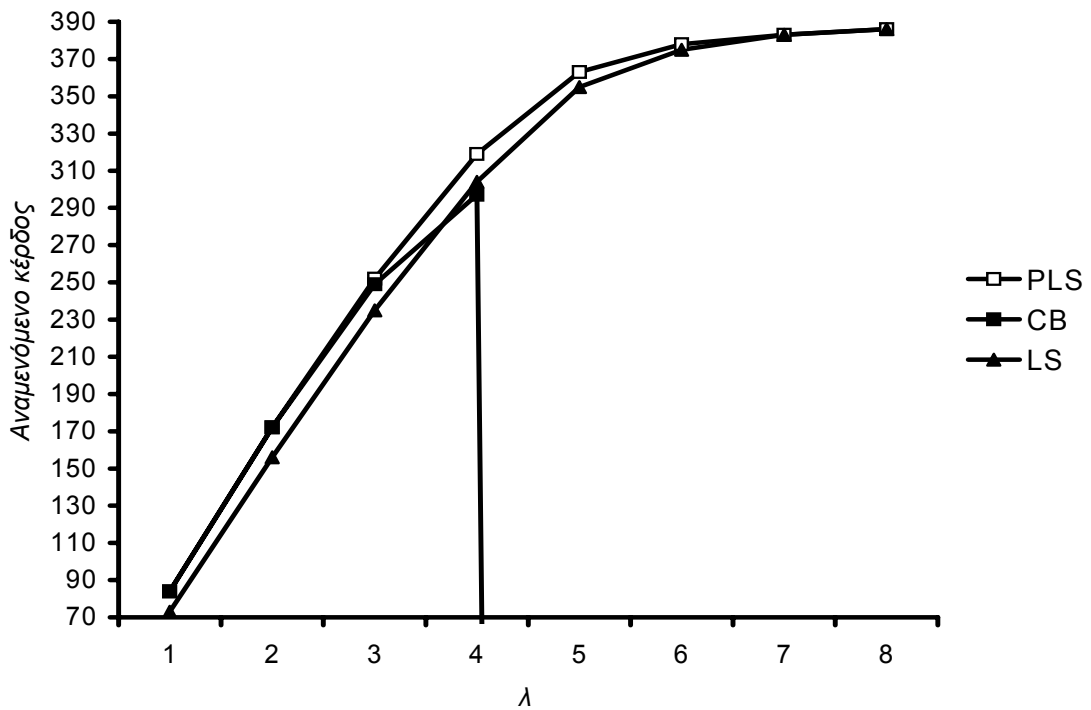
Σχήμα 3.4. Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $P$



Σχήμα 3.5. Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $r$



Σχήμα 3.6.Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $f$



Σχήμα 3.7.Αναμενόμενο κέρδος της γραμμής παραγωγής ως προς  $\lambda$  με διαφορετικά μοναδιαία κόστη αποθεμάτων ανά αποθήκη

Από τα παραπάνω αποτελέσματα φαίνεται ότι η προτεινόμενη πολιτική σε όλες τις υπό εξέταση περιπτώσεις υπερέχει των συνηθέστερα χρησιμοποιούμενων πολιτικών. Για παράδειγμα στο Σχήμα 3.1 που παρουσιάζεται η μεταβολή του αναμενόμενου κέρδους ως προς το ρυθμό ζήτησης παρατηρούμε ότι το κέρδος της PLS και της LS είναι αύξουσα και κοίλη συνάρτηση της ζήτησης. Σε καμία περίπτωση η PLS δεν αποδίδει χειρότερα από τις άλλες πολιτικές. Όταν το  $\lambda \rightarrow 0$  η πολιτική CB επιτυγχάνει το ίδιο αναμενόμενο κέρδος με την PLS. Αυτό συμβαίνει γιατί στην περίπτωση αυτή το μέσο πλήθος παραγγελιών είναι πολύ μικρό και ο περιορισμός του δεν οδηγεί σε σημαντική αύξηση του αναμενόμενου κέρδους.

Αντίθετα όταν η ζήτηση πάρει τιμές ίσες και μεγαλύτερες από την ικανότητα παραγωγής του συστήματος, ( $\lambda > 4.8$ ), η πολιτική CB δεν αποδίδει κέρδη και γίνεται πολύ ζημιογόνος. Αυτό οφείλεται στο ότι το σύστημα παραγωγής δεν μπορεί να εξυπηρετήσει όλη τη ζήτηση με αποτέλεσμα τη συσσώρευση παραγγελιών στη λίστα αναμονής, και τον απειρισμό του κόστους παραγωγής. Όσον αφορά την πολιτική LS παρατηρούμε ότι αυξάνεται με όμοιο τρόπο με την πολιτική PLS αποδίδοντας όμως μικρότερα κέρδη. Βέβαια για τιμές του ρυθμού αφίξεων αρκετά μεγάλες ( $\lambda > 6$ ), βλέπουμε ότι οι PLS και LS ταυτίζονται. Αυτό συμβαίνει γιατί τότε οι αφίξεις των πελατών είναι πολύ συχνές με αποτέλεσμα το μέσο πλήθος εκκρεμών παραγγελιών να είναι αυξημένο. Αυτό έχει ως αποτέλεσμα η απώλεια κέρδους λόγω απόρριψης πελατών να είναι αμελητέα σε σχέση με το κόστος εκκρεμών παραγγελιών. Όταν η ζήτηση προσεγγίζει τη μέγιστη παραγωγικότητα του συστήματος, η προτεινόμενη πολιτική υπερέχει σημαντικά των άλλων. Η διαφορά στην απόδοση της προτεινόμενης πολιτικής σε κάποιες περιπτώσεις φτάνει το 15%.

Η προσεκτική μελέτη του Πίνακα 3.1 στον οποίο παρουσιάζονται οι τιμές των παραμέτρων ελέγχου των συγκρινόμενων πολιτικών για διάφορες τιμές της ζήτησης, επιβεβαιώνει τα συμπεράσματα που προκύπτουν από τη μελέτη του Σχήματος 3.1 και βοηθάει στην καλύτερη κατανόηση των μηχανισμών που διέπουν τη λειτουργία του συστήματος.

Για  $\lambda = 2$ , βλέπουμε ότι οι τιμές των παραμέτρων ελέγχου της PLS συμπίπτουν με αυτές της CB, ενώ όταν  $\lambda = 7$ , οι τιμές των παραμέτρων ελέγχου της PLS είναι παρόμοιες με αυτές της πολιτικής LS. Όταν  $\lambda = 4$ , δηλαδή η ζήτηση προσεγγίζει τη μέγιστη παραγωγικότητα της γραμμής παραγωγής, βλέπουμε ότι οι τιμές των κατωφλίων ελέγχου των αποθηκών για την πολιτική CB είναι μεγαλύτερες από τις αντίστοιχες τιμές της PLS. Αυτό συμβαίνει γιατί στην πολιτική CB δεχόμαστε όλους τους πελάτες και συνεπώς πρέπει ο ρυθμός παραγωγής του συστήματος να είναι τουλάχιστον ίσος με τη ζήτηση. Επειδή ο ρυθμός παραγωγής είναι αύξουσα και κοίλη συνάρτηση των χωρητικότητων των αποθηκών, στην CB χρειαζόμαστε μεγαλύτερες χωρητικότητες αποθηκών. Άμεση συνέπεια αυτού του γεγονότος είναι η αύξηση των μέσων αποθεμάτων και συνεπώς του κόστους αποθέματος που σε συνδυασμό με το αυξημένο κόστος εκκρεμών παραγγελιών οδηγεί σε μειωμένο κέρδος σε σχέση με την πολιτική PLS.

Αντίθετα οι τιμές των χωρητικότητων των αποθηκών της LS, είναι χαμηλότερες της πολιτικής PLS. Αυτό έχει σαν αποτέλεσμα τη μείωση του μέσου κόστους αποθέματος που δεν αντισταθμίζεται όμως από τη μειωμένη κερδοφορία λόγω της αυξημένης απόρριψης πελατών.

Στο Σχήμα 3.2 βλέπουμε πως επηρεάζει την κερδοφορία των συγκρινόμενων πολιτικών η μεταβολή του μοναδιαίου κόστους εκκρεμών παραγγελιών. Και εδώ παρατηρούμε ότι η πολιτική PLS αποδίδει καλύτερα σε σύγκριση με τις άλλες πολιτικές. Σε αυτήν την περίπτωση παρατηρείται μία συνεχής μείωση των κερδών όσο αυξάνεται η τιμή του  $b$ . Αυτό είναι πολύ λογικό και συμβαίνει γιατί όσο αυξάνεται το  $b$  και όλες οι άλλες παράμετροι του συστήματος παραμένουν σταθερές, αυξάνεται το συνολικό κόστος μη ικανοποιημένης ζήτησης, άρα μειώνονται τα κέρδη. Όσον αφορά την πολιτική LS, αυτή παραμένει σταθερή αφού δεν έχει λίστα αναμονής άρα και κόστος  $b$ . Επομένως η μεταβολή του δεν επιφέρει καμία αλλαγή στην απόδοση της. Τέλος αναφορικά με την πολιτική CB, είναι φανερό ότι όπως και η PLS μειώνεται σε συνάρτηση με την αύξηση του  $b$  με την διαφορά ότι αποδίδει πολύ μικρότερα κέρδη και ότι η κλίση της είναι πολύ πιο απότομη. Αυτό συμβαίνει γιατί η πολιτική αυτή έχει σαφώς περισσότερες εκκρεμείς

παραγγελίες αφού έχει λίστα αναμονής άπειρης χωρητικότητας, άρα και μεγαλύτερο κόστος μη ικανοποιημένης ζήτησης.

Οι επιπτώσεις της μεταβολής του μοναδιαίου κόστους αποθέματος φαίνονται στο Σχήμα 3.3. Για ακόμη μια φορά η πολιτική PLS αποδίδει μεγαλύτερο κέρδος. Είναι φανερό ότι όλες οι πολιτικές με την σταδιακή αύξηση του  $h$  μειώνουν τα κέρδη τους. Αυτό είναι αναμενόμενο γιατί όσο αυξάνουμε το κόστος αποθεματοποίησης και διατηρούμε τις άλλες παραμέτρους σταθερές θα μειώνεται αναλόγως και το κέρδος.

Στα Σχήματα 3.5 και 3.6 μελετάται η επίδραση της μεταβολής του ρυθμού επισκευών και βλαβών αντίστοιχα στην κερδοφορία του συστήματος. Στο Σχήμα 3.5 παρατηρούμε ότι με την αύξηση του ρυθμού επισκευής των μηχανών αυξάνεται το κέρδος σε όλες τις πολιτικές, με την PLS να αποδίδει περισσότερο. Η ανοδική αυτή πορεία του κέρδους σε συνάρτηση με την αύξηση του  $r$  είναι αναμενόμενη γιατί αυξάνοντας τον ρυθμό επισκευής, αυξάνεται και η παραγωγικότητα του συστήματος, που αυτό συνεπάγεται περισσότερες πωλήσεις και άρα μεγαλύτερο κέρδος.

Αντίστροφη εικόνα παρατηρείται στο Σχήμα 3.6 όπου έχουμε μείωση του αναμενόμενου κέρδους με την αύξηση του ρυθμού βλαβών. Σημειώνεται ότι η απότομη κλίση μείωσης της πολιτικής CB οφείλεται στο γεγονός ότι με αρκετά αυξημένους ρυθμούς βλαβών μειώνεται πολύ η παραγωγικότητα του συστήματος με αποτέλεσμα ο αριθμός των εκκρεμών παραγγελιών και το κόστος αυτών να γίνεται πολύ μεγάλο και τελικά να τείνει στο άπειρο. Αντίθετη εικόνα εμφανίζεται στο Σχήμα 3.4 που παρουσιάζει την εξέλιξη του αναμενόμενου κέρδους συναρτήσει του μοναδιαίου κέρδους  $p$ . Εδώ όπως αναμενόταν παρατηρείται μία συνεχής αύξηση των κερδών σε συνάρτηση με την αύξηση της τιμής κάθε μονάδας προϊόντος. Αυτό είναι αναμενόμενο αφού με τον ίδιο αριθμό πωλήσεων, μιας και όλες οι άλλες παράμετροι παραμένουν σταθερές, θα αποκομίζουμε περισσότερα κέρδη.

Τέλος εξετάζουμε την περίπτωση όπου τα μοναδιαία κόστη αποθεμάτων είναι διαφορετικά για κάθε αποθήκη. Εδώ εξετάζουμε την

περίπτωση όπου  $h_1 = 1$ ,  $h_2 = 2$ ,  $h_3 = 3$ ,  $h_4 = 4$ ,  $h_5 = 5$ .. Στο Σχήμα 3.7, βλέπουμε πως μεταβάλλεται το αναμενόμενο κέρδος κάθε πολιτικής ελέγχου παραγγελιών συναρτήσει της ζήτησης. Παρατηρούμε ότι τα αποτελέσματα είναι πανομοιότυπα με αυτά του Σχήματος 3.1. Από αυτό συμπεραίνουμε ότι είτε τα  $h_i$  είναι ίδια για όλες τις αποθήκες, είτε διαφέρουν, η συμπεριφορά του συστήματος δεν διαφοροποιείται σημαντικά. Η μόνη διαφορά είναι ως προς το μέγεθος του μέγιστου κέρδους. Εδώ είναι μεγαλύτερο από αυτό του Σχήματος 3.1, κάτι που ήταν αναμενόμενο αφού η μέγιστη τιμή του  $h$  σε αυτήν την περίπτωση είναι 5 ενώ στην άλλη περίπτωση είναι πάντα 8. Άρα έχουμε μεγαλύτερο κόστος αποθεματοποίησης και για το λόγο αυτό μικρότερα κέρδη.

Γενικά λοιπόν συμπεραίνουμε ότι η πολιτική PLS, η οποία προτείνεται σε αυτήν την εργασία, σε όλες τις περιπτώσεις μεταβολής των παραμέτρων του συστήματος που εξετάστηκαν αποδίδει τα μεγαλύτερα κέρδη. Σε γενικές γραμμές όσο αυξάνει η συνεισφορά του κόστους εκκρεμών παραγγελιών στο συνολικό κόστος, τόσο η απόδοση της LS συγκλίνει σε αυτή της PLS., ενώ όταν  $b \rightarrow 0$  η πολιτική CB συμπεριφέρεται παρόμοια με την PLS. Επίσης σημαντικό είναι ότι για μικρούς ρυθμούς αφίξεων ( $\lambda \rightarrow 0$ ) η πολιτική CB επιτυγχάνει το ίδιο αναμενόμενο κέρδος με αυτό της PLS ενώ για  $\lambda > 8$  η LS αποδίδει το ίδιο με την προτεινόμενη πολιτική.



## ΚΕΦΑΛΑΙΟ 4

### Συμπεράσματα

Η εργασία αυτή είχε ως στόχο την επίλυση προβλημάτων συνδυασμένου ελέγχου αποθεμάτων και παραγγελιών σε συστήματα παραγωγής προς αποθήκευση. Εξετάστηκαν γραμμές παραγωγής στις οποίες εφαρμόζεται η πολιτική ελέγχου αποθεμάτων KANBAN. Για τον έλεγχο αποθεμάτων και παραγγελιών, προτείναμε την πολιτική μερικής αποδοχής παραγγελιών (PLS), στην οποία δεχόμαστε παραγγελίες σε περιόδους που δεν υπάρχει απόθεμα μέχρι κάποιο μέγιστο όριο και την συγκρίναμε με διαδεδομένες πολιτικές ελέγχου αποδοχής, όπως οι CB και LS. Το πρόβλημα του ελέγχου διατυπώθηκε ως πρόβλημα μεγιστοποίησης του μέτρου απόδοσης του συστήματος συναρτήσει των παραμέτρων ελέγχου.

Για την μέτρηση της απόδοσης του συστήματος χρησιμοποιήσαμε το αναμενόμενο κέρδος, που είναι το κέρδος πωλήσεων μείον το κόστος αποθέματος και το κόστος εκκρεμών παραγγελιών. Για γραμμές παραγωγής με πολλές μηχανές, το πλήθος των δυνατών καταστάσεων είναι τόσο μεγάλο ώστε είναι πρακτικά αδύνατη η ανάλυση του συστήματος. Για το λόγο αυτό χρησιμοποιήσαμε εργαλεία από το χώρο της προσομοίωσης συστημάτων, για τη μελέτη της συμπεριφοράς των εξεταζόμενων συστημάτων. Επίσης χρησιμοποιήσαμε τεχνικές μη γραμμικής βελτιστοποίησης για την εκτίμηση των βέλτιστων τιμών των παραμέτρων ελέγχου.

Πραγματοποιήσαμε αριθμητικά πειράματα για να συγκρίνουμε την απόδοση της προτεινόμενης πολιτικής με τις άλλες πολιτικές ελέγχου που εφαρμόζονται σήμερα σε πολλά συστήματα παραγωγής. Τα αριθμητικά αποτελέσματα τεκμηρίωσαν την υπεροχή της πολιτικής μερικής αποδοχής παραγγελιών (PLS), σε κάθε περίπτωση. Συγκεκριμένα, στις χειρότερες των περιπτώσεων η κερδοφορία της προτεινόμενης πολιτικής ήταν παρόμοια με αυτές των ανταγωνιστικών πολιτικών, ενώ στην πλειοψηφία των περιπτώσεων η απόδοση της ήταν σημαντικά καλύτερη.

Η επιτυχία της πολιτική μερικής αποδοχής παραγγελιών μας ενθαρρύνει να την επεκτείνουμε και να την εφαρμόσουμε σε άλλα συστήματα καθώς, μέχρι σήμερα, είναι ελάχιστες οι εφαρμογές της (Caldentey [8], Kouikoglou and Phillis [9]) και αυτές εξετάζουν πολύ ειδικές κατηγορίες συστημάτων (συστήματα μίας μηχανής που παράγουν ένα προϊόν). Η πολιτική μερικής αποδοχής παραγγελιών μπορεί να συνδυαστεί με άλλες πολιτικές αποθεμάτων όπως οι πολιτικές τύπου αποθέματος βάσης (base stock), CONWIP καθώς και συνδυασμοί των (γενικευμένες πολιτικές KANBAN, οι εκτεταμένες πολιτικές KANBAN, κ.α.). Ενδιαφέρον θα παρουσίαζε ακόμη η εφαρμογή της PLS σε πολύπλοκα συστήματα παραγωγής, όπως τα δίκτυα συναρμολόγησης και/ή αποσυναρμολόγησης. Τέλος θα μπορούσε να επεκταθεί η εφαρμογή της προτεινόμενης πολιτικής σε προβλήματα συνδυασμένου ελέγχου ποιότητας, αποθεμάτων και πωλήσεων σε γραμμές παραγωγής ή δίκτυα συναρμολόγησης.

## Βιβλιογραφία

- [1] S.B.Gershwin, *Manufacturing Systems Engineering* (Prentice Hall, 1994).
- [2] Liberopoulos, G. and Dallery, Y. (2000) A unified framework for pull control mechanisms in multi-stage manufacturing systems. *Annals of Operations Research*, 93, 325-355.
- [3] Buzacott, J.A. and Shanthikumar, J.G. (1993) *Stochastic Models of Manufacturing Systems* Prentice Hall, Englewood Cliffs, NJ.
- [4] Hadley, G. and Whitin, T.M. (1963) *Analysis of Inventory Systems*, Prentice Hall, Englewood Cliffs, NJ.
- [5] Smith, S.A. (1977) Optimal inventories for an (S-1, S) system with no backorders. *Management Science*, 239, 522-528.
- [6] Zipkin, P.H. (2000) *Foundations of Inventory Management*, McGraw-Hill, New York, NY.
- [7] Moinzadeh, K. (1989) Operating characteristics of the (S-1, S) inventory system with partial backorders and constant resupply times. *Management Science*, 35, 472-477.
- [8] Caldentey, R.A. (2001) Analyzing the make-to stock queue in the supply chain and e-business settings. Ph.D. dissertation, Sloan School of Management, MIT, Massachusetts.
- [9] Kouikoglou, V.S. and Phillis, Y.A. (2002) Design of product specifications and control policies in a single-stage production system. *IIE Transactions*, 34, 590-600.
- [10] S. Ioannidis, V.S. Kouikoglou and Y. A. Phillis, (2004) Analysis of admission and inventory control policies for make-to-stock production networks, *Proceedings of the 9th World Automation Congress 2004*, Seville, Spain.
- [11] Β. Κουϊκόγλου, (2003) Προσομοίωση. Σημειώσεις μαθήματος, Τμήμα ΜΠΔ, Πολυτεχνείο Κρήτης, Χανιά.

**ΠΑΡΑΡΤΗΜΑ**  
**(ΚΩΔΙΚΑΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΔΙΠΛΩΜΑΤΙΚΗΣ)**

```

#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<stdlib.h>
#define MAX 5
#define SIZE 100
#define MODLUS 2147483647
#define MULT1 24112
#define MULT2 26143
#define SEED 123
#define SEED1 1234
double rand11();
void simulator();
void main(void)
{
int k,j,i,Enum1,l,EM1[SIZE],control,conv,gsr,flag,maxi,q,rep;
double laf,m[MAX],smr1,f[MAX],r[MAX],jmax,JNEW;
double E1[SIZE],ET1[SIZE],ms[SIZE][SIZE],MDJ[SIZE],VAR[SIZE],MJ[SIZE],MDJPR[SIZE],VARPR[SIZE];
double JC1[SIZE],DJC1[SIZE],tsim1,p1,b1,h1[SIZE];
double T1[SIZE],za2,e,error,deserror,CONST,JPR,GAM,check,tcheck,check1,eps;
long zi1;

FILE *fp,*fd ;

for(q=1;q<9;q++)
{
    if(q==1)
    {
        if((fp=fopen("dedom1.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
        {
            printf("error1\n");
            exit(1);
        }
        fscanf(fp,"%lf",&tsim1);
        fscanf(fp,"%lf",&laf);
        for(i=0;i<MAX;i++)
        {
            fscanf(fp,"%lf",&m[i]);
            fscanf(fp,"%lf",&f[i]);
            fscanf(fp,"%lf",&r[i]);
        }
    }
}

```

```

    }
    for(i=0;i<=MAX+1;i++)
    fscanf(fp,"%lf",&T1[i]);
    fscanf(fp,"%lf",&e);
    fscanf(fp,"%lf",&b1);
    fscanf(fp,"%lf",&p1);

    for(i=1;i<=MAX;i++)
    fscanf(fp,"%lf",&h1[i]);
    fclose(fp);
}
if(q==2)
{
    if((fp=fopen("dedom2.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
    {
        printf("error1\n");
        exit(1);
    }
    fscanf(fp,"%lf",&tsim1);
    fscanf(fp,"%lf",&laf);
    for(i=0;i<MAX;i++)
    {
        fscanf(fp,"%lf",&m[i]);
        fscanf(fp,"%lf",&f[i]);
        fscanf(fp,"%lf",&r[i]);
    }
    for(i=0;i<=MAX+1;i++)
    fscanf(fp,"%lf",&T1[i]);
    fscanf(fp,"%lf",&e);
    fscanf(fp,"%lf",&b1);
    fscanf(fp,"%lf",&p1);
    for(i=1;i<=MAX;i++)
    fscanf(fp,"%lf",&h1[i]);
    fclose(fp);
}
if(q==3)
{
    if((fp=fopen("dedom3.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
    {
        printf("error1\n");

```

```

        exit(1);
    }
    fscanf(fp,"%lf",&tsim1);
    fscanf(fp,"%lf",&laf);
    for(i=0;i<MAX;i++)
    {
        fscanf(fp,"%lf",&m[i]);
        fscanf(fp,"%lf",&f[i]);
        fscanf(fp,"%lf",&r[i]);
    }
    for(i=0;i<=MAX+1;i++)
    fscanf(fp,"%lf",&T1[i]);
    fscanf(fp,"%lf",&e);
    fscanf(fp,"%lf",&b1);
    fscanf(fp,"%lf",&p1);
    for(i=1;i<=MAX;i++)
    fscanf(fp,"%lf",&h1[i]);
    fclose(fp);
}
if(q==4)
{
    if((fp=fopen("dedom4.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
    {
        printf("error1\n");
        exit(1);
    }
    fscanf(fp,"%lf",&tsim1);
    fscanf(fp,"%lf",&laf);
    for(i=0;i<MAX;i++)
    {
        fscanf(fp,"%lf",&m[i]);
        fscanf(fp,"%lf",&f[i]);
        fscanf(fp,"%lf",&r[i]);
    }
    for(i=0;i<=MAX+1;i++)
    fscanf(fp,"%lf",&T1[i]);
    fscanf(fp,"%lf",&e);
    fscanf(fp,"%lf",&b1);
    fscanf(fp,"%lf",&p1);
}

```

```

        for(i=1;i<=MAX;i++)
        fscanf(fp,"%lf",&h1[i]);
        fclose(fp);
    }
    if(q==5)
    {
        if((fp=fopen("dedom5.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
        {
            printf("error1\n");
            exit(1);
        }
        fscanf(fp,"%lf",&tsim1);
        fscanf(fp,"%lf",&laf);
        for(i=0;i<MAX;i++)
        {
            fscanf(fp,"%lf",&m[i]);
            fscanf(fp,"%lf",&f[i]);
            fscanf(fp,"%lf",&r[i]);
        }
        for(i=0;i<=MAX+1;i++)
        fscanf(fp,"%lf",&T1[i]);
        fscanf(fp,"%lf",&e);
        fscanf(fp,"%lf",&b1);
        fscanf(fp,"%lf",&p1);
        for(i=1;i<=MAX;i++)
        fscanf(fp,"%lf",&h1[i]);
        fclose(fp);
    }
    if(q==6)
    {
        if((fp=fopen("dedom6.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
        {
            printf("error1\n");
            exit(1);
        }
        fscanf(fp,"%lf",&tsim1);
        fscanf(fp,"%lf",&laf);
        for(i=0;i<MAX;i++)
        {

```



```

        fscanf(fp,"%lf",&m[i]);
        fscanf(fp,"%lf",&f[i]);
        fscanf(fp,"%lf",&r[i]);
    }
    for(i=0;i<=MAX+1;i++)
        fscanf(fp,"%lf",&T1[i]);
        fscanf(fp,"%lf",&e);
        fscanf(fp,"%lf",&b1);
        fscanf(fp,"%lf",&p1);
    for(i=1;i<=MAX;i++)
        fscanf(fp,"%lf",&h1[i]);
    fclose(fp);
}
if(q==7)
{

        if((fp=fopen("dedom7.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
        {
            printf("error1\n");
            exit(1);
        }
        fscanf(fp,"%lf",&tsim1);
        fscanf(fp,"%lf",&laf);
        for(i=0;i<MAX;i++)
        {
            fscanf(fp,"%lf",&m[i]);
            fscanf(fp,"%lf",&f[i]);
            fscanf(fp,"%lf",&r[i]);
        }
        for(i=0;i<=MAX+1;i++)
            fscanf(fp,"%lf",&T1[i]);
            fscanf(fp,"%lf",&e);
            fscanf(fp,"%lf",&b1);
            fscanf(fp,"%lf",&p1);
        for(i=1;i<=MAX;i++)
            fscanf(fp,"%lf",&h1[i]);
        fclose(fp);
}
if(q==8)
{

```

```

if((fp=fopen("dedom8.txt","r"))==NULL)//Anoigma arxeiou gia diavasma
{
printf("error1\n");
exit(1);
}
fscanf(fp,"%lf",&tsim1);
fscanf(fp,"%lf",&laf);
for(i=0;i<MAX;i++)
{
fscanf(fp,"%lf",&m[i]);
fscanf(fp,"%lf",&f[i]);
fscanf(fp,"%lf",&r[i]);
}
for(i=0;i<=MAX+1;i++)
fscanf(fp,"%lf",&T1[i]);
fscanf(fp,"%lf",&e);
fscanf(fp,"%lf",&b1);
fscanf(fp,"%lf",&p1);
for(i=1;i<=MAX;i++)
fscanf(fp,"%lf",&h1[i]);
fclose(fp);
}
for(i=0;i<MAX;i++)
{
ms[i][0]=m[i];
}
for(i=0;i<MAX;i++)
{
ms[i][1]=f[i];
}
for(i=0;i<MAX;i++)
{
ms[i][2]=r[i];
}
smr1=0;
for(i=0;i<3;i++)
for(j=0;j<MAX;j++)
{
smr1=smr1+ms[j][i];
}

```

```

}
smr1=smr1+laf;
k=0;
for(i=0;i<MAX;i++)
    for(j=0;j<3;j++)
        {
            E1[k]=ms[i][j]/smr1;
            EM1[k]=i;
            ET1[k]=j;
            k=k+1;
        }
E1[k]=laf/smr1;
EM1[k]=MAX;
Enum1=k+1;
za2=1.96;
printf("za2=%f\n",za2);
if((fd=fopen("l=4,h=5-12.txt","a"))==NULL)//Anoigma arxeiou gia grapsimo
{
    printf("error1\n");
    exit(1);
}
fprintf(fd,"gia e=%d\n",q);
for(i=0;i<=MAX+1;i++)//EYRESH BELTISTOY TSIM
{
    JC1[i]=0;
    DJC1[i]=0;
}
zi1=SEED1;
flag=0;
GAM=0;
simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
for(i=0;i<=MAX+1;i++)
    printf("***statistic*JC1[%d]=%f\n,DJC1[%d]=%f",i,JC1[i],i,DJC1[i]);
control=0;
for(i=1;i<=MAX+1;i++)
{
    MDJ[i]=DJC1[i];
    VAR[i]=0;
    MJ[i]=JC1[i];
}

```

```

k=1;
MJ[0]=JC1[0];
while(control!=6 && k<100)
{
    zi1=zi1+100;
    printf("zi1=%ld\n",zi1);
    simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
    control=0;
    MJ[0]=(k*MJ[0]+JC1[0])/(k+1);
    for(l=1;l<=MAX+1;l++)
    {
        MDJPR[l]=MDJ[l];
        MJ[l]=(k*MJ[l]+JC1[l])/(k+1);
        MDJ[l]=(k*MDJPR[l]+DJC1[l])/(k+1);
        VARPR[l]=VAR[l];
        VAR[l]=((k-1)*VARPR[l]+k*(MDJPR[l]*MDJPR[l]+(DJC1[l]*DJC1[l])-
        (k+1)*(MDJ[l]*MDJ[l]))/k;
        error=za2*sqrt(VAR[l]/(k));
        deserror=e*fabs(MDJ[l])/(1+e);
        if(error<deserror)
            control++;
    }
    fprintf(fd,"control=%d\n",control);
    k=k+1;
}

tsim1=k*tsim1;
fprintf(fd,"tsim1=%lf\n",tsim1);
printf("tsim1=%lf\n",tsim1);
//ARXH ALGORI8MOY BELTISTOPOIHSHS
JPR=0;
CONST=5;
conv=0;
k=1;
eps=0.005;
flag=1;
GAM=CONST;
zi1=SEED;
simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
maxi=0;
jmax=-1000000;

```

```

for(l=0;l<=MAX+1;l++)
{
    if(JC1[l]>=jmax)
    {
        jmax=JC1[l];
        maxi=l;
    }
}
fprintf(fd,"maxi=%d\n",maxi);
GAM=CONST*pow(0.618,maxi);           //YPOLOGISMOS BHMATOS
fprintf(fd,"gam=%f\n",GAM);
if(maxi==MAX+1)
CONST=0.618*CONST;
for(i=1;i<=MAX+1;i++)
{
    T1[i]=(int)(T1[i]+GAM*MDJ[i]+0.5); //NEES      XWRHTIKOTHTES      KAI
    AKERAIOPOIHSH
    if(T1[i]<1 && i!=MAX+1)
        T1[i]=1;
    else
        if(i==MAX+1 && T1[i]<0)
            T1[i]=0;
    fprintf(fd,"T1[%d]=%f\n",i,T1[i]);
    printf("T1[%d]=%f\n",i,T1[i]);
}
JPR=MJ[0];
JNEW=jmax;
fprintf(fd,"jpr=%f\n",JPR);
printf("jpr=%f\n",JPR);
GAM=CONST;
rep=0;

while(conv==0)
{
    k=k+1;
    flag=0;
    zi1=SEED;
    simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
    fprintf(fd,"****jc[0]=%f\n",JC1[0]);
    for(l=1;l<MAX+1;l++)

```

```

fprintf(fd,"JC[%d]=%f,\nDJC[%d]=%f\n",I,JC1[I],I,DJC1[I]);
JNEW=JC1[0];
fprintf(fd,"JNEW=%f\n",JNEW);
printf("JNEW=%f\n",JNEW);
printf("JPR=%f\n",JPR);
check=(JNEW-JPR)/JPR;
check1=-100;
for(i=1;i<=MAX+1;i++)
{
    tcheck=DJC1[i]/JNEW;
    if(tcheck>check1)
        check1=tcheck;
    printf("tcheck=%f\n",tcheck);
}
JPR=JNEW;
fprintf(fd,"check=%f\n",check);
printf("check=%f\n",check);
if(check==0)
    rep=rep+1;
if(rep>0)
{
    fprintf(fd,"rep=%d\n",rep);
    if(rep>=2) //SYN8HKH TERMATISMOY
        conv=1;
else
{
    flag=2;
    zi1=SEED;
    simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
    JNEW=JC1[0];
    maxi=0;
    for(i=1;i<=MAX+1;i++)
    {
        if(JC1[i]>JNEW)
        {
            JNEW=JC1[i];
            maxi=i;
        }
    }
}
if(maxi!=0)

```

```

if(DJC1[maxi]>0)
T1[maxi]=T1[maxi]+1;
else
if(T1[maxi]>0)
if(maxi!=MAX+1)
{
    if(T1[maxi]>1)
    T1[maxi]=T1[maxi]-1;
}
else
T1[maxi]=T1[maxi]-1;
}
}
else
{
flag=1;
gsr=0;
GAM=CONST;
while(flag==1)
{
    GAM=CONST*pow(0.618,(gsr*(MAX+1)));    //ΥΠΟΛΟΓΙΣΜΟΣ ΒΗΜΑΤΟΣ
    zi1=SEED;
    fprintf(fd,"1gam=%lf\n",GAM);
    printf("1gam=%lf\n",GAM);
    simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
    for(l=0;l<MAX+1;l++)
    {
        fprintf(fd,"###JC[%d]=%lf,\n",l,JC1[l]);
        printf("###JC[%d]=%lf,\n",l,JC1[l]);
    }
    maxi=0;
    jmax=-1000000000000;
    for(l=0;l<=MAX+1;l++)
    {
        if(JC1[l]>=jmax)
        {
            jmax=JC1[l];
            maxi=l;
        }
    }
}
}

```

```

fprintf(fd,"maxi=%d\n",maxi);
printf("maxi=%d\n",maxi);
printf("jmax=%lf\n",jmax);
GAM=CONST*pow(0.618,(gsr*(MAX+1)+maxi)); //YPOLOGISMOS BHMATOS
fprintf(fd,"2gam=%lf\n",GAM);
printf("2gam=%lf\n",GAM);
if(jmax<JPR)
gsr=gsr+1;
else
flag=0;
}

for(i=0;i<=MAX+1;i++)
{
    T1[i]=(int)(T1[i]+GAM*DJC1[i]+0.5);
    if(T1[i]<1 && i!=MAX+1)
        T1[i]=1;
    else
        if(i==MAX+1 && T1[i]<0)
            T1[i]=0;
    fprintf(fd,"T1[%d]=%lf\n",i,T1[i]);
    printf("T1[%d]=%lf\n",i,T1[i]);
}

}

}

for(i=0;i<=MAX+1;i++)
{
    fprintf(fd,"T1[%d]=%lf\n",i,T1[i]);
}
tsim1=2*tsim1;
flag=2;
zi1=SEED;
simulator(&tsim1,&Enum1,EM1,E1,ET1,T1,&smr1,JC1,DJC1,&flag,&GAM,&zi1,&p1,&b1,h1);
JNEW=JC1[0];
maxi=0;
for(i=1;i<=MAX+1;i++)
{
    if(JC1[i]>JNEW)
    {
        JNEW=JC1[i];
    }
}

```



```

        maxi=i;
    }
}
if(maxi!=0)
if(DJC1[maxi]>0)
T1[maxi]=T1[maxi]+1;
else
if(T1[maxi]>0)
if(maxi!=MAX+1)
{
    if(T1[maxi]>1)
        T1[maxi]=T1[maxi]-1;
}
else
    T1[maxi]=T1[maxi]-1;
for(i=0;i<=MAX+1;i++)
{
    fprintf(fd,"T1[%d]=%lf\n",i,T1[i]);
}
fprintf(fd,"JNEW=%lf\n",JNEW);
fclose(fd);
}}
//*****PROSOMOIWTHS
void simulator(double *tsim,int *Enum,int EM[SIZE],double E[SIZE],double ET[SIZE],double
T[SIZE],double *smr,double JC[SIZE],double DJC[SIZE],int *flag1,double
*GAM1,long *z1,double *p,double *b,double h[SIZE])
{
int k,j,i,EI,I;
double Sump[SIZE];
double u,t,tp;
double minv[SIZE][SIZE],Bck[SIZE],MBck[SIZE],Stat[SIZE][SIZE],inv[SIZE][SIZE],TH[SIZE];
double S[SIZE][SIZE],Sales[SIZE];
int g;
long z11;

t=0;
for(i=0;i<=MAX+1;i++)
    for(j=0;j<MAX+1;j++)
        {
            Stat[i][j]=1;
            minv[i][j]=0;

```

```

    }
for(k=0;k<=MAX+1;k++)
S[k][0]=T[0];
if(*flag1==0)
{
    printf("fLAG=0\n");
    for(k=0;k<=MAX+1;k++)
    {
        for(i=1;i<=MAX+1;i++)
        {
            S[k][i]=T[i];
            if(k!=0 && k==i)
                S[k][i]=S[k][i]+1;
        }
    }
}
if(*flag1==1)
{
    printf("fLAG=1\n");
    for(l=0;l<=MAX+1;l++)
    {
        for(i=1;i<=MAX+1;i++)
        {
            S[l][i]=(int)(T[i]+(*GAM1)*(pow(0.618,l))*DJC[i]+0.5);
            if(S[l][i]<1 && i!=MAX+1)
                S[l][i]=1;
            else
                if(i==MAX+1 && S[l][i]<0)
                    S[l][i]=0;
        }
    }
}
if(*flag1==2)
{
    printf("fLAG=2\n");
    for(l=0;l<=MAX+1;l++)
    {
        for(i=1;i<=MAX+1;i++)
        {
            S[l][i]=T[i];
            if(l!=0 && l==i)

```

```

        {
            if(DJC[i]>0)
                S[i][i]=S[i][i]+1;
            else
                if(S[i][i]>0)
                    if(i!=MAX+1)
                    {
                        if(S[i][i]>1)
                            S[i][i]=S[i][i]-1;
                    }
                    else
                        S[i][i]=S[i][i]-1;
        }
    }
}
for(i=0;i<SIZE;i++)
    Sales[i]=0;
for(i=0;i<=MAX+1;i++)
{
    inv[i][0]=1000;
    for (j=1;j<MAX+1;j++)
    {
        inv[i][j]=1;
    }
}
for(i=0;i<=MAX+1;i++)
{
    MBck[i]=0;
    Bck[i]=0;
}
g=0;
z11=*z1;
while(t<=*tsim)
{
    tp=t;
    u=rand11(&z11);           //GENNESH TYXAIΟΥ ARI8MOY APO GENNHTRIA
    t=t-(log(u)/(*smr));
    for(i=0;i<*Enum;i++)
        Sump[i]=0;
}

```

```

Sump[0]=E[0];
for(i=1;i<*Enum;i++)
{
    Sump[i]=Sump[i-1]+E[i];
}

for(i=0;i<*Enum;i++)          //EYRESH EPOMENOY GEGONOTOS
if(u<Sump[i])
{
    EI=i;
    i=50;
}
for(l=0;l<=MAX+1;l++)
{
    inv[l][0]=1000;
    for(i=1;i<=MAX;i++)
    {
        minv[l][i]=minv[l][i]+((t-tp)*inv[l][i]);
    }
    MBck[l]=MBck[l]+((t-tp)*Bck[l]);
    if(EM[EI]==MAX)
    {
        if(inv[l][MAX]==0)
        {
            if(Bck[l]<S[l][MAX+1])
            {
                Bck[l]=Bck[l]+1;
            }
        }
        else
        {
            Sales[l]=Sales[l]+1;
            inv[l][MAX]=inv[l][MAX]-1;
        }
    }
    else
    {
        if(ET[EI]==0)
        {
            if(Stat[l][EM[EI]]==1)

```

```

{
    if(EM[EI]==MAX-1)
    {
        if(inv[I][MAX-1]>0)
        {
            if(inv[I][MAX]<S[I][MAX])
            {
                if(Bck[I]<=0)
                {
                    inv[I][MAX]=inv[I][MAX]+1;
                    inv[I][MAX-1]=inv[I][MAX-1]-1;
                }
            }
            else
            {
                inv[I][MAX-1]=inv[I][MAX-1]-1;
                Bck[I]=Bck[I]-1;
                Sales[I]=Sales[I]+1;
            }
        }
    }
    else
    {
        k=EM[EI];
        if(inv[I][k]>0)
        {
            if(inv[I][k+1]<S[I][k+1])
            {
                inv[I][k+1]=inv[I][k+1]+1;
                inv[I][k]=inv[I][k]-1;
            }
        }
    }
}
else
{
    if(ET[EI]==1)
    Stat[I][EM[EI]]=0;
    else

```

```

Stat[l][EM[EI]]=1;
    }
}

}

TH[0]=Sales[0]/(*tsim);
printf("th[0]=%lf\n",TH[0]);
JC[0]=(*p)*TH[0];
for(i=1;i<=MAX;i++)
{
    minv[0][i]=minv[0][i]/(*tsim);
    JC[0]=JC[0]-h[i]*minv[0][i];
}
MBck[0]=MBck[0]/(*tsim);
JC[0]=JC[0]-(*b)*MBck[0];
printf("jc[0]=%lf\n",JC[0]);
for(l=1;l<=MAX+1;l++)
{
    TH[l]=Sales[l]/(*tsim);
    printf("th[%d]=%lf\n",l,TH[l]);
    JC[l]=(*p)*TH[l];
    for(i=1;i<MAX+1;i++)
    {
        minv[l][i]=minv[l][i]/(*tsim);
        JC[l]=JC[l]-h[i]*minv[l][i];
    }
    MBck[l]=MBck[l]/(*tsim);
    JC[l]=JC[l]-(*b)*MBck[l];
    if(*flag1==0)
    {
        DJC[l]=JC[l]-JC[0];
        printf("###JC[%d]=%lf,\n###DJC[%d]=%lf\n",l,JC[l],l,DJC[l]);
    }
}
}

double rand11(long *zi) //GENNHTRIA TYXAIWN ARI8MWN
{
    long lowprd, hi31;
    lowprd = (*zi & 65535) * MULT1;

```

```

hi31 = (*zi >> 16) * MULT1 + (lowprd >> 16);
*zi  = ((lowprd & 65535) - MODLUS) +
      ((hi31 & 32767) << 16) + (hi31 >> 15);
if (*zi < 0) *zi += MODLUS;
lowprd = (*zi & 65535) * MULT2;
hi31 = (*zi >> 16) * MULT2 + (lowprd >> 16);
*zi  = ((lowprd & 65535) - MODLUS) +
      ((hi31 & 32767) << 16) + (hi31 >> 15);
if (*zi < 0) (*zi) += MODLUS;
return ((*zi >> 7 | 1) + 1) / 16777216.0;
}

```