

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Μεταπτυχιακή Εργασία

**ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ ΜΕ ΔΥΟ
ΑΝΑΞΙΟΠΙΣΤΕΣ ΜΗΧΑΝΕΣ ΕΝ ΣΕΙΡΑ ΚΑΙ ΧΩΡΟ
ΕΝΔΙΑΜΕΣΗΣ ΑΠΟΘΗΚΕΥΣΗΣ**

υπό

ΚΡΑΪΑ ΙΩΑΝΝΗ

Διπλωματούχου Μηχανικού Παραγωγής & Διοίκησης, 2011

Υπεβλήθη για την εκπλήρωση μέρους των
απαιτήσεων για την απόκτηση του
Μεταπτυχιακού Διπλώματος Ειδίκευσης



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ & ΚΕΝΤΡΟ ΠΛΗΡΟΦΟΡΗΣΗΣ
ΕΙΔΙΚΗ ΣΥΛΛΟΓΗ «ΓΚΡΙΖΑ ΒΙΒΛΙΟΓΡΑΦΙΑ»**

Αριθ. Εισ.: 9445/1
Ημερ. Εισ.: 30-03-2011
Δωρεά: Συγγραφέα
Ταξιθετικός Κωδικός: Δ
658.5
ΚΡΑ

2011

© 2011 Ιωάννης Κράιας

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα Μέλη της Τριμελούς Εξεταστικής Επιτροπής:

Πρώτος Εξεταστής (Επιβλέπων) Δρ. Γεώργιος Λυμπερόπουλος
Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής Δρ. Γεώργιος Κοζανίδης
Λέκτορας, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Τρίτος Εξεταστής Δρ. Δημήτριος Παντελής
Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της μεταπτυχιακής εργασίας μου, Καθηγητή κ. Γεώργιο Λυμπερόπουλο, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της μεταπτυχιακής εργασίας μου, Καθηγητές κκ. Γεώργιο Κοζανίδη, Δημήτριο Παντελή, για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους. Ευχαριστώ τον συνάδελφο και συνεργάτη μου Κολωνιάρη Γεώργιο για την πολύτιμη βοήθειά του στον προγραμματισμό με C. Ευχαριστώ τους φίλους μου για την ηθική υποστήριξή τους. Επίσης, ευχαριστώ τον συνέταιρο μου Βαγγέλη Βούλγαρη για την κατανόησή του, ιδιαίτερα κατά τη διάρκεια των τελευταίων μηνών της προσπάθειάς μου. Πάνω απ' όλα, είμαι ευγνώμων στην οικογένεια μου, στον αδελφό μου Παναγιώτη Κράια και στη μητέρα μου Κωνσταντίνα Κράια Πετρέλη για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτήν την μεταπτυχιακή εργασία στη μνήμη του πατέρα μου Εμμανουήλ Κράια.

Ιωάννης Κράιας

ΠΡΟΣΟΜΟΙΩΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ ΜΕ ΔΥΟ ΑΝΑΞΙΟΠΙΣΤΕΣ ΜΗΧΑΝΕΣ ΕΝ ΣΕΙΡΑ ΚΑΙ ΧΩΡΟ ΕΝΔΙΑΜΕΣΗΣ ΑΠΟΘΗΚΕΥΣΗΣ

ΙΩΑΝΝΗΣ ΚΡΑΪΑΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2011

Επιβλέπων Καθηγητής: Δρ. Γεώργιος Λυμπερόπουλος, Καθηγητής Στοχαστικών
Μεθόδων στη Διοίκηση Παραγωγής

Περίληψη

Το αντικείμενο αυτής της μεταπτυχιακής εργασίας είναι η εκτίμηση της απόδοσης μέσω προσομοίωσης ενός συστήματος παραγωγής που αποτελείται από δυο αναξιόπιστες μηχανές εν σειρά με ενδιάμεσο χώρο αποθήκευσης. Αρχικά, περιγράφονται αναλυτικά όλες οι πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί το σύστημα καθώς και όλα τα πιθανά γεγονότα που μπορούν να αλλάξουν κάθε κατάσταση. Στη συνέχεια, αναπτύσσεται ένας αλγόριθμος προσομοίωσης του συστήματος, ο οποίος κωδικοποιείται στη γλώσσα προγραμματισμού C. Το προκύπτον πρόγραμμα προσομοίωσης χρησιμοποιείται για να διερευνηθεί η επίδραση σημαντικών παραμέτρων του συστήματος, όπως είναι οι μέσοι ρυθμοί επεξεργασίας, βλάβης και επισκευής των μηχανών και η χωρητικότητα του ενδιάμεσου χώρου αποθήκευσης σε σημαντικά μέτρα απόδοσης του συστήματος, όπως είναι η παροχή, η πιθανότητα μπλοκαρίσματος της πρώτης μηχανής και η πιθανότητα «ατροφίας» της δεύτερης μηχανής. Στο τέλος, γίνεται σύγκριση των αποτελεσμάτων και εξαγωγή συμπερασμάτων.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή	1
1.1 Γραμμές παραγωγής	1
1.2 Βασικά χαρακτηριστικά και ιδιότητες πραγματικών γραμμών παραγωγής.....	3
Κεφάλαιο 2 Προσομοίωση συστήματος δύο μηχανών εν σειρά με χώρο ενδιάμεσης αποθήκευσης.....	7
2.1 Περιγραφή του συστήματος.....	7
2.2 Γεγονότα και καταστάσεις συστήματος	7
2.3 Προσομοίωση συστήματος.....	13
2.4 Μεγέθη που υπολογίζονται με την προσομοίωση του συστήματος.....	16
Κεφάλαιο 3 Αριθμητικά αποτελέσματα προσομοίωσης	19
3.1 Αρχικοποίηση του συστήματος	19
3.1.1 Έναρξη αλγορίθμου και προσομοίωσης.....	19
3.2 Παρουσίαση πειραματικών ελέγχων	19
3.2.1 Πρώτος πειραματικός έλεγχος.....	19
3.2.2 Δεύτερος πειραματικός έλεγχος.....	22
3.2.3 Τρίτος πειραματικός έλεγχος.....	23
3.2.4 Τέταρτος πειραματικός έλεγχος.....	25
3.2.5 Πέμπτος πειραματικός έλεγχος.....	27
3.2.6 Έκτος πειραματικός έλεγχος.....	29
3.2.7 Έβδομος πειραματικός έλεγχος	33
Κεφάλαιο 4 Συμπεράσματα	36
Βιβλιογραφία	37

Κατάλογος πινάκων

Πίνακας 3-1: Μέσος όρος για $P1=P2=10, F1=F2=0.1, R1=R2=2$	20
Πίνακας 3-2: Τυπική απόκλιση για $P1=P2=10, F1=F2=0.1, R1=R2=2$	21
Πίνακας 3-3: Διάστημα εμπιστοσύνης για $P1=P2=10, F1=F2=0.1, R1=R2=2$	21
Πίνακας 3-4: Μέσος όρος για $P1=P2=10, F1=F2=0.1, N=10$	22
Πίνακας 3-5: Τυπική απόκλιση για $P1=P2=10, F1=F2=0.1, N=10$	23
Πίνακας 3-6: Διάστημα εμπιστοσύνης για $P1=P2=10, F1=F2=0.1, N=10$	23
Πίνακας 3-7: Μέσος όρος για $P1=P2=10, R1=R2=2, N=10$	24
Πίνακας 3-8: Τυπική απόκλιση για $P1=P2=10, R1=R2=2, N=10$	25
Πίνακας 3-9: Διάστημα εμπιστοσύνης για $P1=P2=10, R1=R2=2, N=10$	25
Πίνακας 3-10: Μέσος όρος για $P1=P2=10, N=10$	26
Πίνακας 3-11: Τυπική απόκλιση για $P1=P2=10, N=10$	27
Πίνακας 3-12: Διάστημα εμπιστοσύνης για $P1=P2=10, N=10$	27
Πίνακας 3-13: Μέσος όρος για $P1=10, F1=F2=0.1, R1=R2=2, N=10$	28
Πίνακας 3-14: Τυπική απόκλιση για $P1=10, F1=F2=0.1, R1=R2=2, N=10$	29
Πίνακας 3-15: Διάστημα εμπιστοσύνης για $P1=10, F1=F2=0.1, R1=R2=2, N=10$	29
Πίνακας 3-16: Μέσος όρος για $P1=P2$ με $a=0.05$ και $b=0.15, F1=F2=0,1, R1=R2=2$	31
Πίνακας 3-17: Τυπική απόκλιση για $P1=P2$ με $a=0.05$ και $b=0.15, F1=F2=0,1,$ $R1=R2=2$	32
Πίνακας 3-18: Διάστημα εμπιστοσύνης για $P1=P2$ με $a=0.05$ και $b=0.15,$ $F1=F2=0,1, R1=R2=2$	32
Πίνακας 3-19: Μέσος όρος για $P1=P2=10, F1=F2$ με $a=5$ και $b=15, R1=R2$ με $a=0,16$ και $b=0.83$	33
Πίνακας 3-20: Τυπική απόκλιση για $P1=P2=10, F1=F2$ με $a=5$ & $b=15, R1=R2$ με $a=0,16$ & $b=0.83$	35
Πίνακας 3-21: Διάστημα εμπιστοσύνης $P1=P2=10, F1=F2$ με $a=5$ & $b=15, R1=R2$ με $a=0,1$ & $b=0.83$	35

Κατάλογος σχημάτων

Σχήμα 1-1: Διάγραμμα συστήματος παραγωγής.....	1
Σχήμα 2-1 :Κατάσταση $A=3, 0 \leq n \leq N, b=0, s=0$	9
Σχήμα 2-2:Κατάσταση $A=1, 0 \leq n \leq N, b=0, s=0$	10
Σχήμα 2-3:Κατάσταση $A=2, 0 \leq n \leq N, b=0, s=0$	10
Σχήμα 2-4:Κατάσταση $A=0, 0 \leq n \leq N, b=0, s=0$	10
Σχήμα 2-5:Κατάσταση $A=3, n=N, b=1, s=0$	11
Σχήμα 2-6:Κατάσταση $A=1, 0n=N, b=1, s=0$	11
Σχήμα 2-7:Κατάσταση $A=2, n=N, b=1, s=0$	11
Σχήμα 2-8:Κατάσταση $A=0, n=N, b=1, s=0$	12
Σχήμα 2-9:Κατάσταση $A=3, n=0, b=0, s=1$	12
Σχήμα 2-10:Κατάσταση $A=1, n=0, b=0, s=1$	12
Σχήμα 2-11:Κατάσταση $A=2, n=0, b=0, s=1$	13
Σχήμα 2-12:Κατάσταση $A=0, n=0, b=0, s=1$	13
Σχήμα 3-1: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer	20
Σχήμα 3-2 Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer	21
Σχήμα 3-3: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών επιδιόρθωσης	22
Σχήμα 3-4: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών επιδιόρθωσης	23
Σχήμα 3-5: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών βλαβών.....	24
Σχήμα 3-6: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών βλαβών.....	25
Σχήμα 3-7: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών βλαβών και επιδιόρθωσης	26

Σχήμα 3-8: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών βλαβών και επιδιόρθωσης	27
Σχήμα 3-9: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει του ρυθμού παραγωγής της M2	28
Σχήμα 3-10: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει του ρυθμού παραγωγής της M2	29
Σχήμα 3-11: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer	31
Σχήμα 3-12: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer	32
Σχήμα 3-13: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer	34
Σχήμα 3-14: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer	34

Κεφάλαιο 1 Εισαγωγή

1.1 Γραμμές παραγωγής

Τα συστήματα γραμμών παραγωγής συνεχόμενης ροής προϊόντων αποτελούνται από σταθμούς επεξεργασίας και χώρους ενδιάμεσης αποθήκευσης. Τα προϊόντα προς επεξεργασία ρέουν από έναν σταθμό επεξεργασίας στον χώρο ενδιάμεσης αποθήκευσης που έπεται και στη συνέχεια στον επόμενο σταθμό επεξεργασίας. Στο μοντέλο που θα εξεταστεί και περιγραφεί, τα προϊόντα προς επεξεργασία περνάνε μόνο μια φορά από κάθε σταθμό επεξεργασίας και χώρο αποθήκευσης με καθορισμένη σειρά. Στη γενική μορφή, υπάρχει ένας αρχικός σταθμός επεξεργασίας στον οποίο αρχικά εισέρχονται τα προϊόντα και ένας τελικός σταθμός επεξεργασίας από τον οποίο τα προϊόντα εξέρχονται του συστήματος. Οι χρόνοι επεξεργασίας των προϊόντων στον εκάστοτε σταθμό επεξεργασίας μπορεί να θεωρηθούν τυχαίοι, επειδή κατά την λειτουργία ενός τέτοιου συστήματος υπάρχουν παράγοντες που τους επηρεάζουν και που δεν μπορεί να προβλεφτούν, όπως είναι διαφορές στην ποιότητα των υλικών, την ταχύτητα εργασίας των εργαζομένων, την διαθεσιμότητα εργαλείων και αναλωσίμων υλικών, απρόβλεπτες βλάβες που μπορεί να προκύψουν στην λειτουργία των σταθμών επεξεργασίας καθώς και στους χρόνους επιδιόρθωσης αυτών. Οι ενδιάμεσοι αποθηκευτικοί χώροι έχουν συγκεκριμένη χωρητικότητα αποθήκευσης προϊόντων. Είθισται οι χώροι επεξεργασίας να αποκαλούνται μηχανές. Υποθέτουμε ότι κάθε μηχανή δεν μπορεί να επεξεργασθεί ένα καινούργιο προϊόν εφόσον δεν έχει ολοκληρώσει το προϊόν που ήδη επεξεργάζεται. Επίσης, υποθέτουμε ότι κάθε προϊόν προς επεξεργασία ακολουθεί με την σειρά τις μηχανές στις οποίες πρόκειται να επεξεργαστεί καθώς και τους ενδιάμεσους αποθηκευτικούς χώρους.

Μεγάλη χρησιμότητα των εν λόγω γραμμών παραγωγής παρατηρείται στις περιπτώσεις όπου απαιτείται μεγάλη και μαζική παραγωγή. Παρατίθεται διάγραμμα συστήματος γραμμής παραγωγής το οποίο αποτελείται από πέντε μηχανές και τέσσερις αποθηκευτικούς χώρους.



Σχήμα 1-1: Διάγραμμα συστήματος παραγωγής

Οι κύκλοι συμβολίζουν τις μηχανές και τα τετράγωνα τους ενδιάμεσους αποθηκευτικούς χώρους. Οι μηχανές αριθμούνται από 1 έως k όπου k είναι ο συνολικός αριθμός μηχανών στο σύστημα. Σε ένα τέτοιο σύστημα, υπάρχουν $k - 1$ αποθηκευτικοί χώροι και συμβολίζονται με $B_{i,i+1}$, όπου i η μηχανή πριν τον

αποθηκευτικό χώρο και $i + 1$ η επομένη. Έτσι, τα προϊόντα προς επεξεργασία εισέρχονται στην μηχανή M1, στη συνέχεια στον αποθηκευτικό χώρο $B_{1,2}$, στην μηχανή M2 κ.ο.κ. μέχρι την μηχανή M_k , από όπου και εξέρχονται του συστήματος. Σε κάθε έξοδό τους από μια μηχανή, αν η επόμενη μηχανή είναι διαθέσιμη, τότε η παραμονή τους στον χώρο ενδιάμεσης αποθήκευσης είναι ακαριαία. Πρακτικά, αυτό μπορεί να σημαίνει ότι δεν περνούν καν από τον χώρο ενδιάμεσης αποθήκευσης.

Είναι προφανές ότι σε ένα τέτοιο σύστημα παίζει μεγάλο ρόλο ο χρόνος των βλαβών και ο χρόνος των επιδιορθώσεων για αυτό και είναι πολύ σημαντικό να προσδιορίζεται ο τρόπος της τυχαιότητας των χρόνων αυτών. Τα μέτρα απόδοσης τα όποια προσδιορίζονται και επηρεάζονται άμεσα από την αλλαγή ή μετατροπή των χρόνων αυτών, είναι η παροχή του συστήματος, η πιθανότητες μπλοκαρίσματος και ατροφίας των μηχανών, καθώς επίσης και ο μέσος όρος του επιπέδου του ενδιάμεσου αποθηκευτικού χώρου.

Όταν μια μηχανή τελειώνει την επεξεργασία ενός προϊόντος δέχεται το επόμενο είτε από την προηγούμενη μηχανή απευθείας, αν συμπίπτουν οι χρόνοι ολοκλήρωσης, είτε από τον προηγούμενο αποθηκευτικό ενδιάμεσο χώρο. Αυτό έχει ως αποτέλεσμα την μείωση του επιπέδου του αποθηκευτικού χώρου. Από την άλλη πλευρά, όταν μια μηχανή υποστεί βλάβη ή ο χρόνος επεξεργασίας είναι αρκετά μεγάλος τότε το επίπεδο του αποθηκευτικού χώρου που βρίσκεται πριν από αυτή τη μηχανή αυξάνεται. Αντίθετα, το επίπεδο του αποθηκευτικού χώρου μετά την μηχανή αυτή πέφτει. Συνεπώς όταν ο αποθηκευτικός χώρος πριν την μηχανή που έπαθε βλάβη γεμίσει τότε λέμε ότι η μηχανή που είναι πριν από την χαλασμένη μηχανή είναι μπλοκαρισμένη μιας και δεν μπορεί να αδειάσει το προϊόν που μόλις επεξεργάστηκε και να ξεκινήσει άλλο καινούργιο προϊόν. Επίσης όταν ο αποθηκευτικός χώρος μετά την μηχανή αδειάσει τότε η μηχανή που είναι μετά την χαλασμένη μηχανή δεν θα μπορεί πλέον να δουλέψει, επειδή δεν υπάρχει προϊόν διαθέσιμο προς επεξεργασία. Σε αυτήν την περίπτωση λέμε ότι η μηχανή αυτή πεινάει ή βρίσκεται σε ατροφία. Είναι εύκολα αντιληπτό ότι τέτοιες μακροχρόνιες βλάβες ή ακόμα και μεγάλοι χρόνοι επιδιόρθωσης επηρεάζουν άμεσα την απόδοση του συστήματος. Δηλαδή των συνολικό αριθμό τελικών παραγόμενων προϊόντων στο σύνολο όλης της γραμμής παραγωγής. Επομένως ο ρυθμός παραγωγής του συστήματος ή παροχή περιορίζεται από δυο βασικούς παράγοντες

- Ο ρυθμός παραγωγής του συστήματος (παροχή) δεν μπορεί να είναι μεγαλύτερος από το μικρότερο ρυθμό επεξεργασίας οποιασδήποτε μηχανής.

- Όταν οι ρυθμοί επεξεργασίας είναι αρκετά διαφοροποιημένοι ή όταν συμβαίνουν βλάβες, ιδιαίτερα με μεγάλους χρόνους επιδιόρθωσης, είναι συχνά τα φαινόμενα μπλοκαρίσματος και ατροφίας σε κάποιες μηχανές.

Είναι λοιπόν σαφές ότι εάν είναι εφικτό οι χρόνοι επεξεργασίας όλων των μηχανών να είναι σωστά συγχρονισμένοι καθώς επίσης και οι ρυθμοί των βλαβών και των επιδιορθώσεων τότε αυτό θα μειώσει τις περιπτώσεις των φαινομένων μπλοκαρίσματος και ατροφίας με άμεσο αποτέλεσμα την αύξηση της παραγωγικότητας της γραμμής παραγωγής.

Για το σχεδιασμό τέτοιων συστημάτων παραγωγής και την μελέτη αυτών με στόχο την όσο δυνατόν καλύτερη και αξιόπιστη εφαρμογή τους, απαιτείται η δημιουργία και ανάλυση μοντέλων που να ανταποκρίνονται στις πραγματικές καταστάσεις στις οποίες μπορούν να βρεθούν. Κατά την λειτουργία τέτοιων συστημάτων, ένα σύνολο από πιθανές καταστάσεις και ενδεχόμενα είναι υπαρκτά και είναι πιθανόν να συμβούν. Για αυτό το λόγο θα πρέπει πάντα να είναι προσεκτική η επιλογή των οποιοδήποτε μαθηματικών εργαλείων ή μοντελοποιήσεων ώστε ο κάθε πειραματικός έλεγχος να ανταποκρίνεται στις πραγματικές καταστάσεις. Για παράδειγμα είναι τελείως διαφορετικό ένα σύστημα που επεξεργάζεται το ίδιο προϊόν και απλά υπάρχει μια τυχαιότητα στους χρόνους επεξεργασίας και τελείως διαφορετικό η επεξεργασία διαφορετικής φύσης προϊόντων.

Η προσπάθεια μελέτης τέτοιων συστημάτων αναξιόπιστων ως προς την ποικιλία των παραγόντων που μπορούν να τα επηρεάσουν, οδηγεί στην εφαρμογή μοντέλων με ιδανικές συνθήκες και εκτιμήσεις ενσωματώνοντας όμως την τυχαιότητα όλων αυτών των παραγόντων.

1.2 Βασικά χαρακτηριστικά και ιδιότητες πραγματικών γραμμών παραγωγής

Οι γραμμές παραγωγής εκτός από ακαδημαϊκό ενδιαφέρον παρουσιάζουν και μεγάλο οικονομικό ενδιαφέρον μιας και αποτελούν εργαλείο για την μαζική αλλά και ποιοτική παραγωγή μεγάλης γκάμας προϊόντων. Μεγάλο τμήμα των οικονομιών των κρατών παγκοσμίως στηρίζεται σε αυτές. Οπότε η αξία τους είναι πολυσήμαντη και για αυτό ο άνθρωπος επιχειρεί την συνεχή εξέλιξη τους, οργάνωσής τους, βελτιστοποίησή τους, με σκοπό την όσο δυνατόν καλύτερη αξιοποίησή τους.

Τα περισσότερα συστήματα σε πραγματικές συνθήκες είναι ασυγχρόνιστα. Και αυτό διότι οι μηχανές του συστήματος δεν είναι περιορισμένες στο να ξεκινάνε και να τελειώνουν την λειτουργία τους συγχρονισμένα. Ακόμα και στα συστήματα όπου οι χρόνοι λειτουργίας είναι κοντρολαρισμένοι και έχουν ίδιους χρόνους κύκλου η

παρουσία των χωρών αποθήκευσης τους δίνει τη δυνατότητα να ξεκινάνε και τελειώνουν την λειτουργία τους σε μεγάλο βαθμό ανεξάρτητα. Εν τέλει και η ύπαρξη προβλέπτων και ακαθόριστων βλαβών των μηχανών καθώς και των χρόνων επιδιόρθωσης δεν μπορούν να καταστήσουν χρόνους λειτουργίας των μηχανών συγχρονισμένους.

Έτσι λόγω της αβεβαιότητας των χρόνων, τα ασυγχρόνιστα μοντέλα τέτοιων συστημάτων παραγωγής δεν μπορούν να χειριστούν με ντετερμινιστικά μαθηματικά μοντέλα. Ο μη προσδιορισμός ακριβών και μη σταθεροποιημένων χρόνων καθιστά το χειρισμό τους εξαιρετικά δύσκολο.

Για την εύρεση ακριβών λύσεων λοιπόν, έχουν μελετηθεί και δημιουργηθεί αναλυτικές μαθηματικές μέθοδοι. Όταν όμως τα συστήματα αυτά γίνονται μεγάλα και οι παράμετροι παρά πολλοί δεν είναι δυνατόν να επιλυθούν από τις αναλυτικές μεθόδους. Για το λόγο αυτό καταφεύγουμε σε προσεγγιστικά μαθηματικά μοντέλα ή προτιμάται η προσομοίωση τους για μεγάλο χρονικό διάστημα με προσεκτική πάντα επιλογή των μαθηματικών εργαλείων που χρησιμοποιούνται.

Οι πρώτες ύλες που απαιτούνται για την κατασκευή των τελικών προϊόντων φτάνουν και είναι σε διαθεσιμότητα με διαφόρους τρόπους. Καθώς επίσης και μετά το τέλος της συνολικής επεξεργασία τους. Με βάση λοιπόν τον τρόπο που τροφοδοτείται η γραμμή παραγωγής και τον τρόπο με τον οποίο απομακρύνονται τα προϊόντα υπάρχουν δυο βασικές κατηγορίες. Στη μια κατηγορία υποθέτετε ότι το σύστημα συνεχώς τροφοδοτείται ενώ υπάρχει πάντα αρκετός χώρος αποθήκευσης μετά το τέλος της συνολικής επεξεργασίας τους. Σε αυτή την κατηγορία η πρώτη μηχανή δεν υπάρχει περίπτωση να πεινάσει ενώ η τελευταία μηχανή δεν υπάρχει περίπτωση να μπλοκαριστεί. Στη δεύτερη κατηγορία υπάρχουν συνήθως αποθηκευτικοί χώροι στην αρχή και στο τέλος της γραμμής παραγωγής οπότε εισέρχονται και νέα δεδομένα στο σύστημα καθώς πλέον οι ρυθμοί που τροφοδοτείται και αδειάζει το σύστημα είναι συγκεκριμένοι. Στη παρούσα θα γίνει έλεγχος για την πρώτη κατηγορία.

Έχουν αναφερθεί τα φαινόμενα πείνας και μπλοκαρίσματος. Επίσης ότι οι αποθηκευτικοί ενδιάμεσοι χώροι είναι ανάμεσα από δυο μηχανές. Μια μηχανή πεινάει όταν είναι σε λειτουργία και ο αποθηκευτικός χώρος πριν από αυτήν είναι άδειος με αποτέλεσμα ενώ μπορεί να επεξεργαστεί αυτό να μην γίνεται. Μια μηχανή μπλοκάρεται όταν, ενώ είναι σε λειτουργία και έχει ολοκληρώσει ένα προϊόν δεν μπορεί να το διώξει και να δεχθεί ένα επόμενο διότι ο χώρος αποθήκευσης μετά την μηχανή αυτή είναι γεμάτος. Επομένως είναι προφανές ότι όσο μεγαλύτερος είναι αποθηκευτικός χώρος τόσο αυξάνεται ο ρυθμός απόδοσης του συστήματος.

Επιπλέον οι γραμμές παραγωγής χωρίζονται και σε δυο επιπλέον κατηγορίες. Στις γραμμές παραγωγής με αξιόπιστες μηχανές. Και σε γραμμές παραγωγής με αναξιόπιστες μηχανές. Πολλές φορές διάφοροι παράγοντες επιδρούν και απλά συμβαίνουν και οι μηχανές των γραμμών παραγωγής μπορούν να πάθουν βλάβες. Μια ποικιλία υποθέσεων μπορεί να γίνει κάτω από ποιες συνθήκες συμβαίνουν οι βλάβες σε μια μηχανή, πόσο χρόνο διαρκούν, την τυχαιότητα των χρόνων αυτών καθώς και πόσος χρόνος θα περάσει μέχρι την επομένη βλάβη. Μια γραμμή παραγωγή με τέτοιες μηχανές ονομάζεται γραμμή παραγωγής με αναξιόπιστες μηχανές. Μια γραμμή παραγωγής όπου οι μηχανές δεν παθαίνουν βλάβες ονομάζεται γραμμή παραγωγής με αξιόπιστες μηχανές.

Έτσι μεγάλο ενδιαφέρον αποτελεί στις γραμμές παραγωγής με αναξιόπιστες μηχανές πόσο επηρεάζεται το σύστημα από τις βλάβες, πόσο τυχαίες ή όχι είναι οι βλάβες, αν ακολουθούν κάποια κατανομή και ποια είναι αυτή. Έτσι περά από την τυχαιότητα των χρόνων επεξεργασίας το σύστημα επηρεάζεται και από την τυχαιότητα των ρυθμών και την τυχαιότητα των βλαβών. Ενώ στις γραμμές παραγωγής με αξιόπιστες μηχανές η τυχαιότητα τους συστήματος εξαρτάται μονό από την τυχαιότητα των ρυθμών επεξεργασίας. Συνήθως βέβαια οι ρυθμοί βλαβών ακολουθούν εκθετική κατανομή στις περισσότερες περιπτώσεις αλλά στη παρούσα θα μελετηθεί και άλλο είδος κατανομής.

Υπάρχουν δυο είδη ροών στις γραμμές παραγωγής. Οι γραμμές παραγωγής με διακριτό τύπο προϊόντων προς εξεργασία και η συνεχής ροη. Στην πρώτη περίπτωση συγκαταλέγονται διακριτοί τύποι προϊόντων τα όποια μεμονωμένα μπαίνουν στην γραμμή παραγωγής προς επεξεργασία ενώ στην δεύτερη περίπτωση συγκαταλέγονται προϊόντα συνεχούς ροής. Τα συστήματα που έχουν συνεχή ροή πολλές φορές για να μοντελοποιηθούν γίνεται κατηγοριοποίηση τους σε διακριτά μεγέθη. Και στις δυο περιπτώσεις οι μηχανές μπορούν πάθουν βλάβη και να επηρεαστούν οι ρυθμοί παραγωγής τους συστήματος.

Έχει μεγάλο ενδιαφέρον από οικονομικής κυρίως άποψης όταν γίνονται προσπάθειες βελτιώσεις των ρυθμών απόδοσης μιας γραμμής παραγωγής. Στις πραγματικές γραμμές παραγωγής οι χρόνοι επεξεργασίας των μηχανών δεν διαφέρουν πολύ και αυτό για το λόγο που αναφέρθηκε ότι ο ρυθμός παραγωγής δεν μπορεί να είναι μεγαλύτερος από το μικρότερο ρυθμό παραγωγής μιας μηχανής τους συστήματος. Είναι προφανές ότι οποιαδήποτε επένδυση για την αντικατάσταση μιας μηχανής με σκοπό την αύξηση του ρυθμού παραγωγής της αν δε γίνει με προσεκτική μελέτη όλου του συστήματος συνολικά ενδεχομένως να αποδειχθεί ασύμφορη. Και αυτό γιατί η βελτίωση θα είναι ελάχιστη έως μηδαμινή αν γίνουν προσπάθειες για βελτίωση

ρυθμών παραγωγής μηχανών που έχουν αρκετά μεγαλύτερο χρόνο από την μηχανή με το χαμηλότερο ρυθμό παραγωγής.

Η σύνηθες πολιτική διαχείρισης τέτοιων γραμμών παραγωγής είναι η προσπάθεια αύξησης του ρυθμού παραγωγής με τον περιορισμό των καταστάσεων μπλοκαρίσματος και πείνας και αυτό επιτυγχάνεται με τον κατάλληλο προσδιορισμό της χωρητικότητας του ενδιάμεσου χώρου αποθήκευσης.

Κεφάλαιο 2 Προσομοίωση συστήματος δύο μηχανών εν σειρά με χώρο ενδιάμεσης αποθήκευσης

2.1 Περιγραφή του συστήματος

Το σύστημα που θα μελετηθεί και θα προσομοιωθεί αποτελείται από δυο μηχανές σε σειρά. Ανάμεσα στις δυο μηχανές υπάρχει προσωρινός χώρος ενδιάμεσης αποθήκευσης για τα προϊόντα που έχουν τελειώσει την επεξεργασία τους στην πρώτη μηχανή.

Κάθε μηχανή έχει την δυνατότητα να επεξεργάζεται μόνο ένα προϊόν την φορά. Η πρώτη μηχανή μπορεί να τροφοδοτείται συνέχεια με νέα προϊόντα προς επεξεργασία χωρίς κάποιο περιορισμό ενώ ο ενδιάμεσος χώρος αποθήκευσης έχει συγκεκριμένη χωρητικότητα ανάλογα με την επιθυμία του χρήστη.

Καθ' όλη τη διάρκεια λειτουργίας ενός τέτοιου συστήματος, οι χρόνοι επεξεργασίας των προϊόντων σε κάθε μηχανή ακολουθούν κάποια συγκεκριμένη κατανομή με τις αντίστοιχες παραμέτρους της.

Κατά την λειτουργία του συστήματος, μπορεί να συμβεί βλάβη σε οποιαδήποτε μηχανή. Οι χρόνοι μέχρι να συμβεί βλάβη σε κάθε μηχανή ακολουθούν κάποια κατανομή με τις αντίστοιχες παραμέτρους της.

Όταν μια μηχανή πάθει βλάβη, ξεκινά η διαδικασία επιδιόρθωσής της. Οι χρόνοι επιδιόρθωσης κάθε μηχανής ακολουθούν και αυτοί με τη σειρά τους κάποια συγκεκριμένη κατανομή με τις παραμέτρους τους.

Τα κομμάτια ή προϊόντα που είναι προς επεξεργασία πρώτα εισέρχονται στην πρώτη μηχανή και εν συνέχεια για το πέρας της επεξεργασίας τους εισέρχονται στην δεύτερη μηχανή. Σε περίπτωση όπου ένα προϊόν εξέλθει από την πρώτη μηχανή ενώ η δεύτερη ήδη έχει κάποιο προϊόν προς επεξεργασία τότε αποθηκεύεται προσωρινά στον ενδιάμεσο αποθηκευτικό χώρο και περιμένει με την σειρά του να εισέλθει στην δεύτερη μηχανή.

2.2 Γεγονότα και καταστάσεις συστήματος

Υπάρχουν δυο μέθοδοι προσομοίωσης. Αυτές όπου η κατάσταση του συστήματος παρατηρείται σε τακτές χρονικές περιόδους (προσομοίωση βάσει του χρόνου) και αυτές όπου η κατάσταση παρατηρείται κάθε φορά που συμβαίνει ένα γεγονός που αλλάζει την κατάσταση (προσομοίωση βάσει των γεγονότων). Στην παρούσα θα γίνει

προσομοίωση βάσει των γεγονότων. Στο συγκεκριμένο σύστημα που αναλύεται θεωρούνται ως πιθανά γεγονότα τα εξής:

Γεγονότα (events)

- *Event 1*: η πρώτη μηχανή τελειώνει την επεξεργασία ενός προϊόντος
- *Event 2*: η δεύτερη μηχανή τελειώνει την επεξεργασία ενός προϊόντος
- *Event 3*: η πρώτη μηχανή παθαίνει βλάβη
- *Event 4*: η δεύτερη μηχανή παθαίνει βλάβη
- *Event 5*: η πρώτη μηχανή επισκευάζεται
- *Event 6*: η δεύτερη μηχανή επισκευάζεται

Ως κατάσταση συστήματος μια ορισμένη χρονική στιγμή, θεωρείται η κατάσταση στην οποία βρίσκονται οι βασικές παράμετροι τους συστήματος. Δηλαδή σε τι κατάσταση βρίσκονται οι μηχανές, το επίπεδο του αποθηκευτικού χώρου καθώς επίσης και κάποιες ιδιαίτερες καταστάσεις οι οποίες θα προσδιοριστούν στη συνέχεια.

Οι βασικές καταστάσεις των μηχανών προσδιορίζονται με το αν οι μηχανές βρίσκονται η όχι σε λειτουργία.

Η κατάσταση του αποθηκευτικού χώρου προσδιορίζεται από το επίπεδο στο οποίο βρίσκεται σε σχέση με την χωρητικότητα του.

Επίσης υφίστανται δυο ιδιάζουσες καταστάσεις κατά τις οποίες συμβαίνει το εξής. Σε περίπτωση όπου η δεύτερη μηχανή είναι σε λειτουργία χωρίς όμως να επεξεργάζεται κάποιο προϊόν, ο αποθηκευτικός χώρος είναι άδειος και η πρώτη μηχανή είναι χαλασμένη ή δεν έχει τελειώσει την επεξεργασία της τότε λέμε ότι η δεύτερη μηχανή πεινάει ή είναι σε ατροφία.

Στη δεύτερη ιδιάζουσα κατάσταση συμβαίνει το εξής. Όταν η πρώτη μηχανή είναι σε λειτουργία και τελειώσει την επεξεργασία της και ο αποθηκευτικός χώρος είναι γεμάτος, τότε λέμε ότι η πρώτη μηχανή είναι μπλοκαρισμένη.

Χρονική στιγμή για τη συγκεκριμένη προσομοίωση που αναλύεται όπως αναφέρθηκε παραπάνω, θεωρείται η χρονική στιγμή που πραγματοποιείται ένα γεγονός (*event*).

Έτσι κατά την πραγματοποίηση ενός γεγονότος κάποιες από τις καταστάσεις των βασικών παραμέτρων αλλάζουν οπότε αλλάζει και η συνολική κατάσταση του συστήματος.

Ως εκ τούτου διαμορφώνεται ένα γενικό σύνολο καταστάσεων του συστήματος όπου προσδιορίζονται από τις επιμέρους καταστάσεις των παραμέτρων που αναφέρθηκαν

παραπάνω και η μετάβαση από κατάσταση σε κατάσταση γίνεται με την πραγματοποίηση κάποιου γεγονότος.

Προς διευκόλυνση του προσδιορισμού όλων των πιθανών καταστάσεων του συστήματος γίνεται ο ακόλουθος συμβολισμός ο οποίος θα διευκολύνει εκτός από τον προσδιορισμό των καταστάσεων και την προσομοίωση του συστήματος στη συνέχεια.

Με A συμβολίζεται η κατάσταση λειτουργίας των μηχανών, όπου το A παίρνει τιμές από 0 έως 3 ως εξής:

- $A = 0$: και οι δυο μηχανές είναι εκτός λειτουργίας (χαλασμένες)
- $A = 1$: η πρώτη μηχανή είναι εκτός λειτουργίας και η δεύτερη είναι σε λειτουργία
- $A = 2$: η πρώτη μηχανή είναι σε λειτουργία και η δεύτερη είναι εκτός λειτουργίας
- $A = 3$: και οι δυο μηχανές είναι σε λειτουργία

Με n συμβολίζεται το επίπεδο του αποθηκευτικού χώρου, όπου το n παίρνει τις τιμές $n = 0, \dots, N$, όπου N το μέγιστο επιτρεπόμενο επίπεδο του αποθηκευτικού χώρου.

Με b συμβολίζεται η κατάσταση μπλοκαρίσματος της πρώτης μηχανής, όπου το b παίρνει τις τιμές $b = 1$: η μηχανή 1 είναι μπλοκαρισμένη και $b = 0$: η μηχανή 1 δεν είναι μπλοκαρισμένη.

Με s συμβολίζεται η κατάσταση ατροφίας της δεύτερης μηχανής, όπου το s παίρνει τις τιμές $s = 1$: η μηχανή 2 είναι σε ατροφία (πεινασμένη) και $s = 0$: η μηχανή 2 δεν είναι σε ατροφία.

Με βάσει τον παραπάνω συμβολισμό των επιμέρους καταστάσεων οι πιθανές καταστάσεις τους συστήματος είναι οι εξής.

1. $A=3, 0 \leq n \leq N, b=0, s=0$

Και οι δυο μηχανές είναι σε λειτουργία, το επίπεδο του αποθηκευτικού χώρου κυμαίνεται από 0 έως N , η πρώτη μηχανή δεν είναι μπλοκαρισμένη και η δεύτερη δεν πεινάει.



Σχήμα 2-1 :Κατάσταση $A=3, 0 \leq n \leq N, b=0, s=0$

2. $A=1, 0 \leq n \leq N, b=0, s=0$

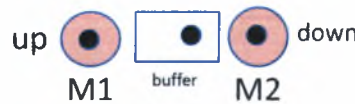
Η πρώτη μηχανή είναι σε λειτουργία και η δεύτερη είναι εκτός λειτουργίας, το επίπεδο του αποθηκευτικού χώρου κυμαίνεται από 0 έως N , η πρώτη μηχανή δεν είναι μπλοκαρισμένη και η δεύτερη δεν πεινάει.



Σχήμα 2-2: Κατάσταση $A=1$, $0 \leq n \leq N$, $b=0$, $s=0$

3. $A=2$, $0 \leq n \leq N$, $b=0$, $s=0$

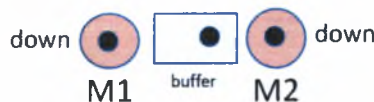
Η πρώτη μηχανή είναι εκτός λειτουργίας και η δεύτερη είναι σε λειτουργία, το επίπεδο του αποθηκευτικού χώρου κυμαίνεται από 0 έως N , η πρώτη μηχανή δεν είναι μπλοκαρισμένη και η δεύτερη δεν πεινάει.



Σχήμα 2-3: Κατάσταση $A=2$, $0 \leq n \leq N$, $b=0$, $s=0$

4. $A=0$, $0 \leq n \leq N$, $b=0$, $s=0$

Η πρώτη και η δεύτερη μηχανή είναι εκτός λειτουργίας, το επίπεδο του αποθηκευτικού χώρου κυμαίνεται από 0 έως N , η πρώτη μηχανή δεν είναι μπλοκαρισμένη και η δεύτερη δεν πεινάει.



Σχήμα 2-4: Κατάσταση $A=0$, $0 \leq n \leq N$, $b=0$, $s=0$

5. $A=3$, $n=N$, $b=1$, $s=0$

Η πρώτη και η δεύτερη μηχανή είναι σε λειτουργία, ο αποθηκευτικός χώρος είναι γεμάτος, η πρώτη μηχανή είναι μπλοκαρισμένη και η δεύτερη μηχανή δεν πεινάει.



Σχήμα 2-5: Κατάσταση $A=3, n=N, b=1, s=0$

6. $A=1, n=N, b=1, s=0$

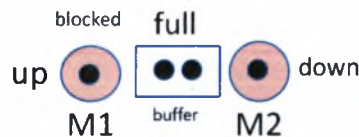
Η δεύτερη μηχανή είναι σε λειτουργία, η πρώτη μηχανή είναι εκτός λειτουργίας, ο αποθηκευτικός χώρος είναι γεμάτος, η πρώτη μηχανή είναι μπλοκαρισμένη, η δεύτερη μηχανή δεν πεινάει.



Σχήμα 2-6: Κατάσταση $A=1, 0n=N, b=1, s=0$

7. $A=2, n=N, b=1, s=0$

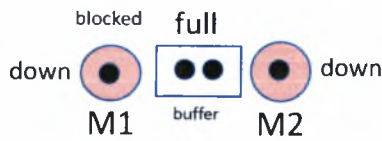
Η πρώτη μηχανή είναι σε λειτουργία, η δεύτερη μηχανή είναι εκτός λειτουργίας, ο αποθηκευτικός χώρος είναι γεμάτος, η πρώτη μηχανή είναι μπλοκαρισμένη, η δεύτερη μηχανή δεν πεινάει.



Σχήμα 2-7: Κατάσταση $A=2, n=N, b=1, s=0$

8. $A=0, n=N, b=1, s=0$

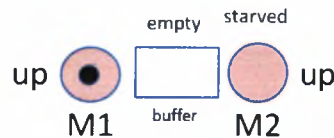
Η πρώτη και η δεύτερη μηχανή είναι εκτός λειτουργίας, ο αποθηκευτικός χώρος είναι γεμάτος, η πρώτη μηχανή είναι μπλοκαρισμένη, η δεύτερη μηχανή δεν πεινάει.



Σχήμα 2-8: Κατάσταση $A=0$, $n=N$, $b=1$, $s=0$

9. $A=3$, $n=0$, $b=0$, $s=1$

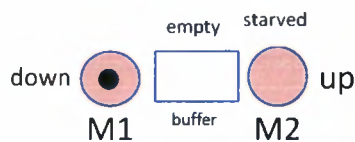
Η πρώτη και η δεύτερη μηχανή είναι σε λειτουργία, ο αποθηκευτικός χώρος είναι άδειος, η πρώτη μηχανή δεν είναι μπλοκαρισμένη, η δεύτερη μηχανή πεινάει.



Σχήμα 2-9: Κατάσταση $A=3$, $n=0$, $b=0$, $s=1$

10. $A=1$, $n=0$, $b=0$, $s=1$

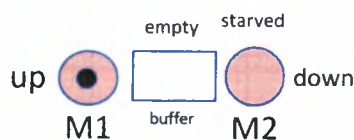
Η πρώτη μηχανή είναι εκτός λειτουργίας, η δεύτερη μηχανή είναι σε λειτουργία, ο αποθηκευτικός χώρος είναι άδειος, η δεύτερη μηχανή πεινάει.



Σχήμα 2-10: Κατάσταση $A=1$, $n=0$, $b=0$, $s=1$

11. $A=2$, $n=0$, $b=0$, $s=1$

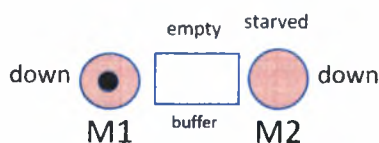
Η πρώτη μηχανή είναι σε λειτουργία, η δεύτερη μηχανή είναι εκτός λειτουργίας, ο αποθηκευτικός χώρος είναι άδειος, η δεύτερη μηχανή πεινάει.



Σχήμα 2-11: Κατάσταση $A=2, n=0, b=0, s=1$

12. $A=0, n=0, b=0, s=1$

Η πρώτη και η δεύτερη μηχανή είναι εκτός λειτουργίας, ο αποθηκευτικός χώρος είναι άδειος, και η δεύτερη μηχανή πεινάει.



Σχήμα 2-12: Κατάσταση $A=0, n=0, b=0, s=1$

Μετά τον προσδιορισμό όλων των πιθανών γεγονότων και πιθανών καταστάσεων γίνεται ανάλυση των μαθηματικών εργαλείων και η προσομοίωση του συστήματος.

2.3 Προσομοίωση συστήματος

Έχοντας καθορίσει όλες οι πιθανές καταστάσεις του συστήματος και τα γεγονότα που μπορούν να τις αλλάξουν, είναι εφικτό πλέον να αναπτυχθεί ο αλγόριθμος της προσομοίωσης. Για να γίνει αυτό, ορίζουμε τους τυχαίους χρόνους μέχρι την πραγματοποίηση κάθε ενός από τα έξι πιθανά γεγονότα που ορίστηκαν στο κεφάλαιο 2.2, ως εξής:

- *Event 1: $tp1$*
- *Event 2: $tp2$*
- *Event 3: $tf1$*
- *Event 4: $tf2$*
- *Event 5: $tr1$*
- *Event 6: $tr2$*

Όλοι οι παραπάνω χρόνοι ακολουθούν κάποια συγκεκριμένη κατανομή.

Η κατανομή προσδιορίζεται από τον χρήστη ανάλογα με τις ανάγκες ή τις απαιτήσεις του. Στη περίπτωση της συγκεκριμένης προσομοίωσης θα γίνει χρήση της εκθετικής

κατανομής για όλους τους παραπάνω χρόνους. Ο τύπος της αθροιστικής συνάρτησης κατανομής μιας εκθετικά κατανεμημένης τυχαίας μεταβλητής x είναι:

$$y = F(x) = 1 - e^{-\lambda x}$$

όπου λ η παράμετρος της εκθετικής κατανομής και $x = tp1, tp2, tf1, tf2, tr1, tr2$.

Το y , όντας πιθανότητα, παίρνει τιμές στο διάστημα $[0, 1]$. Λύνοντας ως προς x τον παραπάνω τύπο προκύπτει:

$$x = -\log(1 - y)/\lambda$$

Με βάση την παραπάνω σχέση, για να γεννηθεί ένας τυχαίος αριθμός που ακολουθεί την κατανομή του x , γεννάται πρώτα ένας τυχαίος αριθμός y που ακολουθεί την ομοιόμορφη κατανομή στο διάστημα $[0, 1]$ (οι συνηθισμένες γεννήτριες τυχαίων αριθμών γεννούν τέτοιους αριθμούς) και στη συνέχεια προκύπτει ο τυχαίος αριθμός x από την παραπάνω σχέση. Μάλιστα, οι συνηθισμένες γεννήτριες τυχαίων αριθμών που είναι ομοιόμορφα κατανεμημένοι στο διάστημα $[0, 1]$ δεν γεννούν πραγματικά τυχαίους αριθμούς y_1, y_2, y_3, \dots αλλά ψευδοτυχαίους αριθμούς χρησιμοποιώντας κάποια επαναληπτική συνάρτηση φ , τέτοια ώστε $y_n = \varphi(y_{n-1})$, $n = 1, 2, \dots$. Ο πρώτος αριθμός που γεννάται είναι ο $y_1 = \varphi(y_0)$, όπου ο αριθμός y_0 ονομάζεται «σπόρος» (*seed*) της γεννήτριας. Με τον τρόπο αυτό εξασφαλίζεται ότι ο ίδιος σπόρος θα γεννήσει το ίδιο δείγμα «τυχαίων» αριθμών. Αυτό είναι πολύ χρήσιμο, γιατί συχνά μπορεί να θέλουμε να συγκρίνουμε μέσω προσομοίωσης δύο ή περισσότερα συστήματα που είναι πανομοιότυπα και διαφέρουν μόνο ως προς κάποιες παραμέτρους. Για να είναι η σύγκριση «δίκαιη», θέλουμε να προσομοιώσουμε τα δύο συστήματα υπό το ίδιο δείγμα τυχαίων αριθμών.

Η παράμετρος λ της εκθετικής κατανομής δίνεται από τον χρήστη για το αντίστοιχο γεγονός. Συμβολίζουμε τη παράμετρο λ για το κάθε γεγονός ως εξής :

- *Event 1: P1*
- *Event 2: P2*
- *Event 3: F1*
- *Event 4: F2*
- *Event 5: R1*
- *Event 6: R2*

Επομένως ο τελικός μαθηματικός τύπος υπολογισμού των τυχαίων χρονών των γεγονότων είναι:

$$tp1, tp2, tf1, tf2, tr1, tr2 = -\log(1 - y)/P1, P2, F1, F2, R1, R2$$

Συνυπολογίζοντας όλα τα παραπάνω δεδομένα που έχουν αναφερθεί μπορεί να δημιουργηθεί και να μοντελοποιηθεί ένας αλγόριθμος ο οποίος θα προσομοιώσει την λειτουργία ενός τέτοιου συστήματος σε βάθος χρόνου και θα μπορεί να δώσει τα επιθυμητά αποτελέσματα που πρέπει να μετρηθούν και ενδιαφέρουν τον χρήστη.

Απαραίτητα στοιχεία για την αρχικοποίηση του αλγορίθμου είναι

- Ο προσδιορισμός της παραμέτρου λ της εκθετικής κατανομής για κάθε γεγονός που είναι πιθανό να συμβεί
- Ο προσδιορισμός της χωρητικότητας του αποθηκευτικού χώρου
- Ο συνολικός χρόνος προσομοίωσης του συστήματος

Εφόσον γίνει η αρχικοποίηση του αλγορίθμου ορίζεται σαν πρώτη κατάσταση από την φύση του συστήματος η κατάσταση κατά την οποία η πρώτη μηχανή και η δεύτερη μηχανή λειτουργούν, ο αποθηκευτικός χώρος είναι άδειος και η δεύτερη μηχανή πεινάει.

Παρατίθενται στη συνέχεια στο σύνολο τους οι επιθυμητές καταστάσεις και τα πιθανά γεγονότα που μπορούν να συμβούν δεδομένου ότι το σύστημα είναι στην κατάσταση αυτή.

1. $A=3, 0 \leq n \leq N, b=0, s=0$

Πιθανά γεγονότα $tp1, tp2, tf1, tf2, tr1, tr2$

2. $A=1, 0 \leq n \leq N, b=0, s=0$

Πιθανά γεγονότα $tp2, tf2, tr1$

3. $A=2, 0 \leq n \leq N, b=0, s=0$

Πιθανά γεγονότα $tp1, tf1, tr2$

4. $A=0, 0 \leq n \leq N, b=0, s=0$

Πιθανά γεγονότα $tr1, tr2$

5. $A=3, n=N, b=1, s=0$

Πιθανά γεγονότα $tp2, tf1, tf2,$

6. $A=1, n=N, b=1, s=0$

Πιθανά γεγονότα $tp2, tf2, tr1$

7. $A=2, n=N, b=1, s=0$

Πιθανά γεγονότα $tp1, tp2, tf1, tf2, tr1, tr2$

8. $A=0, n=N, b=1, s=0$

Πιθανά γεγονότα $tf1, tr2$

$$9. A=3, n=0, b=0, s=1$$

Πιθανά γεγονότα $tp1, tf1, tf2$

$$10. A=1, n=0, b=0, s=1$$

Πιθανά γεγονότα $tf2, tr1$

$$11. A=2, n=0, b=0, s=1$$

Πιθανά γεγονότα $tp1, tf1, tr2$

$$12. A=0, n=0, b=0, s=1$$

Πιθανά γεγονότα $tr1, tr2$

Με βάση την αρχική κατάσταση $A = 3, n = 0, b = 0, s = 1$ και σύμφωνα με το διάγραμμα των συγκεκριμένων πιθανών γεγονότων που θα προκαλέσουν την μετάβαση του συστήματος σε μια άλλη κατάσταση και αφού για κάθε κατάσταση και για κάθε νέα κατάσταση που βρίσκεται το σύστημα παράγονται οι τυχαίοι χρόνοι των επόμενων πιθανών γεγονότων, ο αλγόριθμος με τον τρόπο που είναι δομημένος ελέγχει σε ποια κατάσταση βρίσκεται το σύστημα και ποιες είναι οι πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί το σύστημα και με ποια γεγονότα γίνεται η μετάβαση.

Το κριτήριο επιλογής του πιθανού γεγονότος είναι με βάσει τον τυχαίο χρόνο που έχει παραχθεί. Ο αλγόριθμος επιλέγει και πραγματοποιεί το γεγονός που είναι επιτρεπτό να συμβεί βάσει της κατάστασης που βρίσκεται και έχει το μικρότερο χρόνο. Στη συνέχεια παράγεται νέος τυχαίος χρόνος για το γεγονός που πραγματοποιήθηκε και ανανεώνονται οι χρόνοι των γεγονότων που δεν πραγματοποιήθηκαν.

Με το πέρας του χρόνου προσομοίωσης ο αλγόριθμος τερματίζει και επιστρέφει τα μεγέθη που του έχουν ζητηθεί να υπολογίσει.

2.4 Μεγέθη που υπολογίζονται με την προσομοίωση του συστήματος

Απώτερος και βασικός σκοπός της προσομοίωσης αποτελεί η εξαγωγή όσο το δυνατόν πιο αξιόπιστων και σωστότερων αποτελεσμάτων στα ποσοτικά η ποιοτικά μεγέθη που θέλει ο χρήστης.

Με την εξαγωγή των αποτελεσμάτων ο χρήστης είναι σε θέση να αξιολογήσει όλες τις παραμέτρους του συστήματος και να κάνει τις απαραίτητες μετατροπές για την βελτίωση του.

Τα ποσοτικά και ποιοτικά μεγέθη που μπορούν να υπολογιστούν με τον αλγόριθμο που αναλύθηκε είναι.

- Υπολογίζεται ο συνολικός χρόνος που το σύστημα βρέθηκε σε κάθε επιμέρους κατάσταση. Και εν συνέχεια υπολογίζεται το ποσοστό επί τοις εκατό αυτού του χρόνου στο συνολικό χρόνο προσομοίωσης για κάθε κατάσταση τους συστήματος.

$$Casetime(i) = casetime(i) / tsim$$

όπου,

Casetime = το ποσοστό του συνολικού χρόνου σε μια κατάσταση και *i* από 1 έως 12 όσες και οι πιθανές καταστάσεις

Tsim = ο συνολικός χρόνος προσομοίωσης

- Για κάθε διακριτή στάθμη του αποθηκευτικού χώρου υπολογίζεται ο συνολικός χρόνος στο χρόνο της προσομοίωσης και εν συνέχεια το αντίστοιχο ποσοστό επί τοις εκατό.

$$Buffertime(i) = buffertime(i) / tsim$$

όπου,

Buffertime = το ποσοστό του συνολικού χρόνου σε ένα επίπεδο και *i* από 0 έως *N* όσα και τα πιθανά επίπεδα του αποθηκευτικού χώρου.

Tsim = ο συνολικός χρόνος προσομοίωσης

- Υπολογίζεται η μέση στάθμη του αποθηκευτικού χώρου στο συνολικό χρόνο προσομοίωσης.

$$Averagebuffer = \sum_{i=0}^N i * buffertime(i)$$

- Υπολογίζεται το ποσοστό του συνολικού χρόνου κατά το οποίο η πρώτη μηχανή είναι μπλοκαρισμένη. για να υπολογιστεί αυτό το ποσοστό αθροίζουμε τα ποσοστά χρόνου των αντιστοιχών καταστάσεων κατά τις οποίες η πρώτη μηχανή είναι μπλοκαρισμένη. Αυτό συμβαίνει στις καταστάσεις 5, 6, 7 ,8

$$P_b = casetime(5) + casetime(6) + casetime(7) + casetime(8)$$

- Υπολογίζεται το ποσοστό του συνολικού χρόνου κατά το οποίο η δεύτερη μηχανή πεινάει. Αυτό συμβαίνει στις καταστάσεις 9,10,11,12. Οπότε αθροίζονται τα ποσοστά αυτών των καταστάσεων.

$$P_s = \text{casetime}(9) + \text{casetime}(10) + \text{casetime}(11) + \text{casetime}(12)$$

- Υπολογίζεται ο συνολικός αριθμός παραγόμενων προϊόντων δηλαδή το σύνολο των προϊόντων που παραχώθηκαν από την δεύτερη μηχανή

Throughput = συνολικός αριθμός παραγόμενων προϊόντων στο συνολικό χρόνο προσομοίωσης

- Υπολογίζεται η δυναμικότητα παραγωγής του συστήματος

$$Averthroughput = throughput / tsim$$

Για κάθε αρχικοποίηση και αλλαγή των παραμέτρων τους συστήματος ο αλγόριθμος εξετάζεται τριάντα φορές με διαφορετικό σπόρο (*seed*) κάθε φορά, για πιο αξιόπιστα και ασφαλή δεδομένα. Έτσι για κάθε αλλαγή των παραμέτρων ο αλγόριθμος υπολογίζει:

- Τον μέσο όρο, $X = \frac{\sum_{i=0}^{30} X_i}{30}$

- Την τυπική απόκλιση, $S = \sqrt{\frac{\sum (X_i - X)^2}{30}}$

- Το διάστημα εμπιστοσύνης, $\Delta.E. = X \pm 1,96 * \frac{S}{\sqrt{30}}$

όπου ο αριθμός 1,96 είναι για διάστημα εμπιστοσύνης 95%.

Με τον υπολογισμό όλων αυτών των παραμέτρων είναι εφικτό να γίνει αξιολόγηση διαφόρων σεναρίων με διαφορετική παραμετροποίηση κάθε φορά.

Κεφάλαιο 3 Αριθμητικά αποτελέσματα προσομοίωσης

3.1 Αρχικοποίηση του συστήματος

Για την προσομοίωση του συστήματος πρέπει να οριστούν οι βασικές παράμετροι τους συστήματος και εν συνέχεια με την μεταβολή αυτών των παραμέτρων γίνεται οι διάφοροι πειραματικοί έλεγχοι και η εξαγωγή και επεξεργασία των αντιστοιχών αποτελεσμάτων.

3.1.1 Έναρξη αλγορίθμου και προσομοίωσης

Όπως αναφέρθηκε για την αρχικοποίηση του αλγορίθμου θα πρέπει να οριστούν οι βασικοί παράμετροι του συστήματος.

Ζητείται από τον χρήστη να οριστούν οι ρυθμοί επεξεργασίας στη μηχανή ένα και στην μηχανή δυο. Οι ρυθμοί επεξεργασίας αποτελούν και την παράμετρο λ της εκθετικής κατανομής για κάθε αντίστοιχο πραγματοποιήσιμο γεγονός. Επίσης ζητείται να προσδιοριστούν ο συνολικός χρόνος προσομοίωσης και η μέγιστη χωρητικότητα του αποθηκευτικού χώρου.

Έπειτα εκτελείται ο αλγόριθμος και δίνει τα αποτελέσματα. Ο αλγόριθμος εκτελείται τριάντα φορές για κάθε παραμετροποίηση τους συστήματος με διαφορετικό σπόρο κάθε φορά έτσι ώστε να γίνεται στατιστική ανάλυση των αποτελεσμάτων. Υπολογίζεται ο μέσος όρος των τριάντα επαναλήψεων, η τυπική απόκλιση και το διάστημα εμπιστοσύνης για 95%.

Μετά την εξαγωγή των αποτελεσμάτων για κάθε αλλαγή των παραμέτρων τους συστήματος παρατίθενται πινάκες με τις τιμές του μέσου, της τυπικής απόκλισης και του διαστήματος εμπιστοσύνης για τα μεγέθη του μέσου αποθέματος, του ρυθμού παραγωγής, του ποσοστού του χρόνου όπου η μηχανή ένα είναι μπλοκαρισμένη και του ποσοστού του χρόνου όπου η μηχανή δυο πεινάει. Τέλος δημιουργούνται γραφήματα όπου απεικονίζεται η συμπεριφορά του μέσου ορού για κάθε μετρούμενο μέγεθος και εξάγονται συμπεράσματα και παρατηρήσεις των μεγεθών σε σχέση με την αλλαγή των παραμέτρων.

3.2 Παρουσίαση πειραματικών ελέγχων

3.2.1 Πρώτος πειραματικός έλεγχος

Ορίζονται οι ρυθμοί επεξεργασίας στις μηχανές ένα και δυο που αποτελούν και την παράμετρο λ των αντιστοιχών εκθετικών κατανομών.

$$P1=P2= 10 \text{ parts/hr}$$

Οι ρυθμοί βλαβών για κάθε μηχανή

$$F1=F2= 0,1 \text{ failures/hr}$$

Οι ρυθμοί επιδιορθώσεων για κάθε βλάβη μηχανής

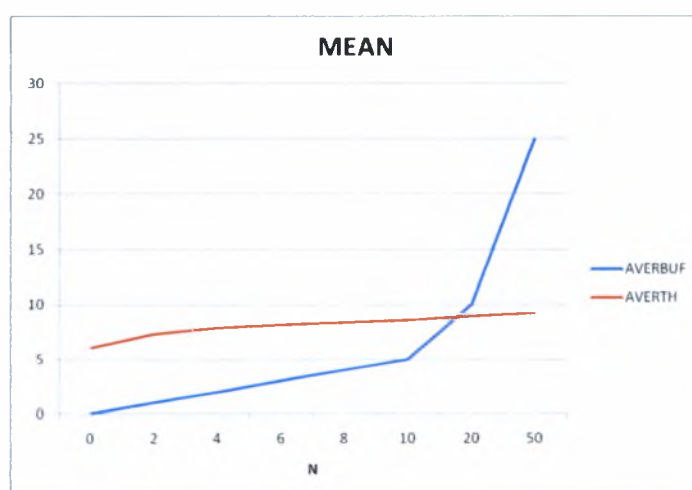
$$R1=R2= 2 \text{ repair/hr}$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή της χωρητικότητας του αποθηκευτικού χώρου και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1=P2= 10 \text{ parts/hr}$, $F1=F2= 0,1 \text{ failures/hr}$, $R1=R2= 2 \text{ repair/hr}$ και για $N=0, 2, 4, 6, 8, 10, 20, 50$ τα αποτελέσματα διαμορφώνονται ως εξής:

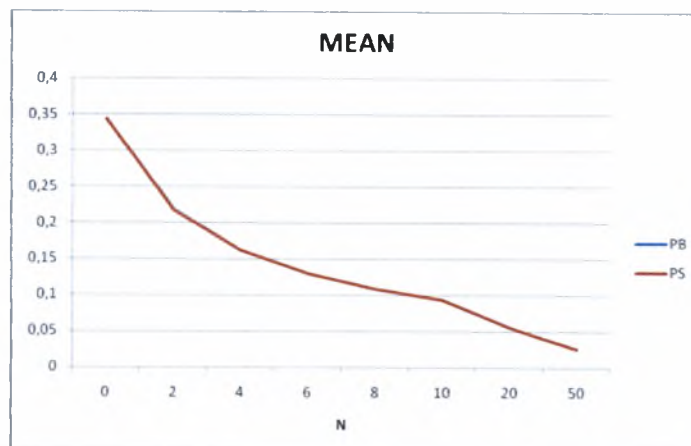
Πίνακας 3-1: Μέσος όρος για $P1=P2=10$, $F1=F2=0.1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	0	6,115486	0,344564	0,34457
2	1,000245	7,366723	0,218242	0,217902
4	1,999324	7,92187	0,162145	0,162238
6	3,000324	8,23764	0,130125	0,129887
8	3,999938	8,444562	0,10919	0,108877
10	5,006757	8,596454	0,094124	0,093784
20	10,004353	8,96819	0,056081	0,056085
50	24,962651	9,271977	0,02535	0,025401



Σχήμα 3-1: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer

Παρατηρείται ότι με την αύξηση του χώρου του ενδιάμεσου χώρου αποθήκευσης η αποδοτικότητα του συστήματος αυξάνεται. Το ίδιο συμβαίνει και με το μέσο επίπεδο του ενδιάμεσου αποθηκευτικού χώρου.



Σχήμα 3-2 Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer

Παρατηρείται ότι καθώς αυξάνεται η χωρητικότητα τους ενδιάμεσου χώρου αποθήκευσης τα ποσοστά μπλοκαρίσματος της πρώτης μηχανής αλλά και τα ποσοστά πείνας της δεύτερης μειώνονται δραστικά.

Πίνακας 3-2: Τυπική απόκλιση για $P1=P2=10$, $F1=F2=0.1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	0	0,010944	0,000852	0,000892
2	0,001885	0,009411	0,000702	0,000774
4	0,005575	0,009063	0,000835	0,001213
6	0,009983	0,01072	0,001103	0,001018
8	0,015962	0,01011	0,00118	0,000959
10	0,014616	0,007465	0,000762	0,000695
20	0,052661	0,010037	0,000916	0,000848
50	0,312987	0,008928	0,000864	0,000708

Πίνακας 3-3: Διάστημα εμπιστοσύνης για $P1=P2=10$, $F1=F2=0.1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	[0.000000-0.000000]	[6.111570-6.119402]	[0.344259-0.344869]	[0.344250-0.344889]
2	[0.999570-1.000920]	[7.363355-7.370091]	[0.217990-0.218493]	[0.217625-0.218179]
4	[1.997329-2.001319]	[7.918626-7.925113]	[0.161847-0.162444]	[0.161804-0.162672]
6	[2.996752-3.003896]	[8.233804-8.241476]	[0.129730-0.130519]	[0.129523-0.130251]
8	[3.994226-4.005650]	[8.440944-8.448180]	[0.108768-0.109612]	[0.108534-0.109221]
10	[5.001527-5.011988]	[8.593782-8.599125]	[0.093852-0.094397]	[0.093535-0.094033]
20	[9.985509-10.023197]	[8.964599-8.971782]	[0.055753-0.056408]	[0.055781-0.056388]
50	[24.850650-25.074652]	[9.268782-9.275172]	[0.025041-0.025659]	[0.025148-0.025654]

3.2.2 Δεύτερος πειραματικός έλεγχος

Ορίζονται οι ρυθμοί επεξεργασίας στις μηχανές ένα και δυο που αποτελούν και την παράμετρο λ των αντιστοιχών εκθετικών κατανομών.

$$P1=P2= 10 \text{ parts/hr}$$

Οι ρυθμοί βλαβών για κάθε μηχανή

$$F1=F2= 0,1 \text{ failures/hr}$$

Η χωρητικότητα του αποθηκευτικού χώρου

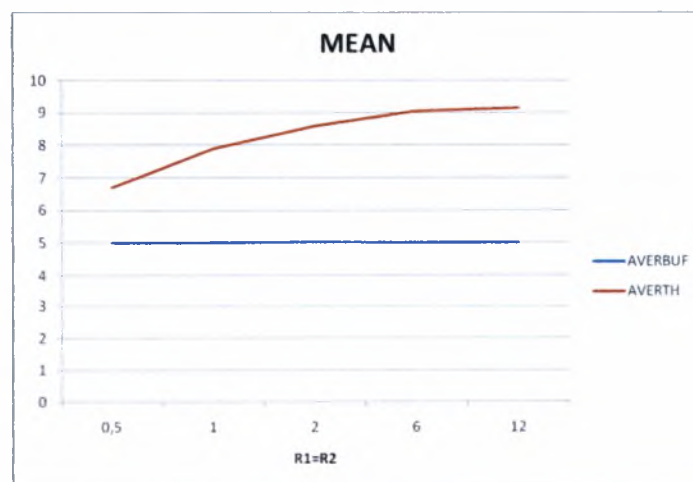
$$N=10$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή των ρυθμών του χρόνου επιδιόρθωσης των βλαβών της κάθε μηχανής και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1=P2= 10 \text{ parts/hr}$, $F1=F2= 0,1 \text{ failures/hr}$, $N=10$ και για $R1=R2 = 0.5, 1, 2, 6, 12$ τα αποτελέσματα διαμορφώνονται ως εξής:

Πίνακας 3-4: Μέσος όρος για $P1=P2=10$, $F1=F2=0.1$, $N=10$

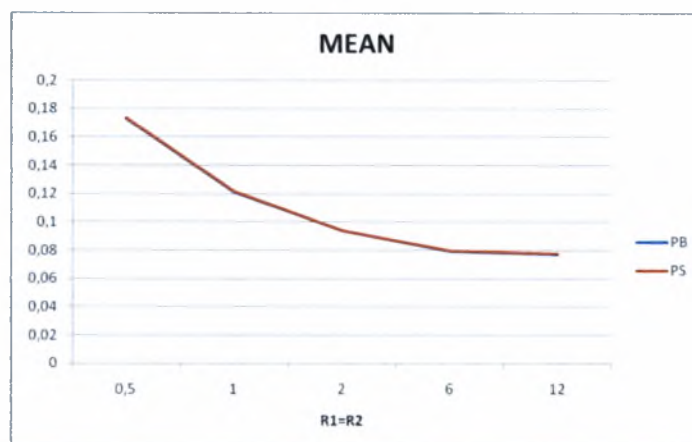
R1=R2	AVERBUF	AVERTH	PB	PS
0,5	4,996165	6,70916	0,172896	0,173427
1	4,993236	7,911644	0,121171	0,12165
2	5,006757	8,596454	0,094124	0,093784
6	4,991873	9,047333	0,079069	0,079692
12	4,997228	9,145937	0,077297	0,077513



Σχήμα 3-3: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών επιδιόρθωσης

Παρατηρείται ότι με την αύξηση των ρυθμών επιδιόρθωσης το επίπεδο του μέσου ενδιάμεσου χωρου απο αποθήκευσης δεν μεταβαλεται ιδιαίτερα. Ενώ αντίθετα η

απόδοση του συστήματος αυξάνεται μιας και το σύστημα είναι περισσότερη ώρα σε λειτουργία.



Σχήμα 3-4: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών επιδιόρθωσης

Με τη αύξηση των ρυθμών επιδιόρθωσης το ποσοστό του χρόνου οπου η πρώτη μηχανή είναι μπλοκαρισμένη και το ποσοστό του χρόνου οπου η δεύτερη μηχανή πεινάσει ελαύνεται.

Πίνακας 3-5: Τυπική απόκλιση για $P1=P2=10$, $F1=F2=0.1$, $N=10$

R1=R2	AVERBUF	AVERTH	PB	PS
0,5	0,022614	0,023761	0,002123	0,001643
1	0,025306	0,017462	0,001596	0,001523
2	0,014616	0,007465	0,000762	0,000695
6	0,024791	0,007525	0,000796	0,000824
12	0,020004	0,007584	0,000735	0,000656

Πίνακας 3-6: Διάστημα εμπιστοσύνης για $P1=P2=10$, $F1=F2=0.1$, $N=10$

R1=R2	AVERBUF	AVERTH	PB	PS
0,5	[4.988072-5.004257]	[6.700657-6.717662]	[0.172136-0.173655]	[0.172839-0.174015]
1	[4.984180-5.002291]	[7.905395-7.917892]	[0.120600-0.121742]	[0.121105-0.122195]
2	[5.001527-5.011988]	[8.593782-8.599125]	[0.093852-0.094397]	[0.093535-0.094033]
6	[4.983001-5.000744]	[9.044640-9.050026]	[0.078784-0.079354]	[0.079397-0.079987]
12	[4.990070-5.004386]	[9.143223-9.148651]	[0.077034-0.077560]	[0.077278-0.077748]

3.2.3 Τρίτος πειραματικός έλεγχος

Ορίζονται οι ρυθμοί επεξεργασίας στις μηχανές ένα και δυο που αποτελούν και την παράμετρο λ των αντιστοιχών εκθετικών κατανομών.

$$P1=P2= 10 \text{ parts/hr}$$

Οι ρυθμοί επιδιορθώσεων για κάθε βλάβη μηχανής

$$R1=R2= 2 \text{ repair/hr}$$

Η χωρητικότητα του αποθηκευτικού χώρου

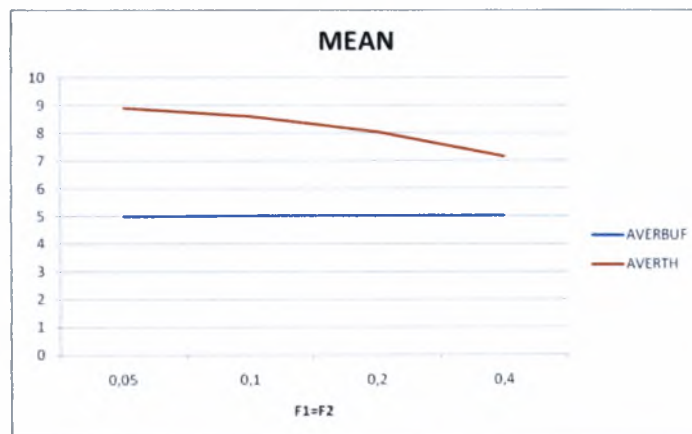
$$N=10$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή των ρυθμών του χρόνου βλαβών της κάθε μηχανής και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1=P2= 10 \text{ parts/hr}$, $R1=R2= 2 \text{ failures/hr}$, $N=10$ και για $F1=F2 = 0.05, 0.1, 0.2, 0.4$ τα αποτελέσματα διαμορφώνονται ως εξής:

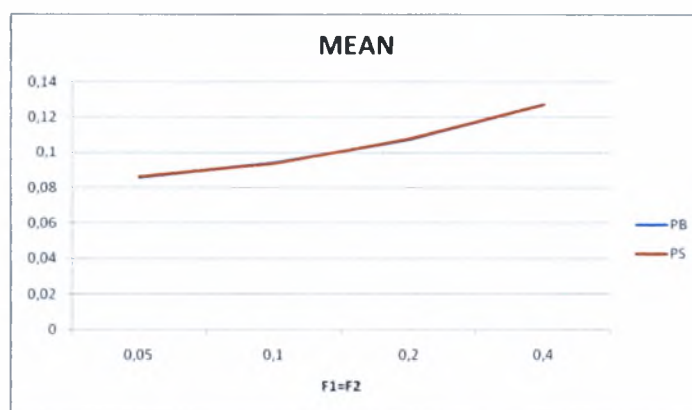
Πίνακας 3-7: Μέσος όρος για $P1=P2=10$, $R1=R2=2$, $N=10$

F1=F2	AVERBUF	AVERTH	PB	PS
0,05	4,994646	8,900764	0,08571	0,086151
0,1	5,006757	8,596454	0,094124	0,093784
0,2	4,997524	8,045004	0,107465	0,107733
0,4	5,000055	7,14358	0,127523	0,127301



Σχήμα 3-5: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών βλαβών

Παρατηρείται ότι με την αύξηση των ρυθμών βλαβών στο σύστημα το μέσο επίπεδο του ενδιάμεσου χώρου αποθήκευσης δε μεταβάλλεται ιδιαίτερα ενώ η απόδοση του συστήματος μειώνεται.



Σχήμα 3-6: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών βλαβών

Είναι λογικό με την αύξηση των ρυθμών βλαβών, τα ποσοστά μπλοκαρίσματος και πείνας να αυξάνονται μιας και το σύστημα λειτουργεί λιγότερο χρόνο.

Πίνακας 3-8: Τυπική απόκλιση για $P1=P2=10$, $R1=R2=2$, $N=10$

F1=F2	AVERBUF	AVERTH	PB	PS
0,05	0,024532	0,010204	0,001047	0,000986
0,1	0,014616	0,007465	0,000762	0,000695
0,2	0,022033	0,009762	0,00105	0,000966
0,4	0,020413	0,012219	0,00122	0,001206

Πίνακας 3-9: Διάστημα εμπιστοσύνης για $P1=P2=10$, $R1=R2=2$, $N=10$

F1=F2	AVERBUF	AVERTH	PB	PS
0,05	[4.985868-5.003425]	[8.897113-8.904415]	[0.085335-0.086085]	[0.085798-0.086504]
0,1	[5.001527-5.011988]	[8.593782-8.599125]	[0.093852-0.094397]	[0.093535-0.094033]
0,2	[4.989639-5.005408]	[8.041510-8.048497]	[0.107089-0.107841]	[0.107387-0.108079]
0,4	[4.992751-5.007360]	[7.139208-7.147953]	[0.127086-0.127959]	[0.126870-0.127733]

3.2.4 Τέταρτος πειραματικός έλεγχος

Ορίζονται οι ρυθμοί επεξεργασίας στις μηχανές ένα και δυο που αποτελούν και την παράμετρο λ των αντιστοιχών εκθετικών κατανομών.

$$P1=P2= 10 \text{ parts/hr}$$

Η χωρητικότητα του αποθηκευτικού χώρου

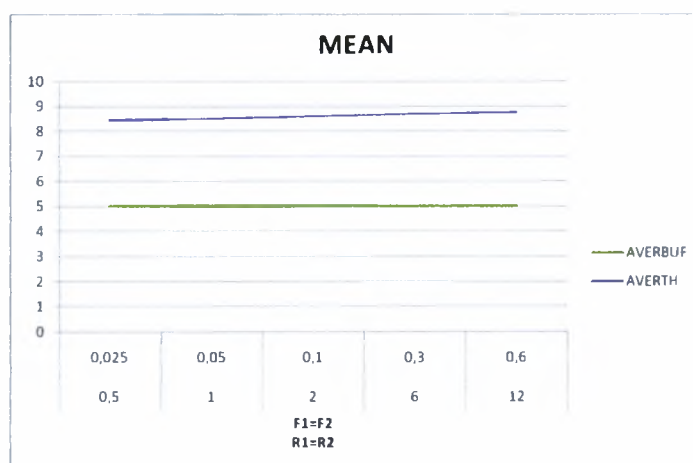
$$N=10$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή των ρυθμών του χρόνου βλαβών της κάθε μηχανής, των ρυθμών του χρόνου επιδιόρθωσης της κάθε μηχανής αντίστοιχα και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1=P2= 10 \text{ parts/hr}$, $N=10$, για $F1=F2 = 0.0025, 0.05, 0.1, 0.3, 0.6 \text{ failures/hr}$ και $R1=R2= 0.5, 1, 2, 6, 12 \text{ repairs/hr}$ τα αποτελέσματα διαμορφώνονται ως εξής:

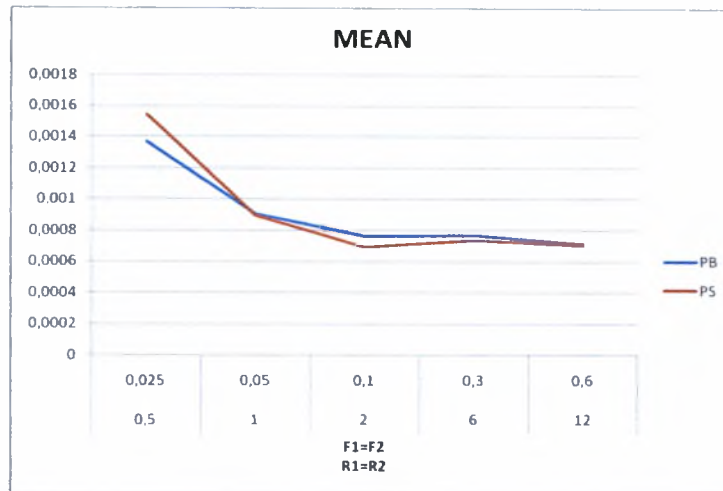
Πίνακας 3-10: Μέσος όρος για $P1=P2=10$, $N=10$

R1=R2	F1=F2	AVERBUF	AVERTH	PB	PS
0,5	0,025	4,99588	8,459364	0,107376	0,107533
1	0,05	4,997467	8,515661	0,101534	0,101696
2	0,1	5,006757	8,596454	0,094124	0,093784
6	0,3	4,996867	8,702561	0,08362	0,083837
12	0,6	4,998391	8,742944	0,08017	0,080196



Σχήμα 3-7: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει των ρυθμών βλαβών και επιδιόρθωσης

Όπως παρατηρήθηκε και στον προηγούμενο πειραματικό έλεγχο η αύξηση των ρυθμών επιδιόρθωσης και βλαβών δεν επηρεάζουν το μέσο επίπεδο του ενδιάμεσου χώρου αποθήκευσης. Το ίδιο συμβαίνει και με την ταυτόχρονη αύξηση τους. Παρόλα αυτά παρατηρείται ότι με την ταυτόχρονη αύξηση των ρυθμών επιδιόρθωσης και βλαβών η απόδοση του συστήματος αυξάνεται.



Σχήμα 3-8: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει των ρυθμών βλαβών και επιδιορθώσεως

Τα ποσοστά του χρόνου μπλοκαρίσματος και πείνας παρατηρείται ότι μειώνεται όσο αυξάνονται οι ρυθμοί βλαβών και επιδιορθώσεων.

Πίνακας 3-11: Τυπική απόκλιση για $P1=P2=10, N=10$

R1=R2	F1=F2	AVERBUF	AVERTH	PB	PS
0,5	0,025	0,022734	0,01824	0,001367	0,001547
1	0,05	0,019341	0,010127	0,000906	0,000895
2	0,1	0,014616	0,007465	0,000762	0,000695
6	0,3	0,018387	0,009413	0,000767	0,00074
12	0,6	0,018124	0,00696	0,000712	0,000702

Πίνακας 3-12: Διάστημα εμπιστοσύνης για $P1=P2=10, N=10$

R1=R2	F1=F2	AVERBUF	AVERTH	PB	PS
0,5	0,025	[4.987745-5.004015]	[8.452837-8.465891]	[0.106887-0.107865]	[0.106979-0.108087]
1	0,05	[4.990546-5.004388]	[8.512037-8.519285]	[0.101210-0.101858]	[0.101376-0.102016]
2	0,1	[5.001527-5.011988]	[8.593782-8.599125]	[0.093852-0.094397]	[0.093535-0.094033]
6	0,3	[4.990287-5.003446]	[8.699193-8.705929]	[0.083345-0.083894]	[0.083572-0.084102]
12	0,6	[4.991906-5.004876]	[8.740453-8.745434]	[0.079915-0.080425]	[0.079945-0.080448]

3.2.5 Πέμπτος πειραματικός έλεγχος

Ορίζεται ο ρυθμός επεξεργασίας για την πρώτη μηχανή

$$P1=10 \text{ parts/hr}$$

Οι ρυθμοί βλαβών για κάθε μηχανή

$$F1=F2= 0,1 \text{ failures/hr}$$

Οι ρυθμοί επιδιορθώσεων για κάθε βλάβη μηχανής

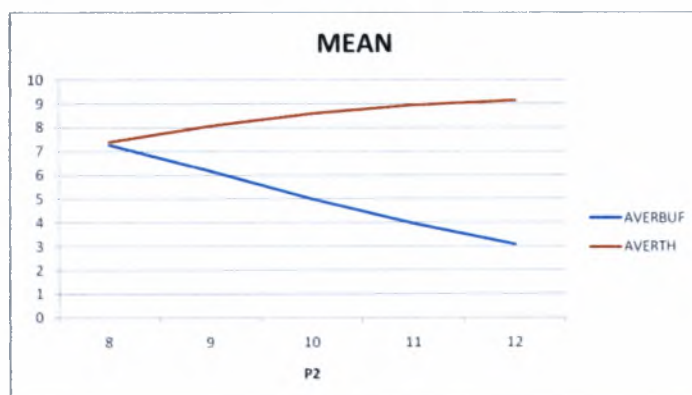
$$R1=R2= 2 \text{ repair/hr}$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή των ρυθμών του χρόνου επεξεργασίας της δεύτερης μηχανής και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1= 10 \text{ parts/hr}$, $N=10$, για $F1=F2 = 0.1 \text{ failures/hr}$ για $R1=R2= 2 \text{ repairs/hr}$ και για $P2 = 8, 9, 10, 11, 12 \text{ parts/hr}$ τα αποτελέσματα διαμορφώνονται ως εξής:

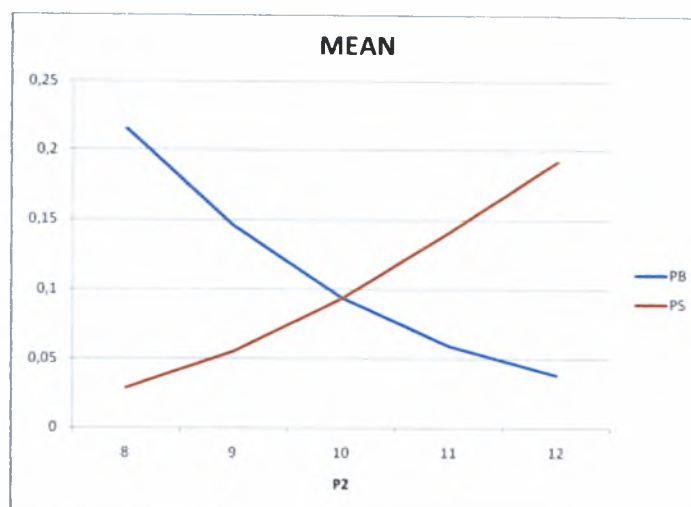
Πίνακας 3-13: Μέσος όρος για $P1=10, F1=F2=0.1, R1=R2=2, N=10$

P2	AVERBUF	AVERTH	PB	PS
8	7,257551	7,393709	0,215323	0,028709
9	6,153303	8,082597	0,145685	0,055324
10	5,006757	8,596454	0,094124	0,093784
11	3,955029	8,937348	0,059503	0,141273
12	3,098394	9,14714	0,038144	0,192292



Σχήμα 3-9: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει του ρυθμού παραγωγής της M2

Όσο αυξάνεται ο ρυθμός επεξεργασίας στην δεύτερη μηχανή είναι λογικό να αυξάνεται η συνολική απόδοση του συστήματος και φυσικά να μειώνεται το μέσο επίπεδο του ενδιάμεσου αποθηκευτικού χώρου.



Σχήμα 3-10: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει του ρυθμού παραγωγής της M2

Η αύξηση του ρυθμού επεξεργασίας της δεύτερης μηχανής είναι λογικό να ελαύνει το ποσοστό του χρόνου όπου η πρώτη μηχανή είναι μπλοκαρισμένη και να αυξάνει το ποσοστό του χρόνου όπου η δεύτερη μηχανή πεινάει.

Πίνακας 3-14: Τυπική απόκλιση για $P1=10$, $F1=F2=0.1$, $R1=R2=2$, $N=10$

P2	AVERBUF	AVERTH	PB	PS
8	0,017713	0,009042	0,001145	0,000564
9	0,019978	0,009809	0,001053	0,00076
10	0,014616	0,007465	0,000762	0,000695
11	0,021897	0,010291	0,000797	0,001094
12	0,021045	0,009576	0,000716	0,001276

Πίνακας 3-15: Διάστημα εμπιστοσύνης για $P1=10$, $F1=F2=0.1$, $R1=R2=2$, $N=10$

P2	AVERBUF	AVERTH	PB	PS
8	[7.251212-7.263889]	[7.390474-7.396945]	[0.214913-0.215732]	[0.028508-0.028911]
9	[6.146154-6.160452]	[8.079086-8.086107]	[0.145308-0.146062]	[0.055052-0.055596]
10	[5.001527-5.011988]	[8.593782-8.599125]	[0.093852-0.094397]	[0.093535-0.094033]
11	[3.947194-3.962865]	[8.933665-8.941031]	[0.059218-0.059788]	[0.140882-0.141665]
12	[3.090863-3.105925]	[9.143713-9.150567]	[0.037887-0.038400]	[0.191835-0.192748]

3.2.6 Έκτος πειραματικός έλεγχος

Σε αυτόν τον πειραματικό έλεγχο θα γίνουν κάποιες μετατροπές. Οι χρόνοι παραγωγής στις μηχανές θα γεννιούνται τυχαία χρησιμοποιώντας διαφορετική

κατανομή. Θα χρησιμοποιηθεί η ομοιόμορφη κατανομή. Ο τύπος της αθροιστικής ομοιόμορφης κατανομής είναι:

$$F(x) = 1 / (b-a) \text{ για } a < x < b$$

Λύνοντας ως προς x τότε

Οι τυχαίοι χρόνοι επεξεργασίας δίνονται από τον τύπο:

$$x = a + (b - a)y$$

Οπού $x = tp1, tp2$

$y =$ τυχαίοι αριθμοί από 0 έως 1

Τα a και b είναι τα όρια για την ομοιόμορφη κατανομή και δίνεται από το χρήστη. Επομένως για κάθε μηχανή ορίζεται κάποιο a και κάποιο b είτε για τους χρόνους επεξεργασίας είτε για τους χρόνους βλαβών είτε για τους χρόνους επιδιόρθωσης.

Στο συγκεκριμένο πειραματικό έλεγχο θα χρησιμοποιηθεί η ομοιόμορφη κατανομή για τους χρόνους παραγωγής και η εκθετική κατανομή για τους χρόνους βλαβών και επιδιορθώσεων.

Ορίζονται οι παράμετροι για τους χρόνους παραγωγής

$$P1=P2 \text{ με } a=0.05\text{hour/part} \text{ και } b=0.15\text{hour/part}$$

Οι ρυθμοί βλαβών για κάθε μηχανή

$$F1=F2= 0,1 \text{ failures/hr}$$

Οι ρυθμοί επιδιορθώσεων για κάθε βλάβη μηχανής

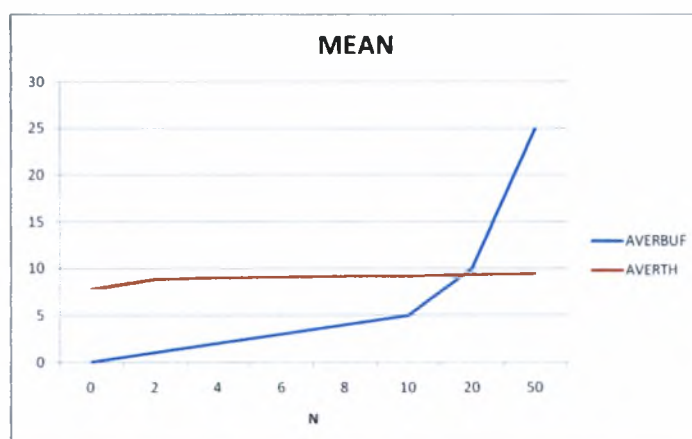
$$R1=R2= 2 \text{ repair/hr}$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή της χωρητικότητας του αποθηκευτικού χώρου και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1= P2$ με $a=0.05\text{hour/part}$ και $b=0.15\text{hour/part}$, για $F1=F2 = 0.1 \text{ failures/hr}$ για $R1=R2= 2 \text{ repairs/hr}$ και για $N=0,2,4,6,8,10,20,50$ τα αποτελέσματα διαμορφώνονται ως εξής:

Πίνακας 3-16: Μέσος όρος για $P1=P2$ με $a=0.05$ και $b=0.15$, $F1=F2=0,1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	0	7,803547	0,173547	0,173594
2	0,999797	8,845665	0,068721	0,068741
4	1,999956	9,022139	0,050912	0,050941
6	3,000206	9,115752	0,04178	0,041662
8	4,001114	9,169278	0,035995	0,036002
10	4,997483	9,213296	0,031582	0,031522
20	9,994384	9,33183	0,01961	0,019587
50	24,902911	9,432858	0,009171	0,009334



Σχήμα 3-11: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer

Παρατηρείται ότι τα αποτελέσματα σε σύγκριση με την εκθετική κατανομή είναι καλύτερα. Το σύστημα έχει καλύτερες αποδόσεις σε σχέση με την εκθετική κατανομή. Αυτό συμβαίνει διότι η εκθετική κατανομή έχει μεγαλύτερη μεταβλητότητα σε σχέση με την ομοιόμορφη. Η μεταβλητότητα της εκθετικής είναι:

$$CV = s / x$$

Όπου CV η μεταβλητότητα

S = τυπική απόκλιση

X = ο μέσος όρος

Για την εκθετική ισχύει

$$s = 1/\lambda \text{ και } x = 1/\lambda$$

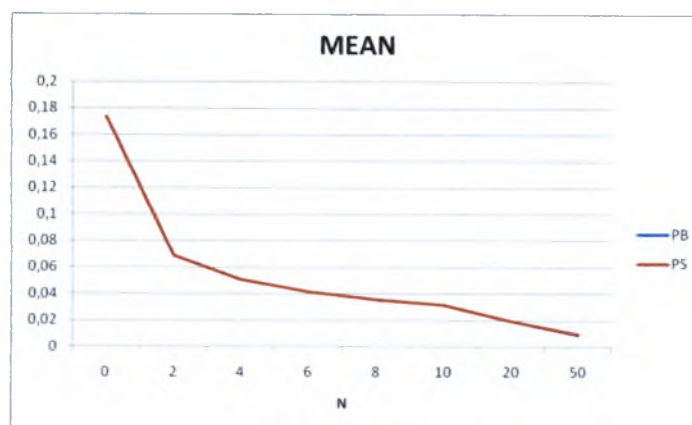
επομένως CV εκθετικής είναι 1

Για την ομοιόμορφη ισχύει

$$s = \frac{b-a}{\sqrt{12}} \text{ και } x = (b-a)/2$$

επομένως για $a=0.05$ και $b=0.15$

$CV = 0.56 < 1$ της εκθετικής



Σχήμα 3-12: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer

Το ίδιο συμβαίνει και για τα ποσοστά μπλοκαρίσματος και πείνας. Εφόσον είναι πιο γρήγορο σχετικά και αποδοτικότερο το σύστημα.

Πίνακας 3-17: Τυπική απόκλιση για $P1=P2$ με $a=0.05$ και $b=0.15$, $F1=F2=0,1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	0	0,005548	0,000498	0,000538
2	0,001998	0,007994	0,000509	0,000493
4	0,00943	0,007934	0,000533	0,000656
6	0,01605	0,008044	0,000504	0,000642
8	0,026872	0,008265	0,000673	0,000617
10	0,037982	0,009169	0,000682	0,000647
20	0,114083	0,006298	0,000543	0,000526
50	0,518385	0,00551	0,000563	0,000537

Πίνακας 3-18: Διάστημα εμπιστοσύνης για $P1=P2$ με $a=0.05$ και $b=0.15$, $F1=F2=0,1$, $R1=R2=2$

N	AVERBUF	AVERTH	PB	PS
0	[0.000000-0.000000]	[7.801561-7.805532]	[0.173369-0.173726]	[0.173401-0.173786]
2	[0.999082-1.000511]	[8.842805-8.848526]	[0.068539-0.068904]	[0.068565-0.068918]
4	[1.996581-2.003330]	[9.019300-9.024979]	[0.050722-0.051103]	[0.050706-0.051176]
6	[2.994462-3.005949]	[9.112874-9.118631]	[0.041600-0.041960]	[0.041432-0.041892]
8	[3.991498-4.010730]	[9.166321-9.172236]	[0.035755-0.036236]	[0.035781-0.036223]
10	[4.983892-5.011075]	[9.210015-9.216577]	[0.031338-0.031826]	[0.031290-0.031753]
20	[9.953560-10.035208]	[9.329577-9.334084]	[0.019416-0.019804]	[0.019399-0.019775]
50	[24.717410-25.088413]	[9.430886-9.434830]	[0.008970-0.009373]	[0.009142-0.009527]

3.2.7 Έβδομος πειραματικός έλεγχος

Σε αυτόν τον πειραματικό έλεγχο οι χρόνοι παραγωγής θα γεννιούνται από την εκθετική κατανομή ενώ αντίθετα οι χρόνοι βλαβών και οι χρόνοι επιδιορθώσεων θα γεννιούνται από την ομοιόμορφη κατανομή.

Ορίζονται οι παράμετροι για τους χρόνους παραγωγής

$$P1=P2= 10 \text{ parts/hr}$$

Οι ρυθμοί βλαβών για κάθε μηχανή

$$F1=F2= \text{ με } a=5\text{hr/failure} \text{ και } b=15\text{hr/failure}$$

Οι ρυθμοί επιδιορθώσεων για κάθε βλάβη μηχανής

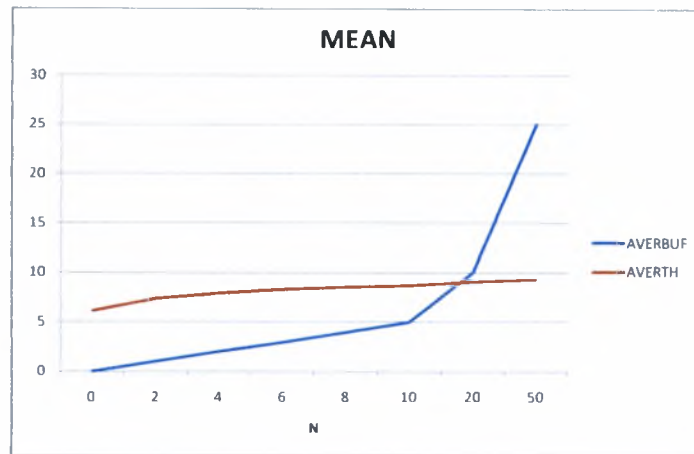
$$R1=R2 \text{ με } a=0.16\text{hr/repair} \text{ και } b=0.83\text{hr/repair}$$

Και στο συγκεκριμένο έλεγχο γίνεται αλλαγή της χωρητικότητας του αποθηκευτικού χώρου και εξετάζονται τα αντίστοιχα αποτελέσματα.

Επομένως για $P1= P2=10 \text{ parts/hr}$, για $F1=F2$ με $a=5\text{hr/failure}$ και $b=0.15\text{hr/failure}$ για $R1=R2$ με $a=0.16\text{hr/repair}$ και $b=0.83\text{hr/repair}$ και για $N=0,2,4,6,8,10,20,50$ τα αποτελέσματα διαμορφώνονται ως εξής:

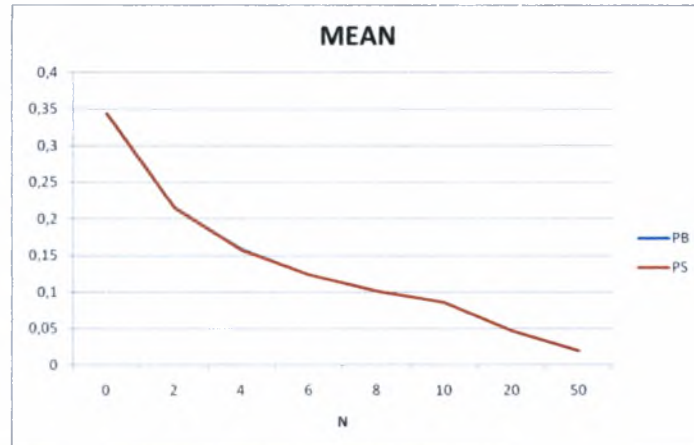
Πίνακας 3-19: Μέσος όρος για $P1=P2=10$, $F1=F2$ με $a=5$ και $b=15$, $R1=R2$ με $a=0,16$ και $b=0.83$

N	AVERBUF	AVERTH	PB	PS
0	0	6,128087	0,343753	0,343572
2	0,999539	7,397322	0,215146	0,215531
4	2,002312	7,966666	0,158107	0,157504
6	3,000701	8,298597	0,124241	0,124048
8	4,002647	8,516712	0,102122	0,101835
10	4,99843	8,674353	0,085949	0,086118
20	9,993689	9,060123	0,046986	0,047056
50	24,969967	9,328052	0,019792	0,01974



Σχήμα 3-13: Μεταβολή μέσου όρου του μέσου επιπέδου του buffer και του μέσου όρου απόδοσης(throughput) συναρτήσει της χωρητικότητας του buffer

Λόγω της μεταβλητότητας που αναφέρθηκε στον προηγούμενο πειραματικό έλεγχο παρατηρείται ότι η απόδοση αυξάνεται ελάχιστα σε σχέση με την εκθετική μιας και η ομοιόμορφη κατανομή στους χρόνους βλαβών και επιδιορθώσεων επιταχύνει ελάχιστα την παραγωγική διαδικασία.



Σχήμα 3-14: Μεταβολή μέσου όρου των ποσοστών μπλοκαρίσματος και πείνας συναρτήσει της χωρητικότητας του buffer

Παρατηρείται επίσης ελάχιστη μείωση στα ποσοστά μπλοκαρίσματος και πείνας μιας και η απόδοση του συστήματος είναι ελάχιστα μεγαλύτερη.

Πίνακας 3-20: Τυπική απόκλιση για $P1=P2=10$, $F1=F2$ με $a=5$ & $b=15$, $R1=R2$ με $a=0,16$ & $b=0.83$

N	AVERBUF	AVERTH	PB	PS
0	0	0,005865	0,000714	0,000726
2	0,00173	0,007396	0,000629	0,000561
4	0,004152	0,008032	0,000583	0,000644
6	0,00947	0,009022	0,000779	0,000921
8	0,013773	0,008149	0,000996	0,000851
10	0,017417	0,008489	0,000742	0,0008
20	0,05938	0,007121	0,000815	0,000888
50	0,367907	0,007025	0,000727	0,000864

Πίνακας 3-21: Διάστημα εμπιστοσύνης $P1=P2=10$, $F1=F2$ με $a=5$ & $b=15$, $R1=R2$ με $a=0,1$ & $b=0.83$

N	AVERBUF	AVERTH	PB	PS
0	[0.000000-0.000000]	[6.125989-6.130186]	[0.343497-0.344008]	[0.343312-0.343831]
2	[0.998920-1.000158]	[7.394676-7.399969]	[0.214921-0.215371]	[0.215331-0.215732]
4	[2.000826-2.003798]	[7.963786-7.969534]	[0.157898-0.158316]	[0.157273-0.157734]
6	[2.997312-3.004090]	[8.295369-8.301826]	[0.123962-0.124520]	[0.123718-0.124377]
8	[3.997718-4.007576]	[8.513796-8.519628]	[0.101765-0.102478]	[0.101530-0.102139]
10	[4.992197-5.004662]	[8.671315-8.677391]	[0.085683-0.086214]	[0.085832-0.086404]
20	[9.972440-10.014938]	[9.057575-9.062671]	[0.046694-0.047278]	[0.046738-0.047374]
50	[24.838313-25.101621]	[9.325538-9.330566]	[0.019532-0.020052]	[0.019431-0.020049]

Κεφάλαιο 4 Συμπεράσματα

Παρατηρώντας προσεκτικά τα αποτελέσματα που εξάγονται από τους πειραματικούς ελέγχους διαπιστώνονται οι αλλαγές και οι μεταβολές που υπόκειται το σύστημα διαφοροποιώντας τις βασικές παραμέτρους του.

Στην περίπτωση όπου οι χρόνοι επεξεργασίας βλαβών και επιδιορθώσεων γεννιούνται από την εκθετική κατανομή εύκολα διαπιστώνεται ότι η αύξηση της χωρητικότητας του ενδιάμεσου αποθηκευτικού χώρου αυξάνει την απόδοση του συστήματος. Αυτό συμβαίνει αφού στην ουσία αυξάνεται ο διαθέσιμος χώρος προς αποθήκευση και οι μηχανές τους συστήματος έχουν την δυνατότητα να επεξεργάζονται περισσότερη ώρα τα προς επεξεργασία προϊόντα. Ελαττώνονται τα ποσοστά όπου η πρώτη μηχανή είναι μπλοκαρισμένη και η δεύτερη μηχανή πεινάει.

Επίσης με την αύξηση του ρυθμού των επιδιορθώσεων η απόδοση τους συστήματος αυξάνεται. Αυτό συμβαίνει μιας και τα ποσοστά μπλοκαρίσματος και πείνας περιορίζονται όσο αυξάνονται οι ρυθμοί επιδιορθώσεων.

Το αντίθετο συμβαίνει με την αύξηση των ρυθμών βλαβών στις μηχανές. Σε αυτή την περίπτωση εφόσον έχουμε περισσότερες βλάβες τα ποσοστά μπλοκαρίσματος και πείνας αυξάνονται.

Ιδιαίτερα σημαντικό είναι ότι με την ταυτόχρονη αύξηση των ρυθμών επιδιόρθωσης αλλά και βλαβών παρατηρείται αύξηση της απόδοσης. Αυτό δείχνει ότι είναι προτιμότερο να έχουμε περισσότερες βλάβες μικρής διάρκειας και γρήγορης επιδιόρθωσης παρά μεγάλης διάρκειας με αργό χρόνο επιδιόρθωσης.

Με την αύξηση του ρυθμού παραγωγικότητας της δεύτερης μηχανής παρατηρείται η αύξηση της απόδοσης. Προφανώς η αύξηση του ρυθμού στην δεύτερη μηχανή μειώνει τα ποσοστά μπλοκαρίσματος στην πρώτη μηχανή επομένως επεξεργάζεται περισσότερο χρόνο προϊόντα και άρα αυξάνει την απόδοση μιας και η συνολική παραγωγικότητα επί της ουσίας είναι η συνολική παραγωγικότητα της δεύτερης μηχανής.

Τέλος η χρησιμοποίηση της ομοιόμορφης κατανομής αντί της εκθετικής επηρεάζει θετικά το σύστημα λόγω της μικρότερης μεταβλητότητας σε σχέση με την εκθετική κατανομή.

Βιβλιογραφία

- [1] Yves Dallery, Manufacturing flow lines systems: a review of models and analytical results 8 (1992)
- [2] T.Altiok and S.Stidham, Jr, A note on transfer lines with unreliable machines, random processing times, and infinite buffers, IIE Trans. 14(1982) 125-127
- [3] M.H. Ammar, Modeling and analysis of unreliable manufacturing assembly networks with finite storages, Technical Report LIDS-TH-1004, MIT Laboratory for Information and Decision Systems(1980)
- [4] T.Altiok, Approximate analysis of exponential tandem queues with blocking, Europ. J. Oper. Res 11 (1982)

ΠΑΡΑΡΤΗΜΑ

```

1 #include <stdio.h>
2 #include <assert.h>
3 #include <ctype.h>
4 #include <errno.h>
5 #include <float.h>
6 #include <limits.h>
7 #include <locale.h>
8 #include <math.h>
9 #include <setjmp.h>
10 #include <signal.h>
11 #include <stddef.h>
12 #include <stdarg.h>
13 #include <string.h>
14 #include <stdlib.h>
15 #include <sys/errno.h>
16 #include <sys/signal.h>
17 #include <time.h>
18
19 double p1,p2,f1,f2,r1,r2;
20 double tend,tsim,tmin;
21 int a,n,N,b,s,cases;
22 double tp1,tp2,tf1,tf2,tr1,tr2;
23
24 double *CUMCASE,*CUMBUF;
25 int CUMTH;
26 double AVERBUF, AVERTH, PB, PS;
27 double sumAVERBUF, sumAVERTH, sumPB, sumPS;
28 double *LAVERBUF, *LAVERTH, *LPB, *LPS;
29
30 double randomize(double l) {
31     double y=drand48();
32     double x=-log(1-y)/l;
33     return x;
34 }
35
36 inline void mintwo(double a, double b, double *t, int *temp) {
37     if(a<b) {
38         *t=a;
39         *temp=1;
40     }
41     else {
42         *t=b;
43         *temp=2;
44     }
45 }
46
47 inline void minthree(double a, double b,
48     double c, double *t, int *temp) {
49     if(a<b && a<c) {
50         *t=a;
51         *temp=1;
52     }
53     else if(b<c) {
54         *t=b;
55         *temp=2;
56     }
57     else {
58         *t=c;
59         *temp=3;
60     }
61 }
62
63 inline void minfour(double a, double b,
64     double c, double d, double *t, int *temp) {
65     if(a<b && a<c && a<d) {
66         *t=a;
67         *temp=1;

```

```

68     }
69     else if(b<c && b<d) {
70         *t=b;
71         *temp=2;
72     }
73     else if(c<d) {
74         *t=c;
75         *temp=3;
76     }
77     else {
78         *t=d;
79         *temp=4;
80     }
81 }
82
83 void configure(int i, int j, double t)
84 {
85     CUMCASE[i]+=t;
86     CUMBUF[j]+=t;
87 }
88
89 void selection() {
90     int temp=0;
91
92     //CASE #1
93     if (a==3 && n>=0 && n<=N && b==0 && s==0)
94     {
95         minfour(tp1,tp2,tf1,tf2,&tmin,&temp);
96         configure(0,n,tmin);
97         switch(temp)
98         {
99             case 1:
100            {
101                if(n==N)
102                    b=1;
103                else
104                    n++;
105                tp1=randomize(p1);
106                tp2-=tmin;
107                tf1-=tmin;
108                tf2-=tmin;
109                break;
110            }
111            case 2:
112            {
113                CUMTH++;
114                if(n==0)
115                    s=1;
116                else
117                    n--;
118                tp2=randomize(p2);
119                tp1-=tmin;
120                tf1-=tmin;
121                tf2-=tmin;
122                break;
123            }
124            case 3:
125            {
126                a=1;
127                tr1=randomize(r1);
128                tp1-=tmin;
129                tp2-=tmin;
130                tf2-=tmin;
131                break;
132            }
133            case 4:
134            {

```

```

135         a=2;
136         tr2=randomize(r2);
137         tp1-=tmin;
138         tp2-=tmin;
139         tf1-=tmin;
140         break;
141     }
142 }
143 }
144
145 //CASE #2
146 else if (a==1 && n>=0 && n<=N && b==0 && s==0)
147 {
148     minthree(tp2,tr1,tf2,&tmin,&temp);
149     configure(1,n,tmin);
150     switch(temp)
151     {
152         case 1:
153         {
154             CUMTH++;
155             if(n==0)
156                 s=1;
157             else
158                 n--;
159             tp2=randomize(p2);
160             tr1-=tmin;
161             tf2-=tmin;
162             break;
163         }
164         case 2:
165         {
166             a=3;
167             tf1=randomize(f1);
168             tp2-=tmin;
169             tf2-=tmin;
170             break;
171         }
172         case 3:
173         {
174             a=0;
175             tr2=randomize(r2);
176             tp2-=tmin;
177             tr1-=tmin;
178             break;
179         }
180     }
181 }
182
183 //CASE #3
184 else if (a==2 && n>=0 && n<=N && b==0 && s==0)
185 {
186     minthree(tp1,tf1,tr2,&tmin,&temp);
187     configure(2,n,tmin);
188     switch(temp)
189     {
190         case 1:
191         {
192             if(n==N)
193                 b=1;
194             else
195                 n++;
196             tp1=randomize(p1);
197             tf1-=tmin;
198             tr2-=tmin;
199             break;
200         }
201         case 2:

```

```

202     {
203         a=0;
204         tr1=randomize(r1);
205         tp1-=tmin;
206         tr2-=tmin;
207         break;
208     }
209     case 3:
210     {
211         a=3;
212         tf2=randomize(f2);
213         tp1-=tmin;
214         tf1-=tmin;
215         break;
216     }
217 }
218 }
219
220 //CASE #4
221 else if (a==0 && n>=0 && n<=N && b==0 && s==0)
222 {
223     mintwo(tr1,tr2,&tmin,&temp);
224     configure(3,n,tmin);
225     switch(temp)
226     {
227         case 1:
228         {
229             a=2;
230             tf1=randomize(f1);
231             tr2-=tmin;
232             break;
233         }
234         case 2:
235         {
236             a=1;
237             tf2=randomize(f2);
238             tr1-=tmin;
239             break;
240         }
241     }
242 }
243
244 //CASE #5
245 else if (a==3 && n==N && b==1 && s==0)
246 {
247     minthree(tp2,tf1,tf2,&tmin,&temp);
248     configure(4,n,tmin);
249     switch(temp)
250     {
251         case 1:
252         {
253             CUMTH++;
254             b=0;
255             tp1=randomize(p1);
256             tp2=randomize(p2);
257             tf1-=tmin;
258             tf2-=tmin;
259             break;
260         }
261         case 2:
262         {
263             a=1;
264             tr1=randomize(r1);
265             tp2-=tmin;
266             tf2-=tmin;
267             break;
268         }

```

```

269         case 3:
270         {
271             a=2;
272             tr2=randomize(r2);
273             tp2-=tmin;
274             tf1-=tmin;
275             break;
276         }
277     }
278 }
279
280 //CASE #6
281 else if (a==1 && n==N && b==1 && s==0)
282 {
283     minthree(tp2,tr1,tf2,&tmin,&temp);
284     configure(5,n,tmin);
285     switch(temp)
286     {
287         case 1:
288         {
289             CUMTH++;
290             b=0;
291             tp2=randomize(p2);
292             tr1-=tmin;
293             tf2-=tmin;;
294             break;
295         }
296         case 2:
297         {
298             a=3;
299             tf1=randomize(f1);
300             tp2-=tmin;
301             tf2-=tmin;
302             break;
303         }
304         case 3:
305         {
306             a=0;
307             tr2=randomize(r2);
308             tp2-=tmin;
309             tr1-=tmin;
310             break;
311         }
312     }
313 }
314
315 //CASE #7
316 else if (a==2 && n==N && b==1 && s==0)
317 {
318     mintwo(tf1,tr2,&tmin,&temp);
319     configure(6,n,tmin);
320     switch(temp)
321     {
322         case 1:
323         {
324             a=0;
325             tr1=randomize(r1);
326             tr2-=tmin;
327             break;
328         }
329         case 2:
330         {
331             a=3;
332             tf2=randomize(f2);
333             tf1-=tmin;
334             break;
335         }

```

```

336     }
337 }
338
339 //CASE #8
340 else if (a==0 && n==N && b==1 && s==0)
341 {
342     mintwo(tr1,tr2,&tmin,&temp);
343     configure(7,n,tmin);
344     switch(temp)
345     {
346         case 1:
347         {
348             a=2;
349             tf1=randomize(f1);
350             tr2--tmin;
351             break;
352         }
353         case 2:
354         {
355             a=1;
356             tf2=randomize(f2);
357             tr1--tmin;
358             break;
359         }
360     }
361 }
362
363 //CASE #9
364 else if (a==3 && n==0 && b==0 && s==1)
365 {
366     minthree(tp1,tf1,tf2,&tmin,&temp);
367     configure(8,n,tmin);
368     switch(temp)
369     {
370         case 1:
371         {
372             s=0;
373             tp1=randomize(p1);
374             tp2=randomize(p2);
375             tf1--tmin;
376             tf2--tmin;
377             break;
378         }
379         case 2:
380         {
381             a=1;
382             tr1=randomize(r1);
383             tp1--tmin;
384             tf2--tmin;
385             break;
386         }
387         case 3:
388         {
389             a=2;
390             tr2=randomize(r2);
391             tp1--tmin;
392             tf1--tmin;
393             break;
394         }
395     }
396 }
397
398 //CASE #10
399 else if (a==1 && n==0 && b==0 && s==1)
400 {
401     mintwo(tr1,tf2,&tmin,&temp);
402     configure(9,n,tmin);

```



```

403     switch(temp)
404     {
405         case 1:
406         {
407             a=3;
408             tf1=randomize(f1);
409             tf2-=tmin;
410             break;
411         }
412         case 2:
413         {
414             a=0;
415             tr2=randomize(r2);
416             tr1-=tmin;
417             break;
418         }
419     }
420 }
421
422 //CASE #11
423 else if (a==2 && n==0 && b==0 && s==1)
424 {
425     minthree(tp1,tf1,tr2,&tmin,&temp);
426     configure(10,n,tmin);
427     switch(temp)
428     {
429         case 1:
430         {
431             s=0;
432             tp1=randomize(p1);
433             tf1-=tmin;
434             tr2-=tmin;
435             break;
436         }
437         case 2:
438         {
439             a=0;
440             tr1=randomize(r1);
441             tp1-=tmin;
442             tr2-=tmin;
443             break;
444         }
445         case 3:
446         {
447             a=3;
448             tf2=randomize(f2);
449             tp1-=tmin;
450             tf1-=tmin;
451             break;
452         }
453     }
454 }
455
456 //CASE #12
457 else if (a==0 && n==0 && b==0 && s==1)
458 {
459     mintwo(tr1,tr2,&tmin,&temp);
460     configure(11,n,tmin);
461     switch(temp)
462     {
463         case 1:
464         {
465             a=2;
466             tf1=randomize(f1);
467             tr2-=tmin;
468             break;
469     }

```

```

470         case 2:
471         {
472             a=1;
473             tf2=randomize(f2);
474             tr1-=tmin;
475             break;
476         }
477     }
478 }
479
480 else
481 {
482     temp=0;
483     printf("-----\n");
484     printf("ERROR: NO CASE\n");
485     printf("-----\n");
486 }
487 }
488
489 int main(void) {
490
491     //EXTERNAL INITIALIZATION
492     int setup1=1;
493     int setup2=1;
494     tend=100000;//total simulation time
495     cases=12;//possible cases
496     int inloops=30;//internal loops per setup
497     int j=0;//counter for internal loops
498     int exloops=27;//external loops
499     int counter=0;//counter for external loops
500     int temp1=setup1;
501     int temp2=setup2;
502
503     //FILE
504     FILE *file;
505     file = fopen("file.csv","w");
506
507     for(counter=0;counter<exloops;counter++)
508     {
509         //CHANGE BUFFER SIZE
510         if (setup1==1)
511         {
512             p1=10;//p1 exp
513             p2=10;//p2 exp
514             f1=0.1;//f1 exp
515             f2=0.1;//f2 exp
516             r1=2;//r1 exp
517             r2=2;//r2 exp
518             N=10;//max buffer size
519
520             temp1=setup1;
521
522             if (setup2==1)
523             {
524                 N=0;
525                 temp2=setup2;
526                 setup2++;
527             }
528             else if (setup2==2)
529             {
530                 N=2;
531                 temp2=setup2;
532                 setup2++;
533             }
534             else if (setup2==3)
535             {
536                 N=4;

```

```

537         temp2=setup2;
538         setup2++;
539     }
540     else if (setup2==4)
541     {
542         N=6;
543         temp2=setup2;
544         setup2++;
545     }
546     else if (setup2==5)
547     {
548         N=8;
549         temp2=setup2;
550         setup2++;
551     }
552     else if (setup2==6)
553     {
554         N=10;
555         temp2=setup2;
556         setup2++;
557     }
558     else if (setup2==7)
559     {
560         N=20;
561         temp2=setup2;
562         setup2++;
563     }
564     else if (setup2==8)
565     {
566         N=50;
567         temp2=setup2;
568         setup1=2;
569         setup2=1;
570     }
571 }
572
573 //CHANGE R1, R2
574 else if (setup1==2)
575 {
576     p1=10;//p1 exp
577     p2=10;//p2 exp
578     f1=0.1;//f1 exp
579     f2=0.1;//f2 exp
580     r1=2;//r1 exp
581     r2=2;//r2 exp
582     N=10;//max buffer size
583
584     temp1=setup1;
585
586     if (setup2==1)
587     {
588         r1=12;
589         r2=12;
590         temp2=setup2;
591         setup2++;
592     }
593     else if (setup2==2)
594     {
595         r1=6;
596         r2=6;
597         temp2=setup2;
598         setup2++;
599     }
600     else if (setup2==3)
601     {
602         r1=2;
603         r2=2;

```

```

604         temp2=setup2;
605         setup2++;
606     }
607     else if (setup2==4)
608     {
609         r1=1;
610         r2=1;
611         temp2=setup2;
612         setup2++;
613     }
614     else if (setup2==5)
615     {
616         r1=0.5;
617         r2=0.5;
618         temp2=setup2;
619         setup1=3;
620         setup2=1;
621     }
622 }
623 //CHANGE F1, F2
624 else if (setup1==3)
625 {
626     p1=10;//p1 exp
627     p2=10;//p2 exp
628     f1=0.1;//f1 exp
629     f2=0.1;//f2 exp
630     r1=2;//r1 exp
631     r2=2;//r2 exp
632     N=10;//max buffer size
633
634     temp1=setup1;
635
636     if (setup2==1)
637     {
638         f1=0.05;
639         f2=0.05;
640         temp2=setup2;
641         setup2++;
642     }
643     else if (setup2==2)
644     {
645         f1=0.1;
646         f2=0.1;
647         temp2=setup2;
648         setup2++;
649     }
650     else if (setup2==3)
651     {
652         f1=0.2;
653         f2=0.2;
654         temp2=setup2;
655         setup2++;
656     }
657     else if (setup2==4)
658     {
659         f1=0.4;
660         f2=0.4;
661         temp2=setup2;
662         setup1=4;
663         setup2=1;
664     }
665 }
666 }
667 //CHANGE F1, F2, R1, R2
668 else if (setup1==4)
669 {
670

```

```

671 p1=10;//p1 exp
672 p2=10;//p2 exp
673 f1=0.1;//f1 exp
674 f2=0.1;//f2 exp
675 r1=2;//r1 exp
676 r2=2;//r2 exp
677 N=10;//max buffer size
678
679 temp1=setup1;
680
681 if (setup2==1)
682 {
683     f1=0.025;
684     f2=0.025;
685     r1=0.5;
686     r2=0.5;
687     temp2=setup2;
688     setup2++;
689 }
690 else if (setup2==2)
691 {
692     f1=0.05;
693     f2=0.05;
694     r1=1;
695     r2=1;
696     temp2=setup2;
697     setup2++;
698 }
699 else if (setup2==3)
700 {
701     f1=0.1;
702     f2=0.1;
703     r1=2;
704     r2=2;
705     temp2=setup2;
706     setup2++;
707 }
708 else if (setup2==4)
709 {
710     f1=0.3;
711     f2=0.3;
712     r1=6;
713     r2=6;
714     temp2=setup2;
715     setup2++;
716 }
717 else if (setup2==5)
718 {
719     f1=0.6;
720     f2=0.6;
721     r1=12;
722     r2=12;
723     temp2=setup2;
724     setup1=5;
725     setup2=1;
726 }
727 }
728
729 //CHANGE P2
730 else if (setup1==5)
731 {
732     p1=10;//p1 exp
733     p2=10;//p2 exp
734     f1=0.1;//f1 exp
735     f2=0.1;//f2 exp
736     r1=2;//r1 exp
737     r2=2;//r2 exp

```

```

738         N=10;//max buffer size
739
740         temp1=setup1;
741
742         if (setup2==1)
743         {
744             p2=8;
745             temp2=setup2;
746             setup2++;
747         }
748         else if (setup2==2)
749         {
750             p2=9;
751             temp2=setup2;
752             setup2++;
753         }
754         else if (setup2==3)
755         {
756             p2=10;
757             temp2=setup2;
758             setup2++;
759         }
760         else if (setup2==4)
761         {
762             p2=11;
763             temp2=setup2;
764             setup2++;
765         }
766         else if (setup2==5)
767         {
768             p2=12;
769             temp2=setup2;
770             setup1=6;
771             setup2=1;
772         }
773     }
774
775     CUMCASE=(double*) malloc(cases*sizeof(double));
776     CUMBUF=(double*) malloc((N+1)*sizeof(double));
777     \AVERBUF=(double*) malloc(inloops*sizeof(double));
778     \AVERTH=(double*) malloc(inloops*sizeof(double));
779     \PB=(double*) malloc(inloops*sizeof(double));
780     \PS=(double*) malloc(inloops*sizeof(double));
781
782     int k;
783     for(k=0;k<inloops;k++)
784     {
785         \AVERBUF[k]=0;
786         \AVERTH[k]=0;
787         \PB[k]=0;
788         \PS[k]=0;
789     }
790
791     sumAVERBUF=0;
792     sumAVERTH=0;
793     sumPB=0;
794     sumPS=0;
795
796     fprintf(file, "%s", "SETUP ");
797     fprintf(file, "%d", temp1);
798     fprintf(file, "%s", ".");
799     fprintf(file, "%d", temp2);
800     fprintf(file, "%s", ",");
801     fprintf(file, "%s", "P1:");
802     fprintf(file, "%f", p1);
803     fprintf(file, "%s", ",");
804     fprintf(file, "%s", "P2:");

```

```

805     fprintf(file, "%f", p2);
806     fprintf(file, "%s", ",");
807     fprintf(file, "%s", "F1:");
808     fprintf(file, "%f", f1);
809     fprintf(file, "%s", ",");
810     fprintf(file, "%s", "F2:");
811     fprintf(file, "%f", f2);
812     fprintf(file, "%s", ",");
813     fprintf(file, "%s", "R1:");
814     fprintf(file, "%f", r1);
815     fprintf(file, "%s", ",");
816     fprintf(file, "%s", "R2:");
817     fprintf(file, "%f", r2);
818     fprintf(file, "%s", ",");
819     fprintf(file, "%s", "N:");
820     fprintf(file, "%d", N);
821     fprintf(file, "%s", ",");
822     fprintf(file, "%s", "\n\n");
823
824     fprintf(file, "%s", ", AVERBUF,");
825     fprintf(file, "%s", "AVERTH,");
826     fprintf(file, "%s", "PB,");
827     fprintf(file, "%s", "PS,");
828     fprintf(file, "%s", "\n\n");
829
830     for(j=0;j<inloops;j++)
831     {
832         //INTERNAL INITIALIZATION
833         tsim=0;//current time
834         tmin=0;//min time
835
836         a=3;//machines state
837         n=0;//current buffer size
838         b=0;//unblocked/blocked
839         s=1;//unstarved/starved
840
841         srand48(j+1);
842
843         //time while 1st machine produces 1 product
844         tp1=randomize(p1);
845
846         //time while 2nd machine produces 1 product
847         tp2=0;
848
849         //time while 1st machine down
850         tf1=randomize(f1);
851
852         //time while 2nd machine down
853         tf2=randomize(f2);
854
855         //time while 1st machine fix
856         tr1=0;
857
858         //time while 2nd machine fix
859         tr2=0;
860
861         int i;
862         for(i=0;i<cases;i++)
863             CUMCASE[i]=0;
864         for(i=0;i<=N;i++)
865             CUMBUF[i]=0;
866
867         CUMTH=0;
868         AVERBUF=0;
869         AVERTH=0;
870
871         //MAIN LOOP

```



```

872     do
873     {
874         selection();
875         tsim+=tmin;
876     }
877     while(tsim<=tend);
878
879     //OUTPUT ANALYSIS
880     for(i=0;i<cases;i++)
881         CUMCASE[i]=CUMCASE[i]/tsim;
882
883
884     for(i=0;i<=N;i++)
885     {
886         CUMBUF[i]=CUMBUF[i]/tsim;
887         AVERBUF=AVERBUF+i*CUMBUF[i];
888     }
889
890     AVERTH=CUMTH/tsim;
891
892     PB = CUMCASE[4] + CUMCASE[5] + CUMCASE[6] + CUMCASE[7];
893     PS = CUMCASE[8] + CUMCASE[9] + CUMCASE[10] + CUMCASE[11];
894
895     \AVERBUF[j]=AVERBUF;
896     \AVERTH[j]=AVERTH;
897     \PB[j]=PB;
898     \PS[j]=PS;
899     sumAVERBUF+=AVERBUF;
900     sumAVERTH+=AVERTH;
901     sumPB+=PB;
902     sumPS+=PS;
903
904     fprintf(file, "%s", ",");
905     fprintf(file, "%f", AVERBUF);
906     fprintf(file, "%s", ",");
907     printf("      AVERBUF: %f\n\n", AVERBUF);
908
909     fprintf(file, "%f", AVERTH);
910     fprintf(file, "%s", ",");
911     printf("      AVERTH: %f\n\n", AVERTH);
912
913     fprintf(file, "%f", PB);
914     fprintf(file, "%s", ",");
915     printf("      PB: %f\n\n", PB);
916
917     fprintf(file, "%f", PS);
918     fprintf(file, "%s", ",");
919     fprintf(file, "%s", "\n");
920     printf("      PS: %f\n\n", PS);
921
922 }
923
924 int l;
925 double sum\AVERBUF=0;
926 double sum\AVERTH=0;
927 double sum\PB=0;
928 double sum\PS=0;
929 for (l=0;l<inloops;l++)
930 {
931     \AVERBUF[l]=powf((\AVERBUF[l]-(sumAVERBUF/inloops)),2);
932     sum\AVERBUF+=\AVERBUF[l];
933     \AVERTH[l]=powf((\AVERTH[l]-(sumAVERTH/inloops)),2);
934     sum\AVERTH+=\AVERTH[l];
935     \PB[l]=powf((\PB[l]-(sumPB/inloops)),2);
936     sum\PB+=\PB[l];
937     \PS[l]=powf((\PS[l]-(sumPS/inloops)),2);
938     sum\PS+=\PS[l];

```


```

939     }
940     sumLAVERBUF=sqrtf(sumLAVERBUF/inloops);
941     sumLAVERTH=sqrtf(sumLAVERTH/inloops);
942     sumLPB=sqrtf(sumLPB/inloops);
943     sumLPS=sqrtf(sumLPS/inloops);
944
945     fprintf(file, "%s", "MEAN,");
946     fprintf(file, "%f", sumAVERBUF/inloops);
947     fprintf(file, "%s", ",");
948     fprintf(file, "%f", sumAVERTH/inloops);
949     fprintf(file, "%s", ",");
950     fprintf(file, "%f", sumPB/inloops);
951     fprintf(file, "%s", ",");
952     fprintf(file, "%f", sumPS/inloops);
953     fprintf(file, "%s", ",\n");
954
955     fprintf(file, "%s", "STD,");
956     fprintf(file, "%f", sumLAVERBUF);
957     fprintf(file, "%s", ",");
958     fprintf(file, "%f", sumLAVERTH);
959     fprintf(file, "%s", ",");
960     fprintf(file, "%f", sumLPB);
961     fprintf(file, "%s", ",");
962     fprintf(file, "%f", sumLPS);
963     fprintf(file, "%s", ",\n");
964
965     fprintf(file, "%s", "TS,");
966     fprintf(file, "%s", "[");
967     fprintf(file, "%f", (sumAVERBUF/inloops)
968         - (1.96*sumLAVERBUF/sqrt(30)));
969     fprintf(file, "%s", "-");
970     fprintf(file, "%f", (sumAVERBUF/inloops)
971         + (1.96*sumLAVERBUF/sqrt(30)));
972     fprintf(file, "%s", "],");
973     fprintf(file, "%s", "[");
974     fprintf(file, "%f", (sumAVERTH/inloops)
975         - (1.96*sumLAVERTH/sqrt(30)));
976     fprintf(file, "%s", "-");
977     fprintf(file, "%f", (sumAVERTH/inloops)
978         + (1.96*sumLAVERTH/sqrt(30)));
979     fprintf(file, "%s", "],");
980     fprintf(file, "%s", "[");
981     fprintf(file, "%f", (sumPB/inloops) - (1.96*sumLPB/sqrt(30)));
982     fprintf(file, "%s", "-");
983     fprintf(file, "%f", (sumPB/inloops) + (1.96*sumLPB/sqrt(30)));
984     fprintf(file, "%s", "],");
985     fprintf(file, "%s", "[");
986     fprintf(file, "%f", (sumPS/inloops) - (1.96*sumLPS/sqrt(30)));
987     fprintf(file, "%s", "-");
988     fprintf(file, "%f", (sumPS/inloops) + (1.96*sumLPS/sqrt(30)));
989     fprintf(file, "%s", "],\n");
990
991     fprintf(file, "%s", "\n\n");
992
993 }
994
995 fclose(file);
996
997 return 0;
998
999 }
1000 }
1001

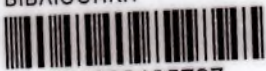
```

<p>Κραίτας Ιωάννης.</p>	
<p>ΣΥΓΓΡΑΦΕΑΣ</p>	
<p>Προβλήματα βυθιτικής παραγωγής</p>	
<p>ΤΙΤΛΟΣ</p>	
<p>DE SUD IONASIONICTES MACHINES EN GARA UTA XURO E. I. ANTEGUS ΑΠΟΣΤΟΛΕΥΣ</p>	
ΛΗΞΗ	ΟΝΟΜΑΤΕΠΩΝΥΜΟ ΔΑΝΕΙΖΟΜΕΝΟΥ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ
Τηλ.: 24210 06300




ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΒΙΒΛΙΟΘΗΚΗ



004000105797