ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Διπλωματική Εργασία

ΕΥΡΕΣΗ ΒΕΑΤΙΣΤΗΣ ΠΟΛΙΤΙΚΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΕΥΕΛΙΚΤΟΥ ΕΞΥΠΗΡΕΤΗΤΗ ΣΕ ΣΥΣΤΗΜΑ ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΔΕΜΕΝΩΝ ΣΤΑΘΜΩΝ ΕΡΓΑΣΙΑΣ

υπό

ΙΩΑΝΝΗ ΜΠΑΣΟΥΚΟΥ

Υπεβλήθη για την εκπλήρωση μέρους των

απαιτήσεων για την απόκτηση του

Διπλώματος Μηχανολόγου Μηχανικού

2010

Η έγκριση της διπλωματικής 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. Δημήτριος Παντελής Επίκουρος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Δεύτερος Εξεταστής	Δρ. Γεώργιος Λυμπερόπουλος Καθηγητής, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
Τρίτος Εξεταστής	Δρ. Γεώργιος Κοζανίδης Λέκτορας, Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα, θέλω να ευχαριστήσω τον επιβλέποντα της διπλωματικής εργασίας μου, Επίκουρο Καθηγητή κ. Δημήτριο Παντελή, για την πολύτιμη βοήθεια και καθοδήγησή του κατά τη διάρκεια της δουλειάς μου. Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της διπλωματικής εργασίας μου, Καθηγητές κκ. Γεώργιο Κοζανίδη και Γεώργιο Λυμπερόπουλο. Ευχαριστώ τους φίλους(ες) μου για την ηθική υποστήριξή τους. Επίσης, οφείλω ευχαριστίες στον αδελφό μου Μάριο Μπασούκο για τον πολύτιμο χρόνο που μου αφιέρωσε, τη βοήθεια και τις συμβουλές που μου έδινε κατά τη διάρκεια της διπλωματικής μου εργασίας. Πάνω απ' όλα, είμαι ευγνώμων στους γονείς μου, Κων/νο και Κων/να Μπασούκου για την ολόψυχη αγάπη και υποστήριξή τους όλα αυτά τα χρόνια. Αφιερώνω αυτή την εργασία στην μητέρα μου και στον πατέρα μου.

Ιωάννης Μπασούκος

ΕΥΡΕΣΗ ΒΕΑΤΙΣΤΗΣ ΠΟΛΙΤΙΚΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΕΥΕΛΙΚΤΟΥ ΕΞΥΠΗΡΕΤΗΤΗ ΣΕ ΣΥΣΤΗΜΑ ΕΝ ΣΕΙΡΑ ΣΥΝΔΕΔΕΜΕΝΩΝ ΣΤΑΘΜΩΝ ΕΡΓΑΣΙΑΣ

ΙΩΑΝΝΗΣ ΜΠΑΣΟΥΚΟΣ

Πανεπιστήμιο Θεσσαλίας, Τμήμα Μηχανολόγων Μηχανικών, 2010

Επιβλέπων Καθηγητής: Δρ. Δημήτριος Παντελής, Επίκουρος Καθηγητής Στοχαστικών Προτύπων Επιχειρησιακής Έρευνας στη Βιομηχανική Διοίκηση

Περίληψη

Τα συστήματα συνδεδεμένων σταθμών εργασίας σε συνδυασμό εφαρμόζονται ευρύτατα σε διάφορες περιοχές που έχουν σχέση με επεξεργασία όπου ευέλικτοι εξυπηρετητές χρησιμοποιούνται ανάλογα με την ζήτηση και τις συνθήκες λειτουργίας.

Σε αυτήν την διπλωματική εργασία καταλήγουμε στην βέλτιστη πολιτική λειτουργίας ενός ευέλικτου εξυπηρετητή. Ιδιαίτερα, θα μας απασχολήσει το θέμα της εν σειρά συνδεδεμένων δύο σταθμών εργασίας, όπου ο αριθμός των εργασιών που βρίσκονται κατά την έναρξη λειτουργίας του συστήματος είναι σταθερός και δεν επηρεάζεται από εξωτερικές αφίξεις.

Αρχικά, διατυπώνουμε το πρόβλημα και αναπτύσσουμε τη μέθοδο, η οποία βασίζεται στην Μαρκοβιανή Θεωρία.

Στη συνέχεια, με τη βοήθεια της γλώσσας προγραμματισμού Fortran επιλύουμε το πρόβλημα και παίρνουμε τα αποτελέσματα.

Τέλος, εφαρμόζουμε την μέθοδο επίλυσης ώστε να καλύπτει περισσότερες περιπτώσεις και το επιλύουμε για δεδομένα διαφορετικού μεγέθους. Κατόπιν σύγκρισης των εξαγόμενων αποτελεσμάτων προκύπτει το συμπέρασμα ότι η μέθοδος λειτουργεί σωστά και αποτελεσματικά, παρέχει τη βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή και επομένως είναι δυνατή η μελλοντική εφαρμογή της σε συστήματα εν σειρά συνδεδεμένων σταθμών εργασίας.

Πίνακας Περιεχομένων

Κατάλογος Πινάκων

Κατάλογος Διαγραμμάτων

Κεφάλαιο 1 Εισαγωγή

Σε αυτό το κεφάλαιο, παρουσιάζουμε πληροφορίες εισαγωγικού χαρακτήρα που δίνουν το κίνητρο και το υπόβαθρο αυτής της διπλωματικής εργασίας, παραθέτουμε μια ανασκόπηση της σχετικής, με την εργασία, βιβλιογραφίας και περιγράφουμε συνοπτικά τις βασικές ενότητες της διπλωματικής εργασίας.

1.1 Κίνητρο και Υπόβαθρο

Τα συστήματα που αποτελούνται από δύο σταθμούς εργασίας εν σειρά συνδεδεμένους και χρησιμοποιούν ευέλικτους εξυπηρετητές είναι συστήματα τα οποία χρησιμοποιούνται σε βιομηχανικές εφαρμογές, όπου οι ευέλικτοι πόροι (επαναπρογραμματισμός μηχανών, πολλαπλά καταρτισμένοι εργαζόμενοι – cross-trained workers) χρησιμοποιούνται κατά περίπτωση, ανάλογα με την ζήτηση και τις συνθήκες λειτουργίας, με σκοπό την βελτίωση της απόδοσης των εν λόγω συστημάτων.

Ο υπολογισμός της πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και η εύρεση της βέλτιστης αυτής, είναι πρωτεύουσας σημασίας για τον σχεδιασμό και την λειτουργία ενός συστήματος με δύο εν σειρά συνδεδεμένους σταθμούς εργασίας και έχει ως γενικότερο αποτέλεσμα την συνολική βελτιστοποίηση της απόδοσης του εν λόγω συστήματος.

Η δυσκολία εύρεσης της βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή έγκειται στο ότι για αρκετά μεγάλο αριθμό εργασιών και στους δύο σταθμούς εμφανίζονται σφάλματα. Τα σφάλματα οφείλονται σε δύο αιτίες: την αρχιτεκτονική του υπολογιστή, στον οποίο εκτελείται κάθε φόρα ο αλγόριθμος που μας δίνει την βέλτιστη πολιτική, και σε διάφορες προγραμματιστικές αδυναμίες, οι οποίες σχετίζονται με την σειρά των πράξεων, την στρογγυλοποίηση των αριθμών, την διάδοση των σφαλμάτων κλπ.

Η συνεισφορά αυτής της διπλωματικής εργασίας είναι ότι μας επιβεβαιώνει πλήρως τα θεωρητικά – μαθηματικά αποτελέσματα τα οποία προκύπτουν μέσα από την διεθνή βιβλιογραφία και μας παρέχει μια ολοκληρωμένη εικόνα για την βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή.

1.2 Βιβλιογραφική Ανασκόπηση

Υπάρχουν αρκετές μελέτες και αναφορές στην διεθνή βιβλιογραφία πάνω σε συστήματα που αποτελούνται από δύο σταθμούς εργασίας εν σειρά συνδεδεμένους και χρησιμοποιούν ευέλικτους εξυπηρετητές. Οι Narongwanich et al. [16] εξέτασαν τη βέλτιστη κατανομή επενδύσεων μεταξύ αφοσιωμένων και αναπροσαρμόσιμων συστήματων για επιχειρήσεις που παράγουν προϊόντα, τα οποία έχουν τυχαία ζήτηση και σε συστήματα όπου πραγματοποιείται η νέα παραγωγή προϊόντων σε τυχαίο μελλοντικό χρόνο. Οι Hopp and Van Oyen [13] παρέχουν ένα πλαίσιο για την αιτιολόγηση του εργατικού δυναμικού cross-training, σε έναν οργανισμό. Τεκμηρίωσαν την σχέση μεταξύ cross-training σε έναν οργανισμό, τις επιδιώξεις αυτού και συζητήθηκαν οι κατάλληλες μέθοδοι για τη συγκεκριμένη εφαρμογή. Μοντέλα για εν σειρά συστήματα παραγωγής με σκοπό τη μεγιστοποίηση της απόδοσης αναλύθηκαν από τους Van Oyen et al. [23], Andradottiret al. [4], [5], [6], Gel et al. [10], [11], Hopp et al. [12], Iravani et al., [15], and Ahn and Righter [3]. Τα συγκεκριμένα μοντέλα περιλαμβάνουν συνεργατικούς και μη- συνεργατικούς τομείς εργασίας, ανοικτά συστήματα με εξωτερικές αφίξεις που είναι ανεξάρτητα από την κατάσταση του συστήματος, κλειστά συστήματα (π.χ., συστήματα CONWIP όπου μια αναχώρηση, δημιουργεί μια νέα άφιξη), καθώς και διάφορες μορφές της πολλαπλής κατάρτισης, όπως για παράδειγμα πλήρης, ζώνης ή ιεραρχικής πολλαπλή κατάρτιση.

Υπάρχει έντονο ερευνητικό ενδιαφέρον και για την βέλτιστη χρήση των ευέλικτων εξυπηρετητών σε εν σειρά συστήματα στα οποία υπάρχει κόστος παραμονής. Επειδή τα μαθηματικά μοντέλα που χρησιμοποιούνται είναι αρκετά περίπλοκα, τα περισσότερα αποτελέσματα περιλαμβάνουν συστήματα δύο σταδίων και εκθετικούς χρόνους εξυπηρέτησης. Οι Rosberg et al. [20] θεωρούν ένα σύστημα που οι αφίξεις του ακολουθούν την κατανομή Poisson, έναν εξυπηρετητή που έχει έναν σταθερό ρυθμό εξυπηρέτησης στον σταθμό που λαμβάνει δεδομένα-downstream, και έναν εξυπηρετητή με έναν ελεγχόμενο ρυθμό εξυπηρέτησης στο σταθμό εργασίας που στέλνει δεδομένα-upstream. Απέδειξαν ότι ο βέλτιστος ρυθμός εξυπηρέτησης είναι μη-μειούμενος καθ' όλη την διάρκεια της πρώτης ουράς και μη-αυξανόμενος καθ' όλη την διάρκεια της δεύτερης ουράς. Παράλληλα όμως, οι Weber and Stidham [24] θεώρησαν ένα σύστημα το οποίο αποτελείται από n σταθμούς με αφίξεις σε κάθε σταθμό, όπου εκτός από τα κυρτά κόστη αναμονής, οι εξυπηρετητές επιβαρύνονται με λειτουργικά κόστη που είναι κυρτές συναρτήσεις των ρυθμών τους εξυπηρέτησης. Επίσης, έδειξαν ότι η βέλτιστη πολιτική είναι γνησίως μονότονη. Δηλαδή, όταν μια εργασία αποχωρεί από μια ουρά, ο βέλτιστος ρυθμός εξυπηρέτησης στην εν λόγω ουρά δεν αυξάνεται και παράλληλα ο βέλτιστος ρυθμός εξυπηρέτησης στις υπόλοιπες ουρές, δεν μειώνεται. Οι Duenyas et al. [7] καθώς επίσης και οι Iravani et al. [14], καθόρισαν τις βέλτιστες πολιτικές για μοντέλα με n και δύο φάσεων συστήματα, αντίστοιχα, με έναν ευέλικτο εξυπηρετητή και κόστη προετοιμασίας. Οι Pandelis and Teneketzis [19] μελέτησαν συστήματα δύο σταδίων χωρίς αφίξεις, αλλά με πολλαπλούς ευέλικτους εξυπηρετητές, όπου οι εργασίες είχαν ενταχθεί στην δεύτερη ουρά με πιθανότητα p, αφού ολοκληρώθηκε η επεξεργασία τους από τον upstream σταθμό. Καθόρισαν τις συνθήκες υπό τις οποίες, η πολιτική που δίνει προτεραιότητα στις εργασίες στον upstream σταθμό είναι η βέλτιστη, για γενικούς χρόνους εξυπηρέτησης. Για ένα σύστημα δύο σταδίων, χωρίς αφίξεις, το οποίο έχει δύο ευέλικτους εξυπηρετητές, οι Ahn et al. [2], παρέθεσαν τις ικανές και αναγκαίες συνθήκες υπό τις οποίες μια πολιτική προτεραιότητας για τον upstream ή τον downstream σταθμό είναι βέλτιστη. Παρόμοια αποτελέσματα ελήφθησαν και από τους Ahn et al. [1] για το μοντέλο με αφίξεις. Τα αποτελέσματα των Ahn et al. [2] έχουν επεκταθεί σε δύο ακόμα κατευθύνσεις. Πρώτον, οι Schiefermayr and Weichbold [21] προσδιορίζουν τη βέλτιστη πολιτική για όλες τις τιμές του κόστους αναμονής και τους χρόνους εξυπηρέτησης. Δεύτερον, οι Weichbold and Schiefermayr [25], καθόρισαν τις συνθήκες για τη βελτιστότητα των αυστηρών πολιτικών, όταν οι εργασίες χρειάζονται τη δεύτερη φάση εξυπηρέτησης με μια ορισμένη πιθανότητα.

Με την εξαίρεση των Rosberg et al. [20] όπου υπάρχει ένα αφοσιωμένος εξυπηρετητής στον downstream σταθμό, τα προαναφερθέντα άρθρα έχουν θεωρήσει ότι δεν υπάρχουν τέτοιοι εξυπηρετητές. Ο Farrar [8], [9] μελέτησε δύο εκδοχές ενός συστήματος χωρίς αφίξεις, με αφοσιωμένους εξυπηρετητές και στους δύο σταθμούς εργασίας και έναν ευέλικτο εξυπηρετητή. Η εκδοχή στην οποία ισχύει αυτός ο περιορισμός, ο ευέλικτος εξυπηρετητής μπορεί μόνο να εξυπηρετεί τον upstream σταθμό, ενώ στην εκδοχή όπου δεν ισχύει αυτός ο περιορισμός, ο ευέλικτος εξυπηρετητής μπορεί να εξυπηρετεί και τους δύο σταθμούς. Και για τις δύο εκδοχές, ο Farrar έδειξε ότι η βέλτιστη πολιτική χαρακτηρίζεται από μία εναλλασσόμενη καμπύλη. Ο ευέλικτος εξυπηρετητής είναι ανενεργός ή αφοσιωμένος στον downstream σταθμό, όταν ο αριθμός των εργασιών εκεί υπερβαίνει ένα ορισμένο όριο που εξαρτάται από τον αριθμό των εργασιών στην πρώτη ουρά. Επίσης, έδειξε ότι η κλίση της καμπύλης εναλλαγής είναι μεγαλύτερη ή ίση με -1. Ο Pandelis [17] έδειξε την εγκυρότητα των αποτελεσμάτων που λαμβάνονται από τον Farrar [8], [9], για την περίπτωση όπου οι εργασίες μπορεί να αποχωρήσουν από το σύστημα αφού έχουν εξυπηρετηθεί από τον πρώτο σταθμό. Επίσης, καθόρισε τα υποσύνολα του χώρου καταστάσεων όπου η βέλτιστη πολιτική μπορεί να καθοριστεί λεπτομερώς. Στην ίδια δομή για την εύρεση της βέλτιστη πολιτικής κατέληξαν και οι Wu et al. [27] για ένα σύστημα που αποτελείται από μια σειρά από ευέλικτους εξυπηρετητές, με το πρόσθετο χαρακτηριστικό, ότι όλοι οι εξυπηρετητές υπόκεινται σε βλάβες. Στο μοντέλο τους, υπέθεσαν ότι οι απαιτήσεις επεξεργασίας είναι ίδιες και στους δύο σταθμούς, με αποτέλεσμα οι πραγματικοί χρόνοι επεξεργασίας να εξαρτώνται από τις ταχύτητες των εξυπηρετητών. Οι Wu et al. [26] έδειξαν την βελτιστότητα μιας γνησίως μονότονης πολιτικής, για το προηγούμενο μοντέλο, με αφίξεις και χωρίς αφοσιωμένους εξυπηρετητές στον upstream σταθμό.

Τέλος, ο Pandelis [18] μελέτησε και ανέπτυξε ένα μαθηματικό μοντέλο για την εύρεση του ελάχιστου αναμενόμενου συνολικού κόστους αναμονής και λειτουργίας ενός συστήματος το οποίο αποτελείται από δύο σταθμούς εργασίας, εν σειρά συνδεδεμένων, με δύο αφοσιωμένους εξυπηρετητές και έναν ευέλικτο εξυπηρετητή, με κόστη λειτουργίας.

1.3 Οργάνωση Διπλωματικής Εργασίας

Το υπόλοιπο αυτής της διπλωματικής εργασίας χωρίζεται σε τέσσερα κεφάλαια. Συγκεκριμένα:

Το Κεφάλαιο 2 περιλαμβάνει την ανάλυση για την εύρεση της βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή σε σύστημα εν σειρά δύο σταθμών εργασίας και διατυπώνεται μαθηματικά.

Στο Κεφάλαιο 3 γίνεται αναφορά στη διαδικασία παραγωγής των αποτελεσμάτων με την βοήθεια της γλώσσας προγραμματισμού Fortran. Επίσης, παραθέτονται τα αποτελέσματα της επίλυσης του προβλήματος και τα συμπεράσματα που προκύπτουν από αυτά.

Στο 0 παρουσιάζονται τα τελικά συμπεράσματα που αποκομίστηκαν από την διπλωματική εργασία και κατευθύνσεις για περαιτέρω έρευνα.

Τέλος, στα Παραρτήματα παραθέτουμε το υπολογιστικό μοντέλο όπως ακριβώς χρησιμοποιείται από την Fortran καθώς και κάποια δεδομένα με τα αποτελέσματα τους.

Κεφάλαιο 2 Ανάλυση του Προβλήματος και Ανάπτυξη του Μαθηματικού Μοντέλου Επίλυσής του

Στο κεφάλαιο αυτό αναλύουμε το πρόβλημα για την εύρεση της βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή σε σύστημα εν σειρά δύο σταθμών εργασίας και αναπτύσσουμε τα κατάλληλα Μαθηματικά Μοντέλα που το διέπουν με σκοπό την επίλυσή του.

2.1 Μαθηματική διατύπωση του προβλήματος

Θεωρούμε ένα σύστημα που αποτελείται από δύο σταθμούς εργασίας εν σειρά συνδεδεμένων. Ο αριθμός των εργασιών που υπάρχουν στο σύστημα είναι συγκεκριμένος και δεν μεταβάλλεται από εξωτερικές αφίξεις. Οι εργασίες που ολοκληρώνονται στον upstream (σταθμός 1) μεταφέρονται στον downstream σταθμό (σταθμός 2) όπου υπόκεινται και σε άλλη επεξεργασία και τέλος φεύγουν από το σύστημα. Κάθε εργασία απαιτεί έναν εκθετικά κατανεμημένο χρόνο αποπεράτωσης σε κάθε σταθμό. Υπάρχουν εξυπηρετητές που διατίθενται αποκλειστικά για τον κάθε ένα σταθμό εργασίας, ξεχωριστά, με ρυθμούς λειτουργίας μ_1 και μ_2. Αυτοί οι εξυπηρετητές προβλέπεται να λειτουργούν όταν υπάρχουν εργασίες σε κάθε έναν από τους αντίστοιχους σταθμούς εργασίας. Επίσης, υπάρχει ένας επιπλέον εξυπηρετητής-ευέλικτος, ο οποίος μπορεί να εξυπηρετήσει και τους δύο σταθμούς εργασίας. Υποθέτουμε ότι αυτός ο εξυπηρετητής μπορεί να μετακινηθεί από την εξυπηρέτηση του ενός σταθμού εργασίας στον άλλον στιγμιαία χωρίς κανένα επιπρόσθετο κόστος και λειτουργεί με ρυθμούς ν_1 και ν_2 στο σταθμό 1 και 2 αντίστοιχα.

Κάθε εργασία στον σταθμό εργασίας i, i=1, 2, επιβαρύνεται με ένα γραμμικό κόστος παραμονής h_i. Επιπλέον, ο ευέλικτος εξυπηρετητής επιβαρύνεται με κόστη c_1 και c_2, κατά την διάρκεια του χρόνου λειτουργίας του, στους σταθμούς εργασίας 1 και 2, αντίστοιχα. Ο αντικειμενικός μας στόχος, είναι να ευρεθεί μία στρατηγική χρήσης του ευέλικτου εξυπηρετητή, η οποία θα ελαχιστοποιεί τα αναμενόμενα συνολικά κόστη αναμονής και λειτουργίας, έως ότου το σύστημα αποδεσμευτεί από όλες τις εργασίες. Επιτρέποντας την «προκατοχή» - preemption, στους χρόνους ολοκλήρωσης της υπηρεσίας και θεωρώντας πως η εργασία που υπήρχε προ της ολοκλήρωσης της υπηρεσίας χάθηκε, μορφοποιούμε το πρόβλημα σαν μια διαδικασία λήψης απόφασης τύπου Markov με χώρο καταστάσεων το $\{(x_1, x_2): x_1, x_2 > 0\}$, όπου x_1 και x_2 είναι ο αριθμός εργασιών στον σταθμό εργασίας 1 και 2 αντίστοιχα, περιλαμβανομένων και των εργασιών που εξυπηρετούνται. Ξεκινώντας από την κατάσταση (x_1, x_2), θεωρούμε ότι $V(x_1, x_2)$ είναι το ελάχιστο αναμενόμενο συνολικό κόστος παραμονής και λειτουργίας με το οποίο επιβαρύνεται το σύστημα, μέχρι αυτό να αδειάσει, με $V(0,0)=0$.

Έχουμε θεωρήσει ότι οι εξυπηρετητές στον ίδιο σταθμό εργασίας μπορούν και συνεργάζονται μεταξύ τους πάνω στην ίδια εργασία και επομένως οι χρόνοι επεξεργασίας είναι προσθετικοί.

Για να μετατρέψουμε το πρόβλημα, από πρόβλημα συνεχούς χρόνου στο ισοδύναμο του πρόβλημα διακριτού χρόνου εφαρμόζουμε μια κανονικοποίηση (Serfozo [22]). Για την εξαγωγή των εξισώσεων, έχουμε θεωρήσει ότι η σχέση κανονικοποίησης είναι $\mu_1 + \mu_2 + \nu_1 + \nu_2 = 1$.

Με βάση τις υποθέσεις που παραθέσαμε και αναλύσαμε παραπάνω οι βέλτιστες εξισώσεις που μας δίνουν το ελάχιστο κόστος σε κάθε περίπτωση είναι οι ακόλουθες.

Για $x_1,\ x_2 > 1$:

$$V(x_1,\ x_2) = h_1x_1 + h_2x_2 + \mu_1 V(x_1 - 1,\ x_2 + 1) + \mu_2 V(x_1,\ x_2 - 1)$$
$$+\min\{(\nu_1+\nu_2)V(x_1,\ x_2), c_1 + \nu_1 V(x_1 - 1,\ x_2 + 1) + \nu_2 V(x_1,\ x_2),$$
$$c_2 + \nu_2 V(x_1,\ x_2 - 1) + \nu_1 V(x_1,\ x_2)\}, \qquad (Εξ.\ 1)$$

Ο πρώτος όρος στην ελαχιστοποίηση αντιστοιχεί με το να κρατάμε ανενεργό τον ευέλικτο εξυπηρετητή. Ο δεύτερος όρος αντιστοιχεί στην ανάθεση του ευέλικτου εξυπηρετητή στον σταθμό εργασίας 1, ενώ ο τρίτος στον σταθμό εργασίας 2.

Για $x_1 > 1,\ x_2 = 0$:

/

$$V(x_1,\ 0) = \min\left\{\frac{h_1x_1}{\mu_1},\ \frac{h_1x_1 + c_1}{\mu_1 + \nu_1}\right\} + V(x_1 - 1,\ 1), \qquad (Εξ.\ 2)$$

Για $x_1 = 0,\ x_2 > 1$:

$$V(0,\ x_2) = \min\left\{\frac{h_2x_2}{\mu_2},\ \frac{h_2x_2 + c_2}{\mu_2 + \nu_2}\right\} + V(0,\ x_2 - 1), \qquad (Εξ.\ 3)$$

Θεωρούμε ότι το μέσο λειτουργικό κόστος για μια εργασία που εξυπηρετείται στον σταθμό εργασίας 2 από τον ευέλικτο εξυπηρετητή, δεν είναι περισσότερο από το αντίστοιχο μέσο κόστος παραμονής και λειτουργίας που επιβαρύνεται κατά την διάρκεια της εξυπηρέτησης της εργασίας από τον αποκλειστικά αφιερωμένο εξυπηρετητή.

Όλα τα παραπάνω διατυπώνονται μαθηματικά στην ακόλουθη Συνθήκη Α:

$$\frac{c_2}{\nu_2} \leq \frac{h_2}{\mu_2}$$

Έχει αποδειχθεί ότι υπό την Συνθήκη Α, το να κρατήσουμε ανενεργό τον ευέλικτο εξυπηρετητή δεν είναι βέλτιστο όταν υπάρχει τουλάχιστον μια εργασία στον σταθμό εργασίας 2 (Pandelis [18]). Στην ουσία η Συνθήκη Α είναι η συνθήκη κάτω από την οποία είναι βέλτιστο να χρησιμοποιήσουμε τον ευέλικτο εξυπηρετητή όταν δεν υπάρχουν εργασίες στον σταθμό εργασίας 1. Αυτό φαίνεται από το γεγονός ότι, όταν ο σταθμός εργασίας 1 ή 2 είναι άδειος από εργασίες, το να κρατάμε ανενεργό τον ευέλικτο εξυπηρετητή είναι βέλτιστο για $x_2 < X_2$ και $x_1 < X_1$ αντίστοιχα, όπου $X_2 = \frac{c_2\mu_2}{h_2\nu_2}$, $X_1 = \frac{c_1\mu_1}{h_1\nu_1}$. Δηλαδή, εάν η Συνθήκη Α ισχύει, το να κρατάμε ανενεργό τον ευέλικτο εξυπηρετητή δεν είναι βέλτιστο όταν $x_1 = 0$.

Θεωρώντας ότι η Συνθήκη Α ισχύει και χρησιμοποιώντας τις παραπάνω εξισώσεις, προκύπτει η βέλτιστη εξίσωση για το ελάχιστο κόστος.

Για $x_1,\ x_2 > 1$:

$$\begin{aligned} V(x_1,\ x_2) &= h_1x_1 + h_2x_2 + \mu_1V(x_1-1,\ x_2+1) + \mu_2V(x_1,\ x_2-1) \\ &+ \min\{c_2 + \nu_2V(x_1,\ x_2-1) + \nu_1V(x_1,\ x_2), \\ &c_1 + \nu_1V(x_1-1,\ x_2+1) + \nu_2V(x_1,\ x_2)\}, \end{aligned} \qquad (Εξ.\ 4)$$

Για $x_1 > 1,\ x_2 = 0$:

$$V(x_1,\ 0) = \min\left\{\frac{h_1 x_1}{\mu_1},\ \frac{h_1 x_1 + c_1}{\mu_1 + \nu_1}\right\} + V(x_1 - 1,\ 1), \qquad (Εξ.\ 5)$$

Για $x_1 = 0,\ x_2 > 1$:

$$V(0,\ x_2) = \frac{h_2 x_2 + c_2}{\mu_2 + \nu_2} + V(0,\ x_2 - 1), \qquad (Εξ.\ 6)$$

2.2 Συμπεράσματα

Στο κεφάλαιο αυτό μελετήσαμε το πρόβλημα για την εύρεση της βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή σε σύστημα εν σειρά δύο σταθμών εργασίας και καταλήξαμε στο Μαθηματικό Μοντέλο το οποίο θα χρησιμοποιήσουμε τελικά για την επίλυση του.

Κεφάλαιο 3 Αριθμητικά Αποτελέσματα

Στο κεφάλαιο αυτό παρουσιάζουμε δεδομένα για τα οποία έχει επιλυθεί το πρόβλημα καθώς και αριθμητικά αποτελέσματα από την εφαρμογή του αριθμητικού μοντέλου που έχουμε αναλύσει.

Τα αρχικά δεδομένα είναι τυχαία και για τις δύο περιπτώσεις, κατά τις οποίες είτε ισχύει, είτε δεν ισχύει η Συνθήκη Α, και στην συνέχεια τα αλλάζουμε κατά περίπτωση με σκοπό να παρατηρήσουμε τις όποιες πιθανές διαφοροποιήσεις θα προκύψουν.

Η εισαγωγή των δεδομένων που καθορίζει χρήστης γίνεται είτε απευθείας από το πληκτρολόγιο, είτε από ένα αρχείο, που δημιουργείται αυτόματα για τον συγκεκριμένο λόγο. Οι τιμές του αριθμού εργασιών σε κάθε σταθμό εργασίας x_1, x_2 εισέρχονται από τον χρήστη, καθώς επίσης και οι τιμές των μη κανονικοποιημένων μεταβλητών h_1, h_2, c_1, c_2, ν_1, ν_2, μ_1 και μ_2 οι οποίες βέβαια μπορεί να είναι τυχαίες ή συγκεκριμένες. Και στις δύο περιπτώσεις οι τιμές των μεταβλητών είναι διπλής ακρίβειας για την αποφυγή πιθανών σφαλμάτων. Οι τυχαίες τιμές των μεταβλητών προκύπτουν από μια γεννήτρια ψευδοτυχαίων αριθμών που είναι ενσωματωμένη στην Fortran 90/95, σύμφωνα με το διάστημα που καθορίζει χρήστης κάθε φορά.

Για τα συγκεκριμένα λοιπόν δεδομένα, το πρόγραμμα εκτελείται, και με βάση το Μαθηματικό Μοντέλο που έχουμε αναπτύξει στο Κεφάλαιο 2, προκύπτει η βέλτιστη πολιτική λειτουργίας για τον ευέλικτο εξυπηρετητή. Τα αποτελέσματα που προκύπτουν αποθηκεύονται σε τέσσερις διαφορετικούς φακέλους, οι οποίοι βρίσκονται στο εξής «μονοπάτι», *C:\output*.

- Ο φάκελος με το όνομα, *Output_Data*, περιέχει, σε ένα αρχείο *.txt*, τα δεδομένα, κανονικοποιημένα και μη, που χρησιμοποιούνται από το πρόγραμμά μας.
- Ο φάκελος με το όνομα, *Matrix_CH*, περιέχει, σε ένα αρχείο *.txt*, τα σημεία στα οποία ο ευέλικτος εξυπηρετητής αλλάζει πολιτική λειτουργίας για πρώτη φορά.
- Ο φάκελος με το όνομα, *Matrix_U*, περιέχει, σε ένα αρχείο *.xls*, για κάθε συνδυασμό αριθμού εργασιών, στους σταθμούς εργασίας, την αντίστοιχη βέλτιστη πολιτική του ευέλικτου εξυπηρετητή.
- Ο φάκελος με το όνομα, *Matrix_V*, περιέχει, σε ένα αρχείο *.txt*, για κάθε συνδυασμό αριθμού εργασιών, στους σταθμούς εργασίας το αντίστοιχο ελάχιστο αναμενόμενο συνολικό κόστος παραμονής και λειτουργίας με το οποίο επιβαρύνεται το υπό εξέταση σύστημα.

Εν συνεχεία, τα αποτελέσματα που προκύπτουν, είναι στην διάθεση του χρήστη με σκοπό την περαιτέρω ανάλυσή τους και την εξαγωγή χρήσιμων συμπερασμάτων μέσω της γραφικής αναπαράστασης αυτών. Η συγκεκριμένη διαδικασία ακολουθήθηκε και στην παρούσα διπλωματική εργασία και ορισμένα ενδεικτικά αποτελέσματα είναι τα ακόλουθα.

3.1 Αποτελέσματα όταν δεν ισχύει η Συνθήκη Α

Αρχικά θα εξετάσουμε το πρόβλημα για την περίπτωση, κατά την οποία, η Συνθήκη Α, η οποία αναλύθηκε στο Κεφάλαιο 2 δεν ισχύει. Ο αριθμός εργασιών σε κάθε σταθμό εργασίας είναι:

$x_1 = 20$ και $x_2 = 20$

Τα δεδομένα για κάθε μη κανονικοποιημένη μεταβλητή που εισάγονται το πρόγραμμα που εκτελούμε επιλέγονται τυχαία στα ακόλουθα διαστήματα:

$h_1 : [0.75,\ 0.77]$, $\quad h_2 : [2.6,\ 2.8]$

$c_1 : [7.53,\ 7.55]$, $\quad c_2 : [7.41,\ 7.43]$

$v_1 : [4.58,\ 4.6]$, $\quad v_2 : [4.30,\ 4.32]$

$\mu_1 : [2.6,\ 2.75]$, $\quad \mu_2 : [6.2,\ 6.4]$

Για κάθε μεταβλητή το πρόγραμμά μας επέλεξε τυχαία μια τιμή. Αυτές οι τιμές εμφανίζονται στο παρακάτω πίνακα, όπου οι μη-κανονικοποιημένες βρίσκονται στην αριστερή στήλη, ενώ οι κανονικοποιημένες στην δεξιά.

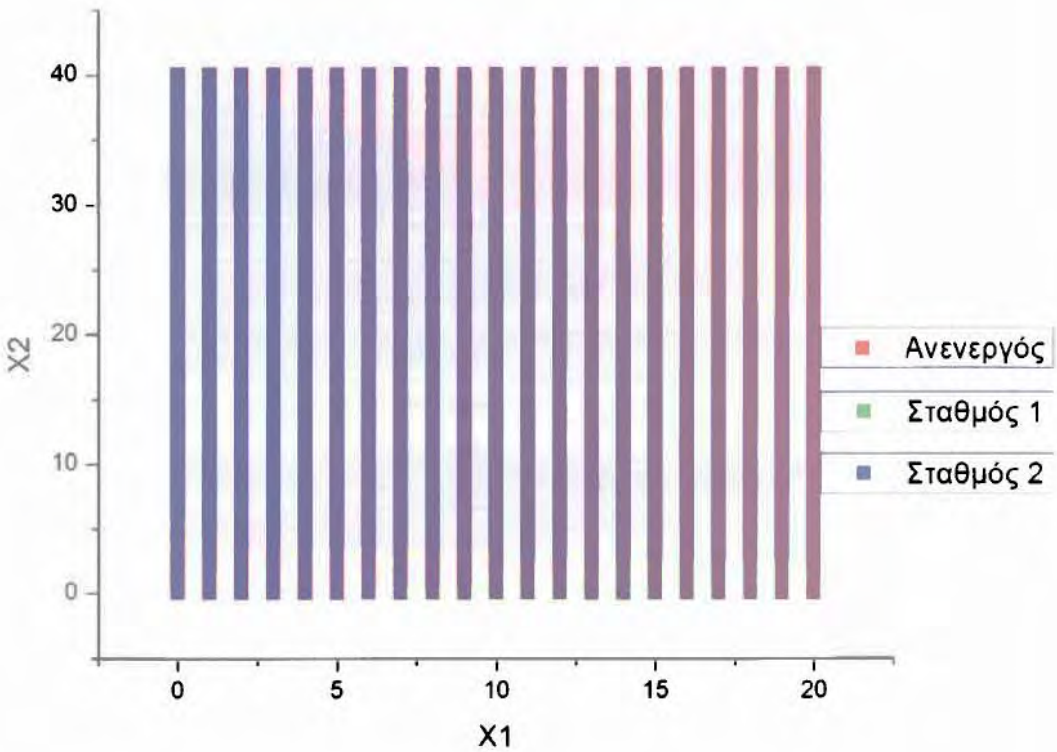
Μεταβλητές	**Μη-κανονικοποιημένες Τιμές**	**Κανονικοποιημένες Τιμές**
h_1	0.7690945	4.0090360E-02
h_2	2.608206	0.1359572
c_1	7.532834	0.3926618
c_2	7.427641	0.3871785
v_1	4.589133	0.2392164
v_2	4.301105	0.2242024
μ_1	2.677692	0.1395793
μ_2	7.616096	0.3970020

Πίνακας 3.1: Κανονικοποιημένες και μη-κανονικοποιημένες αρχικές τυχαίες τιμές για την περίπτωση που δεν ισχύει η Συνθήκη Α

Αναλυτικά όλα τα αριθμητικά αποτελέσματα για κάθε περίπτωση βρίσκονται στο Παράρτημα Ι. Στοχεύοντας στην περαιτέρω επεξεργασία των αποτελεσμάτων που προέκυψαν, θα προβούμε στην αναπαράσταση και ανάλυση τους σε δύο ειδών διαγράμματα.

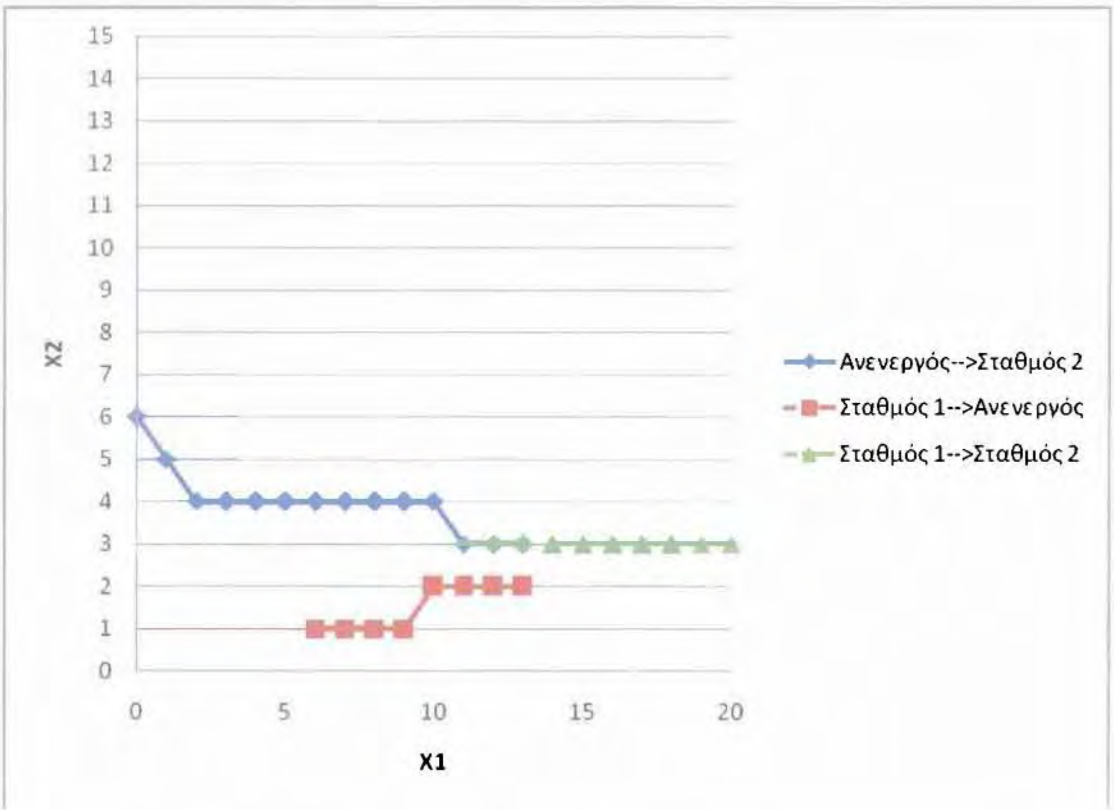
Στο πρώτο διάγραμμα συνοψίζεται η βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για όλους τους αριθμούς εργασιών που έχουμε ως δεδομένα. Αντίστοιχα, στο δεύτερο διάγραμμα αναπαριστώνται τα σημεία αλλαγής της βέλτιστης πολιτικής του ευέλικτου εξυπηρετητή.

Στο διάγραμμα 3.1 αναπαρίσταται η βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή, για τα αρχικά δεδομένα που χρησιμοποιήσαμε. Με το κόκκινο χρώμα υποδηλώνονται τα σημεία εκείνα στα οποία ο ευέλικτος εξυπηρετητής παραμένει ανενεργός. Με πράσινο χρώμα αναπαριστώνται τα σημεία λειτουργίας στον σταθμό εργασίας 1, ενώ με μπλε τα σημεία λειτουργίας στον σταθμό εργασίας 2.



Διάγραμμα 3. 1: Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή

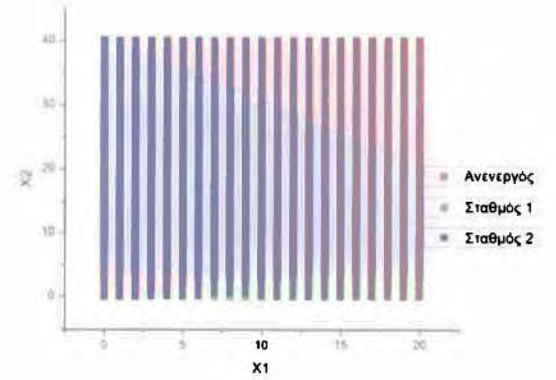
Στο διάγραμμα 3.2 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε σχέση με τον αριθμό εργασιών του σταθμού 1 και 2. Τα σημεία με μπλε χρώμα υποδηλώνουν την αλλαγή λειτουργίας του ευέλικτου εξυπηρετητή από την ανενεργή κατάσταση στην λειτουργία του στον σταθμό εργασίας 2. Αναλόγως, τα σημεία με κόκκινο χρώμα αντιστοιχούν στην αλλαγή λειτουργίας από τον σταθμό εργασίας 1 στην ανενεργή κατάσταση του ευέλικτου εξυπηρετητή. Καταλήγοντας, τα σημεία με πράσινο χρώμα αναπαριστούν την αλλαγή λειτουργίας από τον σταθμό εργασίας 1 στον σταθμό εργασίας 2.



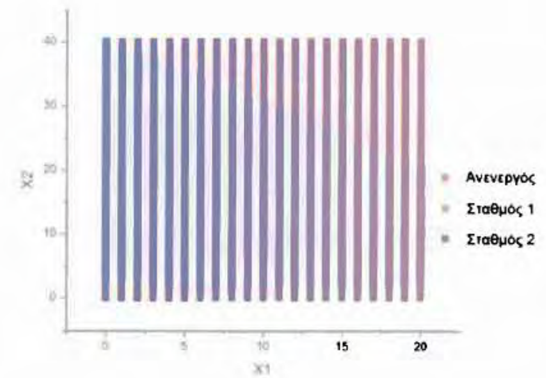
Διάγραμμα 3. 2:Διάγραμμα αλλαγής βέλτιστης απόφασης του ευέλικτου εξυπηρετητή

3.1.1 Αποτελέσματα όταν διαφοροποιείται το γραμμικό κόστος παραμονής κάθε εργασίας σε κάθε σταθμό

Όταν αυξάνεται το γραμμικό κόστος παραμονής h_i με το οποίο επιβαρύνεται κάθε εργασία στον σταθμό εργασίας, i, i=1, 2, τότε αυξάνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.3 και στο Διάγραμμα 3.4 όταν το γραμμικό κόστος παραμονής στον σταθμό εργασίας 1 και 2 είναι $h_1 = 1.5$ και $h_2 = 5$ αντίστοιχα.

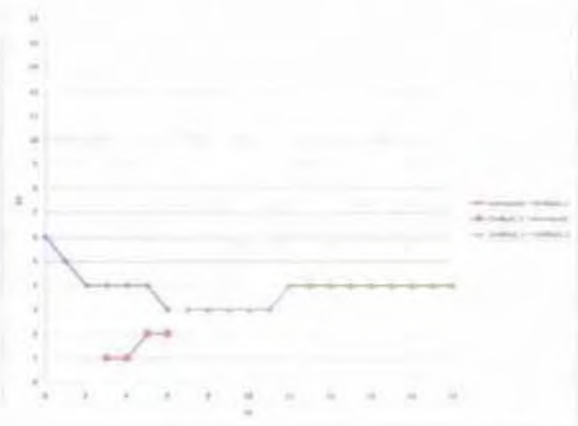


Διάγραμμα 3.3:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $h_1 = 1.5$

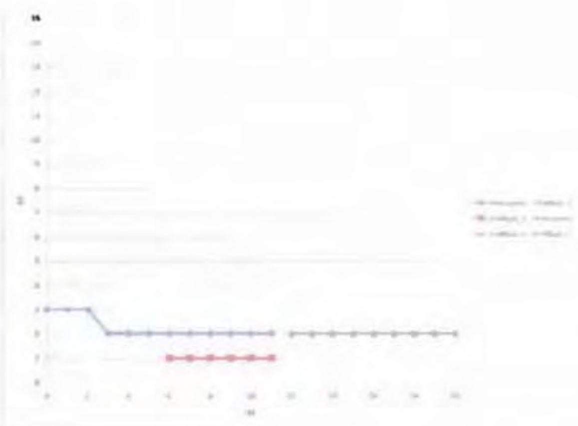


Διάγραμμα 3.4:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $h_2 = 5$

Στο Διάγραμμα 3.5 και στο Διάγραμμα 3.6 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τα νέα γραμμικά κόστη αναμονής στους αντίστοιχους σταθμούς εργασίας.



Διάγραμμα 3.5: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $h_1 = 1.5$

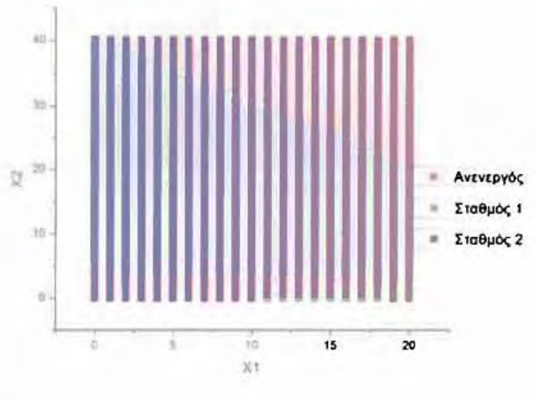


Διάγραμμα 3.6: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $h_2 = 5$

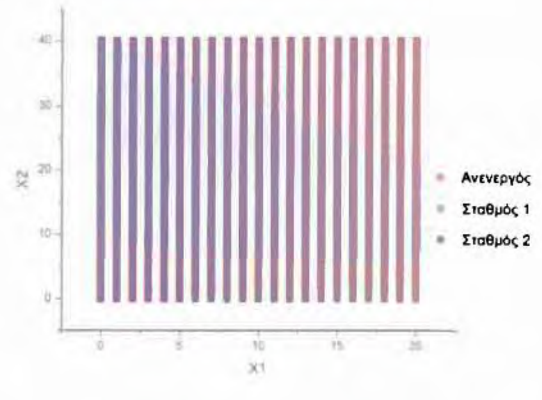
/

3.1.2 Αποτελέσματα όταν διαφοροποιείται το λειτουργικό κόστος του ευέλικτου εξυπηρετητή

Όταν αυξάνεται το λειτουργικό κόστος c_i με οποίο επιβαρύνεται ο ευέλικτος εξυπηρετητής κατά την διάρκεια του χρόνου λειτουργίας του στον σταθμό εργασίας i, i=1, 2, τότε μειώνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.7 και στο Διάγραμμα 3.8 όταν το λειτουργικό κόστος του ευέλικτου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $c_1 = 15$ και $c_2 = 15$ αντίστοιχα.



Διάγραμμα 3.7:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $c_1 = 15$



Διάγραμμα 3.8:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $c_2 = 15$

Στο Διάγραμμα 3.9 και στο Διάγραμμα 3.10 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τα νέα λειτουργικά κόστη του ευέλικτου εξυπηρετητή, στους αντίστοιχους σταθμούς εργασίας.

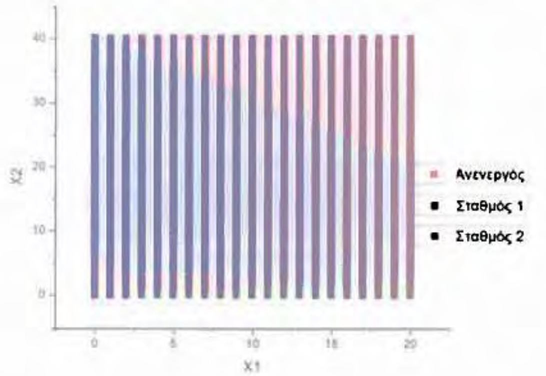
Διάγραμμα 3.9:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $c_1 = 15$

Διάγραμμα 3.10: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $c_2 = 15$

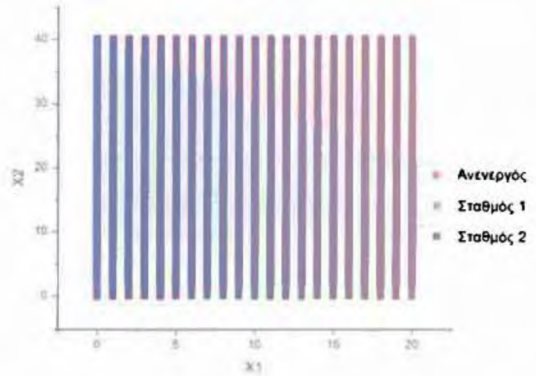
/

3.1.3 Αποτελέσματα όταν διαφοροποιείται ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή

Όταν αυξάνεται ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή ν_i στον σταθμό εργασίας i, i-1, 2, τότε αυξάνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.11 και στο Διάγραμμα 3.12 όταν ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $\nu_1 = 9$ και $\nu_2 = 10$ αντίστοιχα.

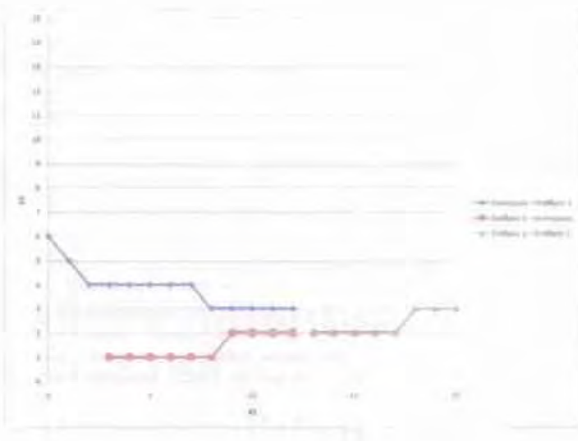


Διάγραμμα 3.11:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\nu_1 = 9$

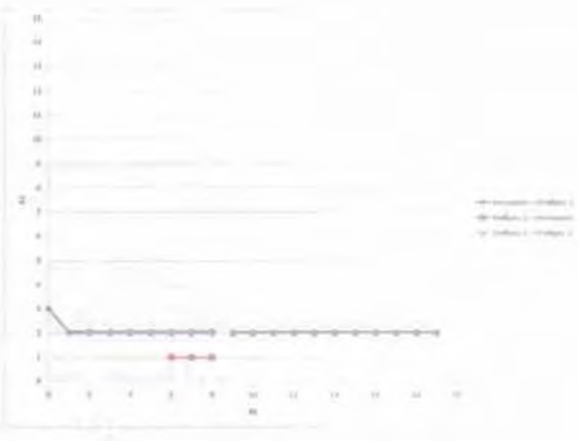


Διάγραμμα 3.12:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\nu_2 = 10$

Στο Διάγραμμα 3.13 και στο Διάγραμμα 3.14 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τους νέους ρυθμούς εργασίας του ευέλικτου εξυπηρετητή στους αντίστοιχους σταθμούς εργασίας.



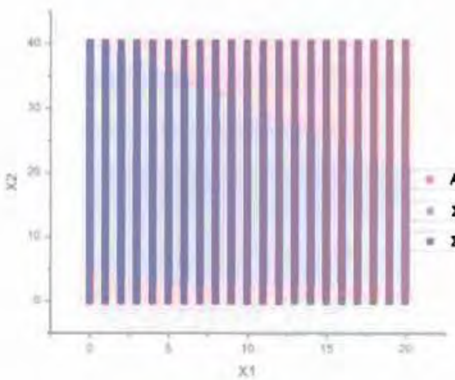
Διάγραμμα 3.13:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\nu_1 = 9$



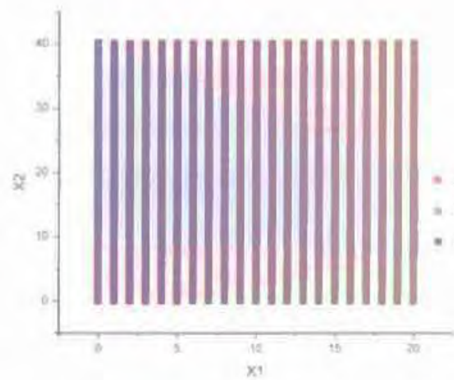
Διάγραμμα 3.14: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\nu_2 = 10$

3.1.4 Αποτελέσματα όταν διαφοροποιείται ο ρυθμός λειτουργίας του κάθε εξειδικευμένου εξυπηρετητή

Όταν αυξάνεται ο ρυθμός λειτουργίας του εξειδικευμένου εξυπηρετητή μ_i στον σταθμό εργασίας i, $i=1,2$, τότε μειώνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.15 και στο Διάγραμμα 3.16 όταν ο ρυθμός λειτουργίας του εξειδικευμένου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $\mu_1 = 6$ και $\mu_2 = 16$ αντίστοιχα.

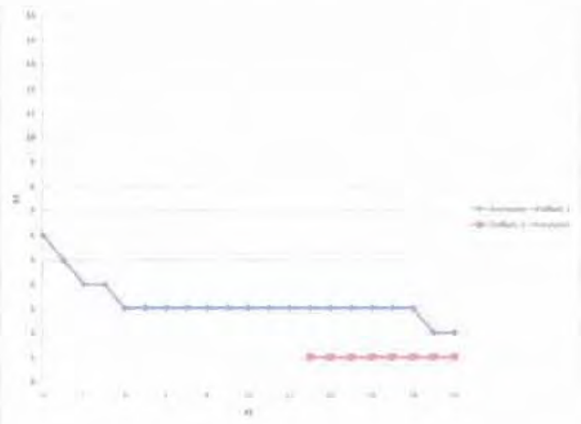


Διάγραμμα 3.15:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\mu_1 = 6$

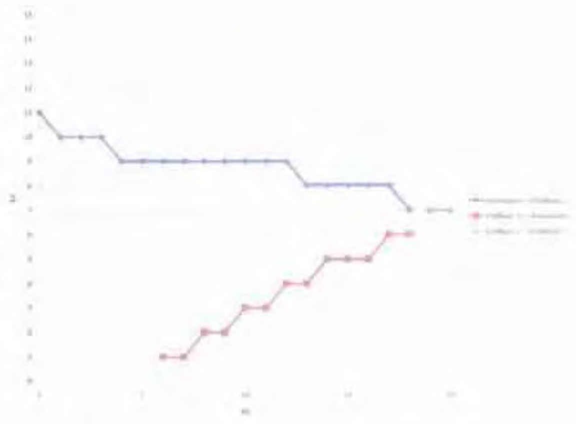


Διάγραμμα 3.16:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\mu_2 = 16$

Στο Διάγραμμα 3.17 και στο Διάγραμμα 3.18 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τους νέους ρυθμούς λειτουργίας κάθε εξειδικευμένου εξυπηρετητή στους αντίστοιχους σταθμούς εργασίας.



Διάγραμμα 3.17:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\mu_1 = 6$



Διάγραμμα 3.18: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\mu_2 = 16$

3.2 Αποτελέσματα όταν ισχύει η Συνθήκη Α

Θα εξετάσουμε το πρόβλημα για την περίπτωση, κατά την οποία, η Συνθήκη Α, η οποία αναλύθηκε στο Κεφάλαιο 2 ισχύει. Για την περίπτωση κατά την οποία ισχύει η Συνθήκη Α εκτελέσαμε το πρόγραμμά μας για 10,000 διαφορετικές επαναλήψεις (το πρόγραμμά μας έχει σχεδιαστεί για επταψήφιο αριθμό επαναλήψεων). Σε όλες τις περιπτώσεις η καμπύλη βέλτιστης απόφασης καμπύλη αλλαγής της βέλτιστης απόφασης του ευέλικτου εξυπηρετητή από τον σταθμό εργασίας 1, στον σταθμό εργασίας 2 είναι μη φθίνουσα, ενώ από την κατάσταση όπου ο ευέλικτος εξυπηρετητής παραμένει ανενεργός στον σταθμό 2 η καμπύλη βέλτιστης απόφασης του ευέλικτου εξυπηρετητή είναι σταθερή. Τα αποτελέσματα αυτά είναι εμφανή στα Διαγράμματα Αλλαγής Βέλτιστης Απόφασης του ευέλικτου εξυπηρετητή.

Ο αριθμός εργασιών σε κάθε σταθμό εργασίας είναι:

$x_1 = 20$ και $x_2 = 20$

Τα δεδομένα για κάθε μη κανονικοποιημένη μεταβλητή που εισάγονται το πρόγραμμα που εκτελούμε επιλέγονται τυχαία στα ακόλουθα διαστήματα:

$h_1 : [6.2,\ 6.5]$, $h_2 : [10,\ 10.2]$

$c_1 : [24,\ 24.2]$, $c_2 : [5.1,\ 5.25]$

$\nu_1 : [5.45,\ 5.5]$, $\nu_2 : [8.64,\ 8.68]$

$\mu_1 : [11.2,\ 11.35]$, $\mu_2 : [12.2,\ 12.4]$

Για κάθε μεταβλητή το πρόγραμμά μας επέλεξε τυχαία μια τιμή. Αυτές οι τιμές εμφανίζονται στο παρακάτω πίνακα, όπου οι μη-κανονικοποιημένες βρίσκονται στην αριστερή στήλη, ενώ οι κανονικοποιημένες στην δεξιά.

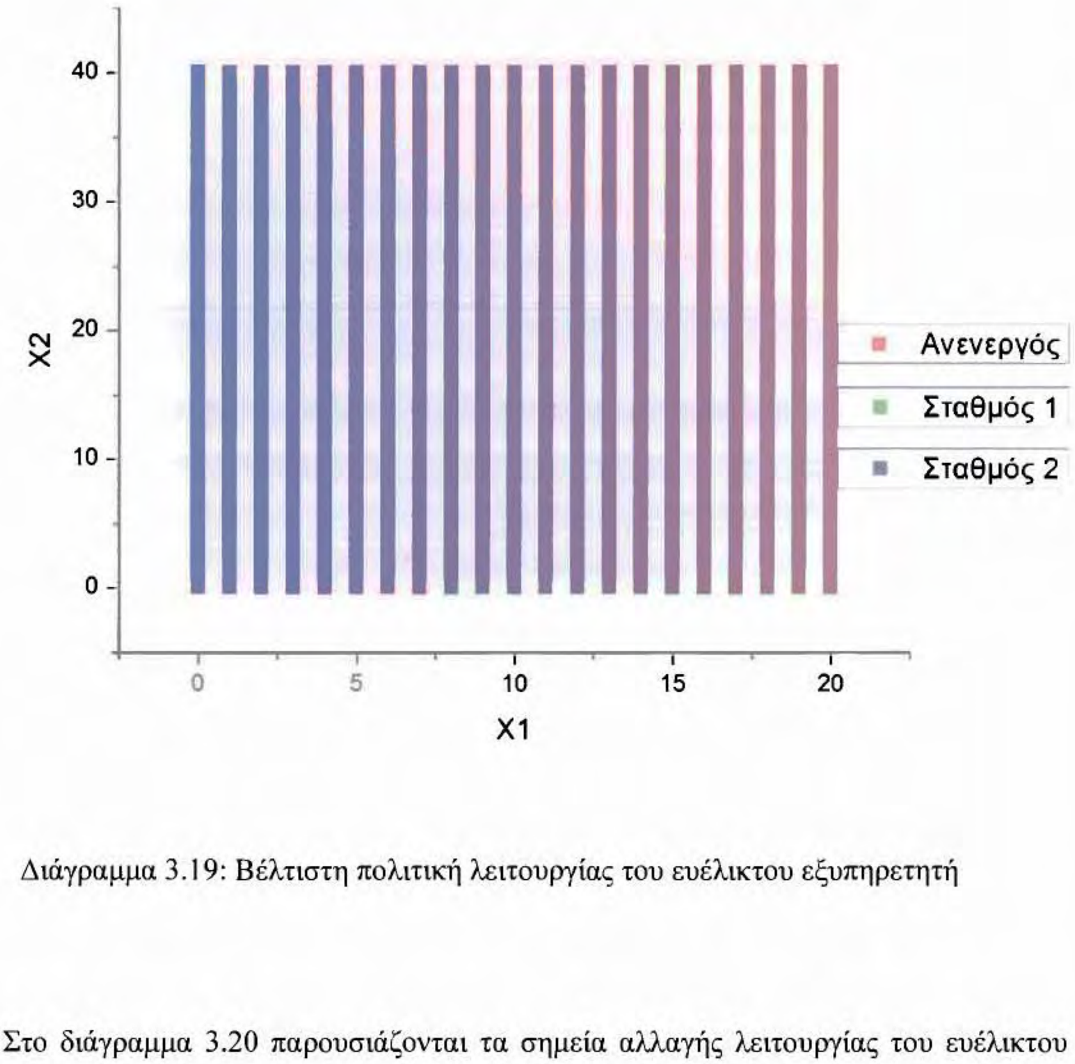
Μεταβλητές	**Μη-κανονικοποιημένες Τιμές**	**Κανονικοποιημένες Τιμές**
h_1	6.317509	0.1680287
h_2	10.18890	0.2709973
c_1	24.09972	0.6409876
c_2	5.180889	0.1377977
v_1	5.465436	0.1453658
v_2	8.663159	0.2304167
μ_1	11.22580	0.2985759
μ_2	12.24341	0.3256416

Πίνακας 3.2: Κανονικοποιημένες και μη-κανονικοποιημένες αρχικές τυχαίες τιμές για την περίπτωση που ισχύει η Συνθήκη Α

Αναλυτικά όλα τα αριθμητικά αποτελέσματα για κάθε περίπτωση βρίσκονται στο Παράρτημα Ι. Στοχεύοντας στην περαιτέρω επεξεργασία των αποτελεσμάτων που προέκυψαν, θα προβούμε στην αναπαράσταση και ανάλυση τους σε δύο ειδών διαγράμματα. Στο πρώτο διάγραμμα συνοψίζεται η βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για όλους τους αριθμούς εργασιών που έχουμε ως δεδομένα. Αντίστοιχα, στο δεύτερο διάγραμμα αναπαριστώνται τα σημεία αλλαγής της βέλτιστης πολιτικής του ευέλικτου εξυπηρετητή.

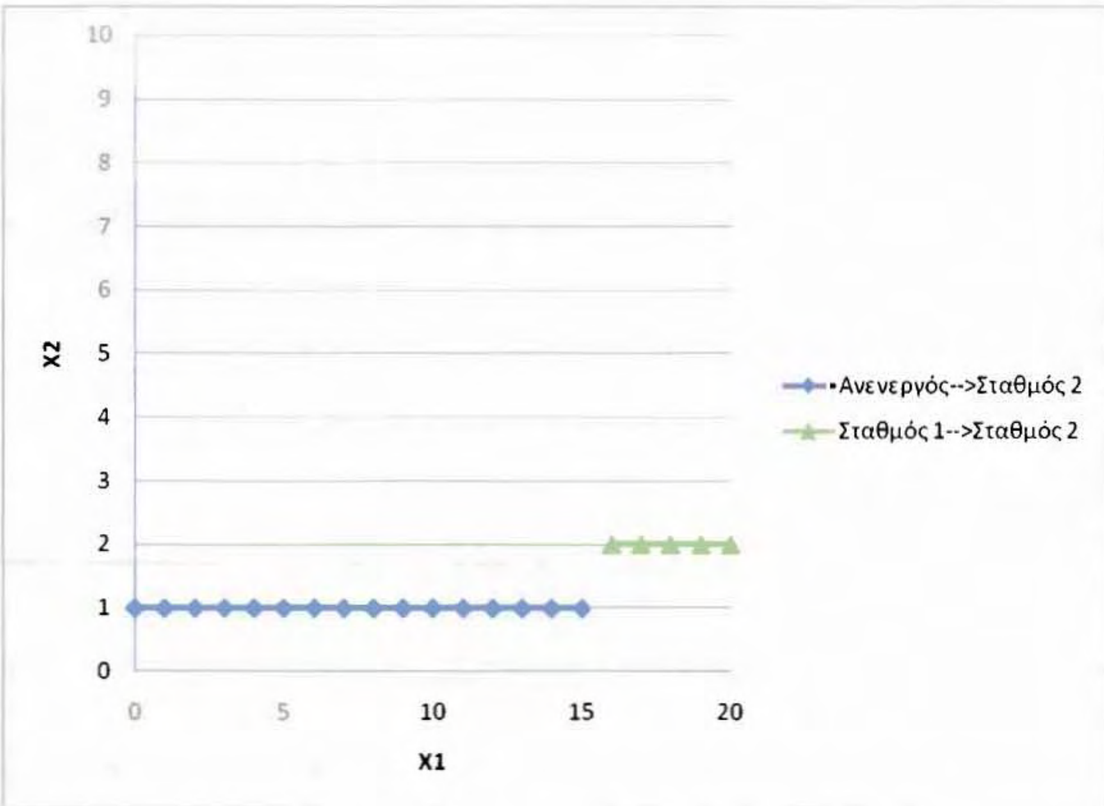
Στο διάγραμμα 3.19 αναπαρίσταται η βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή, για τα αρχικά δεδομένα που χρησιμοποιήσαμε. Με το κόκκινο χρώμα υποδηλώνονται τα σημεία εκείνα στα οποία ο ευέλικτος εξυπηρετητής παραμένει ανενεργός.

Με πράσινο χρώμα αναπαριστώνται τα σημεία λειτουργίας στον σταθμό εργασίας 1, ενώ με μπλε τα σημεία λειτουργίας στον σταθμό εργασίας 2.



Διάγραμμα 3.19: Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή

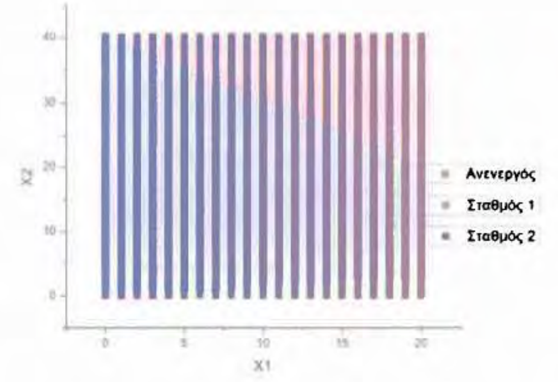
Στο διάγραμμα 3.20 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε σχέση με τον αριθμό εργασιών του σταθμού 1 και 2. Τα σημεία με μπλε χρώμα υποδηλώνουν την αλλαγή λειτουργίας του ευέλικτου εξυπηρετητή από την ανενεργή κατάσταση στην λειτουργία του στον σταθμό εργασίας 2. Αναλόγως, τα σημεία με πράσινο χρώμα αναπαριστούν την αλλαγή λειτουργίας από τον σταθμό εργασίας 1 στον σταθμό εργασίας 2.



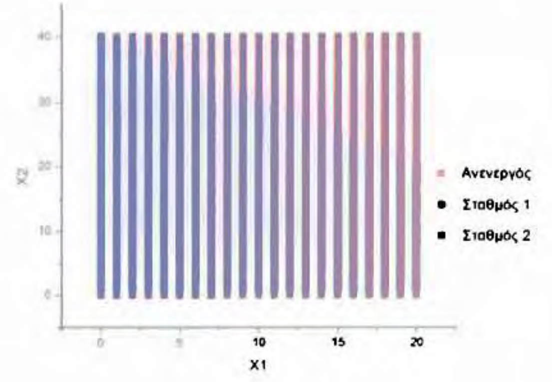
Διάγραμμα 3.20: Διάγραμμα αλλαγής βέλτιστης απόφασης του ευέλικτου εξυπηρετητή

3.2.1 Αποτελέσματα όταν διαφοροποιείται το γραμμικό κόστος παραμονής κάθε εργασίας σε κάθε σταθμό

Όταν αυξάνεται το γραμμικό κόστος παραμονής h_i με το οποίο επιβαρύνεται κάθε εργασία στον σταθμό εργασίας, i, $i=1,2$, τότε αυξάνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.21 και στο Διάγραμμα 3.22 όταν το γραμμικό κόστος παραμονής στον σταθμό εργασίας 1 και 2 είναι $h_1=12$ και $h_2=20$ αντίστοιχα.



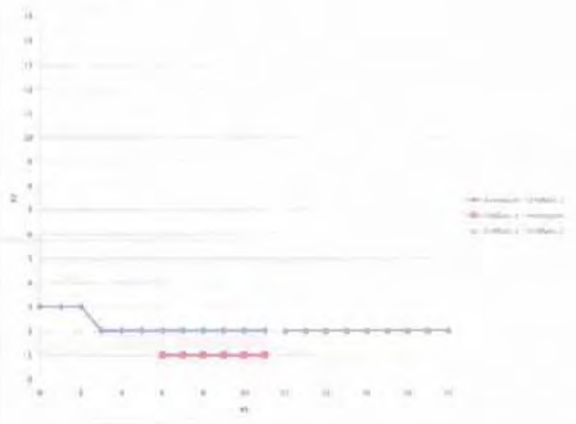
Διάγραμμα 3.21:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $h_1=12$



Διάγραμμα 3.22:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $h_2=20$

Στο Διάγραμμα 3.23 και στο Διάγραμμα 3.24 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τα νέα γραμμικά κόστη αναμονής στους αντίστοιχους σταθμούς εργασίας.

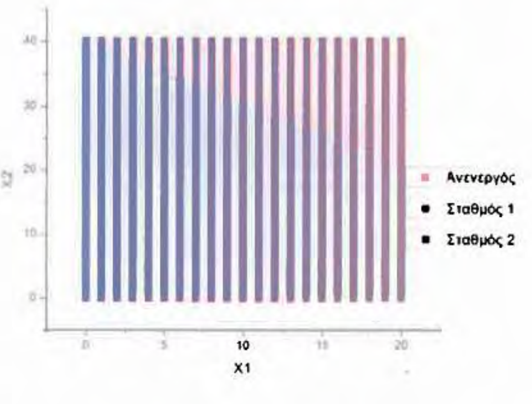
Διάγραμμα 3.23:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $h_1 = 12$



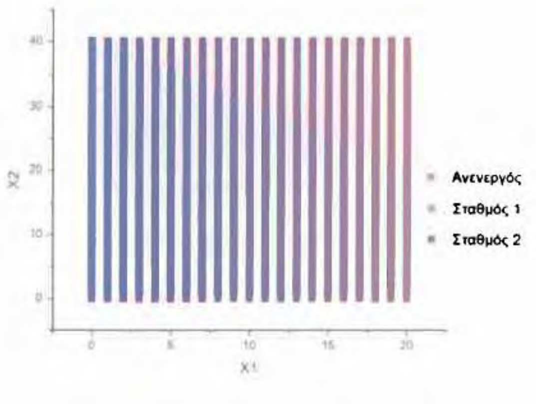
Διάγραμμα 3.24: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $h_2 = 20$

3.2.2 Αποτελέσματα όταν διαφοροποιείται το λειτουργικό κόστος του ευέλικτου εξυπηρετητή

Όταν αυξάνεται το λειτουργικό κόστος c_i με οποίο επιβαρύνεται ο ευέλικτος εξυπηρετητής κατά την διάρκεια του χρόνου λειτουργίας του στον σταθμό εργασίας i, i=1, 2, τότε μειώνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αντίστροφα. Αυτό παρουσιάζεται στο Διάγραμμα 3.25 και στο Διάγραμμα 3.26 όταν το λειτουργικό κόστος του ευέλικτου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $c_1 = 40$ και $c_2 - 2$ αντίστοιχα.

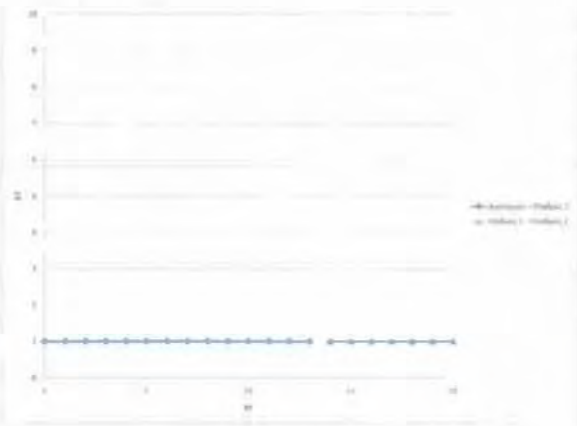


Διάγραμμα 3.25:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $c_1 = 40$

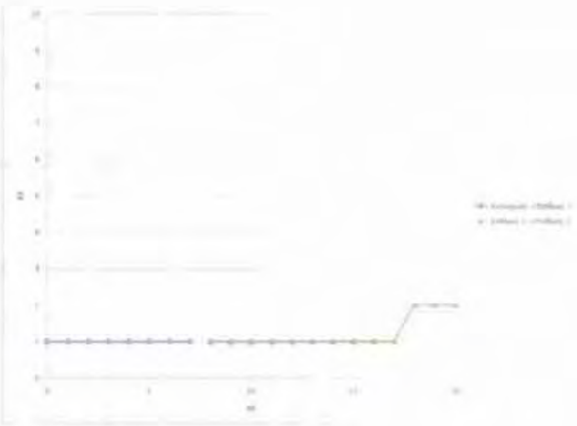


Διάγραμμα 3.26:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $c_2 = 2$

Στο Διάγραμμα 3.27 και στο Διάγραμμα 3.28 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τα νέα λειτουργικά κόστη του ευέλικτου εξυπηρετητή, στους αντίστοιχους σταθμούς εργασίας.



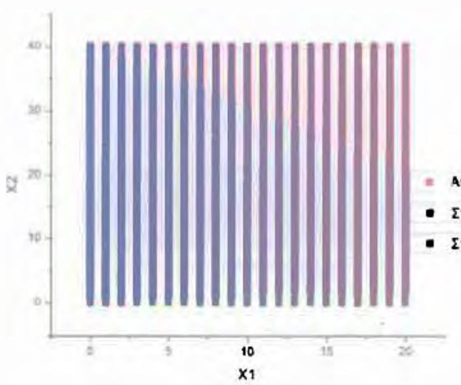
Διάγραμμα 3.27:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $c_1 = 40$



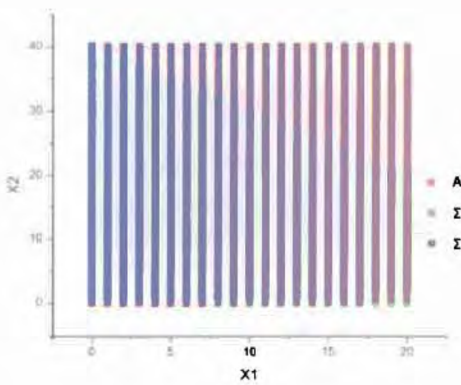
Διάγραμμα 3.28: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $c_2 = 2$

3.2.3 Αποτελέσματα όταν διαφοροποιείται ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή

Όταν αυξάνεται ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή v_i στον σταθμό εργασίας i, $i=1,2$, τότε αυξάνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή. Αυτό παρουσιάζεται στο Διάγραμμα 3.29 και στο Διάγραμμα 3.30 όταν ο ρυθμός εργασίας του ευέλικτου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $v_1=15$ και $v_2=15$ αντίστοιχα.

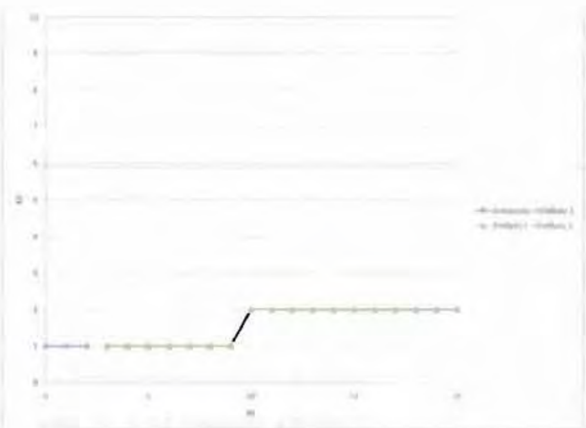


Διάγραμμα 3.29:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $v_1=15$

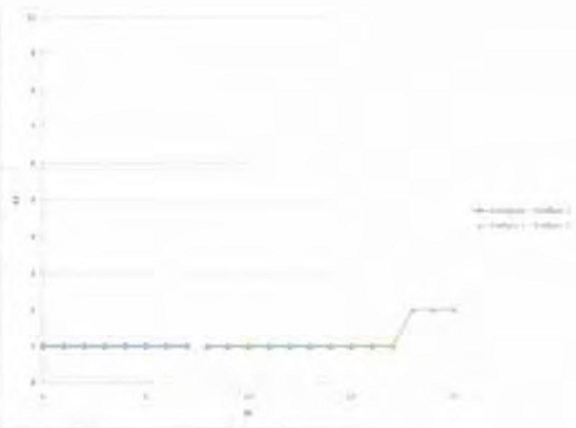


Διάγραμμα 3.30:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $v_2=15$

Στο Διάγραμμα 3.31 και στο Διάγραμμα 3.32 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τους νέους ρυθμούς εργασίας του ευέλικτου εξυπηρετητή στους αντίστοιχους σταθμούς εργασίας.



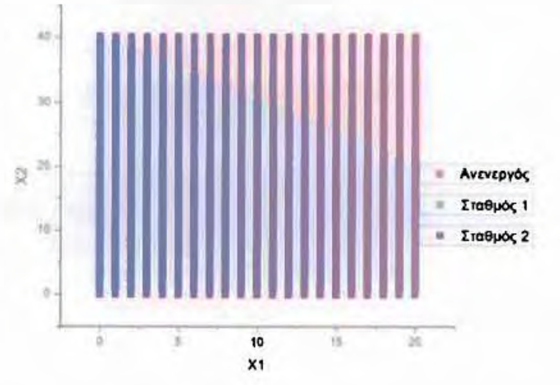
Διάγραμμα 3.31: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $v_1 = 15$



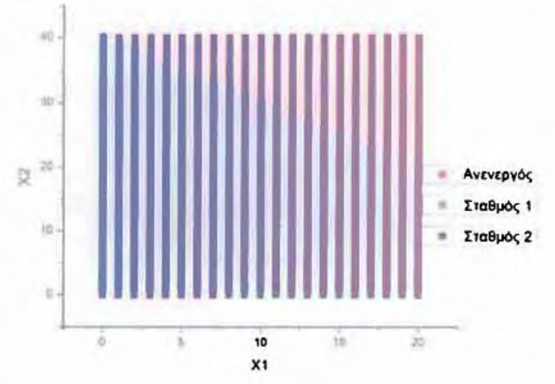
Διάγραμμα 3.32: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $v_2 = 15$

3.2.4 Αποτελέσματα όταν διαφοροποιείται ο ρυθμός λειτουργίας του κάθε εξειδικευμένου εξυπηρετητή

Όταν αυξάνεται ο ρυθμός λειτουργίας του εξειδικευμένου εξυπηρετητή μ_i στον σταθμό εργασίας i, i=1, 2, τότε μειώνεται η περιοχή του αντίστοιχου σταθμού στο διάγραμμα βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αντίστροφα. Αυτό παρουσιάζεται στο Διάγραμμα 3.33 και στο Διάγραμμα 3.34 όταν ο ρυθμός λειτουργίας του εξειδικευμένου εξυπηρετητή στον σταθμό εργασίας 1 και 2 είναι $\mu_1 = 25$ και $\mu_2 = 6$ αντίστοιχα.

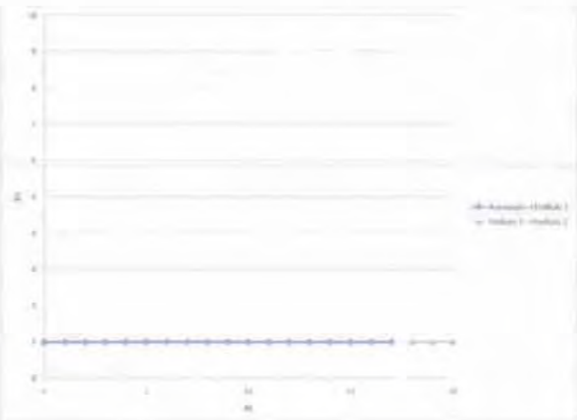


Διάγραμμα 3.33:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\mu_1 = 25$

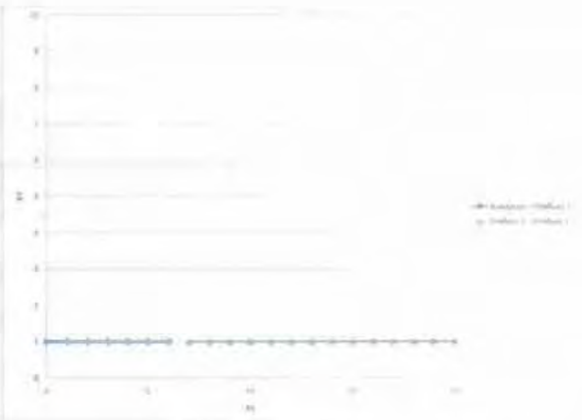


Διάγραμμα 3.34:Βέλτιστη πολιτική λειτουργίας του ευέλικτου εξυπηρετητή για $\mu_2 = 6$

Στο Διάγραμμα 3.35 και στο Διάγραμμα 3.36 παρουσιάζονται τα σημεία αλλαγής λειτουργίας του ευέλικτου εξυπηρετητή σε συνάρτηση με τον αριθμό εργασιών του σταθμού 1 και 2 για τους νέους ρυθμούς λειτουργίας κάθε εξειδικευμένου εξυπηρετητή στους αντίστοιχους σταθμούς εργασίας.



Διάγραμμα 3.35:Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\mu_1 = 25$



Διάγραμμα 3.36: Βέλτιστη απόφαση αλλαγής του ευέλικτου εξυπηρετητή για $\mu_2 = 6$

/

Κεφάλαιο 4 Σύνοψη Διπλωματικής Εργασίας

Σε αυτήν την διπλωματική εργασία εξετάσαμε τον προγραμματισμό ενός ευέλικτου εξυπηρετητή σε ένα σύστημα δίχως αφίξεις και το οποίο αποτελείται από δύο σταθμούς εργασιών με αφοσιωμένους εξυπηρετητές σε κάθε στάδιο, όπου το κόστος λειτουργίας και αναμονής τους συμπεριλαμβάνεται κατά τη διάρκεια του χρόνου εργασίας τους.

Ορίσαμε Συνθήκη Α που όταν δεν ικανοποιείται, είναι προτιμότερο, για κάποιες συγκεκριμένες περιπτώσεις του προβλήματος, ο ευέλικτος εξυπηρετητής να παραμείνει ανενεργός. Κατασκευάσαμε Διαγράμματα Αλλαγής Βέλτιστης Απόφασης του ευέλικτου εξυπηρετητή από τα οποία προκύπτει ότι η καμπύλη βέλτιστης απόφασης του ευέλικτου εξυπηρετητή από την ανενεργή κατάσταση λειτουργίας, στην διάθεσή του στον σταθμό εργασίας 2 είναι μη αύξουσα. Αντίθετα, η καμπύλη βέλτιστης απόφασης του ευέλικτου εξυπηρετητή τόσο από την διάθεσή του στον σταθμό εργασίας 1, στην ανενεργή κατάσταση λειτουργίας, όσο και από την μετακίνησή του από τον σταθμό εργασίας 1, στον σταθμό εργασίας 2 είναι μη φθίνουσα. Η μονοτονία αυτών των καμπυλών δεν έχει αποδειχθεί μαθηματικά.

Για τα συστήματα, τα οποία εξετάσαμε, έχει ήδη δειχθεί ότι υπό την Συνθήκη Α, η εκχώρηση του ευέλικτου εξυπηρετητή στον downstream σταθμό γίνεται βέλτιστη καθώς ο αριθμός των εργασιών εκεί αυξάνεται. Βέβαια για αυτήν την περίπτωση ο ευέλικτος εξυπηρετητής παραμένει ανενεργός μόνο για κάποιες συγκεκριμένες περιπτώσεις του προβλήματος, δηλαδή όταν στον downstream σταθμό (σταθμός 2) δεν έχουν μεταφερθεί εργασίες οι οποίες έχουν ολοκληρωθεί στον upstream σταθμό (σταθμός 1). Όλα τα παραπάνω παρουσιάζονται και στα Διαγράμματα Αλλαγής Βέλτιστης Απόφασης του ευέλικτου εξυπηρετητή. Από τα Διαγράμματα αυτά προκύπτει ότι η καμπύλη βέλτιστης απόφασης του ευέλικτου εξυπηρετητή από την ανενεργή κατάσταση λειτουργίας, στην διάθεσή του στον σταθμό εργασίας 2, όταν συμβαίνει είναι σταθερή. Αντίθετα, η καμπύλη βέλτιστης απόφασης του ευέλικτου εξυπηρετητή από την μετακίνησή του από τον σταθμό εργασίας 1, στον σταθμό εργασίας 2 είναι μη φθίνουσα.

Φυσικά, σκόπιμες είναι κάποιες επεκτάσεις του μοντέλου για την εφαρμογή του σε ένα ευρύτερο πρόβλημα, έτσι ώστε να λάβουμε υπόψη μας όλους τους παράγοντες που μπορεί να επηρεάζουν την εύρεση της βέλτιστης λύσης. Παρόλα αυτά, η συγκεκριμένη προσπάθεια αποτελεί ένα πρωταρχικό βήμα για την εύρεση της βέλτιστης πολιτικής λειτουργίας ενός ευέλικτου εξυπηρετητή σε ένα σύστημα δύο παράλληλα συνδεδεμένων σταθμών εργασίας και όπως όλα δείχνουν μπορεί να έχει ορθά και αξιόπιστα αποτελέσματα.

Μερικά από τα ερωτήματα που μείνανε αναπάντητα και θα μπορούσαν να αποτελέσουν αντικείμενο περαιτέρω εργασίας είναι η εύρεση της βέλτιστης συμπεριφοράς του ευέλικτου εξυπηρετητή σε ένα σύστημα το οποίο θα αποτελείται πάλι από δύο σταθμούς εργασιών και θα διαφοροποιείται στο γεγονός ότι θα δέχεται συνεχώς νέες αφίξεις. Τέλος, για αυτήν την περίπτωση σκόπιμο θα ήταν να μελετήσουμε και την μονοτονία της βέλτιστης πολιτικής.

Παράρτημα I

Το πρόγραμμα της Fortran που υλοποιήσαμε για την εύρεση της βέλτιστης πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή δίνεται παρακάτω:

```
program thesis_implemantation

implicit none

! Variables

integer         ::X1,X2

real            ::hh1,hh2,cc1,cc2,nn1,nn2,mm1,mm2
real            ::h1,h2,c1,c2,n1,n2,m1,m2

real            ::hh1start,hh1end
real            ::hh2start,hh2end

real            ::cc1start,cc1end
real            ::cc2start,cc2end

real            ::nn1start,nn1end
real            ::nn2start,nn2end

real            ::mm1start,mm1end
real            ::mm2start,mm2end

real(8)         ::V0(3),V1(2),V2(2)
```

```
real(8),        allocatable   ::V(:,:)

integer,        allocatable   ::U(:,:)

integer,        allocatable   ::CH1(:),CH2(:),CH3(:)
integer,        allocatable   ::CH(:)

real(8)                       ::p0,p1,p2

real,           allocatable   ::matrix(:)

integer                       ::loops,condition

character(1)                  ::D,E

integer                       ::i,j,t,k,g,b,y,s

character(*),   parameter     ::path_1='C:\Output\Output_Data\'
character(*),   parameter     ::path_2='C:\Output\Matrix_V\'
character(*),   parameter     ::path_3='C:\Output\Matrix_U\'
character(*),   parameter     ::path_4='C:\Output\Matrix_CH\'

character(80)                 ::filename_1
character(80)                 ::filename_2
character(80)                 ::filename_3
character(80)                 ::filename_4

! Body of thesis_implemantation

!Script for inserting the type of data

write   (*,*)   'Insert [R] for Random Data or [S] for Specified ones'
read    (*,*)    D
write   (*,*)   ''
```

```
!Script for inserting random data

if          (D=='R'.or.D=='r')      then

        write  (*,*)  'Press [F] to insert data from File or [K] from Keyboard'
        read   (*,*)   E
        write  (*,*)  ''


        call random_seed()

        !Script for inserting data from file entitled "Input_Data.dat"

        if          (E=='F'.or.E=='f')      then

                allocate(matrix(20))

                matrix=0.

                open   (10,file='Input_Data.dat')

                write  (10,*)  ''
                write  (10,*)  'The number of jobs'
                write  (10,*)  'X1='
                write  (10,*)  ''
                write  (10,*)  'X2='
                write  (10,*)  ''

                write  (10,*)  ''
                write  (10,*)  'The linear holding cost in station 1'
                write  (10,*)  'hh1start='
                write  (10,*)  ''
                write  (10,*)  'hh1end='
```

```
write (10,*) ''

write (10,*) ''
write (10,*) 'The linear holding cost in station 2'
write (10,*) 'hh2start='
write (10,*) ''
write (10,*) 'hh2end='
write (10,*) ''

write (10,*) ''
write (10,*) 'The linear operating cost of flexible in station 1'
write (10,*) 'cc1start='
write (10,*) ''
write (10,*) 'cc1end='
write (10,*) ''

write (10,*) ''
write (10,*) 'The linear operating cost of flexible in station 2'
write (10,*) 'cc2start='
write (10,*) ''
write (10,*) 'cc2end='
write (10,*) ''

write (10,*) ''
write (10,*) 'The processing rate for jobs of flexible in station 1'
write (10,*) 'nn1start='
write (10,*) ''
write (10,*) 'nn1end='
write (10,*) ''

write (10,*) ''
write (10,*) 'The processing rate for jobs of flexible in station 2'
write (10,*) 'nn2start='
write (10,*) ''
```

```
write  (10,*)  'nn2end–'
write  (10,*)  ''

write  (10,*)  ''
write  (10,*)  'The processing rate for jobs in station 1'
write  (10,*)  'mm1start–'
write  (10,*)  ''
write  (10,*)  'mm1end–'
write  (10,*)  ''

write  (10,*)  ''
write  (10,*)  'The processing rate for jobs in station 2'
write  (10,*)  'mm2start='
write  (10,*)  ''
write  (10,*)  'mm2end='
write  (10,*)  ''

write  (10,*)  ''
write  (10,*)  'Insert the number of loops'
write  (10,*)  'loops='
write  (10,*)  ''
write  (10,*)  'If the conditional statement is true/valid insert 1
othewise enter 0'
write  (10,*)  ''

close  (10)

pause

open  (10,file='Input_Data.dat')

do    i=1,20,2

      read  (10,*)
```

```
        read    (10,*)
        read    (10,*)
        read    (10,*)  matrix(i)
        read    (10,*)
        read    (10,*)  matrix(i+1)

enddo

close   (10)

!Script for the allocation of the data comprising the problem:

!The number of jobs

X1=matrix(1)
X2=matrix(2)

!The linear holding cost
!hh1

hh1start=matrix(3)
hh1end=matrix(4)

!hh2

hh2start=matrix(5)
hh2end=matrix(6)

!The linear operating cost
!cc1

cc1start=matrix(7)
cc1end=matrix(8)
```

```
!cc2

cc2start=matrix(9)
cc2end=matrix(10)

!The processing rate for jobs
!nn1

nn1start=matrix(11)
nn1end=matrix(12)

!nn2

nn2start=matrix(13)
nn2end=matrix(14)

!mm1

mm1start=matrix(15)
mm1end=matrix(16)

!mm2

mm2start=matrix(17)
mm2end=matrix(18)

loops=matrix(19)

if          (loops>0)     then

      condition=matrix(20)

endif
```

```
!Script for inserting data from the keyboard

elseif  (E=='K'.or.E=='k')      then

        !Script for the allocation of the data comprising the problem:

        !The number of jobs

        write   (*,*)   'Give the number of jobs in station 1'
        write   (*,*)   'X1='
        read    (*,*)    X1
        write   (*,*)   ''

        write   (*,*)   'Give the number of jobs in station 2'
        write   (*,*)   'X2='
        read    (*,*)    X2
        write   (*,*)   ''

        !The linear holding cost

        write   (*,*)   'Give the first linear holding cost in station 1'
        write   (*,*)   'hh1start='
        read    (*,*)    hh1start
        write   (*,*)   ''

        write   (*,*)   'Give the last linear holding cost in station 1'
        write   (*,*)   'hh1end='
        read    (*,*)    hh1end
        write   (*,*)   ''

        write   (*,*)   'Give the first linear holding cost in station 2'
        write   (*,*)   'hh2start='
        read    (*,*)    hh2start
        write   (*,*)   ''
```

```
write   (*,*)   'Give the last linear holding cost in station 2'
write   (*,*)   'hh2end='
read    (*,*)    hh2end
write   (*,*)   ''

!The linear operating cost

write   (*,*)   'Give the first linear operating cost of flexible in station
1'
write   (*,*)   'cc1start='
read    (*,*)    cc1start
write   (*,*)   ''

write   (*,*)   'Give the last linear operating cost of flexible in station
1'
write   (*,*)   'cc1end='
read    (*,*)    cc1end
write   (*,*)   ''

write   (*,*)   'Give the first linear operating cost of flexible in station
2'
write   (*,*)   'cc2start='
read    (*,*)    cc2start
write   (*,*)   ''

write   (*,*)   'Give the last linear operating cost of flexible in station
2'
write   (*,*)   'cc2end='
read    (*,*)    cc2end
write   (*,*)   ''

!The processing rate for jobs
```

```
write (*,*) 'Give the first processing rate for jobs of flexible in station 1'
write (*,*) 'nn1start='
read (*,*) nn1start
write (*,*) ''

write (*,*) 'Give the last processing rate for jobs of flexible in station 1'
write (*,*) 'nn1end='
read (*,*) nn1end
write (*,*) ''

write (*,*) 'Give the first processing rate for jobs of flexible in station 2'
write (*,*) 'nn2start='
read (*,*) nn2start
write (*,*) ''

write (*,*) 'Give the last processing rate for jobs of flexible in station 2'
write (*,*) 'nn2end='
read (*,*) nn2end
write (*,*) ''

write (*,*) 'Give the first processing rate for jobs in station 1'
write (*,*) 'mm1start='
read (*,*) mm1start
write (*,*) ''

write (*,*) 'Give the last processing rate for jobs in station 1'
write (*,*) 'mm1end='
read (*,*) mm1end
write (*,*) ''

write (*,*) 'Give the first processing rate for jobs in station 2'
```

```
write (*,*) 'mm2start='
read (*,*) mm2start
write (*,*) ''

write (*,*) 'Give the last processing rate for jobs in station 1'
write (*,*) 'mm2end='
read (*,*) mm2end
write (*,*) ''

write (*,*) 'Insert the number of loops'
read (*,*) loops
write (*,*) ''

if (loops>0) then

        write (*,*) 'If the conditional statement is true insert 1
othewise enter 0'
        read (*,*) condition
        write (*,*) ''

endif

endif

!Script for the specification of matrices space
V,V0,V1,V2,CH,CH1,CH2,CH3,U

allocate (V(0:(X1),0:(X1+X2)))

V=0.

V0=0.
V1=0.
V2=0.
```

```
allocate(CH(0:X1))

CH=0

allocate (CH1(0:X1))

CH1=0

allocate (CH2(0:X1))

CH2=0

allocate (CH3(0:X1))

CH3=0

allocate (U(0:(X1),0:(X1+X2)))

U=0

if            (loops==0)     then

        t=1

elseif  (loops>0)       then

        t=loops

endif

do i=1,t

        !Specification of the linear holding cost in Station 1
```

```
call random_number(hh1)
hh1=hh1start+(hh1*(hh1end-hh1start))

!Specification of the linear holding cost in Station 2

call random_number(hh2)
hh2=hh2start+(hh2*(hh2end-hh2start))

!Specification of the linear operating cost of flexible in Station 1

call random_number(cc1)
cc1=cc1start+(cc1*(cc1end-cc1start))

!Specification of the linear operating cost of flexible in Station 2

call random_number(cc2)
cc2=cc2start+(cc2*(cc2end-cc2start))

!Specification of the processing rate for jobs of flexible in Station 1

call random_number(nn1)
nn1=nn1start+(nn1*(nn1end-nn1start))

!Specification of the processing rate for jobs of flexible in Station 2

call random_number(nn2)
nn2=nn2start+(nn2*(nn2end-nn2start))

!Specification of the processing rate for jobs in Station 1

call random_number(mm1)
mm1=mm1start+(mm1*(mm1end-mm1start))
```

```
!Specification of the processing rate for jobs in Station 2

if          (t==1)  then

      call random_number(mm2)

      mm2=mm2start+(mm2*(mm2end-mm2start))

elseif  (t>1)   then

      do
            call random_number(mm2)

            if          (condition==1)        then

                  mm2=mm2start+(mm2*(mm2end-mm2start))

                  if      (mm2<=((hh2*nn2)/cc2))      exit

            elseif  (condition==0)        then

                  mm2=mm2start+(mm2*(hh2*nn2)/cc2)

                  if      (mm2>((hh2*nn2)/cc2))       exit

            endif

      enddo

endif

!Script for the normalization of data

h1=hh1/(mm1+mm2+nn1+nn2)
```

```
h2=hh2/(mm1+mm2+nn1+nn2)
c1=cc1/(mm1+mm2+nn1+nn2)
c2=cc2/(mm1+mm2+nn1+nn2)
n1=nn1/(mm1+mm2+nn1+nn2)
n2=nn2/(mm1+mm2+nn1+nn2)
m1=mm1/(mm1+mm2+nn1+nn2)
m2=mm2/(mm1+mm2+nn1+nn2)

!Script for the creation of files which embody the value of the variables

write(filename_1,'(a,a,i7.7,a)')        path_1,'Output_Data_',i,'.txt'

open(unit=1, file=filename_1,action='write')

write   (1,*)   '------'
write   (1,*)   '|','Data','|'
write   (1,*)   '------'
write   (1,*)   ''
write   (1,*)   ''

write   (1,*)   'The number of jobs in station 1'
write   (1,*)   'X1=',X1
write   (1,*)   ''

write   (1,*)   'The number of jobs in station 2'
write   (1,*)   'X2=',X2
write   (1,*)   ''

write   (1,*)   'The linear holding cost in station 1'
write   (1,*)   'hh1=',hh1
write   (1,*)   ''

write   (1,*)   'The linear holding cost in station 2'
write   (1,*)   'hh2=',hh2
```

```
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 1'
write  (1,*)  'cc1=',cc1
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 2'
write  (1,*)  'cc2=',cc2
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs of flexible in station 1'
write  (1,*)  'nn1=',nn1
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs of flexible in station 2'
write  (1,*)  'nn2=',nn2
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 1'
write  (1,*)  'mm1=',mm1
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 2'
write  (1,*)  'mm2=',mm2
write  (1,*)  ''

write  (1,*)  ''
write  (1,*)  '-----------------'
write  (1,*)  '|','Normalized Data','|'
write  (1,*)  '-----------------'
write  (1,*)  ''
write  (1,*)  ''

write  (1,*)  'The linear holding cost in station 1'
```

```
write  (1,*)  'h1=',h1
write  (1,*)  ''

write  (1,*)  'The linear holding cost in station 2'
write  (1,*)  'h2=',h2
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 1'
write  (1,*)  'c1=',c1
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 2'
write  (1,*)  'c2=',c2
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs of flexible in station 1'
write  (1,*)  'n1=',n1
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs of flexible in station 2'
write  (1,*)  'n2=',n2
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 1'
write  (1,*)  'm1=',m1
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 2'
write  (1,*)  'm2=',m2
write  (1,*)  ''
write  (1,*)  ''

if          (loops==0)    then
```

```
        if      ((cc2/nn2)<=(hh2/mm2))      then

                write   (1,*)   'Condition A is valid'

        elseif  ((cc2/nn2)>(hh2/mm2))       then

                write   (1,*)   'Condition A is invalid'

        endif

elseif  (loops>0.and.condition==1)  then

        write   (1,*)   'Condition A is valid'

elseif  (loops>0.and.condition==0)  then

        write   (1,*)   'Condition A is invalid'

endif

/
!Script for the manipulation of data and allocation of the results in the
matrices V & U

do k=0,X1

        do j=0,X2+X1-k

                if (k==0.and.j==0) then

                        V(k,j)=0.

                        U(k,j)=0

                else if (k==0.and.j/=0) then
```

```
        V2(1)=(h2*j)/(m2)

        V2(2)=((h2*j)+c2)/(m2+n2)

        p2=minval(V2)

        if (p2==V2(1)) then

                U(k,j)=0

        else if (p2==V2(2)) then

                U(k,j)=2

        endif

        V(k,j)=p2 + V(k,j-1)

    else if (k/=0.and.j==0) then

        V1(1)=(h1*k)/(m1)

        V1(2)=((h1*k)+c1)/(m1+n1)

        p1=minval(V1)

        if (p1==V1(1)) then

                U(k,j)=0

        else if (p1==V1(2)) then

                U(k,j)=1
```

```
endif

V(k,j)=p1 + V(k-1,1)

else if (k/=0.and.j/=0) then

V0(1)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)))/(1-(n1+n2))

V0(2)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)) + c1 + (n1*V(k-1,j+1)))/(1-n2)

V0(3)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)) + c2 + (n2*V(k,j-1)))/(1-n1)

p0=minval(V0)

if (p0==V0(1)) then

U(k,j)=0

else if (p0==V0(2)) then

U(k,j)=1

else if (p0==V0(3)) then

U(k,j)=2

endif

V(k,j)=p0

endif
```

```
        enddo
    enddo
    !Script for the creation of files which embody the values-data of the matrices U & V
    write(filename_2,'(a,a,i7.7,a)')        path_2,'Matrix_V_',i,'.txt'
    open(unit=2, file=filename_2,action='write')
    do      y=0,X1
        do      s=0,X1+X2
            write   (2,*)           y,s,V(y,s)
        enddo
    enddo
    close(2)
    write(filename_3,'(a,a,i7.7,a)')        path_3,'Matrix_U_',i,'.xls'
    open(unit=3, file=filename_3,action='write')
    do      y=0,X1
        do      s=0,X1+X2
            write   (3,210)         y,s,U(y,s)
```

```
enddo

enddo

close(3)

!Script for the manipulation of data and allocation of the results in the
matrix CH

if (condition==1) then

!Script for the creation of file which embodies the values-data
of the matrix CH

write(filename_4,'(a,a,i7.7,a)')
path_4,'Matrix_CH_',i,'.txt'

open(unit=4, file=filename_4,action='write')

do b=0,X1

do j=0,((X1)+(X2))-b-1

if (U(b,j)/=U(b,j+1)) then

CH(b)=j+1

exit

endif

enddo

write (4,*) CH(b)
```

```
enddo

close(4)

do g=1,(X1)-1

        if (CH(g)>CH(g+1).and.CH(g+1)>0.) then

                write  (1,*)   ''
                write  (1,*)   'Decreasing'

        endif

enddo

close(1)

elseif  (condition==0)          then

!Script for the creation of files which embody the values-data of
the matrices CH1,CH2,CH3

write(filename_4,'(a,a,i7.7,a)')
path_4,'Matrix_CH_',i,'.txt'

open(unit=4, file=filename_4,action='write')

do b=0,X1

        do j=0,((X1)+(X2))-b-1

                if (U(b,j)==0.and.U(b,j+1)==2) then

                        CH1(b)=j+1
```

```
                elseif(U(b,j)==1.and.U(b,j+1)==0) then
                        CH2(b)=j+1
                elseif (U(b,j)==1.and.U(b,j+1)==2) then
                        CH3(b)=j+1
                endif
        enddo
enddo
do g=0,X1
        if(CH1(g)/=0.and.CH2(g)==0.and.CH3(g)==0) then
                write(4,*) CH1(g),'    ','       '
        elseif(CH1(g)==0.and.CH2(g)/=0.and.CH3(g)==0) then
                write(4,*) '        ',CH2(g),'          '
        elseif(CH1(g)==0.and.CH2(g)==0.and.CH3(g)/=0) then
                write(4,*) '     ','                 ',CH3(g)
        elseif(CH1(g)/=0.and.CH2(g)/=0.and.CH3(g)==0) then
                write(4,*) CH1(g),CH2(g),'   '
        elseif(CH1(g)/=0.and.CH2(g)==0.and.CH3(g)/=0) then
```

```
                write(4,*) CH1(g),'                ',CH3(g)
            elseif(CH1(g)==0.and.CH2(g)/=0.and.CH3(g)/=0) then
                write(4,*) '          ',CH2(i),CH3(g)
            elseif(CH1(g)/=0.and.CH2(g)/=0.and.CH3(g)/=0) then
                write(4,*) CH1(g),CH2(g),CH3(g)
            elseif(CH1(g)==0.and.CH2(g)==0.and.CH3(g)==0) then
                write(4,*) '     ','     ','     '
            endif
        enddo
    endif
    close(4)
    !Process to empty the V,U,CH matrices due to the fact of entering new
loops
    if      (loops>0)       then
        V=0.
        V0=0.
        V1=0.
        V2=0.
```

```
                U=0

                CH=0

                CH1=0
                CH2=0
                Ch3=0

            endif

        enddo

!Script for inserting specific data

elseif  (D=='S'.or.D=='s')      then

        write   (*,*)   'Press [F] to insert data from File or [K] from Keyboard'
        read    (*,*)    E
        write   (*,*)   ''

        !Script for inserting data from file entitled "Input_Data.dat"

        if              (E=='F'.or.E=='f')      then

            allocate(matrix(10))

            matrix=0.

            open    (10,file='Input_Data.dat')

            write   (10,*)  ''
            write   (10,*)  'The number of jobs in station 1'
            write   (10,*)  'X1='
            write   (10,*)  ''
```

```
write  (10,*) ''
write  (10,*) 'The number of jobs in station 2'
write  (10,*) 'X2='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The linear holding cost in station 1'
write  (10,*) 'hh1='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The linear holding cost in station 2'
write  (10,*) 'hh2='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The linear operating cost of flexible in station 1'
write  (10,*) 'cc1='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The linear operating cost of flexible in station 2'
write  (10,*) 'cc2='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The processing rate for jobs of flexible in station 1'
write  (10,*) 'nn1='
write  (10,*) ''

write  (10,*) ''
write  (10,*) 'The processing rate for jobs of flexible in station 2'
write  (10,*) 'nn2='
```

```
write (10,*) ''

write (10,*) ''
write (10,*) 'The processing rate for jobs in station 1'
write (10,*) 'mm1='
write (10,*) ''

write (10,*) ''
write (10,*) 'The processing rate for jobs in station 2'
write (10,*) 'mm2='
write (10,*) ''

close (10)

pause

open (10,file='Input_Data.dat')

do i=1,10

    read (10,*)
    read (10,*)
    read (10,*)
    read (10,*) matrix(i)

enddo

close (10)

!Script for the allocation of the data comprising the problem:

!The number of jobs

X1=matrix(1)
```

```
        X2=matrix(2)

        !The linear holding cost

        hh1=matrix(3)
        hh2=matrix(4)

        !The linear operating cost

        cc1=matrix(5)
        cc2=matrix(6)

        !The processing rate for jobs

        nn1=matrix(7)
        nn2=matrix(8)
        mm1=matrix(9)
        mm2=matrix(10)

!Script for inserting data from the keyboard

elseif  (E=='K'.or.E=='k')     then

        !The number of jobs

        write   (*,*)   'Give the number of jobs in station 1'
        write   (*,*)   'X1='
        read    (*,*)    X1
        write   (*,*)   ''

        write   (*,*)   'Give the number of jobs in station 2'
        write   (*,*)   'X2='
        read    (*,*)    X2
        write   (*,*)   ''
```

```
!The linear holding cost

write  (*,*)  'Give the linear holding cost in station 1'
write  (*,*)  'hh1–'
read   (*,*)   hh1
write  (*,*)  ''

write  (*,*)  'Give the linear holding cost in station 2'
write  (*,*)  'hh2–'
read   (*,*)   hh2
write  (*,*)  ''

!The linear operating cost

write  (*,*)  'Give the linear operating cost of flexible in station 1'
write  (*,*)  'cc1–'
read   (*,*)   cc1
write  (*,*)  ''

write  (*,*)  'Give the linear operating cost of flexible in station 2'
write  (*,*)  'cc2='
read   (*,*)   cc2
write  (*,*)  ''

!The processing rate for jobs

write  (*,*)  'Give the processing rate for jobs of flexible in station 1'
write  (*,*)  'nn1='
read   (*,*)   nn1
write  (*,*)  ''

write  (*,*)  'Give the processing rate for jobs of flexible in station 2'
write  (*.*)  'nn2–'
```

```
        read   (*,*)   nn2
        write  (*,*)   ''

        write  (*,*)   'Give the processing rate for jobs in station 1'
        write  (*,*)   'mm1='
        read   (*,*)   mm1
        write  (*,*)   ''

        write  (*,*)   'Give the processing rate for jobs in station 2'
        write  (*,*)   'mm2='
        read   (*,*)   mm2
        write  (*,*)   ''

endif

!Script for the normalization of data

h1=hh1/(mm1+mm2+nn1+nn2)
h2=hh2/(mm1+mm2+nn1+nn2)
c1=cc1/(mm1+mm2+nn1+nn2)
c2=cc2/(mm1+mm2+nn1+nn2)
n1=nn1/(mm1+mm2+nn1+nn2)
n2=nn2/(mm1+mm2+nn1+nn2)
m1=mm1/(mm1+mm2+nn1+nn2)
m2=mm2/(mm1+mm2+nn1+nn2)

!Script for the creation of files which embody the value of the variables

write(filename_1,'(a,a,a)')      path_1,'Output_Data','.txt'

open(unit=1, file=filename_1,action='write')

write  (1,*)   '------'
write  (1,*)   '|','Data','|'
```

```
write (1,*) '------'
write (1,*) ''
write (1,*) ''

write (1,*) 'The number of jobs in station 1'
write (1,*) 'X1=',X1
write (1,*) ''

write (1,*) 'The number of jobs in station 2'
write (1,*) 'X2=',X2
write (1,*) ''

write (1,*) 'The linear holding cost in station 1'
write (1,*) 'hh1=',hh1
write (1,*) ''

write (1,*) 'The linear holding cost in station 2'
write (1,*) 'hh2=',hh2
write (1,*) ''

write (1,*) 'The linear operating cost of flexible in station 1'
write (1,*) 'cc1=',cc1
write (1,*) ''

write (1,*) 'The linear operating cost of flexible in station 2'
write (1,*) 'cc2=',cc2
write (1,*) ''

write (1,*) 'The processing rate for jobs of flexible in station 1'
write (1,*) 'nn1=',nn1
write (1,*) ''

write (1,*) 'The processing rate for jobs of flexible in station 2'
write (1,*) 'nn2=',nn2
```

```
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 1'
write  (1,*)  'mm1=',mm1
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs in station 2'
write  (1,*)  'mm2=',mm2
write  (1,*)  ''

write  (1,*)  ''
write  (1,*)  '-----------------'
write  (1,*)  '|','Normalized Data','|'
write  (1,*)  '-----------------'
write  (1,*)  ''
write  (1,*)  ''

write  (1,*)  'The linear holding cost in station 1'
write  (1,*)  'h1=',h1
write  (1,*)  ''

write  (1,*)  'The linear holding cost in station 2'
write  (1,*)  'h2=',h2
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 1'
write  (1,*)  'c1=',c1
write  (1,*)  ''

write  (1,*)  'The linear operating cost of flexible in station 2'
write  (1,*)  'c2=',c2
write  (1,*)  ''

write  (1,*)  'The processing rate for jobs of flexible in station 1'
```

```
write (1,*) 'n1=',n1
write (1,*) ''

write (1,*) 'The processing rate for jobs of flexible in station 2'
write (1,*) 'n2=',n2
write (1,*) ''

write (1,*) 'The processing rate for jobs in station 1'
write (1,*) 'm1=',m1
write (1,*) ''

write (1,*) 'The processing rate for jobs in station 2'
write (1,*) 'm2=',m2
write (1,*) ''
write (1,*) ''

if ((cc2/nn2)<=(hh2/mm2)) then

    write (1,*) 'Condition A is valid'

elseif ((cc2/nn2)>(hh2/mm2)) then

    write (1,*) 'Condition A is invalid'

endif

!Script for the specification of matrices space V,V0,V1,V2,CH,CH1,CH2,CH3,U

allocate (V(0:(X1),0:(X1+X2)))

V=0.

V0=0.
```

```
V1=0.
V2=0.

allocate (U(0:(X1),0:(X1+X2)))

U=0

allocate(CH(0:X1))

CH=0

allocate (CH1(0:X1))

CH1=0

allocate (CH2(0:X1))

CH2=0

allocate (CH3(0:X1))

CH3=0

!Script for the manipulation of data and allocation of the results in the matrices
V & U

do k=0,X1

        do j=0,X2+X1-k

                if (k==0.and.j==0) then

                        V(k,j)=0.
```

```
        U(k,j)=0
    else if (k==0.and.j/=0) then
        V2(1)=(h2*j)/(m2)
        V2(2)=((h2*j)+c2)/(m2+n2)
        p2=minval(V2)
        if (p2==V2(1)) then
            U(k,j)=0
        else if (p2==V2(2)) then
            U(k,j)=2
        endif
        V(k,j)=p2 + V(k,j-1)
    else if (k/=0.and.j==0) then
        V1(1)=(h1*k)/(m1)
        V1(2)=((h1*k)+c1)/(m1+n1)
        p1=minval(V1)
        if (p1==V1(1)) then
            U(k,j)=0
```

```
          else if (p1==V1(2)) then

                    U(k,j)=1

          endif

          V(k,j)=p1 + V(k-1,1)

     else if (k/=0.and.j/=0) then

          V0(1)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)))/(1-(n1+n2))

          V0(2)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)) + c1 + (n1*V(k-1,j+1)))/(1-n2)

          V0(3)=((h1*k) + (h2*j) + (m1*V(k-1,j+1)) + (m2*V(k,j-1)) + c2 + (n2*V(k,j-1)))/(1-n1)

          p0=minval(V0)

          if (p0==V0(1)) then

                    U(k,j)=0

          else if (p0==V0(2)) then

                    U(k,j)=1

          else if (p0==V0(3)) then

                    U(k,j)=2

          endif
```

```
                V(k,j)=p0
            endif
        enddo
    enddo
    !Script for the creation of files which embody the values-data of the matrices U
& V
    write(filename_2,'(a,a,a)')     path_2,'Matrix_V','.txt'
    open(unit=2, file=filename_2,action='write')
    do      y=0,X1
        do      s=0,X1+X2
            write   (2,*)           y,s,V(y,s)
        enddo
    enddo
    close(2)
    write(filename_3,'(a,a,a)')     path_3,'Matrix_U','.xls'
    open(unit=3, file=filename_3,action='write')
    do      y=0,X1
        do      s=0,X1+X2
```

```
                write   (3,210)         y,s,U(y,s)

            enddo

        enddo

        close(3)

        !Script for the manipulation of data and allocation of the results in the matrix CH

        if      ((cc2/nn2)<=(hh2/mm2))      then

            !Script for the creation of file which embodies the values-data of the matrix CH

            write(filename_4,'(a,a,a)')      path_4,'Matrix_CH','.txt'

            open(unit=4, file=filename_4,action='write')

            do b=0,X1

                do j=0,((X1)+(X2))-b-1

                    if (U(b,j)/=U(b,j+1)) then

                        CH(b)=j+1

                        exit

                    endif

                enddo
```

```
            write   (4,*)   CH(b)

        enddo

        close(4)

        do g=1,(X1)-1

            if (CH(g)>CH(g+1).and.CH(g+1)>0.) then

                write   (1,*)   ''
                write   (1,*)   'Decreasing'

            endif

        enddo

        close(1)

    elseif  ((cc2/nn2)>(hh2/mm2))       then

        !Script for the creation of files which embody the values-data of the matrices CH1,CH2,CH3

        write(filename_4,'(a,a,a)')     path_4,'Matrix_CH','.txt'

        open(unit=4, file=filename_4,action='write')

        do b=0,X1

            do j=0,((X1)+(X2))-b-1

                if (U(b,j)==0.and.U(b,j+1)==2) then
```

```
                CH1(b)=j+1
            elseif(U(b,j)==1.and.U(b,j+1)==0) then
                CH2(b)=j+1
            elseif (U(b,j)==1.and.U(b,j+1)==2) then
                CH3(b)=j+1
            endif
        enddo
    enddo
    do g=0,X1
        if(CH1(g)/=0.and.CH2(g)==0.and.CH3(g)==0) then
            write(4,*) CH1(g),'    ','          '
        elseif(CH1(g)==0.and.CH2(g)/=0.and.CH3(g)==0) then
            write(4,*) '         ',CH2(g),'          '
        elseif(CH1(g)==0.and.CH2(g)==0.and.CH3(g)/=0) then
            write(4,*) '     ','                   ',CH3(g)
        elseif(CH1(g)/=0.and.CH2(g)/=0.and.CH3(g)==0) then
            write(4,*) CH1(g),CH2(g),'    '
```

```
            elseif(CH1(g)/=0.and.CH2(g)==0.and.CH3(g)/=0) then
                write(4,*) CH1(g),'          ',CH3(g)
            elseif(CH1(g)==0.and.CH2(g)/=0.and.CH3(g)/=0) then
                write(4,*) '      ',CH2(i),CH3(g)
            elseif(CH1(g)/=0.and.CH2(g)/=0.and.CH3(g)/=0) then
                write(4,*) CH1(g),CH2(g),CH3(g)
            elseif(CH1(g)==0.and.CH2(g)==0.and.CH3(g)==0) then
                write(4,*) '    ','     ','    '
            endif
        enddo
    endif
    close(4)
    210     format(I4,('    '),I4,('  '),I2,('  '))
endif
pause
end program thesis_implemantation
```

Παράρτημα ΙΙ

Παρακάτω παρατίθεται τα δεδομένα, κανονικοποιημένα και μη, για τις περιπτώσεις κατά τις οποίες δεν ισχύει ή ισχύει η Συνθήκη Α και τα οποία χρησιμοποιούνται από το πρόγραμμα μας για την εύρεση της πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αποθηκεύονται στον φάκελο με το όνομα, *Output_Data*.

Όταν δεν ισχύει η Συνθήκη Α	**Όταν ισχύει η Συνθήκη Α**
------ |Data| ------ / The number of jobs in station 1 X1= 20 The number of jobs in station 2 X2= 20 The linear holding cost in station 1 hh1= 0.7690945 The linear holding cost in station 2 hh2= 2.608206 The linear operating cost of flexible in station 1	------ |Data| ------ The number of jobs in station 1 X1= 20 The number of jobs in station 2 X2= 20 The linear holding cost in station 1 hh1= 6.317509 The linear holding cost in station 2 hh2= 10.18890 The linear operating cost of flexible in

cc1= 7.532834 The linear operating cost of flexible in station 2 cc2= 7.427641 The processing rate for jobs of flexible in station 1 nn1= 4.589133 The processing rate for jobs of flexible in station 2 nn2= 4.301105 The processing rate for jobs in station 1 mm1= 2.677692 The processing rate for jobs in station 2 mm2= 7.616096 ---------------- Normalized Data| ---------------- The linear holding cost in station 1 h1= 4.0090360E-02 The linear holding cost in station 2 h2= 0.1359572 The linear operating cost of flexible in station 1 c1= 0.3926618	station 1 cc1= 24.09972 The linear operating cost of flexible in station 2 cc2= 5.180889 The processing rate for jobs of flexible in station 1 nn1= 5.465436 The processing rate for jobs of flexible in station 2 nn2= 8.663159 The processing rate for jobs in station 1 mm1= 11.22580 The processing rate for jobs in station 2 mm2= 12.24341 ---------------- |Normalized Data ---------------- The linear holding cost in station 1 h1= 0.1680287 The linear holding cost in station 2 h2= 0.2709972 The linear operating cost of flexible in station 1

The linear operating cost of flexible in station 2 c2= 0.3871784 The processing rate for jobs of flexible in station 1 n1= 0.2392164 The processing rate for jobs of flexible in station 2 n2= 0.2242024 The processing rate for jobs in station 1 m1= 0.1395793 The processing rate for jobs in station 2 m2= 0.3970020 Condition A is invalid	c1= 0.6409874 The linear operating cost of flexible in station 2 c2= 0.1377977 The processing rate for jobs of flexible in station 1 n1= 0.1453658 The processing rate for jobs of flexible in station 2 n2= 0.2304166 The processing rate for jobs in station 1 m1= 0.2985759 The processing rate for jobs in station 2 m2= 0.3256416 Condition A is valid

Παράρτημα III

Παρακάτω παρατίθεται τα αποτελέσματα, δηλαδή τα σημεία στα οποία ο ευέλικτος εξυπηρετητής αλλάζει πολιτική λειτουργίας για πρώτη φορά, για τις περιπτώσεις κατά τις οποίες δεν ισχύει ή ισχύει η Συνθήκη Α. Τα συγκεκριμένα αποτελέσματα εξάγονται από το πρόγραμμα μας για την εύρεση της πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αποθηκεύονται στον φάκελο με το όνομα, *Matrix_CH.*

Όταν δεν ισχύει η Συνθήκη Α			**Όταν ισχύει η Συνθήκη Α**
6			1
5			1
4			1
4			1
4			1
4			1
4	1		1
4	1		1
4	1		1
4	1		1
4	2		1
3	2		1
3	2		1
3	2		1
		3	1
		3	1
		3	2
		3	2
		3	2
		3	2
		3	2

Παράρτημα IV

Παρακάτω παρατίθεται κάποια ενδεικτικά αποτελέσματα για κάθε συνδυασμό αριθμού εργασιών, στους σταθμούς εργασίας, την αντίστοιχη βέλτιστη πολιτική του ευέλικτου εξυπηρετητή, για τις περιπτώσεις κατά τις οποίες δεν ισχύει ή ισχύει η Συνθήκη Α. Τα συγκεκριμένα αποτελέσματα εξάγονται από το πρόγραμμα μας για την εύρεση της πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αποθηκεύονται στον φάκελο με το όνομα, *Matrix_U*.

Όταν δεν ισχύει η Συνθήκη Α

0	0	0	7	0	1	14	0	1
0	1	0	7	1	0	14	1	1
0	2	0	7	2	0	14	2	1
0	3	0	7	3	0	14	3	2
0	4	0	7	4	2	14	4	2
0	5	0	7	5	2	14	5	2
0	6	2	7	6	2	14	6	2
0	7	2	7	7	2	14	7	2

Όταν ισχύει η Συνθήκη Α

0	0	0	7	0	0	14	0	1
0	1	2	7	1	2	14	1	2
0	2	2	7	2	2	14	2	2
0	3	2	7	3	2	14	3	2
0	4	2	7	4	2	14	4	2
0	5	2	7	5	2	14	5	2
0	6	2	7	6	2	14	6	2
0	7	2	7	7	2	14	7	2

Παράρτημα V

Παρακάτω παρατίθεται τα αποτελέσματα, δηλαδή για κάθε συνδυασμό αριθμού εργασιών, στους σταθμούς εργασίας το αντίστοιχο ελάχιστο αναμενόμενο συνολικό κόστος παραμονής και λειτουργίας με το οποίο επιβαρύνεται το υπό εξέταση σύστημα, για τις περιπτώσεις κατά τις οποίες δεν ισχύει ή ισχύει η Συνθήκη Α. Τα συγκεκριμένα αποτελέσματα εξάγονται από το πρόγραμμα μας για την εύρεση της πολιτικής λειτουργίας του ευέλικτου εξυπηρετητή και αποθηκεύονται στον φάκελο με το όνομα, Matrix_V.

Όταν δεν ισχύει η Συνθήκη Α

0	0	0.000000000000000E+000	7	0	11.0458596301672	14	0	31.5753557226311
0	1	0.342459704758799	7	1	11.6277957977561	14	1	32.4592497397745
0	2	1.02737911427640	7	2	12.7005532813360	14	2	33.9757898435588
0	3	2.05475822855279	7	3	14.2303815354165	14	3	35.8400332875878
0	4	3.42459704758799	7	4	16.1074680853136	14	4	37.9741097201908
0	5	5.13689557138199	7	5	18.2604360346966	14	5	40.3781306707852
0	6	7.07332985626397	7	6	20.6888023468589	14	6	43.0522286533648
0	7	9.22862476115364	7	7	23.3918584182611	14	7	45.9965537580349

Όταν ισχύει η Συνθήκη Α

0	0	0.000000000000000E+000	7	0	22.9149499956739	14	0	72.0677504877225
0	1	0.735165505346464	7	1	24.1136770531953	14	1	73.8557484069915
0	2	1.95768497518439	7	2	26.2238838516012	14	2	76.5684000722919
0	3	3.66755840951379	7	3	29.2040046145745	14	3	80.1976642123851
0	4	5.86478580833465	7	4	33.0118765081238	14	4	84.7327966714560
0	5	8.54936717164698	7	5	37.6060894065863	14	5	90.1603008650637
0	6	11.7213024994508	7	6	42.9470808032714	14	6	96.4640434375605
0	7	15.3805917917460	7	7	48.9979488558803	14	7	103.625511249756

Βιβλιογραφία

[1] Ahn, H.S., Duenyas, I., & Lewis, M.E. (2002). The optimal control of a two-stage tandem queueing system with flexible servers. Probability in the Engineering and Informational Sciences 16: 453–469.

[2] Ahn, H.S., Duenyas, I., & Zhang, R. (1999). Optimal stochastic scheduling of a two-stage tandem queue with parallel servers. Advances in Applied Probability 31: 1095–1117.

[3] Ahn, H.S. & Righter, R. (2006). Dynamic load balancing with flexible workers. Advances in Applied Probability 38: 621–642.

[4] Andradottir, S., Ayhan, H., & Down, D.G. (2001). Server assignment policies for maximizing the steady-state throughput of finite queueing systems. Management Science 47: 1421–1439.

[5] Andradottir, S., Ayhan, H., & Down, D.G. (2003). Dynamic server allocation for queueing networks with flexible servers. Operations Research 51: 952–968.

[6] Andradottir, S., Ayhan, H., & Down, D.G. (2007). Compensating for failures with flexible servers. Operations Research 55: 753–768.

[7] Duenyas, I., Gupta, D., & Olsen, T. (1998). Control of a single server tandem queueing system with setups. Operations Research 46: 218–230.

[8] Farrar, T.M. (1992). Resource allocation in systems of queues. Ph.D. dissertation. Cambridge University, Cambridge.

[9] Farrar, T.M. (1993). Optimal use of an extra server in a two station tandem queueing network. IEEE Transactions on Automatic Control 38: 1296–1299.

[10] Gel, E.S., Hopp, W.J., & Van Oyen, M.P. (2002). Factors affecting opportunity of worksharing as a dynamic load balancing mechanism. IIE Transactions 34: 847–863.

[11] Gel, E.S., Hopp, W.J., & Van Oyen, M.P. (2007). Hierarchical cross-training in work-in-processconstrained systems. IIE Transactions 39: 125–143.

[12] Hopp, W.J., Tekin, E., & Van Oyen, M.P. (2004). Benefits of skill-chaining in production lines with cross-trained workers. Management Science 50: 83–98.

[13] Hopp,W.J. & Van Oyen, M.P. (2004). Agile workforce evaluation: A framework for cross-training and coordination. IIE Transactions 36: 919–940.

[14] Iravani, S.M., Posner, M.J., & Buzacott, J.A. (1997). A two-stage tandem queue attended by a moving server with holding and switching costs. Queueing Systems 26: 203–228.

[15] Iravani, S.M., Van Oyen, M.P., & Sims, K.T. (2005). Structural flexibility: A new perspective on the design of manufacturing and service operations. Management Science 51: 151–166.

[16] Narongwanich, W., Duenyas, I., & Birge, J. (2003). Optimal portfolio of reconfigurable and dedicated capacity underuncertainty.

http://users.iems.northwestern.edu/_jrbirge/Public/html/new.html

[17] Pandelis, D.G. (2007). Optimal use of excess capacity in two interconnected queues. Mathematical Methods of Operations Research 65: 179–192.

[18] Pandelis, D.G. (2008). Optimal control of flexible servers in two tandem queues with operating costs. Probability in the Engineering and Informational Sciences 22: 107-131.

[19] Pandelis, D.G. & Teneketzis, D. (1994). Optimal multiserver stochastic scheduling of two interconnected priority queues. Advances in Applied Probability 26: 258–279.

[20] Rosberg, Z., Varaiya, P., & Walrand, J. (1982). Optimal control of service in tandem queues. IEEE Transactions on Automatic Control 27: 600–609.

[21] Schiefermayr, K. & Weichbold, J. (2005). A complete solution for the optimal stochastic scheduling of a two-stage tandem queue with two flexible servers. Journal of Applied Probability 42: 778–796.

[22] Serfozo, R. (1978). An equivalence between continuous and discrete time Markov decision processes. Operations Research 27: 616–620.

[23] Van Oyen, M.P., Gel, E.S., & Hopp, W.J. (2001). Performance opportunity for workforce agility in collaborative and noncollaborative work systems. IIE Transactions 33: 761–777.

[24] Weber, R.R. & Stidham, S. (1987). Optimal control of service rates in networks of queues. Advances in Applied Probability 19: 202–218.

[25] Weichbold, J. & Schiefermayr, K. (2006). The optimal control of a general tandem queue. Probability in the Engineering and Informational Sciences 20: 307–327.

[26] Wu, C.H., Down, D.G., & Lewis, M.E. (2007). Heuristics for allocation of reconfigurable resources in a serial line with reliability considerations. IEEE Transactions.

[27] Wu, C.H., Lewis, M.E., & Veatch, M. (2006). Dynamic allocation of reconfigurable resources in a two-stage tandem queueing system with reliability considerations. IEEE Transactions on Automatic Control 51: 309–314.