

ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΛΕΙΤΟΥΡΓΙΑΣ ΓΡΑΜΜΩΝ ΠΑΡΑΓΩΓΗΣ ΜΕ ΕΥΕΛΙΚΤΟ ΕΡΓΑΤΙΚΟ ΔΥΝΑΜΙΚΟ

ΠΑΠΑΧΡΗΣΤΟΣ ΙΩΑΝΝΗΣ

Επιβλέπων καθηγητής

Δρ ΔΗΜΗΤΡΙΟΣ ΠΑΝΤΕΛΗΣ



ΒΟΛΟΣ 2014

Εξεταστική Επιτροπή

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,
Επίκουρος Καθηγητής,
Τμήμα Μηχανολόγων Μηχανικών,
Πανεπιστήμιο Θεσσαλίας.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της εργασίας κ. Δημήτριο Παντελή, επίκουρο καθηγητή του τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας, για την βοήθεια, τις γνώσεις, την παρότρυνση αλλά και για τις πολύτιμες συμβουλές κατά την εκπόνηση της συγκεκριμένης μεταπτυχιακής εργασίας. Επίσης, είμαι ευγνώμων στα υπόλοιπα μέλη της εξεταστικής επιτροπής της μεταπτυχιακής εργασίας μου, Καθηγητές κ. Γεώργιο Λυμπερόπουλο και τον κ. Γεώργιο Κοζανίδη, για την προσεκτική ανάγνωση της εργασίας μου και για τις πολύτιμες υποδείξεις τους.

Ακόμη, θα ήθελα να ευχαριστήσω όλους τους φίλους και φίλες για την συμπαράσταση, την ανοχή όλο το διάστημα της εργασίας καθώς και τη βοήθεια που λίγο πολύ όλοι μου προσέφεραν. Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για όλα τα εφόδια που μου έχουν προσφέρει και για την συνεχή υποστήριξή τους. Αφιερώνω αυτή την εργασία στην μητέρα μου και στον πατέρα μου.

Περίληψη

Η διπλωματική εργασία που ακολουθεί αναφέρεται στη βελτιστοποίηση λειτουργίας γραμμών παραγωγής με ευέλικτο εργατικό δυναμικό. Η εργασία μας αναφέρεται σε ένα σύστημα που αποτελείται από δύο σταθμούς εξυπηρέτησης σε σειρά, όπου ο καθένας εξυπηρετεί με έναν σταθερό εξυπηρετητή (εξειδικευμένο), ενώ υπάρχει και ένας ευέλικτος ο οποίος μπορεί να τίθεται σε λειτουργία ή να απενεργοποιείται και να μεταφέρεται από τον ένα σταθμό στον άλλο ανάλογα με τις ανάγκες του κάθε σταθμού για πρόσθετο ρυθμό εξυπηρέτησης. Στο σύστημα μας υποθέτουμε εκθετικούς ρυθμούς εξυπηρέτησης ενώ οι εργασίες προσέρχονται στο σύστημα σύμφωνα με την διαδικασία Poisson. Μία εργασία εξέρχεται από το σύστημα μόνο όταν εξυπηρετηθεί και από το δεύτερο σταθμό. Η κάθε εργασία χαρακτηρίζεται από ένα γραμμικό κόστος παραμονής που είναι διαφορετικό για κάθε σταθμό. Επίσης η συνεργασία μεταξύ σταθερού και ευέλικτου εξυπηρετητή για την διεκπεραίωση κοινής εργασίας δεν επιτρέπεται. Ακόμα, επιτρέπεται η αδρανοποίηση (idling) κάθε εξυπηρετητή. Στην εργασία μας διατυπώνουμε τις εξισώσεις βελτιστοποίησης και αποδεικνύουμε ορισμένες ιδιότητες της βέλτιστης πολιτικής για κάποιες ειδικές περιπτώσεις. Τέλος, ελέγχουμε την ισχύ των ιδιοτήτων αυτών, στις περιπτώσεις που δεν είναι δυνατόν να αποδειχθούν, μέσω αριθμητικών παραδειγμάτων.

Περιεχόμενα

Εξεταστική Επιτροπή	3
Ευχαριστίες.....	5
Περίληψη.....	7
Λίστα Πινάκων	11
Λίστα Εικόνων.....	12
Κεφάλαιο 1. Εισαγωγή	15
1.1 Περιγραφή και σχετική βιβλιογραφία.....	15
1.2 Στόχος της εργασίας.....	17
1.3 Υπολογιστικό πακέτο που χρησιμοποιήθηκε.....	17
Κεφάλαιο 2. Περιγραφή και ανάλυση προβλήματος	19
2.1 Διαμόρφωση μοντέλου βελτιστοποίησης	20
2.2 Η βέλτιστη πολιτική.....	21
2.3 Η βέλτιστη πολιτική για $h_1 \geq h_2$	23
2.3.1 Ιδιότητες προς έλεγχο	31
2.4 Η βέλτιστη πολιτική για $h_1 < h_2$	31
2.4.1 Ιδιότητες προς έλεγχο	34
Κεφάλαιο 3. Αριθμητικά αποτελέσματα	35
3.1 Περιγραφή αλγορίθμου.....	35
3.2 Αποτελέσματα για $h_1 \geq h_2$	37
3.2.1 Περίπτωση 1: $v_1 \geq \mu_1$ και $v_2 \geq \mu_2$	37
3.2.2 Περίπτωση 2: $v_1 \geq \mu_1$ και $v_2 < \mu_2$	45
3.2.3 Περίπτωση 3: $v_1 < \mu_1$ και $v_2 \geq \mu_2$	47
3.2.4 Περίπτωση 4: $v_1 < \mu_1$ και $v_2 < \mu_2$	47
3.2.5 Γενικά συμπεράσματα για $h_1 \geq h_2$	48

3.3 Αποτελέσματα για $h_1 < h_2$	49
3.3.1 Περίπτωση 1: $v_1 \geq \mu_1$ και $v_2 \geq \mu_2$	49
3.3.2 Περίπτωση 2: $v_1 \geq \mu_1$ και $v_2 < \mu_2$	58
3.3.3 Περίπτωση 3-4: $(v_1 < \mu_1$ και $v_2 \geq \mu_2)$ - $(v_1 < \mu_1$ και $v_2 < \mu_2)$	58
3.3.4 Γενικά συμπεράσματα για $h_1 < h_2$	59
Κεφάλαιο 4. Σύνοψη αποτελεσμάτων	61
Βιβλιογραφία	63
Παράρτημα	65

Λίστα Πινάκων

Πίνακας 1: Βέλτιστη ανάθεση πόρων για $h_1 \geq h_2$	24
Πίνακας 2: Βέλτιστη ανάθεση πόρων για $h_1 < h_2$	33
Πίνακας 3: Τιμές παραμέτρων για τις οποίες δεν ισχύει η μονοτονία των σημείων αλλαγής.....	46

Λίστα Εικόνων

Εικόνα 1: Σχεδιάγραμμα λειτουργίας του συστήματος	20
Εικόνα 2: Διάγραμμα σημείων αλλαγής με αρχικές τιμές παραμέτρων.....	37
Εικόνα 3: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου v_1	38
Εικόνα 4: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου μ_1	39
Εικόνα 5: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου v_2	40
Εικόνα 6: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου μ_2	41
Εικόνα 7: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου λ	42
Εικόνα 8: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου h_2	43
Εικόνα 9: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου h_1	44
Εικόνα 10: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία	45
Εικόνα 11: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία	47
Εικόνα 12: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία	48
Εικόνα 13: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αρχικές τιμές παραμέτρων.....	50
Εικόνα 14: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου μ_1	51
Εικόνα 15: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου v_1	52
Εικόνα 16: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου v_2	53
Εικόνα 17: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου μ_2	54

Εικόνα 18: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου λ	55
Εικόνα 19: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου h_1	56
Εικόνα 20: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου h_2	57

Κεφάλαιο 1. Εισαγωγή

1.1 Περιγραφή και σχετική βιβλιογραφία

Στο πλαίσιο της εργασίας θα εξεταστεί η βέλτιστη ανάθεση των εξυπηρετητών σε ένα σύστημα που αποτελείται από δύο σταθμούς εξυπηρέτησης σε σειρά που ο καθένας εξυπηρετείται από ένα σταθερό (εξειδικευμένο) εξυπηρετητή ενώ υπάρχει κι ένας ευέλικτος ο οποίος μπορεί να τοποθετείται και στους δύο σταθμούς όπου υπάρχει ανάγκη για περαιτέρω βοήθεια. Πιο συγκεκριμένα, αναζητούμε στρατηγικές κατανομής των πόρων (εξυπηρετητών) για την ελαχιστοποίηση του γραμμικού κόστους αναμονής για συστήματα με αφίξεις που ακολουθούν την κατανομή Poisson και με εκθετικούς ρυθμούς εξυπηρέτησης. Το μαθηματικό μοντέλο που αναλύουμε βρίσκει εφαρμογή σε συστήματα που χρησιμοποιούν ευέλικτους πόρους για την υποστήριξη της παραγωγικής διαδικασίας. Παραπέμπουμε τους ενδιαφερόμενους αναγνώστες στους Horp και Van Oyen [4] και Andradottir, Ayhan και Down [1] για τις εκτεταμένες έρευνες της βιβλιογραφίας σχετικά με την ευελιξία του εργατικού δυναμικού.

Εξαιτίας της πολυπλοκότητας του μαθηματικού μοντέλου που χρησιμοποιείται, η έρευνα για τη βέλτιστη ανάθεση του ευέλικτου εξυπηρετητή με κόστη αναμονής έχει επικεντρωθεί σε συστήματα με δύο σταθμούς. Δύο σχετικές εργασίες πραγματοποίησε ο Farrar [2,3] όπου μελέτησε δύο παραλλαγές ενός συστήματος εκκαθάρισης (χωρίς αφίξεις) με σταθερούς εξυπηρετητές και στους δύο σταθμούς και έναν ευέλικτο εξυπηρετητή. Στην περιορισμένη εκδοχή ο ευέλικτος εξυπηρετητής μπορεί να εργαστεί μόνο στον ανάντη σταθμό, ενώ στην περίπτωση χωρίς περιορισμό μπορεί να εργαστεί και στους δύο σταθμούς. Για τις δύο παραπάνω παραλλαγές έδειξε ότι ο ευέλικτος εξυπηρετητής παραμένει σε αδράνεια ή τοποθετείται στον κατάντη σταθμό όταν ο αριθμός των εργασιών εκεί υπερβαίνει ένα ορισμένο όριο το οποίο εξαρτάται από τον αριθμό των εργασιών στην ουρά του πρώτου σταθμού. Αυτό σημαίνει ότι η βέλτιστη πολιτική χαρακτηρίζεται από μία καμπύλη αλλαγής η οποία αντιπροσωπεύει τα παραπάνω όρια για τα οποία ο ευέλικτος τοποθετείται στον κατάντη σταθμό. Επίσης έδειξε ότι η κλίση της καμπύλης αλλαγής είναι μεγαλύτερη ή ίση του -1. Ο Pandelis [5] έδειξε την εγκυρότητα των αποτελεσμάτων που λαμβάνονται από τον Farrar [2,3] για την περίπτωση που οι εργασίες θα μπορούσαν να εγκαταλείψουν το σύστημα ύστερα από την ολοκλήρωση της εξυπηρέτησης από

τον πρώτο σταθμό. Επίσης προσδιόρισε υποσύνολα του χώρου καταστάσεων όπου η βέλτιστη πολιτική μπορεί να προσδιοριστεί επακριβώς. Η ίδια δομή για τη βέλτιστη πολιτική αποδείχθηκε από τους Wu, Lewis και Veatch [10] όπου υποτέθηκε ότι οι εξυπηρετητές έχουν διαφορετικούς ρυθμούς εξυπηρέτησης και οι ανάγκες επεξεργασίας είναι ίδιες και στους δύο σταθμούς. Για το παραπάνω μοντέλο με αφίξεις και χωρίς σταθερό εξυπηρετητή στον ανάντη σταθμό οι Wu, Down και Lewis [11] έδειξαν την ύπαρξη καμπύλης αλλαγής για την βέλτιστη πολιτική, και ο Pandelis [6] επέκτεινε αυτό το αποτέλεσμα στην περίπτωση που οι εργασίες ενδέχεται να μην απαιτούν εξυπηρέτηση στο κατάντη σταθμό και οι ανάγκες επεξεργασίας δεν είναι ίδιες στους δύο σταθμούς. Τέλος, ο Pandelis [7] μελέτησε ένα μοντέλο με κόστος λειτουργίας των εξυπηρετητών, επιπρόσθετο του κόστους αναμονής, και προσδιόρισε τις συνθήκες κάτω από τις οποίες διατηρείται η δομή της καμπύλης αλλαγής της βέλτιστης πολιτικής.

Με εξαίρεση του Pandelis [5] (περιορισμένη εκδοχή), σε όλα τα άρθρα που αναφέρονται στην προηγούμενη παράγραφο έχει υποτεθεί ότι οι εξυπηρετητές μπορούν να συνεργάζονται για την διεκπεραίωση μίας εργασίας, οπότε ο συνολικός ρυθμός εξυπηρέτησης είναι ίσος με το άθροισμα των επιμέρους ρυθμών εξυπηρέτησης. Επιπλέον, σε όλα τα παραπάνω έχει υποτεθεί ότι δεν αδρανοποιούνται, εάν όχι όλοι, τουλάχιστον οι σταθεροί εξυπηρετητές. Οι παραπάνω δύο υποθέσεις αίρονται από τον Pandelis [8] στην περίπτωση ενός συστήματος εκκαθάρισης. Το ίδιο συμβαίνει και στην εργασία αυτή για το πρόβλημα με αφίξεις. Εδώ αποδεικνύουμε ότι δεν συμφέρει η αδρανοποίηση των εξυπηρετητών όταν το κόστος αναμονής είναι μεγαλύτερο στον ανάντη σταθμό. Για την περίπτωση αυτή δείχνουμε ότι όταν δεν υπάρχει σταθερός εξυπηρετητής στον ανάντη σταθμό και ο σταθερός εξυπηρετητής στον κατάντη σταθμό είναι ταχύτερος από τον ευέλικτο εξυπηρετητή, η βέλτιστη πολιτική προσδιορίζεται από μία καμπύλη αλλαγής. Επιπροσθέτως, παρέχουμε μία προϋπόθεση υπό την οποία δίνεται πάντα προτεραιότητα στον ανάντη σταθμό.

1.2 Στόχος της εργασίας

Στόχος της εργασίας είναι να αποδείξουμε όσες περισσότερες ιδιότητες της βέλτιστης πολιτικής είναι εφικτό. Συγκεκριμένα, να αποδείξουμε εάν η βέλτιστη πολιτική χαρακτηρίζεται από μία καμπύλη αλλαγής και εάν αυτή η καμπύλη είναι μη φθίνουσα. Επιπλέον, για τις ιδιότητες που δεν μπορούν να αποδειχθούν, στόχος είναι να ελεγχθεί η ισχύ τους μέσω αριθμητικών παραδειγμάτων.

Για τον προσδιορισμό της βέλτιστης πολιτικής για διάφορες τιμές των παραμέτρων του προβλήματος χρησιμοποιήσαμε το λογισμικό πακέτο Matlab, μία περιγραφή του οποίου δίνεται στο παρακάτω υποκεφάλαιο.

1.3 Υπολογιστικό πακέτο που χρησιμοποιήθηκε

Το Matlab είναι ένα λογισμικό πακέτο για υψηλής απόδοσης αριθμητικούς υπολογισμούς. Παρέχει στο χρήστη ένα διαδραστικό περιβάλλον με χιλιάδες ενσωματωμένες συναρτήσεις, κατάλληλες για την υλοποίηση απαιτητικών υπολογιστικών αναλύσεων, γραφημάτων καθώς επίσης και για την παραγωγή διαφόρων animations. Επιπλέον, το Matlab προσφέρει τη δυνατότητα επέκτασης σε ποικίλα πεδία εφαρμογών με τη αξιοποίηση την υψηλού επιπέδου γλώσσας προγραμματισμού, την οποία διαθέτει σε όλες τις εκδόσεις του.

Το Matlab αποτελεί ένα εξελιγμένο υπολογιστικό εργαλείο, το οποίο μπορεί να βρει εφαρμογή σε διάφορους τομείς της επιστήμης αλλά βέβαια και της πράξης, όπως για παράδειγμα τη μηχανική, την ιατρική, τις θετικές επιστήμες (Μαθηματικά–Φυσική), την οικονομία καθώς και γενικά τη βιομηχανική παραγωγή. Μάλιστα, το φάσμα των εφαρμογών του συγκεκριμένου πακέτου λογισμικού διευρύνεται συνεχώς και περισσότερο, αναδεικνύοντας με αυτό τον τρόπο τις πολλαπλές δυνατότητες του, όπως:

- Υψηλή απόδοση και ταχύτητα υπολογιστικών αναλύσεων.
- Δυνατότητα προσομοίωσης φυσικών συστημάτων.
- Δυνατότητα υλοποίησης αλγορίθμων.
- Υψηλής ποιότητας γραφικές απεικονίσεις και animations.
- Φιλικότητα προς το χρήστη και διαδραστικός χαρακτήρας.

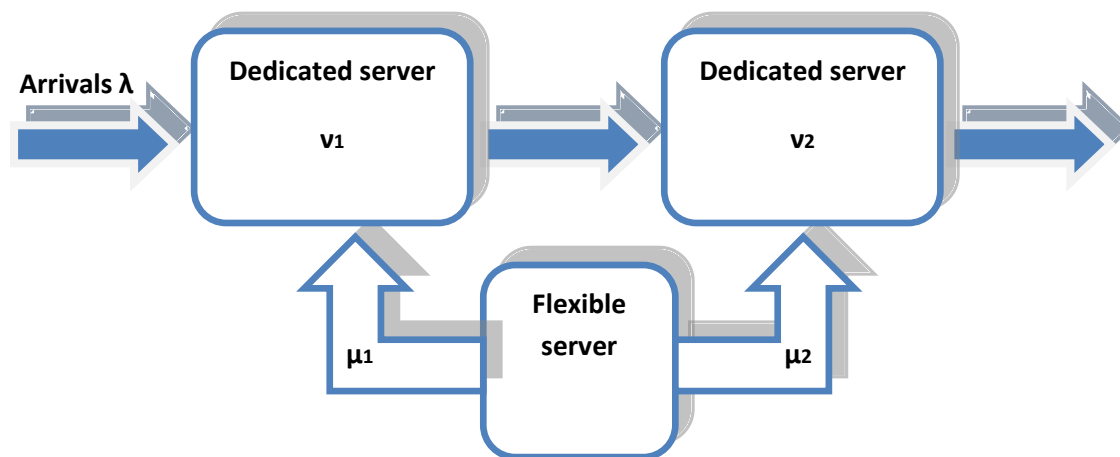
Οι ενσωματωμένες συναρτήσεις του λογισμικού παρέχουν τα απαραίτητα πακέτα εργαλείων για υπολογισμούς γραμμικής άλγεβρας, ανάλυσης δεδομένων, επεξεργασία σημάτων, αριθμητικές λύσεις κανονικών διαφορικών εξισώσεων. Οι περισσότερες από τις προαναφερόμενες συναρτήσεις εφαρμόζουν την πλέον πρόσφατη και εξελιγμένη γνώση στο κάθε τομέα επιστήμης (state-of-the-art algorithms). Επίσης ο χρήστης έχει τη δυνατότητα να αναπτύξει τις δικές του συναρτήσεις, κάνοντας χρήση της δικής του γλώσσας προγραμματισμού. Από τη στιγμή που θα αναπτυχθούν οι συναρτήσεις αυτές, λειτουργούν ως ενσωματωμένες συναρτήσεις του εν λόγω λογισμικού. Επίσης παρέχονται από το Matlab πολλές προαιρετικές εργαλειοθήκες, οι οποίες προορίζονται για την ανάπτυξη ειδικών εφαρμογών.

Κεφάλαιο 2. Περιγραφή και ανάλυση προβλήματος

Θεωρούμε ένα δίκτυο δύο διαδοχικών ουρών αναμονής όπου οι εργασίες φθάνουν σύμφωνα με την κατανομή Poisson με ρυθμό λ . Όταν τελειώσει η εξυπηρέτηση τους από τον ανάντη σταθμό (σταθμός 1) κινούνται προς τον κατάντη σταθμό (σταθμό 2) και όταν εξέλθουν από αυτόν αποχωρούν από το σύστημα (εικόνα 1). Η κάθε εργασία στο σταθμό i , $i=1,2$, χαρακτηρίζεται και από ένα γραμμικό κόστος παραμονής διαφορετικό για τον κάθε σταθμό h_i . Κάθε σταθμός εξυπηρετείται από ένα σταθερό-εξειδικευμένο εξυπηρετητή και υπάρχει και ένας ευέλικτος που τοποθετείται για βοήθεια στο σταθμό 1 ή στο σταθμό 2. Ο ευέλικτος μπορεί να τίθεται σε λειτουργία ή να απενεργοποιείται και να μεταφέρεται από τον ένα σταθμό στον άλλο. Ακόμα, υπάρχει και η δυνατότητα της αδρανοποίησης του σταθερού εξυπηρετητή στο σταθμό 1 και στο σταθμό 2, όπως επίσης και ο τερματισμός λειτουργίας όλου του παραπάνω σταθμού. Ο ευέλικτος εξυπηρετητής δεν μπορεί να συνεργάζεται με το σταθερό του κάθε σταθμού για μία εργασία, αλλά ο καθένας μπορεί να αναλάβει από μία ξεχωριστή εργασία στον ίδιο σταθμό. Επιπλέον πρέπει να τονιστεί ότι οι χρόνοι εξυπηρέτησης ακολουθούν την **εκθετική κατανομή**.

Οι ρυθμοί εξυπηρέτησης για τον κάθε σταθμό θα συμβολίζονται ως εξής :

- Για το σταθμό 1 ο ρυθμός του σταθερού εξυπηρετητή συμβολίζεται με ν_1 ενώ του ευέλικτου με μ_1 .
- Αντίστοιχα για το σταθμό 2 ο ρυθμός του σταθερού συμβολίζεται με ν_2 ενώ του ευέλικτου με μ_2 .



Εικόνα 1: Σχεδιάγραμμα λειτουργίας του συστήματος

2.1 Διαμόρφωση μοντέλου βελτιστοποίησης

Στους χρόνους των αφίξεων και αναχωρήσεων επιτρέπουμε να γίνεται επανατοποθέτηση των εξυπηρετητών και στην συνέχεια διατυπώνουμε το πρόβλημα μας ως μία διαδικασία λήψης αποφάσεων Markov με χώρο καταστάσεων του συστήματος $\{(x_1, x_2) : x_1, x_2 \geq 0\}$, όπου x_1, x_2 είναι ο αριθμός των εργασιών στο σταθμό 1 και 2 αντίστοιχα, συμπεριλαμβανομένων εκείνων που εξυπηρετούνται. Συμβολίζουμε με $A(x_1, x_2)$ το σύνολο των εφικτών τιμών των συνολικών ρυθμών εξυπηρέτησης ρ_1, ρ_2 , στην κατάσταση (x_1, x_2) , όπου $\rho_i, i = 1, 2$, είναι ο ρυθμός που αντιστοιχεί στον αντίστοιχο σταθμό. Οι τιμές ρ_1, ρ_2 είναι εφικτές εφόσον πληρούνται τα ακόλουθα: **i)** $\rho_i \in \{0, v_i, \mu_i, v_i + \mu_i\}$, **ii)** $\rho_i = 0$ εάν $x_i = 0$ (δεν αντιστοιχεί κάποιος ρυθμός εξυπηρέτησης εφόσον ο σταθμός είναι κενός), **iii)** $\rho_i < v_i + \mu_i$ εάν $x_i = 1$ (δεν υφίσταται συνεργασία) και **iv)** $(\rho_1, \rho_2) \notin \{\mu_1, v_1 + \mu_1\} \times \{\mu_2, v_2 + \mu_2\}$ (ο ευέλικτος εξυπηρετητής δε μπορεί να ανατεθεί σε δύο ουρές ταυτόχρονα). Αντί του προβλήματος συνεχούς χρόνου, θεωρούμε το ισοδύναμο πρόβλημα διακριτού χρόνου που λαμβάνεται με ομοιομορφοποίηση (uniformization), όπου, χωρίς βλάβη της γενικότητας υποθέτουμε $v_1 + v_2 + \mu_1 + \mu_2 + \lambda = 1$. Συμβολίζουμε με $V_n(x_1, x_2)$ το συνολικό ελάχιστο αναμενόμενο κόστος για n χρονικές περιόδους ξεκινώντας από την κατάσταση (x_1, x_2) μειωμένο, με έναν εκπτώτικό συντελεστή β . Η συνάρτηση του ελάχιστου κόστους $V_n(x_1, x_2)$ ικανοποιεί την ακόλουθη εξίσωση δυναμικού προγραμματισμού:

$$V_n(x_1, x_2) = h_1 x_1 + h_2 x_2 + \beta \left[\lambda V_{n-1}(x_1 + 1, x_2) + \min_{(\rho_1, \rho_2) \in A(x_1, x_2)} W_{n; \rho_1, \rho_2}(x_1, x_2) \right], \quad (1)$$

όπου,

$$W_{n; \rho_1, \rho_2}(x_1, x_2) = \rho_1 V_{n-1}(x_1 - 1, x_2 + 1) + \rho_2 V_{n-1}(x_1, x_2 - 1) + (1 - \lambda - \rho_1 - \rho_2) V_{n-1}(x_1, x_2), \quad (2)$$

και με αρχική τιμή,

$$V_0(x_1, x_2) = 0. \quad (3)$$

Οι αντίστοιχες εξισώσεις για άπειρο χρονικό ορίζοντα είναι οι εξής:

$$V(x_1, x_2) = h_1 x_1 + h_2 x_2 + \beta \left[\lambda V(x_1 + 1, x_2) + \min_{(\rho_1, \rho_2) \in A(x_1, x_2)} W_{\rho_1, \rho_2}(x_1, x_2) \right], \quad (4)$$

όπου,

$$W_{\rho_1, \rho_2}(x_1, x_2) = \rho_1 V(x_1 - 1, x_2 + 1) + \rho_2 V(x_1, x_2 - 1) + (1 - \lambda - \rho_1 - \rho_2) V(x_1, x_2). \quad (5)$$

Για το μέσο κόστος, ανά μονάδα χρόνου, έχουμε

$$J = \lim_{n \rightarrow \infty} \frac{V_n(x_1, x_2)}{n}. \quad (6)$$

Σημειωτέον, ότι το παραπάνω όριο δεν εξαρτάται από τα x_1, x_2 .

2.2 Η βέλτιστη πολιτική

Στην ενότητα αυτή θα παρουσιαστεί η βέλτιστη πολιτική του μοντέλου για πεπερασμένο χρονικό ορίζοντα, όπου κάτω από τις κατάλληλες προϋποθέσεις, η πολιτική αυτή ισχύει και για τα υπόλοιπα μοντέλα. Θα ξεκινήσουμε με ένα ευρύ χαρακτηρισμό της βέλτιστης πολιτικής που μειώνει την έρευνα για τη βέλτιστη ανάθεση των εξυπηρετητών σε ένα υποσύνολο των εφικτών τιμών των ρυθμών εξυπηρέτησης. Αρχικά με μία απλή επαγωγή μπορούμε να δούμε ότι η συνάρτηση του ελάχιστου κόστους $V_n(x_1, x_2)$ είναι μη φθίνουσα. Επειδή, δεν υπάρχει κανένα κέρδος από τη διατήρηση εργασιών στον κατάντη σταθμό, είναι λογικό να διατεθεί όσο το δυνατόν περισσότερος ρυθμός εξυπηρέτησης σε αυτό το σταθμό για να

ωθήσει εργασίες έξω από το σύστημα. Αυτό έχει καθοριστεί στην πρόταση που ακολουθεί.

Πρόταση 1: Για δεδομένο ρ_1 , η συνάρτηση $W_{n;\rho_1,\rho_2}(x_1, x_2)$ ελαχιστοποιείται όταν μεγιστοποιείται το ρ_2 .

Απόδειξη: Έστω $\rho_2 < \rho_2'$. Τότε, έχουμε

$$W_{n;\rho_1,\rho_2}(x_1, x_2) - W_{n;\rho_1,\rho_2'}(x_1, x_2) = (\rho_2' - \rho_2)[V_{n-1}(x_1, x_2) - V_{n-1}(x_1, x_2 - 1)],$$

το οποίο είναι θετικό λόγω το ότι η συνάρτηση του ελάχιστου κόστους $V_n(x_1, x_2)$ είναι μη φθίνουσα.

Συνέπεια της παραπάνω πρότασης είναι, ότι η βέλτιστη πολιτική δεν αδρανοποιεί το σταθερό εξυπηρετητή στο σταθμό 2 όταν υπάρχουν τουλάχιστον δύο εργασίες εκεί. Στη συνέχεια, όσον αφορά τη βέλτιστη ανάθεση πόρων στο σταθμό 1 ορίζουμε τη συνάρτηση $f_n(x_1, x_2)$ ως ακολούθως:

$$f_n(x_1, x_2) = V_n(x_1, x_2) - V_n(x_1 - 1, x_2 + 1), \quad x_1 \geq 1, x_2 \geq 0.$$

Σημειώστε ότι για δεδομένες τιμές ρυθμών εξυπηρέτησης ρ_1, ρ_2 το κίνητρο για να διατεθεί επιπλέον ρυθμός ρ στο σταθμό 1 δίνεται από

$$W_{n;\rho_1,\rho_2}(x_1, x_2) - W_{n;\rho_1+\rho,\rho_2}(x_1, x_2) = \rho f_{n-1}(x_1, x_2).$$

Ως εκ τούτου, το πρόσημο του $f_n(x_1, x_2)$ καθορίζει εάν θα πρέπει να προστεθεί ρυθμός εξυπηρέτησης στο σταθμό 1 ή όχι. Αυτό επιβεβαιώνεται με την ακόλουθη πρόταση.

Πρόταση 2: Για δεδομένο ρ_2 , το $W_{n;\rho_1,\rho_2}(x_1, x_2)$ ελαχιστοποιείται μεγιστοποιώντας το ρ_1 εφόσον ισχύει $f_{n-1}(x_1, x_2) \geq 0$ και ελαχιστοποιώντας το ρ_1 εφόσον $f_{n-1}(x_1, x_2) < 0$.

Απόδειξη: Έστω $\rho_1 < \rho_1'$. Τότε έχουμε

$$W_{n;\rho_1,\rho_2}(x_1, x_2) - W_{n;\rho_1',\rho_2}(x_1, x_2) = (\rho_1' - \rho_1)f_{n-1}(x_1, x_2),$$

το οποίο αποδεικνύει την παραπάνω πρόταση.

Σύμφωνα με την Πρόταση 2, ανάλογα με το πρόσημο της $f_n(x_1, x_2)$, η βέλτιστη πολιτική θα πρέπει, είτε να διαθέσει όσο περισσότερο ρυθμό εξυπηρέτησης είναι δυνατόν στον ανάντη σταθμό, ή να μην εξυπηρετήσει το σταθμό αυτό καθόλου. Ειδικότερα, όσον αφορά το σταθερό εξυπηρετητή στο σταθμό 1, η βέλτιστη πολιτική δεν θα πρέπει να σταματάει τη λειτουργία του όταν $f_n(x_1, x_2) \geq 0$ και υπάρχουν τουλάχιστον δύο εργασίες στο σταθμό 1, και να τη σταματάει όταν $f_n(x_1, x_2) < 0$.

2.3 Η βέλτιστη πολιτική για $h_1 \geq h_2$

Όταν δεν είναι λιγότερο δαπανηρό να υπάρχουν εργασίες στο σταθμό 1 σε σύγκριση με το σταθμό 2, είναι λογικό να μην αδρανοποιούνται οι εξυπηρετητές και να παραμένουν εργασίες στον παραπάνω σταθμό. Αυτό είναι αποτέλεσμα της παρακάτω λήμμα.

Λήμμα 1: Έστω $h_1 \geq h_2$. Τότε $f_n(x_1, x_2) \geq 0$ για όλα τα $x_1 \geq 1, x_2 \geq 0$.

Απόδειξη: Η απόδειξη γίνεται με επαγωγική υπόθεση ($f_{n-1}(x_1, x_2) \geq 0$ όταν $h_1 \geq h_2$).

Έστω ότι ρ_1^*, ρ_2^* είναι η βέλτιστη κατανομή στην κατάσταση (x_1, x_2) . Τότε,

$$\begin{aligned} f_n(x_1, x_2) &\geq h_1 - h_2 + \beta \left[\lambda f_{n-1}(x_1 + 1, x_2) + W_{n; \rho_1^*, \rho_2^*}(x_1, x_2) - W_{n; 0, \rho_2^*}(x_1 - 1, x_2 + 1) \right] \\ &= h_1 - h_2 + \beta \left[\lambda f_{n-1}(x_1 + 1, x_2) + \rho_2^* f_{n-1}(x_1, x_2 - 1) + (1 - \lambda - \rho_1^* - \rho_2^*) f_{n-1}(x_1, x_2) \right] \geq 0, \end{aligned}$$

από την προαναφερθείσα επαγωγική υπόθεση.

Ως αποτέλεσμα των προτάσεων που αναφέρθηκαν στην προηγούμενη ενότητα αλλά και του παραπάνω αποτελέσματος (Λήμμα 1), η βέλτιστη ανάθεση του εξυπηρετητή, για τις περιπτώσεις $h_1 \geq h_2$, είναι είτε ρητά καθορισμένη (σε ορισμένες περιπτώσεις με μία εργασία σε έναν ή και στους δύο σταθμούς) ή περιορίζεται σε δύο επιλογές (βλέπε Πίνακα 1).

	Περίπτωση 1	Περίπτωση 2	Περίπτωση 3	Περίπτωση 4
	$v_1 \geq \mu_1$ $v_2 \geq \mu_2$	$v_1 \geq \mu_1$ $v_2 < \mu_2$	$v_1 < \mu_1$ $v_2 \geq \mu_2$	$v_1 < \mu_1$ $v_2 < \mu_2$
(1,1)	(v_1, v_2)	(v_1, μ_2)	(μ_1, v_2)	(v_1, μ_2) ή (μ_1, v_2)
(1, x_2)	$(v_1, v_2 + \mu_2)$		$(v_1, v_2 + \mu_2)$ ή (μ_1, v_2)	
(x_1 , 1)	$(v_1 + \mu_1, v_2)$	$(v_1 + \mu_1, v_2)$ ή (v_1, μ_2)	$(v_1 + \mu_1, v_2)$	$(v_1 + \mu_1, v_2)$ ή (v_1, μ_2)
(x_1 , x_2)	$(v_1 + \mu_1, v_2)$ ή $(v_1, v_2 + \mu_2)$			

Πίνακας 1: Βέλτιστη ανάθεση πόρων για $h_1 \geq h_2$

Στο υπόλοιπο αυτής της ενότητας θα αποδείξουμε κάποιες ιδιότητες που χαρακτηρίζουν τη βέλτιστη πολιτική στην περίπτωση όπου δεν υπάρχει σταθερός εξυπηρετητής στο σταθμό 1 και ο σταθερός εξυπηρετητής στο σταθμό 2 είναι πιο γρήγορος από τον ευέλικτο στην αντίστοιχη θέση. Συγκεκριμένα δείχνουμε ότι η βέλτιστη πολιτική χαρακτηρίζεται από μία καμπύλη αλλαγής και καθορίζουμε μία συνθήκη υπό την οποία ανατίθεται ο ευέλικτος πάντα στο σταθμό 1. Όλα τα παραπάνω αναφέρονται στο θεώρημα 1 το οποίο θα αναλυθεί στην συνέχεια. Η συνάρτηση ελάχιστου κόστους στην περίπτωση αυτή ($v_1 = 0$ και $v_2 \geq \mu_2$) είναι:

$$V_n(x_1, x_2) = h_1 x_1 + h_2 x_2 + \beta \left[\begin{array}{l} \lambda V_{n-1}(x_1 + 1, x_2) + \\ \min\{\mu_1 V_{n-1}(x_1 - 1, x_2 + 1) + v_2 V_{n-1}(x_1, x_2 - 1) + (1 - \lambda - \mu_1 - v_2) V_{n-1}(x_1, x_2), \\ (v_2 + \mu_2) V_{n-1}(x_1, x_2 - 1) + (1 - \lambda - v_2 - \mu_2) V_{n-1}(x_1, x_2)\} \end{array} \right], \quad (7)$$

για $x_1, x_2 > 1$.

Ακόμα,

$$V_n(x_1, 1) = h_1 x_1 + h_2 + \beta [\lambda V_{n-1}(x_1 + 1, 1) + \mu_1 V_{n-1}(x_1 - 1, 2) + \nu_2 V_{n-1}(x_1, 0) + (1 - \lambda - \mu_1 - \nu_2) V_{n-1}(x_1, 1)], \quad (8)$$

με $x_1 > 1$,

$$V_n(1, x_2) = h_1 + h_2 x_2 + \beta \left[\begin{array}{l} \lambda V_{n-1}(2, x_2) + \\ \min\{\mu_1 V_{n-1}(0, x_2 + 1) + \nu_2 V_{n-1}(1, x_2 - 1) + (1 - \lambda - \mu_1 - \nu_2) V_{n-1}(1, x_2), \\ (\nu_2 + \mu_2) V_{n-1}(1, x_2 - 1) + (1 - \lambda - \mu_2 - \nu_2) V_{n-1}(1, x_2)\} \end{array} \right], \quad (9)$$

με $x_2 > 1$,

$$V_n(1, 1) = h_1 + h_2 + \beta [\lambda V_{n-1}(2, 1) + \mu_1 V_{n-1}(0, 2) + \nu_2 V_{n-1}(1, 0) + (1 - \lambda - \mu_1 - \nu_2) V_{n-1}(1, 1)]. \quad (10)$$

Όταν μία από τις δύο ουρές είναι άδεια, φαίνεται εύκολα ότι η βέλτιστη πολιτική είναι να ανατίθενται όσο περισσότεροι πόροι είναι δυνατόν στη μη κενή ουρά, έτσι οι εξισώσεις βελτιστοποίησης έχουν την παρακάτω μορφή:

$$V_n(x_1, 0) = h_1 x_1 + \beta [\lambda V_{n-1}(x_1 + 1, 0) + \mu_1 V_{n-1}(x_1 - 1, 1) + (1 - \lambda - \mu_1) V_{n-1}(x_1, 0)], \quad \text{με } x_1 \geq 1, \quad (11)$$

$$V_n(0, 1) = h_2 + \beta [\lambda V_{n-1}(1, 1) + \nu_2 V_{n-1}(0, 0) + (1 - \lambda - \nu_2) V_{n-1}(0, 1)], \quad \text{για } x_2 = 1, \quad (12)$$

$$V_n(0, x_2) = h_2 x_2 + \beta [\lambda V_{n-1}(1, x_2) + (\nu_2 + \mu_2) V_{n-1}(0, x_2 - 1) + (1 - \lambda - \mu_2 - \nu_2) V_{n-1}(0, x_2)]. \quad (13)$$

για $x_2 > 1$,

Όταν και οι δύο ουρές είναι άδειες η εξίσωση βελτιστοποίησης διαμορφώνεται ως εξής:

$$V_n(0, 0) = \beta [\lambda V_{n-1}(1, 0) + (1 - \lambda) V_{n-1}(0, 0)]. \quad (14)$$

Στην συνέχεια ορίζουμε την συνάρτηση $d_n(x_1, x_2)$ ως εξής:

$$d_n(x_1, x_2) = \mu_1 f_n(x_1, x_2) + \mu_2 [V_n(x_1, (x_2 - 1)^+) - V_n(x_1, x_2)], \quad \text{με } x_1 \geq 1, \quad x_2 \geq 0, \quad (15)$$

η οποία για $x_1, x_2 > 1$ είναι ίση με τη διαφορά μεταξύ του δεύτερου και του πρώτου όρου που βρίσκονται μέσα στα άγκιστρα της εξίσωσης (7). Για $x_2 > 1$ η συνάρτηση $d_n(x_1, x_2)$ δίνει το κίνητρο για να τοποθετηθεί ο ευέλικτος εξυπηρετητής στον πρώτο σταθμό αντί του δεύτερου. Έπειτα με την χρήση των εξισώσεων (7)-(14) και, όπου είναι εφαρμόσιμο, των ταυτοτήτων $\min(a, b) = a + (b - a)^-$ και $\min(a, b) = b - (b - a)^+$ για

τον πρώτο και δεύτερο όρο, αντίστοιχα, για τις δύο διαφορές που εμφανίζονται στην εξίσωση (15), παίρνουμε τις παρακάτω αναδρομικές εξισώσεις για την $d_n(x_1, x_2)$:

$$d_n(x_1, 0) = \mu_1(h_1 - h_2) + \beta[\lambda d_{n-1}(x_1 + 1, 0) + \mu_2 d_{n-1}(x_1, 0) + \mu_1^2 f_{n-1}(x_1 - 1, 1)\mathbf{1}(x_1 > 1) + \mu_1 v_2 [V_{n-1}(x_1, 0) - V_{n-1}(x_1 - 1, 0)]] \text{ για } x_1 > 0, \quad (16)$$

$$d_n(x_1, 1) = \mu_1(h_1 - h_2) - \mu_2 h_2 + \beta[\lambda d_{n-1}(x_1 + 1, 1) + v_2 d_{n-1}(x_1, 0) + \mu_1 d_{n-1}(x_1 - 1, 2)^+ \mathbf{1}(x_1 > 1) + \mu_2 d_{n-1}(x_1, 1)] \text{ για } x_1 > 0, \quad (17)$$

$$d_n(1, x_2) = \mu_1(h_1 - h_2) - \mu_2 h_2 + \beta[\lambda d_{n-1}(2, x_2) + v_2 d_{n-1}(1, x_2 - 1) + \mu_2 d_{n-1}(1, x_2 - 1)^- \mathbf{1}(x_2 > 2) + \mu_1 d_{n-1}(1, x_2)^- + \mu_2 d_{n-1}(1, x_2)^+] \text{ για } x_2 > 1, \quad (18)$$

$$d_n(x_1, x_2) = \mu_1(h_1 - h_2) - \mu_2 h_2 + \beta[\lambda d_{n-1}(x_1 + 1, x_2) + v_2 d_{n-1}(x_1, x_2 - 1) + \mu_1 d_{n-1}(x_1 - 1, x_2 + 1)^+ + \mu_2 d_{n-1}(x_1, x_2 - 1)^- \mathbf{1}(x_2 > 2) + \mu_1 d_{n-1}(x_1, x_2)^- + \mu_2 d_{n-1}(x_1, x_2)^+] \text{ για } x_1 > 1, x_2 > 1. \quad (19)$$

Στην συνέχεια θα προχωρήσουμε στην διατύπωση του Θεωρήματος όπου με την χρήση των παραπάνω θα πραγματοποιηθούν και οι ανάλογες αποδείξεις.

Θεώρημα 1: Έστω $h_1 \geq h_2$, $v_1 = 0$, και $v_2 \geq \mu_2$. Τότε,

- 1) Για κάθε $x_1 \geq 1$ υπάρχει μία τιμή $t(x_1) \geq 2$ τέτοια ώστε η βέλτιστη πολιτική να τοποθετεί τον ευέλικτο εξυπηρετητή στο σταθμό 2 εάν $x_2 \geq t(x_1)$, ενώ διαφορετικά τοποθετείται στο σταθμό 1.
- 2) Εάν ισχύει $\mu_1(h_1 - h_2) \geq \mu_2 h_2$, είναι βέλτιστο να τοποθετείται ο ευέλικτος εξυπηρετητής στο σταθμό 1.

Σημείωση: Η συνθήκη του δεύτερου μέρους σημαίνει ότι ο ρυθμός μείωσης του κόστους από μία εξυπηρέτηση στην πρώτη ουρά είναι μεγαλύτερος από μία εξυπηρέτηση στη δεύτερη. Άρα δεδομένου ότι δεν υπάρχει σταθερός εξυπηρετητής στο σταθμό 1 είναι λογικό να τοποθετούμε τον ευέλικτο εκεί.

Απόδειξη 1: Αρκεί να δείξουμε ότι $d_n(x_1, x_2) \geq d_n(x_1, x_2 + 1) \forall x_1 > 0, x_2 \geq 0$.

Για $n=0$ έχουμε $d_0(x_1, x_2) = 0 \forall x_1 > 0, x_2 \geq 0$. Επομένως $d_0(x_1, x_2) = d_0(x_1, x_2 + 1)$ που ικανοποιεί την παραπάνω συνθήκη.

Έστω τώρα ότι ισχύει $d_{n-1}(x_1, x_2) \geq d_{n-1}(x_1, x_2 + 1)$. Με αυτή την υπόθεση θα υπολογίσουμε για την περίοδο n . Αρχικά θα δούμε για την γενική περίπτωση για $x_1 > 1$ και $x_2 \geq 2$.

$$\begin{aligned} d_n(x_1, x_2) - d_n(x_1, x_2 + 1) = & \beta\lambda[d_{n-1}(x_1 + 1, x_2) - d_{n-1}(x_1 + 1, x_2 + 1)] + \beta\nu_2[d_{n-1}(x_1, x_2 - 1) - d_{n-1}(x_1, x_2)] \\ & + \beta\mu_1[d_{n-1}(x_1, x_2)^- - d_{n-1}(x_1, x_2 + 1)^-] + \beta\mu_2[d_{n-1}(x_1, x_2)^+ - d_{n-1}(x_1, x_2 + 1)^+] \\ & + \beta\mu_1[d_{n-1}(x_1 - 1, x_2 + 1)^+ - d_{n-1}(x_1 - 1, x_2 + 2)^+] \\ & + \beta\mu_2[d_{n-1}(x_1, x_2 - 1)^-(x_2 > 2) - d_{n-1}(x_1, x_2)^-]. \end{aligned}$$

Για $x_2 \geq 2$ παρατηρούμε ότι όλοι οι όροι μέσα στις αγκύλες είναι μη αρνητικοί λόγω της επαγωγικής υπόθεσης, $d_{n-1}(x_1, x_2) \geq d_{n-1}(x_1, x_2 + 1)$, επομένως ισχύει ότι $d_n(x_1, x_2) \geq d_n(x_1, x_2 + 1)$. Στην περίπτωση που έχουμε $x_2 = 2$ δεν υπάρχει ο όρος $d_{n-1}(x_1, x_2 - 1)^-$, αλλά ο $d_{n-1}(x_1, x_2)^-$ είναι μικρότερος-ίσος του μηδενός και με το (-) μπροστά είναι μεγαλύτερος-ίσος του μηδενός και ισχύει η ανισότητα και σε αυτή την περίπτωση.

Τώρα προχωράμε στην απόδειξη των περιπτώσεων που δεν ανήκουν στην παραπάνω γενική περίπτωση. Θα ξεκινήσουμε με την παρακάτω:

$$\begin{aligned} d_n(x_1, 0) - d_n(x_1, 1) = & \mu_2 h_2 + \beta\lambda[d_{n-1}(x_1 + 1, 0) - d_{n-1}(x_1 + 1, 1)] + \beta\mu_1[-d_{n-1}(x_1 - 1, 2)^+] \\ & + \beta\mu_2[d_{n-1}(x_1, 0) - d_{n-1}(x_1, 1)] + \beta\mu_1^2 f_{n-1}(x_1 - 1, 1) \\ & + \beta\mu_1 \nu_2 [V_{n-1}(x_1 - 1, 1) - V_{n-1}(x_1 - 1, 0)] \quad , \text{ για } x_1 > 1. \end{aligned}$$

Ο τελευταίος όρος είναι μη αρνητικός διότι η συνάρτηση $V_n(x_1, x_2)$ είναι μη φθίνουσα, ενώ ο πρώτος και ο τρίτος, μέσα στις αγκύλες, λόγω της επαγωγικής υπόθεσης. Τώρα θα δούμε τι συμβαίνει με τους άλλους όρους:

1) Έστω $d_{n-1}(x_1 - 1, 2) \leq 0$. Τότε ο αντίστοιχος όρος γίνεται μηδέν και επειδή έχουμε αποδείξει ότι $f_n(x_1, x_2) \geq 0$ τότε έχουμε όλους τους όρους μη αρνητικούς.

2) Έστω $d_{n-1}(x_1-1,2) \geq 0$. Επειδή γνωρίζουμε ότι $\mu_1 f_{n-1}(x_1-1,1) \geq d_{n-1}(x_1-1,1)$ και $d_{n-1}(x_1-1,1) \geq d_{n-1}(x_1-1,2)$ φτάνουμε στο συμπέρασμα πως όλοι οι όροι είναι μη αρνητικοί.

Έτσι σύμφωνα με τα παραπάνω έχουμε αποδείξει ότι $d_n(x_1,0) \geq d_n(x_1,1) \forall x_1 > 1$.

Στην συνέχεια θα δείξουμε ότι $d_n(x_1,1) \geq d_n(x_1,2)$ για $x_1 > 1$. Έχουμε

$$\begin{aligned} d_n(x_1,1) - d_n(x_1,2) &= \beta\lambda[d_{n-1}(x_1+1,1) - d_{n-1}(x_1+1,2)] + \beta\nu_2[d_{n-1}(x_1,0) - d_{n-1}(x_1,1)] \\ &\quad + \beta\mu_1[d_{n-1}(x_1-1,2)^+ - d_{n-1}(x_1-1,3)^+] + \beta\mu_2[d_{n-1}(x_1,1) - d_{n-1}(x_1,2)^+] \\ &\quad - \beta\mu_1[d_{n-1}(x_1,2)^-]. \end{aligned}$$

Οι όροι στις αγκύλες είναι μη αρνητικοί, λόγω της επαγωγικής υπόθεσης, εκτός από τον όρο $\beta\mu_2[d_{n-1}(x_1,1) - d_{n-1}(x_1,2)^+]$. Ο όρος $[-\beta\mu_1 d_{n-1}(x_1,1)^-]$ με το αρνητικό πρόσημο μπροστά είναι θετικός. Για τον πρώτο ενεργούμε ως εξής:

1) Αν $d_{n-1}(x_1,2) \geq 0$ τότε $d_{n-1}(x_1,1) - d_{n-1}(x_1,2)^+ = d_{n-1}(x_1,1) - d_{n-1}(x_1,2) \geq 0$ λόγω επαγωγής. Επομένως σε αυτή την περίπτωση ισχύει $d_n(x_1,1) \geq d_n(x_1,2)$.

2) Αν $d_{n-1}(x_1,2) \leq 0$, οπότε $d_{n-1}(x_1,2)^+ = 0$, έχουμε τις παρακάτω περιπτώσεις:

1. Εάν $d_{n-1}(x_1,1) \geq 0$ τότε είναι φανερό ότι ισχύει η ζητούμενη ανισότητα.

2. Εάν $d_{n-1}(x_1,1) \leq 0$ τότε αναλύουμε και έχουμε

$$\begin{aligned} &\beta\nu_2[d_{n-1}(x_1,0) - d_{n-1}(x_1,1)] + \beta\mu_2 d_{n-1}(x_1,1) = \\ &= \beta(\mu_2 - \nu_2)d_{n-1}(x_1,1) + \beta\nu_2 d_{n-1}(x_1,0) \text{ όπου } \mu_2 - \nu_2 \leq 0 \text{ και } d_{n-1}(x_1,1) \leq 0 \\ &\text{άρα το γινόμενο τους είναι μη αρνητικό και } d_{n-1}(x_1,0) \geq 0 \text{ διότι είναι ίσο με} \\ &\text{την } f_{n-1}(x_1,0) \text{ που έχει αποδειχτεί ότι είναι γι' αυτές τις περιπτώσεις μη} \\ &\text{αρνητική.} \end{aligned}$$

Άρα αποδείξαμε πως ισχύει $d_n(x_1,1) \geq d_n(x_1,2)$ για $x_1 > 1$.

Μία ακόμα περίπτωση που δεν ανήκει στη γενική, που αποδείξαμε στην αρχή, και την αποδεικνύουμε ξεχωριστά είναι η $d_{n-1}(1,0) \geq d_{n-1}(1,1)$ και αναλύεται ως εξής:

$$\begin{aligned} d_n(1,0) - d_n(1,1) &= \mu_2 h_2 + \beta\lambda[d_{n-1}(2,0) - d_{n-1}(2,1)] + \beta\mu_2[d_{n-1}(1,0) - d_{n-1}(1,1)] \\ &\quad + \beta\mu_1 \nu_2[V_{n-1}(1,0) - V_{n-1}(0,0)] - \beta\nu_2 d_{n-1}(1,0) \end{aligned}$$

Ο όρος $\mu_2 h_2$ είναι θετικός όπως επίσης και οι όροι μέσα στις δύο πρώτες αγκύλες λόγω της επαγωγικής υπόθεσης. Για τους υπόλοιπους όρους έχουμε

$$\beta\mu_1\nu_2[V_{n-1}(1,0) - V_{n-1}(0,0)] - \nu_2\beta d_{n-1}(1,0) = \beta\mu_1\nu_2[V_{n-1}(0,1) - V_{n-1}(0,0)],$$

όπου αυτό που προκύπτει είναι θετικό διότι η $V_n(x_1, x_2)$ είναι μη φθίνουσα. Έτσι αποδείξαμε ότι $d_n(1,0) \geq d_n(1,1)$.

Στην συνέχεια αποδεικνύουμε ότι $d_n(1,1) \geq d_n(1,2)$ όπως φαίνεται παρακάτω:

$$\begin{aligned} d_n(1,1) - d_n(1,2) &= \beta\lambda[d_{n-1}(2,1) - d_{n-1}(2,2)] + \beta\nu_2[d_{n-1}(1,0) - d_{n-1}(1,1)] \\ &\quad + \beta\mu_2[d_{n-1}(1,1) - d_{n-1}(1,2)^+] - \beta\mu_1 d_{n-1}(1,2)^-. \end{aligned}$$

Ο πρώτος όρος μέσα στις αγκύλες είναι θετικός λόγω της υπόθεσης που έχουμε κάνει. Για τους άλλους όρους συνεχίζουμε ως εξής:

1. Εάν $d_{n-1}(1,1) < 0 \Rightarrow d_{n-1}(1,2) < 0$ και ο αντίστοιχος όρος μέσα στην αγκύλη μηδενίζεται και έχουμε:

$$\begin{aligned} &\beta\nu_2[d_{n-1}(1,0) - d_{n-1}(1,1)] + \beta\mu_2 d_{n-1}(1,1) - \beta\mu_1 d_{n-1}(1,2)^- = \\ &\beta(\mu_2 - \nu_2)d_{n-1}(1,1) + \beta\nu_2 d_{n-1}(1,0) - \beta\mu_1 d_{n-1}(1,2)^-, \end{aligned}$$

όπου ο πρώτος και ο τρίτος όρος είναι μη αρνητικός λόγω γινομένου αρνητικών αριθμών ενώ ο δεύτερος διότι για $x_2 = 0$ είναι ίσο με την $f_{n-1}(1,0)$ που έχει αποδειχτεί ότι γι' αυτές τις περιπτώσεις ότι είναι μη αρνητική.

2. Εάν $d_{n-1}(1,1) > 0$, τότε και $d_{n-1}(1,1) \geq d_{n-1}(1,2)$ λόγω της επαγωγικής υπόθεσης, άρα $d_{n-1}(1,1) \geq \max(0, d_{n-1}(1,2)) = d_{n-1}(1,2)^+$. Επομένως ο αντίστοιχος όρος θα είναι μη αρνητικός από την υπόθεση που κάναμε και οι άλλοι όροι είναι επίσης μη αρνητικοί.

Άρα αποδείξαμε ότι ισχύει $d_n(1,1) \geq d_n(1,2)$.

Μία ακόμα περίπτωση που δεν υπάγεται στη γενική περίπτωση και την αποδεικνύουμε παρακάτω είναι η εξής:

$$\begin{aligned} d_n(1,2) - d_n(1,3) &= \beta\lambda[d_{n-1}(2,2) - d_{n-1}(2,3)] + \beta\nu_2[d_{n-1}(1,1) - d_{n-1}(1,2)] + \beta\mu_2[-d_{n-1}(1,2)^-] \\ &\quad + \beta\mu_1[d_{n-1}(1,2)^- - d_{n-1}(1,3)^-] + \beta\mu_2[d_{n-1}(1,2)^+ - d_{n-1}(1,3)^+]. \end{aligned}$$

Όλοι οι όροι μέσα στις αγκύλες είναι θετικοί λόγω της αρχικής μας υπόθεσης εκτός από τον $\beta\mu_2[-d_{n-1}(1,2)^-]$ που είναι θετικός ως γινόμενο δύο αρνητικών αριθμών. Άρα καταλήγουμε πως ισχύει $d_n(1,2) \geq d_n(1,3)$.

Τέλος θα προχωρήσουμε στη γενίκευση της παραπάνω περίπτωσης, δηλαδή να δείξουμε πως ισχύει $d_n(1,x_2) \geq d_n(1,x_2+1)$ για $x_2 \geq 3$. Έχουμε

$$\begin{aligned} d_n(1,x_2) - d_n(1,x_2+1) &= \beta\lambda[d_{n-1}(2,x_2) - d_{n-1}(2,x_2+1)] + \beta\nu_2[d_{n-1}(1,x_2-1) - d_{n-1}(1,x_2)] \\ &\quad + \beta\mu_1[d_{n-1}(1,x_2)^- - d_{n-1}(1,x_2+1)^-] + \beta\mu_2[d_{n-1}(1,x_2)^+ - d_{n-1}(1,x_2+1)^+] \\ &\quad + \beta\mu_2[d_{n-1}(1,x_2-1) - d_{n-1}(1,x_2)]. \end{aligned}$$

Εύκολα καταλαβαίνουμε ότι το ζητούμενο μας ισχύει, διότι όλοι οι όροι μέσα στις αγκύλες είναι θετικοί από την επαγωγική υπόθεση.

Σύμφωνα με όλα τα παραπάνω έχουμε αποδείξει ότι ισχύει $d_n(x_1,x_2) \geq d_n(x_1,x_2+1) \forall x_1 > 0, x_2 \geq 0$ άρα και το πρώτο κομμάτι του θεωρήματος μας.

Απόδειξη 2: Αρκεί να δείξουμε ότι ισχύει $d_n(x_1,x_2) \geq 0 \forall x_1 > 0, x_2 \geq 1$ όταν ισχύει $\mu_1(h_1 - h_2) \geq \mu_2h_2$. Για $x_2 = 0$ το $d_n(x_1,0) = \mu_1f_n(x_1,0)$ για το οποίο έχουμε δείξει ότι είναι μη αρνητικό.

Για $n=0$ έχουμε $d_0(x_1,x_2) = 0 \forall x_1 > 0, x_2 \geq 0$ που ικανοποιεί το παραπάνω.

Έστω ότι $d_{n-1}(x_1,x_2) \geq 0 \forall x_1 > 0, x_2 \geq 0$ τότε από την γενική εξίσωση έχουμε:

$$\begin{aligned} d_n(x_1,x_2) &= \mu_1(h_1 - h_2) - \mu_2h_2 \\ &\quad + \beta\lambda[\lambda d_{n-1}(x_1+1,x_2) + \nu_2 d_{n-1}(x_1,x_2-1) + \mu_1 d_{n-1}(x_1-1,x_2+1)^+ \\ &\quad + \mu_2 d_{n-1}(x_1,x_2-1)^- \mathbf{1}(x_2 > 2) + \mu_1 d_{n-1}(x_1,x_2)^- + \mu_2 d_{n-1}(x_1,x_2)^+] \quad \text{για } x_1 > 1, \\ &\quad x_2 > 1, \end{aligned}$$

όπου όλοι οι όροι μέσα στην αγκύλη είναι μη αρνητικοί λόγω της επαγωγικής υπόθεσης και επειδή έχουμε $\mu_1(h_1 - h_2) \geq \mu_2h_2$, τότε βλέπουμε ότι $d_n(x_1,x_2) \geq 0 \forall x_1 > 1, x_2 > 1$. Όμοια αποδεικνύεται ότι $d_n(1,x_2) \geq 0 \forall x_2 > 1$ και $d_n(x_1,1) \geq 0 \forall x_1 > 0$.

Σύμφωνα με όλα τα παραπάνω έχουμε αποδείξει ότι ισχύει $d_n(x_1, x_2) \geq 0$ $\forall x_1 > 0, x_2 \geq 0$ άρα και το δεύτερο κομμάτι του θεωρήματος μας.

2.3.1 Ιδιότητες προς έλεγχο

Για όλες τις περιπτώσεις θα ελέγξουμε εάν ισχύει η μοναδικότητα του σημείου αλλαγής, όπως είναι λογικό, ενώ για όλες τις περιπτώσεις θα γίνει έλεγχος για τη μονοτονία αυτών των σημείων. Θα περιμένουμε η ακολουθία των σημείων να είναι μη φθίνουσα, δηλαδή το επόμενο σημείο αλλαγής να είναι μεγαλύτερο ή ίσο με το προηγούμενο. Τα παραπάνω θα προσπαθήσουμε να δείξουμε ότι ισχύουν με την βοήθεια του προγράμματος μας (Matlab) χρησιμοποιώντας τις εξισώσεις του ελάχιστου κόστους έτσι όπως διαμορφώνονται για κάθε περίπτωση.

2.4 Η βέλτιστη πολιτική για $h_1 < h_2$

Όταν είναι πιο ακριβό να υπάρχουν εργασίες στο δεύτερο σταθμό από ότι στον πρώτο, μπορεί να συμφέρει να μην εισέρχονται εργασίες στο σταθμό 2, το οποίο προϋποθέτει να μην παρέχονται πόροι στο σταθμό 1 ούτως ώστε να μην εξυπηρετούνται οι εργασίες εκεί. Σύμφωνα με την Πρόταση 2 αυτό θα συμβεί στις καταστάσεις με $f_n(x_1, x_2) < 0$. Επομένως η βέλτιστη τοποθέτηση του ευέλικτου εξυπηρετητή θα δίνεται από τον πίνακα 1 με την προσθήκη της ανάθεσης $(0, \max\{v_2, \mu_2\})$ εάν $x_2 = 1$ και $(0, v_2 + \mu_2)$ εάν $x_2 > 1$ (πίνακας 2).

Επειδή ο αριθμός των πιθανών βέλτιστων αναθέσεων είναι μεγαλύτερος από ότι στην περίπτωση για $h_1 \geq h_2$ η ανάλυση γίνεται πολύ πιο δύσκολη και δεν είμαστε σε θέση να αποδείξουμε γενικές ιδιότητες της βέλτιστης πολιτικής όπως πριν στο Θεώρημα 1. Αντ' αυτού, κάνουμε την ακόλουθη υπόθεση σχετικά με τη δομή της βέλτιστης πολιτικής.

Υπόθεση: Για ένα σταθερό αριθμό εργασιών στο σταθμό 1, ο ρυθμός εξυπηρέτησης που διατίθεται στον εν λόγω σταθμό από τη βέλτιστη πολιτική μειώνεται καθώς ο αριθμός των εργασιών στο σταθμό 2 αυξάνεται.

Αυτή η υπόθεση συνεπάγεται την ακόλουθη δομή της βέλτιστης πολιτικής. Για $v_1 \geq \mu_1$ και μία εργασία στο σταθμό 1 υπάρχει ένα μοναδικό σημείο αλλαγής: όταν ο αριθμός των εργασιών στο σταθμό 2 είναι κάτω από αυτό το σημείο ο σταθερός

εξυπηρετητής στο σταθμό 1 χρησιμοποιείται, αλλιώς παραμένει αδρανής. Για $v_1 < \mu_1$ και μία εργασία στο σταθμό 1 υπάρχουν δύο σημεία αλλαγής: όταν ο αριθμός των εργασιών στο σταθμό 2 είναι κάτω από το κατώτερο σημείο αλλαγής ο ευέλικτος τοποθετείται στο σταθμό 1, όταν είναι μεταξύ των δύο σημείων ο σταθερός του σταθμού 1 χρησιμοποιείται, και όταν είναι πιο πάνω από το ανώτερο σημείο κανένας εξυπηρετητής δεν τοποθετείται στο σταθμό 1. Για περισσότερες από μία εργασίες στο σταθμό 2 η βέλτιστη πολιτική καθορίζεται από δύο καμπύλες αλλαγής: για αριθμό εργασιών στο σταθμό 2 κάτω από την πρώτη καμπύλη ο σταθμός 1 εξυπηρετείται με το σταθερό αλλά και με τον ευέλικτο εξυπηρετητή, για αριθμό εργασιών ανάμεσα στις δύο καμπύλες εξυπηρετείται μόνο με το σταθερό του εξυπηρετητή, και για εργασίες που ο αριθμός τους είναι πάνω από την ανώτερη καμπύλη δεν εξυπηρετείται καθόλου.

	Περίπτωση 1	Περίπτωση 2	Περίπτωση 3	Περίπτωση 4
	$v_1 \geq \mu_1$ $v_2 \geq \mu_2$	$v_1 \geq \mu_1$ $v_2 < \mu_2$	$v_1 < \mu_1$ $v_2 \geq \mu_2$	$v_1 < \mu_1$ $v_2 < \mu_2$
(1,1)	(v_1, v_2) ή $(0, v_2)$	(v_1, μ_2) ή $(0, \mu_2)$	(μ_1, v_2) ή $(0, v_2)$	(v_1, μ_2) ή (μ_1, v_2) ή $(0, \mu_2)$
(1, x_2)	$(v_1, v_2 + \mu_2)$ ή $(0, v_2 + \mu_2)$		$(v_1, v_2 + \mu_2)$ ή (μ_1, v_2) ή $(0, v_2 + \mu_2)$	
(x_1 , 1)	$(v_1 + \mu_1, v_2)$ ή $(0, v_2)$	$(v_1 + \mu_1, v_2)$ ή (v_1, μ_2) ή $(0, \mu_2)$	$(v_1 + \mu_1, v_2)$ ή $(0, v_2)$	$(v_1 + \mu_1, v_2)$ ή (v_1, μ_2) ή $(0, \mu_2)$
(x_1, x_2)	$(v_1 + \mu_1, v_2)$ ή $(v_1, v_2 + \mu_2)$ ή $(0, v_2 + \mu_2)$			

Πίνακας 2: Βέλτιστη ανάθεση πόρων για $h_1 < h_2$

2.4.1 Ιδιότητες προς έλεγχο

Στις παραπάνω περιπτώσεις που ισχύει $h_1 < h_2$ θεωρούμε ότι ισχύει η μοναδικότητα του σημείου αλλαγής του ευέλικτου εξυπηρετητή και μέσα από ένα σύνολο αριθμητικών παραδειγμάτων θα προσπαθήσουμε να ελέγξουμε την παραπάνω θεωρία μας. Ακόμα θα μελετήσουμε τη μονοτονία των σημείων αυτών που θεωρούμε λογικό ότι πρέπει να είναι μη φθίνουσα. Γι' αυτές τις περιπτώσεις θα γίνει έλεγχος για τη μοναδικότητα και του σημείου τερματισμού λειτουργίας του σταθερού στο σταθμό 1, όπως επίσης θα γίνει έλεγχος και για την μονοτονία των σημείων αυτών που όπως θα θεωρήσουμε αρχικά, σύμφωνα με τη λογική, δεν θα είναι φθίνουσα. Όλα τα παραπάνω θα ελεγχθούν με το πρόγραμμα μας (Matlab) χρησιμοποιώντας τις εξισώσεις του ελάχιστου κόστους έτσι όπως διαμορφώνονται για κάθε περίπτωση.

Κεφάλαιο 3. Αριθμητικά αποτελέσματα

Τα συμπεράσματα στα οποία καταλήξαμε για κάθε περίπτωση προήλθαν από έναν αριθμό επαναλήψεων της τάξεως 10^3 με τυχαίες τιμές που προήλθαν από γεννήτρια τυχαίων αριθμών και ακολουθούσαν τους περιορισμούς της εκάστοτε περίπτωσης. Τα παρακάτω αποτελέσματα βασίζονται στον αριθμό των επαναλήψεων.

3.1 Περιγραφή αλγορίθμου

Για την επίλυση κάθε περίπτωσης χρησιμοποιούμε τον αλγόριθμο Value-iteration για το κριτήριο μέσου κόστους [9]. Για την εφαρμογή του αλγορίθμου θεωρήσαμε μέγιστο αριθμό εργασιών σε κάθε σταθμό 300.

Βήμα 1^ο: Διαλέγουμε ως αρχική κατάσταση $V_0(x_1, x_2) = 0 \quad \forall x_1, x_2$. Θέτουμε $n=1$.

Βήμα 2^ο: Υπολογίζουμε το συνολικό ελάχιστο κόστος για n περιόδους από τις εξισώσεις (1), (2).

Βήμα 3^ο: Υπολογίζουμε τα όρια

$$\text{Κάτω όριο: } m_n = \min\{V_n(x_1, x_2) - V_{n-1}(x_1, x_2)\} \quad \forall x_1, x_2$$

$$\text{Άνω όριο: } M_n = \max\{V_n(x_1, x_2) - V_{n-1}(x_1, x_2)\} \quad \forall x_1, x_2.$$

Ο αλγόριθμος σταματάει όταν $0 \leq M_n - m_n \leq \epsilon m_n$ όπου το ϵ είναι ένας πολύ μικρός αριθμός και στην περίπτωση μας είναι $\epsilon = 0.001$. Αλλιώς πηγαίνουμε στο βήμα 4.

Βήμα 4^ο: Θέτουμε $n=n+1$ και πηγαίνουμε πάλι στο βήμα 2.

Το μέσο κόστος είναι πεπερασμένο όταν το σύστημα μας είναι ευσταθές. Για να συμβεί αυτό πρέπει ο ρυθμός αφίξεων (λ) να είναι μικρότερος από το ρυθμό εξυπηρέτησης σε κάθε ουρά. Άρα ο μέγιστος ρυθμός αφίξεων δίνεται από την επίλυση του παρακάτω προβλήματος.

Έστω δ_1, δ_2 το ποσοστό του χρόνου που ο ευέλικτος είναι στην ουρά 1 και 2 αντίστοιχα, τότε ο ρυθμός εξυπηρέτησης σε κάθε ουρά είναι:

- Ουρά 1: Ρυθμός εξυπηρέτησης: $\nu_1 + \mu_1 \delta_1$, $\delta_1 \geq 0$,

- Ουρά 2: Ρυθμός εξυπηρέτησης: $v_2 + \mu_2 \delta_2$, $\delta_2 \geq 0$.

Επομένως εμείς θέλουμε τον μέγιστο ρυθμό αφίξεων υπό τις παραπάνω συνθήκες, τον οποίο παίρνουμε λύνοντας το ακόλουθο πρόβλημα βελτιστοποίησης.

$$\begin{aligned} \max \lambda \\ \text{s.t } \lambda < v_1 + \mu_1 \delta_1 \\ \lambda < v_2 + \mu_2 \delta_2 \\ \delta_1 + \delta_2 \leq 1 \\ \delta_1 \geq 0, \delta_2 \geq 0. \end{aligned}$$

Αρχικά δημιουργούμε την συνάρτηση Lagrange:

$$L(\lambda_1, \lambda_2, \lambda_3, \delta_1, \delta_2) = \lambda - \lambda_1(\lambda - v_1 - \mu_1 \delta_1) - \lambda_2(\lambda - v_2 - \mu_2 \delta_2) - \lambda_3(\delta_1 + \delta_2 - 1),$$

με $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\lambda_3 \geq 0$ και $\delta_1 \geq 0$, $\delta_2 \geq 0$.

Οι Karush–Kuhn–Tucker (KKT) συνθήκες είναι:

- 1) $\lambda < v_1 + \mu_1 \delta_1$
- 2) $\lambda < v_2 + \mu_2 \delta_2$
- 3) $\partial L / \partial \delta_1 = 0$
- 4) $\partial L / \partial \delta_2 = 0$
- 5) $\partial L / \partial \lambda = 0$
- 6) $\lambda_1(\lambda - v_1 - \mu_1 \delta_1) = 0$
- 7) $\lambda_2(\lambda - v_2 - \mu_2 \delta_2) = 0$
- 8) $\lambda_3(\delta_1 + \delta_2 - 1) = 0$
- 9) $\delta_1 \geq 0$, $\delta_2 \geq 0$.
- 10) $\lambda_1 \geq 0$, $\lambda_2 \geq 0$, $\lambda_3 \geq 0$

Λύνοντας τις Karush–Kuhn–Tucker (KKT) συνθήκες για το παραπάνω πρόβλημα προκύπτει η εξής λύση για τον μέγιστο ρυθμό αφίξεων:

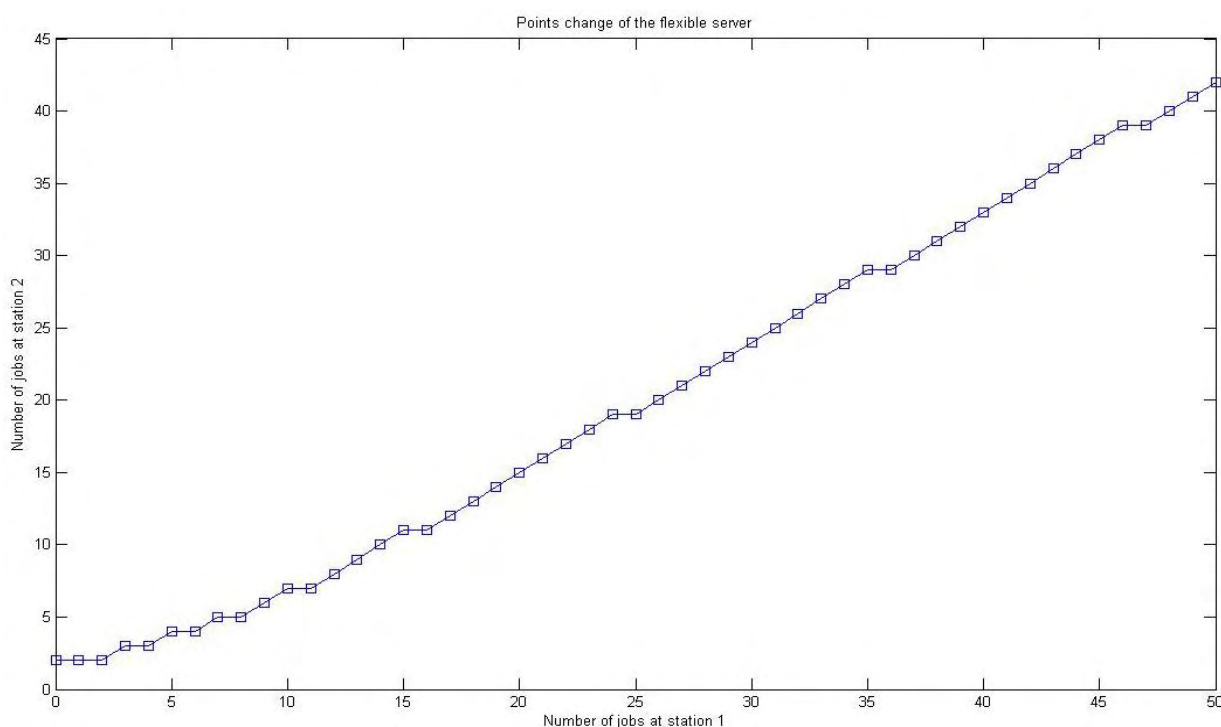
- i)** $\lambda^* = v_1 + \mu_1$ εάν $v_1 + \mu_1 \leq v_2$, **ii)** $\lambda^* = v_2 + \mu_2$ εάν $v_2 + \mu_2 \leq v_1$, **iii)**
 $\lambda^* = (\mu_1 v_2 + v_1 \mu_2 + \mu_1 \mu_2) / (\mu_1 + \mu_2)$ εάν δεν ικανοποιείται καμία από τις δύο παραπάνω συνθήκες.

3.2 Αποτελέσματα για $h_1 \geq h_2$

Σε αυτή την ενότητα θα γίνει παρουσίαση των αποτελεσμάτων της κάθε περίπτωσης για την οποία ισχύει $h_1 \geq h_2$. Θα παρουσιαστούν τα αποτελέσματα που αφορούν τη μοναδικότητα του σημείου αλλαγής σε κάθε περίπτωση, όπως επίσης και τα αποτελέσματα για τον έλεγχο της μονοτονίας των σημείων αυτών. Επιπλέον, θα παρουσιάσουμε ένα αριθμητικό παράδειγμα για να κατανοήσουμε τον τρόπο που επηρεάζει κάθε παράμετρος του προβλήματος τη λήψη των αποφάσεων μας.

3.2.1 Περίπτωση 1: $v_1 \geq \mu_1$ και $v_2 \geq \mu_2$

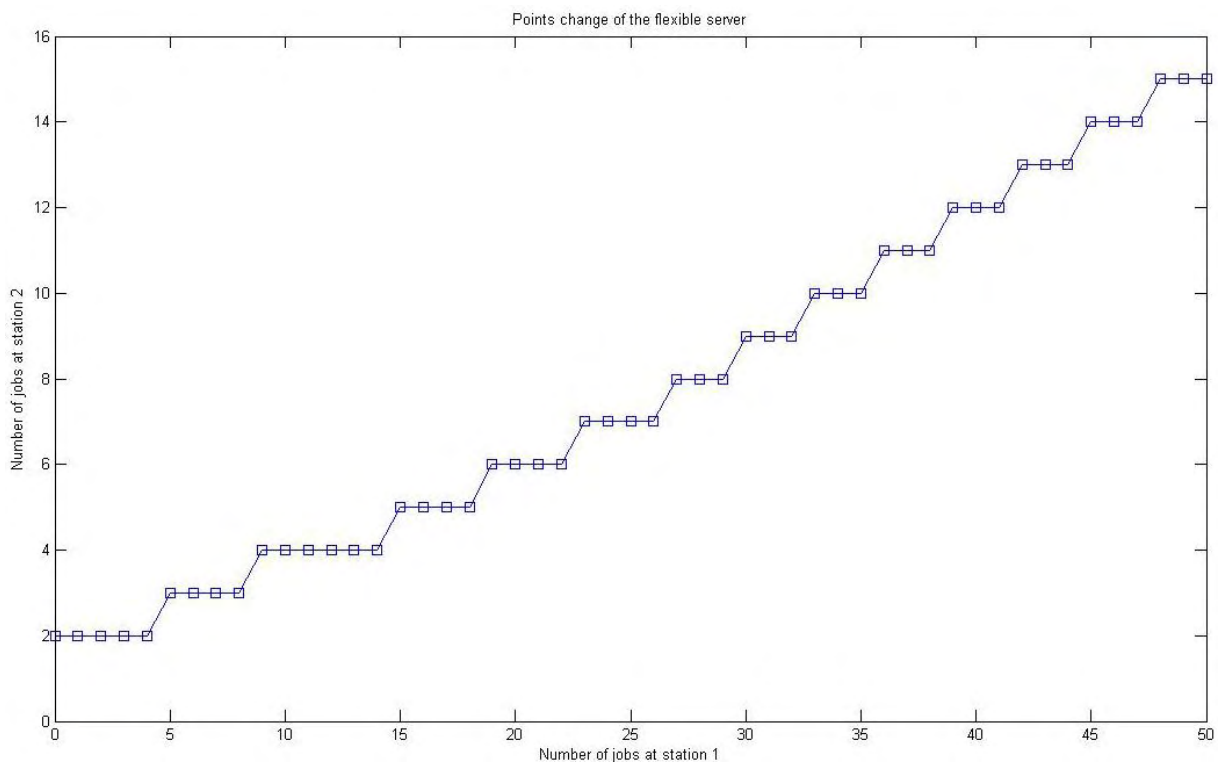
Για την περίπτωση αυτή φαίνεται, σύμφωνα πάντα από τα αποτελέσματα των παραδειγμάτων, να ισχύει η ύπαρξη μοναδικού σημείου αλλαγής του ευέλικτου από τον πρώτο σταθμό στο δεύτερο. Επίσης σε αυτή την περίπτωση ισχύει και η μονοτονία των σημείων αυτών. Γι' αυτή την περίπτωση θα δοθεί ένα παράδειγμα παρακάτω για να γίνει κατανοητός ο ρόλος των παραμέτρων μας στη λήψη της απόφασης αλλαγής. Το παρακάτω διάγραμμα που θα παρουσιαστεί είναι για τις εξής αρχικές τιμές: $v_1 = 0.29$, $v_2 = 0.16$, $\mu_1 = 0.10$, $\mu_2 = 0.14$, $h_1 = 7$, $h_2 = 2$ και $\lambda = 0.15$.



Εικόνα 2: Διάγραμμα σημείων αλλαγής με αρχικές τιμές παραμέτρων

Από το διάγραμμα είναι εμφανές ότι η αλλαγή του ευέλικτου πραγματοποιείται σε μεγαλύτερο σημείο ή στο ίδιο, κάτι το οποίο σημαίνει, πως ισχύει η μη φθίνουσα μονοτονία των σημείων αλλαγής.

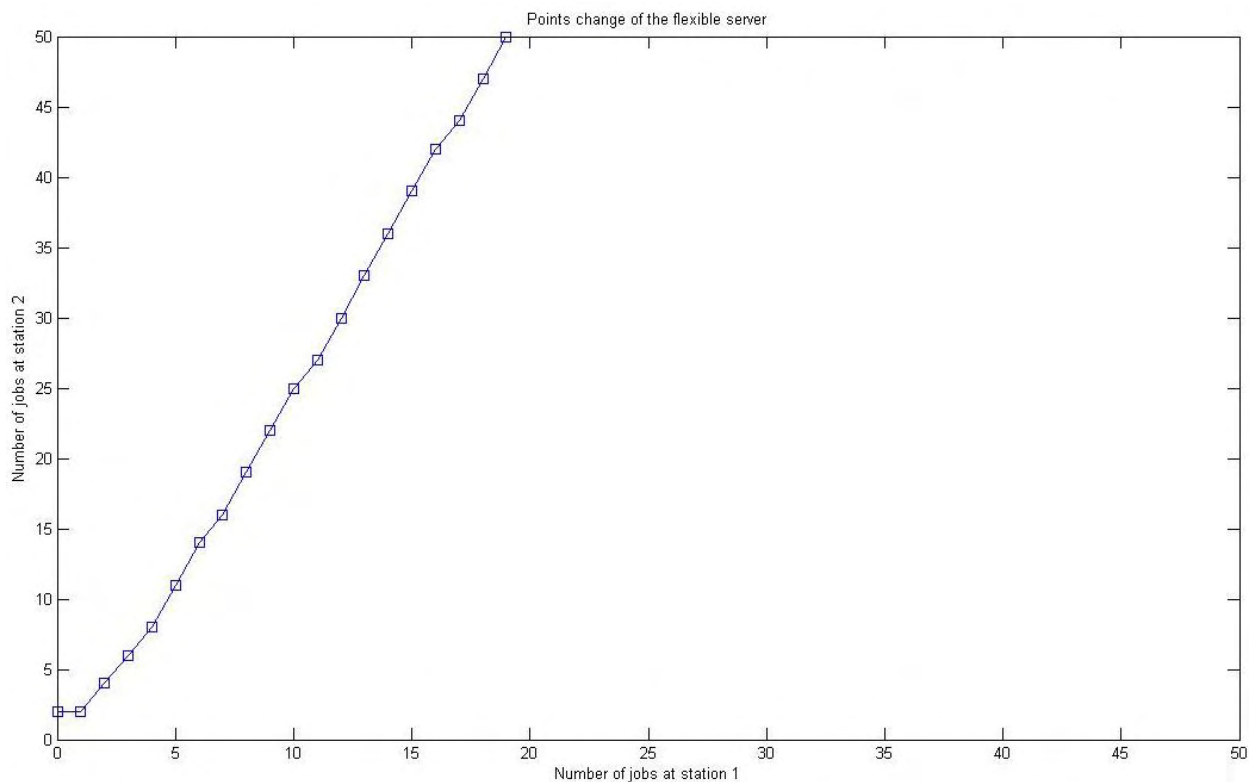
Αρχικά θα παρατηρήσουμε το πώς επηρεάζει η αύξηση του ρυθμού εξυπηρέτησης του σταθερού εξυπηρετητή στο σταθμό 1, από την αρχική του τιμή στην $\nu_1 = 0.35$. Διατηρώντας το ρυθμό αφίξεων, τα κόστη και τους υπόλοιπους ρυθμούς εξυπηρέτησης σταθερά το νέο διάγραμμα διαμορφώνεται ως εξής:



Εικόνα 3: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου ν_1

Με αυτή την αλλαγή είναι φανερό πως ο ευέλικτος τοποθετείται στο σταθμό 2 πιο νωρίς από ότι πριν και αυτό γίνεται, διότι αυξάνοντας το ρυθμό εξυπηρέτησης του σταθερού εξυπηρετητή στο σταθμό 1, μειώσαμε την ανάγκη για πρόσθετη βοήθεια εκεί. Σε αντίθετη περίπτωση με μείωση του ίδιου ρυθμού εξυπηρέτησης θα αυξανόταν η ανάγκη για πρόσθετη βοήθεια στο σταθμό 1 κι έτσι οι αλλαγές θα πραγματοποιούνταν πιο αργά από ότι με τις αρχικές τιμές.

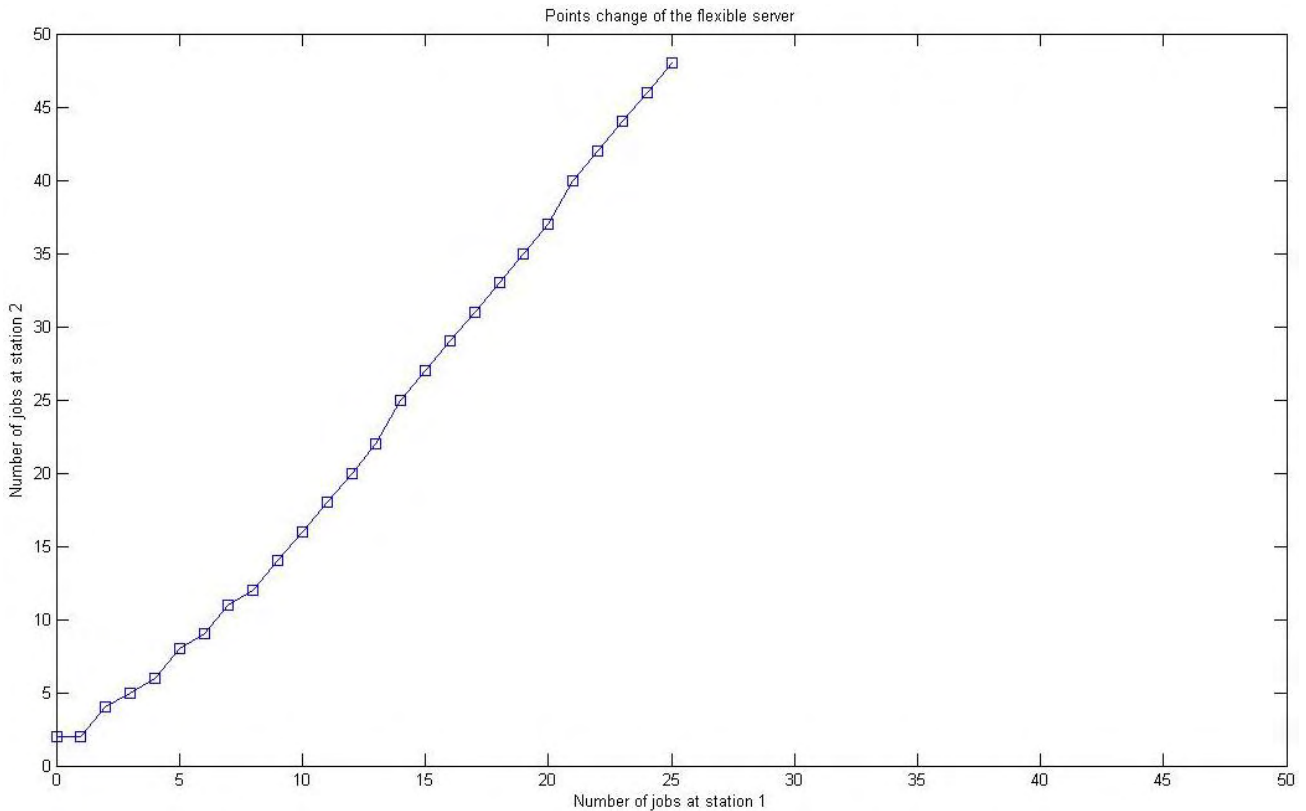
Στη συνέχεια πραγματοποιούμε αύξηση της τιμής του ρυθμού εξυπηρέτησης του ευέλικτου εξυπηρετητή για το σταθμό 1 από την αρχική τιμή στη $\mu_1 = 0.20$. Όπως και στην προηγούμενη περίπτωση, έτσι κι εδώ, διατηρούμε τις τιμές των υπόλοιπων μεταβλητών σταθερές. Το διάγραμμα με την προαναφερθείσα αλλαγή είναι το παρακάτω:



Εικόνα 4: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου μ_1

Από το διάγραμμα είναι φανερό πως με την αλλαγή αυτή ο ευέλικτος εξυπηρετητής τοποθετείται στο δεύτερο σταθμό πιο αργά από ότι με τις αρχικές τιμές. Αυτό οφείλεται στο ότι το $h_1 \geq h_2$ κι αφού αυξάνεται ο ρυθμός εξυπηρέτησης του ευέλικτου που σχετίζεται με το σταθμό 1, λογικό είναι να μας συμφέρει να παραμένει σε αυτή τη θέση περισσότερη ώρα σε σύγκριση με τις αρχικές τιμές. Με μείωση του αντίστοιχου ρυθμού θα υπήρχε μια ταχύτερη τοποθέτηση του ευέλικτου στο σταθμό 2.

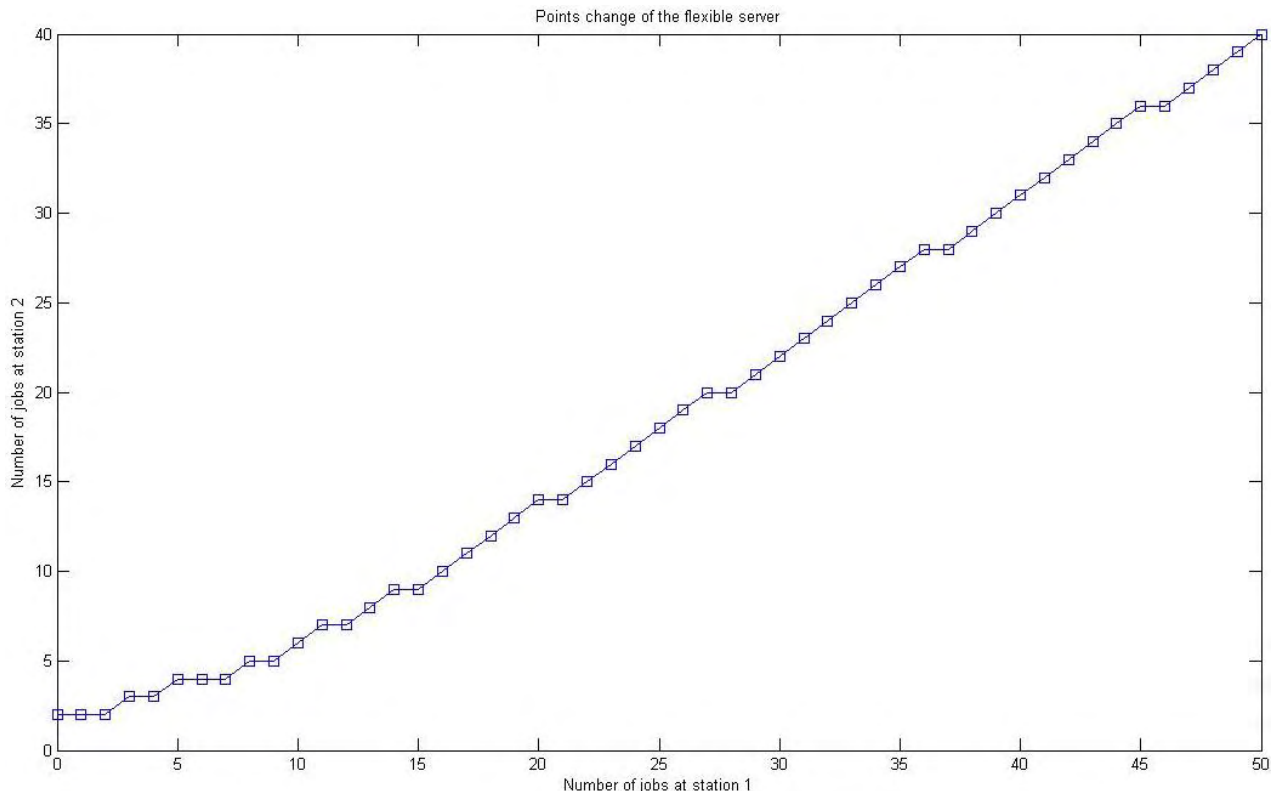
Στην περίπτωση αλλαγής μόνο του ρυθμού εξυπηρέτησης του σταθερού εξυπηρετητή στο σταθμό 2 στην τιμή $\nu_2 = 0.25$ το διάγραμμα είναι το εξής:



Εικόνα 5: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου ν_2

Είναι εύκολο να παρατηρηθεί πως συμβαίνει ό,τι και στην αντίστοιχη αλλαγή για το σταθμό 1. Δηλαδή, όσο αυξάνεται ο ρυθμός εξυπηρέτησης του σταθερού εξυπηρετητή, τόσο η ανάγκη για επιπλέον βοήθεια από τον ευέλικτο παρουσιάζεται αργότερα. Τα σημεία αλλαγής κι εδώ σε σχέση με το αρχικό διάγραμμα είναι πιο μετά και μερικές αλλαγές, στο εύρος των τιμών που δημιουργούμε τα διαγράμματα, δεν πραγματοποιήθηκαν ακόμα.

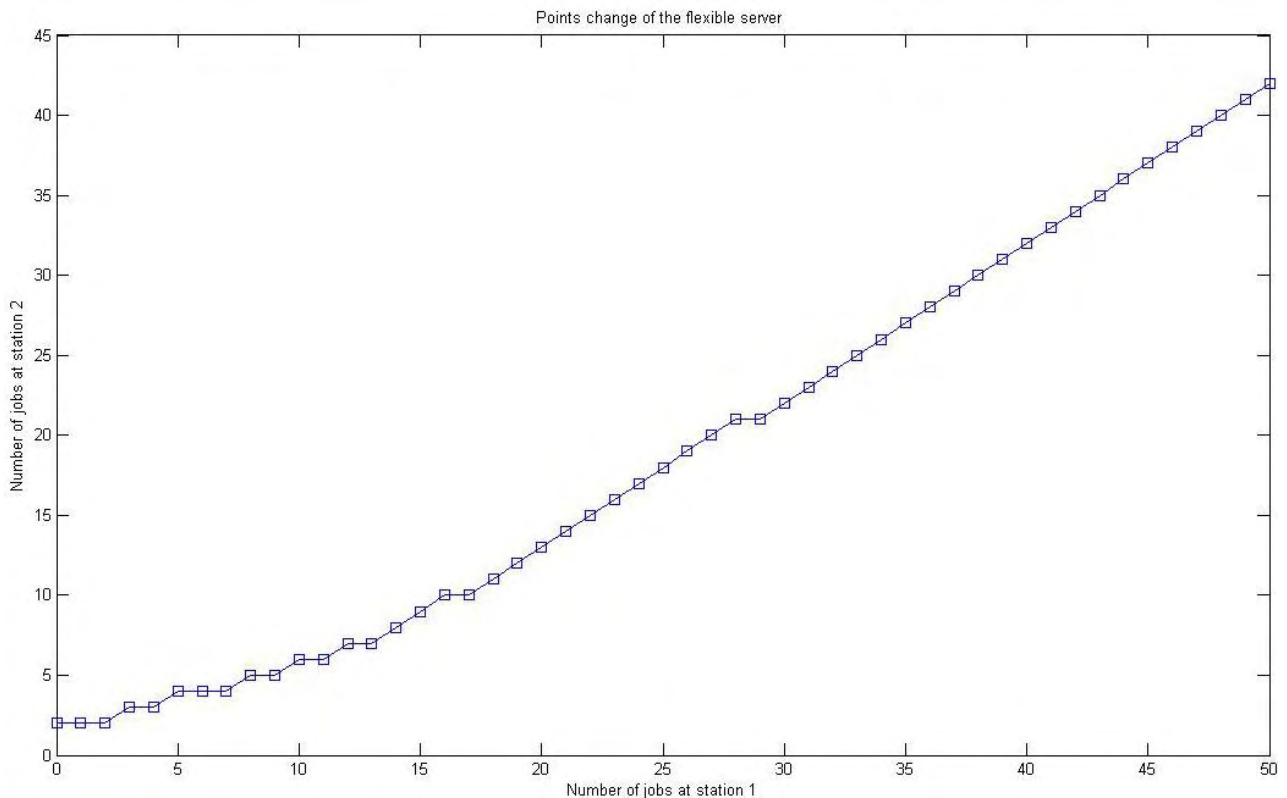
Το επόμενο διάγραμμα αφορά την αύξηση της τιμής του ρυθμού εξυπηρέτησης του ευέλικτου εξυπηρετητή για το σταθμό 2 από την αρχική τιμή σε $\mu_2 = 0.16$ και είναι το εξής:



Εικόνα 6: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου μ_2

Σε αυτή την περίπτωση, αν και η αλλαγή που πραγματοποιήθηκε δεν είναι μεγαλύτερη από την αρχική τιμή φαίνεται ότι η αλλαγή του ευέλικτου πραγματοποιείται νωρίτερα από ότι στην αρχή. Με μία μείωση του αντίστοιχου ρυθμού θα παρατηρούσαμε πως η αλλαγή του ευέλικτου θα γινόταν πιο αργά από ότι στην αρχή, συμπεριφορά όμοια με την περίπτωση για αύξηση του ν_2 .

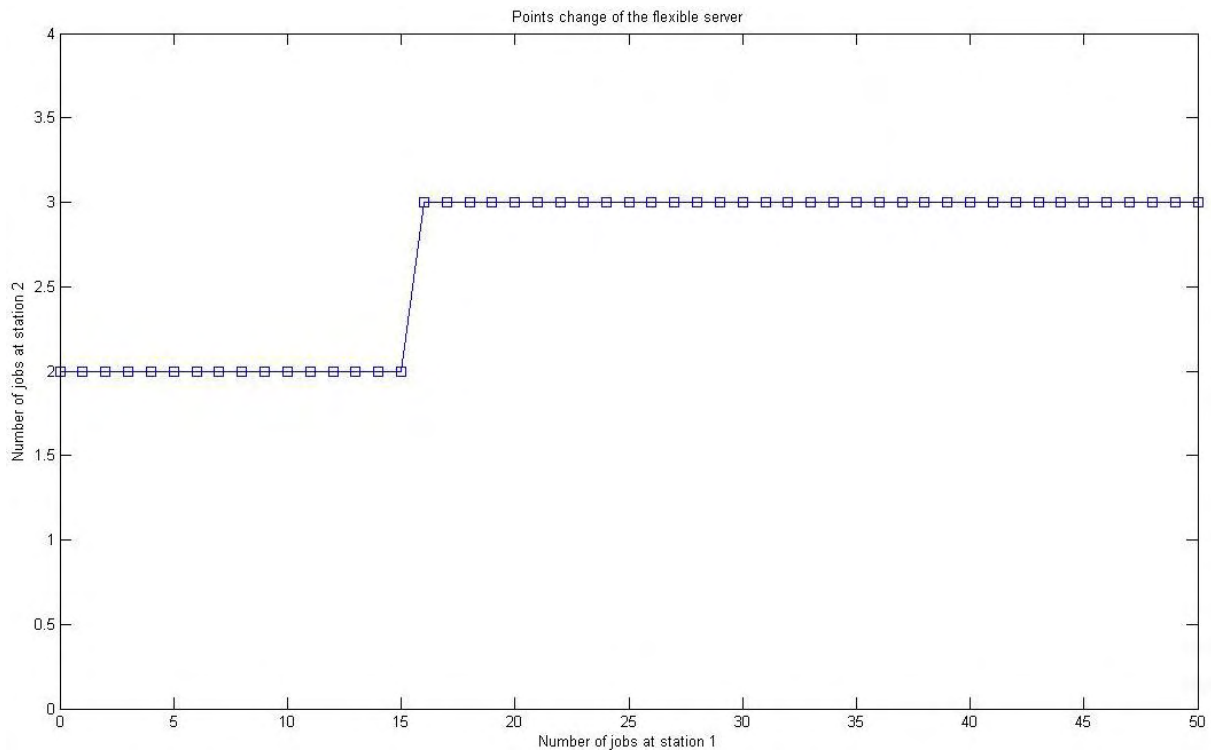
Έχοντας πραγματοποιήσει τον έλεγχο για τις αλλαγές των ρυθμών εξυπηρέτησης θα προχωρήσουμε στην αλλαγή του ρυθμού αφίξεων από την αρχική τιμή στην $\lambda = 0.20$. Το διάγραμμα που προκύπτει από αυτή την αλλαγή είναι το παρακάτω:



Εικόνα 7: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου λ

Εδώ βλέπουμε πως η αλλαγή που πραγματοποιήθηκε στο ρυθμό αφίξεων δεν επηρεάζει ιδιαίτερα την αλλαγή του ευέλικτου σε σχέση με τις αρχικές τιμές. Το μόνο που μπορεί να παρατηρηθεί είναι ότι για $x_1 = 45$ και $x_1 = 46$ στο αρχικό διάγραμμα η αλλαγή γινόταν για ίδιο x_2 ($x_2 = 38$), ενώ τώρα η αλλαγή γίνεται για διαφορετική τιμή. Ακόμα με μείωση που πραγματοποιήσαμε στο ρυθμό αφίξεων στην τιμή $\lambda = 0.10$ δεν παρατηρήθηκε καμία αλλαγή σε σχέση με τις αρχικές μας τιμές.

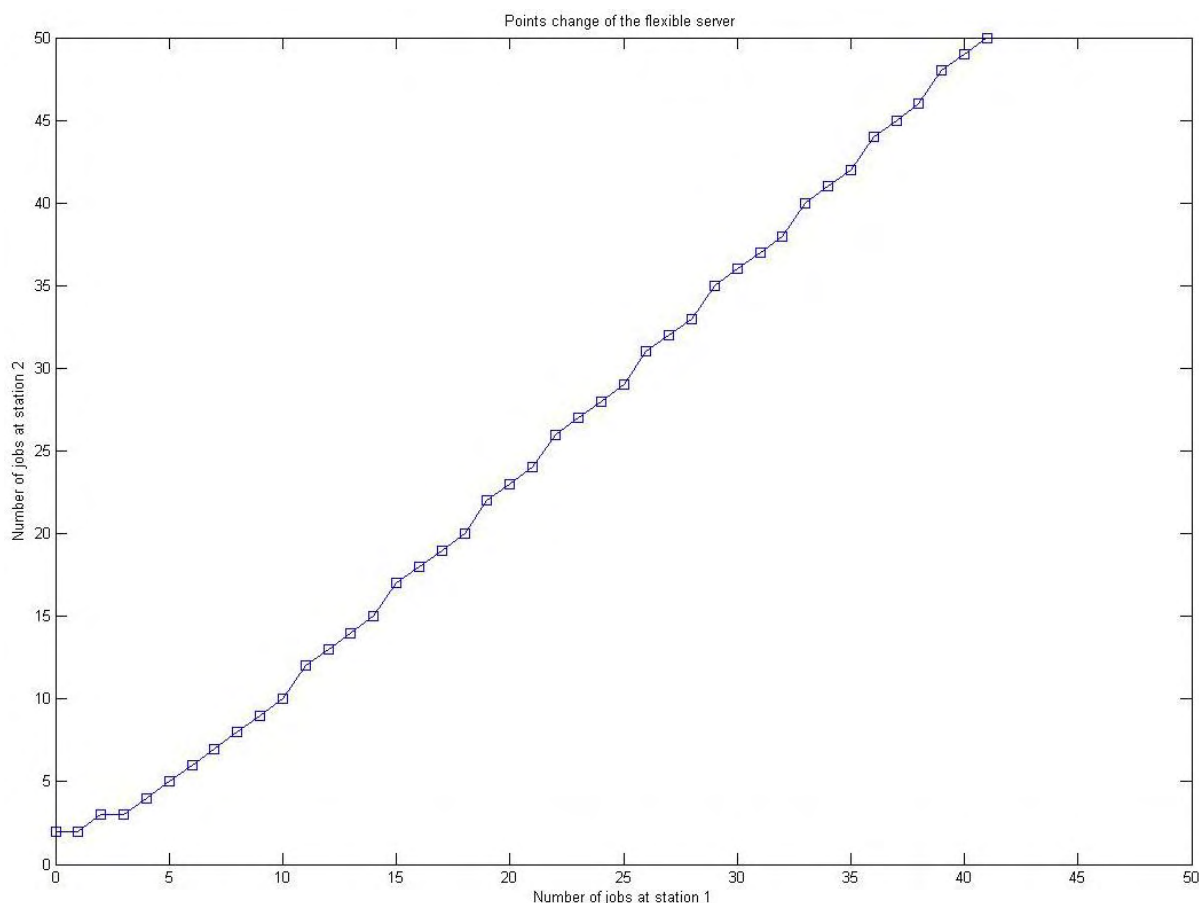
Επίσης τα κόστη έχουν σημαντικό ρόλο στην απόφαση αλλαγής του ευέλικτου εξυπηρετητή και γι' αυτό η επόμενη περίπτωση θα επικεντρωθεί αποκλειστικά στην αλλαγή των τιμών αυτών. Αρχικά θα πραγματοποιήσουμε αύξηση του κόστους παραμονής στο δεύτερο σταθμό στην τιμή $h_2 = 4$ και το διάγραμμα είναι το παρακάτω:



Εικόνα 8: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου h_2

Στο διάγραμμα αυτό η αύξηση του h_2 φαίνεται ξεκάθαρα πως επηρέασε σημαντικά στη μεγάλη πτώση των σημείων αλλαγής. Αυτό γίνεται, διότι αυξήθηκε το κόστος στην ουρά του δεύτερου σταθμού κι έτσι η ανάγκη για πρόσθετο ρυθμό εξυπηρέτησης εμφανίστηκε πιο νωρίς. Επομένως η τοποθέτηση του ευέλικτου εκεί γίνεται συντομότερα σε σχέση με το αρχικό διάγραμμα.

Η τελευταία αλλαγή που θα πραγματοποιήσουμε είναι η αύξηση του κόστους παραμονής στο σταθμό 1 και θα δώσουμε την τιμή $h_1 = 8$. Το διάγραμμα που προκύπτει με αυτή την αλλαγή είναι:



Εικόνα 9: Διάγραμμα σημείων αλλαγής με αύξηση της παραμέτρου h_1

Με την αλλαγή αυτή που πραγματοποιήσαμε είναι εμφανές ότι ο ευέλικτος παραμένει στο σταθμό 1 παραπάνω. Αυτό γίνεται διότι το κόστος στην ουρά εκεί αυξήθηκε και μας συμφέρει να έχουμε εκεί πιο μεγάλο ρυθμό εξυπηρέτησης για να εξυπηρετούνται οι εργασίες συντομότερα. Εάν τώρα είχαμε μειώσει το συγκεκριμένο κόστος τα αποτελέσματα θα ήταν ίδια με την αύξηση του h_2 .

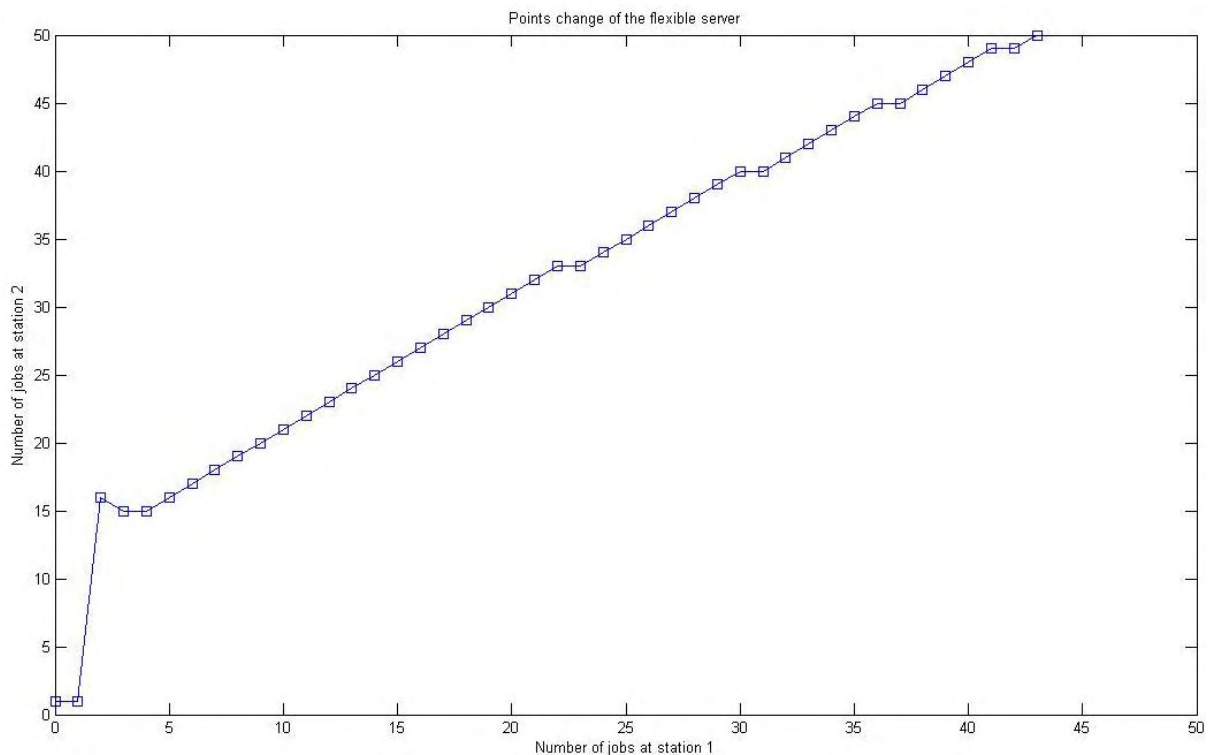
Εν κατακλείδι, από τα παραπάνω, συμπεραίνουμε τα εξής:

- Με την αύξηση του μ_1 ή του h_1 προκύπτει η παραμονή του ευέλικτου για περισσότερο χρόνο στο σταθμό 1.
- Με την αύξηση του μ_2 ή του h_2 η τοποθέτηση του ευέλικτου στο σταθμό 2 γίνεται πιο γρήγορα.

- Με την αύξηση ή την μείωση του ρυθμού αφίξεων λ η τοποθέτηση του ευέλικτου δεν επηρεάζεται σχεδόν καθόλου.
- Εάν υπάρξει αύξηση του ρυθμού εξυπηρέτησης ενός εκ των δύο σταθερών εξυπηρετητών τότε σε εκείνο το σταθμό μειώνεται η ανάγκη για πρόσθετο ρυθμό εξυπηρέτησης κι έτσι ο ευέλικτος ή φεύγει νωρίτερα (σταθμός 1) ή τοποθετείται αργότερα (σταθμός 2).

3.2.2 Περίπτωση 2: $\nu_1 \geq \mu_1$ και $\nu_2 < \mu_2$

Στην συγκεκριμένη περίπτωση η ύπαρξη μοναδικότητας του σημείου αλλαγής, μέσα στο πλαίσιο των επαναλήψεων που πραγματοποιήσαμε και με επιφύλαξη λόγω των αριθμητικών προσεγγίσεων, φαίνεται να ισχύει. Όμως εδώ προέκυψε ένα ποσοστό 0.4-0.6 %, όπου δεν ισχύει η μονοτονία των σημείων αλλαγής. Μία τέτοια περίπτωση είναι για τις παρακάτω τιμές των παραμέτρων: $\nu_1 = 0.03$, $\nu_2 = 0.07$, $\mu_1 = 0.02$, $\mu_2 = 0.53$, $\lambda = 0.025$, $h_1 = 3.5$ και $h_2 = 0.15$ όπου το διάγραμμα των σημείων αλλαγής έχει την εξής μορφή:



Εικόνα 10: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία

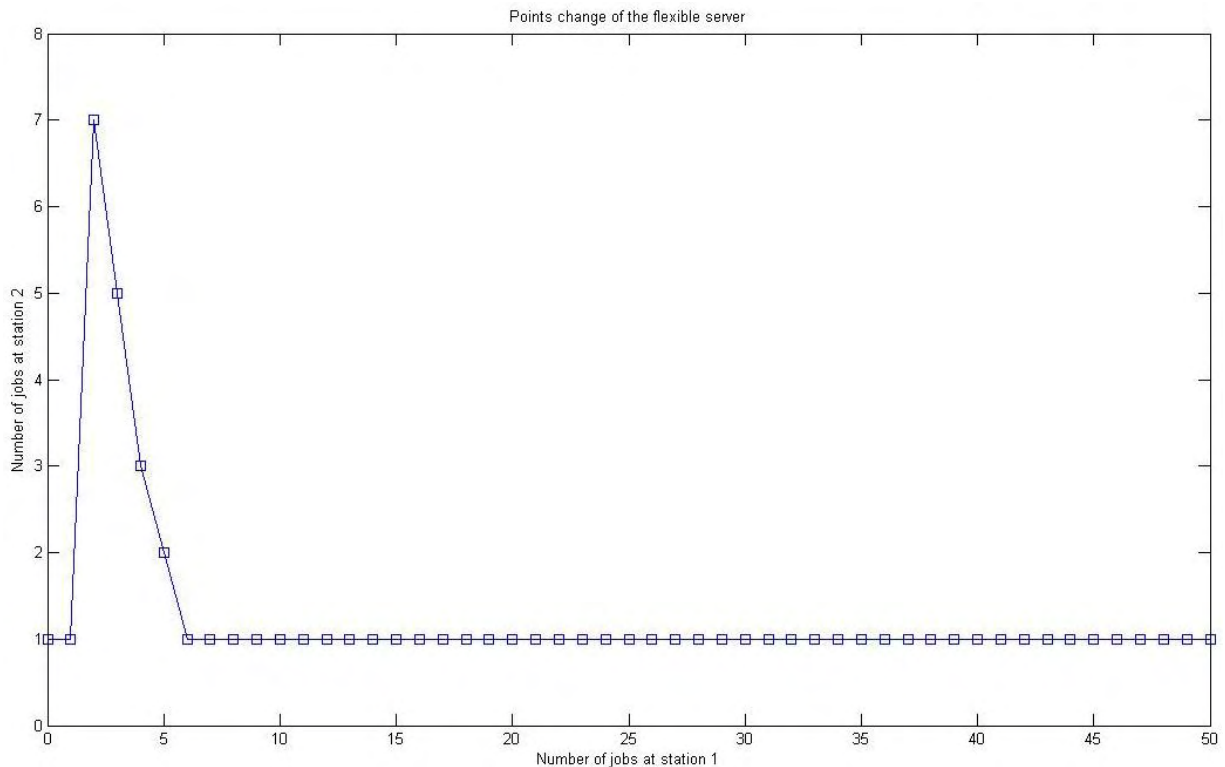
Στο παραπάνω διάγραμμα παρατηρούμε ότι πραγματοποιείται αλλαγή για $x_1 = 3$ στο $x_2 = 16$ και ενώ αναμένουμε οι επόμενες αλλαγές, καθώς αυξάνεται το x_1 , να πραγματοποιηθούν στην ίδια τιμή ή σε μεγαλύτερη, αυτή η αλλαγή πραγματοποιείται σε χαμηλότερη τιμή $x_2 = 15$, όταν έχουμε $x_1 = 4$. Έτσι παρόλο το μικρό ποσοστό εμφάνισης τέτοιων περιπτώσεων, η μονοτονία εδώ δεν ισχύει.

Αξίζει να σημειωθεί πως στις περιπτώσεις αυτές παρατηρήθηκε ότι η τιμή του μ_2 ήταν αρκετά μεγαλύτερη από τους άλλους τρεις ρυθμούς εξυπηρέτησης και επίσης τα κόστη στις δύο ουρές είχαν μεγάλη απόκλιση το ένα με το άλλο. Για παράδειγμα θα δώσουμε έναν πίνακα με τιμές που προήλθαν από πεντακόσιες επαναλήψεις και φαίνεται ξεκάθαρα η παραπάνω παρατήρηση.

v_1	μ_1	v_2	μ_2	λ	h_1	h_2
0.15	0.13	0.06	0.89	0.126	4.8	0.85
0.03	0.02	0.07	0.53	0.025	3.5	0.15
0.15	0.10	0.01	0.70	0.11	4.5	0.80
0.17	0.13	0.12	0.94	0.14	5	1

Πίνακας 3: Τιμές παραμέτρων για τις οποίες δεν ισχύει η μονοτονία των σημείων αλλαγής

Ιδιαίτερο ενδιαφέρον παρουσιάζει ένα παράδειγμα των περιπτώσεων αυτών, το οποίο θα παρουσιαστεί παρακάτω, για το οποίο, ενώ θα περιμέναμε να ισχύει κάτι αντίστοιχο με το θεώρημα 1, δηλαδή για $v_2 = 0$ και $\mu_1(h_1 - h_2) < \mu_2 h_2$ να είναι βέλτιστο να τοποθετείται ο ευέλικτος εξυπηρετητής στο σταθμό 2, δεν συμβαίνει αυτό. Οι τιμές για τις οποίες παρατηρείται αυτό είναι: $v_2 = 0$, $\mu_2 = 0.85$, $v_1 = 0.10$, $\mu_1 = 0.05$, $\lambda = 0.07$, $h_1 = 17$ και $h_2 = 1$. Το διάγραμμα γι' αυτές τις τιμές είναι το παρακάτω:



Εικόνα 11: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία

Αυτό που παρατηρούμε στο παραπάνω διάγραμμα είναι ότι ο ευέλικτος μπορεί να τοποθετείται στο σταθμό 1 ενώ όπως αναφέραμε και πριν περιμέναμε να τοποθετείται στο δεύτερο σταθμό.

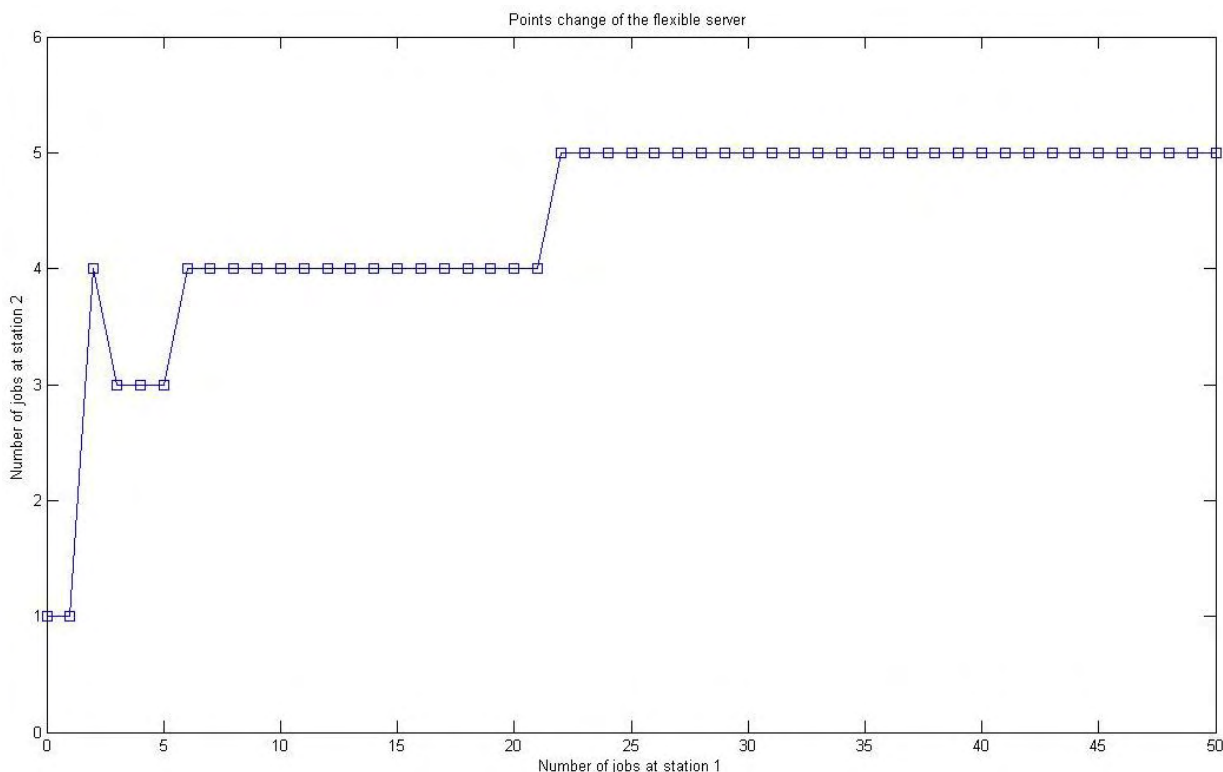
3.2.3 Περίπτωση 3: $\nu_1 < \mu_1$ και $\nu_2 \geq \mu_2$

Για την περίπτωση αυτή μπορούμε να πούμε, σύμφωνα πάντα με τα παραδείγματα που πραγματοποιήθηκαν, ότι για το εύρος τιμών που επιλέξαμε ισχύει η ύπαρξη μοναδικού σημείου αλλαγής του ευέλικτου από τον πρώτο σταθμό στο δεύτερο όπως επίσης και η μονοτονία των σημείων αυτών.

3.2.4 Περίπτωση 4: $\nu_1 < \mu_1$ και $\nu_2 < \mu_2$

Στην περίπτωση αυτή, σύμφωνα με τα παραδείγματα, παρατηρήθηκε πως ισχύει η ύπαρξη μοναδικού σημείου αλλαγής του ευέλικτου από τον πρώτο σταθμό στο δεύτερο, ενώ σε ποσοστό 0.1% παρατηρήθηκε πως δεν ισχύει η μη φθίνουσα μονοτονία των σημείων αυτών. Οι τιμές των παραμέτρων για τις οποίες δεν ισχύει η

μονοτονία είναι: $\nu_1 = 0.10$, $\nu_2 = 0.07$, $\mu_1 = 0.12$, $\mu_2 = 0.90$, $\lambda = 0.1012$, $h_1 = 4.8$ και $h_2 = 0.68$. Στις παραπάνω τιμές αντιστοιχεί το ακόλουθο διάγραμμα:



Εικόνα 12: Διάγραμμα σημείων αλλαγής που δεν ισχύει η μονοτονία

Όπως φαίνεται από το διάγραμμα παρατηρείται μία αλλαγή αργότερα από ότι θα περιμέναμε να πραγματοποιηθεί (για $x_1 = 3$ στο $x_2 = 4$) με αποτέλεσμα να ανατρέπεται η υπόθεση για μη φθίνουσα μονοτονία των σημείων αλλαγής.

3.2.5 Γενικά συμπεράσματα για $h_1 \geq h_2$

Τελικά τα συμπεράσματα μας, από όλα τα παραπάνω, για τις περιπτώσεις που ισχύει $h_1 \geq h_2$ είναι τα εξής:

- Για όλες τις περιπτώσεις ισχύει η μοναδικότητα του σημείου αλλαγής του ευέλικτου εξυπηρετητή.
- Στις περιπτώσεις 1 και 3 η ακολουθία των σημείων αλλαγής είναι μη φθίνουσα ενώ στις περιπτώσεις 2 και 4 δεν ισχύει πάντα αυτό.

- Μεταξύ των δύο περιπτώσεων που δεν ισχύει η μονοτονία που υποθέσαμε, υπήρχε διαφορά στην συχνότητα που εμφανίζεται μία περίπτωση που δεν ισχύει η μονοτονία των σημείων αλλαγής.

3.3 Αποτελέσματα για $h_1 < h_2$

Σε αυτή την ενότητα θα γίνει παρουσίαση των αποτελεσμάτων της κάθε περίπτωσης για την οποία ισχύει $h_1 < h_2$. Θα παρουσιαστούν τα αποτελέσματα που αφορούν την μοναδικότητα του σημείου αλλαγής σε κάθε περίπτωση, όπως επίσης και τα αποτελέσματα για τον έλεγχο της μονοτονίας των σημείων αυτών. Ακόμα θα παρουσιαστούν τα αποτελέσματα για την μοναδικότητα του σημείου τερματισμού λειτουργίας του σταθερού στο σταθμό 1, όπως επίσης και τα αποτελέσματα για την μονοτονία των σημείων αυτών. Επιπλέον, θα παρουσιάσουμε ένα αριθμητικό παράδειγμα για να κατανοήσουμε τον τρόπο που επηρεάζει κάθε παράμετρος του προβλήματος την λήψη των αποφάσεων μας.

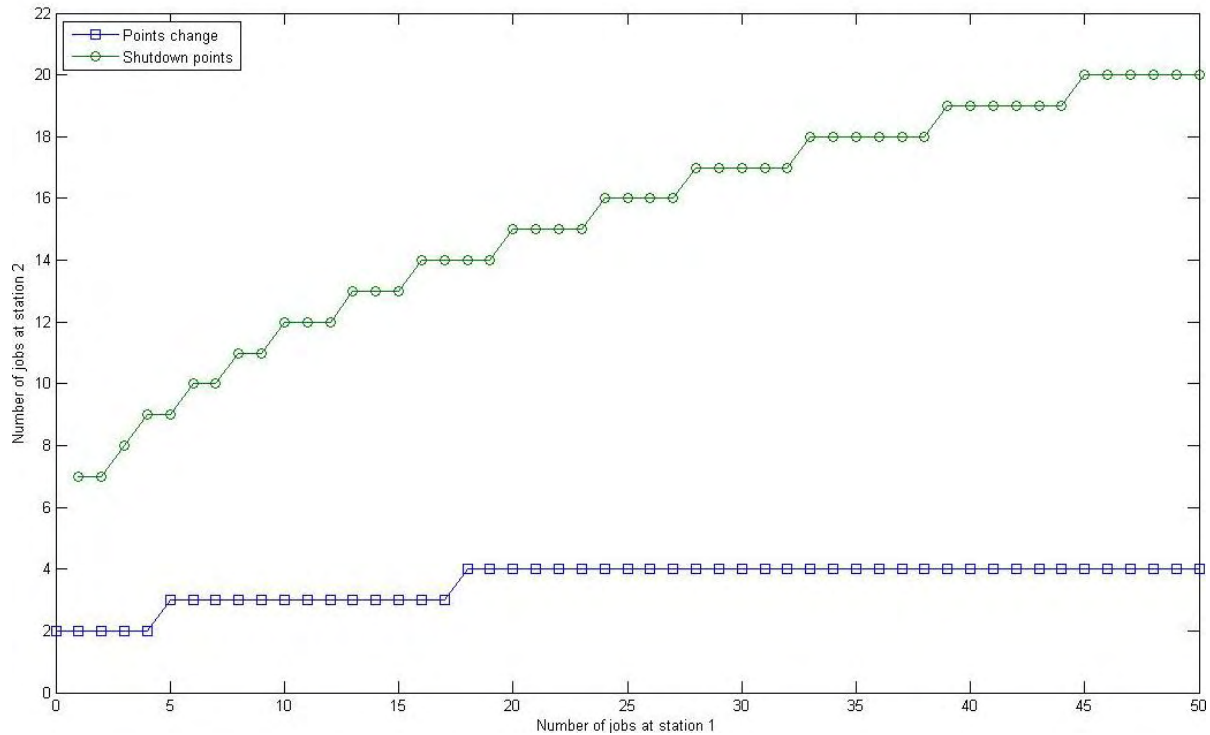
3.3.1 Περίπτωση 1: $\nu_1 \geq \mu_1$ και $\nu_2 \geq \mu_2$

Για την περίπτωση αυτή, ύστερα από τον καθορισμένο αριθμό επαναλήψεων, μπορούμε με βάση τα αποτελέσματα που εξήγαμε να πούμε ότι ισχύει η ύπαρξη μοναδικότητας του σημείου αλλαγής, όπως επίσης και η μονοτονία των σημείων αυτών. Ακόμα, η υπόθεση μας για ύπαρξη μοναδικού σημείου τερματισμού του σταθερού στο σταθμό 1 βγαίνει αληθής όπως και η μη φθίνουσα μονοτονία των σημείων αυτών.

Για να διασαφηνιστεί καλύτερα ο ρόλος των παραμέτρων στη λήψη της απόφασης για τον ευέλικτο εξυπηρετητή αλλά και για να αναλύσουμε το ρόλο αυτών στην απόφαση τερματισμού της λειτουργίας του σταθερού εξυπηρετητή στο σταθμό 1, θα δώσουμε ένα παράδειγμα με ανάλυση διαγραμμάτων και για τις δύο παραπάνω περιπτώσεις.

Στο παράδειγμα που ακολουθεί οι αρχικές τιμές είναι οι εξής: $\nu_1 = 0.95$, $\nu_2 = 0.80$, $\mu_1 = 0.46$, $\mu_2 = 0.11$, $\lambda = 0.45$, $h_1 = 2.46$ και $h_2 = 2.68$. Στο διάγραμμα που ακολουθεί παρουσιάζονται ταυτόχρονα τα σημεία αλλαγής του ευέλικτου

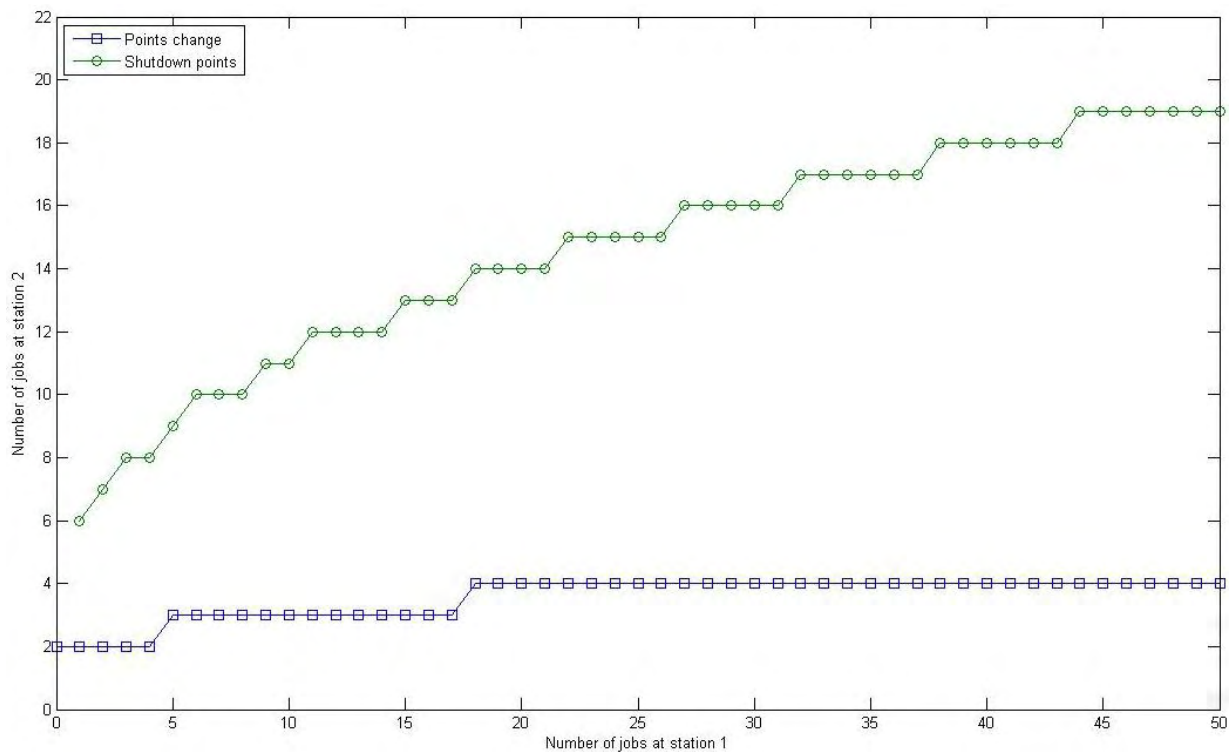
εξυπηρετητή (*Points change*) με τα σημεία τερματισμού λειτουργίας του σταθερού στο σταθμό 1 (*Shutdown points*).



Εικόνα 13: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αρχικές τιμές παραμέτρων

Στο παραπάνω διάγραμμα παρατηρείται εύκολα ότι ο ευέλικτος τοποθετείται σχετικά νωρίς στο σταθμό 2. Με βάση το παραπάνω διάγραμμα θα παρατηρηθεί παρακάτω το πώς οι αλλαγές των παραμέτρων επηρεάζουν την απόφαση τερματισμού και αλλαγής του ευέλικτου.

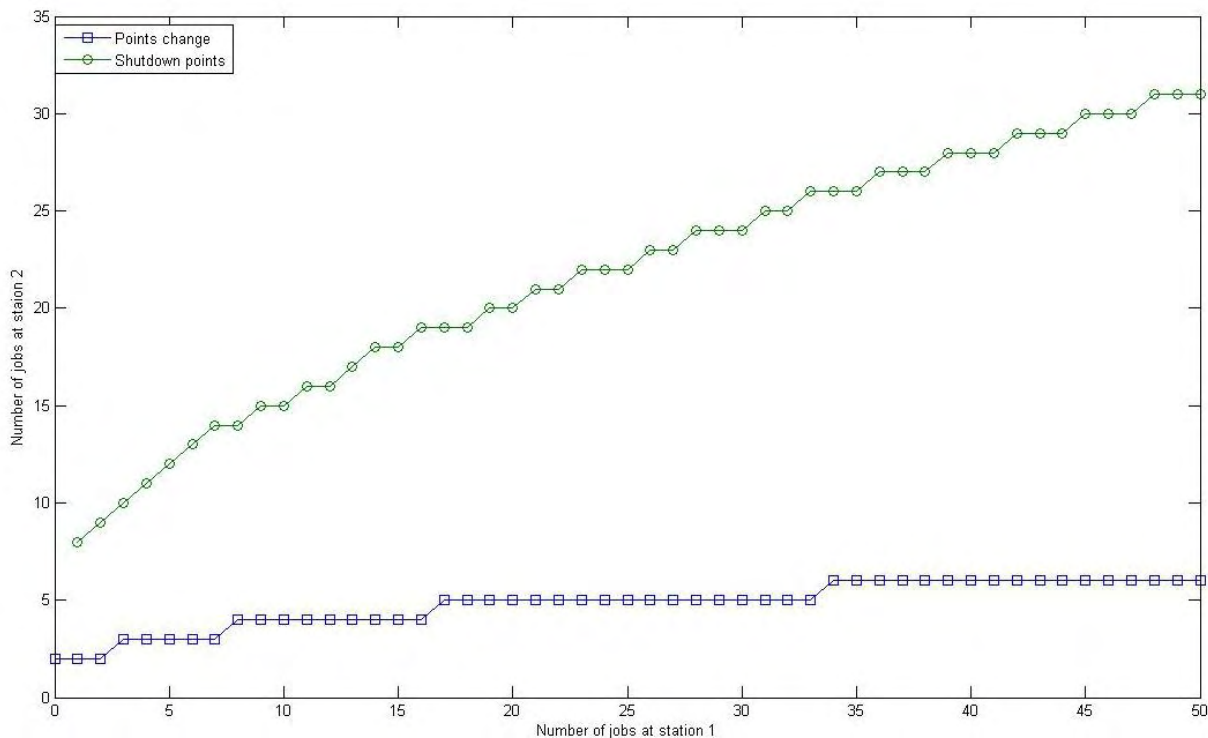
Αρχικά θα ξεκινήσουμε με αλλαγή της τιμής του μ_1 και θα την αυξήσουμε ώστε να πάρει την τιμή $\mu_1 = 0.56$ ενώ οι υπόλοιπες παράμετροι παραμένουν σταθερές. Το παραπάνω διάγραμμα διαμορφώνεται ως εξής:



Εικόνα 14: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου μ_1

Σε αυτό το διάγραμμα ενώ θα περιμέναμε τα σημεία αλλαγής να είναι πιο μετά, σε σχέση με το αρχικό μας, παρατηρούμε πως δεν πραγματοποιείται αλλαγή και αυτό οφείλεται και στο ότι η τιμή του ν_1 είναι αρκετά μεγάλη, με αποτέλεσμα να μην έχει μεγάλη ανάγκη για περαιτέρω βοήθεια. Στο διάγραμμα αυτό παρατηρούμε επίσης ότι ο τερματισμός του σταθερού εξυπηρετητή στο σταθμό 1 γίνεται συντομότερα σε σχέση με το αρχικό μας διάγραμμα.

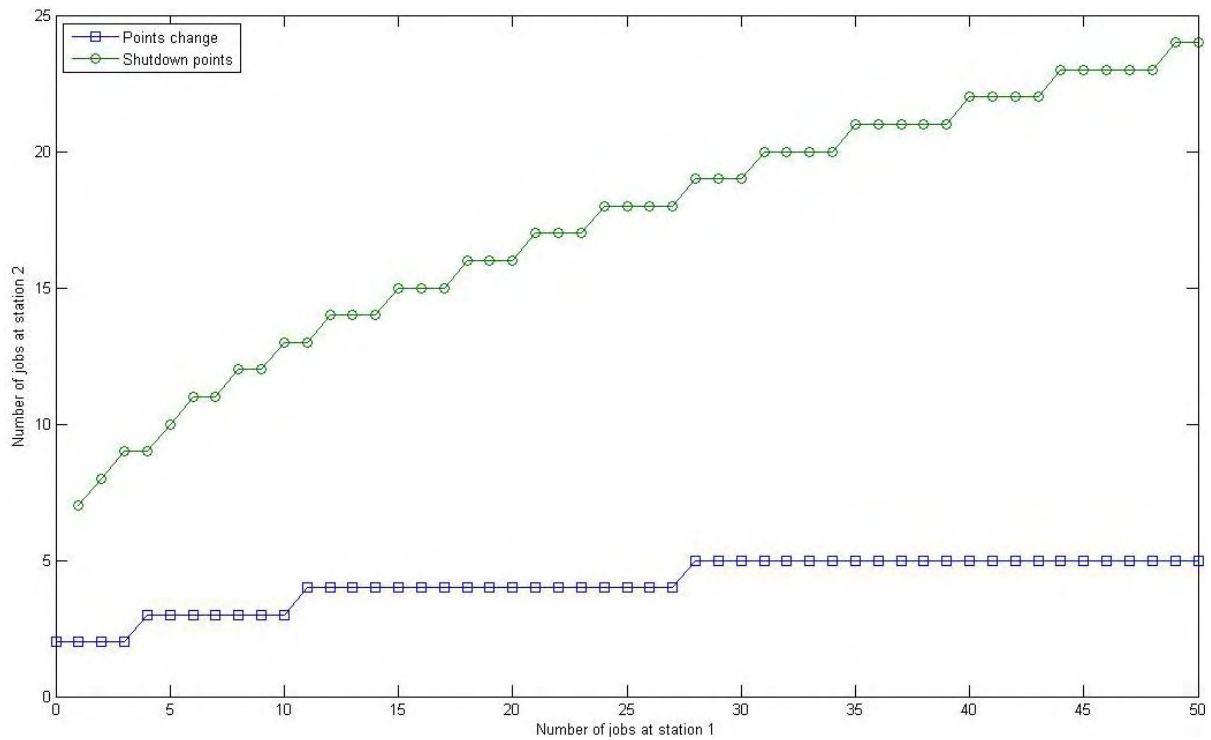
Στην συνέχεια θα πραγματοποιήσουμε αλλαγή στο ν_1 και επειδή η τιμή του είναι ήδη αρκετά μεγάλη θα προχωρήσουμε σε μείωση της τιμής του σε $\nu_1 = 0.80$. Το διάγραμμα που προκύπτει είναι το παρακάτω:



Εικόνα 15: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου V_1

Με την μείωση του v_1 παρατηρούμε πως η αλλαγή του ευέλικτου γίνεται αργότερα από ότι με τις αρχικές μας τιμές, κάτι το οποίο είναι λογικό αφού η ανάγκη για βοήθεια από τον ευέλικτο είναι τώρα πιο αναγκαία από την αρχή. Επίσης παρατηρούμε πως ο τερματισμός του σταθερού εξυπηρετητή στο σταθμό 1 γίνεται αργότερα σε σχέση με το αρχικό μας διάγραμμα. Αυτό μπορεί να οφείλεται στο ότι μειώθηκε ο ρυθμός που εισέρχονται εργασίες στο σταθμό 2 με αποτέλεσμα να μην μας συμφέρει να τερματίζουμε την λειτουργία του πρώτου σταθμού τόσο νωρίς όσο με την αρχική τιμή του v_1 . Με αύξηση της τιμής του v_1 θα προέκυπτε ότι θα τερματιζόταν πιο νωρίς, σε σχέση με τις αρχικές τιμές, η λειτουργία του σταθμού 1 και η αλλαγή του ευέλικτου θα πραγματοποιούνταν συντομότερα.

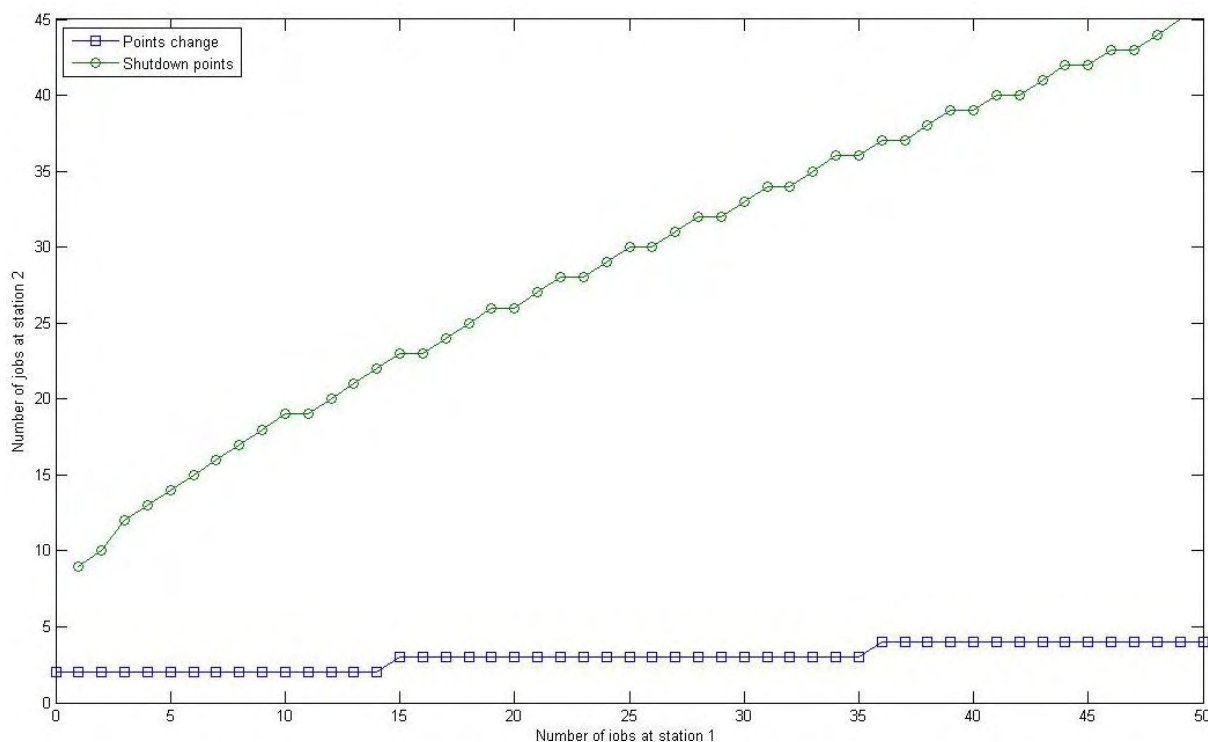
Στην συνέχεια θα ασχοληθούμε με τις παραμέτρους που αφορούν το σταθμό 2 για να παρατηρήσουμε την επίδρασή τους και στις δύο αποφάσεις. Θα ξεκινήσουμε με την αύξηση του v_2 στην τιμή $v_2 = 0.88$. Το διάγραμμα μας διαμορφώνεται ως εξής:



Εικόνα 16: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου V_2

Σε αυτή την περίπτωση παρατηρούμε ακριβώς ό,τι και για την περίπτωση με την μείωση του v_1 . Δηλαδή, ο ευέλικτος τοποθετείται αργότερα από ότι στην αρχή στο σταθμό 2 και αυτό γίνεται διότι με την αύξηση του v_2 η ανάγκη για βοήθεια εκεί παρουσιάζεται αργότερα από ότι με τις αρχικές τιμές. Επιπλέον παρατηρούμε πως ο τερματισμός του σταθερού στο σταθμό 1 γίνεται πιο αργά από ότι στο αρχικό μας διάγραμμα. Είναι λογικό, αφού ο σταθμός 2 εξυπηρετεί πιο γρήγορα, από ότι στην αρχή, ο τερματισμός λειτουργίας του σταθμού 1 να πραγματοποιείται αργότερα.

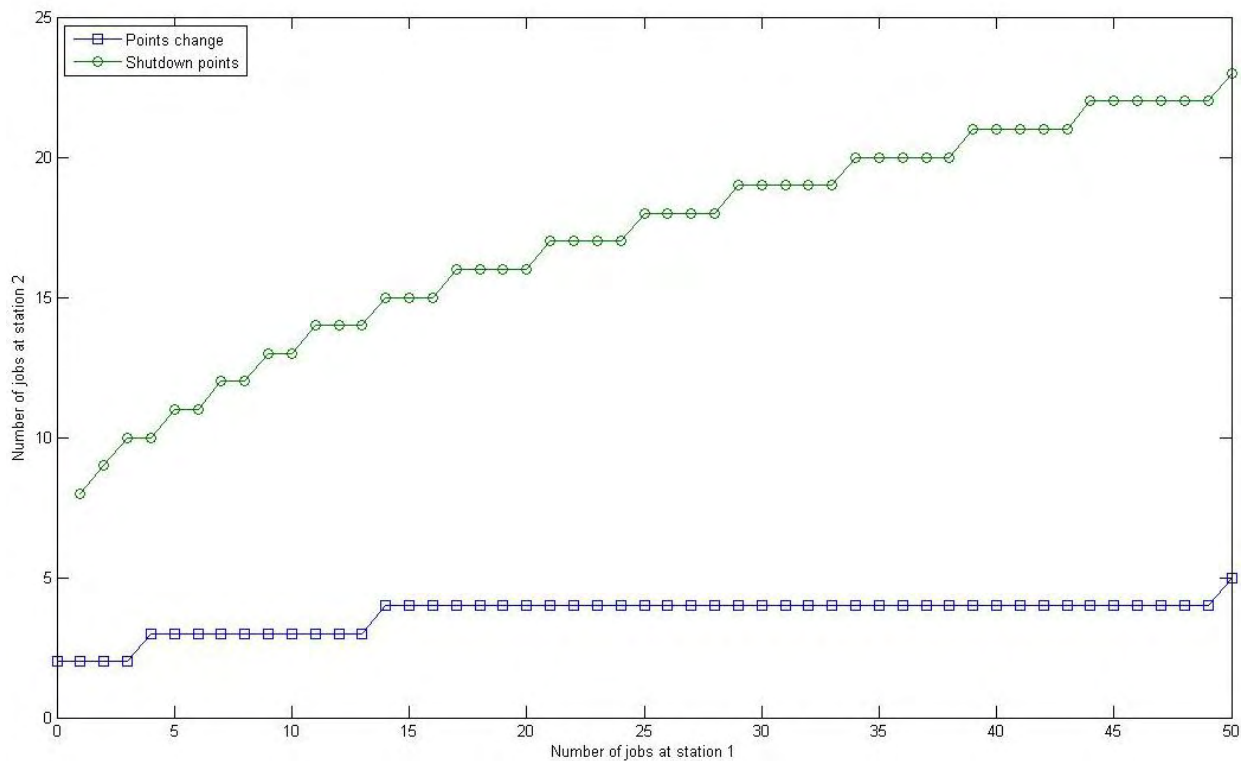
Παραμένοντας στο σταθμό 2 θα πραγματοποιήσουμε αύξηση του μ_2 (από $\mu_2 = 0.11$ σε $\mu_2 = 0.45$) ενώ οι υπόλοιπες παράμετροι διατηρούνται πάντα σταθερές. Το διάγραμμα μας διαμορφώνεται ως εξής:



Εικόνα 17: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου μ_2

Εδώ παρατηρούμε πως η αύξηση του συγκεκριμένου ρυθμού εξυπηρέτησης έχει ως αποτέλεσμα ο σταθερός εξυπηρετητής του σταθμού 1 να σταματάει την λειτουργία του αργότερα, όπως φαίνεται και από το αντίστοιχο διάγραμμα. Αυτό οφείλεται στο ότι οι εργασίες του σταθμού 2 εξυπηρετούνται πιο γρήγορα κι έτσι δεν είναι αναγκαία η διακοπή του σταθερού εξυπηρετητή στον πρώτο σταθμό τόσο νωρίς, όσο στην αρχή, αφού οι εργασίες που αποστέλλει εξυπηρετούνται πιο νωρίς. Επιπρόσθετα, όσο αφορά την τοποθέτηση του ευέλικτου στο σταθμό 2 πραγματοποιείται νωρίτερα, κάτι το οποίο οφείλεται στο μεγαλύτερο κόστος αναμονής στην ουρά του σταθμού 2 (περιπτώσεις $h_1 < h_2$) και είναι λογικό να θέλουμε να εξυπηρετούνται πιο γρήγορα οι εργασίες εκεί. Στο παράδειγμά μας επειδή τα κόστη αναμονής είναι αρκετά κοντά γι' αυτό δεν βλέπουμε μεγάλες αλλαγές, όσο αφορά την τοποθέτηση του ευέλικτου, αλλά και πάλι εύκολα μπορούμε να παρατηρήσουμε την επιρροή της κάθε αλλαγής των παραμέτρων από το ότι, όσο αυξάνεται το x_1 , η αλλαγή πραγματοποιείται για το ίδιο x_2 .

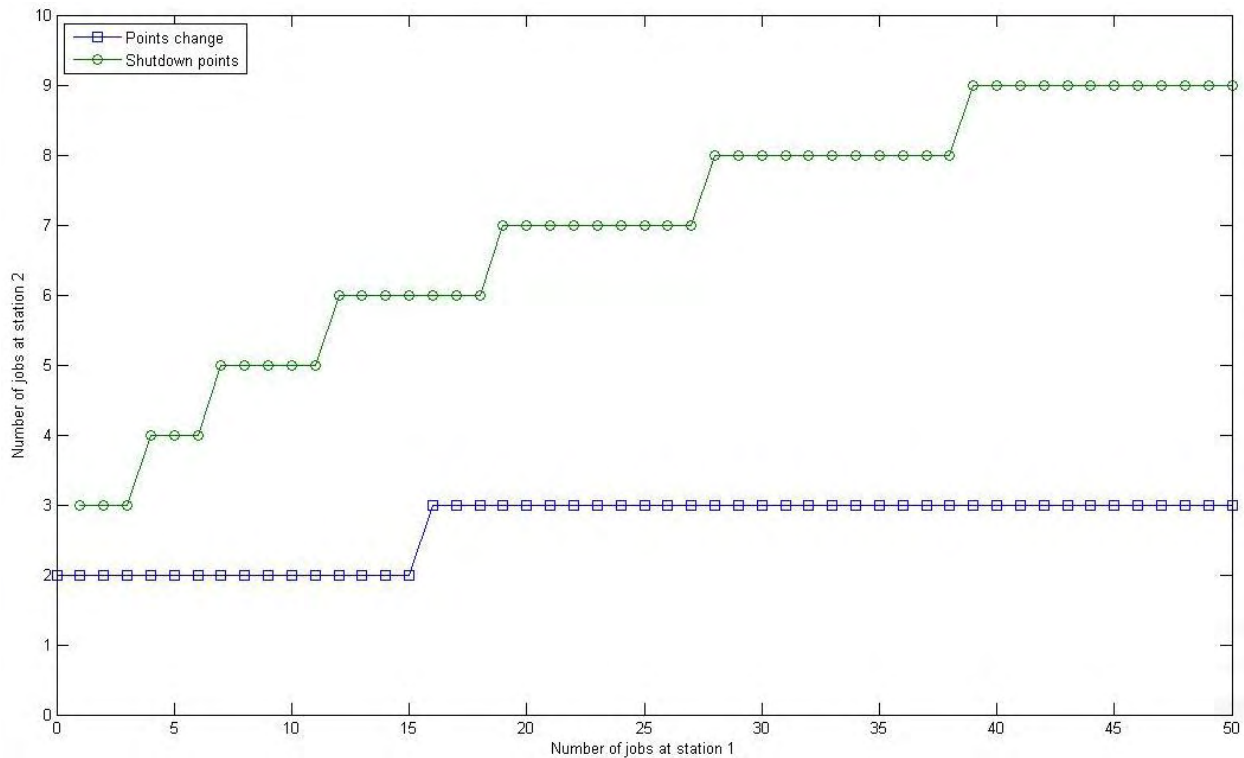
Έχοντας πραγματοποιήσει τον έλεγχο για τις αλλαγές των ρυθμών εξυπηρέτησης θα προχωρήσουμε στην αλλαγή του ρυθμού αφίξεων από την αρχική τιμή στην $\lambda = 0.55$. Το διάγραμμα που προκύπτει από αυτή την αλλαγή είναι το παρακάτω:



Εικόνα 18: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με αύξηση της παραμέτρου λ

Με την αύξηση του λ παρατηρούμε την αλλαγή του ευέλικτου να γίνεται ελάχιστα πιο μετά σε σχέση με το αρχικό διάγραμμα. Επίσης ο τερματισμός του σταθερού στο σταθμό 1 γίνεται αργότερα από ότι με τις αρχικές τιμές. Αυτό είναι λογικό, διότι εισέρχονται περισσότερες εργασίες από ότι στην αρχή.

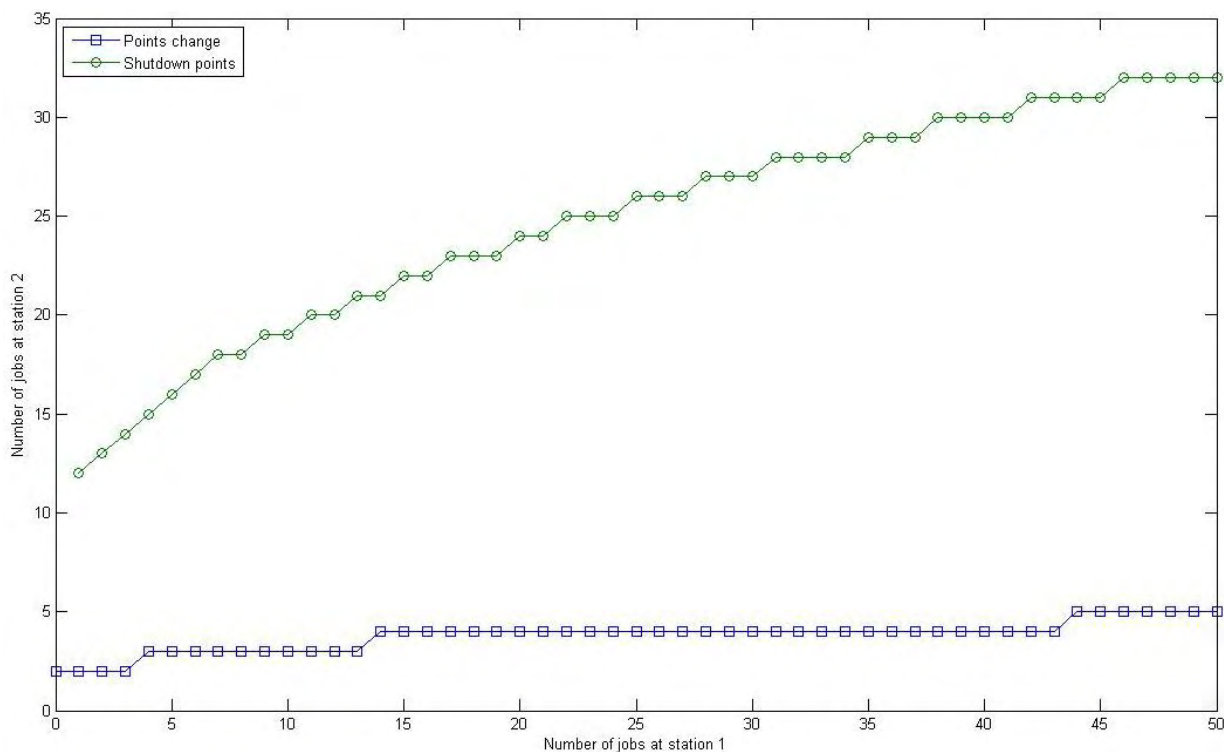
Τέλος θα ασχοληθούμε με το πώς επηρεάζουν τα κόστη των σταθμών στη λήψη των δύο αποφάσεων που μας απασχολούν. Ξεκινώντας θα αλλάξουμε το κόστος παραμονής στην ουρά του σταθμού 1 και θα το μειώσουμε στην τιμή $h_1 = 1.4$. Το διάγραμμα μας είναι το εξής:



Εικόνα 19: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου h_1

Στην περίπτωση αυτή βλέπουμε πως η μετατόπιση του ευέλκτου στο σταθμό 2 γίνεται πιο πριν απ' ό τι στο αρχικό. Αυτό οφείλεται στο ότι μειώσαμε το κόστος στην ουρά του πρώτου σταθμού και είναι λογικό ο ευέλκτος να τοποθετηθεί πιο νωρίς για να ενισχύσει τον σταθερό εξυπηρετητή στο σταθμό 2. Εάν είχαμε αυξήσει το συγκεκριμένο κόστος οι αλλαγές θα παρουσιάζονταν πιο αργά. Επιπλέον βλέπουμε τον τερματισμό του σταθερού στο σταθμό 1 πραγματοποιείται νωρίτερα.

Στο τέλος θα ασχοληθούμε με το κόστος παραμονής στην ουρά του σταθμού 2 (h_2), όπου, εάν το αυξήσουμε η επιρροή που έχει στη λήψη των αποφάσεων που μας ενδιαφέρουν είναι όμοια με την μείωση του h_1 . Έτσι θα το μειώσουμε στην τιμή $h_2 = 2.54$ και το διάγραμμα που προκύπτει είναι το παρακάτω:



Εικόνα 20: Διάγραμμα σημείων αλλαγής και σημείων αδρανοποίησης του σταθερού εξυπηρετητή με μείωση της παραμέτρου h_2

Από αυτό το διάγραμμα παρατηρούμε τα σημεία αλλαγής του ευέλικτου να εμφανίζονται λίγο αργότερα. Αυτό γίνεται διότι μειώσαμε το κόστος παραμονής στην ουρά του δεύτερου σταθμού και γι' αυτό ο ευέλικτος παραμένει στο σταθμό 1 περισσότερο από ότι στο αρχικό διάγραμμα. Επίσης, καταλαβαίνουμε πως ο τερματισμός λειτουργίας του σταθερού εξυπηρετητή πραγματοποιείται αργότερα απ' ότι με τις αρχικές τιμές, διότι τώρα μειώθηκε το κόστος αναμονής στην δεύτερη ουρά, με αποτέλεσμα να μπορεί να στέλνει περισσότερες εργασίες ο σταθμός 1 προς το δεύτερο σταθμό.

Εν τέλει, από τα παραπάνω συμπεραίνουμε τα εξής:

- Με την μείωση του h_1 η τοποθέτηση του ευέλικτου στο σταθμό 2 πραγματοποιείται νωρίτερα, όπως συμβαίνει και για τον τερματισμό του σταθερού στο σταθμό 1.
- Στην περίπτωση της μείωσης του h_2 η τοποθέτηση του ευέλικτου στο δεύτερο σταθμό πραγματοποιείται αργότερα, όπως επίσης αργότερα πραγματοποιείται και ο τερματισμός του σταθερού στο σταθμό 1.

- Με την αύξηση του λ παρατηρούμε την αλλαγή του ευέλικτου να γίνεται αργότερα σε σχέση με το αρχικό διάγραμμα. Επίσης ο τερματισμός του σταθερού στο σταθμό 1 γίνεται αργότερα από ότι με τις αρχικές τιμές.
- Στην περίπτωση αύξησης του μ_1 ο ευέλικτος παραμένει στο σταθμό 1 περισσότερο, ενώ ο τερματισμός του σταθερού 1 πραγματοποιείται νωρίτερα.
- Με την αύξηση του μ_2 ο ευέλικτος εξυπηρετητής τοποθετείται νωρίτερα στο σταθμό 2, ενώ ο τερματισμός του σταθερού στο σταθμό 1 γίνεται αργότερα.
- Εάν υπάρξει αύξηση του v_2 ή μείωση του v_1 ο ευέλικτος τοποθετείται αργότερα στο σταθμό 2, ενώ ο τερματισμός του σταθερού στο σταθμό 1 πραγματοποιείται και αυτός αργότερα.

3.3.2 Περίπτωση 2: $v_1 \geq \mu_1$ και $v_2 < \mu_2$

Στην περίπτωση 2, σύμφωνα με τις επαναλήψεις που πραγματοποιήσαμε, μπορούμε να πούμε ότι η μοναδικότητα του σημείου αλλαγής όπως και η μονοτονία των σημείων αυτών ισχύουν. Για το σημείο τερματισμού λειτουργίας του σταθερού εξυπηρετητή του σταθμού 1 φαίνεται κι εδώ να ισχύει η μοναδικότητά του όπως επίσης και η υπόθεση μας για μη φθίνουσα μονοτονία των σημείων αυτών.

3.3.3 Περίπτωση 3-4: ($v_1 < \mu_1$ και $v_2 \geq \mu_2$)-($v_1 < \mu_1$ και $v_2 < \mu_2$)

Τα αποτελέσματα των δύο περιπτώσεων αυτών τα παρουσιάζουμε σε μία κοινή παράγραφο διότι είναι τα ίδια. Στις περιπτώσεις αυτές, σύμφωνα με το πλήθος των παραδειγμάτων που πραγματοποιήθηκε, μπορούμε να πούμε ότι η μοναδικότητα του σημείου αλλαγής όπως και η μονοτονία των σημείων αυτών ισχύουν. Επιπλέον, φαίνεται να ισχύει και η ύπαρξη μοναδικού σημείου τερματισμού του σταθερού στο σταθμό 1, όπως επίσης και η υπόθεση μας για μη φθίνουσα μονοτονία των σημείων αυτών. Αξίζει να σημειώσουμε ότι, όταν το κόστος αναμονής στην πρώτη ουρά είναι αρκετά μικρότερο από το αντίστοιχο της ουράς στο σταθμό 2, τότε ο σταθερός της ουράς 1 μπορεί να αδρανοποιηθεί όταν δεν υπάρχουν δουλείες στον σταθμό 2.

3.3.4 Γενικά συμπεράσματα για $h_1 < h_2$

Τελικά τα συμπεράσματα μας, από όλα τα παραπάνω, για τις περιπτώσεις που ισχύει $h_1 < h_2$, σύμφωνα με το περιορισμένο αριθμό των παραδειγμάτων, είναι τα εξής:

- Για όλες τις περιπτώσεις ισχύει η μοναδικότητα του σημείου αλλαγής και η μονοτονία των σημείων αυτών.
- Σε όλες τις περιπτώσεις ισχύει η μοναδικότητα του σημείου τερματισμού του σταθερού εξυπηρετητή στο σταθμό 1 όπως επίσης και η μη φθίνουσα μονοτονία αυτών των σημείων.

Κεφάλαιο 4. Σύνοψη αποτελεσμάτων

Όπως παρουσιάστηκαν αναλυτικά τα αποτελέσματα για κάθε περίπτωση είμαστε σε θέση να πούμε, με επιφυλακτικότητα, ότι η βέλτιστη πολιτική φαίνεται να χαρακτηρίζεται από καμπύλη αλλαγής και για το γενικό μοντέλο που μελετήσαμε. Ακόμα οι υποθέσεις που κάναμε σε σχέση με την μονοτονία της καμπύλης αλλαγής φαίνεται ότι δεν βρίσκει εφαρμογή σε όλες τις περιπτώσεις αλλά απορρίπτεται σε δύο από αυτές και συγκεκριμένα όταν έχουμε $h_1 \geq h_2$ και για τους ρυθμούς εξυπηρέτησης ισχύει $v_1 \geq \mu_1$ και $v_2 < \mu_2$ ή $v_1 < \mu_1$ και $v_2 < \mu_2$.

Συνεχίζοντας, για τις περιπτώσεις που ισχύει $h_1 < h_2$ σύμφωνα με τα αποτελέσματα μας μπορούμε να υποστηρίξουμε, με επιφυλακτικότητα, ότι ισχύει σε όλες τις περιπτώσεις η ύπαρξη μοναδικότητας του σημείου τερματισμού του σταθερού εξυπηρετητή στο σταθμό 1, όπως επίσης και η υπόθεσή μας ότι η ακολουθία των σημείων αυτών θα είναι μη φθίνουσα.

Βιβλιογραφία

1. Andradottir S., Ayhan H., Down D.G., (2013). Design principles for flexible systems, *Production and Operations Management*, Vol. 22, pp. 1144-1156.
2. Farrar, T.M. (1992). Resource allocation in systems of queues. Ph.D. dissertation, Cambridge University, Cambridge.
3. Farrar T.M. (1993). Optimal use of an extra server in a two station tandem queueing network, *IEEE Transactions on Automatic Control*, Vol. 38, pp. 1296-1299.
4. Hopp W.J., Van Oyen M.P., (2004). Agile workforce evaluation: a framework for cross-training and coordination, *IIE Transactions*, Vol. 36, pp. 919-940.
5. Pandelis, D.G. (2007). Optimal use of excess capacity in two interconnected queues. *Mathematical Methods of Operations Research* 65: 179–192.
6. Pandelis D.G., (2008a). Optimal stochastic scheduling of two interconnected queues with varying service rates, *Operations Research Letters*, Vol. 36, pp. 492-495.
7. Pandelis D.G., (2008b). Optimal control of flexible servers in two tandem queues with operating costs, *Probability in the Engineering and Informational Sciences*, Vol. 22, pp. 107-131.
8. Pandelis D.G., (2014). Optimal control of noncollaborative servers in two-stage tandem queueing systems, *Naval Research Logistics*, Vol. 61, pp. 435-446.
9. Tijms, H.C., *Stochastic modeling and analysis: A computing approach*, John Wiley, 1986.
10. 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.
11. Wu C.H., Down D.G., Lewis M.E., (2008). Heuristics for allocation of reconfigurable resources in a serial line with reliability considerations, *IIE Transactions*, Vol. 40, pp. 595-611.

Παράρτημα

Στο παράρτημα θα παρατεθούν τα προγράμματα τα οποία συντάξαμε στην γλώσσα προγραμματισμού μας (Matlab).

Αρχικά θα δοθούν τα προγράμματα για τις περιπτώσεις που ισχύει $h_1 \geq h_2$ και έπειτα τα προγράμματα για $h_1 < h_2$.

Συμβολισμοί που χρησιμοποιήθηκαν:

- Παράμετροι που χρησιμοποιήθηκαν για την σύγκριση του κόστους: K, G, T, Q, Z, r1, r2, F, P, MK, MZ, R, E και ER .
- Ο πίνακας που λαμβάνει τις τιμές του ελάχιστου κόστους σε κάθε περίοδο (n) συμβολίζεται με $V_n(x_1, x_2)$ ενώ για την προηγούμενη περίοδο (n-1) ο $V_p(x_1, x_2)$.
- Ο πίνακας που αποθηκεύει την θέση του ευέλκτου σε κάθε κατάσταση είναι ο $W_n(x_1, x_2)$.
- Ο πίνακας που αποθηκεύει σε κάθε περίοδο την τιμή για το κριτήριο του αλγορίθμου της επαναλαμβανόμενης-αξίας είναι ο $rep(kk)$.
- Ο πίνακας που αποθηκεύει τις διαφορές της περιόδου n με την n-1 είναι ο Dif.
- Τα όρια που χρειαζόμαστε για τον αλγόριθμο επαναλαμβανόμενης-αξίας είναι: Mn το άνω όριο και mn το κάτω όριο.
- Μετρητής των περιόδων: kk.
- Ο πίνακας που αποθηκεύει το σημείο αλλαγής του ευέλκτου είναι ο $B(x_1)$.
- Ο πίνακας που αποθηκεύει το σημείο αλλαγής του ευέλκτου στο πλαίσιο που πραγματοποιείτε ο έλεγχος και μας βοηθάει στα διαγράμματα είναι ο $DE(x_1)$.
- Ο πίνακας που αποθηκεύει την θέση του σταθερού του σταθμού 1 σε κάθε κατάσταση είναι ο $O(x_1, x_2)$.
- Ο πίνακας που αποθηκεύει την θέση που αδρανοποιείται ο σταθερός εξυπηρετητής του πρώτου σταθμού είναι ο $M(x_1)$.
- Ρυθμός αφίξεων: μέγιστη τιμή Im και με μείωση προκύπτει ο I4.
- Ρυθμοί εξυπηρέτησης: n3, m3, n4 και m4, όπου με τον δείκτη 3 αναφερόμαστε στο πρώτο σταθμό ενώ με τον 4 στο δεύτερο
- Αντίστοιχες πιθανότητες στο διακριτοποιημένο μοντέλο: n1, m1, n2, m2 και l.
- Κόστη παραμονής: h1 (σταθμός 1) και h2 (σταθμός 2).

- Μέγιστο πλήθος εργασιών: L1 (σταθμός 1) και L2 (σταθμός 2).
- Πίνακες που χρησιμοποιούνται για έλεγχο: A, C, X, U.
- Μετρητές που χρησιμοποιούμε: i, j και rep.
- Βοηθητικές παράμετροι: pl, e, s, d, st, pr, vd.

Προγράμματα για $h_1 \geq h_2$

Περίπτωση 1: $v_1 \geq \mu_1$ και $v_2 \geq \mu_2$

```
% {
  case for n1>=m1
  and n2>=m2 (h1>=h2)
% }
C=[];
A=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
m3=m3*n3;
n4=rand(1,1);
m4=rand(1,1);
m4=m4*n4;
if(n3+m3<=n4)
  lm=n3+m3;
  pr=1;
elseif(n4+m4<=n3)
  lm=n4+m4;
  pr=2;
else
  lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
  pr=3;
end
l4=lm/2;
% set the values of h1, h2
h3=1+4*rand(1,1);
h4=rand(1,1);
h4=h4*h3;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;

st=n1+n2+m1+m2+l;
for i=0:L1
  for j=0:L2
    Vn(i+1,j+1)=0;
```

```

    end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while (rep(kk)>0.001*mn )

% {
    Set the values of the dynamic tables
    and specify the minimum cost
% }

Vn(1,1)=b*(1*Vp(2,1)+(1-1)*Vp(1,1));
Wn(1,1)=0;
Vn(1,2)=h2+b*(n2*Vp(1,1)+1*Vp(2,2)+(1-n2-1)*Vp(1,2));
Wn(1,2)=0;
for j=2:L2
    if (j<L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
        Wn(1,j+1)=2;
    elseif (j==L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
        Wn(1,j+1)=2;
    end
end
Vn(2,1)=h1+b*(n1*Vp(1,2)+1*Vp(3,1)+(1-n1-1)*Vp(2,1));
Wn(2,1)=0;
Vn(2,2)=h1+h2+b*((n1*Vp(1,3))+(n2*Vp(2,1))+(1*Vp(3,2))+((1-n1-n2-1)*Vp(2,2)));
Wn(2,2)=0;
for j=2:L2
    if (j<L2)
        Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-1)*Vp(2,j+1)));
        Wn(2,j+1)=2;
    elseif (j==L2)
        Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-1)*Vp(2,j+1)));
        Wn(2,j+1)=2;
    end
end
for i=2:L1
    if (i<L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+1*Vp(i+2,1)+(1-n1-m1-1)*Vp(i+1,1));
        Wn(i+1,1)=1;
        Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m1-n2-1)*Vp(i+1,2));
        Wn(i+1,2)=1;
    elseif (i==L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+1*Vp(i+1,1)+(1-n1-m1-1)*Vp(i+1,1));
        Wn(i+1,1)=1;
    end
end

```

```

Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3)+(n2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
Wn(i+1,2)=1;
end
for j=2:L2
if (i<L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2)+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
if (Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i==L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2)+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
if (Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i==L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1)+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
if (Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i<L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1)+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
if (Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
end
end
end
Dif=(Vn-Vp);
Mn=max (max (Dif));
mn=min (min (Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end
% Specify the changes points
s=0;
for i=0:L1
for j=0:L2

```

```

        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
end
if (s<L2)
    B(i+1)=s+1;
else
    B(i+1)=L2+100;
end
end
% Specify the changes points
s=0;
for i=0:50
    for j=0:50
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if (s<L2)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end
end
vd=0:50;
%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
    if( DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
end
%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end

```

```
end
end
```

Περίπτωση 2 : $v_1 \geq \mu_1$ και $v_2 < \mu_2$

```
%{
case for n1>=m1
and n2<m2 (h1>=h2)
%}
C=[];
A=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
m3=m3*n3;
n4=rand(1,1);
m4=rand(1,1);
n4=n4*m4;
% in case n2=m2. We want n2<m2
while( n4==m4)
n4=rand(1,1);
n4=n4*m4;
end

if(n3+m3<=n4)
lm=n3+m3;
pr=1;
elseif(n4+m4<=n3)
lm=n4+m4;
pr=2;
else
lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
pr=3;
end
l4=lm/2;
% set the values of h1, h2
h3=1+4*rand(1,1);
h4=rand(1,1);
h4=h4*h3;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;
```

```

st=n1+n2+m1+m2+1;
for i=0:L1
    for j=0:L2
        Vn(i+1,j+1)=0;
    end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn )
    % {
    Set the values of the dynamic tables
    and specify the minimum cost
    % }
    Vn(1,1)=b*(1*Vp(2,1)+(1-1)*Vp(1,1));
    Wn(1,1)=0;
    Vn(1,2)=h2+b*(m2*Vp(1,1)+1*Vp(2,2)+(1-m2-1)*Vp(1,2));
    Wn(1,2)=2;
    for j=2:L2
        if(j<L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
            Wn(1,j+1)=2;
        elseif(j==L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
            Wn(1,j+1)=2;
        end
    end
    Vn(2,1)=h1+b*(n1*Vp(1,2)+1*Vp(3,1)+(1-n1-1)*Vp(2,1));
    Wn(2,1)=0;
    Vn(2,2)=h1+h2+b*((n1*Vp(1,3))+(m2*Vp(2,1))+(1*Vp(3,2))+((1-n1-m2-1)*Vp(2,2)));
    Wn(2,2)=2;
    for j=2:L2
        if(j<L2)
            Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-1)*Vp(2,j+1)));
            Wn(2,j+1)=2;
        elseif(j==L2)
            Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-1)*Vp(2,j+1)));
            Wn(2,j+1)=2;
        end
    end
    for i=2:L1
        if (i<L1)
            Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+1*Vp(i+2,1)+(1-n1-m1-1)*Vp(i+1,1));
            Wn(i+1,1)=1;
            Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+((n2+m2)*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m1-n2-1)*Vp(i+1,2));
            Wn(i+1,2)=1;
            G=(h1*i)+h2+b*((n1)*Vp(i,3))+((m2)*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m2-1)*Vp(i+1,2));
            if(G<=Vn(i+1,2))

```



```

Vn(i+1,2)=G;
Wn(i+1,2)=2;
end
elseif (i==L1)
Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
Wn(i+1,1)=1;
Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3)+(n2*Vp(i+1,1))+l*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
Wn(i+1,2)=1;
G=(h1*i)+h2+b*((n1)*Vp(i,3)+(m2*Vp(i+1,1))+l*Vp(i+1,2))+((1-n1-m2-l)*Vp(i+1,2));
if(G<=Vn(i+1,2))
Vn(i+1,2)=G;
Wn(i+1,2)=2;
end
end
for j=2:L2
if (i<L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+l*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i==L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+l*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1))+n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i==L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+l*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1))+n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
elseif (i<L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+l*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
end
end
end

```

```

end

Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1;
rep(kk)=Mn-mn;
Vp=Vn;
end

%Specify the changes points
for i=0:L1
    for j=0:L2
        if(Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<L2)
        B(i+1)=s+1;
    else
        B(i+1)=L2+100;
    end
end
for i=0:50
    for j=0:50
        if(Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<50)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end
end

end
vd=0:50;
%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
    if( DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)

```

```

        A=[A; n3 m3 n4 m4 h3 h4 l4];
        %A=[A d];
    end
end
end
end
end
%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end
end

```

Περίπτωση 3 : $v_1 < \mu_1$ και $v_2 \geq \mu_2$

```

%{
case for n1<m1
and n2>=m2 (h1>=h2)
%}
C=[];
A=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
n3=n3*m3;
n4=rand(1,1);
m4=rand(1,1);
m4=m4*n4;
% set the values of h1, h2
h3=1+4*rand(1,1);
h4=rand(1,1);
h4=h4*h3;
while(n3==m3)
    n3=rand(1,1);
    n3=n3*m3;
end

if(n3+m3<=n4)
    lm=n3+m3;
    pr=1;
elseif(n4+m4<=n3)
    lm=n4+m4;
    pr=2;
else
    lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
    pr=3;
end
l4=lm/2;

```

```

con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;
st=n1+n2+m1+m2+l;
for i=0:L1
    for j=0:L2
        Vn(i+1,j+1)=0;
    end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn )
    % {
    % Set the values of the dynamic tables
    % and specify the minimum cost
    % }
    Vn(1,1)=b*(l*Vp(2,1)+(1-l)*Vp(1,1));
    Wn(1,1)=0;
    Vn(1,2)=h2+b*(n2*Vp(1,1)+l*Vp(2,2)+(1-n2-l)*Vp(1,2));
    Wn(1,2)=0;
    for j=2:L2
        if(j<L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
            Wn(1,j+1)=2;
        elseif(j==L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
            Wn(1,j+1)=2;
        end
    end
end
Vn(2,1)=h1+b*(m1*Vp(1,2)+l*Vp(3,1)+(1-m1-l)*Vp(2,1));
Wn(2,1)=1;
Vn(2,2)=h1+h2+b*((m1*Vp(1,3))+(n2*Vp(2,1))+(l*Vp(3,2))+((1-m1-n2-l)*Vp(2,2)));
Wn(2,2)=1;
for j=2:L2
    if(j<L2)
        Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(1,j+2))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
        Wn(2,j+1)=1;
        P=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        if(P<=Vn(2,j+1))
            Vn(2,j+1)=P;
            Wn(2,j+1)=2;
        end
    end
end

```

```

elseif(j==L2)
Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(2,j+1))+((n2)*Vp(2,j))+(1*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
Wn(2,j+1)=1;
P=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
if(P<=Vn(2,j+1))
    Vn(2,j+1)=P;
    Wn(2,j+1)=2;
end
end
end

for i=2:L1
if (i<L1)
Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+1*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
Wn(i+1,1)=1;
Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+((n2)*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
Wn(i+1,2)=1;
elseif (i==L1)
Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+1*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
Wn(i+1,1)=1;
Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+((n2)*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
Wn(i+1,2)=1;
end

for j=2:L2
if (i<L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+((n2+m2)*Vp(i+1,j))+1*(Vp(i+2,j+1))+((1-n1-n2-m2-l)*Vp(i+1,j+1)));
if(Q<=Vn(i+1,j+1))
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
end
elseif (i==L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+((n2+m2)*Vp(i+1,j))+1*(Vp(i+1,j+1))+((1-n1-n2-m2-l)*Vp(i+1,j+1)));
if(Q<=Vn(i+1,j+1))
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
end
elseif (i==L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+((n2+m2)*Vp(i+1,j))+1*(Vp(i+1,j+1))+((1-n1-n2-m2-l)*Vp(i+1,j+1)));
if(Q<=Vn(i+1,j+1))
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
end
elseif (i<L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));

```

```

Wn(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1))
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
end
end
end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1;
rep(kk)=Mn-mn;
Vp=Vn;
end

```

%Specify the changes points

```

for i=0:L1
for j=0:L2
if (Wn(i+1,j+1)<2)
s=j;
end
end
if (s<L2)
B(i+1)=s+1;
else
B(i+1)=L2+100;
end
end
end

```

```

for i=0:50
for j=0:50
if(Wn(i+1,j+1)<2)
s=j;
end
end
if (s<50)
DE(i+1)=s+1;
else
DE(i+1)=L2+100;
end
end
end
vd=0:50;

```

%check uniqueness point change

```

for i=0:50
if(DE(i+1)<L2+100)
for j=0:DE(i+1)-1
if(Wn(i+1,j+1)==2)

```

```

        A=[A; n3 m3 n4 m4 h3 h4 l4];
    end
end
end

if(DE(i+1)==L2+100)
    for j=0:50
        if(Wn(i+1,j+1)==2)
            A=[A; n3 m3 n4 m4 h3 h4 l4];
        end
    end
end
end
end

%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end
end

```

Περίπτωση 4: $v_1 < \mu_1$ και $v_2 < \mu_2$

```

% {
    case for n1<m1
        and n2<m2 (h1>=h2)
    % }
C=[];
A=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
n3=n3*m3;
% in case n1=m1. we want n1<m1
while( n3==m3)
n3=rand(1,1);
n3=n3*m3;
end
n4=rand(1,1);
m4=rand(1,1);
n4=n4*m4;
% in case n2=m2. we want n2<m2
while( n4==m4)
n4=rand(1,1);
n4=n4*m4;
end
% set the values of h1, h2

```

```

h3=1+4*rand(1,1);
h4=rand(1,1);
h4=h4*h3;

if(n3+m3<=n4)
    lm=n3+m3;
    pr=1;
elseif(n4+m4<=n3)
    lm=n4+m4;
    pr=2;
else
    lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
    pr=3;
end
l4=lm/2;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;
st=n1+n2+m1+m2+l;

for i=0:L1
    for j=0:L2
        Vn(i+1,j+1)=0;
    end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn)
    % {
    % Set the values of the dynamic tables
    % and specify the minimum cost
    % }
    Vn(1,1)=b*(1*Vp(2,1)+(1-1)*Vp(1,1));
    Wn(1,1)=0;
    Vn(1,2)=h2+b*(m2*Vp(1,1)+l*Vp(2,2)+(1-m2-l)*Vp(1,2));
    Wn(1,2)=2;
    for j=2:L2
        if(j<L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
            Wn(1,j+1)=2;
        elseif(j==L2)
            Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));

```



```

Wn(1,j+1)=2;
end
end
Vn(2,1)=h1+b*(m1*Vp(1,2)+l*Vp(3,1)+(1-m1-l)*Vp(2,1));
Wn(2,1)=1;
Vn(2,2)=h1+h2+b*((m1*Vp(1,3))+(n2*Vp(2,1))+(l*Vp(3,2))+((1-m1-n2-l)*Vp(2,2)));
Wn(2,2)=1;
E=h1+h2+b*((n1*Vp(1,3))+(m2*Vp(2,1))+(l*Vp(3,2))+((1-n1-m2-l)*Vp(2,2)));
if(E<=Vn(2,2))
    Vn(2,2)=E;
    Wn(2,2)=2;
end
for j=2:L2
    if(j<L2)
        Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(1,j+2))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
        Wn(2,j+1)=1;
        P=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        if(P<=Vn(2,j+1))
            Vn(2,j+1)=P;
            Wn(2,j+1)=2;
        end
    elseif(j==L2)
        Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(2,j+1))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
        Wn(2,j+1)=1;
        P=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        if(P<=Vn(2,j+1))
            Vn(2,j+1)=P;
            Wn(2,j+1)=2;
        end
    end
end
for i=2:L1
    if (i<L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+((n2)*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
        Wn(i+1,2)=1;
        R=(h1*i)+h2+b*((n1)*Vp(i,3))+((m2)*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n1-m2-l)*Vp(i+1,2)));
        if(R<=Vn(i+1,2))
            Vn(i+1,2)=R;
            Wn(i+1,2)=2;
        end
    elseif (i==L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+((n2)*Vp(i+1,1))+l*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
        Wn(i+1,2)=1;
        R=(h1*i)+h2+b*((n1)*Vp(i,3))+((m2)*Vp(i+1,1))+l*(Vp(i+1,2))+((1-n1-m2-l)*Vp(i+1,2)));
        if(R<=Vn(i+1,2))
            Vn(i+1,2)=R;
        end
    end
end

```

```

    Wn(i+1,2)=2;
end
end
for j=2:L2
    if (i<L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
        end
    elseif (i==L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
        end
    elseif (i==L1 && j==L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
        end
    elseif (i<L1 && j==L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
        end
    end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end

%Specify the changes points
for i=0:L1
    for j=0:L2

```

```

        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
end
if (s<L2)
    B(i+1)=s+1;
else
    B(i+1)=L2+100;
end
end
end

for i=0:50
    for j=0:50
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if (s<50)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end
end
end
vd=0:50;

%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
    if (DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
end

%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end
end

```

Προγράμματα για $h_1 < h_2$

Περίπτωση 1: $v_1 \geq \mu_1$ και $v_2 \geq \mu_2$

```
% {  
  case for n1>=m1  
  and n2>=m2 (h1>=h2)  
% }  
C=[];  
A=[];  
X=[];  
U=[];  
L1=300;  
L2=300;  
b=1;  
%assigning values to the service rates  
n3=rand(1,1);  
m3=rand(1,1);  
m3=m3*n3;  
n4=rand(1,1);  
m4=rand(1,1);  
m4=m4*n4;  
h4=1+4*rand(1,1);  
h3=rand(1,1);  
h3=h3*h4;  
if(n3+m3<=n4)  
  lm=n3+m3;  
  pr=1;  
elseif(n4+m4<=n3)  
  lm=n4+m4;  
  pr=2;  
else  
  lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);  
  pr=3;  
end  
l4=lm/2;  
con=n3+n4+m3+m4+l4;  
n1=n3/con;  
n2=n4/con;  
m1=m3/con;  
m2=m4/con;  
l=l4/con;  
h1=h3/con;  
h2=h4/con;  
  
st=n1+n2+m1+m2+l;  
for i=0:L1  
  for j=0:L2
```

```

    Vn(i+1,j+1)=0;
end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn )
% {
    Set the values of the dynamic tables
    and specify the minimum cost
% }
Vn(1,1)=b*(1*Vp(2,1)+(1-1)*Vp(1,1));
Wn(1,1)=0;
O(1,1)=0;
Vn(1,2)=h2+b*(n2*Vp(1,1)+1*Vp(2,2)+(1-n2-1)*Vp(1,2));
Wn(1,2)=0;
O(1,2)=0;
for j=2:L2
    if(j<L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
        Wn(1,j+1)=2;
        O(1,j+1)=0;
    elseif(j==L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-1)*Vp(1,j+1));
        Wn(1,j+1)=2;
        O(1,j+1)=0;
    end
end
Vn(2,1)=h1+b*(n1*Vp(1,2)+1*Vp(3,1)+(1-n1-1)*Vp(2,1));
Wn(2,1)=0;
O(2,1)=1;
Mk=h1+b*(1*Vp(3,1)+(1-1)*Vp(2,1));
if(Mk<=Vn(2,1))
    Vn(2,1)=Mk;
    Wn(2,1)=0;
    O(2,1)=0;
end
Vn(2,2)=h1+h2+b*((n1*Vp(1,3))+n2*Vp(2,1))+1*Vp(3,2))+((1-n1-n2-1)*Vp(2,2));
Wn(2,2)=0;
O(2,2)=1;
F=h1+h2+b*((n2*Vp(2,1))+1*Vp(3,2))+((1-n2-1)*Vp(2,2));
if(F<=Vn(2,2))
    Wn(2,2)=0;
    Vn(2,2)=F;
    O(2,2)=0;
end
for j=2:L2
    if(j<L2)

```

```

Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
Wn(2,j+1)=2;
O(2,j+1)=1;
P=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+l*Vp(3,j+1)+((1-n2-m2-l)*Vp(2,j+1));
if(P<=Vn(2,j+1))
    Wn(2,j+1)=2;
    Vn(2,j+1)=P;
    O(2,j+1)=0;
end
elseif(j==L2)
Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
Wn(2,j+1)=2;
O(2,j+1)=1;
P=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+l*Vp(3,j+1)+((1-n2-m2-l)*Vp(2,j+1));
if(P<=Vn(2,j+1))
    Wn(2,j+1)=2;
    Vn(2,j+1)=P;
    O(2,j+1)=0;
end
end
end

for i=2:L1
    if (i<L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        Mz=(h1*i)+b*(l*Vp(i+2,1)+(1-l)*Vp(i+1,1));
        if(Mz<=Vn(i+1,1))
            Vn(i+1,1)=Mz;
            Wn(i+1,1)=0;
            O(i+1,1)=0;
        end
        Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+n2*Vp(i+1,1)+l*Vp(i+2,2)+((1-n1-m1-n2-l)*Vp(i+1,2));
        Wn(i+1,2)=1;
        O(i+1,2)=1;
        R=(h1*i)+h2+b*((n2*Vp(i+1,1))+l*Vp(i+2,2))+((1-n2-l)*Vp(i+1,2));
        if(R<=Vn(i+1,2))
            Wn(i+1,2)=0;
            Vn(i+1,2)=R;
            O(i+1,2)=0;
        end
    end
    elseif (i==L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        Mz=(h1*i)+b*(l*Vp(i+1,1)+(1-l)*Vp(i+1,1));
        if(Mz<=Vn(i+1,1))
            Vn(i+1,1)=Mz;
            Wn(i+1,1)=0;
        end
    end
end

```

```

O(i+1,1)=0;
end
Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2));
Wn(i+1,2)=1;
O(i+1,2)=1;
R=(h1*i)+h2+b*((n2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n2-l)*Vp(i+1,2)));
if(R<=Vn(i+1,2))
    Wn(i+1,2)=0;
    Vn(i+1,2)=R;
    O(i+1,2)=0;
end
end
for j=2:L2
    if (i<L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        O(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
        if (T<=Vn(i+1,j+1)&&T<=Q)
            Vn(i+1,j+1)=T;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=0;
        elseif (Q<T&&Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=1;
        end
    elseif (i==L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        O(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
        if (T<=Vn(i+1,j+1)&&T<=Q)
            Vn(i+1,j+1)=T;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=0;
        elseif (Q<T&&Q<=Vn(i+1,j+1))
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=1;
        end
    end
    elseif (i==L1 && j==L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        O(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
        if (T<=Vn(i+1,j+1)&&T<=Q)

```

```

    Vn(i+1,j+1)=T;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=0;
elseif (Q<T&&Q<=Vn(i+1,j+1))
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=1;
end
elseif(i<L1 && j==L2)
    Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+l*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
    Wn(i+1,j+1)=1;
    O(i+1,j+1)=1;
    Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
    T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+(1-n2-m2-l)*Vp(i+1,j+1));
    if (T<=Vn(i+1,j+1)&&T<=Q)
        Vn(i+1,j+1)=T;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=0;
    elseif (Q<T&&Q<=Vn(i+1,j+1))
        Vn(i+1,j+1)=Q;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=1;
    end
end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end

%Specify the changes points
for i=0:L1
    for j=0:L2
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<L2)
        B(i+1)=s+1;
    else
        B(i+1)=L2+100;
    end
end
end

for i=0:50
    for j=0:50

```



```

    if(Wn(i+1,j+1)<2)
        s=j;
    end
end
if(s<50)
    DE(i+1)=s+1;
else
    DE(i+1)=L2+100;
end

end
vd=0:50;
%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4 i DE(i+1) j];
                %A=[A d];
            end
        end
    end
    if(DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4 i DE(i+1) j];
                %A=[A d];
            end
        end
    end
end
%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4 DE(i+1) DE(i+2)];
    end
end

%Specify the shutdown points
for i=1:50
    for j=0:50
        if(O(i+1,j+1)==1)
            e=j;
        end
    end

    if(e<50)
        M(i+1)=e+1;
    else
        M(i+1)=L2+100;
    end
end

```

```

end
end
%check uniqueness of shutdown points
for i=1:50
if(M(i+1)<L2+100)
for j=0:M(i+1)-1
if(O(i+1,j+1)==0)
U=[U; n3 m3 n4 m4 h3 h4];
%U=[U d];
end
end
end

if (M(i+1)==L2+100)
for j=0:50
if (O(i+1,j+1)==0)
U=[U; n3 m3 n4 m4 h3 h4];
%U=[U d];
end
end
end
end

%check the monotonicity of shutdown points
for i=1:50-2
if (M(i+1)>M(i+2))
X=[X; n3 m3 n4 m4 h3 h4];
end
end

```

Περίπτωση 2 : $v_1 \geq \mu_1$ και $v_2 < \mu_2$

```

%{
case for n1>=m1
and n2<m2 (h1<h2)
%}
C=[];
A=[];
U=[];
X=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
m3=m3*n3;
n4=rand(1,1);
m4=rand(1,1);
n4=n4*m4;

```

```

% in case n2=m2. we want n2<m2
while (n4==m4)
n4=rand(1,1);
n4=n4*m4;
end
% set the values of h1, h2
h4=1+4*rand(1,1);
h3=rand(1,1);
h3=h3*h4;
if(n3+m3<=n4)
lm=n3+m3;
pr=1;
elseif(n4+m4<=n3)
lm=n4+m4;
pr=2;
else
lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
pr=3;
end
l4=lm/2;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;
st=n1+n2+m1+m2+l;
for i=0:L1
for j=0:L2
Vn(i+1,j+1)=0;
end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn )

% {
Set the values of the dynamic tables
and specify the minimum cost
% }
Vn(1,1)=b*(1*Vp(2,1)+(1-l)*Vp(1,1));
Wn(1,1)=0;
O(1,1)=0;
Vn(1,2)=h2+b*(m2*Vp(1,1)+l*Vp(2,2)+(1-m2-l)*Vp(1,2));
Wn(1,2)=2;

```

```

O(1,2)=0;
for j=2:L2
    if(j<L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
        Wn(1,j+1)=2;
        O(1,j+1)=0;
    elseif(j==L2)
        Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+1*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
        Wn(1,j+1)=2;
        O(1,j+1)=0;
    end
end
Vn(2,1)=h1+b*(n1*Vp(1,2)+1*Vp(3,1)+(1-n1-l)*Vp(2,1));
Wn(2,1)=0;
O(2,1)=1;
MK=h1+b*(1*Vp(3,1)+(1-l)*Vp(2,1));
if(MK<Vn(2,1))
    Vn(2,1)=MK;
    Wn(2,1)=0;
    O(2,1)=0;
end
Vn(2,2)=h1+h2+b*((n1*Vp(1,3))+(m2*Vp(2,1))+(1*Vp(3,2))+((1-n1-m2-l)*Vp(2,2)));
Wn(2,2)=2;
O(2,2)=1;
F=h1+h2+b*((m2*Vp(2,1))+(1*Vp(3,2))+((1-m2-l)*Vp(2,2)));
if(F<Vn(2,2))
    Vn(2,2)=F;
    O(2,2)=0;
    Wn(2,2)=2;
end
for j=2:L2
    if(j<L2)
        Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        Wn(2,j+1)=2;
        O(2,j+1)=1;
        P=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
        if(P<=Vn(2,j+1))
            Vn(2,j+1)=P;
            O(2,j+1)=0;
            Wn(2,j+1)=2;
        end
    elseif(j==L2)
        Vn(2,j+1)=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        Wn(2,j+1)=2;
        O(2,j+1)=1;
        P=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(1*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
        if(P<Vn(2,j+1))
            Vn(2,j+1)=P;
            O(2,j+1)=0;
            Wn(2,j+1)=2;
        end
    end
end

```

```

    end
end
end
for i=2:L1
    if (i<L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        MZ=(h1*i)+b*(1*Vp(i+2,1)+(1-l)*Vp(i+1,1));
        if(MZ<Vn(i+1,1))
            Vn(i+1,1)=MZ;
            Wn(i+1,1)=0;
            O(i+1,1)=0;
        end
        Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
        Wn(i+1,2)=1;
        O(i+1,2)=1;
        G=(h1*i)+h2+b*(((n1)*Vp(i,3))+(m2*Vp(i+1,1))+(1*Vp(i+2,2))+((1-n1-m2-l)*Vp(i+1,2)));
        R=(h1*i)+h2+b*((m2*Vp(i+1,1))+(1*Vp(i+2,2))+((1-m2-l)*Vp(i+1,2)));
        if(G<=Vn(i+1,2)&& G<R)
            Vn(i+1,2)=G;
            Wn(i+1,2)=2;
            O(i+1,2)=1;
        elseif(R<=Vn(i+1,2)&&R<=G)
            Vn(i+1,2)=R;
            Wn(i+1,2)=2;
            O(i+1,2)=0;
        end
    elseif (i==L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        MZ=(h1*i)+b*(1*Vp(i+1,1)+(1-l)*Vp(i+1,1));
        if(MZ<Vn(i+1,1))
            Vn(i+1,1)=MZ;
            Wn(i+1,1)=0;
            O(i+1,1)=0;
        end
        Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
        Wn(i+1,2)=1;
        O(i+1,2)=1;
        G=(h1*i)+h2+b*(((n1)*Vp(i,3))+(m2*Vp(i+1,1))+(1*Vp(i+1,2))+((1-n1-m2-l)*Vp(i+1,2)));
        R=(h1*i)+h2+b*((m2*Vp(i+1,1))+(1*Vp(i+1,2))+((1-m2-l)*Vp(i+1,2)));
        if(G<=Vn(i+1,2)&& G<R)
            Vn(i+1,2)=G;
            Wn(i+1,2)=2;
            O(i+1,2)=1;
        elseif(R<=Vn(i+1,2)&&R<=G)
            Vn(i+1,2)=R;
            Wn(i+1,2)=2;

```

```

O(i+1,2)=0;
end
end
for j=2:L2
if (i<L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
O(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1)&& Q<T)
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
O(i+1,j+1)=1;
elseif(T<=Vn(i+1,j+1)&& T<=Q)
Vn(i+1,j+1)=T;
Wn(i+1,j+1)=2;
O(i+1,j+1)=0;
end
elseif (i==L1 && j<L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
O(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1)&& Q<T)
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
O(i+1,j+1)=1;
elseif(T<=Vn(i+1,j+1)&& T<=Q)
Vn(i+1,j+1)=T;
Wn(i+1,j+1)=2;
O(i+1,j+1)=0;
end
elseif(i==L1 && j==L2)
Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
O(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1)&& Q<T)
Vn(i+1,j+1)=Q;
Wn(i+1,j+1)=2;
O(i+1,j+1)=1;
elseif(T<=Vn(i+1,j+1)&& T<=Q)
Vn(i+1,j+1)=T;
Wn(i+1,j+1)=2;
O(i+1,j+1)=0;
end
elseif(i<L1 && j==L2)

```

```

Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
Wn(i+1,j+1)=1;
O(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1)+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1)&& Q<T)
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=1;
elseif(T<=Vn(i+1,j+1)&& T<=Q)
    Vn(i+1,j+1)=T;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=0;
end
end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end
%Specify the changes points
for i=0:L1
    for j=0:L2
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<L2)
        B(i+1)=s+1;
    else
        B(i+1)=L2+100;
    end
end
end

for i=0:50
    for j=0:50
        if(Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<50)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end
end
end

```

```

vd=0:50;
%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
    if(DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
end
%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end
%Specify the shutdown points
for i=1:50
    for j=0:50
        if(O(i+1,j+1)==1)
            e=j;
        end
    end

    if(e<50)
        M(i+1)=e+1;
    else
        M(i+1)=L2+100;
    end
end
%check uniqueness of shutdown points
for i=1:50
    if(M(i+1)<L2+100)
        for j=0:M(i+1)-1
            if(O(i+1,j+1)==0)
                U=[U; n3 m3 n4 m4 h3 h4 l4];
                %U=[U d];
            end
        end
    end
end

```



```

if (M(i+1)==L2+100)
    for j=0:50
        if (O(i+1,j+1)==0)
            U=[U;n3 m3 n4 m4 h3 h4 l4];
            %U=[U d];
        end
    end
end
end
end

%check the monotonicity of shutdown points
for i=1:50-1
    if (M(i+1)>M(i+2))
        X=[X;n3 m3 n4 m4 h3 h4 l4];
    end
end
end

```

Περίπτωση 3 : $v_1 < \mu_1$ και $v_2 \geq \mu_2$

```

%{
case for n1<m1
and n2>=m2 (h1<h2)
%}
C=[];
A=[];
U=[];
X=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
n3=n3*m3;
n4=rand(1,1);
m4=rand(1,1);
m4=m4*n4;
while (n3==m3)
n3=rand(1,1);
n3=n3*m3;
end
% set the values of h1, h2
h4=1+4*rand(1,1);
h3=rand(1,1);
h3=h3*h4;

if(n3+m3<=n4)
    lm=n3+m3;
    pr=1;
elseif(n4+m4<=n3)

```

```

lm=n4+m4;
pr=2;
else
lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
pr=3;
end
l4=lm/2;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;
st=n1+n2+m1+m2+l;
for i=0:L1
for j=0:L2

Vn(i+1,j+1)=0;

end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;
while(rep(kk)>0.001*mn )
% {
Set the values of the dynamic tables
and specify the minimum cost
% }
Vn(1,1)=b*(1*Vp(2,1)+(1-l)*Vp(1,1));
Wn(1,1)=0;
O(1,1)=0;
Vn(1,2)=h2+b*(n2*Vp(1,1)+l*Vp(2,2)+(1-n2-l)*Vp(1,2));
Wn(1,2)=0;
O(1,2)=0;
for j=2:L2
if(j<L2)
Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
Wn(1,j+1)=2;
O(1,j+1)=0;
elseif(j==L2)
Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
Wn(1,j+1)=2;
O(1,j+1)=0;
end
end
end

```

```

Vn(2,1)=h1+b*(m1*Vp(1,2)+l*Vp(3,1)+(1-m1-l)*Vp(2,1));
Wn(2,1)=1;
O(2,1)=0;
MK=h1+b*(l*Vp(3,1)+(1-l)*Vp(2,1));
if(MK<Vn(2,1))
    Vn(2,1)=MK;
    Wn(2,1)=0;
    O(2,1)=0;
end
Vn(2,2)=h1+h2+b*((m1*Vp(1,3))+(n2*Vp(2,1))+(l*Vp(3,2))+((1-m1-n2-l)*Vp(2,2)));
Wn(2,2)=1;
O(2,2)=0;
G=h1+h2+b*((n2*Vp(2,1))+(l*Vp(3,2))+((1-n2-l)*Vp(2,2)));
if(G<Vn(2,2))
    Vn(2,2)=G;
    Wn(2,2)=0;
    O(2,2)=0;
end
for j=2:L2
    if(j<L2)
        Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(1,j+2))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
        Wn(2,j+1)=1;
        O(2,j+1)=0;
        P=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        F=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
        if(P<=Vn(2,j+1)&&P<F)
            Vn(2,j+1)=P;
            Wn(2,j+1)=2;
            O(2,j+1)=1;
        elseif(F<=Vn(2,j+1)&&F<=P)
            Vn(2,j+1)=F;
            Wn(2,j+1)=2;
            O(2,j+1)=0;
        end
    elseif(j==L2)
        Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(2,j+1))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
        Wn(2,j+1)=1;
        P=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
        F=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
        if(P<=Vn(2,j+1)&&P<F)
            Vn(2,j+1)=P;
            Wn(2,j+1)=2;
            O(2,j+1)=1;
        elseif(F<=Vn(2,j+1)&&F<=P)
            Vn(2,j+1)=F;
            Wn(2,j+1)=2;
            O(2,j+1)=0;
        end
    end
end
end

```

```

for i=2:L1
  if (i<L1)
    Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
    Wn(i+1,1)=1;
    O(i+1,1)=1;
    MZ=(h1*i)+b*(l*Vp(i+2,1)+(1-l)*Vp(i+1,1));
    if(MZ<Vn(i+1,1))
      Vn(i+1,1)=MZ;
      Wn(i+1,1)=0;
      O(i+1,1)=0;
    end
  end
  Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
  Wn(i+1,2)=1;
  O(i+1,2)=1;
  K=(h1*i)+h2+b*((n2*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n2-l)*Vp(i+1,2)));
  if(K<=Vn(i+1,2))
    Vn(i+1,2)=K;
    Wn(i+1,2)=0;
    O(i+1,2)=0;
  end
end
elseif (i==L1)
  Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
  Wn(i+1,1)=1;
  O(i+1,1)=1;
  MZ=(h1*i)+b*(l*Vp(i+1,1)+(1-l)*Vp(i+1,1));
  if(MZ<Vn(i+1,1))
    Vn(i+1,1)=MZ;
    Wn(i+1,1)=0;
    O(i+1,1)=0;
  end
end
  Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+l*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
  Wn(i+1,2)=1;
  O(i+1,2)=1;
  K=(h1*i)+h2+b*((n2*Vp(i+1,1))+l*(Vp(i+1,2))+((1-n2-l)*Vp(i+1,2)));
  if(K<=Vn(i+1,2))
    Vn(i+1,2)=K;
    Wn(i+1,2)=0;
    O(i+1,2)=0;
  end
end
  for j=2:L2
    if (i<L1 && j<L2)
      Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+l*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
      Wn(i+1,j+1)=1;
      O(i+1,j+1)=1;
      Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2))+(n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+((1-n1-n2-m2-l)*Vp(i+1,j+1)));
      T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1))+((1-n2-m2-l)*Vp(i+1,j+1)));
      if(Q<Vn(i+1,j+1)&&Q<T)
        Vn(i+1,j+1)=Q;
      end
    end
  end

```

```

Wn(i+1,j+1)=2;
O(i+1,j+1)=1;
elseif(T<Vn(i+1,j+1)&&T<=Q)
    Vn(i+1,j+1)=T;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=0;
end
elseif (i==L1 && j<L2)
    Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i,j+2)+n2*Vp(i+1,j)+l*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
    Wn(i+1,j+1)=1;
    O(i+1,j+1)=1;
    Q=(h1*i)+(h2*j)+b*((n1*Vp(i,j+2)+(n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
    T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
    if(Q<Vn(i+1,j+1)&&Q<T)
        Vn(i+1,j+1)=Q;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=1;
    elseif(T<Vn(i+1,j+1)&&T<=Q)
        Vn(i+1,j+1)=T;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=0;
    end
elseif(i==L1 && j==L2)
    Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+l*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
    Wn(i+1,j+1)=1;
    O(i+1,j+1)=1;
    Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
    T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(l*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
    if(Q<Vn(i+1,j+1)&&Q<T)
        Vn(i+1,j+1)=Q;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=1;
    elseif(T<Vn(i+1,j+1)&&T<=Q)
        Vn(i+1,j+1)=T;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=0;
    end
elseif(i<L1 && j==L2)
    Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+l*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
    Wn(i+1,j+1)=1;
    O(i+1,j+1)=1;
    Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
    T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(l*Vp(i+2,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
    if(Q<Vn(i+1,j+1)&&Q<T)
        Vn(i+1,j+1)=Q;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=1;
    elseif(T<Vn(i+1,j+1)&&T<=Q)
        Vn(i+1,j+1)=T;
        Wn(i+1,j+1)=2;

```

```

        O(i+1,j+1)=0;
    end
end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end

%Specify the changes points
for i=0:L1
    for j=0:L2
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end

    if(s<L2)
        B(i+1)=s+1;
    else
        B(i+1)=L2+100;
    end
end

for i=0:50
    for j=0:50
        if(Wn(i+1,j+1)<2)
            s=j;
        end
    end

    if(s<50)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end

end
vd=0:50;

%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
            end
        end
    end
end

```

```

        %A=[A d];
    end
end
end
if(DE(i+1)==L2+100)
    for j=0:50
        if(Wn(i+1,j+1)==2)
            A=[A; n3 m3 n4 m4 h3 h4 l4];
            %A=[A d];
        end
    end
end
end
end

%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end
end

```

Περίπτωση 4: $v_1 < \mu_1$ και $v_2 < \mu_2$

```

%{
    case for n1<m1
    and n2<m2 (h1<h2)
%}
C=[];
A=[];
U=[];
X=[];
L1=300;
L2=300;
b=1;
%assigning values to the service rates
n3=rand(1,1);
m3=rand(1,1);
n3=n3*m3;
% in case n1=m1. we want n1<m1
while( n3==m3)
n3=rand(1,1);
n3=n3*m3;
end
n4=rand(1,1);
m4=rand(1,1);
n4=n4*m4;
% in case n2=m2. we want n2<m2
while( n4==m4)
n4=rand(1,1);

```

```

n4=n4*m4;
end
% set the values of h1, h2
h4=1+4*rand(1,1);
h3=rand(1,1);
h3=h3*h4;
if(n3+m3<=n4)
    lm=n3+m3;
    pr=1;
elseif(n4+m4<=n3)
    lm=n4+m4;
    pr=2;
else
    lm=(m3*n4+n3*m4+m3*m4)/(m3+m4);
    pr=3;
end
l4=lm/2;
con=n3+n4+m3+m4+l4;
n1=n3/con;
n2=n4/con;
m1=m3/con;
m2=m4/con;
l=l4/con;
h1=h3/con;
h2=h4/con;

st=n1+n2+m1+m2+1;
for i=0:L1
    for j=0:L2
        Vn(i+1,j+1)=0;
    end
end
Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;

while(rep(kk)>0.001*mn )
    % {
    % Set the values of the dynamic tables
    % and specify the minimum cost
    % }
    Vn(1,1)=b*(1*Vp(2,1)+(1-l)*Vp(1,1));
    Wn(1,1)=0;
    O(1,1)=0;
    Vn(1,2)=h2+b*(m2*Vp(1,1)+l*Vp(2,2)+(1-m2-l)*Vp(1,2));
    Wn(1,2)=2;
    O(1,2)=0;
    for j=2:L2

```



```

if(j<L2)
Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
Wn(1,j+1)=2;
O(1,j+1)=0;
elseif(j==L2)
Vn(1,j+1)=(h2*j)+b*((n2+m2)*Vp(1,j)+l*Vp(2,j+1)+(1-n2-m2-l)*Vp(1,j+1));
Wn(1,j+1)=2;
O(1,j+1)=0;
end
end
Vn(2,1)=h1+b*(m1*Vp(1,2)+l*Vp(3,1)+(1-m1-l)*Vp(2,1));
Wn(2,1)=1;
O(2,1)=0;
MK=h1+b*(l*Vp(3,1)+(1-l)*Vp(2,1));
if(MK<Vn(2,1))
Vn(2,1)=MK;
Wn(2,1)=0;
O(2,1)=0;
end
Vn(2,2)=h1+h2+b*((m1*Vp(1,3))+(n2*Vp(2,1))+(l*Vp(3,2))+((1-m1-n2-l)*Vp(2,2)));
Wn(2,2)=1;
O(2,2)=0;
E=h1+h2+b*((n1*Vp(1,3))+(m2*Vp(2,1))+(l*Vp(3,2))+((1-n1-m2-l)*Vp(2,2)));
ER=h1+h2+b*((m2*Vp(2,1))+(l*Vp(3,2))+((1-m2-l)*Vp(2,2)));
if(E<=Vn(2,2)&& E<ER)
Vn(2,2)=E;
Wn(2,2)=2;
O(2,2)=1;
elseif(ER<=Vn(2,2)&& ER<=E)
Vn(2,2)=E;
Wn(2,2)=2;
O(2,2)=0;
end
for j=2:L2
if(j<L2)
Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(1,j+2))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
Wn(2,j+1)=1;
O(2,j+1)=0;
P=h1+(h2*j)+b*((n1*Vp(1,j+2))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
F=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
if(P<=Vn(2,j+1)&& P<F)
Vn(2,j+1)=P;
Wn(2,j+1)=2;
O(2,j+1)=1;
elseif(F<=Vn(2,j+1)&& F<=P)
Vn(2,j+1)=F;
Wn(2,j+1)=2;
O(2,j+1)=0;
end
elseif(j==L2)

```

```

Vn(2,j+1)=h1+(h2*j)+b*((m1*Vp(2,j+1))+((n2)*Vp(2,j))+(l*Vp(3,j+1))+((1-m1-n2-l)*Vp(2,j+1)));
Wn(2,j+1)=1;
O(2,j+1)=0;
P=h1+(h2*j)+b*((n1*Vp(2,j+1))+((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n1-n2-m2-l)*Vp(2,j+1)));
F=h1+(h2*j)+b*((n2+m2)*Vp(2,j))+(l*Vp(3,j+1))+((1-n2-m2-l)*Vp(2,j+1));
if(P<=Vn(2,j+1)&&P<F)
    Vn(2,j+1)=P;
    Wn(2,j+1)=2;
    O(2,j+1)=1;
elseif(F<=Vn(2,j+1)&&F<=P)
    Vn(2,j+1)=F;
    Wn(2,j+1)=2;
    O(2,j+1)=0;
end
end
end

for i=2:L1
    if (i<L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,2)+l*Vp(i+2,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        MZ=(h1*i)+b*(l*Vp(i+2,1)+(1-l)*Vp(i+1,1));
        if(MZ<Vn(i+1,1))
            Vn(i+1,1)=MZ;
            Wn(i+1,1)=0;
            O(i+1,1)=0;
        end
        Vn(i+1,2)=(h1*i)+h2+b*((n1+m1)*Vp(i,3))+((n2)*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n1-m1-n2-l)*Vp(i+1,2));
        Wn(i+1,2)=1;
        O(i+1,2)=1;
        R=(h1*i)+h2+b*((n1)*Vp(i,3))+((m2)*Vp(i+1,1))+(l*Vp(i+2,2))+((1-n1-m2-l)*Vp(i+1,2));
        G=(h1*i)+h2+b*((m2)*Vp(i+1,1))+(l*Vp(i+2,2))+((1-m2-l)*Vp(i+1,2));
        if(R<=Vn(i+1,2)&&R<G)
            Vn(i+1,2)=R;
            Wn(i+1,2)=2;
            O(i+1,2)=1;
        elseif(G<=Vn(i+1,2)&&G<=R)
            Vn(i+1,2)=G;
            Wn(i+1,2)=2;
            O(i+1,2)=0;
        end
    end
    elseif (i==L1)
        Vn(i+1,1)=(h1*i)+b*((n1+m1)*Vp(i,3)+l*Vp(i+1,1)+(1-n1-m1-l)*Vp(i+1,1));
        Wn(i+1,1)=1;
        O(i+1,1)=1;
        MZ=(h1*i)+b*(l*Vp(i+1,1)+(1-l)*Vp(i+1,1));
        if(MZ<Vn(i+1,1))
            Vn(i+1,1)=MZ;
            Wn(i+1,1)=0;
        end
    end
end

```

```

O(i+1,1)=0;
end
Vn(i+1,2)=(h1*i)+h2+b*(((n1+m1)*Vp(i,3))+(n2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m1-n2-l)*Vp(i+1,2)));
Wn(i+1,2)=1;
O(i+1,2)=1;
R=(h1*i)+h2+b*(((n1)*Vp(i,3))+(m2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-n1-m2-l)*Vp(i+1,2)));
G=(h1*i)+h2+b*(((m2*Vp(i+1,1))+1*(Vp(i+1,2))+((1-m2-l)*Vp(i+1,2))));
if(R<=Vn(i+1,2)&&R<G)
    Vn(i+1,2)=R;
    Wn(i+1,2)=2;
    O(i+1,2)=1;
elseif(G<=Vn(i+1,2)&&G<=R)
    Vn(i+1,2)=G;
    Wn(i+1,2)=2;
    O(i+1,2)=0;
end
end
for j=2:L2
    if (i<L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*(((n1+m1)*Vp(i,j+2))+n2*Vp(i+1,j))+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        O(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*(((n1)*Vp(i,j+2))+(n2+m2)*Vp(i+1,j))+1*Vp(i+2,j+1)+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        T=(h1*i)+(h2*j)+b*(((n2+m2)*Vp(i+1,j))+1*Vp(i+2,j+1))+1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1)&&Q<T)
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=1;
        elseif(T<=Vn(i+1,j+1)&&T<=Q)
            Vn(i+1,j+1)=T;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=0;
        end
    end
    elseif (i==L1 && j<L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*(((n1+m1)*Vp(i,j+2))+n2*Vp(i+1,j))+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
        Wn(i+1,j+1)=1;
        O(i+1,j+1)=1;
        Q=(h1*i)+(h2*j)+b*(((n1)*Vp(i,j+2))+(n2+m2)*Vp(i+1,j))+1*Vp(i+1,j+1)+(1-n1-n2-m2-l)*Vp(i+1,j+1));
        T=(h1*i)+(h2*j)+b*(((n2+m2)*Vp(i+1,j))+1*Vp(i+1,j+1))+1-n2-m2-l)*Vp(i+1,j+1));
        if(Q<=Vn(i+1,j+1)&&Q<T)
            Vn(i+1,j+1)=Q;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=1;
        elseif(T<=Vn(i+1,j+1)&&T<=Q)
            Vn(i+1,j+1)=T;
            Wn(i+1,j+1)=2;
            O(i+1,j+1)=0;
        end
    end
    elseif(i==L1 && j==L2)
        Vn(i+1,j+1)=(h1*i)+(h2*j)+b*(((n1+m1)*Vp(i+1,j+1))+n2*Vp(i+1,j))+1*Vp(i+1,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));

```

```

Wn(i+1,j+1)=1;
O(i+1,j+1)=1;
Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+1,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
if(Q<=Vn(i+1,j+1)&&Q<T)
    Vn(i+1,j+1)=Q;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=1;
elseif(T<=Vn(i+1,j+1)&&T<=Q)
    Vn(i+1,j+1)=T;
    Wn(i+1,j+1)=2;
    O(i+1,j+1)=0;
end
elseif(i<L1 && j==L2)
    Vn(i+1,j+1)=(h1*i)+(h2*j)+b*((n1+m1)*Vp(i+1,j+1)+n2*Vp(i+1,j)+1*Vp(i+2,j+1)+(1-n1-m1-n2-l)*Vp(i+1,j+1));
    Wn(i+1,j+1)=1;
    O(i+1,j+1)=1;
    Q=(h1*i)+(h2*j)+b*((n1*Vp(i+1,j+1))+(n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n1-n2-m2-l)*Vp(i+1,j+1));
    T=(h1*i)+(h2*j)+b*((n2+m2)*Vp(i+1,j)+(1*Vp(i+2,j+1)))+(1-n2-m2-l)*Vp(i+1,j+1));
    if(Q<=Vn(i+1,j+1)&&Q<T)
        Vn(i+1,j+1)=Q;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=1;
    elseif(T<=Vn(i+1,j+1)&&T<=Q)
        Vn(i+1,j+1)=T;
        Wn(i+1,j+1)=2;
        O(i+1,j+1)=0;
    end
end
end
end
Dif=(Vn-Vp);
Mn=max(max(Dif));
mn=min(min(Dif));
kk=kk+1
rep(kk)=Mn-mn;
Vp=Vn;
end
%Specify the changes points
for i=0:L1
    for j=0:L2
        if (Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<L2)
        B(i+1)=s+1;
    else
        B(i+1)=L2+100;
    end
end

```

```

end

for i=0:50
    for j=0:50

        if(Wn(i+1,j+1)<2)
            s=j;
        end
    end
    if(s<50)
        DE(i+1)=s+1;
    else
        DE(i+1)=L2+100;
    end
end
vd=0:50;
%check uniqueness point change
for i=0:50
    if(DE(i+1)<L2+100)
        for j=0:DE(i+1)-1
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
    if(DE(i+1)==L2+100)
        for j=0:50
            if(Wn(i+1,j+1)==2)
                A=[A; n3 m3 n4 m4 h3 h4 l4];
                %A=[A d];
            end
        end
    end
end
%check the monotonicity of points change
for i=0:50-1
    if(DE(i+1)>DE(i+2))
        C=[C;n3 m3 n4 m4 h3 h4 l4];
    end
end

```