



Πανεπιστήμιο Θεσσαλίας

Τμήμα: **Μηχανολόγων Μηχανικών**

Τομέας: **Οργάνωση, Παραγωγή και Βιομηχανική Διοίκηση**

**Βέλτιστη Χρήση Ευέλικτου Εξυπηρετητή σε
Συστήματα Ουρών Αναμονής με Δύο
Σταθμούς Εργασίας**

Διπλωματική Εργασία

Μπήτρος Παναγιώτης

ΑΜ:1349

Επιβλέπων: **Δρ. Παντελής Δημήτριος**

Βόλος, Ιούλιος 2017

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του
Διπλώματος Μηχανολόγου Μηχανικού

Copyright© 2017 Μπήτρος Παναγιώτης

Με επιφύλαξη παντός δικαιώματος. Allrightsreserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Πολυτεχνείου Θεσσαλίας.

Εξεταστική Επιτροπή

Πρώτος Εξεταστής
(Επιβλέπων)

Δρ. Δημήτριος Παντελής
Αναπληρωτής Καθηγητής,
Τμήμα Μηχανολόγων Μηχανικών
Πανεπιστήμιο Θεσσαλίας

Δεύτερος Εξεταστής

Δρ. Γεώργιος Λυμπερόπουλος
Καθηγητής,
Τμήμα Μηχανολόγων Μηχανικών
Πανεπιστήμιο Θεσσαλίας

Τρίτος Εξεταστής

Δρ. Γεώργιος Σαχαρίδης
Επίκουρος Καθηγητής,
Τμήμα Μηχανολόγων Μηχανικών
Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Πρώτα απ' όλα θα ήθελα να εκφράσω τις ευχαριστίες μου στον κ. Δημήτριο Παντελή, αναπληρωτή καθηγητή του Τμήματος Μηχανολόγων Μηχανικών του Πανεπιστημίου Θεσσαλίας, για την ευκαιρία που μου παρείχε να ασχοληθώ με το συγκεκριμένο θέμα, την καθοδήγηση του καθ' όλη τη διάρκεια εκπόνησης της διπλωματικής μου εργασίας και τη βοήθειά του σχετικά με την μετέπειτα σταδιοδρομία μου.

Επιπλέον, θα ήθελα να ευχαριστήσω τα υπόλοιπα μέλη της εξεταστικής επιτροπής, τον κ. Γεώργιο Λυμπερόπουλο και τον κ. Γεώργιο Σαχαρίδη για την προσεκτική ανάγνωση της διπλωματικής μου εργασίας.

Τέλος, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου για την υλική, αλλά και ηθική υποστήριξη σε όλη τη διάρκεια των σπουδών μου.

Περίληψη

Η παρούσα διπλωματική εργασία μελετά ένα πρόβλημα καταμερισμού εργατικού δυναμικού σε γραμμές παραγωγής. Συγκεκριμένα, με τη βοήθεια του υπολογιστικού προγράμματος Matlab, θα αναπτυχθεί ο αλγόριθμος με τον οποίο θα αποφασιστεί η βέλτιστη ανάθεση εργασιών σε εργάτες εκπαιδευμένους για διαφορετικούς ρόλους για διάφορα στάδια παραγωγής.

Η εργασία μας αναφέρεται κυρίως σε συστήματα αποτελούμενα από δύο σταθμούς εξυπηρέτησης σε σειρά. Ο κάθε σταθμός εξυπηρέτησης εξυπηρετεί με έναν εξειδικευμένο εξυπηρετητή, ενώ υπάρχει και ένας ευέλικτος εξυπηρετητής, είτε στον πρώτο σταθμό είτε στο δεύτερο, όπου μπορεί να παρέχει εξυπηρέτηση ανάλογα με τις ανάγκες εξυπηρέτησης του εκάστοτε σταθμού. Ο ευέλικτος εξυπηρετητής, όμως, εξυπηρετεί με έναν πιο αργό συνήθως ρυθμό εξυπηρέτησης από τον αντίστοιχο εξειδικευμένο.

Οι αφίξεις φθάνουν στον πρώτο σταθμό εξυπηρέτησης σύμφωνα με τη διαδικασία Poisson. Οι χρόνοι με τους οποίους εξυπηρετεί ο κάθε εξυπηρετητής στο σύστημα είναι εκθετικοί. Κάθε εργασία εξέρχεται από το σύστημα μόνο όταν έχει εξυπηρετηθεί και από το δεύτερο σταθμό. Επίσης, η επεξεργασία μιας εργασίας πρέπει να ολοκληρωθεί με τον εξυπηρετητή που άρχισε να διεκπεραιώνει την εργασία αυτή. Τέλος, δεν επιτρέπεται η συνεργασία μεταξύ εξειδικευμένου και ευέλικτου εξυπηρετητή σε κάθε σταθμό εξυπηρέτησης, προκειμένου να διεκπεραιώσουν μια κοινή εργασία.

Abstract

This diploma thesis studies a problem of workforce allocation in production lines. Specifically, an algorithm, using Matlab, will be developed to decide on the optimal assignment of cross-trained workforce for different production stages.

This dissertation is mainly about two-stage tandem queuing service systems. Each service station has a dedicated server. There is also a flexible server, either at the first station or the second, who can provide service according to the needs of the station. The flexible server, however, usually serves at a slower rate than the corresponding dedicated.

Arrivals arrive at the first service station according to the Poisson process. The processing times are exponential. A job may leave the system only when it is served by the second station. Additionally, preemptions are not allowed. Finally, collaboration between a dedicated and the flexible server to work on the same job is not allowed in each service station.

Περιεχόμενα

Εξεταστική Επιτροπή	iii
Ευχαριστίες	iv
Περίληψη	v
Abstract.....	vi
Κατάλογος Σχημάτων.....	- 2 -
Κεφάλαιο 1: Εισαγωγή	- 3 -
1.1 <i>Κίνητρο και Υπόβαθρο</i>	- 3 -
1.2 Βιβλιογραφική Ανασκόπηση.....	- 3 -
1.3 Υπολογιστικό πακέτο που χρησιμοποιήθηκε	- 4 -
1.4 Δομή Διπλωματικής εργασίας	- 5 -
Κεφάλαιο 2: Γνωστικό Υπόβαθρο	- 6 -
2.1 Συστήματα δύο σταθμών εργασίας με ευέλικτο εργατικό δυναμικό	- 6 -
2.2 Αποτελέσματα προηγούμενης έρευνας	- 7 -
Κεφάλαιο 3: Περιγραφή του προβλήματος	- 8 -
3.1 Ορισμός του προβλήματος	- 8 -
3.2 Διατύπωση εξίσωσης βελτιστοποίησης.....	- 8 -
Κεφάλαιο 4: Ανάλυση περιπτώσεων που μελετήθηκαν	- 11 -
4.1 Ένας σταθμός εργασίας με έναν εξειδικευμένο και έναν αργό εξυπηρετητή- 11 -	
4.2 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον δεύτερο σταθμό εργασίας	- 12 -
4.3 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον πρώτο σταθμό εργασίας	- 13 -
4.4 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και έναν αργό ευέλικτο εξυπηρετητή.....	- 14 -
Κεφάλαιο 5: Αριθμητικά Αποτελέσματα	- 15 -
5.1 Περιγραφή Αλγόριθμου.....	- 15 -
5.2 Αποτελέσματα για την περίπτωση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας.....	- 16 -
5.4 Αποτελέσματα για τη γενική περίπτωση.....	- 30 -
Κεφάλαιο 6: Σύνοψη Αποτελεσμάτων	- 37 -
Βιβλιογραφία	- 38 -
Παράρτημα	- 40 -

Κατάλογος Σχημάτων

Σχήμα 1 - Περιγραφή συστημάτων με δύο σταθμούς εργασίας	- 6 -
Σχήμα 2 - Πρώτη περίπτωση	- 11 -
Σχήμα 3 - Δεύτερη περίπτωση	- 12 -
Σχήμα 4 - Τρίτη Περίπτωση.....	- 13 -
Σχήμα 5 - Τέταρτη περίπτωση.....	- 14 -
Σχήμα 6 – Ευέλικτος στο δεύτερο σταθμό	- 16 -
Σχήμα 7 - Αλλαγή στο ρυθμό αφίξεων.....	- 18 -
Σχήμα 8 – Αλλαγή σε ρυθμό εξυπηρέτησης	- 19 -
Σχήμα 10 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 21 -
Σχήμα 11 – Αλλαγή σε κόστη παραμονής	- 22 -
Σχήμα 12 - Ευέλικτος στον πρώτο σταθμό	- 23 -
Σχήμα 13 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 25 -
Σχήμα 14 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 26 -
Σχήμα 15 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 27 -
Σχήμα 16 – Αλλαγή σε ρυθμό αφίξεων.....	- 28 -
Σχήμα 17 – Αλλαγή στα κόστη παραμονής	- 29 -
Σχήμα 18 – Ευέλικτος είτε στον πρώτο είτε στο δεύτερο σταθμό	- 30 -
Σχήμα 19 – Αλλαγή στα κόστη παραμονής	- 32 -
Σχήμα 20 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 33 -
Σχήμα 21 - Αλλαγή σε ρυθμό εξυπηρέτησης	- 34 -
Σχήμα 22 – Αλλαγή στα κόστη παραμονής	- 35 -
Σχήμα 23 - Αλλαγή στα κόστη παραμονής	- 36 -

Κεφάλαιο 1: Εισαγωγή

1.1 Κίνητρο και Υπόβαθρο

Η διπλωματική εργασία που ακολουθεί είχε ως κίνητρο την πραγματοποίηση της επίλυσης διάφορων προβλημάτων βελτιστοποίησης σε γραμμές παραγωγής με ευέλικτο εργατικό δυναμικό.

Το πρόβλημα που θα μελετήσουμε εντάσσεται στο ευρύτερο πεδίο της Επιχειρησιακής Έρευνας, δηλαδή της επιστήμης που ασχολείται με την βελτιστοποίηση της απόδοσης ενός συστήματος. Συγκεκριμένα, η εργασία αναφέρεται στη θεωρία ουρών. Θεωρείται ως ένας θεωρητικός τομέας της επιστήμης των μαθηματικών, παρόλα αυτά τα συστήματα ουρών αναμονής, καθώς και η μοντελοποίησή τους έχουν πολλές χρήσεις. Χρησιμοποιούνται σε βελτιστοποιήσεις από τις τηλεπικοινωνίες και τις μεταφορές, μέχρι τις τραπεζικές συναλλαγές και τις γραμμές παραγωγής.

1.2 Βιβλιογραφική Ανασκόπηση

Η συγκεκριμένη εργασία βασίστηκε στην εργασία **“On the Optimal Use of a Slow Server in Two-Stage Queuing Systems”** (Papachristos & Pandelis, 2016) η οποία περιέχει την περιγραφή, την ανάλυση καθώς και τις μαθηματικές σχέσεις για το πρόβλημα που θα ασχοληθούμε.

Ακόμα χρησιμοποιήθηκαν τα βιβλία **«Εισαγωγή στην Επιχειρησιακή Έρευνα»** (Hamdy A.Taha, 2012), (N. Δ. Τσάντας & Π. Γ. Βασιλείου, 2000) **“Productions & Operations Analysis”** (Nahmias, 2013) και **“Introduction to Operations Research”** (Frederick S. Hillier & Gerald J. Lieberman, 2010), κυρίως για το θεωρητικό υπόβαθρο της εργασίας. Εκεί μπορεί να ανατρέξει ο αναγνώστης για περισσότερες πληροφορίες για θέματα όπως μοντέλα επιχειρησιακής έρευνας, στοχαστικά πρότυπα, ουρές αναμονής, μαρκοβιανές αλυσίδες και ό,τι άλλο σχετίζεται με την Επιχειρησιακή Έρευνα.

Όσον αφορά την αλληλεπίδραση με το MATLAB, εξαιρετικά χρήσιμο αποδείχτηκε το βιβλίο **“MATLAB 6 για μηχανικούς”** (Adrian Biran & Moshe Breiner, 2012). Τέλος, πολύ σημαντικές ήταν οι πληροφορίες που αντλήθηκαν από την επίσημη ιστοσελίδα της εταιρίας που αναπτύσσει το MATLAB (www.mathworks.com).

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, επεξεργασίας σημάτων, αριθμητικές λύσεις κανονικών διαφορικών εξισώσεων. Οι περισσότερες από τις προαναφερόμενες συναρτήσεις εφαρμόζουν την πλέον πρόσφατη και εξελιγμένη γνώση στο κάθε τομέα επιστήμης. 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.

1.4 Δομή Διπλωματικής εργασίας

Η δομή της διπλωματικής εργασίας έχει ως εξής:

Στο **Κεφάλαιο 2** παρουσιάζεται η θεωρητική οπτική γωνία της παρούσας διπλωματικής εργασίας.

Στο **Κεφάλαιο 3** γίνεται μια αναλυτική περιγραφή του προβλήματος, παρουσιάζεται το μαθηματικό μοντέλο, τα βασικά βήματα επίλυσης καθώς επίσης και οι μεταβλητές που το διέπουν.

Στο **Κεφάλαιο 4** περιγράφονται λεπτομερώς οι περιπτώσεις που μελετήθηκαν.

Στο **Κεφάλαιο 5** παρουσιάζονται τα αριθμητικά αποτελέσματα που προέκυψαν μαζί με τα συμπεράσματα της διπλωματικής εργασίας και τον σχολιασμό τους.

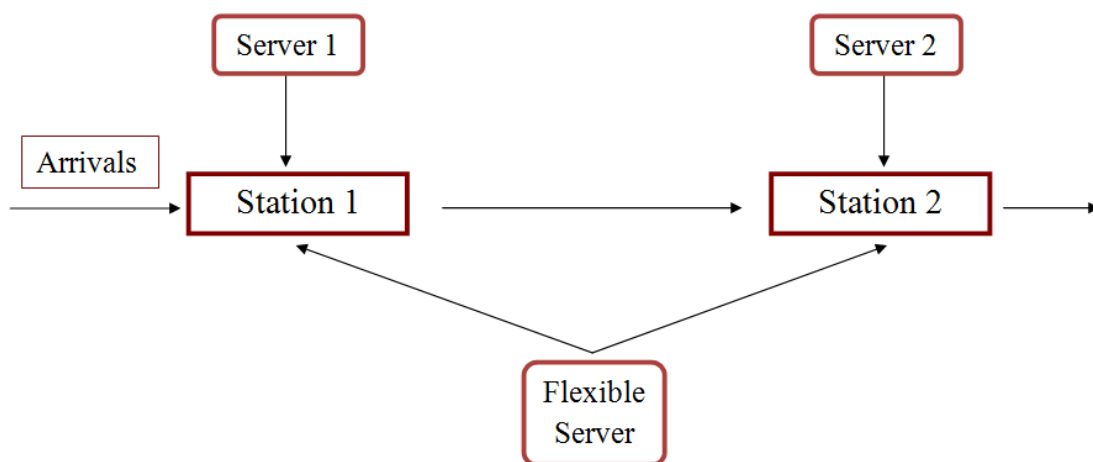
Τέλος, στο **Κεφάλαιο 6** έχουμε τη σύνοψη των αποτελεσμάτων και τα τελικά συμπεράσματα.

Κεφάλαιο 2: Γνωστικό Υπόβαθρο

Σε αυτό το κεφάλαιο, θα καταγραφούν πληροφορίες ενημερωτικού χαρακτήρα. Στόχος είναι να παρουσιαστεί το θεωρητικό πλαίσιο γύρω από θέματα στα οποία βασίζεται η διπλωματική εργασία.

2.1 Συστήματα δύο σταθμών εργασίας με ευέλικτο εργατικό δυναμικό

Όπως φαίνεται και στο παρακάτω σχήμα, οι αφίξεις στο σύστημα γίνονται σύμφωνα με τη διαδικασία Poisson στον πρώτο σταθμό εργασίας, όπου έχουμε τον πρώτο εξειδικευμένο εξυπηρετητή και μετά από επιτυχή εξυπηρέτηση περνούν στον δεύτερο σταθμό εξυπηρέτησης, όπου έχουμε το δεύτερο εξειδικευμένο εξυπηρετητή. Όταν έχουμε εξυπηρέτηση και από το δεύτερο σταθμό, τότε η εργασία θεωρείται ολοκληρωμένη και εξέρχεται από το σύστημα. Ο ευέλικτος εξυπηρετητής μπορεί να βρίσκεται είτε στον πρώτο είτε στο δεύτερο σταθμό εξυπηρέτησης, ανάλογα με τις ανάγκες του εκάστοτε προβλήματος που μελετάμε.



Σχήμα 1 - Περιγραφή συστημάτων με δύο σταθμούς εργασίας

2.2 Αποτελέσματα προηγούμενης έρευνας

Το μαθηματικό μοντέλο που αναλύουμε βρίσκει εφαρμογή σε συστήματα που χρησιμοποιούν ευέλικτους πόρους για την υποστήριξη της παραγωγικής διαδικασίας. Εκτεταμένη αναφορά σε ερευνητικές εργασίες σχετικά με την ευελιξία του εργατικού δυναμικού πραγματοποιήθηκε από τους (Andradottir S. & Ayhan H. & Down D. G., 2013) και από τους (Hopp W. J. & Van Oyen M. P., 2004).

Επίσης, για συστήματα σαν αυτό που μελετά η παρούσα διπλωματική εργασία (δηλαδή συστήματα στα οποία η επεξεργασία μιας εργασίας πρέπει να ολοκληρωθεί με τον εξυπηρετητή που άρχισε να διεκπεραιώνει την εργασία αυτή και που δεν επιτρέπεται η συνεργασία μεταξύ εξειδικευμένου και ευέλικτου εξυπηρετητή σε κάθε σταθμό εξυπηρέτησης, προκειμένου να διεκπεραιώσουν μια κοινή εργασία), αλλά στην απλουστευμένη τους μορφή με έναν σταθμό εργασίας, έχει δειχθεί από διάφορους ερευνητές και με διαφορετικούς τρόπους ότι η βέλτιστη πολιτική είναι να μην απενεργοποιείται ποτέ ο γρήγορος εξυπηρετητής και να ενεργοποιείται ο αργός όταν οι εργασίες ξεπεράσουν έναν συγκεκριμένο αριθμό. (Lin & Kumar, 1984), (Walrand, 1984), (Kooile, 1995)

Έχει, ακόμη, αποδειχθεί για το μοντέλο του ενός σταθμού εργασίας, η βέλτιστη πολιτική για διάφορες επεκτάσεις των συστημάτων αυτών, στα οποία ο σταθμός εργασίας δεν χαρακτηρίζεται μόνο από ένα κόστος παραμονής, αλλά υπάρχουν και λειτουργικά κόστη (Akgun, 2014), καθώς και για συστήματα στα οποία οι εξυπηρετητές μπορεί να βρίσκονται σε κατάσταση βλάβης, δηλαδή δεν είναι αξιόπιστοι (Ozkan & Kharoufeh, 2014).

Τέλος, στη διπλωματική αυτή εργασία ασχολούμαστε με το μοντέλο στο οποίο υπάρχουν δύο σταθμοί εργασίας και στόχο έχουμε να παρατηρήσουμε διάφορες ιδιότητες της βέλτιστης πολιτικής, όσον αφορά την αδρανοποίηση ή μη του ευέλικτου εξυπηρετητή, συγκριτικά με τους παράγοντες που μπορούν να την επηρεάσουν.

Κεφάλαιο 3: Περιγραφή του προβλήματος

3.1 Ορισμός του προβλήματος

Στους χρόνους των αφίξεων και αναχωρήσεων επιτρέπουμε να γίνεται επανατοποθέτηση των εξυπηρετητών και στη συνέχεια διατυπώνουμε το πρόβλημά μας ως μια Μαρκοβιανή Διαδικασία με χώρο καταστάσεων (x_1, i_1, j, x_2, i_2) .

- x_k : εργασίες που περιμένουν στον σταθμό k
- i_k : κατάσταση του γρήγορου εξυπηρετητή στο σταθμό k (0:idle, 1:busy)
- j : κατάσταση του αργού εξυπηρετητή (0:idle, 1,2: working)

Αντί του προβλήματος συνεχούς χρόνου, θεωρούμε το ισοδύναμο πρόβλημα διακριτού χρόνου, δηλαδή έχουμε ομοιομορφοποίηση (uniformization):

$$\lambda + \nu_1 + \nu_2 + \mu_1 + \mu_2 = 1, \text{ όπου:}$$

- λ : ρυθμός αφίξεων σύμφωνα με τη διαδικασία Poisson
- ν_k : ρυθμός εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στο σταθμό k
- μ_k : ρυθμός εξυπηρέτησης του ευέλικτου εξυπηρετητή στο σταθμό k

Το πρόβλημα αυτό χαρακτηρίζεται από το ελάχιστο προσδοκώμενο κόστος $V_n(x_1, i_1, j, x_2, i_2)$ για n χρονικές περιόδους. Το κόστος αυτό είναι μειωμένο κατά έναν εκπτώτικό συντελεστή β .

3.2 Διατύπωση εξίσωσης βελτιστοποίησης

Η συνάρτηση του ελάχιστου κόστους $V_n(x_1, i_1, j, x_2, i_2)$ διέπεται από την ακόλουθη εξίσωση δυναμικού προγραμματισμού:

$$V_n(x_1, i_1, j, x_2, i_2) = \min_{\alpha_1, \alpha_2, \alpha_j} W_n(y_1, \alpha_1, \alpha_j, y_2, \alpha_2)$$

- $\alpha_1, \alpha_2, \alpha_j$: εφικτές τοποθετήσεις εξυπηρετητών
- y_k : αριθμός εργασιών που περιμένουν στον σταθμό k ως αποτέλεσμα των τοποθετήσεων $\alpha_1, \alpha_2, \alpha_j$

Οι εργασίες αυτές προκύπτουν ως εξής:

$$y_k = x_k - a_k(1 - i_k) - \mathbf{1}(j = 0, \alpha_j = k)$$

Μετά τη λήψη των αποφάσεων, το προσδοκώμενο κόστος για την περίπτωση που ο ευέλικτος εξυπηρετητής βρίσκεται στον πρώτο σταθμό εργασίας ($j = 1$), δίνεται από τον ακόλουθο τύπο:

$$\begin{aligned} W_n(x_1, i_1, j, x_2, i_2) = & h_1(x_1 + i_1 + 1) + h_2(x_2 + i_2) \\ & + \beta[\lambda V_{n-1}(x_1 + 1, i_1, 1, x_2, i_2) \\ & + v_1 V_{n-1}(x_1, 0, 1, x_2 + i_1, i_2) \\ & + v_2 V_{n-1}(x_1, i_1, 1, x_2, 0) \\ & + \mu_2 V_{n-1}(x_1, i_1, 1, x_2, i_2) \\ & + \mu_1 V_{n-1}(x_1, i_1, 0, x_2 + 1, i_2)] \end{aligned}$$

Αντίστοιχα το προσδοκώμενο κόστος για την περίπτωση που ο ευέλικτος εξυπηρετητής βρίσκεται στο δεύτερο σταθμό εργασίας ($j = 2$):

$$\begin{aligned} W_n(x_1, i_1, j, x_2, i_2) = & h_1(x_1 + i_1) + h_2(x_2 + i_2 + 1) \\ & + \beta[\lambda V_{n-1}(x_1 + 1, i_1, 2, x_2, i_2) \\ & + v_1 V_{n-1}(x_1, 0, 2, x_2 + i_1, i_2) \\ & + v_2 V_{n-1}(x_1, i_1, 2, x_2, 0) \\ & + \mu_2 V_{n-1}(x_1, i_1, 0, x_2, i_2) \\ & + \mu_1 V_{n-1}(x_1, i_1, 2, x_2, i_2)] \end{aligned}$$

Τέλος, το προσδοκώμενο κόστος για την περίπτωση που ο ευέλικτος εξυπηρετητής είναι απενεργοποιημένος ($j = 0$):

$$\begin{aligned}W_n(x_1, i_1, j, x_2, i_2) &= h_1(x_1 + i_1) + h_2(x_2 + i_2) \\ &+ \beta[\lambda V_{n-1}(x_1 + 1, i_1, 0, x_2, i_2) \\ &+ \nu_1 V_{n-1}(x_1, 0, 0, x_2 + i_1, i_2) \\ &+ \nu_2 V_{n-1}(x_1, i_1, 0, x_2, 0) \\ &+ \mu_2 V_{n-1}(x_1, i_1, 0, x_2, i_2) \\ &+ \mu_1 V_{n-1}(x_1, i_1, 0, x_2, i_2)]\end{aligned}$$

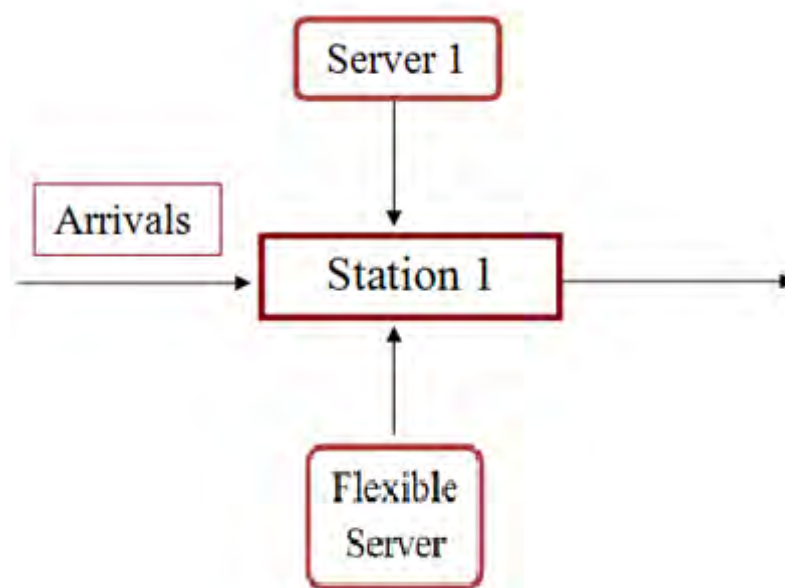
και με αρχική τιμή $W_0(x_1, i_1, j, x_2, i_2) = 0$ για κάθε περίπτωση.

Κεφάλαιο 4: Ανάλυση περιπτώσεων που μελετήθηκαν

Μελετήσαμε τις ακόλουθες περιπτώσεις για γραμμές παραγωγής:

- Ένας σταθμός εργασίας με έναν εξειδικευμένο και έναν αργό εξυπηρετητή
- Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον δεύτερο σταθμό εργασίας
- Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον πρώτο σταθμό εργασίας
- Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό ευέλικτο εξυπηρετητή να έχει τη δυνατότητα να βρίσκεται είτε στον πρώτο είτε στο δεύτερο σταθμό εργασίας

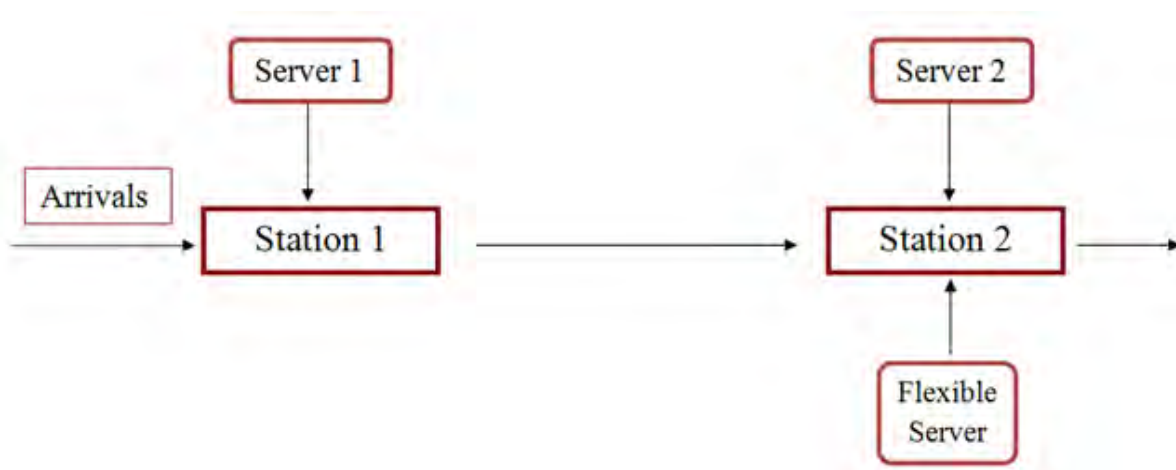
4.1 Ένας σταθμός εργασίας με έναν εξειδικευμένο και έναν αργό εξυπηρετητή



Σχήμα 2 - Πρώτη περίπτωση

Αυτή είναι η πιο απλή περίπτωση που αναλύθηκε. Αυτό έγινε κυρίως για λόγους σύγκρισης με κάποιες από τις περιπτώσεις. Στην περίπτωση αυτή δεν έχουμε δεύτερο σταθμό εργασίας. Υπάρχει μόνο ένας σταθμός με έναν εξειδικευμένο εξυπηρετητή. Στον ίδιο σταθμό υπάρχει και ένας πιο αργός εξυπηρετητής. Όταν μια άφιξη στο σταθμό αυτό εξυπηρετηθεί είτε από τον εξειδικευμένο είτε από τον πιο αργό εξυπηρετητή, ο οποίος έχει το δικαίωμα να εξυπηρετεί μία ή καμία εργασία τη φορά, τότε εξέρχεται από το σύστημα.

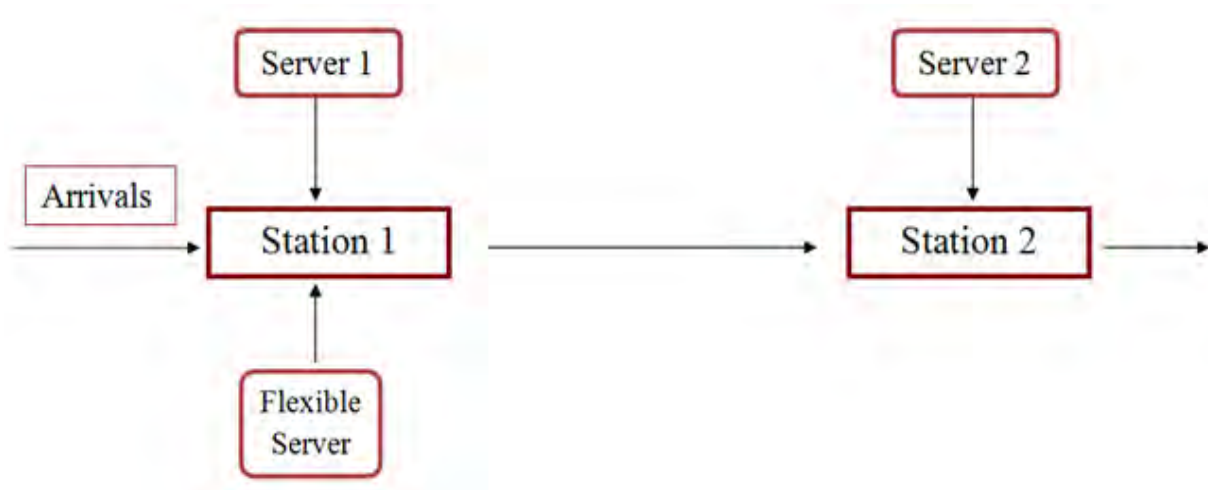
4.2 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον δεύτερο σταθμό εργασίας



Σχήμα 3 - Δεύτερη περίπτωση

Στο σενάριο αυτό έχουμε δύο σταθμούς εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος. Στην περίπτωση αυτή θεωρούμε πως ο ευέλικτος και πιο αργός εξυπηρετητής βρίσκεται στον δεύτερο σταθμό εργασίας. Όπως φαίνεται και στο παραπάνω σχήμα, οι αφίξεις γίνονται στον πρώτο σταθμό εργασίας, όπου εξυπηρετούνται από τον εξειδικευμένο. Στη συνέχεια οι εργασίες μεταφέρονται στο δεύτερο σταθμό, όπου μπορούν να εξυπηρετηθούν από τον αντίστοιχο εξειδικευμένο. Εκεί υπάρχει όμως και ο πιο αργός εξυπηρετητής, ο οποίος έχει το δικαίωμα να εξυπηρετεί μία ή καμία εργασία τη φορά παράλληλα με τον εξειδικευμένο, έτσι ώστε οι εργασίες να εξέλθουν τελικά από το σύστημα.

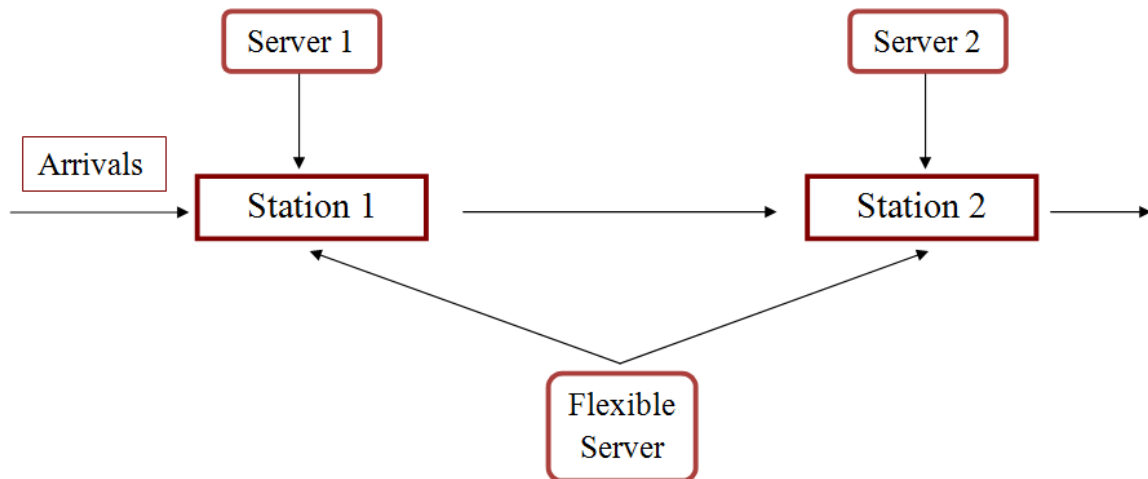
4.3 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και τον αργό εξυπηρετητή στον πρώτο σταθμό εργασίας



Σχήμα 4 - Τρίτη Περίπτωση

Σε αυτό το σενάριο έχουμε ξανά δύο σταθμούς εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος. Στην περίπτωση αυτή θεωρούμε πως ο ευέλικτος και πιο αργός εξυπηρετητής βρίσκεται στον πρώτο σταθμό εργασίας. Όπως φαίνεται και στο παραπάνω σχήμα, οι αφίξεις γίνονται στον πρώτο σταθμό εργασίας, όπου εξυπηρετούνται από τον εξειδικευμένο. Εκεί υπάρχει όμως και ο πιο αργός εξυπηρετητής, ο οποίος έχει το δικαίωμα να εξυπηρετεί μία ή καμία εργασία τη φορά, παράλληλα με τον εξειδικευμένο. Στη συνέχεια οι εργασίες μεταφέρονται στο δεύτερο σταθμό, όπου μπορούν να εξυπηρετηθούν από τον αντίστοιχο εξειδικευμένο και να εξέλθουν τελικά από το σύστημα.

4.4 Δύο σταθμοί εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος και έναν αργό ευέλικτο εξυπηρετητή



Σχήμα 5 - Τέταρτη περίπτωση

Το παραπάνω σενάριο είναι και το πιο γενικό. Έχουμε ξανά δύο σταθμούς εργασίας με έναν εξειδικευμένο εξυπηρετητή έκαστος. Στην περίπτωση αυτή θεωρούμε πως ο ευέλικτος και πιο αργός εξυπηρετητής έχει τη δυνατότητα να παρέχει εξυπηρέτηση είτε στον πρώτο είτε στο δεύτερο σταθμό με βάση τις ανάγκες του κάθε σταθμού. Όπως και στις προηγούμενες περιπτώσεις, οι αφίξεις γίνονται στον πρώτο σταθμό εργασίας και αφού εξυπηρετηθούν και από το δεύτερο σταθμό, εξέρχονται από το σύστημα.

Κεφάλαιο 5: Αριθμητικά Αποτελέσματα

Το μέσο κόστος είναι πεπερασμένο όταν το σύστημά μας είναι ευσταθές. Για να συμβεί αυτό, θα πρέπει οι τιμές των μεταβλητών να ακολουθούν κάθε φορά τους περιορισμούς της εκάστοτε περίπτωσης. Γενικότερα, κάθε φορά θα πρέπει ο ρυθμός αφίξεων λ να είναι μικρότερος από το ρυθμό εξυπηρέτησης σε κάθε ουρά.

5.1 Περιγραφή Αλγόριθμου

Για την επίλυση κάθε περίπτωσης χρησιμοποιούμε τον αλγόριθμο Value-iteration για το κριτήριο μέσου κόστους. Για την εφαρμογή του αλγορίθμου θεωρήσαμε μέγιστο αριθμό εργασιών 100 σε κάθε σταθμό. Ο αλγόριθμος έχει ως εξής:

Βήμα 1^ο: Διαλέγουμε ως αρχική κατάσταση την $V_0(x_1, i_1, j, x_2, i_2) = 0$.

Θέτουμε $n = 1$.

Βήμα 2^ο: Υπολογίζουμε το συνολικό ελάχιστο κόστος για n περιόδους, χρησιμοποιώντας τις εξισώσεις που περιγράφηκαν στο Κεφάλαιο 3.2.

Βήμα 3^ο: Υπολογίζουμε τα όρια:

$$\text{Κάτω όριο: } m_n = \min\{V_n(x_1, i_1, j, x_2, i_2) - V_{n-1}(x_1, i_1, j, x_2, i_2)\}$$

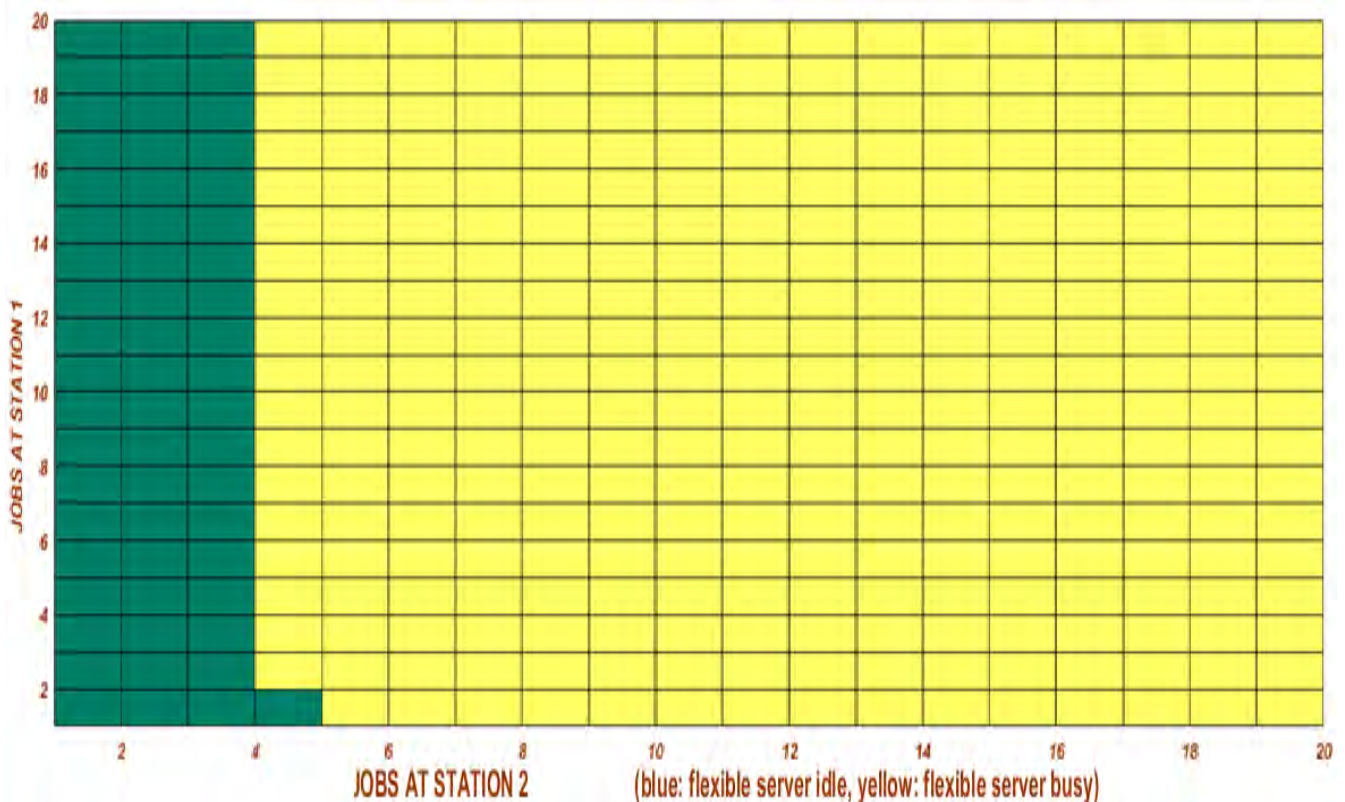
$$\text{Άνω όριο: } M_n = \max\{V_n(x_1, i_1, j, x_2, i_2) - V_{n-1}(x_1, i_1, j, x_2, i_2)\}$$

Βήμα 4^ο: Αν η διαφορά των ορίων που υπολογίστηκαν στο Βήμα 3^ο πάρει τιμή της τάξης 10^{-3} , τότε ο αλγόριθμος τερματίζει. Διαφορετικά, θέτουμε $n = n + 1$ και πηγαίνουμε πάλι στο Βήμα 2^ο.

Μετά τον τερματισμό του αλγορίθμου προκύπτει ένας πίνακας. Γενικότερα, η μορφή των αποτελεσμάτων σε κάθε περίπτωση είναι σε μορφή πίνακα, ο οποίος απεικονίζει τη βέλτιστη χρήση του ευέλικτου εξυπηρετητή. Ο πίνακας αυτός θα περιέχει μόνο τις τιμές 0, 1, 2. Με την τιμή 0 μπορούμε να κατανοήσουμε πως δεν χρησιμοποιείται ο ευέλικτος εξυπηρετητής. Με τις τιμές 1 και 2 δηλώνεται πως συμφέρει ο ευέλικτος εξυπηρετητής να τοποθετείται στον πρώτο ή αντίστοιχα στο δεύτερο σταθμό εργασίας, αναλόγως το πρόβλημα το οποίο μελετάμε.

5.2 Αποτελέσματα για την περίπτωση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας

Στην περίπτωση αυτή θα δούμε το γράφημα που προκύπτει για την περίπτωση δύο σταθμών εργασίας, με τον ευέλικτο εξυπηρετητή να βρίσκεται στο δεύτερο σταθμό. Το γράφημα αυτό προέκυψε χρησιμοποιώντας τις παραμέτρους $\lambda = 0,04$ για το ρυθμό αφίξεων, $\nu_1 = 0,27$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στον πρώτο σταθμό εργασίας, $\nu_2 = 0,35$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στο δεύτερο σταθμό εργασίας και $\mu_2 = 0,1$ για τον ευέλικτο εξυπηρετητή, ο οποίος βρίσκεται στο δεύτερο σταθμό εξυπηρέτησης. Τέλος, κόστος παραμονής στον πρώτο σταθμό $h_1 = 0,37$ και $h_2 = 0,32$, για το δεύτερο σταθμό.



Σχήμα 6 – Ευέλικτος στο δεύτερο σταθμό

Αρχικά παρατηρούμε πως δεν έχουμε ενεργοποίηση του ευέλικτου εξυπηρετητή, πράγμα που σημαίνει πως για μικρό αριθμό εργασιών στη δεύτερη ουρά αναμονής, συμφέρει οι εργασίες να διεκπεραιώνονται αποκλειστικά από τον εξειδικευμένο εξυπηρετητή του δεύτερου σταθμού. Στη συνέχεια, όμως όσο ο αριθμός αυτός αυξάνεται, είναι ξεκάθαρο πως η βέλτιστη πολιτική είναι η ενεργοποίηση του ευέλικτου εξυπηρετητή.

Εδώ αξίζει να γίνει μια σύγκριση της περίπτωσης αυτής με την περίπτωση του ενός σταθμού εργασίας με έναν εξειδικευμένο εξυπηρετητή και έναν ευέλικτο.

Τρέξαμε, λοιπόν, τον κώδικα Matlab για ένα σταθμό εργασίας, χρησιμοποιώντας τις παραμέτρους $\lambda = 0,27$ για το ρυθμό αφίξεων, $\nu = 0,35$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή και $\mu = 0,1$ για τον ευέλικτο εξυπηρετητή, ο οποίος βρίσκεται στο σταθμό αυτό.

Παρατηρήσαμε πως το βέλτιστο είναι να γίνεται χρήση του ευέλικτου εξυπηρετητή από τη στιγμή που θα υπάρχουν 3 εργασίες στην ουρά αναμονής.

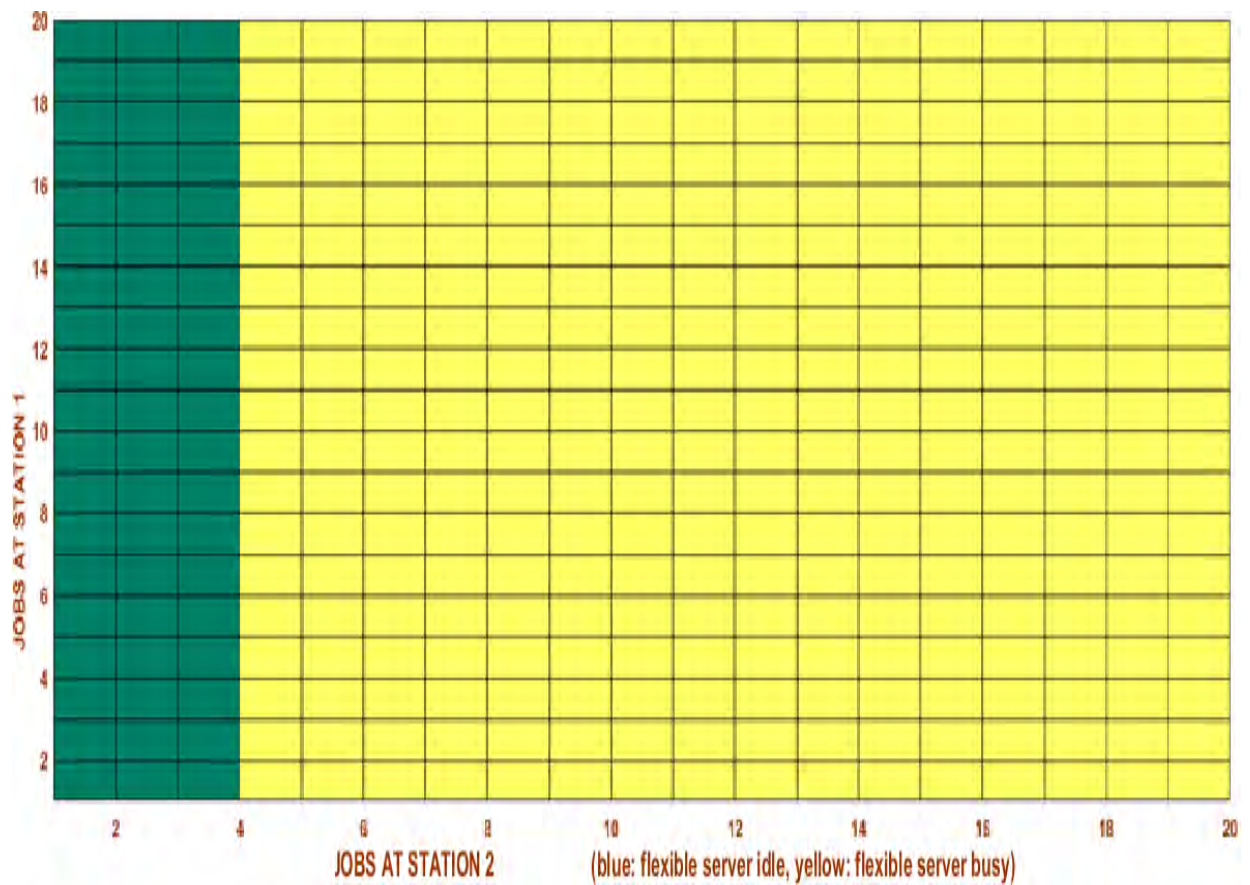
Δηλαδή, η βέλτιστη πολιτική είναι ίδιας φύσεως με την περίπτωση των δύο σταθμών εργασίας, όπου ο ευέλικτος εξυπηρετητής βρίσκεται στο δεύτερο σταθμό.

Αυτό είναι απόλυτα λογικό, εφόσον, ο ευέλικτος βρίσκεται στο δεύτερο σταθμό εργασίας και θα μπορούσε να θεωρηθεί ως ένα ανεξάρτητο σύστημα με ρυθμό αφίξεων ν_1 και έπειτα από τη διεκπεραίωση μιας εργασίας είτε από τον ευέλικτο είτε από τον σταθερό εξυπηρετητή εξέρχεται από το σύστημα.

Στη συνέχεια θα μελετήσουμε πιθανές αλλαγές που μπορούν να προκύψουν στο γράφημα με μεταβολές στις αρχικές τιμές που χρησιμοποιήθηκαν στον κώδικα, δηλαδή αυξομειώσεις είτε στο ρυθμό αυξήσεων, είτε στους ρυθμούς εξυπηρέτησεων, είτε στα κόστη παραμονής στον κάθε σταθμό.

Αρχικά ενδιαφερόμαστε να εξετάσουμε τι θα προκύψει στο σχήμα με μια αλλαγή στο ρυθμό αυξήσεων του συστήματος λ . Συγκεκριμένα, θα αυξήσουμε την τιμή αυτή από 0,04 σε 0,44.

Διατηρώντας, λοιπόν, όλους τους ρυθμούς εξυπηρέτησεων των εξυπηρετητών του συστήματος και τα κόστη παραμονής σταθερά, έχουμε:



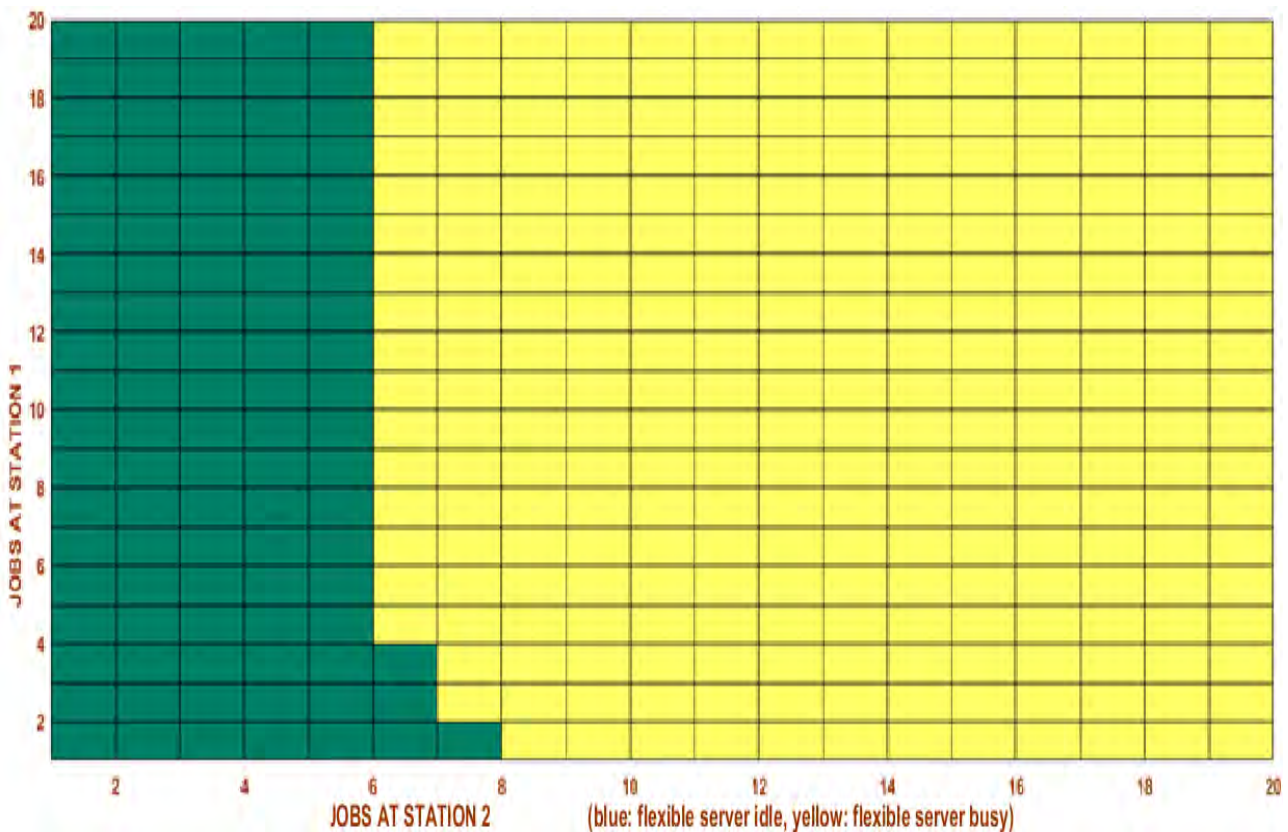
Σχήμα 7 - Αλλαγή στο ρυθμό αφίξεων

Μπορούμε να κατανοήσουμε πως η αλλαγή αυτή στο ρυθμό αφίξεων δεν επηρεάζει ιδιαίτερα το σχήμα. Ο ευέλικτος τίθεται σε λειτουργία για τον ίδιο αριθμό εργασιών στην ουρά αναμονής, όπως και προηγουμένως.

Το σύστημά μας επομένως δεν φαίνεται να εξαρτάται τόσο από το ρυθμό αφίξεων (Poisson).

Στη συνέχεια θα δούμε πως το σύστημά μας επηρεάζεται από την μεταβολή της τιμής του εξειδικευμένου εξυπηρετητή του δεύτερου σταθμού εργασίας, με μια αύξηση της τιμής v_2 από 0,35 σε 0,65.

Διατηρώντας, λοιπόν, τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



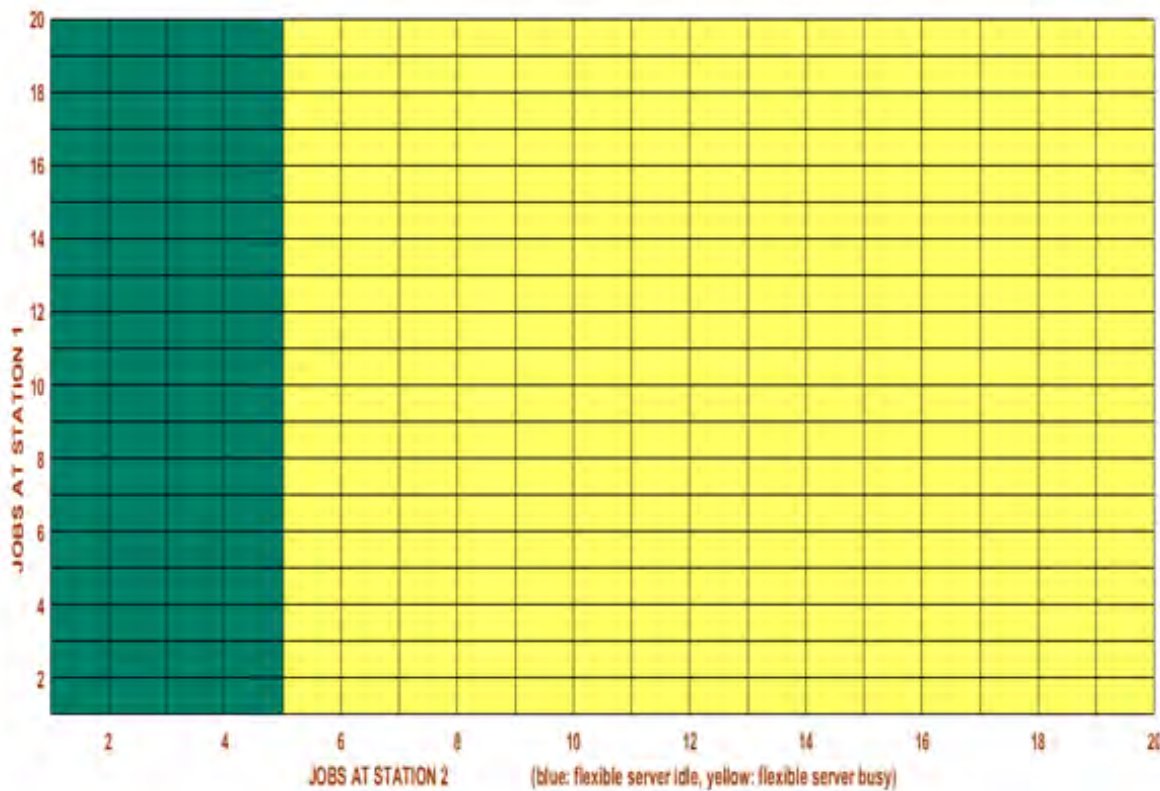
Σχήμα 8 – Αλλαγή σε ρυθμό εξυπηρέτησης

Παρατηρώντας το σχήμα μπορεί κανείς να καταλάβει ότι ο ευέλικτος εξυπηρετητής παραμένει ανενεργός για μεγαλύτερο αριθμό εργασιών στην ουρά αναμονής του δεύτερου σταθμού.

Το αποτέλεσμα αυτό είναι απόλυτα κατανοητό, εφόσον ο σταθερός εξυπηρετητής στο σταθμό αυτό εξυπηρετεί πλέον με πιο γρήγορο ρυθμό, χωρίς να υπάρχουν απαιτήσεις για περισσότερη εξυπηρέτηση.

Μπορούμε να δούμε πώς θα επηρεαστεί το αρχικό σχήμα μειώνοντας τον ρυθμό εξυπηρέτησης ν_1 του εξειδικευμένου εξυπηρετητή στον πρώτο σταθμό εργασίας από 0,27 σε 0,07.

Διατηρώντας τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



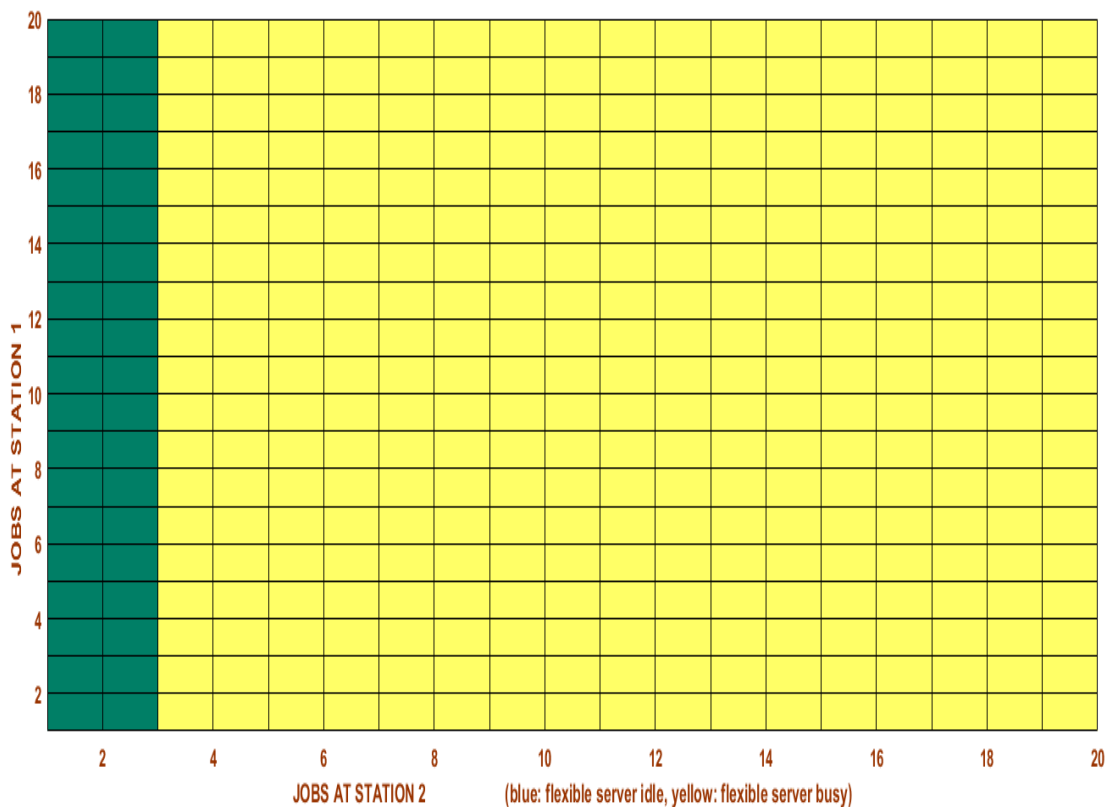
Σχήμα 9 – Αλλαγή σε ρυθμό εξυπηρέτησης

Αλλάζοντας, λοιπόν, το ρυθμό εξυπηρέτησης ν_1 , δεν παρατηρείται σημαντική αλλαγή στο σχήμα. Ο ευέλικτος, ο οποίος σε αυτή την περίπτωση βρίσκεται στο δεύτερο σταθμό εργασίας, ενεργοποιείται για ελαφρώς μεγαλύτερο αριθμό εργασιών στη δεύτερη ουρά αναμονής συγκριτικά με την αρχική περίπτωση.

Μην ξεχνάμε πως από τις αρχικές μας τιμές ισχύει $h_1 \geq h_2$, επομένως ο πρώτος σταθμός είναι αυτός που έχει περισσότερες απαιτήσεις εξυπηρέτησης και όχι ο δεύτερος, πράγμα που δίνει τη δυνατότητα στο ευέλικτο εξυπηρετητή να ενεργοποιηθεί για λίγο μεγαλύτερο αριθμό εργασιών,.

Σε αυτό το βήμα, ενδιαφερόμαστε να εξετάσουμε τι θα προκύψει στο σχήμα με μια αλλαγή στο ρυθμό εξυπηρέτησης του ευέλικτου, ο οποίος σε αυτή την περίπτωση βρίσκεται στο δεύτερο σταθμό εργασίας. Συγκεκριμένα, θα αυξήσουμε την τιμή του μ_2 από 0,1 σε 0,4.

Διατηρώντας, λοιπόν, τους λοιπούς ρυθμούς εξυπηρέτησεων των εξυπηρετητών του συστήματος και τα κόστη παραμονής σταθερά, έχουμε:



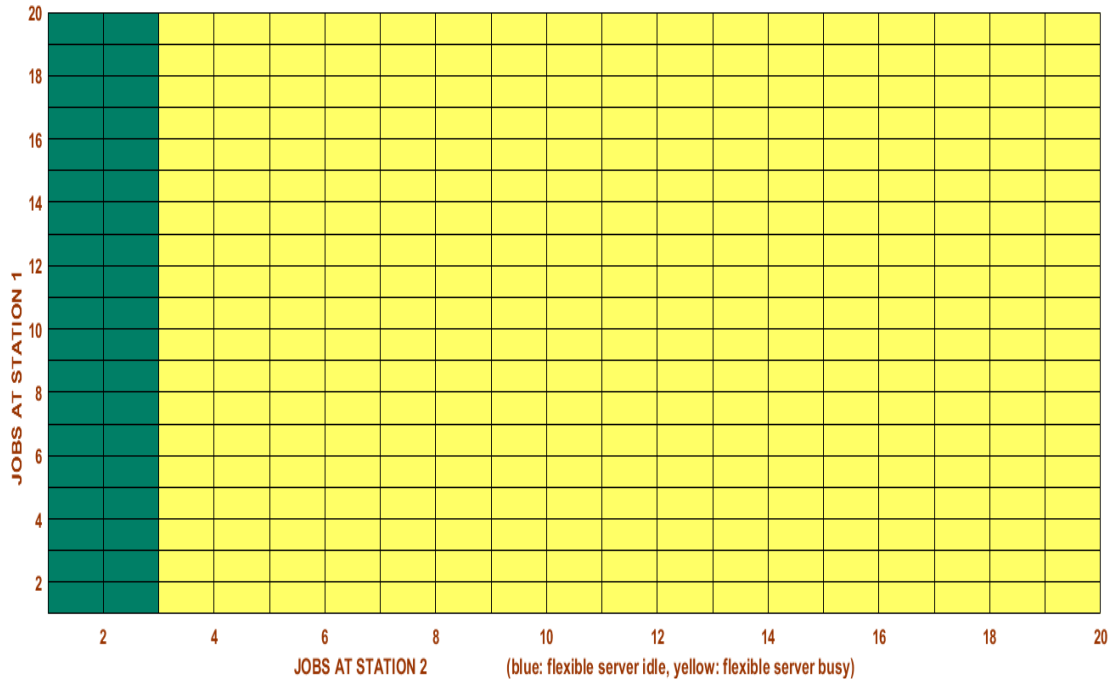
Σχήμα 9 - Αλλαγή σε ρυθμό εξυπηρέτησης

Παρατηρώντας κανείς το σχήμα μπορεί να κατανοήσει πως η μπλε περιοχή μειώθηκε σε σημαντικό βαθμό. Αυτό σημαίνει πως ο ευέλικτος εξυπηρετητής, που βρίσκεται στο δεύτερο σταθμό εργασίας, τέθηκε σε λειτουργία για μικρότερο αριθμό εργασιών στην ουρά αναμονής του δεύτερου σταθμού αυτή τη φορά.

Αυτό γίνεται διότι, αυξάνοντας το ρυθμό εξυπηρέτησης μ_2 , το βέλτιστο είναι ο ευέλικτος να ενεργοποιηθεί για λιγότερες εργασίες, έτσι ώστε να ικανοποιήσει τις ανάγκες για εξυπηρέτηση που υπάρχουν στον δεύτερο σταθμό.

Στις αρχικές τιμές που δώσαμε στον κώδικα, είχαμε υποθέσει ότι $h_1 \geq h_2$. Ενδιαφέρον έχει να δούμε το μετασχηματισμό του γραφήματος για $h_2 \geq h_1$ και συγκεκριμένα για $h_1=0,32$ και $h_2=0,37$.

Διατηρώντας, λοιπόν, το ρυθμό αφίξεων και όλους τους ρυθμούς εξυπηρέτησης σταθερούς, προκύπτει:



Σχήμα 10 – Αλλαγή σε κόστη παραμονής

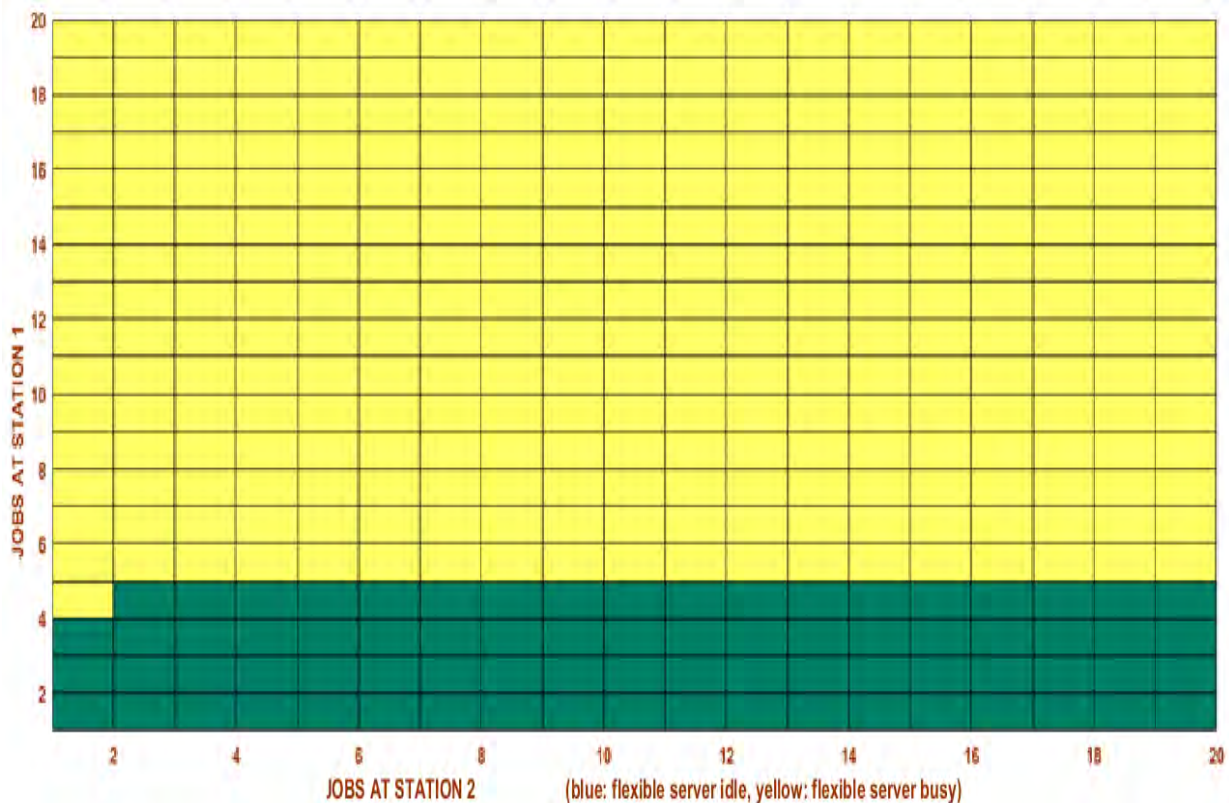
Βρισκόμαστε στην περίπτωση, όπου ο ευέλικτος βρίσκεται στο δεύτερο σταθμό εργασίας, πράγμα που καθιστά λογικό το γράφημα που προέκυψε.

Το να αλλάξει η συνθήκη του κόστους παραμονής σε $h_2 \geq h_1$, οδηγεί σε επιπλέον ανάγκες εξυπηρέτησης στο δεύτερο σταθμό. Επομένως, αυτό κάνει τον ευέλικτο εξυπηρετητή που υπάρχει στο σταθμό αυτό να ενεργοποιηθεί για αρκετά μικρότερο αριθμό εργασιών σε αναμονή στην ουρά του δεύτερου σταθμού.

Γενικότερα, αξίζει να σημειωθεί ότι στην περίπτωση όπου $h_2 > h_1$, πιθανό είναι στη βέλτιστη πολιτική να συμφέρει να απενεργοποιείται και ο σταθερός εξυπηρετητής του πρώτου σταθμού. Στην παρούσα, όμως, διπλωματική έχει υποτεθεί ότι μόνο ο ευέλικτος μπορεί να απενεργοποιείται και όχι οι εξειδικευμένοι εξυπηρετητές.

5.3 Αποτελέσματα για την περίπτωση του ευέλικτου εξυπηρετητή στον πρώτο σταθμό εργασίας

Στην περίπτωση αυτή θα δούμε το γράφημα που προκύπτει για την περίπτωση δύο σταθμών εργασίας, με τον ευέλικτο εξυπηρετητή να βρίσκεται πρώτο σταθμό. Το γράφημα αυτό προέκυψε χρησιμοποιώντας τις παραμέτρους $\lambda = 0,04$ για το ρυθμό αφίξεων, $\nu_1 = 0,35$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στον πρώτο σταθμό εργασίας, $\nu_2 = 0,27$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στο δεύτερο σταθμό εργασίας και $\mu_1 = 0,1$ για τον ευέλικτο εξυπηρετητή, ο οποίος βρίσκεται στο δεύτερο σταθμό εξυπηρέτησης. Τέλος, κόστος παραμονής στον πρώτο σταθμό $h_1 = 0,37$ και $h_2 = 0,32$, για το δεύτερο σταθμό.



Σχήμα 11 - Ευέλικτος στον πρώτο σταθμό

Αρχικά παρατηρούμε ότι δεν έχουμε ενεργοποίηση του ευέλικτου εξυπηρετητή.

Αυτό σημαίνει πως για μικρό αριθμό εργασιών στην ουρά αναμονής του πρώτου σταθμού, συμφέρει οι εργασίες να διεκπεραιώνονται αποκλειστικά από τους εξειδικευμένους εξυπηρετητές.

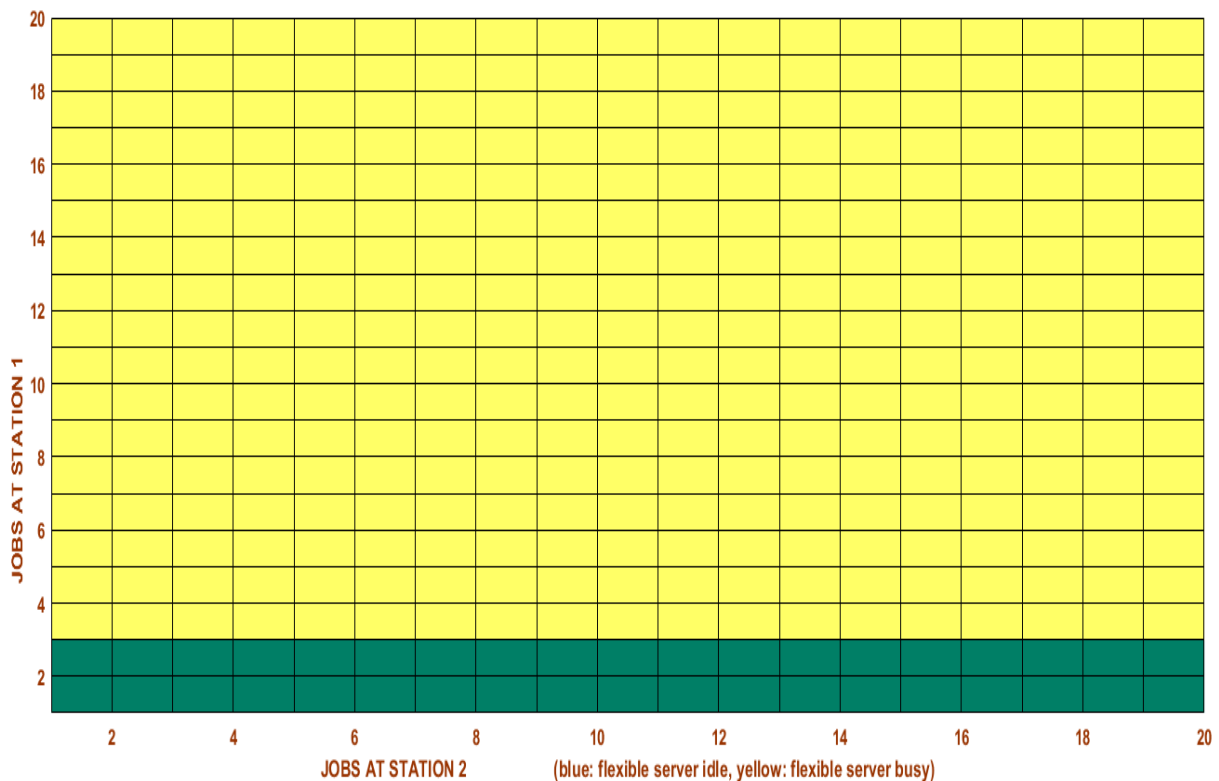
Στη συνέχεια, όμως είναι ξεκάθαρο πως η βέλτιστη πολιτική είναι η ενεργοποίηση του ευέλικτου εξυπηρετητή.

Η βέλτιστη πολιτική, επομένως, είναι ίδιας φύσεως με την περίπτωση που ο ευέλικτος βρισκόταν στο δεύτερο σταθμό εργασίας.

Σε αυτό το βήμα θα μελετήσουμε, ξανά, πιθανές αλλαγές που μπορούν να προκύψουν στο γράφημα με μεταβολές στις αρχικές τιμές που χρησιμοποιήθηκαν στον κώδικα, δηλαδή αυξομειώσεις είτε στο ρυθμό αυξήσεων, είτε στους ρυθμούς εξυπηρετήσεων, είτε στα κόστη παραμονής στον κάθε σταθμό.

Αρχικά, ενδιαφερόμαστε να εξετάσουμε τι θα προκύψει στο σχήμα με μια αλλαγή στο ρυθμό εξυπηρέτησης του ευέλικτου, ο οποίος σε αυτή την περίπτωση βρίσκεται στον πρώτο σταθμό εργασίας. Συγκεκριμένα, θα αυξήσουμε την τιμή του μ_1 από 0,1 σε 0,4.

Διατηρώντας, λοιπόν, όλους τους ρυθμούς εξυπηρέτησεων των εξυπηρετητών του συστήματος και τα κόστη παραμονής σταθερά, έχουμε:



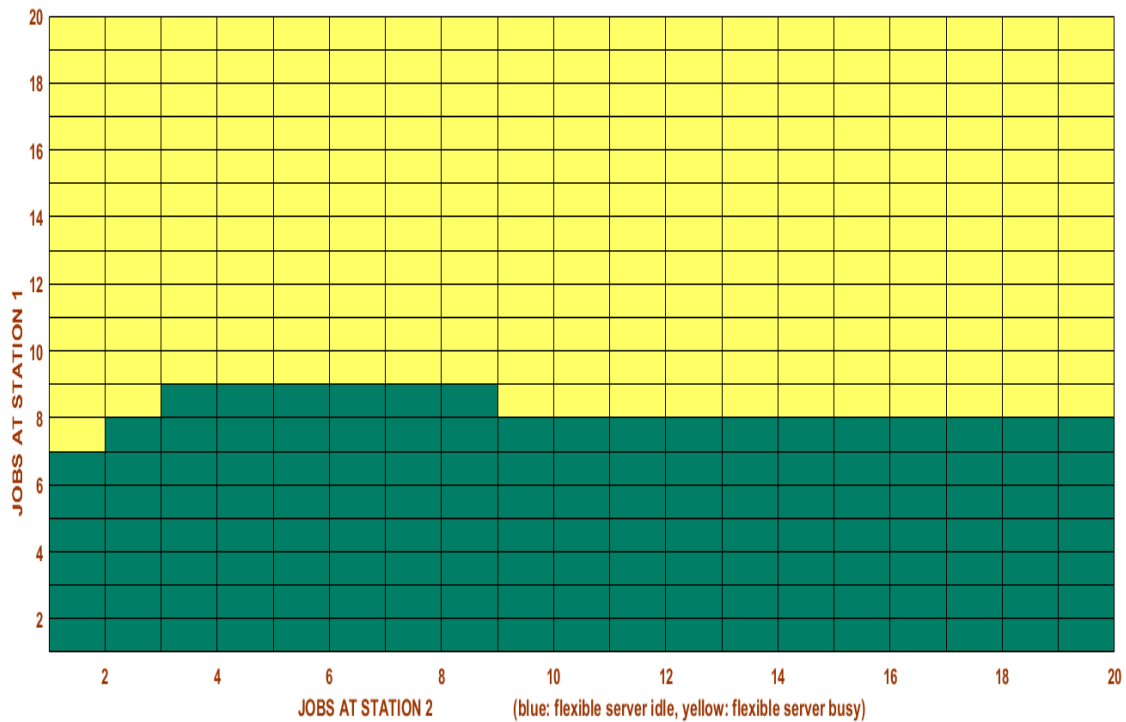
Σχήμα 12 - Αλλαγή σε ρυθμό εξυπηρέτησης

Παρατηρώντας κανείς το σχήμα μπορεί να κατανοήσει πως η μπλε περιοχή μειώθηκε σε σημαντικό βαθμό. Αυτό σημαίνει πως ο ευέλικτος εξυπηρετητής, που βρίσκεται στον πρώτο σταθμό εργασίας, τέθηκε σε λειτουργία για μικρότερο αριθμό εργασιών σε αναμονή αυτή τη φορά.

Αυτό γίνεται διότι, αυξάνοντας το ρυθμό εξυπηρέτησης μ_1 , ενώ ισχύει ταυτόχρονα $h_1 \geq h_2$ από τις αρχικές μας συνθήκες, το βέλτιστο είναι ο ευέλικτος να ενεργοποιηθεί για λιγότερες εργασίες σε αναμονή στην πρώτη ουρά, έτσι ώστε να ικανοποιήσει τις ανάγκες που υπάρχουν στην ουρά αυτή. στον πρώτο σταθμό εργασίας.

Στη συνέχεια θα δούμε πως το σύστημά μας επηρεάζεται από την μεταβολή της τιμής του εξειδικευμένου εξυπηρετητή του πρώτου σταθμού εργασίας, με μια αύξηση της τιμής ν_1 από 0,35 σε 0,65.

Αν διατηρήσουμε, λοιπόν, τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



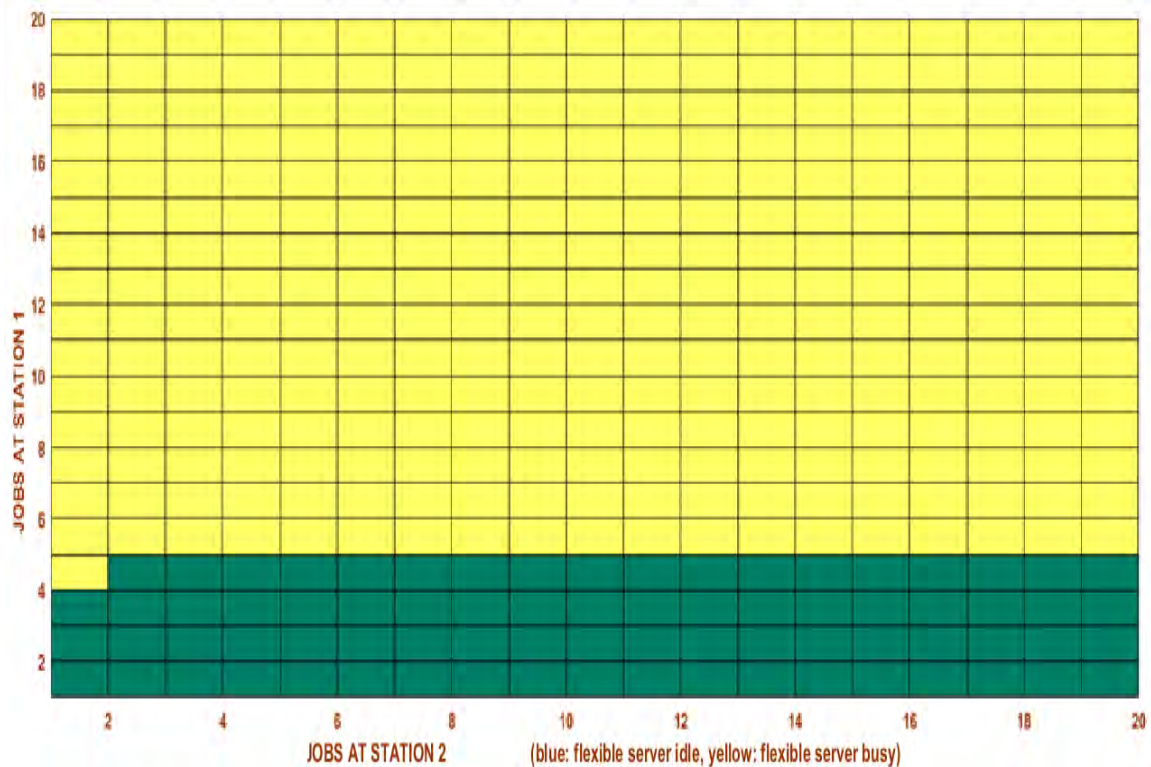
Σχήμα 13 - Αλλαγή σε ρυθμό εξυπηρέτησης

Από το γράφημα που προέκυψε με αύξηση του ρυθμού του σταθερού εξυπηρετητή στον πρώτο σταθμό, βλέπουμε πως ο ευέλικτος εξυπηρετητής, οπου βρίσκεται στον ίδιο σταθμό, παραμένει απενεργοποιημένος για περισσότερες εργασίες σε αναμονή στο σταθμό 2.

Το αποτέλεσμα αυτό είναι λογικό, εφόσον ο σταθερός εξυπηρετεί πιο γρήγορα, υπάρχει μειωμένη ανάγκη για επιπλέον εξυπηρέτηση.

Ενδιαφέρον παρουσιάζει να δούμε πως το σύστημά μας επηρεάζεται από την μεταβολή της τιμής του εξειδικευμένου εξυπηρετητή του δεύτερου σταθμού εργασίας, με μια μείωση της τιμής v_2 από 0,27 σε 0,07.

Διατηρώντας τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



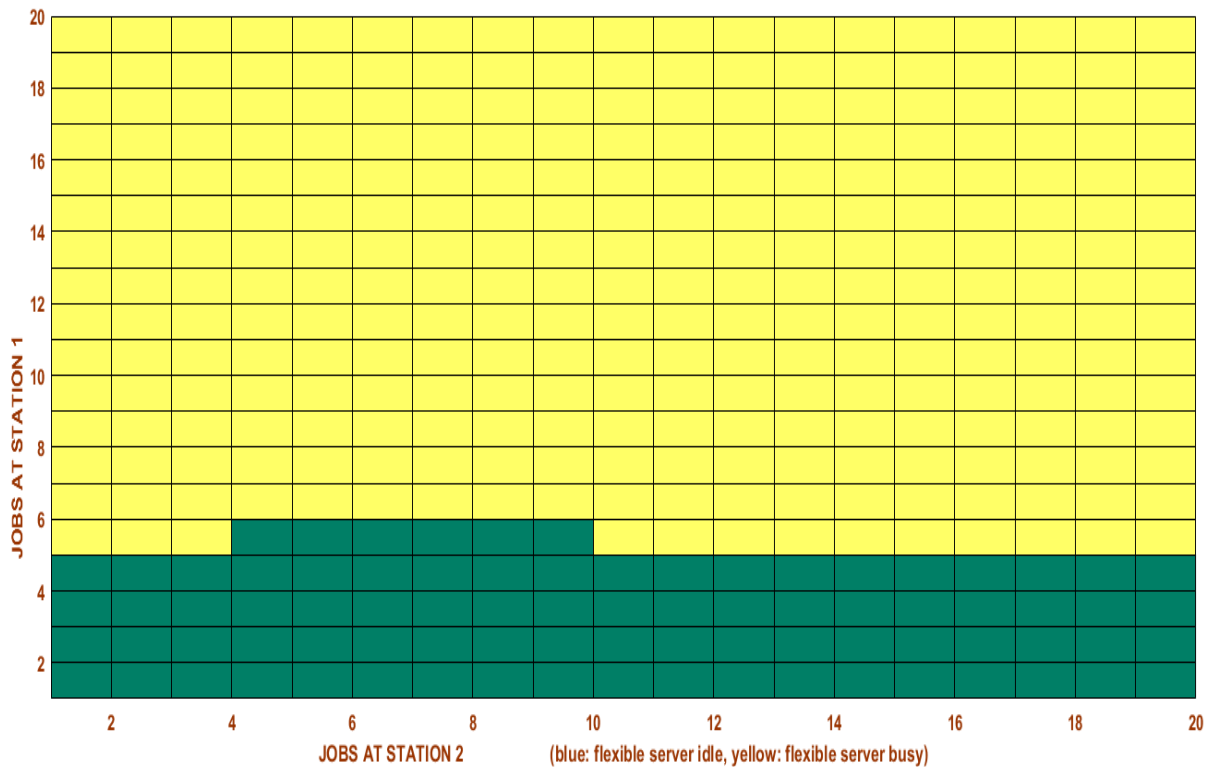
Σχήμα 14 - Αλλαγή σε ρυθμό εξυπηρέτησης

Αλλάζοντας λοιπόν το ρυθμό εξυπηρέτησης v_1 , δεν παρατηρείται καμία αλλαγή στο σχήμα. Ο ευέλικτος, ο οποίος σε αυτή την περίπτωση βρίσκεται στον πρώτο σταθμό εργασίας, ενεργοποιείται για τον ίδιο αριθμό εργασιών σε αναμονή, συγκριτικά με την αρχική περίπτωση.

Αυτό γίνεται διότι οι αυξημένες απαιτήσεις που παρουσιάζονται στο δεύτερο σταθμό εργασίας μετά τη μείωση του σταθερού εξυπηρετητή δεν μπορούν να διεκπεραιωθούν από τον ευέλικτο, αφού αυτός βρίσκεται στον πρώτο σταθμό εργασίας.

Σε αυτό το βήμα, ενδιαφερόμαστε να εξετάσουμε τι θα προκύψει στο σχήμα με μια αλλαγή στο ρυθμό αφίξεων του συστήματος λ . Συγκεκριμένα, θα μειώσουμε την τιμή αυτή από 0,04 σε 0,01.

Αν διατηρήσουμε, λοιπόν, όλους τους ρυθμούς εξυπηρέτησεων των εξυπηρετητών του συστήματος και τα κόστη παραμονής σταθερά, έχουμε:



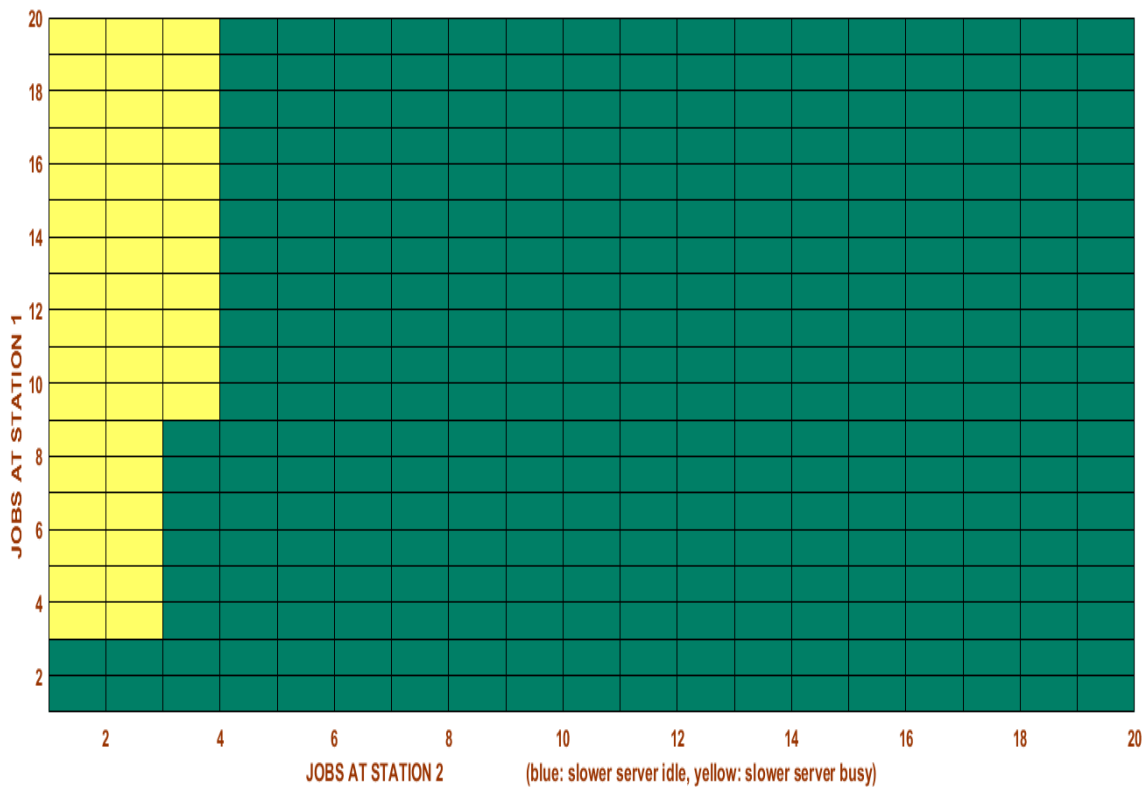
Σχήμα 15 – Αλλαγή σε ρυθμό αφίξεων

Μπορούμε να κατανοήσουμε πως η αλλαγή αυτή στο ρυθμό αφίξεων δεν επηρεάζει ιδιαίτερα το σχήμα. Ο ευέλικτος τίθεται σε λειτουργία περίπου για τον ίδιο αριθμό εργασιών σε αναμονή, όπως και προηγουμένως.

Το σύστημά μας επομένως, για ακόμη μια φορά, δεν φαίνεται να εξαρτάται τόσο από το ρυθμό αφίξεων (Poisson).

Τέλος, στις αρχικές τιμές που δώσαμε στον κώδικα, είχαμε υποθέσει ότι $h_1 \geq h_2$. Ενδιαφέρον έχει να δούμε το μετασχηματισμό του γραφήματος για $h_2 \geq h_1$ και συγκεκριμένα για $h_1=0,32$ και $h_2=0,37$.

Διατηρώντας, λοιπόν, το ρυθμό αφίξεων και όλους τους ρυθμούς εξυπηρέτησης σταθερούς, προκύπτει:



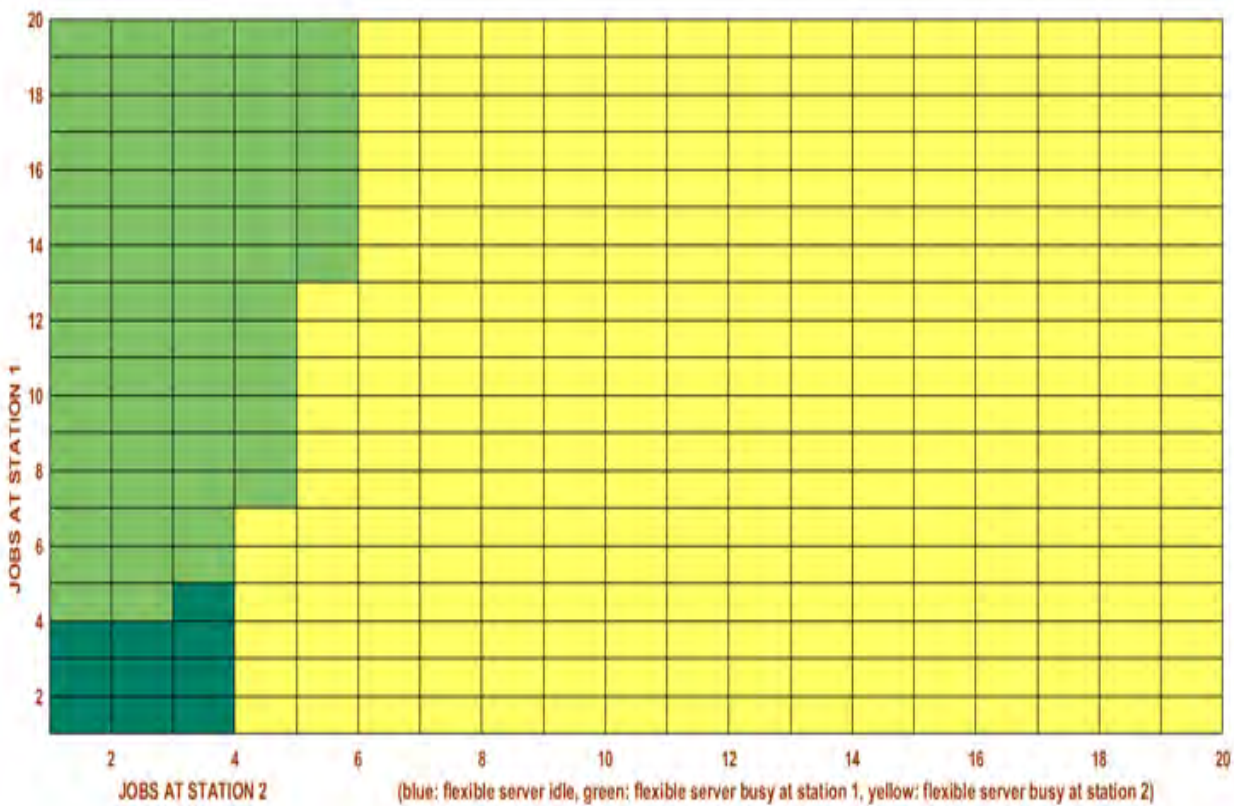
Σχήμα 16 – Αλλαγή στα κόστη παραμονής

Το να αλλάξει η συνθήκη του κόστους παραμονής σε $h_2 \geq h_1$, οδηγεί σε επιπλέον ανάγκες εξυπηρέτησης στο δεύτερο σταθμό.

Βρισκόμαστε, όμως, στην περίπτωση, όπου ο ευέλικτος βρίσκεται στον πρώτο σταθμό εργασίας, πράγμα που καθιστά λογικό το να ενεργοποιείται για μεγαλύτερο αριθμό εργασιών σε αναμον, εφόσον οι απαιτήσεις στον πρώτο σταθμό μειώθηκαν.

5.4 Αποτελέσματα για τη γενική περίπτωση

Στην περίπτωση αυτή θα δούμε το γράφημα που προκύπτει για την περίπτωση δύο σταθμών εργασίας, με τον ευέλικτο εξυπηρετητή να έχει τη δυνατότητα να βρίσκεται είτε στον πρώτο είτε στο δεύτερο σταθμό εργασίας. Το γράφημα αυτό προέκυψε χρησιμοποιώντας τις παραμέτρους $\lambda = 0,2$ για το ρυθμό αφίξεων, $\nu_1 = 0,42$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στον πρώτο σταθμό εργασίας, $\nu_2 = 0,45$ για το ρυθμό εξυπηρέτησης του εξειδικευμένου εξυπηρετητή στο δεύτερο σταθμό εργασίας, $\mu_1 = 0,14$ και $\mu_2 = 0,15$ για τον ευέλικτο εξυπηρετητή ανάλογα με το σταθμό εξυπηρέτησης στον οποίο βρίσκεται. Τέλος, κόστος παραμονής στον πρώτο σταθμό $h_1 = 0,37$ και $h_2 = 0,32$, για το δεύτερο σταθμό.



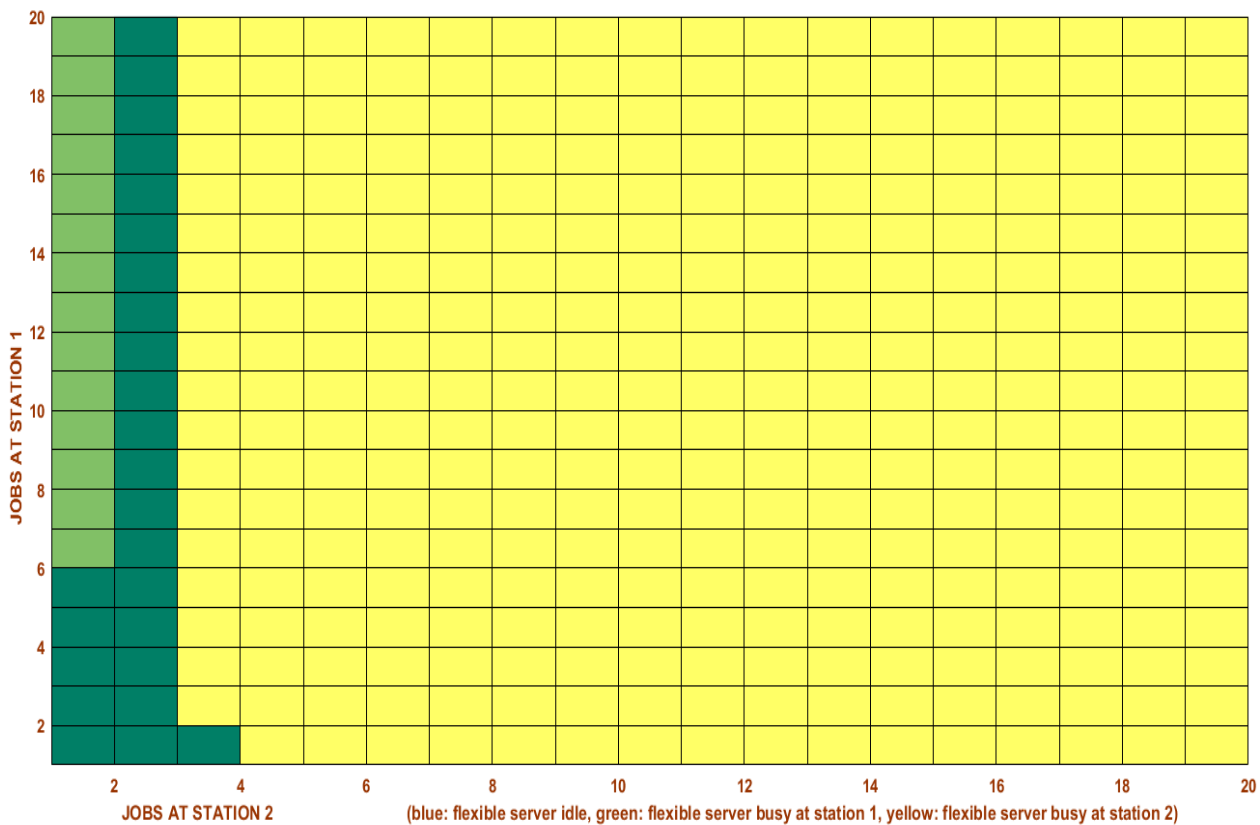
Σχήμα 17 – Ευέλικτος είτε στον πρώτο είτε στο δεύτερο σταθμό

Αρχικά παρατηρούμε πως δεν έχουμε ενεργοποίηση του ευέλικτου εξυπηρετητή, πράγμα που σημαίνει πως για μικρό αριθμό εργασιών σε αναμονή, συμφέρει οι εργασίες να διεκπεραιώνονται αποκλειστικά από τους εξειδικευμένους εξυπηρετητές. Στη συνέχεια, όσο ο αριθμός των εργασιών σε αναμονή αυξάνεται, ο ευέλικτος εξυπηρετητής τοποθετείται στον πρώτο σταθμό εργασίας και τελικά στο δεύτερο.

Ενδιαφέρον παρουσιάζει να δούμε τις μεταβολές που προκύπτουν στο γράφημα με αλλαγές των αρχικών παραμέτρων.

Σε πρώτο στάδιο, μπορούμε να δούμε πώς θα επηρεαστεί το τελευταίο σχήμα αυξάνοντας τον ρυθμό εξυπηρέτησης ν_1 του εξειδικευμένου εξυπηρετητή στον πρώτο σταθμό εργασίας από 0,42 σε 0,72.

Διατηρώντας τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



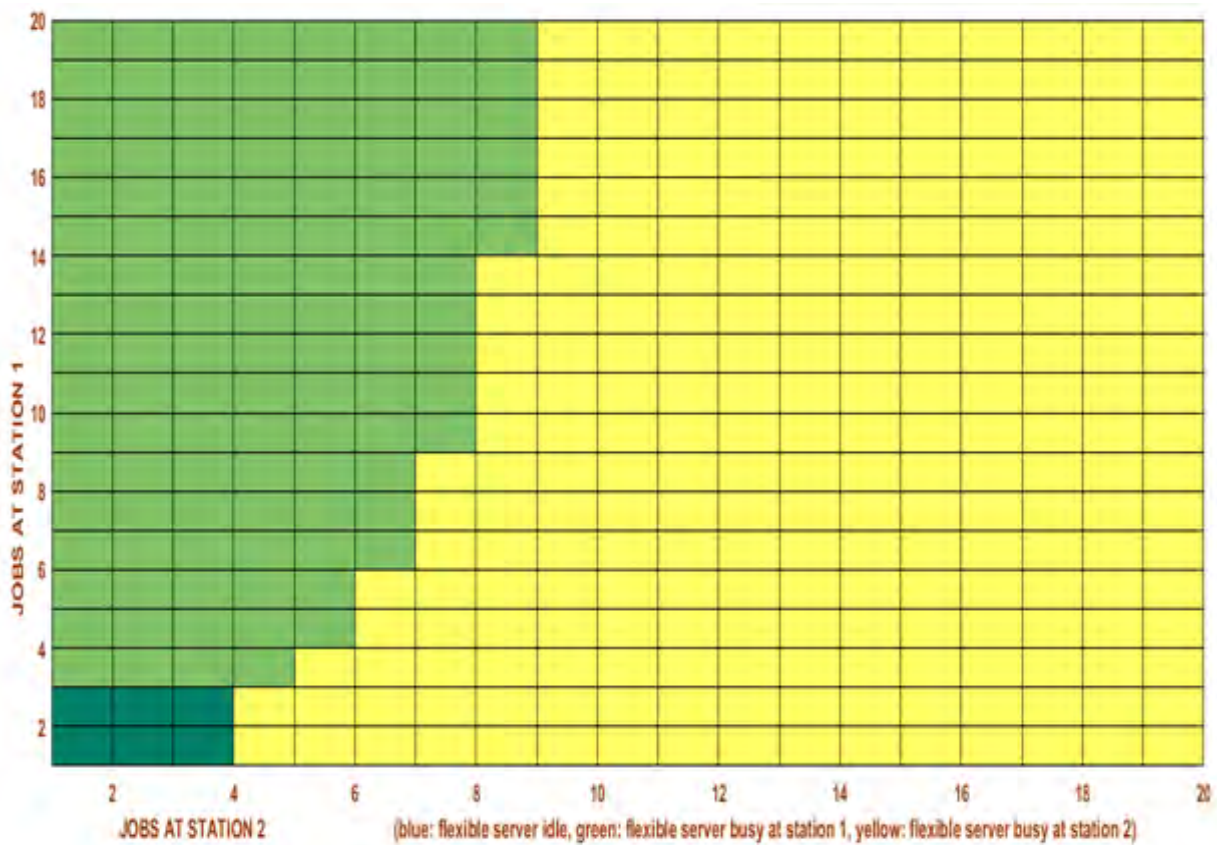
Σχήμα 19 - Αλλαγή σε ρυθμό εξυπηρέτησης

Αλλάζοντας λοιπόν το ρυθμό εξυπηρέτησης ν_1 η τοποθέτηση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας γίνεται για μικρότερο αριθμό εργασιών που περιμένουν και στις δύο ουρές.

Το αποτέλεσμα αυτό είναι απόλυτα λογικό, καθώς με την αύξηση αυτή μειώνονται οι απαιτήσεις για επιπλέον βοήθεια.

Στη συνέχεια, μπορούμε να μελετήσουμε το πώς θα επηρεαστεί το σύστημά μας αυξάνοντας τον ρυθμό εξυπηρέτησης μ_1 του ευέλικτου εξυπηρετητή στον πρώτο σταθμό εργασίας από 0,14 σε 0,64.

Αν διατηρήσουμε τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



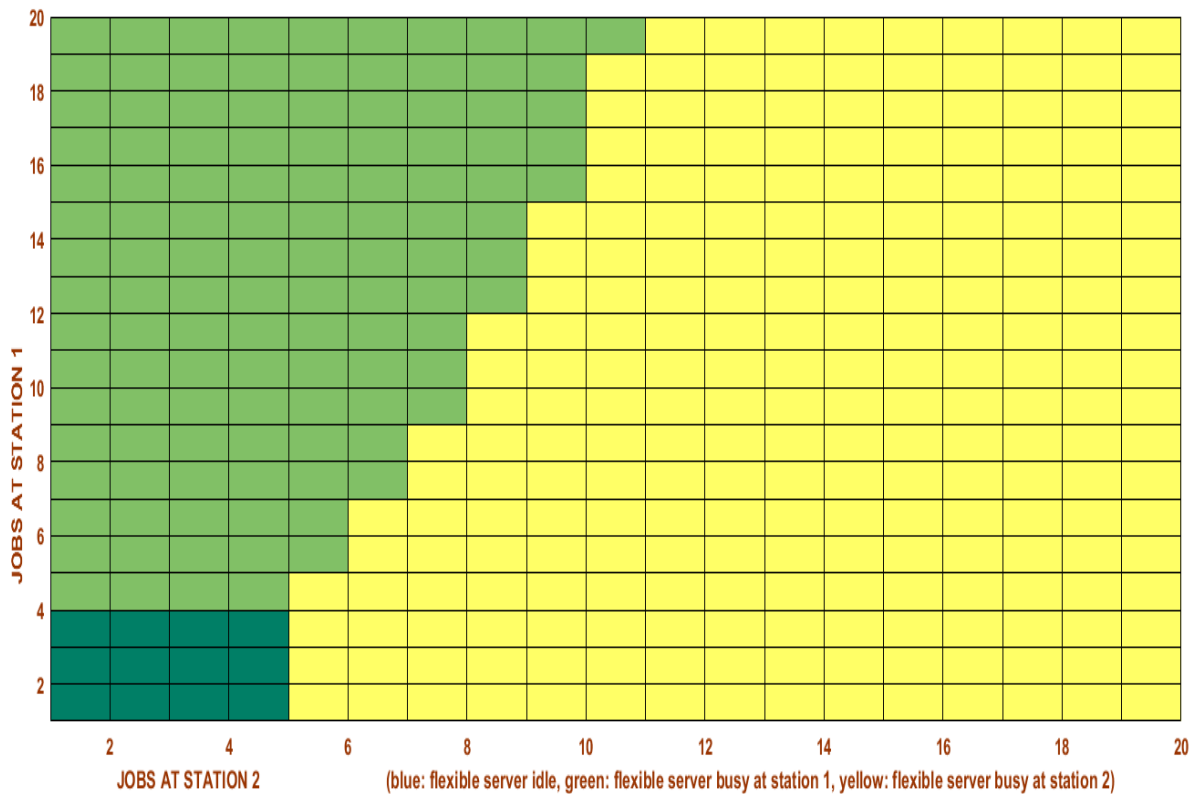
Σχήμα 18 – Αλλαγή στα κόστη παραμονής

Από την αλλαγή αυτή, θα προκύψει η τοποθέτηση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας για μεγαλύτερο αριθμό εργασιών που περιμένουν και στις δύο ουρές.

Αυτό γίνεται διότι, αυξάνοντας το ρυθμό εξυπηρέτησης μ_1 , ενώ ισχύει ταυτόχρονα $h_1 \geq h_2$, το βέλτιστο είναι ο ευέλικτος να παραμείνει στον πρώτο σταθμό εργασίας, για μεγαλύτερο αριθμό εργασιών σε αναμονή.

Μπορούμε, επίσης, να δούμε πώς θα επηρεαστεί το σύστημα αυξάνοντας τον ρυθμό εξυπηρέτησης ν_2 του εξειδικευμένου εξυπηρετητή στο δεύτερο σταθμό εργασίας από 0,45 σε 0,75.

Διατηρώντας τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



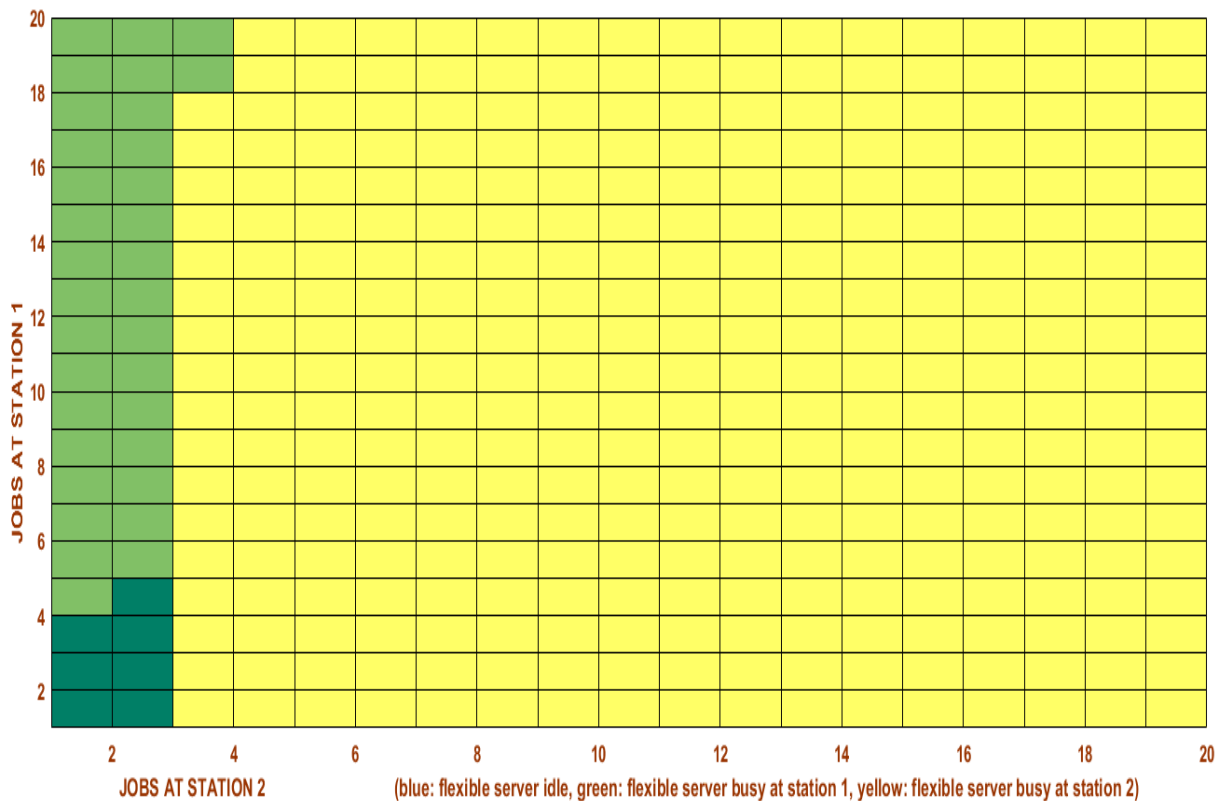
Σχήμα 19 - Αλλαγή σε ρυθμό εξυπηρέτησης

Αλλάζοντας λοιπόν το ρυθμό εξυπηρέτησης ν_2 , η τοποθέτηση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας γίνεται για μεγαλύτερο αριθμό εργασιών σε αναμονή.

Δηλαδή, όσο αυξάνεται ο ρυθμός με τον οποίο εξυπηρετεί ο σταθερός εξυπηρετητής στο δεύτερο σταθμό, τόσο μειώνονται οι απαιτήσεις για επιπλέον βοήθεια, έτσι ώστε να παρουσιαστεί ανάγκη για βοήθεια όταν θα περιμένουν περισσότερες εργασίες στις δύο ουρές.

Επιπλέον, μπορούμε να μελετήσουμε το πώς θα επηρεαστεί το σύστημά μας αυξάνοντας τον ρυθμό εξυπηρέτησης μ_2 του ευέλικτου εξυπηρετητή στον δεύτερο σταθμό εργασίας από 0,15 σε 0,65.

Αν διατηρήσουμε τους υπόλοιπους ρυθμούς εξυπηρέτησης, το ρυθμό αφίξεων και τα κόστη σταθερά, έχουμε:



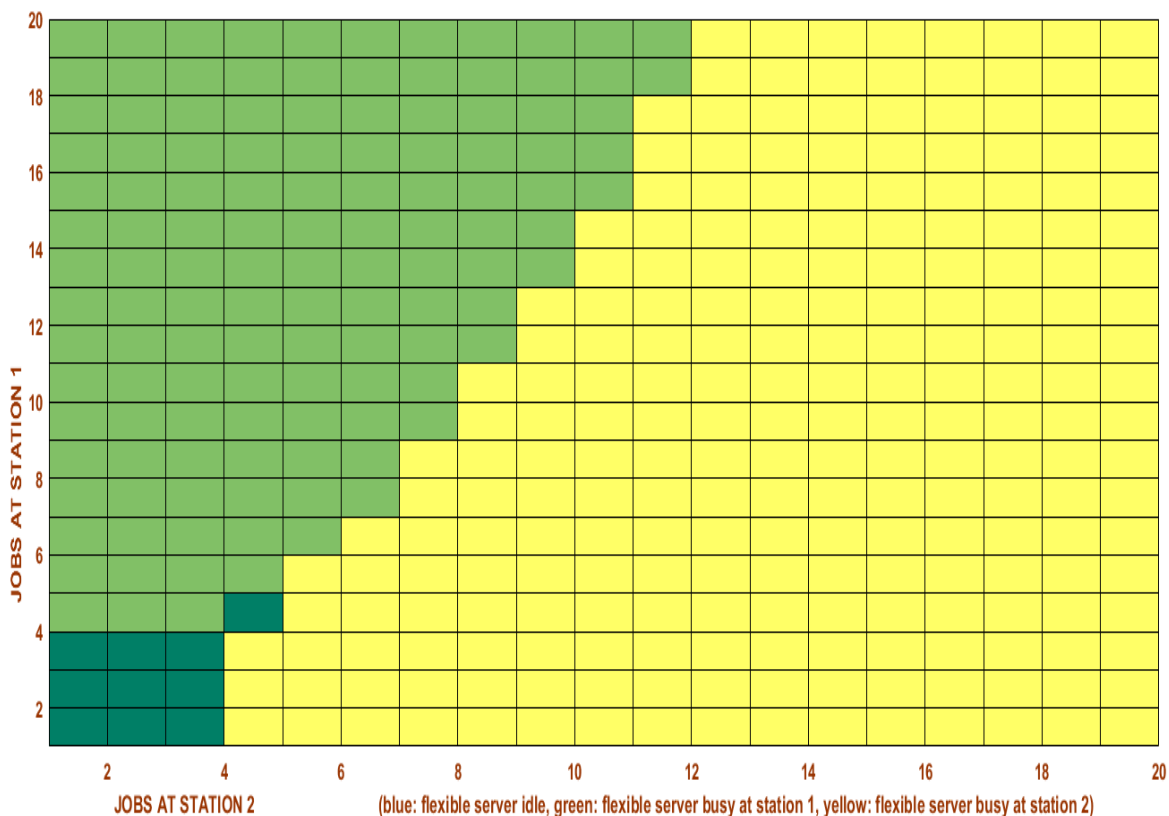
Σχήμα 20 - Αλλαγή σε ρυθμό εξυπηρέτησης

Από την αύξηση αυτή, θα προκύψει η τοποθέτηση του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας για μικρότερο αριθμό εργασιών σε αναμονή.

Αν μειώναμε τον αντίστοιχο αυτό ρυθμό του ευέλικτου εξυπηρετητή στο δεύτερο σταθμό εργασίας, θα μπορούσαμε να παρατηρήσουμε την ακριβώς αντίθετη συμπεριφορά. Δηλαδή, ο ευέλικτος θα είχε τοποθετηθεί στο δεύτερο σταθμό για μεγαλύτερο αριθμό εργασιών σε αναμονή.

Άλλη μια αλλαγή που αξίζει να πραγματοποιήσουμε είναι η αύξηση του κόστους παραμονής στον πρώτο σταθμό εργασίας και να δούμε ποιο θα είναι το αποτέλεσμα που θα προκύψει.

Αν διατηρήσουμε επομένως τις υπόλοιπες παραμέτρους σταθερές, αλλά αυξήσουμε το h_1 από 0,37 σε 0,67, προκύπτει το ακόλουθο σχήμα:



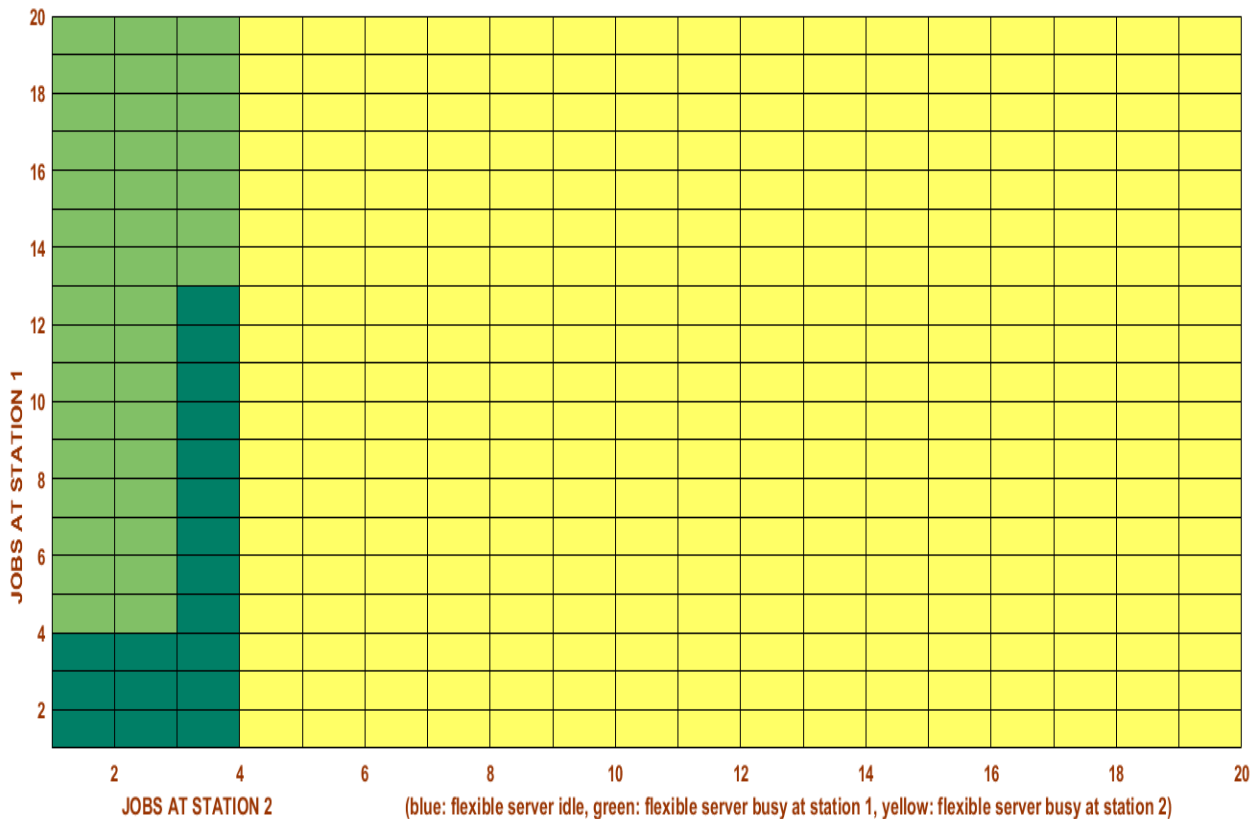
Σχήμα 21 – Αλλαγή στα κόστη παραμονής

Είναι προφανές ότι αυξάνοντας το κόστος παραμονής στον πρώτο σταθμό εργασίας, ο ευέλικτος τοποθετείται στο σταθμό αυτό.

Αυτό γίνεται διότι αυξάνοντας το κόστος παραμονής στην ουρά αυτή, αυξάνονται και οι απαιτήσεις εξυπηρέτησης. Εάν είχαμε μειώσει το κόστος αυτό, θα παρατηρούσαμε τοποθέτηση του ευέλικτου στο δεύτερο σταθμό εργασίας.

Μπορούμε αντίστοιχα να δούμε τι θα προκύψει εάν αυξήσουμε τώρα το κόστος παραμονής στο δεύτερο σταθμό εργασίας.

Διατηρώντας τις υπόλοιπες τιμές ως έχουν και αυξάνοντας το h_2 από 0,32 σε 0,72, έχουμε το παρακάτω σχήμα:



Σχήμα 22 - Αλλαγή στα κόστη παραμονής

Στο διάγραμμα αυτό διακρίνεται ξεκάθαρα πως η αύξηση του κόστους παραμονής στο δεύτερο σταθμό επηρέασε σε μεγάλο βαθμό την τοποθέτηση του ευέλικτου στο 2^ο σταθμό εργασίας.

Το αποτέλεσμα αυτό είναι απόλυτα λογικό, αν σκεφτεί κανείς πως στην περίπτωση αυτή το κόστος στη δεύτερη ουρά είναι αυξημένο, επομένως υπάρχει πρόσθετη ανάγκη για βοήθεια στο δεύτερο σταθμό.

Κεφάλαιο 6: Σύνοψη Αποτελεσμάτων

Εν κατακλείδι, μπορούμε να πούμε για τις περιπτώσεις που ο ευέλικτος βρίσκεται μόνο στον πρώτο ή μόνο στο δεύτερο σταθμό εργασίας:

- Όταν αυξάνεται ο ρυθμός εξυπηρέτησης ενός σταθερού εξυπηρετητή, οι απαιτήσεις για εξυπηρέτηση στο σταθμό αυτό μειώνονται, οπότε μειώνεται η χρήση του αντίστοιχου ευέλικτου. Αντίστοιχα όταν μειώνεται συμβαίνει το αντίστροφο.
- Όταν αυξάνεται ο ρυθμός εξυπηρέτησης του ευέλικτου εξυπηρετητή, οι απαιτήσεις για εξυπηρέτηση στο σταθμό αυτό μειώνονται, οπότε μειώνεται η χρήση του αντίστοιχου εξειδικευμένου. Αντίστοιχα όταν μειώνεται συμβαίνει το αντίστροφο.
- Ο ρυθμός αφίξεων δεν φάνηκε να επηρεάζει ιδιαίτερα τη χρήση του ευέλικτου εξυπηρετητή.
- Το κόστος παραμονής σε έναν σταθμό, δημιουργεί απαιτήσεις για επιπλέον εξυπηρέτηση, οπότε η βέλτιστη πολιτική είναι η χρήση του ευέλικτου εξυπηρετητή στο σταθμό αυτό.

Τέλος, για την περίπτωση που ο ευέλικτος μπορεί να βρίσκεται είτε στον πρώτο είτε στο δεύτερο σταθμό εργασίας:

- Όταν αυξάνεται ο ρυθμός εξυπηρέτησης του σταθερού εξυπηρετητή στον πρώτο σταθμό, οι απαιτήσεις για εξυπηρέτηση στο σταθμό αυτό μειώνονται, οπότε ο ευέλικτος τοποθετείται για μικρότερο αριθμό εργασιών σε αναμονή στο δεύτερο σταθμό εξυπηρέτησης και αντίστοιχα όταν μειώνεται συμβαίνει το αντίστροφο.
- Όταν αυξάνεται ο ρυθμός εξυπηρέτησης του σταθερού εξυπηρετητή στο δεύτερο σταθμό, οι απαιτήσεις για εξυπηρέτηση στο σταθμό αυτό μειώνονται, οπότε ο ευέλικτος τοποθετείται για μεγαλύτερο αριθμό εργασιών σε αναμονή στο δεύτερο σταθμό εξυπηρέτησης και αντίστοιχα όταν μειώνεται συμβαίνει το αντίστροφο.
- Ο ρυθμός αφίξεων ξανά φάνηκε πως δεν επηρεάζει σημαντικά τη χρήση του ευέλικτου εξυπηρετητή.
- Αυξημένο κόστος παραμονής σε έναν σταθμό εργασίας έχει ως βέλτιστη πολιτική τη χρήση του ευέλικτου εξυπηρετητή στο σταθμό αυτό.

Βιβλιογραφία

Akgun, O. T. (2014, March). Energy-Aware Scheduling on Heterogeneous Processors. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, VOL. 59, NO. 3 , σσ. 599-613.

Andradottir S. & Ayhan H. & Down D. G. (2013). Design principles for flexible systems. *Production and Operations Management*, Vol. 22 , σσ. 1144-1156.

Biran, A., & Breiner, M. (2012). *MATLAB 6 για μηχανικούς* (3rd Edition εκδ.). (N. I. Μάργαρης, Επιμ., D. I. Πεταλάς, & Άγγελος Δ. Δημητριάδης, Μεταφρ.) Θεσσαλονίκη: Εκδόσεις ΤΖΙΟΛΑ.

Frederick S. Hillier, & Gerald J. Lieberman. (2010). *Introduction to Operations Research* (Ninth Edition εκδ., Τόμ. International Edition). NewYork: McGraw-Hill Companies.

Hamdy A.Taha. (2012). *Εισαγωγή στην Επιχειρησιακή Έρευνα* (9η Έκδοση εκδ.). (Στέφανος Κατσαβούνης, Επιμ., & Αθανάσιος Ι. Μάργαρης, Μεταφρ.) Αθήνα: Εκδόσεις ΤΖΙΟΛΑ.

Hopp W. J. & Van Oyen M. P. (2004). Agile workforce evaluation: a framework for cross-training and coordination. *IIE Transactions*, Vol. 36 , σσ. 919-940.

Koole, G. (1995). A simple proof of the optimality of a threshold policy in a two-server queueing system. *Systems & Control Letters* 26 , σσ. 301-303.

Lin, W., & Kumar, P. (1984). Optimal control of a queueing system with two heterogeneous servers. *IEEE Transactions on Automatic Control*, Vol. 29 , σσ. 696-703.

Nahmias, S. (2013). *Productions & Operations Analysis* (sixth edition εκδ.). New York: McGraw-Hill/Irwin.

Ozkan, E. & Kharoufeh, J. (2014). Optimal control of a two-server queueing system with failures. *Probability in the Engineering and Informational Sciences*, 28 (4) , σσ. 489-527.

Papachristos, G., & Pandelis, D. G. (2016). On the Optimal Use of a Slow Server in Two-Stage Queueing Systems. *Proceedings of the International Conference of Numerical Analysis and Applied Mathematics (ICNAAM 2016)*. Rodos, Greece.

Walrand, J. (1984). A note on "Optimal Control of a queueing system with two heterogeneous servers". *Systems and Control Letters*, 4 , σσ. 131-134.

Ν. Δ. Τσάντας, & Π. Γ. Βασιλείου. (2000). *Εισαγωγή στην Επιχειρησιακή Έρευνα*. Θεσσαλονίκη: Εκδόσεις ΖΗΤΗ.

Παράρτημα

Παρακάτω υπάρχει ο κώδικας (στη γλώσσα προγραμματισμού Matlab) όλων των εφαρμογών που χρησιμοποιήθηκαν για την παρούσα διπλωματική εργασία.

Συμβολισμοί που χρησιμοποιήθηκαν:

- Ο πίνακας που λαμβάνει τις τιμές του ελάχιστου κόστους σε κάθε περίοδο (n): $\mathbf{Vn}(x_1, x_2)$, ενώ για την προηγούμενη περίοδο: $\mathbf{Vp}(x_1, x_2)$.
- Ο πίνακας που αποθηκεύει την κατάσταση του αργού εξυπηρετητή: $\mathbf{K}(x_1, x_2)$.
- Ο πίνακας που αποθηκεύει τις διαφορές της περιόδου n με την $n - 1$: \mathbf{Dif} .
- Τα όρια που χρειαζόμαστε για το αλγόριθμο επαναλαμβανόμενης αξίας είναι: \mathbf{Mn} για το άνω όριο και \mathbf{mn} για το κάτω όριο.
- Μετρητής των περιόδων: \mathbf{kk} .
- Ο πίνακας που αποθηκεύει σε κάθε περίοδο την τιμή για το κριτήριο του αλγορίθμου: $\mathbf{rep}(kk)$.
- Ρυθμός αφίξεων: \mathbf{L} .
- Ρυθμοί εξυπηρέτησης: \mathbf{N} για τον εξειδικευμένο εξυπηρετητή στην περίπτωση ενός σταθμού εργασίας, $\mathbf{N1}$ για τον πρώτο εξειδικευμένο εξυπηρετητή, $\mathbf{N2}$ για τον δεύτερο εξειδικευμένο εξυπηρετητή στις άλλες περιπτώσεις και \mathbf{M} για τον αργό εξυπηρετητή. Επίσης, $\mathbf{M1}$ για τον αργό εξυπηρετητή στον πρώτο σταθμό και $\mathbf{M2}$ στο δεύτερο.
- Αντίστοιχες πιθανότητες στο διακριτοποιημένο μοντέλο μετά από ομοιομορφοποίηση: \mathbf{l} , \mathbf{n} , $\mathbf{n1}$, $\mathbf{n2}$, \mathbf{m} , $\mathbf{m1}$, $\mathbf{m2}$ αντίστοιχα.
- Κόστη παραμονής: $\mathbf{h1}$ για το σταθμό 1 και $\mathbf{h2}$ για το σταθμό 2.
- Μέγιστο πλήθος εργασιών: $\mathbf{L1}$ για το σταθμό 1 και $\mathbf{L2}$ για το σταθμό 2.
- Εκπτώτικός συντελεστής: \mathbf{b}
- Λοιπές βοηθητικές παράμετροι: \mathbf{i} , \mathbf{j} , \mathbf{k} , \mathbf{uni}

```

% Copyright(C) Bitros Panagiotis 2017

%single station queuing system
%with a dedicated and a slower server

L1=100;

for i=0: L1
for j=0: 1
Vn(i+1,j+1)=0;
end
end

%assigning values

Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;

b=1;

%l<n
L=0.27;
M=0.1;
N=0.35;

%uniformization
uni=L+M+N;
l=L/uni;
n=N/uni;
m=M/uni;

while (rep(kk)>0.001*mn)

%Wn(0,0)
Vn(1,1)=b*(1*Vp(2,1)+n*Vp(1,1)+m*Vp(1,1));
%Wn(0,1)
Vn(1,2)=1+b*(1*Vp(2,2)+n*Vp(1,2)+m*Vp(1,1));
%Wn(1,0)
Vn(2,1)=1+b*(1*Vp(3,1)+n*Vp(1,1)+m*Vp(2,1));
%Wn(1,1)
Vn(2,2)=2+b*(1*Vp(3,2)+n*Vp(1,2)+m*Vp(2,1));

for i=2:L1-1;
%Wn(x,1)
Vn(i+1,2)=i+1+b*(1*Vp(i+2,2)+n*Vp(i,2)+m*Vp(i+1,1));
%Wn(L1,1)
Vn(L1+1,2)=L1+1+b*(1*Vp(L1+1,2)+n*Vp(L1,2)+m*Vp(L1+1,1));
end

for i=2:L1-1;
%Wn(x,0)
V1(i+1,1)=i+b*(1*Vp(i+2,1)+n*Vp(i,1)+m*Vp(i+1,1));

```

```

%Wn(x-1,1)
V2(i+1,1)=i+b*(1*Vp(i+1,2)+n*Vp(i-1,2)+m*Vp(i,1));
Vn(i+1,1)=min(V1(i+1,1),V2(i+1,1));

%DECISION
if Vn(i+1,1)==V1(i+1,1)
K(i+1)=0;
else
K(i+1)=1;
end

%Wn(L1,0)
V1(L1+1,1)=L1+b*(1*Vp(L1+1,1)+n*Vp(L1,1)+m*Vp(L1+1,1));
%Wn(L1-1,1)
V2(L1+1,1)=L1+b*(1*Vp(L1+1,2)+n*Vp(L1-1,2)+m*Vp(L1,1));
Vn(L1+1,1)=min(V1(L1+1,1),V2(L1+1,1));

%DECISION
if Vn(L1+1,1)==V1(L1+1,1)
K(L1+1)=0;
else
K(L1+1)=1;
end
end

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

end

%two stage queuing system
%two dedicated servers and a slower server
%slower server at station 2

L1=100;
L2=100;

for i=0: L1;
for j=0: L2;
for k=0: 1;
Vn(i+1,j+1,k+1)=0;
end
end
end

%assigning values

Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;

```



```

Mn=0;

b=1;

%l<n1 & l<n2+m
L=0.26;
M=0.2;
N1=0.3;
N2=0.24;

H1=0.37587;
H2=0.32645;

%uniformization
uni=L+N1+N2+M;
l=L/uni;
n1=N1/uni;
n2=N2/uni;
m=M/uni;
h1=H1/uni;
h2=H2/uni;

while (rep(kk)>0.001*mn)

%Vn(0,0,0)
Vn(1,1,1)=b*(1*Vp(2,1,1)+(1-l)*Vp(1,1,1));
%Vn(0,0,1)
Vn(1,1,2)=h2+b*(1*Vp(2,1,2)+n1*Vp(1,1,2)+n2*Vp(1,1,2)+m*Vp(1,1,1));
%Vn(0,1,0)
Vn(1,2,1)=h2+b*(1*Vp(2,2,1)+n1*Vp(1,2,1)+n2*Vp(1,1,1)+m*Vp(1,2,1));
%Vn(0,1,1)
Vn(1,2,2)=(2*h2)+b*(1*Vp(2,2,2)+n1*Vp(1,2,2)+n2*Vp(1,1,2)+m*Vp(1,2,1));

for i=1:L1-1;
%Wn(x,0,0)
Vn(i+1,1,1)=h1*i+b*(1*Vp(i+2,1,1)+n1*Vp(i,2,1)+n2*Vp(i+1,1,1)+m*Vp(i+1,1,1));
%Wn(x,0,1)
Vn(i+1,1,2)=h1*i+h2+b*(1*Vp(i+2,1,2)+n1*Vp(i,2,2)+n2*Vp(i+1,1,2)+m*Vp(i+1,1,1));
%Wn(x,1,0)
Vn(i+1,2,1)=h1*i+h2+b*(1*Vp(i+2,2,1)+n1*Vp(i,3,1)+n2*Vp(i+1,1,1)+m*Vp(i+1,2,1));
%Wn(x,1,1)
Vn(i+1,2,2)=h1*i+2*h2+b*(1*Vp(i+2,2,2)+n1*Vp(i,3,2)+n2*Vp(i+1,1,2)+m*Vp(i+1,2,1));
end

for i=1:L1-1;
for j=2:L2-1;
%Wn(x,y,0)

```

```

V1(i+1,j+1,1)=(h1*i)+(h2*j)+b*(1*Vp(i+2,j+1,1)+n1*Vp(i,j+2,1)+
n2*Vp(i+1,j,1)+m*Vp(i+1,j+1,1));
%Wn(x,y-1,1)
V2(i+1,j+1,1)=(h1*i)+(h2*j)+b*(1*Vp(i+2,j,2)+n1*Vp(i,j+1,2)+n2
*Vp(i+1,j-1,2)+m*Vp(i+1,j,1));
Vn(i+1,j+1,1)=min(V1(i+1,j+1,1),V2(i+1,j+1,1));

%DECISION
if Vn(i+1,j+1,1)==V1(i+1,j+1,1)
K(i+1,j+1)=0;
else
K(i+1,j+1)=1;
end
end
end

for i=1:L1-1;
for j=2:L2-1;
%Wn(x,y,1)
Vn(i+1,j+1,2)=(h1*i)+(h2*j)+h2+b*(1*Vp(i+2,j+1,2)+n1*Vp(i,j+2,
2)+n2*Vp(i+1,j,2)+m*Vp(i+1,j+1,1));
end
end

for j=2:L2-1;
%Wn(0,y,0)
V1(1,j+1,1)=(h2*j)+b*(1*Vp(2,j+1,1)+n1*Vp(1,j+1,1)+n2*Vp(1,j,1
)+m*Vp(1,j+1,1));
%Wn(0,y-1,1)
V2(1,j+1,1)=h2*(j-1)+h2+b*(1*Vp(2,j,2)+n1*Vp(1,j,2)+n2*Vp(1,j-
1,2)+m*Vp(1,j,1));
Vn(1,j+1,1)=min(V1(1,j+1,1),V2(1,j+1,1));
%Wn(0,y,1)
Vn(1,j+1,2)=h2*j+h2+b*(1*Vp(2,j+1,2)+n1*Vp(1,j+1,2)+n2*Vp(1,j,
2)+m*Vp(1,j+1,1));

%DECISION
if Vn(1,j+1,1)==V1(1,j+1,1)
K(1,j+1)=0;
else
K(1,j+1)=1;
end
end

for i=1:L1-1;
%Wn(x,L2,0)
V1(i+1,L2+1,1)=h1*i+h2*L2+b*(1*Vp(i+2,L2+1,1)+n1*Vp(i,L2+1,1)+
n2*Vp(i+1,L2,1)+m*Vp(i+1,L2+1,1));
%Wn(x,L2-1,1)
V2(i+1,L2+1,1)=h1*i+h2*(L2-
1)+h2+b*(1*Vp(i+2,L2,2)+n1*Vp(i,L2+1,2)+n2*Vp(i+1,L2-
1,2)+m*Vp(i+1,L2,1));
Vn(i+1,L2+1,1)=min(V1(i+1,L2+1,1),V2(i+1,L2+1,1));
%Wn(x,L2,1)

```

```
Vn (i+1, L2+1, 2) = h1*i+h2*L2+h2+b* (1*Vp (i+2, L2+1, 2) + n1*Vp (i, L2+1, 2) + n2*Vp (i+1, L2, 2) + m*Vp (i+1, L2+1, 1) );
```

```
%DECISION
```

```
if Vn (i+1, L2+1, 1) == V1 (i+1, L2+1, 1)
```

```
K (i+1, L2+1) = 0;
```

```
else
```

```
K (i+1, L2+1) = 1;
```

```
end
```

```
end
```

```
for j = 2:L2-1;
```

```
%Wn (L1, y, 0)
```

```
V1 (L1+1, j+1, 1) = h1*L1+h2*j+b* (1*Vp (L1+1, j+1, 1) + n1*Vp (L1, j+2, 1) + n2*Vp (L1+1, j, 1) + m*Vp (L1+1, j+1, 1) );
```

```
%Wn (L1, y-1, 1)
```

```
V2 (L1+1, j+1, 1) = h1*L1+h2* (j-1) + h2+b* (1*Vp (L1+1, j, 2) + n1*Vp (L1, j+1, 2) + n2*Vp (L1+1, j-1, 2) + m*Vp (L1+1, j, 1) );
```

```
Vn (L1+1, j+1, 1) = min (V1 (L1+1, j+1, 1), V2 (L1+1, j+1, 1) );
```

```
%Wn (L1, y, 1)
```

```
Vn (L1+1, j+1, 2) = h1*L1+h2*j+h2+b* (1*Vp (L1+1, j+1, 2) + n1*Vp (L1, j+2, 2) + n2*Vp (L1+1, j, 2) + m*Vp (L1+1, j+1, 1) );
```

```
%DECISION
```

```
if Vn (L1+1, j+1, 1) == V1 (L1+1, j+1, 1)
```

```
K (L1+1, j+1) = 0;
```

```
else
```

```
K (L1+1, j+1) = 1;
```

```
end
```

```
end
```

```
%Wn (0, L2, 0)
```

```
V1 (1, L2+1, 1) = h2*L2+b* (1*Vp (2, L2+1, 1) + n1*Vp (1, L2+1, 1) + n2*Vp (1, L2, 1) + m*Vp (1, L2+1, 1) );
```

```
%Wn (0, L2-1, 1)
```

```
V2 (1, L2+1, 1) = h2* (L2-1) + h2+b* (1*Vp (2, L2, 2) + n1*Vp (1, L2, 2) + n2*Vp (1, L2-1, 2) + m*Vp (1, L2, 1) );
```

```
Vn (1, L2+1, 1) = min (V1 (1, L2+1, 1), V2 (1, L2+1, 1) );
```

```
%Wn (0, L2, 1)
```

```
Vn (1, L2+1, 2) = h2*L2+h2+b* (1*Vp (2, L2+1, 2) + n1*Vp (1, L2+1, 2) + n2*Vp (1, L2, 2) + m*Vp (1, L2+1, 1) );
```

```
%DECISION
```

```
if Vn (1, L2+1, 1) == V1 (1, L2+1, 1)
```

```
K (1, L2+1) = 0;
```

```
else
```

```
K (1, L2+1) = 1;
```

```
end
```

```
%Wn (L1, L2, 0)
```

```
V1 (L1+1, L2+1, 1) = h1*L1+h2*L2+b* (1*Vp (L1+1, L2+1, 1) + n1*Vp (L1, L2+1, 1) + n2*Vp (L1+1, L2, 1) + m*Vp (L1+1, L2+1, 1) );
```

```
%Wn (L1, L2-1, 1)
```

```

V2 (L1+1, L2+1, 1) = h1*L1+h2* (L2-
1) +h2+b* (1*Vp (L1+1, L2, 2) +n1*Vp (L1, L2+1, 2) +n2*Vp (L1+1, L2-
1, 2) +m*Vp (L1+1, L2, 1));
Vn (L1+1, L2+1, 1) = min (V1 (L1+1, L2+1, 1), V2 (L1+1, L2+1, 1));
%Wn (L1, L2, 1)
Vn (L1+1, L2+1, 2) = h1*L1+h2*L2+h2+b* (1*Vp (L1+1, L2+1, 2) +n1*Vp (L1, L
2+1, 2) +n2*Vp (L1+1, L2, 2) +m*Vp (L1+1, L2+1, 1));

%DECISION
if Vn (L1+1, L2+1, 1) == V1 (L1+1, L2+1, 1)
K (L1+1, L2+1) = 0;
else
K (L1+1, L2+1) = 1;
end

%Wn (L1, 0, 1)
Vn (L1+1, 1, 2) = h1*L1+h2+b* (1*Vp (L1+1, 1, 2) +n1*Vp (L1, 2, 2) +n2*Vp (L1
+1, 1, 2) +m*Vp (L1+1, 1, 1));
%Wn (L1, 1, 1)
Vn (L1+1, 2, 2) = h1*L1+h2+h2+b* (1*Vp (L1+1, 2, 2) +n1*Vp (L1, 3, 2) +n2*Vp
(L1+1, 1, 2) +m*Vp (L1+1, 2, 1));
%Wn (L1, 0, 0)
Vn (L1+1, 1, 1) = h1*L1+b* (1*Vp (L1+1, 1, 1) +n1*Vp (L1, 2, 1) +n2*Vp (L1+1,
1, 1) +m*Vp (L1+1, 1, 1));
%Wn (L1, 1, 0)
Vn (L1+1, 2, 1) = h1*L1+h2+b* (1*Vp (L1+1, 2, 1) +n1*Vp (L1, 3, 1) +n2*Vp (L1
+1, 1, 1) +m*Vp (L1+1, 2, 1));

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

end

%two stage queuing system
%two dedicated servers and a slower server
%slower server at station 1

L1 = 100;
L2 = 100;

for i = 0 : L1;
for j = 0 : L2;
for k = 0 : 1;
Vn (i + 1, j + 1, k + 1) = 0;
end
end
end

%assigning values

Vp = Vn;

```

```

kk=1;
rep(kk)=1;
mn=0;
Mn=0;

b=1;

%l<n1+m & l<n2
L=0.06;
M=0.24;
N1=0.28;
N2=0.26;

H1=0.37587;
H2=0.32645;

%uniformization
uni=L+N1+N2+M;
l=L/uni;
n1=N1/uni;
n2=N2/uni;
m=M/uni;
h1=H1/uni;
h2=H2/uni;

while (rep(kk)>0.001*mn)

%Wn(0,0,0)
Vn(1,1,1)=b*(1*Vp(2,1,1)+(1-1)*Vp(1,1,1));
%Wn(0,0,1)
Vn(1,1,2)=h1+b*(1*Vp(2,1,2)+n1*Vp(1,1,2)+m*Vp(1,2,1)+n2*Vp(1,1,2));
%Wn(0,1,0)
Vn(1,2,1)=h2+b*(1*Vp(2,2,1)+n1*Vp(1,2,1)+m*Vp(1,2,1)+n2*Vp(1,1,1));
%Wn(0,1,1)
Vn(1,2,2)=h2+h1+b*(1*Vp(2,2,2)+n1*Vp(1,2,2)+m*Vp(1,3,1)+n2*Vp(1,1,2));

%Wn(1,0,0)
Vn(2,1,1)=h1+b*(1*Vp(3,1,1)+n1*Vp(1,2,1)+m*Vp(2,1,1)+n2*Vp(2,1,1));
%Wn(1,0,1)
Vn(2,1,2)=h1+h1+b*(1*Vp(3,1,2)+n1*Vp(1,2,2)+m*Vp(2,2,1)+n2*Vp(2,1,2));
%Wn(1,1,0)
Vn(2,2,1)=h1+h2+b*(1*Vp(3,2,1)+n1*Vp(1,3,1)+m*Vp(2,2,1)+n2*Vp(2,1,1));
%Wn(1,1,1)
Vn(2,2,2)=h1+h2+h1+b*(1*Vp(3,2,2)+n1*Vp(1,3,2)+m*Vp(2,3,1)+n2*Vp(2,1,2));

for i=2:L1-1;
%Wn(x,0,0)

```

```

V1(i+1,1,1)=h1*i+b*(1*Vp(i+2,1,1)+n1*Vp(i,2,1)+m*Vp(i+1,1,1)+n
2*Vp(i+1,1,1));
%Wn(x-1,0,1)
V2(i+1,1,1)=h1*(i-1)+h1+b*(1*Vp(i+1,1,2)+n1*Vp(i-
1,2,2)+m*Vp(i,2,1)+n2*Vp(i,1,2));
Vn(i+1,1,1)=min(V1(i+1,1,1),V2(i+1,1,1));
%Wn(x,0,1)
Vn(i+1,1,2)=h1*i+h1+b*(1*Vp(i+2,1,2)+n1*Vp(i,2,2)+m*Vp(i+1,2,1
)+n2*Vp(i+1,1,2));

%DECISION
if Vn(i+1,1,1)==V1(i+1,1,1)
K(i+1,1)=0;
else
K(i+1,1)=1;
end
%Wn(x,1,0)
V1(i+1,2,1)=h1*i+h2+b*(1*Vp(i+2,2,1)+n1*Vp(i,3,1)+m*Vp(i+1,2,1
)+n2*Vp(i+1,1,1));
%Wn(x-1,1,1)
V2(i+1,2,1)=h1*(i-1)+h2+h1+b*(1*Vp(i+1,2,2)+n1*Vp(i-
1,3,2)+m*Vp(i,3,1)+n2*Vp(i,1,2));
Vn(i+1,2,1)=min(V1(i+1,2,1),V2(i+1,2,1));
%Wn(x,1,1)
Vn(i+1,2,2)=h1*i+h2+h1+b*(1*Vp(i+2,2,2)+n1*Vp(i,3,2)+m*Vp(i+1,
3,1)+n2*Vp(i+1,1,2));

%DECISION
if Vn(i+1,2,1)==V1(i+1,2,1)
K(i+1,2)=0;
else
K(i+1,2)=1;
end
end

for i=2:L1-1;
for j=1:L2-1;
%Wn(x,y,0)
V1(i+1,j+1,1)=(h1*i)+(h2*j)+b*(1*Vp(i+2,j+1,1)+n1*Vp(i,j+2,1)+
m*Vp(i+1,j+1,1)+n2*Vp(i+1,j,1));
%Wn(x-1,y,1)
V2(i+1,j+1,1)=h1*(i-1)+(h2*j)+h1+b*(1*Vp(i+1,j+1,2)+n1*Vp(i-
1,j+2,2)+m*Vp(i,j+2,1)+n2*Vp(i,j,2));
Vn(i+1,j+1,1)=min(V1(i+1,j+1,1),V2(i+1,j+1,1));

%DECISION
if Vn(i+1,j+1,1)==V1(i+1,j+1,1)
K(i+1,j+1)=0;
else
K(i+1,j+1)=1;
end
end
end
end

```

```

for j=2:L2-1;
%Wn(1,y,0)
Vn(2,j+1,1)=h1+(h2*j)+b*(1*Vp(3,j+1,1)+n1*Vp(1,j+2,1)+m*Vp(2,j
+1,1)+n2*Vp(2,j,1));
%Wn(1,y,1)
Vn(2,j+1,2)=h1+h2*j+h1+b*(1*Vp(3,j+1,2)+n1*Vp(1,j+2,2)+m*Vp(2,
j+2,1)+n2*Vp(2,j,2));
end

for j=2:L2-1;
%Wn(0,y,0)
Vn(1,j+1,1)=(h2*j)+b*(1*Vp(2,j+1,1)+n1*Vp(1,j+1,1)+m*Vp(1,j+1,
1)+n2*Vp(1,j,1));
%Wn(0,y,1)
Vn(1,j+1,2)=h2*j+h1+b*(1*Vp(2,j+1,2)+n1*Vp(1,j+1,2)+m*Vp(1,j+2
,1)+n2*Vp(1,j,2));
end

for i=1:L1-1;
for j=2:L2-1;
%Wn(x,y,1)
Vn(i+1,j+1,2)=(h1*i)+(h2*j)+h1+b*(1*Vp(i+2,j+1,2)+n1*Vp(i,j+2,
2)+m*Vp(i+1,j+2,1)+n2*Vp(i+1,j,2));
end
end

for i=2:L1-1;
%Wn(x,L2,0)
V1(i+1,L2+1,1)=h1*i+h2*L2+b*(1*Vp(i+2,L2+1,1)+n1*Vp(i,L2+1,1)+
m*Vp(i+1,L2+1,1)+n2*Vp(i+1,L2,1));
%Wn(x-1,L2,1)
V2(i+1,L2+1,1)=h1*(i-
1)+(h2*L2)+h1+b*(1*Vp(i+1,L2+1,2)+n1*Vp(i-
1,L2+1,2)+m*Vp(i,L2+1,1)+n2*Vp(i,L2,2));
Vn(i+1,L2+1,1)=min(V1(i+1,L2+1,1),V2(i+1,L2+1,1));
%Wn(x,L2,1)
Vn(i+1,L2+1,2)=h1*i+h2*L2+h1+b*(1*Vp(i+2,L2+1,2)+n1*Vp(i,L2+1,
2)+m*Vp(i+1,L2+1,1)+n2*Vp(i+1,L2,2));

%DECISION
if Vn(i+1,L2+1,1)==V1(i+1,L2+1,1)
K(i+1,L2+1)=0;
else
K(i+1,L2+1)=1;
end
end

%Wn(1,L2,0)
Vn(2,L2+1,1)=h1+(h2*L2)+b*(1*Vp(3,L2+1,1)+n1*Vp(1,L2+1,1)+m*Vp
(2,L2+1,1)+n2*Vp(2,L2,1));
%Wn(1,L2,1)
Vn(2,L2+1,2)=h1+h2*L2+h1+b*(1*Vp(3,L2+1,2)+n1*Vp(1,L2+1,2)+m*V
p(2,L2+1,1)+n2*Vp(2,L2,2));

```

```

for j=2:L2-1;
%Wn(L1,y,0)
V1(L1+1,j+1,1)=h1*L1+h2*j+b*(1*Vp(L1+1,j+1,1)+n1*Vp(L1,j+2,1)+
m*Vp(L1+1,j+1,1)+n2*Vp(L1+1,j,1));
%Wn(L1-1,y,1)
V2(L1+1,j+1,1)=h1*(L1-1)+h2*j+h1+b*(1*Vp(L1+1,j+1,2)+n1*Vp(L1-
1,j+2,2)+m*Vp(L1,j+2,1)+n2*Vp(L1,j,2));
Vn(L1+1,j+1,1)=min(V1(L1+1,j+1,1),V2(L1+1,j+1,1));
%Wn(L1,y,1)
Vn(L1+1,j+1,2)=h1*L1+h2*j+h1+b*(1*Vp(L1+1,j+1,2)+n1*Vp(L1,j+2,
2)+m*Vp(L1+1,j+2,1)+n2*Vp(L1+1,j,2));

%DECISION
if Vn(L1+1,j+1,1)==V1(L1+1,j+1,1)
K(L1+1,j+1)=0;
else
K(L1+1,j+1)=1;
end
end

%Wn(0,L2,0)
Vn(1,L2+1,1)=h2*L2+b*(1*Vp(2,L2+1,1)+n1*Vp(1,L2+1,1)+m*Vp(1,L2
+1,1)+n2*Vp(1,L2,1));
%Wn(0,L2,1)
Vn(1,L2+1,2)=h2*L2+h1+b*(1*Vp(2,L2+1,2)+n1*Vp(1,L2+1,2)+m*Vp(1
,L2+1,1)+n2*Vp(1,L2,2));
%Wn(L1,0,1)
Vn(L1+1,1,2)=h1*L1+h1+b*(1*Vp(L1+1,1,2)+n1*Vp(L1,2,2)+m*Vp(L1+
1,2,1)+n2*Vp(L1+1,1,2));
%Wn(L1,1,1)
Vn(L1+1,2,2)=h1*L1+h2+h1+b*(1*Vp(L1+1,2,2)+n1*Vp(L1,3,2)+m*Vp(L
1+1,3,1)+n2*Vp(L1+1,1,2));

%Wn(L1,0,0)
V1(L1+1,1,1)=h1*L1+b*(1*Vp(L1+1,1,1)+n1*Vp(L1,2,1)+m*Vp(L1+1,1
,1)+n2*Vp(L1+1,1,1));
%Wn(L1-1,0,1)
V2(L1+1,1,1)=h1*(L1-1)+h1+b*(1*Vp(L1+1,1,2)+n1*Vp(L1-
1,2,2)+m*Vp(L1,2,1)+n2*Vp(L1,1,2));
Vn(L1+1,1,1)=min(V1(L1+1,1,1),V2(L1+1,1,1));

%DECISION
if Vn(L1+1,1,1)==V1(L1+1,1,1)
K(L1+1,1)=0;
else
K(L1+1,1)=1;
end

%Wn(L1,1,0)
V1(L1+1,2,1)=h1*L1+h2+b*(1*Vp(L1+1,2,1)+n1*Vp(L1,3,1)+m*Vp(L1+
1,2,1)+n2*Vp(L1+1,1,1));
%Wn(L1-1,1,1)
V2(L1+1,2,1)=h1*(L1-1)+h2+h1+b*(1*Vp(L1+1,2,2)+n1*Vp(L1-
1,3,2)+m*Vp(L1,3,1)+n2*Vp(L1,1,2));

```



```

Vn (L1+1, 2, 1) = min (V1 (L1+1, 2, 1), V2 (L1+1, 2, 1));

%DECISION
if Vn (L1+1, 2, 1) == V1 (L1+1, 2, 1)
K (L1+1, 2) = 0;
else
K (L1+1, 2) = 1;
end

%Wn (L1, L2, 0)
V1 (L1+1, L2+1, 1) = h1*L1+h2*L2+b*(1*Vp (L1+1, L2+1, 1)+n1*Vp (L1, L2+1, 1)+m*Vp (L1+1, L2+1, 1)+n2*Vp (L1+1, L2, 1));
%Wn (L1-1, L2, 1)
V2 (L1+1, L2+1, 1) = h1*(L1-1)+h2*L2+h1+b*(1*Vp (L1+1, L2+1, 2)+n1*Vp (L1-1, L2+1, 2))+m*Vp (L1, L2+1, 1)+n2*Vp (L1, L2, 2);
Vn (L1+1, L2+1, 1) = min (V1 (L1+1, L2+1, 1), V2 (L1+1, L2+1, 1));
%Wn (L1, L2, 1)
Vn (L1+1, L2+1, 2) = h1*L1+h2*L2+h1+b*(1*Vp (L1+1, L2+1, 2)+n1*Vp (L1, L2+1, 2)+m*Vp (L1+1, L2+1, 1)+n2*Vp (L1+1, L2, 2));

%DECISION
if Vn (L1+1, L2+1, 1) == V1 (L1+1, L2+1, 1)
K (L1+1, L2+1) = 0;
else
K (L1+1, L2+1) = 1;
end

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

end

%two stage queuing system
%two dedicated servers and a flexible slower server
%slower server capable of attending both stations

L1 = 100;
L2 = 100;

for i = 0 : L1;
for j = 0 : L2;
for k = 0 : 2;
Vn (i+1, j+1, k+1) = 0;
end
end
end

```

```

%assigning values

Vp=Vn;
kk=1;
rep(kk)=1;
mn=0;
Mn=0;

b=1;

%l<n1+m1 & l<n2+m2
L=0.22;
M1=0.24;
M2=0.23;
N1=0.58;
N2=0.56;

H1=0.37587;
H2=0.32645;

%uniformization
uni=L+N1+N2+M1+M2;
l=L/uni;
n1=N1/uni;
n2=N2/uni;
m1=M1/uni;
m2=M2/uni;
h1=H1/uni;
h2=H2/uni;

while (rep(kk)>0.001*mn)

%Wn(0,0,0)
Vn(1,1,1)=b*(1*Vp(2,1,1)+(1-1)*Vp(1,1,1));
%Wn(0,0,1)
Vn(1,1,2)=h1+b*(1*Vp(2,1,2)+n1*Vp(1,1,2)+m1*Vp(1,2,1)+n2*Vp(1,1,2)+m2*Vp(1,1,2));
%Wn(0,0,2)
Vn(1,1,3)=h2+b*(1*Vp(2,1,3)+n1*Vp(1,1,3)+m1*Vp(1,1,3)+m2*Vp(1,1,1)+n2*Vp(1,1,3));
%Wn(0,1,0)
Vn(1,2,1)=h2+b*(1*Vp(2,2,1)+n1*Vp(1,2,1)+(m1+m2)*Vp(1,2,1)+n2*Vp(1,1,1));
%Wn(0,1,1)
Vn(1,2,2)=h2+h1+b*(1*Vp(2,2,2)+n1*Vp(1,2,2)+m1*Vp(1,3,1)+n2*Vp(1,1,2)+m2*Vp(1,2,2));
%Wn(0,1,2)
Vn(1,2,3)=h2+h2+b*(1*Vp(2,2,3)+n1*Vp(1,2,3)+m1*Vp(1,2,3)+m2*Vp(1,2,1)+n2*Vp(1,1,3));

%Wn(1,0,0)
Vn(2,1,1)=h1+b*(1*Vp(3,1,1)+n1*Vp(1,2,1)+(m1+m2)*Vp(2,1,1)+n2*Vp(2,1,1));
%Wn(1,0,1)

```

```

Vn(2,1,2)=h1+h1+b*(1*Vp(3,1,2)+n1*Vp(1,2,2)+m1*Vp(2,2,1)+n2*Vp(2,1,2)+m2*Vp(2,1,2));
%Wn(1,0,2)
Vn(2,1,3)=h1+h2+b*(1*Vp(3,1,3)+n1*Vp(1,2,3)+m1*Vp(2,1,3)+m2*Vp(2,1,1)+n2*Vp(2,1,3));
%Wn(1,1,0)
Vn(2,2,1)=h1+h2+b*(1*Vp(3,2,1)+n1*Vp(1,3,1)+(m1+m2)*Vp(2,2,1)+n2*Vp(2,1,1));
%Wn(1,1,1)
Vn(2,2,2)=h1+h2+h1+b*(1*Vp(3,2,2)+n1*Vp(1,3,2)+m1*Vp(2,3,1)+n2*Vp(2,1,2)+m2*Vp(2,2,2));
%Wn(1,1,2)
Vn(2,2,3)=h1+h2+h2+b*(1*Vp(3,2,3)+n1*Vp(1,3,3)+m1*Vp(2,2,3)+m2*Vp(2,2,1)+n2*Vp(2,1,3));

for i=2:L1-1;
%Wn(x,0,0)
V1(i+1,1,1)=h1*i+b*(1*Vp(i+2,1,1)+n1*Vp(i,2,1)+(m1+m2)*Vp(i+1,1,1)+n2*Vp(i+1,1,1));
%Wn(x-1,0,1)
V2(i+1,1,1)=h1*(i-1)+h1+b*(1*Vp(i+1,1,2)+n1*Vp(i-1,2,2)+m1*Vp(i,2,1)+n2*Vp(i,1,2)+m2*Vp(i,1,2));
Vn(i+1,1,1)=min(V1(i+1,1,1),V2(i+1,1,1));
%Wn(x,0,1)
Vn(i+1,1,2)=h1*i+h1+b*(1*Vp(i+2,1,2)+n1*Vp(i,2,2)+m1*Vp(i+1,2,1)+n2*Vp(i+1,1,2)+m2*Vp(i+1,1,2));
%Wn(x,0,2)
Vn(i+1,1,3)=h1*i+h2+b*(1*Vp(i+2,1,3)+n1*Vp(i,2,3)+m1*Vp(i+1,1,3)+m2*Vp(i+1,1,1)+n2*Vp(i+1,1,3));

%DECISION
if Vn(i+1,1,1)==V1(i+1,1,1)
K(i+1,1)=0;
else
K(i+1,1)=1;
end

%Wn(x,1,0)
V1(i+1,2,1)=h1*i+h2+b*(1*Vp(i+2,2,1)+n1*Vp(i,3,1)+(m1+m2)*Vp(i+1,2,1)+n2*Vp(i+1,1,1));
%Wn(x-1,1,1)
V2(i+1,2,1)=h1*(i-1)+h2+h1+b*(1*Vp(i+1,2,2)+n1*Vp(i-1,3,2)+m1*Vp(i,3,1)+n2*Vp(i,1,2)+m2*Vp(i,2,2));
Vn(i+1,2,1)=min(V1(i+1,2,1),V2(i+1,2,1));
%Wn(x,1,1)
Vn(i+1,2,2)=h1*i+h2+h1+b*(1*Vp(i+2,2,2)+n1*Vp(i,3,2)+m1*Vp(i+1,3,1)+n2*Vp(i+1,1,2)+m2*Vp(i+1,2,2));
%Wn(x,1,2)
Vn(i+1,2,3)=h1*i+h2+h2+b*(1*Vp(i+2,2,3)+n1*Vp(i,3,3)+m1*Vp(i+1,2,3)+m2*Vp(i+1,2,1)+n2*Vp(i+1,1,3));

%DECISION
if Vn(i+1,2,1)==V1(i+1,2,1)
K(i+1,2)=0;
else

```

```

K(i+1,2)=1;
end
end

for i=2:L1-1;
for j=2:L2-1;
%Wn(x,y,0)
V1(i+1,j+1,1)=(h1*i)+(h2*j)+b*(1*Vp(i+2,j+1,1)+n1*Vp(i,j+2,1)+(m1+m2)*Vp(i+1,j+1,1)+n2*Vp(i+1,j,1));
%Wn(x-1,y,1)
V2(i+1,j+1,1)=h1*(i-1)+(h2*j)+h1+b*(1*Vp(i+1,j+1,2)+n1*Vp(i-1,j+2,2)+m1*Vp(i,j+2,1)+n2*Vp(i,j,2)+m2*Vp(i,j+1,2));
%Wn(x,y-1,2)
V3(i+1,j+1,1)=(h1*i)+(h2*j)+b*(1*Vp(i+2,j,3)+n1*Vp(i,j+1,3)+n2*Vp(i+1,j-1,3)+m1*Vp(i+1,j,3)+m2*Vp(i+1,j,1));
Vn(i+1,j+1,1)=min([V1(i+1,j+1,1),V2(i+1,j+1,1),V3(i+1,j+1,1)]);
;

%DECISION
if Vn(i+1,j+1,1)==V1(i+1,j+1,1)
K(i+1,j+1)=0;
elseif Vn(i+1,j+1,1)==V2(i+1,j+1,1)
K(i+1,j+1)=1;
else
K(i+1,j+1)=2;
end
end
end

for j=2:L2-1;
%Wn(1,y,0)
V1(2,j+1,1)=h1+(h2*j)+b*(1*Vp(3,j+1,1)+n1*Vp(1,j+2,1)+(m1+m2)*Vp(2,j+1,1)+n2*Vp(2,j,1));
%Wn(1,y-1,2)
V2(2,j+1,1)=h1+(h2*j)+b*(1*Vp(3,j,3)+n1*Vp(1,j+1,3)+n2*Vp(2,j-1,3)+m1*Vp(2,j,3)+m2*Vp(2,j,1));
Vn(2,j+1,1)=min(V1(2,j+1,1),V2(2,j+1,1));
%Wn(1,y,1)
Vn(2,j+1,2)=h1+h2*j+h1+b*(1*Vp(3,j+1,2)+n1*Vp(1,j+2,2)+m1*Vp(2,j+2,1)+n2*Vp(2,j,2)+m2*Vp(2,j+1,2));
%Wn(1,y,2)
Vn(2,j+1,3)=h1+h2*j+h2+b*(1*Vp(3,j+1,3)+n1*Vp(1,j+2,3)+m1*Vp(2,j+1,3)+m2*Vp(2,j+1,1)+n2*Vp(2,j,3));

%DECISION
if Vn(2,j+1,1)==V1(2,j+1,1)
K(2,j+1)=0;
else
K(2,j+1)=2;
end
end
end

```

```

for j=2:L2-1;
%Wn(0,y,0)
V1(1,j+1,1)=(h2*j)+b*(1*Vp(2,j+1,1)+n1*Vp(1,j+1,1)+(m1+m2)*Vp(
1,j+1,1)+n2*Vp(1,j,1));
%Wn(0,y-1,2)
V2(1,j+1,1)=h2*(j-1)+h2+b*(1*Vp(2,j,3)+n1*Vp(1,j,3)+n2*Vp(1,j-
1,3)+m1*Vp(1,j,3)+m2*Vp(1,j,1));
Vn(1,j+1,1)=min(V1(1,j+1,1),V2(1,j+1,1));
%Wn(0,y,1)
Vn(1,j+1,2)=h2*j+h1+b*(1*Vp(2,j+1,2)+n1*Vp(1,j+1,2)+m1*Vp(1,j+
2,1)+n2*Vp(1,j,2)+m2*Vp(1,j+1,2));
%Wn(0,y,2)
Vn(1,j+1,3)=h2*j+h2+b*(1*Vp(2,j+1,3)+n1*Vp(1,j+1,3)+n2*Vp(1,j,
3)+m1*Vp(1,j+1,3)+m2*Vp(1,j+1,1));

%DECISION
if Vn(1,j+1,1)==V1(1,j+1,1)
K(1,j+1)=0;
else
K(1,j+1)=2;
end

end

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

for i=2:L1-1;
%Wn(x,L2,0)
V1(i+1,L2+1,1)=h1*i+h2*L2+b*(1*Vp(i+2,L2+1,1)+n1*Vp(i,L2+1,1)+
(m1+m2)*Vp(i+1,L2+1,1)+n2*Vp(i+1,L2,1));
%Wn(x-1,L2,1)
V2(i+1,L2+1,1)=h1*(i-
1)+(h2*L2)+h1+b*(1*Vp(i+1,L2+1,2)+n1*Vp(i-
1,L2+1,2)+m1*Vp(i,L2+1,1)+n2*Vp(i,L2,2)+m2*Vp(i,L2+1,2));
%Wn(x,L2-1,2)
V3(i+1,L2+1,1)=h1*i+h2*(L2-
1)+h2+b*(1*Vp(i+2,L2,3)+n1*Vp(i,L2+1,3)+n2*Vp(i+1,L2-
1,3)+m1*Vp(i+1,L2,3)+m2*Vp(i+1,L2,1));
Vn(i+1,L2+1,1)=min([V1(i+1,L2+1,1),V2(i+1,L2+1,1),V3(i+1,L2+1,
1)]);
%Wn(x,L2,1)
Vn(i+1,L2+1,2)=h1*i+h2*L2+h1+b*(1*Vp(i+2,L2+1,2)+n1*Vp(i,L2+1,
2)+m1*Vp(i+1,L2+1,1)+n2*Vp(i+1,L2,2)+m2*Vp(i+1,L2+1,2));
%Wn(x,L2,2)
Vn(i+1,L2+1,3)=h1*i+h2*L2+h2+b*(1*Vp(i+2,L2+1,3)+n1*Vp(i,L2+1,
3)+n2*Vp(i+1,L2,3)+m1*Vp(i+1,L2+1,3)+m2*Vp(i+1,L2+1,1));

```

```

%DECISION
if Vn(i+1,L2+1,1)==V1(i+1,L2+1,1)
K(i+1,L2+1)=0;
elseif Vn(i+1,L2+1,1)==V2(i+1,L2+1,1)
K(i+1,L2+1)=1;
else
K(i+1,L2+1)=2;
end
end

%Wn(1,L2,0)
V1(2,L2+1,1)=h1+(h2*L2)+b*(1*Vp(3,L2+1,1)+n1*Vp(1,L2+1,1)+(m1+m2)*Vp(2,L2+1,1)+n2*Vp(2,L2,1));
%Wn(1,L2-1,2)
V2(2,L2+1,1)=h1+h2*(L2-1)+h2+b*(1*Vp(3,L2,3)+n1*Vp(1,L2+1,3)+n2*Vp(2,L2-1,3)+m1*Vp(2,L2,3)+m2*Vp(2,L2,1));
Vn(2,L2+1,1)=min(V1(2,L2+1,1),V2(2,L2+1,1));
%Wn(1,L2,1)
Vn(2,L2+1,2)=h1+h2*L2+h1+b*(1*Vp(3,L2+1,2)+n1*Vp(1,L2+1,2)+m1*Vp(2,L2+1,1)+n2*Vp(2,L2,2)+m2*Vp(2,L2+1,2));
%Wn(1,L2,2)
Vn(2,L2+1,3)=h1+h2*L2+h2+b*(1*Vp(3,L2+1,3)+n1*Vp(1,L2+1,3)+m1*Vp(2,L2+1,3)+n2*Vp(2,L2,3)+m2*Vp(2,L2+1,1));

%DECISION
if Vn(2,L2+1,1)==V1(2,L2+1,1)
K(2,L2+1)=0;
else
K(2,L2+1)=2;
end

for j=2:L2-1;
%Wn(L1,y,0)
V1(L1+1,j+1,1)=h1*L1+h2*j+b*(1*Vp(L1+1,j+1,1)+n1*Vp(L1,j+2,1)+(m1+m2)*Vp(L1+1,j+1,1)+n2*Vp(L1+1,j,1));
%Wn(L1-1,y,1)
V2(L1+1,j+1,1)=h1*(L1-1)+h2*j+h1+b*(1*Vp(L1+1,j+1,2)+n1*Vp(L1-1,j+2,2)+m1*Vp(L1,j+2,1)+n2*Vp(L1,j,2)+m2*Vp(L1,j+1,2));
%Wn(L1,y-1,2)
V3(L1+1,j+1,1)=h1*L1+h2*(j-1)+h2+b*(1*Vp(L1+1,j,3)+n1*Vp(L1,j+1,3)+n2*Vp(L1+1,j-1,3)+m1*Vp(L1+1,j,3)+m2*Vp(L1+1,j,1));
Vn(L1+1,j+1,1)=min([V1(L1+1,j+1,1),V2(L1+1,j+1,1),V3(L1+1,j+1,1)]);
%Wn(L1,y,1)
Vn(L1+1,j+1,2)=h1*L1+h2*j+h1+b*(1*Vp(L1+1,j+1,2)+n1*Vp(L1,j+2,2)+m1*Vp(L1+1,j+2,1)+n2*Vp(L1+1,j,2)+m2*Vp(L1+1,j+1,2));
%Wn(L1,y,2)
Vn(L1+1,j+1,3)=h1*L1+h2*j+h2+b*(1*Vp(L1+1,j+1,3)+n1*Vp(L1,j+2,3)+n2*Vp(L1+1,j,3)+m1*Vp(L1+1,j+1,3)+m2*Vp(L1+1,j+1,1));

%DECISION
if Vn(L1+1,j+1,1)==V1(L1+1,j+1,1)

```

```

K(L1+1, j+1)=0;
elseif Vn(L1+1, j+1, 1)==V2(L1+1, j+1, 1)
K(L1+1, j+1)=1;
else
K(L1+1, j+1)=2;
end
end

%Wn(0, L2, 0)
V1(1, L2+1, 1)=h2*L2+b*(1*Vp(2, L2+1, 1)+n1*Vp(1, L2+1, 1)+(m1+m2)*Vp(1, L2+1, 1)+n2*Vp(1, L2, 1));
%Wn(0, L2-1, 2)
V2(1, L2+1, 1)=h2*(L2-1)+h2+b*(1*Vp(2, L2, 3)+n1*Vp(1, L2, 3)+n2*Vp(1, L2-1, 3)+m1*Vp(1, L2, 3)+m2*Vp(1, L2, 1));
Vn(1, L2+1, 1)=min(V1(1, L2+1, 1), V2(1, L2+1, 1));
%Wn(0, L2, 1)
Vn(1, L2+1, 2)=h2*L2+h1+b*(1*Vp(2, L2+1, 2)+n1*Vp(1, L2+1, 2)+m1*Vp(1, L2+1, 1)+n2*Vp(1, L2, 2)+m2*Vp(1, L2+1, 2));
%Wn(0, L2, 2)
Vn(1, L2+1, 3)=h2*L2+h2+b*(1*Vp(2, L2+1, 3)+n1*Vp(1, L2+1, 3)+n2*Vp(1, L2, 3)+m1*Vp(1, L2+1, 3)+m2*Vp(1, L2+1, 1));

%DECISION
if Vn(1, L2+1, 1)==V1(1, L2+1, 1)
K(1, L2+1)=0;
else
K(1, L2+1)=2;
end

%Wn(L1, 0, 0)
V1(L1+1, 1, 1)=h1*L1+b*(1*Vp(L1+1, 1, 1)+n1*Vp(L1, 2, 1)+(m1+m2)*Vp(L1+1, 1, 1)+n2*Vp(L1+1, 1, 1));
%Wn(L1-1, 0, 1)
V2(L1+1, 1, 1)=h1*(L1-1)+h1+b*(1*Vp(L1+1, 1, 2)+n1*Vp(L1-1, 2, 2)+m1*Vp(L1, 2, 1)+n2*Vp(L1, 1, 2)+m2*Vp(L1, 1, 2));
Vn(L1+1, 1, 1)=min(V1(L1+1, 1, 1), V2(L1+1, 1, 1));
%Wn(L1, 0, 1)
Vn(L1+1, 1, 2)=h1*L1+h1+b*(1*Vp(L1+1, 1, 2)+n1*Vp(L1, 2, 2)+m1*Vp(L1+1, 2, 1)+n2*Vp(L1+1, 1, 2)+m2*Vp(L1+1, 1, 2));
%Wn(L1, 0, 2)
Vn(L1+1, 1, 3)=h1*L1+h2+b*(1*Vp(L1+1, 1, 3)+n1*Vp(L1, 2, 3)+m1*Vp(L1+1, 1, 3)+m2*Vp(L1+1, 1, 1)+n2*Vp(L1+1, 1, 3));

%DECISION
if Vn(L1+1, 1, 1)==V1(L1+1, 1, 1)
K(L1+1, 1)=0;
else
K(L1+1, 1)=1;
end

%Wn(L1, 1, 0)
V1(L1+1, 2, 1)=h1*L1+h2+b*(1*Vp(L1+1, 2, 1)+n1*Vp(L1, 3, 1)+(m1+m2)*Vp(L1+1, 2, 1)+n2*Vp(L1+1, 1, 1));
%Wn(L1-1, 1, 1)

```

```

V2 (L1+1, 2, 1)=h1*(L1-1)+h2+h1+b*(1*Vp(L1+1, 2, 2)+n1*Vp(L1-
1, 3, 2)+m1*Vp(L1, 3, 1)+n2*Vp(L1, 1, 2)+m2*Vp(L1, 2, 2));
Vn(L1+1, 2, 1)=min(V1(L1+1, 2, 1), V2(L1+1, 2, 1));
%Wn(L1, 1, 1)
Vn(L1+1, 2, 2)=h1*L1+h2+h1+b*(1*Vp(L1+1, 2, 2)+n1*Vp(L1, 3, 2)+m1*Vp
(L1+1, 3, 1)+n2*Vp(L1+1, 1, 2)+m2*Vp(L1+1, 2, 2));
%Wn(L1, 1, 2)
Vn(L1+1, 2, 3)=h1*L1+h2+h2+b*(1*Vp(L1+1, 2, 3)+n1*Vp(L1, 3, 3)+m1*Vp
(L1+1, 2, 3)+m2*Vp(L1+1, 2, 1)+n2*Vp(L1+1, 1, 3));

%DECISION
if Vn(L1+1, 2, 1)==V1(L1+1, 2, 1)
K(L1+1, 2)=0;
else
K(L1+1, 2)=1;
end

%Wn(L1, L2, 0)
V1(L1+1, L2+1, 1)=h1*L1+h2*L2+b*(1*Vp(L1+1, L2+1, 1)+n1*Vp(L1, L2+1
, 1)+(m1+m2)*Vp(L1+1, L2+1, 1)+n2*Vp(L1+1, L2, 1));
%Wn(L1-1, L2, 1)
V2(L1+1, L2+1, 1)=h1*(L1-
1)+h2*L2+h1+b*(1*Vp(L1+1, L2+1, 2)+n1*Vp(L1-
1, L2+1, 2))+m1*Vp(L1, L2+1, 1)+n2*Vp(L1, L2, 2)+m2*Vp(L1, L2+1, 2);
%Wn(L1, L2-1, 2)
V3(L1+1, L2+1, 1)=h1*L1+h2*(L2-
1)+h2+b*(1*Vp(L1+1, L2, 3)+n1*Vp(L1, L2+1, 3)+n2*Vp(L1+1, L2-
1, 3)+m1*Vp(L1+1, L2, 3)+m2*Vp(L1+1, L2, 1));
Vn(L1+1, L2+1, 1)=min([V1(L1+1, L2+1, 1), V2(L1+1, L2+1, 1), V3(L1+1, L
2+1, 1)]);
%Wn(L1, L2, 1)
Vn(L1+1, L2+1, 2)=h1*L1+h2*L2+h1+b*(1*Vp(L1+1, L2+1, 2)+n1*Vp(L1, L
2+1, 2)+m1*Vp(L1+1, L2+1, 1)+n2*Vp(L1+1, L2, 2)+m2*Vp(L1+1, L2+1, 2))
;
%Wn(L1, L2, 2)
Vn(L1+1, L2+1, 3)=h1*L1+h2*L2+h2+b*(1*Vp(L1+1, L2+1, 3)+n1*Vp(L1, L2
+1, 3)+m1*Vp(L1+1, L2+1, 3)+m2*Vp(L1+1, L2+1, 1)+n2*Vp(L1+1, L2, 3));

%DECISION
if Vn(L1+1, L2+1, 1)==V1(L1+1, L2+1, 1)
K(L1+1, L2+1)=0;
elseif Vn(L1+1, L2+1, 1)==V2(L1+1, L2+1, 1)
K(L1+1, L2+1)=1;
else
K(L1+1, L2+1)=2;
end

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

```