

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ ΚΑΙ ΔΙΟΙΚΗΣΗΣ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΘΕΜΑ: «Συνδυασμένος έλεγχος αποθεμάτων και αποδοχής παραγγελιών σε συστήματα παραγωγής που εξυπηρετούν δύο κατηγορίες πελατών, επιτρέπονται εκκρεμείς παραγγελίες και μπορούν να υποστούν βλάβες.»

Ονοματεπώνυμο: Ανδρόνογλου Χρήστος
Επιβλέπων Καθηγητής: κ. Ιωαννίδης Ευστράτιος

ΧΑΝΙΑ, ΟΚΤΩΜΒΡΙΟΣ 2017

ΠΕΡΙΛΗΨΗ

Σκοπός της παρούσας διπλωματικής εργασίας είναι να εξετάσουμε ένα σύστημα παραγωγής, που παράγει ένα προϊόν το οποίο διατίθεται σε δυο κατηγορίες πελατών. Οι αφίξεις πελατών είναι τυχαίες και ακολουθούν την κατανομή **Poisson**, ενώ οι χρόνοι παραγωγής είναι εκθετικά κατανεμημένοι. Η μονάδα παραγωγής μπορεί να υποστεί βλάβες. Οι χρόνοι μεταξύ βλαβών και οι χρόνοι επισκευής είναι επίσης εκθετικοί.

Προτείνεται μια απλή πολιτική ελέγχου παραγωγής και αποδοχής παραγγελιών που ονομάζουμε Βασικού Αποθέματος — Εξαρτημένου Διπλού Βασικού Ελλείμματος. Στόχος είναι η ανάλυση της προτεινόμενης πολιτικής και η σύγκρισή της με μια υπάρχουσα πολιτική ελέγχου του αποθέματος και των εισερχόμενων παραγγελιών, ώστε να επιλεγεί αυτή που ελαχιστοποιεί το κόστος λειτουργίας του συστήματος. Για το σκοπό αυτό αναπτύσσονται ακριβή μαθηματικά μοντέλα, που περιγράφουν τη λειτουργία του συστήματος.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	5
1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ - ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ..	6
ΚΕΦΑΛΑΙΟ 2: ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ	8
2.1 ΑΛΥΣΙΔΕΣ MARKOV	8
2.1.1 ΕΞΙΣΩΣΕΙΣ ΚΑΤΑΣΤΑΣΕΩΝ ΛΕΙΤΟΥΡΓΙΑΣ.....	11
2.1.2 ΕΞΙΣΩΣΕΙΣ ΚΑΤΑΣΤΑΣΕΩΝ ΑΔΡΑΝΕΙΑΣ	12
2.2 ΕΥΡΕΣΗ ΠΙΘΑΝΟΤΗΤΩΝ ΜΟΝΙΜΗΣ ΚΑΤΑΣΤΑΣΗΣ.....	13
ΚΕΦΑΛΑΙΟ 3: ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ.....	16
3.1 ΚΟΣΤΟΣ ΑΠΟΡΡΙΨΗΣ ΠΕΛΑΤΩΝ ΤΥΠΟΥ 1 ΚΑΙ 2	16
3.2 ΚΟΣΤΟΣ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ.....	17
3.3 ΜΕΣΟ ΠΛΗΘΟΣ ΕΚΚΡΕΜΩΝ ΠΑΡΑΓΓΕΛΙΩΝ.....	17
ΚΕΦΑΛΑΙΟ 4: ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	18
4.1 ΜΕΛΕΤΗ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ ΓΙΑ ΜΕΤΑΒΛΗΤΕΣ ΤΙΜΕΣ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ	18
4.1.1 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ λ_1	19
4.1.2 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ λ_2	20
4.1.3 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ μ	21
4.1.4 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ β	22
4.1.5 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ ε	23
4.1.6 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ h	24
4.1.7 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ b_1	25
4.1.8 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ b_2	26
4.2 ΜΕΛΕΤΗ ΤΗΣ ΚΥΡΤΟΤΗΤΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ ΩΣ ΠΡΟΣ ΤΙΣ ΜΕΤΑΒΛΗΤΕΣ ΕΛΕΓΧΟΥ	27
4.3 ΜΕΛΕΤΗ ΚΥΡΤΟΤΗΤΑΣ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ	29

ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ	32
ΒΙΒΛΙΟΓΡΑΦΙΑ	33
ΠΑΡΑΡΤΗΜΑ - ΧΡΗΣΗ ΛΟΓΙΣΜΙΚΟΥ ΣΕ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C	34

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

Για πολλούς ανθρώπους ο όρος παραγωγή σημαίνει εργοστάσια, μηχανήματα, εξοπλισμό και προϊόντα. Ωστόσο, ο όρος αυτός περιλαμβάνει και τη διαχείριση μη βιομηχανικών δραστηριοτήτων ή υπηρεσιών όπως είναι οι τράπεζες, οι μεταφορές, η εκπαίδευση, κλπ. Έτσι, θα μπορούσαμε να δώσουμε μια πιο γενικευμένη και ολοκληρωμένη έννοια για τον παραπάνω όρο. Παραγωγή λοιπόν, είναι κάθε οργανωμένη δραστηριότητα που αποσκοπεί στην αύξηση της αξίας ή της χρησιμότητας υλικών πραγμάτων ή στην παροχή υπηρεσιών, με την ανάλωση κάποιων πόρων (υλικών, εργασίας κλπ.)

Παραγωγικές διεργασίες συναντάμε σε όλα τα είδη των οργανωμένων δραστηριοτήτων όπως εργοστάσια, γραφεία, ξενοδοχεία και νοσοκομεία. Η διαχείριση αυτών ακριβώς των παραγωγικών διεργασιών που είναι και το βασικό αντικείμενο της επιστήμης της Οργάνωσης και Διοίκησης Παραγωγής (ΟΔΠ) σχετίζεται με την λήψη αποφάσεων έτσι ώστε να εξασφαλίζεται ότι τα παραγόμενα αγαθά ή υπηρεσίες, παράγονται σύμφωνα με τις προδιαγραφές ποιότητας, στις απαιτούμενες ποσότητες, με το απαιτούμενο πρόγραμμα και με το ελάχιστο δυνατόν κόστος.

Τα παραγωγικά συστήματα είναι εκείνα που τροφοδοτούν τις ανθρώπινες κοινωνίες με υλικά αγαθά και υπηρεσίες απαραίτητες για την επιβίωση, την ικανοποίηση των αναγκών αλλά και τον πολιτισμό. Το κόστος κατασκευής και λειτουργίας κάθε παραγωγικού συστήματος αλλά και το επίπεδο παραγωγικότητάς του, καθορίζουν τα περιθώρια επιβιώσής του. Ένα παραγωγικό σύστημα που είναι αποτελεσματικό όσον αφορά το κόστος παραγωγής αλλά και την ποιότητα των προϊόντων του, είναι ένας οργανισμός που μπορεί να επιβιώσει και να αναπτυχθεί στο σύγχρονο, ανταγωνιστικό, εθνικό ή διεθνές περιβάλλον.

Ο έλεγχος της παραγωγής διαδραματίζει σημαντικό ρόλο στη βελτιστοποίηση της παραγωγικής διαδικασίας μιας βιομηχανικής επιχείρησης με απώτερο σκοπό τη σημαντική μείωση του κόστους παραγωγής ούτως ώστε να εξασφαλίσει την επιβίωσή της στη σύγχρονη ανταγωνιστική αγορά στη διάρκεια του χρόνου. Οι σύγχρονες επιχειρήσεις ανάλογα με την πολυπλοκότητα των παραγωγικών τους συστημάτων και την παραγωγική τους δυναμικότητα προσπαθούν να οργανώσουν με τέτοιο τρόπο την παραγωγική τους διαδικασία ώστε να παράγουν ανταγωνιστικά προϊόντα ή υπηρεσίες για τον σύγχρονο καταναλωτή μειώνοντας ταυτόχρονα το κόστος και το χρόνο παραγωγής.

Η αυξημένη πολυπλοκότητα των παραγωγικών συστημάτων έχει οδηγήσει στην καθιέρωση μιας προσέγγισης κατανεμημένου ελέγχου των διαφόρων δραστηριοτήτων τους, όπου κάθε δραστηριότητα αντιμετωπίζεται και ελέγχεται σαν να ήταν εντελώς ανεξάρτητη από τις υπόλοιπες. Τα τελευταία χρόνια έχουν γίνει αντιληπτές οι

αλληλεπιδράσεις μεταξύ αυτών των δραστηριοτήτων αλλά και η σημασία μιας πιο ολοκληρωμένης προσέγγισης ελέγχου.

Σε αυτή την εργασία εξετάζουμε τις συνέπειες από την χρήση συνδυασμένου ελέγχου παραγωγής και αποδοχής παραγγελιών σε συστήματα, που παράγουν ένα τύπο προϊόντος, για να ικανοποιήσουν τη ζήτηση δύο διαφορετικών κατηγοριών πελατών.

1.1 ΑΝΤΙΚΕΙΜΕΝΟ ΕΡΓΑΣΙΑΣ - ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ

Στην παρούσα εργασία εξετάζουμε ένα στοχαστικό σύστημα παραγωγής το οποίο παράγει δύο προϊόντα και τα διαθέτει σε δύο κατηγορίες πελατών ή σε δύο διαφορετικές αγορές αντίστοιχα. Το ένα προϊόν ή η πρώτη κατηγορία πελατών είναι πιο σημαντική από τη δεύτερη κατηγορία πελατών, από την άποψη ότι η επιχείρηση έχει μεγαλύτερο περιθώριο κέρδους πωλώντας το προϊόν της πρώτης κατηγορίας και αντίστοιχα μεγαλύτερο κόστος όταν καθυστερεί να εξυπηρετήσει έναν πελάτη αυτής της κατηγορίας.

Ο στόχος είναι να καθοριστεί πότε θα σταματάει η παραγωγή και πότε μια παραγγελία θα ικανοποιείται, θα μπαίνει στην αναμονή με κριτήριο την ελαχιστοποίηση του κόστους λόγω αποθεματοποίησης, βλάβης της μηχανής και εκκρεμών παραγγελιών.

Στη συγκεκριμένη έρευνα οι αφίξεις πελατών τύπου 1 και 2 είναι τυχαίες και ακολουθούν την κατανομή **Poisson** με ρυθμούς λ_1 και λ_2 αντίστοιχα. Κάθε πελάτης μπορεί να ζητήσει το πολύ μια μονάδα προϊόντος.

Οι χρόνοι παραγωγής προϊόντων είναι εκθετικά κατανεμημένοι με μέση τιμή $\frac{1}{\mu}$.

Το σύστημα παραγωγής μπορεί να υποστεί και βλάβες. Οι χρόνοι μεταξύ των βλαβών είναι εκθετικά κατανεμημένοι με ρυθμό β . Οι χρόνοι επισκευής είναι επίσης εκθετικά κατανεμημένοι με ρυθμό ε . Καθώς το σύστημα παραγωγής είναι αδρανές, δηλαδή έχει υποστεί βλάβη, δεν παράγει προϊόντα αλλά μπορεί να εξυπηρετεί πελάτες για όσο υπάρχει διαθέσιμο απόθεμα.

Η πολιτική που εξετάζεται σε αυτή την εργασία χρησιμοποιεί 4 κατώφλια ελέγχου της διαδικασίας παραγωγής και της αποδοχής παραγγελιών.

- 1) Κατώφλι βασικού αποθέματος (**S**)
- 2) Κατώφλι αποθέματος προτεραιότητας (**r**)
- 3) Κατώφλι εκκρεμών παραγγελιών πελατών τύπου 1 (**c₁**)
- 4) Κατώφλι εκκρεμών παραγγελιών πελατών τύπου 2 (**c₂**)

Για τα συστήματα παραγωγής που μελετάμε η βέλτιστη πολιτική είναι μία πολιτική τύπου κατώφλιου και την ονομάζουμε πολιτική Βασικού Αποθέματος – Εξαρτημένου διπλού Βασικού Ελλείμματος (BA-ΕΔΒΕ). Το σύστημα παράγει προϊόντα όταν ο αριθμός των προϊόντων του αποθέματός είναι μικρότερος του S και σταματάει την παραγωγή όταν το απόθεμα γίνει S . Έτσι λοιπόν το S αποτελεί το μέγιστο όριο αποθέματος καθώς και ένα κατώφλι που έχει σαν σκοπό τη φύλαξη του συστήματος από ελλείψεις αποθέματος αλλά και από την υπέρμετρη παραγωγή προϊόντων.

Όσον αφορά το κατώφλι προτεραιότητας είναι αρμόδιο να διασφαλίζει την εξυπηρέτηση παραγγελιών πελατών τύπου **1** με την μικρότερη καθυστέρηση καθώς έχουν προτεραιότητα αλλά και την εξυπηρέτηση πελατών τύπου **2** άμεσα, μόνο και μόνο όταν το απόθεμα είναι μεγαλύτερο από το κατώφλι r . Δηλαδή οι πελάτες τύπου **2** εξυπηρετούνται απευθείας από το απόθεμα όταν η στάθμη του αποθέματος είναι μεγαλύτερη του r , διαφορετικά μπαίνουν σε λίστα αναμονής. Όταν υπάρχουν εκκρεμείς παραγγελίες και των δύο κατηγοριών ή το απόθεμα είναι μικρότερο του r , δίδεται προτεραιότητα στους πελάτες τύπου **1** δηλαδή εξυπηρετούμε αυτές τις παραγγελίες ή αυξάνουμε το απόθεμα ανεξάρτητα από το ύψος των εκκρεμών παραγγελιών τύπου **2**.

Καθοριστικό ρόλο στη σωστή λειτουργία του συστήματος παίζουν τα κατώφλια εκκρεμών παραγγελιών πελατών τύπου **1** και **2**. Στη περίπτωση έλλειψης αποθέματος λόγω βλάβης της μηχανής το παραγωγικό σύστημα δέχεται μία παραγγελία πελάτη τύπου **1** εφόσον οι εκκρεμείς παραγγελίες του πελάτη τύπου **1** δεν υπερβαίνουν το κατώφλι εκκρεμών παραγγελιών c_1 . Εάν υπάρχει έτοιμο απόθεμα η παραγγελία ικανοποιείται άμεσα. Μία παραγγελία πελατών τύπου **2** γίνεται αποδεκτή και ικανοποιείται εφόσον το καθαρό έλλειμμα, που είναι το σύνολο των παραγγελιών και από τις δύο κατηγορίες πελατών μείον το υπάρχον απόθεμα, είναι μικρότερο από το κατώφλι c_2 .

ΚΕΦΑΛΑΙΟ 2: ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΠΑΡΑΓΩΓΗΣ

Το σύστημα παραγωγής της παρούσας διπλωματικής έρευνας έχει παρουσιαστεί αναλυτικά στο προηγούμενο κεφάλαιο. Μιας και η διάρκεια παραγωγής μιας μονάδας προϊόντος όπως προείπαμε είναι εκθετικά κατανομημένη με μέση τιμή $\frac{1}{\mu}$ και οι αφίξεις πελατών τύπου **1** και τύπου **2** ακολουθούν την κατανομή **Poisson** με ρυθμούς λ_1 και λ_2 αντίστοιχα, έχουμε τη δυνατότητα να μοντελοποιήσουμε το σύστημα παραγωγής ως μια αλυσίδα **Markov** συνεχούς χρόνου.

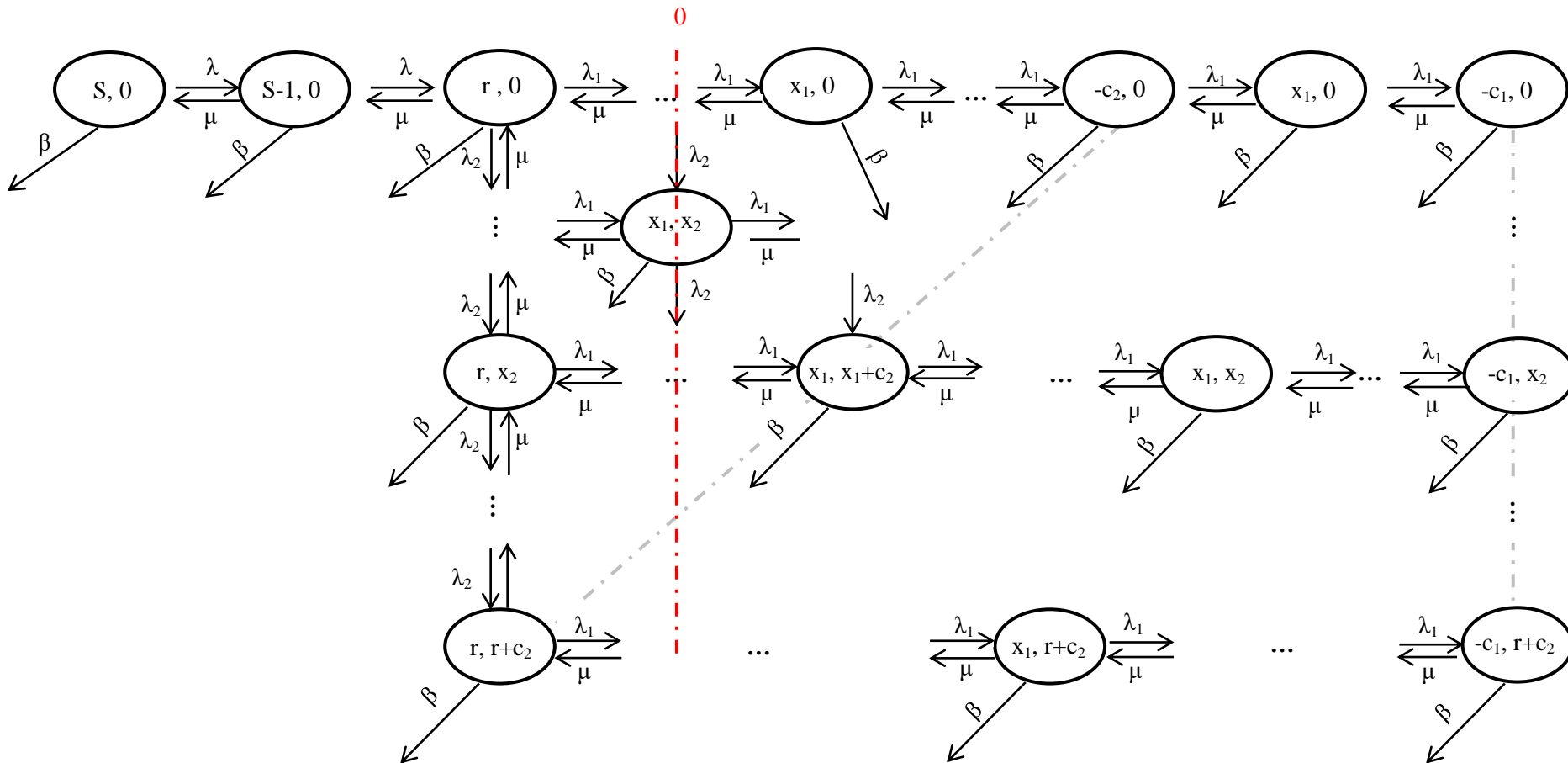
Οι καταστάσεις του συστήματος περιγράφονται από τις παρακάτω μεταβλητές:

- 1) Μεταβλητή x_1 η οποία αντιπροσωπεύει το τρέχον απόθεμα του προϊόντος τύπου **1** ή τις εκκρεμείς παραγγελίες για το ίδιο προϊόν.
- 2) Μεταβλητή x_2 η οποία αντιπροσωπεύει τις εκκρεμείς παραγγελίες του προϊόντος τύπου **2**.
- 3) Μεταβλητή i η οποία παίρνει τη τιμή **0** για να δηλώσει την αδράνεια του συστήματος λόγω βλάβης και τιμή **1** για να δηλώσει τη λειτουργία του συστήματος παραγωγής.

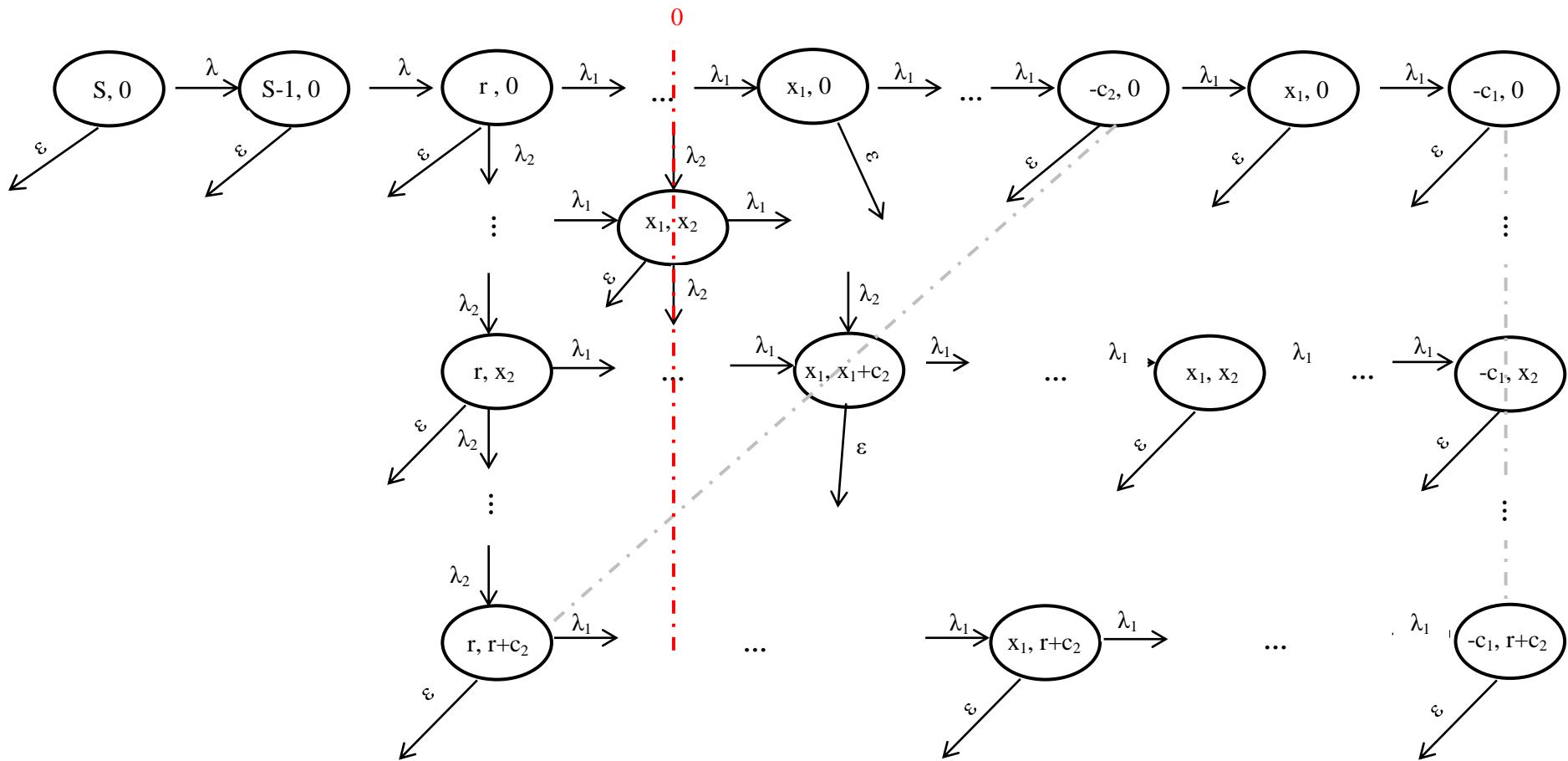
Όταν το $x_1 > 0$ τότε υπάρχει διαθέσιμο απόθεμα για την εξυπηρέτηση πελατών τύπου **1**, ενώ στην αντίθετη περίπτωση όταν δηλαδή το $x_1 < 0$ υπάρχουν εκκρεμείς παραγγελίες πελατών τύπου **1** στο σύστημα ίσες με $-x_1$.

2.1 ΑΛΥΣΙΔΕΣ MARKOV

Η εξέλιξη του συστήματος μπορεί να αναπαρασταθεί από μια αλυσίδα **Markov**, τα διαγράμματα καταστάσεων της οποίας παρουσιάζονται στα παρακάτω σχήματα:



Σχήμα 3.1: Αλυσίδες **Markov** λειτουργικής κατάστασης ($i = 1$).



Σχήμα 3.2: Αλυσίδες **Markov** σε κατάσταση αδράνειας ($i = 0$)

Πιο συγκεκριμένα, αφού το σύστημα παραγωγής που πραγματεύεται η παρούσα διπλωματική, περιγράφεται ως μία αλυσίδα **Markov** συνεχούς χρόνου οι πιθανότητες μόνιμης κατάστασης του συστήματος θα πρέπει να ικανοποιούν τις εξισώσεις **Champan – Kolmogorov (C-K)** για κάθε κατάσταση **k** του συστήματος:

$$P(\mathbf{k}) \times (\text{ρυθμός εξόδου από την κατάσταση } \mathbf{k}) =$$

$$\sum_{\text{όλες οι καταστάσεις } \mathbf{i}} P(\mathbf{i}) \times (\text{ρυθμός μετάβασης από } \mathbf{i} \text{ σε } \mathbf{k}).$$

Έτσι, πιο αναλυτικά στο συγκεκριμένο σύστημα έχουμε τις εξισώσεις καταστάσεων λειτουργίας και καταστάσεων αδράνειας.

2.1.1 ΕΞΙΣΩΣΕΙΣ ΚΑΤΑΣΤΑΣΕΩΝ ΛΕΙΤΟΥΡΓΙΑΣ

$$\text{Όπου } \lambda = \lambda_1 + \lambda_2$$

$$P(S,0,1)(\lambda + \beta) = P(S-1,0,1)\mu + P(S,0,0)\epsilon \quad (1)$$

$$P(x_1,0,1)(\lambda + \mu + \beta) = P(x_1+1,0,1)(\lambda) + P(x_1-1,0,1)\mu + P(x_1,0,0)\epsilon, \quad r < x_1 < S \quad (2)$$

$$P(r,0,1)(\lambda + \mu + \beta) = P(r+1,0,1)(\lambda) + P(r-1,0,1)\mu + P(r,1,1)\mu + P(r,0,0)\epsilon \quad (3)$$

$$P(r,x_2,1)(\lambda + \mu + \beta) = P(r,x_2-1,1)\lambda_2 + P(r,x_2+1,1)\mu + P(r-1,x_2,1)\mu + P(r,x_2,0)\epsilon, \quad 0 < x_2 \leq r + c_2 \quad (4)$$

$$P(r,r+c_2,1)(\lambda_1 + \beta + \mu) = P(r,r+c_2-1,1)\lambda_2 + P(r-1,r+c_2,1)\mu + P(r,r+c_2,0)\epsilon \quad (5)$$

$$P(x_1,x_2,1)(\lambda + \mu + \beta) = P(x_1+1,x_2,1)\lambda_1 + P(x_1-1,x_2,1)\mu + P(x_1,x_2-1,1)\lambda_2 + P(x_1,x_2,0)\epsilon, \quad -c_2 < x_1 < r \quad (6)$$

και $0 < x_2 < x_1 + c_2$

$$P(x_1,0,1)(\lambda + \beta + \mu) = P(x_1+1,0,1)\lambda_1 + P(x_1,0,0)\epsilon + P(x_1-1,0,1)\mu, \quad -c_2 < x_1 < r \quad (7)$$

$$P(x_1,x_1+c_2,1)(\lambda_1 + \mu + \beta) = P(x_1+1,x_1+c_2,1)\lambda_1 + P(x_1,x_1+c_2-1,1)\lambda_2 + P(x_1-1,x_1+c_2,1)\mu + P(x_1,x_1+c_2,0)\epsilon, \quad -c_2 < x_1 < r, \quad x_2 = x_1 + c_2 \quad (8)$$

$$P(-c_2,0,1)(\lambda_1 + \beta + \mu) = P(-c_2+1,0,1)\lambda_1 + P(-c_2,0,0)\epsilon + P(-c_2-1,0,1)\mu \quad (9)$$

$$P(x_1,0,1)(\lambda_1 + \beta + \mu) = P(x_1+1,0,1)\lambda_1 + P(x_1,0,0)\epsilon + P(x_1-1,0,1)\mu, \quad -c_2 < x_1 < -c_1 \quad (10)$$

$$P(-c_1,0,1)(\beta + \mu) = P(-c_1+1,x_2,1)\lambda_1 + P(-c_1,0,0)\epsilon \quad (11)$$

$$P(-c_1, x_2, 1)(\mu + \beta) = P(-c_1 + 1, x_2, 1)\lambda_1 + P(-c_1, x_2, 0)\varepsilon, -c_1 < x_2 < r + c_2 \quad (12)$$

$$P(-c_1, r + c_2, 1)(\beta + \mu) = P(-c_1 + 1, r + c_2, 1)\lambda_1 + P(-c_1, r + c_2, 0)\varepsilon \quad (13)$$

$$P(x_1, r + c_2, 1)(\lambda_1 + \beta + \mu) = P(x_1 + 1, r + c_2, 1)\lambda_1 + P(x_1, r + c_2, 0)\varepsilon + P(x_1 - 1, r + c_2, 1)\mu, r < x_1 < -c_1 \quad (14)$$

$$P(x_1, x_2, 1)(\lambda_1 + \mu + \beta) = P(x_1 + 1, x_2, 1)\lambda_1 + P(x_1 - 1, x_2, 1)\mu + P(x_1, x_2, 0)\varepsilon, -c_1 < x_1 < x_2 - c_2 \text{ και } 0 < x_2 < r + c_2 \quad (15)$$

2.1.2 ΕΞΙΣΩΣΕΙΣ ΚΑΤΑΣΤΑΣΕΩΝ ΑΔΡΑΝΕΙΑΣ

Όπου $\lambda = \lambda_1 + \lambda_2$

$$P(S, 0, 0)(\lambda + \varepsilon) = P(S, 0, 1)\beta \quad (16)$$

$$P(x_1, 0, 0)(\lambda + \mu + \varepsilon) = P(x_1 + 1, 0, 0)(\lambda) + P(x_1, 0, 1)\beta, r < x_1 < S \quad (17)$$

$$P(r, 0, 0)(\lambda + \varepsilon) = P(r + 1, 0, 0)(\lambda) + P(r, 0, 1)\beta \quad (18)$$

$$P(r, x_2, 0)(\lambda + \varepsilon) = P(r, x_2 - 1, 0)\lambda_2 + P(r, x_2, 1)\beta, 0 < x_2 \leq r + c_2 \quad (19)$$

$$P(r, r + c_2, 0)(\lambda_1 + \varepsilon) = P(r, r + c_2 - 1, 0)\lambda_2 + P(r, r + c_2, 1)\beta \quad (20)$$

$$P(x_1, x_2, 0)(\lambda + \varepsilon) = P(x_1 + 1, x_2, 0)\lambda_1 + P(x_1, x_2 - 1, 0)\lambda_2 + P(x_1, x_2, 1)\beta, -c_2 < x_1 < r \text{ και } 0 < x_2 < x_1 + c_2 \quad (21)$$

$$P(x_1, 0, 0)(\lambda + \varepsilon) = P(x_1 + 1, 0, 0)\lambda_1 + P(x_1, 0, 1)\beta, -c_2 < x_1 < r \quad (22)$$

$$P(x_1, x_1 + c_2, 0)(\lambda_1 + \varepsilon) = P(x_1 + 1, x_1 + c_2, 0)\lambda_1 + P(x_1, x_1 + c_2 - 1, 0)\lambda_2 + P(x_1, x_1 + c_2, 1)\beta, -c_2 < x_1 < r, x_2 = x_1 + c_2 \quad (23)$$

$$P(-c_2, 0, 0)(\lambda_1 + \varepsilon) = P(-c_2 + 1, 0, 0)\lambda_1 + P(-c_2, 0, 1)\beta \quad (24)$$

$$P(x_1, 0, 0)(\lambda_1 + \varepsilon) = P(x_1 + 1, 0, 0)\lambda_1 + P(x_1, 0, 1)\beta, -c_2 < x_1 < -c_1 \quad (25)$$

$$P(-c_1, 0, 0)\varepsilon = P(-c_1 + 1, x_2, 0)\lambda_1 + P(-c_1, 0, 1)\beta \quad (26)$$

$$P(-c_1, x_2, 0)\varepsilon = P(-c_1 + 1, x_2, 0)\lambda_1 + P(-c_1, x_2, 1)\beta, -c_1 < x_2 < r + c_2 \quad (27)$$

$$P(-c_1, r + c_2, 0)\varepsilon = P(-c_1 + 1, r + c_2, 0)\lambda_1 + P(-c_1, r + c_2, 1)\beta \quad (28)$$

$$P(x_1, r + c_2, 0)(\lambda_1 + \varepsilon) = P(x_1 + 1, r + c_2, 0)\lambda_1 + P(x_1, r + c_2, 1)\beta, r < x_1 < -c_1 \quad (29)$$

$$P(x_1, x_2, 0)(\lambda_1 + \varepsilon) = P(x_1 + 1, x_2, 0)\lambda_1 + P(x_1, x_2, 1)\beta, -c_1 < x_1 < x_2 - c_2 \text{ και } 0 < x_2 < r + c_2 \quad (30)$$

Οι πιθανότητες μόνιμης κατάστασης $P(x_1, x_2, i)$, είναι εφικτό να υπολογιστούν με την επίλυση των εξισώσεων **Champan – Kolmogorov (C-K)**. Η επαναληπτική διαδικασία επίλυσης, που χρησιμοποιήθηκε σε αυτή την εργασία περιγράφεται αναλυτικά στη συνέχεια.

2.2 ΕΥΡΕΣΗ ΠΙΘΑΝΟΤΗΤΩΝ ΜΟΝΙΜΗΣ ΚΑΤΑΣΤΑΣΗΣ

Για την εύρεση των πιθανοτήτων μόνιμης κατάστασης θα ακολουθήσουμε τη μέθοδο διαδοχικών αντικαταστάσεων. Αρχικά εκφράζουμε όλες τις πιθανότητες συναρτήσει των πιθανοτήτων $P(-c_1, x_2, i)$. Συνεπώς, οι εκφράσεις είναι της μορφής:

$$P(x_1, x_2, i) = \underline{A}^T(x_1, x_2, i)\underline{D} \quad (31)$$

όπου \underline{D} , είναι το διάνυσμα των πιθανοτήτων $P(-c_1, x_2, i)$ και είναι το διάνυσμα των συντελεστών. Τα διανύσματα αυτά περιέχουν κατά συνέπεια $(2(r + c_2) + 1)$ στοιχεία.

Για να εκφράσουμε όλες τις πιθανότητες μόνιμης κατάστασης συναρτήσει των συνοριακών πιθανοτήτων για $x_1 = -c_1$ ακολουθούμε την παρακάτω διαδικασία:

➤ 1^ο Βήμα: Κατασκευή του πίνακα \underline{A} .

Αρχίζοντας από τις καταστάσεις της περιοχής $x_1 = -c_1$ και $0 \leq x_2 < r + c_2$ εκχωρούμε στις θέσεις του πίνακα $A[2x_2]$ για τη λειτουργική κατάσταση ($i = 1$) και $A[2x_2 + 1]$ για την κατάσταση αδράνειας ($i = 0$) την τιμή 1 ενώ σε όλες τις υπόλοιπες θέσεις του πίνακα εκχωρούμε τη τιμή 0. Για τη περιοχή $x_1 = -c_1 + 1$ και $0 \leq x_2 < r + c_2$ λύνουμε τις εξισώσεις (11) και (26) για $i = 0, 1$ ως προς τις συνοριακές πιθανότητες αντίστοιχα:

$$P(-c_1 + 1, x_2, 1) = (1/\lambda_1)[P(-c_1, x_2, 1)(\beta + \mu) - P(-c_1, x_2, 0)\varepsilon]$$

$$P(-c_1 + 1, x_2, 0) = (1/\lambda_1)[P(-c_1, x_2, 0)\varepsilon - P(-c_1, x_2, 1)\beta]$$

Αντικαθιστώντας τις εξισώσεις (11) και (26) στην εξίσωση (31) έχουμε τις παρακάτω εκφράσεις:

$$P(-c_1+1, x_2, 1) = (1/\lambda_1) [\underline{A}(-c_1, x_2, 1)^T (\beta + \mu) - \underline{A}(-c_1, x_2, 0)^T \varepsilon] \underline{D} \text{ και}$$

$$P(-c_1+1, x_2, 0) = (1/\lambda_1) [\underline{A}(-c_1, x_2, 0)^T \varepsilon - \underline{A}(-c_1, x_2, 1)^T \beta] \underline{D}$$

Για τις καταστάσεις της περιοχής όπου $0 \leq x_2 < r+c_2$ και $x_1 > -c_1$ έως $x_1 \leq x_2-c_2$, λύνουμε τις εξισώσεις (15) και (30) για να πάρουμε τις ακόλουθες εκφράσεις:

$$P(x_1+1, x_2, 0) = (1/\lambda_1) [\underline{A}(x_1, x_2, 0)^T (\lambda_1 + \varepsilon) - \underline{A}(x_1, x_2, 1)^T \beta] \underline{D} \text{ και}$$

$$P(x_1+1, x_2, 1) = (1/\lambda_1) [\underline{A}(x_1, x_2, 1)^T (\lambda_1 + \mu + \beta) - \underline{A}(x_1-1, x_2, 1)^T \mu - \underline{A}(x_1, x_2, 0)^T \varepsilon] \underline{D}.$$

Όταν το $x_1 = r+c_2-1$ και το $x_2 = r$ λύνουμε τις εξισώσεις (5) και (20) και έχουμε:

$$P(r, r+c_2-1, 0) = (1/\lambda_2) [\underline{A}(r, r+c_2, 0)^T (\lambda_1 + \varepsilon) - \underline{A}(r, r+c_2, 1)^T \beta] \underline{D} \text{ και}$$

$$P(r, r+c_2-1, 1) = (1/\lambda_2) [\underline{A}(r, r+c_2, 1)^T (\lambda_1 + \beta + \mu) - \underline{A}(r-1, r+c_2, 1)^T \mu - \underline{A}(r, r+c_2, 0)^T \varepsilon] \underline{D}.$$

Στη συνέχεια, για τις καταστάσεις της περιοχής όπου $1-c_2 < x_1 \leq r$ και $x_2 = x_1+c_2-1$ χρησιμοποιούμε τις εξισώσεις (8) και (23) και οδηγούμαστε στις εκφράσεις:

$$P(x_1, x_1+c_2-1, 0) = (1/\lambda_2) [\underline{A}(x_1, x_1+c_2, 0)^T (\lambda_1 + \varepsilon) - \underline{A}(x_1+1, x_1+c_2, 0)^T \lambda_1 - \underline{A}(x_1, x_1+c_2, 1)^T \beta] \underline{D} \text{ και}$$

$$P(x_1, x_1+c_2-1, 1) = (1/\lambda_2) [\underline{A}(x_1, x_1+c_2, 1)^T (\lambda_1 + \mu + \beta) - \underline{A}(x_1+1, x_1+c_2, 1)^T \lambda_1 - \underline{A}(x_1-1, x_1+c_2, 1)^T \mu - \underline{A}(x_1, x_1+c_2, 0)^T \varepsilon] \underline{D}.$$

Έπειτα, για τις καταστάσεις κάθε διαγωνίου που ορίζεται από την εξίσωση $x_2 = x_1 + c_2 - j$, $j-c_2 \leq x_1 \leq r$, $j = 2, \dots, r + c_2 - 1$, χρησιμοποιούμε τις εξισώσεις (4) και (19) για να εκτιμήσουμε τις εκφράσεις των πιθανοτήτων $P(r, r+c_2-j-1, i)$:

$$P(r, r+c_2-1, 0) = (1/\lambda_2) [\underline{A}(r, x_2, 0)^T (\lambda + \varepsilon) - \underline{A}(r, x_2, 1)^T \beta] \underline{D} \text{ και}$$

$$P(r, x_2-1, 1) = (1/\lambda_2) [\underline{A}(r, x_2, 1)^T (\lambda + \mu + \beta) - \underline{A}(r, x_2+1, 1)^T \mu - \underline{A}(r-1, x_2, 1)^T \mu - \underline{A}(r, x_2, 0)^T \varepsilon] \underline{D}.$$

και στη συνέχεια λύνουμε τις εξισώσεις (6) και (21) για να εκφράσουμε τις πιθανότητες:

$$P(x_1, x_2-1, 0) = (1/\lambda_2) [\underline{A}(x_1, x_2, 0)^T (\lambda + \varepsilon) - \underline{A}(x_1+1, x_2, 0)^T \lambda_1 + \underline{A}(x_1, x_2, 1)^T \beta] \underline{D} \text{ και}$$

$$P(x_1, x_2-1, 1) = (1/\lambda_2) [\underline{A}(x_1, x_2, 1)^T (\lambda + \mu + \beta) - \underline{A}(x_1+1, x_2, 1)^T \lambda_1 - \underline{A}(x_1-1, x_2, 1)^T \mu - \underline{A}(x_1, x_2, 0)^T \varepsilon] \underline{D}.$$

Για να εκφράσουμε τις καταστάσεις $(r+1, 0, i)$ συναρτήσει του \underline{D} παίρνουμε τις εξισώσεις (3) και (18) που μας οδηγούν στις ακόλουθες εκφράσεις:

$$P(r+1, 0, 0) = (1/\lambda) [\underline{A}(r, 0, 0)^T (\lambda + \varepsilon) - \underline{A}(r, 0, 1)^T \beta] \underline{D} \text{ και}$$

$$P(r+1, 0, 1) = (1/\lambda) [\underline{A}(r, 0, 1)^T (\lambda + \mu + \beta) - \underline{A}(r-1, 0, 1)^T \mu - \underline{A}(r, 1, 1)^T \mu - \underline{A}(r, 0, 0)^T \varepsilon] \underline{D}.$$

Τέλος, για την περιοχή των καταστάσεων για $r+1 < x_1 \leq S$ και $x_2 = 0$ χρησιμοποιούμε τις εξισώσεις (2) και (18) και παίρνουμε τις παρακάτω εκφράσεις:

$$P(x_1+1,0,0) = (1/\lambda)[\underline{A}(x_1,0,0)^T(\lambda+\mu+\epsilon) - \underline{A}(x_1,0,1)^T\beta]\underline{D} \text{ και}$$

$$P(x_1+1,0,1) = (1/\lambda)[\underline{A}(x_1,0,1)^T(\lambda+\mu+\beta) - \underline{A}(x_1-1,0,1)^T\mu - \underline{A}(x_1,0,0)^T\epsilon]\underline{D}.$$

➤ **2^ο Βήμα: Κατασκευή του πίνακα \underline{D} .**

Για τον υπολογισμό του διανύσματος πιθανοτήτων \underline{D} , θα πρέπει να επιλύσουμε ένα σύστημα $2((r+c_2)+1)$ εξισώσεων. Για το σκοπό αυτό θα αξιοποιήσουμε τις εξισώσεις που δεν χρησιμοποιήσαμε ως τώρα, αλλά και την εξίσωση κανονικοποίησης, που μας λέει ότι το άθροισμα όλων των πιθανοτήτων ισούται με την μονάδα.

Οι εξισώσεις που χρησιμοποιήθηκαν για τον υπολογισμό του διανύσματος \underline{D} , είναι: οι εξισώσεις (9) και (24) για $x_1 = -c_2$, οι εξισώσεις (7) και (22) για $c_2 \leq x_1 \leq r$, η εξίσωση (1) και η εξίσωση κανονικοποίησης. Επιλύοντας το σύστημα αυτών των εξισώσεων υπολογίζουμε το διάνυσμα \underline{D} .

➤ **3^ο Βήμα: Χρήση των σχέσεων $P(x_1, x_2, i) = \underline{A}^T(x_1, x_2, i) \times \underline{D}$ και εύρεση των πιθανοτήτων μόνιμης κατάστασης για κάθε κατάσταση του συστήματός μας, λειτουργική και μη.**

ΚΕΦΑΛΑΙΟ 3: ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ

Το μέτρο απόδοσης του συστήματος παραγωγής της μελέτης μας, είναι η συνάρτηση μέσου κόστους η οποία προσδίδει το συνολικό αναμενόμενο κόστος λειτουργίας του συστήματος στη μονάδα του χρόνου.

Το συνολικό κόστος ή συνάρτηση κόστους δίνεται από τον παρακάτω τύπο:

$$K = P_1LS_1 + P_2LS_2 + hH + b_1B_1 + b_2B_2$$

Όπως είναι φανερό, η συνάρτηση κόστους απαρτίζεται και από κάποια επιμέρους κόστη, η αναφορά των οποίων παρουσιάζεται παρακάτω.

3.1 ΚΟΣΤΟΣ ΑΠΟΡΡΙΨΗΣ ΠΕΛΑΤΩΝ ΤΥΠΟΥ 1 ΚΑΙ 2

Γνωρίζοντας ότι οι πελάτες τύπου 1 είναι πιο σημαντικοί από τους πελάτες τύπου 2 στο σύστημα παραγωγής που μελετάμε, το μοναδιαίο κόστος απόρριψης πελατών της κατηγορίας 1, P_1 θα είναι μεγαλύτερο από το μοναδιαίο κόστος απόρριψης πελατών της κατηγορίας 2, P_2 και αυτός είναι και ένας από τους λόγους που δίνουμε προτεραιότητα στη εξυπηρέτησή τους. Δηλαδή ισχύει ότι $P_1 > P_2$. Για την εύρεση όμως του κόστους απόρριψης, είναι απαραίτητη και η γνώση του ρυθμού με τον οποίο χάνονται οι πελάτες των δύο κατηγοριών στη μονάδα του χρόνου, δηλαδή το ισούται με τη συνολική πιθανότητα, το σύστημα να βρίσκεται σε κατάσταση που δεν μπορεί να εξυπηρετήσει τη συγκεκριμένη κατηγορία πελατών, επί ρυθμό άφιξης πελατών τύπου 1, λ_1 . Αυτό συμβαίνει στην περίπτωση όπου το σύστημά μας βρίσκεται στις καταστάσεις που ισχύει ότι $x_1 = c_1$ και δίνεται από το τύπο:

$$LS_1 = \lambda_1 \left(\sum_{x_2=0}^{r+c_2} \sum_{i=0}^1 P(-c_1, x_2, i) \right)$$

Αντίστοιχα, οι παραγγελίες της δεύτερης κατηγορίας πελατών απορρίπτονται με ρυθμό που ισούται με τη συνολική πιθανότητα, το σύστημα να βρίσκεται στις καταστάσεις που απορρίπτονται οι πελάτες της συγκεκριμένης κατηγορίας, επί το ρυθμό άφιξής τους λ_2 . Ο ρυθμός αυτός δίνεται από το τύπο:

$$LS_2 = \lambda_2 \left(\sum_{x_2=0}^{r+c_2} \sum_{x_1=-c_1}^{x_2-c_2} \sum_{i=0}^1 P(x_1, x_2, i) \right)$$

3.2 ΚΟΣΤΟΣ ΑΠΟΘΕΜΑΤΟΠΟΙΗΣΗΣ

Το κόστος αποθέματος είναι ανάλογο, του μοναδιαίου κόστους αποθεματοποίησης h αλλά και του μέσου αποθέματος H του συστήματος. Πιο συγκεκριμένα είναι το γινόμενο αυτών των δύο παραμέτρων hH . Το μέσο απόθεμα είναι το άθροισμα του γινομένου για κάθε τιμή του x_1 από 1 έως S επί την αντίστοιχη πιθανότητα και δίνεται από την παρακάτω εξίσωση:

$$H = \left(\sum_{x_1=1}^r x_1 \sum_{x_2=0}^{r+c_2} \sum_{i=0}^1 P(x_1, x_2, i) \right) + \left(\sum_{x_1=r+1}^S x_1 \sum_{i=0}^1 P(x_1, 0, i) \right)$$

3.3 ΜΕΣΟ ΠΛΗΘΟΣ ΕΚΚΡΕΜΩΝ ΠΑΡΑΓΓΕΛΙΩΝ

Το κόστος εκκρεμών παραγγελιών είναι μία γραμμική συνάρτηση του μέσου πλήθους εκκρεμών παραγγελιών για κάθε μία από τις δύο κατηγορίες πελατών.

Εκκρεμείς παραγγελίες για τους πελάτες τύπου 1 έχουμε όταν το $x_1 < 0$ και το μέσο πλήθος της συγκεκριμένης κατηγορίας δίνεται από την παρακάτω εξίσωση:

$$B_1 = \left(\sum_{x_1=-c_1}^{-1} (-x_1) \sum_{x_2=0}^{r+c_2} \sum_{i=0}^1 P(x_1, x_2, i) \right)$$

Αντίστοιχα το πλήθος εκκρεμών παραγγελιών της δεύτερης κατηγορίας πελατών που αντιπροσωπεύεται από τη μεταβλητή x_2 όταν $x_2 > 1$, δίνεται από τη σχέση:

$$B_2 = \left(\sum_{x_2=1}^{r+c_2} x_2 \sum_{x_1=-c_1}^r \sum_{i=0}^1 P(x_1, x_2, i) \right)$$

Έτσι λοιπόν, έχοντας εκτιμήσει τα διάφορα μέτρα απόδοσης του συστήματος που καθορίζουν τη τιμή του κόστους λειτουργίας, μπορούμε να την υπολογίσουμε κάνοντας χρήση του τύπου όπως αναφέρθηκε στην αρχή του κεφαλαίου. Το μέσο κόστος λειτουργίας είναι συνάρτηση των καταωφλίων ελέγχου της προτεινόμενης πολιτικής (BA-EΔBE), που καθορίζεται από τις τιμές των παραμέτρων ελέγχου S , r , c_1 , c_2 . Για να βρούμε τη βέλτιστη πολιτική, της μορφής αυτής, πρέπει να ελαχιστοποιήσουμε την συνάρτηση μέσου κόστους λειτουργίας ως προς τις παραμέτρους ελέγχου. Η βελτιστοποίηση γίνεται με εξαντλητική αναζήτηση στον χώρο των μεταβλητών ελέγχου ανάμεσα σε ένα εύρος τιμών που εμείς θέτουμε.

ΚΕΦΑΛΑΙΟ 4: ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Ακολουθώντας τη μεθοδολογία που παρουσιάστηκε στα προηγούμενα κεφάλαια, δημιουργήθηκε λογισμικό σε γλώσσα προγραμματισμού C, για την εύρεση των πιθανοτήτων μόνιμης κατάστασης καθώς και του μέσου συνολικού κόστους σε κάθε περίπτωση. Με τον υπολογισμό της τιμής του συνολικού μέσου κόστους έγινε σύγκριση μεταξύ των πολιτικών (BA-EΔBE - BAEΠ). Στο παρόν κεφάλαιο μελετάμε την επίδραση των διαφόρων παραμέτρων του συστήματος στο μέσο κόστος αλλά και στις βέλτιστες τιμές των παραμέτρων ελέγχου κάνοντας πολλά αριθμητικά πειράματα με διαφορετικές τιμές των παραμέτρων λ_1 , λ_2 , ε , β και μ .

Ακόμα εξετάζουμε αν η συνάρτηση μέσου κόστους είναι κυρτή συνάρτηση ή έχει κάποια μορφή κυρτότητας ως προς τις παραμέτρους ελέγχου. Η κυρτότητα είναι αυτή που μας εξασφαλίζει ότι ένα τοπικό βέλτιστο είναι και ολικό βέλτιστο.

Το λογισμικό που χρησιμοποιήθηκε σε γλώσσα προγραμματισμού C, παρουσιάζεται στο τέλος της διπλωματικής εργασίας.

4.1 ΜΕΛΕΤΗ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ ΓΙΑ ΜΕΤΑΒΛΗΤΕΣ ΤΙΜΕΣ ΤΩΝ ΠΑΡΑΜΕΤΡΩΝ

Σε αυτό το σημείο της διπλωματικής εξετάζουμε την μεταβολή του βέλτιστου συνολικού κόστους καθώς μεταβάλλουμε τις τιμές των παραμέτρων. Για κάθε συνδυασμό των διαφόρων τιμών των παραμέτρων εκτιμούμε τις βέλτιστες τιμές των κατωφλίων ελέγχου της πολιτικής μας που ελαχιστοποιούν το μέσο συνολικό κόστος.

Η βέλτιστη τιμή του μέσου συνολικού κόστους και τα βέλτιστα κατώφλια που αντιστοιχούν στο συγκεκριμένο συνολικό κόστος, παρουσιάζονται παρακάτω σε πίνακες ανά πολιτική. Οι εξεταζόμενες πολιτικές είναι δύο. Η πολιτική Βασικού Αποθέματος – Εξαρτημένου διπλού Βασικού Ελλείμματος (BA-EΔBE) που είναι και η βασική μας πολιτική, η οποία χρησιμοποιεί τέσσερα κατώφλια ελέγχου S , r , c_1 , c_2 για τις εκκρεμείς παραγγελίες αλλά και για τον έλεγχο του αποθέματος. Η δεύτερη πολιτική είναι η πολιτική Βασικού Αποθέματος με Εξαρτημένη Απόδοση Προτεραιότητας (BAEΠ), η οποία απαρτίζεται από ένα κατώφλι βασικού αποθέματος S και δύο κατώφλια απόδοσης προτεραιότητας r_0 και r_1 , ενώ δεν επιτρέπονται εκκρεμείς παραγγελίες.

Οι τιμές των που χρησιμοποιήθηκαν για το βασικό σενάριο διεξαγωγής των αριθμητικών πειραμάτων είναι οι εξής: $\lambda_1=1$, $\lambda_2=1$, $\beta=1$, $\varepsilon=10$, $\mu=6$, $h=1$, $P_1=35$, $P_2=6$, $b_1=4$, $b_2=2$.

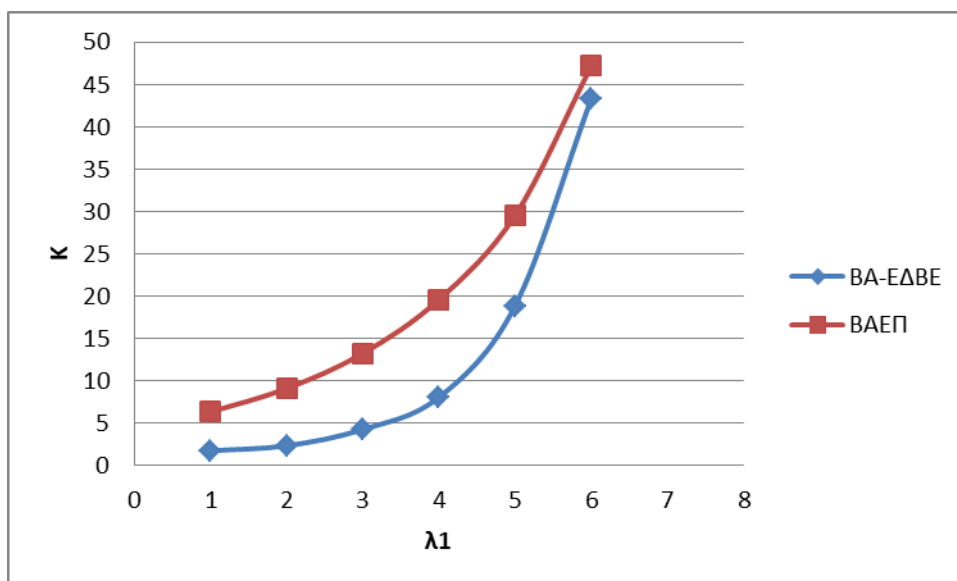
4.1.1 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ λ_1

Αρχικά εξετάζουμε τη περίπτωση μεταβολής του ρυθμού άφιξης πελατών τύπου 1, λ_1 . Από τα αποτελέσματα, που παρουσιάζονται στον Πίνακα 1, παρατηρούμε ότι όσο αυξάνεται ο ρυθμός έλευσης πελατών αυξάνεται και το βασικό απόθεμα S , πράγμα λογικό, για να υπάρχει περισσότερο διαθέσιμο απόθεμα που χρειαζόμαστε για να μην απορρίπτεται μεγάλος αριθμός πελατών. Πρακτικά, με την αύξηση του βασικού αποθέματος οδηγούμαστε σε μεγαλύτερους ρυθμούς παραγωγής και σε λιγότερες ανικανοποίητες παραγγελίες. Ακόμα παρατηρούμε ότι όσο μικρότερος ο ρυθμός άφιξης πελατών τόσο μικρότερο είναι και το μέσο συνολικό κόστος, και αυτό συμβαίνει γιατί το σύστημα προλαβαίνει να τους εξυπηρετήσει με αποτέλεσμα να μειώνεται το κόστος απόρριψης. Τέλος, όπως είναι φανερό, τα κατώφλια εκκρεμών παραγγελιών μειώνονται καθώς αυξάνεται ο ρυθμός λ_1 . Γνωρίζοντας ότι ο ονομαστικός ρυθμός παραγωγής παραμένει σταθερός, το σύστημα θα πρέπει να δέχεται λιγότερες παραγγελίες έτσι ώστε να μπορεί να τις εξυπηρετήσει και να μην δημιουργούνται ουρές εκκρεμών παραγγελιών αυξάνοντας κατ'επέκταση και το συνολικό κόστος.

	Πολιτική (BA-EΔBE)					Πολιτική (BAEP)			
λ_1	K	S	r	c_1	c_2	K	S	σ_1	σ_0
1	1,7261	2	0	-10	-7	6,3101	6	1	2
2	2,3236	2	0	-10	-6	9,0852	9	2	33
3	4,2548	4	0	-10	-3	13,2214	13	3	4
4	8,0504	8	2	-10	0	19,5832	19	6	7
5	18,8455	15	2	-6	0	29,5231	29	11	12
6	43,3212	15	3	-4	0	47,1558	46	23	24

Πίνακας 1: Επίδραση του ρυθμού λ_1 στην απόδοση του συστήματος.

Κάνοντας χρήση του Γραφήματος 1, μπορούμε να κάνουμε σύγκριση ανάμεσα στις δύο πολιτικές. Παρατηρούμε ότι εξ'αρχής υπάρχει μια φανερή διαφορά ανάμεσα στα κόστη των πολιτικών όπως φαίνεται και από το γράφημα. Για μικρές τιμές του λ_1 σε σχέση με το ρυθμό παραγωγής μ το σύστημα μπορεί να ικανοποιήσει το σύνολο των παραγγελιών, οπότε δεν υπάρχει λόγος απόρριψης από την πλευρά της BAEP και δημιουργίας εκκρεμών παραγγελιών από τη πλευρά της BA-EΔBE. Όσο όμως αυξάνεται ο ρυθμός άφιξης τόσο δυσκολότερη γίνεται η ικανοποίηση του συνόλου των παραγγελιών, και είναι απαραίτητο να δίδεται αυξημένη προτεραιότητα στους πελάτες τύπου 1 για την πολιτική BAEP, και δημιουργία εκκρεμών παραγγελιών για την πολιτική BA-EΔBE αυξάνοντας και στις δύο περιπτώσεις το συνολικό μέσο κόστος.



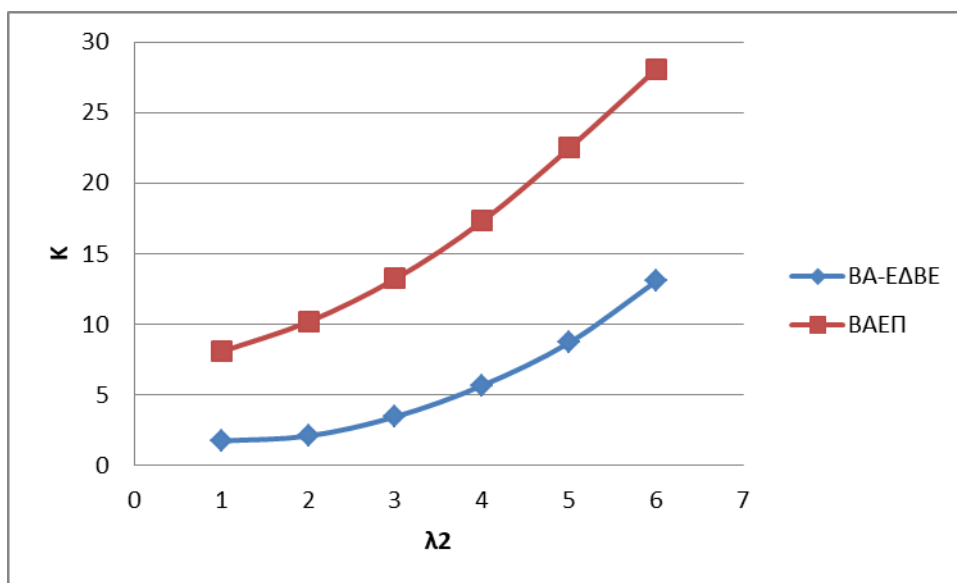
Γράφημα 1: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού λ_1 .

4.1.2 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ λ_2

Για την μεταβολή του ρυθμού άφιξης πελατών τύπου 2 ισχύει το ίδιο με την περίπτωση μεταβολής του ρυθμού λ_1 . Τα αποτελέσματα από τη μεταβολή του λ_2 εμφανίζονται στον Πίνακα 2. Παρατηρούμε και εδώ, ότι το βέλτιστο βασικό απόθεμα S γίνεται μεγαλύτερο όσο αυξάνεται το λ_2 . Ακόμα όσον αφορά τα κατώφλια εκκρεμών παραγγελιών, παρατηρούμε ότι μειώνονται με την αύξηση του ρυθμού έλευσης πελατών τύπου 2, όπως ακριβώς συνέβη και με το λ_1 .

λ_2	Πολιτική (BA-EΔBE)					Πολιτική (BAΕΠ)			
	K	S	r	c_1	c_2	K	S	σ_1	σ_0
1	1,7261	2	0	-10	-7	8,058	8	2	3
2	2,0851	2	0	-10	-7	10,1774	10	3	4
3	3,4505	3	0	-9	-5	13,2214	13	3	4
4	5,6624	5	0	-7	-3	17,3093	17	4	5
5	8,713	8	0	-10	-2	22,4938	22	4	5
6	13,0762	13	0	-9	-2	28,092	28	5	6

Πίνακας 2: Επίδραση του ρυθμού λ_2 στην απόδοση του συστήματος.



Γράφημα 2: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού λ_2 .

4.1.3 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ μ

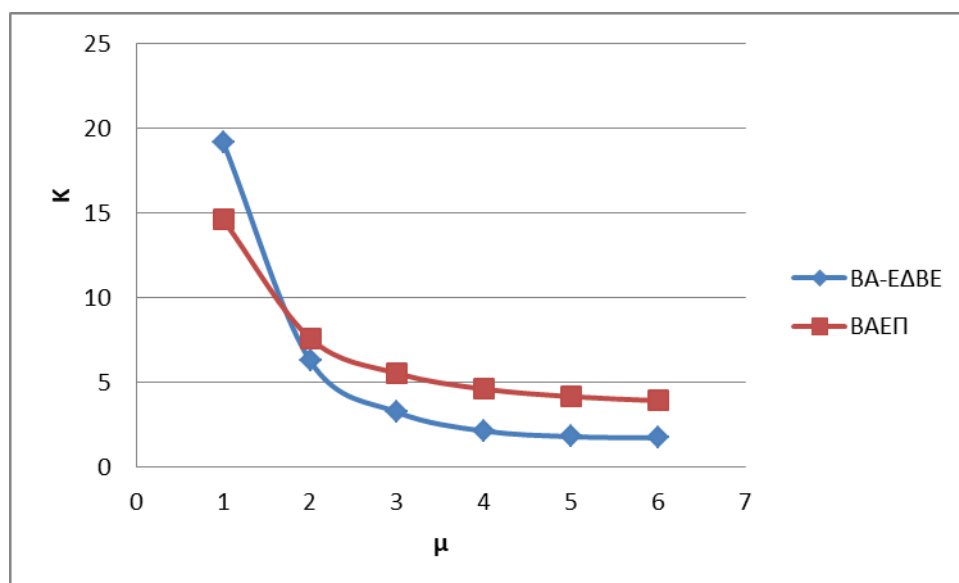
Για να είναι μικρότερο το συνολικό μέσο κόστος, θα πρέπει ο ρυθμός παραγωγής μ να είναι όσο το δυνατόν μεγαλύτερος. Από τη μελέτη των αριθμητικών αποτελεσμάτων που παρουσιάζονται στον Πίνακα 3, καθώς αυξάνεται το μ το βέλτιστο βασικό απόθεμα μειώνεται καθώς παράγονται προϊόντα με μεγαλύτερο ρυθμό, οπότε απαιτείται μικρότερο απόθεμα. Επίσης για μεγάλες τιμές του ρυθμού παραγωγής αυξάνονται τα κατώφλια εκκρεμών παραγγελιών με αποτέλεσμα τη μείωση του κόστους εκκρεμών παραγγελιών και κατ'επέκταση του συνολικού μέσου κόστους. Επί της ουσίας η αύξηση του μ έχει παρόμοια επίπτωση στο σύστημα με την μείωση των λ_1 και λ_2 .

μ	Πολιτική (BA-ΕΔΒΕ)					Πολιτική (ΒΑΕΠ)			
	K	S	r	c_1	c_2	K	S	σ_1	σ_0
1	19,1348	9	2	-2	0	14,578	14	7	8
2	6,2549	6	1	-7	-1	7,5821	7	2	3
3	3,2429	3	0	-10	-2	5,5243	5	1	2
4	2,1282	2	0	-10	-6	4,6033	4	1	2
5	1,8009	2	0	-10	-7	4,1605	4	1	2
6	1,7261	2	0	-10	-7	3,9149	3	1	2

Πίνακας 3: Επίδραση του ρυθμού μ στην απόδοση του συστήματος.

Για μικρές τιμές του μ η πολιτική ΒΑΕΠ δείχνει να είναι καλύτερη από την προτεινόμενη πολιτική BA-ΕΔΒΕ. Και στις δύο περιπτώσεις παρατηρούμε ότι με την με την αύξηση του ρυθμού μ το βέλτιστο συνολικό μέσο κόστος μειώνεται όπως φαίνεται και στο Γράφημα 3 και έτσι αφού η πλειοψηφία των παραγγελιών ικανοποιείται όπως συμβαίνει σε υψηλούς ρυθμούς παραγωγής δεν υπάρχει λόγος

απόρριψης παραγγελιών για την πολιτική ΒΑΕΠ αλλά και δημιουργία εκκρεμών παραγγελιών για την πολιτική ΒΑ-ΕΔΒΕ.



Γράφημα 3: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού μ .

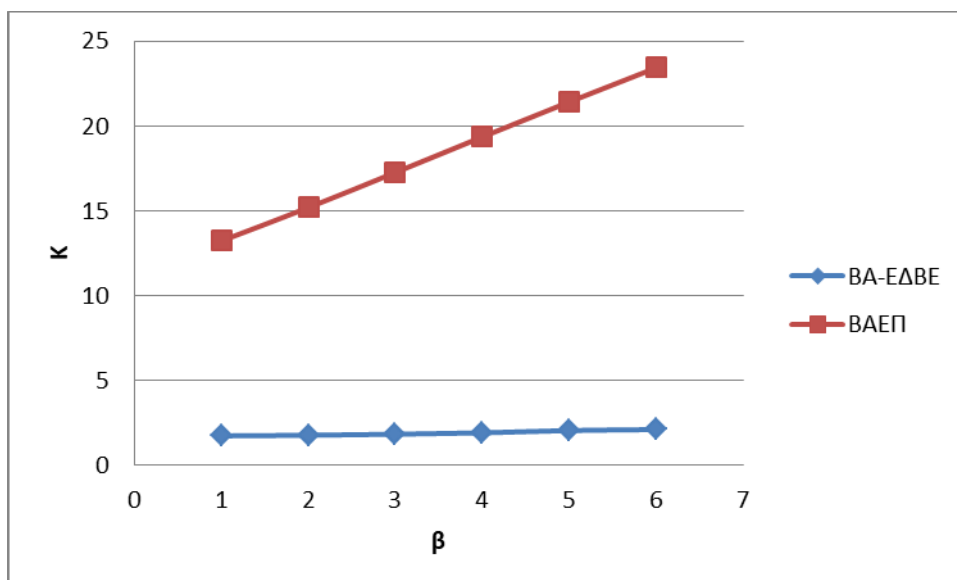
4.1.4 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ β

Όσον αφορά το ρυθμό που το σύστημα παθαίνει βλάβη, θέλουμε ο ρυθμός αυτός να είναι όσο το δυνατόν μικρότερος καθώς έτσι επιτυγχάνεται σημαντική μείωση στο μέσο συνολικό κόστος. Με αυτό τον τρόπο το σύστημα παραγωγής βρίσκεται λιγότερο χρόνο σε μη λειτουργικές καταστάσεις και έτσι προλαβαίνει να παράγει περισσότερο με αποτέλεσμα να μειώνεται με αυτό τον τρόπο το κόστος εκκρεμών παραγγελιών και εν συνεχεία το μέσο συνολικό κόστος λειτουργίας. Αυτό επιβεβαιώνεται από τα αριθμητικά αποτελέσματα που παρουσιάζονται στον Πίνακα 4. Επίσης παρατηρούμε ότι όσο αυξάνεται ο ρυθμός βλάβης το βέλτιστο απόθεμα S παραμένει ίδιο, γεγονός που δηλώνει ότι η γραμμή παραγωγής είναι σε θέση να εξυπηρετήσει την πλειοψηφία των παραγγελιών μειώνοντας το κόστος των εκκρεμών παραγγελιών. Η μείωση αυτή επιτυγχάνεται με την μείωση των τιμών των κατωφλίων αποδοχής εκκρεμών παραγγελιών c_1, c_2 .

β	Πολιτική (BA-ΕΔΒΕ)					Πολιτική (ΒΑΕΠ)			
	K	S	r	c_1	c_2	K	S	σ_1	σ_0
1	1,7261	2	0	-10	-7	13,2214	13	3	4
2	1,7646	2	0	-8	-6	15,187	15	4	5
3	1,8261	2	0	-8	-6	17,2583	17	5	6
4	1,9102	2	0	-10	-4	19,3595	19	6	7
5	2,0452	2	0	-7	-5	21,4215	21	6	7
6	2,1016	2	0	-8	-4	23,4373	23	7	8

Πίνακας 4: Επίδραση του ρυθμού β στη απόδοση του συστήματος.

Με βάση το Γράφημα 4, είμαστε σε θέση να συγκρίνουμε τις δύο παραπάνω πολιτικές. Βλέπουμε ότι καθώς αυξάνεται το β η τιμή του μέσου συνολικού κόστους δεν αλλάζει πολύ για την προτεινόμενη πολιτική BA-EΔBE σε αντίθεση με την συγκρινόμενη πολιτική BAEP όπου η αύξηση του κόστους είναι αρκετά αισθητή, δίνοντας έτσι ένα σημαντικό προβάδισμα στην προτεινόμενη πολιτική.



Γράφημα 4: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού β .

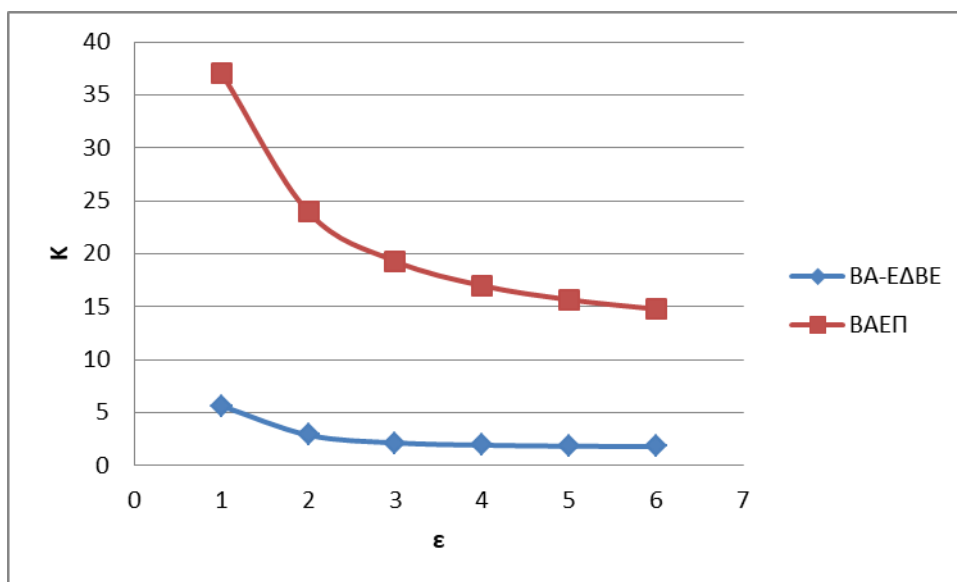
4.1.5 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ ε

Σε αυτή τη περίπτωση και όπως φαίνεται και από το Πίνακα 5, όσο πιο γρήγορα μεγαλώνει ο ρυθμός επισκευής ε , δηλαδή όσο πιο γρήγορα επισκευάζεται το σύστημα παραγωγής τόσο μειώνεται ο χρόνος που το σύστημα βρίσκεται σε αδράνεια με αποτέλεσμα να έχουμε αύξηση της παραγωγικότητας του συστήματος και συνεπώς μικρότερο κόστος εκκρεμών παραγγελιών και συνεπώς μικρότερο βέλτιστο συνολικό μέσο κόστος. Το μέγιστο απόθεμα S αυξάνεται σε χαμηλά επίπεδα του ρυθμού ε καθώς όταν αργεί να επισκευαστεί το σύστημα απαιτούνται μεγαλύτερα αποθέματα για να μην αυξηθεί το κόστος εκκρεμών παραγγελιών και κατ'επέκταση το μέσο συνολικό κόστος.

ε	Πολιτική (BA-EΔBE)					Πολιτική (BAEP)			
	K	S	r	c_1	c_2	K	S	σ_1	σ_0
1	5,6317	5	1	-10	-1	37,1018	35	15	18
2	2,8654	2	0	-9	-6	23,9344	23	8	9
3	2,1325	2	0	-9	-7	19,2875	18	6	7
4	1,9076	2	0	-10	-7	16,9769	16	5	6
5	1,8175	2	0	-10	-7	15,6453	15	4	5
6	1,776	2	0	-10	-6	14,7681	14	4	5

Πίνακας 5: Επίδραση του ρυθμού ε στη απόδοση του συστήματος.

Στο Γράφημα 5, βλέπουμε ότι από χαμηλές έως υψηλές τιμές του ε , η προτεινόμενη πολιτική έχει χαμηλό κόστος, σε αντίθετη περίπτωση με την συγκρινόμενη πολιτική η οποία σε χαμηλές τιμές του ρυθμού ε έχει αρκετά υψηλό κόστος και σε υψηλές τιμές μειώνεται σημαντικά.



Γράφημα 5: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού ε .

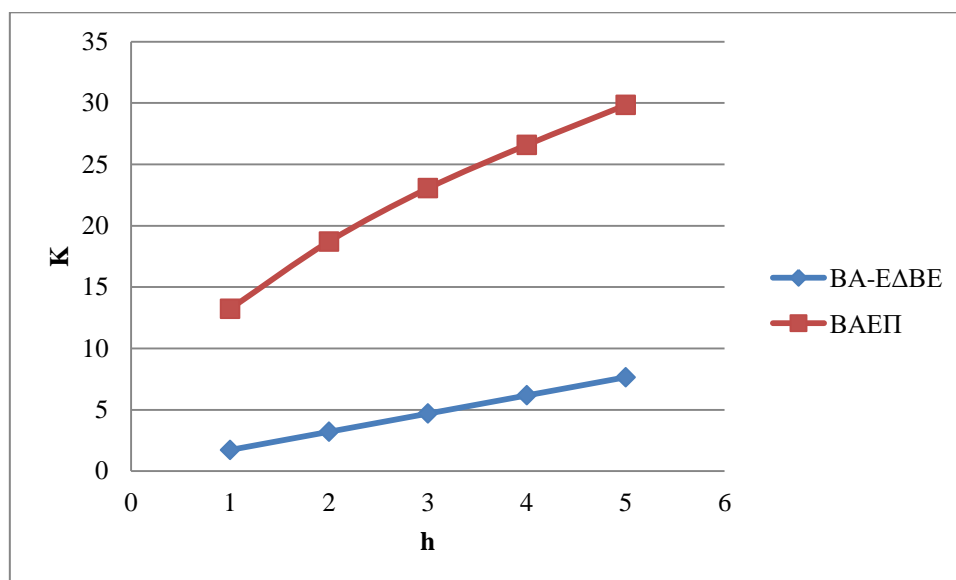
4.1.6 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ h

Αναφορικά με την αύξηση του μοναδιαίου κόστους αποθέματος ανά μονάδα προϊόντος και ανά μονάδα χρόνου h , παρατηρούμε βάση του Πίνακα 6, ότι το μέσο συνολικό κόστος αυξάνεται σε αντίθετη περίπτωση με το βασικό απόθεμα S το οποίο παραμένει σταθερό σε όλες τις τιμές του h από τις χαμηλότερες έως τις υψηλότερες. Το ίδιο ακριβώς συμβαίνει και στις περιπτώσεις των κατωφλίων εκκρεμών παραγγελιών αλλά και στο κατώφλι προτεραιότητας r .

h	Πολιτική (BA-ΕΔΒΕ)					Πολιτική (ΒΑΕΠ)			
	K	S	r	c ₁	c ₂	K	S	σ ₁	σ ₀
1	1,7261	2	0	-10	-7	13,2214	13	3	4
2	3,2068	2	0	-10	-7	18,7082	9	3	4
3	4,6872	2	0	-10	-7	23,0444	7	2	3
4	6,1683	2	0	-10	-7	26,5797	6	2	3
5	7,649	2	0	-10	-7	29,828	5	2	3

Πίνακας 6: Επίδραση του ρυθμού h στη απόδοση του συστήματος.

Η προτεινόμενη πολιτική και στην περίπτωση μεταβολής του μοναδιαίου κόστους αποθέματος δείχνει να έχει καλύτερα αποτελέσματα συγκριτικά με την πολιτική ΒΑΕΠ όπως φαίνεται και στο Γράφημα 6.



Γράφημα 6: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού h .

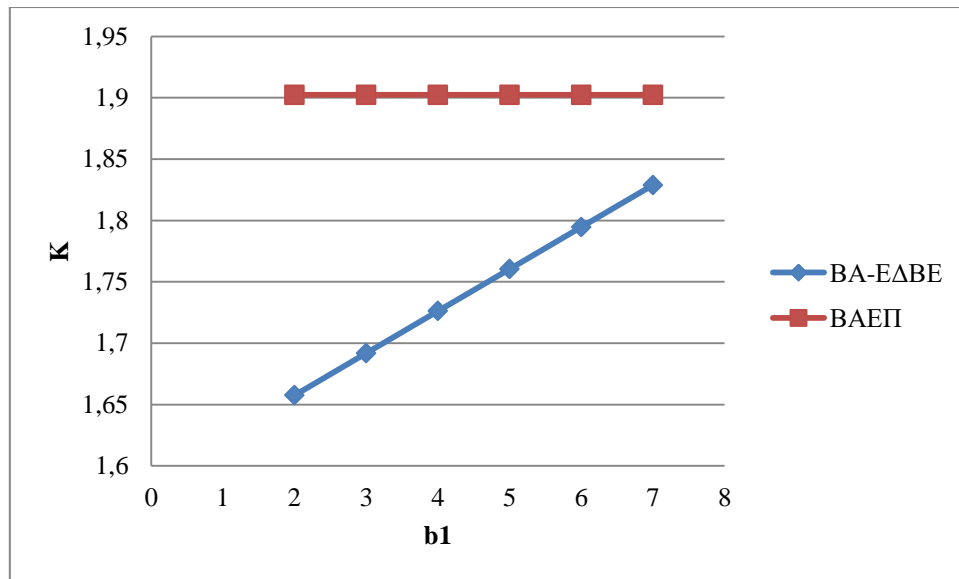
4.1.7 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ b_1

Από τον Πίνακα 7, βλέπουμε ότι το μοναδιαίο κόστος εκκρεμών παραγγελιών προϊόντος τύπου 1, επηρεάζει το μέσο συνολικό κόστος αυξάνοντας τις τιμές του σταδιακά. Παρατηρούμε επίσης, πως τα κατώφλια ελέγχου του συστήματος παραμένουν αμετάβλητα στη μεταβολή του μοναδιαίου κόστους b_1 .

b_1	Πολιτική (BA-ΕΔΒΕ)					Πολιτική (ΒΑΕΠ)
	K	S	r	c_1	c_2	K
2	1,6575	2	0	-10	-7	1,9024
3	1,6918	2	0	-10	-7	1,9024
4	1,7261	2	0	-10	-7	1,9024
5	1,7603	2	0	-10	-7	1,9024
6	1,7946	2	0	-10	-7	1,9024
7	1,8288	2	0	-10	-7	1,9024

Πίνακας 7: Επίδραση του ρυθμού b_1 στη απόδοση του συστήματος.

Με τη βοήθεια του Γραφήματος 7, παρατηρούμε ότι η συγκρινόμενη πολιτική ΒΑΕΠ παραμένει ανεπηρέαστη για όλες τις τιμές του b_1 . Πρακτικά το μέσο συνολικό κόστος παραμένει σταθερό, για τον λόγο ότι η πολιτική αυτή δεν δέχεται εκκρεμείς παραγγελίες.



Γράφημα 7: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού b_1 .

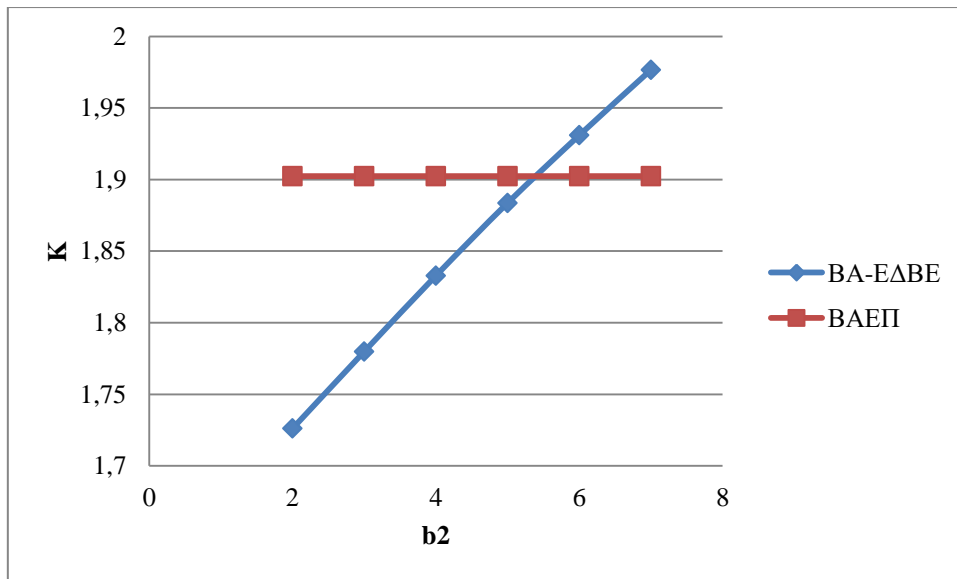
4.1.8 ΕΠΙΔΡΑΣΗ ΣΤΗ ΣΥΝΑΡΤΗΣΗ ΚΟΣΤΟΥΣ ΛΟΓΩ ΜΕΤΑΒΟΛΗΣ ΤΟΥ ΡΥΘΜΟΥ b_2

Όπως και με το b_1 , η αύξηση του μοναδιαίου κόστους εκκρεμών παραγγελιών προϊόντος τύπου 2 b_2 , οδηγεί σε αύξηση του μέσου συνολικού κόστους. Για την περίπτωση αυτή βλέπουμε, από τον Πίνακα 8, πως το κατώφλι βασικού αποθέματος και το κατώφλι προτεραιότητας παραμένουν ανεπηρέαστα στη μεταβολή b_2 , ενώ ταυτόχρονα έχουμε μείωση στα κατώφλια εκκρεμών παραγγελιών c_1 και c_2 , με μικρότερη αυτή του c_1 , έτσι ώστε να μην έχουμε πολλές εκκρεμείς παραγγελίες με αποτέλεσμα την αύξηση του μέσου συνολικού κόστους.

b_2	Πολιτική (BA-ΕΔΒΕ)					Πολιτική (ΒΑΕΠ)
	K	S	r	c_1	c_2	K
2	1,7261	2	0	-10	-7	1,9024
3	1,7798	2	0	-10	-7	1,9024
4	1,8328	2	0	-10	-5	1,9024
5	1,8836	2	0	-7	-4	1,9024
6	1,9309	2	0	-7	-3	1,9024
7	1,9766	2	0	-7	-2	1,9024

Πίνακας 8: Επίδραση του ρυθμού b_1 στη απόδοση του συστήματος.

Όπως συμβαίνει και με το κόστος εκκρεμών παραγγελιών τύπου 1 (b_1), η πολιτική ΒΑΕΠ παραμένει σταθερή αφού δεν επηρεάζεται από τη μεταβολή του b_2 .



Γράφημα 8: Απόδοση των δύο πολιτικών για διάφορες τιμές του ρυθμού b_2 .

4.2 ΜΕΛΕΤΗ ΤΗΣ ΚΥΡΤΟΤΗΤΑΣ ΤΗΣ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ ΩΣ ΠΡΟΣ ΤΙΣ ΜΕΤΑΒΛΗΤΕΣ ΕΛΕΓΧΟΥ

Λόγω της πολυπλοκότητας της συνάρτησης κόστους η οποία έχει τέσσερις μεταβλητές ελέγχου θα προχωρήσουμε στο ακόλουθο πείραμα για να εξετάσουμε αν η συνάρτηση είναι κυρτή ή μονοκόρυφη ως προς κάθε μία από τις μεταβλητές. Ξεκινώντας κρατάμε σταθερές τις τιμές όλων των μεταβλητών εκτός από μία, που μεταβάλλουμε και υπολογίζουμε το μέσο κόστος για κάθε τιμή της κυμαινόμενης μεταβλητής. Στη συνέχεια αυξάνουμε την τιμή της κυμαινόμενης μεταβλητής μέχρι εκεί που μας επιτρέπεται και επαναυπολογίζουμε το μέσο κόστος. Η διαδικασία αυτή συνεχίζεται για όλες τις μεταβλητές, μέχρι να καλύψουμε όλο το εύρος τιμών της κάθε μεταβλητής που εξετάζουμε.

Για το παράδειγμα που εξετάζουμε, υπολογίζουμε το μέσο κόστος ενώ μεταβάλλονται οι τιμές των κατωφλίων ελέγχου S , r , c_1 , c_2 . Το σενάριο που χρησιμοποιήθηκε για τον υπολογισμό απαρτίζεται από τις παρακάτω τιμές των παραμέτρων και μεταβλητών ελέγχου: $\lambda_1 = 7$, $\lambda_2 = 2$, $\mu = 10$, $\beta = 1$, $\varepsilon = 8$, $P_1 = 12$, $P_2 = 10$, $h = 2$, $b_1 = 5$, $b_2 = 3$, $S = 15$, $r = 1$, $c_1 = -10$, $c_2 = -1$.

Τα αποτελέσματα του πειράματος παρουσιάζονται στους παρακάτω πίνακες ξεχωριστά για κάθε κατώφλι ελέγχου:

Πολιτική (BA-EΛBE)				
Μεταβλητό S	K	r	c ₁	c ₂
2	26,357	1	-10	-1
3	24,1393	1	-10	-1
4	22,5789	1	-10	-1
5	21,4877	1	-10	-1
6	20,7437	1	-10	-1
7	20,264	1	-10	-1
8	19,9898	1	-10	-1
9	19,8785	1	-10	-1
10	19,8983	1	-10	-1
11	20,0251	1	-10	-1
12	20,2401	1	-10	-1
13	20,5287	1	-10	-1
14	20,8791	1	-10	-1
15	21,2818	1	-10	-1

Πίνακας 9: Μέσο κόστος λειτουργίας συναρτήσει βασικού αποθέματος S.

Πολιτική (BA-EΛBE)				
Μεταβλητό r	K	S	c ₁	c ₂
1	21,2818	15	-10	-1
2	21,4616	15	-10	-1
3	21,8645	15	-10	-1

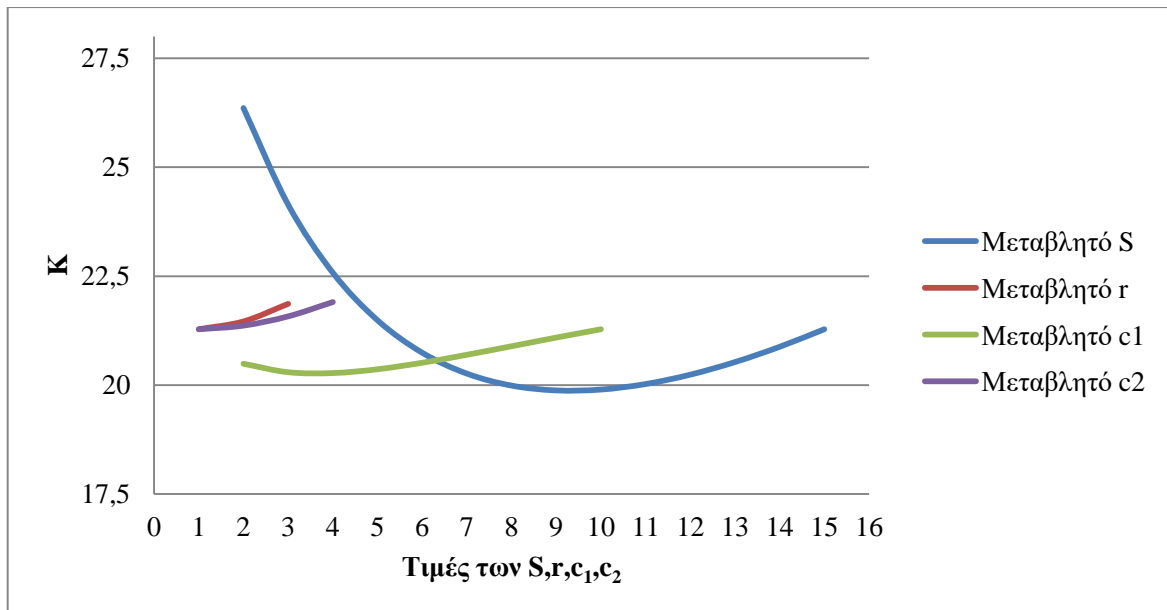
Πίνακας 10: Μέσο κόστος λειτουργίας συναρτήσει κατωφλίου προτεραιότητας r.

Πολιτική (BA-EΛBE)				
Μεταβλητό -c ₁	K	S	r	c ₂
2	20,4909	15	1	-1
3	20,2962	15	1	-1
4	20,277	15	1	-1
5	20,3643	15	1	-1
6	20,5136	15	1	-1
7	20,6961	15	1	-1
8	20,8924	15	1	-1
9	21,0903	15	1	-1
10	21,2818	15	1	-1

Πίνακας 11: Μέσο κόστος λειτουργίας συναρτήσει κατωφλίου εκκρεμών παραγγελιών c₁.

Πολιτική (BA-EΛBE)				
Μεταβλητό -c ₂	K	S	r	c ₁
1	21,2818	15	1	-10
2	21,3653	15	1	-10
3	21,5776	15	1	-10
4	21,9059	15	1	-10

Πίνακας 12: Μέσο κόστος λειτουργίας συναρτήσει κατωφλίου εκκρεμών παραγγελιών c₂.



Γράφημα 9: Αποτελέσματα κυρτότητας αν μεταβλητή ελέγχου.

Παρατηρώντας τα αποτελέσματα βλέπουμε, ότι η συνάρτηση είναι κυρτή ή μονοκόρυφη ως προς κάθε μία από τις παραμέτρους S , r , c_1 , c_2 . Εκτός των αποτελεσμάτων που παρουσιάζουμε και μία άλλη σειρά από άλλα πειράματα που εκτελέσαμε, συμφωνούν με αυτό το συμπέρασμα.

4.3 ΜΕΛΕΤΗ ΚΥΡΤΟΤΗΤΑΣ ΣΥΝΑΡΤΗΣΗΣ ΚΟΣΤΟΥΣ

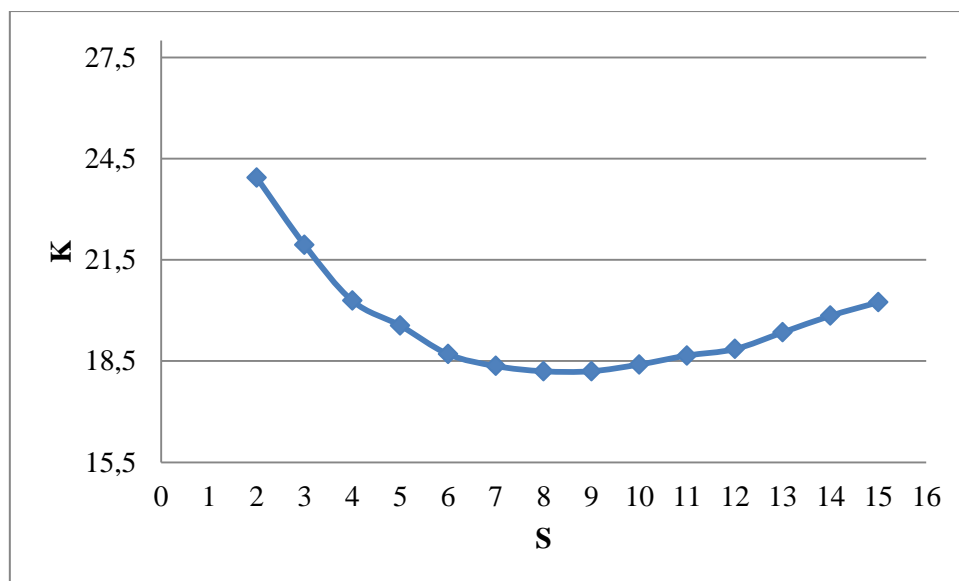
Για την μελέτη συνολικής της κυρτότητας της συνάρτησης κόστους στο σύστημα παραγωγής που εξετάζουμε, ακολουθήσαμε την παρακάτω διαδικασία. Κάνοντας χρήση του αλγορίθμου που πραγματοποιήθηκε σε γλώσσα προγραμματισμού C, για κάθε πιθανή τιμή του κατωφλίου αποθέματος S , από τη μικρότερη έως τη μεγαλύτερη, βρήκαμε τα βέλτιστα κατώφλια ελέγχου, χάρη των οποίων ελαχιστοποιείτο το μέσο συνολικό κόστος.

Η διαδικασία αυτή πραγματοποιήθηκε για τις εξής τιμές των παραμέτρων: $\lambda_1 = 7$, $\lambda_2 = 2$, $\mu = 10$, $\beta = 1$, $\varepsilon = 8$, $P_1 = 12$, $P_2 = 10$, $h = 2$, $b_1 = 5$, $b_2 = 3$. Τα αριθμητικά αποτελέσματα παρουσιάζονται στο παρακάτω πίνακα.

Πολιτική (BA-EΔBE)				
S	r	c ₁	c ₂	K
2	0	-5	-2	23,93535
3	1	-5	-3	21,94265
4	0	-4	-2	20,28877
5	1	-3	-1	19,55302
6	0	-4	-2	18,7073
7	0	-4	-2	18,35312
8	0	-4	-2	18,19857
9	0	-4	-2	18,19979
10	1	-3	-1	18,39845
11	1	-4	-1	18,66403
12	0	-4	-2	18,85547
13	1	-3	-1	19,3492
14	1	-5	-2	19,84263
15	0	-5	-2	20,24266

Πίνακας 13: Τιμές κόστους και βέλτιστων κατωφλίων της πολιτικής BA-EΔBE.

Η γραφική παράσταση που προκύπτει από τις παραπάνω τιμές φαίνεται στο Γράφημα 10. Παρατηρούμε ότι υπάρχει ένα μοναδικό ελάχιστο, είναι δηλαδή μονοκόρυφη. Ακόμα, οι διάφορες τιμές του K μας δείχνουν πως η συνάρτηση κόστους είναι κυρτή ως προς το S. Γενικά από τα πειράματα που παρουσιάζονται αλλά και από μια σειρά πειραμάτων που πραγματοποιήσαμε, αλλά δεν παρουσιάζουμε για λόγους οικονομίας χώρου, φαίνεται ότι η συνάρτηση κόστους είναι μονοκόρυφη. Αυτό το αποτέλεσμα θα έχει σημαντικές επιπτώσεις στην διαδικασία εκτίμησης των βέλτιστων τιμών των παραμέτρων ελέγχου. Αν ισχύει ότι η συνάρτηση κόστους είναι μονοκόρυφη μπορούμε να κατασκευάσουμε πολύ αποτελεσματικούς υπολογιστικά αλγορίθμους για την εύρεση της βέλτιστης πολιτικής, χωρίς να χρειάζεται να εξετάζουμε εξαντλητικά όλο τον χώρο των παραμέτρων ελέγχου.



Γράφημα 10: Κυρτότητα πολιτικής ΒΑ-ΕΔΒΕ.

ΚΕΦΑΛΑΙΟ 5: ΣΥΜΠΕΡΑΣΜΑΤΑ

Στη παρούσα διπλωματική εργασία εξετάσαμε ένα πρόβλημα ελέγχου σε ένα σύστημα παραγωγής που παράγει ένα προϊόν για δύο κατηγορίες πελατών και επιτρέπονται εκκρεμείς παραγγελίες. Η προτεινόμενη πολιτική ήταν η πολιτική βασικού αποθέματος – εξαρτημένου διπλού βασικού ελλείμματος (BA-EΔBE) την οποία τη συγκρίναμε με την πολιτική Βασικού Αποθέματος με Εξαρτημένη Απόδοση Προτεραιότητας (BAΕΠ).

Το σύστημα μοντελοποιήθηκε ως μία αλυσίδα Markov συνεχούς χρόνου. Αναπτύχθηκε λογισμικό σε γλώσσα προγραμματισμού C, για να εκτιμηθούν οι πιθανότητες μόνιμης κατάστασης αλλά και τα μέτρα απόδοσης που καθορίζουν τη συνάρτηση κόστους. Τέλος έγινε βελτιστοποίηση για την εύρεση των βέλτιστων τιμών των μεταβλητών ελέγχου που ελαχιστοποιούν το μέσο συνολικό κόστος λειτουργίας του συστήματος.

Έπειτα από μια σειρά αριθμητικών πειραμάτων, η προτεινόμενη πολιτική έδειξε να είναι καλύτερη σε σχέση με τη συγκρινόμενη πολιτική απόδοσης προτεραιότητας, καθώς επιτυγχάνεται σημαντική μείωση του κόστους λειτουργίας του συστήματος παραγωγής σε όλες τις περιπτώσεις που εξετάστηκαν. Στο σημείο αυτό θα πρέπει να αναφερθούμε και στην ευστάθεια του προτεινόμενου αλγορίθμου ανάλυσης της BA-EΔBE, που δείχνει να αντιμετωπίζει πρόβλημα στη σωστή εκτίμηση της συνάρτησης κόστους όταν οι τιμές των κατωφλίων c_1 και c_2 είναι σχετικά μεγάλες.

Κλείνοντας, τα αριθμητικά αποτελέσματα που λάβαμε από τη διερεύνηση της κυρτότητας της συνάρτησης, φανερώνουν ότι η συνάρτηση κόστους μπορεί να είναι κυρτή ή μονοκόρυφη άρα και πιθανόν να υπάρχει μοναδικό τοπικό και ολικό βέλτιστο, κάτι που μας κάνει να ελπίζουμε ότι η εύρεση της βέλτιστης λύσης είναι εφικτή με μειωμένο υπολογιστικό φόρτο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- Adan I. and van der Wal. J., Difference and differential equations in Stochastic Operations Research, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1998.
- Ha A. Y., “Inventory rationing in a make-to-stock production system with several demand classes and lost sales”, Management Science, vol. 51, pp. 1093-1103, 1997.
- Ioannidis S., “An inventory and order admission control policy for production systems with two customer classes”, International Journal of Production Economics, vol. 131, pp. 663-673, 2011.
- Σαράντης Ι., “Συνδυασμένος έλεγχος αποθεμάτων και αποδοχής παραγγελιών σε συστήματα παραγωγής που εξυπηρετούν δύο κατηγορίες πελατών και απαιτείται προετοιμασία για την έναρξη λειτουργίας της μονάδας παραγωγής”, Τμήμα Μ.Π.Δ, Πολυτεχνείο Κρήτης, Διπλωματική εργασία, 2013.
- Ασκούνης Δ., “Διοίκηση Παραγωγής και Συστημάτων Υπηρεσιών”, Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Η/Υ, Εθνικό Μετσόβιο Πολυτεχνείο, διδακτικές σημειώσεις, 2008.
- Τσαουσίδης Σ. “Έλεγχος αποθεμάτων σε συστήματα παραγωγής, που εξυπηρετούν δύο κατηγορίες πελατών, δεν επιτρέπονται εκκρεμείς παραγγελίες και μπορούν να υποστούν βλάβες”, Τμήμα Μ.Π.Δ, Πολυτεχνείο Κρήτης, Διπλωματική εργασία, 2014.
- Φίλης Ι. Α. Αναμονητικά Συστήματα-Γραμμές Παραγωγής-FMS, Τμήμα Μ.Π.Δ, Πολυτεχνείο Κρήτης, διδακτικές σημειώσεις, 2003.

ΠΑΡΑΡΤΗΜΑ - ΧΡΗΣΗ ΛΟΓΙΣΜΙΚΟΥ ΣΕ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C

```
//Βιβλιοθήκες που χρησιμοποιήθηκαν

#include <stdio.h>

#include <math.h>

#include <stdlib.h>


//Σταθεροί παράμετροι

#define l1 7.0d

#define l2 2.0d

#define m 10.0d

#define b 1.0d

#define e 8.0d

#define P1 12.0

#define P2 10.0

#define h 2.0

#define b1 5.0

#define b2 3.0


//Δηλώσεις συναρτήσεων που χρησιμοποιήθηκαν

double * create1dArray(int size);

double ** create2dArray(int x, int y);

double *** create3dArray(int x, int y, int z);

double **** create4dArray(int x, int y, int z, int w);


void free1dArray(double *A);
```

```
void free2dArray(double **A, int x);  
void free3dArray(double ***A, int x, int y);  
void free4dArray(double ****A, int x, int y, int z);
```

```
double calcP(int x1, int x2, int f);  
double getP(int x1, int x2, int f);  
void setP(int x1, int x2, int f, double p);  
void printP();
```

```
void getD();  
void printD();
```

```
void getF();  
void printF();  
void printStartF();  
void inverse(double **A, double **B);
```

```
void getA(int x1, int x2, int f);  
void printA();  
double getAi(int x1, int x2, int f, int I);
```

```
void pickTable(int x1, int x2, int f, double *B);  
void mulTable(double *G, double k);  
void minusTable(double *G, double *B);  
void storeTable(int x1, int x2, int f, double *B);
```

```
double calcCost(int S, int r, int c1, int c2, int print);
```

```
void calcAll();
```

```
int N, M, S, r, c1, c2, absc1, absc2;
```

```
double l, LS1, LS2, H, temp, B1, B2, Ctotal, total0=0, total1=0,
```

```
****A,
```

```
***P,
```

```
**F,
```

```
**F1,
```

```
*D;
```

```
int main(){
```

```
    calcAll();
```

```
    return 0;
```

```
}
```

```
void calcAll(){
```

```
    int S, minS = 2, maxS = 15,
```

```
        r, minr = 0, maxr,
```

```
        c1, minc1 = -10, maxc1 = -2,
```

```
        c2, c2offset = 2,
```

```
        i, current, best, allCosts = 300000;
```

```

double ** costs = create2dArray(allCosts,5), bestCost;

current = 0;

for(S=minS;S<=maxS;S++){

    maxr = S-2;

    for(r=minr;r<=maxr;r++){

        for(c1=minc1;c1<=maxc1;c1++){

            for(c2=c1+c2offset;c2<=maxc1+c2offset;c2++){

                if(r-c2 < 2 || r-c2 >= 8 ) continue;

                double cst = calcCost(S, r, c1, c2, -1);

                //if(cst < 0 ) continue;

                //if(cst > 20) continue;

                double total = total0 + total1;

                if(total > 1 || total < 0.95) continue;

                printf("Costs [%d] (%d,%d,%d,%d) = ",current,S,r,c1,c2);

                costs[current][0] = cst;

                costs[current][1] = (double) S;

                costs[current][2] = (double) r;

                costs[current][3] = (double) c1;

                costs[current][4] = (double) c2;

                printf("%lf\n",costs[current][0]);

                current++;
            }
        }
    }
}

```

```

        }
    }
}

// εύρεση μικρότερου κόστους
best = 0;
for(i=best+1;i<current;i++){
    if(costs[i][0]<costs[best][0]){
        best = i;
    }
}

// βέλτιστα κατώφλια ελέγχου
bestCost = costs[best][0];
S = (int) costs[best][1];
r = (int) costs[best][2];
c1 = (int) costs[best][3];
c2 = (int) costs[best][4];

calcCost(S, r, c1, c2, best);

free2dArray(costs,allCosts);
}

double calcCost(int Sc, int rc, int c1c, int c2c, int print){

```

```
int x1,x2,f;
```

```
S = Sc;
```

```
r = rc;
```

```
c1 = c1c;
```

```
c2 = c2c;
```

```
absc1 = c1 < 0 ? -c1 : c1;
```

```
absc2 = c2 < 0 ? -c2 : c2;
```

```
l = l1+l2;
```

```
N = (r+absc2+1)*2;
```

```
M = S+absc1+1;
```

```
A = create4dArray(M,N,2,N);
```

```
P = create3dArray(M,N,2);
```

```
F = create2dArray(N,N);
```

```
F1 = create2dArray(N,N);
```

```
D = create1dArray(N);
```

```
// υπολογισμός A με συγκεκριμένη διαπέραση
```

```
// πρώτες κάθετες στο Γ
```

```
for(x1=c1;x1<=c2-1;x1++){
```

```
    for(x2=r+absc2;x2>=0;x2--){
```

```
        for(f=0;f<=1;f++){
```

```
            getA(x1,x2,f);
```

```

    }
}
}

// τρίγωνο στο Γ
for(x1=c2;x1<=r;x1++){
    for(x2=r+absc2;x2>=x1+absc2;x2--){
        for(f=0;f<=1;f++){
            getA(x1,x2,f);
        }
    }
}

```

```

// διαγώνιοι πάνω από το Γ (B)
for(x2=r+absc2-1;x2>=0;x2--){
    for(x1=0;x1<=x2;x1++){
        for(f=0;f<=1;f++){
            getA(r-x1,x2-x1,f);
        }
    }
}

```

```

// σημείο r
for(f=0;f<=1;f++){
    getA(r,0,f);
}

```

```

// περιοχή A
for(x1=r+1;x1<=S;x1++){
    for(f=0;f<=1;f++){
        getA(x1,0,f);
    }
}

getF();
getD();

total0 = 0;
total1 = 0;

for(x1=c1;x1<=S;x1++){
    if(x1<=r){
        for(x2=r+absc2;x2>=0;x2--){
            total1 += calcP(x1,x2,1);
            total0 += calcP(x1,x2,0);
        }
    }else{
        total1 += calcP(x1,0,1);
        total0 += calcP(x1,0,0);
    }
}

//printf("\ntotal1 = %.50lf\n",total1);
//printf("\ntotal0 = %.50lf\n",total0);

```

```
//printf("\nTOTAL = %.50lf\n",total1+total0);
```

```
// υπολογισμός συνάρτησης κόστους
```

```
LS1=0;
```

```
LS2=0;
```

```
H=0;
```

```
temp=0;
```

```
B1=0;
```

```
B2=0;
```

```
for(x2=0;x2<=r+absc2;x2++){
```

```
    for(f=0;f<=1;f++){
```

```
        LS1 += getP(c1,x2,f);
```

```
    }
```

```
}
```

```
LS1 *= l1;
```

```
for(x2=0;x2<=r+absc2;x2++){
```

```
    for(x1=c1;x1<=c2+x2;x1++){
```

```
        for(f=0;f<=1;f++){
```

```
            LS2 += getP(x1,x2,f);
```

```
        }
```

```
    }
```

```
}
```

```
LS2 *= l2;
```

```

for(x1=S;x1>r;x1--){
    temp = 0;
    for(f=0;f<=1;f++){
        temp += getP(x1,0,f);
    }
    H += temp * x1;
}

for(x1=r;x1>=1;x1--){
    temp = 0;
    for(x2=0;x2<=r+absc2;x2++){
        for(f=0;f<=1;f++){
            temp += getP(x1,x2,f);
        }
    }
    H += temp * x1;
}

for(x1=-1;x1>=c1;x1--){
    temp = 0;
    for(x2=0;x2<=r+absc2;x2++){
        for(f=0;f<=1;f++){
            temp += getP(x1,x2,f);
        }
    }
    B1 += temp * -x1;
}

```

```

for(x2=1;x2<=r+absc2;x2++){
    temp = 0;
    for(x1=c1;x1<=S;x1++){
        for(f=0;f<=1;f++){
            temp += getP(x1,x2,f);
        }
    }
    B2 += temp * x2;
}

Ctotal = P1*LS1+P2*LS2+h*H+b1*B1+b2*B2;

if(print > -1){

    //printA();
    //printStartF();
    //printF();
    //printD();

    printP();

    printf("\nbest position = %d\n\n",print);
    printf("best cost = %.50lf\n\n", Ctotal);

    printf("best S = %d\n", S);
    printf("best r = %d\n", r);
    printf("best c1 = %d\n", c1);

```

```

    printf("best c2 = %d\n\n", c2);

    printf("best LS1 = %.50lf\n", LS1);
    printf("best LS2 = %.50lf\n", LS2);
    printf("best H = %.50lf\n", H);
    printf("best B1 = %.50lf\n", B1);
    printf("best B2 = %.50lf\n\n", B2);

}

free4dArray(A,M,N,2);
free3dArray(P,M,N);
free2dArray(F,N);
free2dArray(F1,N);
free1dArray(D);

return Ctotal;
}

// υπολογισμός πιθανοτήτων
double calcP(int x1, int x2, int f){
    int i;
    double p = 0;
    for(i=0;i<N;i++){
        p += getAi(x1,x2,f,i)*D[i];
    }
    setP(x1,x2,f,p);
}

```

```

        return p;
    }

// εμφάνιση πιθανοτήτων
void printP(){
    int i,j,k;
    for(i=0;i<M;i++){
        for(j=0;j<N;j++){
            for(k=0;k<2;k++){
                printf("P(%d,%d,%d) = %.50lf\n",i,j,k,getP(i+1,j,k));
            }
        }
    }
}

double getP(int x1, int x2, int f){
    return P[x1-1][x2][f];
}

void setP(int x1, int x2, int f, double p){
    P[x1-1][x2][f]=p;
}

// υπολογισμός διανύσματος D
void getD(){
    double *W = create1dArray(N);
    int i,j;

```

```

W[N-1]=1.0d;

for(i=0;i<N;i++){

    D[i] = 0;

    for(j=0;j<N;j++){

        D[i] += F[i][j]*W[j];

    }

}

free1dArray(W);

}

// εμφάνιση πίνακα D
void printD(){

    int j;

    for(j=0;j<N;j++){

        printf("D[%d] = %.50lf\n",j,D[j]);

    }

}

// υπολογισμός πίνακα F
void getF(){

    int i, j, k, L;

    for(j=0;j<N;j++){

        for(i=0;i<N;i++){

            F1[j][i]=0;

        }

    }

}

```

```

// απο 0 εως 1
for(i=0;i<N;i++){

    F1[0][i]=getAi(1+c2,0,0,i)*l1-getAi(c2,0,0,i)*(l1+e)+getAi(c2,0,1,i)*b;

    F1[1][i]=getAi(1+c2,0,1,i)*l1-getAi(c2,0,1,i)*(l1+b+m)+getAi(c2,0,0,i)*e+getAi(c2-
1,0,1,i)*m;

}

// από 2 εως N-3
for(j=0;j<=r+absc2-2;j++){

    int x1=j+c2+1;

    for(i=0;i<N;i++){

        F1[2*(j+1)][i]=getAi(x1,0,0,i)*(l+e)-getAi(x1+1,0,0,i)*l1-getAi(x1,0,1,i)*b;

        F1[2*(j+1)+1][i]=getAi(x1,0,1,i)*(l+m+b)-getAi(x1+1,0,1,i)*l1-getAi(x1,0,0,i)*e-
getAi(x1-1,0,1,i)*m;

    }

}

// N-2
for(i=0;i<N;i++){

    F1[N-2][i]=getAi(S,0,1,i)*(l+b)-getAi(S-1,0,1,i)*m-getAi(S,0,0,i)*e;

}

// N-1
for(k=c1;k<=r;k++){

    for(L=0;L<=absc2+r;L++){

        for(j=0;j<=1;j++){

            for(i=0;i<N;i++){

                F1[N-1][i] += getAi(k,L,j,i);

```

```

        }
    }
}

// N-1
for(k=r+1;k<=S;k++){
    for(j=0;j<=1;j++){
        for(i=0;i<N;i++){
            F1[N-1][i] += getAi(k,0,j,i);
        }
    }
}

inverse(F1, F);
}

void printF(){
    int i,j;
    printf("antistrofos F:\n");
    for(j=0;j<N;j++){
        for(i=0;i<N;i++){
            printf("F[%d][%d] = %.50lf\n",j,i,F[j][i]);
        }
    }
}
}

```

```

void printStartF(){
    int i,j;

    printf("kanonikos F:\n");

    for(j=0;j<N;j++){
        for(i=0;i<N;i++){
            printf("F[%d][%d] = %.50lf\n",j,i,F1[j][i]);
        }
    }
}

```

// υπολογισμός αντίστροφου πίνακα F

```

void inverse(double **A, double **B){

    int i, j, k;

    double num, **C = create2dArray(N,N);

    for(i=0;i<N;i++){
        for(j=0;j<N;j++){
            if(i==j){
                B[i][j]=1;
            }else{
                B[i][j]=0;
            }

            C[i][j]=A[i][j];
        }
    }
}

```

```

// για κάθε γραμμή
for(i=0;i<N;i++){

    // μετατρέπουμε τα στοιχεία της διαγωνίου σε 1
    num = C[i][i];

    for(j=0;j<N;j++){

        C[i][j] /= num;

        B[i][j] /= num;

    }

    // γραμμοπράξεις κάτω
    for(k=i+1;k<N;k++){

        num = C[k][i];

        for(j=0;j<N;j++){

            C[k][j] -= C[i][j] * num;

            B[k][j] -= B[i][j] * num;

        }

    }

    // γραμμοπράξεις πάνω
    for(k=i-1;k>=0;k--){

        num = C[k][i];

        for(j=0;j<N;j++){

            C[k][j] -= C[i][j] * num;

            B[k][j] -= B[i][j] * num;

        }

    }

```

```

    }

    free2dArray(C,N);
}

double getAi(int x1, int x2, int f, int l){
    if(l >= N || l < 0){
        return 0;
    }else{
        return A[x1-c1][x2][f][l];
    }
}

// κατασκευή διανύσματος A
void getA(int x1, int x2, int f){
    int i;

    double *U = create1dArray(N);
    double *P = create1dArray(N);

    if(x1 >= c1 && x2 <= r+absc2){
        if(x1 > r+1 && x1 <= S && x2 == 0){
            // περιοχή A
            if(f == 1){

                pickTable(x1-1,x2,1,U);

                mulTable(U,l+m+b);
            }
        }
    }
}

```

```

    pickTable(x1-2,x2,1,P);

    mulTable(P,m);


    minusTable(U,P);


    pickTable(x1-1,x2,0,P);

    mulTable(P,e);


    minusTable(U,P);

    mulTable(U,1/l);


}else if(f == 0){


    pickTable(x1-1,x2,0,U);

    mulTable(U,l+e);


    pickTable(x1-1,x2,1,P);

    mulTable(P,b);


    minusTable(U,P);

    mulTable(U,1/l);

}

}else if(x1 == r+1 && x2 == 0){

// κατάσταση 3

    if(f == 1){

```

pickTable(x1-1,x2,1,U);

mulTable(U,l+m+b);

pickTable(x1-2,x2,1,P);

mulTable(P,m);

minusTable(U,P);

pickTable(x1-1,x2+1,1,P);

mulTable(P,m);

minusTable(U,P);

pickTable(x1-1,x2,0,P);

mulTable(P,e);

minusTable(U,P);

mulTable(U,1/l);

}else if(f == 0){

pickTable(x1-1,x2,0,U);

mulTable(U,l+e);

pickTable(x1-1,x2,1,P);

mulTable(P,b);

```

        minusTable(U,P);

        mulTable(U,1/l);

    }

}

else if(x1 == r && x2 == 0){

// σημείο r

    if(f == 1){

        pickTable(x1,x2+1,1,U);

        mulTable(U,l+m+b);

        pickTable(x1,x2+2,1,P);

        mulTable(P,m);

        minusTable(U,P);

        pickTable(x1-1,x2+1,1,P);

        mulTable(P,m);

        minusTable(U,P);

        pickTable(x1,x2+1,0,P);

        mulTable(P,e);

        minusTable(U,P);

        mulTable(U,1/l2);

    }

}

else if(f == 0){

```

```

    pickTable(x1,x2+1,0,U);
    mulTable(U,l+e);

    pickTable(x1,x2+1,1,P);
    mulTable(P,b);

    minusTable(U,P);
    mulTable(U,1/l2);
}
}else if(x1 == r && x2 == absc2+r-1){
// σημείο x2 = 5
    if(f == 1){

        pickTable(x1,x2+1,1,U);
        mulTable(U,l1+b+m);

        pickTable(x1-1,x2+1,1,P);
        mulTable(P,m);

        minusTable(U,P);

        pickTable(x1,x2+1,0,P);
        mulTable(P,e);

        minusTable(U,P);
        mulTable(U,1/l2);

```

```

}else if(f == 0){

    pickTable(x1,x2+1,0,U);

    mulTable(U,l1+e);


    pickTable(x1,x2+1,1,P);

    mulTable(P,b);


    minusTable(U,P);

    mulTable(U,1/l2);

}

}else if(x1 == r && x2 < absc2+r-1){

// περιοχή B1

    if(f == 1){


        pickTable(x1,x2+1,1,U);

        mulTable(U,l+m+b);


        pickTable(x1,x2+2,1,P);

        mulTable(P,m);


        minusTable(U,P);


        pickTable(x1-1,x2+1,1,P);

        mulTable(P,m);

```

```

        minusTable(U,P);

        pickTable(x1,x2+1,0,P);

        mulTable(P,e);

        minusTable(U,P);

        mulTable(U,1/l2);

    }else if(f == 0){

        pickTable(x1,x2+1,0,U);

        mulTable(U,l+e);

        pickTable(x1,x2+1,1,P);

        mulTable(P,b);

        minusTable(U,P);

        mulTable(U,1/l2);

    }

    }else if(x1 > c2 && x2 == x1+absc2-1){

// περιοχή B4 (από πάνω)

        if(f == 1){

            pickTable(x1,x2+1,1,U);

            mulTable(U,l1+m+b);

            pickTable(x1+1,x2+1,1,P);

```

mulTable(P,l1);

minusTable(U,P);

pickTable(x1-1,x2+1,1,P);

mulTable(P,m);

minusTable(U,P);

pickTable(x1,x2+1,0,P);

mulTable(P,e);

minusTable(U,P);

mulTable(U,1/l2);

}else if(f == 0){

pickTable(x1,x2+1,0,U);

mulTable(U,l1+e);

pickTable(x1+1,x2+1,0,P);

mulTable(P,l1);

minusTable(U,P);

pickTable(x1,x2+1,1,P);

mulTable(P,b);

```

        minusTable(U,P);

        mulTable(U,1/l2);

    }

} else if (x1 > c2 && x2 < x1+absc2-1){

// περιοχή B

    if(f == 1){

        pickTable(x1,x2+1,1,U);

        mulTable(U,l+m+b);


        pickTable(x1+1,x2+1,1,P);

        mulTable(P,l1);


        minusTable(U,P);


        pickTable(x1-1,x2+1,1,P);

        mulTable(P,m);


        minusTable(U,P);


        pickTable(x1,x2+1,0,P);

        mulTable(P,e);


        minusTable(U,P);

        mulTable(U,1/l2);

    }

} else if (f == 0){

    pickTable(x1,x2+1,0,U);

```

```

mulTable(U,l+e);

pickTable(x1+1,x2+1,0,P);
mulTable(P,l1);

minusTable(U,P);

pickTable(x1,x2+1,1,P);
mulTable(P,b);

minusTable(U,P);
mulTable(U,1/l2);
}
}else if(x1 > c1+1){ // x>=c2
// περιοχή Γ (Τρίγωνο πάνω από Β)
if(f == 1){
pickTable(x1-1,x2,1,U);
mulTable(U,l1+b+m);

pickTable(x1-2,x2,1,P);
mulTable(P,m);

minusTable(U,P);

pickTable(x1-1,x2,0,P);
mulTable(P,e);

```

```

        minusTable(U,P);

        mulTable(U,1/l1);


    }else if(f == 0){

        pickTable(x1-1,x2,0,U);

        mulTable(U,l1+e);


        pickTable(x1-1,x2,1,P);

        mulTable(P,b);


        minusTable(U,P);

        mulTable(U,1/l1);

    }

    }else if(x1 == c1+1){

// περιοχή Γ (προτελευταία στήλη που υπολογίζεται από την τελευταία)

        if(f == 1){

            pickTable(x1-1,x2,1,U);

            mulTable(U,b+m);


            pickTable(x1-1,x2,0,P);

            mulTable(P,e);


            minusTable(U,P);

            mulTable(U,1/l1);


        }else if(f == 0){

            pickTable(x1-1,x2,0,U);

```

```

        mulTable(U,e);

        pickTable(x1-1,x2,1,P);

        mulTable(P,b);

        minusTable(U,P);

        mulTable(U,1/l1);
    }
}
else if(x1 == c1){
    // περιοχή Γ (τελευταία στήλη)

    if(f == 1){
        U[2*x2] = 1;
    }else if(f == 0){
        U[2*x2+1] = 1;
    }
}

storeTable(x1,x2,f,U);
}

free1dArray(U);

free1dArray(P);
}

// εμφάνιση πίνακα A
void printA(){
    int x1,x2,f,i;

    for(x1=c1;x1<=r;x1++){
        for(x2=0;x2<=r+absc2;x2++){
            for(f=1;f>=0;f--){

```

```

        printf("A(%d,%d,%d)=[\n",x1,x2,f);
        for(i=0;i<N;i++){
            printf("\n\t%.50lf",getAi(x1,x2,f,i));
        }
        printf("\n]\n");
    }
}

for(x1=r+1;x1<=S;x1++){
    for(f=1;f>=0;f--){
        printf("A[%d][%d][%d]=[\n",x1,0,f);
        for(i=0;i<N;i++){
            printf("\n\t%.50lf",getAi(x1,0,f,i));
        }
        printf("\n]\n");
    }
}
}

```

// πράξεις πινάκων

```

void pickTable(int x1, int x2, int f, double *B){
    int i;
    for(i=0;i<N;i++){
        B[i]=A[x1-c1][x2][f][i];
    }
}

```

```

void mulTable(double *G, double k){
    int i;
    for(i=0;i<N;i++){
        G[i]=G[i]*k;
    }
}

```

```

void minusTable(double *G, double *B){
    int i;
    for(i=0;i<N;i++){
        G[i]=G[i]-B[i];
    }
}

```

```

void storeTable(int x1, int x2, int f, double *B){
    int i;
    for(i=0;i<N;i++){
        A[x1-c1][x2][f][i]=B[i];
    }
}

```

// δημιουργία και απελευθέρωση δυναμικών πινάκων

```

double * create1dArray(int size){
    int i;
    double *A = (double *)malloc(sizeof(double) * size);
    for(i=0;i<size;i++){

```

```

        A[i]=0;
    }
    return A;
}

```

```

void free1dArray(double *A){
    free(A);
}

```

```

double ** create2dArray(int x, int y){
    int i;

    double **A = (double **)malloc(sizeof(double *) * x);

    for (i = 0; i < x; i++){
        A[i] = create1dArray(y);
    }

    return A;
}

```

```

void free2dArray(double **A, int x){
    int i;

    for (i = 0; i < x; i++){
        free(A[i]);
    }
}

```

```

    free(A);
}

double *** create3dArray(int x, int y, int z){
    int i;

    double ***A = (double ***)malloc(sizeof(double **) * x);

    for (i = 0; i < x; i++){
        A[i] = create2dArray(y,z);
    }

    return A;
}

void free3dArray(double ***A, int x, int y){
    int i,j;

    for (i = 0; i < x; i++){
        for (j = 0; j < y; j++){
            free(A[i][j]);
        }
        free(A[i]);
    }
    free(A);
}

```

```

double **** create4dArray(int x, int y, int z, int w){
    int i;

    double ****A = (double ****)malloc(sizeof(double ***) * x);

    for (i = 0; i < x; i++){
        A[i] = create3dArray(y,z,w);
    }

    return A;
}

void free4dArray(double ****A, int x, int y, int z){
    int i,j,k;

    for (i = 0; i < x; i++){
        for (j = 0; j < y; j++){
            for (k = 0; k < z; k++){
                free(A[i][j][k]);
            }
            free(A[i][j]);
        }
        free(A[i]);
    }
    free(A);
}

```