ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΣΥΝΤΟΝΙΣΜΕΝΕΣ ΠΟΛΙΤΙΚΕΣ ΕΛΕΓΧΟΥ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΠΑΡΑΓΩΓΗΣ ΣΕ ΕΡΓΟΣΤΑΣΙΑ ΜΕ ΠΟΙΚΙΛΙΑ ΑΓΟΡΑΣΤΩΝ

Διατριβή που υπεβλήθη για την μερική ικανοποίηση των απαιτήσεων για την απόκτηση Μεταπτυχιακού Διπλώματος Ειδίκευσης

υπό

Γιώργου Γ. Λοΐζου

Χανιά, 2005

© Copyright υπό Γιώργου Γ. Λοΐζου, 2005

Η διατριβή του Γιώργου Λοΐζου, εγκρίνεται από τους Βασίλη Κουϊκόγλου (επιβλέποντα), Νικόλαο Τσουρβελούδη και Ευάγγελο Γρηγορούδη.

1) Βασίλης Κουϊκόγλου (επιβλέπων)

2) Τσουρβελούδης Νικόλαος

3) Ευάγγελος Γρηγορούδης

Περιεχόμενα

Ευχαριστώ ιδιαίτερα τον καθηγητή μου, κ. Βασίλη Κουϊκόγλου για την αμέριστη βοήθεια και συμπαράσταση που μου πρόσφερε. Ευχαριστώ επίσης το Στράτο Ιωαννίδη για τη φιλία και τη βοήθειά του όταν τη χρειαζόμουν.

Ο Γιώργος Λοΐζος γεννήθηκε στην Αθήνα το 1974. Έχει πτυχίο Μηχανικού Παραγωγής και Διοίκησης από το Πολυτεχνείο Κρήτης και ολοκληρώνει το μεταπτυχιακό πρόγραμμα σπουδών «Συστήματα Παραγωγής» του τμήματος Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης. Ο τομέας της έρευνάς του είναι η ανάλυση και ο έλεγχος συστημάτων παραγωγής.

Περίληψη

Στην εργασία αυτή εξετάζουμε ένα πρόβλημα ελέγχου ποιότητας και ελέγχου αποθεμάτων για ένα σύστημα παραγωγής που διαθέτει το προϊόν του σε διαφορετικές αγορές συγχρόνως. Τα προϊόντα έχουν ένα ποιοτικό χαρακτηριστικό το οποίο είναι μετρίσιμο και για το οποίο υπάρχει ιδανική τιμή (πχ. διάμετρος πιστονιού). Τυχών αποκλίσεις απο αυτήν την τιμή οδηγούν σε ένα κόστος ποιότητας το οποίο προκύπτει από επιστροφές ελαττωματικών προϊόντων για επιδιορθώσεις ή αντικατάσταση, κόστος δυσφήμισης, κλπ τα οποία διαφέρουν απο αγορά σε αγορά.

Οι Hong και Elsayed (1998) ταξινομούν τα προϊόντα μιας μονάδας παραγωγής με βάση την απόκλιση ενός ποιοτικού τους χαρακτηριστικού απο την ιδανική τιμή. Τα κομμάτια κάθε κατηγορίας K(i) πωλούνται σε διαφορετική αγορά A(i). Όσο μικρότερη είναι η απόκλιση των κομματιών της κατηγορίας από την ιδανική τιμή του χαρακτηριστικού, σε τόσο απαιτητικότερες αγορές πωλούνται. Όσο πιο απαιτητική είναι μια αγορά, τόσο πιο ακριβά αγοράζει το προϊόν, αλλά έχει και μεγαλύτερο κόστος ποιότητας για την εταιρία.

«Κατώφλι ποιότητας» ονομάζουμε τη μέγιστη απόκλιση ενός ποιοτικού χαρακτηριστικού των παραγόμενων κομματιών από την ιδανική τιμή, πέραν της οποίας ένα κομμάτι κατατάσσεται στην επόμενη κατηγορία ποιότητας ή απορρίπτεται ως προϊόν διαλογής. Στην εργασία τους οι Hong και Elsayed υπολογίζουν τα «κατώφλια ποιότητας» που διαχωρίζουν τις διαλογές, έτσι ώστε η ταξινόμηση των προϊόντων να αποδώσει το μέγιστο κέρδος απο την πώληση τους στις διαθέσιμες αγορές. Λαμβάνουν υπόψη τους το κόστος ποιότητας, το κόστος απόρριψης κομματιού, και το κόστος ποιοτικού ελέγχου του κάθε κομματιού. Από την κάθε κατηγορία ποιότητας K(i) επιτρέπεται να πωληθούν προϊόντα μόνο στην αντίστοιχη αγορά A(i). Όσο πιο καλή η κατηγορία ποιότητας, σε τόσο πιο απαιτητική αγορά πωλούνται τα προϊόντα αυτής της κατηγορίας.

Στην εργασία αυτή εξετάζουμε και το κόστος αποθέματος. Τροποποιούμε την προηγούμενη πολιτική, ώστε να επιτρέπεται η πώληση προϊόντων απο μια κατηγορία ποιότητας K(i) όχι μόνο προς την αντίστοιχη αγορά A(i) , αλλά και στην αμέσως λιγότερο απαιτητική αγορά A(i+1). Από αριθμητικά πειράματα προκύπτει ότι

το κέρδος που αποκομίζει η εταιρία είναι μεγαλύτερο απο εκείνο της πολιτικής των Hong και Elsayed.

1 Εισαγωγή

1.1 Αντικείμενο της διατριβής

Η εξέλιξη στην τεχνολογία του αυτοματισμού και των διαδικασιών ελέγχου που εφαρμόζονται στις παραγωγικές μονάδες έχει οδηγήσει τα τελευταία χρόνια σε μια ριζική αλλαγή της νοοτροπίας σε ότι αφορά τη συσχέτιση της μεθοδολογίας του ποιοτικού ελέγχου με την πολιτική προώθησης των παραγόμενων προϊόντων.

Η διαδικασία του ποιοτικού ελέγχου έχει γίνει πολύ ταχύτερη και πολύ οικονομικότερη, με αποτέλεσμα όλο και περισσότερες εταιρίες παραγωγής εξαρτημάτων να εγκαταλείπουν την παραδοσιακή διαδικασία δειγματοληπτικού ελέγχου, και να υιοθετούν τη διαδικασία εξαντλητικού ελέγχου. Στην περίπτωση αυτή όλα τα παραγόμενα κομμάτια εξετάζονται ένα προς ένα αν πληρούν τις προδιαγραφές που ορίζονται από την εταιρία, οι οποίες στην πραγματικότητα επιβάλλονται είτε από το νόμο της αγοράς, είτε από ελεγκτικούς μηχανισμούς.

Ο εξαντλητικός έλεγχος γίνεται επιτακτικός όταν η αστοχία των παραγόμενων προϊόντων που καταλήγουν στον καταναλωτή είναι ιδιαίτερα επιβλαβής για την εταιρία, ενώ η τιμή πώλησής τους αρκετά μεγαλύτερη από το κόστος ελέγχου ανά κομμάτι. «Ο εξαντλητικός έλεγχος αναδεικνύεται σε μια ελκυστική πρακτική για τις εταιρίες, καθώς εξασφαλίζει την απομάκρυνση των ακατάλληλων κομματιών και αναπόφευκτα θα γίνει απαραίτητος στις σύγχρονες παραγωγικές μονάδες» (Tang and Tang 1994).

Θεωρώντας ως δεδομένη την ύπαρξη διαδικασίας εξαντλητικού ελέγχου σε μια παραγωγική μονάδα, είναι δυνατό να εφαρμοστεί περαιτέρω μια άλλη διαδικασία, η οποία συσχετίζει τον ποιοτικό έλεγχο με την αύξηση του κέρδους από τις πωλήσεις της εταιρίας. Η διαδικασία αυτή εκμεταλλεύεται το γεγονός ότι υπάρχει μια ποικιλία αγορών, με διαφορετικές απαιτήσεις, που είναι διατεθειμένες να πληρώσουν για το προϊόν τιμή ανάλογη με τις απαιτήσεις τους.

Εφαρμόζεται λοιπόν μια πολιτική διάθεσης των παραγόμενων προϊόντων στις υπάρχουσες αγορές αντιστοιχώντας το κάθε - επιθεωρημένο ποιοτικά - προϊόν στην κατάλληλη αγορά, ώστε να εξασφαλίζεται το μέγιστο κέρδος για την επιχείρηση, μεγιστοποιώντας τα έσοδα από τις πωλήσεις και ελαχιστοποιώντας την δυσαρέσκεια των πελατών.

1.2 Βιβλιογραφική ανασκόπηση

Από τη δεκαετία του 1980 γνωστοί ερευνητές έχουν προτείνει οικονομικά μοντέλα τα οποία βασίζονται σε πολιτικές εξαντλητικού έλεγχού βάση της επίδοσης των προϊόντων σε συγκεκριμένο χαρακτηριστικό. Τρεις συνιστώσες κόστους λαμβάνονται κυρίως υπόψη σε αυτές τις μελέτες. Αυτές είναι το κόστος ποιότητας, που αναφέρεται στη οικονομική απώλεια της εταιρίας λόγω της δυσαρέσκειας του πελάτη, εφόσον η ποιότητα του προϊόντος που αγόρασε δεν τον ικανοποιεί, το κόστος απόρριψης ελαττωματικών κομματιών, και το κόστος επιθεώρησης (ποιοτικού ελέγχου) για κάθε κομμάτι, (Tang and Schneider (1987), Tang (1988), Hui (1990), Duffuaa and Al-Najjar (1995), Ng and Hui (1996)).

Οι Hong και Elsayed (1998) ταξινομούν τα προϊόντα μιας γραμμής παραγωγής όχι μόνο σε αποδεκτά και απορριπτέα, αλλά και σε κατηγορίες ποιότητας (διαλογές), με βάση την απόκλιση απο την ιδανική τιμή, που παρουσιάζει το ποιοτικό χαρακτηριστικό που ελέγχεται κατά τον εξαντλητικό έλεγχο. Με χρήση αυτής της νέας πληροφορίας οι Hong και Elsayed προτείνουν μια πολιτική προώθησης των προϊόντων στις διαθέσιμες αγορές για βελτιστοποίηση του κέρδους απο τις πωλήσεις, συνυπολογίζοντας τις παραπάνω συνιστώσες κόστους, αλλά χωρίς να λάβουν υπόψη τους το κόστος αποθεματοποίησης. Θεωρούν δηλαδή ότι όταν παράγεται το προϊόν τότε πωλείται αμέσως.

Η παρούσα εργασία προτείνει μια βελτιωμένη εκδοχή της πολιτικής των Hong και Elsayed καθώς εισάγει στη συνάρτηση του υπολογισμού του κέρδους και το κόστος αποθέματος. Λίγες είναι οι εταιρίες που παράγουν προϊόντα μόνο κατά παραγγελία, ενώ οι περισσότερες διατηρούν αποθέματα ασφαλείας για τα προϊόντα τους. Επομένως το κόστος αποθέματος είναι πολύ ρεαλιστικός παράγοντας καθορισμού του συνολικού κέρδους μιας επιχείρησης και πρέπει να ληφθεί υπ’όψιν. Αποδεικνύεται ότι στην περίπτωση συνυπολογισμού και του κόστους αποθέματος, η νέα πολιτική αποδίδει πάντοτε μεγαλύτερο κέρδος από αυτή των Hong και Elsayed.

1.3 Δομή της διατριβής

Στο Κεφάλαιο 2 που ακολουθεί, γίνεται μια αναλυτική περιγραφή του προβλήματος καταδεικνύοντας όλες τις παραμέτρους. Παρουσιάζονται οι δύο εξεταζόμενες πολιτικές προώθησης των προϊόντων, η πολιτική των Hong και Elsayed και αυτή που προτείνεται στην παρούσα διατριβή, διασαφηνίζοντας τις ομοιότητες και τις διαφορές τους .

Στο Κεφάλαιο 3 γίνεται παρουσίαση των αλγόριθμων που υπολογίζουν διάφορα μέτρα απόδοσης του συστήματος και για τις δύο πολιτικές, ώστε να είναι δυνατό να συγκριθεί το μέσο κέρδος που αποδίδει η κάθε μια. Οι αλγόριθμοι βασίζονται σε θεωρίες ουρών αναμονής και περιλαμβάνουν μαθηματικές σχέσεις που υπολογίζουν το κέρδος και τις συνιστώσες της κάθε πολιτικής.

Στο Κεφάλαιο 4 παρουσιάζονται τα αποτελέσματα των πειραμάτων που έγιναν με τη χρήση των παραπάνω αλγόριθμων προσομοίωσης. Γίνεται σύγκριση του μέσου κέρδους των δύο πολιτικών, καθώς και διερεύνηση της επίδρασης που έχουν στο κέρδος οι παράμετροι του προβλήματος. Τα αποτελέσματα δείχνουν ότι στην περίπτωση που λαμβάνουμε υπόψη το κόστος αποθέματος η νέα πολιτική είναι αποδοτικότερη. Η διατριβή καταλήγει με τα συμπεράσματα και ένα παράρτημα με τους κώδικες των προγραμμάτων.

2 Πολιτικές διάθεσης προϊόντων στις κατάλληλες αγορές με βάση την απόκλιση απο την ιδανική τιμή.

2.1 Περιγραφή του συστήματος παραγωγής

Έστω ένα σύστημα παραγωγής που παράγει ένα προϊόν και το διαθέτει σε δύο αγορές (Αγορά 1 και Αγορά 2). Το προϊόν χαρακτηρίζεται από ένα ποιοτικό χαρακτηριστικό y που έχει ιδανική τιμή t (πχ. διάμετρος πιστονιού). Το σύστημα παράγει κομμάτια που η τιμή του ποιοτικού χαρακτηριστικού δίνεται από γνωστή συνάρτηση πυκνότητας πιθανότητας f(y) (πχ. κανονική κατανομή με γνωστές μέση τιμή α και διασπορά σ^2).

Το κόστος ποιότητας είναι η χρηματική απώλεια της εταιρίας όταν ο πελάτης δεν είναι ικανοποιημένος από την ποιότητα του προϊόντος που αγόρασε. Η χρηματική απώλεια αφορά την περίπτωση που ο πελάτης επιστρέφει το προϊόν, αλλά και το - πολύ δύσκολα μετρήσιμο - κόστος της δυσφήμισης. Για να διευκολυνθεί η δυνατότητα μελέτης πολιτικών διάθεσης των προϊόντων ανάλογα με την ποιότητά τους, το κόστος ποιότητας ορίζεται ως συνάρτηση μετρήσιμων μεγεθών. Η συναρτησιακή μορφή του κόστους ποιότητας υιοθετείται πολύ συχνά τα τελευταία χρόνια και αποτελεί την κεντρική υπόθεση της φιλοσοφίας ποιότητας σύμφωνα με τον Taguchi.

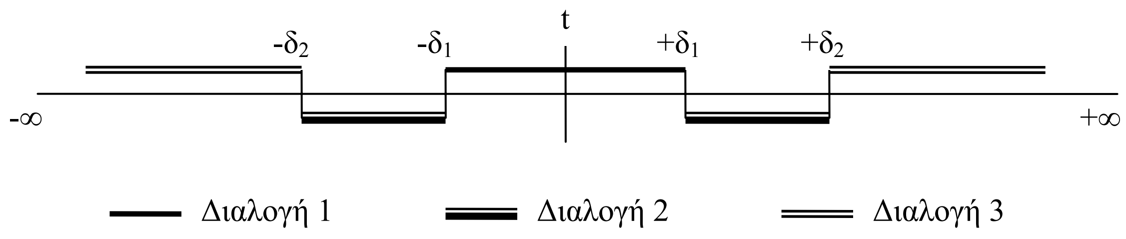
Το κόστος ποιότητας λόγω απόκλισης του ποιοτικού χαρακτηριστικού από την ιδανική τιμή t υποθέτουμε ότι είναι $b(y-t)^2$ ανά κομμάτι που παράγεται, για το οποίο το ποιοτικό χαρακτηριστικό του έχει τιμή y, και b είναι μια σταθερά. Δηλαδή όσο πιο μεγάλη η απόκλιση τόσο μεγαλύτερη η πιθανότητα αστοχίας του προϊόντος οπότε και η πιθανότητα μη ικανοποίησης του πελάτη, με αποτέλεσμα να ζημιώνεται το σύστημα παραγωγής.

Τώρα υποθέτουμε ότι το σύστημα παράγει κομμάτια με μέσο ρυθμό μ. Επίσης, οι δύο αγορές (1 και 2) έχουν διαφορετικά χαρακτηριστικά κέρδους. Η αγορά i προσφέρει κέρδος A_i ανά κομμάτι και έχει παράμετρο κόστους ποιότητας b_i. Υπάρχει επίσης η δυνατότητα απόρριψης κομματιού (ή πώλησης σε χαμηλή τιμή) αν η απόκλιση είναι μεγάλη. Τη δυνατότητα αυτή την περιγράφουμε με μία φανταστική αγορά 3 που προσφέρει κέρδος A_3 (μπορεί να ισχύει και $A_3 < 0$) με παράμετρο κόστους ποιότητας $b_3 = 0$.

2.2 Η πολιτική των Hong και Elsayed

Οι Hong and Elsayed (1998) έχουν υπολογίσει πού συμφέρει να διαθέτουμε ένα προϊόν ανάλογα με το μέγεθος της απόκλισης. Οι αγορές 1, 2 και 3 προσφέρουν κέρδος A_1, A_2 και A_3. Έχουν ταξινομηθεί οι αγορές έτσι, ώστε $A_1>A_2>A_3$. Αποδεικνύεται ότι για τις παραμέτρους κόστους ποιότητας bi δεν μπορούμε παρά να θεωρήσουμε ότι $b_1>b_2$. Αν δεν συνέβαινε αυτό, τότε θα ίσχυε $b_1 \leq b_2$. Αλλά τότε η αγορά 1 θα πρόσφερε αφενός μεγαλύτερο κέρδος, ενώ αφετέρου θα είχε και μικρότερο κόστος ποιότητας. Άρα δε θα συμφέρει να πουλάμε καθόλου στην αγορά 2. Σε αυτήν την περίπτωση δεν έχει νόημα να διερευνηθούν διαφορετικές πολιτικές, αφού το πιο ευνοϊκό είναι να πωλούνται όλα τα προϊόντα στην αγορά 1. Η αγορά 3 (απόρριψη προϊόντων) έχει συντελεστή κόστους ποιότητας $b_3=0$, αφού το κόστος απόρριψης είναι A_3 για όλα τα προϊόντα, χωρίς να ενδιαφέρει η απόκλισή τους από την ιδανική τιμή t.

Αποδεικνύεται στην εργασία εκείνη ότι υπάρχουν «κατώφλια διαλογής» δ_1 και δ_2, τα οποία οριοθετούν στην πραγματικότητα μια απόσταση από την ιδανική τιμή t. Έτσι όσα προϊόντα βρίσκονται σε απόσταση ίση ή μικρότερη με δ_1 από το t, ανήκουν στην πρώτη διαλογή. Η περιοχή της πρώτης διαλογής είναι η $[t–\delta_1, t+\delta_1]$ ή $[t–\delta_1, t]\cup[t, t+\delta_1]$. Η περιοχή της δεύτερης διαλογής είναι τα κομμάτια που «απέχουν» από την ιδανική τιμή t περισσότερο από δ_1 και λιγότερο από δ_2. Άρα η τιμή y του ποιοτικού χαρακτηριστικού για αυτά τα κομμάτια ανήκει στα διαστήματα $[t–\delta_2, t–\delta_1]\cup[t+\delta_1, t+\delta_2]$. Όταν η απόκλιση από την ιδανική τιμή είναι μεγαλύτερη από δ2 τότε τα κομμάτια ανήκουν στην τρίτη αγορά και η τιμή y για αυτά ανήκει στο $[-\infty, t–\delta_2]\cup [t+\delta_2, +\infty]$. Γραφικά μπορούμε να αποδώσουμε τις διαλογές με το παρακάτω σχήμα:



Σχ. 1. Κατηγορίες προϊόντων σύμφωνα με την απόκλιση δ_i

Διαφορετικά μπορούμε να περιγράψουμε τις διαλογές ως εξής:

Διαλογή 1, y: $0 \leq |y-t| < \delta_1$

Διαλογή 2, y: $\delta_1 \leq |y-t| < \delta_2$

Διαλογή 3, y: $\delta_2 \leq |y-t| < \infty$

Έτσι αν $A_1 > A_2 > A_3$ και $b_1 > b_2$ τότε πουλάμε τα "πολύ καλά κομμάτια" ($0 \leq |y-t| < \delta_1$) στην αγορά 1 , τα λιγότερο καλά ($\delta_1 \leq |y-t| < \delta_2$) στην αγορά 2 και τα άλλα ($\delta_2 \leq |y-t| < \infty$) τα απορρίπτουμε. Τα κατώφλια διαλογής δ_1 και δ_2 υπολογίζονται θεωρητικά στην εργασία του Elsayed, έτσι ώστε να βελτιστοποιείται το κέρδος. Το συνολικό κέρδος Jel είναι συνάρτηση του κέρδους από τις πωλήσεις, του κόστους ποιότητας, του κόστους επιθεώρησης του κάθε κομματιού και του κόστους απόρριψης ελαττωματικών κομματιών.

2.3 Η βελτιωμένη πολιτική

Η λύση που προτείνουν οι ανωτέρω ερευνητές είναι σωστή αν αγνοηθεί το κόστος αποθέματος και το κόστος από την απώλεια πελατών, οπότε θεωρούν ότι οι αγορές 1 και 2 είναι πάντα διαθέσιμες όταν παράγουμε ένα κομμάτι. Για παράδειγμα αν η αγορά 1 (που είναι απαιτητική σε ποιότητα αλλά και πιο κερδοφόρα) δεν έχει μεγάλη ζήτηση, τότε θα συνέφερε ένα καλό προϊόν ($0 \leq |y-t| < \delta_1$) να το προσφέραμε στην 2 αν από αυτήν έχουμε μεγάλη (συχνή) ζήτηση παρά να το αποθηκεύσουμε περιμένοντας την 1.

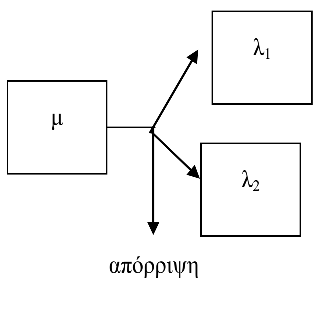
Στη δική μας εργασία κάνουμε τις ίδιες παραδοχές σε ότι αφορά το σύστημα παραγωγής που κάνουν και οι Hong και Elsayed, με μια μετατροπή. Θεωρούμε επιπλέον ότι όταν ένα κομμάτι παράγεται δεν είναι βέβαιο ότι θα υπάρχει διαθέσιμος πελάτης. Οπότε τα παραγόμενα κομμάτια που δεν βρίσκουν αγοραστή πρέπει να αποθηκεύονται έως ότου πουληθούν.

Τώρα λοιπόν πρέπει να εξετάσουμε τα προβλήματα της παραγωγής, ελέγχου ποιότητας και πώλησης προϊόντων από κοινού. Για το πρόβλημα ελέγχου παραγωγής υποθέτουμε ότι μία μονάδα έτοιμου προϊόντος που μένει αποθηκευμένο στην αποθήκη του συστήματος για 1 μονάδα χρόνου κοστίζει h. Οπότε η συνάρτηση του κέρδους συμπεριλαμβάνει και το κόστος αποθέματος. Μία απλή πολιτική είναι η ακόλουθη:

Τα προϊόντα για τα οποία $|y-t|<\delta_2$ ονομάζονται αποδεκτά. Ορίζουμε ένα απόθεμα ασφαλείας s (base stock – απόθεμα βάσης). Παράγουμε κομμάτι. Αν είναι αποδεκτό τότε το αποθηκεύουμε. Αλλιώς το απορρίπτουμε. Για όσο διάστημα το απόθεμα είναι κάτω από S συνεχίζουμε την παραγωγή. Όταν το απόθεμα φτάσει την τιμή S η παραγωγή σταματά. Όταν ζητείται κομμάτι από την αγορά i τότε ελέγχουμε το απόθεμα. Αν στην αποθήκη υπάρχουν διαθέσιμα κομμάτια "κατάλληλης ποιότητας" γι' αυτήν την αγορά τότε διαθέτουμε ένα (μειώνεται το αντίστοιχο απόθεμα). Αν δεν υπάρχουν τότε δεν διαθέτουμε τίποτε και η παραγγελία δε γίνεται δεκτή.

Η κατάσταση του συστήματος περιγράφεται από το ζεύγος (n_1, n_2) όπου n_i είναι ο αποθηκευμένος αριθμός κομματιών κατάλληλων για την αγορά i. Θα πρέπει $0\leq n_1+n_2 \leq S$, αφού το S είναι η χωρητικότητα της αποθήκης.

Οι Hong και Elsayed από κάθε διαλογή i επιτρέπουν την πώληση μόνο στην αντίστοιχη αγορά i. Στην περίπτωση που θεωρηθεί ότι ένα κομμάτι πωλείται αμέσως μόλις παραχθεί, και άρα δεν έχουμε κόστος αποθέματος, τότε η πολιτική Hong και Elsayed μπορεί να παρασταθεί γραφικά κάπως έτσι:

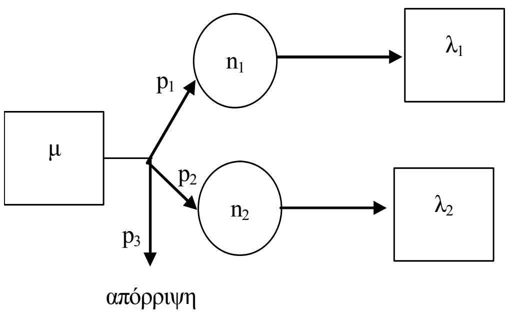


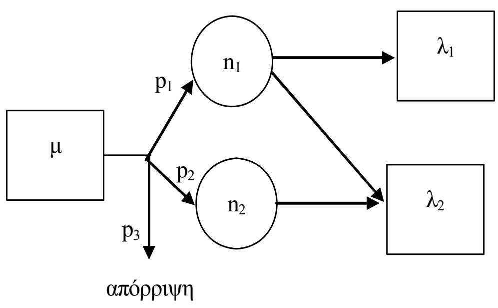
Σχ. 2. Πολιτική Hong και Elsayed για τρεις αγορές χωρίς κόστος αποθέματος

Τώρα όμως προκύπτει το θέμα να παρουσιαστεί πελάτης της αγοράς i και να υπάρχει στην αποθήκη διαθέσιμο προϊόν που προορίζεται για άλλη αγορά.

Σε αυτήν την περίπτωση εξετάζουμε δύο ενδεχόμενα. Αν ο πελάτης προέρχεται από τη αγορά 1 (πιο απαιτητική) και διατίθενται στην αποθήκη μόνο κομμάτια δεύτερης ποιότητας n_2, τότε δεν του πουλάμε, αφού δεν καλύπτονται οι απαιτήσεις ποιότητας του πελάτη αυτού. Εντούτοις όπως είδαμε τα κομμάτια n_1 είναι κατάλληλα και για την αγορά 2, αφού υπερκαλύπτουν τις απαιτήσεις ποιότητας της αγορά 2. Οπότε όταν έρχεται πελάτης της αγοράς 2 και έχουμε διαθέσιμα μόνο κομμάτια n_1 μπορούμε να πουλήσουμε από αυτά. Επομένως η βασική διαφορά με την πολιτική των Hong και Elsayed είναι ότι εμείς επιτρέπουμε την πώληση κομματιών ni στην αντίστοιχη αγορά , αλλά και στις λιγότερο απαιτητικές αγορές n_j, όπου $j>i$. Στο εξής θα ονομάζουμε την πολιτική των Hong και Elsayed πολιτική $n_1 \rightarrow \lambda_1$ ή $n_1\lambda_1$ (πωλούνται κομμάτια n_1 μόνο σε πελάτες λ_1), ενώ τη νέα πολιτική $n_1 \rightarrow \lambda_1, \lambda_2$ ή $n_1\lambda_1\lambda_2$ (πωλούνται κομμάτια n_1 σε πελάτες λ_1 είτε σε πελάτες λ_2)

Διαγραμματικά μπορούμε να περιγράψουμε τις δύο πολιτικές, όπως εφαρμόζονται στο σύστημα παραγωγής που εξετάζουμε στα παρακάτω σχήματα.





Σχ. 3. Πολιτική Hong και Elsayed $n_1\lambda_1$

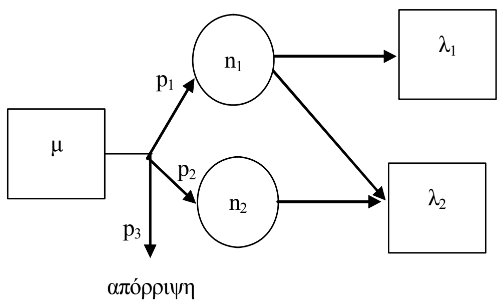
Σχ. 4. Νέα Πολιτική $n_1\lambda_1\lambda_2$

Τα λ_1 και λ_2 είναι οι ρυθμοί με τους οποίους καταφθάνουν οι πελάτες της αγοράς 1 και 2 αντίστοιχα, ενώ p_1, p_2 και p_3 είναι οι πιθανότητες τα κομμάτια που θα παραχθούν να έχουν τιμή του ποιοτικού χαρακτηριστικού τέτοια, ώστε να υπάγονται στην διαλογή 1, την διαλογή 2 ή να απορρίπτονται αντίστοιχα. Αναλυτικότερα θα φανούν αυτά στην μοντελοποίηση που αναφέρεται στο επόμενο κεφάλαιο.

3 Μοντελοποίηση του προβλήματος – Αλγόριθμος επίλυσης

3.1 Το σύστημα παραγωγής (παραγωγή – αποθέματα – ζήτηση)

Έχουμε το σύστημα παραγωγής που περιγράφηκε στο προηγούμενο κεφάλαιο. Για τη μοντελοποίηση, υποθέτουμε ότι η διάρκεια παραγωγής κομματιού είναι εκθετική με μέση τιμή μ (ρυθμός παραγωγής). Οι αφίξεις από αγορά i είναι Poisson με ρυθμό λ_i. Άρα ο ρυθμός με τον οποίο καταφθάνουν πελάτες που ανήκουν στην αγορά 1 είναι λ_1, ενώ οι πελάτες της αγοράς 2 έρχονται με ρυθμό λ_2. Για την αγορά 3 δεν αναμένουμε πελάτες. Αμέσως μόλις παραχθεί ένα προϊόν με $\delta_2 \leq |y-t| < \infty$ απορρίπτεται. Θα μπορούσαμε να πούμε ότι $\lambda_3=0$. Για λόγους διευκόλυνσης του αναγνώστη παραθέτουμε ξανά το σχ. 4 για τη νέα πολιτική.



Σχ. 5. Νέα Πολιτική($n_1\lambda_1\lambda_2$)

Όταν παράγεται ένα κομμάτι τότε αυξάνεται το n_1 ή το n_2, ή κανένα (λόγω απόρριψης), ανάλογα με την απόκλιση $|y-t|$. Γενικά ο ρυθμός αύξησης ni είναι μp_i όπου $p_i = P(\delta_{i-1} \leq |y-t| < \delta_i)$, όπου μ είναι ο ρυθμός παραγωγής κομματιών και p_i είναι η πιθανότητα το κομμάτι που παράγεται να ανήκει στη διαλογή i.

Τα κόστη που λαμβάνονται υπόψη στον υπολογισμό του συνολικού κέρδους είναι το κόστος επιθεώρησης κομματιού, το κόστος αποθέματος και τα κόστη ποιότητας κάθε αγοράς. Το κόστος ποιότητας είναι διαφορετικό για κάθε κομμάτι που πουλιέται, αφού για κάθε κομμάτι είναι διαφορετική η τιμή $|y-t|$.

Το μέσο κέρδος ανά κομμάτι από πωλήσεις στην αγορά i είναι η αξία πώλησης του κομματιού Ai μείον το μέσο κόστος ποιότητας των προϊόντων που πωλούνται στην αγορά i (θα υπολογιστεί παρακάτω). Μας ενδιαφέρει να συγκρίνουμε το μέσο συνολικό κέρδος $\mathbf{Jn_1\lambda_1\lambda_2}$ της νέας πολιτικής με το μέσο συνολικό κέρδος $\mathbf{Jn_1\lambda_1}$ της πολιτικής Hong και Elsayed.

3.2 Υπολογισμός του μέσου κέρδους

Το συνολικό κέρδος είναι συνάρτηση των κατωφλίων ποιότητας και της μέγιστης χωρητικότητας της αποθήκης S:

$J(\delta_1, \delta_2, S)$ = καθαρά κέρδη (πλην κόστους ποιότητας) από πωλήσεις

– κόστος αποθέματος – κόστος επιθεώρησης.

Επειδή ο ρυθμός παραγωγής μ είναι ο ίδιος και στις δύο πολιτικές, ο αριθμός των κομματιών που πρέπει να ελεγχθούν είναι ο ίδιος. Επομένως και το συνολικό κόστος επιθεώρησης είναι το ίδιο. Για αυτό δεν συμπεριλαμβάνουμε το κόστος επιθεώρησης στο εξής στον υπολογισμό του μέσου κέρδους. Άρα για την σύγκριση των δύο πολιτικών έχουμε:

$J(\delta_1, \delta_2, S)$ = καθαρά κέρδη (πλην κόστους ποιότητας) από πωλήσεις

– κόστος αποθέματος

Ορίζουμε

J_1: μέσο καθαρό κέρδος (έχοντας αφαιρέσει το κόστος ποιότητας) της αγοράς 1

J_2: μέσο καθαρό κέρδος (έχοντας αφαιρέσει το κόστος ποιότητας) της αγοράς 2

J_3: μέσο καθαρό κέρδος λόγω της απόρριψης κομματιών (συνήθως αρνητικός αριθμός), και

J_H: μέσο κόστος αποθέματος.

Τότε το μέσο κόστος J γράφεται ως:

$$J = J_1 + J_2 + J_3 - J_H \tag{1}$$

Οι συνιστώσες J_i του κόστους υπολογίζονται στη συνέχεια ξεχωριστά για κάθε πολιτική.

3.2.1 Υπολογισμός συνιστωσών J_i

Η τιμή y του ποιοτικού χαρακτηριστικού των παραγόμενων προϊόντων θεωρούμε ότι ακολουθεί κανονική κατανομή με μέση τιμή α και διασπορά σ. Έστω c η μέση απόκλιση του |y-t| για προϊόντα τέτοια ώστε |y–t|<δ όπου δ≥0.

Είναι

$$c = \int_{t-\delta}^{t+\delta} (y-t)^2 f(y)dy$$

Στο άρθρο Kouikoglou and Phillis (2002) έχει υπολογιστεί ο τύπος που υπολογίζει το παραπάνω ολοκλήρωμα. Ο σχετικός τύπος βρίσκεται στον κώδικα του προγράμματος που παρατίθενται στα παραρτήματα.

Ονομάζουμε λοιπόν

c_1: τη μέση απόκλιση απο την ιδανική τιμή t των κομματιών τέτοιων ώστε $|y–t|<\delta_1$

και

c_2: τη μέση απόκλιση απο την ιδανική τιμή t των κομματιών τέτοιων ώστε $|y–t|< \delta_2$

Είναι :

$$c_1 = \int_{t-\delta_1}^{t+\delta_1} (y-t)^2 f(y)dy \text{ και}$$

$$c_2 = \int_{t-\delta_2}^{t+\delta_2} (y-t)^2 f(y)dy$$

Έστω C_2 η μέση απόκλιση απο την ιδανική τιμή t των κομματιών τέτοιων ώστε $\delta_1 \leq |y–t|< \delta_2$. Είναι:

$$C_2 = \int_{t-\delta_2}^{t-\delta_1} (y-t)^2 f(y)dy + \int_{t+\delta_1}^{t+\delta_2} (y-t)^2 f(y)dy \quad (1)$$

Ισχύει η σχέση:

$$\int_{t-\delta_2}^{t} (y-t)^2 f(y)dy = \int_{t-\delta_2}^{t-\delta_1} (y-t)^2 f(y)dy + \int_{t-\delta_1}^{t} (y-t)^2 f(y)dy \Leftrightarrow$$

$$\int_{t-\delta_2}^{t-\delta_1}(y-t)^2 f(y)dy = \int_{t-\delta_2}^{t}(y-t)^2 f(y)dy - \int_{t-\delta_1}^{t}(y-t)^2 f(y)dy \quad (2)$$

Αντίστοιχα ισχύει επίσης:

$$\int_{t+\delta_1}^{t+\delta_2}(y-t)^2 f(y)dy = \int_{t}^{t+\delta_2}(y-t)^2 f(y)dy - \int_{t}^{t+\delta_1}(y-t)^2 f(y)dy \quad (3)$$

Επομένως από τις σχέσεις (1), (2) και (3) έχουμε:

$$C_2 = \int_{t-\delta_2}^{t}(y-t)^2 f(y)dy - \int_{t-\delta_1}^{t}(y-t)^2 f(y)dy + \int_{t}^{t+\delta_2}(y-t)^2 f(y)dy - \int_{t}^{t+\delta_1}(y-t)^2 f(y)dy \quad (4)$$

Αλλά

$$\int_{t-\delta_1}^{t}(y-t)^2 f(y)dy + \int_{t}^{t+\delta_1}(y-t)^2 f(y)dy = \int_{t-\delta_1}^{t+\delta_1}(y-t)^2 f(y)dy \quad (5)$$

και

$$\int_{t-\delta_2}^{t}(y-t)^2 f(y)dy + \int_{t}^{t+\delta_2}(y-t)^2 f(y)dy = \int_{t-\delta_2}^{t+\delta_2}(y-t)^2 f(y)dy \quad (6)$$

Η (4) συνεπάγεται από τι (5) και (6):

$$C_2 = \int_{t-\delta_2}^{t+\delta_2}(y-t)^2 f(y)dy - \int_{t-\delta_1}^{t+\delta_1}(y-t)^2 f(y)dy \Leftrightarrow$$

$$C_2 = c_2 - c_1$$

3.2.1.1 Συνιστώσες J_i για την πολιτική $n_1\lambda_1\lambda_2$

Ο ρυθμός με τον οποίο πωλούνται προϊόντα διαλογής 1 στην αγορά 1 είναι $\lambda_1 \times P(n_1>0)$. Για την πολιτική $n_1\lambda_1\lambda_2$ το μέσο κόστος ποιότητας Q_1, λόγω των πωλήσεων στην αγορά 1 είναι:

$Q_1 = \lambda_1 \, P(n_1>0) \, b_1 \, c_1$

όπου b_1 είναι ο συντελεστής κόστους ποιότητας της αγοράς 1 .

Ο ρυθμός με τον οποίο πωλούνται προϊόντα διαλογής 2 στην αγορά 2 είναι

$\lambda_2 \times P(n_2>0)$, ενώ ο ρυθμός με τον οποίο πωλούνται προϊόντα διαλογής 1 στην αγορά 2 είναι $\lambda_2 \times P(n_1>0,\ n_2 = 0)$. Το μέσο κόστος ποιότητας Q_2, λόγω των πωλήσεων στην αγορά 2 είναι:

Q_2 = (κόστος ποιότητας λόγω πώλησης προϊόντων διαλογής 2)

+ (κόστος ποιότητας λόγω πώλησης προϊόντων διαλογής 1)

$= \lambda_2\, b_2\, C_2\, P(n_2>0) + \lambda_2\, b_2\, c_1\, P(n_1>0,\, n_2=0)$

$= \lambda_2\, [b_2\, C_2\, P(n_2>0) + b_2\, c_1\, P(n_1>0,\, n_2=0)]$

Επομένως το μέσο κέρδος J_1 από τις πωλήσεις στην αγορά 1 είναι:

J_1 = (αξία πωλήσεων στην αγορά 1) $- Q_1$

$= \lambda_1\, P(n_1>0)\, A_1 - Q_1$

$= \lambda_1\, P(n_1>0)\, (A_1 - b_1\, c_1)$.

Το μέσο κέρδος J_2 από τις πωλήσεις στην αγορά 2 είναι:

J_2 = (αξία πωλήσεων προϊόντων διαλογής 1 και 2 στην αγορά 2) $- Q_2$

$= A_2\, [\lambda_2\, P(n_2>0) + \lambda_2\, (n_1>0,\, n_2 = 0)] - Q_2$

όπου $\lambda_2 P(n_2>0)$ είναι ο ρυθμός με τον οποίο πωλούνται προϊόντα διαλογής 2 στην αγορά 2 και $\lambda_2 P(n_1>0,\ n_2 = 0)$ είναι ο ρυθμός με τον οποίο πωλούνται προϊόντα διαλογής 1 στην αγορά 2. Αντικαθιστώντας στην παραπάνω σχέση το Q_2 έχουμε:

$$J_2 = \lambda_2 \left[A_2\, [P(n_2>0)+P(n_1>0,\, n_2 = 0)] - b_2\, C_2\, P(n_2>0) - b_2\, C_1\, P(n_1>0,\, n_2 = 0)\right]$$

Το μέσο κέρδος J_3 διάθεσης απορριπτέων κομματιών, το οποίο συνήθως είναι αρνητικό, είναι:

$J_3 = A_3 \mu p_3$

όπου το γινόμενο μp_3 είναι ο ρυθμός παραγωγής μ του συστήματος επί την πιθανότητα p_3 να παραχθεί κομμάτι απορριπτέο.

3.2.1.2 Συνιστώσες J_i για την πολιτική $n_1\lambda_1$

Σε αυτήν την περίπτωση το μέσο κόστος ποιότητας Q_1, λόγω των πωλήσεων στην αγορά 1 είναι το ίδιο με της πολιτικής $n_1\lambda_1\lambda_2$:

$Q_1 = Q_1 = \lambda_1 \, P(n_1>0) \, b_1 \, c_1$

ενώ το μέσο κόστος ποιότητας Q_2, λόγω των πωλήσεων στην αγορά 2 είναι:

$Q_2 = \lambda_2 \, P(n_2>0) \, b_2 \, C_2$ όπου

$\lambda_2 \, P(n_2>0)$ είναι ο ρυθμός πώλησης προϊόντων διαλογής 2 στην αγορά 2.

Το μέσο κέρδος J_1 από τις πωλήσεις στην αγορά 1 είναι:

$J_1 = \lambda_1 \, P(n_1>0) \, A_1 - Q_1 \Leftrightarrow J_1 = \lambda_1 \, P(n_1>0) \, (A_1 - b_1 \, c_1)$.

και το μέσο κέρδος J_2 από τις πωλήσεις στην αγορά 2 είναι:

J_2 = (αξία πωλήσεων προϊόντων διαλογής 2 στην αγορά 2) $- \, Q_2 \Leftrightarrow$

$J_2 = \lambda_2 \, P(n_2>0) \, A_2 - Q_2 \Leftrightarrow$

$J_2 = \lambda_2 \, P(n_2>0) \, (A_2 - b_2 \, C_2)$.

Το μέσο κέρδος J_3 παραμένει όπως ήταν:

$J_3 = A_3 \mu p_3$

3.2.2 Μέσο κόστος αποθέματος

Το μέσο κόστος αποθέματος J_H είναι το γινόμενο του κόστους h της αποθήκευσης ενός κομματιού για μια μονάδα χρόνου, επί το μέσο απόθεμα Η το οποίο υπολογίζεται παρακάτω.

3.3 Υπολογισμός πιθανοτήτων των καταστάσεων των αποθεμάτων

Για τον υπολογισμό των πιθανοτήτων $P(n_1>0)$ και $P(n_2=0, \, n_1>0)$ θα χρησιμοποιήσουμε μοντέλο Markov. Υποθέτουμε ότι οι χρόνοι παραγωγής κομματιών και οι χρόνοι ανάμεσα σε διαδοχικές αφίξεις πελατών ακολουθούν εκθετικές κατανομές.

Όλες οι πιθανές καταστάσεις (n_1, n_2) οι οποίες μπορούν να παρατηρηθούν στην αποθήκη του συστήματός μας φαίνονται στην παρακάτω αλυσίδα Markov (σχ. 6). Κάθε κατάσταση (n_1, n_2) εμφανίζεται με πιθανότητα $P(n_1, n_2)$

Η πιθανότητα $P(n_1>0)$ είναι το άθροισμα των πιθανοτήτων όλων των καταστάσεων, εκτός από τις καταστάσεις που στην αλυσίδα βρίσκονται στην κάτω γραμμή, όπου $n_1=0$. Είναι:

$$p(n_1 > 0) = \sum p(n_1, n_2) - \sum p(0, n_2)\ , \quad \text{όπου } s= n_1+n_2=1, 2, \dots, S$$

Η πιθανότητα $P(n_2=0,\ n_1>0)$ είναι το άθροισμα των πιθανοτήτων των καταστάσεων της κάτω γραμμής της αλυσίδας όπου $n_1=0$, εκτός από την κατάσταση 0, 0 όπου ισχύει και $n_1=0$.

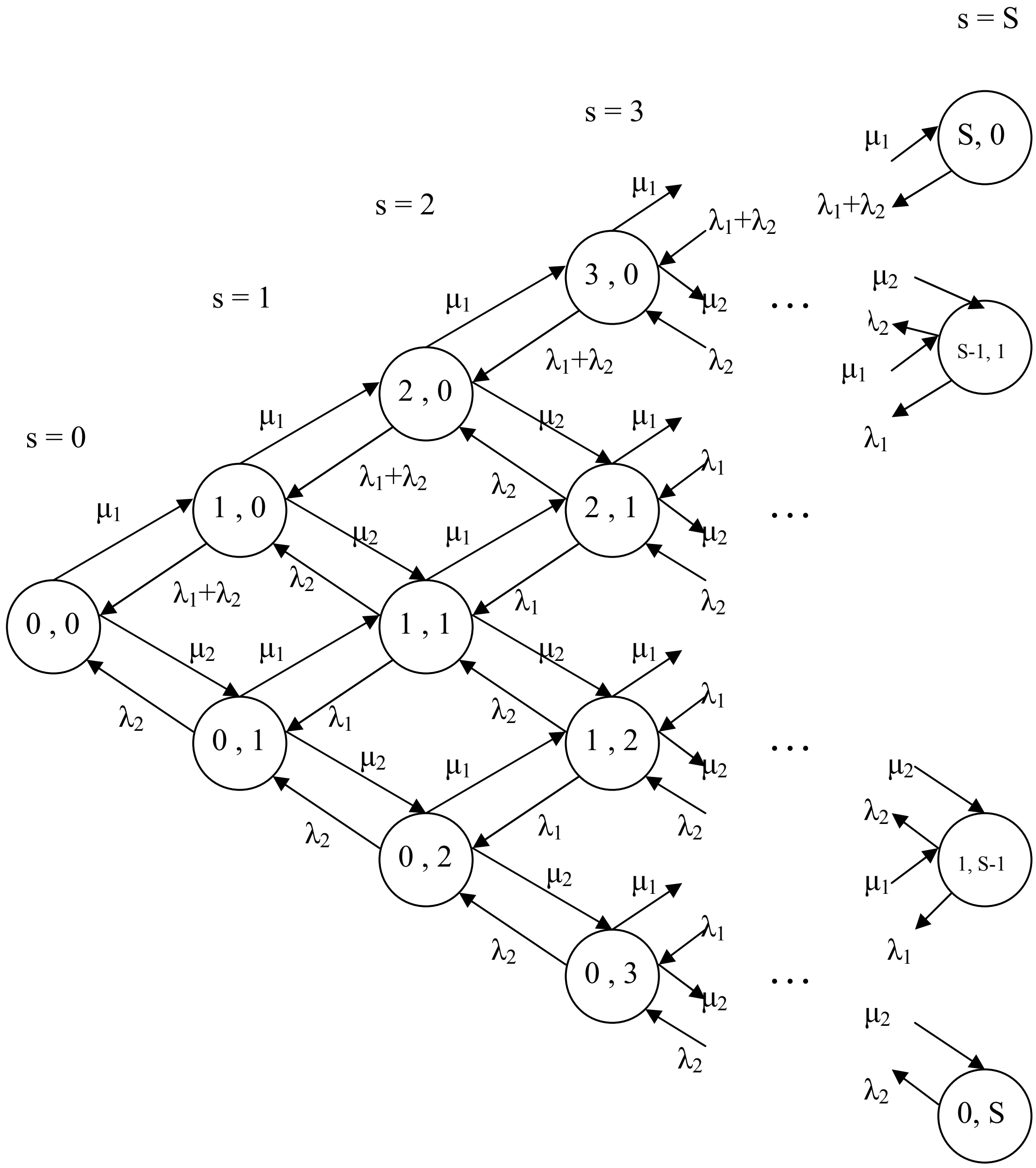
$$P(n_2=0, n_1>0) = [\sum P(0,n_2)] - P(0,0)\ , \quad \text{όπου } s= n_1+n_2=0, 1, ..., S$$

Το μέσο απόθεμα Η ισούται με:

$$H = \sum_{n_1}\sum_{n_2}(n_1 + n_2)P(n_1, n_2)$$

3.3.1 Αλυσίδα Markov για την πολιτική $n_1\lambda_1\lambda_2$

Το σύστημα αποθηκεύει προϊόντα συνεχώς, μέχρις ότου το απόθεμα γίνει S. Κατόπιν παύει να παράγει μέχρις ότου το απόθεμα μειωθεί. Οι επιτρεπτές καταστάσεις είναι οι $(n_1,\ n_2)$ όπου $n_1\geq 0$, $n_2\geq 0$ και $n_1+n_2\leq S$. Ορίζουμε τη βοηθητική μεταβλητή καταστάσεως $s=n_1+n_2$. Επίσης ορίζουμε ως ρυθμό παραγωγής μ_1 των κομματιών n_1, το γινόμενο μp_1 και ως ρυθμό παραγωγής μ_2 των κομματιών n_2, το γινόμενο μp_2.



Σχ. 6. Αλυσίδα Markov για την πολιτική $n_1\lambda_1\lambda_2$

Η χωρητικότητα S της αποθήκης είναι παράμετρος του προβλήματος. Συνεπώς για κάθε διαφορετικό S έχουμε διαφορετικό μέγεθος αλυσίδας. Η αλυσίδα Markov έχει πεπερασμένο αριθμό καταστάσεων, συνεπώς υπάρχει μόνιμη κατάσταση. Στην μόνιμη κατάσταση υπάρχει μία ισορροπία μεταξύ των πιθανοτήτων η οποία εκφράζεται από τις εξισώσεις Chapman-Kolmogorov:

P(το σύστημα να ευρίσκεται στην κατάσταση σ) × (μέσος ρυθμός εξόδου από αυτήν)

= $\sum$ P(να ευρίσκεται σε άλλη κατάσταση) × (μέσος ρυθμός μετάβασης στην σ)

Η συνθήκη ισορροπίας στην κατάσταση (0, 0), όπου $n_1+n_2 = 0$, δίνει τη σχέση:

$$[\mu_1+\mu_2]P(0,0)=[\lambda_1+\lambda_2\ \lambda_2]\cdot\begin{bmatrix}P(1,0)\\P(0,1)\end{bmatrix}$$

Για τις καταστάσεις όπου $n_1+n_2=1$ έχουμε τη σχέση:

$$\begin{bmatrix}\lambda_1+\lambda_2+\mu_1+\mu_2 & 0\\ 0 & \lambda_2+\mu_1+\mu_2\end{bmatrix}\begin{bmatrix}P(1,0)\\P(0,1)\end{bmatrix}=\begin{bmatrix}\mu_1\\ \mu_2\end{bmatrix}P(0,0)+\begin{bmatrix}\lambda_1+\lambda_2 & \lambda_2 & 0\\ 0 & \lambda_1 & \lambda_2\end{bmatrix}\cdot\begin{bmatrix}P(2,0)\\P(1,1)\\P(0,2)\end{bmatrix}$$

Για τις καταστάσεις όπου $n_1+n_2=2$ προκύπτουν οι

$$\begin{bmatrix}\lambda_1+\lambda_2+\mu_1+\mu_2 & 0 & 0\\ 0 & \lambda_1+\lambda_2+\mu_1+\mu_2 & 0\\ 0 & 0 & \lambda_2+\mu_1+\mu_2\end{bmatrix}\begin{bmatrix}P(2,0)\\P(1,1)\\P(0,2\end{bmatrix}$$

$$=\begin{bmatrix}\mu_1 & 0\\ \mu_2 & \mu_1\\ 0 & \mu_2\end{bmatrix}\cdot\begin{bmatrix}P(1,0)\\P(0,1)\end{bmatrix}+\begin{bmatrix}\lambda_1+\lambda_2 & \lambda_2 & 0 & 0\\ 0 & \lambda_1 & \lambda_2 & 0\\ 0 & 0 & \lambda_1 & \lambda_2\end{bmatrix}\cdot\begin{bmatrix}P(3,0)\\P(2,1)\\P(1,2)\\P(0,3)\end{bmatrix}$$

Όμοια, στην κατάσταση όπου $n_1+n_2=3$ έχουμε

$$\begin{bmatrix}\lambda_1+\lambda_2+\mu_1+\mu_2 & 0 & 0 & 0\\ 0 & \lambda_1+\lambda_2+\mu_1+\mu_2 & 0 & 0\\ 0 & 0 & \lambda_1+\lambda_2+\mu_1+\mu_2 & 0\\ 0 & 0 & 0 & \lambda_2+\mu_1+\mu_2\end{bmatrix}\begin{bmatrix}P(3,0)\\P(2,1)\\P(1,2)\\P(0,3)\end{bmatrix}$$

$$=\begin{bmatrix}\mu_1 & 0 & 0\\ \mu_2 & \mu_1 & 0\\ 0 & \mu_2 & \mu_1\\ 0 & 0 & \mu_2\end{bmatrix}\cdot\begin{bmatrix}P(2,0\\P(1,1)\\P(0,2\end{bmatrix}+\begin{bmatrix}\lambda_1+\lambda_2 & \lambda_2 & 0 & 0 & 0\\ 0 & \lambda_1 & \lambda_2 & 0 & 0\\ 0 & 0 & \lambda_1 & \lambda_2 & 0\\ 0 & 0 & 0 & \lambda_1 & \lambda_2\end{bmatrix}\cdot\begin{bmatrix}P(4,0)\\P(3,1)\\P(2,2)\\P(1,3)\\P(0,4)\end{bmatrix}$$

Γενικά η σχέση που περιγράφει την ισορροπία για τις καταστάσεις $P(n_1, n_2)$ όταν το άθροισμα n_1+n_2 είναι s έχει τη μορφή:

$$A_s\,P_s = B_s\,P_{s-1} + C_{s+1}\,P_{s+1}$$

Ο πίνακας A_s πολλαπλασιάζει το διάνυσμα P_s που περιέχει όλες τις πιθανότητες για $n_1+n_2=s$. Είναι διαγώνιος με γενική μορφή:

$$A_n = \begin{bmatrix} \lambda_1+\lambda_2+\mu_1+\mu_2 & 0 & \dots & 0 \\ 0 & \lambda_1+\lambda_2+\mu_1+\mu_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & \lambda_1+\lambda_2+\mu_1+\mu_2 & 0 \\ 0 & \dots & 0 & \lambda_2+\mu_1+\mu_2 \end{bmatrix}$$

Ο πίνακας B_s πολλαπλασιάζει το διάνυσμα P_{s-1} που περιέχει όλες τις πιθανότητες για $n_1+n_2=s-1$. Έχει διαστάσεις n, n–1 με γενική μορφή:

$$B_n = \begin{bmatrix} \mu_1 & 0 & \dots & 0 & 0 \\ \mu_2 & \mu_1 & & 0 & 0 \\ 0 & \mu_2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \mu_1 & 0 \\ 0 & & \dots & \mu_2 & \mu_1 \\ 0 & & \dots & 0 & \mu_2 \end{bmatrix}$$

Ενώ ο πίνακας C_{s+1} πολλαπλασιάζει το διάνυσμα P_{s+1} που περιέχει τις πιθανότητες για $n_1+n_2=s+1$ και έχει διαστάσεις n, n+1.

$$C_n = \begin{bmatrix} \lambda_1+\lambda_2 & \lambda_2 & 0 & \dots & 0 \\ 0 & \lambda_1 & \lambda_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \lambda_1 & \lambda_2 & 0 \\ 0 & \cdots & 0 & \lambda_1 & \lambda_2 \end{bmatrix}$$

Θεωρούμε έναν νέο πίνακα G, ο οποίος ορίζεται παρακάτω.

$$\left.\begin{aligned} I\cdot P_n &= A_n^{-1}B_nP_{n-1}+A_n^{-1}C_{n+1}P_{n+1} \\ P_{n-1} &= G_nP_n \end{aligned}\right\} \Rightarrow I\cdot P_n = A_n^{-1}B_nG_nP_n + A_n^{-1}C_{n+1}P_{n+1} \Rightarrow$$

$$\Rightarrow \left(I - A_n^{-1}B_nG_n\right)P_n = A_n^{-1}C_{n+1}P_{n+1} \Rightarrow P_n = \left(I - A_n^{-1}B_nG_n\right)^{-1}A_n^{-1}C_{n+1}P_{n+1}$$

$$G_{n+1} = \left(I - A_n^{-1}B_nG_n\right)^{-1}A_n^{-1}C_{n+1}$$

Με τον πίνακα G_n μπορούμε εύκολα να συσχετίσουμε το διάνυσμα πιθανοτήτων P_{n-1} με το επόμενό του P_n

Ορίζουμε τώρα και τους πίνακες F_n, n=1, 2, ..., s–1 Οι πίνακες Fn συσχετίζουν οποιοδήποτε διάνυσμα P_n, n=0, 1, 2, …s–1 με το τελευταίο διάνυσμα πιθανοτήτων P_S όπου $n_1+n_2=S$.

$$
\begin{aligned}
&P_0 = G_0P_1 = ... = G_0G_1...G_{s-1}P_s = F_0P_s \\
&P_1 = G_1P_2 = ... = G_1G_2...G_{s-1}P_s = F_1P_s \\
&\vdots \\
&P_{s-1} = G_{s-1}P_s = F_{s-1}P_s
\end{aligned}
$$

Από συνοριακές συνθήκες έχουμε:

$$IP_s = A_s^{-1}B_sP_{s-1}$$

όπου $A_s^{-1}B_s$ ένας πίνακας με διαστάσεις s, s–1 που είναι ίσος με

$$A_s^{-1}B_s = \begin{bmatrix} \frac{\mu_1}{\lambda_1+\lambda_2} & 0 & \cdots & 0 \\ \frac{\mu_2}{\lambda_1+\lambda_2} & \frac{\mu_1}{\lambda_1+\lambda_2} & 0 & \vdots \\ 0 & \ddots & \ddots & 0 \\ \vdots & \cdots 0 & \frac{\mu_2}{\lambda_1+\lambda_2} & \frac{\mu_1}{\lambda_1+\lambda_2} \\ 0 & \cdots & 0 & \frac{\mu_2}{\lambda_2} \end{bmatrix}$$

και από την προηγούμενη εξίσωση έχουμε

$$IP_s = A_s^{-1}B_sG_{s-1}P_s \Rightarrow \left(I - A_s^{-1}B_sG_{s-1}\right)P_s = 0$$

Αυτή η εξίσωση πινάκων είναι ένα σύστημα εξισώσεων με S+1 αγνώστους και S+1 εξισώσεις. Μία όμως από τις εξισώσεις είναι εξαρτημένη από τις υπόλοιπες, και άρα είναι περιττή. Αντί αυτής χρησιμοποιούμε την

$$\sum p(i,j) = 1$$

ή αλλιώς $\sum$ $[\text{των στοιχείων των διανυσμάτων } P_n] = 1 \Leftrightarrow$

$$\left.\begin{aligned} P_0 + I_1P_1 + ... + I_sP_s = 1 \\ P_n = F_nP_s \end{aligned}\right\} \Leftrightarrow$$

όπου I_s=[1 1 1 1 . . . 1] με διαστάσεις 1, s+1.

$$\Leftrightarrow F_0P_s + F_1P_s + ... + F_{s-1}P_s + P_s = 1 \Leftrightarrow \left[F_0 + F_1 + ... + F_{s-1} + I\right]\begin{bmatrix} P(s,0) \\ P(s-1,1) \\ \vdots \\ P(0,s) \end{bmatrix} = 1$$

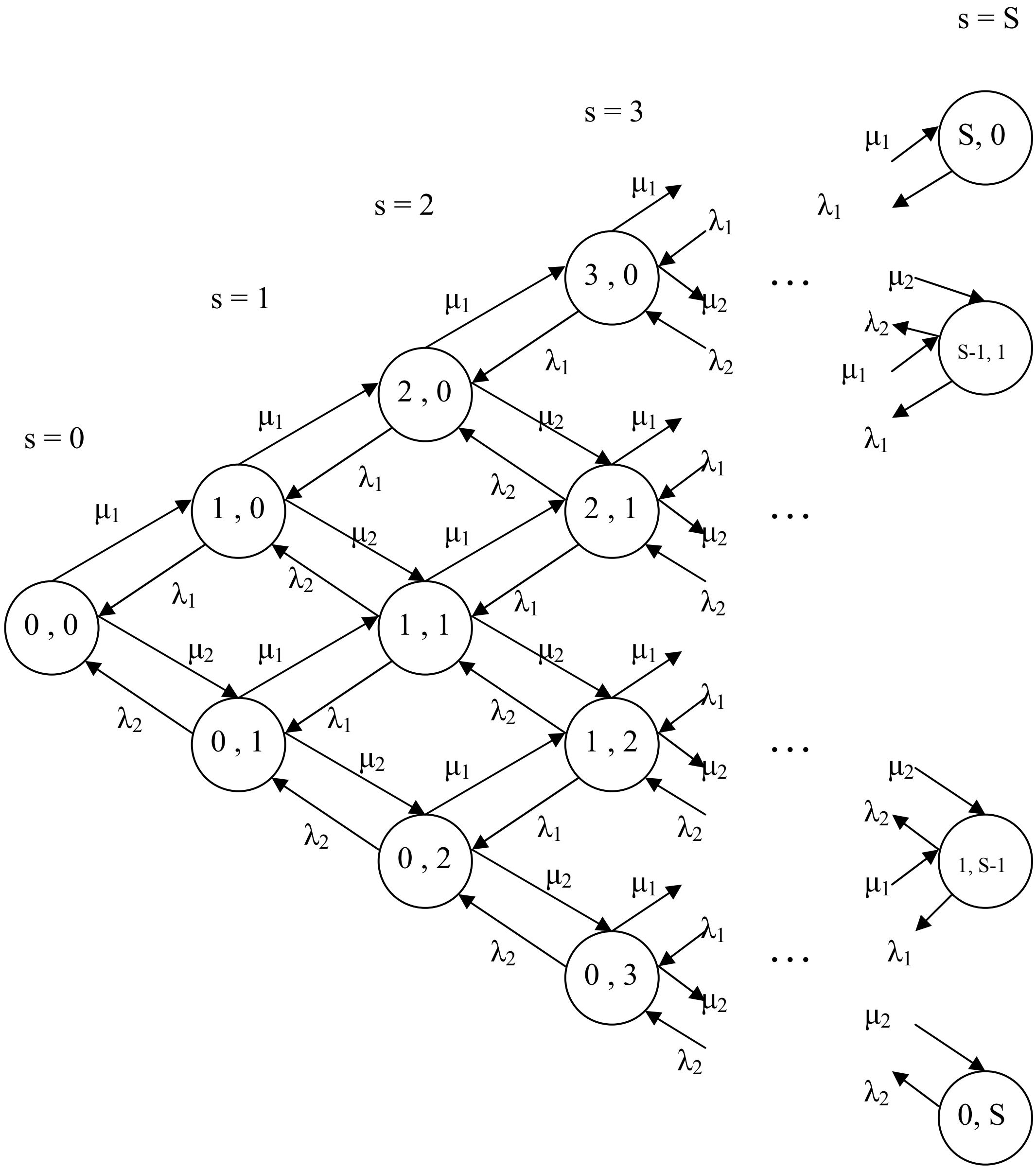
Οπότε λύνουμε το σύστημα

$$
\left[\begin{array}{c}
\mathrm{I}-A_s^{-1}B_sG_{s-1} \\
\text{εκτός τελευταίας γραμμής} \\
\hline
F_0+F_1+.....+F_{s-1}++I
\end{array}\right]
\begin{bmatrix} P(s,0) \\ P(s-1,1) \\ \vdots \\ P(0,s) \end{bmatrix}
=
\begin{bmatrix} 0 \\ \vdots \\ \underline{0} \\ 1 \end{bmatrix}
$$

Γνωρίζοντας πλέον το διάνυσμα πιθανοτήτων P_s υπολογίζουμε και τις πιθανότητες των υπόλοιπων διανυσμάτων P_n από τις σχέσεις $P_n = F_n P_s$. Οπότε υπολογίζονται όλες οι πιθανότητες p(i,j) που είναι απαραίτητες για τον υπολογισμό του μέσου κέρδους.

3.3.2 Αλυσίδα Markov για την πολιτική $n_1\lambda_1$

Η αλυσίδα Markov που περιγράφει τις καταστάσεις του αποθέματος και τις δυνατές μεταβάσεις του συστήματος από μια κατάσταση σε άλλη, για την πολιτική **$n_1\lambda_1$** φαίνεται στο παρακάτω σχήμα. Η μόνη διαφορά απο την αλυσίδα Markov για την πολιτική **$n_1\lambda_1\lambda_2$** είναι ότι απο τις πιθανότητες $p(s+1,0)$ προς τις πιθανότητες $p(s,0)$ η μετάβαση γίνεται με ρυθμό λ_1, αντί για $\lambda_1+\lambda_2$.



Για την αλυσίδα Markov της πολιτικής **$n_1\lambda_1$** οι πίνακες είναι οι παρακάτω:

$$A_n = \begin{bmatrix} \lambda_1+\mu_1+\mu_2 & 0 & \cdots & 0 \\ 0 & \lambda_1+\lambda_2+\mu_1+\mu_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & \lambda_1+\lambda_2+\mu_1+\mu_2 & 0 \\ 0 & \cdots & 0 & \lambda_2+\mu_1+\mu_2 \end{bmatrix},$$

$$B_n = \begin{bmatrix} \mu_1 & 0 & \dots & 0 & 0 \\ \mu_2 & \mu_1 & & 0 & 0 \\ 0 & \mu_2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \mu_1 & 0 \\ 0 & & \dots & \mu_2 & \mu_1 \\ 0 & & \dots & 0 & \mu_2 \end{bmatrix}$$

$$\text{και} \quad C_n = \begin{bmatrix} \lambda_1 & \lambda_2 & 0 & \dots & 0 \\ 0 & \lambda_1 & \lambda_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \lambda_1 & \lambda_2 & 0 \\ 0 & \cdots & 0 & \lambda_1 & \lambda_2 \end{bmatrix}$$

όλη η υπόλοιπη διαδικασία επίλυσης του συστήματος προκειμένου να υπολογιστούν οι πιθανότητες $p(i,j)$ είναι ακριβώς η ίδια.

3.4 Αλγόριθμοι υπολογισμού μέσου κέρδους

Τα δεδομένα που δίδονται σαν είσοδος στο πρόβλημα καταγράφονται στον παρακάτω πίνακα:

μ	Ρυθμός παραγωγής κομματιών
λ_1	Ρυθμός πώλησης στην αγορά 1
λ_2	Ρυθμός πώλησης στην αγορά 2
h	Κόστος αποθέματος ενός κομματιού σε μια μονάδα χρόνου
b_1	Συντελεστής κόστους ποιότητας αγοράς 1
b_2	Συντελεστής κόστους ποιότητας αγοράς 1
A_1	Τιμή πώλησης κομματιού στην αγορά 1
A_2	Τιμή πώλησης κομματιού στην αγορά 2
A_3	Τιμή πώλησης κομματιού στην αγορά 3
t	Ιδανική τιμή εξεταζόμενου ποιοτικού χαρακτηριστικού
α	μέση τιμή του εξεταζόμενου ποιοτικού χαρακτηριστικού
σ	διασπορά του εξεταζόμενου ποιοτικού χαρακτηριστικού

Τα στοιχεία που υπολογίζονται απο τον αλγόριθμο και αποτελούν την έξοδο του προβλήματος είναι τα κατώφλια ποιότητας δ_1 , δ_2 και η βέλτιστη χωρητικότητα S της αποθήκης.

3.4.1 Υπολογισμός μέσου κέρδους της πολιτικής $n_1\lambda_1$

Για τον υπολογισμό του βέλτιστου μέσου κέρδους $J_{opt}EL$ της πολιτικής Hong και Elsayed ακολουθεί τα παρακάτω βήματα.

Βήμα 1: Υπολογίζουμε με βάση τη δημοσίευση των Hong και Elsayed τα βέλτιστα κατώφλια δ_1 και δ_2, τα οποία αντιστοιχούν στην περίπτωση κατά την οποία αμέσως μόλις παράγεται ένα κομμάτι απορροφάται από την αντίστοιχη αγορά. Υπενθυμίζουμε ότι αυτή η θεώρηση αγνοεί το κόστος αποθέματος.

Βήμα 2: Θέτουμε $J_{opt}EL = -\infty$ και ορίζουμε τη μέγιστη τιμή S_{max} για το απόθεμα ασφαλείας S. Θα αναζητήσουμε το βέλτιστο S από το σύνολο $\{0, 1, 2, \ldots, S_{max}\}$, όπου S_{max} αρκούντως μεγάλος αριθμός

Βήμα 3: Για κάθε $S=0, 1, 2, \ldots, S_{max}$:

α. Υπολογίζουμε το μέσο κέρδος JEL_S

β. Αν $JEL_S > J_{opt}EL$, τότε ενημερώνουμε το νέο βέλτιστο κέρδος και το βέλτιστο απόθεμα ασφαλείας θέτοντας $J_{opt}EL = JEL_S$ και $S_{opt} = S$.

Ο κώδικας με τον οποίο υλοποιήθηκε ο αλγόριθμος είναι γραμμένος σε γλώσσα προγραμματισμού Fortran77 και βρίσκεται στο Παράρτημα 1.

3.4.2 Υπολογισμός μέσου κέρδους της πολιτικής $n_1\lambda_1\lambda_2$

Αναζητούμε τώρα τα δ_1, δ_2 και S που βελτιστοποιούν το κέρδος από κοινού. Ο αλγόριθμος διαμορφώνεται ως εξής:

Βήμα 1: Θέτουμε $J_{opt} = -\infty$ και ορίζουμε τη μέγιστη τιμή S_{max} για το απόθεμα ασφαλείας S. Καθορίζουμε μία μέγιστη τιμή Δ για το δ_2 και κάνουμε μία διαμέριση του τμήματος [0, Δ] σε n ισαπέχοντα σημεία:

$$\delta_2 = 1\frac{\Delta}{n},\quad 2\frac{\Delta}{n},\ \ldots,\ (n-1)\frac{\Delta}{n},\quad n\frac{\Delta}{n} = \Delta$$

Για κάθε τιμή του δ_2 δοκιμάζουμε διάφορες τιμές για τα όρια δ_1 της πρώτης διαλογής, τέτοιες ώστε $\delta_1 \leq \delta_2$. Έχουμε δηλαδή

$$\delta_1 = 0\frac{\Delta}{n},\quad 1\frac{\Delta}{n},\ \ldots,\ \delta_2$$

Βήμα 2: Ξεκινάμε με $\delta_2 = \Delta$ και $\delta_1 = \Delta$.

Βήμα 3: Για κάθε $S = 0, 1, 2, \ldots, S_{max}$:

α. Υπολογίζουμε το μέσο κέρδος $J(\delta_1, \delta_2, S)$.

β. Αν $J(\delta_1, \delta_2, S) > J_{opt}$, τότε ενημερώνουμε το νέο βέλτιστο κέρδος και τις βέλτιστες παραμέτρους θέτοντας $J_{opt} = J(\delta_1, \delta_2, S)$, και $S_{opt} = S$, $\delta_{1opt} = \delta_1$, $\delta_{2opt} = \delta_2$.

Βήμα 4: Μειώνουμε το δ_1 κατά Δ/n. Αν το αποτέλεσμα είναι ≥ 0, τότε έχουμε μία νέα αποδεκτή τιμή για το δ_1 και πηγαίνουμε στο Βήμα 3. Αν το νέο $\delta_1<0$, τότε έχουμε εξαντλήσει όλες τις τιμές του δ_1, οπότε θέτουμε $\delta_2 = \delta_2 - \Delta/n$. Αν η νέα τιμή του δ_2 είναι $\geq \Delta/n$, που είναι η μικρότερη τιμή για το δ_2, τότε θέτουμε $\delta_1 = \delta_2$ και πηγαίνουμε στο Βήμα 3. Αλλιώς έχουμε $\delta_2 = 0$, οπότε έχουμε εξαντλήσει όλες τις αποδεκτές τιμές του δ_2 και περατώνουμε τον αλγόριθμο.

Ο κώδικας με τον οποίο υλοποιήθηκε ο αλγόριθμος είναι γραμμένος σε γλώσσα προγραμματισμού Fortran77 και βρίσκεται στο Παράρτημα 2.

Δε μένει πια παρά να συγκρίνουμε το μέγιστο μέσο κέρδος των δύο πολιτικών για να διαπιστώσουμε αν η πολιτική $n_1\lambda_1\lambda_2$ αποδίδει καλύτερα από την $n_1\lambda_1$. Αυτό όμως θα γίνει στο επόμενο κεφάλαιο.

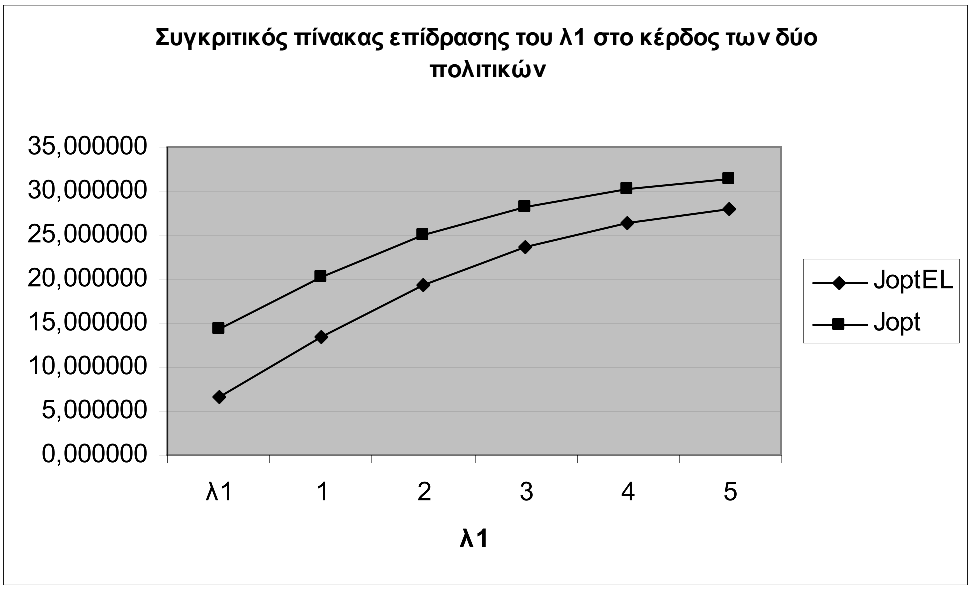
4 Πειραματική σύγκριση του κέρδους των δύο πολιτικών

Στο προηγούμενο κεφάλαιο είδαμε ποιες είναι οι παράμετροι εισόδου και ποια η έξοδος του συστήματος. Σε αυτό το κεφάλαιο δίνουμε τιμές στις παραμέτρους εισόδου, μεταβάλλοντας βηματικά κάθε μια από αυτές, ώστε να δούμε πώς επηρεάζεται η έξοδος του συστήματος. Από τα τρία δεδομένα εξόδου περισσότερο μας ενδιαφέρει να συγκρίνουμε το μέσο κέρδος $J_{opt}EL$ της πολιτικής $n_1\lambda_1$ με το μέσο κέρδος J_{opt} της πολιτική $n_1\lambda_1\lambda_2$.

Για κάθε μια από τις παραμέτρους εισόδου παρουσιάζεται το διάγραμμα της μεταβολής του μέσου κέρδους των δύο πολιτικών σε συνάρτηση με τη μεταβολή της παραμέτρου αυτής.

4.1 Επίδραση της μεταβολής του λ_1

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές, και μεταβάλλοντας μόνο το ρυθμό άφιξης λ_1 πελατών αγοράς 1, παίρνουμε το Σχήμα 1.

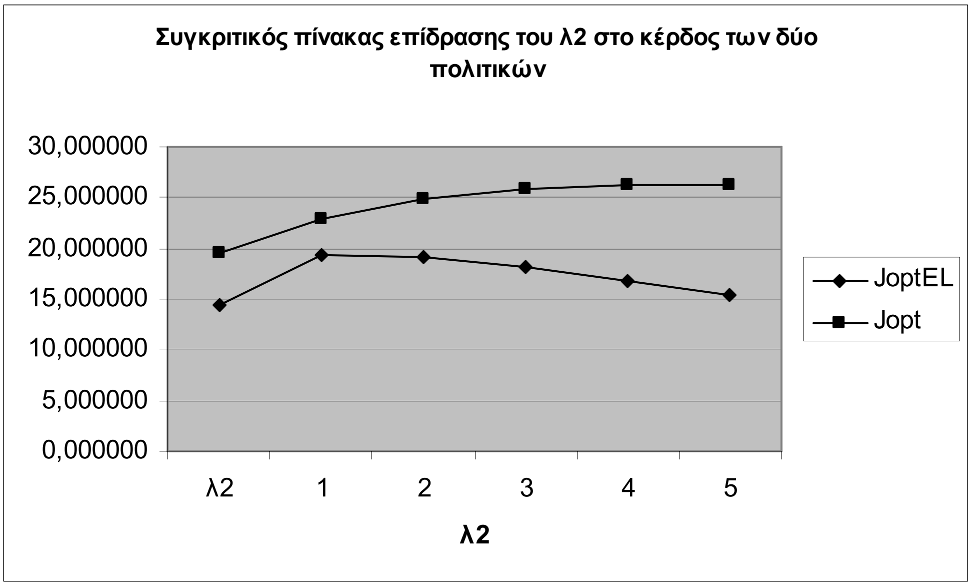


Σχήμα 1 Συγκριτικός πίνακας επίδρασης του λ_1 στο κέρδος των δύο πολιτικών

Το μέσο κέρδος τη πολιτικής $n_1\lambda_1\lambda_2$ υπερκαλύπτει γραφικά συνεχώς το μέσο κέρδος της πολιτικής $n_1\lambda_{1.}$

4.2 Επίδραση της μεταβολής του λ_2

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το ρυθμό άφιξης λ_2 πελατών αγοράς 2, παίρνουμε το Σχήμα 2.

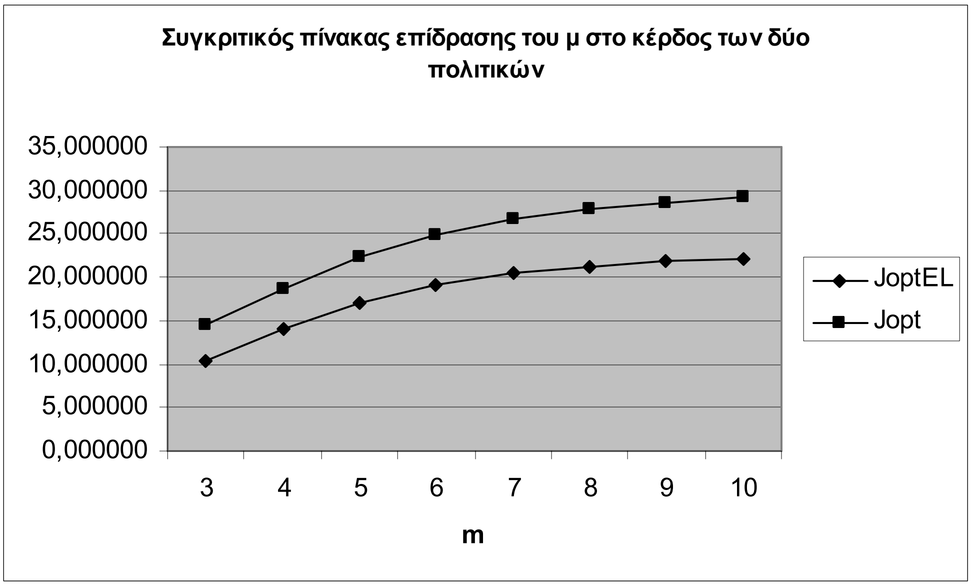


Σχήμα 2 Συγκριτικός πίνακας επίδρασης του λ_2 στο κέρδος των δύο πολιτικών

Το κέρδος της πολιτική $n_1\lambda_1\lambda_2$ είναι μεγαλύτερο της πολιτικής $n_1\lambda_1$. Όσο το λ_2 μεγαλώνει, τόσο αυξάνει και η διαφορά του κέρδους. Αυτό συμβαίνει γιατί όταν το λ2 γίνεται μεγαλύτερο από το λ1, στην πολιτική $n_1\lambda_1$ «μένουν» συχνά απούλητα αποθέματα, ενώ στην πολιτική $n_1\lambda_1\lambda_2$ εκμεταλλευόμαστε το αυξημένο ρυθμό ζήτησης της αγοράς 2, ακόμα κι αν δεν έχουμε κομμάτια διαλογής 1 να προσφέρουμε.

4.3 Επίδραση της μεταβολής του μ

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το ρυθμό παραγωγής μ του συστήματος, παίρνουμε το Σχήμα 3.

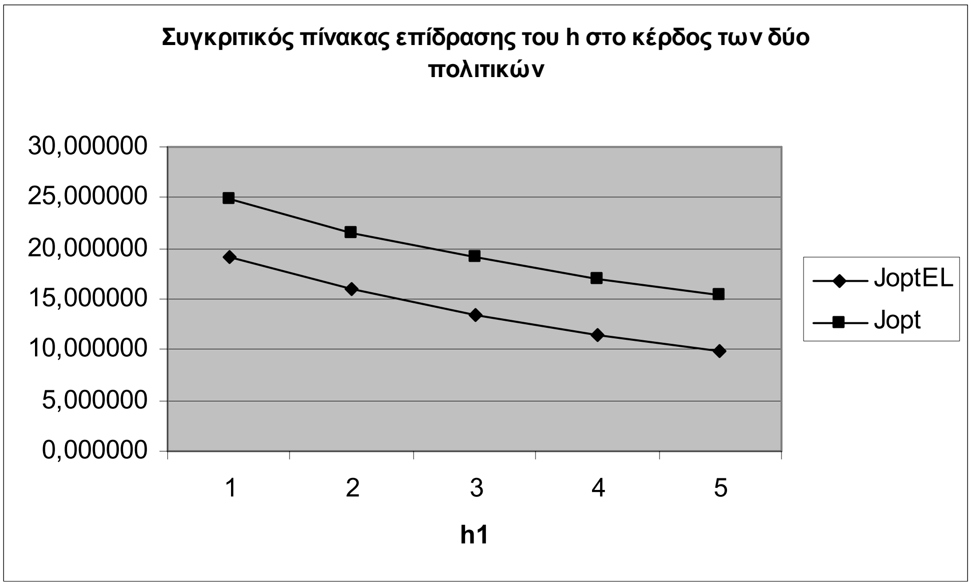


Σχήμα 3 Συγκριτικός πίνακας επίδρασης του μ στο κέρδος των δύο πολιτικών

Παρατηρούμε ότι καθώς αυξάνεται ο ρυθμός παραγωγής αυξάνεται ασυμπτωτικά και το κέρδος, ενώ το κέρδος της πολιτικής $n_1\lambda_1\lambda_2$ παραμένει πάντα υψηλότερο από της πολιτικής $n_1\lambda_1$.

4.4 Επίδραση της μεταβολής του h

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το κόστος αποθέματος h ενός κομματιού ανά μονάδα χρόνου, παίρνουμε το Σχήμα 4.

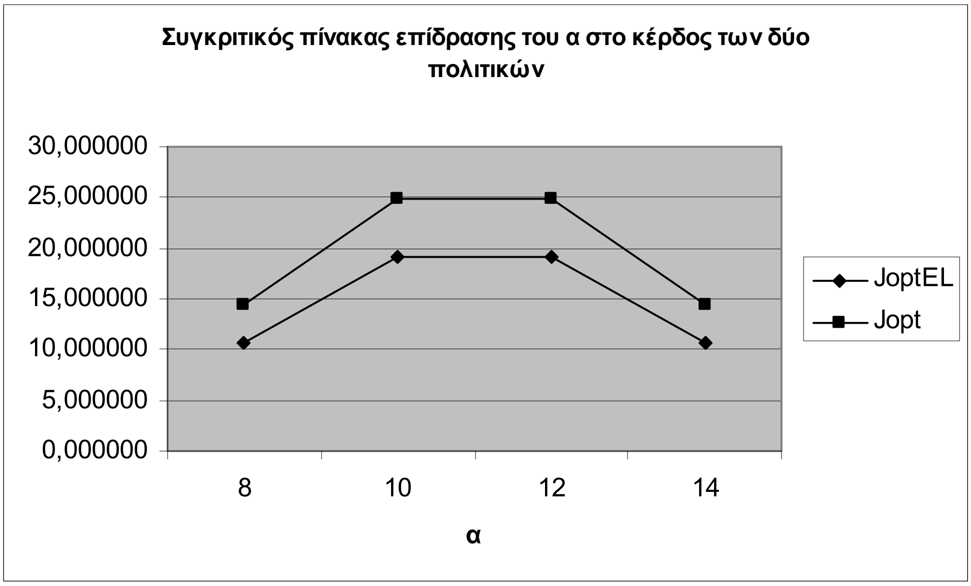


Σχήμα 4 Συγκριτικός πίνακας επίδρασης του h στο κέρδος των δύο πολιτικών

Όπως είναι φυσικό, όσο αυξάνεται το κόστος αποθέματος κομματιού, τόσο μειώνονται τα κέρδη. Το κέρδος J_{opt} είναι πάντα μεγαλύτερο του J_{opt}EL.

4.5 Επίδραση της μεταβολής του α

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο τη μέση τιμή α της κανονικής κατανομής του χαρακτηριστικού που εξετάζεται, παίρνουμε το Σχήμα 5.

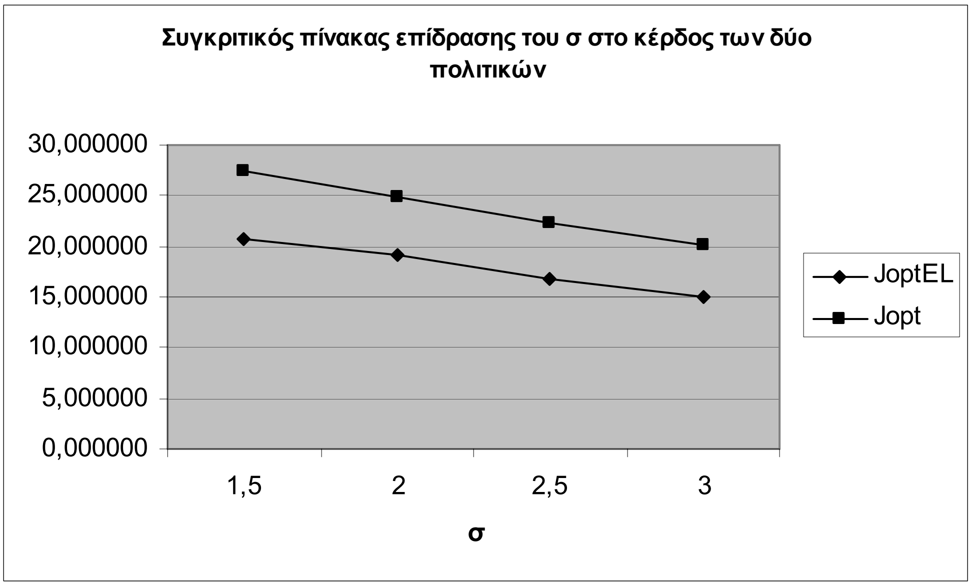


Σχήμα 5Συγκριτικός πίνακας επίδρασης του α στο κέρδος των δύο πολιτικών

Το κέρδος γίνεται μέγιστο, όταν η μέση τιμή α γίνεται ίση με την ιδανική τιμή t. Το κέρδος J_{opt} είναι μεγαλύτερο του J_{opt}EL για κάθε τιμή του α.

4.6 Επίδραση της μεταβολής του σ

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο τη διασπορά σ της κανονικής κατανομής του χαρακτηριστικού που εξετάζεται, παίρνουμε το Σχήμα 6.

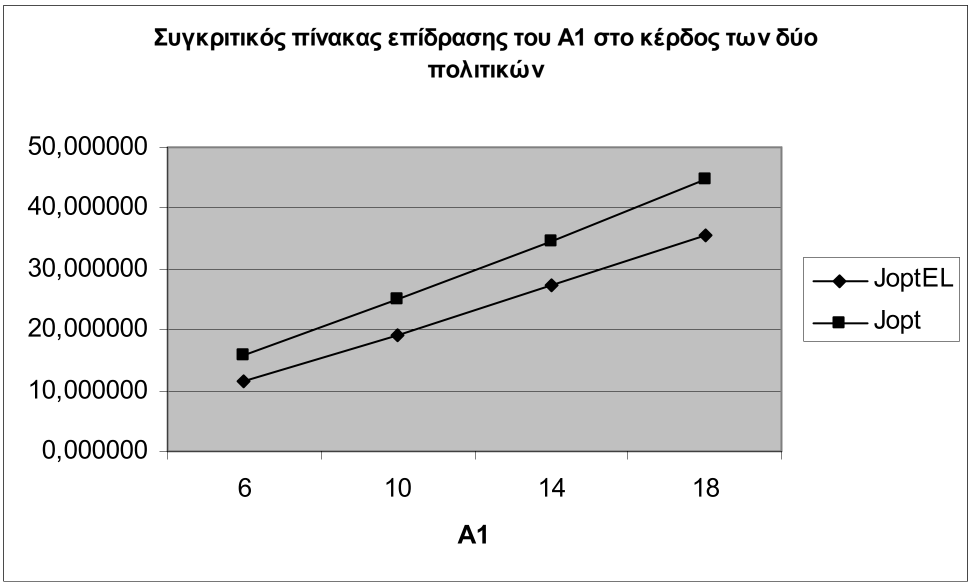


Σχήμα 6 Συγκριτικός πίνακας επίδρασης του σ στο κέρδος των δύο πολιτικών

Καθώς η διασπορά αυξάνει, τα κέρδη μειώνονται, γιατί όλο και λιγότερα κομμάτια έχουν τιμή εξεταζόμενου χαρακτηριστικού κοντά στην ιδανική τιμή t, άρα μεγαλώνει το κόστος ποιότητας. Το γραμμή του κέρδους της νέας πολιτικής υπερκαλύπτει τη γραμμή του κέρδους της πολιτικής Hong και Elsayed.

4.7 Επίδραση της μεταβολής του A_1

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο την αξία πώλησης A_1 κομματιού στην αγορά 1, παίρνουμε το Σχήμα 7.

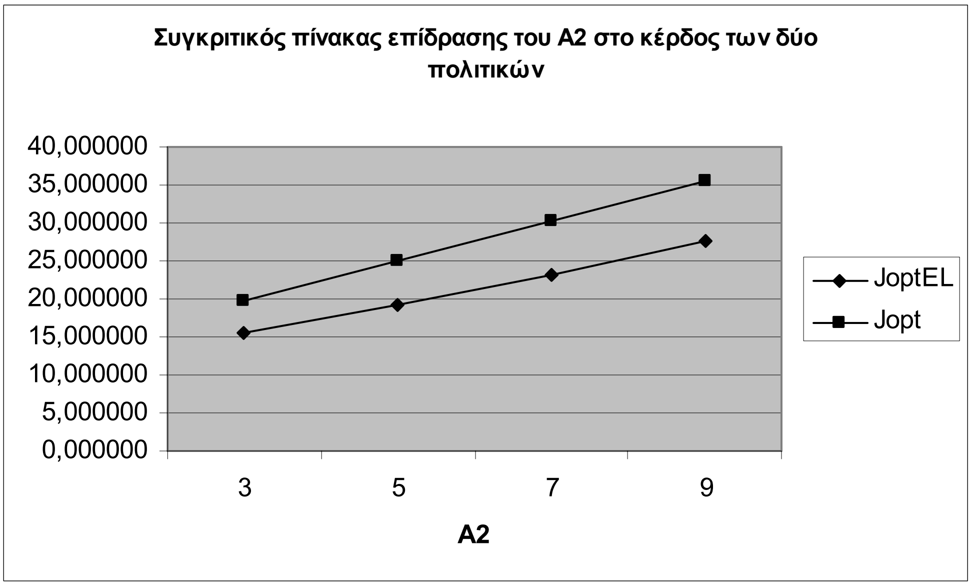


Σχήμα 7 Συγκριτικός πίνακας επίδρασης του Α1 στο κέρδος των δύο πολιτικών

Όσο αυξάνει η αξία πώλησης A_1, τόσο αυξάνονται και τα κέρδη, με το J_{opt} να είναι μεγαλύτερο.

4.8 Επίδραση της μεταβολής του A_2

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο την αξία πώλησης A_2 κομματιού στην αγορά 2, παίρνουμε το Σχήμα 8.

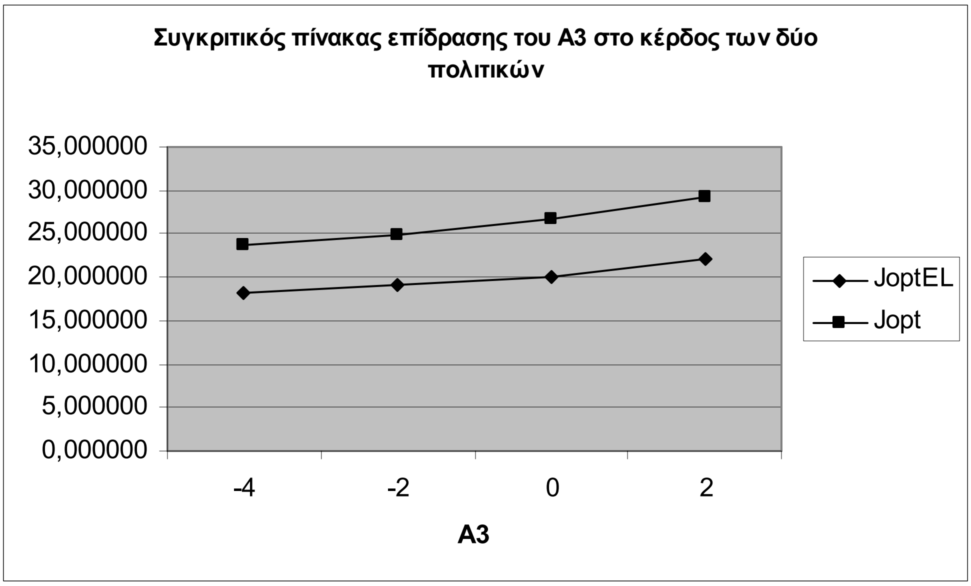


Σχήμα 8 Συγκριτικός πίνακας επίδρασης του Α2 στο κέρδος των δύο πολιτικών

Όπως και στο προηγούμενο γράφημα, όσο αυξάνει η αξία A_2 τόσαο αυξάνουν τα κέρδη. Το κέρδος της πολιτικής $n_1\lambda_1\lambda_2$ είναι υψηλότερο.

4.9 Επίδραση της μεταβολής του A_3

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το κόστος A_3 (θετικό ή αρνητικό) απόρριψης κομματιού, παίρνουμε το Σχήμα 9.

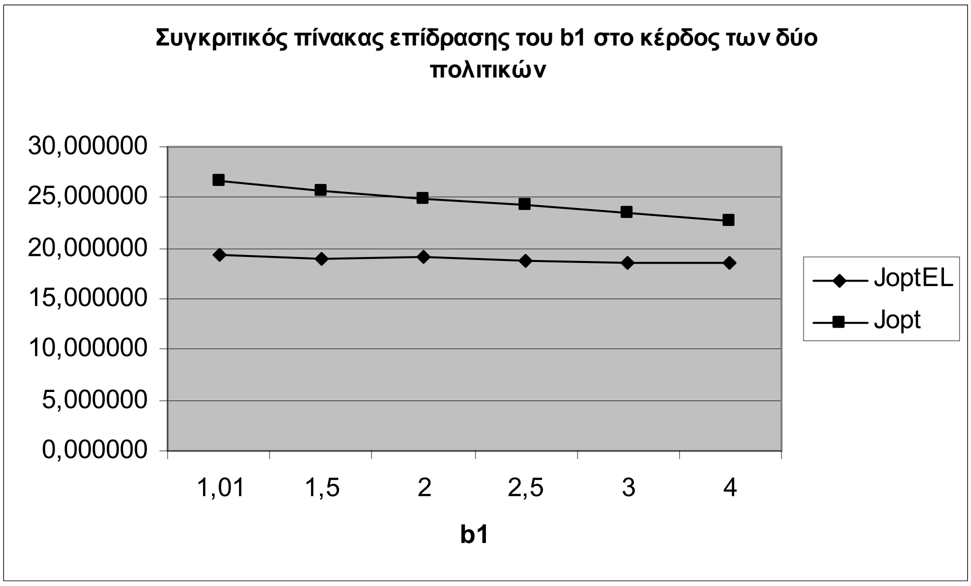


Σχήμα 9 Συγκριτικός πίνακας επίδρασης του A3 στο κέρδος των δύο πολιτικών

Όσο μεγαλύτερο είναι το A_3, τόσο αυξάνει το κέρδος, με το κέρδος J_{opt} να είναι πάντα καλύτερο.

4.10 Επίδραση της μεταβολής του b_1

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το συντελεστή κόστους ποιότητας b_1 για την αγορά 1, παίρνουμε το Σχήμα 10.

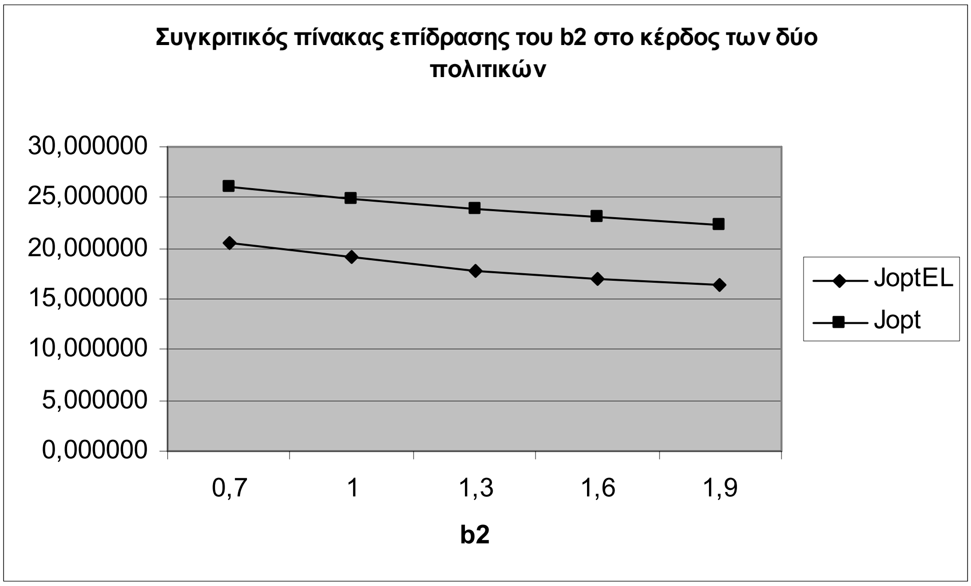


Σχήμα 10 Συγκριτικός πίνακας επίδρασης του b1 στο κέρδος των δύο πολιτικών

Όσο ο συντελεστής κόστους ποιότητας b_1 αυξάνεται, τόσο τα κέρδη ελαττώνονται. Το κέρδος της πολιτικής $n_1\lambda_1\lambda_2$ είναι συνεχώς μεγαλύτερο από το κέρδος της πολιτικής $n_1\lambda_1$.

4.11 Επίδραση της μεταβολής του b_2

Διατηρώντας τις υπόλοιπες παραμέτρους σταθερές και μεταβάλλοντας μόνο το συντελεστή κόστους ποιότητας b_2 για την αγορά 2, παίρνουμε το Σχήμα 11.



Σχήμα 11 Συγκριτικός πίνακας επίδρασης του b2 στο κέρδος των δύο πολιτικών

Όσο ο συντελεστής κόστους ποιότητας b_2 αυξάνεται, τόσο τα κέρδη ελαττώνονται. Το κέρδος της πολιτικής $n_1\lambda_1\lambda_2$ είναι συνεχώς μεγαλύτερο από το κέρδος της πολιτικής $n_1\lambda_1$.

4.12 Συμπεράσματα

Στα παραπάνω πειράματα, όπου μεταβάλαμε όλες τις παραμέτρους του προβλήματος φάνηκε ότι σε κάθε περίπτωση το κέρδος J_{opt} που αποδίδει η πολιτική $n_1\lambda_1\lambda_2$ είναι μεγαλύτερο από το κέρδος $J_{opt}EL$ που αποδίδει η πολιτική $n_1\lambda_1$.

Οπότε αν θεωρήσουμε ένα σύστημα παραγωγής με εναλλακτικές αγορές, όπου δεν έχουμε συνεχώς διαθέσιμους πελάτες και άρα πρέπει να τηρείται απόθεμα συμφέρει το εξής: Αντί να δίνουμε αυστηρά στην αγορά n μόνο προϊόντα διαλογής n, να δίνουμε στην αγορά n και προϊόντα της αμέσως καλύτερης διαλογής, εφόσον δεν έχουμε της διαλογής n.

Δεν είναι δυνατό να εγγυηθούμε ότι η πολιτική $n_1\lambda_1\lambda_2$ είναι βέλτιστη. Πιθανόν να υπάρχει και καλύτερη πολιτική. Αυτό που μπορούμε να πούμε με βεβαιότητα είναι ότι αποτελεί βελτίωση της πολιτικής $n_1\lambda_1$ για την περίπτωση που μας ενδιαφέρει να εξετάζουμε και το κόστος αποθέματος.

Βιβλιογραφία

Duffuaa, S. O. and Al-Najjar, H. J. (1995) An optimal complete inspection plan for critical multicharacteristic components. *Journal of Operational Research Society*, **46**, 930–942.

Hong S. H. and Elsayed E. A. (1998) Economic complete inspection plans with multi-decision alternatives. *International Journal of Production Research*, **26**, 203–217.

Hui, Y. V. (1990) Economic design of a complete inspection plan for bivariate products. *International Journal of Production Research*, **28**, 259–265.

Kouikoglou, V. S. and Phillis, Y. A. (2002) Design of product specifications and control policies in a single-stage production system. *IIE Transactions*, **34**, 590–600.

Ng, W. C. and Hui, Y. V. (1996). Economic design of a complete inspection plan with interactive quality improvement. *European Journal of Operational Research*, **96**, 122–129.

Stidham, S. Jr and Weber, R.R. (1993) A survey of Markov decision models for control of networks of queues. *Queueing Systems*, **13**, 291–314.

Taguchi, G. (1984) *Quality evaluation for quality assurance* (Romulus, MI: American Supplier Institute).

Tang, J. and Tang, K. (1994) Design of screening procedures: a review. *Journal of Quality Technology*, **26**, 209–226.

Tang, K. (1987) Economic design of a one-sided screening procedure using a correlated variable. *Technometrics*, **29**, 477–485.

Tang, K. (1988) Economic design of product speci®cations for a complete inspection plan. *International Journal of Production Research*, **26**, 203–217.

Tang, K. and Schneider, H. (1987) The e)ects of inspection error on a complete inspection plan. *IIE Transactions*, **19**, 421–428.

Ιωαννίδης, Ε. (2004) Συνεργαζόμενες πολιτικές ελέγχου αποθεμάτων και αποδοχής παραγγελιών σε συστήματα παραγωγής. Διδακτορική διατριβή, Πολυτεχνείο Κρήτης, Χανιά

Παράρτημα 1. Αλγόριθμος υπολογισμού κέρδους Jopt της πολιτικής $n_1\lambda_1\lambda_2$

```
      PROGRAM QUICK6GR
$DEBUG
$LARGE

C PROGRAMMA YPOLOGISMOU PITHANOTHTWN
C DHMIOYRGOUME PINAKES A, B, C, OPOY ARXIKA OLA TOYS TA STOIXEIA EINAI 0

      DOUBLE PRECISION A, B, C, D, E, F, H, P, Q, R, L1, L2, M, M1, M2, SUM
      DOUBLE PRECISION D2MIN, D2MAX, D1MIN, D1MAX, D1STEP, D2STEP
      DOUBLE PRECISION HH, JJ1, A1, A2, H1, JOPT, D1OPT, D2OPT, MEAN
      DOUBLE PRECISION PNISS, QCA1, QCA2, QCB, AA1, AA2, PII, TEMP, NEW
      DOUBLE PRECISION B1, B2, A3, PN1IS0, PN, SDV, Q1, Q2, D1, D2, PN2IS0
C      DOUBLE PRECISION F1
      DIMENSION A(50, 50), B(50, 50), C(50, 50), D(50, 50), E(50, 50)
      DIMENSION F(36, 36, 36), H(50, 50), R(50), Q(50)
C      DIMENSION F1(7, 40, 40)
      DIMENSION P(50, 50)
      INTEGER S, I, J, K, N1, N, SMAX, L, ND1, ND2, MAXST, MAXST1, SOPT

      DO 5 I=1, 50
       DO 6 J=1, 50
          A(I, J)=0
          B(I, J)=0
          C(I, J)=0
    6  CONTINUE
    5 CONTINUE

C      WRITE(*, *)'DWSE L1, L2, M1, M2, S'
C      READ(*, *) L1
C      READ(*, *) L2
C      READ(*, *) M
C      READ(*, *) S
      L1=3.D0
      L2=3.D0
      M=6.D0
      H1=1.D0
      B1=2.D0
      B2=1.D0
      A1=10.D0
      A2=5.D0
      A3=-2.D0
      T=11.D0
      MEAN=10.D0
      SDV=3.D0
      SMAX=15
      MAXST=20
      D2MIN=0.01*SDV
      D2MAX=3*SDV
      D1MIN=D2MIN*1.01
      D2STEP=(D2MAX-D2MIN)/MAXST
      DO 102 ND2=0, MAXST
       D2=D2MAX-ND2*D2STEP
       D1=D2
       MAXST1=MAXST-ND2
       IF (MAXST1.EQ.0) THEN
        MAXST1=1
        D1STEP=0.D0
       ELSE
        D1STEP=(D2-D1MIN)/MAXST1
       ENDIF
       DO 101 ND1=1, MAXST1
        D1=D2-ND1*D1STEP
        CALL TATA(T-D1, T+D1, MEAN, SDV**2, NEW)
        Q1=NEW
        CALL TATA(T-D2, T+D2, MEAN, SDV**2, NEW)
        Q2=NEW
        WRITE(*, *)'Q1=', Q1, '  Q2=', Q2
        Q2=Q2-Q1
        WRITE(*, *)'Q1-Q2=', Q2
        WRITE(*, *)'D1=', D1, '  D2=', D2
```

```
        WRITE(*, *)'T=', T, '  MEAN=', MEAN, '  SDV=', SDV
        M1=M*Q1
        M2=M*Q2
        IF (D1.EQ.D2) M2=M1


      DO 100 S=1, SMAX
      DO 99 N=S, S+1

C H LOUPA "DO 100" EINAI I "MEGALI" LOUPA TOY ALGORITHMOU.
C 1.TO STOIXEIO (1, 1) EINAI "SYNORIAKO" KAI DIAFEREI APO TA
C   YPOLOIPA THS DIAGWNIOY:

       IF (N.EQ.1) THEN
        A(1, 1)=M1+M2

C 2.GIA TA YPOLOIPA STOIXEIA THS DIAGWNIOY MEXRI N-1:

        ELSEIF (N.LT.S+1) THEN
         DO 10 I=1, N-1
          A(I, I)=L1+L2+M1+M2
   10    CONTINUE
         A(N, N)=L2+M1+M2

C 3.ENW TO TELEYTAIO STOIXEIO THS DIAGWNIOY EINAI:

        ELSE
         DO 15 I=1, S
          A(I, I)=L1+L2
   15    CONTINUE
         A(S+1, S+1)=L2
       ENDIF

       OPEN (7, FILE='AMATRIX.OUT', STATUS='NEW')
       WRITE(7, *) 'N=', N
       DO 11 I=1, S+1
        DO 12 J=1, S+1
         WRITE(7, *) 'A(', I, ', ', J, ')=', A(I, J)
12      CONTINUE
11     CONTINUE

C??????????????????????????????

C GIA TON PINAKA B TA PRAGMATA EINAI PIO APLA:



        IF (N.GT.1) THEN
         DO 20 I=1, N-1
          B(I, I)=M1
          B(I+1, I)=M2
   20    CONTINUE
        ENDIF

C GIA TON C EXOUME:

       IF (N.LT.S+1) THEN
         C(1, 1)=L1+L2
         C(1, 2)=L2
         DO 25 I=2, N
          C(I, I)=L1
          C(I, I+1)=L2
   25    CONTINUE
       ENDIF

C: YPOLOGISMOS TOU G

       IF (N.EQ.1) THEN
        CALL INVERT(A, D, 1)
        CALL MUL(D, C, E, 1, 1, 2)
        DO 30 I=1, 1
         DO 35 J=1, 2
          F(N, I, J)=E(I, J)
   35    CONTINUE
   30   CONTINUE
```

```
      ELSE IF(N.LT.S+1) THEN
        OPEN (11, FILE='GGMATRIX.OUT', STATUS='NEW')
       DO 31 I=1, N-1
        DO 36 J=1, N
         E(I, J)=F(N-1, I, J)
          WRITE(11, *) 'G(', I, ', ', J, ')=', E(I, J)
   36   CONTINUE
   31  CONTINUE
       CALL MUL(B, E, D, N, N-1, N)
       CALL SUB(A, D, E, N, N)

C       IF (N.EQ.4) THEN
C        OPEN (8, FILE='GMATRIX.OUT', STATUS='NEW')
C        WRITE(8, *) 'N=', N
C        DO 41 I=1, N
C         DO 46 J=1, N+1
C          WRITE(8, *) 'E(', I, ', ', J, ')=', E(I, J)
C          WRITE(8, *) E(I, J)
C   46    CONTINUE
C   41   CONTINUE
C       ENDIF

       CALL INVERT(E, D, N)
       CALL MUL(D, C, E, N, N, N+1)
       OPEN (7, FILE='G4MATRIX.OUT', STATUS='NEW')
       WRITE(7, *) 'N=', N
       DO 40 I=1, N
        DO 45 J=1, N+1
         F(N, I, J)=E(I, J)
         WRITE(7, *) 'G(', I, ', ', J, ')=', E(I, J)
   45   CONTINUE
   40  CONTINUE
       DO 50 N1=1, N-1
        DO 52 I=1, N1
         DO 54 J=1, N
          D(I, J)=F(N1, I, J)
   54    CONTINUE
   52   CONTINUE
        CALL MUL(D, E, H, N1, N, N+1)
        DO 56 I=1, N1
         DO 58 J=1, N+1
          F(N1, I, J)=H(I, J)
   58    CONTINUE
   56   CONTINUE
   50  CONTINUE

        OPEN (8, FILE='FMATRIX.OUT', STATUS='NEW')
        WRITE(8, *) 'N=', N
        DO 41 I=1, N
         DO 46 J=1, N+1
          WRITE(8, *) 'F(', I, ', ', J, ')=', F(N, I, J)
C          WRITE(8, *) E(I, J)
   46    CONTINUE
   41   CONTINUE

      ELSE
       CALL MUL(B, E, D, S+1, S, S+1)
       CALL SUB(A, D, E, S+1, S+1)
      ENDIF
  99  CONTINUE

     DO 60 J=1, S+1
      E(S+1, J)=1
   60 CONTINUE

     DO 70 J=1, S+1
      DO 72 N1=1, S
       DO 74 I=1, N1
        E(S+1, J)=E(S+1, J)+F(N1, I, J)
   74  CONTINUE
   72 CONTINUE
   70 CONTINUE

     DO 76 I=1, S
      Q(I)=0
```

```
 76 CONTINUE
    Q(S+1)=1

    CALL SOLVE(E, Q, R, S+1)
    SUM=0
    DO 78 I=1, S+1
     P(S+1, I)=R(I)
     SUM=SUM+P(S+1, I)
 78 CONTINUE
    DO 80 N=1, S
     DO 82 I=1, N
      P(N, I)=0
       DO 84 J=1, S+1
       P(N, I)=P(N, I)+F(N, I, J)*R(J)
 84   CONTINUE
      SUM=SUM+P(N, I)
 82  CONTINUE
 80 CONTINUE


       PII=3.141592654

       QCA1=(T+D1-MEAN)*DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
       QCA1=QCA1-((T-D1-MEAN)*DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2))))
       TEMP=DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
       TEMP=TEMP-DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2)))
       QCA1=QCA1+2*(MEAN-T)*TEMP
       QCA1=-(SDV/(SQRT(2*PII)))*QCA1
       QCA1=QCA1+Q1*(SDV**2)+Q1*((MEAN-T)**2)

       QCA2=(T+D2-MEAN)*DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
       QCA2=QCA2-((T-D2-MEAN)*DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2))))
       TEMP=DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
       TEMP=TEMP-DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2)))
       QCA2=QCA2+2*(MEAN-T)*TEMP
       QCA2=-(SDV/(SQRT(2*PII)))*QCA2
       QCA2=QCA2+(Q1+Q2)*(SDV**2)+(Q1+Q2)*((MEAN-T)**2)

       QCB=QCA2-QCA1
       AA1=A1-B1*QCA1
       HH=0.D0
       PNISS=0.D0
       PN1IS0=0.D0
       PN2IS0=0.D0
       DO 93 N=1, S+1
        PN=0.D0
        DO 94 I=1, N
         PN=PN+P(N, I)
         IF (N.EQ.S+1) THEN
          PNISS=PNISS+P(N, I)
         ENDIF
 94     CONTINUE
        HH=HH+(N-1)*PN
        PN1IS0=PN1IS0+P(N, N)
        PN2IS0=PN2IS0+P(N, 1)
 93    CONTINUE
       AA2=A2-B2*(PN2IS0*QCA1+(1-PN2IS0)*QCB)
       JJ1=AA1*L1*(1-PN1IS0)+AA2*L2*(1-P(1, 1))
       JJ1=JJ1+A3*(1-Q1-Q2)*M*(1-PNISS)-H1*HH
       IF (JJ1.GT.JOPT) THEN
        JOPT=JJ1
        SOPT=S
        D1OPT=D1
        D2OPT=D2
       ENDIF
100   CONTINUE
101  CONTINUE
102 CONTINUE

    OPEN (9, FILE='PROB.OUT', STATUS='NEW')
    WRITE(9, *) 'SUM PROB=', SUM
    DO 91 N=1, S
     DO 92 I=1, N
      WRITE(9, *) 'P(', N, ', ', I, ')=', P(N, I)
 92  CONTINUE
```

```
   91 CONTINUE
      WRITE(9, *) 'JOPT=', JOPT
      WRITE(9, *) 'SOPT=', SOPT
      WRITE(9, *) 'D1=', D1
      WRITE(9, *) 'D2=', D2
      WRITE(9, *) 'D1OPT=', D1OPT
      WRITE(9, *) 'D2OPT=', D2OPT
      WRITE(9, *) 'Q1=', Q1
      WRITE(9, *) 'Q2=', Q2

      STOP
      END



      SUBROUTINE SUB(A, B, C, L, M)
     INTEGER L, M, I, J
C            C=A-B
      DOUBLE PRECISION A, B, C
      DIMENSION A(50, 50), B(50, 50), C(50, 50)
      DO 1 I=1, L
       DO 2 J=1, M
        C(I, J)=A(I, J)-B(I, J)
    2  CONTINUE
    1 CONTINUE
      RETURN
      END

C
      SUBROUTINE MUL(A, B, C, L, M, N)
     INTEGER L, M, N, I, J, K
C           C=AB
      DOUBLE PRECISION A, B, C
      DIMENSION A(50, 50), B(50, 50), C(50, 50)
      DO 1 I=1, L
       DO 2 J=1, N
        C(I, J)=0
        DO 3 K=1, M
         C(I, J)=C(I, J)+A(I, K)*B(K, J)
    3   CONTINUE
    2  CONTINUE
    1 CONTINUE
      RETURN
      END
C

      SUBROUTINE SOLVE(A, B, P, N)
C THE FOLLOWING PROGRAM SOLVES THE LINEAR SYSTEM
C           AP=B   => P=[INVERSE OF A]B
C WHERE
C   A IS AN N*N MATRIX (KNOWN)
C   P IS A COLUMN VECTOR OF SIZE N (UNKNOWN)
C   B IS A COLUMN VECTOR OF SIZE N (KNOWN)
C   AND N < OR = 200
C A, B, P MUST BE DECLARED EITHER AS
C DOUBLE PRECISION A, B, P AND THEN DIMENSION A(200, 200), B(200), P(200)
C OR AS
C REAL*8 A(100, 100), B(100), P(100)
C THE TWO DECLARATIONS ARE EQUIVALENT IN FORTRAN.

      DOUBLE PRECISION A, B, P, D, E
      DIMENSION A(50, 50), B(50), P(50), D(50, 50), E(50)
     INTEGER N, I
C---    THIS PERFORMS "LU DECOMPOSITION" ON A

         OPEN (9, FILE='GMATRIX.OUT', STATUS='NEW')
         WRITE(9, *) 'N=', N
         DO 42 I=1, N
          DO 47 J=1, N
           WRITE(9, *) 'A(', I, ', ', J, ')=', A(I, J)
C           WRITE(9, *) E(I, J)
   47     CONTINUE
   42    CONTINUE
```

```
      CALL INVERT(A, D, N)
      CALL MUL(D, B, E, N, N, 1)
C---    LATER, WE SHALL SOLVE FOR THE UNKNOWN VECTOR BY "BACKSUBSTITUTION"
C---    USING LUBKSB, WHICH NEEDS TO INITIALIZE THE SOLUTION AS P=B.
      DO 10 I=1, N
       P(I)=E(I)
10    CONTINUE
      RETURN
      END


      SUBROUTINE INVERT(A, B, N)
C THE FOLLOWING PROGRAM SOLVES THE LINEAR SYSTEM
C           AB=I   => B=[INVERSE OF A]
C WHERE
C   A, B ARE AN N*N MATRICES (KNOWN)
C   AND N < OR = 200
C SYMFWNA ME THN INVGR

      DOUBLE PRECISION A(50, 50), B(50, 50), D11, D22
      DO 10 I=1, N
       DO 15 J=1, N
        B(I, J)=0.
  15   CONTINUE
       B(I, I)=1.
  10  CONTINUE
      D11=1.D0
      DO 20 I=1, N
      D22=A(I, I)
      D11=D11*D22
       DO 25 J=1, N
        A(I, J)=A(I, J)/D22
        B(I, J)=B(I, J)/D22
 25    CONTINUE
       DO 30 J=1, N
        IF(I.EQ.J) GO TO 30
         D11=A(J, I)
         DO 35 K=1, N
          A(J, K)=A(J, K)-D11*A(I, K)
          B(J, K)=B(J, K)-D11*B(I, K)
 35      CONTINUE
 30    CONTINUE
 20   CONTINUE
      RETURN
      END

      SUBROUTINE TATA(A, B, AVE, VAR, OUT)
      IMPLICIT REAL*8 (A-H, O-Z)
      OUT=F(A, B, AVE, VAR)
      RETURN
      END

      FUNCTION F(A, B, AVE, VAR)
C-- COMPUTES THE PROBABILITY THAT A NORMAL(MEAN ( AVE, VARIANCE ( VAR)
C   VARIABLE IS IN THE INTERVAL (A, B). ALL VARIABLES MUST BE REAL*8.
C-- NOTE: IN FORTRAN WE CAN ASSIGN A TYPE (E.G REAL*8, ETC.) TO ALL
C   VARIABLES WHOSE NAME STARTS WITH A GIVEN LETTER USING IMPLICIT.
C   THUS, WE AVOID USING SPECIFYING THE DETAILED NAMES IN DOUBLE PRECISION.
C-- NOTE: ACCEPT IN THE CALLING PROGRAM SHOULD BE DECLARED AS REAL*8 USING
C   IMPLICIT.
      IMPLICIT REAL*8 (A-H, O-Z)
      IF(B.LT.A)THEN
       WRITE(*, *)'NEGATIVE AREA A>B', A, B
       STOP
      ENDIF
      Q1=.5*ERF(DABS(B-AVE)/(2*VAR)**.5)
      Q2=.5*ERF(DABS(A-AVE)/(2*VAR)**.5)
C     WRITE(*, '(20H A, B, AVE, VAR, Q1, Q2  , 6F8.3)')A, B, AVE, VAR, Q1, Q2
      IF ((A-AVE)*(B-AVE).LE.0.D0) THEN
       F=Q1+Q2
      ELSE
       F=DABS(Q1-Q2)
      ENDIF
      RETURN
      END
```

```
      FUNCTION ERF(Y)
      IMPLICIT REAL*8 (A-H, O-Z)
C      DOUBLE PRECISION ERF, X, GAMMP
      IF(Y.LT.0.D0)THEN
      WRITE(*, *)'NEGATIVE ARGUMENT IN ERF'
      STOP
      ELSE
      A=.5D0
      X=Y**2
      IF(X.LT.0..OR.A.LE.0.)PAUSE 'BAD ARGUMENTS IN GAMMP'
      IF(X.LT.A+1.)THEN
       CALL GSER(GAMSER, A, X, GLN)
       ERF=GAMSER
      ELSE
       CALL GCF(GAMMCF, A, X, GLN)
       ERF=1.-GAMMCF
      ENDIF
      ENDIF
      RETURN
      END

      SUBROUTINE GCF(GAMMCF, A, X, GLN)
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (ITMAX=100, EPS=3.E-7, FPMIN=1.E-30)
C     USES GAMMLN
      GLN=GAMMLN(A)
      B=X+1.-A
      C=1./FPMIN
      D=1./B
      H=D
      DO 11 I=1, ITMAX
       AN=-I*(I-A)
       B=B+2.
       D=AN*D+B
       IF(DABS(D).LT.FPMIN)D=FPMIN
       C=B+AN/C
       IF(DABS(C).LT.FPMIN)C=FPMIN
       D=1./D
       DEL=D*C
       H=H*DEL
       IF(DABS(DEL-1.).LT.EPS)GOTO 1
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL IN GCF'
1     GAMMCF=DEXP(-X+A*DLOG(X)-GLN)*H
      RETURN
      END

      SUBROUTINE GSER(GAMSER, A, X, GLN)
      IMPLICIT REAL*8 (A-H, O-Z)
C     INTEGER ITMAX
C     REAL A, GAMSER, GLN, X, EPS
      PARAMETER (ITMAX=100, EPS=3.E-7)
CU    USES GAMMLN
C     INTEGER N
C     REAL AP, DEL, SUM, GAMMLN
      GLN=GAMMLN(A)
      IF(X.LE.0.)THEN
       IF(X.LT.0.)PAUSE 'X < 0 IN GSER'
       GAMSER=0.
       RETURN
      ENDIF
      AP=A
      SUM=1./A
      DEL=SUM
      DO 11 N=1, ITMAX
       AP=AP+1.
       DEL=DEL*X/AP
       SUM=SUM+DEL
       IF(DABS(DEL).LT.DABS(SUM)*EPS)GOTO 1
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL IN GSER'
1     GAMSER=SUM*DEXP(-X+A*DLOG(X)-GLN)
      RETURN
```

```
      END

      FUNCTION GAMMLN(XX)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION COF(6)
      SAVE COF, STP
      DATA COF, STP/76.18009172947146D0, -86.50532032941677D0,
     *24.01409824083091D0, -1.231739572450155D0, .1208650973866179D-2,
     *-.5395239384953D-5, 2.5066282746310005D0/
      X=XX
      Y=X
      TMP=X+5.5D0
      TMP=(X+0.5D0)*DLOG(TMP)-TMP
      SER=1.000000000190015D0
      DO 11 J=1, 6
       Y=Y+1.D0
       SER=SER+COF(J)/Y
11    CONTINUE
      GAMMLN=TMP+DLOG(STP*SER/X)
      RETURN
      END
```

Παράρτημα 2. Αλγόριθμος υπολογισμού κέρδους JoptEL της πολιτικής $n_1\lambda_1$

```
      PROGRAM PROASS
$DEBUG
$LARGE

C PROGRAMMA YPOLOGISMOU PITHANOTHTWN

C DHMIOYRGOUME PINAKES A, B, C, OPOY ARXIKA OLA TOYS TA STOIXEIA EINAI 0

      DOUBLE PRECISION A, B, C, D, E, F, H, P, Q, R, L1, L2, M, M1, M2, SUM
      DOUBLE PRECISION D2MIN, D2MAX, D1MIN, D1MAX, D1STEP, D2STEP
      DOUBLE PRECISION HH, JJ1, A1, A2, H1, JOPT, D1OPT, D2OPT, MEAN
      DOUBLE PRECISION PNISS, QCA1, QCA2, QCB, AA1, AA2, PII, TEMP, NEW
      DOUBLE PRECISION B1, B2, A3, PN1IS0, PN, SDV, Q1, Q2, D1, D2
      DIMENSION A(50, 50), B(50, 50), C(50, 50), D(50, 50), E(50, 50)
      DIMENSION F(36, 36, 36), H(50, 50), R(50), Q(50)
      DIMENSION P(50, 50)
      INTEGER S, I, J, K, N1, N, SMAX, L, ND1, ND2, MAXST, MAXST1, SOPT

      DO 5 I=1, 50
       DO 6 J=1, 50
          A(I, J)=0
          B(I, J)=0
          C(I, J)=0
    6  CONTINUE
    5 CONTINUE

C      WRITE(*, *)'DWSE L1, L2, M1, M2, S'
C      READ(*, *) L1
C      READ(*, *) L2
C      READ(*, *) M
C      READ(*, *) S
      L1=3.D0
      L2=3.D0
      M=6.D0
      H1=1.D0
      B1=2.D0
      B2=1.D0
      A1=10.D0
      A2=5.D0
      A3=-2.D0
      T=11.D0
      MEAN=10.D0
      SDV=3.D0
      SMAX=15
      MAXST=50
      D2MIN=0.01*SDV
      D2MAX=3*SDV
      D1MIN=D2MIN*1.01
      D2STEP=(D2MAX-D2MIN)/MAXST
      DO 102 ND2=0, MAXST
       D2=D2MAX-ND2*D2STEP
       D1=D2
       MAXST1=MAXST-ND2
       IF (MAXST1.EQ.0) THEN
        MAXST1=1
        D1STEP=0.D0
       ELSE
        D1STEP=(D2-D1MIN)/MAXST1
       ENDIF
       DO 101 ND1=1, MAXST1
        D1=D2-ND1*D1STEP
        CALL TATA(T-D1, T+D1, MEAN, SDV**2, NEW)
        Q1=NEW
        CALL TATA(T-D2, T+D2, MEAN, SDV**2, NEW)
        Q2=NEW
        WRITE(*, *)'Q1=', Q1, '  Q2=', Q2
```

```
        Q2=Q2-Q1
        WRITE(*, *)'Q2=', Q2
        WRITE(*, *)'D1=', D1, '  D2=', D2
        WRITE(*, *)'T=', T, '  MEAN=', MEAN, '  SDV=', SDV
        PII=3.141592654

        QCA1=(T+D1-MEAN)*DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
        QCA1=QCA1-((T-D1-MEAN)*DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2))))
        TEMP=DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
        TEMP=TEMP-DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2)))
        QCA1=QCA1+2*(MEAN-T)*TEMP
        QCA1=-(SDV/(SQRT(2*PII)))*QCA1
        QCA1=QCA1+Q1*(SDV**2)+Q1*((MEAN-T)**2)

        QCA2=(T+D2-MEAN)*DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
        QCA2=QCA2-((T-D2-MEAN)*DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2))))
        TEMP=DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
        TEMP=TEMP-DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2)))
        QCA2=QCA2+2*(MEAN-T)*TEMP
        QCA2=-(SDV/(SQRT(2*PII)))*QCA2
        QCA2=QCA2+(Q1+Q2)*(SDV**2)+(Q1+Q2)*((MEAN-T)**2)

        QCB=QCA2-QCA1
        AA1=A1-B1*QCA1
        AA2=A2-B2*QCB
        JJ1=AA1*Q1+AA2*Q2+A3*(1-Q1-Q2)
        IF (JJ1.GT.JOPT) THEN
         JOPT=JJ1
         SOPT=S
         D1OPT=D1
         D2OPT=D2
        ENDIF
  101  CONTINUE
  102 CONTINUE
      D1=D1OPT
      D2=D2OPT
      CALL TATA(T-D1, T+D1, MEAN, SDV**2, NEW)
      Q1=NEW
      CALL TATA(T-D2, T+D2, MEAN, SDV**2, NEW)
      Q2=NEW
      Q2=Q2-Q1
      M1=M*Q1
      M2=M*Q2
      QCA1=(T+D1-MEAN)*DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
      QCA1=QCA1-((T-D1-MEAN)*DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2))))
      TEMP=DEXP(-((T+D1-MEAN)**2)/(2*(SDV**2)))
      TEMP=TEMP-DEXP(-((T-D1-MEAN)**2)/(2*(SDV**2)))
      QCA1=QCA1+2*(MEAN-T)*TEMP
      QCA1=-(SDV/(SQRT(2*PII)))*QCA1
      QCA1=QCA1+Q1*(SDV**2)+Q1*((MEAN-T)**2)
      QCA2=(T+D2-MEAN)*DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
      QCA2=QCA2-((T-D2-MEAN)*DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2))))
      TEMP=DEXP(-((T+D2-MEAN)**2)/(2*(SDV**2)))
      TEMP=TEMP-DEXP(-((T-D2-MEAN)**2)/(2*(SDV**2)))
      QCA2=QCA2+2*(MEAN-T)*TEMP
      QCA2=-(SDV/(SQRT(2*PII)))*QCA2
      QCA2=QCA2+(Q1+Q2)*(SDV**2)+(Q1+Q2)*((MEAN-T)**2)
      QCB=QCA2-QCA1
      AA1=A1-B1*QCA1
      AA2=A2-B2*QCB

      OPEN (9, FILE='PROB7AEL.OUT', STATUS='NEW')
      WRITE(9, *) 'SUM PROB=', SUM
C      DO 91 N=1, S
C       DO 92 I=1, N
C        WRITE(9, *) 'P(', N, ', ', I, ')=', P(N, I)
C   92  CONTINUE
   91 CONTINUE
      WRITE(9, *) 'JOPT=', JOPT
      WRITE(9, *) 'SOPT=', SOPT
      WRITE(9, *) 'D1=', D1
      WRITE(9, *) 'D2=', D2
      WRITE(9, *) 'D1OPT=', D1OPT
      WRITE(9, *) 'D2OPT=', D2OPT
      WRITE(9, *) 'Q1=', Q1
```

```
      WRITE(9, *) 'Q2=', Q2

C   YPOLOGISMOS BELTISTOU S ME DEDOMEN TA D1OPT, D2OPT


      JOPT=0.D0
      DO 100 S=1, SMAX
      DO 99 N=S, S+1

C H LOUPA "DO 100" EINAI I "MEGALI" LOUPA TOY ALGORITHMOU.
C 1.TO STOIXEIO (1, 1) EINAI "SYNORIAKO" KAI DIAFEREI APO TA
C   YPOLOIPA THS DIAGWNIOY:

       IF (N.EQ.1) THEN
        A(1, 1)=M1+M2

C 2.GIA TA YPOLOIPA STOIXEIA THS DIAGWNIOY MEXRI N-1:

        ELSEIF (N.LT.S+1) THEN
         DO 10 I=1, N-1
          A(I, I)=L1+L2+M1+M2
   10    CONTINUE
         A(N, N)=L2+M1+M2

C 3.ENW TO TELEYTAIO STOIXEIO THS DIAGWNIOY EINAI:

        ELSE
         DO 15 I=1, S
          A(I, I)=L1+L2
   15    CONTINUE
         A(S+1, S+1)=L2
       ENDIF

C       OPEN (7, FILE='AMATRIX.OUT', STATUS='NEW')
C       WRITE(7, *) 'N=', N
C       DO 11 I=1, S+1
C        DO 12 J=1, S+1
C         WRITE(7, *) 'A(', I, ', ', J, ')=', A(I, J)
C12      CONTINUE
C11     CONTINUE

C??????????????????????????????

C GIA TON PINAKA B TA PRAGMATA EINAI PIO APLA:



        IF (N.GT.1) THEN
         DO 20 I=1, N-1
          B(I, I)=M1
          B(I+1, I)=M2
   20    CONTINUE
        ENDIF

C GIA TON C EXOUME:

       IF (N.LT.S+1) THEN
         C(1, 1)=L1+L2
         C(1, 2)=L2
         DO 25 I=2, N
          C(I, I)=L1
          C(I, I+1)=L2
   25    CONTINUE
       ENDIF

C: YPOLOGISMOS TOU G

       IF (N.EQ.1) THEN
        CALL INVERT(A, D, 1)
        CALL MUL(D, C, E, 1, 1, 2)
        DO 30 I=1, 1
         DO 35 J=1, 2
          F(N, I, J)=E(I, J)
   35    CONTINUE
   30   CONTINUE
```

```
       ELSE IF(N.LT.S+1) THEN
C         OPEN (11, FILE='GGMATRIX.OUT', STATUS='NEW')
        DO 31 I=1, N-1
         DO 36 J=1, N
          E(I, J)=F(N-1, I, J)
C           WRITE(11, *) 'G(', I, ', ', J, ')=', E(I, J)
   36    CONTINUE
   31   CONTINUE
        CALL MUL(B, E, D, N, N-1, N)
        CALL SUB(A, D, E, N, N)

C        IF (N.EQ.4) THEN
C         OPEN (8, FILE='GMATRIX.OUT', STATUS='NEW')
C         WRITE(8, *) 'N=', N
C         DO 41 I=1, N
C          DO 46 J=1, N+1
C           WRITE(8, *) 'E(', I, ', ', J, ')=', E(I, J)
C           WRITE(8, *) E(I, J)
C   46     CONTINUE
C   41    CONTINUE
C        ENDIF

        CALL INVERT(E, D, N)
        CALL MUL(D, C, E, N, N, N+1)
C        OPEN (7, FILE='G4MATRIX.OUT', STATUS='NEW')
C        WRITE(7, *) 'N=', N
        DO 40 I=1, N
         DO 45 J=1, N+1
          F(N, I, J)=E(I, J)
C          WRITE(7, *) 'G(', I, ', ', J, ')=', E(I, J)
   45    CONTINUE
   40   CONTINUE
        DO 50 N1=1, N-1
         DO 52 I=1, N1
          DO 54 J=1, N
           D(I, J)=F(N1, I, J)
   54     CONTINUE
   52    CONTINUE
         CALL MUL(D, E, H, N1, N, N+1)
         DO 56 I=1, N1
          DO 58 J=1, N+1
           F(N1, I, J)=H(I, J)
   58     CONTINUE
   56    CONTINUE
   50   CONTINUE

C         OPEN (8, FILE='FMATRIX.OUT', STATUS='NEW')
C         WRITE(8, *) 'N=', N
C         DO 41 I=1, N
C          DO 46 J=1, N+1
C           WRITE(8, *) 'F(', I, ', ', J, ')=', F(N, I, J)
C           WRITE(8, *) E(I, J)
C   46     CONTINUE
C   41    CONTINUE

       ELSE
        CALL MUL(B, E, D, S+1, S, S+1)
        CALL SUB(A, D, E, S+1, S+1)
       ENDIF
  99  CONTINUE

      DO 60 J=1, S+1
       E(S+1, J)=1
   60 CONTINUE

      DO 70 J=1, S+1
       DO 72 N1=1, S
        DO 74 I=1, N1
         E(S+1, J)=E(S+1, J)+F(N1, I, J)
   74   CONTINUE
   72  CONTINUE
   70 CONTINUE

      DO 76 I=1, S
       Q(I)=0
```

```
   76 CONTINUE
      Q(S+1)=1

      CALL SOLVE(E, Q, R, S+1)
      SUM=0
      DO 78 I=1, S+1
       P(S+1, I)=R(I)
       SUM=SUM+P(S+1, I)
   78 CONTINUE
      DO 80 N=1, S
       DO 82 I=1, N
        P(N, I)=0
         DO 84 J=1, S+1
         P(N, I)=P(N, I)+F(N, I, J)*R(J)
   84   CONTINUE
        SUM=SUM+P(N, I)
   82  CONTINUE
   80 CONTINUE

         HH=0.D0
         PNISS=0.D0
         PN1IS0=0.D0
         PN2IS0=0.D0
         DO 53 N=1, S+1
          PN=0.D0
          DO 64 I=1, N
           PN=PN+P(N, I)
           IF (N.EQ.S+1) THEN
            PNISS=PNISS+P(N, I)
           ENDIF
   64     CONTINUE
          HH=HH+(N-1)*PN
          PN1IS0=PN1IS0+P(N, N)
          PN2IS0=PN2IS0+P(N, 1)
   53    CONTINUE
         JJ1=AA1*L1*(1-PN1IS0)+AA2*L2*(1-PN2IS0)
         JJ1=JJ1+A3*((1-Q1-Q2)*M)*(1-PNISS)-H1*HH
         WRITE(*, *) 'JJ1=', JJ1
         IF (JJ1.GT.JOPT) THEN
          JOPT=JJ1
          SOPT=S
          D1OPT=D1
          D2OPT=D2
         ENDIF

  100   CONTINUE

      OPEN (11, FILE='PROB7EL.OUT', STATUS='NEW')
      WRITE(11, *) 'SUM PROB=', SUM
      DO 51 N=1, S
       DO 62 I=1, N
        WRITE(11, *) 'P(', N, ', ', I, ')=', P(N, I)
   62  CONTINUE
   51 CONTINUE
      WRITE(11, *) 'JOPT1=', JOPT
      WRITE(11, *) 'SOPT=', SOPT
      WRITE(11, *) 'D1=', D1
      WRITE(11, *) 'D2=', D2
      WRITE(11, *) 'D1OPT=', D1OPT
      WRITE(11, *) 'D2OPT=', D2OPT
      WRITE(11, *) 'Q1=', Q1
      WRITE(11, *) 'Q2=', Q2

      STOP
      END


      SUBROUTINE SUB(A, B, C, L, M)
     INTEGER L, M, I, J
C            C=A-B
      DOUBLE PRECISION A, B, C
      DIMENSION A(50, 50), B(50, 50), C(50, 50)
      DO 1 I=1, L
       DO 2 J=1, M
        C(I, J)=A(I, J)-B(I, J)
```

```
    2  CONTINUE
    1 CONTINUE
      RETURN
      END

C
      SUBROUTINE MUL(A, B, C, L, M, N)
     INTEGER L, M, N, I, J, K
C           C=AB
      DOUBLE PRECISION A, B, C
      DIMENSION A(50, 50), B(50, 50), C(50, 50)
      DO 1 I=1, L
       DO 2 J=1, N
        C(I, J)=0
        DO 3 K=1, M
         C(I, J)=C(I, J)+A(I, K)*B(K, J)
    3   CONTINUE
    2  CONTINUE
    1 CONTINUE
      RETURN
      END
C

      SUBROUTINE SOLVE(A, B, P, N)
C THE FOLLOWING PROGRAM SOLVES THE LINEAR SYSTEM
C           AP=B   => P=[INVERSE OF A]B
C WHERE
C   A IS AN N*N MATRIX (KNOWN)
C   P IS A COLUMN VECTOR OF SIZE N (UNKNOWN)
C   B IS A COLUMN VECTOR OF SIZE N (KNOWN)
C   AND N < OR = 200
C A, B, P MUST BE DECLARED EITHER AS
C DOUBLE PRECISION A, B, P AND THEN DIMENSION A(200, 200), B(200), P(200)
C OR AS
C REAL*8 A(100, 100), B(100), P(100)
C THE TWO DECLARATIONS ARE EQUIVALENT IN FORTRAN.

      DOUBLE PRECISION A, B, P, D, E
      DIMENSION A(50, 50), B(50), P(50), D(50, 50), E(50)
     INTEGER N, I
C---    THIS PERFORMS "LU DECOMPOSITION" ON A

C         OPEN (9, FILE='GMATRIX.OUT', STATUS='NEW')
C         WRITE(9, *) 'N=', N
C         DO 42 I=1, N
C          DO 47 J=1, N
C           WRITE(9, *) 'A(', I, ', ', J, ')=', A(I, J)
C           WRITE(9, *) E(I, J)
C   47     CONTINUE
C   42    CONTINUE


      CALL INVERT(A, D, N)
      CALL MUL(D, B, E, N, N, 1)
C---    LATER, WE SHALL SOLVE FOR THE UNKNOWN VECTOR BY "BACKSUBSTITUTION"
C---    USING LUBKSB, WHICH NEEDS TO INITIALIZE THE SOLUTION AS P=B.
      DO 10 I=1, N
       P(I)=E(I)
10    CONTINUE
      RETURN
      END


      SUBROUTINE INVERT(A, B, N)
C THE FOLLOWING PROGRAM SOLVES THE LINEAR SYSTEM
C           AB=I   => B=[INVERSE OF A]
C WHERE
C   A, B ARE AN N*N MATRICES (KNOWN)
C   AND N < OR = 200
C SYMFWNA ME THN INVGR

      DOUBLE PRECISION A(50, 50), B(50, 50), D11, D22
      DO 10 I=1, N
       DO 15 J=1, N
        B(I, J)=0.
```

```
  15   CONTINUE
       B(I, I)=1.
  10  CONTINUE
      D11=1.D0
      DO 20 I=1, N
      D22=A(I, I)
      D11=D11*D22
       DO 25 J=1, N
        A(I, J)=A(I, J)/D22
        B(I, J)=B(I, J)/D22
 25    CONTINUE
       DO 30 J=1, N
        IF(I.EQ.J) GO TO 30
         D11=A(J, I)
         DO 35 K=1, N
          A(J, K)=A(J, K)-D11*A(I, K)
          B(J, K)=B(J, K)-D11*B(I, K)
 35      CONTINUE
 30    CONTINUE
 20   CONTINUE
      RETURN
      END

      SUBROUTINE TATA(A, B, AVE, VAR, OUT)
      IMPLICIT REAL*8 (A-H, O-Z)
      OUT=F(A, B, AVE, VAR)
      RETURN
      END

      FUNCTION F(A, B, AVE, VAR)
C-- COMPUTES THE PROBABILITY THAT A NORMAL(MEAN ( AVE, VARIANCE ( VAR)
C   VARIABLE IS IN THE INTERVAL (A, B). ALL VARIABLES MUST BE REAL*8.
C-- NOTE: IN FORTRAN WE CAN ASSIGN A TYPE (E.G REAL*8, ETC.) TO ALL
C   VARIABLES WHOSE NAME STARTS WITH A GIVEN LETTER USING IMPLICIT.
C   THUS, WE AVOID USING SPECIFYING THE DETAILED NAMES IN DOUBLE PRECISION.
C-- NOTE: ACCEPT IN THE CALLING PROGRAM SHOULD BE DECLARED AS REAL*8 USING
C   IMPLICIT.
      IMPLICIT REAL*8 (A-H, O-Z)
      IF(B.LT.A)THEN
       WRITE(*, *)'NEGATIVE AREA A>B', A, B
       STOP
      ENDIF
      Q1=.5*ERF(DABS(B-AVE)/(2*VAR)**.5)
      Q2=.5*ERF(DABS(A-AVE)/(2*VAR)**.5)
C     WRITE(*, '(20H A, B, AVE, VAR, Q1, Q2  , 6F8.3)')A, B, AVE, VAR, Q1, Q2
      IF ((A-AVE)*(B-AVE).LE.0.D0) THEN
       F=Q1+Q2
      ELSE
       F=DABS(Q1-Q2)
      ENDIF
      RETURN
      END


      FUNCTION ERF(Y)
      IMPLICIT REAL*8 (A-H, O-Z)
C      DOUBLE PRECISION ERF, X, GAMMP
      IF(Y.LT.0.D0)THEN
      WRITE(*, *)'NEGATIVE ARGUMENT IN ERF'
      STOP
      ELSE
      A=.5D0
      X=Y**2
      IF(X.LT.0..OR.A.LE.0.)PAUSE 'BAD ARGUMENTS IN GAMMP'
      IF(X.LT.A+1.)THEN
       CALL GSER(GAMSER, A, X, GLN)
       ERF=GAMSER
      ELSE
       CALL GCF(GAMMCF, A, X, GLN)
       ERF=1.-GAMMCF
      ENDIF
      ENDIF
      RETURN
      END
```

```
      SUBROUTINE GCF(GAMMCF, A, X, GLN)
      IMPLICIT REAL*8 (A-H, O-Z)
      PARAMETER (ITMAX=100, EPS=3.E-7, FPMIN=1.E-30)
C     USES GAMMLN
      GLN=GAMMLN(A)
      B=X+1.-A
      C=1./FPMIN
      D=1./B
      H=D
      DO 11 I=1, ITMAX
       AN=-I*(I-A)
       B=B+2.
       D=AN*D+B
       IF(DABS(D).LT.FPMIN)D=FPMIN
       C=B+AN/C
       IF(DABS(C).LT.FPMIN)C=FPMIN
       D=1./D
       DEL=D*C
       H=H*DEL
       IF(DABS(DEL-1.).LT.EPS)GOTO 1
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL IN GCF'
1     GAMMCF=DEXP(-X+A*DLOG(X)-GLN)*H
      RETURN
      END

      SUBROUTINE GSER(GAMSER, A, X, GLN)
      IMPLICIT REAL*8 (A-H, O-Z)
C     INTEGER ITMAX
C     REAL A, GAMSER, GLN, X, EPS
      PARAMETER (ITMAX=100, EPS=3.E-7)
CU    USES GAMMLN
C     INTEGER N
C     REAL AP, DEL, SUM, GAMMLN
      GLN=GAMMLN(A)
      IF(X.LE.0.)THEN
       IF(X.LT.0.)PAUSE 'X < 0 IN GSER'
       GAMSER=0.
       RETURN
      ENDIF
      AP=A
      SUM=1./A
      DEL=SUM
      DO 11 N=1, ITMAX
       AP=AP+1.
       DEL=DEL*X/AP
       SUM=SUM+DEL
       IF(DABS(DEL).LT.DABS(SUM)*EPS)GOTO 1
11    CONTINUE
      PAUSE 'A TOO LARGE, ITMAX TOO SMALL IN GSER'
1     GAMSER=SUM*DEXP(-X+A*DLOG(X)-GLN)
      RETURN
      END

      FUNCTION GAMMLN(XX)
      IMPLICIT REAL*8 (A-H, O-Z)
      DIMENSION COF(6)
      SAVE COF, STP
      DATA COF, STP/76.18009172947146D0, -86.50532032941677D0,
     *24.01409824083091D0, -1.231739572450155D0, .1208650973866179D-2,
     *-.5395239384953D-5, 2.5066282746310005D0/
       X=XX
       Y=X
      TMP=X+5.5D0
      TMP=(X+0.5D0)*DLOG(TMP)-TMP
      SER=1.000000000190015D0
      DO 11 J=1, 6
       Y=Y+1.D0
       SER=SER+COF(J)/Y
11    CONTINUE
      GAMMLN=TMP+DLOG(STP*SER/X)
      RETURN
      END
```