

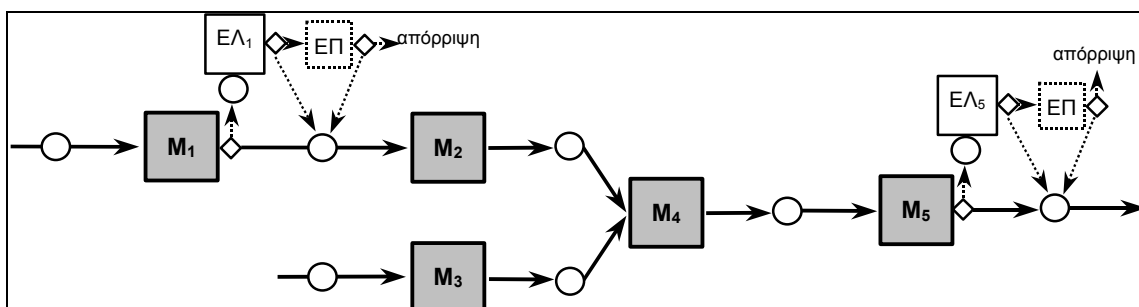


ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΑΡΑΓΩΓΗΣ & ΔΙΟΙΚΗΣΗΣ

***Ένα μοντέλο προσομοίωσης δικτύων συναρμολόγησης για την
βέλτιστη κατανομή σημείων ελέγχου ποιότητας***

Νικήτας Σπήλιος

Διπλωματική Εργασία



Τριμελής εξεταστική επιτροπή:
Κουϊκόγλου Βασίλειος (επιβλέπων)
Γρηγορούδης Ευάγγελος
Νικολός Ιωάννης

Χανιά, 2007

Copyright 2007 υπό Νικήτα Σπήλιο

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	5
1 ΕΙΣΑΓΩΓΗ.....	6
1.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	6
1.2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ	6
2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	8
2.1 ΔΙΚΤΥΑ ΣΥΝΑΡΜΟΛΟΓΗΣΗΣ	9
2.2 ΠΟΛΙΤΙΚΗ CONWIP	10
2.3 ΜΗΧΑΝΕΣ ΖΗΤΗΣΗΣ ΚΑΙ ΠΩΛΗΣΗΣ	10
2.4 ΣΤΑΘΜΟΙ ΕΛΕΓΧΟΥ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΕΠΙΘΕΩΡΗΣΗΣ	11
2.5 ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ	13
2.5.1 Λειτουργικές παράμετροι	13
2.5.2 Παράμετροι κόστους	14
3 ΑΞΙΟΛΟΓΗΣΗ ΕΝΑΛΛΑΚΤΙΚΩΝ ΛΥΣΕΩΝ.....	16
3.1 ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ	17
3.2 ΒΑΣΙΚΗ ΔΟΜΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	18
3.2.1 Χρόνοι επόμενων γεγονότων και πιθανότητες βλάβης	18
3.2.2 Τρόπος λειτουργίας του αλγορίθμου	19
3.3 ΓΕΓΟΝΟΤΑ ΣΥΣΤΗΜΑΤΟΣ	20
4 ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ	24
4.1 ΔΕΔΟΜΕΝΑ	24
4.2 ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΑΚΟΛΟΥΘΙΕΣ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ	26
4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ 1	27
4.4 ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΚΤΥΟΥ ΓΙΑ ΔΙΑΦΟΡΕΤΙΚΑ $Q(i)$ ΚΑΙ $S(i)$	28

5	ΣΥΜΠΕΡΑΣΜΑΤΑ - ΕΠΕΚΤΑΣΕΙΣ	30
6	ΒΙΒΛΙΟΓΡΑΦΙΑ	32
	ΠΑΡΑΡΤΗΜΑ: ΚΩΔΙΚΑΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ	33

ΠΡΟΛΟΓΟΣ

Στο σημείο αυτό, νιώθω την ανάγκη να ευχαριστήσω κάποιους ανθρώπους που με στήριξαν στην πορεία μου έως εδώ.

Τον πατέρα μου, Κανέλλο, την μητέρα μου, Κατερίνα, την αδερφή μου, Δήμητρα, και την γιαγιά μου Ασπασία για την αγάπη τους. Τους οφείλω μια συγνώμη... ξέρουν γιατί!

Τον επιβλέποντα καθηγητή και δάσκαλό μου Κ.Κουϊκόγλου γιατί με εμπιστεύθηκε και απλόχερα μου προσέφερε πολλά περισσότερα από ότι τόλμησα ποτέ μου να φανταστώ.

Όλους τους φίλους μου που με ανέχτηκαν όλα αυτά τα χρόνια. Τους υπόσχομαι ότι θα προσπαθήσω να με ανεχτούν πολλά ακόμη.

1 ΕΙΣΑΓΩΓΗ

1.1 ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Στην εποχή μας οι σύγχρονες και μεγάλες βιομηχανικές μονάδες έχουν εισάγει στην καθημερινή λειτουργία τους τον έλεγχο ποιότητας και έχουν αναγάγει σε βασική προτεραιότητά τους την διασφάλιση της παραγωγής άρτιων και αξιόπιστων προϊόντων. Ο λόγος είναι ότι το κόστος από την πώληση ενός ελαττωματικού προϊόντος είναι κατά πολύ μεγαλύτερο από το κόστος της πρόληψης και του ελέγχου κατά το στάδιο της παραγωγής.

Στην παρούσα διπλωματική εργασία αναπτύσσεται ένας αλγόριθμος προσομοίωσης δικτύων συναρμολόγησης που παράγουν ένα τελικό προϊόν. Ο αλγόριθμος αυτός έχει την δυνατότητα να περιγράφει τη λειτουργία στο σύστημα διαφόρων κέντρων επιθεώρησης, ελέγχου ποιότητας δηλαδή, αλλά και ενός κέντρου επιδιόρθωσης των ελαττωματικών κομματιών. Τα ελαττώματα αυτά δημιουργούνται στα επιμέρους στάδια της παραγωγικής διαδικασίας, σε κάποιες από τις κατεργασίες που γίνονται στις μηχανές του δικτύου.

Σκοπός είναι, προσομοιώνοντας ένα τέτοιο αναξιόπιστο δίκτυο, να προσδιορισθεί η βέλτιστη κατανομή των σημείων ελέγχου-επιθεώρησης, ώστε να μεγιστοποιηθεί το μέσο κέρδος λειτουργίας του συστήματος ανά μονάδα χρόνου. Για την διαμόρφωση του συνολικού κέρδους συμβάλλουν διάφορες επιμέρους παράμετροι κόστους ή κέρδους: κόστος λειτουργίας των σταθμών ελέγχου και του σταθμού επιδιόρθωσης, κόστος από πιθανές επιστροφές έτοιμων προϊόντων τα οποία όμως έχουν κάποιο ελαττωματικό χαρακτηριστικό, κόστος επιδιόρθωσης ή αντικατάστασης ελαττωματικών κομματιών που εντοπίστηκαν εντός του δικτύου κατά την διάρκεια της παραγωγής, κέρδος από τις πωλήσεις, κόστος καθυστέρησης παραγγελιών και κόστος αποθέματος.

1.2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΑΝΑΣΚΟΠΗΣΗ

Μοντέλα και αλγόριθμοι βελτιστοποίησης για προβλήματα κατανομής σταθμών ελέγχου ξεκίνησαν ήδη να μελετώνται από το 1965, από τους Lindsay και Bishop [1] οι οποίοι και πρότειναν ένα μοντέλο δυναμικού προγραμματισμού. Στις πρώτες μελέτες εξετάστηκαν γραμμές παραγωγής με λίγες μηχανές ενώ τα τελευταία χρόνια έχουν αναπτυχθεί και αλγόριθμοι για πιο πολύπλοκα δίκτυα, λόγω των υπολογιστικών δυνατοτήτων που προσφέρουν οι σύγχρονοι υπολογιστές.

Οι Cochran και Erol [2] παρουσίασαν ένα αναλυτικό μοντέλο για απλή γραμμή παραγωγής με σταθμούς ελέγχων και επιδιόρθωσης. Σε αυτό υπολογίζεται ο ρυθμός ελαττωματικών και αξιόπιστων προϊόντων, που παράγονται στο σύστημα, με βάση τις θέσεις των σημείων ελέγχου. Το μοντέλο εξετάζει την ποιότητα μόνο και δεν μπορεί να περιγράψει αποθέματα, ανικανοποίητη ζήτηση, το ύψος των πωλήσεων και συνεπώς δεν

μπορεί να υπολογίσει μέσο ρυθμό κερδοφορίας.

Στην εργασία [3] γίνεται μια προσπάθεια εντοπισμού βέλτιστων σημείων ελέγχου εντός δικτύου. Κάνοντας χρήση μιας ευρετικής (εμπειρικής) λύσης για το πρόβλημα εντοπίζονται τα βέλτιστα αυτά σημεία, ενώ δίνεται και μια εφαρμογή για μια γραμμή παραγωγής ηλεκτρονικών εξαρτημάτων, χωρίς να γίνεται κάποια βελτιστοποίηση κέρδους.

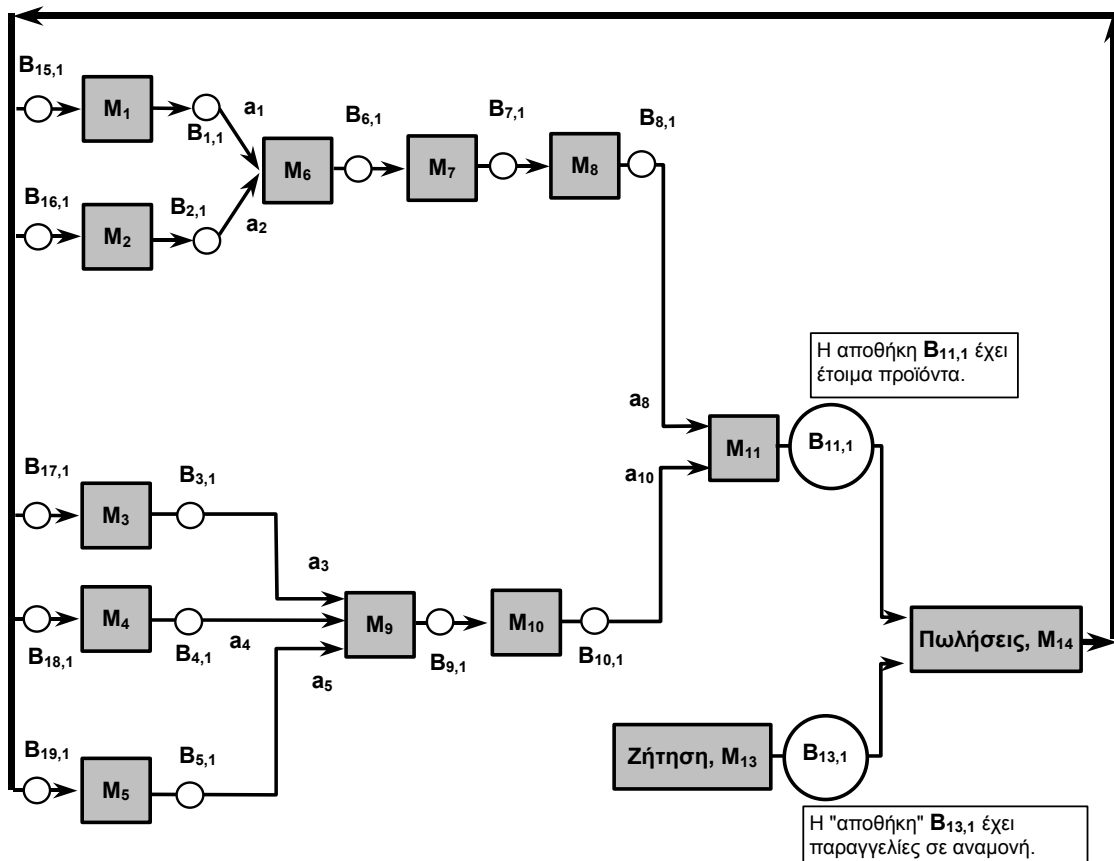
Το 2005 παρουσιάστηκε από τους Voslem, et al. [4], ένας γενετικός αλγόριθμος, που χρησιμοποιεί προσομοίωση διακριτών γεγονότων για την εκτίμηση του κέρδους λειτουργίας μιας γραμμής παραγωγής χωρίς συναρμολογήσεις. Σύμφωνα με αυτή την εργασία σε κάθε μηχανή μπορεί να γίνεται επιθεώρηση όμως δεν επιτρέπονται επιδιορθώσεις και όταν εντοπίζεται ελαττωματικό κομμάτι τότε απορρίπτεται.

Στην εργασία [5] οι Penn και Ravin παρουσιάζουν μια μελέτη για την εύρεση βέλτιστων σημείων ελέγχου σε γραμμή συναρμολόγησης όπου και εδώ στόχος είναι η βελτιστοποίηση του αναμενόμενου κέρδους που μπορεί να επιφέρει το σύστημα. Στο δίκτυο δεν γίνονται επιδιορθώσεις κομματιών. Το σύστημα παράγει κατά παραγγελία, δηλαδή δεν επιτρέπεται η διατήρηση αποθέματος ασφαλείας. Όταν φθάσει μια παραγγελία τότε μια πρώτη ύλη εισέρχεται στην αποθήκη που τροφοδοτεί την πρώτη μηχανή. Οι αφίξεις παραγγελιών ακολουθούν κατανομή Poisson, ήτοι οι χρόνοι μεταξύ διαδοχικών παραγγελιών έχουν εκθετική κατανομή.

Στην παρούσα διπλωματική εργασία εξετάζονται δίκτυα συναρμολόγησης με γενικές κατανομές χρόνων κατεργασίας, βλάβης και επισκευής και συνεκτιμώνται στη συνάρτηση κέρδους το κόστος αποθέματος και το κόστος καθυστερήσεων στην παράδοση προϊόντων.

2 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Το σύστημα είναι ένα δίκτυο συναρμολόγησης, που παράγει ένα τελικό προϊόν. Ένα παράδειγμα φαίνεται στο Σχ. 1, όπου οι κύκλοι συμβολίζουν αποθηκευτικούς χώρους και τα τετράγωνα μηχανές. Στο σύστημα διατηρείται συνολικό απόθεμα που αρκεί για την παραγωγή ενός σταθερού αριθμού τελικών προϊόντων. Αυτή η πολιτική αποθέματος θα περιγραφεί εκτενέστερα σε επόμενη παράγραφο.



Σχ. 1. Σύστημα συναρμολόγησης τύπου CONWIP: **1)** Κάθε μηχανή M_i τροφοδοτεί την αποθήκη $B_{i,1}$. **2)** Κάθε μία από τις μηχανές M_6 , M_9 και M_{11} , προκειμένου να ξεκινήσει τη συναρμολόγηση ενός κομματιού, χρειάζεται a_i εξαρτήματα από τις αποθήκες $B_{i,1}$ των μηχανών M_i που την τροφοδοτούν. **3)** Τη χρονική στιγμή 0, στο σύστημα υπάρχουν μόνο πρώτες ύλες. Αυτές αρκούν για να παραχθούν s προϊόντα. **4)** Μετά από κάθε πώληση, νέες πρώτες ύλες που αντιστοιχούν στο προϊόν που πωλήθηκε φορτώνονται στις αποθήκες $B_{15,1}, \dots, B_{19,1}$. **5)** Το συνολικό απόθεμα του συστήματος ανά πάσα χρονική στιγμή αρκεί ώστε να παραχθούν ακριβώς s τελικά προϊόντα, αν δεν γίνει άλλη τροφοδοσία σε πρώτες ύλες.

2.1 ΔΙΚΤΥΑ ΣΥΝΑΡΜΟΛΟΓΗΣΗΣ

Όπως φαίνεται και στο σχήμα, μια μηχανή σημειώνεται ως M_i , όπου i ο δείκτης της. Τα κομμάτια που παράγονται από τη μηχανή M_i μεταφέρονται στην έμπροσθεν αποθήκη της $B_{i,1}$. Υπάρχουν πρώτες μηχανές στο δίκτυο, M_1, M_2, M_3, M_4 και M_5 , οι οποίες επεξεργάζονται τις πρώτες ύλες. Μετά τις πρώτες κατεργασίες τα εξαρτήματα που παράγονται προχωρούν σταδιακά προς τις επόμενες μηχανές, οι οποίες είτε εκτελούν κάποια κατεργασία πάνω σε κομμάτι, είτε απλά συναρμολογούν κομμάτια, είτε κάνουν και τα δυο αυτά μαζί. Γι' αυτό το λόγο κάθε αποθήκη $B_{i,1}$ που τροφοδοτεί με εξαρτήματα μια μηχανή έχει κάποιο συντελεστή συναρμολόγησης a_i (*Assembly Factor*), που δίνει την πληροφορία για το πόσα κομμάτια από την $B_{i,1}$ απαιτούνται για την συναρμολόγηση. Για παράδειγμα στο Σχ. 1 η μηχανή M_6 χρειάζεται a_1 κομμάτια από την αποθήκη $B_{1,1}$ και a_2 από την αποθήκη $B_{2,1}$ ώστε να πραγματοποιήσει μια συναρμολόγηση και να μπει σε κατάσταση πλήρους λειτουργίας (κατάσταση 1). Αν ο αριθμός των εξαρτημάτων είναι μικρότερος του a_i , τότε η μηχανή δεν μπορεί να παράγει και παραμένει σε κατάσταση αποστέρησης (κατάσταση 0). Άρα μια μηχανή μπορεί να έχει δυο καταστάσεις, κατάσταση λειτουργίας 1 και κατάσταση αποστέρησης 0.

Επιπλέον κάθε μηχανή έχει έναν τυχαίο χρόνο παραγωγής, που ακολουθεί γνωστή κατανομή. Πρόκειται για το χρόνο που απαιτείται ώστε η μηχανή να εκτελέσει έναν πλήρη κύκλο, δηλαδή μια κατεργασία σε ένα κομμάτι.

Είναι πιθανό κάποιες από τις μηχανές να υποστούν μια ή περισσότερες βλάβες κατά την διάρκεια λειτουργίας της. Πράγματι, εξαιτίας διαφόρων λόγων όπως διακοπές ρεύματος, κακή συντήρηση, κακή χρήση αλλά ακόμα και λόγω ανάγκης αλλαγής κάποιων εξαρτημάτων ή λόγω προγραμματισμένης συντήρησης οι μηχανές σε μια παραγωγική μονάδα σταματούν να λειτουργούν. Όλα αυτά τα φαινόμενα οδηγούν σε διακοπή της λειτουργίας της και θα αναφέρονται ως βλάβη. Υποθέτουμε ότι οι πιθανότητες βλάβης των μηχανών είναι γνωστές. Επιπλέον εδώ εισάγεται και ένας χρόνος, τυχαίος πάλι, που αφορά την διάρκεια επισκευής μιας μηχανής στην περίπτωση που αυτή υποστεί βλάβη. Οι διάρκειες επισκευής των βλαβών έχουν γνωστές κατανομές. (Για τον προσδιορισμό των κατανομών των χρόνων κατεργασιών και επισκευών αλλά και των πιθανοτήτων βλάβης γίνεται δειγματοληψία και στατιστική ανάλυση αλλά αυτό δεν εξετάζεται στην παρούσα εργασία). Όταν μια μηχανή είναι υπό επισκευή τότε δεν εκτελεί καμία κατεργασία. Τα εξαρτήματα που υφίστανται κατεργασία σε αυτή την μηχανή περιμένουν έως ότου αυτή επισκευαστεί και είναι έτοιμη πάλι για να συνεχίσει την κατεργασία της.

Όπως αναφέρθηκε, μεταξύ των μηχανών μεσολαβούν αποθήκες. Κάθε μηχανή M_i τροφοδοτεί την αποθήκη $B_{i,1}$ με εξαρτήματα. Όταν δηλαδή μια μηχανή τερματίσει έναν κύκλο της, αποστέλλει το κομμάτι στην αποθήκη της. Εκεί το κομμάτι περιμένει έως ότου φορτωθεί στη μηχανή που ακολουθεί και εκείνη με την σειρά της ξεκινήσει ένα νέο δικό της κύκλο παραγωγής. Υποθέτουμε ότι το σύστημα λειτουργεί με τέτοιο τρόπο ώστε να αποφεύγονται υπερχειλίσεις στις αποθήκες. Έτσι ποτέ μια μηχανή δεν μπλοκάρει. Αυτό επιτυγχάνεται εφαρμόζοντας διάφορες πολιτικές. Η πλέον διαδεδομένη πολιτική ελέγχου του συνολικού αποθέματος στο σύστημα, είναι η CONWIP (*Constant work in process*).

2.2 ΠΟΛΙΤΙΚΗ CONWIP

Με την πολιτική CONWIP ελέγχεται συνεχώς το πλήθος των κομματιών που βρίσκονται εντός του δικτύου, των κομματιών δηλαδή που ακόμα είναι στο στάδιο της παραγωγής ή είναι στην αποθήκη έτοιμων προϊόντων διαθέσιμα για να πωληθούν όταν υπάρξει ζήτηση. Κάθε φορά που για οποιονδήποτε λόγο κομμάτι εγκαταλείπει το σύστημα, αυτό αναπληρώνεται άμεσα με πρώτες ύλες.

Έτσι, για παράδειγμα, όταν πωλείται ένα τελικό προϊόν τότε αμέσως εισάγονται πρώτες ύλες στις αρχικές αποθήκες. Σε κάθε μια από αυτές μπαίνουν τόσες πρώτες ύλες όσες απαιτούνται για την παραγωγή ενός τελικού έτοιμου προϊόντος. Με αυτό τον τρόπο μπορεί να κρατηθεί σταθερό το σύνολο των «εν δυνάμει» έτοιμων προϊόντων και κατά επέκταση να ελεγχθούν και οι χωρητικότητες των ενδιάμεσων αποθηκών έτσι ώστε να μην δημιουργηθούν προβλήματα στην παραγωγή. Επιπλέον με την πολιτική CONWIP διατηρείται πεπερασμένο το κόστος αποθέματος, το οποίο συχνά αποτελεί μια πολύ σημαντική συνιστώσα του κόστους παραγωγής.

Θα μπορούσε να ειπωθεί για να γίνει πιο κατανοητό ότι η τελική μηχανή του συστήματος κάνει μιας μορφής αποσυναρμολόγηση προς της πρώτες μηχανές πάλι με κάποιο συντελεστή αποσυναρμολόγησης, που είναι στην ουσία το πλήθος των πρώτων υλών για να παραχθεί ένα τελικό προϊόν. Διαμορφώνεται έτσι ένα κλειστό σύστημα ελέγχου των αποθεμάτων σε όλη την γραμμή παραγωγής.

2.3 ΜΗΧΑΝΕΣ ΖΗΤΗΣΗΣ ΚΑΙ ΠΩΛΗΣΗΣ

Όταν παράγεται τελικό προϊόν, μεταφέρεται στην αποθήκη έτοιμων προϊόντων όπου αναμένει πελάτη που θα το αγοράσει. Εισάγονται δηλαδή δυο έννοιες, η ζήτηση και η πώληση. Για να γίνει μια πώληση ενός τελικού προϊόντος πρέπει να υπάρξει ζήτηση, κάποιος να ενδιαφερθεί με άλλα λόγια να αγοράσει το προϊόν. Αυτά τα δυο γεγονότα προσομοιώνονται μέσω δυο μηχανών που είναι οι δύο από τις τρεις τελευταίες μηχανές του δικτύου.

Η διαδικασία της ζήτησης μπορεί να θεωρηθεί ως μια μηχανή που παράγει και αυτή με κάποια τυχαιότητα. Αυτό που παράγει δεν είναι τίποτα άλλο από πελάτες που έρχονται και είναι υποψήφιοι αγοραστές. Στο Σχ. 1, όταν η μηχανή της ζήτησης M_{12} ολοκληρώνει έναν κύκλο της, ένας πελάτης μπαίνει στο σύστημα και περιμένει έως ότου ικανοποιηθεί. Αυτό θα συμβεί όταν βρεθεί διαθέσιμο προϊόν στην αποθήκη τελικών προϊόντων. Όπως φαίνεται και στο σχήμα εκεί πραγματοποιείται μια ιδιότυπη συναρμολόγηση: (πελάτης + προϊόν = πώληση). Η αποθήκη τελικών προϊόντων ($B_{11,1}$ στο Σχ. 1) θα δώσει σε πελάτη την ποσότητα που θέλει να αγοράσει, αν βέβαια στην αποθήκη της μηχανής ζήτησης ($B_{13,1}$ στο Σχ. 1) υπάρχει πελάτης εν αναμονή. Εδώ πρέπει να τονισθεί ότι η αποθήκη της ζήτησης, και μόνο αυτής, έχει πεπερασμένη χωρητικότητα και αυτό είναι λογικό: Ένα σύστημα παραγωγής δεν μπορεί και δεν πρέπει να θέτει σε αναμονή όλους τους πελάτες που φθάνουν, όταν το ίδιο δεν έχει απόθεμα. Κάτι τέτοιο θα είχε κόστος δυσφήμισης και πιθανό κόστος (ρήτρες) καθυστέρησης. Πολλές φορές υπάρχει δυσφήμιση αν η αναμονή των πελατών είναι μεγάλη σε διάρκεια. Για αυτούς τους λόγους η χωρητικότητα της αποθήκης ζήτησης είναι καθορισμένη. Έτσι δεν γίνονται δεκτές παραγγελίες πελατών όταν ο αριθμός εκείνων που ήδη περιμένουν είναι

ίσος με κάποιο άνω όριο.

Από την άλλη πλευρά η μηχανή της πώλησης (M_{14} στο Σχ. 1) είναι αυτή που θα κανονίσει πότε θα ικανοποιηθεί μια παραγγελία ενός πελάτη. Αυτή ως μηχανή συναρμολόγησης για να τεθεί σε λειτουργία πρέπει να βρει το ανάλογο απόθεμα τόσο στην μηχανή ζήτησης όσο και στην τελευταία μηχανή M_{11} . Επιπροσθέτως είναι και αυτή που παίζει τον ρόλο του ρυθμιστή στην πολιτική CONWIP. Όταν ένα κομμάτι αναχωρεί από αυτή την μηχανή τότε αναπληρώνεται στο σύστημα με ανάλογη ποσότητα πρώτων υλών.

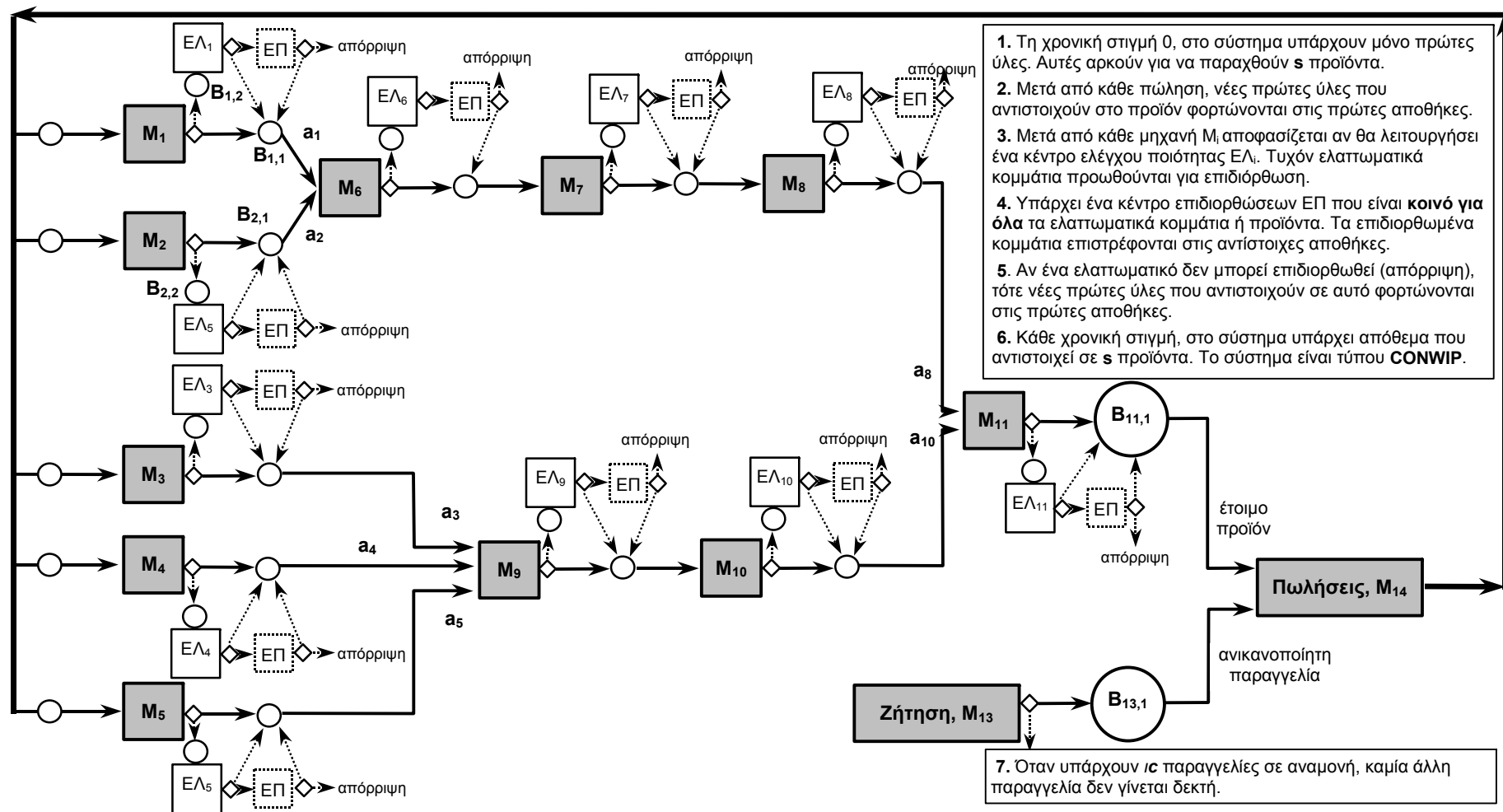
2.4 ΣΤΑΘΜΟΙ ΕΛΕΓΧΟΥ ΠΟΙΟΤΗΤΑΣ ΚΑΙ ΕΠΙΘΕΩΡΗΣΗΣ

Έως τώρα περιγράφηκε ένα απλό σύστημα συναρμολόγησης στο οποίο εφαρμόζεται μια πολιτική διαχείρισης αποθέματος CONWIP. Όμως όπως έχει προαναφερθεί στην παρούσα εργασία γίνεται και έλεγχος ποιότητας των κατεργασιών και συναρμολογήσεων των μηχανών αλλά και ενδεχόμενη επιδιόρθωση. Σε ένα δίκτυο μηχανών είναι προφανές ότι κάποιες από αυτές κάνουν κακές κατεργασίες προσδίδοντας κάποιο ελάττωμα στο κομμάτι που επεξεργάζονται. Αυτό συμβαίνει με μια τυχαιότητα. Για την περιγραφή της επιθεώρησης σε μια μηχανή M_i εισάγονται στο δίκτυο σταθμοί ελέγχου EL_i , όπως φαίνεται στο Σχ. 2 .

Βασικός σκοπός ενός σταθμού ελέγχου είναι να διερευνήσει αν κάποιες κατεργασίες και συναρμολογήσεις εντός του δικτύου έγιναν σωστά. Εδώ θα πρέπει να τονισθεί ότι σε ένα δίκτυο μπορεί να υπάρχουν πολλοί σταθμοί ελέγχου.

Ένα επιθεωρημένο κομμάτι, που διαπιστώνεται ότι είναι ελαττωματικό, από κάποιο σταθμό ελέγχου οδηγείται στον σταθμό επιδιόρθωσης. Αυτός είναι ένας κοινός σταθμός για όλο το δίκτυο. Εκεί θα διερευνηθεί περαιτέρω αν μπορεί να γίνει επιδιόρθωση στο κομμάτι. Αν ναι, τότε μεταφέρεται στην αποθήκη της μηχανής, στην οποία είχε υποστεί την τελευταία κατεργασία αλλιώς απορρίπτεται.

Το σύστημα επιβαρύνεται με κάποιο κόστος για κάθε σταθμό επιθεώρησης και για το σταθμό επιδιόρθωσης, εφ'όσον αποφασισθεί η λειτουργία τους. Το ζητούμενο είναι αν αυτό το κόστος είναι τόσο υψηλό που δεν συμφέρει να γίνει καθόλου έλεγχος ποιότητας ή, αν συμφέρει, σε ποια σημεία της παραγωγής θα πρέπει να τοποθετηθούν οι σταθμοί επιθεώρησης.



Σχ. 2. Ενδεικτικό σύστημα συναρμολόγησης τύπου CONWIP με έλεγχο ποιότητας, επιδιόρθωση, και μέγιστο επιτρεπτό αριθμό ανικανοποίητων παραγγελιών: **1).** Κομμάτι που εξέρχεται από την M_i , είτε πηγαίνει στην $B_{i,1}$ είτε, εφ' όσον λειτουργεί κέντρο ελέγχου EL_i , μεταφέρεται στο χώρο αποθήκευσης $B_{i,2}$ του EL_i . **2)** Αν τα ελεγχθέντα κομμάτια είναι αποδεκτά, μεταφέρονται στην $B_{i,1}$, αλλιώς μεταφέρονται στο κέντρο επιδιόρθωσης $ΕΠ$. **3)** Επιδιορθωμένα κομμάτια επιστρέφουν στην αποθήκη $B_{i,1}$. Τα μη επιδιορθώσιμα απορρίπτονται και αντίστοιχες πρώτες ύλες μεταφέρονται στις πρώτες μηχανές.

2.5 ΠΑΡΑΜΕΤΡΟΙ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

Σε αυτό το σημείο και για την καλύτερη περαιτέρω κατανόηση του συστήματος θα δοθούν κάποιες παράμετροι, που χρησιμοποιούνται στον αλγόριθμο για την περιγραφή της τοπολογίας του δικτύου και την περιγραφή της λειτουργίας όλων των στοιχείων του (μηχανές, αποθήκες, κέντρα ελέγχου, κέντρα επιδιόρθωσης) κάθε χρονική στιγμή. Για συντομία αποφεύγεται ο συμβολισμός του τρέχοντος χρόνου.

2.5.1 Λειτουργικές παράμετροι

Οι λειτουργικές παράμετροι του συστήματος αφορούν χρόνους και πιθανότητες εμφάνισης γεγονότων, τα οποία επηρεάζουν τη λειτουργία των μηχανών στο σύστημα.

Το πλήθος των μηχανών του δικτύου, που επεξεργάζονται κομμάτια έως την παραγωγή ενός τελικού έτοιμου προϊόντος, συμβολίζεται με M . Ως N θα συμβολίζεται το σύνολο των μηχανών:

$$M + \{\text{μηχανή επιδιόρθωσης}\} + \{\text{μηχανή ζήτησης}\} + \{\text{μηχανή πώλησης}\} = M + 3 = N$$

όπου $N = M + 3$: η μηχανή πώλησης

$N - 1 = M + 2$: η μηχανή ζήτησης

$N - 2 = M + 1$: η μηχανή επιδιόρθωσης

Θεωρούμε μια «μηχανή» $M_{i,j}$ όπου $j=1$ αν είναι πραγματική μηχανή και $j=2$ αν είναι σημείο επιθεώρησης EL_i . Οι χρόνοι κατεργασιών της $M_{i,j}$ μπορεί να είναι:

- i. Σταθεροί και ίσοι με $PRA(i,j)$
- ii. Εκθετικά κατανομημένοι με μέση τιμή $PRA(i,j)$
- iii. Ομοιόμορφοι στο διάστημα $[PRA(i,j), PRB(i,j)]$

Από την παραδοχή ότι οι βλάβες συμβαίνουν αν η μηχανή λειτουργεί, προκύπτει ότι ο χρόνος μεταξύ δυο διαδοχικών βλαβών εξαρτάται από το πλήθος των κομματιών που επεξεργάζεται. Για αυτό για την περιγραφή των βλαβών χρησιμοποιείται η μεταβλητή του πλήθους κομματιών που θα παραχθούν μέχρι την επόμενη βλάβη, $NPTF(i,j)$. Το πλήθος αυτό μπορεί να είναι:

- i. Άπειρο (καμία βλάβη)
- ii. Τυχαίο από γεωμετρική κατανομή με παράμετρο $FRA(i,j)$
- iii. Τυχαίο από ομοιόμορφη κατανομή στο διάστημα $[FRA(i,j), FRB(i,j)]$

Οι χρόνοι επισκευής, όπως και οι χρόνοι κατεργασιών, μπορεί να είναι:

- i. Σταθεροί και ίσοι με $REA(i,j)$
- ii. Εκθετικά κατανομημένοι με μέση τιμή $REA(i,j)$
- iii. Ομοιόμορφοι στο διάστημα $[REA(i,j), REB(i,j)]$

Όταν ένα εξάρτημα είναι ελαττωματικό τότε αυτό ή θα επιδιορθωθεί ή θα απορριφθεί. Επομένως οι μηχανές έχουν και κάποιες πιθανότητες να παράγουν

ελαττωματικό. Η πιθανότητα να παραχθεί επιδιορθώσιμο ελαττωματικό συμβολίζεται με $Q(i)$ και με $S(i)$ να παραχθεί απορριπτό ελαττωματικό.

Κάθε μηχανή πρέπει να έχει κάποιες πληροφορίες που την αφορούν και την χαρακτηρίζουν εντός του δικτύου, διαμορφώνοντας την τοπολογία του. Έτσι πρέπει κάθε μια να γνωρίζει την επόμενη μηχανή της, που συμβολίζεται ως $MD(i)$ (*machine down i*). Με βάση αυτή την πληροφορία μπορεί κανείς να γνωρίζει: πόσες είναι οι μηχανές που τροφοδοτούν την $M_{i,1}$, $NU(i)$ (*number up i*), αλλά και ποιες είναι αυτές $MU(i,k)$ (*machine upstream i,k*), όπου k είναι ο αύξων αριθμός της προηγούμενης μηχανής $k=1,2,\dots,NU(i)$ η οποία τροφοδοτεί την i .

Επιπλέον κάθε μηχανή έχει και έναν συντελεστή συναρμολόγησης $MA(i)$ (*machine assembly*). Επίσης μια μηχανή μπορεί να μπει σε δυο διαφορετικές καταστάσεις 1 ή 0 (*λειτουργεί ή αποστερείται*) που συμβολίζονται ως $MS(i)$ (*machine state*).

Μια σημαντική παράμετρος του συστήματος είναι και αυτή που καθορίζει την συνολική αποθεματική χωρητικότητά του εξαιτίας της πολιτικής CONWIP και συμβολίζεται ως $ICON$. Είναι στην ουσία το πλήθος των τελικών κομματιών, τα οποία μπορούν να κατασκευασθούν από το σύνολο των εξαρτημάτων και πρώτων υλών που βρίσκονται εντός όλων των αποθηκών του δικτύου.

Τέλος, λειτουργική παράμετρος του συστήματος είναι και η χωρητικότητα της αποθήκης ζήτησης, IC , που επίσης δίνεται από τον χρήστη. Αυτή περιγράφει τον μέγιστο επιτρεπτό αριθμό πελατών σε αναμονή. Όταν υπάρχουν IC ανικανοποίητες παραγγελίες στο σύστημα, τότε καμία άλλη δεν γίνεται δεκτή, έως ότου το πλήθος τους μειωθεί.

2.5.2 Παράμετροι κόστους

Πέραν των λειτουργικών παραμέτρων που εξετάστηκαν παραπάνω υπάρχουν και παράμετροι κόστους, οι οποίες θα διαμορφώσουν στο τέλος κατά πόσο μια πολιτική με μια συγκεκριμένη τοπολογία μηχανών και σταθμών επιθεώρησης είναι συμφέρουσα. Σε μια γραμμή παραγωγής υπάρχουν παράμετροι που συνδέονται με το κέρδος και άλλες που συνδέονται με το κόστος.

Στο κέρδος συμβάλλει κυρίως η τιμή πώλησης που διαμορφώνεται για το τελικό προϊόν. Αν από την τιμή πώλησης αφαιρεθούν κάποια κόστη, σχετιζόμενα με το λειτουργικό κόστος ανά προϊόν που υπάρχει κατά την παραγωγική διαδικασία ενός κομματιού, όπως κόστος πρώτων υλών, κόστος λειτουργίας μηχανών ανά προϊόν κ.α., διαμορφώνεται το τελικό καθαρό κέρδος πωλήσεως ανά μονάδα προϊόντος.

Σε μια παραγωγική διαδικασία, όπου λαμβάνει χώρα και έλεγχος ποιότητας ένα επιπλέον κέρδος είναι αυτό που μπορεί να προκύψει από την απόρριψη ελαττωματικών προϊόντων. Απορρίπτοντας ένα ελαττωματικό εξάρτημα στο σταθμό επιδιόρθωσης, προκύπτει ένα καθαρό κέρδος, συνεκτιμώντας το προφανές κόστος που υπάρχει. Το κέρδος αυτό σχετίζεται με την διάθεση ελαττωματικών κομματιών, που πιθανόν να υπάρχει. Αν δεν υπάρχει η απόρριψη έχει αρνητικό κέρδος.

Όμως πέραν των κερδών υπάρχουν και κόστη. Κάθε δίκτυο παραγωγής οποιασδήποτε μορφής επιβαρύνεται με κόστος αποθέματος ανά μονάδα χρόνου. Πρόκειται για το κόστος που οφείλεται στη δέσμευση κεφαλαίου εξαιτίας της ύπαρξης αποθεμάτων και το κόστος συντήρησης τους στις αποθήκες εντός του δικτύου κατά την παραγωγική διαδικασία. Στην προκειμένη περίπτωση της πολιτικής CONWIP το κόστος

αυτό υπολογίζεται εύκολα αφού πάντα εντός του δικτύου υπάρχουν εξαρτήματα και πρώτες ύλες τόσες ώστε να παραχθούν ICON τελικά προϊόντα.

Αν στο σύστημα επιτρέπεται να υπάρχει ανικανοποίητη ζήτηση είναι πιθανό να υπάρχει κόστος καθυστέρησης είτε άμεσο (ρήτρες καθυστέρησης) είτε έμμεσο (δυσφορία πελατών). Με αυτό τον τρόπο διαμορφώνεται ένα κόστος ανικανοποίητης ζήτησης ανά μονάδα χρόνου.

Το σύστημα επιβαρύνεται με κόστος επιστροφών από τις πωλήσεις ελαττωματικών προϊόντων. Αν για κάποιο λόγο δεν υπάρξουν σταθμοί επιθεώρησης στις τελευταίες μηχανές και κομμάτι ελαττωματικό φτάσει στα χέρια ενός πελάτη αυτός το επιστρέφει εξασφαλίζοντας κάποιου είδους χρηματική αποζημίωση. Ταυτόχρονα όμως και εδώ υπάρχει κάποιο κόστος δυσφήμισης και δυσαρέσκειας του πελάτη που παραλαμβάνει αναξιόπιστα προϊόντα. Συνολικά λοιπόν διαμορφώνεται και ένα κόστος από την διάθεση τέτοιων κομματιών στον πελάτη.

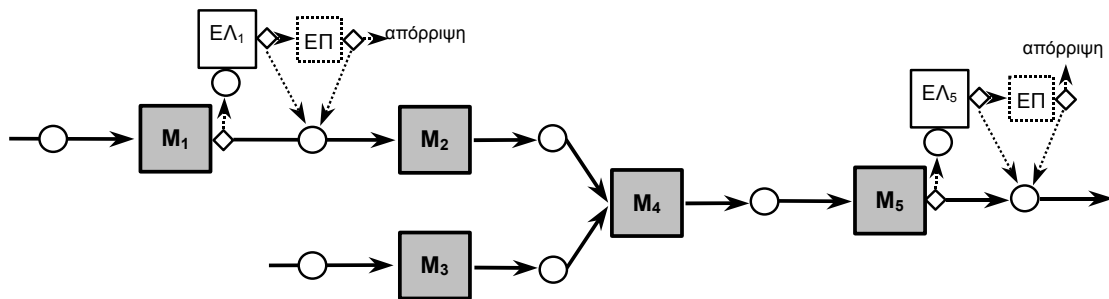
Όπως είναι λογικό οι σταθμοί επιθεώρησης καθώς και ο σταθμός επιδιόρθωσης έχουν κόστος λειτουργίας. Πέραν του πάγιου κόστους, που σχετίζεται με εξοπλισμό που πιθανά χρησιμοποιείται εκεί ή το πλήθος των ατόμων που απασχολούνται σε αυτούς, διαμορφώνεται και ένα μεταβαλλόμενο κόστος. Στους μεν σταθμούς ελέγχου καθορίζεται από το πλήθος των μηχανών που ελέγχονται ενώ στο σταθμό επιδιόρθωσης από τις επιδιορθώσεις. Επιπλέον λοιπόν υπάρχουν και δυο ακόμα κόστη, κόστος λειτουργίας σταθμού επιδιόρθωσης και κόστος λειτουργίας σταθμών επιθεώρησης.

Αθροίζοντας τα παραπάνω κέρδη και κόστη δίνεται η δυνατότητα στον χρήστη να έχει μια συγκριτική εικόνα μεταξύ των εναλλακτικών που του προσφέρονται και να αποφασίσει ποια πολιτική ταιριάζει καλύτερα στα ζητούμενά του καθώς και στις απαιτήσεις του.

3 ΑΞΙΟΛΟΓΗΣΗ ΕΝΑΛΛΑΚΤΙΚΩΝ ΛΥΣΕΩΝ

Βασικός σκοπός του αλγορίθμου που αναπτύσσεται είναι να προσδιορισθούν τα σημεία εντός του δικτύου όπου θα γίνεται έλεγχος ποιότητας. Τα σημεία μπορεί να βρίσκονται μεταξύ οποιονδήποτε διαδοχικών μηχανών, ελέγχοντας μια ή περισσότερες από αυτές αλλά και το πλήθος των κατεργασιών που έκαναν πάνω σε ένα εξάρτημα. Τελικός στόχος να υπάρξει μια τοπολογία συγκεκριμένη με μέγιστο ρυθμό κέρδους για το σύστημα.

Σε κάθε ένα από τα σημεία ελέγχου πραγματοποιείται επιθεώρηση σε ένα εξάρτημα. Ελέγχονται όλες οι κατεργασίες που έχουν γίνει πάνω σε αυτό και δεν έχουν έως εκείνη την στιγμή ελεγχθεί. Συνεπώς ελέγχονται όλες οι κατεργασίες από τους αμέσως προηγούμενους σταθμούς επιθεώρησης, ή από τις πρώτες μηχανές αν δεν μεσολαβούν άλλοι τέτοιοι σταθμοί, έως και τον σταθμό που εξετάζεται. Όπως φαίνεται και στο Σχ. 3, έστω ότι τα σημεία ελέγχου λειτουργούν μετά τις μηχανές M_1 και M_5 . Όταν επιθεωρούμε εξάρτημα που τελείωσε κατεργασία στην μηχανή M_5 , τότε θα ελεγχθεί το σύνολο των κατεργασιών των μηχανών M_2 , M_3 , M_4 και M_5 επειδή η κατεργασία από την M_1 έχει ήδη ελεγχθεί.



Σχ. 3. Το κέντρο ελέγχου EL_5 ελέγχει κάθε κομμάτι που εξέρχεται από την μηχανή M_5 για τις κατεργασίες που υπέστη στις μηχανές M_2 , M_3 , M_4 και M_5 , ενώ **δεν ελέγχει** την κατεργασία της M_1 γιατί εκείνη έχει ήδη ελεγχθεί από το κέντρο EL_1 .

Κάθε κατανομή σημείων ελέγχου περιγράφεται από έναν πίνακα μονοδιάστατο $ISE = [ISE_1 \ ISE_2 \ \dots \ ISE_M]$ όπου το στοιχείο ISE_i έχει τιμή 1 αν υπάρχει σημείο ελέγχου αμέσως μετά τη μηχανή M_{i-1} ή 0 αν δεν ακολουθεί μετά από αυτή σημείο επιθεώρησης. Για παράδειγμα, για το δίκτυο του Σχ. 3 που αποτελείται από 5 μηχανές έχουμε τον πίνακα:

$$ISE = [1 \ 0 \ 0 \ 0 \ 1]$$

επειδή υπάρχουν σημεία ελέγχου μόνο την πρώτη και τελευταία μηχανή.

Αναλυτικότερα ο αλγόριθμος παράγει όλους τους δυνατούς συνδυασμούς για 0 και 1 που μπορούν να υπάρξουν για όλες τις μηχανές $M_{i,1}$. Αρχικά δοκιμάζεται η περίπτωση να μην υπάρξει κανένα σημείο ελέγχου στο δίκτυο και το διάνυσμα είναι:

$$[0 \ 0 \ 0 \ \dots \ 0 \ 0 \ 0]$$

στη συνέχεια επιλέγεται ένας συνδυασμός σημείων ελέγχου. Αυτός προκύπτει προσθέτοντας στο δυαδικό αριθμό της προηγούμενης επανάληψης το 1. Για παράδειγμα η δεύτερη επανάληψη δίνει τον συνδυασμό $[0 \ 0 \ \dots \ 0 \ 1]$, η τρίτη τον $[0 \ 0 \ \dots \ 1 \ 0]$ κ.ο.κ. .

Κάθε φορά που ένας νέος συνδυασμός παράγεται, εκτελείται προσομοίωση του συστήματος βάζοντας σημεία ελέγχου μετά από τις μηχανές που ο αντίστοιχος συνδυασμός έχει δώσει. Η διαδικασία αυτή επαναλαμβάνεται έως ότου προσομοιωθεί και ο τελευταίος δυνατός συνδυασμός σημείων:

$$[1 \ 1 \ 1 \ \dots \ 1 \ 1 \ 1]$$

Τέλος, συγκρίνοντας τα κέρδη κάθε προσομοίωσης, προκύπτει ο βέλτιστος συνδυασμός σημείων επιθεώρησης που δίνουν το μέγιστο ρυθμό κέρδους.

Επισημαίνεται ότι δίνεται η δυνατότητα στην περίπτωση που επιθυμούνται σημεία επιθεώρησης σε συγκεκριμένα σημεία εντός του δικτύου, αυτά να ορισθούν εξ αρχής και να εξετασθούν μόνο οι συνδυασμοί που τα περιέχουν.

Στις επόμενες παραγράφους συνοψίζονται τα βήματα που ακολουθεί ο αλγόριθμος προσομοίωσης με τον οποίο εκτιμάται το μέσο κέρδος από τη λειτουργία του συστήματος για μία συγκεκριμένη κατανομή κέντρων ελέγχου.

3.1 ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ

Ο αλγόριθμος προσομοίωσης μεταχειρίζεται τους σταθμούς ελέγχου ως μηχανές. Πράγματι ο έλεγχος ενός κομματιού απαιτεί κάποιο χρόνο και μπορεί το κέντρο ελέγχου να υφίσταται βλάβες. Για την περιγραφή του συστήματος, συμβολίζουμε $M_{i,1}$ την μηχανή M_i και το αντίστοιχο κέντρο ελέγχου $ΕΛ_i$ συμβολίζεται $M_{i,2}$. Έτσι στο σύστημα υπάρχουν οι μηχανές $M_{i,1}$

$$M_{1,1}, M_{2,1}, \dots, M_{M,1}$$

αλλά και τα κέντρα ελέγχου

$$M_{1,2}, M_{2,2}, \dots, M_{M,2}.$$

Αν το στοιχείο ISE_i είναι 0, τότε **δεν λαμβάνεται υπ' όψιν** το κέντρο $M_{i,2}$.

Το κέντρο επιδιόρθωσης, εφ' όσον υπάρχει, συμβολίζεται $M_{M+1,1}$. Τέλος, υπάρχουν δύο ψευδομηχανές: η $M_{M+2,1}$ που προσομοιώνει την ζήτηση και η $M_{M+3,1}$ που περιγράφει την πώληση.

Κάθε μηχανή μπορεί να συναρμολογεί περισσότερα του ενός εξαρτήματα, τα οποία και προέρχονται από προγενέστερες μηχανές, προκειμένου να παράγει ένα κομμάτι. Συνεπώς ο συντελεστής συναρμολόγησης μπορεί να είναι και διάφορος του 1.

Ο χρόνος που θα λάβει χώρα το επόμενο γεγονός για κάθε μηχανή, οποιουδήποτε

τύπου, συμβολίζεται ως $EVT(i,1)$ αν αφορά κάποια από τις N μηχανές ή ως $EVT(i,2)$ αν πρόκειται για χρόνο γεγονότος σε σημείο ελέγχου. Είναι μια μεταβλητή που δίνει την σχέση μηχανής και χρόνου ενώ ταυτόχρονα μέσω αυτών μπορούμε να προσομοιώσουμε το σύστημα διακριτών γεγονότων.

Μετά από κάθε μηχανή υπάρχει μια αποθήκη, που αποστέλλονται τα εξαρτήματα που παράγονται από αυτή, και περιμένουν έως ότου φορτωθούν στην επόμενη μηχανή. Μόνο στην περίπτωση που μετά την μηχανή υπάρχει ένα σημείο ελέγχου, το κομμάτι πρέπει πρώτα να περιμένει να ελεγχθεί και εν συνεχεία με την σειρά του να μπει στην αποθήκη της μηχανής. Για αυτό το λόγο κάθε μηχανή διαθέτει δυο αποθήκες εμπρός της. Όταν έχουμε μετά από μια μηχανή $M_{i,1}$, ένα σημείο ελέγχου $M_{i,2}$, τότε κομμάτι που εξέρχεται αυτής, φορτώνεται στην αποθήκη $B(i,2)$ όπου και αναμένει έως ότου αυτό ελεγχθεί. Εν συνεχεία είτε επιδιορθώνεται και οδηγείται στην αποθήκη $B(i,1)$ είτε αν κριθεί αξιόπιστο από το σημείο ελέγχου οδηγείται εξαρχής στην αποθήκη $B(i,1)$.

3.2 ΒΑΣΙΚΗ ΔΟΜΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Τώρα θα εξεταστεί ο αλγόριθμος προσομοίωσης αναλυτικά και θα ορισθούν τα γεγονότα του συστήματος.

3.2.1 Χρόνοι επόμενων γεγονότων και πιθανότητες βλάβης

Ο αλγόριθμος παρατηρεί τους χρόνους $EVT(i, j)$ στους οποίους ένα κομμάτι εξέρχεται από τη μηχανή $M_{i,j}$ (που μπορεί να είναι απλή μηχανή, κέντρο επιθεώρησης, κέντρο επιδιόρθωσης) καθώς και τους χρόνους ζήτησης και πώλησης. Οι χρόνοι αυτοί μπορεί να είναι είτε σταθεροί είτε εκθετικοί είτε ομοιόμορφοι. Για την παραγωγή των τυχαίων χρόνων γίνεται χρήση κατάλληλων γεννητριών, αναλόγως την κατανομή που αυτοί ακολουθούν.

Έστω ότι οι χρόνοι κατεργασιών στην $M_{i,j}$ ακολουθούν εκθετική κατανομή με μέση τιμή $PRA(i,j)$. Αρχικά μια συνάρτηση-γεννήτρια παράγει έναν τυχαίο αριθμό U . Αν η κατεργασία ξεκινήσει τη στιγμή TN , τότε θα περατωθεί τη στιγμή

$$EVT(i, j) = TN - PRA(i, j) \ln U$$

όπου $-PRA(i, j) \ln U$ είναι η γεννήτρια εκθετικών τυχαίων μεταβλητών με μέσες τιμές $PRA(i, j)$ [6]. Αν η κατανομή του χρόνου κατεργασίας είναι ομοιόμορφη στο διάστημα $[PRA(i,j), PRB(i,j)]$ τότε η γεννήτρια χρόνου επόμενου γεγονότος είναι

$$EVT(i, j) = TN + PRA(i, j) + [PRB(i,j) - PRA(i,j)] U.$$

Ο αλγόριθμος στην τρέχουσα μορφή του έχει αυτές τις δύο γεννήτριες, επειδή οι εκθετική και η ομοιόμορφη κατανομή είναι οι πιο συνήθεις. Άλλες κατανομές μπορούν να συμπεριληφθούν με ελάχιστο προγραμματισμό.

Ό,τι ισχύει για τους χρόνους κατεργασιών ισχύει και για τους χρόνους επιδιόρθωσης των μηχανών, για τους χρόνους ζήτησης αλλά και πώλησης. Οι χρόνοι στους σταθμούς επιθεώρησης και επιδιόρθωσης βασίζονται επιπλέον και στο πλήθος των ελεγχόμενων κατεργασιών ή στο πλήθος των επιδιορθώσιμων κατεργασιών αντίστοιχα.

Αν το πλήθος $NPTF(i, j)$ κομματιών που θα παράγει η μηχανή $M_{i,j}$ έως ότου αυτή να υποστεί επόμενη βλάβη ακολουθεί γεωμετρική κατανομή με παράμετρο $FRA(i,j)$

(=πιθανότητα να συμβεί μία βλάβη κατά τη διάρκεια της κατεργασίας ενός κομματιού), η αντίστοιχη γεννήτρια είναι

$$NPTF(i, j) = INT\left(\frac{\ln U}{\ln(1 - FRA(i, j))}\right)$$

όπου $INT(x)$ είναι το ακέραιο μέρος του αριθμού x . Αν τα κομμάτια μέχρι τη βλάβη έχουν ομοιόμορφη κατανομή στο διάστημα $[FRA(i, j), FRB(i, j)]$ τότε η γεννήτριά τους είναι

$$NPTF(i, j) = FRA(i, j) + [FRB(i, j) - FRA(i, j) + 1] U.$$

Και πάλι, το μοντέλο μπορεί να χρησιμοποιήσει χρόνους που ακολουθούν οποιοδήποτε τύπου κατανομή με ελάχιστες τροποποιήσεις.

3.2.2 Τρόπος λειτουργίας του αλγορίθμου

Ο αλγόριθμος προσομοίωσης που αναπτύσσεται στην εργασία αυτή βασίζεται στην σύγκριση των χρόνων επόμενων γεγονότων που πρόκειται να συμβούν στις μηχανές του συστήματος. Το επόμενο γεγονός του συστήματος είναι εκείνο που θα συμβεί στον συντομότερο χρόνο.

Στην αρχή υπάρχει αρχικό απόθεμα μόνο για τις πρώτες μηχανές του δικτύου, λόγω της πολιτικής CONWIP. Σε αυτές υπάρχουν τόσες πρώτες ύλες ώστε να παραχθούν ICON τελικά προϊόντα. Συνεπώς οι μηχανές αυτές είναι σε κατάσταση λειτουργίας 1 και γι' αυτές υπολογίζονται οι χρόνοι επόμενων γεγονότων $EVT(i, j)$. Ο χρόνος επόμενου γεγονότος του συστήματος υπολογίζεται από τον τύπο

$$TN = \min_{\substack{i=1, \dots, M+3 \\ j=1, 2}} EVT(i, j)$$

Εν συνεχεία, η μηχανή στην οποία θα συμβεί το επόμενο γεγονός τροφοδοτεί με εξαρτήματα την επόμενη μηχανή θέτοντας και αυτήν σε κατάσταση λειτουργίας. Με αυτό τον τρόπο το δίκτυο αρχίζει να παράγει.

Κατά την διάρκεια της προσομοίωσης καταγράφεται η εξέλιξη του συστήματος μέσω κάποιων μεταβλητών. Οι αυξομειώσεις που συντελούνται στα αποθέματα των αποθηκών, οι πελάτες που έρχονται στο σύστημα, καθώς και αυτοί που αποχωρούν ως ανικανοποίητοι, καθώς και οι πωλήσεις που λαμβάνουν χώρα είναι οι πιο σημαντικές μεταβλητές του συστήματος. Επιπροσθέτως μελετώνται και οι επιστροφές προϊόντων από πελάτες λόγω ελαττωμάτων που υπάρχουν σε αυτά, αλλά και οι απορρίψεις και επιδιορθώσεις ελαττωματικών κομματιών, που έγιναν εντός της παραγωγικής διαδικασίας.

Όλα αυτά θα συνδράμουν έτσι ώστε να υπολογισθεί το μέσο κέρδος του συστήματος ανά μονάδα χρόνου για μια συγκεκριμένη τοπολογία σημείων ελέγχου. Αυτό θα μπορέσει εν συνεχεία να συγκριθεί με τα αντίστοιχα άλλων τοπολογιών με σκοπό την επιλογή της βέλτιστης πολιτικής.

3.3 ΓΕΓΟΝΟΤΑ ΣΥΣΤΗΜΑΤΟΣ

Μηχανή $M_{i,1}$ παράγει ένα κομμάτι

Όταν μηχανή $M_{i,1}$ παράγει ένα κομμάτι μια αλληλουχία επιμέρους γεγονότων λαμβάνουν χώρα στο δίκτυο.

Αρχικά μειώνεται το πλήθος των εξαρτημάτων που έχει η $M_{i,1}$ να κατεργασθεί έως ότου υποστεί την επόμενη βλάβη της.

Το παραγόμενο εξάρτημα οδηγείται στην αντίστοιχη αποθήκη που τροφοδοτεί την επόμενη μηχανή της. Στην περίπτωση που η επόμενη μηχανή της $M_{i,1}$ είναι σημείο ελέγχου ($M_{i,2}$) οδηγείται στην αποθήκη $B(i,2)$ όπου και αναμένει να επιθεωρηθεί. Διαφορετικά πηγαίνει στην αποθήκη $B(i,1)$ η οποία και τροφοδοτεί την $MD(i)$.

Η επόμενη μηχανή, σημείο ελέγχου ή απλή μηχανή, στην περίπτωση που ήταν σε κατάσταση λειτουργίας 0, δηλαδή ήταν αδρανής, ενεργοποιείται αν αυτό κριθεί εφικτό. Αν πρόκειται για σημείο ελέγχου η ενεργοποίηση γίνεται άμεσα. Σε διαφορετική περίπτωση εξετάζεται πρώτα αν όλες οι προηγούμενες μηχανές της $MD(i)$ έχουν το απαιτούμενο πλήθος εξαρτημάτων MA έτσι ώστε να την ενεργοποιήσουν. Μόνο τότε μπαίνει σε κατάσταση λειτουργίας 1 και ξεκινά νέο κύκλο κατεργασίας.

Η μηχανή $M_{i,1}$ τώρα έχει μείνει ελεύθερη, χωρίς εξάρτημα υπό κατεργασία. Ελέγχονται τώρα όλες οι προηγούμενες μηχανές της, $MU(i,k)$, πλήθους k . Αν στις αποθήκες τους που τροφοδοτούν την $M_{i,1}$ υπάρχει αρκετό απόθεμα τότε την ενεργοποιούν και αυτή ξεκινά ένα νέο κύκλο κατεργασίας. Στην περίπτωση που δεν υπάρχουν εξαρτήματα τότε αυτή μένει αδρανής, κατάσταση 0, και περιμένει έως ότου κάποια από τις προηγούμενες μηχανές της, την ενεργοποιήσουν.

Τέλος σε αυτό το σημείο θα πρέπει να τονισθεί ότι υπάρχει μια ειδική περίπτωση, αν η μηχανή $M_{i,1}$ είναι η τελική μηχανή $M_{M,1}$ και αυτή δεν έχει εμπρός της σημείο ελέγχου, παρά μόνο την πώληση. Τότε, επιπλέον των άλλων, γίνεται και μια εικονική επιθεώρηση του κομματιού που εξέρχεται αυτής. Με τον όρο εικονική επιθεώρηση νοείται ένας έλεγχος ποιότητας πάνω σε ένα εξάρτημα ή προϊόν χωρίς αυτό να προωθηθεί στον σταθμό επιδιόρθωσης αν αυτό βρεθεί να έχει ελαττώματα. Σκοπός αυτής της εικονικής επιθεώρησης είναι να είναι γνωστό εκ των προτέρων το πλήθος των ελαττωματικών προϊόντων που αποστέλλονται για πώληση και με αυτόν τον τρόπο να υπολογίζονται οι επιστροφές που γίνονται από τους πελάτες καθώς και το κόστος τους.

Σημείο ελέγχου $M_{i,2}$ ολοκληρώνει την επιθεώρηση

Ένα άλλο γεγονός του συστήματος είναι όταν σημείο ελέγχου $M_{i,2}$ ολοκληρώνει επιθεώρηση σε εξάρτημα που προήλθε από την μηχανή $M_{i,1}$. Όταν ένα εξάρτημα ελέγχεται, δυο ενδεχόμενα μπορούν να συμβούν: Ή το εξάρτημα να βρεθεί ελαττωματικό ή να βρεθεί αξιόπιστο. Ένα εξάρτημα ελέγχεται για όλο το πλήθος των ανέλεγκτων κατεργασιών που έχει υποστεί έως την στιγμή του ελέγχου. Δηλαδή αυτό που ελέγχεται είναι για κάθε μια ανέλεγκτη μηχανή το σύνολο των κατεργασιών που έχει κάνει πάνω στο εξάρτημα. Η προσομοίωση του ελέγχου γίνεται ως εξής:

1. Βρίσκονται ποιες κατεργασίες πρέπει να ελεγχθούν με βάση τη θέση του τρέχοντος σημείου ελέγχου i και των ελέγχων που έχουν προηγηθεί. Προφανώς μία από τις κατεργασίες θα πρέπει να είναι και η τελευταία (i).

2. Για κάθε κατεργασία k που πρέπει να ελεγχθεί υπολογίζεται το πλήθος εξαρτημάτων που έχουν χρησιμοποιηθεί για την παραγωγή του κομματιού που ελέγχεται (το κομμάτι που έφυγε από την $M(i, 1)$ έχει πολλά εξαρτήματα). Αυτός ο υπολογισμός γίνεται χρησιμοποιώντας τις πληροφορίες για τις διασυνδέσεις του δικτύου πριν από τη μηχανή $M(i, 1)$ και με βάση τους δείκτες συναρμολόγησης $MA(k)$ όλων των μηχανών πριν από την $M(i, 1)$.

3. Χρησιμοποιείται η γεννήτρια τυχαίων αριθμών, προκειμένου να ευρεθεί αν κάποια από τις κατεργασίες στις προηγούμενες μηχανές $M(k, i)$ έχει ελάττωμα (επιδιορθώσιμο ή μη επιδιορθώσιμο), ή δεν έχει ελάττωμα:

α) Έστω ότι για το κομμάτι που επιθεωρείται έχουν χρησιμοποιηθεί J_k εξαρτήματα της μηχανής $M(k, 1)$. Στην αρχή, το πλήθος ελαττωμάτων που έχουν βρεθεί είναι 0.

β) Για κάθε εξάρτημα γίνονται τα ακόλουθα:

i) Παράγεται τυχαίος αριθμός U ομοιόμορφος στο διάστημα $(0, 1)$.

ii) Αν

$$U < Q(k) + S(k)$$

τότε υπάρχει ελάττωμα στην κατεργασία του εξαρτήματος και αυξάνεται το πλήθος ελαττωμάτων αυτής της κατεργασίας κατά 1.

γ) Επαναλαμβάνονται τα βήματα α) και β) για όλα τα είδη εξαρτημάτων k που δεν έχουν ελεγχθεί ως τώρα.

δ) Αν δεν βρεθούν ελαττωματικές κατεργασίες, τότε το κομμάτι είναι αξιόπιστο αλλιώς θεωρείται ελαττωματικό.

Στην περίπτωση που το κομμάτι μετά τον έλεγχο βρεθεί ελαττωματικό αποστέλλεται στο κέντρο επιδιόρθωσης μαζί με μία λίστα που περιλαμβάνει το πλήθος των και το είδος των κατεργασιών που έχουν ελαττώματα. Αν όμως βρεθεί αξιόπιστο τότε φορτώνεται στην αποθήκη μετά την $M_{i,1}$ μηχανή, $B(i,1)$, και συνεχίζει την πορεία του εντός του δικτύου παραγωγής. Τότε εξετάζεται αν η επόμενη μηχανή της $M_{i,1}$, $MD(i)$, είναι σε κατάσταση λειτουργίας 0. Αν ναι εξετάζονται όλες οι προηγούμενες αποθήκες της, και διερευνάται αν αυτές είναι ικανές με την προσθήκη και του επιπλέον αυτού εξαρτήματος να την ενεργοποιήσουν. Διαφορετικά παραμένει αδρανής και περιμένει και άλλα εξαρτήματα για να ενεργοποιηθεί.

Γεγονός στο κέντρο επιδιόρθωσης $i=M+1$

Στο κέντρο επιδιόρθωσης ένα εξάρτημα θα εξετασθεί περαιτέρω αν είναι ελαττωματικό και επιδιορθώσιμο ή ελαττωματικό και απορριπτέο. Η διαδικασία είναι η ακόλουθη:

1. Το κέντρο ελέγχου έχει ήδη προσδιορίσει ποιες κατεργασίες έχουν ελαττώματα και για κάθε τέτοια κατεργασία k είναι γνωστό το πλήθος εξαρτημάτων που έχουν εντοπισθεί ελαττώματα.

2. Χρησιμοποιείται η γεννήτρια τυχαίων αριθμών, προκειμένου να ευρεθεί αν κάποιο εξάρτημα είναι μη επιδιορθώσιμο.

α) Έστω ότι για το κομμάτι που επιθεωρείται έχουν εντοπισθεί JB_k ελαττωματικά εξαρτήματα της μηχανής $M(k, 1)$.

β) Για κάθε εξάρτημα γίνονται τα ακόλουθα:

- i) Παράγεται τυχαίος αριθμός U ομοιόμορφος στο διάστημα $(0, 1)$.
- ii) Αν

$$U < S(k)/[Q(k) + S(k)]$$

τότε υπάρχει μη επιδιορθώσιμο ελάττωμα και ολόκληρο το κομμάτι απορρίπτεται.

γ) Επαναλαμβάνονται τα βήματα α) και β) για όλα τα είδη εξαρτημάτων k στα οποία έχουν εντοπισθεί ελαττώματα από το κέντρο ελέγχου.

δ) Αν όλες οι ελαττωματικές κατεργασίες είναι επιδιορθώσιμες τότε ολόκληρο το κομμάτι θεωρείται επιδιορθώσιμο και υπολογίζεται ένας τυχαίος χρόνος επιδιόρθωσης από κατάλληλη γεννήτρια τυχαίων αριθμών.

Αν βρεθεί επιδιορθώσιμο τότε προγραμματίζεται ο χρόνος ολοκλήρωσης της επιδιόρθωσης στον οποίο το κομμάτι θα επιστρέψει στην αποθήκη έμπροσθεν του σημείου ελέγχου από το οποίο και προήλθε. Στο ενδεχόμενο που η μηχανή που τροφοδοτείται από αυτή την αποθήκη είναι ανενεργή (κατάσταση λειτουργίας 0) τότε άμεσα ελέγχεται μήπως μπορεί να ενεργοποιηθεί και να βρεθεί σε κατάσταση πλήρους λειτουργίας 1. Αλλιώς παραμένει ανενεργή.

Στην περίπτωση που το εξάρτημα πρέπει να απορριφθεί, αυξάνεται το πλήθος των κομματιών που απορρίφθηκαν από την παραγωγική διαδικασία. Επιπροσθέτως το κενό που δημιουργείται στο συνολικό απόθεμα του συστήματος αναπληρώνεται άμεσα για να κρατηθεί σταθερό, και να εφαρμόζεται σωστά η πολιτική CONWIP. Φορτώνονται επομένως πρώτες ύλες στις αποθήκες των πρώτων μηχανών, μόνο αυτών όμως που έχουν κάνει κατεργασίες πάνω στο απορριπτέο εξάρτημα. Στην πράξη, για να γίνει αντιληπτό, εισάγονται τόσες πρώτες ύλες στο σύστημα όσες χρειάστηκαν για την κατασκευή του εξαρτήματος έως το σημείο απόρριψής του.

Γεγονός στην μηχανή της ζήτησης, $i=M+2$

Την στιγμή που η μηχανή της ζήτησης «παράγει» έναν πελάτη, ελέγχεται αν η αποθήκη της είναι γεμάτη. Αν υπάρχουν ήδη στην αναμονή IC σε πλήθος πελάτες, τότε ο πελάτης αποχωρεί αυξάνοντας το πλήθος των πελατών που εγκαταλείπουν το σύστημα.

Σε διαφορετική περίπτωση φορτώνεται στην αποθήκη μετά την μηχανή της ζήτησης και περιμένει να εξυπηρετηθεί, να γίνει δηλαδή μια πώληση. Για αυτό το λόγο κάθε φορά που γίνεται άφιξη πελάτη στο σύστημα ελέγχεται αν η μηχανή της πώλησης είναι αδρανής. Αν ναι και ταυτόχρονα υπάρχουν και έτοιμα προϊόντα στην αποθήκη της M μηχανής (αποθήκη έτοιμων προϊόντων), τότε ενεργοποιείται και ξεκινά ένα δικό της κύκλο κατεργασίας – πώλησης.

Γεγονός στην μηχανή της πώλησης, $I=M+3$

Η μηχανή της πώλησης θεωρείται ως τελευταία μηχανή του συστήματος. Εκτέλεση γεγονότος σε αυτή, συνεπάγεται άμεσα την φόρτωση πρώτων υλών στις αποθήκες των αρχικών μηχανών της γραμμής παραγωγής. Αυτό επιβάλλεται για να διατηρηθεί το συνολικό απόθεμα του συστήματος σταθερό και να εφαρμόζεται η πολιτική CONWIP. Βάζοντας πρώτες ύλες στις πρώτες μηχανές θα πρέπει να διερευνηθεί και αν κάποια από αυτές ήταν σε κατάσταση λειτουργίας 0 και συνεπώς τώρα θα πρέπει να ενεργοποιηθεί.

Τέλος ελέγχεται αν οι προηγούμενες μηχανές της πώλησης, η ζήτηση και η τελική μηχανή της παραγωγικής διαδικασίας M , έχουν αποθέματα έτσι ώστε να ξεκινήσει ένα νέο κύκλο πώλησης. Αν κάποια από τις δυο αποθήκες είναι άδεια τότε αδρανοποιείται και περιμένει έως ότου ενεργοποιηθεί.

Μέσο πλήθος ανικανοποίητων παραγγελιών

Για το κόστος ανικανοποίητης ζήτησης απαιτείται ο υπολογισμός του μέσου αποθέματος A στην αποθήκη της ζήτησης. Αυτό προκύπτει ως εξής [6]. Στην αρχή θέτουμε $A = 0$. Κάθε φορά που κάποια παραγγελία γίνεται δεκτή, αφαιρείται ο τρέχων χρόνος (TN) και προστίθεται αν έχουμε γεγονός πώλησης. Άρα, σε κάθε άφιξη παραγγελίας

$$A = A - TN$$

ενώ σε κάθε πώληση:

$$A = A + TN .$$

Στο τέλος της προσομοίωσης, το μέσο πλήθος πελατών σε αναμονή είναι

$$A = \frac{A + L(M+2,1) * TSIM}{TSIM},$$

όπου $L(M+2,1)$ είναι το πλήθος των πελατών σε αναμονή τη στιγμή που περατώνεται η προσομοίωση. Ο αριθμητής του κλάσματος ισούται με το εμβαδόν του αποθέματος [6], οπότε το κλάσμα ισούται με το μέσο απόθεμα.

4 ΑΡΙΘΜΗΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Αφού αναλύθηκε ο τρόπος λειτουργίας του αλγορίθμου προσομοίωσης, στη συνέχεια διερευνώνται κάποια από τα αποτελέσματα που έδωσε, υπό διαφορετικές συνθήκες, για ένα δεδομένο δίκτυο 11 μηχανών και τριών ψευδομηχανών, όπως φαίνεται στο Σχ. 2. Επισημαίνεται ότι στα πειράματα που έγιναν δεν τέθηκαν περιορισμοί ως προς τις θέσεις των σημείων ελέγχου, εξετάζοντας με αυτό τον τρόπο όλες τις δυνατές περιπτώσεις.

Το δίκτυο του Σχ. 2 επελέγη ως βάση για αριθμητικά πειράματα διότι συνδυάζει ένα πλήθος συναρμολογήσεων χωρίς να είναι εξαιρετικά πολύπλοκο. Έχει για παράδειγμα 3 μηχανές συναρμολόγησης ενώ ειδικά στην M_9 συναρμολογούνται τρία διαφορετικά εξαρτήματα.

Στην επόμενη παράγραφο παρουσιάζονται δεδομένα για τους χρόνους κατεργασιών, τις πιθανότητες βλάβης, τους χρόνους επισκευής, και τις πιθανότητες ελαττωματικών κατεργασιών.

Η βασική παραδοχή για τους χρόνους κατεργασιών είναι η εξής: Υιοθετείται ένας βασικός κύκλος εκτέλεσης μίας τελικής κατεργασίας στην M_{11} που διαρκεί 0.2 μονάδες χρόνου. Αν το τελικό προϊόν χρειάζεται J_k εξαρτήματα που παράγονται από κάποια μηχανή M_k , τότε ο μέσος χρόνος παραγωγής της M_k τίθεται ίσος με περίπου $0.2/J_k$ έτσι ώστε σε 0.2 μονάδες χρόνου να προλαβαίνει η M_k να παράγει όσα εξαρτήματα χρειάζονται για ένα τελικό προϊόν. Τέτοια συστήματα συναντώνται συχνά στην πράξη και ονομάζονται ισορροπημένα (balanced). Αν η M_k είχε μεγαλύτερους χρόνους παραγωγής τότε θα αποτελούσε μία "κρίσιμη μηχανή" για το σύστημα. Στα συνήθη συστήματα παραγωγής υπάρχει η τάση να αποφεύγονται οι κρίσιμες μηχανές. Παρόλα αυτά σε αρκετά συστήματα παραγωγής κανείς μπορεί να συναντήσει τέτοιες.

Για τις βλάβες, επελέγη μία πιθανότητα βλάβης της τάξης του 0.005 ή ισοδύναμα, μία βλάβη κάθε 200 κομμάτια που παράγονται. Προφανώς, υπάρχουν συστήματα με συχνότερες ή λιγότερο συχνές βλάβες.

Ο αλγόριθμος μπορεί να δεχθεί οιοσδήποτε τιμές για όλες τις παραμέτρους.

4.1 ΔΕΔΟΜΕΝΑ

Ο χρόνος προσομοίωσης TSIM μπορεί να διαφέρει κατά περίπτωση. Για να είναι αξιόπιστα τα αποτελέσματα της προσομοίωσης επιλέγεται χρόνος

$$TSIM = 20000 \text{ μονάδες χρόνου}$$

Για την επιλογή του εξετάστηκαν διάφοροι χρόνοι $TSIM = 1000, 10000, 20000$ και βρέθηκε ότι ο τελευταίος δίνει πάντα την ίδια βέλτιστη κατανομή όταν προσομοιώνεται το σύστημα με διαφορετικούς τυχαίους αριθμούς.

Το συνολικό απόθεμα του συστήματος (σύμφωνα με την πολιτική CONWIP) λαμβάνεται ως $ICON=4$ ενώ η αποθήκη της ζήτησης θεωρείται ότι έχει χωρητικότητα 4.

Τα δεδομένα για τις (11+3) μηχανές που χρησιμοποιούνται, ενδεικτικά, για την εκτέλεση των προσομοιώσεων παρουσιάζονται στον Πίνακα 1. Αυτά προέκυψαν εμπειρικά. Για παράδειγμα ο μέσος χρόνος κατεργασιών PRA(i,1) παίρνει τιμές τέτοιες ώστε στο σύστημα να μην παρατηρούνται μεγάλες καθυστερήσεις στις μηχανές.

Πίνακας 1. Παράμετροι του Συστήματος

M_i	MD(i)	MA(i)	Q(i)	S(i)	PRA(i,1)	PRA(i,2)	FRA(i,1)	FRA(i,2)	REA(i,1)	REA(i,2)
1	6	2	0.015	0.001	0.1	0.005	0.005	0	0.005	-
2	6	1	0.015	0.001	0.2	0.005	0.005	0	0.005	-
3	9	3	0.015	0.001	0.03	0.005	0.005	0	0.005	-
4	9	2	0.015	0.001	0.05	0.005	0.005	0	0.005	-
5	9	1	0.015	0.001	0.1	0.005	0.005	0	0.005	-
6	7	1	0.015	0.001	0.2	0.005	0.005	0	0.005	-
7	8	1	0.015	0.001	0.2	0.005	0.005	0	0.005	-
8	11	1	0.015	0.001	0.12	0.005	0.005	0	0.005	-
9	10	1	0.015	0.001	0.1	0.005	0.005	0	0.005	-
10	11	2	0.015	0.001	0.06	0.005	0.005	0	0.005	-
11	0	1	0.015	0.001	0.2	0.005	0.005	0	0.005	-
12	0	1	0	0	0,005	0	0	0	0	-
13	0	1	0	0	0.2	0	0	0	0	-
14	0	1	0	0	0.2	0	0	0	0	-

Στην πρώτη στήλη δίνονται οι μηχανές ενώ στην δεύτερη οι επόμενές τους. Επισημαίνεται ότι οι μηχανές επιδιόρθωσης (12), ζήτησης (13) και πώλησης (14) δεν έχουν επόμενη μηχανή, γιατί ο αλγόριθμος έχει την δυνατότητα να τις αναγνωρίσει και να τις ορίσει. Ακολουθούν οι συντελεστές συναρμολόγησης, οι πιθανότητες επιδιορθώσιμης ελαττωματικής κατεργασίας και οι πιθανότητες ελαττωματικής κατεργασίας που οδηγούν σε απόρριψη του εξαρτήματος.

Εν συνεχεία ακολουθούν 6 στήλες με χρόνους κατεργασιών, πιθανότητες βλάβης και χρόνους επισκευών τόσο για τις μηχανές όσο και για τα υποψήφια σημεία ελέγχου. Παρατηρείται ότι για τα σημεία ελέγχου οι χρόνοι επιθεώρησης είναι μικροί ενώ για να αποφευχθεί εμφάνιση βλάβης σε αυτά έχουν ορισθεί μηδενικές πιθανότητες βλάβης. Βέβαια δίνεται η δυνατότητα να ορισθούν τιμές στην περίπτωση που κάτι τέτοιο είναι επιθυμητό.

Τέλος στον Πίνακα 2 δίνονται κάποια ενδεικτικά κόστη, τα οποία και θα διαμορφώσουν την πολιτική με τον βέλτιστο ρυθμό κέρδους. Τόσο τα κέρδη όσο και τα κόστη επειδή βελτιστοποιείται ρυθμό κέρδους, στο τέλος κάθε προσομοίωσης διαιρούνται με τον χρόνο TSIM.

Το συνολικό κέρδος απόρριψης σε κάθε προσομοίωση συνυπολογίζεται, αθροίζοντας το πλήθος των εξαρτημάτων που απορρίφθηκαν κατά την παραγωγή.

Το κόστος αποθέματος προκύπτει εύκολα, αφού λόγω της πολιτικής CONWIP στο σύστημα υπάρχουν ICON εν δυνάμει έτοιμα προϊόντα.

Για το κόστος ελέγχου πέραν του πάγιου υπάρχει και το τρέχον, μεταβλητό, κόστος το οποίο διαμορφώνεται βάση του πλήθους των κατεργασιών που ελέγχει το εκάστοτε σημείο ελέγχου.

Πίνακας 2. Ενδεικτικές τιμές κέρδους και κόστους

Καθαρό κέρδος πωλήσεων ανά προϊόν	1000
Καθαρό κέρδος απόρριψης ανά ελαττωματικό εξάρτημα	10
Κόστος αποθέματος ανά μονάδα χρόνου και ανά μονάδα προϊόντος	20
Κόστος ανικανοποίητης ζήτησης ανά μονάδα χρόνου και ανά μέσο απόθεμα	20
Κόστος πώλησης ελαττωματικού προϊόντος	2000
Πάγιο κόστος ανά μονάδα χρόνου για κάθε σημείο ελέγχου	20
Μέσο κόστος ελέγχου μίας κατεργασίας για κάθε σημείο ελέγχου	10
Πάγιο κόστος ανά μονάδα χρόνου λειτουργίας κέντρου επιδιόρθωσης	100
Κόστος ανά κατεργασία που επιδιορθώνεται στο κέντρο επιδιόρθωσης	10

4.2 ΠΡΟΣΟΜΟΙΩΣΕΙΣ ΜΕ ΔΙΑΦΟΡΕΤΙΚΕΣ ΑΚΟΛΟΥΘΙΕΣ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ

Ένα από τα σημαντικότερα κριτήρια σωστής λειτουργίας ενός τέτοιου είδους αλγορίθμου είναι το να διερευνηθεί αν επιτυγχάνεται σύγκλιση. Για να διαπιστωθεί αν υπάρχει σύγκλιση, αρκεί να μελετηθεί η συμπεριφορά του αλγορίθμου για διαφορετικές ακολουθίες τυχαίων αριθμών. Επιλέγεται και εδώ συνολικός χρόνος προσομοίωσης: TSIM=20000 μονάδες χρόνου (ο χρόνος προσομοίωσης επιλέγεται μεγάλος για να εξομαλυνθεί η τυχαιότητα και να έχουμε μεγαλύτερη ακρίβεια). Ο χρόνος που απαιτείται για την προσομοίωση όλων των εναλλακτικών κατανομών είναι περίπου 1 ώρα.

Εκτελούνται τρεις διαφορετικές προσομοιώσεις, κάθε μία με διαφορετική ακολουθία τυχαίων αριθμών. Οι εναλλακτικές κατανομές σημείων επιθεώρησης προσομοιώνονται χρησιμοποιώντας κοινούς τυχαίους αριθμούς (από την ίδια δηλαδή ακολουθία), έτσι ώστε να υπάρχει δίκαιη σύγκριση όλων των εναλλακτικών κατανομών για τα σημεία επιθεώρησης σε κάθε προσομοίωση.

Τα αποτελέσματα που έδωσαν οι τρεις προσομοιώσεις συνοψίζονται στον ακόλουθο πίνακα.

Πίνακας 3. Βέλτιστες πολιτικές για τις τρεις διαφορετικές προσομοιώσεις

α/α Προσομοίωσης	1	2	3
Βέλτιστος ρυθμός κέρδους	1824	1816	1825
Σημεία ελέγχου	00000000011	00000000011	00000000011
Συνολικές πωλήσεις	44409	44257	44426

Όπως διαφαίνεται από τα αποτελέσματα των προσομοιώσεων και οι τρεις δίνουν τα ίδια βέλτιστα σημεία ελέγχου, μετά τις μηχανές $M_{10,1}$ και $M_{11,1}$. Ακόμα και οι συνολικές πωλήσεις που πραγματοποιήθηκαν από το σύστημα για τις τρεις προσομοιώσεις αποκλίνουν ελάχιστα ($<0.5\%$).

Συνεπώς ο αλγόριθμος συμπεριφέρεται ικανοποιητικά και επιτυγχάνεται η σύγκλιση και μάλιστα σε αρκετά ικανοποιητικό επίπεδο αφού και οι τρεις προσομοιώσεις δίνουν τα ίδια βέλτιστα σημεία ενώ και η απόκλιση στους ρυθμούς κέρδους είναι εξαιρετικά μικρή. Επομένως το χρονικό διάστημα της προσομοίωσης TSIM=20000 έχει ικανοποιητική διάρκεια.

4.3 ΑΠΟΤΕΛΕΣΜΑΤΑ ΤΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ 1

Σε αυτό το σημείο, ενδεικτικά, θα διερευνηθούν περισσότερο τα αποτελέσματα που προέκυψαν από την πρώτη προσομοίωση.

Οι τρεις καλύτερες πολιτικές καθώς και η χειρότερη, που προέκυψαν από τη προσομοίωση συνοψίζονται στο Πίνακα 4.

Σε ότι αφορά τις τρεις καλύτερες πολιτικές, είναι εμφανές ότι οι αποκλίσεις των ρυθμών κέρδους είναι μικρές. Σε ό,τι αφορά τις τοπολογίες, προτείνονται σημεία ελέγχου μετά τις τελευταίες μηχανές του δικτύου και μάλιστα δύο στο πλήθος. Με αυτό τον τρόπο συγκρατείται σε χαμηλά επίπεδα το κόστος από τις πωλήσεις ελαττωματικών προϊόντων (μηδενικό στην βέλτιστη πολιτική).

Από την άλλη μεριά, η χειρότερη πολιτική εξαιτίας της έλλειψης σταθμών ελέγχου στις τελικές μηχανές δεν αποφεύγει την πώληση ελαττωματικών προϊόντων στους πελάτες, με αποτέλεσμα να υπάρχουν επιστροφές, κρατώντας το κόστος σε υψηλά επίπεδα.

Επιπρόσθετα παρατηρείται ότι, αν και η πολιτική αυτή έχει 5 σημεία ελέγχου, το κόστος τους είναι σχετικά χαμηλό σε σύγκριση με τις βέλτιστες πολιτικές που έχουν μόλις δυο. Αυτό εξηγείται από το γεγονός ότι οι έλεγχοι γίνονται σε πρώιμα στάδια της παραγωγικής διαδικασίας, με αποτέλεσμα οι κατεργασίες που ελέγχονται να είναι λίγες στο πλήθος και οι έλεγχοι να βρίσκουν λιγότερα σφάλματα. Για τον ίδιο λόγο και το ύψος των πωλήσεων είναι μεγαλύτερο. Όμως, από την στιγμή που στην παραγωγική μονάδα έχουμε «κακό» έλεγχο ποιότητας αυξάνεται το πλήθος των επιστροφών σημαντικά, επιδρώντας αρνητικά στο κέρδος. Με αυτό τον τρόπο διαμορφώνεται τελικά μια διαφορά στον ρυθμό κέρδους της τάξεως των 351 χρηματικών μονάδων (χ.μ.), η οποία είναι αρκετά σημαντική, αν ληφθεί υπόψη ότι αναφερόμαστε σε ρυθμούς.

Πίνακας 4. Οι τρεις καλύτερες πολιτικές και η χειρότερη για την προσομοίωση 1

Πολιτική	00000000011	00000001010	00000000101	11000111000
Συνολικές πωλήσεις	44409	44676	44212	44863
Ρυθμός κέρδους από πωλήσεις	2220.45	2233.8	2210.6	2243.15
Κόστος αποθέματος ανά μονάδα χρόνου	80	80	80	80
Κόστος ανικανοποίητης ζήτησης ανά μονάδα χρόνου	66.632	66.5	66.756	66.384
Πλήθος ανικανοποίητων πελατών	55820	55550	56014	55364
Κόστος πώλησης ελαττωματικών προϊόντων	0	25.7	0	373.3
Κόστος ανά μονάδα χρόνου για σημεία ελέγχου	150	140	150	150
Καθαρό κέρδος απόρριψης ελαττωματικών εξαρτημάτων	0.5095	0.49	0.51	0.147
Συνολικό κέρδος	1824	1822	1814	1473

4.4 ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΚΤΥΟΥ ΓΙΑ ΔΙΑΦΟΡΕΤΙΚΑ $Q(i)$ ΚΑΙ $S(i)$

Τώρα εξετάζεται η συμπεριφορά του δικτύου για διαφορετικές τιμές στις πιθανότητες $Q(i)$ (επιδιορθώσιμη ελαττωματική κατεργασία) και $S(i)$ (ελαττωματική κατεργασία, που οδηγεί σε απόρριψη εξαρτήματος) για κάθε μηχανή $M_{i,1}$. Και εδώ θα χρησιμοποιήσουμε TSIM 20000. Τα δεδομένα θα παραμείνουν τα ίδια όπως στον Πίνακα 1 με την διαφορά ότι στις στήλες των $Q(i)$ και $S(i)$ θα έχουμε τις τιμές 0,01 και 0,0008 αντίστοιχα. Στον Πίνακα 5 συνοψίζονται κάποια από τα αποτελέσματα των δυο προσομοιώσεων. Παρουσιάζονται για κάθε περίπτωση η βέλτιστη και η χειρότερη πολιτική.

Πίνακας 5. Βέλτιστη και χειρότερη πολιτική για δυο διαφορετικές πιθανότητες ελαττωματικών

	Q(i)=0,015	S(i)=0,01	Q(i)=0,001	S(i)=0,0008
Πολιτική	00000000011	11000111000	00000001010	11001111000
Συνολικές πωλήσεις	44409	44863	44705	44893
Ρυθμός κέρδους από πωλήσεις	2220.45	2243.15	2235.25	2244.65
Κόστος αποθέματος ανά μονάδα χρόνου	80	80	80	80
Κόστος ανικανοποίητης ζήτησης ανά μονάδα χρόνου	66.632	66.384	66.586	66.4
Πλήθος ανικανοποίητων πελατών	55820	55364	55522	55335
Κόστος πώλησης ελαττωματικών προϊόντων	0	373.3	17.6	236.1
Κόστος ανά μονάδα χρόνου για σημεία ελέγχου	150	150	140	180
Καθαρό κέρδος απόρριψης ελαττωματικών εξαρτημάτων	0.5095	0.147	0.3955	0.15
Συνολικό κέρδος	1824	1473	1831	1582

Και εδώ παρατηρούνται φαινόμενα όπως ότι πολλά σημεία ελέγχου στην αρχή του δικτύου δεν ωφελούν, αφού το κόστος αυξάνεται. Η ζήτηση και στις δυο περιπτώσεις συμπεριφέρεται αμερόληπτα, χωρίς να επηρεάζει τα αποτελέσματα της προσομοίωσης (κοντά στο 66,4 για τις δυο χειρότερες πολιτικές και κοντά 66,6 για τις δυο βέλτιστες).

Αυτό όμως που διαφέρει σημαντικά στις δυο περιπτώσεις που προσομοιώνονται είναι ο ρυθμός κέρδους, που διαμορφώνεται από την χειρότερη πολιτική. Αν και οι βέλτιστες πολιτικές τους είναι πολύ κοντά, αυτό που τις διαφοροποιεί είναι ο ρυθμός κέρδους των δυο χειρότερων πολιτικών. Σε αυτές η διαφορά ανέρχεται πάνω από τις 100 χ.μ. . Μάλιστα στην δεύτερη περίπτωση, που η χειρότερη πολιτική έχει ρυθμό κέρδους 1582 χ.μ., σημειώνεται ότι η αμέσως χειρότερη πλησιάζει τις 1600 χ.μ. Αυτό εξηγείται από το γεγονός ότι η δεύτερη περίπτωση έχει μικρές πιθανότητες ελαττωματικών κατεργασιών, $Q(i)=0.001$ και $S(i)=0.0008$. Λιγότερα ελαττωματικά εξαρτήματα συνεπάγονται χαμηλότερο κόστος πώλησης ελαττωματικών προϊόντων, άρα υψηλότερο κέρδος.

Κάτι ανάλογο όμως δεν παρατηρείται στις βέλτιστες πολιτικές. Αυτό γιατί αυτές έχουν σταθμούς ελέγχου στις τελευταίες μηχανές, προλαμβάνοντας το γεγονός της πώλησης ελαττωματικών. Εδώ τα βέλτιστα κέρδη ανά μονάδα χρόνου είναι πολύ κοντά (διαφέρουν μόλις κατά 7χ.μ.).

5 ΣΥΜΠΕΡΑΣΜΑΤΑ - ΕΠΕΚΤΑΣΕΙΣ

Στην παρούσα εργασία περιγράφηκε ένας αλγόριθμος προσομοίωσης δικτύων παραγωγής με σκοπό την εξεύρεση των βέλτιστων σημείων εντός της παραγωγικής διαδικασίας όπου θα πρέπει να τοποθετηθούν κέντρα επιθεώρησης με στόχο την βελτιστοποίηση του κέρδους.

Η εφαρμογή του περιορίζεται σε δίκτυα συναρμολόγησης. Στο σύστημα παρατηρούνται και βλάβες, για την αποκατάσταση των οποίων απαιτούνται τυχαίοι χρόνοι επισκευής. Ακολουθείται μια πολιτική διαχείρισης των αποθεμάτων του συστήματος τύπου CONWIP, ενώ επιπλέον στο σύστημα παρατηρούνται αφίξεις πελατών και πωλήσεις.

Μπορούν να εξετασθούν όλες οι δυνατές περιπτώσεις τοποθέτησης των σταθμών ελέγχου. Δίνεται όμως και η δυνατότητα στον χρήστη να προεπιλέξει σημεία εντός του δικτύου για κάποια επιθυμητά κέντρα ελέγχου, περιορίζοντας τις εναλλακτικές. Στο σύστημα υπάρχει ένα κέντρο επιδιόρθωσης στο οποίο αποστέλλονται όλα τα ελαττωματικά εξαρτήματα. Εκεί αυτά είτε απορρίπτονται είτε επιστρέφουν στο δίκτυο επισκευασμένα.

Τέλος υπολογίζεται μια σειρά από κέρδη και κόστη, για την κάθε περίπτωση, τα οποία συγκρίνονται, έτσι ώστε τελικά ο αλγόριθμος να προτείνει την βέλτιστη πολιτική τοποθέτησης σημείων ελέγχου εντός του δικτύου, που θα επιφέρει και μέγιστο συνολικό ρυθμό κέρδους του συστήματος.

Ο αλγόριθμος που αναπτύχθηκε κάνει μια εξαντλητική έρευνα όλων των πιθανών σημείων όπου μπορούν να εισέλθουν σταθμοί ελέγχου ποιότητας, διαμορφώνοντας και διερευνώντας όλους τους δυνατούς συνδυασμούς. (Κάτι τέτοιο επιβάλλει υψηλό υπολογιστικό φόρτο σε έναν υπολογιστή.)

Ενδεικτικά για ένα σύστημα 11 μηχανών, όπως αυτό που αναλύθηκε, προκύπτουν $2^{11}=2048$ δυνατοί συνδυασμοί, που πρέπει να προσομοιωθούν, ενώ για 20 μηχανές ο αριθμός των συνδυασμών είναι 1048576. Ένας επεξεργαστής των 32-bit περιορίζει τις μηχανές που μπορεί να έχει το δίκτυο στις 31. Με την εφαρμογή όμως ενός γενετικού αλγορίθμου θα μπορούσε να επιτευχθεί σημαντική μείωση του υπολογιστικού φόρτου.

Ένα ακόμα από τα σημεία που θα μπορούσαν να μελετηθούν περαιτέρω είναι η επέκταση του αλγορίθμου, ώστε να περιγράφει πιο πολύπλοκα δίκτυα παραγωγής, με πολλά τελικά προϊόντα. Ο παρών αλγόριθμος εφαρμόζεται σε δίκτυα που παράγουν ένα τελικό προϊόν. Οι σύγχρονες γραμμές παραγωγής, για λόγους εξοικονόμησης χρόνου και χρήματος λειτουργούν με παράλληλα δίκτυα, που παράγουν πολλά τελικά προϊόντα. Επιπλέον θα ήταν δυνατό να επεκταθεί και σε δίκτυα που κάνουν και αποσυναρμολογήσεις.

Μπορεί λοιπόν η παρούσα εργασία να αποτελέσει αφετηρία για την ανάπτυξη ενός βελτιωμένου αλγορίθμου, με λιγότερο υπολογιστικό φόρτο και περισσότερες δυνατότητες.

6 ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] G.F. Lindsay and A.B. Bishop, Allocation of screening inspection effort: A dynamic programming approach, *Manage Sci* 10, 342-352, 1965.
- [2] J.K. Cochran and R. Erol, Performance modeling of serial production lines with inspection/repair stations, *International Journal of Production Research*, 39(8), 1707-1720, 2001.
- [3] V. Kakade, J. F. Valenzuela and J. S. Smith, "An optimization model for selective inspection in serial manufacturing systems," *International Journal of Production Research*, 42(18), 3891-3909, 2004.
- [4] S. Van Volsem, W. Dullaert, H. Van Landeghem, "An evolutionary algorithm and discrete event simulation for optimizing inspection strategies for multi-stage processes," *European Journal of Operational Research*, 179, 621-633, 2005.
- [5] Michal Penn, Tal Raviv, "Optimizing the quality control station configuration," *Naval Research Logistics*, 54, 1-14, 2006.
- [6] Β. Κουϊκόγλου, *Προσομοίωση, Σημειώσεις μαθήματος*, Πολυτεχνείο Κρήτης, 2006.

ΠΑΡΑΡΤΗΜΑ: ΚΩΔΙΚΑΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ο κώδικας που ακολουθεί προσομοιώνει δίκτυα συναρμολόγησης για όλους τους δυνατούς συνδυασμούς κέντρων ελέγχου και επιλέγει τον συνδυασμό με το μεγαλύτερο μέσο ρυθμό κέρδους. Αναπτύχθηκε σε γλώσσα προγραμματισμού FORTRAN.

```
PROGRAM DIKTYO
IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4(I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)
COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+ ISCRAP,ICON
COMMON/S6/ICRN0(20,2,5),IFI(40),IISE(20)
COMMON/S7/BSALE,BICON,BUNSD,EBIBAD,BCKEA,BCKEB,BISCRAP,BCE
CALL INPUT
CALL SEE
STOP
END

SUBROUTINE INPUT

!GIA NA DWSOUME TA DEDOMENA STON ALGORITHMOS MAS PREPEI NA
!DHMIOURGHISOYME ENA ARXEIO *.TXT STO OPOIO 8A EXOUME TA DEDOMENA
! WS EKSHS:
! IE ARXIKOSW SPOROS
! M PLH8OS MHXANWN SUNOLIKA
! TSIM XRONOS PROSOMOIWSHS 'H PLHTHOS KOMMATIWN
! ICON CONWIP
! IC XWRHTTIKOTHTA APOTHHKHS ZHTHSHS
! GIA TIS N MHXANES TOU DIKTUOU EXOUME:
! MD(I) MA(I) PRA(I) PRB(I) FRA(I) FRB(I) REA(I) REB(I)
! TA SHMEIA ELEGXOU EXEI MONO PRA(I)
! H ZHTHSH EXEI MONO PRA(I)
! H PWLHSH EXEI MONO PRA(I)
! H EPIDIOR8WSH EXEI PRA(I)
! MD EPOMENH MHXANH
! MA SYNTELESTHS SYNARMOLOGHSHS

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4(I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)
COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+ ISCRAP,ICON
COMMON/S6/ICRN0(20,2,5),IFI(40),IISE(20)
```

```

COMMON/S7/BSALE,BICON,BUNSD,BIBAD,BCKEA,BCKEB,BISCRAP,BCE
LOGICAL THERE ! T: FILES DOES PRE-EXIST
CHARACTER*80 CMMD ! COMMAND FOR OPERATING SYSTEM
C REAL STATUS ! STATUS OF EXECUTING OS COMMAND

REAL*8 Z

INQUIRE(FILE='DIKTYO.OUT', EXIST=THERE)
IF (THERE) THEN
    CMMD = 'DEL ' // 'DIKTYO.OUT'
    STATUS = SYSTEM(CMMD)
END IF
OPEN(4,FILE='DIKTYO.TXT',STATUS='OLD')
OPEN(7,FILE='DIKTYO.OUT',STATUS='NEW')
READ(4,*) IE,M,TSIM,ICON,IC
WRITE(7,*) IE,M,TSIM,ICON,IC

IF (IE.GT.21474) IE=1
Z=1973272912
DO 48 I=1,IE
    Z=DMOD(715.D0*Z, 2147483647.D0)
    Z=DMOD(1058.D0*Z, 2147483647.D0)
    Z=DMOD(1385.D0*Z, 2147483647.D0)
48 CONTINUE

N=M+3
DO 54 I=1,N
    DO 55 J=1,2
        Z=DMOD(715.D0*Z, 2147483647.D0)
        Z=DMOD(1058.D0*Z, 2147483647.D0)
        Z=DMOD(1385.D0*Z, 2147483647.D0)
        ICRN0(I,J,1)=IDINT(Z)
55 CONTINUE
    DO 50 J=2,5
        Z=DMOD(715.D0*Z, 2147483647.D0)
        Z=DMOD(1058.D0*Z, 2147483647.D0)
        Z=DMOD(1385.D0*Z, 2147483647.D0)
        ICRN0(I,1,J)=IDINT(Z)
50 CONTINUE
54 CONTINUE

DO 68 I=1,20
    MD(I)=0
    MA(I)=1
    MASS(I)=1
    NU(I)=0
    ISE(I)=0
    L(I,1)=0
    L(I,2)=0
    NMACHA=0
    MACHA(I)=0
    DO 76 J=1,20
        MU(I,J)=0
76 CONTINUE
68 CONTINUE

DO 134 I=1,N
    READ(4,*) MD(I),MA(I),IDISTP(I,1),IDISTP(I,2),IDISTF(I,1),
+    IDISTF(I,2),IDISTR(I,1),IDISTR(I,2),Q(I),S(I)
    IF ((IDISTP(I,1).LE.1)) THEN !GIA APLH MHXANH XWRIS SE EMPROS THS I STA8ERH 'H
EK8ETIKH
        READ(4,*) PRA(I,1)
    ELSE
        READ(4,*) PRA(I,1),PRB(I,1) !OMOIOMORFH
    ENDIF
    IF (IDISTP(I,2).LE.1) THEN !GIA SE EMPROS THS I STA8ERH 'H EK8ETIKH
        READ(4,*) PRA(I,2)
    ELSE
        READ(4,*) PRA(I,2),PRB(I,2)
    ENDIF
    IF (IDISTF(I,1).LE.1) THEN

```

```

        READ(4,*) FRA(I,1)
    ELSE
        READ(4,*) FRA(I,1),FRB(I,1)
    ENDIF
    IF (IDISTF(I,2).LE.1) THEN
        READ(4,*) FRA(I,2)
    ELSE
        READ(4,*) FRA(I,2),FRB(I,2)
    ENDIF
    IF (IDISTR(I,1).LE.1) THEN
        READ(4,*) REA(I,1)
    ELSE
        READ(4,*) REA(I,1),REB(I,1)
    ENDIF
    IF (IDISTR(I,2).LE.1) THEN
        READ(4,*) REA(I,2)
    ELSE
        READ(4,*) REA(I,2),REB(I,2)
    ENDIF
134    CONTINUE

    READ(4,*) (IISE(I),I=1,M)
    READ(4,*) BSALE,BICON,BUNSD,BIBAD,BCKEA,BCKEB,BISCRAP,BCE
    CLOSE(5)

    DO 77 I=1,M      !ME GNWSTES TIS MD(I) BRISKW NU(I) & MU(I,J)
        J=MD(I)
        IF (J.GT.0) THEN
            NU(J)=NU(J)+1
            MU(J,NU(J))=I
        ENDIF
77    CONTINUE

    !BRISKW A' MHXANES DIKTYOU KAI TIS APO8HKEUW STON PINAKA
    !MACHA(I,1) ENW STO NMACHA KRATW TO PLH8OS TOUS
    !EPIPLEON BRISKOU ME PRWTES MHXANES PLH8OS KAI POIES EINAI GIA KA8E
    !MHXANH TOU DIKTUOU MAS NMF,MF
    NMACHA=0
    DO 173 I=1,M
        NMF(I)=0
        IF (NU(I).EQ.0) THEN
            NMACHA=NMACHA+1
            MACHA(NMACHA)=I
            NMF(I)=1
            MF(I,1)=I
        ELSE
            DO 181 J=1,NU(I)
                I1=MU(I,J)
                DO 183 J1=1,NMF(I1)
                    I2=MF(I1,J1)
                    NMF(I)=NMF(I)+1
                    MF(I,NMF(I))=I2
                CONTINUE
            CONTINUE
            ENDIF
173    CONTINUE

    !TWRA 8A UPOLOGISOUME TO MASS KA8E MHXANHS, DHLADH POSA KOMMATIA APO
    !KA8E MHXANH 8ELOUME GIA NA PAROUME ENA TELIKO PROION
    MASS(M)=1
    DO 197 I=1,M-1
        MASS(I)=MA(I)
        I1=MD(I)
201    IF (I1.EQ.M) GO TO 197
        MASS(I)=MASS(I)*MA(I1)
        I1=MD(I1)
        GO TO 201
197    CONTINUE

    !PROS8ETOUME MHXANES ZHTHSHS KAI PWLHSH STO DIKTUO

```

```

MD(N-1)=N
NU(N)=2
MU(N,1)=M
MU(N,2)=N-1
MD(M)=N

!FTIAXNW NU,MU TWN PRWTWN MHXANWN
DO 123 I=1,NMACHA
    K=MACHA(I)
    MASS(N+I)=MASS(K)
    NU(K)=1
    MU(K,1)=N+I
    L(N+I,1)=0
123    CONTINUE

!FTIAXNW PINAKA IFI POU MOU LEEI POIES EINAI OI MHXANES
!PSEUTIKES PRIN TIS A' MHXANES
DO 111 I=1,N
    IFI(I)=0
111    CONTINUE
DO 110 I=N+1,N+NMACHA
    IFI(I)=1
110    CONTINUE

RETURN
END

!-----

SUBROUTINE SEE
IMPLICIT INTEGER*4(I-N)
IMPLICIT REAL*8 (A-H,O-Y)

COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+   L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+   NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+   PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)
COMMON/S5/OPTIMAL,UNSD,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+   ISCRAP,ICON
COMMON/S6/ICRNOLD(20,2,5),IFI(40),IISE(20)

DO 212 I=1,M
    ISE(I)=0
212    CONTINUE
    NE=1
    WRITE(7,*)'*** SIMULATION NO.',NE
    WRITE(7,*) 'SE=',(ISE(I),I=1,M)
    WRITE(*,*)'*** SIMULATION NO.',NE
    WRITE(*,*) 'SE=',(ISE(I),I=1,M)
    CALL SIMULATION
217    K=M
218    IF (K.EQ.0) RETURN
    IF (ISE(K).EQ.0) THEN
        ISE(K)=1
        DO 333 I=1,M
            IF ((IISE(I).EQ.1).AND.(ISE(I).EQ.0)) THEN
                WRITE(7,*) 'SE=',(ISE(I1),I1=1,M)
                WRITE(*,*) 'SE=',(ISE(I1),I1=1,M)
                GOTO 217
            ENDIF
333        CONTINUE
        NE=NE+1
        WRITE(7,*) ' '
        WRITE(7,*)'*** SIMULATION NO.',NE
        WRITE(7,*) 'SE=',(ISE(I1),I1=1,M)

```

```

        WRITE(*,*) ' '
        WRITE(*,*) '*** SIMULATION NO.',NE
        WRITE(*,*) 'SE=',(ISE(I1),I1=1,M)
        CALL SIMULATION
        GOTO 217
    ELSE
        ISE(K)=0
        K=K-1
        GOTO 218
    ENDIF

    RETURN

END

!-----

SUBROUTINE SIMULATION

    IMPLICIT REAL*8 (A-H,O-Y)
    IMPLICIT INTEGER*4(I-N)
    COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
    COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
    COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
    COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)
    COMMON/S6/ICRN0(20,2,5),IFI(40),IISE(20)
    COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+ ISCRAP,ICON

    TN=0.D0

    DO 250 I=1,N
        DO 251 J=1,2
            ICRN(I,J,1)=ICRN0(I,J,1)
251        CONTINUE
        DO 252 J=2,5
            ICRN(I,1,J)=ICRN0(I,1,J)
252        CONTINUE
250    CONTINUE

    DO 200 I=1,N
        MS(I,1)=0
        MS(I,2)=0
        NPTF(I,1)=0
        NPTF(I,2)=0
        EVT(I,1)=TSIM+1
        EVT(I,2)=TSIM+1
        TE(I)=TSIM+1
        IEV(I)=0
        L(I,1)=0
        L(I,2)=0
        NMYD(I)=0
        DO 101 J=1,N
            MYD(I,J)=0
101        CONTINUE
200    CONTINUE

    NLE=0
    ISALE=0
    UNSDE=0
    IBADCUST=0
    IXCUST=0
    ISCRAP=0

    !ME DEDOMENO PLEON TON ISE PINAKA BRISKW MHXANES UPODIKTUOU GIA
    !KA8E SHMEIO ELEGXOU-MHXANH- ISE(i)=1

```

```

DO 400 K=1,NMACHA
  NMYD(K)=1
  MYD(K,1)=K
400 CONTINUE
DO 401 K=NMACHA+1,M
  NMYD(K)=1
  MYD(K,1)=K
  DO 402 I=1,NU(K)
    IP=MU(K,I)
    IF(ISE(IP).EQ.1) GOTO 402
    NMY=NMYD(K)
    NMYD(K)=NMY+NMYD(IP)
    DO 403 I1=1,NMYD(IP)
      MYD(K,NMY+I1)=MYD(IP,I1)
403 CONTINUE
402 CONTINUE
401 CONTINUE

DO 257 I=1,M
  IF (ISE(I).EQ.0) THEN
    IEV(I)=1
    EVT(I,2)=TSIM+10
  ENDIF
257 CONTINUE

!BAZW A' YLES STIS PRWTES MHXANES GIA APO8EMA OSO LEEI XRHSTHS
!KAI GIA KA8E MHXANH TOU LEW XRONOUS NEOUS

DO 262 I=1,NMACHA
  K=MACHA(I)
  L(MU(K,1),1)=MASS(K)*ICON
  IF (IDISTP(K,1).EQ.0)THEN
    EVT(K,1)=PRA(K,1)
  ELSEIF (IDISTP(K,1).EQ.1) THEN
    IE=ICRN(K,1,1)
    EVT(K,1)=-ALOG(ZRAND(IE))*PRA(K,1)
    ICRN(K,1,1)=IE
  ELSE
    IE=ICRN(K,1,1)
    EVT(K,1)=PRA(K,1)+(PRB(K,1)-PRA(K,1))*ZRAND(IE)
    ICRN(K,1,1)=IE
  ENDIF
  !KOMMATIA BLABWN
260 IF (IDISTF(K,1).EQ.0) THEN
    NPTF(K,1)=200000000
  ELSEIF (IDISTF(K,1).EQ.1) THEN
    IE=ICRN(K,1,2)
    NPTF(K,1)=IDINT(ALOG(ZRAND(IE))/DLOG(1-FRA(K,1)))
    ICRN(K,1,2)=IE
  ELSE
    IE=ICRN(K,1,2)
    NPTF(K,1)=FRA(K,1)+INT((FRB(K,1)-FRA(K,1)+1)*ZRAND(IE))
    ICRN(K,1,2)=IE
  ENDIF
  IF(NPTF(K,1).GT.0) GOTO 262
  !XRONOI EPISKEUHS
  IF (IDISTR(K,1).EQ.0) THEN
    EVT(K,1)=EVT(K,1)+REA(K,1)
  ELSEIF (IDISTR(K,1).EQ.1) THEN
    IE=ICRN(K,1,3)
    EVT(K,1)=EVT(K,1)-ALOG(ZRAND(IE))*REA(K,1)
    ICRN(K,1,3)=IE
  ELSE
    IE=ICRN(K,1,3)
    EVT(K,1)=EVT(K,1)+REA(K,1)+(REB(K,1)-REA(K,1))*ZRAND(IE)
    ICRN(K,1,3)=IE
  ENDIF
  GO TO 260
262 CONTINUE

```

```

      IF (TE(N-1).GE.TSIM) THEN
        CALL PROCESS(N-1,1)
      ENDIF

      DO 350 J=1,NMACHA
        I=MACHA(J)
        IF (EVT(I,1).LT.EVT(I,2)) THEN
          TE(I)=EVT(I,1)
          IEV(I)=1
        ELSE
          TE(I)=EVT(I,2)
          IEV(I)=2
        ENDIF
350    CONTINUE

      !APEIRIZW XRONOU GIA TIS 3 EIDIKES MHXANES ETSI WSTE
      !NA MHN DOYLEYOYN WS SHMEIA ELEGXOU
      EVT(N,2)=TSIM+10
      EVT(N-1,2)=TSIM+10
      EVT(N-2,2)=TSIM+10

      !BRISKOUME EDW THN MHXANH M OPOU KAI 8A SUMBEI TO EPOMENO
      !GEGONOS KAI EINAI H YPO EKSETASH MHXANH
      WRITE(7,*) 'EVT1=',(EVT(I,1),I=1,N)
      WRITE(7,*) 'EVT2=',(EVT(I,2),I=1,M)
500    TN=TSIM+1
      DO 303 I=1,N
        IF (TE(I).LT.TN) THEN
          TN=TE(I)
          MACHINE=I
          IEV(MACHINE)=IEV(I)
        ENDIF
303    CONTINUE

      IF (TN.GE.TSIM) THEN
        CALL OUTPUT(ISE)
        RETURN
      ENDIF

      IF ((MACHINE.LE.M).AND.(IEV(MACHINE).EQ.1)) THEN
        CALL SIMPLE(MACHINE)
      ELSEIF (MACHINE.LE.M) THEN
        CALL INSPECTION(MACHINE)
      ELSEIF (MACHINE.EQ.N-1) THEN
        CALL ZHTSH
      ELSEIF (MACHINE.EQ.N-2) THEN
        CALL REPAIR
      ELSE
        CALL PWLHSH
      ENDIF

      GO TO 500

      END

      !-----
      !UPOROUTINA GIA MHXANH K OTAN EINAI EITE JJ=1,APLH MHXANH,
      !EITE JJ=2, SHMEIO ELEGXOU

      SUBROUTINE PROCESS(K,JJ)

      IMPLICIT REAL*8 (A-H,O-Y)
      IMPLICIT INTEGER*4(I-N)
      COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
      COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)

```

```

COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)

!XRONOI KATERGASIWN
IF (IDISTP(K,JJ).EQ.0) THEN
    TTPR=PRA(K,JJ)
ELSEIF (IDISTP(K,JJ).EQ.1) THEN
    IE=ICRN(K,JJ,1)
    TTPR=-ALOG(ZRAND(IE))*PRA(K,JJ)
    ICRN(K,JJ,1)=IE
ELSE
    IE=ICRN(K,JJ,1)
    TTPR=PRA(K,JJ)+(PRB(K,JJ)-PRA(K,JJ))*ZRAND(IE)
    ICRN(K,JJ,1)=IE
ENDIF
IF (JJ.EQ.1) THEN
    EVT(K,1)=TN+TTPR
ELSE
    EVT(K,2)=TN+TTPR*NMYD(K)
ENDIF

IF (NPTF(K,JJ).GT.0) GOTO 540
!XRONOI EPISKEUHS
555 IF (IDISTR(K,JJ).EQ.0) THEN
    EVT(K,JJ)=EVT(K,JJ)+REA(K,JJ)
ELSEIF (IDISTR(K,JJ).EQ.1) THEN
    IE=ICRN(K,JJ,3)
    EVT(K,JJ)=EVT(K,JJ)-ALOG(ZRAND(IE))*REA(K,JJ)
    ICRN(K,JJ,3)=IE
ELSE
    IE=ICRN(K,JJ,3)
    EVT(K,JJ)=EVT(K,JJ)+REA(K,JJ)+(REB(K,JJ)-REA(K,JJ))*ZRAND(IE)
    ICRN(K,JJ,3)=IE
ENDIF
!KOMMATIA BLABWN
IF (IDISTF(N,JJ).EQ.0) THEN
    NPTF(K,JJ)=200000000
ELSEIF (IDISTF(K,JJ).EQ.1) THEN
    IE=ICRN(K,JJ,2)
    NPTF(K,JJ)=IDINT(ALOG(ZRAND(IE))/DLOG(1-FRA(K,JJ)))
    ICRN(K,JJ,2)=IE
ELSE
    IE=ICRN(K,JJ,2)
    NPTF(K,JJ)=FRA(K,JJ)+INT((FRB(K,JJ)-FRA(K,JJ)+1)*ZRAND(IE))
    ICRN(K,JJ,2)=IE
ENDIF
IF (NPTF(K,JJ).LE.0) GO TO 555

!KRATAME XRONO MIKROTERO GIA MIA MHXANH I ANALOGA 1 KAI 2
540 IF (EVT(K,1).LT.EVT(K,2)) THEN
    TE(K)=EVT(K,1)
    IEV(K)=1
ELSE
    TE(K)=EVT(K,2)
    IEV(K)=2
ENDIF

RETURN

END

!-----
SUBROUTINE SIMPLE (K)

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4 (I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,

```



```

+   NMF(20),MF(20,20),NMYD(20),MYD(20,20)
+   COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+   PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
+   COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)

DO 580 I=1,NU(K)
    IM=MU(K,I)
    L(IM,1)=L(IM,1)-MA(IM)
580 CONTINUE

NPTF(K,1)=NPTF(K,1)-1
IF (ISE(K).EQ.0) THEN !AN META K OXI SHMEIO ELEGXOU
    L(K,1)=L(K,1)+1
    IM=MD(K)
    IF ((MS(IM,1).EQ.0).AND.(L(K,1).GE.MA(K))) THEN
        DO 569 I=1,NU(IM)
            M1=MU(IM,I)
            IF (L(M1,1).LT.MA(M1)) GOTO 599
569 CONTINUE
            MS(IM,1)=1
            CALL PROCESS (IM,1)
        ENDIF
    ELSEIF (ISE(K).EQ.1) THEN !AN META K SHMEIO ELEGXOU
        L(K,2)=L(K,2)+1
        IF (MS(K,2).EQ.0) THEN
            CALL PROCESS (K,2)
        ENDIF
    ENDIF

!H K EINAI PEINASMENH?
599 DO 587 I=1,NU(K)
    IM=MU(K,I)
    IF (L(IM,1).LT.MA(IM)) THEN
        MS(K,1)=0
        EVT(K,1)=TSIM+1
        IF (EVT(K,1).LE.EVT(K,2)) THEN
            TE(K)=EVT(K,1)
            IEV(K)=1
        ELSE
            TE(K)=EVT(K,2)
            IEV(K)=2
        ENDIF
        RETURN
    ENDIF
587 CONTINUE

CALL PROCESS (K,1)

IF ((K.EQ.M).AND.(ISE(M).NE.1)) CALL FALSINSPE

RETURN
END

!-----
SUBROUTINE INSPECTION (I)

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4 (I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)

NPTF(I,2)=NPTF(I,2)-1
NFOUND=0

```

```

DO 629 K=1,NMYD(I)
  M1=MYD(I,K)
  NFOP=0
  DO 762 NOP=1,MASS(M1)/MASS(I)
    IE=ICRN(M1,1,4)
    CHECK=ZRAND(IE)
    ICRN(M1,1,4)=IE
    IF (CHECK.LE.(Q(M1)+S(M1))) NFOP=NFOP+1
762  CONTINUE
    IF (NFOP.GT.0) THEN
      IF (NFOUND.EQ.0) THEN
        NFOUND=1
        NLE=NLE+1
        IF (NLE.GT.1000) THEN
          WRITE(*,*) 'ERROR: SIZE OF MLE EXCEEDS 1000',
+           'TIME=',TN,' INSP=',I
          WRITE(7,*) 'ERROR: SIZE OF MLE EXCEEDS 1000',
+           'TIME=',TN,' INSP=',I
          STOP
        ENDIF
        NBAD(NLE)=1
        MBAD(NLE,1)=M1
        NMBAD(NLE,1)=NFOP
      ELSE
        NBAD(NLE)=NBAD(NLE)+1
        MBAD(NLE,NBAD(NLE))=M1
        NMBAD(NLE,NBAD(NLE))=NFOP
      ENDIF
    ENDIF
629  CONTINUE

  IF (NFOUND.EQ.1) THEN !BRHKE SFALMATA
    !PAEI STHN MHXANH REPAIR
    L(M+1,1)=L(M+1,1)+1
    MLE(NLE)=I
    IF (MS(M+1,1).EQ.0) THEN
      IE=ICRN(M+1,1,1)
      TTPR=-ALOG(ZRAND(IE))*PRA(M+1,1)*NMYD(I)/NBAD(NLE)
      ICRN(M+1,1,1)=IE
      EVT(M+1,1)=TN+TTPR
      IF (EVT(M+1,1).LT.EVT(M+1,2)) THEN
        TE(M+1)=EVT(M+1,1)
        IEV(M+1)=1
      ELSE
        TE(M+1)=EVT(M+1,2)
        IEV(M+1)=2
      ENDIF
      MS(M+1,1)=1
    ENDIF
  ELSE !NFOUND=0 OXI SFALMATA
    L(I,1)=L(I,1)+1
    M1=MD(I) !TROFODOTOUME EMPROS THS I
    IF ((MS(M1,1).EQ.0).AND.(L(I,1).GE.MA(I))) THEN
      IOK=1
      DO 670 J=1,NU(M1)
        M2=MU(M1,J)
        IF (L(M2,1).LT.MA(M2)) THEN
          IOK=0
        ENDIF
670  CONTINUE
      IF (IOK.EQ.0) GO TO 721 !PARE KOMMATI APO APO8HKH ELEGXOU
      MS(M1,1)=1
      CALL PROCESS (M1,1)
    ENDIF
  ENDIF

721  L(I,2)=L(I,2)-1
  IF (L(I,2).EQ.0) THEN
    MS(I,2)=0
    EVT(I,2)=TSIM+1
    IF (EVT(I,1).LE.EVT(I,2)) THEN

```

```

        TE(I)=EVT(I,1)
        IEV(I)=1
    ELSE
        TE(I)=EVT(I,2)
        IEV(I)=2
    ENDIF
    RETURN
ENDIF

CALL PROCESS(I,2) !FTIAXNEI XRONO PARAGWGHS GIA SHMEIO ELEGXOU META THN I

RETURN

END

!-----

SUBROUTINE REPAIR

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4 (I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+   L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+   NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+   PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
        COMMON/S4/NLE,MLE(1000),MBAD(1000,20),NBAD(1000),NMBAD(1000,20)
COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+   ISCRAP,ICON

IF (MS(M+1,1).EQ.1) THEN

NPTF(M+1,1)=NPTF(M+1,1)-1
IO=MLE(1)
IGOOD=1
DO 700 K1=1,NBAD(1)
    K=MBAD(1,K1)
    DO 701 J1=1,NMBAD(1,K1)
        IE=ICRN(K,1,5)
        CHECK=ZRAND(IE)
        ICRN(K,1,5)=IE
        IF (CHECK.GT.Q(K)/(Q(K)+S(K))) THEN
            IGOOD=0
            GO TO 705
        ENDIF
701    CONTINUE
700    CONTINUE

705    IF (IGOOD.EQ.0) THEN
        ISCRAP=ISCRAP+1
        DO 702 J=1,NMF(IO)
            I=MF(IO,J)
            IUP=MU(I,1)
            L(IUP,1)=L(IUP,1)+MASS(I)/MASS(IO)
            IF (MS(I,1).EQ.0) THEN
                MS(I,1)=1
                CALL PROCESS(I,1)
            ENDIF
702    CONTINUE
    ELSE
        !PAEI STHN APO8HKH THS I ALLA THN 1
        L(IO,1)=L(IO,1)+1
        I=MD(IO) !TROFODOTOUME EMPROS TIS I
        IF ((MS(I,1).EQ.0).AND.(L(IO,1).GE.MA(IO))) THEN
            IOK=0
            DO 703 J=1,NU(I)
                K=MU(I,J)
                IF (L(K,1).LT.MA(K)) IOK=1
            END DO
        END IF
    END IF
END SUBROUTINE REPAIR

```

```

703      CONTINUE
          IF (IOK.EQ.1) GO TO 750
          MS(I,1)=1
          CALL PROCESS (I,1)
      ENDIF
ENDIF

!AFAIROUME KOMMATI APO THN LISTA ME TA UPO DIOR8WSH KOMMATIA
750      L(M+1,1)=L(M+1,1)-1
          NLE=NLE-1
          DO 740 K=1,NLE
              MLE(K)=MLE(K+1)
              NBAD(K)=NBAD(K+1)
              DO 745 J=1,NBAD(K)
                  MBAD(K,J)=MBAD(K+1,J)
                  NMBAD(K,J)=NMBAD(K+1,J)
745          CONTINUE
740      CONTINUE

      ENDIF

      IF (L(M+1,1).EQ.0) THEN
          MS(M+1,1)=0
          EVT(M+1,1)=TSIM+1
          IEV(M+1)=1
          TE(M+1)=EVT(M+1,1)
      ELSE
          CALL PROCESS(M+1,1)
      ENDIF

      RETURN

      END

!-----

      SUBROUTINE PWLHSH

          IMPLICIT REAL*8 (A-H,O-Y)
          IMPLICIT INTEGER*4 (I-N)
          COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+      L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+      NMF(20),MF(20,20),NMYD(20),MYD(20,20)
          COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+      PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
          COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
          COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+      ISCRAP,ICON

          NPTF(N,1)=NPTF(N,1)-1
          ISALE=ISALE+1 !PWLHSH ENOS KOMMATIOY
          UNSDE=UNSDE+TN

          DO 780 I=1,NU(N)
              ID=MU(N,I)
              L(ID,1)=L(ID,1)-1
780          CONTINUE

          DO 787 I=1,NMACHA
              IK=MACHA(I)
              L(MU(IK,1),1)=L(MU(IK,1),1)+MASS(IK)
              IF (MS(IK,1).EQ.0) THEN
                  MS(IK,1)=1
                  CALL PROCESS (IK,1)
              ENDIF
787          CONTINUE

          !ELEGXOYME AN MU EXOUN MA GIA NA TROFODOTHOUN THN MHXANH PWLHSH

```

```

DO 797 I=1,NU(N)
  IK=MU(N,I)
  IF (L(IK,1).LT.MA(IK)) THEN
    MS(N,1)=0
    IEV(N)=1
    TE(N)=TSIM+1
    RETURN
  ENDIF
797 CONTINUE

CALL PROCESS (N,1)

RETURN

END

!-----

SUBROUTINE ZHTHSH

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4(I-N)

COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S2/IDISTP(20,2),IDISTF(20,2),IDISTR(20,2),PRA(20,2),
+ PRB(20,2),FRA(20,2),FRB(20,2),REA(20,2),REB(20,2),NPTF(20,2)
COMMON/S3/EVT(20,2),IEV(20),TN,TE(20),ISE(20),MS(20,2)
COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+ ISCRAP,ICON

NPTF(M+2,1)=NPTF(M+2,1)-1
IF (L(N-1,1).GE.IC) THEN
  IXCUST=IXCUST+1
ELSE
  !TROFODOTOYME PWLHSH
  UNSDE=UNSDE-TN
  L(N-1,1)=L(N-1,1)+1
  IF (MS(N,1).EQ.0) THEN
    DO 833 I=1,NU(N)
      M1=MU(N,I)
      IF (L(M1,1).LT.MA(M1)) GO TO 830
833 CONTINUE
      MS(N,1)=1
      CALL PROCESS (N,1) !XRONOI GIA PWLHSH
    ENDIF
  ENDIF

830 CALL PROCESS (N-1,1)

RETURN

END

!-----

SUBROUTINE FALSINSPE

IMPLICIT REAL*8 (A-H,O-Y)
IMPLICIT INTEGER*4(I-N)
COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+ L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+ NMF(20),MF(20,20),NMYD(20),MYD(20,20)
COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+ ISCRAP,ICON

DO 860 K=1,NMYD(M)

```

```

      II=MYD(M,K)
      DO 870 NOP=1,MASS(II)/MASS(M)
        IE=ICRN(II,1,4)
        CHECK=ZRAND(IE)
        ICRN(II,1,4)=IE
        IF (CHECK.LE.(Q(II)+S(II))) THEN
          IBADCUST=IBADCUST+1
          RETURN
        ENDIF
870      CONTINUE
860      CONTINUE

      RETURN
      END

!-----

      SUBROUTINE OUTPUT (ISE)

      IMPLICIT REAL*8 (A-H,O-Y)
      IMPLICIT INTEGER*4 (I-N)
      INTEGER*4 ISE(20)
      COMMON/S1/M,N,NU(20),MU(20,20),MD(20),MASS(20),MA(20),
+      L(20,2),TSIM,Q(20),S(20),ICRN(20,2,5),IC,MACHA(20),NMACHA,
+      NMF(20),MF(20,20),NMYD(20),MYD(20,20)
      COMMON/S5/OPTIMAL,UNSDE,ISEOPTIMAL(20),ISALE,IXCUST,IBADCUST,
+      ISCRAP,ICON
      COMMON/S7/BSALE,BICON,BUNSD,EBIBAD,BCKEA,BCKEB,BISCRAP,BCE

      WRITE(*,*) 'PLH8OS PWLHSEWN APO TO SUSTHMA: ', ISALE
      WRITE(7,*) 'PLH8OS PWLHSEWN APO TO SUSTHMA: ', ISALE
      EP=BSALE*ISALE/TSIM
      WRITE(*,*) 'SUNOLIKO Kerdos PwlHsewn Gia to Susthma: ', EP
      WRITE(7,*) 'SUNOLIKO Kerdos PwlHsewn Gia to Susthma: ', EP

      CQ=BICON*ICON
      WRITE(*,*) 'KOSTOS APO8EMATOS ANA MON. XRONOY: ', CQ
      WRITE(7,*) 'KOSTOS APO8EMATOS ANA MON. XRONOY: ', CQ

      CW=(UNSDE+L(N-1,1)*TSIM)/TSIM
      CW=BUNSD*CW
      WRITE(*,*) 'KOSTOS ANIKANOP. ZHTSHS ANA MON. XRONOY: ', CW
      WRITE(7,*) 'KOSTOS ANIKANOP. ZHTSHS ANA MON. XRONOY: ', CW

      CR=BIBAD*IBADCUST/TSIM
      WRITE(*,*) 'KOSTOS APO PwlHseis elattwmatikwn: ', CR
      WRITE(7,*) 'KOSTOS APO PwlHseis elattwmatikwn: ', CR

      WRITE(*,*) 'PELATES POU EFUGAN: ', IXCUST
      WRITE(7,*) 'PELATES POU EFUGAN: ', IXCUST

      CKE=0
      DO 913 J=1,M
        IF (ISE(J).EQ.1) THEN
          CKE=CKE+BCKEA+BCKEB*NMYD(J)
        ENDIF
913      CONTINUE

      WRITE(*,*) 'KOSTOS ANA MON. XRONOY GIA SHMEIA ELEGXOU: ', CKE
      WRITE(7,*) 'KOSTOS ANA MON. XRONOY GIA SHMEIA ELEGXOU: ', CKE

      EG=BISCRAP*ISCRAP/TSIM
      WRITE(*,*) 'Kerdos APORRIPSHS elattwmatikwn: ', EG
      WRITE(7,*) 'Kerdos APORRIPSHS elattwmatikwn: ', EG

      DO 10 I=1,M
10      IF (ISE(I).EQ.1) GOTO 20
      CE=0

```

```

      GOTO 30
20    CE=BCE
30    WRITE(*,*) 'KOSTOS ANA MONADA XRONOU KENTROU EPIDIOR8WSHS: ', CE
      WRITE(7,*) 'KOSTOS ANA MONADA XRONOU KENTROU EPIDIOR8WSHS: ', CE
      !KATHARO KERDOS/XRONO APO XRHSH SYGKEKRIMENHS TOPOLOGIAS DIKTUOU

      KERDOS=EP+EG-CW-CQ-CR-CKE-CE
      WRITE(*,*) '**SUNOLIKOS RYTH KERDOYS APO SYGKEKRIMENH TOPOLOGIA',
+      KERDOS
      WRITE(7,*) '**SUNOLIKOS RYTH KERDOYS APO SYGKEKRIMENH TOPOLOGIA',
+      KERDOS

      IF (KERDOS.GT.OPTIMAL) THEN
        OPTIMAL=KERDOS
        DO 1000 J=1,M
          ISEOPTIMAL(J)=ISE(J)
1000    CONTINUE
      ENDIF
      WRITE(*,*) 'BELTISTO KERDOS EWS TWRA:', OPTIMAL
      WRITE(*,*) 'ME BELTISTA SHMEIA ELEGXOU:', (ISEOPTIMAL(I), I=1,M)
      WRITE(7,*) 'BELTISTO KERDOS EWS TWRA:', OPTIMAL
      WRITE(7,*) 'ME BELTISTA SHMEIA ELEGXOU:', (ISEOPTIMAL(I), I=1,M)

      RETURN
      END

!-----

      FUNCTION ZRAND(IE)
C*****C
C THIS FUNCTION GENERATES UNIFORM (0,1) RANDOM NUMBERS C
C*****C

      INTEGER*4 A1,A2,P,B15,B16,XHI,XALO,LEFTLO,FHI,K,IE
      DATA B15/32768/,B16/65536/,P/2147483647/
      DATA A1/24112/,A2/26143/
      XHI=IE/B16
      XALO=(IE-XHI*B16)*A1
      LEFTLO=XALO/B16
      FHI=XHI*A1+LEFTLO
      K=FHI/B15
      IE=((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IE.LT.0)IE=IE+P
      XHI=IE/B16
      XALO=(IE-XHI*B16)*A2
      LEFTLO=XALO/B16
      FHI=XHI*A2+LEFTLO
      K=FHI/B15
      IE=((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IE.LT.0)IE=IE+P
      ZRAND=(2*(IE/256)+1)/16777216.
      RETURN
      END

```