



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΣΗΜΕΙΩΣΕΙΣ ΤΟΥ ΜΑΘΗΜΑΤΟΣ
“ΠΡΟΣΟΜΟΙΩΣΗ”

Ι. Δαρζέντας
Θ. Σπύρου
Χ. Τσαγγάρης

ΚΑΡΛΟΒΑΣΙ 2001

Περιεχόμενα

1. Προσομοίωση	1
1.1 Εισαγωγή	2
1.2 Συστήματα και Μοντέλα	3
1.2.1 Σύστημα	3
1.2.2 Μοντέλο	7
1.3 Αντιμετώπιση Κοινωνικο-Οικονομικών Προβλημάτων	10
1.4 Συστήματα Υποστήριξης Αποφάσεων	13
1.5 Τι είναι Προσομοίωση?	14
1.6 Γιατί Προσομοίωση?	15
1.7 Εφαρμογές της Προσομοίωσης	17
1.8 Ο ρόλος της Στατιστικής στη Προσομοίωση	19
1.9 Μελλοντικές Δυνατότητες της Προσομοίωσης	20
2. Προσομοίωση Διακριτών Γεγονότων	21
2.1 Εισαγωγή	22
2.2 Ροές και Ανακυκλώσεις	24
2.3 Μέθοδος Γεγονότων	26
2.4 Μέθοδος Δραστηριοτήτων	43
2.5 Μέθοδος Διαδικασιών	53
3. Τυχαίοι αριθμοί – Έλεγχος τυχειότητας	63
3.1 Τυχαίοι αριθμοί	64
3.2 Ψευδοτυχαίοι αριθμοί	65
3.3 Γεννήτριες τυχαίων αριθμών	66
3.3.1 Μέθοδος Μέσων Τετραγώνων	66
3.3.2 Μέθοδος Μέσων Γινομένων	67
3.3.3 Congruential Μέθοδοι	68
3.3.3.1 Πολλαπλασιαστικός τύπος	68
3.3.3.2 Μικτός τύπος	69
3.4 Έλεγχος τυχειότητας	70
3.4.1 χ^2 -test	70
3.4.2 Poker test	75
3.4.3 Gap test	76
3.4.4 Runs test	76

4. Δειγματοληψία από συνεχές και διακριτές κατανομές	77
4.1 Εισαγωγή	78
4.2 Μέθοδος της αντίστροφης	78
4.3 Μέθοδος της απόρριψης	82
4.4 Δειγματοληψία από κανονική κατανομή $N(\mu, \sigma^2)$	84
4.5 Δειγματοληψία από ομοιόμορφη διακριτή κατανομή $U'(\alpha, \beta)$	85
4.6 Μέθοδος του αντίστροφου μετασχηματισμού για διακριτές κατανομές	86
4.6.1 Δειγματοληψία από γεωμετρική κατανομή	87
4.6.2 Δειγματοληψία από Poisson κατανομή	88
4.7 Δειγματοληψία από διωνυμική κατανομή $B(n, p)$	88
4.8 Δειγματοληψία από εμπειρική κατανομή $E(n, x_i, p_i)$	89
5. Η γλώσσα Simgscript II.5	91
5.1 Η γλώσσα Simgscript II.5 και το περιβάλλον εργασίας SimLab	92
5.1.1 Ιστορική αναδρομή	92
5.1.2 Simgscript II.5 – Προσομοίωση	92
5.1.3 Το περιβάλλον SimLab	92
5.1.4 Ο editor κειμένου SimEdit	93
5.1.4.1 Οι κλασσικές λειτουργίες του editor	93
5.1.4.2 Δουλεύοντας με τον SimEdit	94
5.1.5 Ο Compiler	95
5.1.6 Ο Linker	96
5.1.7 Ο Debugger σφαλμάτων SimDebug	96
5.1.8 Οι εντολές του περιβάλλοντος SimLab	97
5.2 Γράφοντας ένα μοντέλο προσομοίωσης στο περιβάλλον εργασίας SimLab	98
5.2.1 Κάνοντας το μοντέλο του κουρείου πιο πολύπλοκο	107
5.3 Κατανοώντας τη λειτουργία της Simgscript II.5	113
Παράρτημα 1ο: Οι εντολές του SimEdit	124
Παράρτημα 2ο: Οι εντολές του SimLab	127
Παράρτημα 3ο: Τελεστές σχέσεων	128
Παράρτημα 4ο: Συναρτήσεις κατανομών	129
Παράρτημα 5ο: Στατιστικές εντολές	131

ΚΕΦΑΛΑΙΟ 1ο ΠΡΟΣΟΜΟΙΩΣΗ

1.1 Εισαγωγή

Προβλήματα ή καταστάσεις που απασχολούν αυτούς που αποφασίζουν, και που παρουσιάζονται (ανήκουν) σε κοινωνικούς-οικονομικούς χώρους, είναι συνήθως αδύνατο να προσδιορισθούν επακριβώς. Με άλλα λόγια καταστάσεις που αναγνωρίζονται ως συστήματα ανθρώπινης δραστηριότητας –όπως στον χώρο της υγείας, σε επιχειρησιακούς και οικονομικούς χώρους, στις μεταφορές, στο γενικό πρόβλημα της ακτοπλοΐας κ.τ.λ.- είναι ασαφώς προσδιορισμένες κυρίως διότι ο κοινωνικός (ανθρώπινος) παράγοντας δεν ελέγχεται κατά ένα μεγάλο ποσοστό ούτε ακόμη στοχαστικά, δηλαδή ούτε με τη χρήση της στατιστικής.

Αποτέλεσμα αυτής της αναγνώρισης είναι η αποδοχή της άποψης ότι οι τεχνολογίες και οι μεθοδολογίες που χρησιμοποιούνται και χρησιμοποιούνται για την αντιμετώπιση απόλυτα προσδιορισμένων προβλημάτων, δεν μπορούν να χρησιμοποιηθούν και για την αντιμετώπιση προβλημάτων με ισχυρή συμμετοχή του κοινωνικού (ανθρώπινου) στοιχείου. Παραδοσιακά η αντιμετώπιση προβλημάτων, γενικά αλλά και ειδικότερα στα στάδια της εξεύρεσης της λύσης, εξορισμού καταφεύγει στην γνωστή ποσοτικοποίηση ή μαθηματικοποίησή τους. Η μαθηματικοποίηση όμως αυτή απαιτεί την ικανοποίηση μιας σειράς από αυστηρά καθορισμένες υποθέσεις και αξιώματα. Αυτό όμως σημαίνει ότι γίνονται μια σειρά από μη πραγματιστικές υποθέσεις με αποτέλεσμα συνήθως να θυσιάζεται η πραγματικότητα στον βώμο της επεξεργασιμότητας. Είναι λογικό να αναμένει κανείς ότι εάν η ευαισθησία του προβλήματος είναι μεγάλη τότε και οι λύσεις που προτείνονται μετά από μαθηματικοποιημένη επεξεργασία να μην ακολουθούνται, δηλαδή να μην αντιμετωπίζουν με επιτυχία το πρόβλημα.

Η αντιμετώπιση των πραγματικών κοινωνικο-οικονομικών προβλημάτων απαιτεί μεγάλη προσοχή και ευελιξία αρχικά για τη σωστή αναγνώριση και τον προσδιορισμό τους. Ο βασικός σκοπός του παρόντος κεφαλαίου δεν είναι να παρουσιάσει όλες τις θεωρίες, τις μεθοδολογίες και την τεχνολογία για την αντιμετώπιση αυτών των προβλημάτων αλλά να παρουσιάσει και να αιτιολογήσει γιατί επιλέγονται τα συστήματα υποστήριξης αποφάσεων και ειδικότερα η προσομοίωση σε ηλεκτρονικούς υπολογιστές για την αντιμετώπιση αυτών των προβλημάτων.

1.2 Συστήματα και Μοντέλα

Σε αυτή τη παράγραφο θα ορίσουμε και θα παρουσιάσουμε δύο βασικές έννοιες – κλειδιά, τις έννοιες του “Συστήματος” και του “Μοντέλου”. Αυτές οι έννοιες είναι απαραίτητες για τον καθορισμό ενός συγκεκριμένου επιπέδου επικοινωνίας με τον αναγνώστη. Επίσης οι έννοιες των συστημάτων και των μοντέλων είναι ουσιώδεις στην κατανόηση της μεθοδολογίας της προσομοίωσης.

1.2.1 Σύστημα

Οι περισσότεροι υποστηρικτές της συστημικής θεωρίας δέχονται ότι το σύστημα είναι ένα ενιαίο σύνολο που αποτελείται από αλληλένδετα μέρη, δηλαδή από στοιχεία, που έχουν διασυνδέσεις, αλληλεξαρτήσεις και αλληλεπιδράσεις. Κυρίως, όμως δέχονται ότι αυτό το σύνολο είναι κάτι περισσότερο από το άθροισμα των μερών του. Συγκεκριμένα ο Checkland αναφέρει το παράδειγμα που έδωσε ένας διδάσκων σ’ ένα φοιτητή. «Οπωσδήποτε εσύ είσαι κάτι περισσότερο από το άθροισμα των μερών σου, γιατί είσαι βλάκας». Και όπως σχολίασε ο Checkland «σίγουρα μπορούμε να βάλουμε αυτή την ταμπέλα στον φοιτητή, ενώ δεν μπορούμε να αποκαλέσουμε βλάκα τον αστράγαλο ή τον αγκώνα του».

Με βάση αυτό τον ορισμό μπορούμε να κάνουμε μερικές χρήσιμες παρατηρήσεις:

1. Η συμπεριφορά (ή οι ιδιότητες) καθενός στοιχείου του συνόλου έχει επίδραση πάνω στην συμπεριφορά (ή στις ιδιότητες) ολόκληρου του συνόλου.
2. Κανένα στοιχείο δεν επιδρά ανεξάρτητα πάνω σε όλο το σύστημα. Οι ιδιότητες και η συμπεριφορά κάθε συνιστώσας και ο τρόπος που επιδρά πάνω στο σύστημα εξαρτάται από τις ιδιότητες και τη συμπεριφορά των άλλων συνιστωσών.
3. Κάθε επί μέρους σύνολο στοιχείων του συστήματος έχει τις παραπάνω δύο ιδιότητες. Με άλλα λόγια κανένα στοιχείο του συστήματος δεν μπορεί να οργανωθεί σε τελείως ανεξάρτητες υποομάδες στοιχείων.

Το σύστημα είναι μια ανθρώπινη σύλληψη από μέρη που αλληλοσυνδέονται και αλληλεξαρτώνται μεταξύ τους, δηλ. βρίσκονται σε μια οργανωμένη τάξη. Και εάν μεν, αυτό το κατασκεύασμα είναι ένα τέχνημα, είναι κατανοητό. Στην περίπτωση όμως των κοινωνικών ή φυσικών φαινομένων, τι εννοείται; Το κοινωνικό ή φυσικό φαινόμενο είναι το σύστημα; Εδώ θα πρέπει να κατανοηθεί ότι το νοητικό κατασκεύασμα που καταρτίζεται προκειμένου να συλληφθεί το φαινόμενο, εάν βεβαίως το φαινόμενο αντιμετωπίζεται ως σύστημα, συνιστά το σύστημα. Δηλ. ο μελετητής είναι υποχρεωμένος, μελετώντας το όλο, να καθορίσει εκείνα τα μέρη που νομίζει ότι το συνιστούν από τη σκοπιά που ενδιαφέρεται να το μελετήσει, να συντάξει τα μέρη σε οργανωμένη τάξη, ώστε μεταξύ τους να βρίσκονται σε σχέση αλληλοσύνδεσης και αλληλεπίδρασης και τότε το σύνολο αυτών των μερών αποτελεί το σύστημα που θα μελετήσει. Αυτό το νοητικό κατασκεύασμα είναι στην ουσία το θεωρητικό πρότυπο (μοντέλο) της πραγματικότητας που επιθυμεί να μελετήσει ο επιστήμονας και αυτό είναι το σύστημα. Το τεχνητό πρότυπο (μοντέλο) που θα κατασκευάσει εκ των υστέρων

για να μελετήσει τη συμπεριφορά του θεωρητικού προτύπου, δηλ. του συστήματος είναι το μοντέλο που απεικονίζει το σύστημα.

Στις τεχνολογικές απόψεις περί συστημάτων, ο ορισμός του συστήματος ως προτύπου (μοντέλου) δεν κάνει την διάκριση μεταξύ θεωρητικού και εφαρμοσμένου προτύπου (μοντέλου), αλλά ονομάζει απ' ευθείας το σύστημα ως 'μια απεικόνιση (αντιστοιχία) των στοιχείων ενός συνόλου (είσοδος) σε ένα άλλο σύνολο (έξοδο)'.

Η διάκριση αυτή είναι πολύ σημαντική στη θεωρία των συστημάτων. Η υπέρβαση δηλαδή της σύγχυσης που έφερε η πρώιμη συστημική σκέψη ανάμεσα στο σύστημα ως έκφραση του πραγματικού κόσμου και στο σύστημα ως τρόπο σκέψης που έχει εφεύρει ο άνθρωπος για να ερμηνεύσει το πραγματικό κόσμο. Για το λόγο αυτό, πολλοί συστημικοί επιστήμονες προτείνουν να χρησιμοποιείται άλλη ορολογία για το πραγματικό φαινόμενο - σύστημα- και άλλη για την θεωρητική του σύλληψη -π.χ. Όλον-.

Τα όρια του συστήματος καθορίζονται κάθε φορά από τον ερευνητή-παρατηρητή. Κριτήριο για τον καθορισμό των ορίων είναι έως ποιο επίπεδο δέχεται ή επιτυγχάνει ο ερευνητής την ικανότητα του συστήματος για έλεγχο. Ότι τοποθετείται πέραν της περιοχής ελέγχου του συγκεκριμένου συστήματος, δηλ. πέραν των ορίων του, συνιστά το περιβάλλον αυτού. Το περιβάλλον του συστήματος εμπεριέχει εκείνες τις δυνάμεις οι οποίες, χωρίς να ανήκουν στο σύστημα, είναι ικανές να το επηρεάζουν. Είναι εκείνος ο χώρος ο οποίος παρέχει στο σύστημα τα στοιχεία εισόδου του υπό μορφή ύλης - ενέργειας - πληροφορίας και στον οποίο το σύστημα παρέχει τα τελικά του προϊόντα υπό μορφή ύλης - ενέργειας - πληροφορίας.

Η έννοια του συστήματος κατανοείται εάν γίνει πλήρως αντιληπτός ο ρόλος του παρατηρητή.

Παράδειγμα 1ο:

Υπάρχει μια γέφυρα. Εάν ο παρατηρητής είναι πολιτικός μηχανικός και θέλει να μελετήσει την αντοχή της γέφυρας με συστημικό τρόπο, θα εντάξει στο σύστημα την γέφυρα, από τεχνική άποψη και το φορτίο που δέχεται καθημερινώς αυτή (πεζοί και αυτοκίνητα). Εάν ο παρατηρητής είναι πολεοδόμος και μελετά την κυκλοφοριακή αναδιοργάνωση της περιοχής σε σχέση με την γέφυρα, θα εντάξει στο σύστημα τη γέφυρα (όχι από τεχνική άποψη, αλλά ως οδό διέλευσης της κυκλοφορίας), την κίνηση που εξυπηρετεί και τις παράπλευρες οδικές αρτηρίες που μαζί με την γέφυρα εξυπηρετούν την κίνηση της δεδομένης περιοχής.

Παράδειγμα 2ο:

Έστω ότι έχουμε τον ασθενή X ο οποίος υποφέρει από συχνούς πονοκεφάλους. Ο παθολόγος Ψ, ο οποίος τον εξετάζει, θεωρεί ως "σύστημα X" τον οργανισμό του ασθενή. Εξετάζει τα όργανα του σώματός του και τελικά καταλήγει ότι ο ασθενής υποφέρει από πονοκεφάλους εντάσεως ή άγχους. Του συστήνει να μην στενοχωριέται και όταν έχει πονοκέφαλο να παίρνει ασπιρίνες. Κατόπιν αυτού, ο X επισκέπτεται τον ψυχολόγο Ω. Ο Ω προσπαθεί να εντοπίσει τις αιτίες του

άγχους γι' αυτό “στο σύστημα X” εντάσσει την οικογένεια του X και τον επαγγελματικό του χώρο. Μελετά τις αλληλεπιδράσεις αυτών των χώρων με την προσωπικότητα του X και καταλήγει ότι η αιτία του άγχους είναι η επιβάρυνση του X με καθήκοντα που υπερβαίνουν τις ψυχικές του δυνάμεις.

Παράδειγμα 3ο:

Έστω ότι έχουμε ένα σπίτι. Εάν ο παρατηρητής είναι ένας αρχιτέκτονας θεωρεί το σπίτι ως κύριο σύστημα και τους ενοίκους του, τα συστήματα φωτισμού, θερμάνσεως, ύδρευσης, εξαερισμού, κ.λπ. ως υποσυστήματα του κυρίου συστήματος. Εάν ο παρατηρητής είναι ένας μηχανολόγος μηχανικός θεωρεί τα συστήματα φωτισμού, θερμάνσεως, ύδρευσης, εξαερισμού, κ.λπ. ως το κύριο σύστημα, το δε υπόλοιπο σπίτι ως το περιβάλλον του συστήματός του. Τέλος εάν ο παρατηρητής είναι ένας ερευνητής κοινωνικής ψυχολογίας, που ασχολείται με την οικογένεια, θεωρεί αυτή ως σύστημα, το σπίτι ως περιβάλλον του συστήματός του και τα συστήματα φωτισμού, θερμάνσεως, ύδρευσης, εξαερισμού, κ.λπ. ως συστήματα που ανήκουν στο ίδιο περιβάλλον.

Τα συστήματα μπορούν να ταξινομηθούν σε διάφορες κατηγορίες ανάλογα με τη σχέση τους με το περιβάλλον, με τα γεγονότα, με την πολυπλοκότητά τους, με τη συμπεριφορά τους κ.λ.π. Έτσι έχουμε:

1. Σε σχέση με τα στοιχεία τους τα συστήματα είναι *φυσικά (physical systems)* αν υπάρχουν αυτούσια στη φύση (π.χ. ο άνθρωπος, ένα οικοσύστημα, το ηλιακό σύστημα, κ.λ.π.) ή *σχεδιασμένα (designed systems)* αν τα κατασκευάζει ο άνθρωπος. Στη δεύτερη αυτή κατηγορία έχουμε:
 - α) *Θεωρητικά* συστήματα (π.χ. συστήματα αρίθμησης).
 - β) *Τεχνολογικά* συστήματα (π.χ. ένα ασανσέρ, ο Η.Υ.).
 - γ) *Κοινωνικά* συστήματα με στοιχεία ανθρώπους και τις διαπροσωπικές τους σχέσεις.
 - δ) *Συστήματα Ανθρώπινης Δραστηριότητας* όπου οι άνθρωποι εκτελούν δραστηριότητες για κάποιο σκοπό (π.χ. ένα Πληροφοριακό Σύστημα). Τα ΣΑΔ εμπεριέχουν ένα κοινωνικό σύστημα (π.χ. ομάδα απασχολούμενων) αλλά επί πλέον ένα ιδεατικό σύστημα δραστηριοτήτων, το οποίο εκφράζει το “τι κάνει” το κοινωνικό σύστημα. Δηλαδή: ΣΑΔ = Σύστημα Δραστηριοτήτων + Κοινωνικό Σύστημα.
2. Σε σχέση με το περιβάλλον τα συστήματα είναι *ανοικτά (open)* ή *κλειστά (closed)* ανάλογα με το εάν υπάρχει ή όχι αλληλοεπίδραση μεταξύ συστήματος και περιβάλλοντος.
3. Σε σχέση με τη συμπεριφορά τους μπορούν να ταξινομηθούν ακολουθώντας μια πορεία που ξεκινάει από απλές μορφές συμπεριφοράς και καταλήγει στις σύνθετες:
 - α) *Σταθερής κατάστασης (state maintaining system)* είναι το σύστημα που μπορεί να αντιδρά κατά ένα και μόνο τρόπο σε ένα εσωτερικό ή εξωτερικό γεγονός, αλλά αντιδρά διαφορετικά σε διαφορετικά εσωτερικά ή εξωτερικά γεγονότα, όμως οι αντιδράσεις αυτές έχουν πάντα το ίδιο αποτέλεσμα. Παράδειγμα: το σύστημα θερμάνσεως (με θερμοστάτη) των πολυκατοικιών.

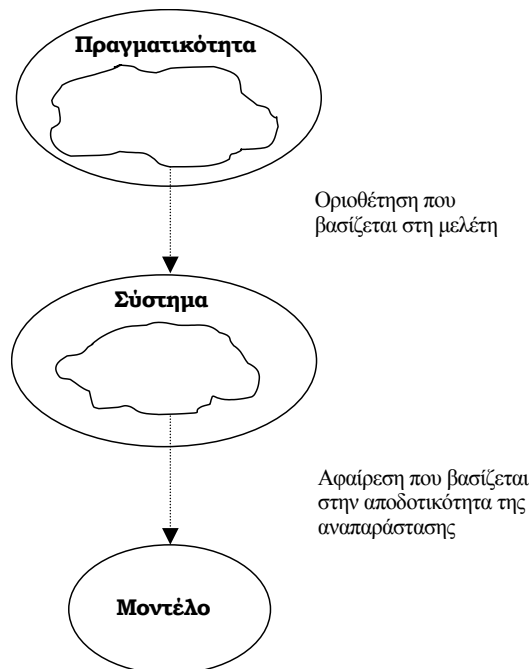
- β) *Συγκεκριμένου στόχου (goal seeking)* είναι το σύστημα που αποκρίνεται διαφορετικά σε ένα ή περισσότερα εσωτερικά ή εξωτερικά γεγονότα μέχρις ότου επιτύχει μια συγκεκριμένη κατάσταση. Ένα σύστημα της κατηγορίας αυτής έχει επιλογή συμπεριφοράς και διαθέτει “μνήμη”. Παράδειγμα: συστήματα με αυτόματους πιλότους.
- γ) *Σύστημα πολλών στόχων (multi-goal seeking)* είναι αυτό που αποκρίνεται διαφορετικά σε ένα ή περισσότερα διαφορετικά εσωτερικά ή εξωτερικά γεγονότα και έχει την ιδιότητα να αναζητά διαφορετικούς στόχους. Οι στόχοι μπορεί να έχουν ή όχι μια κοινή ιδιότητα. Ο στόχος όμως που πραγματοποιείται εξαρτάται από κάποιες αρχικές συνθήκες. Τα συστήματα της κατηγορίας αυτής δεν εκλέγουν το στόχο τους, γιατί αυτός είναι καθορισμένος από μερικές αρχικές συνθήκες, εκλέγουν όμως τα μέσα επιδιώξεως του στόχου τους. Παράδειγμα: Ένα σύστημα Η/Υ που έχει προγραμματισθεί να παίζει περισσότερα του ενός παιχνίδια.
- δ) *Σύστημα που εκδηλώνει πρόθεση (purposeful system)* είναι το σύστημα εκείνο, που παράγει το ίδιο αποτέλεσμα με διάφορους τρόπους στην ίδια (εσωτερική ή εξωτερική) κατάσταση, και μπορεί και παράγει διάφορα αποτελέσματα στην ίδια ή σε διαφορετικές καταστάσεις. Έτσι ένα σύστημα της κατηγορίας αυτής έχει την δυνατότητα να αλλάξει τους στόχους κάτω από σταθερούς περιορισμούς, να επιλέξει τους σκοπούς του και τα μέσα επιδιώξεών του. Είναι συστήματα που εκδηλώνουν θέληση. Παράδειγμα: ο άνθρωπος.
4. Τα συστήματα διακρίνονται ως προς τα γεγονότα στις εξής κατηγορίες:
- α) Στα *στατικά* (ή μιας καταστάσεως) συστήματα, που είναι εκείνα στα οποία η αλληλεξάρτηση των στοιχείων τους είναι στατική, δηλαδή δεν συμβαίνουν γεγονότα.
- β) Στα *δυναμικά*, που είναι εκείνα στα οποία συμβαίνουν γεγονότα και αλλάζει η κατάσταση του συστήματος.
- γ) Στα *ομοιοστατικά*, που είναι στατικά συστήματα ως ολότητα, αλλά τόσο τα στοιχεία τους όσο και το περιβάλλον τους είναι δυναμικά. Ένα ομοιοστατικό σύστημα δηλαδή είναι ένα σύστημα που διατηρεί σταθερή την κατάστασή του σε ένα μεταβαλλόμενο περιβάλλον με τη βοήθεια εσωτερικών μετατροπών.
5. Αν ως κριτήριο χρησιμοποιήσουμε το βαθμό πρόβλεψης της συμπεριφοράς του συστήματος τότε έχουμε
- α) *αιτιοκρατικά (deterministic)* συστήματα, όταν η συμπεριφορά του συστήματος σε διαφορετικές καταστάσεις μπορεί να προβλεφθεί με ακρίβεια, και
- β) *πιθανολογικά (probabilistic)* συστήματα όταν καμία πρόβλεψη δεν μπορεί να γίνει με ακρίβεια ως προς τη συμπεριφορά του συστήματος σε διαφορετικές καταστάσεις.

1.2.2 Μοντέλο

Πρόκειται για το τεχνητό πρότυπο (μοντέλο) του συστήματος. Για να μπορέσει κάποιος να αποδώσει την περιγραφή ενός συστήματος, ή να το αναλύσει, πρέπει πρώτα να εκφράσει το σύστημα με κάποια μορφής αναπαράστασης. Αυτή η αναπαράσταση ονομάζεται μοντέλο. Για παράδειγμα, ένας ποιητής μπορεί να δει ένα δέντρο (σύστημα) και να το αναπαραστήσει με λέξεις που να το περιγράφουν (λεκτική αναπαράσταση). Ένας ζωγράφος μπορεί να δει το ίδιο δένδρο και να το εκφράσει με μια ζωγραφιά (εικονογραφική αναπαράσταση). Αν και η ζωγραφιά και το ποίημα είναι πολύ διαφορετικά μεταξύ τους, είναι μοντέλα του ίδιου συστήματος.

Ο Wilson ορίζει το μοντέλο ως εξής: 'Είναι η σαφής διερμηνευση της αντίληψης που έχει κάποιος για μια κατάσταση. Μπορεί να εκφρασθεί με μαθηματικά σύμβολα ή λέξεις, αλλά ουσιαστικά είναι μια περιγραφή στοιχείων και των σχέσεών τους. Μπορεί να είναι δεοντολογική ή αναπαραστατική, αλλά πάνω απ' όλα πρέπει να είναι χρήσιμη'. Δηλαδή το μοντέλο είναι μια καρικατούρα της πραγματικότητας.

Τα μοντέλα σπάνια αποδίδουν όλα τα συμβαίνοντα σχετικά με το σύστημα που αναπαριστούν, άρα είναι στην πράξη αφαιρέσεις του συστήματος. Όμως, ορισμένες μορφές αναπαραστάσεων είναι περισσότερο αποτελεσματικές στο να αποδώσουν ορισμένα συμβαίνοντα. Για παράδειγμα, η ζωγραφιά ενός δέντρου μπορεί να περιγράψει καλύτερα τα χρώματα του σκηνικού, αλλά το ποίημα μπορεί να περιγράψει καλύτερα την κίνηση των φύλλων και το κελάηδημα των πουλιών. Στο σχήμα 1.1 παρουσιάζονται οι σχέσεις ανάμεσα στην πραγματικότητα, το σύστημα και το μοντέλο.



Σχήμα 1.1: Σχέση ανάμεσα στην πραγματικότητα, το σύστημα και το μοντέλο

Τα μοντέλα, δηλαδή, είναι συνήθως πολύ πιο απλά από την πραγματικότητα, αλλά πρέπει να εξηγούν και να προβλέπουν ορισμένες καταστάσεις της πραγματικότητας. Αυτό που πρέπει να τονιστεί είναι ότι τα μοντέλα δεν είναι μια ερμηνεία ή αναπαράσταση της πραγματικότητας αλλά της αντίληψης, ή της σύλληψης που κάνει κάποιος για την πραγματικότητα. Είναι δηλαδή διερμηνευση ή αναπαράσταση του συστήματος όπως το έχει συλλάβει στο μυαλό του για να το μελετήσει ο παρατηρητής.

Μερικοί θετικοί λόγοι για την χρησιμοποίηση των μοντέλων είναι:

1. *Οργάνωση θεωρητικών ‘πιστεύω’ και εμπειρικών παρατηρήσεων για ένα σύστημα.* Με την κατασκευή ενός μοντέλου είναι πιο εύκολο να μεταδοθούν οι ιδέες για κάποιο σύστημα απ’ ότι με την περιγραφή του συστήματος. Για παράδειγμα, ένας αρχιτέκτονας κατασκευάζει μια μακέτα του κτιρίου που έχει σχεδιάσει και μ’ αυτήν δίνει πολύ περισσότερες πληροφορίες στον πελάτη απ’ ότι με λεκτική περιγραφή ή αρχιτεκτονικά σχέδια.
2. *Οδηγούν σε καλύτερη κατανόηση του συστήματος.* Το μοντέλο είναι συχνά πολύ πιο απλό στην κατανόηση από το ίδιο το σύστημα, γιατί κατά την κατασκευή του μοντέλου διατηρούνται μόνο τα χαρακτηριστικά του συστήματος που ενδιαφέρουν στη συγκεκριμένη μελέτη. Με τον τρόπο αυτό, ο μελετητής δεν “χάνεται” στις λεπτομέρειες του συστήματος αλλά επικεντρώνει την προσοχή του μόνο στα σημαντικά στοιχεία.
3. *Αντιμετωπίζουν το πρόβλημα της αδυναμίας πρόσβασης.* Μερικές φορές η πρόσβαση στο πραγματικό σύστημα είναι αδύνατη ή επικίνδυνη. Κατασκευάζοντας ένα μοντέλο είναι δυνατόν να μελετηθεί το σύστημα χωρίς να κινδυνεύσει ο μελετητής ή το ίδιο το σύστημα.
4. *Αποτελούν εργαλείο πρόβλεψης.* Ορισμένα συστήματα παρουσιάζουν πολύ αργές μεταβολές της κατάστασής τους με αποτέλεσμα να είναι αδύνατη η πρόβλεψη της συμπεριφοράς τους για ένα μακρύ χρονικό διάστημα. Κατασκευάζοντας ένα μοντέλο του συστήματος είναι δυνατή η επιτάχυνση των χρονικών μεταβολών έτσι ώστε να προβλεφθεί η μελλοντική συμπεριφορά του πραγματικού συστήματος.
5. *Βοηθούν στην εκπαίδευση.* Με την κατασκευή ενός μοντέλου είναι δυνατόν να εκπαιδεύουν χειριστές χωρίς τον κίνδυνο καταστροφών από λάθος των εκπαιδευόμενων. Είναι επίσης δυνατόν να εκπαιδεύουν οι χειριστές ενός συστήματος το οποίο δεν έχει κατασκευασθεί ακόμη.
6. *Επιτρέπουν τον έλεγχο αλλαγών με ελάχιστο κόστος.* Με την κατασκευή ενός μοντέλου είναι δυνατό να ελεγχθεί η συμπεριφορά του συστήματος για διάφορες τιμές των παραμέτρων του με ελάχιστο κόστος. Από τη μελέτη του μοντέλου που έχει κατασκευασθεί διαπιστώνεται ο αποδοτικότερος συνδυασμός παραμέτρων, και στη συνέχεια, οι παράμετροι αυτοί εφαρμόζονται στο πραγματικό σύστημα.

Ο Ackoff (1962) προτείνει τρεις τύπους μοντέλων:

1. *Εικονικά* μοντέλα (φωτογραφίες, σχέδια, μακέτες): Είναι μια αναπαράσταση, σε σμίκρυνση της πραγματικότητας π.χ. η μακέτα ενός υπό κατασκευή κτιρίου.
2. *Αναλογικά* μοντέλα: Αναπαριστούν την συμπεριφορά της πραγματικής κατάστασης π.χ. ροή νερού ανάμεσα από μικρές πλαστικές δεξαμενές σε θερμοκρασία δωματίου, για να ερευνηθεί η συμπεριφορά του τετηγμένου γυαλιού σε κάμινους με θερμοκρασία 1000°C.
3. *Αναλυτικά* μοντέλα: Μαθηματικές ή λογικές σχέσεις οι οποίες αναπαριστούν φυσικούς νόμους που διέπουν την υπό μελέτη κατάσταση. Τα αναλογικά μοντέλα περιέχουν αριθμούς, γράμματα και σύμβολα τα οποία εκφράζουν τις σχέσεις μεταξύ των μεταβλητών του μοντέλου καθώς και τις ίδιες τις μεταβλητές. Συνήθως προηγούνται των αναλογικών μοντέλων.

Τα αναλυτικά μοντέλα χωρίζονται συνήθως σε 4 κατηγορίες αναλόγως της εξαρτήσεώς τους από το χρόνο (στατικά - δυναμικά) και της εξαρτήσεώς τους από φυσικούς νόμους ή τυχαίους παράγοντες (ντετερμινιστικά - μη ντετερμινιστικά). Βάσει αυτών των διακρίσεων τα αναλυτικά μοντέλα ταξινομούνται σε:

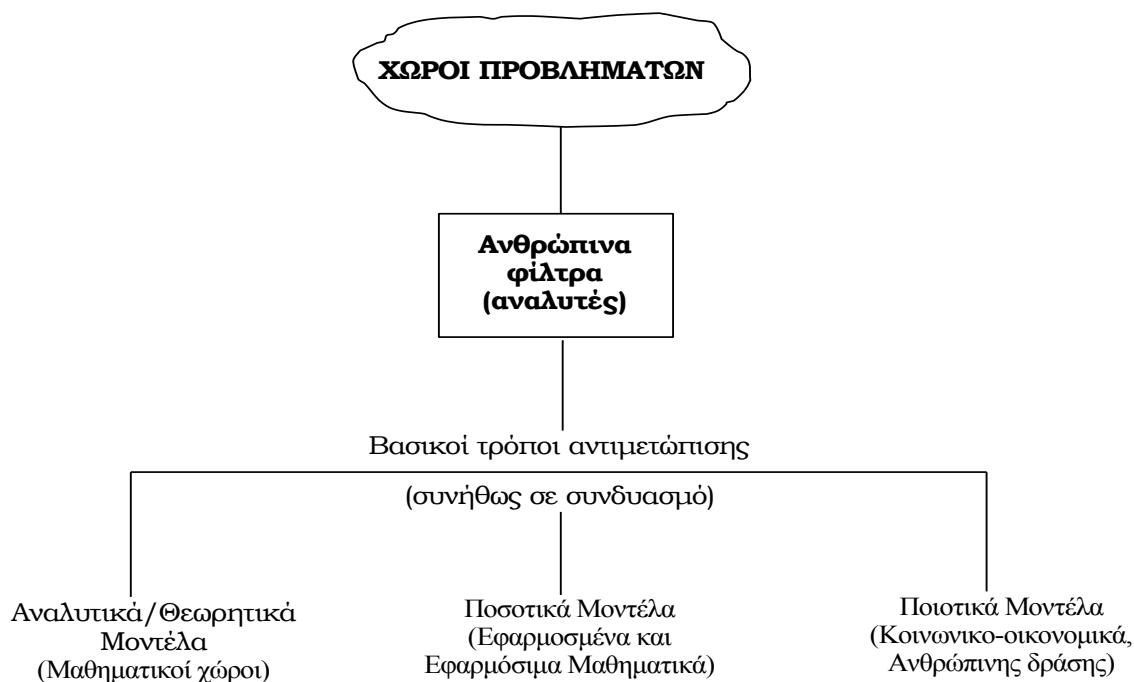
	Στατικά	Δυναμικά
Ντετερμινιστικά	Αλγεβρικές σχέσεις	Διαφορικές εξισώσεις
Στοχαστικά	Στατιστικές και πιθανολογικές σχέσεις	Προσομοίωση

Εκτός από τις παραπάνω κατηγορίες μοντέλου που προτείνει ο Ackoff, νεώτεροι ερευνητές (Checkland, Wilson) αναφέρουν και έναν τέταρτο τύπο μοντέλων, τα εννοιολογικά (conceptual) μοντέλα.

4. *Εννοιολογικά* μοντέλα: Πρόκειται για γραφικές - συμβολικές αναπαραστάσεις, οι οποίες αναπαριστούν τα ποιοτικά χαρακτηριστικά μιας κατάστασης, όπως αυτή εννοείται από τον παρατηρητή. Τα εννοιολογικά μοντέλα μπορούν να είναι πολύ ευρέα, όπως π.χ. μια επιστήμη ή πολύ περιορισμένα, που να εκφράζουν μια ιδέα. Από το 1970 όμως και έπειτα, χρησιμοποιούνται εκτεταμένα στα συστήματα ανθρώπινης δράσης (ΣΑΔ). Τα ΣΑΔ εκφράζουν μια ένσκηπη ανθρώπινη δραστηριότητα και ερευνούν το “τι κάνει” ένα ανθρώπινο σύστημα. Το εννοιολογικό μοντέλο αυτού του συστήματος εκφράζει τις πράξεις του συστήματος στη φάση που ερευνούμε.

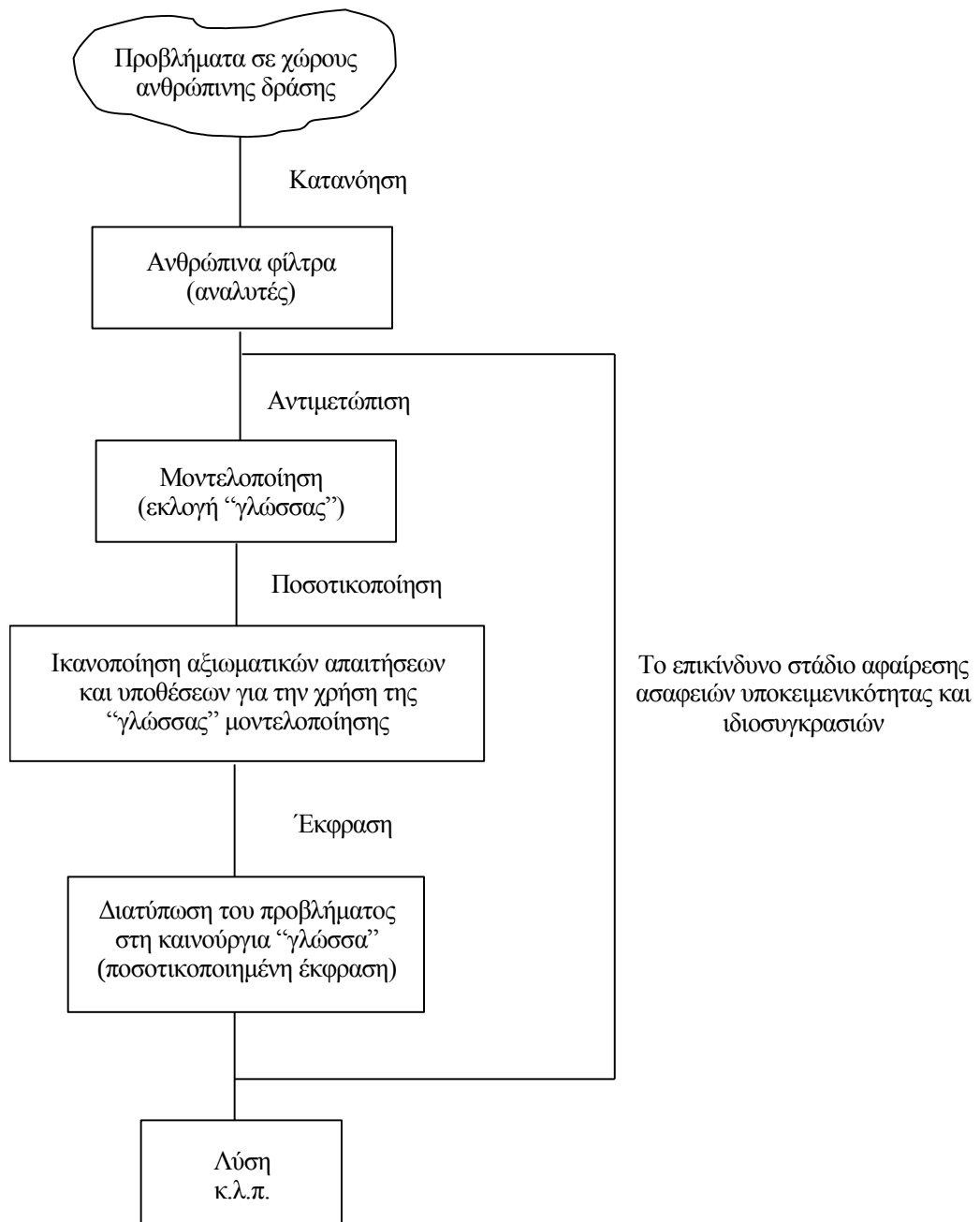
1.3 Αντιμετώπιση Κοινωνικο-Οικονομικών Προβλημάτων

Ο ρόλος της ανθρώπινης υποκειμενικότητας και ιδιοσυγκρασίας στην αντίληψη και διατύπωση των προβλημάτων, καθώς και στην αντιμετώπισή τους (αναλυτές, αποφασίζοντες) είναι πολύ σημαντικός και με κανένα τρόπο δεν πρέπει να παραβλέπεται. Η παρουσία του αναλυτή ουσιαστικά σαν ένα ανθρώπινο φίλτρο, όπως φαίνεται στο σχήμα 1.2, έχει σαν σκοπό να τονίσει τη σημασία του κοινωνικού στοιχείου στην αντιμετώπιση των κοινωνικο-οικονομικών προβλημάτων. Ο διαχωρισμός των μοντέλων στο ίδιο διάγραμμα σε θεωρητικά, ποσοτικά και ποιοτικά μοντέλα είναι χαρακτηριστικός. Συνήθως τα ποιοτικά συστήματα / μοντέλα περιέχουν ποσοτικά και ίσως θεωρητικά υποσυστήματα / μοντέλα. Το αντίθετο φυσικά δεν συμβαίνει.



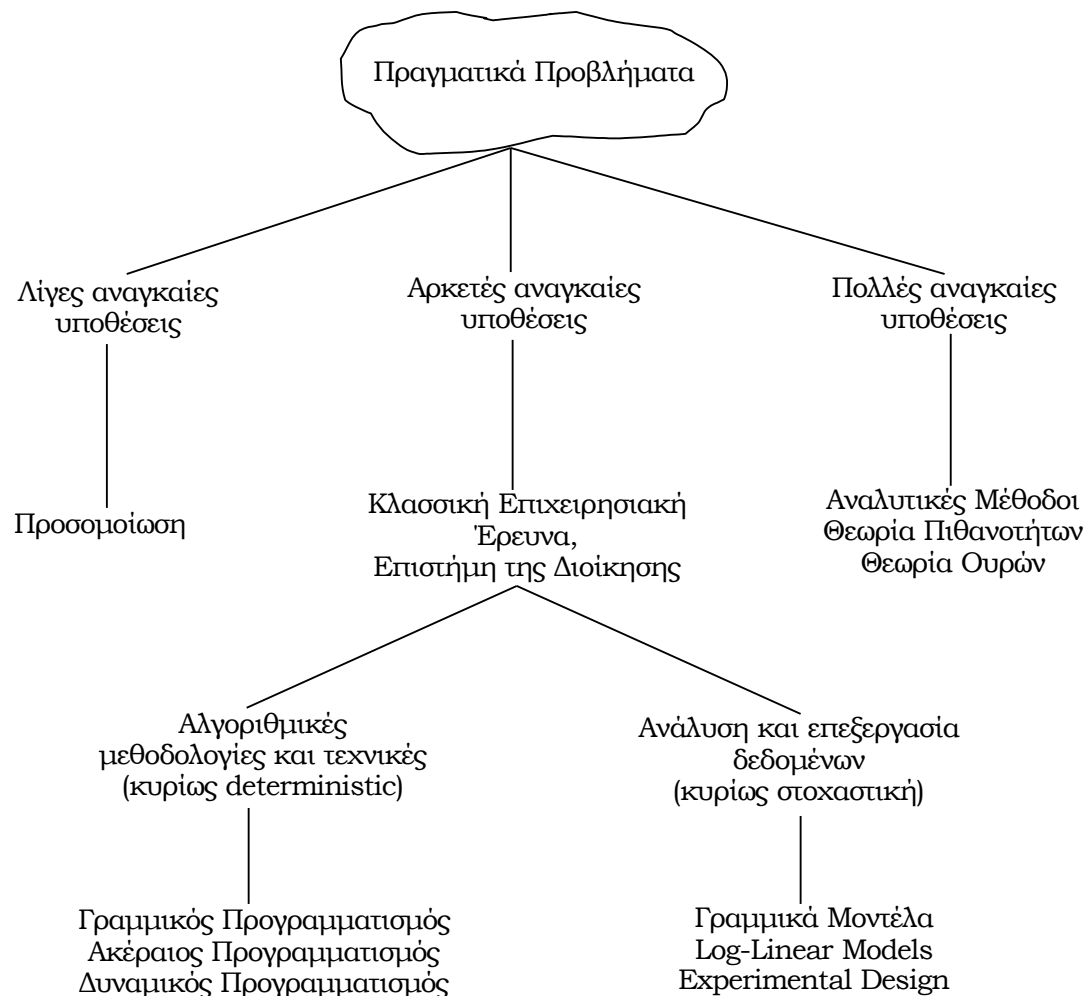
Σχήμα 1.2

Το σχήμα 1.3 δίνει παραστατικά μια γενική διαδικασία αντιμετώπισης κοινωνικο-οικονομικών προβλημάτων. Ξεκινώντας την αντιμετώπιση του πραγματικού προβλήματος στο χώρο ανθρώπινης δραστηριότητας, οι αναλυτές μετά από μια γενική διαδικασία κατανόησης, αποφασίζουν με βάση αυτή την κατανόηση, πως θα αντιμετωπίσουν το πρόβλημα. Η ανάπτυξη μοντέλου ή μοντέλων στο στάδιο αυτό είναι ουσιαστικά η εισαγωγή σε μια διαδικασία ποσοτικοποίησης, δηλαδή μετατροπής του προβλήματος από γενικό και αόριστο σε επεξεργάσιμο. Η διαδικασία αυτή αναφέρεται ως “επικίνδυνη” στο σχήμα 1.3 λόγω του γεγονότος που αναφέρθηκε και προηγουμένως, ότι δηλαδή για να είναι επεξεργάσιμο ένα μοντέλο ενός πραγματικού προβλήματος πρέπει να ικανοποιεί πολλές περιοριστικές υποθέσεις και αξιώματα. Αυτό φυσικά σημαίνει ότι ουσιαστικά αφαιρείται αρκετή “πραγματικότητα” από το πρόβλημα.



Σχήμα 1.3

Τέλος το σχήμα 1.4 δίνει παραστατικά τις μεθόδους και τις τεχνικές αντιμετώπισης πραγματικών προβλημάτων ως προς τη λύση τους. Ο διαχωρισμός τους γίνεται ενδεικτικά σε τρεις κατηγορίες με βάση τον αριθμό των αναγκαίων υποθέσεων που χρησιμοποιούνται στη λύση των αντίστοιχων προβλημάτων.



Σχήμα 1.4

Η προσομοίωση, η οποία εμφανίζεται σαν η μέθοδος με τις λιγότερες αναγκαίες υποθέσεις για την αντιμετώπιση κοινωνικο-οικονομικών προβλημάτων, παρουσιάζεται λεπτομερώς στα επόμενα κεφάλαια που ακολουθούν. Είναι όμως γεγονός ότι για να εφαρμοστεί η προσομοίωση χρειάζεται να υπάρχει ένα σύστημα – μοντέλο προκαθορισμένο σε συγκεκριμένο επεξεργάσιμο επίπεδο. Απλώς οι υποθέσεις που απαιτούνται για να εκφρασθεί και να αναπτυχθεί αυτό το μοντέλο για προσομοίωση είναι σαφώς λιγότερες από τις υποθέσεις που πρέπει να ικανοποιηθούν σε άλλες περιπτώσεις όπου επιλέγονται άλλες μέθοδοι και τεχνικές. Ένα χαρακτηριστικό παράδειγμα αναγκαίας υπόθεσης για την ανάπτυξη μοντέλων προσομοίωσης σε κοινωνικο-οικονομικά προβλήματα είναι η υπόθεση της στοχαστικότητας, δηλαδή η υπόθεση ότι η έννοια του “τυχαίου” αντιμετωπίζεται στα μοντέλα της προσομοίωσης με την χρήση της στατιστικής.

1.4 Συστήματα Υποστήριξης Αποφάσεων

Οι επιστημονικοί χώροι που ασχολούνται άμεσα ή έμμεσα με τις “διαδικασίες αποφάσεων”, ειδικά σε περιβάλλοντα (συστήματα) ανθρώπινης δραστηριότητας, αποτελούνται κυρίως από τους κλασσικούς χώρους της επιχειρησιακής έρευνας, της διοίκησης επιχειρήσεων, της στατιστικής, της θεωρία συστημάτων και των μαθηματικών. Οι θεωρίες, οι μεθοδολογίες και οι τεχνολογίες οι οποίες βασίζονται και προέρχονται από αυτούς τους επιστημονικούς χώρους, δίνουν θεωρητικά λύσεις σε προβλήματα προσφέροντας έτσι την “απόφαση – λύση” στους αποφασίζοντες.

Στην πραγματικότητα όμως τα πράγματα είναι φυσικά διαφορετικά και αναγκαστικά αναφέρεται κανείς σε συστήματα “υποστήριξης αποφάσεων”. Το πρόβλημα ή η προβληματική κατάσταση που απασχολεί αυτούς που πρέπει να αποφασίσουν, ζητά μία ή περισσότερες λύσεις οι οποίες βελτιστοποιούν την κατάσταση με δεδομένους πάντα τους περιορισμούς.

Προβλήματα που εμφανίζονται σε συστήματα ανθρώπινων δραστηριοτήτων, όπου το ανθρώπινο (κοινωνικό) στοιχείο κυριαρχεί, περιέχουν συνήθως πολύ μεγάλο αριθμό παραμέτρων και μεταβλητών που επιδρούν σε αυτά. Στην πράξη οι παράμετροι και οι μεταβλητές εκφράζονται σε φυσική γλώσσα όπου εξ’ ορισμού περιλαμβάνονται ασάφειες, αντιφάσεις και παράδοξα. Παρά το γεγονός ότι έχουν αναπτυχθεί τα τελευταία χρόνια θεωρίες και μεθοδολογίες για την αντιμετώπιση (όχι αναγκαστικά ποσοτικοποιήσεις) αυτών των προβλημάτων, διαφαίνεται ότι συγκεκριμένα, αυτόνομα εργαλεία που θα δίνουν τη “λύση” σε τέτοιες πολύπλοκες καταστάσεις δεν είναι δυνατόν να υπάρξουν. Επίσης είναι δεδομένο ότι όσο περιορισμένου εύρους και αν είναι κάποια κοινωνικο-οικονομικά προβλήματα, διατηρούν σημαντική πολυπλοκότητα έτσι ώστε να μην είναι δυνατόν να “χωρέσουν” σε συγκεκριμένα θεωρητικά περιβάλλοντα ώστε να λυθούν αυτόνομα.

Το συμπέρασμα που βγαίνει από τα παραπάνω είναι ότι τα πραγματικά κοινωνικο-οικονομικά (ανθρώπινης δραστηριότητας) προβλήματα εκτός ότι αναγκαστικά αντιμετωπίζονται διεπιστημονικά, δεν είναι δυνατόν να “λυθούν” από εργαλεία που ανήκουν σε έναν ή περισσότερους θεωρητικούς χώρους χωρίς τη συμμετοχή της ανθρώπινης “διαίσθησης” των αποφασίζόντων.

Ο όρος συστήματα υποστήριξης αποφάσεων είναι αποτέλεσμα αυτής της αποδοχής για την αντιμετώπιση των πραγματικών κοινωνικο-οικονομικών προβλημάτων. Η σημασία του όρου είναι ότι οι διαδικασίες αποφάσεων που αφορούν την λύση τέτοιων προβλημάτων βασίζονται σε ανθρώπους αποφασίζοντες, οι οποίοι ακολουθούν συγκεκριμένες γενικές μεθοδολογίες διαδικασιών αποφάσεων, και οι οποίοι υποβοηθούνται (υποστηρίζονται) από διάφορες άλλες θεωρίες, μεθοδολογίες και τεχνολογίες στην διαδικασία αυτή.

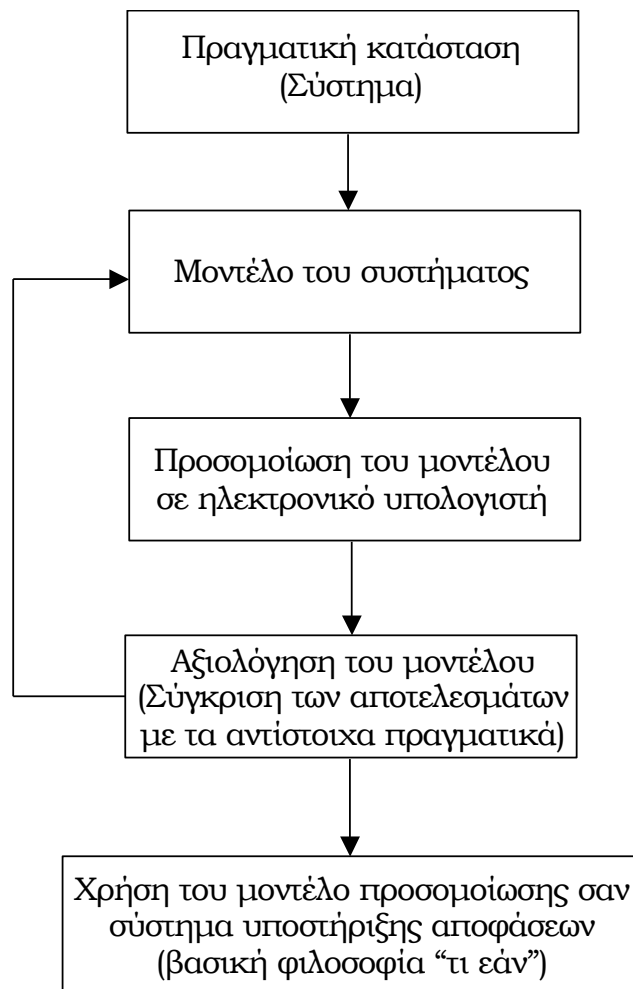
Σαν ένα τυπικό παράδειγμα συστήματος υποστήριξης αποφάσεων μπορεί να θεωρηθεί η προσομοίωση με την χρήση ηλεκτρονικών υπολογιστών.

1.5 Τι είναι Προσομοίωση?

Το εννοιολογικό μοντέλο μιας κατάστασης (κοινωνικο-οικονομικού προβλήματος) είναι συνήθως, όπως έχει ήδη αναφερθεί, αρκετά πολύπλοκο για να μπορεί να ελεγχθεί από έναν ολοκληρωμένο θεωρητικό χώρο.

Διάφορες συστημικές θεωρίες και μεθοδολογίες χρησιμοποιούνται στην ανάλυση γενικών εννοιολογικών μοντέλων, με σκοπό να μετασχηματιστούν αυτά σε πιο συγκεκριμένα και επεξεργάσιμα μοντέλα. Προϊόντα αυτών των αναλύσεων είναι μοντέλα υποσυστημάτων του γενικότερου προβλήματος εκφρασμένα σε εκείνο το επίπεδο περιγραφής που προσφέρεται ιδιαίτερα για την ανάπτυξη μοντέλων προσομοίωσης σε ηλεκτρονικούς υπολογιστές (computer simulation).

Με απλά λόγια προσομοίωση είναι η ανάπτυξη προγραμμάτων σε ηλεκτρονικό υπολογιστή τα οποία “μιμούνται” τις πραγματικές καταστάσεις δηλαδή η έκφραση των μοντέλων προσομοίωσης, που αναπαριστούν τις πραγματικές καταστάσεις, με προγράμματα σε ηλεκτρονικό υπολογιστή. Στο σχήμα 1.5, που ακολουθεί, δίνεται μια απλή περιγραφή της διαδικασίας ανάπτυξης μοντέλων προσομοίωσης.



Σχήμα 1.5

Ένα πολύ σημαντικό στάδιο στη διαδικασία ανάπτυξης είναι το στάδιο της αξιολόγησης (validation), όπου η συμπεριφορά του μοντέλου ελέγχεται έναντι των πραγματικών καταστάσεων τις οποίες “μιμείται”. Ο κλασσικός τρόπος ελέγχου των αποτελεσμάτων του μοντέλου είναι ο στατιστικός έλεγχος, δηλαδή η στατιστική σύγκριση των αποτελεσμάτων με τα αντίστοιχα δεδομένα που έχουν παρατηρηθεί στη πραγματικότητα ή γενικά με τα δεδομένα με τα οποία θεωρείται ότι πρέπει να συγκριθούν.

Η σκοπιμότητα ενός μοντέλου προσομοίωσης είναι η ύπαρξη της δυνατότητας ανάλυσης και μελέτης διαφόρων εναλλακτικών λύσεων για την αξιολόγησή τους ως προς εφαρμοσιμότητά τους.

Παραδείγματος χάριν, στην περίπτωση της προβληματικής κατάστασης της ακτοπλοϊκής συγκοινωνίας στο Αιγαίο, οι διάφορες μεταβλητές και παράμετροι όπως ο αριθμός των επιβατών, ο αριθμός και ο τύπος των αυτοκινήτων, ο καιρός που επικρατεί ανάλογα με τις εποχές, το κόστος λειτουργίας κ.λ.π. λαμβάνονται υπόψη από το μοντέλο προσομοίωσης. Η “απομίμηση” των διάφορων ακτοπλοϊκών καταστάσεων βασίζεται στις διάφορες τιμές των μεταβλητών και των παραμέτρων ανάλογα με τα σενάρια που εξετάζονται. Άλλες σημαντικές μεταβλητές – παράμετροι που μπορούν να θεωρηθούν από το μοντέλο της προσομοίωσης στην περίπτωση των ακτοπλοϊκών συγκοινωνιών είναι οι βλάβες των πλοίων, οι ματαιώσεις δρομολογίων λόγω καιρού, η υποχρεωτική κάλυψη δρομολογίων, η ακινησία πλοίων για διάφορους λόγους κ.λ.π.

Για συγκεκριμένα σενάρια με δεδομένους, παρατηρούμενους ή προβλεπόμενους τρόπους “συμπεριφοράς” των μεταβλητών και των παραμέτρων, το μοντέλο προσομοίωσης παρέχει μετρήσεις αποτελεσματικότητας και απόδοσής του ανάλογα με διάφορα κριτήρια. Με άλλα λόγια δίνεται η δυνατότητα να μελετηθούν οι θετικές και οι αρνητικές επιπτώσεις που θα έχει η εφαρμογή διαφόρων σεναρίων έτσι ώστε οι αποφασίζοντες να μπορούν να εκτιμήσουν και να αποφασίσουν σε σχετικά ελάχιστο χρόνο.

1.6 Γιατί Προσομοίωση?

Όπως έχει αναφερθεί, οι υποθέσεις που απαιτούνται για να εκφρασθεί και να αναπτυχθεί ένα μοντέλο προσομοίωσης είναι σαφώς λιγότερες από τις υποθέσεις που πρέπει να ικανοποιηθούν σε άλλες περιπτώσεις όπου επιλέγονται άλλες μέθοδοι και τεχνικές. Έτσι η προσομοίωση εμφανίζεται ως η μέθοδος με τις λιγότερες αναγκαίες υποθέσεις για την αντιμετώπιση κοινωνικο-οικονομικών προβλημάτων.

Άλλοι θετικοί λόγοι για την χρήση της προσομοίωσης στην αντιμετώπιση προβληματικών καταστάσεων σε συστήματα ανθρώπινων δραστηριοτήτων είναι:

1. *Η προσομοίωση ουσιαστικά αντικαθιστά την πειραματική προσέγγιση.* Έτσι μπορούν να γίνουν πειράματα με ελάχιστο κόστος, πολύ γρήγορα και με

ασφάλεια σε σενάρια που μέχρι τώρα δεν είναι εφικτά. Πειραματισμός με τη βοήθεια των υπολογιστών σημαίνει ότι εκτός της μεγάλης μείωσης που επιτυγχάνεται στο κόστος, υπάρχει και μεγάλη αύξηση στον βαθμό και στην ποιότητα της ασφάλειας.

2. *Η προσομοίωση αντιμετωπίζει το πρόβλημα της αδυναμίας της πρόσβασης στο σύστημα.* Η προσομοίωση κάνει δυνατή τη μελέτη και τον πειραματισμό σε περίπλοκες προβληματικές καταστάσεις σε συστήματα που μερικές φορές η πρόσβαση είναι αδύνατη ή επικίνδυνη. Κατασκευάζοντας ένα μοντέλο προσομοίωσης μπορεί να μελετηθεί το σύστημα χωρίς να κινδυνεύσει ο μελετητής ή το σύστημα.
3. *Η προσομοίωση οδηγεί στην καλύτερη κατανόηση του συστήματος.* Μια λεπτομερής παρατήρηση του συστήματος του οποίου γίνεται η προσομοίωση, μέσα από τη διαδικασία ανάπτυξη του μοντέλου προσομοίωσης, μπορεί να οδηγήσει σε καλύτερη κατανόηση του συστήματος, και σε προτάσεις βελτίωσής του, που σε διαφορετική περίπτωση θα ήταν κάτι το ανέφικτο.
4. *Η προσομοίωση μπορεί να χρησιμοποιηθεί σαν εκπαιδευτικός μηχανισμός.* Με την ανάπτυξη μοντέλων προσομοίωσης είναι δυνατόν να εκπαιδεύουν χειριστές χωρίς τον κίνδυνο καταστροφών από λάθος των εκπαιδευόμενων. Επίσης είναι δυνατό να εκπαιδεύουν οι χειριστές ενός συστήματος το οποίο ακόμη δεν έχει κατασκευασθεί.
5. *Η προσομοίωση αποτελεί εργαλείο πρόβλεψης.* Πολλά συστήματα παρουσιάζουν αργές μεταβολές της κατάστασής τους με αποτέλεσμα να είναι αδύνατη η πρόβλεψη της συμπεριφοράς τους για μεγάλα χρονικά διαστήματα. Κατασκευάζοντας και υλοποιώντας ένα μοντέλο προσομοίωσης σε ηλεκτρονικούς υπολογιστές είναι δυνατή η επιτάχυνση των χρονικών μεταβολών, έτσι ώστε να προβλεφθεί η μελλοντική συμπεριφορά του πραγματικού συστήματος για μεγάλο χρονικό διάστημα μέσα σε λίγο πραγματικό χρόνο.
6. *Η προσομοίωση επιτρέπει τον έλεγχο αλλαγών σε ένα σύστημα σε κοινωνικο-οικονομικούς χώρους με το ελάχιστο κόστος.* Με την κατασκευή ενός μοντέλου προσομοίωσης είναι δυνατό να ελεγχθεί η συμπεριφορά του συστήματος για διάφορες τιμές των μεταβλητών και των παραμέτρων του με ελάχιστο κόστος. Από τη μελέτη του μοντέλου προσομοίωσης που έχει κατασκευασθεί διαπιστώνεται ο αποδοτικότερος συνδυασμός μεταβλητών και παραμέτρων, και στη συνέχεια, οι μεταβλητές και οι παράμετροι εφαρμόζονται στο πραγματικό σύστημα.
7. *Η προσομοίωση μπορεί να χρησιμεύσει ως έλεγχος συντήρησης,* αφού με το μοντέλο προσομοίωσης μπορούμε να δοκιμάσουμε νέες πολιτικές και κανόνες απόφασης για την λειτουργία του συστήματος πριν την έκθεση στο κίνδυνο του πειραματισμού του πραγματικού συστήματος. Επίσης η προσομοίωση μπορεί να χρησιμοποιηθεί για τον πειραματισμό με νέες συνθήκες, για τις οποίες έχουμε λίγες ή και καθόλου πληροφορίες ώστε να προετοιμαζόμαστε για το τι μπορεί να συμβεί.

1.7 Εφαρμογές της Προσομοίωσης

Η προσομοίωση έχει χρησιμοποιηθεί σε ευρύ πεδίο περιοχών, μερικές από τις οποίες θα αναφέρουμε στις επόμενες παραγράφους. Ορισμένες από τις περιοχές εφαρμογής της προσομοίωσης στο χώρο παροχής υπηρεσιών και οι αντίστοιχοι στόχοι ανάλυσης και σχεδιασμού, είναι οι ακόλουθοι:

1. *Επικοινωνίες.* Η εφαρμογή της προσομοίωσης είναι ιδιαίτερα ζωτική στο χώρο των επικοινωνιών. Τοπικά δίκτυα υπολογιστών και δίκτυα ευρείας ζώνης, τηλεφωνικά συστήματα, εθνικά και διηπειρωτικά συστήματα δορυφορικών επικοινωνιών, δίκτυα καλωδιακής τηλεόρασης και συστήματα κινητής τηλεφωνίας είναι μερικά από τα παραδείγματα πολύπλοκων συστημάτων που απαιτούν τη συμβολή της προσομοίωσης με υπολογιστή για αποδοτική σχεδίαση και λειτουργία. Η εμφάνιση των λεωφόρων επικοινωνίας και η άνθισή τους στο τέλος της δεκαετίας του 90 δημιουργεί ακόμα μεγαλύτερη ζήτηση για μελέτες με χρήση προσομοίωσης. Για την ανάλυση και το σχεδιασμό επικοινωνιακών συστημάτων διατίθενται ήδη αρκετά εμπορικά εργαλεία προσομοίωσης με ειδικούς σκοπούς.
2. *Οικονομία.* Υπάρχουν πολλές αναφορές σχετικά με την εφαρμογή της προσομοίωσης σε τράπεζες, εταιρείες ασφαλειών και προστασίας. Μερικές από τις δραστηριότητες που μπορούν να εκτιμηθούν με την προσομοίωση είναι: η ανάλυση των συναλλαγών και της ροής χρημάτων, ο σχεδιασμός συστημάτων γραφείων, ο προγραμματισμός υλικών και προμηθειών, ο σχεδιασμός επεξεργασίας δεδομένων και δικτύων υπολογιστών, ο σχεδιασμός αυτόματων ταμειακών μηχανών και συστημάτων εξυπηρέτησης.
3. *Υπηρεσίες τροφίμων.* Συστήματα όπως διάφορα είδη εστιατορίων και συστήματα αποθήκευσης τροφίμων μπορούν να γίνουν αντικείμενα μελετών προσομοίωσης με σκοπούς όπως η διαχείριση της αποθήκης των υλικών και ο σχεδιασμός προμηθειών, ο σχεδιασμός διανομής, η επιλογή θέσεων και ο σχεδιασμός εργατικού δυναμικού.
4. *Υγεία.* Νοσοκομεία, μονάδες εντατικής θεραπείας, ιατρεία, οδοντιατρεία και παραϊατρικά γραφεία μελετώνται συχνά μέσω της προσομοίωσης για να καθοριστούν οι βάρδιες για τις νοσοκόμες και τους γιατρούς, οι πολιτικές για τη διαχείριση των αποθεμάτων φαρμάκων και τροφίμων και τα σχέδια τα σχετικά με τη διαθεσιμότητα σε πόρους όπως κρεβάτια, χώρους αναμονής, χειρουργεία, εξοπλισμούς εργαστηρίων και ασθενοφόρα. Επιπλέον, μελέτες επιδημιών όπως οι προβλεπόμενοι ρυθμοί εξάπλωσης μιας ασθένειας ή ανάλυση εναλλακτικών πολιτικών ελέγχου ασθενειών, θεωρούνται ως εργασίες ρουτίνας για την προσομοίωση.
5. *Υπηρεσίες φιλοξενίας και ξενοδοχείων.* Συστήματα όπως ξενοδοχεία, πανδοχεία και θέρετρα μπορούν να μελετηθούν με χρήση προσομοίωσης για να καθοριστούν παράγοντες όπως: κατάλληλες δυναμικότητες, περιοχές, πολιτικές διαχείρισης αποθηκών και πόρων, μέθοδοι σχεδιασμού και προγραμματισμού ανθρώπινου δυναμικού και συστήματα κρατήσεων.

6. *Μεταφορές.* Εδώ τα συστήματα περιέχουν έναν ή περισσότερους τύπους οχημάτων (π.χ. ταξί, λεωφορεία, τρένα αεροπλάνα, πλοία), επιβάτες, φορτία και διαδρομές μεταφοράς. Η μελέτη προσομοίωσης μπορεί να έχει ως στόχο της τον καθορισμό της χωρητικότητας των οχημάτων το εργατικό δυναμικό (οδηγούς, προσωπικό συντήρησης κ.λ.π.), το σχεδιασμό αποθεμάτων ανταλλακτικών, το σχεδιασμό συντήρησης, τη δρομολόγηση των οχημάτων, το σχεδιασμό λεωφόρων, το σχεδιασμό συστημάτων ελέγχου κίνησης αέρα και εδάφους και το σχεδιασμό των δομών και των χώρων στάθμευσης.
7. *Πρόγνωση καιρού, περιβάλλον και οικολογία.* Για την πρόγνωση του καιρού χρησιμοποιείται μόνιμα και απαρέγκλιτα η προσομοίωση με υπολογιστή. Μεγάλος αριθμός μεταβλητών διαχειρίζονται από προγράμματα προσομοίωσης, που συνήθως εκτελούνται σε υπερ-υπολογιστές, για την πρόβλεψη τοπικών και παγκόσμιων καιρικών συνθηκών σε διάφορες χρονικές στιγμές. Οι μελέτες που αφορούν τον έλεγχο της ρύπανσης, το φαινόμενο του θερμοκηπίου, τους πληθυσμούς των εντόμων και άλλα περιβαλλοντικά και οικολογικά ζητήματα πραγματοποιούνται επίσης με χρήση προσομοίωσης με υπολογιστή.

Τα συστήματα παραγωγής και κατασκευών είναι μια ακόμα σημαντική ενότητα για εφαρμογές προσομοίωσης. Ορισμένα τυπικά συστήματα στην ενότητα αυτή και οι αντίστοιχοι στόχοι για μελέτες προσομοίωσης είναι οι ακόλουθοι:

8. *Εξόρυξη φυσικών πόρων.* Βιομηχανικές μονάδες στον τομέα των ορυχείων, της υλοτομίας, της άντλησης υδάτων και της αλιείας χρησιμοποιούν προσομοίωση με υπολογιστή στη σχεδίαση σχετικών δραστηριοτήτων και τη δημιουργία πολιτικών για έγκαιρη προκήρυξη διαγωνισμών για την αγορά ακριβών μέσων όπως μεγάλων τρακτέρ, φορτωτών, εκσκαφέων, ανυψωτών, γερανών, μεταφορέων, προωθητών γαιών και πλοίων.
9. *Παραγωγή ενέργειας.* Τα συστήματα παραγωγής ενέργειας τα οποία βασίζονται σε πηγές όπως ο ατμός, τα καύσιμα, το νερό, τα πυρηνικά υλικά, ο ήλιος ή ο άνεμος συνήθως προσομοιώνονται για το σχεδιασμό συστημάτων δυναμικότητας, σύνθεση και διανομής και για την ανάλυση και το σχεδιασμό λειτουργικών συστημάτων σχετικά με το προγραμματισμό ρυθμών παραγωγής, το προγραμματισμό διανομής, το σχεδιασμό συστημάτων ελέγχου, το σχεδιασμό συστημάτων ασφάλειας και αξιοπιστίας, τον προγραμματισμό συντήρησης και τον έλεγχο των περιβαλλοντικών επιπτώσεων.
10. *Βιομηχανίες.* Κάθε είδος βιομηχανίας, συμπεριλαμβανομένων των εργοστασίων επεξεργασίας χημικών, των αυτοκινητοβιομηχανιών, των βιομηχανιών κατασκευής αεροπλάνων, μηχανημάτων, ηλεκτρονικών συσκευών κ.λ.π. χρησιμοποιεί εκτεταμένα την προσομοίωση σε εφαρμογές όπως ο στρατηγικός σχεδιασμός, η μέση παραγωγική ικανότητα και ο σχεδιασμός παραγωγής, η μελέτη κατασκευής εργοστασίου, ο σχεδιασμός πολιτικής επιλογής εξοπλισμού, αντικατάστασης και συντήρησης, ελέγχου και προγραμματισμού αποθεμάτων, προγραμματισμού παραγωγής, αποθήκευσης και διαχείρισης υλικών, βελτίωσης γραμμής παραγωγής και

πολλά άλλα ζητήματα που σχετίζονται με το σχεδιασμό, την κατασκευή, τη συναρμολόγηση, τον έλεγχο ποιότητας, την αποθήκευση και τη διανομή. Η ζήτηση των μελετών προσομοίωσης στα βιομηχανικά συστήματα αυξάνεται συνέχεια. Κατά συνέπεια υπάρχει ήδη ένας σημαντικός αριθμός εργαλείων προσομοίωσης ειδικού σκοπού, που προσφέρονται ως εμπορικά προϊόντα για σχεδιασμό και ανάλυση βιομηχανικών συστημάτων.

1.8 Ο ρόλος της Στατιστικής στη Προσομοίωση

Για την ανάπτυξη, την υποστήριξη και τη χρήση μοντέλων προσομοίωσης απαιτούνται στατιστικά μοντέλα με τη μορφή κυρίως στατιστικών κατανομών. Τα στατιστικά αυτά μοντέλα “συμπεριφοράς” είτε αναπτύσσονται με βάση διαθέσιμα στοιχεία (δεδομένα) τα οποία έχουν συγκεντρωθεί στο παρελθόν, είτε αναπτύσσονται με βάση την αναμενόμενη συμπεριφορά των διαφόρων μεταβλητών και παραμέτρων.

Στην περίπτωση που χρησιμοποιούνται στατιστικά μοντέλα που βασίζονται σε στοιχεία που παρατηρήθηκαν στο παρελθόν, τα μοντέλα προσομοίωσης μελετούν (“μιμούνται”) το σύστημα (προβληματική κατάσταση) μέσω διάφορων εναλλακτικών σεναρίων αλλά με την υπάρχουσα υποδομή. Ενώ στη περίπτωση που χρησιμοποιούνται στατιστικά μοντέλα που βασίζονται σε αναμενόμενη συμπεριφορά το μοντέλο μελετά μελλοντικά σενάρια.

Η σπουδαιότητα του ρόλου της στατιστικής διαφαίνεται από το γεγονός ότι η προσομοίωση (“απομίμηση”) της πραγματικότητας είναι πιο ρεαλιστική όσο περισσότερο αναγνωρίζεται ο στοχαστικός χαρακτήρας των διαφόρων μεταβλητών και παραμέτρων. Για παράδειγμα, οι πιθανότητες επικράτησης συγκεκριμένων καιρικών συνθηκών στις διάφορες περιοχές του Αιγαίου και για διαφορετικές περιοχές λαμβάνονται υπόψη από μοντέλο προσομοίωσης των ακτοπλοϊκών συγκοινωνιών στο Αιγαίο. Ο τρόπος με τον οποίο το μοντέλο αναγνωρίζει τον στοχαστικό χαρακτήρα των καιρικών συνθηκών είναι ο ακόλουθος: μέσα σε τακτά χρονικά διαστήματα (μιας ημέρας ή μερικών ωρών) οι καιρικές συνθήκες που θα “επικρατήσουν” στο μοντέλο επιλέγονται δειγματοληπτικά από το στοχαστικό προφίλ (στατιστική κατανομή) που προϋπάρχει και που βασίζεται σε στατιστικές κατανομές. Στη συγκεκριμένη περίπτωση το μοντέλο δειγματοληπτικά θέτει την ισχύ του ανέμου στην κλίμακα μποφόρ. Ένα άλλο παράδειγμα είναι η ζήτηση των επιβατών και των οχημάτων. Η ζήτηση αυτή βασίζεται επίσης σε στοχαστικά προφίλ τα οποία έχουν δημιουργηθεί με βάσει παρατηρήσεις ανά εποχή και φυσικά ανά διαδρομή. Το μοντέλο κάνει δειγματοληψία από τις κατάλληλες στατιστικές κατανομές για να υπολογίσει τη ζήτηση κάθε διαδρομής. Η χρήση και ο ρόλος της στατιστικής στη δομή του μοντέλου παρουσιάζεται αναλυτικότερα στα επόμενα κεφάλαια. Τέλος τα αποτελέσματα της χρήσης των μοντέλων προσομοίωσης παρουσιάζονται και αυτά συνήθως στατιστικά.

Με άλλα λόγια η “απομίμηση” της πραγματικότητας γίνεται για προσομοιωμένα χρονικά διαστήματα όπως π.χ. μερικών ετών για την περίπτωση των ακτοπλοϊκών συγκοινωνιών. Κατά τη διάρκεια του προσομοιωμένου αυτού

χρονικού διαστήματος, οι διάφορες στοχαστικές μεταβλητές συμπεριφέρονται ανάλογα με προϋπολογισμένες στατιστικές μεταβλητές, σε μια προσπάθεια να “μιμηθούν” την πραγματικότητα. Τα προσομοιωμένα από το μοντέλο χρονικά διαστήματα π.χ. μερικών ετών, αντιστοιχούν σε πραγματικό χρόνο του ηλεκτρονικού υπολογιστή μερικών χιλιοστών του δευτερολέπτου έως μερικών λεπτών.

1.9 Μελλοντικές Δυνατότητες της Προσομοίωσης

Η χρήση της προσομοίωσης με υπολογιστή αναμένεται να αυξηθεί με ακόμα πιο γοργούς ρυθμούς στο εγγύς μέλλον. Η κινητήρια δύναμη αυτής της ανάπτυξης μπορεί να αναζητηθεί στους ακόλουθους παράγοντες:

1. Ο βαθμός της πολυπλοκότητας των φυσικών, βιολογικών, κοινωνικο-τεχνικών και κοινωνικο-οικονομικών συστημάτων που μελετώνται από τους σημερινούς σχεδιαστές και αναλυτές συστημάτων απαγορεύει τη χρήση των κλασσικών ή ακόμα και των σύγχρονων μαθηματικών εργαλείων στην πλειοψηφία των πραγματικών μελετών.
2. Οι συνεχώς αυξανόμενες δυνατότητες των υπολογιστών, οι οποίοι επιτυγχάνουν εξαιρετικά υψηλότερες αποδόσεις (κυρίως σε μνήμη και ταχύτητα) απ’ ότι οι προηγούμενες γενιές, επιτρέπουν προσομοίωση συστημάτων ευρείας κλίμακας, ενώ η ταχύτητα επεξεργασίας προσφέρει έγκαιρη παραγωγή πληροφοριών. Οι ισχυροί μικροϋπολογιστές, χαμηλού κόστους, εξασφαλίζουν στις μικρές επιχειρήσεις πρόσβαση στις δυνατότητες που προσφέρει η προσομοίωση.
3. Η εξέλιξη του υπάρχοντος λογισμικού για μοντελοποίηση, το οποίο προσφέρει νέες δυνατότητες και η παραγωγή νέων εργαλείων λογισμικού που περιλαμβάνουν πιο εύκολες και πιο προηγμένες τεχνικές για το περιβάλλον επικοινωνίας με το χρήστη, προσφέρουν στον τελευταίο το αναγκαίο τεχνικό υπόβαθρο και τις προγραμματιστικές δυνατότητες ώστε να χρησιμοποιήσει την προσομοίωση.
4. Η αυξανόμενη αναγνώριση των πλεονεκτημάτων της προσομοίωσης από τους διευθυντές διάφορων οργανισμών και έργων και η δυνατότητα χρήσης σύγχρονων εργαλείων προσομοίωσης και ισχυρών χαμηλού κόστους υπολογιστών, θα οδηγήσει σε επέκταση της χρήσης της προσομοίωσης σε διάφορες δραστηριότητες λήψης αποφάσεων, οι οποίες μέχρι τώρα βασιζόνταν στη διαίσθηση.

ΚΕΦΑΛΑΙΟ 2ο
ΠΡΟΣΟΜΟΙΩΣΗ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ

2.1 Εισαγωγή

Η μέθοδος που θα χρησιμοποιηθεί για να προσομοιωθεί ένα σύστημα έχει άμεση σχέση με το μοντέλο που δημιουργήθηκε για το σύστημα. Αυτό ισχύει και αντιστρόφως: η μεθοδολογία προσομοίωσης, πολλές φορές, υπαγορεύει το είδος του μοντέλου που θα αναπτυχθεί. Έτσι η προσομοίωση διακριτών γεγονότων αφορά την ανάπτυξη μοντέλων σε ηλεκτρονικούς υπολογιστές που περιγράφουν διακριτά συστήματα. Συστήματα στα οποία οι αλλαγές της κατάστασής τους συμβαίνουν μόνο σε συγκεκριμένες χρονικές στιγμές, δηλαδή οι αλλαγές στη κατάστασή τους αναπαρίστανται από ένα σύνολο διακριτών γεγονότων. Τα γεγονότα αυτά μπορούν να προσομοιωθούν με διάφορους τρόπους, ανάλογα με τον τρόπο αντιμετώπισης του μοντέλου. Στο κεφάλαιο αυτό αναπτύσσονται τρεις μεθοδολογίες προσομοίωσης, συγκεκριμένα: η μέθοδος γεγονότων, η μέθοδος δραστηριοτήτων και η μέθοδος διαδικασιών.

Σε μία προσομοίωση με διακριτά γεγονότα μια αλλαγή στη κατάσταση του συστήματος, επέρχεται όταν συμβαίνει ένα γεγονός. Εφόσον η κατάσταση των οντοτήτων παραμένει σταθερή μεταξύ διαδοχικών γεγονότων δεν υπάρχει λόγος καταγραφής αυτής της περιόδου.

Στη συνέχεια θα δώσουμε μερικούς ορισμούς εννοιών που θα χρησιμοποιήσουμε στις επόμενες παραγράφους αυτού του κεφαλαίου.

Οντότητες: Είναι τα αυθύπαρκτα στοιχεία του συστήματος τα οποία προσομοιώνονται και των οποίων η συμπεριφορά ορίζεται επακριβώς. Για παράδειγμα αυτές μπορεί να περιλαμβάνουν ασθενείς σε ένα νοσοκομείο, αεροπλάνα σε ένα αεροδρόμιο κ.λ.π. Το πρόγραμμα που υλοποιεί το μοντέλο προσομοίωσης στον ηλεκτρονικό υπολογιστή, διατηρεί πληροφορίες γύρω από την κατάσταση των οντοτήτων. Όταν συμβεί μία αλλαγή στην κατάσταση της οντότητας τότε το πρόγραμμα καταγράφει αυτή την αλλαγή. Η ολική κατάσταση του συστήματος είναι αποτέλεσμα της αλληλεπίδρασης των οντοτήτων.

Πόροι: Είναι επίσης αυθύπαρκτα στοιχεία του συστήματος αλλά δεν μοντελοποιούνται μεμονωμένα. Οι πόροι έχουν περιορισμένη διαθεσιμότητα. Πολλές φορές οι οντότητες χρειάζονται να χρησιμοποιήσουν και μετά να απελευθερώσουν πόρους. Στις περιπτώσεις αυτές οι οντότητες ίσως χρειασθεί να αναμένουν να έρθει η σειρά τους για τη χρήση των πόρων. Για παράδειγμα σε ένα σύστημα πόροι θα μπορούσαν να είναι τα διαθέσιμα κουτιά ενός προϊόντος σε μια αποθήκη εμπορευμάτων, οι ταμίες σε μία τράπεζα, οι υπάλληλοι σε ένα γραφείο κ.λ.π. Οι πόροι μπορούν να θεωρηθούν και ως οι μόνιμες οντότητες του συστήματος.

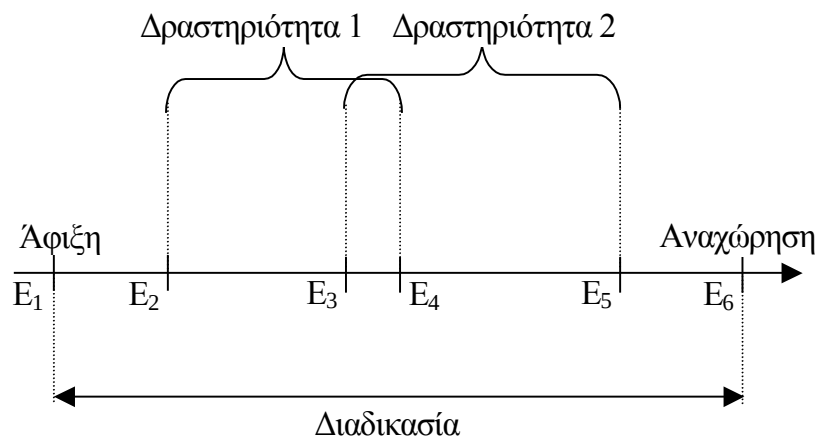
Ρολόι Προσομοίωσης: Είναι το σημείο στο οποίο φυλάσσεται ο τρέχων χρόνος προσομοίωσης. Δηλαδή το ρολόι προσομοίωσης δείχνει πάντα τον τρέχων χρόνο προσομοίωσης. Ο χρόνος προσομοίωσης είναι πάντα διαφορετικός από το πραγματικό χρόνο. Συνήθως μεγάλα προσομοιούμενα χρονικά διαστήματα αντιστοιχούν σε λίγα λεπτά πραγματικού χρόνου.

Γεγονός: Ένα γεγονός αντιστοιχεί σε συγκεκριμένη (διακριτή) χρονική στιγμή κατά την οποία η κατάσταση (η τιμή) μίας ή περισσότερων μεταβλητών αλλάζει, π.χ. εργασία Α αρχίζει, εργασία Β τελειώνει κ.λ.π.

Δραστηριότητα: Μια δραστηριότητα καταναλώνει χρόνο, δηλαδή διαρκεί κάποιο προσομοιούμενο χρονικό διάστημα και περιορίζονται από ένα “αρχικό” και από ένα “τελικό” γεγονός. Π.χ. μία εξυπηρέτηση πελάτη από έναν υπάλληλο που χαρακτηρίζεται από το αρχικό γεγονός: αρχή εξυπηρέτησης.

Διαδικασία: Οι διαδικασίες περιέχουν διάφορα γεγονότα και δραστηριότητες και ουσιαστικά περιγράφουν όλο το ιστορικό μια οντότητας μέσα στο σύστημα που προσομοιώνεται. Π.χ. η διαδικασία ενός οχηματαγωγού πλοίου μπορεί να περιλαμβάνει: α) αναχώρηση από λιμάνι 1 (γεγονός) β) άφιξη στο λιμάνι 2 (γεγονός) γ) αποβίβαση επιβατών – αυτοκινήτων (δραστηριότητα) δ) επιβίβαση επιβατών – αυτοκινήτων (δραστηριότητα) ε) αναχώρηση από λιμάνι 2 (γεγονός) κ.λ.π.

Οι τρεις τελευταίες έννοιες αναπαρίστανται γραφικά στο σχήμα 2.1 που ακολουθεί, όπου μία οντότητα φθάνει σε κάποιο χώρο επεξεργασίας και περνάει από δύο δραστηριότητες πριν αναχωρήσει.



Σχήμα 2.1

2.2 Διαγράμματα Ροών και Κύκλων

Σε πολλά μοντέλα προσομοίωσης διακριτών γεγονότων οι οντότητες και οι πόροι (μόνιμες οντότητες) περιέχονται, είτε σε ροές είναι σε ανακυκλώσεις (κύκλους).

Ροές: Στις ροές περιέχονται οι προσωρινές οντότητες του συστήματος όπως πελάτες που φθάνουν, προωθούνται μέσα από ένα αριθμό εργασιών και μετά αναχωρούν από το σύστημα.

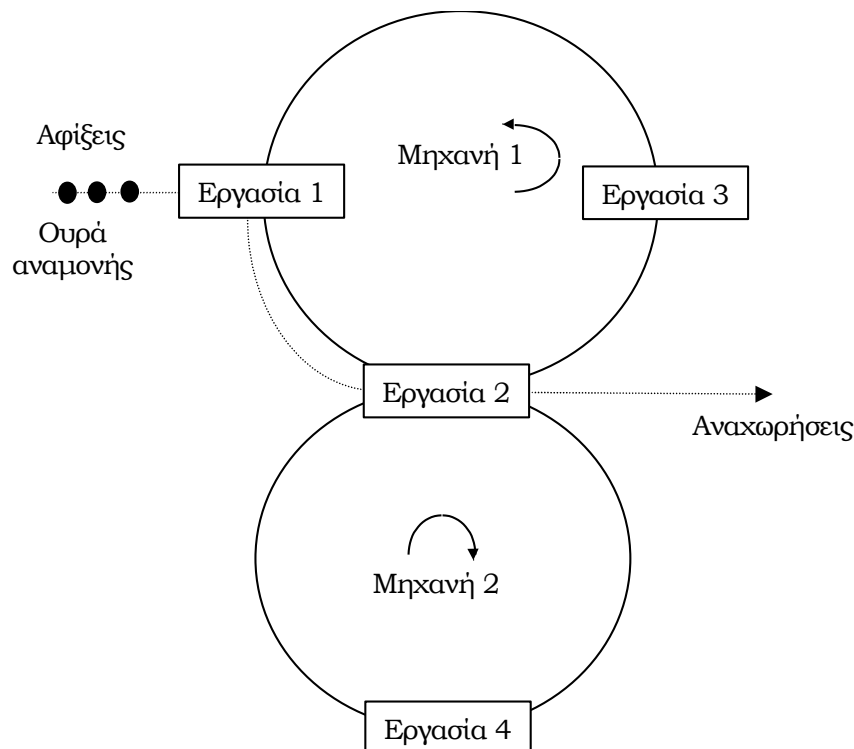
Ανακυκλώσεις: Στις ανακυκλώσεις περιέχονται οι πόροι (μόνιμες οντότητες) του συστήματος, όπως μηχανές, υπάλληλοι κ.λ.π. που ακολουθούν ένα προσχεδιασμένο κύκλο εργασιών.

Οι ροές και οι ανακυκλώσεις είναι πολύ χρήσιμες έννοιες στην αρχική αναπαράσταση του μοντέλου (μοντέλο προσομοίωσης) ενός συστήματος που πρόκειται προσομοιωθεί με προσομοίωση διακριτών γεγονότων.

Έστω ότι έχουμε το ακόλουθο σύστημα:

Έχουμε δύο μηχανές (υπάλληλους κ.λ.π.) που κάνουν ένα αριθμό εργασιών. Συγκεκριμένα η μηχανή 1 κάνει τις εργασίες 1, 2 και 3 ενώ η μηχανή 2 κάνει τις εργασίες 2 και 3. Η εργασία 2 πρέπει να γίνεται ταυτόχρονα και από τις δύο μηχανές. Οι πελάτες (προϊόντα κ.λ.π.) όταν φθάνουν στο σύστημα περνούν από τις εργασίες 1 και 2 και μετά αναχωρούν.

Το μοντέλο προσομοίωσης του παραπάνω συστήματος με τη βοήθεια των ροών και ανακυκλώσεων μπορεί να αναπαρασταθεί ως εξής:



Σχήμα 2.2

Η κατάσταση του συστήματος κάθε φορά εκφράζεται από τις τιμές των μεταβλητών και των παραμέτρων του συστήματος.

Οι μεταβλητές του συστήματος είναι οι:

- NQ: Είναι η μεταβλητή που δηλώνει το μήκος της ουράς.
- M1: Είναι η μεταβλητή που δηλώνει την κατάσταση της μηχανής 1.
- M2: Είναι η μεταβλητή που δηλώνει την κατάσταση της μηχανής 2.
- ST: Είναι η μεταβλητή που δηλώνει τον τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

Κατά τη διάρκεια της προσομοίωσης αλλάζει η τιμή αυτών των μεταβλητών με αποτέλεσμα να αλλάζει η κατάσταση του συστήματος που προσομοιώνεται.

Οι παράμετροι του συστήματος είναι οι:

- ARRI: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.

Κατά τη διάρκεια της προσομοίωσης δεν αλλάζουν οι τιμές των παραμέτρων. Μπορούμε όμως για τις ίδιες αρχικές τιμές των μεταβλητών και για διαφορετικές τιμές των παραμέτρων να ξανακάνουμε προσομοίωση του συστήματος.

Υπάρχουν τρεις διαφορετικές μεθοδολογίες για την ανάπτυξη μοντέλων προσομοίωσης διακριτών γεγονότων, οι οποίες βασίζονται αντίστοιχα στις τρεις έννοιες *γεγονός, δραστηριότητα, διαδικασία*:

- Μέθοδος Γεγονότων.
- Μέθοδος Δραστηριοτήτων.
- Μέθοδος Διαδικασιών.

2.3 Μέθοδος Γεγονότων

Η μέθοδος των γεγονότων μελετάει τα στοιχειώδη γεγονότα του συστήματος. Αυτά είναι οι συγκεκριμένες χρονικές στιγμές στις οποίες η τιμή μίας ή περισσότερων μεταβλητών του συστήματος αλλάζει.

Π.χ. Αφίξη

 Έναρξη εργασίας 1

 Τέλος εργασίας 2 κ.λ.π.

Δομή της μεθόδου

Η μέθοδος των γεγονότων λειτουργεί μέσω:

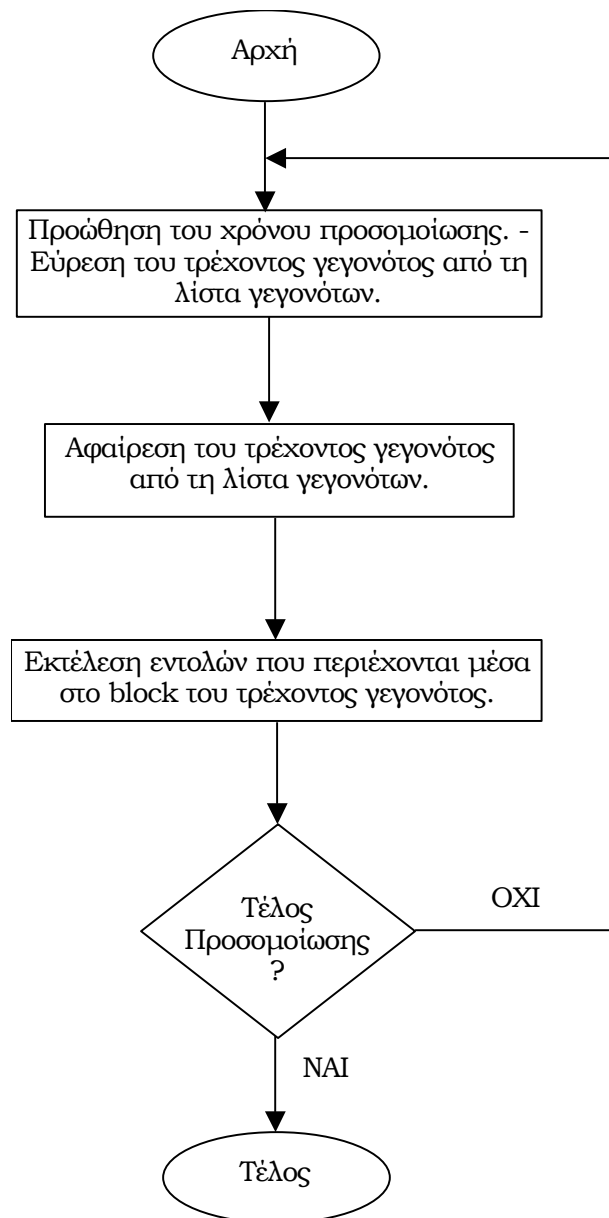
1. Ενός “block” για κάθε γεγονός το οποίο περιέχει μία περιγραφή του πως αλλάζουν οι μεταβλητές του συστήματος όταν συμβεί το γεγονός και συγχρόνως σχεδιάζει όλα τα καινούργια γεγονότα που μπορούν να σχεδιασθούν.
2. Ενός αρχείου (λίστα) γεγονότων το οποίο περιέχει όλα τα τρέχοντα και μελλοντικά γεγονότα καταγράφοντας τον τύπο τους και το χρόνο που θα συμβούν.

Διαδικασία της μεθόδου

Η διαδικασία προσομοίωσης με τη μέθοδο των γεγονότων ακολουθεί τα εξής επαναληπτικά βήματα:

1. Ευρίσκει στη λίστα γεγονότων το πιο κοντινό χρονικά γεγονός, αλλάζει το χρόνο προσομοίωσης (ρολόι προσομοίωσης) στο χρόνο του γεγονότος αυτού και καταστρέφει τα στοιχεία του γεγονότος (όνομα και χρόνο) από τη λίστα των γεγονότων.
2. Εκτελεί τις εντολές που αντιστοιχούν στο block του γεγονότος που διέγραψε από τη λίστα των γεγονότων.

Αυτή η διαδικασία αναπαρίσταται στο λογικό διάγραμμα (Σχήμα 2.3) της εκτελεστικής ρουτίνας της μεθόδου, που ακολουθεί.



Σχήμα 2.3: Εκτελεστική Ρουτίνα για τη Μέθοδο των Γεγονότων

Εφαρμογή 1η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.2 (σελ 24).

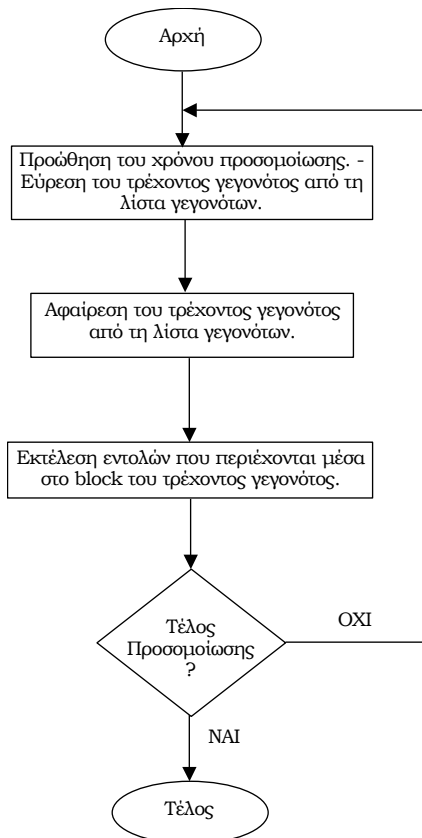
Μεταβλητές Συστήματος:

- $NQ = n$ Το μήκος της ουράς είναι n πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει την εργασία 1.
= 2 Η μηχανή 1 κάνει την εργασία 2.
= 3 Η μηχανή 1 κάνει την εργασία 3.
= 10 Η μηχανή 1 έχει κάνει την εργασία 1 και περιμένει.
= 30 Η μηχανή 1 έχει κάνει την εργασία 3 και περιμένει.
- $M2 = 2$ Η μηχανή 2 κάνει την εργασία 2.
= 4 Η μηχανή 2 κάνει την εργασία 4.
= 40 Η μηχανή 2 έχει κάνει την εργασία 4 και περιμένει.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

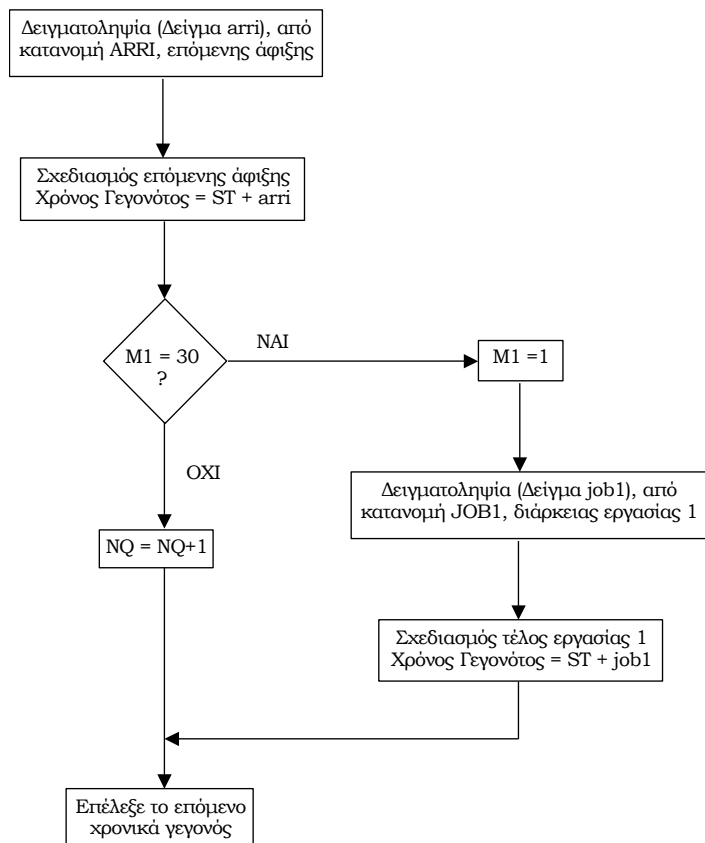
Παράμετροι Συστήματος:

- $ARRI$: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- $JOB1$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- $JOB2$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- $JOB3$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- $JOB4$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.

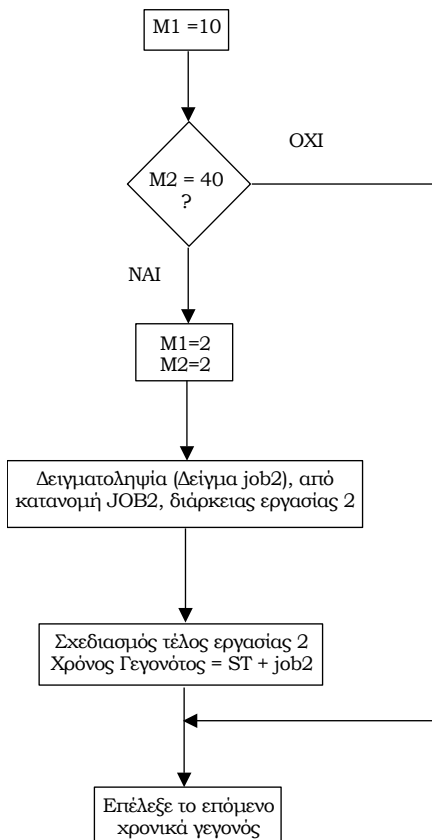
Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.2) με τη μέθοδο των γεγονότων θα πρέπει να περιγράψουμε τα blocks των γεγονότων: Αφίξη, Job1Ends (τέλος εργασίας 1), Job2Ends (τέλος εργασίας 2), Job3Ends (τέλος εργασίας 3), Job4Ends (τέλος εργασίας 4) καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου.



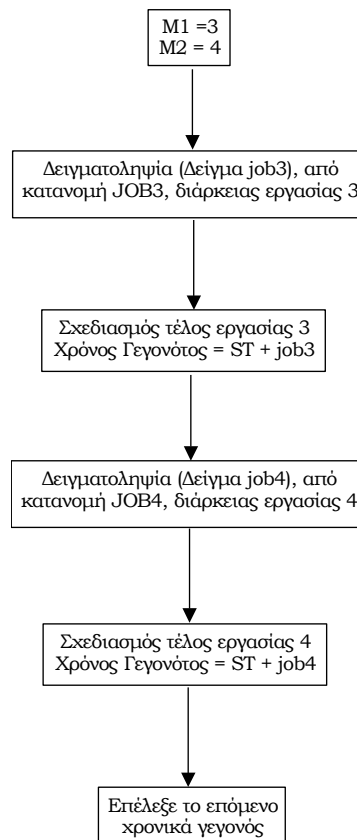
Σχήμα 2.4: Εκτελεστική ρουτίνα



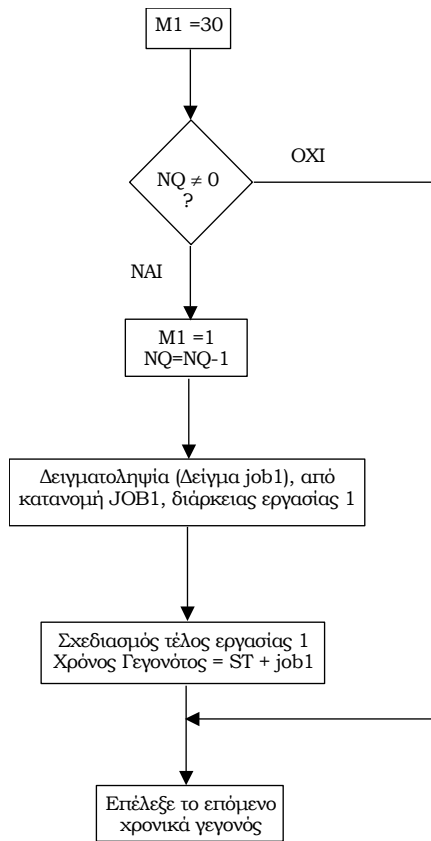
Σχήμα 2.5: Block Άφιξης



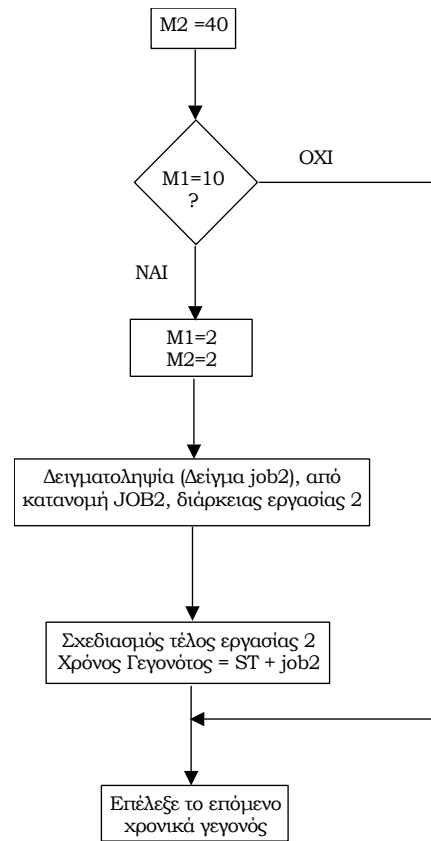
Σχήμα 2.6: Block Job1Ends



Σχήμα 2.7: Block Job2Ends



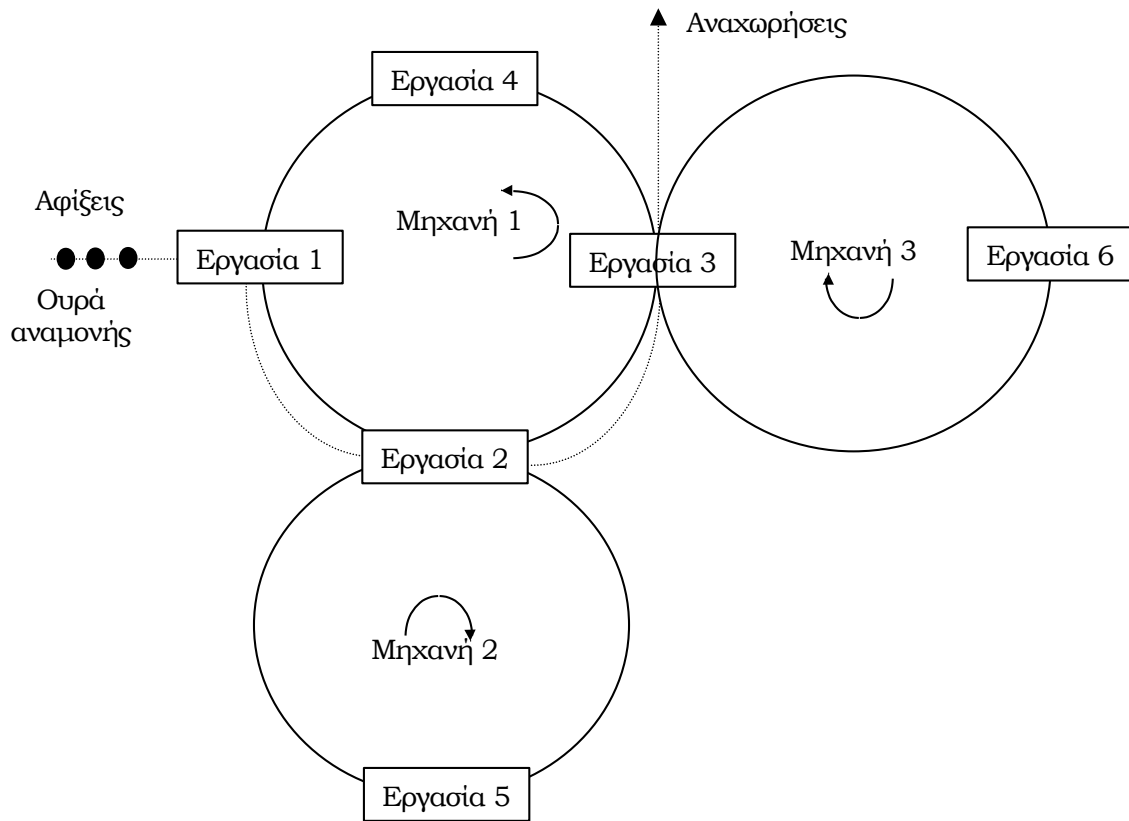
Σχήμα 2.8: Block Job3Ends



Σχήμα 2.9: Block Job4Ends

Εφαρμογή 2η

Έστω ότι έχουμε το παρακάτω μοντέλο προσομοίωσης (σχήμα 2.10).



Σχήμα 2.10

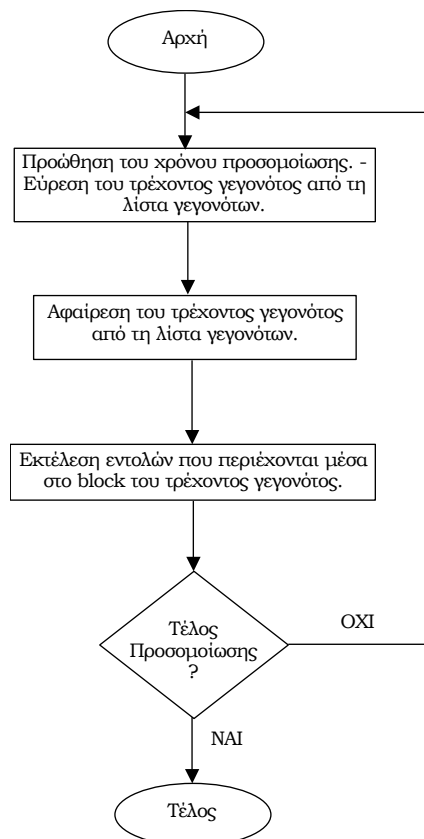
Μεταβλητές Συστήματος:

- $NQ = n$ Το μήκος της ουράς είναι n πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει την εργασία 1.
= 2 Η μηχανή 1 κάνει την εργασία 2.
= 3 Η μηχανή 1 κάνει την εργασία 3.
= 4 Η μηχανή 1 κάνει την εργασία 4.
= 10 Η μηχανή 1 έχει κάνει την εργασία 1 και περιμένει.
= 20 Η μηχανή 1 έχει κάνει την εργασία 2 και περιμένει.
= 40 Η μηχανή 1 έχει κάνει την εργασία 4 και περιμένει.
- $M2 = 2$ Η μηχανή 2 κάνει την εργασία 2.
= 5 Η μηχανή 2 κάνει την εργασία 5.
= 50 Η μηχανή 2 έχει κάνει την εργασία 5 και περιμένει.
- $M3 = 3$ Η μηχανή 3 κάνει την εργασία 3.
= 6 Η μηχανή 3 κάνει την εργασία 6.
= 60 Η μηχανή 3 έχει κάνει την εργασία 6 και περιμένει.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

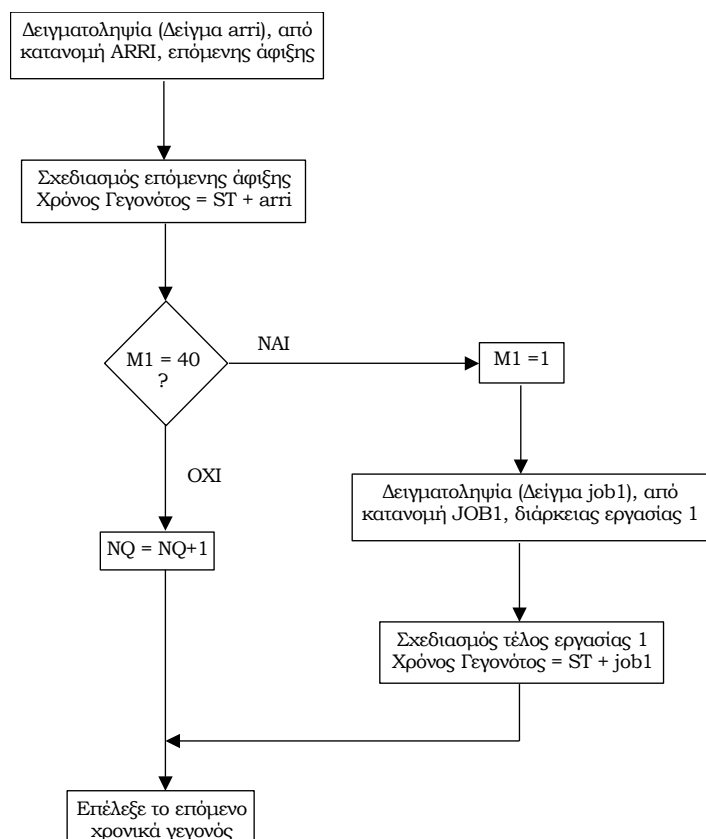
Παράμετροι Συστήματος:

- ARRI: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.
- JOB5: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 5.
- JOB6: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 6.

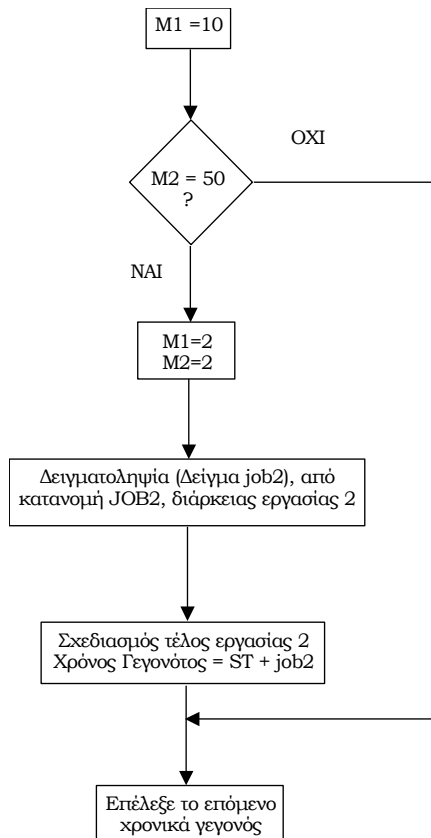
Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.10) με τη μέθοδο των γεγονότων θα πρέπει να περιγράψουμε τα blocks των γεγονότων: Άφιξη, Job1Ends (τέλος εργασίας 1), Job2Ends (τέλος εργασίας 2), Job3Ends (τέλος εργασίας 3), Job4Ends (τέλος εργασίας 4), Job5Ends (τέλος εργασίας 5), Job6Ends (τέλος εργασίας 6) καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου.



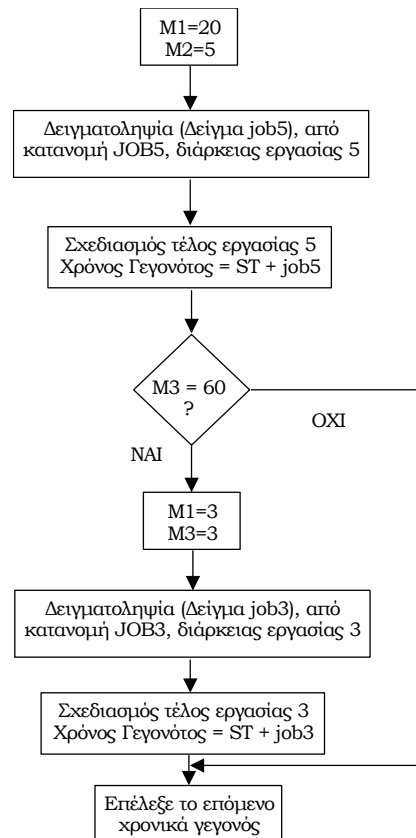
Σχήμα 2.11: Εκτελεστική ρουτίνα



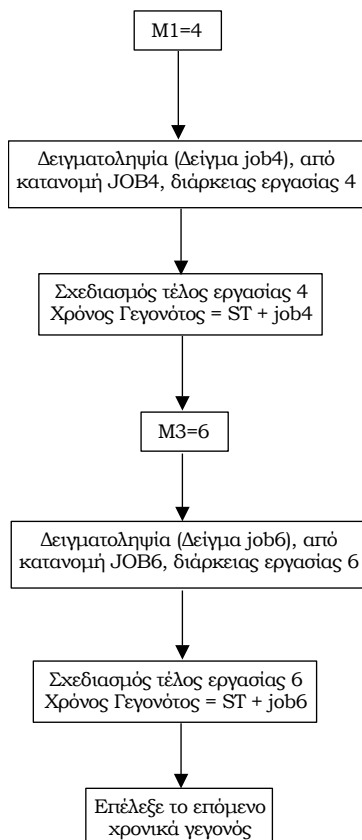
Σχήμα 2.12: Block Άφιξης



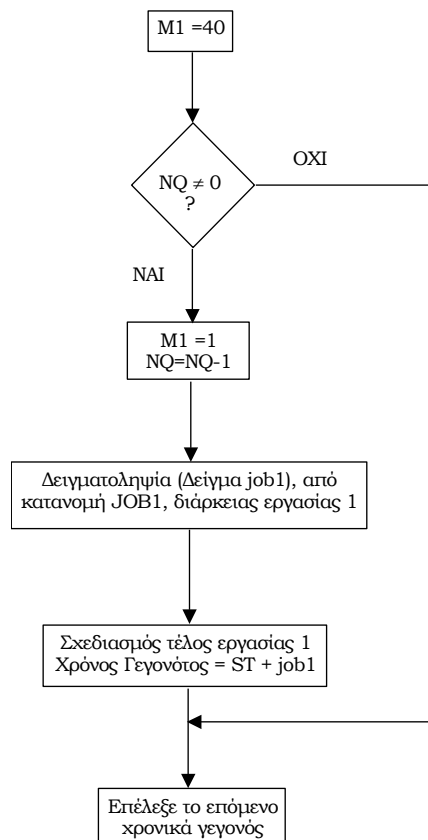
Σχήμα 2.13: Block Job1Ends



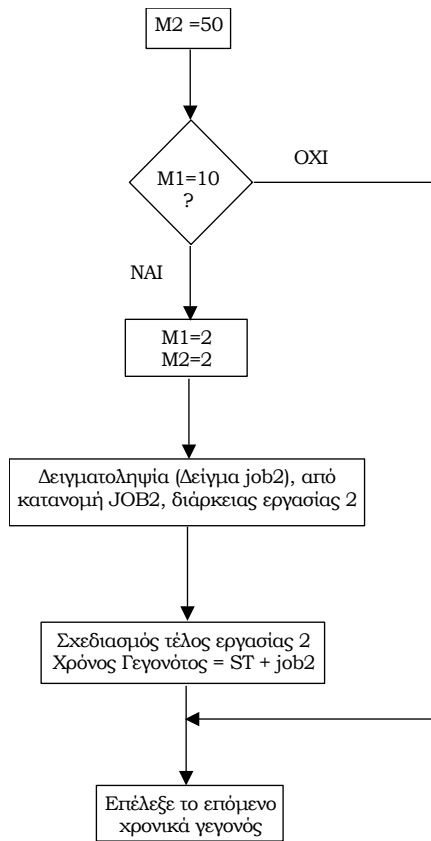
Σχήμα 2.14: Block Job2Ends



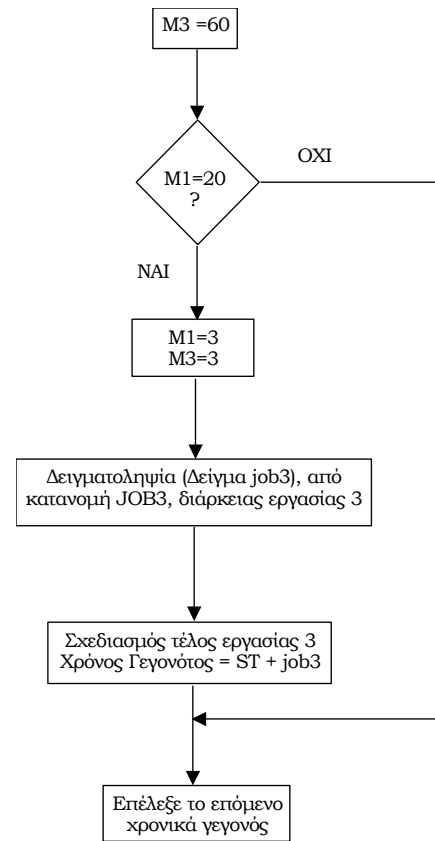
Σχήμα 2.15: Block Job3Ends



Σχήμα 2.16: Block Job4Ends



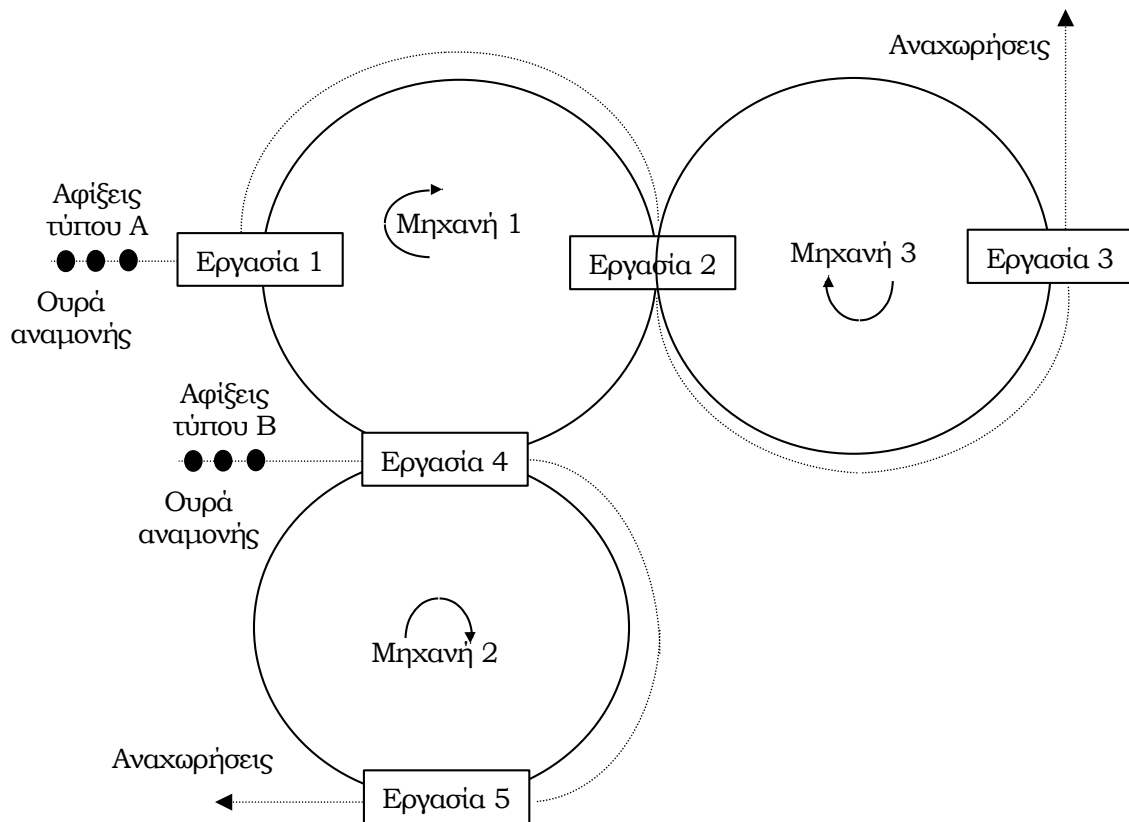
Σχήμα 2.17: Block Job5Ends



Σχήμα 2.18: Block Job6Ends

Εφαρμογή 3η

Έστω ότι έχουμε το παρακάτω μοντέλο προσομοίωσης (σχήμα 2.19).



Σχήμα 2.19

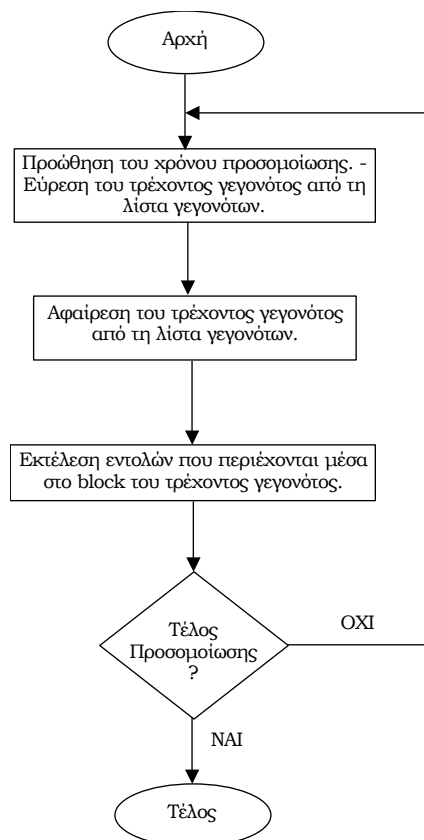
Μεταβλητές Συστήματος:

- $NQ_A = n$ Το μήκος της ουράς Α είναι n πελάτες.
- $NQ_B = m$ Το μήκος της ουράς Β είναι m πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει την εργασία 1.
= 2 Η μηχανή 1 κάνει την εργασία 2.
= 4 Η μηχανή 1 κάνει την εργασία 4.
= 10 Η μηχανή 1 έχει κάνει την εργασία 1 και περιμένει.
= 20 Η μηχανή 1 έχει κάνει την εργασία 2 και περιμένει.
= 40 Η μηχανή 1 έχει κάνει την εργασία 4 και περιμένει.
- $M2 = 4$ Η μηχανή 2 κάνει την εργασία 4.
= 5 Η μηχανή 2 κάνει την εργασία 5.
= 50 Η μηχανή 2 έχει κάνει την εργασία 5 και περιμένει.
- $M3 = 3$ Η μηχανή 3 κάνει την εργασία 3.
= 2 Η μηχανή 3 κάνει την εργασία 2.
= 30 Η μηχανή 3 έχει κάνει την εργασία 3 και περιμένει.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

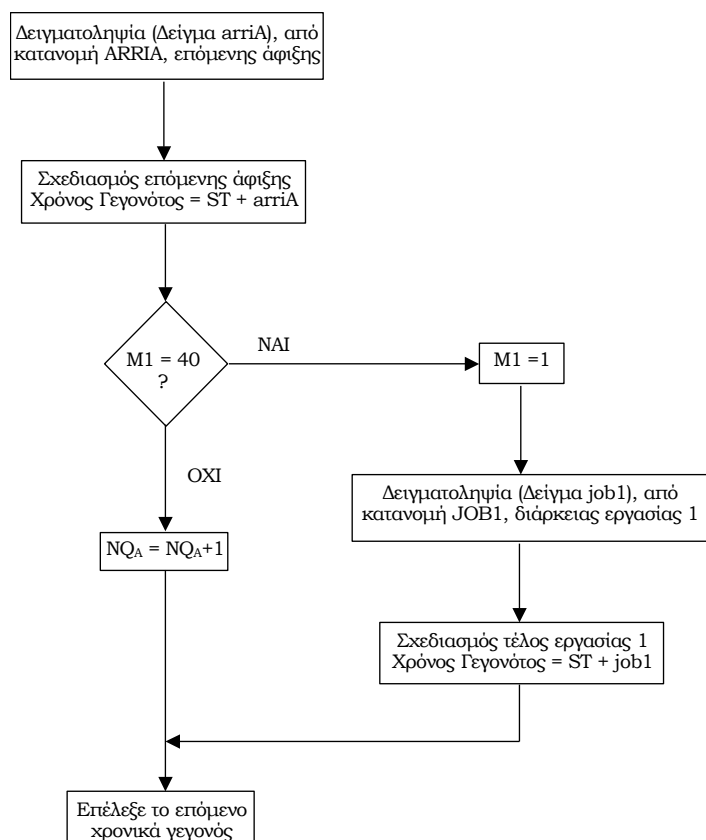
Παράμετροι Συστήματος:

- ARRIA: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου A.
- ARRIB: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου B.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.
- JOB5: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 5.

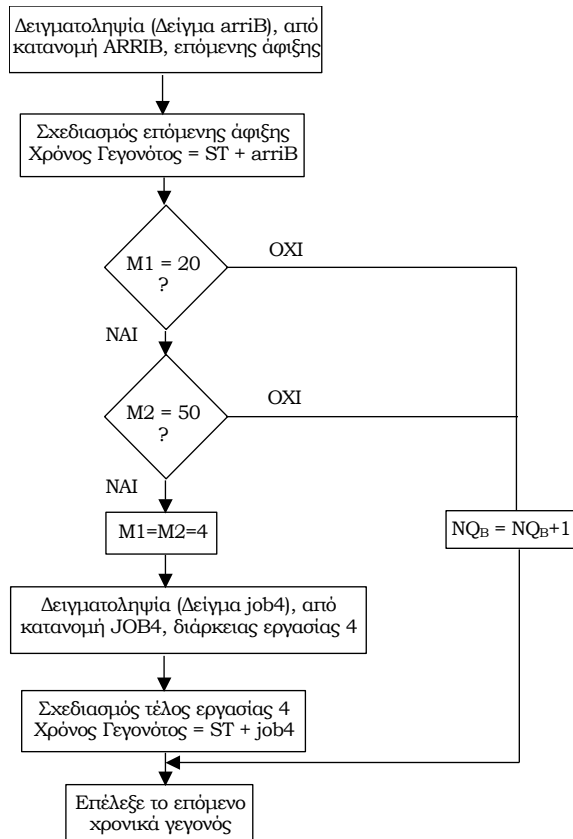
Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.19) με τη μέθοδο των γεγονότων θα πρέπει να περιγράψουμε τα blocks των γεγονότων: ΆφιξηA, ΆφιξηB, Job1Ends (τέλος εργασίας 1), Job2Ends (τέλος εργασίας 2), Job3Ends (τέλος εργασίας 3), Job4Ends (τέλος εργασίας 4), Job5Ends (τέλος εργασίας 5) καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου.



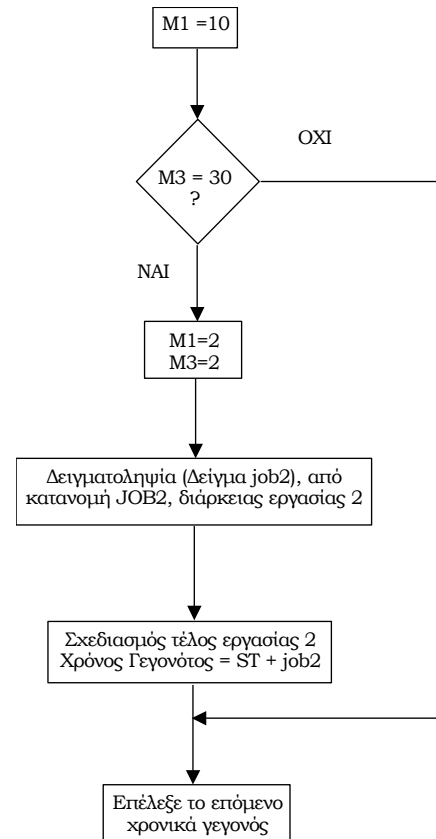
Σχήμα 2.20: Εκτελεστική ρουτίνα



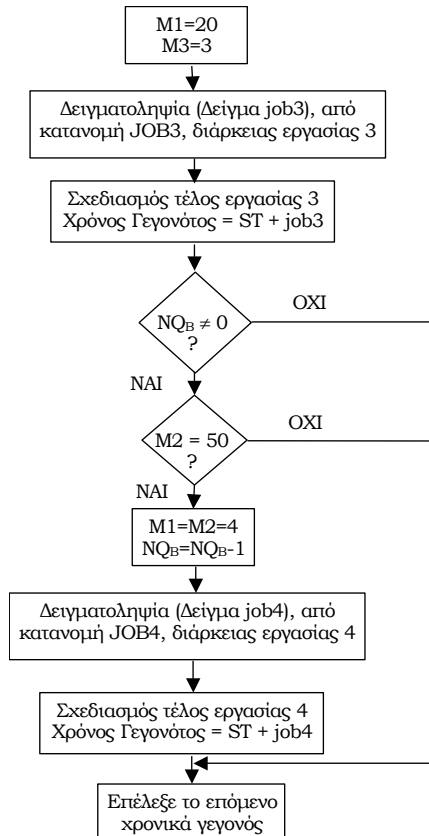
Σχήμα 2.21: Block Άφιξης τύπου A



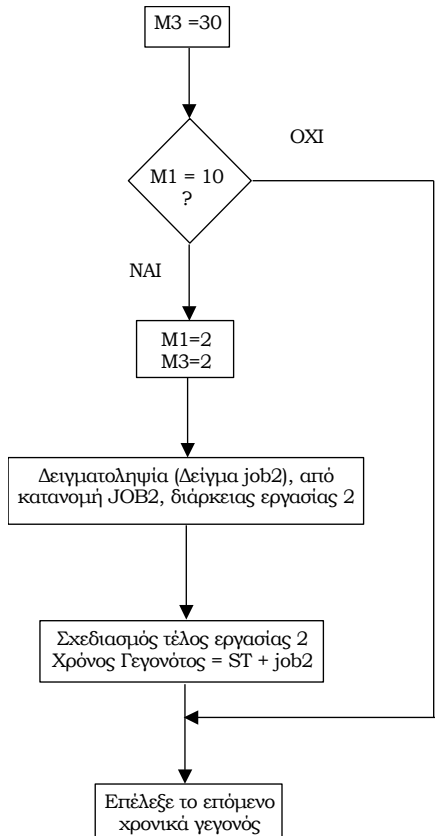
Σχήμα 2.22: Block Αφίξης τύπου B



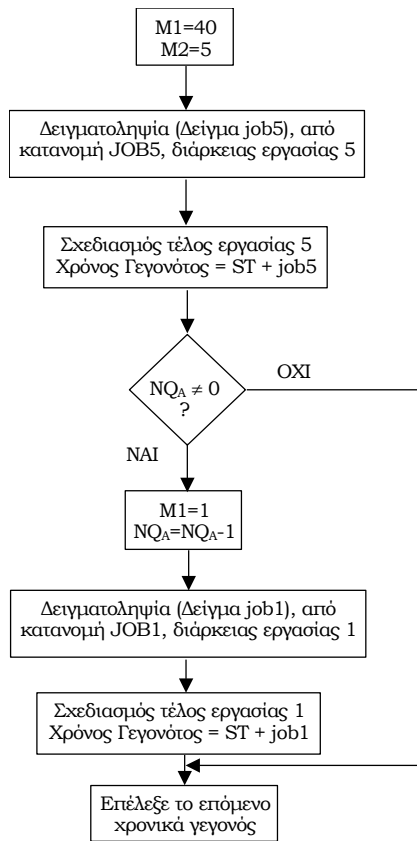
Σχήμα 2.23: Block Job1Ends



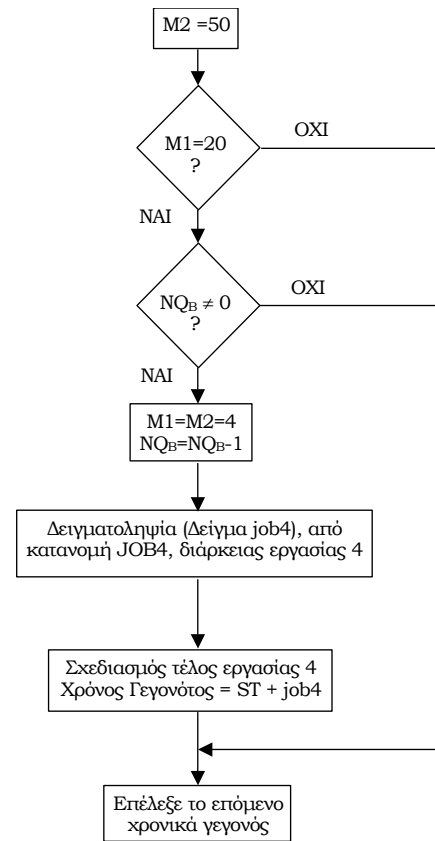
Σχήμα 2.24: Block Job2Ends



Σχήμα 2.25: Block Job3Ends



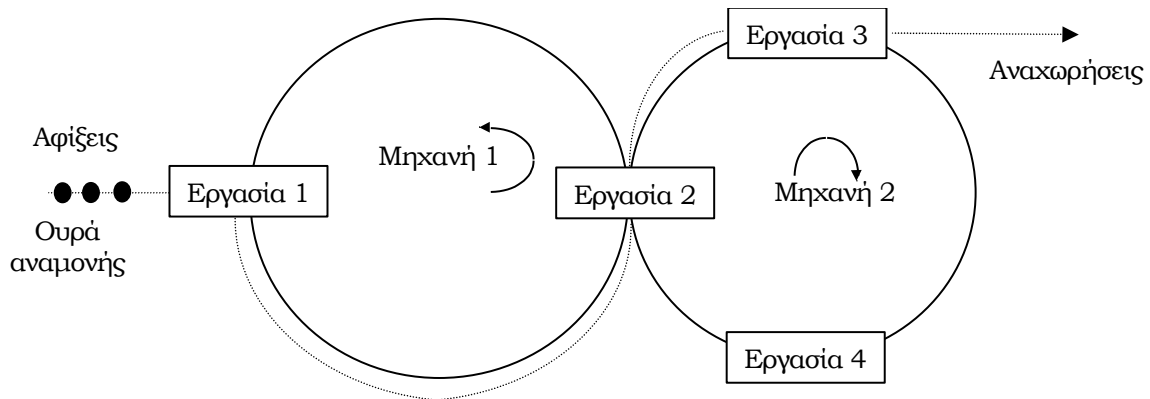
Σχήμα 2.26: Block Job4Ends



Σχήμα 2.27: Block Job5Ends

Εφαρμογή 4η

Έστω ότι έχουμε το παρακάτω μοντέλο προσομοίωσης (σχήμα 2.28).



Σχήμα 2.28

Το μοντέλο το προσομοιώνουμε με τη μέθοδο των γεγονότων και τη χρονική στιγμή 46 η κατάσταση του συστήματος καθώς και η λίστα των γεγονότων είναι αυτή που φαίνεται στο σχήμα 2.29.

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1

Λίστα Γεγονότων
Άφιξη 49
Job1Ends 47
Job3Ends 48

Σχήμα 2.29: Κατάσταση συστήματος και Λίστα γεγονότων

Για τις μεταβλητές και τις παραμέτρους του συστήματος καθώς επίσης και για τα blocks των γεγονότων ισχύουν ότι και στις προηγούμενες εφαρμογές.

Επίσης η δειγματοληψία στις κατανομές ARRI, JOB1, JOB2, JOB3 και JOB4 έδωσε τα παρακάτω δείγματα:

ARRI: 4, 6, 1, 3, 2,

JOB1: 3, 2, 1, 3, 2

JOB2: 1, 1, 1, 1, 1

JOB3: 2, 4, 1, 1, 3

JOB4: 2, 1, 2, 1, 2

Συνεχίστε την προσομοίωση του συστήματος με τη μέθοδο των γεγονότων μέχρι να φύγει από το σύστημα ο πελάτης που φθάνει τη χρονική στιγμή 49.

Αρχικά ψάχνουμε στη λίστα των γεγονότων για το κοντινότερο χρονικά γεγονός. Αυτό είναι το Job1Ends που γίνεται τη χρονική στιγμή 47. Στη συνέχεια το αφαιρούμε από τη λίστα των γεγονότων και εκτελούμε τις εντολές που υπάρχουν στο block του. Η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.30.

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1

Λίστα Γεγονότων
Άφιξη 49
Job3Ends 48

Σχήμα 2.30

Το κοντινότερο χρονικά γεγονός είναι το Job3Ends τη χρονική στιγμή 48. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.31.

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job3Ends	48	10	4	1

Λίστα Γεγονότων
Άφιξη 49
Job4Ends 50

Σχήμα 2.31

Παρατηρούμε ότι στη λίστα γεγονότων προστέθηκε το γεγονός Job4Ends με χρόνο το 50, αφού η μηχανή 2 μετά το τέλος της εργασίας 3 (Job3Ends) κάνει την εργασία 4 και το πρώτο δείγμα από την κατανομή JOB4 είναι 2 και άρα ο χρόνος Job4Ends = 48 + 2 = 50.

Το κοντινότερο, τώρα, χρονικά γεγονός είναι η Άφιξη τη χρονική στιγμή 49. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.32. Παρατηρούμε ότι πελάτης που μας ενδιαφέρει βρίσκεται δεύτερος στην ουρά και ότι σχεδιάστηκε νέο γεγονός (άφιξη) τη χρονική στιγμή 53 = 49 + 4 (πρώτο δείγμα από κατανομή ARRI).

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2

Λίστα Γεγονότων
Job4Ends 50
Άφιξη 53

Θέση του πελάτη που
(2) μας ενδιαφέρει στην
ουρά

Σχήμα 2.32

Το κοντινότερο, τώρα, χρονικά γεγονός είναι το Job4Ends τη χρονική στιγμή 50. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.33. Παρατηρούμε ότι σχεδιάστηκε νέο γεγονός (Job2Ends) τη χρονική στιγμή 51 = 50 + 1 (πρώτο δείγμα από κατανομή JOB2).

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2
Job4Ends	50	2	2	2

(2)

Σχήμα 2.33

Το κοντινότερο, τώρα, χρονικά γεγονός είναι το Job2Ends τη χρονική στιγμή 51. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.34. Παρατηρούμε ότι σχεδιάσθηκαν δύο νέα γεγονότα: το γεγονός Job1Ends τη χρονική στιγμή $54 = 51 + 3$ (πρώτο δείγμα από κατανομή JOB1) και το γεγονός Job3Ends τη χρονική στιγμή $53 = 51 + 2$ (πρώτο δείγμα από κατανομή JOB3). Επίσης ο πελάτης που μας ενδιαφέρει έχει έρθει στην κεφαλή της ουράς.

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2
Job4Ends	50	2	2	2
Job2Ends	51	1	3	1

(1)

Σχήμα 2.34

Το κοντινότερο, τώρα, χρονικά γεγονός είναι το Job3Ends και η Άφιξη τη χρονική στιγμή 53. Επιλέγουμε αυθαίρετα ένα, έστω το Job3Ends. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.35. Παρατηρούμε ότι σχεδιάσθηκε το γεγονός Job4Ends τη χρονική στιγμή $54 = 53 + 1$ (δεύτερο δείγμα από κατανομή JOB4).

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2
Job4Ends	50	2	2	2
Job2Ends	51	1	3	1
Job3Ends	53	1	4	1

(1)

Σχήμα 2.35

Λίστα Γεγονότων
Άφιξη 53
Job2Ends 51

Λίστα Γεγονότων
Άφιξη 53
Job1Ends 54
Job3Ends 53

Λίστα Γεγονότων
Άφιξη 53
Job1Ends 54
Job4Ends 54

Το κοντινότερο, τώρα, χρονικά γεγονός είναι η άφιξη τη χρονική στιγμή 53. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.36. Παρατηρούμε ότι σχεδιάσθηκε το γεγονός Άφιξη τη χρονική στιγμή $59 = 53 + 6$ (δεύτερο δείγμα από κατανομή ARRI).

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2
Job4Ends	50	2	2	2
Job2Ends	51	1	3	1
Job3Ends	53	1	4	1
Άφιξη	53	1	4	2

(1)

Λίστα Γεγονότων
Job1Ends 54
Job4Ends 54
Άφιξη 59

Σχήμα 2.36

Το κοντινότερο, τώρα, χρονικά γεγονός είναι το Job1Ends τη χρονική στιγμή 54. Το αφαιρούμε από τη λίστα των γεγονότων και η νέα κατάσταση του συστήματος καθώς και η λίστα γεγονότων φαίνεται στο σχήμα 2.37.

Κατάσταση Συστήματος				
Γεγονός	ST	M1	M2	NQ
Άφιξη	46	1	3	1
Job1Ends	47	10	3	1
Job2Ends	48	10	4	1
Άφιξη	49	10	4	2
Job4Ends	50	2	2	2
Job2Ends	51	1	3	1
Job3Ends	53	1	4	1
Άφιξη	53	1	4	2
Job1Ends	54	10	4	2

(1)

Λίστα Γεγονότων
Job4Ends 54
Άφιξη 59

Σχήμα 2.37

Συνεχίζουμε τη προσομοίωση μέχρι να εκτελεστεί το γεγονός Job3Ends και η θέση του πελάτη που μας ενδιαφέρει να είναι 0, δηλαδή να είναι αυτός που εξυπηρετείται.

2.4 Μέθοδος Δραστηριοτήτων

Η μέθοδος εξετάζει μόνο τις δραστηριότητες.

Δομή της Μεθόδου

Σε κάθε προώθηση χρόνου η μέθοδος εξετάζει εάν η “μηχανή” λειτουργεί ή όχι και αν δεν λειτουργεί τότε εξετάζει εάν ικανοποιούνται οι προϋποθέσεις για να αρχίσει μια καινούρια δραστηριότητα.

Μαζί με κάθε “μηχανή” υπάρχει ένα ρολόι (μια μεταβλητή χρόνου) το οποίο δείχνει:

1. Τον χρόνο που θα σταματήσει η μηχανή εάν αυτή λειτουργεί.
2. Τον χρόνο που σταμάτησε η μηχανή εάν αυτή δεν λειτουργεί.

Σημείωση: Δεν υπάρχει ανάγκη για την ύπαρξη δεικτών λειτουργίας / μη λειτουργίας των μηχανών όπως στη μέθοδο των γεγονότων. Σύγκριση του χρόνου προσομοίωσης (ρολόι προσομοίωσης) ST με το ρολόι κάθε μηχανής απαντά στο ερώτημα λειτουργίας της μηχανής.

Οι προϋποθέσεις που απαιτούνται, για να γίνει μια δραστηριότητα, είναι συνήθως της μορφής:

1. Διαθεσιμότητα της μηχανής.
2. Διαθεσιμότητα περισσότερων μηχανών.
3. Οι μηχανές πρέπει να βρίσκονται στο κατάλληλο σημείο του κύκλου που ακολουθούν.
4. Άλλες προϋποθέσεις ανεξάρτητες από τις 1, 2, 3 όπως πελάτης στην ουρά κ.λ.π.

Οι επιπτώσεις που έχουμε στο μοντέλο προσομοίωσης, όταν γίνεται μια δραστηριότητα, είναι της μορφής:

1. Το ρολόι της μηχανής (των μηχανών) που κάνει (-ουν) την δραστηριότητα αλλάζει και δείχνει το χρόνο που θα σταματήσει (ελευθερωθεί) η μηχανή (οι μηχανές).
2. Αλλάζει η κατάσταση της μηχανής ή των μηχανών που εκτελούν την δραστηριότητα.
3. Άλλες επιπτώσεις ανεξάρτητες από τις 1, 2, όπως πρόσθεση πελάτη στην ουρά κ.λ.π.

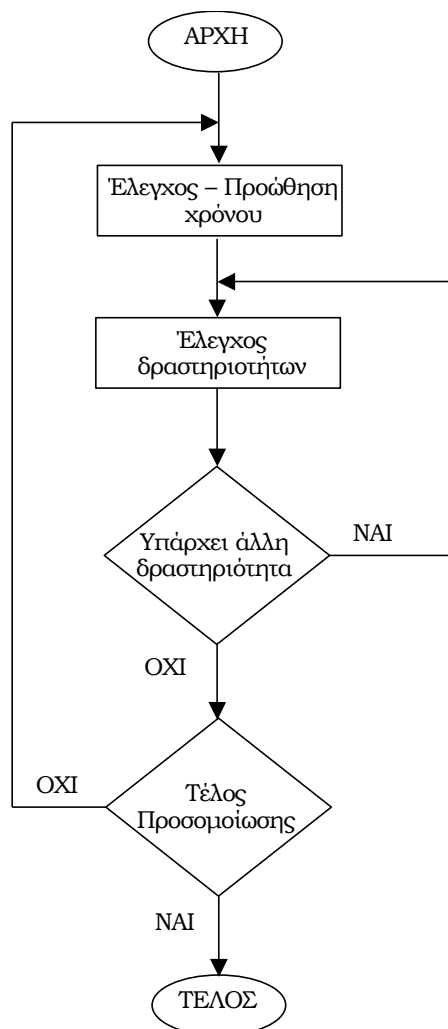
Σημείωση: 1. Ο χρόνος προσομοίωσης κινείται πάντα προς τα μπρος.
2. Αλλαγή του χρόνου προσομοίωσης σημαίνει ότι μια μηχανή σταμάτησε να λειτουργεί και δεν υποδεικνύει τι θα συμβεί μετά.

Διαδικασία της μεθόδου

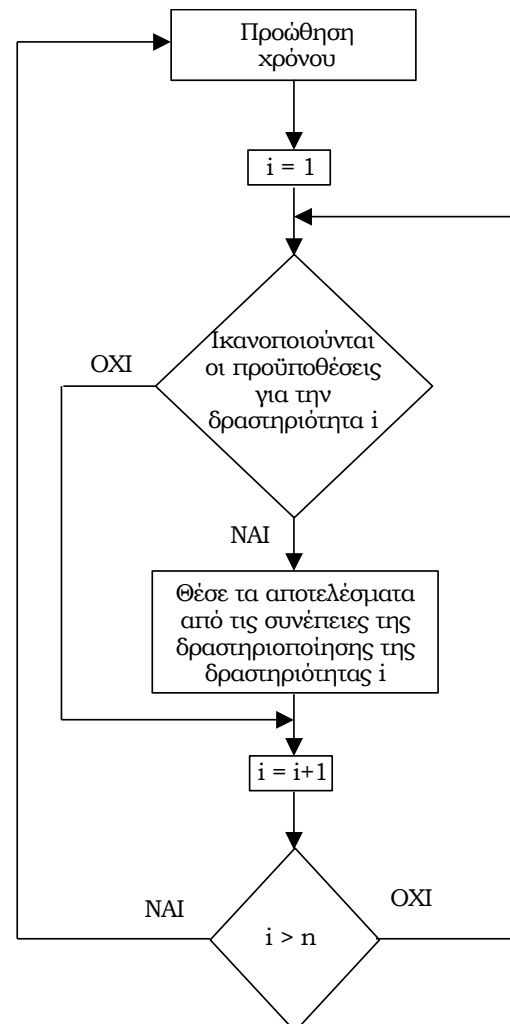
Η διαδικασία προσομοίωσης με τη μέθοδο των δραστηριοτήτων ακολουθεί τα εξής επαναληπτικά βήματα:

1. Ευρίσκει το χρόνο του επόμενου γεγονότος και αλλάζει το χρόνο προσομοίωσης (ρολόι προσομοίωσης) στο χρόνο του γεγονότος αυτού.
2. Για αυτή τη χρονική στιγμή ελέγχει αν ικανοποιούνται οι προϋποθέσεις για να γίνει μία δραστηριότητα και αν ναι εκτελεί τις συνέπειές της. Αυτό τον έλεγχο τον κάνει για κάθε μία από τις δραστηριότητες που έχουν οριστεί.

Η διαδικασία αυτή, γενικά, αναπαρίσταται στο λογικό διάγραμμα (Σχήμα 2.38) της εκτελεστικής ρουτίνας της μεθόδου, που ακολουθεί, ενώ το βήμα 2 (έλεγχος των δραστηριοτήτων) αναπαρίσταται στο σχήμα 2.39.

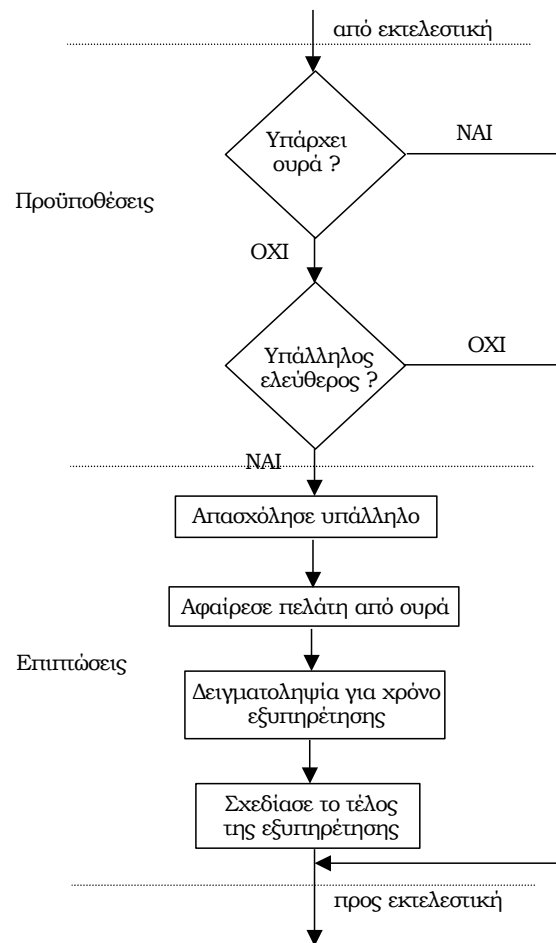
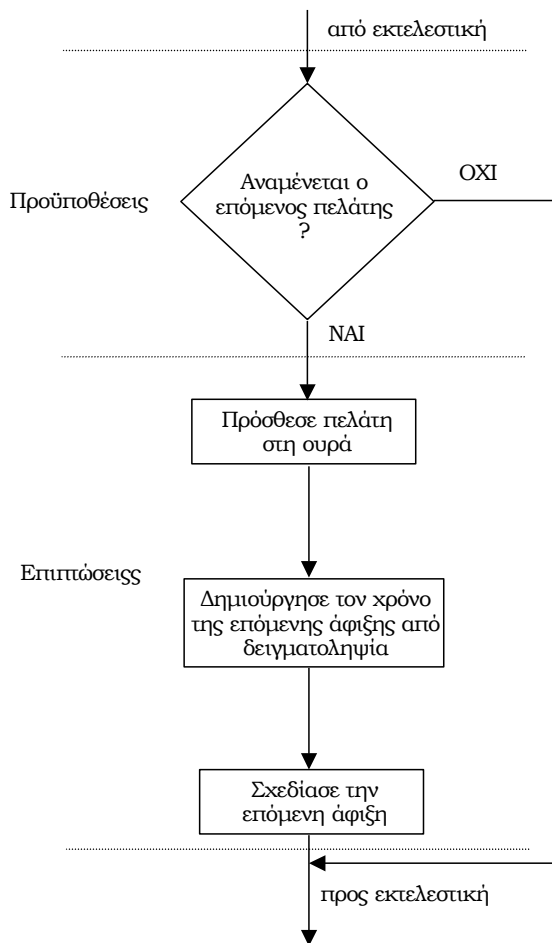


Σχήμα 2.38: Εκτελεστική ρουτίνα μεθόδου δραστηριοτήτων



Σχήμα 2.39: Έλεγχος δραστηριοτήτων για κάθε προώθηση χρόνου

Στη συνέχεια θα δώσουμε δύο παραδείγματα δραστηριοτήτων. Συγκεκριμένα, θα περιγράψουμε την δραστηριότητα άφιξης σε ένα σύστημα και την δραστηριότητα εξυπηρέτησης ενός πελάτη από ένα υπάλληλο.



Σχήμα 2.40: Δραστηριότητα απλής άφιξης **Σχήμα 2.41:** Δραστηριότητα εξυπηρέτησης

Εφαρμογή 1η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.2 (σελ 24).

Μεταβλητές Συστήματος:

- $NQ = n$ Το μήκος της ουράς είναι n πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 1.
= 2 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 3 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 3.
- $M2 = 2$ Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 4 Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 4.

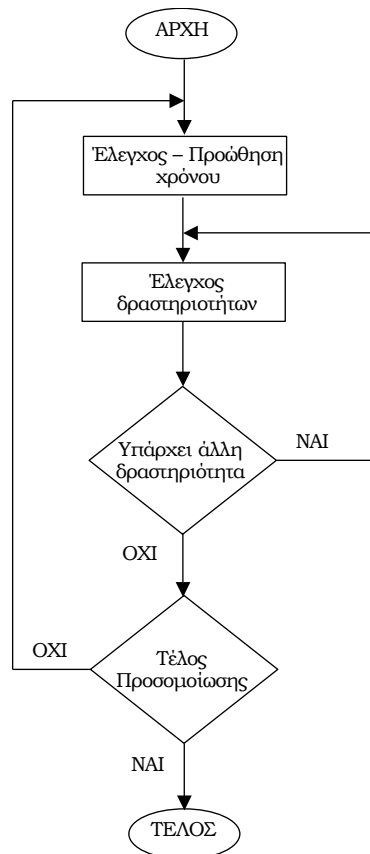
Δηλαδή οι μεταβλητές $M1$ και $M2$ παίρνουν ως τιμές τις τελευταίες ή τις τρέχουσες εργασίες.

- $T1 =$ Το ρολόι της μηχανής 1.
- $T2 =$ Το ρολόι της μηχανής 2.
- $T_A =$ Το ρολόι για τις αφίξεις.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

Παράμετροι Συστήματος:

- $ARRI$: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- $JOB1$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- $JOB2$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- $JOB3$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- $JOB4$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.

Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.2) με τη μέθοδο των δραστηριοτήτων θα πρέπει να περιγράψουμε τις δραστηριότητες: της άφιξης, της εργασίας 1, της εργασίας 2, της εργασίας 3 και της εργασίας 4 καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου.



Σχήμα 2.42: Εκτελεστική ρουτίνα μεθόδου δραστηριοτήτων

Δραστηριότητα της Άφιξης

Προϋποθέσεις: $T_A \leq ST$.

Συνέπειες: $T_A = ST + \text{Δείγμα}(\text{ARRI})$, $NQ = NQ + 1$.

Δραστηριότητα της Εργασίας 1

Προϋποθέσεις: $T_1 \leq ST$, $M1 = 3$, $NQ \neq 0$.

Συνέπειες: $T_1 = ST + \text{Δείγμα}(\text{JOB1})$, $M1 = 1$, $NQ = NQ - 1$.

Δραστηριότητα της Εργασίας 2

Προϋποθέσεις: $T_1, T_2 \leq ST$, $M1 = 1$, $M2 = 4$.

Συνέπειες: $T_1, T_2 = ST + \text{Δείγμα}(\text{JOB2})$, $M1 = M2 = 2$.

Δραστηριότητα της Εργασίας 3

Προϋποθέσεις: $T_1 \leq ST$, $M1 = 2$.

Συνέπειες: $T_1 = ST + \text{Δείγμα}(\text{JOB3})$, $M1 = 3$.

Δραστηριότητα της Εργασίας 4

Προϋποθέσεις: $T_2 \leq ST$, $M2 = 2$.

Συνέπειες: $T_2 = ST + \text{Δείγμα}(\text{JOB4})$, $M2 = 4$.

Εφαρμογή 2η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.10 (σελ. 31).

Μεταβλητές Συστήματος:

- $NQ = n$ Το μήκος της ουράς είναι n πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 1.
= 2 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 3 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 3.
= 4 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 4.
- $M2 = 2$ Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 5 Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 5.
- $M3 = 3$ Η μηχανή 3 κάνει ή έχει κάνει τελευταία την εργασία 3.
= 6 Η μηχανή 3 κάνει ή έχει κάνει τελευταία την εργασία 6.
- $T_1 =$ Το ρολόι της μηχανής 1.
- $T_2 =$ Το ρολόι της μηχανής 2.
- $T_3 =$ Το ρολόι της μηχανής 3.
- $T_A =$ Το ρολόι για τις αφίξεις.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

Παράμετροι Συστήματος:

- ARRI: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.
- JOB5: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 5.
- JOB6: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 6.

Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.10) με τη μέθοδο των δραστηριοτήτων θα πρέπει να περιγράψουμε τις δραστηριότητες: της άφιξης, της εργασίας 1, της εργασίας 2, της εργασίας 3, της εργασίας 4, της εργασίας 5 και της εργασίας 6 καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου (σχήμα 2.38, σελ 44).

Δραστηριότητα της Άφιξης

Προϋποθέσεις: $T_A \leq ST$.

Συνέπειες: $T_A = ST + \text{Δείγμα(ARRI)}$, $NQ = NQ + 1$.

Δραστηριότητα της Εργασίας 1

Προϋποθέσεις: $T_1 \leq ST$, $M1 = 4$, $NQ \neq 0$.

Συνέπειες: $T_1 = ST + \text{Δείγμα(JOB1)}$, $M1 = 1$, $NQ = NQ - 1$.

Δραστηριότητα της Εργασίας 2

Προϋποθέσεις: $T_1, T_2 \leq ST$, $M1 = 4$, $M2 = 5$.

Συνέπειες: $T_1, T_2 = ST + \text{Δείγμα(JOB2)}$, $M1 = M2 = 2$.

Δραστηριότητα της Εργασίας 3

Προϋποθέσεις: $T_1, T_3 \leq ST$, $M1 = 2$, $M3 = 6$.

Συνέπειες: $T_1, T_3 = ST + \text{Δείγμα(JOB3)}$, $M1 = M3 = 3$.

Δραστηριότητα της Εργασίας 4

Προϋποθέσεις: $T_1 \leq ST$, $M1 = 3$.

Συνέπειες: $T_1 = ST + \text{Δείγμα(JOB4)}$, $M1 = 4$.

Δραστηριότητα της Εργασίας 5

Προϋποθέσεις: $T_2 \leq ST$, $M2 = 2$.

Συνέπειες: $T_2 = ST + \text{Δείγμα(JOB5)}$, $M2 = 5$.

Δραστηριότητα της Εργασίας 6

Προϋποθέσεις: $T_3 \leq ST$, $M3 = 3$.

Συνέπειες: $T_3 = ST + \text{Δείγμα(JOB6)}$, $M3 = 6$.

Εφαρμογή 3η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.19 (σελ. 35).

Μεταβλητές Συστήματος:

- $NQ_A = n$ Το μήκος της ουράς A είναι n πελάτες.
- $NQ_B = m$ Το μήκος της ουράς B είναι m πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 1.
= 2 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 4 Η μηχανή 1 κάνει ή έχει κάνει τελευταία την εργασία 4.
- $M2 = 4$ Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 4.
= 5 Η μηχανή 2 κάνει ή έχει κάνει τελευταία την εργασία 5.
- $M3 = 2$ Η μηχανή 3 κάνει ή έχει κάνει τελευταία την εργασία 2.
= 3 Η μηχανή 3 κάνει ή έχει κάνει τελευταία την εργασία 3.
- $T_1 =$ Το ρολόι της μηχανής 1.
- $T_2 =$ Το ρολόι της μηχανής 2.
- $T_3 =$ Το ρολόι της μηχανής 3.
- $T_A =$ Το ρολόι για τις αφίξεις τύπου A.
- $T_B =$ Το ρολόι για τις αφίξεις τύπου B.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

Παράμετροι Συστήματος:

- ARRIA: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου A.
- ARRIB: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου B.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.
- JOB5: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 5.

Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.19) με τη μέθοδο των δραστηριοτήτων θα πρέπει να περιγράψουμε τις δραστηριότητες: της άφιξης τύπου Α, της άφιξης τύπου Β, της εργασίας 1, της εργασίας 2, της εργασίας 3, της εργασίας 4 και της εργασίας 5 καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου (σχήμα 2.38, σελ 44).

Δραστηριότητα της Άφιξης τύπου Α

Προϋποθέσεις: $T_A \leq ST$.

Συνέπειες: $T_A = ST + \text{Δείγμα}(\text{ARRIA})$, $NQ_A = NQ_A + 1$.

Δραστηριότητα της Εργασίας 1

Προϋποθέσεις: $T_1 \leq ST$, $M1 = 4$, $NQ_A \neq 0$.

Συνέπειες: $T_1 = ST + \text{Δείγμα}(\text{JOB1})$, $M1 = 1$, $NQ_A = NQ_A - 1$.

Δραστηριότητα της Εργασίας 2

Προϋποθέσεις: $T_1, T_3 \leq ST$, $M1 = 1$, $M3 = 3$.

Συνέπειες: $T_1, T_3 = ST + \text{Δείγμα}(\text{JOB2})$, $M1 = M3 = 2$.

Δραστηριότητα της Εργασίας 3

Προϋποθέσεις: $T_3 \leq ST$, $M3 = 2$.

Συνέπειες: $T_3 = ST + \text{Δείγμα}(\text{JOB3})$, $M3 = 3$.

Δραστηριότητα της Εργασίας 4

Προϋποθέσεις: $T_1, T_2 \leq ST$, $M1 = 2$, $M2 = 5$, $NQ_B \neq 0$.

Συνέπειες: $T_1, T_2 = ST + \text{Δείγμα}(\text{JOB4})$, $M1 = M2 = 4$, $NQ_B = NQ_B - 1$.

Δραστηριότητα της Εργασίας 5

Προϋποθέσεις: $T_2 \leq ST$, $M2 = 4$.

Συνέπειες: $T_2 = ST + \text{Δείγμα}(\text{JOB5})$, $M2 = 5$.

Σύγκριση των μεθόδων γεγονότων και δραστηριοτήτων

Γεγονότα	Δραστηριότητες
Μοντέλα δύσκολα να αναπτυχθούν.	Μοντέλα εύκολα να αναπτυχθούν
Είναι απαραίτητη η ύπαρξη μιας δυναμικά εξελισσόμενης λίστας γεγονότων.	Δεν χρειάζεται λίστα.
Πιο αποτελεσματική αφού η λίστα προσδιορίζει τι συμβαίνει και πότε.	Συνήθως γίνεται αρκετό μη παραγωγικό ψάξιμο.
Χρειάζεται ξεχωριστό πέρασμα για κάθε γεγονός.	Ταυτόχρονα γεγονότα γίνονται με ένα ταυτόχρονο πέρασμα.
Προβλήματα προτεραιότητας ξεπερνιούνται με extra tests στις μεταβλητές απόφασης που περιλαμβάνονται στα blocks γεγονότων για να καθορίσουν ποια γεγονότα μπορούν να σχεδιασθούν.	Προβλήματα προτεραιότητας.

2.5 Μέθοδος Διαδικασιών

Η μέθοδος συνδυάζει χαρακτηριστικά και από τις δύο προηγούμενες μεθόδους.

Αποτελείται από ένα σύνολο διαδικασιών. Κάθε μία διαδικασία αποτελεί ξεχωριστό κομμάτι του προγράμματος προσομοίωσης.

Παράδειγμα:

Διαδικασία απλής αναμονής και εξυπηρέτησης από ένα υπάλληλο σε ένα γκισέ.

Η διαδικασία του πελάτη θα είναι:

- Άφιξη του πελάτη στο σύστημα.
- Αναμονή μέχρι ο πελάτης να φτάσει στην κορυφή της ουράς.
- Μετακίνηση του πελάτη στον γκισέ όταν ελευθερωθεί ο υπάλληλος.
- Παραμονή στο γκισέ μέχρι το τέλος της εξυπηρέτησης του πελάτη από τον υπάλληλο.
- Αναχώρηση του πελάτη από το σύστημα.

Δομή της Μεθόδου

Η διαδικασία ορίζεται σαν μια σειρά λειτουργιών μέσα από τις οποίες περνάει μια οντότητα κατά τη διάρκεια της ζωής της στο σύστημα.

Η οντότητα παραμένει στη διαδικασία μέχρι την επαναδραστηριοποίηση της.

Η μέθοδος αντιμετωπίζει κάθε οντότητα ως μετακινούμενη μέσα από τις διάφορες λειτουργίες από τις οποίες αποτελείται η διαδικασία.

Η προώθησή της συνεχίζεται μέχρι να εμποδιστεί ή καθυστερήσει για κάποιο λόγο.

Δύο τύποι καθυστερήσεων υπάρχουν:

1. Καθυστερήσεις χωρίς συνθήκες, π.χ. χρόνος εξυπηρέτησης (προκαθορισμένος)
2. Καθυστερήσεις με συνθήκες (περίμενε μέχρι), π.χ. ο πελάτης περιμένει μέχρι να έρθει στην κορυφή της ουράς και ελευθερωθεί ο υπάλληλος.

Συνήθως υπάρχουν δύο λίστες αρχείων (records):

1. Λίστα των μελλοντικών γεγονότων.
2. Λίστα των τρεχόντων γεγονότων.

Η λίστα των μελλοντικών γεγονότων περιλαμβάνει αυτές τις οντότητες των οποίων η κίνηση καθυστερεί και των οποίων ο χρόνος επαναδραστηριοποίησης είναι γνωστός. Π.χ. αναμενόμενη άφιξη ή προσχεδιασμένο τέλος εξυπηρέτησης.

Η λίστα των τρεχόντων γεγονότων περιλαμβάνει δύο τύπους οντοτήτων:

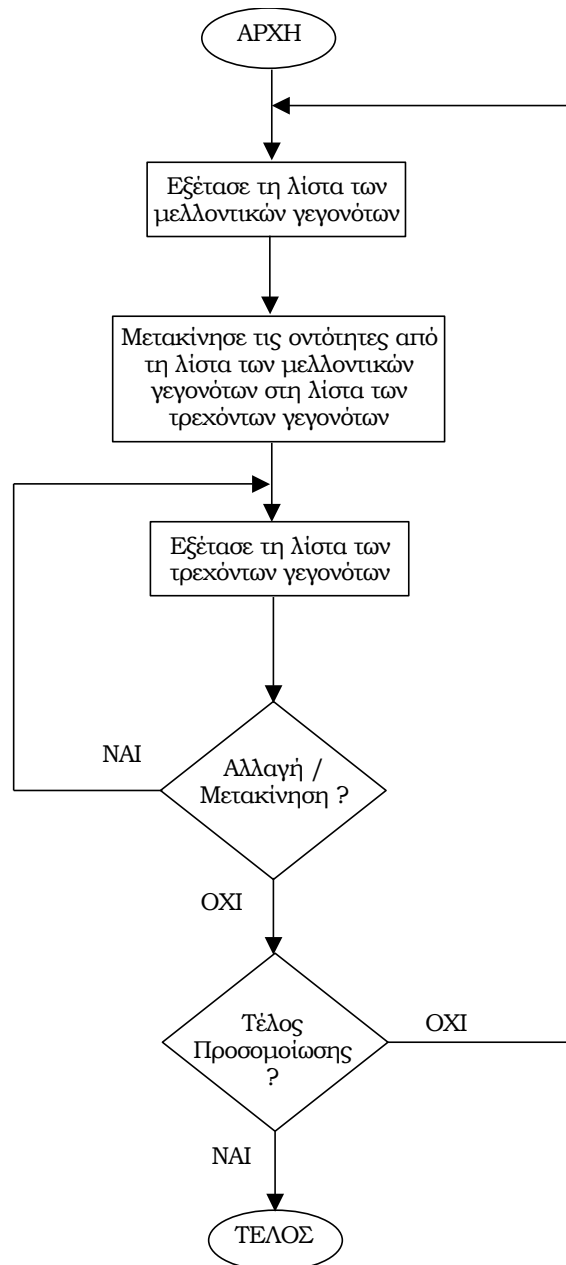
1. Αυτές των οποίων η κίνηση καθυστερεί και των οποίων η επαναδραστηριοποίηση είναι σχεδιασμένη για τον τρέχοντα χρόνο προσομοίωσης. Π.χ. μια άφιξη είναι σχεδιασμένη για τώρα.
2. Αυτές των οποίων η κίνηση καθυστερεί από τις προϋποθέσεις της προσομίωσης και η επαναδραστηριοποίησή τους θα γίνει όταν ικανοποιηθούν αυτές οι προϋποθέσεις. Π.χ. ένας πελάτης (οντότητα) μπορεί να παραμένει στη λίστα μετά την άφιξη του και μέχρι να φτάσει στην κεφαλή της ουράς.

Διαδικασία της Μεθόδου

Η διαδικασία προσομοίωσης με τη μέθοδο των διαδικασιών ακολουθεί τα εξής επαναληπτικά βήματα:

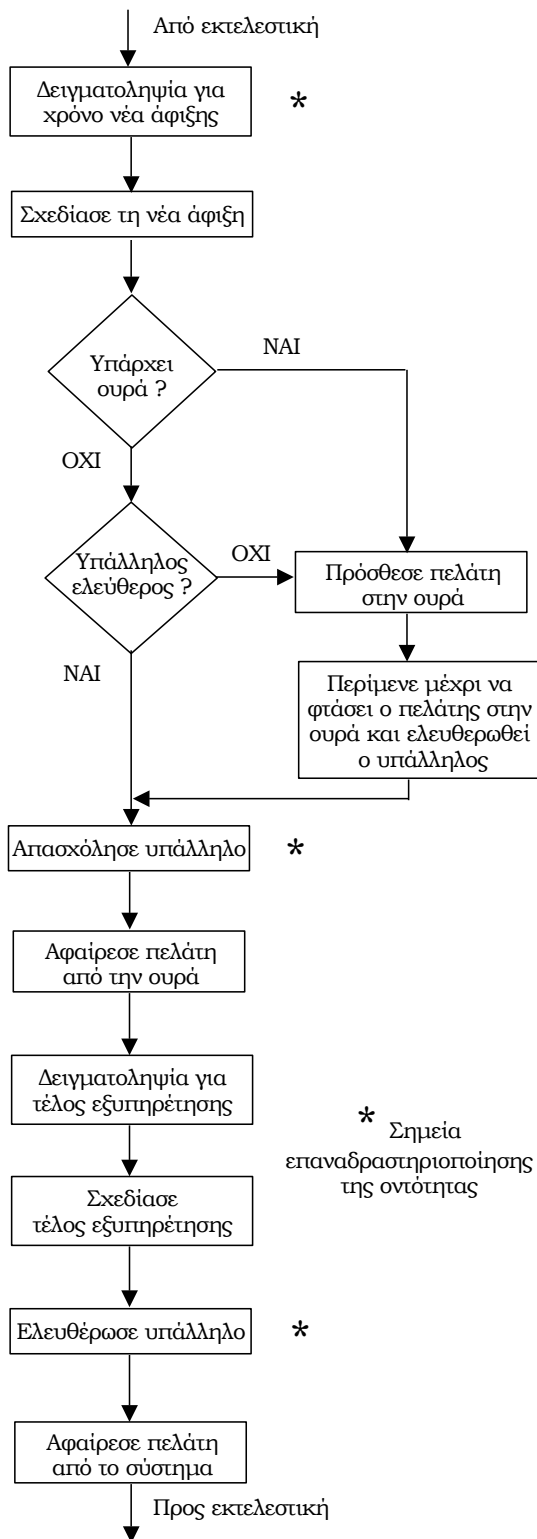
1. Εξετάζει τη λίστα των μελλοντικών γεγονότων. Από τη λίστα αυτή επιλέγονται οι οντότητες με το πιο κοντινό χρόνο επαναδραστηριοποίησης. Το ρολόι προσομοίωσης μετακινείται και δείχνει το χρόνο αυτό.
2. Μετακινεί τις οντότητες που τώρα θα είναι τρέχουσες από τη λίστα των μελλοντικών γεγονότων στη λίστα των τρεχόντων γεγονότων.
3. Εξετάζει τη λίστα των τρεχόντων γεγονότων. Εδώ γίνεται προσπάθεια να κινηθεί κάθε οντότητα που βρίσκεται στη λίστα των τρεχόντων γεγονότων μέσα από τη διαδικασία της, από το σημείο επαναδραστηριοποίησης της.
 - Εάν η οντότητα κινηθεί τότε σημειώνει που τελικώς σταματά πάλι, δηλαδή σημειώνει το χρόνο που η οντότητα σταματά πάλι.
 - Εάν η καθυστέρηση που έχει τώρα η οντότητα είναι χωρίς συνθήκες τότε ανοίγει ένα νέο αρχείο στη λίστα των μελλοντικών γεγονότων. Το αρχείο αυτό θα πρέπει να αναγνωρίζει το σημείο (χρόνο) επαναδραστηριοποίησης της οντότητας.
 - Εάν η καθυστέρηση είναι με συνθήκες τότε ανοίγει ένα αρχείο στη λίστα των τρεχόντων γεγονότων.
 - Και στις δύο περιπτώσεις (καθυστέρηση με συνθήκες / χωρίς συνθήκες) αφαιρεί το αρχείο της οντότητας από τη λίστα των τρεχόντων γεγονότων. Επίσης εάν μία οντότητα τελειώσει τη διαδικασία της τότε αφαιρεί το αρχείο της από όλες τις λίστες.
 - Επαναλαμβάνει την εξέταση των τρεχόντων γεγονότων μέχρι να μη κινείται καμία οντότητα.

Η διαδικασία αυτή, γενικά, αναπαρίσταται στο λογικό διάγραμμα (Σχήμα 2.43) της εκτελεστικής ρουτίνας της μεθόδου, που ακολουθεί.

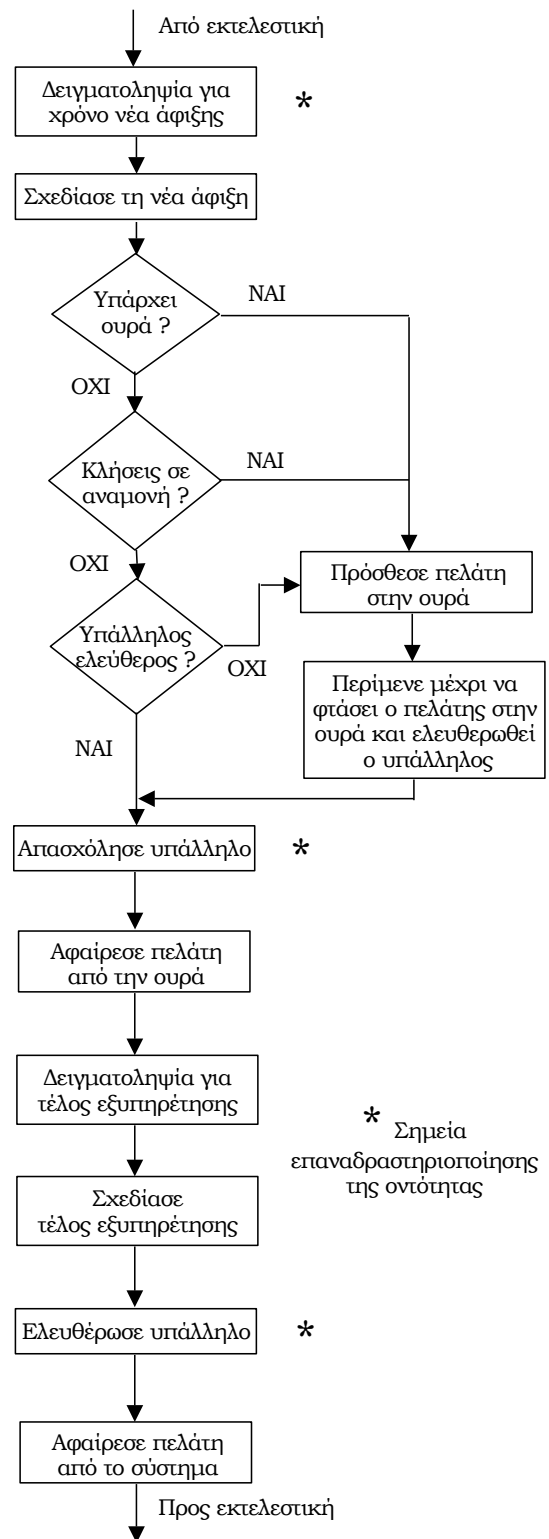


Σχήμα 2.43: Εκτελεστική ρουτίνα μεθόδου διαδικασιών

Στη συνέχεια θα δώσουμε δύο παραδείγματα διαδικασιών. Συγκεκριμένα, θα περιγράψουμε την διαδικασία άφιξης και εξυπηρέτησης σε ένα σύστημα και την διαδικασία άφιξης και εξυπηρέτησης από ένα υπάλληλο που απαντάει και σε τηλεφωνικές κλήσεις, οι τηλεφωνικές κλήσεις έχουν προτεραιότητα από τη εξυπηρέτηση του πελάτη.



Σχήμα 2.44: Διαδικασία απλής εξυπηρέτησης



Σχήμα 2.45: Διαδικασία εξυπηρέτησης τηλεφωνικών κλήσεων και πελατών

Εφαρμογή 1η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.2 (σελ 24).

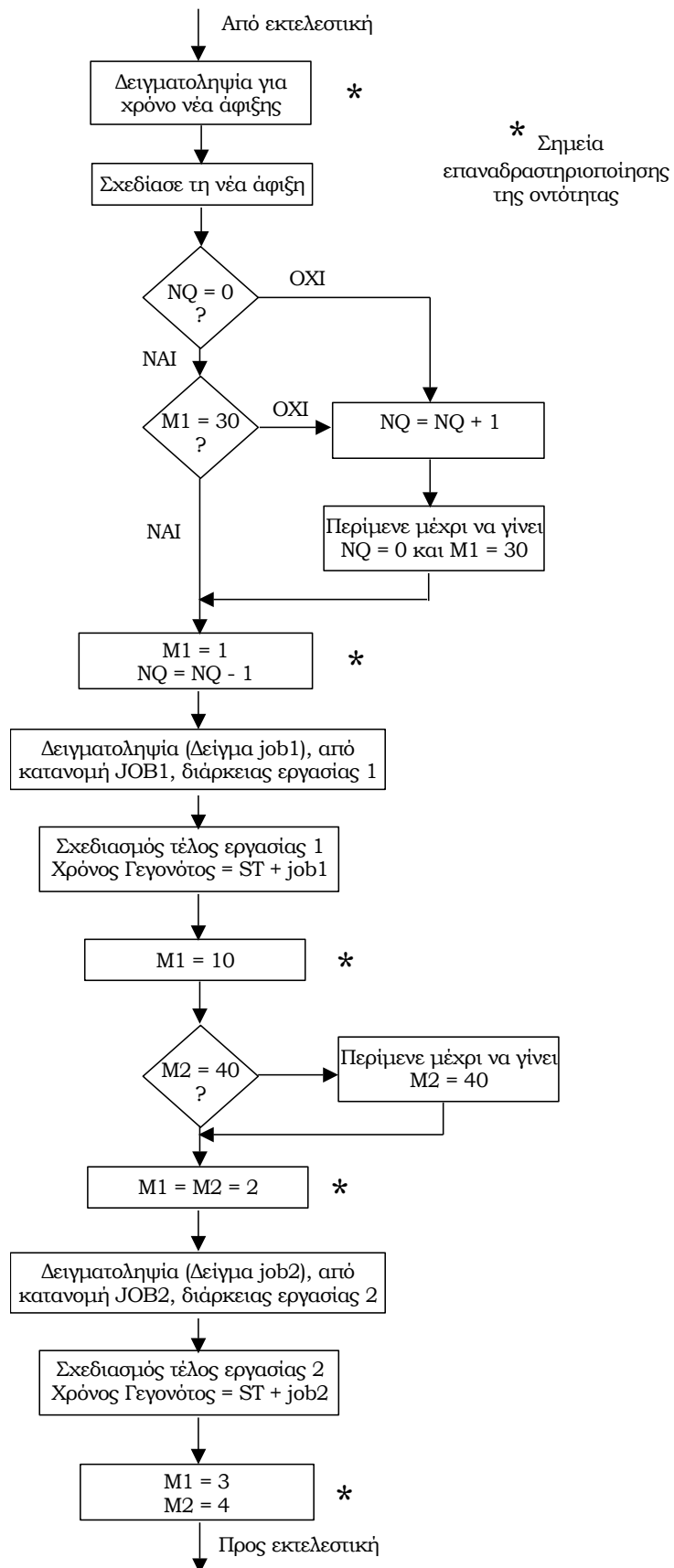
Μεταβλητές Συστήματος:

- $NQ = n$ Το μήκος της ουράς είναι n πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει την εργασία 1.
= 2 Η μηχανή 1 κάνει την εργασία 2.
= 3 Η μηχανή 1 κάνει την εργασία 3.
= 10 Η μηχανή 1 έχει κάνει την εργασία 1 και περιμένει.
= 30 Η μηχανή 1 έχει κάνει την εργασία 3 και περιμένει.
- $M2 = 2$ Η μηχανή 2 κάνει την εργασία 2.
= 4 Η μηχανή 2 κάνει την εργασία 4.
= 40 Η μηχανή 2 έχει κάνει την εργασία 4 και περιμένει.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

Παράμετροι Συστήματος:

- ARRI: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων.
- JOB1: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- JOB2: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- JOB3: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- JOB4: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.

Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.2) με τη μέθοδο των διαδικασιών θα πρέπει να περιγράψουμε τη διαδικασία της οντότητας “πελάτης” καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου (σχήμα 2.43 σελ 55).



Σχήμα 2.46: Διαδικασία εξυπηρέτησης πελάτη

Εφαρμογή 2η

Έστω ότι έχουμε το μοντέλο προσομοίωσης του σχήματος 2.19 (σελ 35).

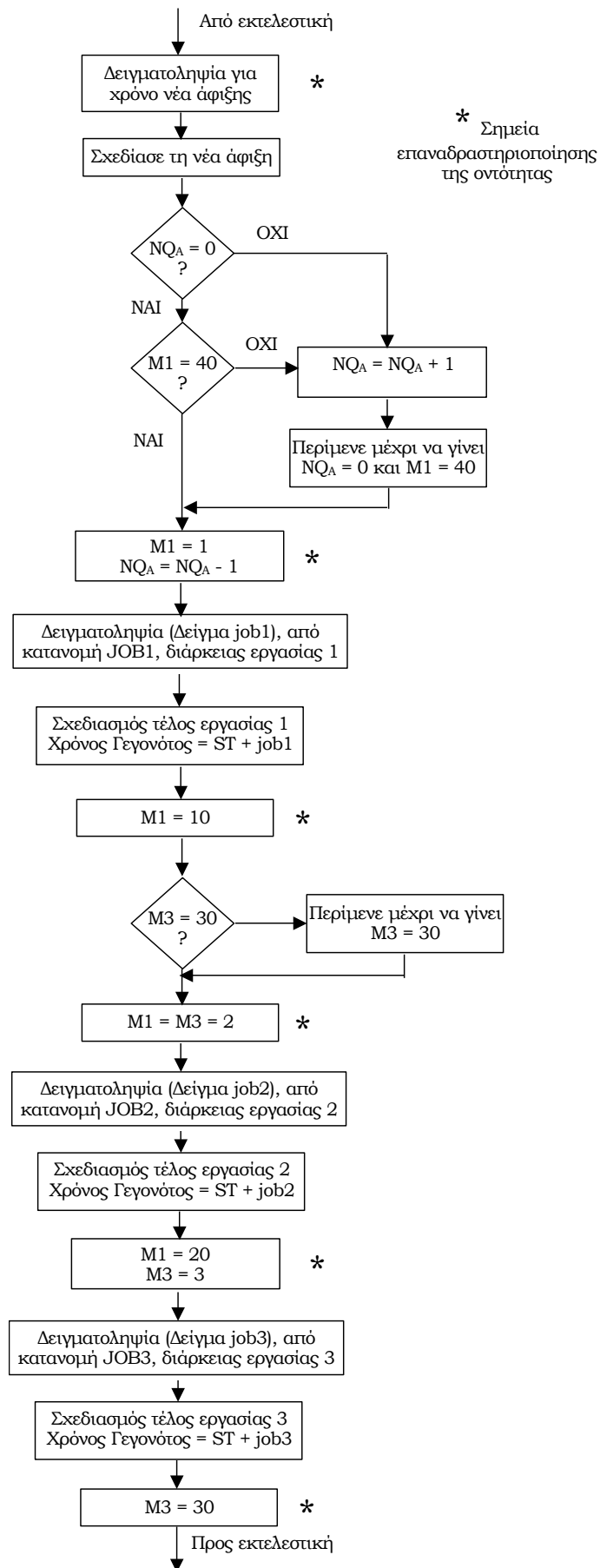
Μεταβλητές Συστήματος:

- $NQ_A = n$ Το μήκος της ουράς A είναι n πελάτες.
- $NQ_B = m$ Το μήκος της ουράς B είναι m πελάτες.
- $M1 = 1$ Η μηχανή 1 κάνει την εργασία 1.
= 2 Η μηχανή 1 κάνει την εργασία 2.
= 4 Η μηχανή 1 κάνει την εργασία 4.
= 10 Η μηχανή 1 έχει κάνει την εργασία 1 και περιμένει.
= 20 Η μηχανή 1 έχει κάνει την εργασία 2 και περιμένει.
= 40 Η μηχανή 1 έχει κάνει την εργασία 4 και περιμένει.
- $M2 = 4$ Η μηχανή 2 κάνει την εργασία 4.
= 5 Η μηχανή 2 κάνει την εργασία 5.
= 50 Η μηχανή 2 έχει κάνει την εργασία 5 και περιμένει.
- $M3 = 3$ Η μηχανή 3 κάνει την εργασία 3.
= 2 Η μηχανή 3 κάνει την εργασία 2.
= 30 Η μηχανή 3 έχει κάνει την εργασία 3 και περιμένει.
- $ST =$ Ο τρέχων χρόνο προσομοίωσης (ρολόι προσομοίωσης).

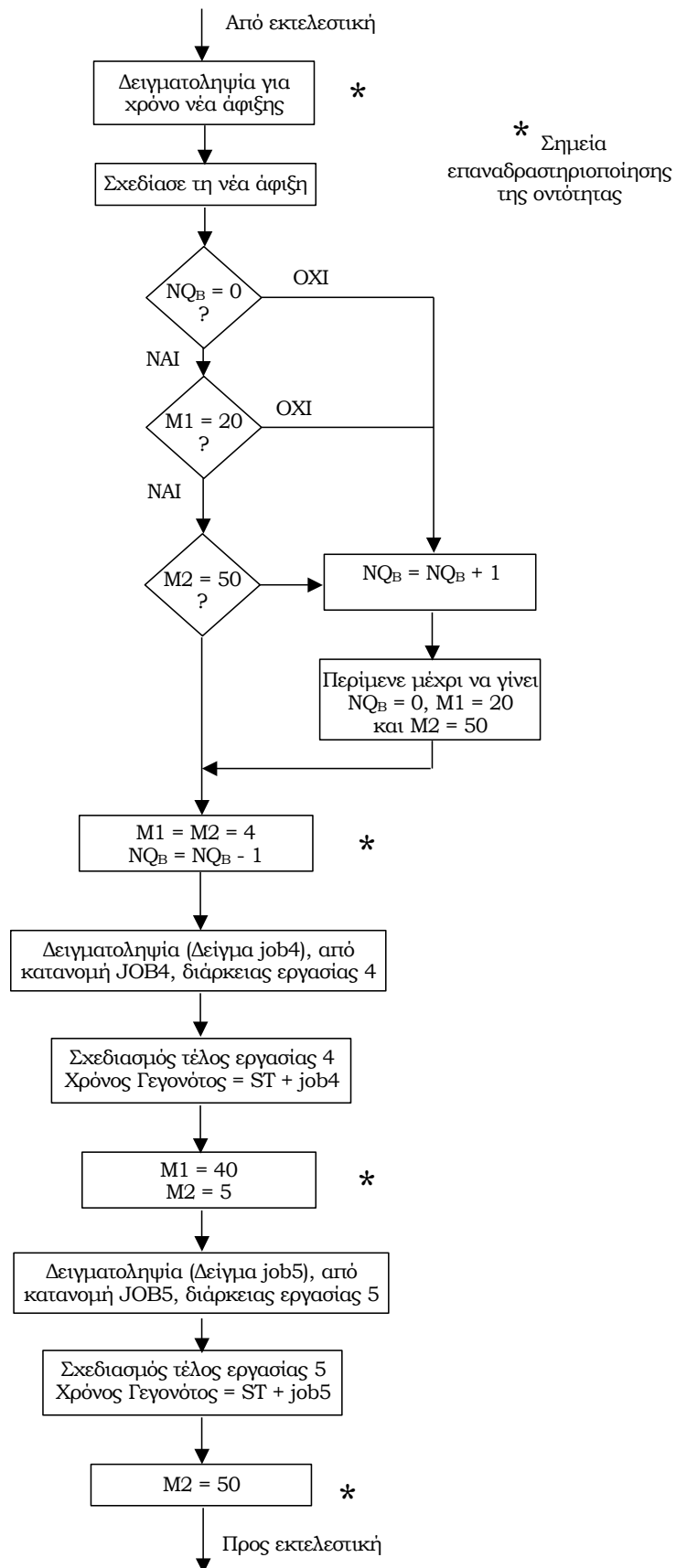
Παράμετροι Συστήματος:

- $ARRIA$: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου A.
- $ARRIB$: Είναι η κατανομή που ακολουθούν τα ενδιάμεσα χρονικά διαστήματα μεταξύ δύο αφίξεων τύπου B.
- $JOB1$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 1.
- $JOB2$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 2.
- $JOB3$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 3.
- $JOB4$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 4.
- $JOB5$: Είναι η κατανομή που ακολουθούν οι χρόνοι διάρκειας της εργασίας 5.

Για να προσομοιώσουμε το παραπάνω μοντέλο προσομοίωσης (σχήμα 2.19) με τη μέθοδο των διαδικασιών θα πρέπει να περιγράψουμε τη διαδικασία της οντότητας “πελάτης τύπου A”, τη διαδικασία της οντότητας “πελάτης τύπου B” καθώς επίσης και την εκτελεστική ρουτίνα της μεθόδου (σχήμα 2.43 σελ 55).



Σχήμα 2.47: Διαδικασία εξυπηρέτησης πελάτη τύπου Α



Σχήμα 2.48: Διαδικασία εξυπηρέτησης πελάτη τύπου Β

ΚΕΦΑΛΑΙΟ 3ο
ΤΥΧΑΙΟΙ ΑΡΙΘΜΟΙ – ΕΛΕΓΧΟΣ ΤΥΧΑΙΟΤΗΤΑΣ

3.1 Τυχαίοι αριθμοί

Στην προσομοίωση διακριτών γεγονότων γίνεται χρήση ακολουθίας τυχαίων αριθμών στις περιπτώσεις που απαιτείται η δημιουργία στοχαστικών μεταβλητών και παραμέτρων σε κατάλληλες διακριτές στιγμές κατά τη διάρκεια της προσομοίωσης.

Μια ακολουθία από αριθμούς που παράγονται από κάποια πηγή θεωρείται τυχαία όταν ικανοποιούνται οι εξής δύο βασικές ιδιότητες:

1. Όλοι οι αριθμοί της ακολουθίας είναι ομοιόμορφα κατανεμημένοι στο πεδίο ορισμού τους.
2. Οι παραγόμενοι αριθμοί είναι στατιστικά ανεξάρτητοι.

Η πρώτη ιδιότητα σημαίνει, ότι στην ακολουθία των αριθμών που παράγει η πηγή κάθε νέος αριθμός έχει την ίδια πιθανότητα να πάρει οποιαδήποτε τιμή μέσα στο πεδίο ορισμού της ακολουθίας.

Η δεύτερη ιδιότητα σημαίνει, ότι κάθε παραγόμενος αριθμός είναι στατιστικά ανεξάρτητος από τους άλλους αριθμούς της ακολουθίας που έχουν ήδη εμφανιστεί και δεν επηρεάζει την σειρά εμφάνισης των επομένων αριθμών. Δηλαδή, δεν είναι δυνατόν να γίνει χρήση ιστορικών πληροφοριών έτσι ώστε να προβλεφθούν μελλοντικά στοιχεία της ακολουθίας.

Γεννήτριες τυχαίων αριθμών

Παραδείγματα μεθόδων δημιουργίας τυχαίων αριθμών είναι γνωστά από την αρχαιότητα ακόμα, όπως το ρίξιμο ζαριών, το μοίρασμα χαρτιών, η λοταρία κ.λ.π. Πολλές από τις μεθόδους αυτές χρησιμοποιούνται ακόμα και σήμερα είτε σε τυχερά παιχνίδια είτε σε κληρώσεις. Η πιο συνηθισμένη μέθοδος στις κληρώσεις είναι η τυχαία επιλογή μιας αριθμημένης σφαίρας μέσα από ένα καλά ανακατεμένο δοχείο. Η ρουλέτα είναι επίσης ένα κλασσικό παράδειγμα δημιουργίας τυχαίων αριθμών.

Επίσης, είναι δυνατή η δημιουργία τυχαίων αριθμών και με ηλεκτρονικά μέσα όπως οι γεννήτριες λευκού θορύβου. Η γεννήτρια του λευκού θορύβου παράγει ηλεκτρονικό θόρυβο με την ίδια ισχύ σε όλες τις συχνότητες. Ο θόρυβος αυτός εισάγεται σε μια γεννήτρια παλμών, η οποία παράγει παλμούς όταν η ένταση του θορύβου είναι μεγαλύτερη από κάποια προκαθορισμένη τιμή. Τέλος, ένας μετρητής μετρά τους παλμούς που δημιουργήθηκαν σε κάποιο προκαθορισμένο χρονικό διάστημα. Ο αριθμός των παλμών είναι ο παραγόμενος τυχαίος αριθμός.

Μια άλλη μέθοδος παραγωγής τυχαίων αριθμών με ηλεκτρονικά μέσα χρησιμοποιεί παλλόμενες ηλεκτρονικές λυχνίες για τη δημιουργία τυχαίων αριθμών με βάση τους παλμούς που δημιουργούνται σε ένα προκαθορισμένο χρονικό διάστημα.

Τυχαίοι αριθμοί που έχουν δημιουργηθεί με παρόμοιες μεθόδους είναι διαθέσιμοι με τη μορφή πινάκων οι οποίοι μπορούν να εισαχθούν ως δεδομένα

σε ένα πρόγραμμα. Για παράδειγμα, η εταιρεία Rand Corporation (1955) δημιούργησε πίνακα με 1.000.000 τυχαίους αριθμούς. Η χρήση όμως αυτών των πινάκων, παρά το γεγονός ότι είναι διαθέσιμοι σε ψηφιακή μορφή, δεν ενδείκνυται για δύο λόγους. Ο πρώτος είναι η αδυναμία αποθήκευσης των τεραστίων αυτών πινάκων στην περιορισμένη μνήμη ενός υπολογιστή, επειδή αυτό θα αποτελούσε σπατάλη πολύτιμων πόρων. Επίσης αν υποθεθεί ότι αποθηκεύονται σε σκληρούς δίσκους, η χαμηλή ταχύτητα πρόσβασης σε αυτούς θα καθιστούσε την προσομοίωση απαράδεκτα αργή. Ο δεύτερος λόγος είναι ότι σε πολλές προσομοιώσεις απαιτείται η δημιουργία πολύ περισσότερων από 1.000.000 τυχαίων αριθμών, γιατί η επαναχρησιμοποίηση των ίδιων αριθμών θα προκαλούσε στατιστικά απαράδεκτη πόλωση στα αποτελέσματα.

3.2 Ψευδοτυχαίοι αριθμοί

Σήμερα, για τους λόγους που αναφέρθηκαν παραπάνω, στην πράξη χρησιμοποιούνται αποκλειστικά σχεδόν, οι ηλεκτρονικοί υπολογιστές για τη δημιουργία τυχαίων αριθμών. Ο συνηθέστερος τρόπος παραγωγής είναι η χρησιμοποίηση μιας επαναληπτικής διαδικασίας, επειδή έχει το πλεονέκτημα να απασχολεί ελάχιστο τμήμα της μνήμης του υπολογιστή και τη δυνατότητα να παράγει τους τυχαίους αριθμούς με μεγάλη ταχύτητα. Ένα άλλο πλεονέκτημα της χρήσης των υπολογιστών, πολύ χρήσιμο σε συγκρίσεις εναλλακτικών λύσεων, είναι η δυνατότητα επανάληψης μιας σειράς τυχαίων αριθμών.

Οι τυχαίοι αριθμοί που παράγονται από ένα ηλεκτρονικό υπολογιστή ονομάζονται ψευδοτυχαίοι, επειδή έχουν το μειονέκτημα σε κάποιο σημείο να κλείνει ο κύκλος παραγωγής τους και η εμφάνισή τους να συνεχίζεται σε δεύτερο κύκλο με την ίδια ακριβώς σειρά όπως στο πρώτο. Αυτό πάντως δεν επηρεάζει την γνησιότητα μιας προσομοίωσης όταν ο κύκλος είναι αρκετά μεγάλος, ώστε να μην εξαντλείται πριν τελειώσει η προσομοίωση.

Μία γεννήτρια ψευδοτυχαίων αριθμών θα πρέπει να ικανοποιεί τις παρακάτω ιδιότητες:

1. Οι αριθμοί που παράγει πρέπει να είναι όσο το δυνατόν πιο ανεξάρτητοι μεταξύ τους. Στην ιδανική περίπτωση δεν πρέπει να υπάρχει αυτοσυσχέτιση μεταξύ τους. Δηλαδή για κάποιον που δεν γνωρίζει τον αλγόριθμο της επαναληπτικής διαδικασίας οι αριθμοί που γεννιούνται να είναι τυχαίοι.
2. Θα πρέπει να είναι γρήγορη και να μην απαιτεί μεγάλη υπολογιστική μνήμη.
3. Οι αριθμοί που παράγει θα πρέπει να έχουν μεγάλο κύκλο παραγωγής (περίοδος).
4. Θα πρέπει να έχει την δυνατότητα να αναπαράγει την ίδια ακολουθία αριθμών, ώστε να επαναλαμβάνεται η προσομοίωση του μοντέλου, για διαφορετικά εναλλακτικά σενάρια, στις ίδιες συνθήκες.

3.3 Γεννήτριες ψευδοτυχαίων αριθμών

Στις επόμενες παραγράφους θα παρουσιάσουμε ορισμένες αντιπροσωπευτικές τεχνικές παραγωγής (γεννήτριες) ψευδοτυχαίων αριθμών.

3.3.1 Μέθοδος Μέσων Τετραγώνων

Μια από τις πρώτες επαναληπτικές διαδικασίες που χρησιμοποιήθηκε για την παραγωγή ψευδοτυχαίων αριθμών είναι η μέθοδος του μέσων τετραγώνων. Στη μέθοδο αυτή, που προτάθηκε από τον Von Neuman το 1948, κάθε νέος αριθμός της ακολουθίας των ψευδοτυχαίων παράγεται από τα ρ μεσαία ψηφία του τετραγώνου ενός ρ -ψήφιου αριθμού. Η μέθοδος ακολουθεί τα εξής βήματα:

1. Επιλέγουμε ένα ρ -ψήφιο αριθμό.
2. Το αριθμό αυτό τον υψώνουμε στο τετράγωνο και προσθέτουμε μηδενικά στα αριστερά του αριθμού, αν χρειάζονται για να γίνει 2ρ -ψήφιος αριθμός.
3. Τα ρ μεσαία ψηφία του αποτελούν τον επόμενο ψευδοτυχαίο αριθμό.
4. Επαναλαμβάνουμε τα βήματα 2 και 3 για κάθε νέο ψευδοτυχαίο αριθμό που δημιουργούμε.

Η μέθοδος των μέσων τετραγώνων, παρά την απλότητά της, δεν χρησιμοποιείται στην πράξη, γιατί παρουσιάζει αρκετά μειονεκτήματα. Μειονέκτημα της μεθόδου είναι ότι η ακολουθία των ψευδοτυχαίων που δημιουργείται επαναλαμβάνει τον εαυτό της με μικρή περίοδο, δηλαδή έχουμε σύντομα επανάληψη ψευδοτυχαίου αριθμού. Ένα άλλο μειονέκτημα της μεθόδου είναι η εμφάνιση του αριθμού 0 σε κάποια φάση της διαδικασίας, κάτι που όμως αντιμετωπίζεται με προγραμματισμένη αντικατάστασή του.

Ας δούμε μερικά παραδείγματα της μεθόδου:

i	x_i	x_i^2
1	21	0441
2	44	1936
3	93	8649
4	64	4096
5	09	0081
6	08	0064
7	06	0036
8	03	0009
9	00	0000
10	00	0000

Η περίοδος της παραγόμενης ακολουθίας ψευδοτυχαίων αριθμών είναι 9.

i	x_i	x_i^2
1	4500	20250000
2	2500	06250000
3	2500	06250000

Η περίοδος της παραγόμενης ακολουθίας ψευδοτυχαίων αριθμών είναι 2.

3.3.2 Μέθοδος Μέσων Γινομένων

Η μέθοδος αυτή μοιάζει με τη μέθοδο των μέσων τετραγώνων με τη μόνη διαφορά ότι κάθε νέος ψευδοτυχαίος αριθμός προκύπτει από τα μεσαία ρ ψηφία του γινομένου των δύο προηγούμενων ρ -ψήφιων αριθμών της ακολουθίας των ψευδοτυχαίων.

Τα βήματα της μεθόδου είναι:

1. Αρχικά επιλέγουμε δύο ρ -ψήφιους αριθμούς.
2. Τους αριθμούς αυτούς τους πολλαπλασιάζουμε και προσθέτουμε στα αριστερά του γινομένου μηδενικά, αν χρειάζονται για να γίνει 2ρ -ψήφιος το γινόμενο.
3. Τα ρ μεσαία ψηφία του γινομένου αποτελούν τον επόμενο ψευδοτυχαίο αριθμό.
4. Πολλαπλασιάζουμε τον νέο ψευδοτυχαίο αριθμό με τον προηγούμενο αριθμό της ακολουθίας των ψευδοτυχαίων και επαναλαμβάνουμε το βήμα 3.

Παράδειγμα:

i	x_i	$x_i \cdot x_{i-1}$
1	21	
2	20	0420
3	42	0840
4	84	3528
5	52	4368
6	36	1872
7	87	8132
8	13	κ.λ.π.

Η μέθοδος των μέσων γινομένων δεν εφαρμόζεται στη πράξη για παρόμοιους λόγους με τη μέθοδο των μέσων τετραγώνων.

3.3.3 Congruential Μέθοδοι

Οι Congruential μέθοδοι που εισήγαγε πρώτος ο Lehmer (1951) είναι οι πιο ευρέως χρησιμοποιούμενες επαναληπτικές διαδικασίες γέννησης ψευδοτυχαίων αριθμών.

Υπάρχουν τρεις τύποι congruential μεθόδων:

- Ο πολλαπλασιαστικός τύπος:

$$x_{n+1} = (a \cdot x_n) \bmod m$$

- Ο μικτός τύπος:

$$x_{n+1} = (a \cdot x_n + c) \bmod m$$

- Ο προσθετικός (αθροιστικός) τύπος:

$$x_{n+1} = (x_n + x_{n-1}) \bmod m$$

όπου a , c και m είναι σταθερές και x_0 είναι η αρχική τιμή (seed) που χρησιμοποιείται για τη δημιουργία της ακολουθίας των ψευδοτυχαίων.

Είναι προφανές ότι οι ψευδοτυχαίοι μπορούν να πάρουν τιμές μεταξύ 0 και $m-1$, δηλαδή $0 \leq x_i \leq m-1$, αφού η πράξη $(\bmod m)$ σημαίνει το υπόλοιπο της διαίρεσης με m και άρα δεν είναι δυνατόν να δώσει ποτέ αποτέλεσμα $\geq m$.

Επομένως οι congruential μέθοδοι μπορούν να δώσουν ψευδοτυχαίους αριθμούς μεταξύ 0 και 1, δηλαδή $0 \leq r_i \leq 1$, εάν θέσουμε $r_i = \frac{x_i}{m}$.

Επίσης και σ' αυτές τις επαναληπτικές διαδικασίες έχουμε ανακύκλωση, αλλά γίνονται αποδεκτές εφόσον ο κύκλος της ακολουθίας των ψευδοτυχαίων είναι μεγαλύτερος από την ακολουθία τυχαίων αριθμών που χρειαζόμαστε στην προσομοίωση.

Στις επόμενες παραγράφους θα παρουσιάσουμε τους δύο πρώτους τύπους, δηλαδή των πολλαπλασιαστικό και τον μικτό.

3.3.3.1 Πολλαπλασιαστικός τύπος

Είναι ο απλούστερος τύπος congruential μεθόδου:

$$x_{n+1} = (a \cdot x_n) \bmod m.$$

Για να υπάρξει μεγάλος κύκλος ψευδοτυχαίων αριθμών πρέπει να γίνει η κατάλληλη επιλογή των a , m και x_0 . Τα x_0 και m πρέπει να πρώτοι αριθμοί μεταξύ τους.

Κύκλος ψευδοτυχαίων με m τιμές δεν μπορεί να υπάρξει αφού η τιμή 0 δεν μπορεί να βρίσκεται στην ακολουθία των ψευδοτυχαίων. Άρα η μέγιστη δυνατή

περίοδος μιας ακολουθίας ψευδοτυχαίων που γεννιούνται με τον πολλαπλασιαστικό τύπο είναι $m-1$ αριθμοί.

Επειδή οι congruential μέθοδοι χρησιμοποιούνται σε ηλεκτρονικούς υπολογιστές για να έχουν καλύτερα υπολογιστικά αποτελέσματα θα πρέπει το m να είναι δύναμη του 2, δηλαδή $m = 2^b$.

Έχει αποδειχτεί ότι η μέγιστη περίοδος ψευδοτυχαίων που παράγονται με τον πολλαπλασιαστικό τύπο, όταν $m = 2^b$ και ισχύουν οι συνθήκες:

$$i) \ a \bmod 8 = \pm 3 \text{ ή } 5 \text{ και}$$

$$ii) \ x_0 \text{ και } m \text{ πρώτοι αριθμοί μεταξύ τους,}$$

είναι ίση με 2^{b-2} .

Παράδειγμα:

Έστω $m = 16 = 2^4$, $a = 5$ ($5 \bmod 8 = 5$) και $x_0 = 3$, τότε έχουμε:

i	x_i	$5x_i$	$5x_i \bmod 16$
1	3	15	15
2	15	75	11
3	11	55	7
4	7	35	3
5	3		

Επομένως η περίοδος της παραγόμενης ακολουθίας των ψευδοτυχαίων είναι $4 = 2^{4-2} = 2^2$.

3.3.2 Μικτός τύπος

Είναι ο γενικός τύπος της congruential μεθόδου:

$$x_{n+1} = (a \cdot x_n + c) \bmod m.$$

Για να υπάρξει μεγάλος κύκλος ψευδοτυχαίων αριθμών πρέπει να γίνει η κατάλληλη επιλογή των a , c , m και x_0 .

Και στον μικτό τύπο, για καθαρά υπολογιστικούς σκοπούς αφού η μέθοδος χρησιμοποιείται σε ηλεκτρονικούς υπολογιστές, πρέπει το m να είναι δύναμη του 2, δηλαδή $m = 2^b$.

Έχει αποδειχτεί ότι η μέγιστη περίοδος ακολουθίας ψευδοτυχαίων που παράγονται με τον μικτό τύπο, όταν $m = 2^b$ και ισχύουν οι συνθήκες:

$$i) \ a \bmod 4 = 1 \text{ και}$$

$$ii) \ c \text{ και } m \text{ πρώτοι αριθμοί μεταξύ τους,}$$

είναι ίση με $m = 2^b$.

Παράδειγμα:

Έστω $m = 8 = 2^3$, $a = 5$ ($5 \bmod 4 = 1$), $c = 3$ και $x_0 = 3$, τότε έχουμε:

i	x_i	$5x_i + 3$	$5x_i + 3 \bmod 8$
1	3	18	2
2	2	13	5
3	5	28	4
4	4	23	7
5	7	38	6
6	6	33	1
7	1	8	0
8	0	3	3
9	3		

Επομένως η περίοδος της παραγόμενης ακολουθίας των ψευδοτυχαίων είναι $8 = 2^3 = m$.

3.4 Έλεγχος τυχαιότητας

Στις προηγούμενες παραγράφους εξετάσαμε τρόπους με τους οποίους μπορούμε να δημιουργήσουμε ψευδοτυχαίους αριθμούς. Όπως είδαμε, οι αριθμοί αυτοί δεν μπορούν να είναι εντελώς τυχαίοι, αφού δεν δημιουργούνται τυχαία αλλά με μια επαναληπτική διαδικασία. Αυτό όμως που μας ενδιαφέρει είναι οι αριθμοί που παράγονται από τις επαναληπτικές διαδικασίες να έχουν τις στατιστικές ιδιότητες των τυχαίων αριθμών. Γι αυτό πριν χρησιμοποιήσουμε μια ακολουθία ψευδοτυχαίων αριθμών που δημιουργείται από μια επαναληπτική διαδικασία θα πρέπει να ελέγξουμε κατά πόσο αυτοί μπορούν να θεωρηθούν στατιστικώς τυχαίοι. Μια σειρά από στατιστικούς ελέγχους τυχαιότητας χρησιμοποιούνται για τον έλεγχο ψευδοτυχαίων αριθμών. Μερικούς από αυτούς θα τους παρουσιάσουμε στις επόμενες παραγράφους.

3.4.1 χ^2 -test

Ο έλεγχος αυτός διατυπώθηκε από τον Karl Pearson το 1903, στηρίζεται στην χ^2 κατανομή και χρησιμοποιείται για τον έλεγχο της ομοιομορφίας. Με τον έλεγχο αυτό συγκρίνουμε τη συχνότητα των παρατηρούμενων δεδομένων με τη συχνότητα των αναμενόμενων δεδομένων σύμφωνα με μια θεωρητική κατανομή ως προς τη οποία γίνεται ο έλεγχος.

Συγκεκριμένα ο K. Pearson απόδειξε ότι η μεταβλητή

$$\chi^2 = \sum_{j=1}^n \frac{(O_j - E_j)^2}{E_j}$$

όπου O_j = είναι τα παρατηρούμενα δεδομένα στο j διάστημα (κλάση δεδομένων),

E_j = είναι τα αναμενόμενα δεδομένα, σύμφωνα με τη θεωρητική κατανομή, στο διάστημα (κλάση δεδομένων) j ,

ακολουθεί την κατανομή X^2 με $v = n-1-\lambda$ βαθμούς ελευθερίας. όπου n είναι ο αριθμός των κλάσεων (διαστημάτων) που χωρίζονται τα δεδομένα και λ είναι ο αριθμός των παραμέτρων της θεωρητικής κατανομής που πρέπει να εκτιμηθούν από τα στατιστικά δεδομένα.

Αν $x^2 = 0$, τότε η παρατηρούμενη συχνότητα και η αναμενόμενη συμφωνούν απόλυτα, ενώ αν $x^2 > 0$ δεν συμφωνούν. Όσο μεγαλύτερη είναι η τιμή του x^2 , τόσο μεγαλύτερη είναι η ασυμφωνία ανάμεσα στη παρατηρούμενη και την αναμενόμενη συχνότητα. Αν $x^2 > 0$, τότε πρέπει να συγκρίνουμε την τιμή της μεταβλητής x^2 που υπολογίσαμε με εκείνες που δίνονται από τους πίνακες της κατανομής X^2 για να αποφασίσουμε σε ορισμένο επίπεδο σημαντικότητας. Στην πράξη ελέγχουμε την υπόθεση ομοιομορφίας H_0 που λέει ότι δεν υπάρχει σημαντική διαφορά ανάμεσα στην παρατηρούμενη και στην αναμενόμενη συχνότητα δεδομένων, δηλαδή η παρατηρούμενη συχνότητα συμφωνεί με την αναμενόμενη. Αν κάτω από την υπόθεση αυτή η τιμή της μεταβλητής x^2 που υπολογίζουμε είναι μεγαλύτερη από τη κρίσιμη τιμή που δίνεται από τον πίνακα της X^2 κατανομής σε ορισμένο επίπεδο σημαντικότητας και με τους κατάλληλους βαθμούς ελευθερίας, μπορούμε να συμπεράνουμε ότι η παρατηρούμενη συχνότητα διαφέρει σημαντικά από την αναμενόμενη σ' αυτό το επίπεδο σημαντικότητας και έτσι απορρίπτεται η υπόθεση H_0 .

Βήματα ελέγχου x^2 -test:

1. Διατύπωσε την υπόθεση ομοιομορφίας H_0 : η παρατηρούμενη συχνότητα συμφωνεί με την αναμενόμενη συχνότητα.
2. Καθόρισε το επίπεδο σημαντικότητας α : 5% ή 1%.
3. Υπολόγισε την τιμή της μεταβλητής x^2 σύμφωνα με τον τύπο
$$x^2 = \sum_{j=1}^n \frac{(O_j - E_j)^2}{E_j}$$
 και καθόρισε τους αντίστοιχους βαθμούς ελευθερίας.
4. Σύγκρινε την τιμή της μεταβλητής x^2 που υπολογίσαμε με εκείνη που δίνεται από τους πίνακες της κατανομής X^2 για επίπεδο σημαντικότητας α και με τους αντίστοιχους βαθμούς ελευθερίας.

Ιδιότητες της X^2 κατανομής

1. Έστω μια ακολουθία τυχαίων μεταβλητών $Y_1, Y_2, \dots, Y_n$ που ακολουθούν την κανονική κατανομή με μέσο $E(Y) = \mu$ και διασπορά $Var(Y) = \sigma^2$ και $X_1, X_2, \dots, X_n$ οι αντίστοιχες τυποποιημένες τυχαίες μεταβλητές με μέσο $E(X) = 0$ και διασπορά $Var(X) = 1$, που δίνονται από τον τύπο: $X_i = \frac{Y_i - \mu}{\sigma}$. Τότε η μεταβλητή $X_1^2 + X_2^2 + \dots + X_n^2$ ακολουθεί την X^2 κατανομή με συνάρτηση πυκνότητας πιθανότητας την $f(x) = \frac{1}{(\frac{n}{2}-1)!} 2^{-\frac{n}{2}} x^{\frac{n}{2}-1} e^{-\frac{1}{2}x}$.

2. Έστω $X_1^2, X_2^2, \dots, X_k^2$ ακολουθία ανεξάρτητων μεταβλητών που ακολουθούν την X^2 κατανομή με βαθμούς ελευθερίας $n_1, n_2, \dots, n_k$. Τότε και η μεταβλητή $X_1^2 + X_2^2 + \dots + X_k^2$ ακολουθεί την X^2 κατανομή με $n_1+n_2+\dots+n_k$ βαθμούς ελευθερίας.

Παράδειγμα 1ο

Έστω ότι έχουμε την παρακάτω ακολουθία μονοψήφιων ψευδοτυχαίων αριθμών

8, 0, 2, 0, 1, 5, 8, 8, 9, 8, 6, 5, 8, 6, 7, 3, 2, 8, 6, 0,
6, 0, 2, 9, 1, 8, 9, 0, 9, 3, 7, 3, 2, 1, 4, 5, 7, 6, 9, 6,
9, 4, 5, 3, 5, 7, 9, 6, 4, 3, 6, 5, 8, 2, 9, 1, 0, 3, 2, 6,
6, 1, 5, 4, 7, 7, 1, 3, 9, 3, 8, 6, 1, 8, 6, 6, 5, 9, 0, 1,

και θέλουμε να ελέγξουμε κατά πόσο μπορεί να θεωρηθούν στατιστικά τυχαίοι, δηλαδή εάν εμφανίζονται με περίπου ίδια συχνότητα.

1. Αρχικά θα ορίσουμε την υπόθεση ομοιομορφίας H_0 : οι αριθμοί εμφανίζονται περίπου με την ίδια συχνότητα.

2. Στη συνέχεια ορίζουμε το επίπεδο σημαντικότητας. Έστω $\alpha = 5\% = 0.05$.

3. Μετά θα υπολογίσουμε τη μεταβλητή χ^2 από τον τύπο $\chi^2 = \sum_{j=1}^n \frac{(O_j - E_j)^2}{E_j}$ και θα υπολογίσουμε τους βαθμούς ελευθερίας.

Αφού οι αριθμοί είναι 80 και οι μονοψήφιοι αριθμοί 10 τότε αφού υποθέτουμε ότι οι ψευδοτυχαίοι εμφανίζονται με την ίδια συχνότητα η αναμενόμενη συχνότητα για κάθε μονοψήφιο είναι $E_j = \frac{80}{10} = 8$.

Στη συνέχεια μετράμε τις εμφανίσεις των μονοψήφιων αριθμών στην ακολουθία και έτσι έχουμε την παρατηρούμενη συχνότητα για κάθε μονοψήφιο, η οποία φαίνεται στο παρακάτω πίνακα.

Μονοψήφιος αριθμός	Παρατηρούμενη συχνότητα E_j	Αναμενόμενη συχνότητα O_j
0	7	8
1	8	8
2	6	8
3	8	8
4	4	8
5	8	8
6	13	8
7	6	8
8	10	8
9	10	8

Αφού έχουν βρεθεί οι παρατηρούμενες και οι αναμενόμενες συχνότητες για κάθε μονοψήφιο αριθμό μπορούμε να χρησιμοποιήσουμε τον τύπο για τον υπολογισμό της χ^2 μεταβλητής:

$$\chi^2 = \sum_{j=1}^{10} \frac{(O_j - E_j)^2}{E_j} = \frac{(7-8)^2}{8} + \frac{(8-8)^2}{8} + \frac{(6-8)^2}{8} + \frac{(8-8)^2}{8} + \frac{(4-8)^2}{8} + \frac{(8-8)^2}{8} + \frac{(13-8)^2}{8} + \frac{(6-8)^2}{8} + \frac{(10-8)^2}{8} + \frac{(10-8)^2}{8} = 7.2$$

Εφόσον ο αριθμός των κλάσεων (οι μονοψήφιοι αριθμοί) στις οποίες χωρίσαμε την ακολουθία των ψευδοτυχαίων είναι 10, τότε οι βαθμοί ελευθερίας είναι

$$v = k - 1 - \lambda = 10 - 1 - 0 = 9.$$

Το λ είναι ίσο με 0 επειδή δεν χρειάστηκε να εκτιμήσουμε καμία παράμετρο της θεωρητικής κατανομής.

4. Τέλος θα συγκρίνουμε την τιμή της μεταβλητής χ^2 που υπολογίσαμε με εκείνη που δίνεται από τους πίνακες της κατανομής χ^2 για επίπεδο σημαντικότητας 0.05 και με 9 βαθμούς ελευθερίας.

Από τους πίνακες της χ^2 κατανομής βρίσκουμε

$$\chi^2_{(1-\alpha);v} = \chi^2_{0.95;9} = 16,92.$$

Επειδή $7.2 < 16,92$ η υπόθεση ομοιομορφίας H_0 δεν απορρίπτεται. Επομένως οι ψευδοτυχαίοι αριθμοί της παραπάνω ακολουθίας, στο επίπεδο σημαντικότητας 0.05, μπορούν θεωρηθούν ότι εμφανίζονται με την ίδια συχνότητα.

Παράδειγμα 2ο

Υπάρχουν 5 τμήματα φοιτητών του Τμήματος Μαθηματικών, στο μάθημα της Προσομοίωσης, με 50 φοιτητές το καθένα. Τα αποτελέσματα προηγούμενων ετών έχουν δείξει ότι συνήθως το $\frac{1}{5}$ του κάθε τμήματος αποτυγχάνει στις τελικές εξετάσεις του μαθήματος. Τα αποτελέσματα της φετινής τελικής εξέτασης φαίνεται στο παρακάτω πίνακα:

Τμήμα	Αποτυχόντες	Επιτυχόντες
1	8	42
2	5	45
3	13	37
4	15	35
5	5	45

Τα 5 τμήματα μπορούν να θεωρηθούν ως 5 ανεξάρτητα δείγματα που χωρίζονται σε 2 κλάσεις ($k = 2$), τους αποτυχόντες και τους επιτυχόντες.

Έστω η υπόθεση ομοιομορφίας H_0 : το $\frac{1}{5}$ του κάθε τμήματος αποτυγχάνει στις τελικές εξετάσεις του μαθήματος.

Ορίζουμε το επίπεδο σημαντικότητας $\alpha = 0.05$.

Στη συνέχεια θα υπολογίσουμε τη μεταβλητή χ^2 , σύμφωνα με τον τύπο, για κάθε τμήμα ξεχωριστά, καθώς και τους αντίστοιχους βαθμούς ελευθερίας.

$$\chi_1^2 = \frac{(8-10)^2}{10} + \frac{(42-40)^2}{40} = 0.5 \text{ και } \nu_1 = k - 1 - \lambda = 2 - 1 - 0 = 1,$$

$$\chi_2^2 = \frac{(5-10)^2}{10} + \frac{(45-40)^2}{40} = 3.1 \text{ και } \nu_2 = k - 1 - \lambda = 2 - 1 - 0 = 1,$$

$$\chi_3^2 = \frac{(13-10)^2}{10} + \frac{(37-40)^2}{40} = 1.1 \text{ και } \nu_3 = k - 1 - \lambda = 2 - 1 - 0 = 1,$$

$$\chi_4^2 = \frac{(15-10)^2}{10} + \frac{(35-40)^2}{40} = 3.1 \text{ και } \nu_4 = k - 1 - \lambda = 2 - 1 - 0 = 1,$$

$$\chi_5^2 = \frac{(5-10)^2}{10} + \frac{(45-40)^2}{40} = 3.1 \text{ και } \nu_5 = k - 1 - \lambda = 2 - 1 - 0 = 1,$$

Αφού οι μεταβλητές χ_i^2 είναι ανεξάρτητες, επειδή αντιστοιχούν σε ανεξάρτητα δείγματα, θα ελέγξουμε αν το άθροισμα τους

$$\chi^2 = \chi_1^2 + \chi_2^2 + \chi_3^2 + \chi_4^2 + \chi_5^2 = 10.9$$

ακολουθεί την χ^2 κατανομή με βαθμούς ελευθερίας $\nu = \nu_1 + \nu_2 + \nu_3 + \nu_4 + \nu_5 = 1 + 1 + 1 + 1 + 1 = 5$.

Από τους πίνακες της χ^2 κατανομής, για επίπεδο σημαντικότητας 0.05 και για 5 βαθμούς ελευθερίας, βρίσκουμε

$$\chi_{(1-\alpha);\nu}^2 = \chi_{0.95;5}^2 = 11.1.$$

Επειδή $10.9 < 11.1$ η υπόθεση ομοιομορφίας H_0 δεν απορρίπτεται. Επομένως μπορούμε να πούμε ότι με βάση και τα φετινά αποτελέσματα το $\frac{1}{5}$ του κάθε τμήματος αποτυγχάνει στις τελικές εξετάσεις του μαθήματος της προσομοίωσης.

Η κυκλική ακολουθία 0, 1, ..., 9, 0, 1, ..., 9, ..., 0, 1, ..., 9 προφανώς δεν είναι ακολουθία τυχαίων αριθμών όμως περνάει τους έλεγχους συχνότητας όπως το χ^2 -test.

Για αυτό χρησιμοποιούνται άλλοι έλεγχοι οι οποίοι ελέγχουν την ανεξαρτησία των αριθμών (δεύτερη ιδιότητα τυχαίων αριθμών) ενώ οι έλεγχοι συχνότητας, όπως το χ^2 -test ελέγχουν την ομοιόμορφη κατανομή (πρώτη ιδιότητα τυχαίων αριθμών). Σε όλους τους ελέγχους υπολογίζονται τα αναμενόμενα (θεωρητικά αποτελέσματα), υποθέτοντας τυχειότητα, και συγκρίνονται με τα αντίστοιχα παρατηρούμενα.

Στη συνέχεια θα παρουσιάσουμε περιληπτικά μερικούς τέτοιους ελέγχους.

3.4.2 Poker test

Σ' αυτό τον έλεγχο δημιουργούμε από τους ψευδοτυχαίους της ακολουθίας που θέλουμε να ελέγξουμε 5-άδες και υπολογίζουμε όλες τις παρατηρούμενες πεντάδες, ως εξής:

- O_1 : οι 5-άδες με όλα τα νούμερα διαφορετικά,
- O_2 : οι 5-άδες με ένα ζευγάρι αριθμών και τρεις άλλους διαφορετικούς αριθμούς,
- O_3 : οι 5-άδες με δύο ζευγάρια αριθμών,
- O_4 : οι 5-άδες με μια τριάδα και ένα ζευγάρι αριθμών,
- O_5 : οι 5-άδες με τετράδες αριθμών και ένα άλλο διαφορετικό αριθμό,
- O_6 : οι 5-άδες με όλους τους αριθμούς ίδιους.

Στη συνέχεια με τη βοήθεια των πιθανοτήτων και της συνδυαστικής υπολογίζουμε τη πιθανότητα εμφάνισης αυτών των πεντάδων. Για παράδειγμα αν έχουμε μια ακολουθία μονοψήφιων ψευδοτυχαίων οι πιθανότητες αυτές είναι:

- f_1 : 0.3024 (αφού ο πρώτος αριθμός της πεντάδας γίνεται δεκτός με πιθανότητα 1, ο δεύτερος με πιθανότητα 0.9, ο τρίτος με πιθανότητα 0.8, ο τέταρτος με πιθανότητα 0.7 και ο πέμπτος με πιθανότητα 0.6, άρα $f_1 = 1 \times 0.9 \times 0.8 \times 0.7 \times 0.6 = 0.3024$),
- f_2 : 0.5040,
- f_3 : 0.1080,
- f_4 : 0.0810,
- f_5 : 0.0045,
- f_6 : 0.0001.

Αυτές τις πιθανότητες τις χρησιμοποιούμε για να υπολογίσουμε τις αναμενόμενες πεντάδες σύμφωνα με τον τύπο

$$E_j = f_j \cdot n, \quad j = 1, 2, \dots, 6$$

όπου n το πλήθος των δυνατών πεντάδων που σχηματίζονται από τους ψευδοτυχαίους.

Τέλος χρησιμοποιείται το χ^2 -test για να συγκρίνουμε τα αναμενόμενα με τα παρατηρούμενα αποτελέσματα.

3.4.3 Gap test

Σε αυτό τον έλεγχο υπολογίζεται η κατανομή των αριθμών, δηλαδή το πλήθος των αριθμών, μέχρι την επανάληψη του ίδιου αριθμού στη ακολουθία των ψευδοτυχαίων. Στη συνέχεια χρησιμοποιείται το χ^2 -test για να συγκριθούν τα αναμενόμενα με τα παρατηρούμενα αποτελέσματα.

3.4.4 Runs test

Αυτός ο έλεγχος βασίζεται στις κατανομές του μήκους (πλήθος αριθμών) των προς τα άνω ή των προς τα κάτω διαδρομών (runs) στη ακολουθία των ψευδοτυχαίων. Π.χ.

1, 4, 8, 9: προς τα πάνω διαδρομή με μήκος 4

9, 6, 3: προς τα κάτω διαδρομή με μήκος 3.

Στη συνέχεια χρησιμοποιείται το χ^2 -test για να συγκριθούν τα αναμενόμενα με τα παρατηρούμενα αποτελέσματα.

ΚΕΦΑΛΑΙΟ 4ο
ΔΕΙΓΜΑΤΟΛΗΨΙΑ ΑΠΟ ΣΥΝΕΧΕΙΣ
ΚΑΙ ΔΙΑΚΡΙΤΕΣ ΚΑΤΑΝΟΜΕΣ

4.1. Εισαγωγή

Στην προσομοίωση σε πολλές περιπτώσεις είναι απαραίτητη η δημιουργία δειγμάτων τυχαίων μεταβλητών που ακολουθούν κάποια καθορισμένη κατανομή πιθανότητας. Στις επόμενες παραγράφους παρουσιάζονται μέθοδοι παραγωγής είτε από συνεχείς κατανομές είτε από διακριτές κατανομές.

Συγκεκριμένα θα παρουσιαστούν δύο γενικές μέθοδοι, η μέθοδος της αντίστροφης και η μέθοδος της απόρριψης, για δειγματοληψία από συνεχείς κατανομές, η μέθοδος δειγματοληψίας από κανονική κατανομή, η μέθοδος δειγματοληψίας από διακριτή ομοιόμορφη κατανομή, η μέθοδος του αντίστροφου μετασχηματισμού για δειγματοληψία από διακριτές κατανομές, η μέθοδος δειγματοληψίας από διωνυμική κατανομή και τέλος η μέθοδος δειγματοληψίας από εμπειρική διακριτή κατανομή.

Σε όλες αυτές τις μεθόδους η δειγματοληψία του δείγματος της τυχαίας μεταβλητής που ακολουθεί κάποια καθορισμένη κατανομή πιθανότητας, ξεκινάει από μια δειγματοληψία τυχαίας μεταβλητής που ακολουθεί μία ομοιόμορφη κατανομή στο διάστημα $(0,1)$, δηλαδή ξεκινάει από έναν τυχαίο αριθμό στο διάστημα $[0,1]$. Η γέννηση αυτού του τυχαίου αριθμού επιτυγχάνεται με τη βοήθεια των γεννητριών ψευδοτυχαίων αριθμών που παρουσιάσθηκαν στο προηγούμενο κεφάλαιο.

4.2 Μέθοδος της αντίστροφης

Η μέθοδος αυτή χρησιμοποιεί την αντίστροφη συνάρτηση της αθροιστικής συνάρτησης πιθανότητας της κατανομής για να δημιουργήσει το δείγμα της τυχαίας μεταβλητής που ακολουθεί την κατανομή από έναν τυχαίο αριθμό στο διάστημα $[0, 1]$.

Έστω μια συνεχής κατανομή με συνάρτηση πυκνότητας πιθανότητας την $f(x)$, τότε η αθροιστική συνάρτηση πιθανότητας της κατανομής θα είναι η

$$F(x) = \int_{-\infty}^x f(x)dx.$$

Έστω η τυχαία μεταβλητή U που ακολουθεί την ομοιόμορφη κατανομή $U(0,1)$.

Η συνάρτηση πυκνότητας πιθανότητας της ομοιόμορφης κατανομής είναι:

$$f(x) = \frac{1}{\beta - \alpha} \quad \alpha < x < \beta$$

Αρα η αθροιστική συνάρτηση πιθανότητας της ομοιόμορφης κατανομής είναι:

$$F(x) = \int_{\alpha}^x f(x)dx = \int_{\alpha}^x \frac{1}{\beta - \alpha} dx = \frac{1}{\beta - \alpha} \int_{\alpha}^x dx = \frac{1}{\beta - \alpha} x \Big|_{\alpha}^x = \frac{x - \alpha}{\beta - \alpha}$$

Για την ομοιόμορφη κατανομή $U(0,1)$ ($\alpha=0, \beta=1$) έχουμε:

$$F(x) = \frac{x-0}{1-0} = x.$$

Επομένως

$$F(u) = u \Leftrightarrow P(U \leq u) = u.$$

Για να γεννήσουμε ένα δείγμα x μιας τυχαίας μεταβλητής X που ακολουθεί μια συνεχή κατανομή με δεδομένο ένα δείγμα u της τυχαίας μεταβλητής U που ακολουθεί την ομοιόμορφη κατανομή $U(0,1)$, πρέπει να βρούμε μια συνάρτηση G τέτοια ώστε:

$$x = G(u) \text{ και}$$

$$P(X \leq x) = P(U \leq u).$$

Έστω ότι υπάρχει αυτή η συνάρτηση G , τότε έχουμε:

$$P(X \leq x) = P(U \leq u) \Leftrightarrow F_X(x) = F_U(u) \Leftrightarrow F_X(G(u)) = F_U(u) \Leftrightarrow F_X(G(u)) = u \Leftrightarrow$$

$$F_X \circ G = I \Leftrightarrow G = F_X^{-1}.$$

Επίσης αν υπάρχει η αντίστροφη συνάρτηση της αθροιστικής συνάρτησης πιθανότητας, δηλαδή υπάρχει η F_X^{-1} τότε έχουμε:

$$P(X \leq x) = P(U \leq u) \Leftrightarrow F_X(x) = F_U(u) \Leftrightarrow F_X(x) = u \Leftrightarrow F_X^{-1}(F_X(x)) = F_X^{-1}(u) \Leftrightarrow$$

$$I_X(x) = F_X^{-1}(u) \Leftrightarrow x = F_X^{-1}(u).$$

Άρα $G = F_X^{-1}$.

Επομένως μια δειγματοληψία, με τη μέθοδο της αντίστροφης, από μία συνεχή κατανομή ακολουθεί τα επόμενα βήματα:

1. Ολοκληρώνουμε την $f(x)$ και παίρνουμε την $F(x)$.
2. Βρίσκουμε την αντίστροφη $F^{-1}(x)$ της $F(x)$.
3. Γεννάμε ένα τυχαίο δείγμα u από την ομοιόμορφη κατανομή $U(0,1)$.
4. Θέτουμε $x = F^{-1}(u)$.

ή

1. Ολοκληρώνουμε την $f(x)$ και παίρνουμε την $F(x)$.
2. Γεννάμε ένα τυχαίο δείγμα u από την ομοιόμορφη κατανομή $U(0,1)$.
3. Θέτουμε $u = F(x)$ και λύνουμε ως προς x .
4. Το x είναι ένα τυχαίο δείγμα της τυχαίας μεταβλητής X .

Μειονεκτήματα της μεθόδου:

1. Η $f(x)$ δεν είναι πάντα ολοκληρώσιμη.
2. Δεν υπάρχει πάντα η αντίστροφη της $F(x)$.

Παράδειγμα 1ο

Να γίνει δειγματοληψία, με τη μέθοδο της αντιστροφής, από μία εκθετική κατανομή με μέσο $1/\lambda$.

Η εκθετική κατανομή με μέσο $1/\lambda$ έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \lambda e^{-\lambda x} \quad x \geq 0.$$

$$1. F(x) = \int_0^x \lambda e^{-\lambda x} dx = 1 - e^{-\lambda x}.$$

2. Γεννάμε ένα τυχαίο δείγμα u της ομοιόμορφης κατανομής $U(0,1)$.

$$3. \text{Θέτουμε } u = 1 - e^{-\lambda x} \Leftrightarrow e^{-\lambda x} = 1 - u \Leftrightarrow -\lambda x = \ln(1-u) \Leftrightarrow x = \frac{1}{\lambda} \ln(1-u)$$

Εάν το u είναι τυχαίο δείγμα της $U(0,1)$ τότε και το $1-u$ είναι τυχαίο δείγμα της $U(0,1)$, άρα μπορούμε αντί για $1-u$ να χρησιμοποιήσουμε u . Δηλαδή:

$$x = \frac{1}{\lambda} \ln(u).$$

Παράδειγμα 2ο

Να γίνει δειγματοληψία, με τη μέθοδο της αντιστροφής, από μία ομοιόμορφη κατανομή $U(\alpha, \beta)$.

Η ομοιόμορφη κατανομή $U(\alpha, \beta)$ έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \frac{1}{\beta - \alpha} \quad \alpha < x < \beta.$$

$$1. F(x) = \int_{\alpha}^x f(x) dx = \int_{\alpha}^x \frac{1}{\beta - \alpha} dx = \frac{1}{\beta - \alpha} \int_{\alpha}^x dx = \frac{1}{\beta - \alpha} x \Big|_{\alpha}^x = \frac{x - \alpha}{\beta - \alpha}.$$

2. Γεννάμε ένα τυχαίο δείγμα u της ομοιόμορφης κατανομής $U(0,1)$.

$$3. \text{Θέτουμε } u = \frac{x - \alpha}{\beta - \alpha} \Leftrightarrow x - \alpha = u(\beta - \alpha) \Leftrightarrow x = \alpha + u(\beta - \alpha)$$

Παράδειγμα 3ο

Να γίνει δειγματοληψία, με τη μέθοδο της αντίστροφης, από μια συνεχή κατανομή που έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \begin{cases} 0 & x < 0 \\ 4x & 0 \leq x \leq 0.5 \\ 4(1-x) & 0.5 \leq x \leq 1 \\ 0 & x > 1 \end{cases}.$$

1. $F(x) = \int_0^x f(x)dx$. Εδώ διακρίνουμε δύο περιπτώσεις:

α) Έστω $0 \leq x \leq 0.5$ τότε $F(x) = \int_0^x 4x dx = 2x^2$.

β) Έστω $0.5 \leq x \leq 1$ τότε $F(x) = \int_0^x f(x)dx = \int_0^{0.5} 4x dx + \int_{0.5}^x 4(1-x)dx =$
 $= 2x^2 \Big|_0^{0.5} + \int_{0.5}^x 4 dx - \int_{0.5}^x 4x dx = 2(0.5)^2 - 0 + 4x \Big|_{0.5}^x - 2x^2 \Big|_{0.5}^x =$
 $= 0.5 + 4x - 4 \cdot (0.5) - 2x^2 + 2(0.5)^2 = 0.5 + 4x - 2 - 2x^2 + 0.5 =$
 $= -2x^2 + 4x - 1.$

Επομένως

$$F(x) = \begin{cases} 0 & x < 0 \\ 2x^2 & 0 \leq x \leq 0.5 \\ -2x^2 + 4x - 1 & 0.5 \leq x \leq 1 \\ 1 & x > 1 \end{cases}.$$

2. Γεννάμε ένα τυχαίο δείγμα u της ομοιόμορφης κατανομής $U(0,1)$.

3. Διακρίνουμε δύο περιπτώσεις:

α) Εάν $u \leq 0.5$ τότε έχουμε $u = 2x^2 \Leftrightarrow x^2 = \frac{u}{2} \Leftrightarrow x = \pm \sqrt{\frac{u}{2}} \Rightarrow$

$$x = \sqrt{\frac{u}{2}} \quad \text{και} \quad x = -\sqrt{\frac{u}{2}}.$$

Η τιμή $x = -\sqrt{\frac{u}{2}}$ απορρίπτεται αφού πρέπει $x \geq 0$. Επομένως

$$x = \sqrt{\frac{u}{2}}.$$

β) Εάν $u \geq 0.5$ τότε έχουμε $u = -2x^2 + 4x - 1 \Leftrightarrow 2x^2 - 4x + 1 + u = 0 \Rightarrow$

$$x_{1,2} = \frac{4 \pm \sqrt{16 - 4 \cdot 2 \cdot (1+u)}}{4} \Rightarrow x_1 = 1 + \sqrt{\frac{1-u}{2}} \quad \text{και} \quad x_2 = 1 - \sqrt{\frac{1-u}{2}}.$$

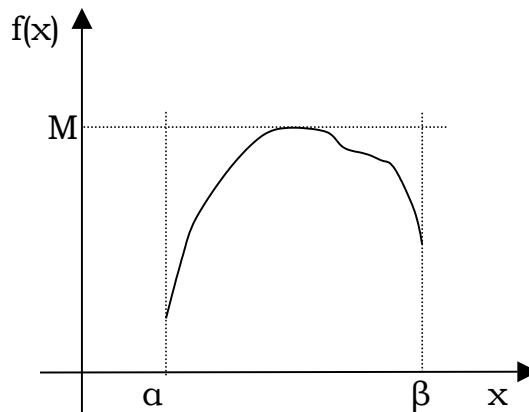
Η τιμή $x_2 = 1 - \sqrt{\frac{1-u}{2}}$ απορρίπτεται αφού πρέπει $x \leq 1$. Επομένως

$$x = 1 + \sqrt{\frac{1-u}{2}}.$$

4.3 Μέθοδος της απόρριψης

Η μέθοδος αυτή χρησιμοποιείται κυρίως στις περιπτώσεις όπου η συνάρτηση πυκνότητας πιθανότητας $f(x)$ δεν είναι ολοκληρώσιμη.

Έστω ότι έχουμε την παρακάτω φραγμένη συνάρτηση πυκνότητας πιθανότητας:



όπου

$$0 \leq f(x) \leq M \quad \text{για} \quad a \leq x \leq \beta.$$

Μια απλή διαδικασία δειγματοληψίας, από μια φραγμένη συνεχή κατανομή, με τη μέθοδο της απόρριψης ακολουθεί τα παρακάτω βήματα:

1. Επέλεξε μια σταθερά c , έτσι ώστε $c \cdot f(x) \leq 1$ για $a \leq x \leq \beta$.
2. Γεννήστε δύο τυχαία δείγματα u_1, u_2 από μία ομοιόμορφη κατανομή $U(0,1)$.
3. Θέστε $x = a + (\beta - a)u_1$.
4. Εάν $u_2 \leq c \cdot f(a + (\beta - a)u_1)$ τότε το x που υπολογίσαμε στο 3ο βήμα είναι αποδεκτό ως δείγμα της φραγμένης συνεχούς κατανομής. Σε αντίθετη περίπτωση γεννάμε δύο νέα τυχαία δείγματα u_1', u_2' και επαναλαμβάνουμε τα βήματα 3 και 4.

Παρατηρήσεις:

1. Το βήμα 3 ουσιαστικά σημαίνει ότι το διάστημα $[a, \beta]$ είναι ισοπίθανο στη δειγματοληψία.
2. Για ένα σημείο, στο διάστημα $[a, \beta]$, η πιθανότητα να γίνει αποδεκτό είναι ανάλογη της συνάρτησης πυκνότητας πιθανότητας της κατανομής στο σημείο αυτό. Π.χ. εάν $f(x_1) = y$ και $f(x_2) = 2y$ τότε τα x_1 και x_2 έχουν την ίδια πιθανότητα να εμφανισθούν στη δειγματοληψία αλλά το x_2 είναι δύο φορές πιο πιθανόν να γίνει αποδεκτό λόγω του βήματος 4.
3. Ελάττωμα της μεθόδου είναι ότι απαιτεί ορισμένο (φραγμένο) πεδίο ορισμού, ενώ για τις περισσότερες κατανομές έχουμε $0 \leq x < \infty$ ή $-\infty < x < \infty$. Όμως οι περισσότερες κατανομές μπορούν να περιορισθούν με ελάχιστο “χάσιμο” ακρίβειας.

Παράδειγμα 1ο

Να γίνει δειγματοληψία, με τη μέθοδο της απόρριψης, από μια συνεχή κατανομή που έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \begin{cases} 0 & x < 0 \\ 4x & 0 \leq x \leq 0.5 \\ 4(1-x) & 0.5 \leq x \leq 1 \\ 0 & x > 1 \end{cases}.$$

1. Επιλέγουμε $c = \frac{1}{2}$ αφού πρέπει $c \cdot f(x) \leq 1$. Πράγματι για

$$c \cdot f(x) = \begin{cases} 2x & 0 \leq x \leq 0.5 \\ 2(1-x) & 0.5 \leq x \leq 1 \end{cases}$$

έχουμε $c \cdot f(x) \leq 1$ για κάθε x στο διάστημα $[0, 1]$.

2. Γεννάμε δύο τυχαία δείγματα u_1, u_2 από μία ομοιόμορφη κατανομή $U(0,1)$. Διακρίνουμε δύο περιπτώσεις:

α) $0 \leq u_1 \leq 0.5$.

3. Θέτουμε $x = 0 + (0.5 - 0)u_1 \Rightarrow x = 0.5u_1$.

4. Εάν $u_2 \leq 2(0.5u_1) \Rightarrow u_2 \leq u_1$ τότε το τυχαίο δείγμα της κατανομής θα είναι το $x = 0.5u_1$, αλλιώς επαναλαμβάνουμε τα βήματα 2, 3 και 4 της μεθόδου.

β) $0.5 \leq u_1 \leq 1$.

3. Θέτουμε $x = 0.5 + (1 - 0.5)u_1 \Rightarrow x = 0.5 + 0.5u_1$.

4. Εάν $u_2 \leq 2(1 - 0.5 - 0.5u_1) \Rightarrow u_2 \leq 2(0.5 - 0.5u_1) \Rightarrow u_2 \leq 1 - u_1$ τότε το τυχαίο δείγμα της κατανομής θα είναι το $x = 0.5 + 0.5u_1$, αλλιώς επαναλαμβάνουμε τα βήματα 2, 3 και 4 της μεθόδου.

4.4 Δειγματοληψία από κανονική κατανομή $N(\mu, \sigma^2)$

Για την κανονική κατανομή $N(\mu, \sigma^2)$ η συνάρτηση πυκνότητας πιθανότητας είναι:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2},$$

ενώ για την κανονική κατανομή $N(0, 1)$ η συνάρτηση πυκνότητας πιθανότητας είναι:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}.$$

Επίσης ισχύει ότι εάν X μια τυχαία μεταβλητή που ακολουθεί την κανονική κατανομή $N(\mu, \sigma^2)$ τότε η τυχαία μεταβλητή Z που δίνεται από τον μετασχηματισμό $Z = \frac{X-\mu}{\sigma}$, ακολουθεί την κανονική κατανομή $N(0, 1)$.

Αντίστροφα, εάν Z μια τυχαία μεταβλητή που ακολουθεί την κανονική κατανομή $N(0, 1)$ τότε η τυχαία μεταβλητή X που δίνεται από τον μετασχηματισμό $X = Z \cdot \sigma + \mu$, ακολουθεί την κατανομή $N(\mu, \sigma^2)$.

Με βάση αυτή την ιδιότητα, η διαδικασία δειγματοληψίας από κανονική κατανομή $N(\mu, \sigma^2)$ ακολουθεί τα παρακάτω βήματα:

1. Δειγματοληψία τυχαίου δείγματος z από κανονική κατανομή $N(0, 1)$.
2. Θέσε ως τυχαίο δείγμα από κανονική κατανομή $N(\mu, \sigma^2)$ την τιμή x που δίνεται από τον μετασχηματισμό $x = \mu + \sigma \cdot z$.

Στη συνέχεια θα παρουσιάσουμε τη διαδικασία δειγματοληψίας από κανονική κατανομή $N(0,1)$.

Από το κεντρικό οριακό θεώρημα έχουμε ότι εάν X_i ανεξάρτητες τυχαίες μεταβλητές που ακολουθούν οποιαδήποτε κατανομή $\Delta(\mu, \sigma^2)$ με μέσο μ και διασπορά σ^2 τότε για $n \rightarrow \infty$ έχουμε ότι η τυχαία μεταβλητή $\sum_{i=1}^n X_i$ ακολουθεί

την κανονική κατανομή $N(\sum_{i=1}^n \mu, \sum_{i=1}^n \sigma^2)$.

Επομένως για τυχαία δείγματα u_i από την ομοιόμορφη κατανομή $U(0, 1)$ ($\mu_U = \frac{1}{2}, \sigma_U^2 = \frac{1}{12}$) με βάση το παραπάνω θεώρημα έχουμε ότι η τυχαία μεταβλητή $\sum_{i=1}^n u_i$ ακολουθεί την κανονική κατανομή $N(\frac{n}{2}, \frac{n}{12})$. Έτσι στη συνέχεια χρησιμοποιώντας τον μετασχηματισμό που παρουσιάσαμε πιο πάνω

έχουμε ότι η τυχαία μεταβλητή $\frac{\sum_{i=1}^n u_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}}$ ακολουθεί την κανονική κατανομή $N(0, 1)$.

Επομένως η διαδικασία δειγματοληψίας από κανονική κατανομή $N(0, 1)$ ακολουθεί τα παρακάτω βήματα:

1. Γέννηση τυχαίων δειγμάτων $u_1, u_2, \dots, u_n$ από ομοιόμορφη κατανομή $U(0, 1)$.

2. Θέτουμε $z = \frac{\sum_{i=1}^n u_i - \frac{n}{2}}{\sqrt{\frac{n}{12}}}$ ως τυχαίο δείγμα από κανονική κατανομή $N(0, 1)$.

Η μέθοδος είναι προσεγγιστική (ισχύει για $n \rightarrow \infty$) και δεν είναι πάντα κατάλληλη για περιπτώσεις όπου υπάρχουν αρκετές ακραίες τιμές. Μεγάλο n (δείγμα) προτιμάται για ακρίβεια και μικρό n για ταχύτητα.

4.5 Δειγματοληψία από ομοιόμορφη διακριτή κατανομή $U'(a, \beta)$

Η ομοιόμορφη διακριτή κατανομή $U'(a, \beta)$ έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x) = \frac{1}{\beta - \alpha + 1},$$

δηλαδή η πιθανότητα p , η τυχαία μεταβλητή X , που ακολουθεί την ομοιόμορφη διακριτή κατανομή, να πάρει μια τιμή στο διάστημα $[a, \beta]$ είναι:

$$p = P(X = i) = \frac{1}{\beta - \alpha + 1}, \quad x = a, a+1, \dots, \beta-1, \beta.$$

Η διαδικασία δειγματοληψίας από ομοιόμορφη διακριτή κατανομή $U'(a, \beta)$ ακολουθεί τα παρακάτω βήματα:

1. Γέννηση τυχαίου δείγματος u από ομοιόμορφη συνεχή κατανομή $U(0, 1)$.
2. Θέσε $x = a + [u \cdot (\beta - a + 1)]$.

Στο βήμα 2 οι αγκύλες συμβολίζουν το ακέραιο μέρος.

Αφού η πιθανότητα η τυχαία μεταβλητή X , που ακολουθεί την ομοιόμορφη διακριτή κατανομή, να πάρει μια τιμή στο διάστημα $[a, \beta]$ είναι:

$$p = P(X=i) = \frac{1}{\beta - \alpha + 1} \Rightarrow (\beta - \alpha + 1) \cdot p = 1 \Rightarrow (\beta - \alpha + 1) = \frac{1}{p},$$

τότε στο βήμα 2, θέτουμε $x = a + \left\lfloor \frac{u}{p} \right\rfloor$.

Επομένως το βήμα 2 δίνει, ως τυχαίο δείγμα της τυχαίας μεταβλητής X που ακολουθεί την ομοιόμορφη διακριτή κατανομή, τις εξής τιμές:

a	$0 \leq u \leq p$
$a+1$	$p \leq u \leq 2p$
$a+2$	$2p \leq u \leq 3p$
.....	
$a+i$	$i \cdot p \leq u \leq (i+1) \cdot p$
.....	
β	$(\beta - a) \cdot p \leq u \leq (\beta - a + 1) \cdot p = 1$

4.6 Μέθοδος του αντίστροφου μετασχηματισμού για διακριτές κατανομές.

Η μέθοδος αυτή, όπως και η μέθοδος της αντίστροφης για συνεχείς κατανομές, χρησιμοποιεί την αθροιστική συνάρτηση πιθανότητας της κατανομής. Η μόνη απαίτηση είναι η συνάρτηση πυκνότητας πιθανότητας που τώρα εκφράζεται ως $P(X=x)$ να είναι γνωστή.

Στις διακριτές κατανομές ισχύουν

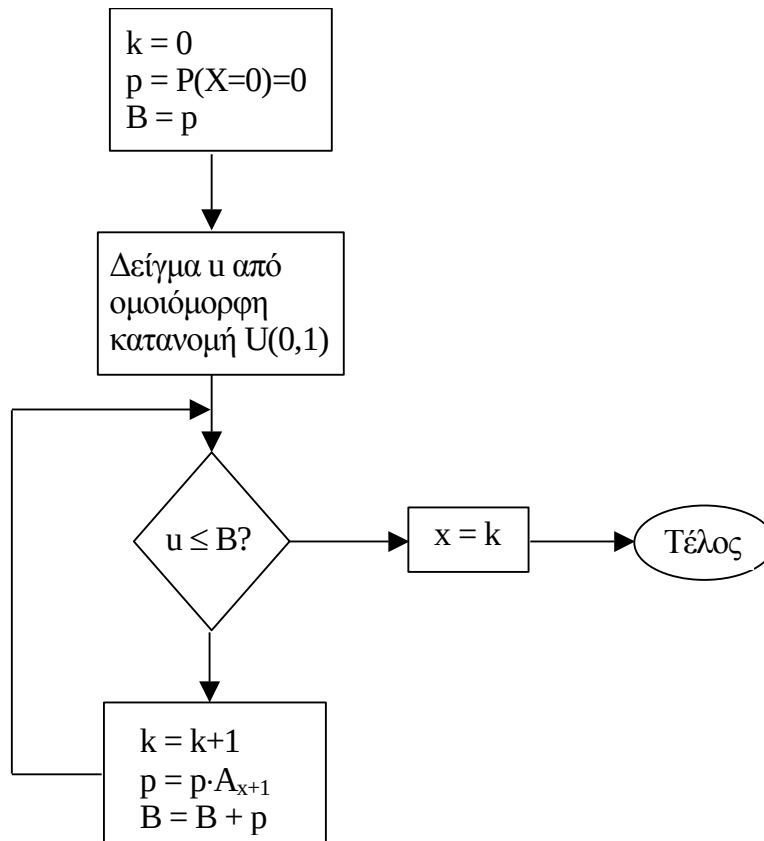
$$P(X \leq x+1) = P(X \leq x) + P(X = x+1) \text{ και}$$

$$P(X = x+1) = A_{x+1} \cdot P(X = x) \text{ όπου } A_{x+1} \text{ εξαρτάται από την κατανομή.}$$

Δηλαδή $P(X \leq x+1) = P(X \leq x) + A_{x+1} \cdot P(X = x)$.

Στη μέθοδο αυτή γεννιέται ένα τυχαίο δείγμα u από μια ομοιόμορφη κατανομή $U(0, 1)$ και στη συνέχεια αυτό το δείγμα συγκρίνεται με διαδοχικές τιμές της αθροιστικής συνάρτησης πιθανότητας $P(X \leq k)$, $k = 0, 1, \dots$. Όταν $u \leq P(X \leq k)$ τότε το τυχαίο δείγμα από τη διακριτή κατανομή είναι το $x = k$.

Έτσι με κατάλληλη επιλογή του A_{x+1} για τη συγκεκριμένη διακριτή κατανομή από την οποία θέλουμε να γίνει η δειγματοληψία, ο αλγόριθμος της μεθόδου θα μπορούσε να είναι ο εξής:



4.6.1 Δειγματοληψία από γεωμετρική κατανομή

Για την γεωμετρική κατανομή ισχύει

$$f(x) = P(X = x) = p(1-p)^x, \quad x = 0, 1, 2, \dots$$

όπου p είναι η πιθανότητα κάποιου γεγονότος να συμβεί και x είναι ο αριθμός των ανεξάρτητων δοκιμών πριν να συμβεί αυτό το γεγονός.

Αρχικά θα υπολογίσουμε το A_{x+1} για την γεωμετρική κατανομή:

$$P(X = x+1) = A_{x+1} \cdot P(X = x) \Rightarrow$$

$$p(1-p)^{x+1} = A_{x+1} \cdot p(1-p)^x \Rightarrow$$

$$A_{x+1} = (1-p).$$

Στη συνέχεια, αφού υπολογίστηκε το A_{x+1} για τη γεωμετρική κατανομή, μπορεί να χρησιμοποιηθεί ο παραπάνω αλγόριθμος για δειγματοληψία από γεωμετρική κατανομή.

4.6.2 Δειγματοληψία από Poisson κατανομή

Για την Poisson κατανομή ισχύει

$$f(x) = P(X = x) = \frac{e^{-\lambda} \lambda^x}{x!}, \quad x = 0, 1, 2, \dots$$

όπου λ είναι ο μέσος αριθμός “επιτυχιών” στη μονάδα του χρόνου και x είναι ο αριθμός των “επιτυχιών” στη συγκεκριμένη μονάδα του χρόνου.

Αρχικά θα υπολογίσουμε το A_{x+1} για την Poisson κατανομή:

$$P(X = x+1) = A_{x+1} \cdot P(X = x) \Rightarrow$$

$$\frac{e^{-\lambda} \lambda^{x+1}}{(x+1)!} = A_{x+1} \cdot \frac{e^{-\lambda} \lambda^x}{x!} \Rightarrow$$

$$A_{x+1} = \frac{\lambda}{x+1}.$$

Στη συνέχεια, αφού υπολογίστηκε το A_{x+1} για τη Poisson κατανομή, μπορεί να χρησιμοποιηθεί ο παραπάνω αλγόριθμος για δειγματοληψία από Poisson κατανομή.

4.7 Δειγματοληψία από δυωνυμική κατανομή $B(n, p)$

Για την δυωνυμική κατανομή, η συνάρτηση πυκνότητας πιθανότητας, είναι:

$$f(x) = \binom{n}{x} p^x (1-p)^{n-x}, \quad x = 0, 1, 2, \dots$$

όπου n είναι ο αριθμός των ανεξάρτητων δοκιμών Bernoulli και x είναι ο συνολικός αριθμός των επιτυχιών.

Υποθέτουμε ότι έχουμε n ανεξάρτητες δοκιμών Bernoulli για τις οποίες η πιθανότητα επιτυχίας είναι p και η πιθανότητα αποτυχίας $(1-p)$. Εάν x είναι ο συνολικός αριθμός των επιτυχιών, τότε το x είναι δείγμα από δυωνυμική κατανομή $B(n, p)$.

Μια δοκιμή Bernoulli ακολουθεί τα εξής βήματα:

1. Γέννηση τυχαίου δείγματος από ομοιόμορφη κατανομή $U(0, 1)$.
2. Εάν $u \leq p$ τότε έχουμε επιτυχία, ενώ εάν $u > p$ τότε έχουμε αποτυχία.

Έτσι η δειγματοληψία από δυωνυμική κατανομή $B(n, p)$ ακολουθεί την εξής διαδικασία:

Γίνεται ένας αριθμός δοκιμών Bernoulli, σύμφωνα με τη παραπάνω διαδικασία, και στη συνέχεια μετريέται ο αριθμός των επιτυχιών (x). Αυτός ο αριθμός είναι ένα δείγμα από μια δυωνυμική κατανομή $B(n, p)$.

Δειγματοληψία από δυωνυμική κατανομή $B(n,p)$ με προσέγγιση κανονικής κατανομής

Για $n \rightarrow \infty$ μια δυωνυμική κατανομή $B(n, p)$ προσεγγίζει μία κανονική κατανομή $N(np, np(1-p))$.

Η παραπάνω προσέγγιση είναι επιτυχής για:

1. $np > 5$ όταν $p \leq \frac{1}{2}$
2. $n(1-p) > 5$ όταν $p \geq \frac{1}{2}$.

Εφόσον ισχύουν οι παραπάνω συνθήκες, η διαδικασία δειγματοληψίας από δυωνυμική με προσέγγιση κανονικής κατανομής ακολουθεί τα εξής βήματα:

1. Παίρνουμε ένα δείγμα z από την κανονική κατανομή $N(0,1)$.
2. Θέτουμε $x = np + z\sqrt{np(1-p)}$.

Επειδή το x είναι δείγμα διακριτής κατανομής, στρογγυλοποιείται στο πλησιέστερο ακέραιο.

4.8 Δειγματοληψία από εμπειρική κατανομή $E(n, x_i, p_i)$

Σε μια εμπειρική κατανομή $E(n, x_i, p_i)$ η συνάρτηση πυκνότητας πιθανότητας είναι της μορφής:

$$f(x_i) = P(X = x_i) = p_i, \quad i = 1, 2, \dots, n.$$

Η μέθοδος δειγματοληψίας από μια εμπειρική κατανομή ακολουθεί τα παρακάτω βήματα:

1. Ταξινομούμε τα x_i έτσι ώστε $x_1 < x_2 < \dots < x_n$.
2. Υπολογίζουμε τη πιθανότητα $p_i = P(X \leq x_i) = \sum_{j=1}^i p_j$, αφού θέσουμε $p_0 = 0$.
3. Γεννάμε ένα τυχαίο δείγμα u από την ομοιόμορφη κατανομή $U(0,1)$.
4. Θέτουμε $x = x_i$ εάν $p_{i-1} \leq u \leq p_i$.

Το x είναι ένα τυχαίο δείγμα από μια εμπειρική κατανομή $E(n, x_i, p_i)$.

ΚΕΦΑΛΑΙΟ 5ο

Η ΓΛΩΣΣΑ SIMSCRIPT II.5

5.1 Η γλώσσα SImscript II.5 και το περιβάλλον εργασίας Simlab

5.1.1 Ιστορική αναδρομή

Η γλώσσα προσομοίωσης SImscript II.5 βρίσκει τις ρίζες της στην εταιρία Rank Corporation όπου ο Harry Markowitz δημιούργησε ένα περιβάλλον στο οποίο κωδικοποιημένες φόρμες συμπληρώνονταν για την περιγραφή και μοντελοποίηση των αντικειμένων, για την αρχικοποίηση των μεταβλητών και για τη δημιουργία reports και μετά ένα κωδικοποιημένο σε Fortran πρόγραμμα αναλάμβανε την υλοποίηση της προσομοίωσης.

Στην συνέχεια στην εταιρία CACI ο ίδιος ο Markowitz δημιούργησε μια βελτιωμένη έκδοση αυτής της γλώσσας γνωστή ως SImscript I.5, ελαχιστοποιώντας πλέον τις εξαρτήσεις από την γλώσσα Fortran.

Για τους υπολογιστές των εταιριών IBM, CDC, UNIVAC, PHILCO, RCA και NCR κατασκευάστηκαν αντίστοιχες εκδόσεις της SImscript I.5 από παράλληλους κατασκευαστές όπως οι Henry Kleine, Glen Johnson και Claude Delfosse.

Τέλος, ενώ ο Markowitz βελτίωσε τη γλώσσα φτιάχνοντας την έκδοση SImscript II, στην εταιρία CACI φτιάχτηκε η τελική έκδοση SImscript II.5 για υπολογιστές εταιριών όπως IBM, CDC, Honeywell, Univac, Digital Equipment, NCR και PRIME.

Η SImscript II.5 είναι και η σημερινή έκδοση της γλώσσας που είναι ευρέως διαδεδομένη σαν πρότυπο γλώσσας και γενικότερα περιβάλλοντος προσομοίωσης και είναι η γλώσσα που θα παρουσιαστεί στην συνέχεια.

5.1.2 SImscript II.5 – Προσομοίωση

Η γλώσσα SImscript II.5 ασχολείται με την προσομοίωση διακριτών γεγονότων (discrete event simulation) και έχει τη δυνατότητα υλοποίησης μοντέλων προσομοίωσης τόσο με μέθοδο των γεγονότων όσο και με την μέθοδο των διαδικασιών.

Στη συνέχεια θα ασχοληθούμε κύρια με την μέθοδο των διαδικασιών δίνοντας έμφαση στο πως αυτή υλοποιείται στην SImscript II.5.

5.1.3 Το περιβάλλον SimLab

Το περιβάλλον SimLab είναι ένα πλήρες και ολοκληρωμένο περιβάλλον εργασίας ειδικά κατασκευασμένο για την ανάπτυξη και υλοποίηση μοντέλων προσομοίωσης στη γλώσσα SImscript II.5 περιλαμβάνοντας ένα σύνολο από σχετικά εργαλεία:

- **SimEdit** Αυτός είναι ένας editor κειμένου με τη βοήθεια του οποίου εισάγονται τα κείμενα των προγραμμάτων.
- **Compiler** Αυτός είναι ο compiler της Simsript II.5 που ετοιμάζει τα διάφορα τμήματα (modules) των προγραμμάτων στην πορεία τους να γίνουν εκτελέσιμο πρόγραμμα.
- **Linker** Είναι το πρόγραμμα (κομμάτι του SimLab) που, παίρνοντας τα αποτελέσματα του compiler, δημιουργεί πλήρες πρόγραμμα προς εκτέλεση.
- **SimDebug** Είναι το εργαλείο με τη βοήθεια του οποίου αυτός που αναπτύσσει εφαρμογές, βρίσκει λάθη που παρατηρούνται κατά τη διάρκεια της εκτέλεσης του προγράμματος (run time errors).
- **Commands** Είναι ένα σύνολο εντολών του SimLab περιβάλλοντος που χρησιμοποιούνται για τη διαχείριση του περιβάλλοντος.

5.1.4 Ο editor κειμένου SimEdit

Ο editor αυτός είναι ένας συνηθισμένος editor που έχει όλα τα βασικά χαρακτηριστικά ενός επεξεργαστή κειμένου. Χρησιμοποιώντας αυτόν τον editor πληκτρολογούμε το πρόγραμμα προσομοίωσης που επιθυμούμε ή τα διάφορα τμήματά (modules) του, υλοποιώντας έτσι ένα μοντέλο προσομοίωσης.

Οι εντολές που χρησιμοποιεί είναι παρόμοιες με αυτές που χρησιμοποιεί ο επεξεργαστής κειμένου WordStar της εταιρίας MicroPro International Corporation. Υπάρχει φυσικά η δυνατότητα να χρησιμοποιηθεί οποιοσδήποτε άλλος editor, αλλά θα πρέπει να αλλάζουμε περιβάλλον από SimLab σε MS-DOS, από όπου θα πρέπει να εκτελούμε το πρόγραμμα του editor που θέλουμε και επίσης το κείμενο που πληκτρολογούμε θα πρέπει να το μετακινούμε από και προς το περιβάλλον SimLab με τη χρήση των εντολών Export και Import.

5.1.4.1 Οι κλασσικές λειτουργίες του editor

- ✓ **Οι κινήσεις του δείκτη θέσης (cursor).** Ο δείκτης θέσης (cursor) μπορεί να μετακινηθεί μέσα στο παράθυρο κατά ένα χαρακτήρα, κατά μία λέξη ή κατά μία σελίδα (16 γραμμές).
- ✓ **Διαγραφή κειμένου** γίνεται κατά χαρακτήρα, κατά λέξη, κατά γραμμή ή κατά κομμάτι (block). Σαν κομμάτι (block) ορίζουμε ένα πλήθος γραμμών κειμένου που καθορίζεται κάθε φορά από τον χρήστη (user).
- ✓ **Πληκτρολόγηση κειμένου** γίνεται σε μια από τις δύο καταστάσεις (modes), την κατάσταση παρεμβολής (insert mode) και την κατάσταση αντικατάστασης (overtyping mode). Στην κατάσταση παρεμβολής (insert mode) εισάγουμε χαρακτήρες χωρίς να σβήνουμε άλλους που ήδη

υπάρχουν αλλά ωθώντας τους προς τα δεξιά. Στην κατάσταση αντικατάστασης (overtyping mode) εισάγουμε τους νέους χαρακτήρες ακριβώς πάνω στους ήδη υπάρχοντες, διαγράφοντας έτσι τους τελευταίους. Ξεχωρίζουμε κάθε στιγμή την κατάσταση (mode) στην οποία βρισκόμαστε από το σχήμα του δείκτη θέσης (cursor).

- ✓ **Αναζήτηση / αντικατάσταση.** Υπάρχει η δυνατότητα για αναζήτηση συγκεκριμένου κειμένου μέσα σε ένα αρχείο που το επεξεργαζόμαστε στον editor και επίσης υπάρχει η δυνατότητα αντικατάστασης αυτού του κομματιού με ένα άλλο.
- ✓ **Διαχείριση κομματιών κειμένου (blocks)** γίνεται σημειώνοντας αυτά τα κομμάτια (mark blocks) και μετακινώντας τα, αντιγράφοντας τα, σβήνοντας τα ή και γράφοντας τα σε άλλα αρχεία. Όταν ένα κομμάτι κειμένου σημειωθεί (marked block) τότε ο editor το κάνει να φαίνεται έντονα φωτισμένο.
- ✓ **Αυτόματη στοίχιση.** Γράφοντας ένα πρόγραμμα στον SimEdit αυτός φροντίζει για την κατάλληλη στοίχιση των γραμμών στο αριστερό περιθώριο.
- ✓ **Εμφάνιση των σφαλμάτων.** Τα λάθη (errors) και οι προειδοποιήσεις (warnings) που δημιουργούνται κατά τη διάρκεια του compilation εμφανίζονται σαν προσωρινές γραμμές (phantom lines) αμέσως μετά τη γραμμή σφάλματος, δίνοντας έτσι τη δυνατότητα άμεσης διόρθωσης.
- ✓ **Αυτόματη αποθήκευση.** Ο SimEdit αυτόματα αποθηκεύει (save) το κείμενο / πρόγραμμα που πληκτρολογείται κάθε 15 λεπτά, τιμή που μπορεί να μεταβληθεί.

5.1.4.2 Δουλεύοντας με τον SimEdit

Με τον SimEdit μπορούμε να επεξεργαστούμε αρχεία που αντιστοιχούν σε κομμάτια (modules) προγραμμάτων που ήδη έχουμε φτιάξει, αρχεία δεδομένων, ή το αρχείο εργασίας (workfile).

Το αρχείο εργασίας (workfile) είναι εκείνο το προσωρινό αρχείο που χρησιμοποιείται για πρόσθεση νέου κώδικα στον ήδη υπάρχοντα, με τη μορφή νέου τμήματος (module). Κατά τη διάρκεια του compilation ο compiler τεμαχίζει το αρχείο εργασίας (workfile) σε ένα ή περισσότερα τμήματα (modules) και δημιουργεί αρχεία με αντίστοιχα ονόματα.

Όταν επεμβαίνουμε με τον SimEdit σε ένα τμήμα (module) τότε σημειώνεται εσωτερικά αυτή η επέμβασή μας ώστε να γνωρίζει το λειτουργικό περιβάλλον του SimLab τις τυχόν αλλαγές και να περιμένει αντίστοιχα compilation.

Ο SimEdit καλείται, ανάλογα με την περίπτωση, με την εντολή EDIT ως εξής:

EDIT “όνομα τμήματος (module)”	για την περίπτωση που θέλουμε να επεμβούμε σε ένα ήδη έτοιμο τμήμα (module).
EDIT/NEW	για την περίπτωση που θέλουμε να δημιουργήσουμε κάποιο νέο τμήμα του προγράμματος πληκτρολογώντας το στο αρχείο εργασίας (workfile).
EDIT/WORK	για να διορθώσουμε ένα αρχείο εργασίας (workfile) που ήδη υπάρχει.
EDIT/DATA “όνομα αρχείου”	για να εισάγουμε ή να τροποποιήσουμε αριθμητικά ή άλλα δεδομένα σε ένα αρχείο δεδομένων.

Από τη στιγμή που έχουμε καλέσει τον SimEdit είτε έχει δημιουργηθεί ένα νέο αρχείο (workfile ή data file) είτε έχει έρθει στην οθόνη για επεξεργασίας ένα ήδη δημιουργημένο αρχείο. Στην πρώτη περίπτωση έχουμε κενή οθόνη έτοιμη να δεχτεί δεδομένα ενώ στη δεύτερη έχουμε στην οθόνη το έτοιμο αρχείο. Και στις δύο περιπτώσεις μετά την κλήση του editor βρισκόμαστε στην κατάσταση επεξεργασίας (Video mode).

Στο παράρτημα 1 παρουσιάζονται οι διάφορες δυνατότητες που μας παρέχονται όταν βρισκόμαστε στην κατάσταση αυτή (Video mode).

Ενώ βρισκόμαστε στην κατάσταση επεξεργασίας του κειμένου, πιέζοντας το πλήκτρο Esc ερχόμαστε στην κατάσταση εντολών ανοίγοντας ένα παράθυρο στην οθόνη όπου αναμένεται η πληκτρολόγηση κάποιας από τις εντολές του SimEdit που παρουσιάζονται και αυτές στο παράρτημα 1.

5.1.5 Ο Compiler

Ο Compiler του περιβάλλοντος SimLab είναι αυτός που αναλαμβάνει να κάνει τη μετάφραση του κώδικα, που πληκτρολογούμε για κάθε τμήμα (module) του προγράμματος, σε μία γλώσσα κατάλληλη για τον επεξεργαστή του κάθε H/Y.

Τεχνικά, η μετάφραση γίνεται σε δύο στάδια (passes). Στο πρώτο έχουμε μετάφραση του πηγαίου κώδικα (source) σε μια ενδιάμεση μορφή γλώσσας συνοψίζοντας τη δομή και την οργάνωση του προγράμματος που έχουμε εισάγει. Σε αυτό το σημείο ο μεταφραστής (compiler) χωρίζει το αρχείο εργασίας στο οποίο πληκτρολογήσαμε τον πηγαίο κώδικα σε άλλα μικρότερα αρχεία, ένα για κάθε διαδικασία, ένα για κάθε ρουτίνα και ένα για τα PREAMBLE και MAIN. Τα αρχεία αυτά παίρνουν το όνομα της διαδικασίας ή της ρουτίνας αντίστοιχα και με αυτά τα ονόματα τα καλούμε στο εξής. Τα φυσικά ονόματα των αρχείων αυτών, τα ονόματα δηλαδή με τα οποία το MsDos τα αναγνωρίζει

είναι διαφορετικά αλλά δεν χρειάζεται να τα γνωρίζουμε όσο βρισκόμαστε στο περιβάλλον του SimLab.

Στο δεύτερο στάδιο ο μεταφραστής μεταφράζει την ενδιάμεση φόρμα (αποτέλεσμα του πρώτου σταδίου) σε ένα σύνολο υπορουτίνων, ειδικές για ένα νοητό υπολογιστή με συμβολική ονομασία S-machine. Τέλος υπάρχει ειδικό κομμάτι του SimLab που αναλαμβάνει να αντιστοιχεί τη μορφή αυτή της μετάφρασης στο συγκεκριμένο τύπο επεξεργαστή.

Το compilation ξεκινά με την εντολή COMPILE και εκτελείται στο background, δηλαδή ενώ εκτελείται αφήνει ελεύθερο το περιβάλλον για να εκτελέσουμε και κάποιες άλλες εργασίες ταυτόχρονα. Η δυνατότητα αυτή ονομάζεται multitasking.

5.1.6 Ο Linker

Μετά και το δεύτερο πέραςμα (δεύτερο στάδιο) από τον μεταφραστή (compiler) το πρόγραμμα δεν είναι έτοιμο ακόμα να εκτελεστεί. Χρειάζεται να γίνει η κατάλληλη ένωση των διαφόρων μεταφρασμένων τμημάτων (modules) σε ένα ενιαίο εκτελέσιμο πρόγραμμα. Την εργασία αυτή αναλαμβάνει ο linker του περιβάλλοντος SimLab. Όπως φαίνεται και στο παράρτημα 2 των εντολών του SimLab ο linker καλείται με την εντολή LINK. Επίσης ο linker καλείται αυτόματα μόλις ζητήσουμε την εκτέλεση ενός έτοιμου προγράμματος που το μόνο που του απομένει ακόμα είναι το linking.

Ο linker στην ουσία φτιάχνει δύο αρχεία, ένα αρχείο προς φόρτωση (load file) και ένα αρχείο δεικτών που δείχνουν στο πρώτο αρχείο (index file). Με αυτό τον τρόπο είναι δυνατόν να εκτελεστεί πρόγραμμα που χρειάζεται μνήμη πολύ μεγαλύτερη από την υπάρχουσα φυσική μνήμη. Στη μνήμη κάθε στιγμή φορτώνονται από το πρώτο αρχείο μόνο τα τμήματα που είναι απαραίτητα και το υπόλοιπο μένει στην τοπική περιοχή αποθήκευσης (swap area). Τον έλεγχο του που ακριβώς βρίσκεται το κατάλληλο κομμάτι στο load file τον αναλαμβάνει το δεύτερο αρχείο το index file.

5.1.7 Ο Debugger σφαλμάτων SimDebug

Ο SimDebug είναι ένα εργαλείο με το οποίο γίνεται ανίχνευση και διόρθωση σφαλμάτων στα προγράμματα κατά τη διάρκεια της εκτέλεσής της (run time errors). Οι κατηγορίες των σφαλμάτων αυτών είναι:

- **Σφάλματα της γλώσσας Simscript II.5.** Τέτοια σφάλματα είναι αυτά που παραβιάζουν τους κανόνες, συντακτικούς και νοηματικούς, τη γλώσσας Simscript II.5. Για παράδειγμα αν σε κάποιο σημείο ζητείται η τετραγωνική ρίζα ενός αρνητικού αριθμού τότε στο σημείο αυτό θα έχουμε ένα σφάλμα εκτέλεσης (run time error).

- **Σφάλματα εισαγωγής δεδομένων.** Το SimLab έχει τη δυνατότητα να ανιχνεύει εισαγωγές δεδομένων που δεν είναι του τύπου που αναμένεται από την αντίστοιχη εντολή του προγράμματος. Για παράδειγμα, αν προσπαθήσουμε να εισάγουμε ένα γράμμα εκεί που αναμένεται να πληκτρολογήσουμε έναν αριθμό θα παρατηρήσουμε ένα λάθος αυτού του τύπου.
- **Σφάλματα εισόδου / εξόδου.** Εδώ έχουμε τα διάφορα λάθη που συμβαίνουν κατά τη διάρκεια διαχείρισης αρχείων, δηλαδή δημιουργία, άνοιγμα, ανάγνωση, εγγραφή, κλείσιμο, διαγραφή και εκτύπωση αρχείου. Για παράδειγμα έχουμε τέτοιο λάθος όταν προσπαθούμε να ανοίξουμε ένα αρχείο που δεν υπάρχει.
- **Μηχανικά σφάλματα.** Τέτοια είναι τα σφάλματα που οφείλονται σε λάθη του υλικού (hardware). Ο SimDebug αναφέρει ακριβώς το σφάλμα που παρατηρήθηκε.

5.1.8 Οι εντολές του περιβάλλοντος SimLab

Στο παράρτημα 2 παρουσιάζονται όλες οι εντολές του περιβάλλοντος SimLab με την αντίστοιχη επεξήγησή τους.

5.2 Γράφοντας ένα μοντέλο προσομοίωσης στο περιβάλλον εργασίας SimLab

Το πρώτο βήμα για τη δημιουργία ενός μοντέλου προσομοίωσης είναι η σε βάθος κατανόηση του προβλήματος και η σωστή και ολοκληρωμένη σχεδίαση της λύσης του. Υποθέτοντας σε αυτό το σημείο ότι υπάρχει η βαθύτερη κατανόηση της λειτουργίας των μοντέλων προσομοίωσης, ασχολούμαστε με την τεχνική πλευρά της υλοποίησης του μοντέλου, δηλαδή με το πως εισάγουμε τον κώδικα στον Η/Υ, πως δημιουργούμε ένα εκτελέσιμο πρόγραμμα και τέλος πως εκτελούμε αυτό το πρόγραμμα ώστε να πάρουμε τα επιθυμητά αποτελέσματα.

Η πρώτη ενέργεια που έχουμε να κάνουμε είναι να δημιουργήσουμε ένα κομμάτι στην περιφερειακή μνήμη του υπολογιστή μας (συνήθως στο σκληρό δίσκο) όπου θα υπάρχουν όλα τα κομμάτια (modules) τα σχετικά με το συγκεκριμένο μοντέλο. Η εντολή την οποία χρησιμοποιούμε γι αυτό είναι:

SELECT/NEW “όνομα μοντέλου”

Εκτελώντας αυτή την εντολή δημιουργείται ένα χώρος στο δίσκο (subdirectory) με όνομα το συγκεκριμένο όνομα του μοντέλου. Κάθε φορά που ξεκινάμε το πρόγραμμα και θέλουμε να εκτελέσουμε ή να επεξεργαστούμε αυτό το μοντέλο πληκτρολογούμε την ίδια εντολή ως εξής:

SELECT “όνομα μοντέλου”

Παρατηρούμε ότι μη πληκτρολογώντας το “/NEW” το πρόγραμμα αναζητά ήδη υπαρχών μοντέλο με αυτό το όνομα.

Έχοντας δημιουργήσει το χώρο που θα φτιάξουμε το μοντέλο μας και έχοντας επιλέξει αυτόν τον χώρο, πρέπει τώρα να δημιουργήσουμε τα διάφορα κομμάτια (modules) του προγράμματός μας όπως το PREAMBLE, το MAIN και τις διάφορες ρουτίνες και διαδικασίες του μοντέλου. Με τη βοήθεια του editor (SimEdit) θα πληκτρολογήσουμε τα modules αυτά στο αρχείο εργασίας (workfile) και κατόπιν με το compilation θα δημιουργηθούν αντίστοιχα αρχεία. Η εντολή με την οποία ξεκινάμε την πληκτρολόγηση είναι:

EDIT/NEW

Ας δούμε για παράδειγμα πως φτιάχνουμε ένα συγκεκριμένο μοντέλο, έστω το μοντέλο ενός κουρείου.

Πληκτρολογούμε:

SELECT/NEW BARBER

με αυτήν την εντολή δημιουργήσαμε και επιλέξαμε ένα νέο subdirectory στο οποίο θα περιέχονται όλα τα τμήματα του μοντέλου του κουρείου. Αν έχουμε ήδη δημιουργήσει αυτό το μοντέλο, τότε πληκτρολογούμε:

SELECT BARBER

Ξεκινάμε κατόπιν να πληκτρολογούμε το πρόγραμμά μας, με τη βοήθεια του SimEdit, στο αρχείο εργασίας (workfile) δίνοντας την εντολή:

EDIT/NEW

Το κείμενο / πρόγραμμα που πληκτρολογούμε φαίνεται αμέσως πιο κάτω:

```
1 Preamble
2
3   processes include arrival.generator, close.doors, and customer
4
5   resources include barber
6
7   define day.length, delay.in.queue, max.service.time,
8       mean.interarrival.time, and min.service.time as real variables
9
10  define num.barbers as an integer variable
11
12  define .minutes to mean units
13
14  tally max.delay.in.queue as the maximum, mean.delay.in.queue as
15      the mean, and num.served as the number of delay.in.queue
16
17  accumulate avg.number.in.queue as the average and
18      max.number.in.queue as the maximum of n.q.barber
19
20  accumulate util.barber as the average of n.x.barber
21
22 End "preamble
23
24 Main
25
26   call read.data
27   call initialize
28
29   start simulation
30
31   call report
32
33 End "Main
```

Οι αριθμοί γραμμών δεν πληκτρολογούνται. Εδώ βρίσκονται μόνο για να μπορούμε να αναφερόμαστε στη συγκεκριμένη γραμμή.

Παρατηρούμε ότι πληκτρολογήσαμε ταυτόχρονα στο αρχείο εργασίας (workfile) δύο από τα modules του μοντέλου μας, το PREAMBLE και το MAIN. Μπορούμε βέβαια να τα πληκτρολογήσουμε το καθένα ξεχωριστά με τη σειρά, αλλά έτσι και αλλιώς ο compiler στη συνέχεια θα τα ξεχωρίσει δημιουργώντας δύο νέα αρχεία με αντίστοιχα ονόματα PREAMBLE και MAIN ελευθερώνοντας ταυτόχρονα το Workfile.

Σημειώνουμε ότι για κάθε μοντέλο, υποχρεωτικά πρέπει να δημιουργήσουμε πρώτο το PREAMBLE που είναι το τμήμα των δηλώσεων και μετά το αρχείο MAIN που περιέχει τη βασική ροή του προγράμματος.

Στη συνέχεια αποθηκεύουμε το κείμενο που πληκτρολογήσαμε και βγαίνουμε από τον SimEdit με τις εξής εντολές:

Πατάμε το πλήκτρο Esc

με αυτόν τον τρόπο ερχόμαστε στην κατάσταση των εντολών και πληκτρολογούμε την εντολή:

EXIT

με την οποία αποθηκεύεται το workfile και βγαίνουμε από τον SimEdit. Τώρα είμαστε στο περιβάλλον των εντολών του SimLab όπου πρέπει να δώσουμε την εντολή για να ξεκινήσει το compilation πληκτρολογώντας:

COMPILE/ALL

Με την εντολή αυτή ξεκινά η μετάφραση του προγράμματος από κείμενο σε γλώσσα κατάλληλη για τον υπολογιστή. Όσο διαρκεί το compilation το περιβάλλον του SimLab είναι διαθέσιμο, γιατί η όλη εργασία γίνεται στο background. Μόλις τελειώσει το compilation εμφανίζεται σχετικό μήνυμα που συνοδεύεται από χαρακτηριστικό ήχο. Πληκτρολογώντας:

STATUS

βλέπουμε την κατάσταση των αρχείων του μοντέλου μας. Εδώ παρατηρούμε ότι έχουν δημιουργηθεί δύο αρχεία το PREAMBLE και το MAIN χαρακτηρισμένα OK (αν δεν έχει γίνει κανένα λάθος στην πληκτρολόγηση) και ότι το αρχείο εργασίας (workfile) έχει χαρακτηριστεί old. Τώρα πλέον είμαστε σε θέση να δημιουργήσουμε ένα νέο κομμάτι (module) του μοντέλου μας πληκτρολογώντας το στο workfile, ή να διορθώσουμε κάποιο από τα ήδη δημιουργημένα τμήματα (modules).

Εδώ επίσης παρατηρούμε ότι κατά τη διάρκεια του compilation δημιουργήθηκαν τα αρχεία READ.DATA, INITIALIZE, REPORT, CUSTOMER, CLOSE.DOORS και ARRIVAL.GENERATOR τα οποία είχαμε δηλώσει μέσα στο PREAMBLE ή τα είχαμε καλέσει στο MAIN. Τα αρχεία αυτά είναι χαρακτηρισμένα σαν not written. Δίνοντας τώρα για παράδειγμα την εντολή:

EDIT MAIN

ανοίγεται με τον SimEdit το αρχείο MAIN και είμαστε έτοιμοι να κάνουμε τυχόν διορθώσεις. Για τα αρχεία που είναι χαρακτηρισμένα σαν not written πρέπει να χρησιμοποιήσουμε το workfile δίνοντας τη γνωστή εντολή:

EDIT/WORK

όπου πληκτρολογούμε το αντίστοιχο κείμενο. Αφού αποθηκεύσουμε το αρχείο και κάνουμε compilation τότε τακτοποιείται το αντίστοιχο module.

Το ίδιο κάνουμε και για όλα τα άλλα τμήματα (modules) πληκτρολογώντας τα εξής:

```

1 Process arrival.generator
2
3   while time.v < day.length
4   do
5       wait exponential.f(mean.interarrival.time, 1) .minutes
6       activate a customer now
7   loop
8
9 End "arrival.generator

1 Process close.doors
2
3   interrupt arrival.generator
4
5 End "close.doors

1 Process customer
2
3   define time.of.arrival as a real variable
4
5   let time.of.arrival = time.v
6   request 1 barber(1)
7   let delay.in.queue = time.v - time.of.arrival
8   work uniform.f (min.service.time, max.service.time, 2) .minutes
9   relinquish 1 barber(1)
10
11 End "customer

1 Routine initialize
2
3   create every barber(1)
4   let u.barber(1) = num.barbers
5
6   activate an arrival.generator now
7   activate a close.doors in day.length .minutes
8
9 End "initialize

1 Routine read.data
2
3   read day.length, mean.interarrival.time, min.service.time,
4       max.service.time, and num.barbers
5
6   print 4 lines thus
7
8 BARBERSHOP MODEL - INPUT PARAMETERS
9 -----
10
11   skip 3 lines

```



```

12
13     print 9 lines with num.barbers, mean.interarrival.time,
14         min.service.time, max.service.time, and day.length thus
15
16 Number of barbers **
17
18 Interarrival times are exponential with mean ***.* minutes
19
20 Service times are uniform on the interval (***.*, ***.*) minutes
21
22 Doors close after ****.* minutes
23
24     start new page
25
26 End "read.data

```

```

1 Routine report
2
3     print 4 lines thus
4
5 SIMULATION RESULTS (ALL TIMES ARE IN MINUTES)
6 -----
7
8     skip 3 lines
9
10    print 3 lines with time.v thus
11
12 Simulation ended at ****.*
13
14    skip 3 lines
15
16    print 6 lines with num.served, mean.delay.in.queue, and
17        max.delay.in.queue thus
18
19 Total number of          Mean time          Max time
20 customers served        in queue          in queue
21 -----
22      *****          *****.*          *****.*
23
24 skip 3 lines
25
26 print 6 lines with util.barber(1)/num.barbers, avg.number.in.queue(1),
27     and max.number.in.queue(1) thus
28
29 Barber          Average number of          Maximum number of
30 utilization      customers in queue          customers in queue
31 -----
32    ***.*          *****.*          *****.*
33
34 End "report

```

Αυτή τη στιγμή έχουμε πληκτρολογήσει όλα τα τμήματα του μοντέλου μας και είμαστε έτοιμοι για το τελικό compilation που γίνεται δίνοντας πάλι την εντολή:

COMPILE/ALL

Όταν τελειώσει το compilation και αν δεν έχει γίνει κανένα λάθος πληκτρολόγησης, τότε πληκτρολογώντας την εντολή:

STATUS

Θα πάρουμε σαν αποτέλεσμα μια λίστα με όλα τα τμήματα (modules) του μοντέλου μας, χαρακτηρισμένα ok και έτσι είμαστε έτοιμοι να προχωρήσουμε στο linking του προγράμματος έτσι ώστε αυτό να γίνει ένα εκτελέσιμο πρόγραμμα. Κάνουμε link αυτόματα πληκτρολογώντας την εντολή:

RUN

που είναι και η εντολή με την οποία τρέχουμε το μοντέλο μας (αυτόματα ανιχνεύεται ότι το μοντέλο είναι έτοιμο και το μόνο που λείπει είναι το link που ξεκινά μόνο του).

Αφού δώσουμε τις αρχικές τιμές των μεταβλητών που ζητούνται παίρνουμε σαν αποτέλεσμα τα εξής:

BARBERSHOP MODEL - INPUT PARAMETERS

Number of barbers 1

Interarrival times are exponential with mean 30.0 minutes

Service times are uniform on the interval (15.0, 25.0) minutes

Doors close after 480.0 minutes

SIMULATION RESULTS (ALL TIMES ARE IN MINUTES)

Simulation ended at 508.35

Total number of
customers served

21

Mean time
in queue

30.08

Max time
in queue

72.58

Barber
utilization

.84

Average number of
customers in queue

1.24

Maximum number of
customers in queue

4.00

Στο πιο πάνω μοντέλο χρησιμοποιήθηκαν οι πιο κάτω μεταβλητές που δημιουργούνται από το σύστημα όταν ζητάμε να δημιουργηθεί μια θέση εξυπηρέτησης (server):

N.resource	Αριθμός τύπων resources μιας κατηγορίας
U.resource	Αριθμός μονάδων resources ανά resource
Q.resource	Σύνολο δραστηριοτήτων στην αναμονή μιας resource
N.Q.resource	Αριθμός δραστηριοτήτων στην αναμονή μιας resource
X.resource	Σύνολο δραστηριοτήτων που χρησιμοποιούν μια resource
N.X.resource	Αριθμός δραστηριοτήτων που χρησιμοποιούν μια resource

Επίσης χρησιμοποιήσαμε και τις παρακάτω μεταβλητές:

avg.number.in.queue	ο μέσος αριθμός πελατών στην αναμονή
day.length	ο συνολικός χρόνος με ανοιχτές τις πόρτες
delay.in.queue	η καθυστέρηση στην αναμονή ενός πελάτη
max.delay.in.queue	η μέγιστη καθυστέρηση πελατών στην αναμονή
mean.delay.in.queue	η μέση καθυστέρηση πελατών στην αναμονή
max.number.in.queue	ο μέγιστος αριθμός πελατών στην αναμονή
max.service.time	ο μέγιστος χρόνος εξυπηρέτησης
min.service.time	ο ελάχιστος χρόνος εξυπηρέτησης
mean.interarrival.time	ο μέσος χρόνος μεταξύ αφίξεων
num.barbers	ο αριθμός κουρέων
num.served	ο αριθμός πελατών που εξυπηρετήθηκαν
time.of.arrival	ο χρόνος άφιξης πελάτη στο κουρείο
util.barber	η καθαρή απασχόληση του κουρέα

Στη συνέχεια παρουσιάζονται αναλυτικά οι ρουτίνες και οι διαδικασίες του προγράμματος που μοντελοποιεί το σύστημα του κουρείου.

PREAMBLE

Το Preamble είναι το τμήμα (module) του προγράμματος που είναι απαραίτητο για κάθε μοντέλο προσομοίωσης. Είναι το τμήμα στο οποίο γίνονται οι δηλώσεις των μεταβλητών, των διαδικασιών, των σημείων εξυπηρέτησης (resources) και άλλων σταθερών που χρειάζονται για την εκτέλεση της προσομοίωσης του μοντέλου. Στο συγκεκριμένο πρόγραμμα ξεκινάμε με τον ορισμό τριών διαδικασιών που περιλαμβάνονται στο μοντέλο του κουρείου, όπως περιγράφεται στη γραμμή 3 του Preamble. Οι διαδικασίες αυτές είναι η διαδικασία γέννησης αφίξεων (arrival.generator), η διαδικασία εξυπηρέτησης πελατών (customer) και η διαδικασία παύσης των γεννήσεων αφίξεων (close.doors).

Με τη δήλωση της γραμμής 5, ορίζουμε το μοναδικό σημείο εξυπηρέτησης (resource) του μοντέλου μας που είναι ο κουρέας (barber).

Στη συνέχεια ορίζονται οι διάφορες γενικές (global) μεταβλητές, η χρήση των οποίων έχει ήδη περιγραφεί.

Στη γραμμή 12 ορίζουμε τα λεπτά (minutes) να είναι η μονάδα μέτρησης (unit) του χρόνου.

Τέλος στις γραμμές 14 ως 20 καθορίζονται οι τιμές που θα παίρνουν κάποιες μεταβλητές με τις εντολές TALLY και ACCUMULATE. Έτσι μπορούμε να έχουμε τη μέγιστη, την ελάχιστη, τη μέση τιμή κ.λ.π. μιας μεταβλητής που παίρνει διάφορες τιμές κατά τη διάρκεια της εκτέλεσης της προσομοίωσης.\

MAIN

Στο τμήμα (module) Main περιγράφεται γενικά η διαδικασία της εκτέλεσης της προσομοίωσης. Εδώ καλούνται οι διάφορες ρουτίνες (read.data, initialize και report) και υπάρχει ο βασικός έλεγχος ροής της προσομοίωσης. Εδώ έχουμε την πιο απλή μορφή της ρουτίνας Main όπου καλείται η ρουτίνα με την οποία διαβάζονται οι αρχικές τιμές για τις διάφορες μεταβλητές του συστήματος και μετά δημιουργούνται οι κατάλληλες αρχικές συνθήκες. Κατόπιν ξεκινά η προσομοίωση και αφού αυτή συμπληρωθεί, εκτυπώνεται η αναφορά (report) με τα αποτελέσματα.

READ.DATA

Εδώ διαβάζονται από την είσοδο του υπολογιστή (πληκτρολόγιο) οι τιμές για τις μεταβλητές day.length, mean.interarrival.time, min.service.time και num.barber. Αμέσως μετά τυπώνονται αυτές οι τιμές για να έχουμε άμεση αντίληψη για τις τιμές που έχουμε δώσει στις μεταβλητές του συστήματος. Παρατηρούμε τον τρόπο με τον οποίο δηλώνουμε το πως θέλουμε να εκτυπωθούν κάποιοι αριθμοί ανάμεσα σε κείμενο με συγκεκριμένη μορφή (format).

INITIALIZE

Η εντολή CREATE στη γραμμή 3 της συγκεκριμένης ρουτίνας είναι αυτή με την οποία δεσμεύουμε την κατάλληλη μνήμη για την εξυπηρέτηση της μόνιμης οντότητας του συστήματός μας, τον κουρέα (barber). Εδώ παίρνει αυτόματα την τιμή της και η μεταβλητή του συστήματος N.BARBER που όπως έχει αναφερθεί, δείχνει πόσους τύπους κουρέων έχουμε, στην περίπτωσή μας έναν. Στην επόμενη εντολή δίνεται τιμή στην άλλη μεταβλητή του συστήματος που έχει σχέση με το σημείο εξυπηρέτησης, την U.BARBER(1). Αυτή είναι η μεταβλητή που δείχνει τον αριθμό των μονάδων που διαθέτουμε για το συγκεκριμένο τύπο κουρέα που στη περίπτωσή μας είναι πάλι 1. Για παράδειγμα αν θέλαμε να παραστήσουμε ένα σύστημα με δύο τύπους κουρέων (αυτούς που κάνουν καλλιτεχνικό κούρεμα και αυτούς που κάνουν απλό

κούρεμα) και από τον πρώτο τύπο έχουμε 3 κουρείς ενώ από τον δεύτερο 4, τότε οι αντίστοιχες μεταβλητές θα έχουν τις εξής τιμές:

$$N.BARBER = 2 \text{ (δύο τύπων κουρείς)}$$

$$U.BARBER(1) = 3$$

$$U.BARBER(2) = 4$$

Με την εντολή ACTIVATE στις γραμμές 6 και 7 προσθέτουμε στη λίστα σημειώσεων διαδικασιών σημειώσεις για τις δύο διαδικασίες της γέννησης αφίξεων και της παύσης γεννήσεων αφίξεων. Η πρώτη σημείωση παίρνει σαν χρόνο ενεργοποίησης τον τρέχων χρόνο προσομοίωσης, έτσι ώστε να είναι η πρώτη σημείωση που θα εξεταστεί, ενώ η δεύτερη χαρακτηρίζεται με χρόνο ενεργοποίησης το χρόνο που έχουμε δώσει σαν χρόνο διάρκειας της εργάσιμης μέρας. Η μεταβλητή του συστήματος που κάθε στιγμή περιέχει τον τρέχων χρόνο προσομοίωσης είναι η TIME.V την οποία μπορούμε να χρησιμοποιήσουμε σαν κανονική μεταβλητή.

ARRIVAL.GENERATOR

Το τμήμα (module) αυτό του προγράμματος είναι εκείνο που γεννά τις αφίξεις των πελατών χρησιμοποιώντας τη συνάρτηση $\text{exponential.f}(\alpha, \beta)$ για δειγματοληψία από μια εκθετική κατανομή για τους ενδιάμεσους χρόνους αφίξεων. Έχοντας τον χρόνο που προκύπτει από τη δειγματοληψία, ενεργοποιεί (δημιουργεί σημείωση για) μια διαδικασία εξυπηρέτησης πελάτη με χρόνο ενεργοποίησης αυτόν το χρόνο. Η διαδικασία γέννησης αφίξεων συνεχίζεται μέχρι να ικανοποιηθεί η συνθήκη τερματισμού, μέχρι δηλαδή να εξαντληθεί η εργάσιμη μέρα (γραμμή 3). Η εντολή WHILE ... DO ... LOOP είναι αυτή με την οποία πετυχαίνεται αυτή η επανάληψη.

CUSTOMER

Στη τρίτη γραμμή αυτού του τμήματος (module) του προγράμματος ορίζεται μια τοπική (local) μεταβλητή, η TIME.OF.ARRIVAL. Είναι η μεταβλητή στην οποία θα φυλάξουμε τον χρόνο προσομοίωσης στην οποία έγινε η άφιξη του πελάτη στο σύστημα του κουρείου, έτσι ώστε αφαιρώντας την τιμή αυτή από τον τρέχων χρόνο προσομοίωσης στον οποίο ο πελάτης αυτός αρχίζει να εξυπηρετείται, θα έχουμε σαν αποτέλεσμα την ώρα καθυστέρησής του στην ουρά (γραμμή 5 και 7). Στη γραμμή 6 έχουμε αίτηση προς το κουρέα (το σημείο εξυπηρέτησης) για απασχόληση μια μονάδας (από τον τύπο 1 του κουρέα, που εδώ είναι και ο μόνος που υπάρχει). Εδώ το πακέτο εργασίας (SimLab) αναλαμβάνει να διαχειριστεί μόνο του την διαδικασία ελέγχου του σημείου εξυπηρέτησης, δηλαδή να ελέγξει αν είναι ο κουρέας απασχολημένος ή ελεύθερος. Στην πρώτη περίπτωση θα βάλει τον πελάτη στην ουρά ενώ στη δεύτερη θα αφαιρέσει 1 από την μεταβλητή U.BARBER(1) κάνοντάς την 0 από 1, πράγμα που σημαίνει ότι δεν έχουμε άλλη μονάδα κουρέα να εξυπηρετήσει άλλη αίτηση πελάτη (ή με απλά λόγια ο κουρέας είναι απασχολημένος). Στην επόμενη εντολή (γραμμή 8) ο κουρέας απασχολείται με το συγκεκριμένο πελάτη κατά τόσα λεπτά όσα θα μας επιστρέψει ως τιμή η συνάρτηση που καλείται: $\text{uniform}(\alpha, \beta, \gamma)$. Με τη βοήθεια αυτής της συνάρτησης δειγματοληπτούμε από μία ομοιόμορφη

κατανομή με ελάχιστο και μέγιστο τα α και β αντίστοιχα. Το όρισμα γ της συνάρτησης είναι η αρχική τιμή (seed) που χρειάζεται για να αρχίσει η δειγματοληψία. Στην επόμενη εντολή στη γραμμή 9, ελευθερώνεται ο κουρέας οπότε προστίθεται το 1 στην μεταβλητή του συστήματος U.BARBER(1), που έτσι γίνεται στην περίπτωση μας 1 δίνοντας τη δυνατότητα εξυπηρέτησης άλλου πελάτη.

CLOSE.DOORS

Αυτό το τμήμα (module) του προγράμματος καλείται μόνο μία φορά τη στιγμή της λήξης της εργάσιμης μέρας για να διακόψει τη γέννηση νέων αφίξεων πελατών και να αφαιρέσει από τη λίστα σημειώσεων διαδικασιών κάθε σημείωση διαδικασίας που έχει σχέση με τη γέννηση αφίξεων πελατών.

REPORT

Τέλος το τμήμα (module) REPORT του προγράμματος είναι αυτό που δημιουργεί την τελική έκθεση (report) αποτελεσμάτων από την εκτέλεση της προσομοίωσης του συστήματος του κουρείου.

5.2.1 Κάνοντας το μοντέλο του κουρείου πιο πολύπλοκο

Ας θεωρήσουμε ότι το κουρείο αποκτά και ένα νέο υπάλληλο, τον ταμιά, ο οποίος αναλαμβάνει την εξυπηρέτηση των πελατών αμέσως μόλις τελειώσουν από τον κουρέα και θα πρέπει να πληρώσουν. Επίσης πουλά σε αυτούς τους πελάτες κάποια προϊόντα σχετικά με το αντικείμενο της δουλειάς. Θεωρούμε ότι ο χρόνος εξυπηρέτησης από τον ταμιά ακολουθεί ομοιόμορφη κατανομή. Επίσης θεωρούμε ότι το κουρείο έχει έξι πολυθρόνες αναμονής που αν είναι κατειλημμένες οι νέοι πελάτες φεύγουν έχοντας έτσι χαμένους πελάτες (balks) για το κουρείο. Επίσης λόγω αύξησης της ζήτησης σχεδιάζεται η πρόσληψη ενός ή δύο ακόμα ατόμων για να κουρεύουν.

Στη συνέχεια θα παρουσιάσουμε το πρόγραμμα που μοντελοποιεί το διαμορφωμένο (πολύπλοκο) σύστημα του κουρείου.

```

1 Preamble
2
3   processes include arrival.generator, close.doors, and customer
4
5   resources include barber and cashier
6
7   define  day.length, delay.in.queue.in.barber, delay.in.queue.in.cashier,
8           mean.interarrival.time, min.service.time.in.barber,
9           max.service.time.in.barber, min.service.time.in.cashier,
10          and max.service.time.in.cashier as real variables
11
12   define  num.balks, num.barbers, num.cashiers, and num.chairs as
13          integer variables
14
15   define .minutes to mean units
16
17   tally  max.delay.in.queue.in.barber as the maximum,
18          mean.delay.in.queue.in.barber as the mean,
19          and num.served as the number of delay.in.queue.in.barber
20
21   tally  max.delay.in.queue.in.cashier as the maximum,
22          and mean.delay.in.queue.in.cashier as the mean
23          of delay.in.queue.in.cashier
24
25   accumulate  avg.number.in.queue.in.barber as the average and
26               max.number.in.queue.in.barber as the maximum of n.q.barber
27
28   accumulate  avg.number.in.queue.in.cashier as the average and
29               max.number.in.queue.in.cashier as the maximum of n.q.cashier
30
31   accumulate util.barber as the average of n.x.barber
32
33   accumulate util.cashier as the average of n.x.cashier
34
35 End "Preamble

```

```

1 Main
2
3   call read.data
4   call initialize
5
6   start simulation
7
8   call report
9
10 End "Main

```

```

1 Process close.doors
2
3   interrupt arrival.generator
4
5 End "close.doors

```

```

1 Process arrival.generator
2
3   while time.v < day.length
4     do
5       wait exponential.f(mean.interarrival.time, 1) .minutes
6       activate a customer now
7     loop
8
9 End "arrival.generator

1 Routine initialize
2
3   create every barber(1)
4   let u.barber(1) = num.barbers
5
6   create every cashier(1)
7   let u.cashier(1) = num.cashiers
8
9   activate an arrival.generator now
10  activate a close.doors in day.length .minutes
11
12 End "initialize

1 Process customer
2
3   define haircut.finish.time and time.of.arrival as real variables
4
5   if n.q.barber(1) < num.chairs
6
7     let time.of.arrival = time.v
8     request 1 barber(1)
9     let delay.in.queue.in.barber = time.v - time.of.arrival
10    work uniform.f(min.service.time.in.barber,
11                  max.service.time.in.barber, 2) .minutes
12    relinquish 1 barber(1)
13
14    let haircut.finish.time = time.v
15    request 1 cashier(1)
16    let delay.in.queue.in.cashier = time.v - haircut.finish.time
17    work uniform.f(min.service.time.in.cashier,
18                  max.service.time.in.cashier, 3) .minutes
19    relinquish 1 cashier(1)
29
30  else
31
32    add 1 to num.balks
33
34  always
35
36 End "customer

```



```

1 Routine read.data
2
3   read day.length, mean.interarrival.time, min.service.time.in.barber,
4       max.service.time.in.barber, min.service.time.in.cashier,
5       max.service.time.in.cashier, num.barbers, num.cashiers,
6       and num.chairs
7
8   print 4 lines thus
9
10 BARBERSHOP MODEL -- INPUT PARAMETERS
11 -----
12
13   skip 3 lines
14
15   print 15 lines with num.barbers, num.cashiers, num.chairs,
16       mean.interarrival.time, min.service.time.in.barber,
17       max.service.time.in.barber, min.service.time.in.cashier,
18       max.service.time.in.cashier, and day.length thus
19
20 Number of barbers **
21
22 Number of cashiers **
23
24 Number of chairs **
25
26 Interarrival times are exponential with mean **.* minutes
27
28 Service times in barber are uniform on the interval (**.*, **.*) minutes
29
30 Service times in cashier are uniform on the interval (**.*, **.*) minutes
31
32 Doors close after ****.* minutes
33
34   start new page
35
36 End "read.data

```



```

1 Routine report
2
3   print 4 line thus
4
5 SIMULATION RESULTS (ALL TIMES ARE IN MINUTES)
6 -----
7
8   skip 3 lines
9
10   print 5 lines with time.v and num.balks thus
11
12 Simulation ended at ****.**
13
14 Number of balks ***
15

```

```

16 skip 3 lines
17
18 print 9 lines with num.served, mean.delay.in.queue.in.barber, and
19 max.delay.in.queue.in.barber thus
20
21 Barber
22 -----
23
24 Total number of          Mean time          Max time
25 customers served        in queue        in queue
26 -----
27      *****          *****.**          *****.**
28
29 skip 3 lines
30
31 print 6 lines with util.barber(1)/num.barbers,
32 avg.number.in.queue.in.barber(1), and
33 max.number.in.queue.in.barber(1) thus
34
35 Barber          Average number of          Maximum number of
36 utilization      customers in queue        customers in queue
37 -----
38      ***.**          *****.**          *****.**
39
40 skip 3 lines
41
42 print 9 lines with mean.delay.in.queue.in.cashier,
43 and max.delay.in.queue.in.cashier thus
44
45 Cashier
46 -----
47
48 Mean time          Max time
49 in queue          in queue
50 -----
51      *****.**          *****.**
52
53 skip 3 lines
54
55 print 6 lines with util.cashier(1)/num.cashiers,
56 avg.number.in.queue.in.cashier(1), and
57 max.number.in.queue.in.cashier(1) thus
58
59 Cashier          Average number of          Maximum number of
60 utilization      customers in queue        customers in queue
61 -----
62      ***.**          *****.**          *****.**
63
64 End "report

```

Στο παραπάνω μοντέλο προσομοίωσης χρησιμοποιήσαμε τις εξής μεταβλητές:

day.length	ο συνολικός χρόνος με ανοιχτές τις πόρτες
delay.in.queue.in.barber	η καθυστέρηση ενός πελάτη στην ουρά του κουρέα
delay.in.queue.in.cashier	η καθυστέρηση ενός πελάτη στην ουρά του ταμιά
mean.interarrival.time	ο μέσος χρόνος μεταξύ αφίξεων
min.service.time.in.barber	ο ελάχιστος χρόνος εξυπηρέτησης του κουρέα
max.service.time.in.barber	ο μέγιστος χρόνος εξυπηρέτησης του κουρέα
min.service.time.in.cashier	ο ελάχιστος χρόνος εξυπηρέτησης του ταμιά
max.service.time.in.cashier	ο μέγιστος χρόνος εξυπηρέτησης του ταμιά
num.barbers	ο αριθμός των κουρέων
num.cashiers	ο αριθμός των ταμιών
num.balks	ο αριθμός των χαμένων πελατών
num.served	ο αριθμός πελατών που εξυπηρετήθηκαν
mean.delay.in.queue.in.barber	η μέση καθυστέρηση πελατών στην ουρά του κουρέα
mean.delay.in.queue.in.cashier	η μέση καθυστέρηση πελατών στην ουρά του ταμιά
max.delay.in.queue.in.barber	η μέγιστη καθυστέρηση πελατών στην ουρά του κουρέα
max.delay.in.queue.in.cashier	η μέγιστη καθυστέρηση πελατών στην ουρά του ταμιά
avg.number.in.queue.in.barber	ο μέσος αριθμός πελατών στην ουρά του κουρέα
avg.number.in.queue.in.cashier	ο μέσος αριθμός πελατών στην ουρά του ταμιά
max.number.in.queue.in.barber	ο μέγιστος αριθμός πελατών στην ουρά του κουρέα
max.number.in.queue.in.cashier	ο μέγιστος αριθμός πελατών στην ουρά του ταμιά
util.barber	η καθαρή απασχόληση του κουρέα
util.cashier	η καθαρή απασχόληση του ταμιά
time.of.arrival	ο χρόνος άφιξης πελάτη στο κούρεο
haircut.finish.time	ο χρόνος που τελειώνει το κούρεμα

5.3 Κατανοώντας τη λειτουργία της Simgscript II.5

Μέχρι τώρα είδαμε πως χρησιμοποιούμε το περιβάλλον SimLab για να υλοποιήσουμε ένα ήδη έτοιμο σχεδιασμένο μοντέλο. Σε αυτό το σημείο θα δούμε πως εσωτερικά αναπαριστά ο υπολογιστής τις μεταβλητές και τις διάφορες λειτουργίες που χρειάζονται στην προσομοίωση με σκοπό να κατανοήσουμε τη λειτουργία του ώστε να μπορούμε να σχεδιάσουμε μόνοι μας ένα μοντέλο.

Θυμίζουμε ότι η Simgscript II.5 χρησιμοποιεί τη μέθοδο των διαδικασιών (process) για προσομοίωση διακριτών γεγονότων (discrete event simulation).

Για παράδειγμα θα χρησιμοποιήσουμε τη μελέτη του μοντέλου ενός κουρείου. Οι βασικές διαδικασίες που χαρακτηρίζουν αυτό το μοντέλο είναι η διαδικασία της γέννησης αφίξεων πελατών στο κουρείο και η διαδικασία εξυπηρέτησης του πελάτη από τον κουρέα. Υπάρχει επίσης και η διαδικασία του κλεισίματος του κουρείου όπου στην ουσία εδώ γίνεται η απενεργοποίηση της διαδικασίας γέννησης αφίξεων. Στο μοντέλο έχουμε επίσης έναν κουρέα, δηλαδή μια θέση εξυπηρέτησης (server).

Για την εσωτερική αναπαράσταση του μοντέλου και διαδικασίας της προσομοίωσης στον υπολογιστή, πρέπει να ορίσουμε τις εξής έννοιες:

Ρολόι προσομοίωσης: Δείχνει τον τρέχων χρόνο προσομοίωσης.

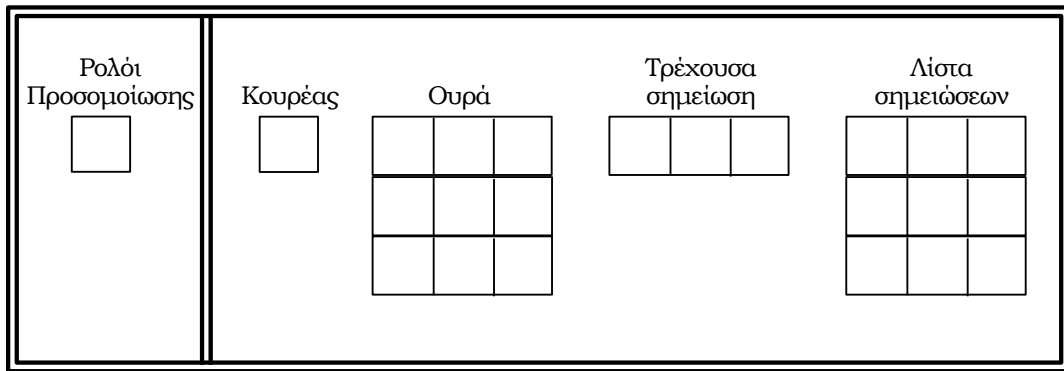
Θέση εξυπηρέτησης (server): Δείχνει τη κατάσταση του σημείου εξυπηρέτησης. Παίρνει τιμή B (busy = απασχολημένος) ή τιμή I (idle = ελεύθερος).

Ουρά (queue): Δομή δεδομένων (λίστα) τύπου FIFO. Στην ουρά κρατούνται σημειώσεις διαδικασιών (process notices) που δεν εξυπηρετούνται γιατί η θέση εξυπηρέτησης είναι απασχολημένη.

Τρέχουσα σημείωση διαδικασίας (current process notice): Η σημείωση της διαδικασίας που εκείνη τη στιγμή ενεργοποιείται.

Λίστα σημειώσεων διαδικασιών (process notices list): Λίστα όπου κρατούνται οι σημειώσεις για τις διαδικασίες που δημιουργούνται ή που πρόκειται να εκτελεστούν ύστερα από διακοπή τους.

Στο πιο κάτω σχήμα 5.1 βλέπουμε μια εσωτερική αναπαράσταση και με βάση αυτό το σχήμα θα παρακολουθήσουμε πως εξελίσσεται η διαδικασία της προσομοίωσης δοκιμάζοντας ένα υπολογιστικό / αριθμητικό παράδειγμα για το μοντέλο του κουρείου.



Σχήμα 5.1

Στο σχήμα 5.2 βλέπουμε το λογικό διάγραμμα της διαδικασίας γέννησης αφίξεων (process arrival.generator) και στο σχήμα 5.3 βλέπουμε το λογικό διάγραμμα της διαδικασίας εξυπηρέτησης πελάτη (process customer).

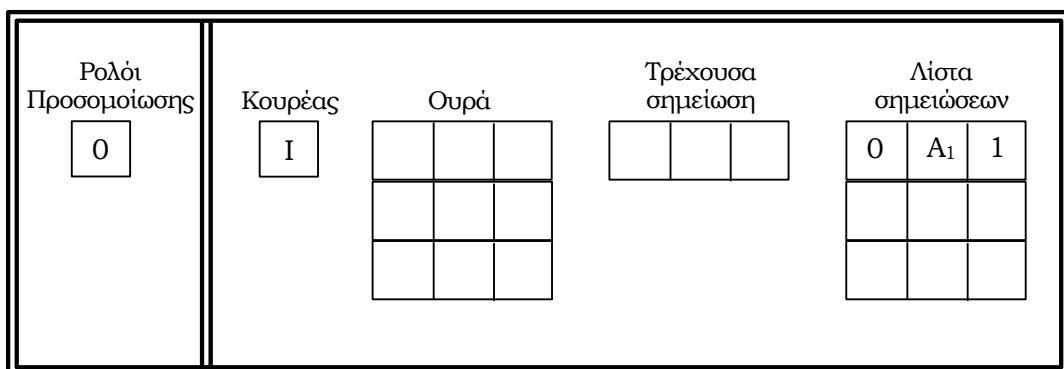
Θα εξηγήσουμε την εσωτερική αναπαράσταση κατά τη διάρκεια της εκτέλεσης της προσομοίωσης δείχνοντας τη διαδικασία με την οποία η ρουτίνα (ελέγχου ροής) χρόνου (timing routine) επιτρέπει την εκτέλεση διαδικασιών που έχουν σημείωση στη λίστα σημειώσεων κάνοντας τις τρέχουσες διαδικασίες.

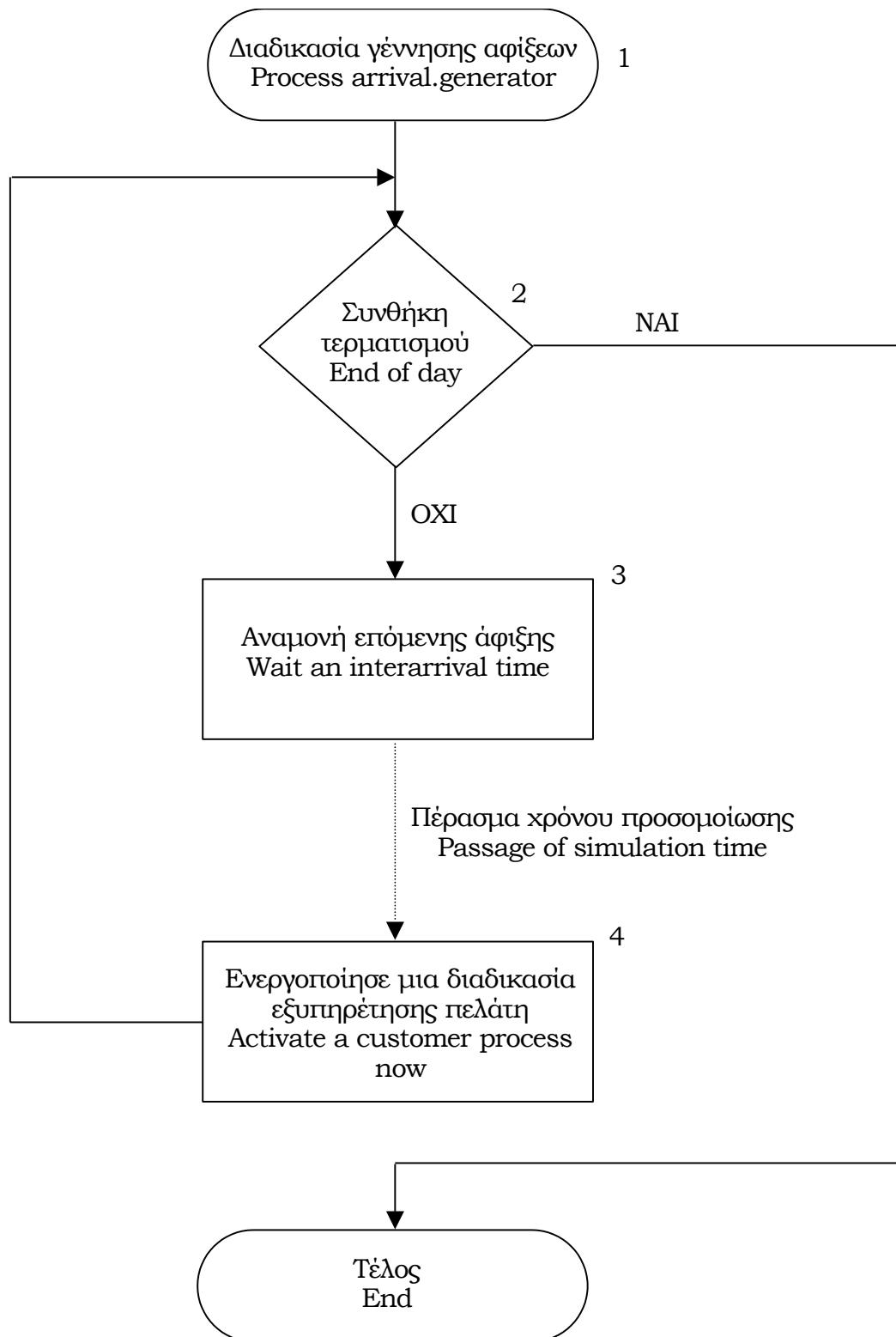
Έστω t_i ο χρόνος άφιξης του i πελάτη στο σύστημα ($t_0=0$) και έστω $T_i = t_i - t_{i-1}$ ο ενδιάμεσος χρόνος αφίξεων (interarrival time) μεταξύ της $(i-1)$ και της i άφιξης πελάτη. Επίσης έστω S_i ο χρόνος εξυπηρέτησης του i πελάτη από τον κουρέα

Ας θεωρήσουμε ότι αυτοί οι χρόνοι είναι γνωστοί και είναι οι εξής:

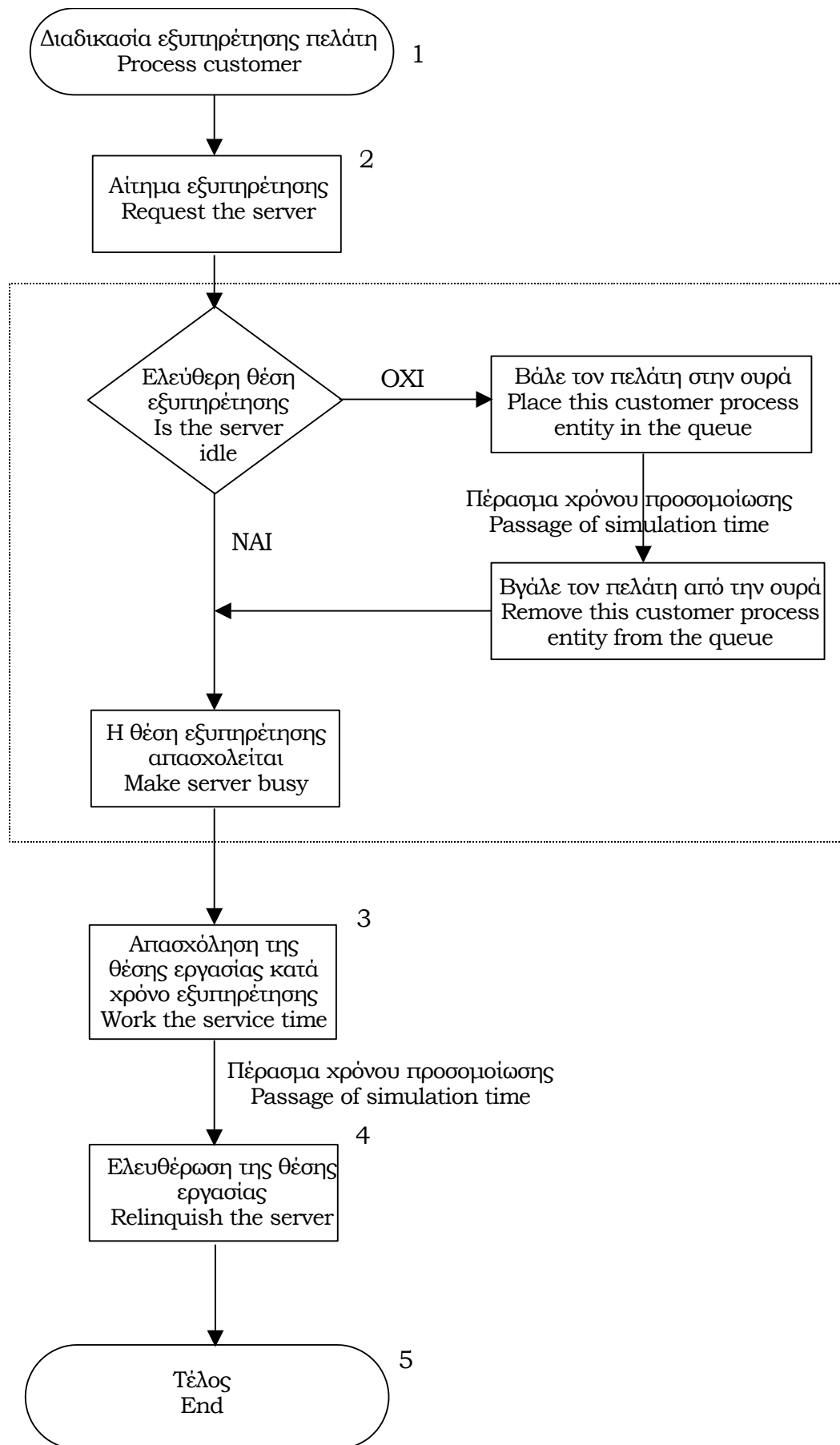
$T_1 = 55, T_2 = 32, T_3 = 24 \dots$ και $S_1 = 43, S_2 = 36 \dots$

Τη χρονική στιγμή 0 το κύριο τμήμα (Main) του προγράμματος καλεί τη ρουτίνα (initialize) που αρχικοποιεί το μοντέλο μας. Ύστερα από αυτή την αρχικοποίηση η εσωτερική αναπαράσταση φαίνεται στο παρακάτω σχήμα.





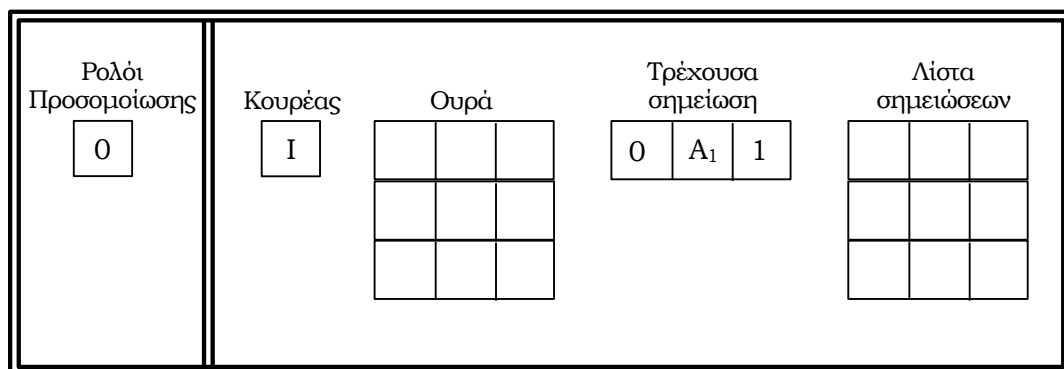
Σχήμα 5.2



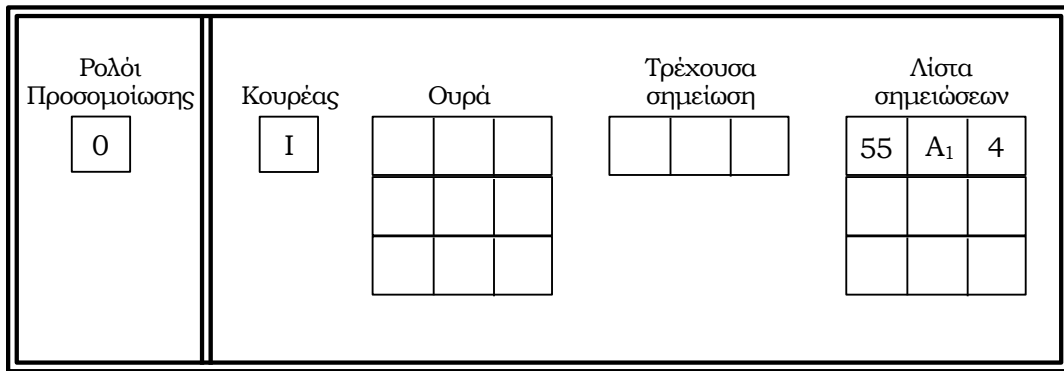
Σχήμα 5.3

Βλέπουμε ότι η θέση εξυπηρέτησης έχει χαρακτηριστεί σαν ελεύθερη, η ουρά είναι κενή, το ρολόι προσομοίωσης δείχνει μηδέν και έχει σημειωθεί στην αντίστοιχη λίστα σημειώσεων (process notices list) η σημείωση της διαδικασίας γέννησης αφίξεων, η οποία συμβολίζεται με το γράμμα A. Ο αντίστοιχος συμβολισμός της σημείωσης της διαδικασίας είναι $(0, A_1, 1)$. Στην ουσία η σημείωση διαδικασίας (process notice) είναι μια εγγραφή (record) που αναπαριστάται από την τριάδα (t, P_i, b) , όπου t είναι ο χρόνος ενεργοποίησης της διαδικασίας, το P_i συμβολίζει την i υλοποίηση της διαδικασίας τύπου P και b είναι το σημείο του λογικού διαγράμματος της διαδικασίας από όπου θα συνεχιστεί η εκτέλεση της διαδικασίας όταν αυτή επανενεργοποιηθεί.

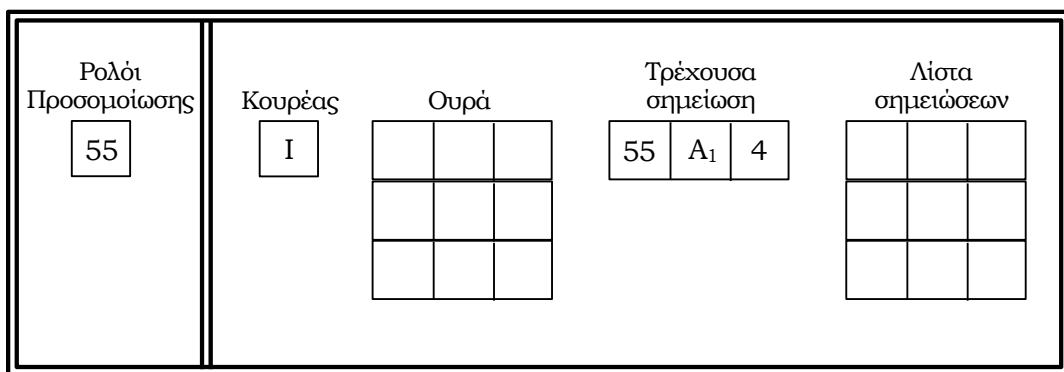
Έτσι τη χρονική στιγμή 0 και αφού ξεκινήσει η διαδικασία της προσομοίωσης με την εντολή START SIMULATION, η ρουτίνα χρόνου (timing routine) καλείται να καθορίσει την πιο πρόσφατη σημείωση από τη λίστα σημειώσεων, αυτή δηλαδή που ο χρόνος ενεργοποίησής της είναι πιο κοντά στο χρόνο που δείχνει το ρολόι της προσομοίωσης. Εδώ, επειδή έχουμε μόνο μια σημείωση στη λίστα, γίνεται τρέχουσα αυτή η σημείωση διαδικασίας. Η εκτέλεση αυτής της διαδικασίας ξεκινά από το σημείο 1 του λογικού διαγράμματός της, όπως λέει και η σχετική σημείωση. Σε αυτό το σημείο ενημερώνεται το ρολόι προσομοίωσης παίρνοντας σαν τιμή το χρόνο ενεργοποίησης της τρέχουσας διαδικασίας, εδώ την τιμή 0. Η εσωτερική αναπαράσταση αυτή τη στιγμή είναι ως εξής:



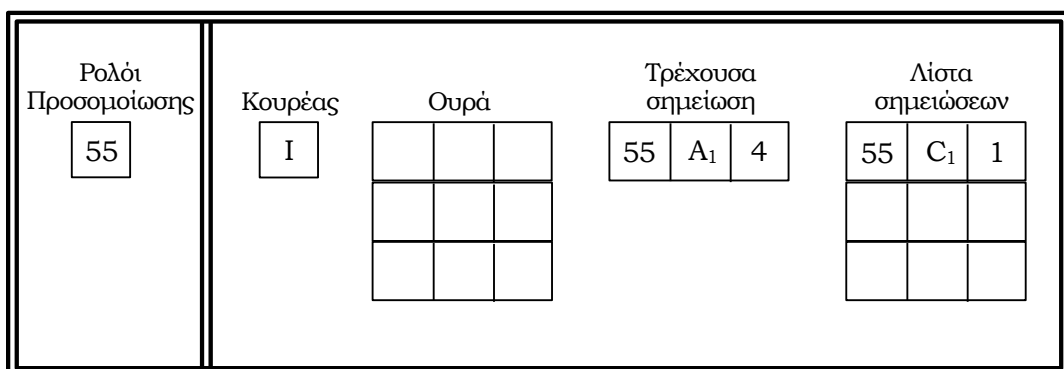
Ξεκινώντας λοιπόν η εκτέλεση αυτής της διαδικασίας από το σημείο 1 του λογικού της διαγράμματός της, περνά από το σημείο 2 όπου γίνεται ο έλεγχος για το αν εκπληρώνεται η συνθήκη τερματισμού της συγκεκριμένης διαδικασίας. Αρχικά, για το παράδειγμά μας, θεωρούμε πως από αυτό το σημείο θα περνάμε έχοντας πάντα σαν απάντηση το όχι και άρα περνάμε στο σημείο 3. Εδώ έχουμε πέρασμα χρόνου προσομοίωσης ίσο με τον πρώτο ενδιάμεσο χρόνο αφίξεων, δηλαδή $T_1 = 55$ λεπτά. Εφόσον θα έχουμε πέρασμα χρόνου προσομοίωσης, η σημείωση ενημερώνεται ότι ο χρόνος επανενεργοποίησής της είναι ο 55 ($0 + 55$) και ότι όταν επανενεργοποιηθεί η συγκεκριμένη διαδικασία θα συνεχίσει από το σημείο 4 που είναι το ακριβώς επόμενο σημείο από εκεί που απαιτήθηκε πέρασμα χρόνου προσομοίωσης. Έτσι η ενημερωμένη σημείωση ξαναπερνά στη λίστα σημειώσεων περιμένοντας για τη νέα κλήση της. Η εσωτερική αναπαράσταση τώρα είναι:



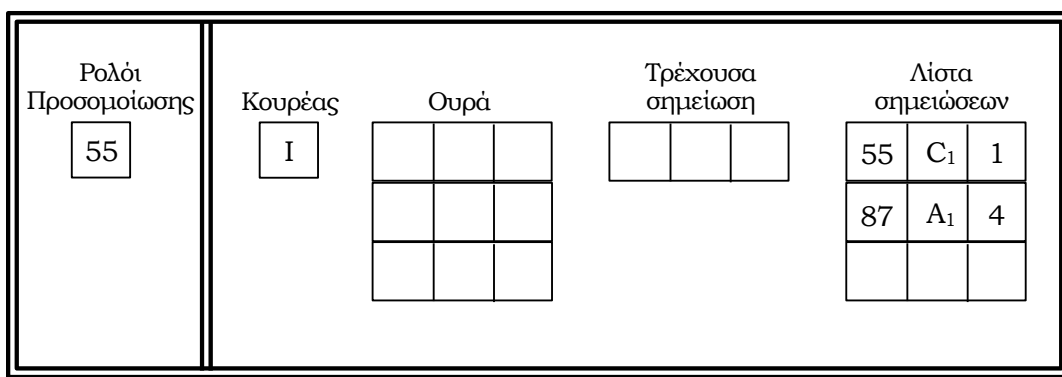
Ο έλεγχος γυρνά τώρα στη ρουτίνα χρόνου (timing routine), η οποία καλεί από τη λίστα σημειώσεων την επόμενη σημείωση (process notice) την πιο κοντινή χρονικά στο 0 που είναι ο τρέχων χρόνος προσομοίωσης (ρολόι προσομοίωσης). Η σημείωση που αυτή τη φορά γίνεται τρέχουσα, είναι πάλι η A₁ που έχει χρόνο ενεργοποίησης 55. Το ρολόι προσομοίωσης ενημερώνεται αμέσως παίρνοντας σαν τιμή το 55 που είναι η χρονική στιγμή ενεργοποίησης της τρέχουσας διαδικασίας. Η εσωτερική αναπαράσταση πλέον είναι:



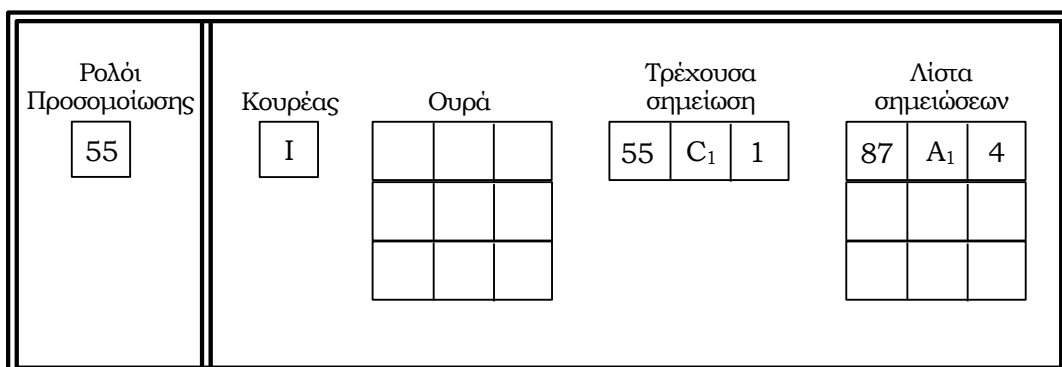
Καλείται λοιπόν να συνεχιστεί η διαδικασία γέννησης αφίξεων A₁ από το σημείο 4 του λογικού διαγράμματος. Αφού εκτελεστεί η εντολή “ενεργοποίησε” (activate) τοποθετείται μια νέα σημείωση διαδικασίας στη λίστα σημειώσεων, η σημείωση που αντιπροσωπεύει τη νέα διαδικασία εξυπηρέτησης πελάτη (process customer). Ο χρόνος ενεργοποίησης της είναι 55 εφόσον στην εντολή “ενεργοποίησε” (activate) δηλώνεται ότι ο χρόνος ενεργοποίησης είναι “τώρα” (now). Αφού είναι νέα διαδικασία, προφανώς μόλις ενεργοποιηθεί θα ξεκινήσει από το σημείο 1. Η εσωτερική αναπαράσταση στο σημείο αυτό είναι:



Η τοποθέτηση της νέας σημείωσης διαδικασίας (process notice) που μόλις δημιουργήθηκε δεν σημαίνει πέρασμα χρόνου προσομοίωσης και έτσι συνεχίζεται να εκτελείται η προηγούμενη (ενεργή) διαδικασία A_1 περνώντας από το σημείο 2. Επειδή η απάντηση έχουμε θεωρήσει ότι θα είναι “όχι” προχωράμε στο σημείο 3 όπου έχουμε πάλι πέρασμα χρόνου προσομοίωσης. Η σημείωση θα ενημερωθεί ως προς το νέο χρόνο ενεργοποίησής της, προσθέτοντας στο 55 την επόμενη τιμή (T_2) από τη δειγματοληψία των ενδιάμεσων χρόνων αφίξεων, που έχουμε θεωρήσει ότι είναι 32, δηλαδή ο νέος χρόνος επανενεργοποίησης της διαδικασίας A_1 θα είναι $55 + 32 = 87$. Το σημείο από όπου θα συνεχιστεί η εκτέλεση της διαδικασίας A_1 , είναι το 4. Η διαδικασία συνεχίζει να συμβολίζεται με A_1 αφού δεν έχει δημιουργηθεί νέα διαδικασία γέννησης αφίξεων αλλά έχουμε συνεχή επανάληψη της αρχικά δημιουργημένης διαδικασίας. Έτσι προστίθεται ακόμα μια σημείωση διαδικασίας (process notice) στη λίστα σημειώσεων και η εσωτερική αναπαράσταση τώρα γίνεται:

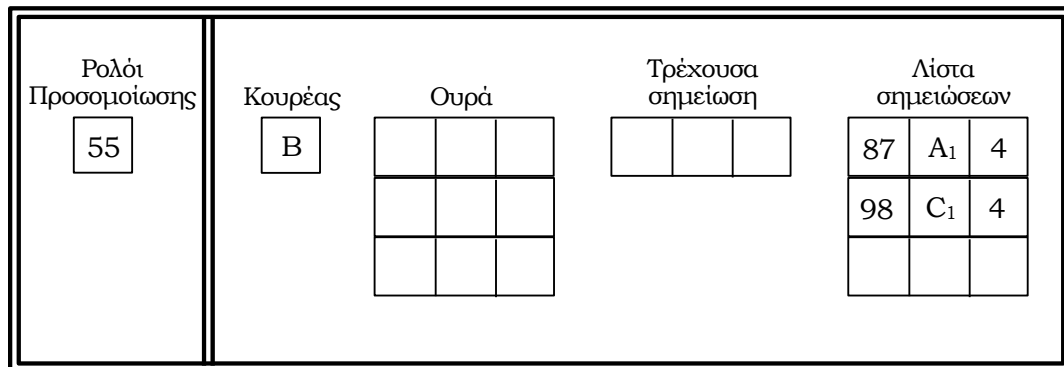


Η ρουτίνα χρόνου (timing routine) για άλλη μια φορά αναζητά στη λίστα σημειώσεων την πιο κοντινή χρονικά στο 55 που είναι ο χρόνος του ρολογιού προσομοίωσης. Αυτή τη φορά η σημείωση που γίνεται τρέχουσα είναι αυτή που αντιστοιχεί στη διαδικασία C_1 , δηλαδή στη διαδικασία εξυπηρέτησης πελάτη. Το ρολόι προσομοίωσης παραμένει στη τιμή 55 αφού η τιμή του χρόνου ενεργοποίησης της τρέχουσας διαδικασίας είναι 55. Η εσωτερική αναπαράσταση τώρα γίνεται:

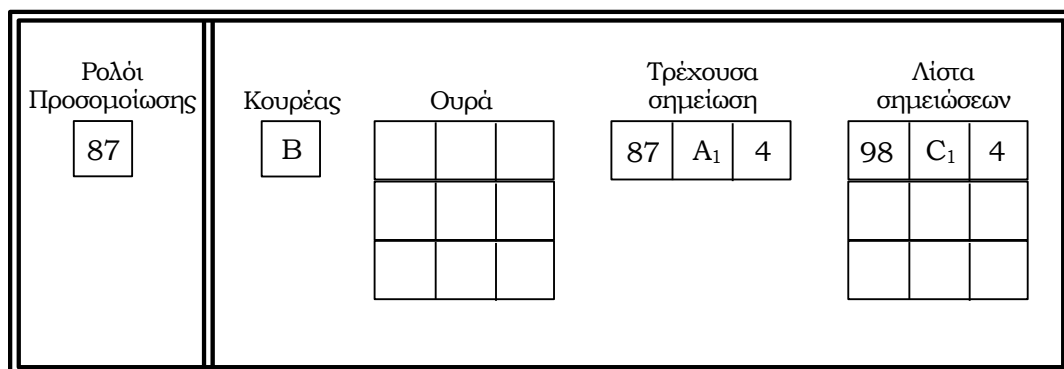


Τώρα ξεκινά η εκτέλεση της διαδικασίας C_1 (διαδικασία εξυπηρέτησης πελάτη – process customer) από το σημείο 1. Περνώντας στο επόμενο σημείο 2 γίνεται αίτηση για απασχόληση της θέσης εξυπηρέτησης (server). Η απάντηση από τον έλεγχο του επόμενου σημείου είναι “ναι” εφόσον ο κουρέας είναι ελεύθερος

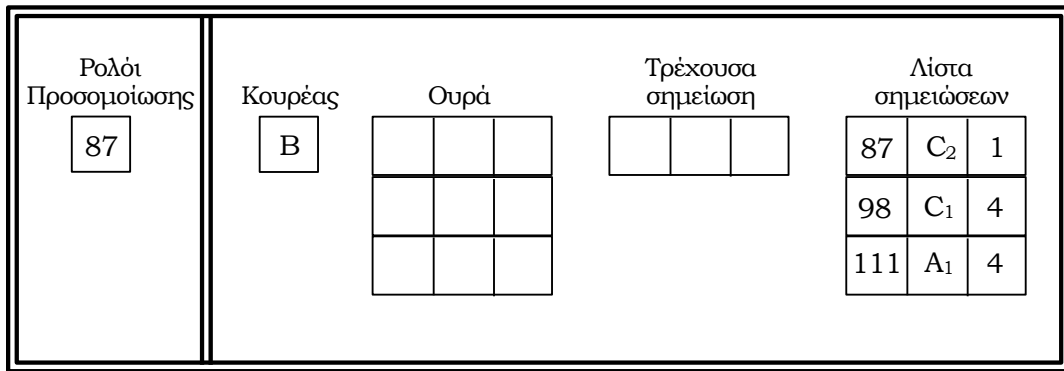
(idle). Η μεταβλητή που μας δείχνει τη κατάσταση του server τώρα γίνεται B (busy – απασχολημένος) και η διαδικασία προχωράει στο σημείο 3. Εδώ έχουμε πέρασμα χρόνου προσομοίωσης ίσο με το χρόνο εξυπηρέτησης που για τον πρώτο πελάτη είναι $S_1 = 43$ λεπτά. Αφού πάλι έχουμε πέρασμα χρόνου η διαδικασία θα διακοπή και η σημείωσή της θα περάσει στη λίστα σημειώσεων, αφού πρώτα ενημερωθεί με το νέο χρόνο ενεργοποίησης της διαδικασίας ($55+43=98$) και με τον αριθμό του σημείου (εδώ το 4) από το οποίο θα συνεχισθεί η διαδικασία όταν επανενεργοποιηθεί. Τώρα η εσωτερική αναπαράσταση γίνεται:



Συνεχίζοντας η ρουτίνα χρόνου (timing routine) καλεί τη σημείωση (87, A₁, 4) από τη λίστα σημειώσεων, ενημερώνοντας το ρολόι προσομοίωσης με την τιμή 87 και συνεχίζοντας τη διαδικασία A από το σημείο 4. Έτσι η εσωτερική αναπαράσταση γίνεται:

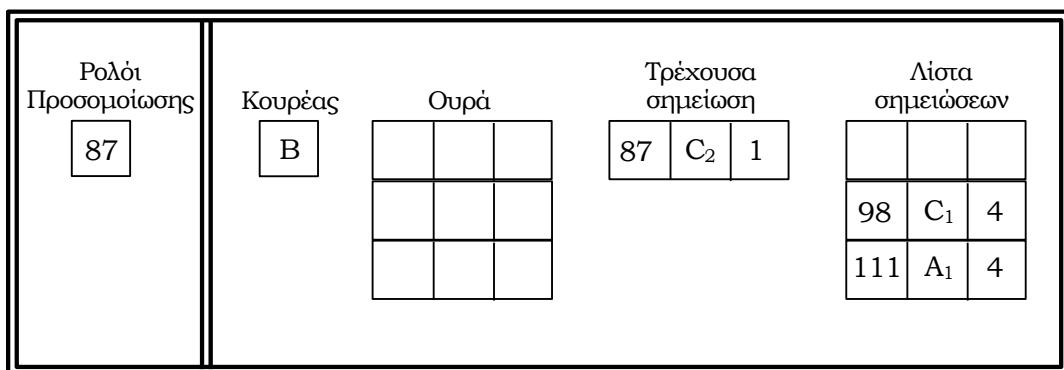


Στο σημείο 4 της διαδικασίας A₁ δημιουργείται μια καινούργια σημείωση στη λίστα σημειώσεων για την διαδικασία C₂ (όπως πριν). Στη συνέχεια περνάμε από το σημείο 2 (εδώ έχουμε απάντηση “όχι”) και καταλήγουμε στο σημείο 3, όπου έχουμε πέρασμα χρόνου προσομοίωσης ίσο με τον ενδιάμεσο χρόνο αφίξεων $T_3 = 24$ λεπτά και έτσι επιστρέφοντας η σημείωση στη λίστα σημειώσεων η εσωτερική αναπαράσταση γίνεται:

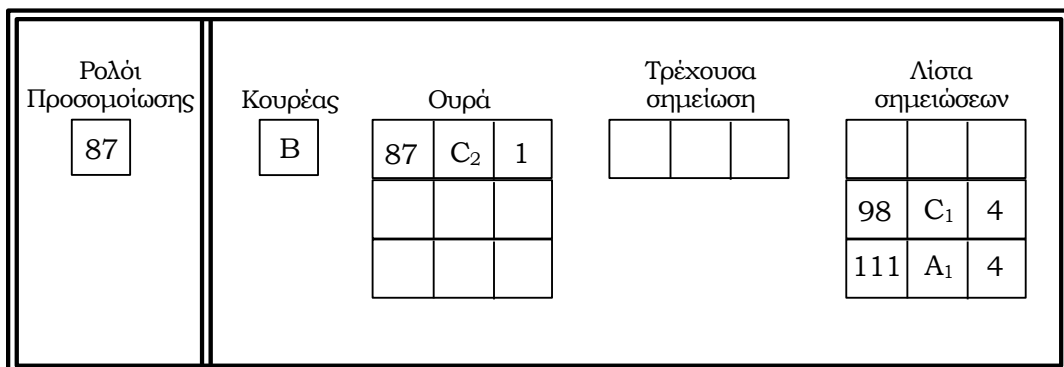


Παρατηρούμε ότι ο πελάτης 1 παραμένει μέσα στο σύστημα παρόλο που διακόπηκε η διαδικασία εξυπηρέτησής του. Η διακοπή αυτή δεν αντιπροσωπεύει φυσική διακοπή εξυπηρέτησης του πελάτη αλλά διακοπή της διαδικασίας της προσομοίωσης για να ελέγξει ενδιάμεσες διαδικασίες που πρέπει να προωθηθούν πριν ολοκληρωθεί φυσικά η διαδικασία εξυπηρέτησης του πελάτη 1 (C₁). (Πράγματι όπως θα δούμε στη συνέχεια πρέπει να προωθηθεί η διαδικασία C₂ ώστε ο πελάτης που έρχεται στο σύστημα να μπει στην ουρά αναμονής όσο εξυπηρετείται ο πρώτος πελάτης).

Η ρουτίνα χρόνου (timing routine) καλεί τώρα από την λίστα σημειώσεων τη λίστα (87, C₂, 1) κάνοντάς της τρέχουσα. Η εσωτερική αναπαράσταση τώρα γίνεται:



Η τρέχουσα διαδικασία (C₂) ξεκινά από το σημείο 1 και ζητά εξυπηρέτηση από το server. Επειδή όμως αυτός είναι απασχολημένος (B = busy) η διαδικασία εξυπηρέτησης πελάτη (C₂) μπαίνει σε αναμονή και η σημείωσή της στην ουρά. Έτσι η εσωτερική αναπαράσταση γίνεται:



Η σημείωση αυτής της διαδικασίας παραμένει στην ουρά μέχρι να ελευθερωθεί η θέση εξυπηρέτησης (server). Ο χρόνος ενεργοποίησης και το σημείο από όπου θα συνεχίσει πλέον δεν ισχύουν μέχρι να ξανατοποθετηθεί η σημείωση στη λίστα σημειώσεων.

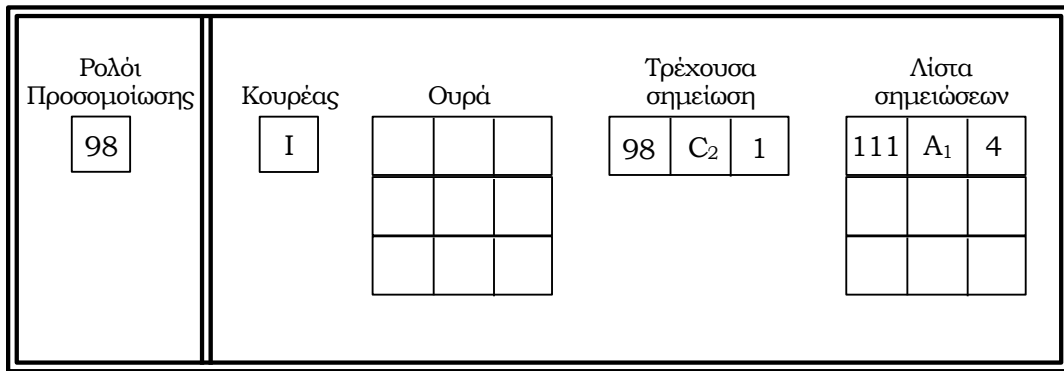
Επειδή η αναμονή στην ουρά σημαίνει πέρασμα χρόνου προσομοίωσης, αναλαμβάνει πάλι τον έλεγχο η ρουτίνα χρόνου (timing routine) που καλεί από τη λίστα σημειώσεων την επόμενη σημείωση (98, C_1 , 4) την οποία την κάνει τρέχουσα. Το ρολόι προσομοίωσης ενημερώνεται και η εσωτερική αναπαράσταση γίνεται:

Ρολόι Προσομοίωσης	Κουρέας	Ουρά	Τρέχουσα σημείωση	Λίστα σημειώσεων
98	B	87 C ₂ 1	98 C ₁ 4	111 A ₁ 4

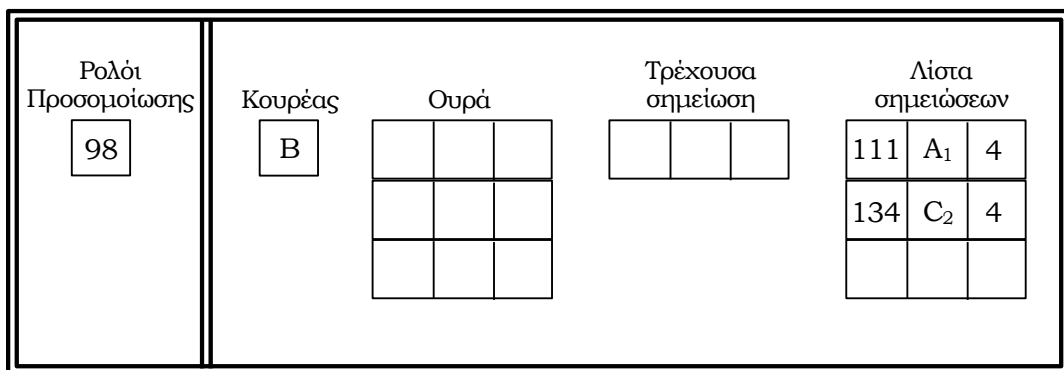
Η διαδικασία C_1 συνεχίζεται από σημείο 4 όπου ελευθερώνεται ο server (idle) και τελειώνει η διαδικασία αυτή, οπότε σβήνεται και η σημείωσή της. Αυτό είναι και το σημείο που αποχωρεί ο πρώτος πελάτης από το κουρείο. Τώρα η εσωτερική αναπαράσταση γίνεται:

Ρολόι Προσομοίωσης	Κουρέας	Ουρά			Τρέχουσα σημείωση			Λίστα σημειώσεων		
98	I	87	C ₂	1				111	A ₁	4

Με την ελευθέρωση της θέσης εξυπηρέτησης, γίνεται έλεγχος αν υπάρχει σημείωση στην ουρά (δηλαδή αν υπάρχει διαδικασία σε αναμονή εξυπηρέτησης). Αν υπάρχει, όπως στην περίπτωση μας, αφαιρείται από την ουρά και γίνεται τρέχουσα, οπότε η εσωτερική αναπαράσταση γίνεται:



Τώρα η θέση εξυπηρέτησης (server) χαρακτηρίζεται πάλι B (busy) και η διαδικασία C₂ που ξεκινάει από το σημείο 1, περνώντας από το σημείο 2, φτάνει στο σημείο 3 του λογικού διαγράμματός της. Εδώ έχουμε πέρασμα χρόνου προσομοίωσης ίσο με το χρόνο εξυπηρέτησης που για τον δεύτερο πελάτη είναι S₂ = 36 λεπτά. Αφού πάλι έχουμε πέρασμα χρόνου η διαδικασία θα διακοπή και η σημείωσή της θα περάσει στη λίστα σημειώσεων, αφού πρώτα ενημερωθεί με το νέο χρόνο ενεργοποίησης της διαδικασίας (98+36=134) και με τον αριθμό του σημείου (εδώ το 4) από το οποίο θα συνεχισθεί η διαδικασία όταν επανενεργοποιηθεί. Τώρα η εσωτερική αναπαράσταση γίνεται:



Έτσι συνεχίζεται η διαδικασία της προσομοίωσης μέχρι να ικανοποιηθεί η συνθήκη τερματισμού γέννησης νέων αφίξεων και να εξυπηρετηθούν όλες οι διαδικασίες που περιμένουν στην ουρά αναμονής.

Παράρτημα 1ο: Οι εντολές του SimEdit

Μετακινήσεις

Ctrl+E = βελάκι πάνω	Κινεί τον δείκτη θέσης (cursor) μια γραμμή πάνω.
Ctrl+X = βελάκι κάτω	Κινεί τον δείκτη θέσης (cursor) μια γραμμή κάτω.
Ctrl+D = βελάκι δεξιά	Κινεί τον δείκτη θέσης (cursor) κατά ένα χαρακτήρα δεξιά.
Ctrl+S = βελάκι αριστερά	Κινεί τον δείκτη θέσης (cursor) κατά ένα χαρακτήρα αριστερά.
Ctrl+R = PgUp	Κινεί τον δείκτη θέσης (cursor) κατά μία σελίδα πάνω.
Ctrl+C = PgDn	Κινεί τον δείκτη θέσης (cursor) κατά μία σελίδα κάτω.
Ctrl+F	Κινεί τον δείκτη θέσης (cursor) στον πρώτο χαρακτήρα της επόμενης λέξης.
Ctrl+A	Κινεί τον δείκτη θέσης (cursor) στον πρώτο χαρακτήρα της προηγούμενης λέξης.
End	Κινεί τον δείκτη θέσης (cursor) στο τέλος της γραμμής.
Home	Κινεί τον δείκτη θέσης (cursor) στην αρχή της γραμμής.
Ctrl+W	Κάνει την οθόνη να κυλήσει (scroll) κατά μία γραμμή πάνω.
Ctrl+Z	Κάνει την οθόνη να κυλήσει (scroll) κατά μία γραμμή κάτω.
Ctrl+Q+R = Ctrl+PgUp	Κινεί τον δείκτη θέσης στην αρχή του αρχείου.
Ctrl+Q+C = Ctrl+PgDn	Κινεί τον δείκτη θέσης στο τέλος του αρχείου.

Διαγραφές

Backspace	Διαγράφει τον χαρακτήρα που βρίσκεται αριστερά από τον δείκτη θέσης.
Ctrl+G = Del	Διαγράφει τον χαρακτήρα που βρίσκεται ακριβώς στον δείκτη θέσης.
Ctrl+T	Διαγράφει τη λέξη δεξιά από τον δείκτη θέσης.
Ctrl+Y	Διαγράφει τη γραμμή στην οποία βρίσκεται ο δείκτης θέσης.
Ctrl+U+Y	Επαναφέρει την τελευταία γραμμή που διαγράφηκε.

Εντολές τμηματοποίησης (block)

Ctrl+K+B	Σημειώνει (mark) την αρχή του τμήματος (block) κειμένου.
Ctrl+K+K	Σημειώνει (mark) το τέλος του τμήματος (block) κειμένου.
Ctrl+K+C	Αντιγράφει (copy) το σημειωμένο τμήμα κειμένου.
Ctrl+K+V	Μετακινεί (move) το σημειωμένο τμήμα κειμένου.
Ctrl+K+Y	Διαγράφει (delete) το σημειωμένο τμήμα κειμένου.
Ctrl+K+H	Εξαφανίζει (hide) τα σημάδια σημείωσης του τμήματος κειμένου.
Ctrl+K+R	Διαβάξει τμήμα κειμένου από ένα ενδιάμεσο σημειωματάριο (clipboard).
Ctrl+K+W	Γράφει το σημειωμένο τμήμα (block) κειμένου στο σημειωματάριο (clipboard).
Ctrl+K+αριθμ.	Σημειώνει μια θέση μέσα στο κείμενο με έναν αριθμό.
Ctrl+Q+αριθμ.	Επιστρέφει το δείκτη θέσης σε μια ήδη σημειωμένη με αριθμό θέση στο κείμενο.
Ctrl+K+S	Ρυθμίζει (indent) προς τα αριστερά τα αριστερά περιθώρια του σημειωμένου τμήματος κειμένου.
Ctrl+K+D	Ρυθμίζει (outdent) προς τα δεξιά τα αριστερά περιθώρια του σημειωμένου τμήματος κειμένου.
Ctrl+Q	Αλλάζει μεταξύ των καταστάσεων “ρύθμιση / όχι ρύθμιση” αριστερών περιθωρίων.

Ενώ βρισκόμαστε στην κατάσταση επεξεργασίας του κειμένου, πιέζοντας το πλήκτρο Esc ερχόμαστε στην κατάσταση εντολών ανοίγοντας ένα παράθυρο στην οθόνη όπου αναμένεται η πληκτρολόγηση κάποιας από τις πιο κάτω εντολές:

EXIT	“Βγαίνει” από τον SimEdit αφού αποθηκεύσει το κείμενο.
SAVE	Αποθηκεύει το κείμενο χωρίς να “βγαίνει” από τον SimEdit.
QUIT	“Βγαίνει” από τον SimEdit χωρίς να αποθηκεύσει το κείμενο.
VIDEO	Επαναφέρει στην κατάσταση επεξεργασίας του κειμένου κλείνοντας το παράθυρο.
AUTOSAVE αριθμ.	Καθορίζει το ρυθμό αυτόματης αποθήκευσης του κειμένου.
COPY	Αντιγράφει το σημειωμένο τμήμα (block) κειμένου.

CURRENT		Δείχνει τον αριθμό της τρέχουσας γραμμής.
DELETE		Διαγράφει το σημειωμένο τμήμα.
FIND		Βρίσκει μια σειρά χαρακτήρων (string) που αναζητούμε.
επιλογή	C	Ελέγχει μικρά / κεφαλαία γράμματα.
	G	Συνολικό ψάξιμο.
	number	Δείχνει ποια από τις επιτυχίες ψαξίματος σε σειρά είναι η ζητούμενη.
GOTO		Μετακινεί τον δείκτη θέσης σε μια συγκεκριμένη γραμμή.
INDENT		Καθορίζει το μέγεθος του αριστερού περιθωρίου.
MOVE		Μετακινεί ένα σημειωμένο τμήμα (block) κειμένου.
READ		Εισάγει αντίγραφο ενός αρχείου.
REPLACE		Αντικαθιστά μια σειρά χαρακτήρων (string) με μια νέα.
επιλογή	G	Συνολικό ψάξιμο.
	C	Ελέγχει μικρά / κεφαλαία γράμματα.
	N	Αντικαθιστά χωρίς να ζητά επιβεβαίωση.
	number	Δείχνει πόσες αντικαταστάσεις να γίνουν.
WRITE		Αντιγράφει το σημειωμένο τμήμα κειμένου (marked block) σε ένα αρχείο.

Παράρτημα 2ο: Οι εντολές του SimLab

DELETE		Διαγράφει ένα συγκεκριμένο αρχείο ή όλα τα αρχεία ενός συγκεκριμένου τύπου.
επιλογή	/DATA	Διαγράφει όλα τα αρχεία τύπου DATA.
	/όνομα	Διαγράφει το αρχείο με το συγκεκριμένο όνομα.
	/WORK	Διαγράφει το αρχείο εργασίας (workfile).
EDIT		Ανοίγει αρχεία για επεξεργασία από τον επεξεργαστή SimEdit του SimLab.
επιλογή	/DATA	Ανοίγει αρχείο δεδομένων για επεξεργασία.
	/NEW	Ανοίγει ένα νέο αρχείο.
	/WORK	Ανοίγει το αρχείο εργασίας (workfile) για επεξεργασία.
	/όνομα	Ανοίγει το αρχείο με το συγκεκριμένο όνομα για επεξεργασία.
KILL		Σταματάει την μετάφραση (compilation) που εκτελείται σε ένα συγκεκριμένο υπό-κατάλογο (subdirectory).
LINK		Ενεργοποιεί τον linker για την κατάλληλη ένωση των διαφόρων μεταφρασμένων τμημάτων (modules) σε ένα ενιαίο εκτελέσιμο πρόγραμμα, αλλά δεν εκτελεί το πρόγραμμα.
QUIT		Μας βγάζει από τον περιβάλλον του SimLab.
RUN		Εκτελεί ένα πρόγραμμα προσομοίωσης.
SELECT		Εμφανίζει λίστα με την κατάσταση των αρχείων που βρίσκονται στο επιλεγμένο (selected) υπο-κατάλογο (subdirectory).
UNLOCK		Ξεκλειδώνει ένα συγκεκριμένο κλειδωμένο (locked) αρχείο ή όλα.
επιλογή	/All	Ξεκλειδώνει όλα τα αρχεία.
	/όνομα	Ξεκλειδώνει το αρχείο με το συγκεκριμένο όνομα.
	/WORK	Ξεκλειδώνει το αρχείο εργασίας (workfile).

Παράρτημα 3ο: Τελεστές σχέσεων

Μαθηματικά Σύμβολα	Επιτρεπτές Μορφές στη Simscript II.5
=	= EQ EQUALS EQUAL TO
≠	≠ NE NOT EQUAL TO
<	< LT LS LESS THAN
>	> GT GR GREATER THAN
≤	≤ LE NO GREATER THAN NOT GREATER THAN
≥	≥ LE NO GREATER THAN NOT GREATER THAN

Παράρτημα 4ο: Συναρτήσεις κατανομών

BETA.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή beta q_1 : δύναμη του x ($q_1 > 0$) q_2 : δύναμη του $1-x$ ($q_2 > 0$) q_3 : καθορισμός σειράς ψευδοτυχαίων.
BINOMIAL.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή binomial q_1 : αριθμός δοκιμών q_2 : πιθανότητα επιτυχίας q_3 : καθορισμός σειράς ψευδοτυχαίων.
ERLANG.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή erlang q_1 : μέσος q_2 : k q_3 : καθορισμός σειράς ψευδοτυχαίων.
EXPONENTIAL.F(q_1, q_2)	τυχαίο δείγμα από κατανομή exponential q_1 : μέσος q_2 : καθορισμός σειράς ψευδοτυχαίων.
GAMMA.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή gamma q_1 : μέσος q_2 : k q_3 : καθορισμός σειράς ψευδοτυχαίων.
ISTEP.F(q_1, q_2)	τυχαίο δείγμα από πίνακα look-up χωρίς interpolation q_1 : μεταβλητή που δείχνει στον πίνακα q_2 : καθορισμός σειράς ψευδοτυχαίων.
LIN.F(q_1, q_2)	τυχαίο δείγμα από πίνακα look-up με γραμμική interpolation q_1 : μεταβλητή που δείχνει στον πίνακα q_2 : καθορισμός σειράς ψευδοτυχαίων.

LOG.NORMAL.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή log-normal q_1: μέσος q_2: τυπική απόκλιση (standard deviation) q_3: καθορισμός σειράς ψευδοτυχαίων.
NORMAL.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή normal q_1: μέσος q_2: τυπική απόκλιση (standard deviation) q_3: καθορισμός σειράς ψευδοτυχαίων.
POISSON.F(q_1, q_2)	τυχαίο δείγμα από κατανομή poisson q_1: μέσος q_2: καθορισμός σειράς ψευδοτυχαίων.
RANDI.F(q_1, q_2, q_3)	τυχαίο δείγμα ομοιόμορφα κατανεμημένο μεταξύ εύρους τιμών q_1: αρχική τιμή q_2: τελική τιμή q_3: καθορισμός σειράς ψευδοτυχαίων.
RANDOM.F(q)	ψευδοτυχαίος αριθμός μεταξύ 0 και 1 q: καθορισμός σειράς ψευδοτυχαίων.
RSTEP.F(q_1, q_2)	τυχαίο δείγμα από πίνακα look-up q_1: μεταβλητή που δείχνει στον πίνακα q_2: καθορισμός σειράς ψευδοτυχαίων.
UNIFORM.F(q_1, q_2, q_3)	τυχαίο δείγμα από ομοιόμορφη κατανομή q_1: αρχική τιμή q_2: τελική τιμή q_3: καθορισμός σειράς ψευδοτυχαίων.
WEIBULL.F(q_1, q_2, q_3)	τυχαίο δείγμα από κατανομή weibull q_1: παράμετρος σχήματος q_2: παράμετρος κλίμακας q_3: καθορισμός σειράς ψευδοτυχαίων.

Παράρτημα 5ο: Στατιστικές Εντολές

Στατιστικές Εντολές της Simsript	Συνώνυμα	Υπολογισμός	Ορισμός
NUMBER	NUM		Ο αριθμός των επιλεγμένων τιμών.
SUM		Σ (expression)	Το άθροισμα των επιλεγμένων τιμών.
MEAN	AVG AVERAGE	SUM/NUMBER	Το άθροισμα των επιλεγμένων τιμών διαιρεμένο με τον αριθμό των επιλεγμένων τιμών.
SUM.OF.SQUARES	SSQ	Σ (expression) ²	Το άθροισμα των τετραγώνων των επιλεγμένων τιμών.
MEAN.SQUARE	MSQ	SUM.OF.SQUARES/ NUMBER	Το άθροισμα των τετραγώνων των επιλεγμένων τιμών διαιρεμένο με τον αριθμό των επιλεγμένων τιμών.
VARIANCE	VAR	MEAN.SQUARE – (MEAN) ²	
STD.DEV	STD	SQRT.F(VARIANCE)	Η τετραγωνική ρίζα της απόκλισης.
MAXIMUM	MAX		Η μέγιστη τιμή από τις επιλεγμένες τιμές.
MINIMUM	MIN		Η ελάχιστη τιμή από τις επιλεγμένες τιμές.

