ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σημειώσεις για το Μάθημα

ΤΕΧΝΙΚΕΣ ΕΚΤΙΜΗΣΗΣ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Γιάννης Δ. Γαροφαλάκης
Καθηγητής

5η Έκδοση
Πάτρα, Σεπτέμβριος 2010

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

1.1 Η ΑΝΑΓΚΗ ΓΙΑ ΜΕΛΕΤΗ ΤΗΣ ΑΠΟΔΟΣΗΣ ΤΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Ο χρήστης, ο διαχειριστής και ο σχεδιαστής ενός *πληροφοριακού συστήματος,* ενδιαφέρονται κατ' αρχήν το σύστημα να *αποδίδει* σωστά στις λειτουργίες για τις οποίες προορίζεται.

> Με τον όρο **πληροφοριακό σύστημα** αναφερόμαστε σε οποιαδήποτε συλλογή τμημάτων υλικού ή λογισμικού. Μπορεί να αναφερόμαστε σε έναν Ηλεκτρονικό Υπολογιστή (Η/Υ), στο Λειτουργικό Σύστημά του, μία Βάση Δεδομένων, ένα σύστημα μετάδοσης και επικοινωνίας δεδομένων, έναν εξυπηρετητή παγκόσμιου ιστού (WWW server) κ.α.

Η δεύτερη μέριμνά τους θα είναι το πληροφοριακό σύστημα να έχει "ικανοποιητική" *απόδοση* με "λογικό" *κόστος*. Οι έννοιες *ικανοποιητική απόδοση* και *λογικό κόστος* είτε προσδιορίζονται με ακρίβεια (ποσοτικά), ή συνάγονται από τις λειτουργίες που περιμένουμε να εξυπηρετεί το σύστημα. Για παράδειγμα, ο χρήστης ενός επεξεργαστή κειμένου είναι φυσικό να περιμένει "στιγμιαία" απόκριση στις περισσότερες από τις εντολές που θα δώσει. Ο χρήστης ενός συστήματος πλοήγησης στον παγκόσμιο ιστό μπορεί να περιμένει περισσότερο, ίσως 10 δευτερόλεπτα αδιαμαρτύρητα για να "κατεβάσει" μια σελίδα, αλλά θα θεωρήσει το σύστημα απαράδεκτο αν περιμένει πολύ περισσότερο. Από την άλλη πλευρά, ένας προγραμματιστής που θέλει να μεταγλωττίσει ένα μεγάλο πρόγραμμα, θα μπορούσε να περιμένει και για αρκετά λεπτά χωρίς να ενοχληθεί.

Δυστυχώς η *απόδοση* και το *κόστος* δεν γίνονται συνήθως αντικείμενα προσοχής παρά μόνο στα τελευταία στάδια ανάπτυξης ενός πληροφοριακού συστήματος. Οι σχεδιαστές και οι υπεύθυνοι υλοποίησης ίσως να θεωρούν ότι θα βελτιώσουν τυχόν προβλήματα στην απόδοση του συστήματος, απλώς χρησιμοποιώντας μια ταχύτερη Κεντρική Μονάδα Επεξεργασίας (CPU), περισσότερη κεντρική μνήμη ή μεγαλύτερους δίσκους. Η ιστορία έχει δείξει πάντως, ότι αν καταλήξουμε σε απαράδεκτη απόδοση ενός συστήματος χρησιμοποιώντας λογικά και γενικά αποδεκτά μεγέθη υλικού (hardware), τότε οι μόνες επιλογές που έχουμε είναι είτε να απορρίψουμε εντελώς το σύστημα (γεγονός που συμβαίνει συχνά), ή να προχωρήσουμε σε επανασχεδιασμό και υλοποίηση από την αρχή, μέχρι το σύστημα να είναι αποδεκτό. Και οι δύο επιλογές κοστίζουν τελικά πολύ περισσότερο από μία σχεδίαση και διαδικασία υλοποίησης που παίρνει σαφώς υπόψη της το ζήτημα της απόδοσης.

Η *μελέτη* ή *ανάλυση της απόδοσης* είναι απαραίτητη σε όλα τα στάδια του κύκλου ζωής ενός πληροφοριακού συστήματος: στο σχεδιασμό, στην κατασκευή – υλοποίηση, στην πώληση/αγορά, στη χρήση του, στη ρύθμιση, στην αναβάθμισή του κ.λ.π. Μία μελέτη της απόδοσης είναι απαραίτητη όταν ο σχεδιαστής ενός πληροφοριακού συστήματος θέλει να συγκρίνει εναλλακτικές λύσεις υλοποίησης. Είναι απαραίτητη επίσης, όταν ο διευθυντής ενός υπολογιστικού κέντρου θέλει να συγκρίνει διαφορετικά συστήματα για να αποφασίσει ποιο είναι καλύτερο για ένα δεδομένο σύνολο εφαρμογών. Ακόμα και αν δεν υπάρχουν εναλλακτικές επιλογές, η ανάλυση της απόδοσης ενός δεδομένου συστήματος βοηθάει στον προσδιορισμό του βαθμού απόδοσής του, καθώς και σε τυχόν αποφάσεις για βελτίωσή του. Ατυχώς, υπάρχουν τόσο πολλές μορφές

εφαρμογών των πληροφοριακών συστημάτων, ώστε δεν είναι δυνατόν να έχουμε ένα απόλυτο και συγκεκριμένο *μέτρο απόδοσης,* ένα *περιβάλλον μέτρησης,* ή μία συγκεκριμένη *τεχνική* ανάλυσης της απόδοσης για όλες τις περιπτώσεις. Έτσι το πρώτο βήμα στην ανάλυση της απόδοσης πληροφοριακών συστημάτων είναι η επιλογή των κατάλληλων μέτρων απόδοσης, περιβαλλόντων μέτρησης και τεχνικών ανάλυσης.

1.2 ΜΕΘΟΔΟΛΟΓΙΑ ΑΝΑΛΥΣΗΣ ΤΗΣ ΑΠΟΔΟΣΗΣ

Τα περισσότερα προβλήματα απόδοσης πληροφοριακών συστημάτων είναι μοναδικά. Οι μετρικές, τα φορτία εργασίας και οι τεχνικές μελέτης που χρησιμοποιούνται για ένα πρόβλημα, γενικά δεν μπορούν να χρησιμοποιηθούν για άλλο πρόβλημα. Πάντως, υπάρχουν κάποια κοινά *βήματα* σε όλες τις μελέτες απόδοσης, τα οποία μπορούν να μας βοηθήσουν να αποφύγουμε τα λάθη που αναφέρονται στην προηγούμενη Παράγραφο. Τα βήματα αυτά είναι τα εξής:

1. *Καταγραφή Στόχων και Καθορισμός του Συστήματος*: Είναι το πρώτο βήμα σε οποιαδήποτε μελέτη απόδοσης συστήματος. Με δεδομένο το σύνολο υλικού και λογισμικού, ο ορισμός του συστήματος μπορεί να διαφέρει, ανάλογα με τους στόχους της μελέτης. Για παράδειγμα, αν έχουμε δύο CPU, ο στόχος μπορεί να είναι η εκτίμηση της επίδρασής τους στο χρόνο απόκρισης των αλληλεπιδραστικών (interactive) χρηστών. Στην περίπτωση αυτή το σύστημα θα αποτελείται από το αλληλεπιδραστικό υποσύστημα και τα συμπεράσματα της μελέτης μπορεί να εξαρτώνται σε μεγάλο βαθμό από τμήματα εξωτερικά της CPU. Αντίθετα, αν οι δύο CPU είναι όμοιες εκτός από τις Αριθμητικές – Λογικές Μονάδες τους (ALUs – Arithmetic Logic Units) και ο στόχος είναι να αποφασίσουμε ποια ALU θα επιλέξουμε, μπορεί να θεωρήσουμε ως μέρη του συστήματος μόνο εσωτερικά τμήματα της CPU. Η επιλογή των ορίων του συστήματος επηρεάζει τις *μετρικές απόδοσης*, καθώς και τα *φορτία εργασίας* που θα χρησιμοποιηθούν για τη σύγκριση των συστημάτων.

   Μία **μετρική απόδοσης** ορίζεται ως ένα κριτήριο ποσοτικοποίησης της απόδοσης ενός συστήματος.
   Φορτίο εργασίας ονομάζουμε τις αιτήσεις που κάνουν οι χρήστες στο σύστημα.

   ΠΙΘΑΝΑ ΛΑΘΗ:

   - *Έλλειψη Στόχων:* Είναι συνηθισμένο φαινόμενο η έλλειψη ή η ασάφεια των στόχων μιας μελέτης. Πολλές φορές ο ειδικός της ανάλυσης της απόδοσης ενός συστήματος που χρησιμοποιείται στην ομάδα σχεδιασμού, αρχίζει μια *αναλυτική προσέγγιση* ή μία *προσομοίωση*, θεωρώντας ότι το *μοντέλο* που κατασκευάζει μπορεί να απαντήσει οποιαδήποτε σχεδιαστική ερώτηση τεθεί. Όμως, οι έμπειροι αναλυτές γνωρίζουν ότι δεν έχει κανένα νόημα ένα μοντέλο γενικού σκοπού. Ένα μοντέλο κατασκευάζεται πάντα έχοντας υπόψη συγκεκριμένους στόχους. Πριν γραφεί η πρώτη γραμμή κώδικα *προσομοίωσης*, ή η πρώτη εξίσωση ενός *αναλυτικού μοντέλου*, ή πριν οργανωθεί ένα *πείραμα μέτρησης*, είναι σημαντικό για τον αναλυτή να έχει κατανοήσει το σύστημα και να έχει προσδιορίσει τα προβλήματα που πρέπει να λυθούν. Αυτό θα τον βοηθήσει να καθορίσει τις σωστές μετρικές, τα σωστά φορτία εργασίας και τις κατάλληλες *τεχνικές* ανάλυσης.

     Ο καθορισμός στόχων δεν είναι πάντα εύκολη διαδικασία. Για παράδειγμα, ένα πρόβλημα που είχε αρχικά διατυπωθεί ως η εύρεση ενός αλγορίθμου για αναμετάδοση πακέτων σε ένα δίκτυο, θα μπορούσε αργότερα να αναγνωρισθεί ως

ένα πρόβλημα προσαρμογής του φορτίου του δικτύου όταν έχουμε απώλειες πακέτων.

- *Προκατειλημμένοι Στόχοι:* Ένα άλλο κοινό λάθος είναι η προκατάληψη στον ορισμό των στόχων της μελέτης. Αν για παράδειγμα θέσουμε ως στόχο "να δείξουμε ότι το ΔΙΚΟ ΜΑΣ σύστημα είναι καλύτερο από το ΔΙΚΟ ΤΟΥΣ", τότε το πρόβλημα από εύρεση των κατάλληλων μετρικών και φορτίων εργασίας ώστε να συγκρίνουμε σωστά τα δύο συστήματα, μετασχηματίζεται σε πρόβλημα εύρεσης των μετρικών και φορτίων εργασίας που θα εμφανίσουν το δικό μας σύστημα καλύτερο. Ο ρόλος του αναλυτή της απόδοσης ενός συστήματος πρέπει να είναι απολύτως αμερόληπτος. Δεν πρέπει να έχει υπόψη του προσχηματισμένες απόψεις για την απόδοση του συστήματος, ενώ όλα τα συμπεράσματά του πρέπει να βασίζονται αποκλειστικά στα αποτελέσματα της ανάλυσής του.
- *Ακατάλληλο Επίπεδο Λεπτομέρειας:* Στη σύγκριση εναλλακτικών λύσεων που διαφέρουν λίγο μόνο από μια βασική λύση, ένα λεπτομερές μοντέλο που μπορεί να αναπαραστήσει τις διαφοροποιήσεις, θα είναι πιο χρήσιμο από ένα μοντέλο υψηλότερου επιπέδου. Αντίθετα, για τη σύγκριση εναλλακτικών λύσεων αρκετά διαφορετικών μεταξύ τους, ένα απλό μοντέλο υψηλού επιπέδου θα επιτρέψει την ανάλυση πολλών εναλλακτικών λύσεων γρήγορα και χωρίς μεγάλο κόστος.
- *Υπόθεση ότι δεν θα Υπάρξουν Αλλαγές στο Μέλλον:* Συχνά χρησιμοποιούμε ένα μοντέλο βασισμένο σε παρελθόντα φορτία εργασίας και τιμές παραμέτρων, για να μελετήσουμε την απόδοση ενός συστήματος στο μέλλον. Ο μελετητής θα πρέπει να προσδιορίζει το χρονικό πλαίσιο ισχύος του μοντέλου που κατασκευάζει.

2. *Καταγραφή των Υπηρεσιών και των Αποτελεσμάτων*: Κάθε σύστημα προσφέρει ένα σύνολο υπηρεσιών. Για παράδειγμα, ένα δίκτυο επιτρέπει στους χρήστες του να στέλνουν πακέτα σε συγκεκριμένους προορισμούς. Μία Βάση Δεδομένων (ΒΔ) αποκρίνεται σε αιτήσεις (queries). Ένας επεξεργαστής εκτελεί έναν αριθμό διαφορετικών εντολών. Το επόμενο βήμα στην ανάλυση ενός συστήματος είναι η καταγραφή των υπηρεσιών αυτών. Όταν ένας χρήστης ζητήσει κάποια από τις υπηρεσίες αυτές, μπορούν να εμφανιστούν διάφορα πιθανά αποτελέσματα. Άλλα από τα αποτελέσματα αυτά είναι επιθυμητά και άλλα όχι. Για παράδειγμα, μια ΒΔ μπορεί να αποκριθεί σε μια αίτηση σωστά, λανθασμένα (λόγω κακής ενημέρωσης), ή να μην απαντήσει καθόλου (λόγω deadlock ή άλλων προβλημάτων). Μία λίστα υπηρεσιών και πιθανών αποτελεσμάτων είναι χρήσιμη στην επιλογή των σωστών μετρικών και φορτίων εργασίας.

3. *Επιλογή των Μετρικών Απόδοσης*: Το επόμενο βήμα είναι η επιλογή των κριτηρίων για τη σύγκριση της απόδοσης του συστήματος. Γενικά οι μετρικές σχετίζονται με την *ταχύτητα*, την *ακρίβεια* και τη *διαθεσιμότητα* των υπηρεσιών. Για παράδειγμα, η απόδοση ενός δικτύου μετράται με την ταχύτητα (ρυθμαπόδοση και καθυστέρηση), την ακρίβεια (ποσοστό λαθών) και τη διαθεσιμότητα του δικτύου. Η απόδοση ενός επεξεργαστή μετράται με την ταχύτητα (χρόνος εκτέλεσης) διαφόρων εντολών.
   ΠΙΘΑΝΑ ΛΑΘΗ:
   - *Λάθος Μετρικές Απόδοσης:* Παραδείγματα συχνά χρησιμοποιούμενων μετρικών είναι η *ρυθμαπόδοση* (throughput) και ο *χρόνος απόκρισης* (response time). Η επιλογή των σωστών μετρικών απόδοσης εξαρτάται από τις υπηρεσίες που προσφέρει το σύστημα ή το υποσύστημα που μελετούμε. Για παράδειγμα, η απόδοση των Κεντρικών Μονάδων Επεξεργασίας (CPUs) συγκρίνεται χρησιμοποιώντας τις ρυθμαποδόσεις τους, οι οποίες συνήθως μετρώνται σε

εκατομμύρια εντολών ανά δευτερόλεπτο (MIPS - millions of instructions per second). Όμως η σύγκριση των MIPS δύο διαφορετικών αρχιτεκτονικών CPU, όπως οι RISC (Reduced Instruction Set Computers) και CISC (Complex Instruction Set Computers), δεν έχει νόημα αφού οι δύο αρχιτεκτονικές δεν έχουν τις ίδιες εντολές. Είναι εύκολο στην περίπτωση αυτή να αλλάξουν τα συμπεράσματα της μελέτης με κατάλληλο χειρισμό των μετρικών. Ακόμα, είναι πολλές φορές φυσικό να παραβλέπονται οι μετρικές που είναι δύσκολο να υπολογισθούν ή να μετρηθούν.

4. *Καταγραφή των Παραμέτρων του Συστήματος*:

Τα χαρακτηριστικά του συστήματος και του φορτίου εργασίας που επηρεάζουν την απόδοση του συστήματος ονομάζονται **παράμετροι.**

Το επόμενο βήμα είναι η καταγραφή όλων των παραμέτρων που επηρεάζουν την απόδοση. Η λίστα μπορεί να χωρισθεί σε παραμέτρους συστήματος και παραμέτρους φορτίου εργασίας. Οι *παράμετροι συστήματος* περιλαμβάνουν παραμέτρους υλικού και λογισμικού που γενικά δεν μεταβάλλονται με τις διάφορες εγκαταστάσεις του συστήματος. Οι *παράμετροι φορτίου εργασίας* χαρακτηρίζουν τις αιτήσεις των χρηστών, οι οποίες μεταβάλλονται από εγκατάσταση σε εγκατάσταση. Η λίστα των παραμέτρων μπορεί να μην είναι πλήρης στην αρχή. Δηλαδή, μετά από ένα πρώτο πέρασμα της μελέτης, μπορεί να ανακαλύψουμε ότι υπάρχουν επιπλέον παράμετροι που επηρεάζουν την απόδοση. Μπορούμε τότε να προσθέσουμε τις παραμέτρους αυτές στη λίστα.

ΠΙΘΑΝΑ ΛΑΘΗ:

- *Παράβλεψη Σημαντικών Παραμέτρων:* Είναι καλή πρακτική να κάνουμε στην αρχή της μελέτης, μια πλήρη λίστα των παραπάνω παραμέτρων. Για παράδειγμα, οι παράμετροι συστήματος μπορεί να περιλαμβάνουν το μέγεθος του quantum (για τη χρήση της CPU) ή το μέγεθος του συνόλου εργασίας (για τη χρήση της κεντρικής μνήμης). Οι παράμετροι φορτίου εργασίας μπορεί να περιλαμβάνουν τον αριθμό των χρηστών, τον τρόπο άφιξης των αιτήσεων χρήσης του συστήματος, την προτεραιότητα κ.α. Ο αναλυτής επιλέγει τιμές για τις παραπάνω παραμέτρους. Το τελικό αποτέλεσμα της μελέτης εξαρτάται σε μεγάλο βαθμό από τις επιλογές αυτές, καθώς επίσης και από το αν έχει διακρίνει και περιλάβει τις σημαντικές παραμέτρους.
- *Αγνόηση των Λαθών στα Δεδομένα Εισόδου:* Συχνά μια σημαντική παράμετρος δεν μπορεί να μετρηθεί και έτσι χρησιμοποιείται για την εκτίμησή της κάποια άλλη μετρήσιμη μεταβλητή. Για παράδειγμα, ας υποθέσουμε ότι σε μια συσκευή δικτύου τα πακέτα αποθηκεύονται σε μια συνδεμένη λίστα ενταμιευτών (buffers). Κάθε ενταμιευτής έχει μέγεθος 512 οκτάδες. Με δεδομένο τον αριθμό των ενταμιευτών που απαιτούνται για την αποθήκευση των πακέτων, δεν είναι δυνατό να υπολογισθούν με ακρίβεια ο αριθμός των πακέτων ή ο αριθμός των οκτάδων στα πακέτα. Τέτοιες περιπτώσεις εισάγουν επιπλέον αβεβαιότητες στα δεδομένα εισόδου και ο μελετητής πρέπει να προσαρμόσει κατάλληλα το επίπεδο εμπιστοσύνης των αποτελεσμάτων του.
- *Κακή Αντιμετώπιση των Εξαιρετικών Τιμών των Δεδομένων:* Οι πολύ μεγαλύτερες ή πολύ μικρότερες τιμές των δεδομένων εισόδου και εξόδου σε σχέση με την πλειοψηφία των τιμών, μπορούν να δημιουργήσουν πρόβλημα. Αν δεν οφείλονται σε πραγματικό φαινόμενο του συστήματος θα πρέπει να αγνοούνται. Στην αντίθετη περίπτωση θα πρέπει να περιλαμβάνονται στο μοντέλο αναπαράστασης του συστήματος.

- *Αγνόηση της Διακύμανσης:* Πολλές φορές βασίζουμε την ανάλυση και τα αποτελέσματά μας αποκλειστικά στη μέση τιμή των μεγεθών εισόδου και εξόδου, ιδιαίτερα όταν είναι δύσκολο να υπολογίσουμε τη διακύμανσή τους. Αν όμως η διακύμανση είναι μεγάλη, οι διαχειριστές του συστήματος μπορεί να οδηγηθούν σε λάθος συμπεράσματα και αποφάσεις. Για παράδειγμα, αποφάσεις βασισμένες στις ημερήσιες μέσες τιμές των απαιτήσεων χρηστών ενός υπολογιστικού κέντρου ίσως να μην είναι ιδιαίτερα χρήσιμες, αν υπάρχουν ώρες με απότομες αυξήσεις των απαιτήσεων.

5. *Επιλογή των Παραγόντων της Μελέτης*:

Οι παράμετροι που μεταβάλλονται κατά τη διάρκεια της μελέτης ονομάζονται **παράγοντες**, ενώ οι τιμές τους καλούνται *επίπεδα*.

Γενικά η λίστα των παραγόντων και τα πιθανά επίπεδά τους είναι μεγαλύτερα από ότι επιτρέπουν οι διαθέσιμοι πόροι μας για τη μελέτη. Με άλλα λόγια, συνήθως η λίστα μεγαλώνει συνέχεια μέχρι να γίνει φανερό ότι δεν υπάρχουν αρκετοί πόροι για τη μελέτη του προβλήματος. Είναι προτιμότερο να αρχίσουμε με μια μικρή λίστα παραγόντων και ένα μικρό αριθμό επιπέδων για κάθε παράγοντα και να επεκτείνουμε τη λίστα στην επόμενη φάση της μελέτης, αν το επιτρέπουν οι πόροι. Για παράδειγμα, μπορεί να αποφασίσουμε να έχουμε μόνο δύο παράγοντες: το μέγεθος του quantum και τον αριθμό των χρηστών. Για κάθε ένα από αυτά μπορεί να επιλέξουμε να έχουμε μόνο δύο επίπεδα: χαμηλό και ψηλό. Το μέγεθος του συνόλου εργασίας (working set) και ο τύπος του φορτίου εργασίας μπορούν να είναι σταθερά.

ΠΙΘΑΝΑ ΛΑΘΗ:

- *Αγνόηση Σημαντικών Παραγόντων:* Για παράδειγμα, από τις παραμέτρους του φορτίου εργασίας που αναφέρθηκαν παραπάνω, μόνο ο αριθμός των χρηστών μπορεί να θεωρηθεί παράγοντας. Οι παράμετροι δεν έχουν όλες την ίδια επίδραση στην απόδοση του συστήματος. Είναι σημαντικό να μπορούμε να προσδιορίσουμε τις παραμέτρους που μεταβαλλόμενες επηρεάζουν ουσιαστικά την απόδοση του συστήματος. Οι παράμετροι αυτές θα πρέπει να αντιμετωπίζονται ως παράγοντες. Για παράδειγμα, αν ο ρυθμός άφιξης πακέτων επηρεάζει το χρόνο απόκρισης μιας πύλης δικτύου περισσότερο από το μέγεθος των πακέτων, θα πρέπει να χρησιμοποιήσουμε αρκετούς διαφορετικούς ρυθμούς αφίξεων κατά τη μελέτη.

6. *Επιλογή της Τεχνικής Μελέτης*: Υπάρχουν τρεις γενικές κατηγορίες τεχνικών μελέτης της απόδοσης ενός συστήματος: *μέτρηση* (πείραμα με το πραγματικό σύστημα), *προσομοίωση* και *αναλυτικά* μοντέλα. Η επιλογή της κατάλληλης τεχνικής εξαρτάται από την πολυπλοκότητα του μοντέλου, τους διαθέσιμους πόρους για τη μελέτη και το επιθυμητό επίπεδο ακρίβειας. Με το ζήτημα αυτό θα ασχοληθούμε στην επόμενη παράγραφο του παρόντος Κεφαλαίου.

ΠΙΘΑΝΑ ΛΑΘΗ:

- *Λάθος Τεχνική Μελέτης:* Οι αναλυτές συχνά έχουν μια προτίμηση για κάποια τεχνική την οποία χρησιμοποιούν για κάθε πρόβλημα. Για παράδειγμα, ένας αναλυτής με ευχέρεια στις αναλυτικές τεχνικές της θεωρίας αναμονής (queueing theory) τείνει να μετασχηματίσει κάθε πρόβλημα ανάλυσης απόδοσης, σε πρόβλημα αναμονής, έστω και αν το σύστημα είναι πολύπλοκο ή είναι διαθέσιμο για απ' ευθείας μετρήσεις. Από την άλλη πλευρά, ένας αναλυτής καλός στον προγραμματισμό, τείνει να επιλύει οποιοδήποτε πρόβλημα με προσομοίωση. Η εμμονή σε μια συγκεκριμένη τεχνική μπορεί να οδηγήσει σε ένα μοντέλο εύκολο στη επίλυση, αλλά όχι απαραίτητα σε ένα μοντέλο που θα λύσει το πρόβλημα. Ακόμα, μπορούν να εισαχθούν στο μοντέλο φαινόμενα που δεν υπάρχουν στο

αρχικό σύστημα, ή να αγνοηθούν σημαντικά φαινόμενα που υπάρχουν στο πραγματικό σύστημα. Ένας αναλυτής πρέπει να έχει βασικές τουλάχιστον γνώσεις και από τις τρεις κατηγορίες τεχνικών.

7. *Επιλογή του Φορτίου Εργασίας*: Το φορτίο εργασίας αποτελείται από ένα σύνολο αιτήσεων για υπηρεσίες του συστήματος. Για παράδειγμα, το φορτίο εργασίας για τη σύγκριση συστημάτων Βάσεων Δεδομένων, μπορεί να αποτελείται από ένα σύνολο αιτήσεων. Ανάλογα με την τεχνική μελέτης που έχει επιλεγεί, το φορτίο εργασίας μπορεί να εκφραστεί με διαφορετικές μορφές. Στα αναλυτικά μοντέλα, το φορτίο εργασίας συνήθως εκφράζεται ως πιθανότητες διαφόρων αιτήσεων για υπηρεσίες. Στην προσομοίωση, θα μπορούσε να χρησιμοποιηθεί ένα *ίχνος* (trace) αιτήσεων σε ένα υπαρκτό σύστημα. Για τεχνική μέτρησης, το φορτίο εργασίας θα μπορούσε να αποτελείται από μικρά προγράμματα χρηστών (scripts), τα οποία θα εκτελεσθούν από το σύστημα. Σε κάθε περίπτωση είναι βασικό να είναι αντιπροσωπευτικό της χρήσης του συστήματος στην πραγματική ζωή. Για να δημιουργήσουμε αντιπροσωπευτικά φορτία εργασίας, πρέπει να μετρήσουμε και να χαρακτηρίσουμε το φορτίο εργασίας σε υπαρκτά συστήματα.
   ΠΙΘΑΝΑ ΛΑΘΗ:
   - *Μη Αντιπροσωπευτικά Φορτία Εργασίας:* Το φορτίο εργασίας (workload) που χρησιμοποιείται για τη σύγκριση δύο συστημάτων, πρέπει να είναι αντιπροσωπευτικό της πραγματικής χρήσης των συστημάτων. Για παράδειγμα, αν τα πακέτα σε ένα δίκτυο είναι μίξη δύο μεγεθών – μικρά και μεγάλα – το φορτίο εργασίας για τη σύγκριση δύο δικτύων πρέπει να περιλαμβάνει και τα δύο μεγέθη.

8. *Σχεδιασμός των Πειραμάτων*: Έχοντας επιλέξει τους παράγοντες και τα επίπεδά τους, χρειάζεται να αποφασίσουμε μια ακολουθία πειραμάτων η οποία θα προσφέρει τη μέγιστη πληροφορία με την ελάχιστη προσπάθεια. Πρακτικά αυτό σημαίνει να εκτελέσουμε το πείραμα σε δύο φάσεις. Στην πρώτη φάση ο αριθμός των παραγόντων μπορεί να είναι μεγάλος, αλλά ο αριθμός των επιπέδων μικρός. Στόχος είναι να προσδιορίσουμε τη σχετική επιρροή των διαφόρων παραγόντων. Στη δεύτερη φάση μειώνουμε τον αριθμό των παραγόντων και αυξάνουμε τον αριθμό των επιπέδων για τους παράγοντες που έχουν σημαντική επίπτωση στην απόδοση του συστήματος.
   ΠΙΘΑΝΑ ΛΑΘΗ:
   - *Ακατάλληλος Σχεδιασμός των Πειραμάτων:* Ο σχεδιασμός των πειραμάτων έχει να κάνει με την επιλογή του αριθμού πειραμάτων μέτρησης ή προσομοίωσης που θα εκτελεσθούν, καθώς και με τις τιμές των παραμέτρων που θα χρησιμοποιηθούν σε κάθε πείραμα. Η κατάλληλη επιλογή των τιμών αυτών μπορεί να παράγει περισσότερη πληροφορία από τον ίδιο αριθμό πειραμάτων, ενώ η μη σωστή επιλογή μπορεί να οδηγήσει σε χάσιμο χρόνου και πόρων κατά τη μελέτη.
   - *Έλλειψη Ανάλυσης Ευαισθησίας:* Συχνά οι μελετητές αγνοούν το γεγονός ότι τα αποτελέσματα μπορούν να είναι ευαίσθητα στο φορτίο εργασίας και στις παραμέτρους του συστήματος. Χωρίς ανάλυση ευαισθησίας δεν μπορούμε να είμαστε σίγουροι αν τα συμπεράσματά μας θα άλλαζαν σε ένα λίγο διαφορετικό περιβάλλον. Ακόμα, θα είναι δύσκολο να βρούμε τη σχετική σημασία των διαφόρων παραμέτρων.

9. *Ανάλυση και Εξήγηση των Αποτελεσμάτων*: Είναι σημαντικό να έχουμε υπόψη ότι τα αποτελέσματα των μετρήσεων και των προσομοιώσεων είναι τυχαίες ποσότητες αφού θα είχαν διαφορετικές τιμές σε κάθε επανάληψη του πειράματος. Στη σύγκριση δύο εναλλακτικών λύσεων, είναι απαραίτητο να παίρνουμε υπόψη τη μεταβλητότητα των αποτελεσμάτων. Η απλή σύγκριση των μέσων τιμών μπορεί να οδηγήσει σε ανακριβή

αποτελέσματα. Επίσης, η εξήγηση των αποτελεσμάτων μιας μελέτης είναι σημαντική, αν δεν ξεχνάμε ότι η μελέτη παράγει μόνο αποτελέσματα και όχι συμπεράσματα. Δεν είναι σπάνιο δύο αναλυτές με το ίδιο σύνολο αποτελεσμάτων, να καταλήγουν σε διαφορετικά συμπεράσματα.

ΠΙΘΑΝΑ ΛΑΘΗ:

- *Έλλειψη Ανάλυσης Δεδομένων:* Αν ο μελετητής δεν έχει γνώσεις και εμπειρία στην ανάλυση δεδομένων, είναι πιθανό μία μελέτη απόδοσης με μετρήσεις να καταλήξει στη συλλογή τεράστιων ποσοτήτων δεδομένων, αλλά χωρίς καμία ανάλυση ή έστω χρήσιμη μορφή παρουσίασης.
- *Παράλειψη Υποθέσεων και Περιορισμών:* Οι αρχικές υποθέσεις και περιορισμοί της μελέτης συχνά παραλείπονται από την τελική αναφορά. Αυτό μπορεί να οδηγήσει το χρήστη στην εφαρμογή της μελέτης σε ένα υποθετικό περιβάλλον λειτουργίας του συστήματος στο οποίο οι υποθέσεις δεν ισχύουν.

10. *Παρουσίαση των Αποτελεσμάτων*: Το τελικό βήμα σε όλες τις μελέτες απόδοσης είναι η μετάδοση των αποτελεσμάτων στα υπόλοιπα μέλη της ομάδας λήψης αποφάσεων. Είναι σημαντικό τα αποτελέσματα να παρουσιάζονται με εύκολα κατανοητό τρόπο. Αυτό συνήθως απαιτεί την κατάλληλη χρήση γραφικών.

    ΠΙΘΑΝΑ ΛΑΘΗ:

    - *Ακατάλληλη παρουσίαση των Αποτελεσμάτων:* Ο τελικός στόχος κάθε μελέτης απόδοσης πληροφοριακού συστήματος, είναι η υποβοήθηση της λήψης αποφάσεων. Μία μελέτη με αποτελέσματα που δεν μπορούν να γίνουν κατανοητά από τους υπεύθυνους για τη λήψη αποφάσεων, είναι το ίδιο αποτυχημένη με μια μελέτη που δεν έχει παράγει χρήσιμα αποτελέσματα. Η μεταβίβαση των αποτελεσμάτων της μελέτης στον υπεύθυνο για τη λήψη αποφάσεων, είναι υπευθυνότητα του μελετητή. Για το σκοπό αυτό είναι σημαντική η σωστή χρήση λέξεων, εικόνων και γραφικών.

    Στο σημείο αυτό πολλές φορές, η γνώση που έχουμε αποκτήσει από τη μελέτη, μπορεί να αναγκάσει το μελετητή να γυρίσει πίσω κάποια βήματα και να ξανασκεφθεί μερικές από τις αποφάσεις του. Για παράδειγμα, μπορεί να χρειαστεί να ορίσει ξανά τα όρια του συστήματος ή να περιλάβει και άλλες παραμέτρους και μετρικές που δεν είχαν ληφθεί υπόψη. Δηλαδή, μια πλήρης μελέτη συνήθως αποτελείται από μερικούς κύκλους των βημάτων που αναφέραμε και όχι μόνο από ένα μόνο ακολουθιακό πέρασμά τους.

Τα παραπάνω βήματα μιας μελέτης απόδοσης πληροφοριακών συστημάτων φαίνονται περιληπτικά στον Πίνακα 1.1.

1. Δηλώστε τους στόχους της μελέτης και καθορίστε τα όρια του συστήματος.
2. Καταγράψετε τις υπηρεσίες του συστήματος και τα πιθανά αποτελέσματα.
3. Επιλέξτε τις μετρικές απόδοσης.
4. Καταγράψτε τις παραμέτρους του συστήματος και του φορτίου εργασίας.
5. Επιλέξτε τους παράγοντες και τα επίπεδά τους.
6. Επιλέξτε την τεχνική μελέτης.
7. Επιλέξτε το φορτίο εργασίας.
8. Σχεδιάστε τα πειράματα.
9. Αναλύστε και εξηγήστε τα δεδομένα.
10. Παρουσιάστε τα αποτελέσματα. Επαναλάβετε τα βήματα αν χρειάζεται.

Πίνακας 1.1. Τα Βήματα μιας Μελέτης Απόδοσης

Στον Πίνακα 1.2 φαίνεται μια λίστα ερωτήσεων που πρέπει να απαντώνται θετικά, αν θέλουμε να αποφεύγουμε τα συνήθη λάθη που αναφέρθηκαν κατά την παρουσίαση των βημάτων μιας μελέτης απόδοσης:

1. Έχουν ορισθεί σωστά και με σαφήνεια το σύστημα και οι στόχοι της μελέτης;
2. Έχουν καθορισθεί οι στόχοι της μελέτης χωρίς προκατάληψη;
3. Είναι κατάλληλο το επίπεδο λεπτομέρειας του μοντέλου;
4. Έχουν περιληφθεί στο μοντέλο οι μελλοντικές αλλαγές στο σύστημα και το φορτίο εργασίας;
5. Είναι οι μετρικές απόδοσης κατάλληλες για το πρόβλημα;
6. Είναι πλήρης η λίστα των παραμέτρων που επηρεάζουν την απόδοση;
7. Τυχόν λάθη στα δεδομένα εισόδου θα δημιουργούσαν μικρή διαφοροποίηση στα αποτελέσματα;
8. Έχουν αντιμετωπισθεί σωστά οι πολύ μικρές ή οι πολύ μεγάλες τιμές των δεδομένων εισόδου και εξόδου;
9. Έχει ληφθεί υπόψη η διακύμανση των τιμών των παραμέτρων εισόδου;
10. Έχουν επιλεγεί ως παράγοντες οι σωστές παράμετροι;
11. Είναι κατάλληλη η τεχνική μελέτης της απόδοσης;
12. Είναι το φορτίο εργασίας το σωστό για το πρόβλημα;
13. Είναι ο σχεδιασμός των πειραμάτων αποτελεσματικός αναφορικά με το χρόνο και τα αποτελέσματα;
14. Είναι η ανάλυση σωστή στατιστικά;
15. Έχει γίνει ανάλυση ευαισθησίας;
16. Έχει αναλυθεί η διακύμανση των αποτελεσμάτων;
17. Είναι εύκολο να εξηγηθούν τα αποτελέσματα;
18. Έχουν τεκμηριωθεί με σαφήνεια οι υποθέσεις και οι περιορισμοί της μελέτης;
19. Είναι το στυλ της παρουσίασης των αποτελεσμάτων κατάλληλο για το ακροατήριο για το οποίο προορίζεται;

Πίνακας 1.2. Λίστα Ερωτήσεων για Αποφυγή Λαθών κατά τη Μελέτη Απόδοσης

Στο Παράδειγμα 1.1 που ακολουθεί παρουσιάζεται ένα παράδειγμα συστηματικής προσέγγισης μιας μελέτης απόδοσης:

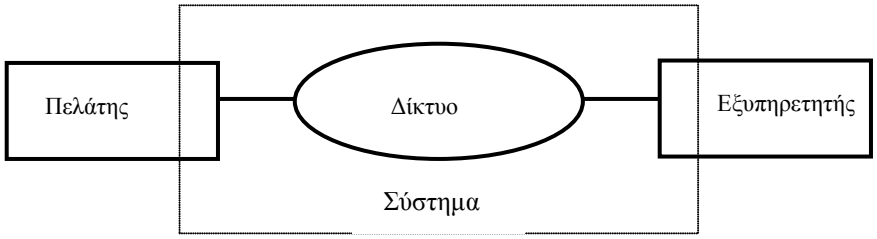
Παράδειγμα 1.1

Συστηματική προσέγγιση μιας μελέτης απόδοσης.

Έστω το πρόβλημα της σύγκρισης απομακρυσμένων σωληνώσεων (remote pipes) με απομακρυσμένες κλήσεις διαδικασιών (remote procedure calls). Σε μια κλήση διαδικασίας το πρόγραμμα που καλεί μπλοκάρεται, ο έλεγχος μεταφέρεται στην καλούμενη διαδικασία μαζί με μερικές παραμέτρους και όταν ολοκληρώνεται η διαδικασία, τα αποτελέσματα μαζί με τον έλεγχο επιστρέφουν στο πρόγραμμα που έχει καλέσει τη διαδικασία. Η *απομακρυσμένη κλήση διαδικασίας* είναι μια επέκταση της αρχής αυτής σε ένα κατανεμημένο σύστημα υπολογιστών: Ένα πρόγραμμα που εκτελείται σε ένα υπολογιστικό σύστημα, καλεί μία διαδικασία που βρίσκεται σε ένα άλλο υπολογιστικό σύστημα. Το πρόγραμμα που καλεί περιμένει μέχρι να ολοκληρωθεί η διαδικασία και δέχεται πίσω τα αποτελέσματα. Η *απομακρυσμένη σωλήνωση* είναι αντικείμενο παρόμοιο με την απομακρυσμένη διαδικασία, με τη διαφορά πως όταν κληθεί, δεν μπλοκάρεται το πρόγραμμα που την έχει καλέσει. Η εκτέλεση της σωλήνωσης γίνεται ταυτόχρονα με τη συνεχιζόμενη εκτέλεση του προγράμματος που την κάλεσε και τα αποτελέσματά της, αν υπάρχουν, μεταβιβάζονται αργότερα (ασύγχρονα).

Το παρακάτω πλάνο μελέτης βασίζεται στα βήματα του Πίνακα 1.1:

1. *Ορισμός Συστήματος*: Στόχος της μελέτης είναι η σύγκριση της απόδοσης εφαρμογών που χρησιμοποιούν απομακρυσμένες σωληνώσεις, με την απόδοση εφαρμογών που χρησιμοποιούν απομακρυσμένες κλήσεις διαδικασιών. Το βασικό στοιχείο που θα μελετηθεί είναι το κανάλι. Ένα *κανάλι* είναι μια ιδεατή σύνδεση επικοινωνίας και αντιστοιχεί είτε σε μια διαδικασία ή σε μια σωλήνωση. Το σύστημα αποτελείται από δύο υπολογιστές συνδεδεμένους μέσω ενός δικτύου, όπως φαίνεται στο Σχήμα 1.1.



Σχήμα 1.1. Ορισμός του Συστήματος για το Παράδειγμα 1.1.

Οι αιτήσεις στέλνονται μέσω του καναλιού από τον υπολογιστή – πελάτη στον υπολογιστή – εξυπηρετητή. Μόνο τα τμήματα του πελάτη και του εξυπηρετητή που προσφέρουν υπηρεσίες καναλιού, θεωρούνται μέρη του συστήματος. Η μελέτη θα γίνει έτσι ώστε η επίδραση των υπολοίπων τμημάτων να ελαχιστοποιείται.

2. *Υπηρεσίες*: Οι υπηρεσίες που προσφέρονται από το σύστημα είναι οι δύο τύποι κλήσεων μέσω καναλιού: Απομακρυσμένη κλήση διαδικασίας και απομακρυσμένη σωλήνωση. Οι πόροι που χρησιμοποιούνται από τις κλήσεις εξαρτώνται από τον αριθμό των παραμέτρων που μεταβιβάζονται και από τις ενέργειες που χρειάζεται να γίνουν στις παραμέτρους αυτές. Στο παράδειγμά μας, ως εφαρμογή επιλέγεται η μεταφορά δεδομένων, ενώ οι κλήσεις θα ταξινομούνται απλώς ως μικρές ή μεγάλες, ανάλογα με το μέγεθος των δεδομένων που θα μεταφερθεί στον απομακρυσμένο υπολογιστή. Με άλλα λόγια, το σύστημα προσφέρει μόνο δύο υπηρεσίες: Μεταφορά λίγων δεδομένων και μεταφορά πολλών δεδομένων.
3. *Μετρικές*: Λόγω των περιορισμών σε πόρους, δεν θα μελετήσουμε τα λάθη και τις αποτυχίες της επικοινωνίας. Δηλαδή, θα περιοριστούμε στη σωστή λειτουργία του συστήματος. Για κάθε υπηρεσία θα συγκρίνουμε: το ρυθμό υλοποίησης της υπηρεσίας, το χρόνο που χρειάζεται η υπηρεσία και τους πόρους που καταναλώνει. Πόροι είναι: ο τοπικός υπολογιστής (πελάτης), ο απομακρυσμένος υπολογιστής (εξυπηρετητής) και η δικτυακή σύνδεση. Έτσι οδηγούμαστε στις παρακάτω μετρικές απόδοσης:
   (I) Χρόνος που έχει περάσει, ανά κλήση.
   (II) Μέγιστος ρυθμός κλήσεων ανά χρονική μονάδα, ή ισοδύναμα ο χρόνος που απαιτείται για την ολοκλήρωση ενός μπλοκ από *n* διαδοχικές κλήσεις.
   (III) Χρόνος τοπικής CPU ανά κλήση.
   (IV) Χρόνος απομακρυσμένης CPU ανά κλήση.
   (V) Αριθμός bytes που στέλνονται ανά κλήση.
4. *Παράμετροι*: Οι παράμετροι του συστήματος που επηρεάζουν την απόδοση μιας δεδομένης εφαρμογής και το μέγεθος των δεδομένων είναι οι εξής:
   (I) Ταχύτητα της τοπικής CPU.
   (II) Ταχύτητα της απομακρυσμένης CPU.
   (III) Ταχύτητα του δικτύου.
   (IV) Επιβάρυνση του Λειτουργικού Συστήματος για επικοινωνία με τα κανάλια.
   (V) Επιβάρυνση του Λειτουργικού Συστήματος για επικοινωνία με το δίκτυο.

(VI) Αξιοπιστία του δικτύου σε σχέση με τον αριθμό των αναμεταδόσεων που χρειάζονται.

Οι παράμετροι του φορτίου εργασίας που επηρεάζουν την απόδοση είναι:

(I) Χρόνος ανάμεσα σε διαδοχικές κλήσεις.
(II) Αριθμός και μεγέθη των παραμέτρων κλήσης.
(III) Αριθμός και μεγέθη των αποτελεσμάτων των κλήσεων.
(IV) Τύπος του καναλιού.
(V) Άλλα φορτία στην τοπική και την απομακρυσμένη CPU.
(VI) Άλλα φορτία στο δίκτυο.

5. *Παράγοντες*: Οι βασικοί παράγοντες που επιλέγονται για τη μελέτη του παραδείγματός μας είναι:

(I) *Τύπος του καναλιού*. Θα συγκριθούν δύο τύποι καναλιού: απομακρυσμένες σωληνώσεις και απομακρυσμένες κλήσεις διαδικασιών.
(II) *Ταχύτητα του δικτύου*. Θα εξαρτάται από δύο πιθανές τοποθεσίες του απομακρυσμένου υπολογιστή: σε μικρή απόσταση (μέσα στον οργανισμό) και σε μεγάλη απόσταση (οπουδήποτε αλλού στη χώρα).
(III) *Μεγέθη των παραμέτρων κλήσης που θα μεταδοθούν*. Θα χρησιμοποιηθούν δύο επίπεδα: Μικρό και μεγάλο.
(IV) *Αριθμός n διαδοχικών κλήσεων*. Θα χρησιμοποιηθούν 11 διαφορετικές τιμές του *n*: 1, 2, 4, 8, 16, 32,…, 512,1024.

Οι παράγοντες επιλέχθηκαν με βάση τη διαθεσιμότητα πόρων και τα ενδιαφέροντα των τελικών αποδεκτών της μελέτης. Οι τιμές των υπολοίπων παραμέτρων θα είναι σταθερές. Συνεπώς, τα αποτελέσματά μας θα ισχύουν μόνο για τους τύπους των CPU και Λειτουργικών Συστημάτων που χρησιμοποιούνται στη μελέτη. Η αναμετάδοση λόγω λαθών στο δίκτυο θα αγνοείται. Τα πειράματα θα εκτελούνται όταν θα υπάρχει πολύ μικρό άλλο φορτίο στους υπολογιστές και στο δίκτυο.

6. *Τεχνική Μελέτης*: Αφού έχουν ήδη υλοποιηθεί πρωτότυπα και των δύο τύπων καναλιών, θα χρησιμοποιήσουμε τεχνικές μέτρησης του πραγματικού συστήματος. Επίσης, θα χρησιμοποιήσουμε αναλυτικές τεχνικές για να επιβεβαιώσουμε τη συνέπεια των τιμών που θα μετρήσουμε για διαφορετικές παραμέτρους.

7. *Φορτίο Εργασίας*: Το φορτίο εργασίας θα αποτελείται από ένα *συνθετικό πρόγραμμα* που θα παράγει τους καθορισμένους τύπους κλήσεων. Το πρόγραμμα θα *παρακολουθεί* επίσης τους πόρους που θα καταναλώνονται και θα *καταγράφει* τα αποτελέσματα που θα μετρώνται. Για τον προσδιορισμό των πόρων που θα καταναλώνονται για την παρακολούθηση και την καταγραφή, θα χρησιμοποιούνται "ψεύτικες" αιτήσεις για κλήσεις, οι οποίες δεν θα προβλέπουν πραγματική δουλειά από το απομακρυσμένο σύστημα.

8. *Σχεδιασμός πειραμάτων*: Θα χρησιμοποιηθεί ένας πλήρης παραγοντικός σχεδιασμός πειραμάτων με $2^3 \times 11 = 88$ πειράματα.

9. *Ανάλυση δεδομένων*: Για την ποσοτικοποίηση της επίδρασης των τριών πρώτων παραγόντων, θα χρησιμοποιηθεί ανάλυση διακύμανσης, ενώ για την ποσοτικοποίηση της επίδρασης του αριθμού *n* διαδοχικών κλήσεων, θα χρησιμοποιηθεί η μέθοδος της παλινδρόμησης.

10. *Παρουσίαση αποτελεσμάτων*: Τα τελικά αποτελέσματα θα εμφανιστούν σε γραφική παράσταση ως συνάρτηση του μεγέθους *n*.

1.3 Η ΕΠΙΛΟΓΗ ΤΕΧΝΙΚΗΣ ΜΕΛΕΤΗΣ

Η επιλογή της κατάλληλης τεχνικής, καθώς και των μετρικών της απόδοσης του συστήματος, είναι τα σημαντικότερα βήματα σε κάθε μελέτη. Υπάρχουν πολλές πλευρές του θέματος που πρέπει να ληφθούν υπόψη. Στην παράγραφο αυτή θα παρουσιάσουμε τα σημαντικότερα ζητήματα που σχετίζονται με την επιλογή της τεχνικής μελέτης.

Οι τρεις τεχνικές για μελέτη της απόδοσης ενός πληροφοριακού συστήματος είναι τα αναλυτικά μοντέλα, η προσομοίωση και η μέτρηση. Υπάρχει ένας αριθμός κριτηρίων που μπορούν να μας βοηθήσουν να επιλέξουμε τεχνική. Στον Πίνακα 1.3 φαίνονται τα κριτήρια αυτά με φθίνουσα σειρά σημασίας.

Κριτήριο	Ανάλυση	Προσομοίωση	Μέτρηση
1. Στάδιο Συστήματος	Οποιοδήποτε	Οποιοδήποτε	Υλοποιημένο
2. Χρόνος	Μικρός	Μέτριος	Μεταβλητός
3. Εργαλεία	Αναλυτές	Γλώσσες Προγρ/σμού	Συσκευές Μέτρησης
4. Ακρίβεια	Χαμηλή	Ικανοποιητική	Μεταβλητή
5. Μεταβλητότητα	Εύκολη	Ικανοποιητική	Δύσκολη
6. Κόστος	Μικρό	Μέτριο	Υψηλό
7. Εμπιστοσύνη	Μικρή	Μέτρια	Υψηλή

Πίνακας 1.3. Κριτήρια για την Επιλογή Τεχνικής Μελέτης ενός Συστήματος

Το σημαντικότερο κριτήριο στην επιλογή της τεχνικής μελέτης, είναι το *στάδιο του κύκλου ζωής* στο οποίο βρίσκεται το σύστημα. Τεχνικές μέτρησης είναι δυνατές μόνο αν υπάρχει υλοποιημένο το σύστημα, έστω και ως πρωτότυπο. Αν ενδιαφερόμαστε για μια καινούργια ιδέα που βρίσκεται στο στάδιο του σχεδιασμού, μπορούμε να χρησιμοποιήσουμε μόνο ανάλυση ή προσομοίωση.

Το επόμενο κριτήριο είναι ο *διαθέσιμος χρόνος* για τη μελέτη. Τις περισσότερες φορές τα αποτελέσματα πρέπει να είναι διαθέσιμα "χθες". Στις περιπτώσεις αυτές τα αναλυτικά μοντέλα είναι η μόνη λύση, έστω και με υιοθέτηση προσεγγιστικών υποθέσεων όσον αφορά τις παραμέτρους και τα όρια του συστήματος. Οι τεχνικές μέτρησης καταναλώνουν γενικά λιγότερο χρόνο από την προσομοίωση, αλλά είναι ευάλωτες σε λάθη, τεχνικά προβλήματα κ.λ.π., ώστε να θεωρούνται οι περισσότερο μεταβλητές σε χρονική διάρκεια.

Επόμενο κριτήριο είναι η *διαθεσιμότητα εργαλείων* για την εκτέλεση της μελέτης. Οι τρεις κατηγορίες τεχνικών μελέτης απαιτούν αντίστοιχα: δυνατότητες ανάλυσης, γλώσσες προγραμματισμού και συσκευές μέτρησης. Πολλές φορές η έλλειψη γνώσεων σε γλώσσες προσομοίωσης και τις ανάλογες τεχνικές από έναν μελετητή με ικανότητες στην ανάλυση, αποθαρρύνουν τους μελετητές από τις τεχνικές αυτές. Συχνό είναι και το αντίστροφο φαινόμενο.

Το επίπεδο της *απαιτούμενης ακρίβειας* είναι ένα άλλο κριτήριο. Γενικά οι αναλυτικές τεχνικές περιλαμβάνουν πολλές απλοποιήσεις και προσεγγιστικές υποθέσεις, έτσι ώστε να είναι σχεδόν αδύνατο να πάρουμε ένα ικανοποιητικό επίπεδο ακρίβειας στα αποτελέσματά μας. Η προσομοίωση μπορεί να περιλάβει στα μοντέλα της περισσότερες λεπτομέρειες και συνεπώς να δώσει αποτελέσματα πιο κοντά στην πραγματικότητα. Οι τεχνικές μέτρησης φαίνεται ίσως ότι μπορούν να προσεγγίσουν περισσότερο την πραγματικότητα, αλλά υπάρχουν λόγοι για τους οποίους πολλές φορές φθάνουμε σε αντίθετο αποτέλεσμα: όρια συστήματος και φορτίο εργασίας που συνδέονται μόνο με το συγκεκριμένο πείραμα, τιμές παραμέτρων που δεν αντιπροσωπεύουν το εύρος των τιμών που θα αντιμετωπίσει το σύστημα κ.α.

Στόχος οποιασδήποτε μελέτης απόδοσης είναι είτε η σύγκριση *διαφορετικών εναλλακτικών επιλογών*, ή η εύρεση της *βέλτιστης τιμής* μιας ή περισσοτέρων παραμέτρων. Οι αναλυτικές τεχνικές γενικά δίνουν καλύτερη εξήγηση των επιπτώσεων των διαφόρων παραμέτρων και των αλληλεπιδράσεών τους. Στην προσομοίωση είναι δυνατόν να επιλεγεί η βέλτιστη τιμή μιας παραμέτρου, αλλά συχνά δεν είναι σαφής ο ποιοτικός προσδιορισμός της αλληλεπίδρασης πολλών παραμέτρων. Η μέτρηση είναι η λιγότερο αποδοτική τεχνική κάτω από αυτό το πρίσμα. Δεν είναι εύκολο να συμπεράνουμε αν η βελτιωμένη απόδοση είναι αποτέλεσμα κάποιας τυχαίας αλλαγής στο περιβάλλον ή οφείλεται σε συγκεκριμένες τιμές των παραμέτρων.

Το *κόστος* της μελέτης είναι επίσης μια σημαντική παράμετρος. Οι τεχνικές μέτρησης απαιτούν τη χρήση κάποιου πραγματικού συστήματος, συσκευές μέτρησης και χρόνο για να υλοποιηθούν. Είναι οι πιο ακριβές τεχνικές. Το κόστος και η δυνατότητα αλλαγής των παραμέτρων και των ορίων του συστήματος, είναι οι συνηθέστεροι λόγοι επιλογής της προσομοίωσης για τη μελέτη μεγάλων και ακριβών συστημάτων. Οι αναλυτικές τεχνικές απαιτούν μόνο χαρτί, μολύβι και χρόνο του μελετητή και κατά συνέπεια είναι οι φθηνότερες τεχνικές.

Η δυνατότητα να γίνουν τα αποτελέσματα *αποδεκτά με εμπιστοσύνη* από τους υπεύθυνους για τη λήψη αποφάσεων, είναι ένα ακόμα σημαντικό κριτήριο. Είναι ευκολότερο να πείσουμε τους άλλους κάνοντας πραγματικές μετρήσεις. Πολλοί είναι επιφυλακτικοί με τις αναλυτικές τεχνικές, απλά διότι δεν τις κατανοούν. Έτσι, οι ερευνητές που αναπτύσσουν καινούργιες αναλυτικές τεχνικές, χρησιμοποιούν προσομοίωση ή μετρήσεις για να επιβεβαιώσουν τα αποτελέσματά τους.

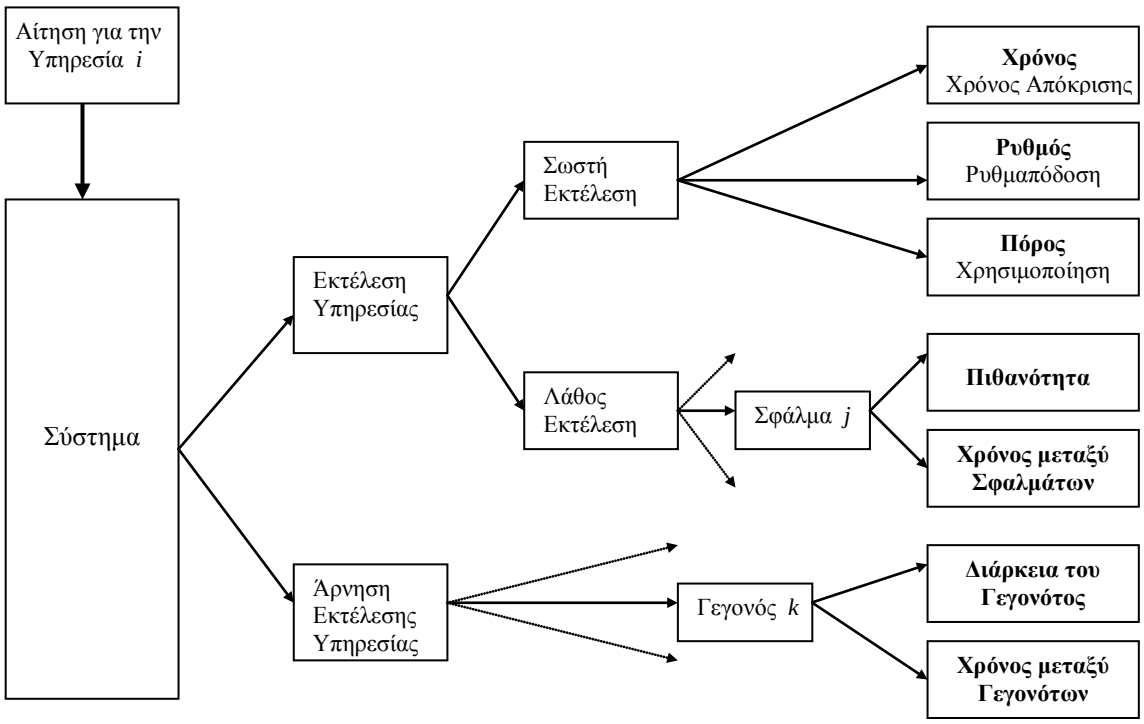
Πολλές φορές είναι αναγκαίο να χρησιμοποιήσουμε δύο ή περισσότερες τεχνικές *ακολουθιακά*. Για παράδειγμα, μια απλή αναλυτική τεχνική μπορεί να χρησιμοποιηθεί για τον προσδιορισμό του κατάλληλου εύρους τιμών μιας παραμέτρου συστήματος και μια προσομοίωση να χρησιμοποιηθεί μετά για τη μελέτη της απόδοσης στο εύρος αυτό. Η διαδικασία αυτή μπορεί να μειώσει σημαντικά τον αριθμό των εκτελέσεων του προσομοιωτή και έτσι να έχουμε πιο παραγωγική χρήση των διαθέσιμων πόρων για τη μελέτη. Επίσης μπορούμε να χρησιμοποιήσουμε δύο ή περισσότερες τεχνικές για την επιβεβαίωση των αποτελεσμάτων της μιας με τα αποτελέσματα της άλλης.

1.4 Η ΕΠΙΛΟΓΗ ΜΕΤΡΙΚΩΝ ΑΠΟΔΟΣΗΣ

Σε κάθε μελέτη απόδοσης, πρέπει να επιλεγούν κριτήρια ή μετρικές απόδοσης. Ένας τρόπος είναι να καταγράψουμε τις υπηρεσίες που προσφέρει το σύστημα. Κάθε αίτηση για υπηρεσία από το σύστημα μπορεί να έχει διάφορα αποτελέσματα. Τα πιθανά αυτά αποτελέσματα κατατάσσονται σε τρεις κατηγορίες όπως φαίνεται στο Σχήμα 1.2. Το σύστημα μπορεί να εκτελέσει την υπηρεσία σωστά, λάθος, ή να αρνηθεί να την εκτελέσει. Για παράδειγμα, μία πύλη (gateway) σε ένα δίκτυο προσφέρει την υπηρεσία προώθησης πακέτων στους καθορισμένους προορισμούς σε ετερογενή δίκτυα. Μπορεί να προωθήσει σωστά ένα πακέτο, μπορεί να το προωθήσει σε λάθος προορισμό, ή μπορεί να είναι εκτός λειτουργίας (η πύλη), οπότε δεν θα το προωθήσει καθόλου. Ανάλογα, μια Βάση Δεδομένων προσφέρει την υπηρεσία απάντησης σε ερωτήσεις (queries). Μπορεί να απαντήσει σωστά σε μια ερώτηση, να απαντήσει λάθος, ή να είναι εκτός λειτουργίας και να μην απαντήσει καθόλου.

Αν το σύστημα εκτελέσει σωστά την υπηρεσία, η απόδοσή του θα μετρηθεί με: το χρόνο για την εκτέλεση της αίτησης για την υπηρεσία (π.χ. *χρόνος απόκρισης* – response time), το ρυθμό με τον οποίο η υπηρεσία προσφέρεται (π.χ. *ρυθμαπόδοση* – throughput)

και τους πόρους που θα καταναλώσει για την προσφορά της υπηρεσίας (π.χ. *χρησιμοποίηση* – utilization).



Σχήμα 1.2. Πιθανά Αποτελέσματα Αίτησης για Υπηρεσία και Μετρικές

Για παράδειγμα, στη μελέτη μιας πύλης ενός δικτύου, ο χρόνος απόκρισης μετρά το χρονικό διάστημα μεταξύ της άφιξης ενός πακέτου και της επιτυχούς άφιξης στον προορισμό του. Η ρυθμαπόδοση θα αναφέρεται στον αριθμό πακέτων που προωθούνται στη μονάδα του χρόνου από την πύλη.

Η χρησιμοποίηση θα είναι ένα μέτρο για το ποσοστό του χρόνου που οι πόροι της πύλης είναι απασχολημένοι με το δεδομένο επίπεδο φορτίου εργασίας. Ο πόρος με τη μεγαλύτερη χρησιμοποίηση ονομάζεται *σημείο συμφόρησης* (bottleneck). Η βελτιστοποίηση της απόδοσης του πόρου αυτού θα μας δώσει τα μεγαλύτερα ωφελήματα. Κατά συνέπεια, ο προσδιορισμός της χρησιμοποίησης των διαφόρων πόρων μέσα στο σύστημα είναι ένα σημαντικό μέρος της μελέτης απόδοσής του.

Αν το σύστημα εκτελέσει λάθος την υπηρεσία, έχει εμφανιστεί ένα *σφάλμα*. Είναι χρήσιμο να κατατάσσουμε τα σφάλματα σε κατηγορίες και να προσδιορίζουμε τις πιθανότητες εμφάνισης κάθε κατηγορίας. Στο παράδειγμα της πύλης, ίσως πρέπει να βρούμε τις πιθανότητες εμφάνισης σφάλματος ενός bit, σφάλματος δύο bits κ.λ.π. Επίσης την πιθανότητα μερικής παράδοσης του πακέτου στον προορισμό του.

Αν το σύστημα δεν εκτελέσει την υπηρεσία, τότε λέμε ότι είναι *εκτός λειτουργίας* ή *μη διαθέσιμο*. Και στην περίπτωση αυτή είναι χρήσιμο να κατηγοριοποιήσουμε τα γεγονότα που οδηγούν στη μη διαθεσιμότητα και να προσδιορίσουμε τις πιθανότητες εμφάνισης των γεγονότων. Για παράδειγμα, η πύλη μπορεί να είναι εκτός λειτουργίας το 0.01% του χρόνου λόγω προβλήματος του επεξεργαστή της και 0.03% του χρόνου λόγω προβλημάτων του λογισμικού.

Οι μετρικές που έχουν σχέση με τα τρία πιθανά αποτελέσματα, προσδιορίζονται ως μετρικές *ταχύτητας*, *αξιοπιστίας* και *διαθεσιμότητας* αντίστοιχα. Είναι φανερό ότι για κάθε υπηρεσία που προσφέρεται από το σύστημα, μπορούμε να έχουμε έναν αριθμό από μετρικές ταχύτητας, αξιοπιστίας και διαθεσιμότητας.

Στα πληροφοριακά συστήματα που χρησιμοποιούνται από πολλούς χρήστες, πρέπει να παίρνουμε υπόψη δύο τύπους μετρικών απόδοσης: ατομικές και καθολικές. Οι *ατομικές* μετρικές αναφέρονται στην απόδοση και τα ενδιαφέροντα κάθε χρήστη ξεχωριστά, ενώ οι *καθολικές* μετρικές αντανακλούν τα "ενδιαφέροντα" του συστήματος. Η χρησιμοποίηση, η αξιοπιστία και η διαθεσιμότητα των πόρων του συστήματος είναι καθολικές μετρικές, ενώ ο χρόνος απόκρισης και η ρυθμαπόδοση μπορούν να μετρηθούν τόσο ως ατομικές, όσο και ως καθολικές μετρικές. Σε αρκετές περιπτώσεις η απόφαση για τη βελτιστοποίηση μιας ατομικής μετρικής μπορεί να επηρεάσει αρνητικά μια καθολική μετρική και το αντίθετο. Για παράδειγμα, σε ένα δίκτυο όπου διατηρείται σταθερός αριθμό πακέτων, η αύξηση του αριθμού των πακέτων από μια πηγή μπορεί να αυξήσει τη ρυθμαπόδοση της πηγής αυτής, θα μειώσει όμως τη ρυθμαπόδοση κάποιας άλλης πηγής και πιθανότατα θα επηρεάσει την καθολική απόδοση του δικτύου.

1.5 ΣΥΝΗΘΕΙΣ ΜΕΤΡΙΚΕΣ ΑΠΟΔΟΣΗΣ

Παρακάτω παρουσιάζονται μερικές από τις πιο συχνά χρησιμοποιούμενες μετρικές απόδοσης. Οι ορισμοί που δίνονται δεν είναι οι μοναδικοί. Πολλές φορές υπάρχουν διαφοροποιήσεις ανάλογα με τη συγκεκριμένη εφαρμογή.

- *Χρόνος Απόκρισης (response time)*: Ορίζεται ως το χρονικό διάστημα μεταξύ της αίτησης του χρήστη προς το σύστημα και της απόκρισης του συστήματος. Ο ορισμός αυτός, πάντως, είναι απλουστευτικός, αφού συνήθως οι αιτήσεις και οι αποκρίσεις δεν είναι ακαριαίες. Οι χρήστες χρειάζονται χρόνο για να συντάξουν και να αποστείλουν την αίτησή τους και το σύστημα καταναλώνει επίσης χρόνο για να παραδώσει την απάντησή του. Στις περιπτώσεις αυτές ο χρόνος απόκρισης προσδιορίζεται με δύο πιθανούς τρόπους: Είτε ως το χρονικό διάστημα μεταξύ του τέλους αποστολής της αίτησης και της αρχής της απόκρισης του συστήματος, ή ως το χρονικό διάστημα μεταξύ του τέλους αποστολής της αίτησης και του τέλους της απόκρισης του συστήματος. Ο δεύτερος προσδιορισμός είναι προτιμότερος όταν η απάντηση του συστήματος καταναλώνει πολύ χρόνο. Έτσι, σε ένα αλληλεπιδραστικό σύστημα ο χρόνος απόκρισης θα είναι το χρονικό διάστημα μεταξύ της πληκτρολόγησης του τελευταίου "return" ή "enter" από το χρήστη και της εμφάνισης του τελευταίου χαρακτήρα της απάντησης του συστήματος στην οθόνη του τερματικού.
- *Χρόνος Ολοκλήρωσης (turnaround time)*: Αναφέρεται συνήθως σε batch συστήματα και ορίζεται ως ο χρόνος μεταξύ της υποβολής της batch εργασίας και της ολοκλήρωσης της απάντησης του συστήματος. Στο χρόνο ολοκλήρωσης περιλαμβάνεται και ο χρόνος ανάγνωσης της αίτησης από το σύστημα.
- *Χρόνος Αντίδρασης (reaction time)*: Ορίζεται ως ο χρόνος μεταξύ της υποβολής της αίτησης και της αρχής εκτέλεσής της από το σύστημα. Για να μετρηθεί ο χρόνος αντίδρασης, πρέπει να υπάρχει η δυνατότητα παρακολούθησης και καταγραφής των εσωτερικών λειτουργιών του συστήματος, αφού είναι πιθανό η έναρξη εκτέλεσης να μη γίνεται αντιληπτή στο χρήστη ή το διαχειριστή του συστήματος.
- *Συντελεστής Επιμήκυνσης (stretch factor)*: Γενικά ο χρόνος απόκρισης αυξάνεται όσο αυξάνεται το φορτίο στο σύστημα. Ο λόγος του χρόνου απόκρισης με ένα συγκεκριμένο φορτίο προς το χρόνο απόκρισης με το ελάχιστο φορτίο ονομάζεται συντελεστής επιμήκυνσης. Για παράδειγμα, σε ένα αλληλεπιδραστικό σύστημα ο συντελεστής επιμήκυνσης μπορεί να ορισθεί ως ο λόγος του χρόνου απόκρισης σε συνθήκες πολυπρογραμματισμού, προς το χρόνο απόκρισης με ένα χρήστη.

- *Ρυθμαπόδοση (throughput)*: Ορίζεται ως ο ρυθμός (αιτήσεις ανά μονάδα χρόνου) με τον οποίο οι αιτήσεις μπορούν να ικανοποιούνται από το σύστημα. Σε batch συστήματα η ρυθμαπόδοση μετράται σε εργασίες ανά δευτερόλεπτο. Σε αλληλεπιδραστικά συστήματα μετράται σε αιτήσεις ανά δευτερόλεπτο. Όταν αναφερόμαστε σε κεντρικές μονάδες επεξεργασίας (CPU), η ρυθμαπόδοση μετράται σε *MIPS* (εκατομμύρια εντολών ανά δευτερόλεπτο), ή *MFLOPS* (εκατομμύρια πράξεων κινητής υποδιαστολής ανά δευτερόλεπτο). Όταν αναφερόμαστε σε δίκτυα, η ρυθμαπόδοση μετράται σε *bps* (bits ανά δευτερόλεπτο) ή *pps* (πακέτα ανά δευτερόλεπτο). Για συστήματα επεξεργασίας διεργασιών (transaction processing systems), μετράμε σε TPS (διεργασίες ανά δευτερόλεπτο).
- *Χωρητικότητα (capacity)*: Η ρυθμαπόδοση ενός συστήματος γενικά αυξάνεται όταν αυξάνεται αρχικά το φορτίο. Μετά από ένα κατώφλι, σταματάει να αυξάνεται και σε πολλές περιπτώσεις αρχίζει να μειώνεται. Η μέγιστη δυνατή ρυθμαπόδοση υπό ιδανικές συνθήκες φορτίου ονομάζεται χωρητικότητα. Σε δίκτυα υπολογιστών η χωρητικότητα ονομάζεται επίσης *εύρος ζώνης* (bandwidth) και μετράται συνήθως σε bps.
- *Χρησιμοποίηση (utilization)*: Αναφέρεται σε έναν πόρο του συστήματος και είναι το ποσοστό του χρόνου που είναι απασχολημένος στην εξυπηρέτηση των αιτήσεων. Τα χρονικά διαστήματα στα οποία ο πόρος δεν χρησιμοποιείται, ονομάζονται *άεργα* διαστήματα. Οι διαχειριστές συστημάτων ενδιαφέρονται συνήθως για την εξισορρόπηση του φορτίου εργασίας έτσι ώστε η χρησιμοποίηση των πόρων να μη διαφέρει πολύ μεταξύ τους.
- *Αξιοπιστία (reliability)*: Συνήθως μετράται με την πιθανότητα λάθους ή με το μέσο χρόνο μεταξύ λαθών.
- *Διαθεσιμότητα (availability)*: Ορίζεται ως το ποσοστό του χρόνου στο οποίο το σύστημα είναι διαθέσιμο για την εξυπηρέτηση των αιτήσεων των χρηστών. Το σύστημα μπορεί να μην είναι διαθέσιμο στους χρήστες που μας ενδιαφέρουν, είτε λόγω βλάβης, ή επειδή ασχολείται με την εξυπηρέτηση διεργασιών του ίδιου του συστήματος, ή ακόμα επειδή διαμοιράζεται σε περισσότερες ομάδες χρηστών.
- *Λόγος Κόστους/Απόδοσης (cost/performance ratio)*: Το μέγεθος αυτό χρησιμοποιείται ως μετρική για τη σύγκριση δύο ή περισσοτέρων συστημάτων όταν πρόκειται να γίνει προμήθεια ενός συστήματος. Το κόστος περιλαμβάνει την αγορά υλικού και λογισμικού, την εγκατάσταση και τη συντήρηση του συστήματος για ορισμένα χρόνια (συνήθως 3 – 5). Η απόδοση μετράται με τη ρυθμαπόδοση υπό σταθερό χρόνο απόκρισης. Για παράδειγμα, δύο συστήματα επεξεργασίας διεργασιών μπορούν να συγκριθούν με το μέγεθος “ευρώ ανά TPS”.

ΚΕΦΑΛΑΙΟ 2
ΦΟΡΤΙΟ ΕΡΓΑΣΙΑΣ, ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΕΙΡΑΜΑΤΑ

2.1 ΦΟΡΤΙΟ ΕΡΓΑΣΙΑΣ

Στην παράγραφο 1.2 ορίσαμε το *φορτίο εργασίας* (workload) ως «τις αιτήσεις που κάνουν οι χρήστες στο σύστημα». Ένας πιο ακριβής ορισμός είναι:

Ο όρος **φορτίο εργασίας** αναφέρεται στο σύνολο των απαιτήσεων επεξεργασίας που επιβάλλονται σε ένα σύστημα από την κοινότητα των χρηστών του, στη διάρκεια κάποιας χρονικής περιόδου.

Για παράδειγμα, ας θεωρήσουμε ένα χρήστη που εργάζεται σε περιβάλλον UNIX στο σταθμό εργασίας του, ο οποίος είναι συνδεδεμένος σε ένα δίκτυο υπολογιστών. Στη διάρκεια του χρόνου παρατήρησης ο χρήστης έστειλε ένα μήνυμα αποθηκευμένο στο αρχείο `text`, σε 12 συναδέλφους του συνδεδεμένους επίσης στο δίκτυο. Δηλαδή, ο χρήστης εκτέλεσε 12 φορές την εντολή `% mail "συνάδελφος" < text`. Το φορτίο εργασίας που επιβλήθηκε στο σταθμό εργασίας και στο δίκτυο αποτελείται από την ακολουθία εντολών `mail` που εκτέλεσε ο συγκεκριμένος χρήστης.

Τα φορτία εργασίας αντιπροσωπεύουν την «είσοδο» (input) στο σύστημα που θέλουμε να μελετήσουμε αυτόνομα, ή να συγκρίνουμε με άλλα αντίστοιχα συστήματα. Κατά συνέπεια η σωστή επιλογή και περιγραφή τους είναι σημαντικό βήμα στη μελέτη απόδοσης.

Χρησιμοποιούμε τον όρο *φορτίο εργασίας μελέτης* όταν αναφερόμαστε σε φορτίο εργασίας που χρησιμοποιείται σε μελέτη απόδοσης πληροφοριακού συστήματος. Ένα φορτίο εργασίας μελέτης μπορεί να είναι πραγματικό ή συνθετικό. *Πραγματικό φορτίο εργασίας* είναι αυτό που παρατηρείται σε ένα σύστημα κάτω από κανονικές συνθήκες λειτουργίας, όπως στο προηγούμενο παράδειγμα. Γενικά, τα πραγματικά φορτία εργασίας δεν μπορούν να επαναληφθούν εύκολα κάτω από τις ίδιες συνθήκες και έτσι συνήθως δεν είναι κατάλληλα για φορτία εργασίας μελέτης. Αντίθετα, το *συνθετικό φορτίο εργασίας* είναι μια τεχνητή κατασκευή ειδική για μελέτες, με χαρακτηριστικά όμως παρόμοια με αυτά πραγματικού φορτίου εργασίας το οποίο μπορεί να χρησιμοποιείται όσες φορές χρειαστεί με ελεγχόμενο τρόπο.

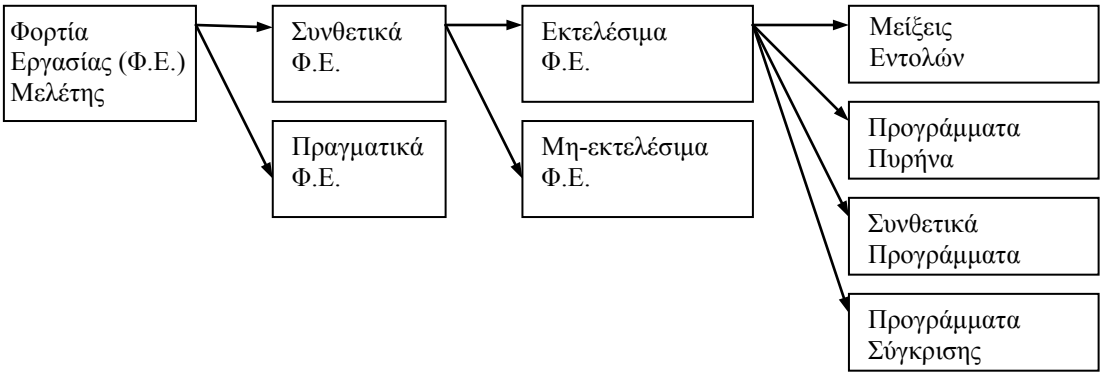
Οι *λόγοι* για τους οποίους προτιμούνται τα συνθετικά φορτία εργασίας είναι:

- Αποτελούν αξιόπιστα μοντέλα των πραγματικών φορτίων εργασίας
- Δεν χρειάζονται συλλογές πραγματικών δεδομένων, οι οποίες συνήθως είναι τεράστιες και είναι πιθανό να περιέχουν ευαίσθητα στοιχεία
- Μπορούν να τροποποιηθούν εύκολα χωρίς να επηρεάζεται η λειτουργία του συστήματος
- Μπορούν εύκολα να μεταφερθούν σε διάφορα συστήματα, κυρίως λόγω του μικρού μεγέθους τους
- Πολλές φορές έχουν ενσωματωμένους μηχανισμούς μετρήσεων

Τα συνθετικά φορτία εργασίας διακρίνονται σε εκτελέσιμα και μη-εκτελέσιμα. *Εκτελέσιμο* φορτίο εργασίας είναι αυτό που μπορεί να χρησιμοποιηθεί και σε πειράματα με το πραγματικό σύστημα. *Μη-εκτελέσιμο* είναι το φορτίο εργασίας που περιγράφεται με τέτοιο τρόπο ώστε δεν είναι δυνατή η χρησιμοποίησή του σε πειράματα απευθείας στο

πραγματικό σύστημα. Τα τελευταία συνήθως περιγράφονται (χαρακτηρίζονται) από τη μέση τιμή, τη διασπορά ή την κατανομή πιθανότητας των παραμέτρων τους.

Τα *εκτελέσιμα συνθετικά φορτία εργασίας* που έχουν αναπτυχθεί και χρησιμοποιούνται ακόμα, κατατάσσονται στους παρακάτω τύπους: μείξεις εντολών (instruction mixes), προγράμματα πυρήνα (kernels), συνθετικά προγράμματα (synthetic programs), προγράμματα σύγκρισης (benchmarks). Στο Σχήμα 2.1 φαίνεται η ιεραρχία των διαφόρων κατηγοριών φορτίων εργασίας:



Σχήμα 2.1 Η ιεραρχία των κατηγοριών φορτίων εργασίας μελέτης.

2.1.1 Τύποι Εκτελέσιμων Συνθετικών Φορτίων Εργασίας

1. Μείξεις εντολών.

Μία **μείξη εντολών** είναι ένα σύνολο διαφόρων εντολών χαμηλού επιπέδου του συστήματος, μαζί με τις σχετικές συχνότητες εμφάνισης των εντολών αυτών.

Οι μείξεις εντολών εφαρμόστηκαν αρχικά για τη μελέτη των επεξεργαστών ηλεκτρονικών υπολογιστών και οι εντολές που χρησιμοποιήθηκαν ήταν βασικά εντολές επεξεργασίας δεδομένων.

Ο προσδιορισμός και η χρήση των μείξεων εντολών γίνεται ως εξής: Με την εκτέλεση ενός συνόλου αντιπροσωπευτικών προγραμμάτων σε κάθε ένα από τα συστήματα που θα μελετήσουμε και την καταγραφή του αριθμού εμφανίσεων των επιμέρους τύπων εντολών όπως οι Add/Subtract, Multiply, Divide, Load, Store, Shift κ.λ.π., μπορούμε να υπολογίσουμε τη σχετική συχνότητα f_i εμφάνισης του τύπου εντολών i. Γνωρίζοντας τους χρόνους εκτέλεσης t_i κάθε τύπου εντολών i για το συγκεκριμένο σύστημα, μπορούμε να υπολογίσουμε το μέσο χρόνο εκτέλεσης της μείξης εντολών για κάθε σύστημα, $\tau = \sum_i f_i t_i$. Συγκρίνοντας τις τιμές του τ για τα διαφορετικά συστήματα μπορούμε να πάρουμε αποφάσεις σχετικά με τη σχετική απόδοσή τους. Αν διαλέξουμε ως μονάδα χρόνου το microsecond (μsec), τότε η ποσότητα $1/\tau$ αντιπροσωπεύει την ταχύτητα της CPU σε MIPS.

Μία γνωστή μείξη εντολών είναι η *Gibson*, η οποία κατασκευάστηκε το 1959 από τον Jack C. Gibson και φαίνεται στον Πίνακα 2.1. Οι σχετικές συχνότητες εμφάνισης των εντολών υπολογίστηκαν μετά από μετρήσεις σε συστήματα IBM 704 και 650.

Οι μείξεις εντολών έχουν αρκετά *μειονεκτήματα*: Οι σημερινοί υπολογιστές υποστηρίζουν περισσότερους πολύπλοκους τύπους εντολών που δεν αντιπροσωπεύονται στις γνωστές μείξεις. Ο χρόνος εκτέλεσης μιας εντολής δεν είναι πλέον σταθερός. Οι μείξεις δεν παίρνουν υπόψη τους νέες δυνατότητες όπως η ιδεατή διευθυνσιοδότηση κ.α.

Οι μείξεις εντολών είναι χρήσιμες ουσιαστικά μόνο όταν πρόκειται να συγκρίνουμε την ταχύτητα επεξεργαστών σε συστήματα με παρόμοιες αρχιτεκτονικές. Για τη μελέτη της απόδοσης ενός πλήρους συστήματος, συνήθως πρέπει να συνυπολογίσουμε τη συμπεριφορά και των άλλων στοιχείων που το συνθέτουν. Το μέγεθος τ και τα MIPS του επεξεργαστή έχουν νόημα μόνο αν ο επεξεργαστής είναι το βασικό σημείο συμφόρησης (bottleneck) του συστήματος.

i	Τύπος Εντολής	f_i
1	Load and Store	31.2
2	Fixed-Point Add and Subtract	6.1
3	Compares	3.8
4	Branches	16.6
5	Floating Add and Subtract	6.9
6	Floating Multiply	3.8
7	Floating Divide	1.5
8	Fixed-Point Multiply	0.6
9	Fixed-Point Divide	0.2
10	Shifting	4.4
11	Logical, And, Or	1.6
12	Εντολές που δεν χρησιμοποιούν καταχωρητές	5.3
13	Indexing	18.0
		100.0

Πίνακας 2.1. Η Μείξη Εντολών Gibson

2. Προγράμματα πυρήνα (kernels).

Τα προγράμματα πυρήνα είναι γενίκευση των μείξεων εντολών. Όταν έγινε φανερό ότι οι εντολές από μόνες τους δεν μπορούσαν να περιλάβουν τις όλο και πιο πολύπλοκες λειτουργίες των επεξεργαστών, οι ερευνητές προσπάθησαν να ορίσουν σύνολα εντολών που συγκροτούν μια ανώτερου επιπέδου *λειτουργία* ή υπηρεσία του επεξεργαστή.

Πρόγραμμα πυρήνα ονομάζουμε ένα σύνολο εντολών ή μια λειτουργία που εμφανίζεται συχνά στο βασικό πυρήνα εκτέλεσης μιας ή περισσοτέρων κατηγοριών εφαρμογών του συστήματος.

Ουσιαστικά, ένα πρόγραμμα πυρήνα είναι ένα μικρό τμήμα προγράμματος που αντιπροσωπεύει τον εσωτερικό βρόχο μιας συχνά χρησιμοποιούμενης εφαρμογής. Για παράδειγμα, σε επιστημονικές εφαρμογές, μπορούν να επιλεγούν ως προγράμματα πυρήνα, μία ρουτίνα αντιστροφής πίνακα και ένα πρόγραμμα επίλυσης διαφορικών εξισώσεων. Ένα πρόγραμμα πυρήνα είναι χρήσιμο στα αρχικά στάδια σχεδιασμού τμημάτων ενός συστήματος, όπως η CPU, για την αξιολόγηση διαφορετικών συνόλων εντολών και διαφορετικών αρχιτεκτονικών ιδεών. Και αυτό διότι τα προγράμματα πυρήνα περιέχουν σημαντική πληροφορία, όπως η ακολουθία των εντολών και η σχετική θέση των εντολών διακλάδωσης. Επίσης επιτρέπουν την αξιολόγηση τμημάτων λογισμικού. Συγκεκριμένα, η ποιότητα ενός μεταφραστή μπορεί να αξιολογηθεί με την υλοποίηση και μετάφραση ενός προγράμματος πυρήνα και τον έλεγχο του αντικειμενικού κώδικα.

Σε αντίθεση με τις μείξεις εντολών, τα προγράμματα πυρήνα δεν χρησιμοποιούν πραγματικές μετρήσεις του συστήματος. Έγιναν δημοφιλή λόγω της χρήσης τους από ερευνητές που ήθελαν να συγκρίνουν τις αποδόσεις διαφόρων αρχιτεκτονικών επεξεργαστών.

Τα προγράμματα πυρήνα διατηρούν τα περισσότερα *μειονεκτήματα* των μείξεων εντολών. Το βασικό μειονέκτημά τους, πάντως, είναι ότι συνήθως δεν κάνουν χρήση των συσκευών εισόδου/εξόδου του συστήματος και έτσι δεν μπορούν να μετρήσουν τη συνολική απόδοση του συστήματος.

3. Συνθετικά προγράμματα (synthetic programs).

Τα προγράμματα πυρήνα δεν παίρνουν υπόψη τους τις υπηρεσίες που προσφέρει το λειτουργικό σύστημα, καθώς και τις συσκευές εισόδου/εξόδου. Η ανάγκη να συνυπολογισθούν οι παραπάνω παράγοντες, οδήγησε τους ερευνητές στη δημιουργία απλών προγραμμάτων δοκιμής τα οποία εκτελούμενα, κάνουν κλήσεις των υπηρεσιών του λειτουργικού συστήματος και αιτήσεις για χρήση των συσκευών εισόδου/εξόδου. Τα προγράμματα αυτά ονομάζονται *συνθετικά προγράμματα* και είναι γραμμένα συνήθως σε γλώσσα υψηλού επιπέδου όπως FORTRAN, Pascal, C.

Τα **συνθετικά προγράμματα** είναι «τεχνητά» προγράμματα που περιέχουν όλα τα σημαντικά συστατικά των προγραμμάτων, για να δοκιμάσουν διάφορα τμήματα των συστημάτων.

Περιλαμβάνουν μεταβλητές παραμέτρους ώστε να μπορούν να περιγράψουν ένα ευρύ φάσμα χαρακτηριστικών των προγραμμάτων. Οι παράμετροι αυτοί είναι, για παράδειγμα, το ύψος των απαιτήσεων σε CPU, οι απαιτήσεις σε χώρο αποθήκευσης, ο αριθμός προσπελάσεων στο δίσκο. Η δομή μιας συνθετικής εργασίας μπορεί να είναι αρκετά απλή: το υπολογιστικό τμήμα της, μπορεί να είναι ένας βρόχος που απλώς προσθέτει ακεραίους.

Το βασικό *πλεονέκτημα* των συνθετικών προγραμμάτων είναι το γεγονός ότι κατασκευάζονται και τροποποιούνται εύκολα και είναι ανεξάρτητα από συγκεκριμένο υλικό συστήματος. Επίσης, δεν χρησιμοποιούν πραγματικά δεδομένα, ενώ τις περισσότερες φορές διαθέτουν ενσωματωμένους μηχανισμούς μετρήσεων.

Τα *μειονεκτήματα* των συνθετικών προγραμμάτων είναι: επειδή είναι μικρά δεν μπορούν να κάνουν αντιπροσωπευτικές αναφορές στη μνήμη και στους δίσκους του συστήματος. Οι μηχανισμοί σφαλμάτων σελίδας και cache του δίσκου μπορεί να μην παίρνονται ικανοποιητικά υπόψη. Δεν είναι κατάλληλα για πολυχρηστικά περιβάλλοντα διότι μπορούν να δημιουργήσουν συγχρονισμούς με απρόβλεπτες συνέπειες στην απόδοση του συστήματος.

4. Προγράμματα σύγκρισης (benchmarks).

Τα **προγράμματα σύγκρισης** είναι πλήρη προγράμματα γραμμένα σε γλώσσα υψηλού επιπέδου, τα οποία θεωρούνται αντιπροσωπευτικά κλάσεων προγραμμάτων εφαρμογών.

Τα προγράμματα σύγκρισης μπορούν να περιλαμβάνουν, για παράδειγμα, τμήματα ταξινόμησης και ενημέρωσης αρχείων. Με την εκτέλεση διαφόρων προγραμμάτων σύγκρισης, μπορεί να αξιολογηθεί η απόδοση του συνολικού συστήματος σε ρεαλιστικές συνθήκες. Είναι κατασκευασμένα έτσι ώστε να χρησιμοποιούν και να αξιολογούν τους περισσότερους πόρους του συστήματος, όπως επεξεργαστές, συσκευές εισόδου/εξόδου, δίκτυα, βάσεις δεδομένων κ.λ.π. Τα τεστ με προγράμματα σύγκρισης χρησιμοποιούνται συχνά στη διαδικασία επιλογής προμηθευτή για ένα σύστημα: ένα πρόγραμμα σύγκρισης εκτελείται σε κάθε προτεινόμενο σύστημα. Η σύγκριση του συνολικού χρόνου εκτέλεσης, είναι μία καλή ένδειξη της σχετικής απόδοσης των συστημάτων.

Γνωστά προγράμματα σύγκρισης είναι τα Whetstone, Linpack, Dhrystone, TPC benchmarks, SPEC. Ακολουθεί σύντομη περιγραφή των δύο τελευταίων που έχουν και μεγαλύτερη βαρύτητα.

TPC benchmarks

Τα TPC benchmarks προδιαγράφηκαν από το TPC (Transaction Processing Performance Council), ένα συμβούλιο στο οποίο συμμετέχουν κατασκευαστές, χρήστες,

σύμβουλοι συστημάτων επεξεργασίας διεργασιών (transaction processing systems) και δημιουργήθηκε το 1988. Τα συστήματα επεξεργασίας διεργασιών αποκτούν όλο και μεγαλύτερη σημασία με εφαρμογές στον τραπεζικό τομέα, στις κρατήσεις θέσεων κ.λ.π. Τα TPC benchmarks εκτιμούν την απόδοση των επεξεργαστών, του υποσυστήματος εισόδου/εξόδου, του λειτουργικού συστήματος και του υποσυστήματος διαχείρισης βάσεων δεδομένων.

Το βασικό TPC benchmark είναι το *TPC-C*, του οποίου η τρέχουσα έκδοση είναι η 5.11. Είναι πιο πολύπλοκο από προηγούμενα OLTP (On-Line Transaction Processing) benchmarks, λόγω των ποικίλλων τύπων διεργασιών, της πιο σύνθετης βάσης δεδομένων και της συνολικής δομής εκτέλεσής του. Το *TPC-C* εκτιμά την απόδοση διαφορετικών τύπων διεργασιών μέσω της εκτέλεσης μιας μείξης πέντε συγχρονισμένων διεργασιών διαφορετικού τύπου και πολυπλοκότητας.

Η βασική *μετρική απόδοσης* που επιστρέφεται από τα TPC benchmarks είναι η ρυθμαπόδοση του συστήματος μετρημένη σε TPS (διεργασίες ανά δευτερόλεπτο), υπό τον περιορισμό ότι το 90% των διεργασιών έχουν χρόνο απόκρισης το πολύ 2 sec.

Σήμερα το TPC έχει αναπτύξει νέα προγράμματα σύγκρισης για σύγχρονες εφαρμογές, όπως το *TPC-App* το οποίο είναι πρόγραμμα σύγκρισης για Application Servers και web services και το *TPC-H* για περιβάλλοντα υποστήριξης αποφάσεων (decision support). Περισσότερες πληροφορίες μπορείτε να βρείτε στη σελίδα του TPC: http://www.tpc.org

SPEC

Το SPEC (Standard Performance Evaluation Corporation) είναι ένας οργανισμός που δημιουργήθηκε, επίσης, το 1988 από εξέχουσες βιομηχανίες παραγωγής υπολογιστών, με σκοπό να αναπτύξει ένα πρότυπο σύνολο προγραμμάτων σύγκρισης. Η έκδοση 1.0 του SPEC benchmark suite (1990), αποτελείται από τα παρακάτω 10 προγράμματα σύγκρισης που αντιπροσωπεύουν διάφορες επιστημονικές και τεχνολογικές εφαρμογές:

1. *GCC*: Μετρά το χρόνο που χρειάζεται ο μεταγλωττιστής GNU C για να μετατρέψει 19 προεπεξεργασμένα αρχεία πηγαίου κώδικα σε γλώσσα assembly. Το πρόγραμμα αυτό σύγκρισης, είναι αντιπροσωπευτικό ενός περιβάλλοντος τεχνολογίας λογισμικού εκτιμώντας την αποδοτικότητα της μεταγλώττισης σε ένα σύστημα.
2. *Espresso*: Είναι ένα εργαλείο αυτοματισμού ηλεκτρονικού σχεδιασμού (EDA – Electronic Design Automation), το οποίο κάνει ελαχιστοποίηση Boolean συναρτήσεων για προγραμματιζόμενους λογικούς πίνακες (PLA – Programmable Logic Arrays). Μετρά το χρόνο που απαιτείται για την εκτέλεση ενός συνόλου επτά μοντέλων εισόδου.
3. *Spice 2g6*: Ένας ακόμα εκπρόσωπος του EDA, το Spice 2g6 είναι ένα εργαλείο προσομοίωσης αναλογικών κυκλωμάτων. Μετρά το χρόνο προσομοίωσης ενός διπολικού κυκλώματος.
4. *Doduc*: Εκτελεί μία προσομοίωση Monte Carlo για συγκεκριμένες λειτουργίες πυρηνικού αντιδραστήρα. Δοκιμάζει ιδιαίτερα την αποτελεσματικότητα της cache μνήμης.
5. *NASA7*: Συλλογή από επτά προγράμματα πυρήνα που χρησιμοποιούν αριθμητική κινητής υποδιαστολής, τα οποία εκτελούν λειτουργίες πινάκων με δεδομένα διπλής ακρίβειας.
6. *LI*: Μετρά το χρόνο επίλυσης του προβλήματος των 9 βασιλισσών με τη χρήση του διερμηνευτή της γλώσσας LISP.
7. *Eqntott*: Μεταφράζει μία λογική αναπαράσταση μιας Boolean εξίσωσης, σε ένα πίνακα αληθείας.

8. *Matrix300*: Εκτελεί διάφορες λειτουργίες πινάκων με τη χρήση ρουτινών LINPACK, σε πίνακες μεγέθους 300 x 300. Χρησιμοποιεί αριθμητική κινητής υποδιαστολής διπλής ακρίβειας.
9. *Fpppp*: Είναι ένα πρόγραμμα σύγκρισης από την περιοχή της κβαντικής φυσικής, το οποίο είναι γραμμένο σε FORTRAN και χρησιμοποιεί αριθμητική κινητής υποδιαστολής διπλής ακρίβειας.
10. *Tomcatv*: Είναι ένα πρόγραμμα παραγωγής πλεγμάτων με τα ίδια κατασκευαστικά χαρακτηριστικά όπως το Fpppp.

Τα παραπάνω προγράμματα σύγκρισης SPEC, χρησιμοποιούν κυρίως την CPU, τη μονάδα κινητής υποδιαστολής (FPU – Floating Point Unit) και λιγότερο το υποσύστημα μνήμης. Χρησιμοποιούνται ιδιαίτερα για τη σύγκριση των ταχυτήτων διαφορετικών CPU. Η ισχύουσα σήμερα έκδοση του παραπάνω συνόλου προγραμμάτων σύγκρισης ονομάζεται *SPEC CPU2006* που αποτελείται από 30 προγράμματα.

Η *συνολική απόδοση* του συστήματος που θέλουμε να μελετήσουμε χρησιμοποιώντας την παραπάνω SPEC benchmark suite, ονομάζεται **SPECmark** και ορίζεται ως ο γεωμετρικός μέσος όρος των SPECthruput των *n* προγραμμάτων σύγκρισης. Δηλαδή
SPECmark = $\sqrt[n]{\alpha_1 \cdot \alpha_2 \cdots \alpha_n}$ όπου α_i είναι το SPECthruput του προγράμματος *i*.

Το **SPECthruput** είναι ο λόγος του χρόνου που απαιτεί η εκτέλεση του προγράμματος στο υπό μελέτη σύστημα, προς το χρόνο που απαιτεί η εκτέλεση του προγράμματος σε ένα σύστημα αναφοράς (το Sun Ultra Enterprise 2 – 1997, με ULTRASPARK II processor, για το *SPEC CPU2006*).

Η SPEC έχει προγράμματα σύγκρισης και για άλλες κατηγορίες πληροφοριακών συστημάτων, όπως το *SPECweb2009* για WWW εξυπηρετητές, το *SPECsfs2008* για εξυπηρετητές αρχείων, το *SPECjAppServer2004* για Java Servers κ.α. Περισσότερες πληροφορίες μπορείτε να βρείτε στη σελίδα της SPEC: http://www.spec.org

2.2 ΕΠΙΛΟΓΗ ΚΑΙ ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΦΟΡΤΙΟΥ ΕΡΓΑΣΙΑΣ

2.2.1 Η Επιλογή Φορτίου Εργασίας

Οι τέσσερις βασικοί παράγοντες από τους οποίους εξαρτάται η επιλογή του κατάλληλου φορτίου εργασίας είναι: οι υπηρεσίες που προσφέρει το υπό μελέτη σύστημα, το επίπεδο λεπτομέρειας στην περιγραφή των απαιτήσεων των χρηστών, η αντιπροσωπευτικότητα του φορτίου εργασίας και η παρακολούθηση των αλλαγών στη συμπεριφορά των χρηστών με την εξέλιξη του χρόνου.

1. Υπηρεσίες.

Όπως είδαμε στο Κεφάλαιο 1, η καταγραφή των υπηρεσιών που προσφέρει το σύστημα που θέλουμε να μελετήσουμε, είναι ένα από τα πρώτα βήματα που πρέπει να ακολουθήσουμε σε μια συστηματική μελέτη απόδοσης. Συχνά χρησιμοποιείται ο όρος *Σύστημα Υπό Μέτρηση* (System Under Test – *SUT*) για την περιγραφή του ευρύτερου συστήματος ή συνόλου στοιχείων που έχουμε στη διάθεσή μας. Αν υπάρχει κάποιο στοιχείο του SUT το οποίο θέλουμε να μελετήσουμε ιδιαίτερα, το ονομάζουμε *Στοιχείο Υπό Μελέτη* (Component Under Study – *CUS*). Για παράδειγμα, ένας σχεδιαστής μιας CPU μπορεί να θέλει να κατανοήσει την επίδραση στην απόδοση της CPU, διαφορετικών αρχιτεκτονικών λύσεων της Αριθμητικής – Λογικής Μονάδας (Arithmetic – Logic Unit – ALU). Στην περίπτωση αυτή η CPU είναι το SUT και η ALU είναι το CUS. Αντίστοιχα, μια τράπεζα που θέλει να προμηθευτεί ένα σύστημα επεξεργασίας διεργασιών, ίσως θέλει

να συγκρίνει διαφορετικά υποσυστήματα δίσκων. Στην περίπτωση αυτή, το σύστημα επεξεργασίας διεργασιών είναι το SUT και το υποσύστημα δίσκων είναι το CUS. Ο σαφής καθορισμός του SUT και του (ή των) CUS είναι πολύ σημαντικός σε μια μελέτη απόδοσης, αφού το μεν SUT καθορίζει τις υπηρεσίες και τις μετρικές απόδοσης, ενώ το CUS καθορίζει τις παραμέτρους και τους παράγοντες της μελέτης. Στο εξής ο όρος *σύστημα* θα αντιστοιχεί στο SUT, ενώ ο όρος *στοιχείο* θα αντιστοιχεί στο CUS.

Το φορτίο εργασίας που θα επιλεγεί, επίσης, πρέπει να αναφέρεται στις προσφερόμενες υπηρεσίες σε επίπεδο συστήματος και όχι στοιχείου. Αν το σύστημα προσφέρει *πολλαπλές* υπηρεσίες, το φορτίο εργασίας θα πρέπει να χρησιμοποιεί όσο το δυνατόν περισσότερες από αυτές. Έτσι, δεν είναι σωστό να μετρήσουμε την απόδοση μιας ALU που υποστηρίζει τόσο αριθμητική κινητής υποδιαστολής όσο και ακέραιης, με τη χρήση φορτίου εργασίας που χρησιμοποιεί μόνο ακέραιη αριθμητική.

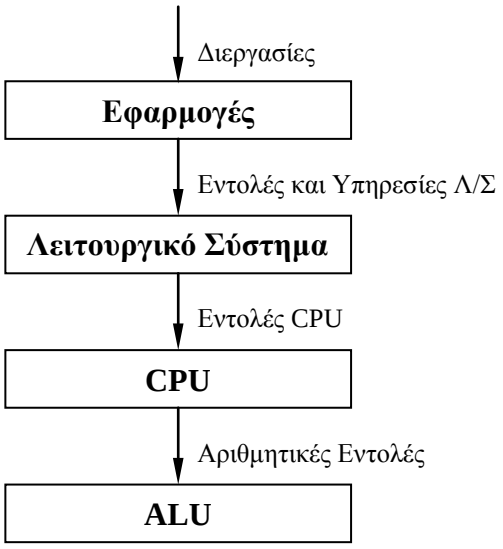
Εξετάζοντας τις υπηρεσίες του συστήματος, πρέπει επίσης να πάρουμε υπόψη το *στόχο* της μελέτης. Το φορτίο εργασίας πρέπει να χρησιμοποιεί τις πιο σημαντικές από τις υπηρεσίες. Για παράδειγμα, ένας συντάκτης γραφικών (graphics editor) ίσως να μην είναι αποδοτικός ως συντάκτης κειμένου (text editor). Κατά συνέπεια, ένα φορτίο εργασίας με χαρακτηριστικά συντάκτη κειμένου δεν είναι κατάλληλο σε μια μελέτη για συντάκτες, αν θέλουμε αυτοί να χρησιμοποιούνται κυρίως για εργασία με γραφικά.

Η επιλογή τύπου φορτίου εργασίας από τους περιγραφόμενους στην ενότητα 2.1, ακολουθεί τις αρχές που αναφέρθηκαν παραπάνω. Με το παράδειγμα που ακολουθεί μπορούμε να ερευνήσουμε το θέμα αυτό.

Παράδειγμα 2.1

Στο Σχήμα 2.2 φαίνεται μια ιεραρχική άποψη ενός συστήματος επεξεργασίας διεργασιών. Ο τυπικός χρήστης του συστήματος ενεργοποιεί μια διεργασία στο σύστημα, για παράδειγμα υποβάλλει μια αίτηση ανάληψης μετρητών από μια ATM τραπέζης.

Το λογισμικό εφαρμογών θα μεταφράσει την αίτηση σε ένα αριθμό αιτήσεων για υπηρεσίες από το Λειτουργικό Σύστημα (Λ/Σ). Το Λ/Σ στη συνέχεια, θα μεταφράσει τις αιτήσεις αυτές σε αιτήσεις για εξυπηρέτηση από τα διάφορα στοιχεία υλικού του συστήματος, όπως επεξεργαστές ειδικού και γενικού σκοπού, συσκευές εισόδου/εξόδου και δικτυακές συνδέσεις. Οι αιτήσεις προς τη CPU θα μεταφραστούν σε έναν αριθμό εντολών, η κάθε μία από τις οποίες μπορεί να δημιουργήσει μία ή περισσότερες αιτήσεις προς την ALU. Συνεπώς, όπως φαίνεται στο Σχήμα 2.2, υπάρχει μια ιεραρχία επιπέδων διεπαφής (interface levels) στα οποία εξυπηρετούνται οι αιτήσεις. Μία αίτηση σε ένα επίπεδο, μπορεί να καταλήξει σε μία ή περισσότερες αιτήσεις χαμηλότερου επιπέδου.



Σχήμα 2.2. Ιεραρχική άποψη ενός συστήματος επεξεργασίας διεργασιών.

Το φορτίο εργασίας θα μπορούσε να περιγραφεί, με την ποσοτικοποίηση των αιτήσεων σε οποιοδήποτε από τα τέσσερα επίπεδα διεπαφής, ανάλογα με το ποιο είναι το SUT. Αν συγκρίνουμε δύο CPU, τότε η CPU είναι το σύστημα και το σύνολο εντολών της CPU είναι οι προσφερόμενες υπηρεσίες. Κατάλληλο φορτίο εργασίας στην περίπτωση αυτή θα μπορούσε να είναι μία *μείξη εντολών*. Αν όμως η απόδοση μιας εντολής εξαρτάται από την απόδοση άλλων «συγγενών» εντολών, το φορτίο εργασίας θα πρέπει να εκφραστεί με βάση σύνολα εντολών που συνήθως χρησιμοποιούνται μαζί. Τέτοια σύνολα είναι τα *προγράμματα πυρήνα*.

Αν θέλουμε να συγκρίνουμε δύο συστήματα στο επίπεδο του Λειτουργικού Συστήματος, τότε οι προσφερόμενες υπηρεσίες είναι οι εντολές και υπηρεσίες Λ/Σ. Το φορτίο εργασίας θα μπορούσε να είναι ένα ή περισσότερα *συνθετικά προγράμματα*.

Τέλος, αν θέλουμε να συγκρίνουμε δύο ολοκληρωμένα συστήματα επεξεργασίας διεργασιών (transaction processing systems), τότε αναφερόμαστε στο επίπεδο Εφαρμογών ως σύστημα και οι αιτήσεις στο επίπεδο αυτό, δηλαδή οι διεργασίες, θα πρέπει να συγκροτούν τη βάση του φορτίου εργασίας. Τα *προγράμματα σύγκρισης* TPC benchmarks θα ήταν μια καλή επιλογή στην περίπτωση αυτή.

2. Επίπεδο λεπτομέρειας.

Έχοντας προσδιορίσει το σύστημα, το επίπεδο διεπαφής για τις υπηρεσίες και έχοντας δημιουργήσει μια λίστα των υπηρεσιών, το επόμενο βήμα στη διαδικασία επιλογής φορτίου εργασίας, είναι η επιλογή του *επιπέδου λεπτομέρειας* στην καταγραφή (και άρα στην αναπαραγωγή) των αιτήσεων για τις υπηρεσίες αυτές. Η περιγραφή του φορτίου εργασίας μπορεί να είναι μεγάλη όσο η ακολουθία όλων των αιτήσεων με τους χρόνους υποβολής τους, ή μικρή όσο μία μόνο αίτηση (π.χ. η περισσότερο χρησιμοποιούμενη). Οι πιθανές λύσεις είναι οι παρακάτω:

- *Περισσότερο χρησιμοποιούμενη αίτηση*: Είναι η επιλογή που προσφέρει τη μικρότερη πληροφορία για το σύστημα, αλλά συχνά χρησιμοποιείται ως αρχικό φορτίο εργασίας. Είναι χρήσιμη επιλογή όταν ο τύπος αυτός αιτήσεων κυριαρχεί σε σχέση με τις υπόλοιπες, ή καταναλώνει το μεγαλύτερο μέρος των πόρων του συστήματος. Παράδειγμα τέτοιας επιλογής είναι μία εντολή πρόσθεσης για ALU, διάφορα προγράμματα πυρήνα για τη σύγκριση επεξεργαστών, ή προγράμματα σύγκρισης για συστήματα επεξεργασίας διεργασιών.
- *Συχνότητα τύπων αιτήσεων*: Μία δεύτερη επιλογή είναι η καταγραφή διαφόρων αιτήσεων, των χαρακτηριστικών τους και των συχνοτήτων εμφάνισής τους. Οι μείξεις εντολών είναι παραδείγματα της προσέγγισης αυτής.
- *Ακολουθίες αιτήσεων*: Στην περίπτωση αυτή χρησιμοποιούμε μία ακολουθία πραγματικών αιτήσεων μαζί με τους χρόνους υποβολής τους, την οποία ονομάζουμε *ίχνος* (trace). Το βασικό πρόβλημα εδώ είναι το πολύ μεγάλο και συχνά ανεπιθύμητο επίπεδο λεπτομέρειας. Σίγουρα δεν είναι αποδεκτή λύση για *αναλυτικές τεχνικές* μελέτης απόδοσης.
- *Μέση απαίτηση χρήσης πόρων*: Στις αναλυτικές τεχνικές χρησιμοποιούνται ως φορτία εργασίας οι απαιτήσεις για χρήση των πόρων του συστήματος και όχι οι ίδιες οι αιτήσεις. Για παράδειγμα, το δεδομένο ότι κάθε χρήστης έχει μέσο χρόνο χρήσης της CPU ίσο με 50 msec και κάνει κατά μέσο όρο 20 ενέργειες εισόδου/εξόδου ανά αίτηση, πιθανόν να είναι ικανοποιητικό φορτίο εργασίας σε μια αναλυτική τεχνική.

- *Κατανομή απαιτήσεων χρήσης πόρων*: Σε πολλές περιπτώσεις δεν είναι αρκετή η μέση τιμή των απαιτήσεων χρήσης των πόρων, αλλά χρειάζεται η πλήρης περιγραφή της κατανομής πιθανότητάς τους. Αυτό συμβαίνει σε περιπτώσεις που έχουμε μεγάλη διασπορά στις απαιτήσεις, ή δεν έχουμε μια γνωστή θεωρητική κατανομή. Θα αντιμετωπίσουμε το πρόβλημα αυτό ιδιαίτερα στην προσομοίωση.

Οι δύο τελευταίες περιπτώσεις είναι εμφανές ότι αναφέρονται σε *μη-εκτελέσιμα* φορτία εργασίας.

3. Αντιπροσωπευτικότητα.

Ένα *φορτίο εργασίας μελέτης* πρέπει να είναι αντιπροσωπευτικό της πραγματικής *εφαρμογής* που θα επιβάλλεται στο σύστημα. Συνεπώς, το φορτίο εργασίας μελέτης και η πραγματική εφαρμογή θα πρέπει να ταιριάζουν στα παρακάτω τρία χαρακτηριστικά:

1. *Ρυθμός Αφίξεων*: Ο ρυθμός αφίξεων αιτήσεων του φορτίου εργασίας πρέπει να είναι ίδιος ή ανάλογος του ρυθμού αφίξεων αιτήσεων της εφαρμογής.
2. *Απαιτήσεις Χρήσης Πόρων*: Οι συνολικές απαιτήσεις για χρήση των βασικών πόρων στο φορτίο εργασίας, πρέπει να είναι ίδιες ή ανάλογες με αυτές της εφαρμογής.
3. *Τρόπος Χρήσης Πόρων*: Σχετίζεται με την ακολουθία και το μέγεθος χρήσης των διαφορετικών πόρων του συστήματος από μια εφαρμογή.

4. Παρακολούθηση αλλαγών.

Τα φορτία εργασίας πρέπει να παρακολουθούν τις αλλαγές στη συμπεριφορά των χρηστών με την εξέλιξη του χρόνου. Οι χρήστες αλλάζουν συμπεριφορά ανάλογα με τις υπηρεσίες που τους προσφέρονται από τα νέα συστήματα και έτσι τα φορτία εργασίας γίνονται πολλές φορές άχρηστα μόλις γίνουν κατανοητά.

Γενικά η παρατηρούμενη συμπεριφορά των χρηστών έχει ισχύ σε ένα συγκεκριμένο περιβάλλον, για ένα συγκεκριμένο σύστημα, σε μια συγκεκριμένη χρονική περίοδο. Οι χρήστες αλλάζουν συμπεριφορά με το χρόνο και μάλιστα οι αλλαγές στην απόδοση του συστήματος τους οδηγούν σε αλλαγές της συμπεριφοράς τους, δημιουργώντας έτσι ένα φαύλο κύκλο. Οι χρήστες έχουν την τάση να προσαρμόζουν τις απαιτήσεις τους έτσι ώστε να βελτιστοποιούν την απόκριση από το σύστημα, προσανατολιζόμενοι στις υπηρεσίες που το σύστημα προσφέρει πιο αποδοτικά. Ακόμα και στο ίδιο σύστημα, η συμπεριφορά των χρηστών αλλάζει με το χρόνο. Για παράδειγμα, αρχικά οι υπολογιστές χρησιμοποιούνταν για επιστημονικές εφαρμογές που απαιτούσαν γρήγορες αριθμητικές λειτουργίες. Σήμερα σε πολλά συστήματα, όπως οι βάσεις δεδομένων, μεγαλύτερη βαρύτητα έχουν άλλες λειτουργίες όπως αυτές της εισόδου/εξόδου.

Η παρακολούθηση της συμπεριφοράς των χρηστών έχει ιδιαίτερη σημασία σε μελέτες απόδοσης συστημάτων που βρίσκονται στη φάση σχεδιασμού. Είναι παρακινδυνευμένο να χρησιμοποιούνται φορτία εργασίας βασισμένα σε υπάρχοντα συστήματα. Πρέπει να παρακολουθούμε τη συμπεριφορά των χρηστών συνέχεια και να χρησιμοποιούμε φορτία εργασίας βασισμένα σε *πρόσφατες* μετρήσεις *παρόμοιων* συστημάτων, που λειτουργούν σε *παρόμοια* περιβάλλοντα.

2.2.2 Χαρακτηρισμός του Φορτίου Εργασίας

Η διαδικασία μελέτης του πραγματικού περιβάλλοντος των χρηστών, παρατήρησης των βασικών χαρακτηριστικών του και κατασκευής ενός φορτίου εργασίας που μπορεί να χρησιμοποιηθεί πολλές φορές, ονομάζεται **χαρακτηρισμός του φορτίου εργασίας.**

Με την έννοια *χρήστης*, εννοούμε την οντότητα που υποβάλλει τις αιτήσεις για εξυπηρέτηση στο επίπεδο διεπαφής SUT. Ο χρήστης μπορεί να είναι άνθρωπος, μπορεί

και να μην είναι. Για παράδειγμα, αν το σύστημα είναι ένας επεξεργαστής, οι χρήστες μπορεί να είναι προγράμματα. Αν το σύστημα είναι ένα τοπικό δίκτυο, οι χρήστες θα είναι οι σταθμοί εργασίας που είναι συνδεδεμένοι στο δίκτυο. Πολλές φορές, αντί για τον όρο «χρήστης», χρησιμοποιείται ο όρος *στοιχείο φορτίου εργασίας* (workload component), ή *μονάδα φορτίου εργασίας* (workload unit).

Οι ποσότητες που μετρώνται (αιτήσεις εξυπηρέτησης, ή απαιτήσεις χρήσης πόρων του συστήματος), με σκοπό να χρησιμοποιηθούν για το χαρακτηρισμό του φορτίου εργασίας, ονομάζονται *παράμετροι του φορτίου εργασίας*. Παραδείγματα τέτοιων παραμέτρων είναι οι τύποι διεργασιών, οι εντολές, τα μεγέθη πακέτων, οι προορισμοί των πακέτων, οι τρόποι αναφοράς σε σελίδες. Στην επιλογή των παραμέτρων πρέπει να προσέχουμε ώστε αυτές να εξαρτώνται από το φορτίο εργασίας και όχι από το σύστημα. Για παράδειγμα, είναι προτιμότερο να χαρακτηρίσουμε μία διεργασία με το ρυθμό αφίξεών της στο σύστημα και όχι με το χρόνο απόκρισής της.

Διάφορα *χαρακτηριστικά* των αιτήσεων εξυπηρέτησης, ή των απαιτήσεων χρήσης πόρων του συστήματος, έχουν ενδιαφέρον και μπορούν να αντιπροσωπεύονται στο φορτίο εργασίας: Οι χρονικές στιγμές άφιξης των αιτήσεων, οι τύποι αιτήσεων, οι διάρκειες των αιτήσεων, τα μεγέθη χρήσης των πόρων που επιβάλλουν. Τα χαρακτηριστικά με τη μεγαλύτερη επίπτωση στην απόδοση του συστήματος, πρέπει να περιλαμβάνονται στο φορτίο εργασίας, σε αντίθεση με τα χαρακτηριστικά που έχουν μικρή επίδραση. Για παράδειγμα, σε ένα δίκτυο δεδομένων, αν το μέγεθος των πακέτων δεν επηρεάζει το χρόνο προώθησης των πακέτων σε ένα δρομολογητή, δεν χρειάζεται να το περιλάβουμε στη λίστα των παραμέτρων του φορτίου εργασία, σε αντίθεση με άλλα χαρακτηριστικά, όπως ο αριθμός των πακέτων και οι χρονικές στιγμές άφιξής τους.

Παρακάτω παρουσιάζονται με συντομία οι βασικές *τεχνικές χαρακτηρισμού των παραμέτρων* του φορτίου εργασίας. Σημειώστε ότι πολλές φορές χρησιμοποιούμε περισσότερες από μία τεχνικές σε συνδυασμό, προκειμένου να χαρακτηρίσουμε ένα πολύπλοκο φορτίο εργασίας.

1. Μέση τιμή.

Η απλούστερη μέθοδος χαρακτηρισμού είναι να δημιουργήσουμε έναν αριθμό που θα αντιπροσωπεύει όλες τις τιμές της παραμέτρου που έχουμε παρατηρήσει. Η λογικότερη επιλογή είναι η *μέση τιμή*. Αν έχουμε παρατηρήσει τις n τιμές $\{x_1, x_2, \ldots, x_n\}$ μιας παραμέτρου, η πιο απλή επιλογή, η *αριθμητική μέση τιμή* $\bar{x}$ ορίζεται ως:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Υπάρχουν περιπτώσεις που η αριθμητική μέση τιμή δεν είναι κατάλληλη και χρησιμοποιούμε άλλες μέσες τιμές όπως η γεωμετρική, η αρμονική κ.α.

2. Διασπορά.

Η μέση τιμή δεν είναι από μόνη της αντιπροσωπευτικό μέγεθος όταν οι τιμές έχουν μεγάλη διασπορά. Η *διακύμανση* s^2 είναι ένα μέγεθος που περιγράφει τη διασπορά των τιμών γύρω από την αριθμητική μέση τιμή τους:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$$

Πολλές φορές είναι πιο βολικό να χρησιμοποιούμε το μέγεθος *τυπική απόκλιση* s που είναι η τετραγωνική ρίζα της διακύμανσης.

Ο λόγος είναι ότι εκφράζεται με τις ίδιες μονάδες όπως η μέση τιμή. Επίσης χρησιμοποιείται το μέγεθος *συντελεστής διασποράς* $C = s / \bar{x}$.

Άλλες επιλογές για την περιγραφή της διασποράς, είναι το εύρος των τιμών (μέγιστο και ελάχιστο), οι περιοχές στις οποίες βρίσκονται τα ποσοστά 10% ή 90% των τιμών κ.α.

3. Ιστογράμματα.

Ένα *ιστόγραμμα* παρουσιάζει τις σχετικές συχνότητες των διαφόρων τιμών μιας παραμέτρου. Για παραμέτρους με συνεχές πεδίο τιμών, πρέπει να χωρίσουμε το εύρος τιμών σε μικρότερα τμήματα (*κυψέλες)* και να μετρήσουμε τις παρατηρήσεις (τιμές) που περιλαμβάνονται σε κάθε κυψέλη. Το ιστόγραμμα έχει τη μορφή διαγράμματος στο επίπεδο *x*, *y*, ή μπορεί να έχει τη μορφή πίνακα. Σε κάθε περίπτωση, στη μία διάσταση εμφανίζονται οι σχετικές συχνότητες των τιμών, ενώ στην άλλη οι διακριτές τιμές ή οι κυψέλες τιμών, για διακριτά ή συνεχή πεδία τιμών αντίστοιχα.

Τα δεδομένα του ιστογράμματος μπορούν να χρησιμοποιηθούν σε πειράματα μετρήσεων ή σε προσομοιώσεις για τη δημιουργία φορτίων εργασίας μελέτης. Σε αναλυτικά μοντέλα τα ιστογράμματα χρησιμοποιούνται για τον προσδιορισμό ή την επιβεβαίωση της χρησιμοποιούμενης κατανομής πιθανότητας.

Αν υπάρχει σημαντική *συσχέτιση* μεταξύ διαφορετικών παραμέτρων φορτίου εργασίας, τότε η χρήση ιστογράμματος μιας παραμέτρου όπως περιγράφηκε παραπάνω, ίσως να μην είναι ικανοποιητική. Στην περίπτωση αυτή μπορούμε να δημιουργήσουμε *πολυπαραμετρικά* ιστογράμματα.

4. Ακολουθίες Markov.

Πολλές φορές είναι απαραίτητο να έχουμε εκτός από τον αριθμό (π.χ. τη μέση τιμή) των αιτήσεων και τη σειρά με την οποία εμφανίζονται οι διάφοροι τύποι αιτήσεων. Αφορά περιπτώσεις που ο τύπος της επόμενης αίτησης καθορίζεται από τον τύπο ορισμένων από τις τελευταίες που έχουν γίνει. Αν κάνουμε την απλοποιητική παραδοχή ότι καθορίζεται μόνο από την τελευταία αίτηση, τότε η ακολουθία των αιτήσεων συμπεριφέρεται ως μια *ακολουθία Markov*. Οι ακολουθίες Markov αποτελούν χρήσιμα μοντέλα στις αναλυτικές τεχνικές όπως τα μοντέλα θεωρίας αναμονής και θα τις μελετήσουμε με λεπτομέρεια στο Κεφάλαιο 3.

Για τις ανάγκες του χαρακτηρισμού του φορτίου εργασίας, είναι αρκετό να σημειώσουμε ότι οι ακολουθίες Markov περιγράφονται από ένα *πίνακα μεταβάσεων* ο οποίος περιέχει τις πιθανότητες να έχουμε επόμενη αίτηση ενός τύπου, δεδομένου του τύπου της τελευταίας αίτησης.

Οι ακολουθίες Markov μπορούν να περιγράψουν γενικότερα συστήματα που μεταπηδούν από μια *κατάσταση* σε άλλη, όπως φαίνεται στο παρακάτω παράδειγμα.

Παράδειγμα 2.2

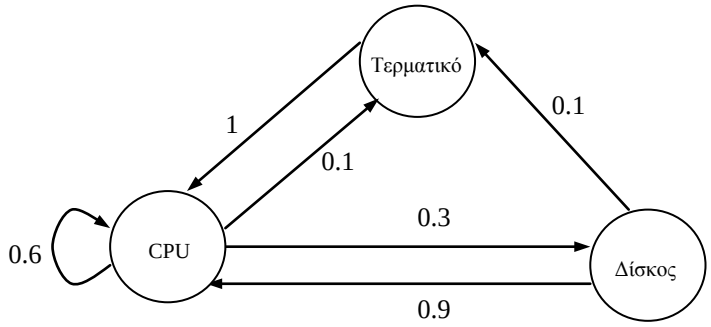
Ο πίνακας μεταβάσεων του Πίνακα 2.2., αντιστοιχεί σε σύστημα που αναφέρεται στη μετάβαση μιας εργασίας μεταξύ της CPU, του δίσκου και του τερματικού σε έναν υπολογιστή.

Από/Προς	CPU	Δίσκος	Τερματικό
CPU	0.6	0.3	0.1
Δίσκος	0.9	0	0.1
Τερματικό	1	0	0

Πίνακας 2.2. Πίνακας Μεταβάσεων

Στο Σχήμα 2.3 φαίνεται το αντίστοιχο *διάγραμμα καταστάσεων – πιθανοτήτων μεταβάσεων* για το ίδιο σύστημα. Οι κύκλοι αντιστοιχούν στις καταστάσεις του συστήματος και τα βέλη στις μη-μηδενικές πιθανότητες μετάβασης του Πίνακα 2.2. Έτσι,

μετά από κάθε επίσκεψη της εργασίας στη CPU, με πιθανότητα 0.3 θα μετακινηθεί στο δίσκο, με πιθανότητα 0.1 θα μετακινηθεί στο τερματικό και με πιθανότητα 0.6 θα ξαναχρησιμοποιήσει τη CPU.



Σχήμα 2.3. Διάγραμμα καταστάσεων – πιθανοτήτων μεταβάσεων.

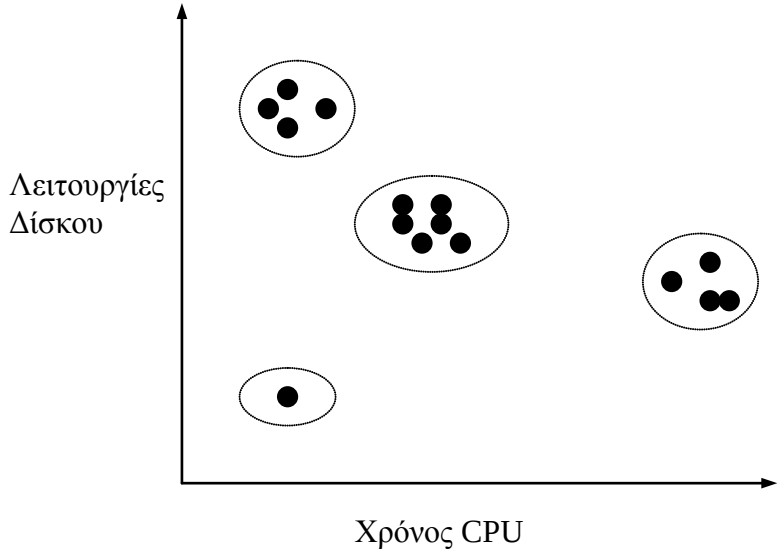
Οι ακολουθίες Markov δεν είναι χρήσιμα μοντέλα μόνο για την περιγραφή των ακολουθιών αιτήσεων, ή των μεταβάσεων μεταξύ πόρων του συστήματος (όπως στο τελευταίο παράδειγμα), αλλά και για μεταβολές μεταξύ διαφορετικών τύπων εφαρμογών, ή ακόμα για την περιγραφή της *τοπικότητας αναφορών σελίδων*. Στην τελευταία περίπτωση ο πίνακας μεταβάσεων περιέχει τις πιθανότητες να αναφερθούν από ένα πρόγραμμα οι σελίδες *i* αμέσως μετά την αναφορά στις σελίδες *j*.

5. Ομαδοποίηση.

Γενικά τα φορτία εργασίας που μετράμε αποτελούνται από ένα μεγάλο αριθμό στοιχείων. Για παράδειγμα, μπορεί να έχουμε καταγράψει μερικές χιλιάδες διαφορετικά προφίλ συμπεριφοράς χρηστών. Είναι χρήσιμο, τότε, να ταξινομήσουμε αυτά τα στοιχεία σε ένα μικρό αριθμό κλάσεων ή *ομάδων*, έτσι ώστε τα στοιχεία κάθε ομάδας να είναι παρόμοια. Αργότερα, ένα μέλος κάθε ομάδας μπορεί να επιλεγεί ώστε να αντιπροσωπεύει την ομάδα και η χρήση του στη μελέτη ως φορτίο εργασίας, να επιτρέπει την εξαγωγή συμπερασμάτων για ολόκληρη την ομάδα.

Παράδειγμα 2.3

Το Σχήμα 2.4 παρουσιάζει τις απαιτήσεις σε χρόνο CPU και λειτουργίες εισόδου/εξόδου δίσκου, από 15 εργασίες. Οι εργασίες ταξινομούνται σε τέσσερις ομάδες. Έτσι, αντί να χρησιμοποιήσουμε 15 διαφορετικούς τύπους εργασιών, μπορούμε να περιοριστούμε σε 4 τύπους που αντιπροσωπεύουν τη μέση απαίτηση σε χρήση των δύο πόρων από κάθε ομάδα.



Σχήμα 2.4. Ομαδοποίηση απαιτήσεων χρήσης πόρων.

2.3 ΕΛΕΓΚΤΕΣ

Οι **ελεγκτές** (monitors) είναι εργαλεία που χρησιμοποιούνται για τη μέτρηση του επιπέδου δραστηριότητας σε ένα πληροφοριακό σύστημα.

Η βασική λειτουργία ενός ελεγκτή μετρήσεων είναι η συλλογή δεδομένων σχετικά με τη λειτουργία του συστήματος. Στην ιδανική περίπτωση, ο ελεγκτής πρέπει να είναι ένας «παρατηρητής» του συστήματος και όχι μέρος του. Πρέπει, φυσικά, να κατασκευάζεται και να χρησιμοποιείται με τρόπο ώστε να μην επηρεάζει τη λειτουργία του συστήματος που μελετάμε.

Δύο χαρακτηριστικά των ελεγκτών που έχουν ενδιαφέρον, είναι ο *τύπος* τους και ο *τρόπος λειτουργίας*. Υπάρχουν τρεις βασικοί τύποι ελεγκτών σε σχέση με τη μορφή υλοποίησής τους: *υλικοί* ελεγκτές, *λογισμικοί* ελεγκτές και *υβριδικοί*. Όσον αφορά τον τρόπο λειτουργίας τους, έχουμε δύο κυρίως τρόπους συλλογής δεδομένων από τους ελεγκτές: *καταγραφή γεγονότων* (event trace) και *δειγματοληψία*.

2.3.1 Τύποι Ελεγκτών

1. Υλικοί ελεγκτές.

Οι *υλικοί ελεγκτές* είναι εργαλεία μέτρησης που εντοπίζουν γεγονότα σε ένα πληροφοριακό σύστημα μέσω της παρακολούθησης προκαθορισμένων σημάτων. Εξετάζουν την κατάσταση του συστήματος με ηλεκτρονικά στοιχεία τα οποία συνδέονται στα κυκλώματα και καταγράφουν τις μετρήσεις. Τα ηλεκτρονικά στοιχεία μπορούν να παρακολουθούν την κατάσταση των τμημάτων υλικού του συστήματος, όπως οι καταχωρητές, οι θέσεις μνήμης και τα κανάλια εισόδου/εξόδου. Για παράδειγμα, ένας υλικός ελεγκτής μπορεί να εντοπίσει μία λειτουργία ανάγνωσης μνήμης αναγνωρίζοντας ότι η αίτηση ανάγνωσης μιας θέσης μνήμης αλλάζει από τη μη-ενεργή στην ενεργή κατάσταση.

Ως *πλεονεκτήματα* των υλικών ελεγκτών μπορούμε να εντοπίσουμε τα εξής: Δεν καταναλώνουν πόρους του συστήματος ούτε επιβάλλουν επιπλέον φορτίο που θα άλλαζε τη συμπεριφορά του, αφού ουσιαστικά είναι εξωτερικές συσκευές. Είναι διαθέσιμοι συνήθως για χρήση σε περισσότερα από ένα συστήματα, αφού δεν εξαρτώνται από το λειτουργικό σύστημα ή άλλα υποσυστήματα λογισμικού.

Οι υλικοί ελεγκτές έχουν και *μειονεκτήματα* όπως το γεγονός ότι χρειάζεται εξειδίκευση για τη χρησιμοποίησή τους, δεν έχουν γενικά φιλικότητα στη χρήση τους και χρειάζεται αυξημένη προσπάθεια για το σχεδιασμό πειραμάτων βασισμένων σε υλικούς ελεγκτές. Ακόμα, είναι δύσκολο να εντοπίσουν γεγονότα που είναι βασισμένα στο λογισμικό, όπως για παράδειγμα, ο τύπος μιας διαδικασίας ή ενός φορτίου εργασίας, ο αριθμός των διεργασιών που έχουν δημιουργήσει ένα συγκεκριμένο γεγονός κ.λ.π.

Παραδείγματα σύγχρονων υλικών ελεγκτών είναι τα *HWMonitor, SpeedFan*

2. Λογισμικοί ελεγκτές.

Ένας *λογισμικός ελεγκτής* αποτελείται από ρουτίνες που έχουν εισαχθεί στο λογισμικό ενός πληροφοριακού συστήματος με στόχο να καταγράφουν την κατάσταση και τα

γεγονότα του συστήματος. Συλλέγουν δεδομένα απόδοσης σχετικά με την εκτέλεση ενός ή περισσοτέρων προγραμμάτων, καθώς και για τα τμήματα υλικού του συστήματος. Οι ρουτίνες του ελεγκτή ενεργοποιούνται είτε με την εμφάνιση συγκεκριμένων γεγονότων, ή σε συγκεκριμένες χρονικές στιγμές, ανάλογα με τον τρόπο λειτουργίας του ελεγκτή, που θα συζητήσουμε στην επόμενη υποενότητα.

Οι λογισμικοί ελεγκτές μπορούν γενικά να καταγράψουν οποιαδήποτε πληροφορία διαθέσιμη σε προγράμματα και το λειτουργικό σύστημα. Η δυνατότητα αυτή μαζί με τη μεγάλη ευελιξία στην επιλογή των δεδομένων που θα καταγραφούν, κάνει τους λογισμικούς ελεγκτές πολύ δημοφιλείς.

Πάντως, επειδή οι λογισμικοί ελεγκτές καταναλώνουν πόρους του συστήματος για την εκτέλεση των ρουτινών τους, δημιουργούν συνήθως μεγάλη επιβάρυνση στη λειτουργία του συστήματος και πιθανότατα αλλοίωση των μετρήσεων.

Πλεονεκτήματα των λογισμικών ελεγκτών θεωρούνται η ευκολία εγκατάστασης και χρήσης τους. *Μειονεκτήματα* είναι η επιβάρυνση στη λειτουργία του συστήματος και η πιθανή εξάρτηση από ένα συγκεκριμένο λειτουργικό σύστημα.

Οι *IBM Resource Management Facility (RMF)* και *Unisys Software Instrumentation (SIP)* είναι παραδείγματα λογισμικών ελεγκτών που μετρούν και δίνουν πληροφορίες για μετρικές απόδοσης όπως η ρυθμαπόδοση, η χρησιμοποίηση συσκευών, η δραστηριότητα εισόδου/εξόδου κ.α.

3. Υβριδικοί ελεγκτές.

Οι *υβριδικοί ελεγκτές* είναι συνδυασμοί υλικών και λογισμικών ελεγκτών, τέτοιοι ώστε να εμφανίζουν τα καλύτερα χαρακτηριστικά των δύο βασικών τύπων. Σε έναν υβριδικό ελεγκτή, οι ρουτίνες λογισμικού είναι υπεύθυνες για τον εντοπισμό των γεγονότων και τη μεταβίβαση της πληροφορίας σε ειδικούς καταχωρητές. Το υλικό τμήμα του υβριδικού ελεγκτή είναι υπεύθυνο για την καταγραφή των δεδομένων, αποφεύγοντας την επιβάρυνση των λειτουργιών εισόδου/εξόδου του συστήματος. Ένα σημαντικό πρόβλημα των υβριδικών ελεγκτών είναι η απαίτηση εξειδικευμένου υλικού, όπως οι καταχωρητές και τα σημεία ελέγχου στο σύστημα. Οι υβριδικοί ελεγκτές είναι ουσιαστικά χρήσιμοι μόνο αν είναι ενσωματωμένοι στο υπό μελέτη σύστημα.

Ένα παράδειγμα υβριδικού ελεγκτή για υπολογιστικά συστήματα, είναι ο ελεγκτής *Diamond*, του οποίου το υλικό τμήμα είναι μια πλακέτα που καταγράφει την κίνηση των δεδομένων στο δίαυλο (bus) του συστήματος, ενώ το λογισμικό τμήμα μπορεί να διαβάζει διευθύνσεις εντολών, καταστάσεις του επεξεργαστή κ.α.

2.3.2 Τρόποι Λειτουργίας των Ελεγκτών

1. Καταγραφή γεγονότων.

Ένα σήμα διακοπής (interrupt) εισόδου/εξόδου που σηματοδοτεί την ολοκλήρωση μιας λειτουργίας ανάγνωσης/γραφής στο δίσκο, μπορεί να θεωρηθεί ως ένα *γεγονός* που αλλάζει την κατάσταση του συστήματος.

Η **κατάσταση** ενός συστήματος ορίζεται ως η συλλογή των τιμών των παραμέτρων που είναι απαραίτητες για την περιγραφή του συστήματος σε μια χρονική στιγμή.

Ένας *ελεγκτής καταγραφής γεγονότων* (event trace monitor) συλλέγει πληροφορία από το σύστημα, τις χρονικές στιγμές εμφάνισης συγκεκριμένων γεγονότων. Για παράδειγμα, στο επίπεδο του λειτουργικού συστήματος, η κατάσταση του συστήματος μπορεί να εκφράζεται από την τριάδα αριθμών που δείχνουν πόσες διαδικασίες είναι σε φάση εκτέλεσης, στην ουρά έτοιμων για εκτέλεση και στην ουρά μπλοκαρισμένων διαδικασιών

αντίστοιχα. Στην περίπτωση αυτή, γεγονότα θα μπορούσαν να θεωρηθούν μία κλήση για υπηρεσία του λειτουργικού συστήματος, ή μια αίτηση για λειτουργία εισόδου/εξόδου.

Συνήθως ένας *λογισμικός* ελεγκτής καταγραφής γεγονότων αποτελείται από ειδικά προγράμματα ενσωματωμένα σε συγκεκριμένα σημεία του λειτουργικού συστήματος. Με τον εντοπισμό ενός γεγονότος καλείται μια κατάλληλη ρουτίνα η οποία δημιουργεί μια εγγραφή που περιέχει τον τύπο, την ημερομηνία και την ώρα εμφάνισης του γεγονότος, καθώς και άλλα δεδομένα σχετιζόμενα με το συγκεκριμένο γεγονός. Για παράδειγμα, μία εγγραφή για το γεγονός της ολοκλήρωσης μιας διαδικασίας μπορεί να περιέχει το χρόνο CPU που χρησιμοποιήθηκε από τη διαδικασία, τον αριθμό των σφαλμάτων σελίδας που προκάλεσε, τον αριθμό λειτουργιών εισόδου/εξόδου που εκτελέστηκαν, την ποσότητα μνήμης που χρειάστηκε κ.α.

Όταν ο ρυθμός εμφάνισης γεγονότων γίνει πολύ μεγάλος, οι ρουτίνες του ελεγκτή εκτελούνται πολύ συχνά, με συνέπεια να εισάγεται μεγάλη επιβάρυνση στη λειτουργία και στις μετρήσεις του συστήματος. Πολλές φορές η επιβάρυνση αυτή μπορεί να φτάσει σε ανεπιθύμητα επίπεδα, όπως για παράδειγμα 30%. Ανεκτά θεωρούνται επίπεδα επιβάρυνσης έως 5%. Με δεδομένο ότι ο ρυθμός των γεγονότων δεν ελέγχεται από τον ελεγκτή, το επίπεδο επιβάρυνσης είναι ουσιαστικά απρόβλεπτο. Το τελευταίο αποτελεί το βασικό μειονέκτημα των ελεγκτών καταγραφής γεγονότων.

2. Δειγματοληψία.

Ένας *ελεγκτής δειγματοληψίας* συλλέγει πληροφορία για το σύστημα σε συγκεκριμένες χρονικές στιγμές. Οι ρουτίνες συλλογής πληροφορίας ενός *λογισμικού* ελεγκτή δειγματοληψίας, αντί να ενεργοποιούνται με την εμφάνιση κάποιου γεγονότος, καλούνται σε προκαθορισμένες χρονικές στιγμές οι οποίες ορίζονται κατά την έναρξη εκτέλεσης των μετρήσεων. Η δειγματοληψία οδηγείται από χρονικά σήματα διακοπής βασισμένα σε εσωτερικό ρολόι του συστήματος.

Η επιβάρυνση στη λειτουργία του συστήματος που επιβάλλεται από έναν ελεγκτή δειγματοληψίας εξαρτάται από δύο παράγοντες: τον αριθμό των μεταβλητών που μετρώνται και το μέγεθος του διαστήματος δειγματοληψίας. Και οι δύο παράγοντες μπορούν να καθοριστούν από τον ελεγκτή δειγματοληψίας και κατά συνέπεια μπορούμε να ελέγξουμε το ύψος της επιβάρυνσης. Εδώ υπάρχει μια καθαρή περίπτωση δοσοληψίας μεταξύ της επιβάρυνσης και της ακρίβειας των μετρήσεων: μεγάλα διαστήματα δειγματοληψίας έχουν σαν αποτέλεσμα μικρή επιβάρυνση στη λειτουργία του συστήματος, αλλά από την άλλη πλευρά μειώνουν τον αριθμό των δειγμάτων και κατά συνέπεια το διάστημα εμπιστοσύνης των μετρήσεων.

Σε σύγκριση με τους ελεγκτές καταγραφής γεγονότων, οι ελεγκτές δειγματοληψίας προσφέρουν μάλλον μικρότερη λεπτομέρεια στις μετρήσεις τους, αφού οι πρώτοι καθοδηγούνται από τα γεγονότα του συστήματος και καταγράφουν όλες τις αλλαγές στην κατάστασή του.

2.4 ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΔΙΑΧΕΙΡΙΣΗ ΧΩΡΗΤΙΚΟΤΗΤΑΣ

Ο *σχεδιασμός της χωρητικότητας* (capacity planning) σε ένα πληροφοριακό σύστημα, έχει στόχο να διασφαλίσει ότι θα υπάρχουν διαθέσιμοι αρκετοί υπολογιστικοί πόροι, ώστε να εξυπηρετούνται με αποδοτικό τρόπο τα μελλοντικά φορτία εργασίας, ικανοποιώντας παράλληλα τις απαιτήσεις απόδοσης του συστήματος. Από την άλλη πλευρά, ο όρος *διαχείριση χωρητικότητας* (capacity management) χρησιμοποιείται για την περιγραφή του προβλήματος της μεγιστοποίησης της απόδοσης του υπάρχοντος

συστήματος. Δηλαδή, μπορούμε να πούμε ότι η διαχείριση χωρητικότητας ασχολείται με το παρόν, ενώ ο σχεδιασμός χωρητικότητας με το μέλλον ενός συστήματος.

Μία εναλλακτική προσέγγιση στο σχεδιασμό χωρητικότητας είναι η προμήθεια περισσοτέρων υπολογιστικών πόρων, ενώ για τη διαχείριση χωρητικότητας είναι η *ρύθμιση απόδοσης* (performance tuning), δηλαδή η αλλαγή των τιμών των παραμέτρων του συστήματος ώστε να μεγιστοποιείται η απόδοση.

2.4.1 Τα Βήματα στο Σχεδιασμό και Διαχείριση Χωρητικότητας

Τα βήματα που ακολουθούμε είτε για σχεδιασμό ή για διαχείριση της χωρητικότητας ενός συστήματος, είναι βασικά τα ίδια:

1. Προετοιμασία του συστήματος για μετρήσεις.
2. Καταγραφή της χρήσης του συστήματος.
3. Χαρακτηρισμός του φορτίου εργασίας.
4. Εκτίμηση της απόδοσης για διαφορετικές εναλλακτικές επιλογές.
5. Υιοθέτηση της επιλογής με το μικρότερο λόγο κόστους/απόδοσης.

Στο πρώτο βήμα προσπαθούμε να εξασφαλίσουμε ότι θα υπάρχουν τα κατάλληλα σημεία μέτρησης και ελεγκτές στο σύστημα, ώστε να μπορούμε να καταγράφουμε τη χρήση του.

Τα επόμενα δύο βήματα αφορούν, όπως έχουμε ήδη δει, τη συγκέντρωση δεδομένων για μια χρονική περίοδο, την ανάλυσή τους και την περιγραφή τους με μορφή τέτοια που μπορεί να χρησιμοποιηθεί ως είσοδος σε ένα *μοντέλο* του συστήματος για εκτίμηση της απόδοσής του.

Τα βήματα 1 και 2 εκτελούνται στο υπό μελέτη σύστημα αν αυτό είναι υπαρκτό και διαθέσιμο. Αλλιώς μπορεί να χρησιμοποιηθεί ένα αντίστοιχο σύστημα υπό συνθήκες παρόμοιες με αυτές που περιμένουμε να ισχύουν στο υπό μελέτη σύστημα.

Στη *διαχείριση χωρητικότητας*, η ισχύουσα διάταξη και το φορτίο εργασίας του συστήματος δίνονται ως είσοδοι στο μοντέλο, στο οποίο γίνεται ρύθμιση απόδοσης ώστε να πάρουμε συμπεράσματα για τις αναγκαίες αλλαγές στις τιμές των παραμέτρων του συστήματος. Το μοντέλο αυτό είναι συνήθως είτε ένας προσομοιωτής, ή ένα λεπτομερές αναλυτικό μοντέλο, δηλαδή ένα σύνολο κανόνων και σχέσεων που έχουν αναπτυχθεί ειδικά για το υπό μελέτη σύστημα.

Στο *σχεδιασμό χωρητικότητας*, πρώτα γίνεται πρόβλεψη του φορτίου εργασίας με βάση μετρήσεις του συστήματος (ή αντίστοιχου υπαρκτού). Στη συνέχεια, διαφορετικές εναλλακτικές διατάξεις και μελλοντικά προβλεπόμενα φορτία εργασίας του συστήματος, δίνονται ως είσοδος στο μοντέλο που προορίζεται για την εκτίμηση της απόδοσης του συστήματος. Τα μοντέλα που χρησιμοποιούνται για σχεδιασμό χωρητικότητας είναι γενικά λιγότερο λεπτομερή από τα μοντέλα που χρησιμοποιούνται για διαχείριση χωρητικότητας. Μία δημοφιλής επιλογή είναι τα μοντέλα της *θεωρίας αναμονής* (queueing models).

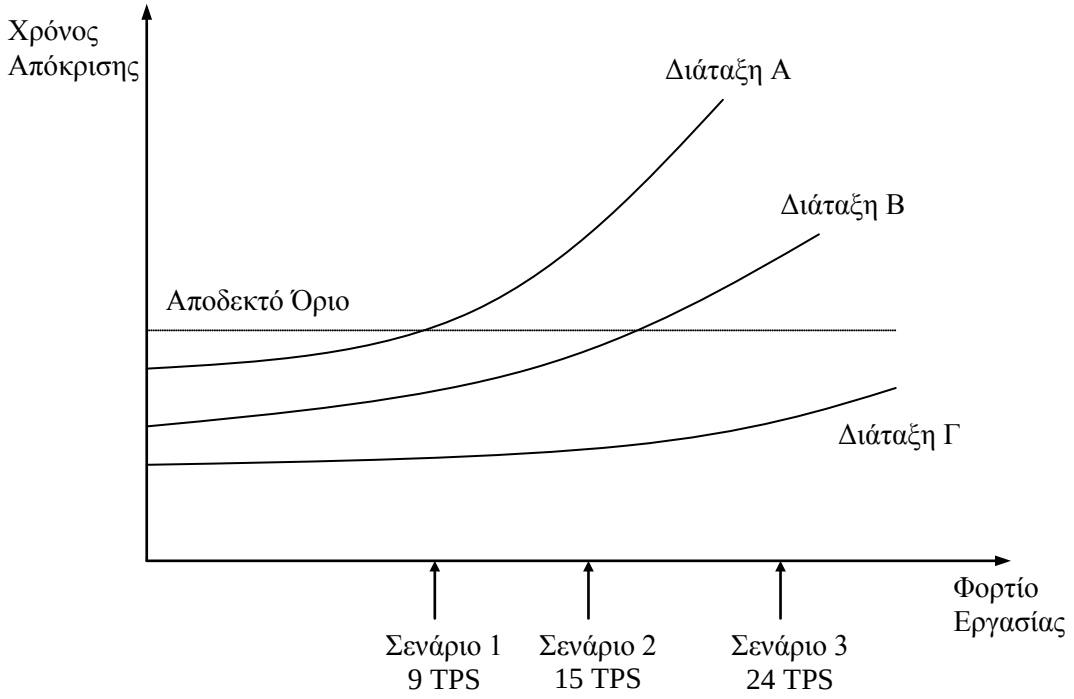
2.4.2 Η Εκτίμηση Απόδοσης Διαφορετικών Εναλλακτικών Επιλογών

Το βήμα αυτό είναι το βασικότερο στην όλη διαδικασία σχεδιασμού χωρητικότητας. Τα υπό μελέτη συστήματα αποτελούνται πολλές φορές από συνδυασμούς πολλών υποσυστημάτων όπως σταθμοί εργασίας, εξυπηρετητές, ειδικά συστήματα (π.χ. συστήματα επεξεργασίας διεργασιών) και δίκτυα. Τα υποσυστήματα αυτά μπορούν φυσικά να συνδυαστούν σε πολλές διαφορετικές πιθανές διατάξεις.

Η πρόβλεψη πάντα εμπεριέχει ένα βαθμό αβεβαιότητας και κατά συνέπεια πρέπει να ληφθούν υπόψη διάφορα σενάρια μελλοντικών φορτίων εργασίας. Στο σημείο αυτό χρειαζόμαστε μια τεχνική εκτίμησης της απόδοσης του συστήματος η οποία θα χρησιμοποιήσει ως είσοδο τις εναλλακτικές διατάξεις και τα μελλοντικά φορτία εργασίας του συστήματος. Όπως είδαμε στην προηγούμενη υποενότητα, μπορούμε να χρησιμοποιήσουμε αναλυτικές τεχνικές, όπως μοντέλα της θεωρία αναμονής. Με δεδομένο ότι δεν χρειαζόμαστε μεγάλη ακρίβεια στο σχεδιασμό χωρητικότητας, το ενδιαφέρον μας εστιάζεται κυρίως στην αποκάλυψη των πραγματικών *τάσεων* στην απόδοση του συστήματος και στη σωστή *ανάλυση και εξήγηση* των αποτελεσμάτων.

Παράδειγμα 2.4

Στο Σχήμα 2.5 φαίνεται μια τυπική έξοδος (σύνολο αποτελεσμάτων) κάποιας τεχνικής εκτίμησης της απόδοσης. Η γραφική παράσταση του Σχήματος 2.5 παρουσιάζει το χρόνο απόκρισης ενός συστήματος επεξεργασίας διεργασιών (transaction processing system), ως συνάρτηση του μεγέθους του φορτίου εργασίας, για διάφορες διατάξεις του συστήματος. Κάθε σενάριο αντιστοιχεί σε μια συγκεκριμένη πρόβλεψη για το φορτίο εργασίας του συστήματος κατά το επόμενο έτος. Ας υποθέσουμε ότι το Σενάριο 2 το οποίο προβλέπει 15 TPS (διεργασίες ανά δευτερόλεπτο), είναι το πιο πιθανό. Παίρνουμε υπόψη και δύο εναλλακτικά σενάρια: Το Σενάριο 1 που αντιστοιχεί σε μια απαισιόδοξη εκτίμηση του φορτίου εργασίας στο 60% του πιθανότερου σεναρίου, δηλαδή 9 TPS και το Σενάριο 3 που αντιστοιχεί σε μια αισιόδοξη πρόβλεψη στο 160% της πιθανότερης τιμής, δηλαδή 24 TPS. Η διακεκομμένη οριζόντια γραμμή υποδεικνύει το αποδεκτό όριο για το χρόνο απόκρισης των διεργασιών. Όπως βλέπουμε, η Διάταξη Γ έχει την καλύτερη απόδοση, εμφανίζοντας χρόνους απόκρισης πάντα μέσα στα αποδεκτά όρια. Από την άλλη πλευρά η Διάταξη Α είναι απαράδεκτη αφού δεν βρίσκεται μέσα στα αποδεκτά όρια για κανένα από τα πιθανά σενάρια φορτίου εργασίας.



Σχήμα 2.5. Αποτελέσματα εκτίμησης της απόδοσης συστήματος επεξεργασίας διεργασιών.

Η επιλογή βρίσκεται μεταξύ των Διατάξεων Β και Γ και γίνεται παίρνοντας υπόψη το κόστος κάθε διάταξης. Η τελευταία σύγκριση μας οδηγεί στο τελευταίο Βήμα 5 του σχεδιασμού χωρητικότητας, όπως περιγράφεται στην προηγούμενη υποενότητα, δηλαδή στην υιοθέτηση της επιλογής με το μικρότερο λόγο κόστους/απόδοσης.

2.5 ΣΧΕΔΙΑΣΜΟΣ ΠΕΙΡΑΜΑΤΩΝ

Ο βασικός *στόχος* ενός σωστού σχεδιασμού των πειραμάτων μέτρησης ή προσομοίωσης σε μια μελέτη απόδοσης, είναι η εξασφάλιση της μέγιστης δυνατής πληροφορίας από τον ελάχιστο δυνατό αριθμό πειραμάτων.

Στη συνέχεια, η κατάλληλη ανάλυση των πειραμάτων θα βοηθήσει στο διαχωρισμό των επιπτώσεων διαφόρων *παραγόντων* που μπορούν να επηρεάσουν την απόδοση του συστήματος, καθώς και στη διακρίβωση αν ένας παράγοντας έχει σημαντική επίδραση ή αν οι παρατηρούμενες διαφορές στην τιμή του οφείλονται σε τυχαίες διακυμάνσεις λόγω μετρητικών λαθών και της παρουσίας παραμέτρων που δεν ελέγχονται.

Ο **σχεδιασμός πειραμάτων** ασχολείται με τον προσδιορισμό του αριθμού των διαφορετικών πειραμάτων, των συνδυασμών των *επιπέδων* των παραγόντων για κάθε πείραμα και των αριθμών επαναλήψεων για κάθε πείραμα.

Παράδειγμα 2.5

Ας υποθέσουμε ότι έχουμε μια μελέτη για το σχεδιασμό ενός νέου σταθμού εργασίας, στην οποία έχουμε προσδιορίσει πέντε παράγοντες: τον τύπο της CPU, το μέγεθος της κεντρικής μνήμης, τον αριθμό των οδηγών δίσκων, το φορτίο εργασίας και το επίπεδο εκπαίδευσης των χρηστών. Ο πρώτος παράγοντας, δηλαδή ο τύπος της CPU έχει τρία επίπεδα: Pentium III, UltraSPARC III και R12000 RISC. Το μέγεθος της μνήμης έχει τρία επίπεδα: 16 Mbytes, 32 Mbytes, 64 Mbytes. Ο αριθμός των οδηγών δίσκων έχει τέσσερα επίπεδα: 1, 2, 3, 4. Το φορτίο εργασίας έχει τρία επίπεδα: γραμματειακό, διοικητικό, επιστημονικό. Τέλος, οι χρήστες συγκαταλέγονται σε τρία επίπεδα εκπαίδευσης: απόφοιτοι λυκείου, πτυχιούχοι ΑΕΙ και ΤΕΙ, μεταπτυχιακοί φοιτητές και διδάκτορες. Ένας σχεδιασμός πειραμάτων θα μπορούσε να προδιαγράψει όλα τα πειράματα που αντιστοιχούν σε όλους τους πιθανούς συνδυασμούς των επιπέδων των πέντε παραγόντων. Δηλαδή, 3x3x4x3x3, ή 324 πειράματα. Αν αποφασίσουμε να επαναλάβουμε κάθε πείραμα πέντε φορές, θα έχουμε συνολικά 1620 παρατηρήσεις.

□

Για δύο παράγοντες Α και Β λέμε ότι **αλληλεπιδρούν**, αν η επίδραση ενός από τους δύο εξαρτάται από το επίπεδο του άλλου.

Για παράδειγμα, ας υποθέσουμε ότι οι παράγοντες Α και Β ενός συστήματος, έχουν από δύο επίπεδα A_1, A_2 και B_1, B_2 αντίστοιχα. Αν παρατηρήσουμε ότι μια μετρική του συστήματος αυξάνεται ή μειώνεται κατά σταθερό μέγεθος όταν αλλάζει το επίπεδο του Α από A_1 σε A_2 ανεξάρτητα από το επίπεδο του Β, μπορούμε να συμπεράνουμε ότι δεν έχουμε αλληλεπίδραση μεταξύ των παραγόντων Α και Β. Αντίθετα, αν παρατηρούμε διαφορετική συμπεριφορά στην αύξηση (ή μείωση) της τιμής της μετρικής ανάλογα με το επίπεδο του Β, τότε οι Α και Β αλληλεπιδρούν. Στον Πίνακα 2.3 φαίνεται η απόδοση του συστήματος σε δύο τέτοιες περιπτώσεις.

	A. Οι παράγοντες δεν αλληλεπιδρούν		**B. Οι παράγοντες αλληλεπιδρούν**	
	A_1	A_2	A_1	A_2
B_1	3	5	3	5
B_2	6	8	6	9

Πίνακας 2.3. Απόδοση συστήματος με και χωρίς αλληλεπίδραση των παραγόντων.

2.5.1 Μέθοδοι Σχεδιασμού Πειραμάτων

Υπάρχουν αρκετές μέθοδοι σχεδιασμού πειραμάτων. Οι τρεις πιο δημοφιλείς είναι ο *απλός* σχεδιασμός, οι *πλήρως παραγοντικοί* σχεδιασμοί και οι *κλασματικοί παραγοντικοί* σχεδιασμοί. Στη συνέχεια παρουσιάζονται με συντομία οι τρεις αυτές μέθοδοι.

1. Απλός Σχεδιασμός.

Στη μέθοδο αυτή αρχίζουμε με μια τυπική διάταξη του συστήματος και αλλάζουμε την τιμή ενός παράγοντα τη φορά, με στόχο να δούμε πώς ο παράγοντας αυτός επηρεάζει την απόδοση του συστήματος.

Παράδειγμα 2.5 (Συνέχεια)

Στο παράδειγμα του σχεδιασμού ενός σταθμού εργασίας, μια τυπική διάταξη θα μπορούσε να αποτελείται από μια R12000 CPU, 32MB μνήμη, με δύο οδηγούς δίσκων που χρησιμοποιείται για διοικητικές εργασίες από πτυχιούχους AEI. Υπολογίζουμε την απόδοση της διάταξης αυτής πρώτα. Στη συνέχεια μεταβάλλουμε την τιμή του πρώτου παράγοντα (CPU) και η απόδοση συγκρίνεται με διατάξεις που έχουν άλλες CPU με ίδια τα υπόλοιπα στοιχεία. Αυτό θα μας βοηθήσει να αποφασίσουμε ποια CPU είναι προτιμότερη. Μετά αλλάζουμε τον αριθμό των οδηγών δίσκων σε ένα, τρία και τέσσερα, συγκρίνοντας την απόδοση έτσι ώστε να βρούμε το βέλτιστο αριθμό.

□

Αν έχουμε k παράγοντες, με τον i-στό παράγοντα να έχει n_i επίπεδα, ένας απλός σχεδιασμός θα απαιτούσε n πειράματα, όπου

$$n = 1 + \sum_{i=1}^{k} (n_i - 1)$$

Στην παραπάνω σχέση το 1 αντιστοιχεί στην αρχική τυπική διάταξη, ενώ κάθε όρος του αθροίσματος αντιστοιχεί στις διατάξεις που προκύπτουν από τη μεταβολή των επιπέδων του αντίστοιχου παράγοντα σε σχέση με την αρχική τυπική διάταξη.

Ο σχεδιασμός αυτός δεν κάνει την καλύτερη δυνατή χρήση της προεργασίας που έχει γίνει. Δεν είναι στατιστικά αποδοτικός, ενώ δεν αντιμετωπίζει ικανοποιητικά την περίπτωση της αλληλεπίδρασης των παραγόντων. Για παράδειγμα, αν η επίδραση της CPU εξαρτάται από το μέγεθος της μνήμης, ο βέλτιστος συνδυασμός των δύο παραγόντων δεν είναι εύκολο να προσδιορισθεί μέχρι να δοκιμασθούν όλοι οι πιθανοί συνδυασμοί.

2. Πλήρως Παραγοντικός Σχεδιασμός.

Ένας πλήρως παραγοντικός σχεδιασμός αξιοποιεί όλους τους δυνατούς συνδυασμούς επιπέδων των παραγόντων. Μία μελέτη απόδοσης με k παράγοντες, με τον i-στό παράγοντα να έχει n_i επίπεδα, απαιτεί n πειράματα, όπου

$$n = \prod_{i=1}^{k} n_i$$

Παράδειγμα 2.5 (Συνέχεια)

Στο παράδειγμα σχεδιασμού του σταθμού εργασίας, ο αριθμός των πειραμάτων θα είναι:

n = (3 CPU)(3 επίπεδα μνήμης)(4 οδηγοί δίσκων) x (3 φορτία εργασίας)(3 επίπ. εκπαίδ.)
= 324 πειράματα

□

Το βασικό πλεονέκτημα του πλήρως παραγοντικού σχεδιασμού είναι ότι λαμβάνονται υπόψη όλοι οι συνδυασμοί διατάξεων και φορτίων εργασίας. Το μειονέκτημα είναι το μεγάλο κόστος της μελέτης, ιδιαίτερα αν σκεφθούμε ότι χρειάζονται περισσότερες από μία επαναλήψεις για κάθε πείραμα. Υπάρχουν τρεις τρόποι να μειώσουμε τον αριθμό των πειραμάτων:

- Να μειώσουμε τον αριθμό των επιπέδων κάθε παράγοντα.
- Να μειώσουμε τον αριθμό των παραγόντων.
- Να χρησιμοποιήσουμε κλασματικό παραγοντικό σχεδιασμό.

Ιδιαίτερα δημοφιλής είναι η πρώτη επιλογή. Μπορούμε, για παράδειγμα, να χρησιμοποιήσουμε μόνο δύο βασικά επίπεδα για κάθε παράγοντα (μεγάλο και μικρό) και έτσι να προσδιορίσουμε τη *σχετική* επίδραση κάθε παράγοντα. Αν έχουμε k παράγοντες, τους οποίους χρησιμοποιούμε με δύο επίπεδα τον καθένα, τότε θα απαιτηθούν 2^k πειράματα. Ο τελευταίος αυτός σχεδιασμός χρησιμοποιείται πολύ συχνά και ονομάζεται 2^k *παραγοντικός σχεδιασμός*.

3. Κλασματικός Παραγοντικός Σχεδιασμός.

Μερικές φορές ο αριθμός των πειραμάτων που απαιτούνται για έναν πλήρως παραγοντικό σχεδιασμό, είναι πολύ μεγάλος και κατά συνέπεια ασύμφορος, είτε λόγω του μεγάλου αριθμού παραγόντων της μελέτης, ή λόγω του μεγάλου αριθμού επιπέδων κάθε παράγοντα. Στις περιπτώσεις αυτές μπορούμε να χρησιμοποιήσουμε ένα κλάσμα μόνο του πλήρως παραγοντικού σχεδιασμού.

Παράδειγμα 2.5 (Συνέχεια)

Στην εφαρμογή του σχεδιασμού του σταθμού εργασίας, μπορούμε να πάρουμε υπόψη μόνο τέσσερις από τους πέντε παράγοντες, αγνοώντας για παράδειγμα, τον αριθμό των οδηγών δίσκων. Θα έχουμε έτσι τέσσερις παράγοντες, με τρία επίπεδα ο καθένας. Συνεπώς θα απαιτούνται

n = (3 CPU)(3 επίπεδα μνήμης)(3 φορτία εργασίας)(3 επίπ. εκπαίδευσης) = 81 πειράματα

Δηλαδή, ο σχεδιασμός αυτός απαιτεί 81 πειράματα και ονομάζεται 3^4 κλασματικός παραγοντικός σχεδιασμός, αφού έχουμε 4 παράγοντες με 3 επίπεδα ο καθένας. Στον Πίνακα 2.4 φαίνεται ένας 3^{4-2} κλασματικός παραγοντικός σχεδιασμός για το παράδειγμά μας, ο οποίος αποτελείται μόνο από εννέα πειράματα. Μπορείτε να παρατηρήσετε ότι κάθε ένας από τους τέσσερις παράγοντες χρησιμοποιείται από τρεις φορές για κάθε ένα από τα τρία επίπεδά του.

Αριθμός Πειράματος	CPU	Επίπεδο Μνήμης	Τύπος Φορτίου Εργασίας	Επίπεδο Εκπαίδευσης
1	SPARC	16M	Διοικητικό	Λύκειο
2	SPARC	32M	Επιστημονικό	Μεταπτυχιακό
3	SPARC	64M	Γραμματειακό	ΑΕΙ – ΤΕΙ
4	R12000	16M	Επιστημονικό	ΑΕΙ – ΤΕΙ
5	R12000	32M	Γραμματειακό	Λύκειο
6	R12000	64M	Διοικητικό	Μεταπτυχιακό
7	Pentium	16M	Γραμματειακό	Μεταπτυχιακό
8	Pentium	32M	Διοικητικό	ΑΕΙ – ΤΕΙ
9	Pentium	64M	Επιστημονικό	Λύκειο

Πίνακας 2.4. Κλασματικός Παραγοντικός Σχεδιασμός

Είναι προφανές ότι ο κλασματικός παραγοντικός σχεδιασμός μας εξοικονομεί χρόνο και άλλους πόρους, συγκρινόμενος με τον πλήρως παραγοντικό σχεδιασμό. Όμως, αυτό το πληρώνουμε σε πληροφορία, η οποία είναι μικρότερη στην περίπτωση αυτή. Για παράδειγμα, είναι πιθανό να μην εντοπισθούν όλες οι αλληλεπιδράσεις μεταξύ των παραγόντων. Από την άλλη πλευρά, αν γνωρίζουμε ότι ορισμένες αλληλεπιδράσεις είναι αμελητέες, μπορούμε χωρίς ιδιαίτερο πρόβλημα να τις αγνοήσουμε και να προχωρήσουμε σε κλασματικό παραγοντικό σχεδιασμό.

ΚΕΦΑΛΑΙΟ 3
ΜΟΝΤΕΛΑ ΘΕΩΡΙΑΣ ΑΝΑΜΟΝΗΣ

3.1 ΕΙΣΑΓΩΓΗ ΣΤΗ ΘΕΩΡΙΑ ΑΝΑΜΟΝΗΣ

Τα *συστήματα αναμονής* (queueing systems) βρίσκονται πίσω από τα περισσότερα μοντέλα μελέτης της απόδοσης υπολογιστικών συστημάτων, αφού φαινόμενα καθυστερήσεων λόγω της απαίτησης χρήσης περιορισμένων πόρων από πολλούς "πελάτες" (αιτήσεις), μπορούν να εντοπισθούν τόσο σε απλούς υπολογιστές (π.χ. CPU, I/O, μνήμη), όσο και σε πολύπλοκα συστήματα, όπως τα δίκτυα υπολογιστών (π.χ. κανάλια μετάδοσης δεδομένων). Τα συστήματα αναμονής αναφέρονται συχνά και ως *συστήματα ουρών*.

Κάθε σύστημα στο οποίο αφίξεις «πελατών» δημιουργούν απαιτήσεις εξυπηρέτησης από πόρους πεπερασμένης δυνατότητας εξυπηρέτησης, είναι ένα **σύστημα αναμονής.**

Όταν οι χρονικές στιγμές αφίξεων ή το μέγεθός τους δεν είναι δυνατόν να προβλεφθούν, τότε θα εμφανιστούν απαιτήσεις συγχρόνου χρησιμοποίησης των πόρων και θα σχηματισθούν ουρές πελατών που περιμένουν. Με δεδομένη τη μέγιστη δυνατότητα εξυπηρέτησης του συστήματος, δύο μεγέθη επηρεάζουν τα μήκη αυτών των ουρών: Η *μέση τιμή* και η *στατιστική διακύμανση* του *ρυθμού αφίξεων*. Φυσικά όταν η μέση τιμή του ρυθμού αφίξεων ξεπερνά τη μέγιστη δυνατότητα εξυπηρέτησης, το σύστημα «πέφτει» και ουρές ανεξέλεγκτου μήκους αρχίζουν να σχηματίζονται. Πάντως, ακόμα και στην περίπτωση που η μέση τιμή του ρυθμού αφίξεων είναι μικρότερη από τη μέγιστη δυνατότητα εξυπηρέτησης, θα έχουμε το σχηματισμό ουρών λόγω των στατιστικών διακυμάνσεων και των «ξεσπασμάτων» που μπορεί να έχουν οι αφίξεις.

Τα παραπάνω φαινόμενα αναμονής τα αντιμετωπίζουμε όλοι μας κατά τη διάρκεια των καθημερινών συναλλαγών μας σε τράπεζες, υπηρεσίες, διόδια κ.λ.π., όπου «πελάτες» είμαστε εμείς και πόροι του συστήματος είναι συνήθως οι υπάλληλοι που μας εξυπηρετούν. Σε ένα πληροφοριακό σύστημα φανταστείτε ως «πελάτες» τις αιτήσεις χρήσης μιας CPU από ένα πρόγραμμα, τα πακέτα δεδομένων σε ένα δίκτυο και αντίστοιχα ως πόρους του συστήματος το χρόνο της CPU, το κανάλι μετάδοσης των πακέτων κ.λ.π.

Προκειμένου να μελετήσουμε ένα σύστημα αναμονής, θα χρησιμοποιήσουμε *μοντέλα αναμονής* (queueing models), τα οποία είναι *μαθηματικά* μοντέλα. Το παρόν Κεφάλαιο θα ασχοληθεί με την *αναλυτική λύση* των μοντέλων αναμονής. Τα θεωρητικά εργαλεία που χρησιμοποιούνται για το σκοπό αυτό, συγκροτούν τη *θεωρία αναμονής* (queueing theory).

Η **θεωρία αναμονής** ασχολείται με τη μελέτη συστημάτων, η απόδοση των οποίων επηρεάζεται από φαινόμενα αναμονής.

3.1.1 Ορισμοί, Μετρικές και Συμβολισμοί των Συστημάτων Αναμονής

1. Φορτίο εργασίας ή Δεδομένα εισόδου.

Προκειμένου να ορίσουμε πλήρως το φορτίο εργασίας στο σύστημα αναμονής του Σχήματος 3.1, πρέπει να προσδιορίσουμε τις *Στοχαστικές Διαδικασίες* (Σ.Δ.) που περιγράφουν το ρεύμα αφίξεων, καθώς και τη μορφή της εξυπηρέτησης. Είναι προφανές ότι εδώ αναφερόμαστε σε *μη-εκτελέσιμο συνθετικό φορτίο εργασίας*.

Γενικά η διαδικασία αφίξεων περιγράφεται από τη *Συνάρτηση Κατανομής Πιθανότητας* (ΣΚΠ) των *χρόνων μεταξύ διαδοχικών αφίξεων* (Χ.Α.) $A(t)$, όπου:

$$A(t) = \text{Prob}[\text{χρόνος μεταξύ διαδοχικών αφίξεων} \leq t]$$

Η υπόθεση στο μεγαλύτερο τμήμα της θεωρίας ουρών είναι ότι αυτοί οι Χ.Α. είναι ανεξάρτητες, όμοια κατανεμημένες *Τυχαίες Μεταβλητές* (Τ.Μ.).

Η δεύτερη στατιστική ποσότητα που πρέπει να περιγραφεί είναι το ποσόν της εξυπηρέτησης που επιβάλλουν αυτές οι αφίξεις στο σύστημα. Αυτό αναφέρεται συνήθως σαν *χρόνος εξυπηρέτησης* (Χ.Ε.) με ΣΚΠ την $B(x)$:

$$B(x) = \text{Prob}[\text{χρόνος εξυπηρέτησης} \leq x]$$

Δηλαδή ο Χ.Ε. αναφέρεται στο χρόνο που ένας πελάτης απασχολεί έναν εξυπηρετητή.

2. Άλλα μεγέθη περιγραφής της δομής και των χαρακτηριστικών του συστήματος.

Εξετάζοντας τη δομή του συστήματος και τα χαρακτηριστικά των πελατών, πρέπει να γνωρίζουμε ή να υποθέσουμε και μια σειρά άλλες ποσότητες, όπως:

- *Αριθμός εξυπηρετητών* (servers) στο σύστημα m.
- *Χωρητικότητα* του συστήματος σε πελάτες K. (Συνήθως υποθέτουμε $K = \infty$).
- *Πληθυσμός* υποψηφίων πελατών M. (Συνήθως υποθέτουμε $M = \infty$).
- *Πολιτική εξυπηρέτησης*, δηλαδή ο τρόπος επιλογής πελατών από την ουρά για τον (τους) εξυπηρετητές. (Συνήθως υποθέτουμε FCFS, δηλ. First Come - First Serve, πολιτική στην οποία επιλέγεται πάντα ο πελάτης που περιμένει περισσότερο).
- *Κλάσεις* πελατών. (Συνήθως υποθέτουμε μόνο μία)
- *Ομάδες προτεραιότητας* πελατών. (Συνήθως υποθέτουμε μόνο μία)
- *Διαθεσιμότητα* εξυπηρετητή. (Συνήθως υποθέτουμε 100%)

3. Μετρικές απόδοσης.

Οι ποσότητες που ενδιαφέρουν από πλευράς μέτρησης της απόδοσης και της αποτελεσματικότητας του συστήματος, είναι :

- *Χρόνος απόκρισης* (συνολικός χρόνος στο σύστημα) για ένα πελάτη.
- *Χρόνος αναμονής* για ένα πελάτη.
- *Αριθμός πελατών* στο σύστημα.
- Το μήκος μιας *περιόδου απασχόλησης* (busy period).
- Το μήκος μιας *περιόδου αργίας* (idle period).

Όλες αυτές οι τελευταίες ποσότητες είναι Σ.Δ. και πιθανώς ψάχνουμε την πλήρη περιγραφή τους (π.χ. την ΣΚΠ). Συνήθως όμως μας αρκούν μόνο οι λίγες πρώτες ροπές τους (μέση τιμή, διασπορά κ.λ.π.).

4. Συμβολισμός συστημάτων αναμονής.

Ένα Σύστημα Αναμονής συμβολίζεται σαν *A/B/m* που καθορίζει ένα σύστημα με *m* εξυπηρετητές, ενώ *A* και *B* περιγράφουν τις ΣΚΠ των Χ.Α. και Χ.Ε. αντίστοιχα.

Τα *A* και *B* εμφανίζονται στην παραπάνω έκφραση με τις εξής μορφές :

- *M* (για την εκθετική κατανομή).
- *D* (για τη ντετερμινιστική [σταθερή] κατανομή).
- E_r (για την κατανομή Erlang r-βαθμίδων).
- *G* (για ΟΠΟΙΑΔΗΠΟΤΕ ΚΑΤΑΝΟΜΗ). Ο συμβολισμός *G* χρησιμοποιείται όταν αναφερόμαστε σε κάποιο χαρακτηριστικό το οποίο ισχύει για οποιαδήποτε μορφή της κατανομής, ή όταν η κατανομή αυτή δεν μας είναι γνωστή.

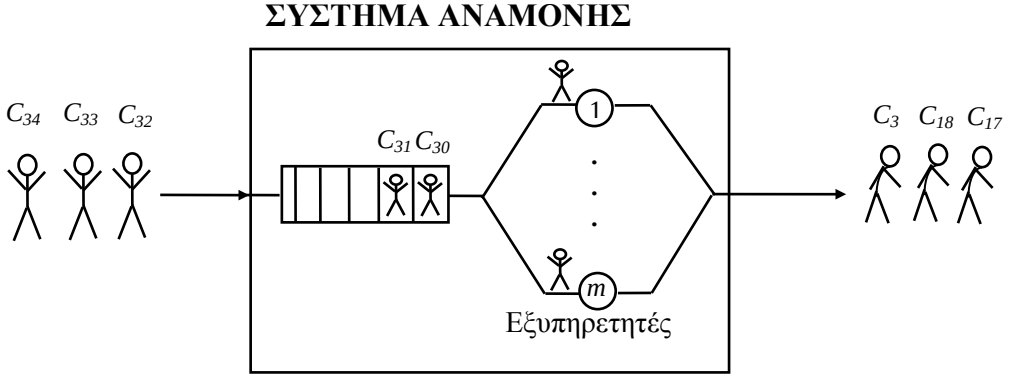
Πολλές φορές χρειάζεται να καθορίσουμε τη χωρητικότητα του συστήματος *K* ή το μέγεθος του πληθυσμού των πελατών *M*, όταν αυτά είναι διαφορετικά από ∞. Στην περίπτωση αυτή ο συμβολισμός μας γίνεται *A/B/m/K/M*. Αν λείπει ένα τουλάχιστον απ' τα δύο τελευταία τμήματα, τότε τα υποθέτουμε ίσα με ∞.

Για παράδειγμα, το σύστημα *D/M/2//200* είναι ένα σύστημα δύο εξυπηρετητών με σταθερούς (ντετερμινιστικούς) Χ.Α., με εκθετικά κατανεμημένους Χ.Ε., με άπειρη χωρητικότητα ουράς και διαθέσιμο πληθυσμό πελατών ίσο με 200.

5. Ορισμοί και συμβολισμοί των βασικών μεγεθών.

Ξεκινάμε τη μελέτη μας έχοντας υπόψη ένα πολύ γενικό σύστημα *G/G/m*, δηλαδή ένα σύστημα στο οποίο τα *A(t)*, *B(x)* είναι τελείως αυθαίρετα (όλοι οι Χ.Α. και οι Χ.Ε. υποτίθενται ανεξάρτητοι μεταξύ τους). Το σύστημα έχει *m* εξυπηρετητές και η σειρά εξυπηρέτησης είναι επίσης σχετικά αυθαίρετη (συγκεκριμένα, δεν χρειάζεται να είναι FCFS). Συγκεντρώνουμε την προσοχή μας στη ροή των πελατών καθώς φθάνουν, περνούν και τελικά φεύγουν από το σύστημα.

Αριθμούμε τους πελάτες με το δείκτη *n* και ορίζουμε το C_n να συμβολίζει τον *n*-οστό πελάτη που εισέρχεται στο σύστημα. Το σύστημα φαίνεται σχηματικά στο Σχήμα 3.1.



Σχήμα 3.1. Ένα σύστημα αναμονής.

Μπορούμε αμέσως να ορίσουμε μερικές Σ.Δ. που μας ενδιαφέρουν. Για παράδειγμα, μας ενδιαφέρει το $N(t) \equiv \{$*Αριθμός πελατών στο σύστημα τη στιγμή t*$\}$ και το $U(t) \equiv \{$*η ατελείωτη δουλειά μέσα στο σύστημα τη στιγμή t*$\}$. Το $U(t)$ ουσιαστικά είναι ο χρόνος που μένει για να αδειάσει το σύστημα από όλους τους πελάτες που είναι παρόντες τη στιγμή t. Όταν $U(t) > 0$, το σύστημα λέγεται *απασχολημένο* (busy) και μόνο όταν $U(t) = 0$ το σύστημα λέγεται *άεργο* (idle). Η διάρκεια και η θέση (στο χρόνο) αυτών των περιόδων απασχόλησης και αργίας είναι επίσης ποσότητες που ενδιαφέρουν.

Ορίζουμε τα παρακάτω μεγέθη:

Α. Για τους χρόνους μεταξύ διαδοχικών αφίξεων:

$\tau_n \equiv$ χρονική στιγμή άφιξης του C_n

$t_n \equiv$ χρόνος μεταξύ των αφίξεων των C_{n-1}, C_n

Δηλαδή: $t_n \equiv \tau_n - \tau_{n-1}$, όπου $n \geq 2$ και $t_1 = \tau_1$.

Μια και έχουμε υποθέσει ότι *όλοι* οι χρόνοι μεταξύ διαδοχικών αφίξεων (Χ.Α.) έχουν την ίδια ΣΚΠ $A(t)$, ισχύει : $\text{Prob}[\, t_n \leq t\,] = A(t)$ που είναι σχέση ανεξάρτητη του n.

Ο *μέσος χρόνος μεταξύ διαδοχικών αφίξεων* συμβολίζεται $\bar{t}$.

Ο *ρυθμός αφίξεων* (arrival rate) των πελατών συμβολίζεται με λ και ισχύει: $\lambda = \dfrac{1}{\bar{t}}$

Β. Για τους χρόνους εξυπηρέτησης:

$x_n \equiv$ χρόνος εξυπηρέτησης του C_n

Από τις υποθέσεις μας, έχουμε : $\text{Prob}[\, x_n \leq x\,] = B(x)$.

Ο *μέσος χρόνος εξυπηρέτησης* συμβολίζεται $\bar{x}$.

Ο *ρυθμός εξυπηρέτησης* (service rate) των πελατών συμβολίζεται με μ και $\mu = \dfrac{1}{\bar{x}}$

Οι ακολουθίες $\{t_n\}$ και $\{x_n\}$ θεωρούνται *μεταβλητές εισόδου* (input variables) για το σύστημα αναμονής. Ο τρόπος με τον οποίο το σύστημα χειρίζεται τους πελάτες αυτούς, δημιουργεί ουρές και χρόνους αναμονής οι οποίοι είναι *μεταβλητές εξόδου* (output variables) που πρέπει να ορίσουμε :

Γ. Για το χρόνο αναμονής ενός πελάτη στην ουρά:

$w_n \equiv$ *χρόνος αναμονής* (στην ουρά) του C_n.

Ο *μέσος χρόνος αναμονής* εμφανίζεται συνήθως με το συμβολισμό $W \equiv \bar{w}$.

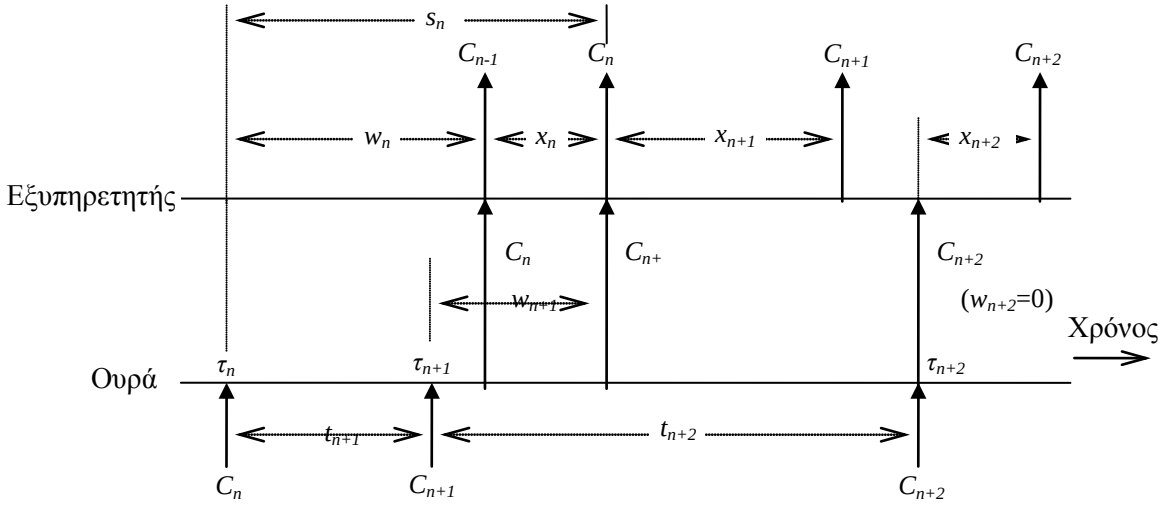
Δ. Για το συνολικό χρόνο ενός πελάτη στο σύστημα (χρόνο απόκρισης):

Ο συνολικός *χρόνος συστήματος* s_n, που δαπανά μέσα στο σύστημα ο C_n, είναι το άθροισμα του χρόνου αναμονής w_n και του χρόνου εξυπηρέτησής του x_n:

$s_n \equiv$ χρόνος συστήματος (ουρά + εξυπηρέτηση) του $C_n = w_n + x_n$

Ο *μέσος χρόνος συστήματος* εμφανίζεται συνήθως με το συμβολισμό $T \equiv \bar{s}$. Φυσικά ισχύει πάντοτε $T = W + \bar{x}$.

Στο Σχήμα 3.2 φαίνεται ένα *χρονικό διάγραμμα* για ένα σύστημα αναμονής, που επιτρέπει να δούμε τη δυναμική του συστήματος, ενώ παράλληλα μας δίνει λεπτομέρειες για τις διάφορες Σ.Δ. Το σχήμα δείχνει ένα σύστημα FCFS με έναν εξυπηρετητή, αλλά εύκολα θα μπορούσε να μετατραπεί ώστε να δέχεται οποιαδήποτε σειρά εξυπηρέτησης ή/και περισσότερους εξυπηρετητές.



Σχήμα 3.2. Χρονικό διάγραμμα ενός συστήματος αναμονής.

3.1.2 Νόμος του Little

Σε ένα γενικό σύστημα αναμονής, περιμένουμε πως όταν ο αριθμός των πελατών είναι μεγάλος, το ίδιο θα πρέπει να ισχύει και για το χρόνο αναμονής.

Η παραπάνω παρατήρηση εκφράζεται με μια πολύ απλή σχέση μεταξύ του *μέσου αριθμού πελατών στο σύστημα* $\overline{N}$, του μέσου ρυθμού αφίξεων πελατών λ και του μέσου χρόνου συστήματος των πελατών T:

$$\overline{N} = \lambda \cdot T \tag{3.1}$$

Η τελευταία εξίσωση, γνωστή σαν *Νόμος του Little*, δηλώνει ότι: «Ο μέσος αριθμός πελατών σε ένα σύστημα αναμονής είναι ίσος με το μέσο ρυθμό αφίξεων πελατών στο σύστημα επί το μέσο χρόνο που ξοδεύει ένας πελάτης σ' αυτό».

Μία διαισθητική απόδειξη του Νόμου του Little για σύστημα FCFS βγαίνει από την παρατήρηση ότι ένας πελάτης που φθάνει στο σύστημα θα *βρει* μέσα κατά μέσο όρο τον ίδιο αριθμό πελατών $\overline{N}$ που θα *υπάρχει* όταν φύγει. Όμως κατά το διάστημα της παρουσίας του ήρθαν $\lambda \cdot T$ πελάτες κατά μέσο όρο. Η τελευταία ποσότητα είναι οι $\overline{N}$ πελάτες που αφήνει πίσω φεύγοντας.

Ο Νόμος του Little ισχύει για σύστημα *G/G/m* και με οποιαδήποτε σειρά εξυπηρέτησης, δηλαδή για την πιο γενική περίπτωση. Με τον ίδιο τρόπο μπορούμε να πάρουμε και άλλες μορφές του Νόμου του Little, μια και δεν έπαιξαν κανένα ρόλο κατά την απόδειξή μας τα όρια του συστήματος και η ακριβής περιγραφή του.

Αν, δηλαδή, ορίσουμε τα όρια του συστήματος με τρόπο ώστε να περιλαμβάνεται σ' αυτό μόνο η ουρά, ο Νόμος του Little δίνει:

$$\overline{N}_q = \lambda \cdot W \tag{3.2}$$

όπου $\overline{N}_q$ ο *μέσος αριθμός πελατών στην ουρά* και W ο μέσος χρόνος αναμονής (στην ουρά).

Αν, επίσης, τα όρια του συστήματος περιλαμβάνουν μόνο τον εξυπηρετητή (ή τους εξυπηρετητές), ισχύει:

$$\overline{N}_s = \lambda \cdot \overline{x} \tag{3.3}$$

όπου $\overline{N}_s$ ο *μέσος αριθμός πελατών στον εξυπηρετητή* (ή στους εξυπηρετητές) και $\overline{x}$ ο μέσος χρόνος εξυπηρέτησης ενός πελάτη.

Πρέπει να παρατηρήσουμε, πάντως, ότι ο Νόμος του Little, αν και δίνει μια χρήσιμη σχέση μεταξύ ορισμένων βασικών μεγεθών ενός συστήματος αναμονής, δεν αποτελεί «λύση» στο γενικό μας πρόβλημα: Ουσιαστικά συνδέει ένα γνωστό μέγεθος εισόδου (λ), με δύο άγνωστα μεγέθη ($\overline{N}$ και T) τα οποία είναι μετρικές απόδοσης που θέλουμε να βρούμε. Θα πρέπει να επιλύσουμε αναλυτικά το μοντέλο για να βρούμε ένα από τα δύο άγνωστα μεγέθη και στη συνέχεια να χρησιμοποιήσουμε το Νόμο του Little, αν βολεύει.

3.1.3 Συντελεστής Απασχόλησης ή Χρησιμοποίηση

Ο **συντελεστής απασχόλησης** ή **χρησιμοποίηση** ρ, ορίζεται ως ο λόγος του ρυθμού με τον οποίο εισέρχεται «δουλειά» στο σύστημα, προς το μέγιστο ρυθμό με τον οποίο το σύστημα μπορεί να εκτελέσει αυτή τη «δουλειά».

Η «δουλειά» που προσθέτει στο σύστημα ένας πελάτης που φθάνει σ' αυτό, είναι ίση με τον αριθμό των δευτερολέπτων εξυπηρέτησης που χρειάζεται. Συνεπώς, στην περίπτωση συστήματος με *έναν εξυπηρετητή*, ο ορισμός του ρ γίνεται:

$$\rho \equiv (\text{ μέσος ρυθμός άφιξης πελατών }) \times (\text{ μέσος χρόνος εξυπηρέτησης })/1$$

ή
$$\rho = \lambda \cdot \overline{x} \tag{3.4}$$

Το τελευταίο ισχύει αφού έχουμε έναν εξυπηρετητή και συνεπώς το σύστημα έχει μέγιστο ρυθμό εξυπηρέτησης ίσο με 1sec/1sec, ενώ κάθε πελάτης που φθάνει, φέρνει όγκο δουλειάς ίσο με $\overline{x}$ sec κατά μέσο όρο.

Στην περίπτωση συστήματος με m εξυπηρετητές ισχύει βέβαια:

$$\rho = \frac{\lambda \cdot \overline{x}}{m} \tag{3.5}$$

Οι δύο τελευταίες εξισώσεις ισχύουν στην περίπτωση που ο μέγιστος ρυθμός εξυπηρέτησης είναι ανεξάρτητος από την *κατάσταση* (αριθμό παρόντων πελατών) του συστήματος. Στην αντίθετη περίπτωση πρέπει να δοθεί ένας πιο προσεκτικός ορισμός. Ο ρυθμός εισόδου «δουλειάς» στο σύστημα αναφέρεται μερικές φορές ως *ένταση κυκλοφορίας* του συστήματος.

Σε σύστημα ενός εξυπηρετητή, ο συντελεστής απασχόλησης ισούται προς την ένταση κυκλοφορίας ενώ σε σύστημα m εξυπηρετητών η ένταση κυκλοφορίας ισούται με $m \cdot \rho$. Συνδυάζοντας τις σχέσεις (3.4) ή (3.5) με την (3.3) συμπεραίνουμε ότι:

$\rho =$ {Μέση τιμή του ποσοστού εξυπηρετητών που είναι απασχολημένοι}

Σταθερό σύστημα αναμονής, είναι αυτό στο οποίο δεν επιτρέπεται να δημιουργούνται ουρές ανεξέλεγκτου (άπειρου) μήκους. Δηλαδή η μέγιστη δυνατότητα εξυπηρέτησης του συστήματος υπερκαλύπτει τη μέση εισερχόμενη «δουλειά» στο σύστημα.

Για να είναι ένα σύστημα *G/G/1* σταθερό, πρέπει $0 \le \rho < 1$. Σπάνια, επιτρέπουμε την περίπτωση $\rho = 1$ (και ειδικά για το σύστημα *D/D/1*).

Στην περίπτωση αυτή ($0 \le \rho < 1$) έχουμε τους παρακάτω υπολογισμούς:

Έστω τ ένα αυθαίρετα μεγάλο χρονικό διάστημα. Κατά τη διάρκεια αυτού του διαστήματος περιμένουμε ο αριθμός των αφίξεων να είναι πολύ κοντά στην τιμή $\lambda \cdot \tau$. Επίσης, έστω p_0 η *πιθανότητα ο εξυπηρετητής να είναι άεργος* σε κάποιο τυχαία εκλεγμένο χρονικό διάστημα. Μπορούμε λοιπόν να πούμε ότι κατά τη διάρκεια του διαστήματος *τ*, ο εξυπηρετητής είναι απασχολημένος για $\tau - \tau \cdot p_0$ sec και άρα ο αριθμός των πελατών που εξυπηρετούνται στο χρονικό διάστημα *τ*, είναι περίπου $\frac{(\tau - \tau \cdot p_0)}{\bar{x}}$.

Μπορούμε να εξισώσουμε τώρα τον αριθμό των αφίξεων με τον αριθμό πελατών που εξυπηρετήθηκαν κατά το διάστημα αυτό, που δίνει για μεγάλο *τ*:

$\lambda \cdot \tau \cong \frac{(\tau - \tau \cdot p_0)}{\bar{x}}$ οπότε για $\tau \to \infty$, έχουμε: $\lambda \bar{x} = 1 - p_0$

Χρησιμοποιώντας την Εξ. (3.4) παίρνουμε για σύστημα *G/G/1*:

$$\rho = 1 - p_0 \qquad (3.6)$$

δηλαδή, το ρ είναι απλώς το ποσοστό του χρόνου που ο εξυπηρετητής είναι απασχολημένος. Το τελευταίο ενισχύει το αποτέλεσμα της Εξ. (3.3), στην οποία φαίνεται ότι το $\lambda \cdot \bar{x} = \rho$ είναι ίσο με το μέσο αριθμό πελατών που είναι μέσα στον εξυπηρετητή.

□

Στη συνέχεια θα αποδείξουμε ότι $\rho =$ {Μέση τιμή του ποσοστού εξυπηρετητών που είναι απασχολημένοι}.

Α. Για σύστημα με έναν εξυπηρετητή:

Από τις σχέσεις (3.3) και (3.4) ισχύει: $\rho = \bar{N}_s$, δηλαδή η χρησιμοποίηση ρ είναι ίση με το μέσο αριθμό πελατών στον εξυπηρετητή. Προφανώς ο μέσος αριθμός πελατών στον εξυπηρετητή είναι αριθμός μικρότερος από 1, αφού η μέση τιμή αναφέρεται σε ένα μεγάλο χρονικό διάστημα, το οποίο θα έχει και διαστήματα κατά τα οποία το σύστημα είναι άεργο. Και επειδή $\bar{N}_s = \bar{N}_s / 1$, όπου το 1 αντιπροσωπεύει τον αριθμό των εξυπηρετητών (και το μέγιστο αριθμό πελατών στον εξυπηρετητή), το ρ είναι ίσο και με το ποσοστό των εξυπηρετητών που είναι κατειλημμένοι.

Β. Για σύστημα με *m* εξυπηρετητές:

Από τις σχέσεις (3.3) και (3.5) ισχύει: $\rho \cdot m = \bar{N}_s$, δηλαδή η χρησιμοποίηση ρ είναι ίση με $\bar{N}_s / m$. Αφού το *m* εκφράζει το μέγιστο αριθμό πελατών στους εξυπηρετητές, με την ίδια λογική με παραπάνω, το ρ είναι ίσο με το ποσοστό των εξυπηρετητών που είναι κατειλημμένοι.

3.2 ΤΑΞΙΝΟΜΗΣΗ ΤΩΝ ΣΤΟΧΑΣΤΙΚΩΝ ΔΙΑΔΙΚΑΣΙΩΝ

Η *Στοχαστική Διαδικασία* (Σ.Δ.), ορίζεται ως μία οικογένεια Τυχαίων Μεταβλητών (Τ.Μ.), *X(t)*, όπου οι Τ.Μ. έχουν δεικτοδοτηθεί με τη χρονική παράμετρο *t*.

Η ταξινόμηση των Σ.Δ. εξαρτάται από 3 ποσότητες : το *χώρο καταστάσεων*, τη *χρονική παράμετρο* και τη *στατιστική σχέση* μεταξύ των Τ.Μ. *X(t)* για διαφορετικές τιμές της χρονικής παραμέτρου *t*. Ας εξετάσουμε κάθε μία από αυτές:

- Πρώτα για το χώρο καταστάσεων. «Το σύνολο των πιθανών τιμών (ή καταστάσεων) που μπορεί να πάρει η *X(t)* καλείται χώρος καταστάσεων». Αν αυτές οι τιμές είναι πεπερασμένες ή αριθμήσιμες, έχουμε μία Σ.Δ. *διακριτών-καταστάσεων*, που συχνά αναφέρεται σαν *αλυσίδα*. Ο χώρος καταστάσεων για μία αλυσίδα είναι συνήθως το σύνολο των ακεραίων {0,1,2,...}. Αντίθετα, αν η *X(t)* παίρνει τιμές από ένα πεπερασμένο ή άπειρο συνεχές διάστημα (ή σύνολο τέτοιων διαστημάτων), τότε έχουμε μία Σ.Δ. *συνεχών-καταστάσεων*.

- Όσον αφορά τη χρονική παράμετρο τώρα, αν οι *επιτρεπτές* χρονικές στιγμές που μπορεί να γίνει αλλαγή κατάστασης είναι πεπερασμένες ή μετρήσιμες, έχουμε μία Σ.Δ. *διακριτού-χρόνου*, ενώ αν οι αλλαγές αυτές μπορούν να εμφανιστούν οποτεδήποτε μέσα σε ένα πεπερασμένο ή άπειρο συνεχές διάστημα (ή σύνολο τέτοιων διαστημάτων) του χρονικού άξονα, τότε έχουμε μία Σ.Δ. *συνεχούς-χρόνου*. Στην πρώτη περίπτωση συνήθως γράφουμε X_n αντί για *X(t)*. Η X_n συχνά καλείται *στοχαστική ακολουθία* σε αντίθεση με την *X(t)* που καλείται *στοχαστική διαδικασία*.

- Η σημαντικότερη ποσότητα που διαφοροποιεί τις Σ.Δ. είναι η στατιστική σχέση μεταξύ των Τ.Μ. *X(t)* ή X_n και των άλλων μελών της ίδιας οικογένειας. Όπως έχουμε πει, πρέπει να προσδιορίσουμε την πλήρη από κοινού PDF πάνω στις Τ.Μ. $\vec{X} = [X(t_1), X(t_2), \ldots]$, δηλαδή την:

$$F_{\vec{X}}(\vec{x};\vec{t}) \equiv P\left[X(t_1) \le x_1, \ \ldots, X(t_n) \le x_n\right] \qquad (3.7)$$

για όλα τα $\vec{x} = (x_1, x_2, \ldots, x_n)$, $\vec{t} = (t_1, t_2, \ldots, t_n)$ και *n*.

Η δουλειά αυτή είναι σχεδόν απαγορευτική, αλλά μερικές Σ.Δ. επιτρέπουν μια απλούστερη περιγραφή. Σε κάθε περίπτωση πάντως, η συνάρτηση $F_{\vec{X}}(\vec{x};\vec{t})$ είναι αυτή που περιγράφει πραγματικά τις εξαρτήσεις μεταξύ των Τ.Μ. μιας Σ.Δ. Στα επόμενα, περιγράφουμε μερικούς συνήθεις τύπους Σ.Δ. που χαρακτηρίζονται από διαφορετικά είδη εξαρτήσεων μεταξύ των Τ.Μ. τους.

a) **<u>Στάσιμες Σ.Δ.</u>** Μία Σ.Δ. *X(t)* καλείται στάσιμη αν είναι αμετάβλητη στις ολισθήσεις στο χρόνο, για όλες τις τιμές των ορισμάτων της. Δηλαδή για οποιοδήποτε σταθερό τ, πρέπει:

$$F_{\vec{X}}(\vec{x};\vec{t}+\tau) = F_{\vec{X}}(\vec{x};\vec{t}) \qquad (3.8)$$

όπου $\vec{t} + \tau = (t_1 + \tau, t_2 + \tau, \ldots, t_n + \tau)$.

Στη *στάσιμης Σ.Δ. υπό την ευρεία έννοια*, πρέπει να ισχύει για την *X(t)*:

$E[X(t)]$ ανεξάρτητη του *t*, και

$E[X(t)X(t+\tau)]$ εξαρτάται *μόνο* από το *τ*.

Παρατηρήσαμε επίσης ότι όλες οι στάσιμες Σ.Δ. είναι στάσιμες υπό την ευρεία έννοια, αλλά όχι αντίστροφα.

b) **<u>Ανεξάρτητες Σ.Δ.</u>** είναι οι πιο απλές Σ.Δ. για τις οποίες ισχύει:

$$f_{\bar{X}}(\bar{x};\bar{t}) \equiv f_{X_1\ldots X_n}(x_1,\ldots,x_n;t_1,\ldots,t_n) = f_{X1}(x_1;t_1)\ldots f_{Xn}(x_n;t_n) \qquad (3.9)$$

Ουσιαστικά εδώ δεν έχουμε να κάνουμε με Σ.Δ. μια και δεν υπάρχει καμία δομή ή εξάρτηση των Τ.Μ. της.

c) **<u>Διαδικασίες Markov</u>:** Μία διαδικασία Markov με διακριτό χώρο καταστάσεων καλείται *αλυσίδα Markov διακριτού-χρόνου*. Είναι οι πιο εύκολες από πλευράς ορισμού και κατανόησης των αρχών τους. ΄Ενα σύνολο Τ.Μ. $\{X_n\}$ σχηματίζει μία αλυσίδα Markov, αν η πιθανότητα ότι η επόμενη τιμή (κατάσταση) θα είναι X_{n+1}, εξαρτάται μόνο από την παρούσα τιμή (κατάσταση) X_n και όχι από οποιαδήποτε προηγούμενη'. Δηλαδή η επιρροή ολόκληρου του παρελθόντος πάνω στο μέλλον της διαδικασίας, περιορίζεται στην τρέχουσα τιμή της.

Στην περίπτωση αλυσίδας Markov διακριτού-χρόνου, οι χρονικές στιγμές που ,εμφανίζονται αλλαγές κατάστασης, ορίζεται να είναι οι ακέραιοι *0,1,2,...,n,...* Αντίθετα, στην περίπτωση αλυσίδας Markov συνεχούς-χρόνου, οι αλλαγές κατάστασης μπορούν να γίνουν οποιαδήποτε χρονική στιγμή. Συνεπώς, οδηγούμαστε να εξετάσουμε την Τ.Μ. που περιγράφει πόσο χρόνο η Σ.Δ. παραμένει στην παρούσα διακριτή κατάστασή της, πριν μεταβεί σε κάποια άλλη. Η μορφή της διαδικασίας Markov, προσθέτει ένα μεγάλο περιορισμό όσον αφορά την κατανομή του χρόνου που η διαδικασία παραμένει σε μια δεδομένη κατάσταση. Στην πραγματικότητα, όπως θα δούμε αργότερα, αυτός ο 'χρόνος κατάστασης' *πρέπει* να είναι *εκθετικά* κατανεμημένος.

Παρόμοια, για αλυσίδα Markov διακριτού-χρόνου, η διαδικασία μπορεί να παραμείνει σε μια δεδομένη κατάσταση για χρόνο που είναι *γεωμετρικά* κατανεμημένος. Η εκθετική κατανομή είναι η μόνη συνεχής *PDF* που 'δεν θυμάται το παρελθόν' και η γεωμετρική η μόνη διακριτή *pmf* που επίσης 'δεν θυμάται το παρελθόν'. Αυτή η ιδιότητα της 'αμνησίας' χαρακτηρίζει όλες τις διαδικασίες Markov.

Αναλυτικά η ιδιότητα Markov γράφεται: (για *αλυσίδα* Markov)

$$P\left[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n, X(t_{n-1}) = x_{n-1},\ldots,X(t_1) = x_1\right] =$$
$$= P\left[X(t_{n+1}) = x_{n+1} \mid X(t_n) = x_n\right] \qquad (3.10)$$

όπου $t_1 < t_2 < \ldots < t_n < t_{n+1}$, ενώ τα x_i περιέχονται σε κάποιο διακριτό χώρο καταστάσεων.

d) **<u>Διαδικασίες Γεννήσεων - Θανάτων</u> :** Είναι μια πολύ σημαντική κλάση αλυσίδων Markov. Μπορούν να είναι είτε διακριτού είτε συνεχούς χρόνου διαδικασίες, στις οποίες η συνθήκη ορισμού είναι:

'Οι αλλαγές κατάστασης γίνονται μόνο μεταξύ *γειτονικών* καταστάσεων'

Δηλαδή αν $X(t_n) = i$, τότε $X(t_{n+1}) = i - 1$ ή $X(t_{n+1}) = i + 1$ μόνο. (Το σύνολο καταστάσεων εδώ ορίζεται σαν το σύνολο των ακεραίων, χωρίς απώλεια της γενικότητας). Οι διαδικασίες γεννήσεων - θανάτων παίζουν τεράστιο ρόλο στη θεωρία ουρών ,όπως θα δούμε αργότερα.

e) **<u>Διαδικασίες Semi Markov</u> :** Θέλοντας να ξεπεράσουμε τον περιορισμό που μας επιβάλλει η ιδιότητα Markov στην κατανομή του «χρόνου κατάστασης» μιας Στοχαστικής Διαδικασίας Markov, δηλαδή να επιτρέψουμε αυθαίρετη κατανομή του

χρόνου που η Στοχαστική διαδικασία μπορεί να παραμείνει σε μια κατάσταση, εισάγουμε την έννοια της διαδικασίας Semi - Markov.

Πάντως και σε αυτήν την περίπτωση, η διαδικασία συμπεριφέρεται σαν μια Markov κατά τις χρονικές στιγμές αλλαγής κατάστασης, και στην πραγματικότητα *σε αυτές τις στιγμές* λέμε ότι έχουμε μια *συμπυκνωμένη (imbedded) αλυσίδα Markov*. Όπως είναι φανερό, η κλάση των διαδικασιών Markov περιέχεται μέσα στην κλάση των διαδικασιών Semi - Markov.

f) <u>Τυχαίοι περίπατοι</u> : Μια Στοχαστική διαδικασία λέγεται «τυχαίος περίπατος» όταν η επόμενη θέση (κατάσταση) που θα πάρει η Στοχαστική διαδικασία, ισούται με την προηγούμενη θέση, συν μια τυχαία μεταβλητή της οποίας την τιμή παίρνουμε από κάποια αυθαίρετη, που πάντως δεν αλλάζει (η κατανομή) με την κατάσταση της διαδικασίας. Δηλαδή, μια ακολουθία τυχαίων μεταβλητών $\{S_n\}$ καλείται 'τυχαίος περίπατος' αν:

$$S_n = X_1+X_2+ \dots +X_n \quad , \quad \text{όπου } n = 1, 2, \dots \qquad (3.11)$$

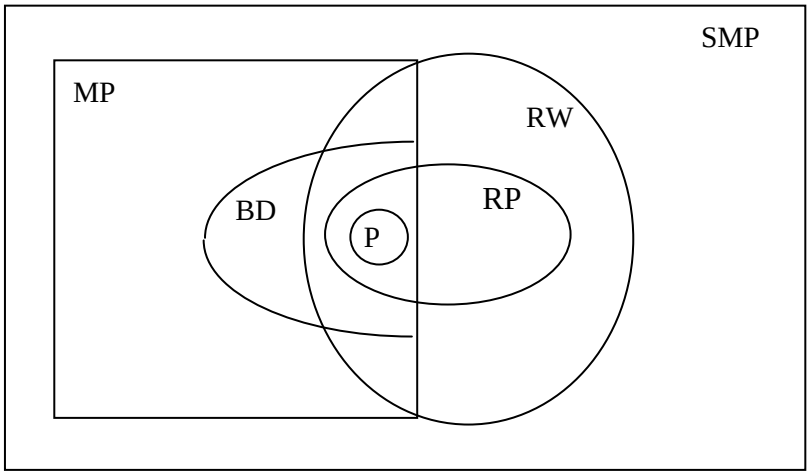
όπου $S_0 = 0$ και $X_1 , X_2 ,\dots$ είναι μια ακολουθία ανεξάρτητων τυχαίων μεταβλητών με κοινή κατανομή. Οι τυχαίοι περίπατοι είναι μια ειδική περίπτωση των διαδικασιών Semi - Markov.

g) <u>Διαδικασίες ανανέωσης</u> : Οι διαδικασίες ανανέωσης (ή επανάληψης) είναι ειδική περίπτωση των τυχαίων περιπάτων. Χαρακτηρίζονται από την εξίσωση Εξ. (3.11) όπου όμως S_n είναι τώρα η τυχαία μεταβλητή που καθορίζει τη *χρονική στιγμή* στην οποία γίνεται η *n*-οστή μεταβολή κατάστασης και $\{X_n\}$είναι ένα σύνολο ανεξάρτητων, όμοια κατανεμημένων τυχαίων μεταβλητών, όπου η X_n αντιπροσωπεύει τώρα το χρόνο μεταξύ της *(n-1)*-οστής και *n*-οστής μεταβολής κατάστασης . Οι μεταβολές γίνονται μόνο μεταξύ γειτονικών καταστάσεων.

Στο σχήμα 3.3 φαίνονται οι σχέσεις των κλάσεων των Στοχαστικών διαδικασιών που έχουμε ως τώρα ορίσει όπου ισχύουν οι παρακάτω συμβολισμοί:

SMP: Διαδικασία Semi - Markov
MP: Διαδικασία Markov
RW: Τυχαίος περίπατος
RP: Διαδικασία ανανέωσης
BD: Διαδικασία Γεννήσεων-Θανάτων
P: Διαδικασία Poisson

Η διαδικασία Poisson, που παίζει κεντρικό ρόλο στη θεωρία ουρών, θα μελετηθεί αργότερα.



Σχήμα 3.3. Σχέσεις μεταξύ των κλάσεων Στοχαστικών Διαδικασιών

3.3 ΑΛΥΣΙΔΕΣ MARKOV ΔΙΑΚΡΙΤΟΥ ΧΡΟΝΟΥ

Οι αλυσίδες Markov χρησιμοποιούνται για την περιγραφή της κίνησης ενός αντικειμένου μέσα σε κάποιο χώρο. Εξετάζουμε τώρα αλυσίδες Markov διακριτού-χρόνου, που επιτρέπουν στο αντικείμενο να καταλαμβάνει διακριτές θέσεις (καταστάσεις) και επιτρέπει οι αλλαγές μεταξύ αυτών των θέσεων να γίνονται μόνο σε διακριτές χρονικές στιγμές. Αρχίζουμε την παρουσίαση της θεωρίας με ένα παράδειγμα.

Έστω ένας τουρίστας που ταξιδεύει από πόλη σε πόλη σε μια χώρα. Έστω X_n η πόλη που θα βρεθεί ο τουρίστας το μεσημέρι της μέρας n. Όταν είναι σε κάποια πόλη i, θα φύγει το απόγευμα από την πόλη με το πρώτο auto-stop που θα τον πάρει. Υποθέτουμε ότι ο χρόνος ταξιδιού μεταξύ δύο οποιονδήποτε πόλεων είναι αμελητέος. Φυσικά είναι πιθανό να μην τον πάρει auto-stop εκείνη την μέρα, οπότε μένει στην πόλη i μέχρι το απόγευμα της επόμενης μέρας. Μια και τα αυτοκίνητα που πηγαίνουν στις διάφορες γειτονικές πόλεις, εμφανίζονται με απρόβλεπτο προορισμό, η θέση του τουρίστα σε κάποια στιγμή στο μέλλον, είναι σίγουρα μια τυχαία μεταβλητή που μπορεί κάλλιστα να περιγραφεί από μια αλυσίδα Markov διακριτού-χρόνου.

Ορισμός:

Η ακολουθία τυχαίων μεταβλητών $X_1, X_2, \ldots$ συνθέτει μια αλυσίδα Markov διακριτού-χρόνου, αν για όλα τα n $(n = 1, 2, \ldots)$ και όλες τις πιθανές τιμές των τυχαίων μεταβλητών ισχύει ότι:

$$P\left[X_n = j \mid X_1 = i_1, X_2 = i_2, \ldots, X_{n-1} = i_{n-1}\right] = P\left[X_n = j \mid X_{n-1} = i_{n-1}\right] \quad (3.12)$$

□

Όσον αφορά το παράδειγμα μας, ο ορισμός αυτός απλώς λέει ότι η επόμενη πόλη που θα επισκεφθεί ο τουρίστας, εξαρτάται *μόνο* από την πόλη στην οποία είναι τώρα. Δηλαδή η μνήμη της αλυσίδας Markov πηγαίνει πίσω μόνο στην πιο πρόσφατη θέση (κατάσταση) του αντικειμένου. Όταν $X_n = j$ (ο τουρίστας είναι στην πόλη j την μέρα n) τότε λέμε ότι το σύστημα βρίσκεται στην *κατάσταση* E_j την χρονική στιγμή n (ή κατά το n-οστό βήμα). Για να ξεκινήσει ο τουρίστας μας την ημέρα 0, αρχίζουμε με κάποια *αρχική κατανομή πιθανότητας* $P[X_0 = j]$.

Η έκφραση στο δεξιό μέρος της Εξ. (3.12) αναφέρεται σαν η *πιθανότητα μετάβασης* (ενός-βήματος) και δίνει την υπό συνθήκη πιθανότητα να γίνει η μετάβαση (αλλαγή) της διαδικασίας από την κατάσταση E_i όπου είναι στο βήμα $(n-1)$, στην κατάσταση E_j κατά

το βήμα *n*. Είναι φανερό πώς αν μας δοθούν η κατανομή πιθανότητας της αρχικής κατάστασης και οι πιθανότητας μετάβασης, τότε μπορούμε με μοναδικό τρόπο να βρούμε να είμαστε σε διάφορες καταστάσεις την χρονική στιγμή *n*.

Αν οι πιθανότητες μετάβασης είναι *ανεξάρτητες του n*, τότε έχουμε μια *ομογενή αλυσίδα Markov* και ορίζουμε:

$$p_{ij} \equiv P\left[X_n = j \mid X_{n-1} = i\right] \tag{3.13}$$

που δίνει την πιθανότητα να μεταβούμε στην κατάσταση E_j στο επόμενο βήμα, δεδομένου ότι είμαστε στην κατάσταση E_i. *Η ανάλυση που ακολουθεί αναφέρεται μόνο σε ομογενείς αλυσίδες Markov*.

Οι αλυσίδες αυτές είναι τέτοιες που οι πιθανότητες μετάβασης είναι στάσιμες στο χρόνο. (Σημειώστε εδώ ότι, αν και έχουμε αλυσίδες Markov με στάσιμες μεταβάσεις, δεν είναι απαραίτητο, οι ίδιες να είναι στάσιμές στοχαστικές διαδικασίες). Συνεπώς, δεδομένης της παρούσας κατάστασης, η πιθανότητα διαφόρων καταστάσεων μετά m βήματα, εξαρτάται μόνο από το m και όχι από την παρούσα χρονική στιγμή. Είναι βολικό να ορίσουμε εδώ τις πιθανότητες μετάβασης *m*-βημάτων:

$$p_{ij}^{(m)} \equiv P\left[X_{n+m} = j \mid X_n = i\right] \tag{3.14}$$

Είναι εύκολο να πάρουμε την ακόλουθη αναδρομική σχέση για το $p_{ij}^{(m)}$:

$$p_{ij}^{(m)} = \sum_k p_{ik}^{(m-1)} p_{kj} \tag{3.15}$$

Η εξίσωση αυτή απλώς λέει ότι αν πρόκειται να ταξιδέψουμε από την E_i στην E_j μέσα σε *m* βήματα, πρέπει να το κάνουμε 'ταξιδεύοντας' πρώτα από την E_i σε κάποια E_k μέσα σε *(m-1)* βήματα και μετά από την E_k στην E_j στο επόμενο βήμα. Τα δυο τελευταία γεγονότα είναι ανεξάρτητα, μια και πρόκειται για μια αλυσίδα Markov, ισχύει δηλαδή :

$$P\left[\left(X_m = j \mid X_{m-1} = k, X_0 = i\right) \middle| \left(X_{m-1} = k \middle| X_0 = i\right)\right]$$

$$= P\left[\left(X_m = j \mid X_{m-1} = k, X_0 = i\right)\right] \overset{(2.39)}{=} P\left[X_m = j \mid X_{m-1} = k\right]$$

Ο.Ε.Δ.

Άρα η πιθανότητα τους είναι το γινόμενο των πιθανοτήτων κάθε μιας, και αν προσθέσουμε το γινόμενα για όλες τις πιθανές ενδιάμεσες καταστάσεις E_k, φτάνουμε στο $p_{ij}^{(m)}$ της Εξ. (3.15).

Ορισμοί:

Μια αλυσίδα Markov λέγεται *αμείωτη* αν κάθε κατάσταση της μπορεί να προσπελασθεί από όλες τις υπόλοιπες καταστάσεις. Δηλαδή για κάθε ζευγάρι καταστάσεων (E_i και E_j) υπάρχει ένας ακέραιος m_0 (που μπορεί να εξαρτάται από το *i* και το *j*) έτσι ώστε

$$p_{ij}^{(m_0)} > 0$$

Έστω *A* το σύνολο όλων των καταστάσεων σε μια αλυσίδα Markov. Ένα υποσύνολο καταστάσεων A_1 λέγεται *κλειστό* αν δεν είναι δυνατή καμία μετάβαση ενός βήματος από οποιαδήποτε κατάσταση του A_1 σε οποιαδήποτε κατάσταση του A_1^c. Αν το A_1 αποτελείται από μια μόνο κατάσταση, έστω E_i , τότε αυτή καλείται *απορροφητική* κατάσταση. Μια αναγκαία και ικανή συνθήκη ώστε να είναι η E_i απορροφητική, είναι $p_{ii} = 1$. Αν το *A* είναι κλειστό και δεν περιέχει κανένα κλειστό υποσύνολο τότε έχουμε *αμείωτη αλυσίδα Markov*. Αντίθετα, αν το *A* περιέχει κλειστά υποσύνολα η αλυσίδα λέγεται *μειώσιμη*. Αν ένα κλειστό υποσύνολο μιας μειώσιμης αλυσίδας Markov δεν περιέχει κλειστά

υποσύνολα, τότε καλείται (το υποσύνολο) *αμείωτη υπο-αλυσίδα Markov*. Αυτές οι υπο-αλυσίδες μπορούν να μειωθούν να μελετηθούν ανεξάρτητα από τις άλλες καταστάσεις.

Στο παράδειγμα μας, ο τουρίστας μπορεί να μην θελήσει να επιστρέψει σε κάποια πόλη που έχει επισκεφθεί. Όμως αυτό μπορεί να γίνει λόγω του τρόπου που ταξιδεύει, και είναι ενδιαφέρον να καθορίσουμε εδώ αυτή την ποσότητα:

$f_j^{(n)} \equiv P[$*Η πρώτη επιστροφή στην* E_j *γίνεται μετά από n βήματα από την αναχώρηση από την* $E_j]$

και η πιθανότητα να επιστρέψουμε κάποτε στην E_j είναι:

$$f_j = \sum_{n=1}^{\infty} \mathrm{f}_j^{(n)} = P[\ \textit{Κάποτε να επιστρέψουμε στην}\ E_j\ ]$$

Είναι τώρα δυνατόν να ταξινομήσουμε τις καταστάσεις μιας διαδικασίας Markov, ανάλογα με την τιμή του f_j. Συγκεκριμένα, αν $f_j = 1$ η κατάσταση E_j λέγεται *επαναληπτική*. Αν αντίθετα $f_j < 1$, τότε η κατάσταση E_j λέγεται *μεταβατική*. Παραπέρα, αν τα μόνα δυνατά βήματα κατά τα οποία μπορούμε να επιστρέψουμε στην E_j είναι $\gamma, 2\gamma, 3\gamma, \ldots$ (όπου $\gamma > 0$, ο μεγαλύτερος τέτοιος ακέραιος), τότε η E_j λέγεται *περιοδική* με περίοδο γ. Αν $\gamma = 1$ τότε η E_j είναι *μη-περιοδική*.

Εξετάζοντας τις καταστάσεις για τις οποίες $f_j = 1$, μπορούμε να ορίσουμε τον *μέσο χρόνο επανάληψης* της E_j:

$$\mathrm{M}_j \equiv \sum_{n=1}^{\infty} n\mathrm{f}_j^{(n)} \tag{3.16}$$

Το M_j είναι απλώς ο μέσος χρόνος επιστροφής στην E_j. Τώρα μπορούμε να ταξινομήσουμε τις καταστάσεις ακόμα περισσότερο. Συγκεκριμένα, αν $M_j = \infty$, η E_j λέγεται *μηδενικά επαναληπτική*, ενώ αν $M_j < \infty$, η E_j λέγεται *βέβαια επαναληπτική*.

Έστω $\pi_j^{(n)}$ η πιθανότητα να βρεθεί το σύστημα (η αλυσίδα Markov) στην κατάσταση E_j κατά το n-οστό βήμα, δηλαδή:

$$\pi_j^{(n)} \equiv P\left[X_n = j\right] \tag{3.17}$$

Παρουσιάζουμε τώρα δύο σημαντικά θεωρήματα χωρίς απόδειξη:

Θεώρημα 1.

Οι καταστάσεις μιας αμείωτης αλυσίδας Markov είναι είτε όλες μεταβατικές, είτε όλες βέβαια επαναληπτικές ή όλες μηδενικά επαναληπτικές. Αν είναι περιοδικές, τότε όλες οι καταστάσεις έχουν την ίδια περίοδο γ.

□

Αν υποθέσουμε στο παράδειγμα μας, ότι ο τουρίστας περιπλανιέται για πάντα, θα περάσει από τις διάφορες πόλεις της χώρας πολλές φορές. Μπαίνει το ερώτημα αν υπάρχει μια *στάσιμη* κατανομή πιθανότητας $\{\pi_j\}$ που περιγράφει την πιθανότητα να βρεθεί στην πόλη j κάποια χρονική στιγμή στο απώτερο μέλλον. [Μια κατανομή πιθανότητας P_j λέγεται *στάσιμη κατανομή* αν, όταν την διαλέξουμε για κατανομή της αρχικής κατάστασης (δηλαδή $\pi_j^{(0)} = P_j$), τότε για όλα τα n θα είναι $\pi_j^{(n)} = P_j$]. Το να βρούμε τα $\{\pi_j\}$ είναι το πιο σημαντικό τμήμα της ανάλυσης των αλυσίδων Markov.

Θεώρημα 2.

Σε μια αμείωτη και μη-περιοδική ομογενή αλυσίδα Markov, οι οριακές πιθανότητες

$$\pi_j = \lim_{n\to\infty} \pi_j^{(n)} \tag{3.18}$$

υπάρχουν πάντα , και είναι ανεξάρτητες από την κατανομή της αρχικής κατάστασης.
Επίσης ισχύει:

(a) Είτε όλες οι καταστάσεις είναι μεταβατικές ή όλες είναι μηδενικά επαναληπτικές, οπότε $\pi_j = 0$ και δεν υπάρχει στάσιμη κατανομή.

(b) Είτε όλες οι καταστάσεις είναι βέβαια επαναληπτικές και τότε $\pi_j > 0$ για όλα τα j, στην οποία περίπτωση το σύνολο $\{ \pi_j \}$ είναι μια στάσιμη κατανομή και

$$\pi_j = \frac{1}{M_j} \tag{3.19}$$

Στην τελευταία περίπτωση οι ποσότητες π_j καθορίζονται κατά μοναδικό τρόπο από τις εξής εξισώσεις:

$$1 = \sum_i \pi_i \tag{3.20}$$

$$\pi_j = \sum_i \pi_i p_{ij} \tag{3.21}$$

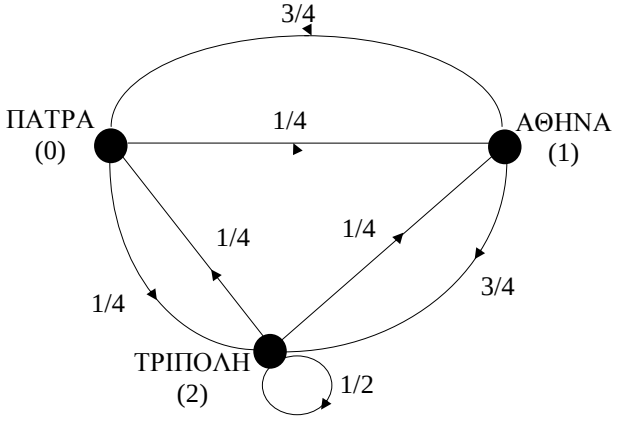
□

Ορισμός:

Μια κατάσταση E_j λέγεται εργοδική, αν είναι μη-περιοδική και βέβαια επαναληπτική. Δηλαδή αν $f_j = 1$, $M_j < \infty$ και $\gamma = 1$. Αν όλες οι καταστάσεις μιας αλυσίδας Markov είναι εργοδικές, τότε η αλυσίδα Markov λέγεται και η ίδια *εργοδική*.

Μια αλυσίδα Markov είναι εργοδική αν η κατανομή μάζας πιθανότητας $\{\pi_j^{(n)}\}$ συγκλίνει πάντα (σαν συνάρτηση του n) σε μια στάσιμη οριακή κατανομή $\{\pi_j\}$ που είναι ανεξάρτητη από την κατανομή αρχικής κατάστασης. Είναι εύκολο να δειχθεί ότι όλες οι καταστάσεις μιας *πεπερασμένης*, μη-περιοδικής, αμείωτης αλυσίδας Markov είναι εργοδικές. (Μια αλυσίδα Markov είναι πεπερασμένη, αν έχει πεπερασμένο αριθμό καταστάσεων. Αν μια αμείωτη αλυσίδα Markov είναι μεταβατική ή μηδενικά επαναληπτική, *δεν* μπορεί να είναι πεπερασμένη). Επίσης, έχει αποδειχθεί ότι μια αμείωτη και μη-περιοδική αλυσίδα Markov είναι εργοδική, αν το σύνολο των γραμμικών Εξ. (3.21) έχει μια μη μηδενική λύση για την οποία $\sum_j |\pi_j| = \infty$. Οι οριακές πιθανότητες $\{\pi_j\}$ μιας εργοδικής αλυσίδας Markov συχνά αναφέρονται σαν οι *πιθανότητες μόνιμης κατάστασης* με την έννοια ότι η επίδραση της κατανομής αρχικής κατάστασης $\pi_j^{(0)}$ έχει εξαφανιστεί.

Γυρνώντας στο παράδειγμα μας, έστω ότι ο τουρίστας βρίσκεται στην Ελλάδα, και έστω ότι οι 3 μοναδικές πόλεις που μπορεί να επισκεφθεί είναι η Πάτρα, η Αθήνα και η Τρίπολη. Το οδικό δίκτυο φαίνεται στο Σχήμα 3.4. Στο Σχήμα αυτό οι κατευθυνόμενες πλευρές αντιπροσωπεύουν επιτρεπτές κατευθύνσεις κυκλοφορίας Οι αριθμοί πάνω αντιπροσωπεύουν την πιθανότητα (p_{ij}) ότι ο τουρίστας θα ταξιδέψει σε αυτό το δρόμο, δεδομένου ότι ξεκινά από την πόλη που είναι στην αρχή του βέλους.



Σχήμα 3.4. Μια αλυσίδα Markov

Οι αριθμοί στις παρενθέσεις κάτω από τα ονόματα των πόλεων, θα χρησιμοποιούνται στο εξής αντί για τα ονόματα. Σημειώστε ότι ο τουρίστας έχει πιθανότητα ½ να παραμείνει στην Τρίπολη για μια μέρα, όταν βρεθεί εκεί.

Το διάγραμμα αυτό ονομάζεται *διάγραμμα καταστάσεων -πιθανοτήτων μεταβάσεων*.

Ορίζουμε τον *πίνακα μεταβάσεων* $\vec{\mathbf{P}}$:

$$\vec{\mathbf{P}} = \left[p_{ij} \right] \tag{3.22}$$

και το *διάνυσμα πιθανοτήτων* $\vec{\boldsymbol{\pi}}$:

$$\vec{\boldsymbol{\pi}} = \left[\pi_0, \pi_1, \pi_2, ... \right] \tag{3.23}$$

οπότε οι εξισώσεις Εξ. 3.21 γράφονται:

$$\vec{\boldsymbol{\pi}} = \vec{\boldsymbol{\pi}} \cdot \vec{\mathbf{P}} \tag{3.24}$$

Στο παράδειγμα μας, είναι:

$$\vec{\mathbf{P}} = \begin{bmatrix} 0 & 3/4 & 1/4 \\ 1/4 & 0 & 3/4 \\ 1/4 & 1/4 & 1/2 \end{bmatrix}$$

και συνεπώς μπορούμε να λύσουμε την Εξ. (3.24) παίρνοντας τις τρεις παρακάτω εξισώσεις από αυτήν (τις Εξ. (3.21)).

$$\begin{aligned} \pi_0 &= 0 \cdot \pi_0 + 1/4 \cdot \pi_1 + 1/4 \cdot \pi_2 \\ \pi_1 &= 3/4 \cdot \pi_0 + 0 \cdot \pi_1 + 1/4 \cdot \pi_2 \\ \pi_2 &= 1/4 \cdot \pi_0 + 3/4 \cdot \pi_1 + 1/2 \cdot \pi_2 \end{aligned} \tag{3.25}$$

Παρατηρούμε ότι μόνο οι δύο από τις τρεις εξισώσεις είναι ανεξάρτητες (η Τρίτη προκύπτει από τις άλλες δύο) πράγμα που συμβαίνει *πάντα* (μόνο οι *(n-1)* από τις εξισώσεις που δίνουν τα π_j είναι ανεξάρτητες), και συνεπώς χρειάζεται μια επιπλέον σχέση, που είναι η Εξ. (3.20), η οποία στο παράδειγμα μας είναι:

$$1 = \pi_0 + \pi_1 + \pi_2 \tag{3.26}$$

Λύνοντας το σύστημα των Εξ. (3.25) και (3.26) παίρνουμε :

$$\begin{aligned} \pi_0 &= 1/5 = 0.20 \\ \pi_1 &= 7/25 = 0.28 \\ \pi_2 &= 13/25 = 0.52 \end{aligned} \tag{3.27}$$

Αυτές είναι οι (στάσιμες) πιθανότητες μόνιμης κατάστασης. Είναι καθαρό ότι έχουμε μια εργοδική αλυσίδα Markov (είναι πεπερασμένη και αμείωτη).

Συχνά ενδιαφερόμαστε για τη *μεταβατική* συμπεριφορά του συστήματος. Αυτό σημαίνει να βρούμε τα $\pi_j^{(n)}$, δηλαδή την πιθανότητα να βρεθούμε στην κατάσταση E_j τη χρονική στιγμή *n*. Ορίζουμε το διάνυσμα πιθανοτήτων στο βήμα *n*:

$$\vec{\boldsymbol{\pi}}^{(n)} \equiv \left[\pi_0^{(n)}, \pi_1^{(n)}, \pi_2^{(n)}, \quad ... \right] \tag{3.28}$$

Ισχύει:

$$\vec{\boldsymbol{\pi}}^{(1)} = \vec{\boldsymbol{\pi}}^{(0)} \cdot \vec{\mathbf{P}}$$

$$\text{και } \vec{\boldsymbol{\pi}}^{(2)} = \vec{\boldsymbol{\pi}}^{(1)} \cdot \vec{\mathbf{P}}$$

$$= \left[\vec{\boldsymbol{\pi}}^{(0)} \cdot \vec{\mathbf{P}} \right] \cdot \vec{\mathbf{P}}$$

$$= \vec{\boldsymbol{\pi}}^{(0)} \cdot \left(\vec{\mathbf{P}} \right)^2$$

αποτελέσματα που οδηγούν στις γενικές εξισώσεις:

$$\vec{\boldsymbol{\pi}}^{(n)} = \vec{\boldsymbol{\pi}}^{(n-1)} \cdot \vec{\mathbf{P}} \qquad (3.29)$$

και

$$\vec{\boldsymbol{\pi}}^{(n)} = \vec{\boldsymbol{\pi}}^{(0)} \cdot (\vec{\mathbf{P}})^n \qquad (3.30)$$

Από τους μέχρι τώρα ορισμούς συνεπάγεται:

$$\vec{\boldsymbol{\pi}} = \lim_{n \to \infty} \vec{\boldsymbol{\pi}}^{(n)}$$

αν υπάρχει το όριο, πράγμα που ισχύει αν έχουμε μια αμείωτη, περιοδική, ομογενή αλυσίδα Markov (Θεώρημα 2).

Υπό τις προϋποθέσεις αυτές η Εξ. (3.29) δίνει:

$$\lim_{n \to \infty} \vec{\boldsymbol{\pi}}^{(n)} = \lim_{n \to \infty} \vec{\pi}^{(n-1)} \cdot \vec{\mathbf{P}} \text{ , δηλαδή } \vec{\boldsymbol{\pi}} = \vec{\boldsymbol{\pi}} \cdot \vec{\mathbf{P}} \text{ που είναι πάλι η Εξ. (3.24).}$$

Εφαρμόζοντας τα τελευταία αποτελέσματα στο παράδειγμα μας, ας υποθέσουμε ότι ο τουρίστας ξεκινά από την Πάτρα τη χρονική στιγμή 0 με πιθανότητα 1, δηλαδή:

$$\vec{\pi}^{(0)} = [1,0,0]$$

Στον παρακάτω πίνακα φαίνεται η ακολουθία των τιμών $\vec{\pi}^{(n)}$. Φαίνεται επίσης η οριακή τιμή $\vec{\pi}$ των Εξ. (3.27).

n	0	1	2	3	4	. . . .	∞
$\pi_0^{(n)}$	1	0	0.250	0.187	0.203	. . . .	0.20
$\pi_1^{(n)}$	0	0.75	0.062	0.359	0.254	. . . .	0.28
$\pi_2^{(n)}$	0	0.25	0.688	0.454	0.543	. . . .	0.52

Πίνακας 3.1

Παρατηρούμε ότι οι ποσότητες $\pi_i^{(n)}$ συγκλίνουν πολύ γρήγορα προς τις οριακές τιμές της μόνιμης κατάστασης. Αν συμπληρώσετε τον παραπάνω πίνακα για διαφορετικές τιμές του $\vec{\pi}^{(0)}$ (δηλαδή εκκίνηση από διαφορετική πόλη) θα πάρετε τιμές που θα συγκλίνουν επίσης πολύ γρήγορα στις ίδιες οριακές τιμές.

Π.χ. Για $\vec{\pi}^{(0)}$ = [0,0,1] παίρνουμε:

n	0	1	2	3	4	. . . .	∞
$\pi_0^{(n)}$	0	0.25	0.187	0.203	0.199	. . . .	0.20
$\pi_1^{(n)}$	0	0.25	0.313	0.266	0.285	. . . .	0.28
$\pi_2^{(n)}$	1	0.50	0.50	0.531	0.516	. . . .	0.52

Πίνακας 3.2

Παρατηρούμε δηλαδή ότι οι οριακές τιμές είναι ανεξάρτητες από την αρχική θέση του αντικειμένου (δηλαδή την κατανομή πιθανότητας αρχικής κατάστασης).

Ας ασχοληθούμε τώρα με το χρόνο που το σύστημα μένει σε μια δεδομένη κατάσταση. Θα αποδείξουμε ότι ο αριθμός των μονάδων χρόνου που το σύστημα μένει στην ίδια κατάσταση, είναι κατανεμημένος *γεωμετρικά*. Όπως έχουμε πει, η γεωμετρική κατανομή είναι η *μοναδική διακριτή κατανομή χωρίς μνήμη*.

Έστω ότι το σύστημα έχει μόλις εισέλθει στην κατάσταση E_i. Θα μείνει στην κατάσταση αυτή και κατά το επόμενο βήμα, με πιθανότητα p_{ii}.

Αντίθετα, θα φύγει από την κατάσταση αυτή στο επόμενο βήμα, με πιθανότητα $(1-p_{ii})$.

Αν όντως παραμείνει στην ίδια κατάσταση, έχει πάλι πιθανότητα p_{ii} να μείνει και στο επόμενο βήμα, ενώ έχει πάλι πιθανότητα $(1-p_{ii})$ να φύγει στο επόμενο βήμα κ.ο.κ. Με βάση την ιδιότητα Markov, το γεγονός ότι έχει μείνει σε μια δεδομένη κατάσταση για ένα γνωστό αριθμό βημάτων, δεν επηρεάζει καθόλου την πιθανότητα να φύγει στο επόμενο βήμα. Οι πιθανότητες μετάβασης, λοιπόν, είναι ανεξάρτητες και μπορούμε να γράψουμε:

$$P\,[\text{ Το σύστημα να παραμείνει στην } E_i \text{ για ακριβώς } m \text{ επιπλέον βήματα, δεδομένου ότι έχει μόλις εισέλθει στην } E_i\,] = (1 - p_{ii})\, p_{ii}^{\,m} \quad (3.31)$$

Η τελευταία πρόταση είναι η *Γεωμετρική κατανομή μάζας πιθανότητας*.

□

3.4 ΑΛΥΣΙΔΕΣ MARKOV ΣΥΝΕΧΟΥΣ ΧΡΟΝΟΥ ΚΑΙ ΔΙΑΔΙΚΑΣΙΕΣ ΓΕΝΝΗΣΕΩΝ - ΘΑΝΑΤΩΝ

Τα απλούστερα συστήματα αναμονής, τα οποία και θα μελετήσουμε, είναι της μορφής *M/M/m/K*. Τα συστήματα αυτά έχουν χρόνους μεταξύ διαδοχικών αφίξεων (Χ.Α.) και χρόνους εξυπηρέτησης (Χ.Ε.) *εκθετικά κατανεμημένους* σύμφωνα με το συμβολισμό που περιγράφεται στην παράγραφο 3.1.1. Η εκθετική κατανομή είναι η μοναδική συνεχής κατανομή πιθανότητας που έχει τη σημαντική *ιδιότητα της αμνησίας*, έτσι ώστε τα συστήματα αυτά να έχουν ιδιαίτερα εύκολη αναλυτική λύση και να χρησιμοποιούνται εκτεταμένα στη μελέτη συστημάτων κάθε τύπου, περιλαμβανομένων των πληροφοριακών.

Η υπόθεση *εκθετικής κατανομής* για τους Χ.Α. και Χ.Ε., σημαίνει ότι οι Συναρτήσεις Κατανομής Πιθανότητάς τους (ΣΚΠ), δίνονται από τις σχέσεις

$$A(t) = 1 - e^{-\lambda t}, \quad t \geq 0 \quad \text{και} \quad B(x) = 1 - e^{-\mu x}, \quad x \geq 0 \qquad (3.32)$$

αντίστοιχα, όπου λ είναι ο ρυθμός αφίξεων και μ ο ρυθμός εξυπηρέτησης (κάθε εξυπηρετητή αν είναι παραπάνω από ένας) στο σύστημα. Υπενθυμίζεται εδώ, ότι τα λ και μ είναι οι αντίστροφοι του μέσου χρόνου μεταξύ διαδοχικών αφίξεων $\bar{t}$ και του μέσου χρόνου εξυπηρέτησης $\bar{x}$, αντίστοιχα.

Η *ιδιότητα της αμνησίας* αναφέρει ότι «ο χρόνος ως το επόμενο γεγονός, είναι ανεξάρτητος από το χρόνο που έχει περάσει από το τελευταίο γεγονός». Πρακτικά, για τα συστήματα *M/M/m/K* που θα μελετήσουμε, αυτό σημαίνει:

Για τη διαδικασία αφίξεων: Αν από την τελευταία άφιξη στο σύστημα έχει περάσει χρονικό διάστημα έστω t_0, τότε το χρονικό διάστημα μέχρι την άφιξη του επόμενου πελάτη είναι ανεξάρτητο από το t_0. Μαθηματικά αυτό γράφεται ως:

$$\text{Prob}[\, t_n \le t + t_0 \mid t_n > t_0 \,] = \text{Prob}[\, t_n \le t \,]$$

όπου t_n είναι η *τυχαία μεταβλητή* που αντιπροσωπεύει το χρόνο μεταξύ των αφίξεων του (*n-1*)-στού και του *n*-στού πελάτη στο σύστημα, ενώ t είναι μια παράμετρος.

Για τη διαδικασία εξυπηρέτησης: Αν ένας πελάτης έχει εισέλθει και χρησιμοποιεί έναν εξυπηρετητή του συστήματος για χρονικό διάστημα x_0, τότε το χρονικό διάστημα μέχρι την ολοκλήρωση της εξυπηρέτησής του είναι ανεξάρτητο του x_0. Αντίστοιχα εδώ:

$$\text{Prob}[\, x_n \le x + x_0 \mid x_n > x_0 \,] = \text{Prob}[\, x_n \le x \,]$$

όπου x_n είναι η *τυχαία μεταβλητή* που αντιπροσωπεύει το χρόνο εξυπηρέτησης του *n*-στού πελάτη, ενώ *x* είναι μια παράμετρος.

3.4.1 Χαρακτηριστικά και Μελέτη των Αλυσίδων Markov

Τα συστήματα *M/M/m/K* που θα μελετήσουμε, εντάσσονται στην κατηγορία των Στοχαστικών Διαδικασιών (Σ.Δ.) που αναφέρονται ως *αλυσίδες Markov*. Αυτές είναι Σ.Δ. που έχουν ένα διακριτό σύνολο *καταστάσεων*, δηλαδή τιμών που παίρνει η Σ.Δ. Στα συστήματά μας, ως κατάσταση θεωρούμε τον *αριθμό των πελατών* που βρίσκονται μέσα στο σύστημα αναμονής (ουρά και εξυπηρετητές) κάθε χρονική στιγμή. Συνεπώς, το σύνολο των πιθανών καταστάσεων σε ένα σύστημα *M/M/m/K*, είναι το $\{0,1,2,3,\ldots,K\}$, αφού στο σύστημα μπορούν να βρίσκονται το πολύ *K* πελάτες στην ουρά και στους εξυπηρετητές. Αν η ουρά έχει άπειρο μήκος, τότε το σύνολο των πιθανών καταστάσεων είναι το $\{0,1,2,3,\ldots\}$. Η ιδιότητα Markov αναφέρεται στην ιδιότητα αμνησίας που διαθέτουν τα συστήματα που θα μελετήσουμε.

Επιπλέον, θεωρούμε ότι τα συστήματα *M/M/m/K* είναι *διαδικασίες Γεννήσεων – Θανάτων* (birth – death processes). Στις διαδικασίες γεννήσεων - θανάτων επιτρέπονται *μεταβάσεις* (αλλαγές κατάστασης) από μια κατάσταση, μόνο σε γειτονικές καταστάσεις. Δηλαδή αν το σύστημα βρίσκεται στην κατάσταση *j*, τότε στην επόμενη αλλαγή κατάστασης θα βρεθεί σε μια από τις καταστάσεις $j-1$ ή $j+1$. Αυτό σημαίνει για τα συστήματα που θα μελετήσουμε, ότι δεν επιτρέπονται πολλαπλές αφίξεις, πολλαπλές αναχωρήσεις, ούτε ταυτόχρονη αναχώρηση με άφιξη. Δηλαδή, επιτρέπεται η (πιθανή) εμφάνιση μόνο ενός γεγονότος κάθε χρονική στιγμή. Η υπόθεση αυτή είναι συμβατή με τα χαρακτηριστικά ενός συστήματος που λειτουργεί σε συνεχή χρόνο, δηλαδή οι αλλαγές καταστάσεων μπορούν να συμβούν οποτεδήποτε. Τα συστήματά μας έχουν τέτοια λειτουργία.

Η επίλυση του γενικού συστήματος *M/M/m/K* απαιτεί τον προσδιορισμό του

$$P_k(t) = \text{Prob}[\text{ να υπάρχουν } k \text{ πελάτες στο σύστημα τη χρονική στιγμή } t\,] \qquad (3.33)$$

$$\text{για } 0 \le k \le K, \quad t \ge 0$$

Θεωρώντας ότι το σύστημά μας είναι σταθερό (δηλαδή $0 \le \rho < 1$, βλέπε υποενότητα 3.1.3.), μπορούμε να υποθέσουμε ότι η παραπάνω κατανομή πιθανότητας (3.32) συγκλίνει σε μια *κατανομή μόνιμης κατάστασης* p_k, όπου

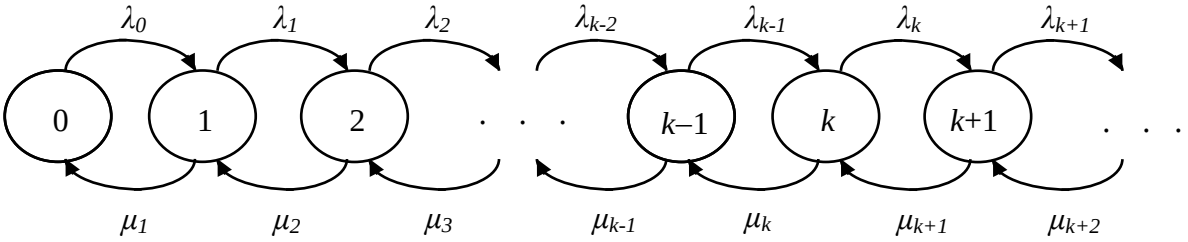
$$p_k = \lim_{t\to\infty} P_k(t) \tag{3.34}$$

Ουσιαστικά η ύπαρξη κατανομής μόνιμης κατάστασης υποδηλώνει ότι το σύστημα έρχεται σε μια στατιστική ισορροπία μετά από κάποιο αρχικό χρονικό διάστημα που χαρακτηρίζεται ως *μεταβατική περίοδος*. Δηλαδή, στη μόνιμη κατάσταση δεν έχει νόημα να ζητάμε την «πιθανότητα να είναι *k* πελάτες στο σύστημα τη χρονική στιγμή *t*», αλλά αρκεί να γνωρίζουμε το p_k, δηλαδή την *«πιθανότητα να είναι k πελάτες στο σύστημα κάποια χρονική στιγμή»*, αφού μετά από ένα μεγάλο διάστημα, για κάθε χρονική στιγμή *t* η απάντηση θα είναι ίδια.

Ο προσδιορισμός του p_k είναι αρκετός για την επίλυση των περισσοτέρων συστημάτων αναμονής που έχουν κάποιο ενδιαφέρον. Συνεπώς θα προσανατολίσουμε την ανάλυσή μας στην εύρεση του p_k για τα συστήματα της μορφής *M/M/m/K*.

3.4.2 Η Γενική Λύση των Αλυσίδων Markov Γεννήσεων - Θανάτων

Ένας βολικός τρόπος αναπαράστασης των αλυσίδων Markov είναι με τη χρήση του *διαγράμματος καταστάσεων – ρυθμών μεταβάσεων*. Στο Σχήμα 3.5 φαίνεται ένα τέτοιο διάγραμμα για τη γενική περίπτωση αλυσίδας Markov Γεννήσεων – Θανάτων.



Σχήμα 3.5. Διάγραμμα καταστάσεων-ρυθμών μεταβάσεων της γενικής αλυσίδας Markov Γεννήσεων-Θανάτων.

Στο διάγραμμα αυτό, οι κύκλοι συμβολίζουν τις πιθανές καταστάσεις στις οποίες μπορεί να βρεθεί η αλυσίδα, δηλαδή σε ένα σύστημα αναμονής αντιστοιχούν στον αριθμό των πελατών στο σύστημα. Το βέλος λ_k συμβολίζει το ρυθμό αφίξεων όταν υπάρχουν *k* πελάτες στο σύστημα και το βέλος μ_k συμβολίζει το ρυθμό εξυπηρέτησης όταν υπάρχουν *k* πελάτες στο σύστημα.

Η λύση p_k του παραπάνω συστήματος στη μόνιμη κατάσταση, είναι εύκολο να υπολογιστεί, με τη χρήση του ισχύοντος εδώ *νόμου ισορροπίας της ροής πιθανότητας*:

Στη μόνιμη κατάσταση, ο «ρυθμός ροής πιθανότητας» μιας αλυσίδας Markov *από* κάθε κατάσταση, είναι ίσος με το «ρυθμό ροής πιθανότητας» *προς* την κατάσταση.

ΣΗΜΕΙΩΣΗ: Ο παραπάνω νόμος ισχύει για οποιαδήποτε αλυσίδα Markov κι όχι μόνο για διαδικασίες Γεννήσεων-Θανάτων.

Αν συγκεντρώσουμε την προσοχή μας σε μια οποιαδήποτε κατάσταση *k* του διαγράμματος του σχήματος 3.5, ο νόμος αυτός μεταφράζεται στα παρακάτω:

{Ρυθμός ροής πιθανότητας *από* την κατάσταση k} = $p_k \cdot (\lambda_k + \mu_k)$

{Ρυθμός ροής πιθανότητας *προς* την κατάσταση k} = $p_{k-1} \cdot \lambda_{k-1} + p_{k+1} \cdot \mu_{k+1}$

Ο νόμος ισορροπίας της ροής πιθανότητας δηλώνει ότι οι παραπάνω δύο εκφράσεις είναι ίσες. Δηλαδή:

$$p_k \cdot (\lambda_k + \mu_k) = p_{k-1} \cdot \lambda_{k-1} + p_{k+1} \cdot \mu_{k+1} \quad \text{για} \quad k \geq 1 \tag{3.35}$$

Αντίστοιχη σχέση ισχύει και για την (ειδικής μορφής) αρχική κατάσταση 0:

$$p_0 \cdot \lambda_0 = p_1 \cdot \mu_1 \qquad (\text{για } k = 0) \tag{3.36}$$

Φυσικά, ισχύει πάντα και η παρακάτω προφανής σχέση:

$$\sum_{k=0}^{\infty} p_k = 1 \tag{3.37}$$

Το σύνολο των σχέσεων (3.35), (3.36) και (3.37) μας δίνει εύκολα τη λύση της γενικής διαδικασίας γεννήσεων – θανάτων του σχήματος 3.5. Αρκεί να ξεκινήσουμε από τη σχέση (3.36) επιλύνοντας ως προς p_1 και στη συνέχεια να χρησιμοποιήσουμε αναδρομικά την (3.35) για να εκφράσουμε όλα τα p_k συναρτήσει του p_0. Στο τέλος θα αντικαταστήσουμε τα p_k στην (3.37) και θα έχουμε και τη σχέση για το p_0. Με τον τρόπο αυτό θα πάρουμε την παρακάτω λύση:

$$p_k = p_0 \cdot \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}} \quad \text{για} \quad k = 1, 2, 3, \ldots \tag{3.38}$$

$$p_0 = \frac{1}{1 + \sum_{k=1}^{\infty} \prod_{i=0}^{k-1} \frac{\lambda_i}{\mu_{i+1}}} \tag{3.39}$$

Η παραπάνω λύση θα υπάρχει (δηλαδή θα υπάρχει μόνιμη κατάσταση του συστήματος), αν ο παρονομαστής της σχέσης (3.39) είναι μικρότερος από ∞. Για να ισχύει το τελευταίο, θα πρέπει η ακολουθία λ_k / μ_k να συγκλίνει, δηλαδή θα πρέπει να υπάρχει κάποιο k_0 τέτοιο ώστε

$$\frac{\lambda_k}{\mu_k} < 1 \qquad \text{για όλα τα} \quad k \geq k_0 \tag{3.40}$$

Είναι σαφές ότι η τελευταία απαίτηση (3.40) είναι απαραίτητη για να έχουμε *σταθερό* σύστημα (βλέπε και παράγραφο 3.1.3.).

Η λύση που περιγράφεται από τις σχέσεις (3.38) και (3.39) μαζί με την απαραίτητη συνθήκη (3.40), αποτελούν τη βάση για την επίλυση των συγκεκριμένων απλών συστημάτων αναμονής που θα μελετήσουμε στην ενότητα 3.6.

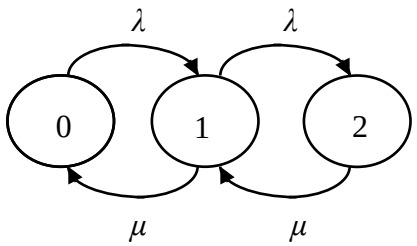
Παράδειγμα

Μας δίνεται μια αλυσίδα Markov γεννήσεων – θανάτων, η οποία έχει μόνο τρεις καταστάσεις {0, 1, 2}, ενώ ισχύει: $\lambda_k = \lambda$ για $k = 0,1$ και $\mu_k = \mu$ για $k = 1,2$. Ας

προσπαθήσουμε να κατασκευάσουμε το διάγραμμα καταστάσεων – ρυθμών μεταβάσεων για την αλυσίδα. Στη συνέχεια, θα επιλύσουμε την αλυσίδα στη μόνιμη κατάσταση και θα δούμε σε ποιο σύστημα αναμονής αντιστοιχεί.

Το παράδειγμα αυτό είναι μια εφαρμογή σε μικρότερη κλίμακα και με απλούστερα αριθμητικά δεδομένα, της μεθοδολογίας που περιγράφεται παραπάνω.

Το διάγραμμα καταστάσεων – ρυθμών μεταβάσεων είναι το εξής:



Για να βρούμε τη λύση του συστήματος στη μόνιμη κατάσταση, καταστρώνουμε τις σχέσεις του νόμου ισορροπίας της ροής για τις τρεις καταστάσεις:

Για την κατάσταση 0: $p_0 \cdot \lambda = p_1 \cdot \mu$

Για την κατάσταση 1: $p_1 \cdot (\lambda + \mu) = p_0 \cdot \lambda + p_2 \cdot \mu$

Για την κατάσταση 2: $p_2 \cdot \mu = p_1 \cdot \lambda$

Από τις παραπάνω τρεις σχέσεις, μόνο οι δύο είναι ανεξάρτητες. Η τρίτη «βγαίνει» με συνδυασμό των υπολοίπων δύο. Όμως χρησιμοποιούμε και την προφανή σχέση (3.37), η οποία εδώ είναι $p_0 + p_1 + p_2 = 1$ μαζί με δύο από τις παραπάνω τρεις σχέσεις και εύκολα καταλήγουμε στη λύση:

$$p_0 = \frac{1}{1 + \lambda/\mu + (\lambda/\mu)^2} \qquad p_1 = \frac{\lambda/\mu}{1 + \lambda/\mu + (\lambda/\mu)^2} \qquad p_2 = \frac{(\lambda/\mu)^2}{1 + \lambda/\mu + (\lambda/\mu)^2}$$

Η διαδικασία αυτή αντιστοιχεί στο σύστημα *M/M/1/2*. Ο λόγος είναι ότι υπάρχουν μόνο τρεις επιτρεπτές καταστάσεις που αντιστοιχούν: σε άδειο σύστημα (η κατάσταση 0), σε σύστημα με έναν πελάτη μέσα (η κατάσταση 1) και σε σύστημα με δύο πελάτες μέσα (η κατάσταση 2). Επειδή, μάλιστα, ο ρυθμός εξυπηρέτησης είναι πάντα ίσος με μ, είτε υπάρχει ένας, είτε δύο πελάτες στο σύστημα, συμπεραίνουμε ότι υπάρχει μόνο ένας εξυπηρετητής που εξυπηρετεί με ρυθμό μ και ο δεύτερος πελάτης (αν υπάρχει) βρίσκεται στη ουρά αναμονής του συστήματος.

3.5 ΔΙΑΔΙΚΑΣΙΕΣ POISSON

Είναι ειδική περίπτωση των διαδικασιών Γ-Θ στις οποίες έχουμε $\mu_\kappa = 0$ για όλα τα κ και $\lambda_\kappa = \lambda$ για όλα τα $\kappa = 0, 1, 2, \ldots$

Για λόγους απλότητας υποθέτουμε ότι το σύστημα ξεκινά τη χρονική στιγμή 0 με 0 μέλη στο πληθυσμό (βρίσκεται στην κατάσταση E_0 την στιγμή $t = 0$).

Δηλαδή:
$$P_\kappa(0) = \begin{cases} 1, & \kappa = 0 \\ 0, & \kappa \neq 0 \end{cases} \tag{3.41}$$

Για τη χρονική στιγμή t:

$$P_\kappa(t) = \frac{(\lambda t)^\kappa}{\kappa!} e^{-\lambda t} \text{ , για } \kappa \geq 0 \text{ , } t \geq 0 \qquad (3.42)$$

Η Εξ. (3.42) είναι η γνωστή *κατανομή Poisson*, που δίνει και την ονομασία στις *διαδικασίες Poisson*.

Μπορούμε να θεωρήσουμε την άφιξη πελατών σε κάποιο σύστημα αναμονής, σαν μια διαδικασία Poisson, όπου λ είναι ο μέσος ρυθμός άφιξης τους στο σύστημα. Με την αρχική συνθήκη της (3.41), το $P_\kappa(t)$ μας δίνει την πιθανότητα να φτάσουν κ πελάτες στο χρονικό διάστημα (0, t).

Διαισθητικά, υπολογίζουμε το μέσο αριθμό αφίξεων σε ένα χρονικό διάστημα μήκους *t*, ίσο με λt. Ας αποδείξουμε αυτό το αποτέλεσμα. έστω *K* η τυχαία μεταβλητή που εκφράζει τον αριθμό αφίξεων σε αυτό το διάστημα μήκους *t*.

Ισχύει:

$$E[K] = \sum_{\kappa=0}^{\infty} \kappa P_\kappa(t) = \sum_{\kappa=0}^{\infty} \kappa \frac{(\lambda t)^\kappa}{\kappa!} = e^{-\lambda t} \sum_{\kappa=1}^{\infty} \frac{(\lambda t)^\kappa}{(\kappa-1)!} = e^{-\lambda t} \lambda t \sum_{\kappa=0}^{\infty} \frac{(\lambda t)^\kappa}{\kappa!}$$

Εξ' ορισμού: $e^x = 1 + x + \frac{x^2}{2!} + \ldots$ και συνεπώς

$$e^{-\lambda t} = \left(e^{\lambda t}\right)^{-1} = \left[1 + \lambda t + \frac{\lambda t}{2!} + \cdots\right]^{-1} = \left(\sum_{\kappa=0}^{\infty} \frac{(\lambda t)^\kappa}{\kappa!}\right)^{-1}$$

Άρα: $$E[K] = \lambda t \qquad (3.43)$$

Για να βρούμε τη διακύμανση του αριθμού αφίξεων, υπολογίζουμε πρώτα την εξής ροπή:

$$E[K(K-1)] = \sum_{\kappa=0}^{\infty} \kappa(\kappa-1) P_\kappa(t) = e^{-\lambda t} \sum_{\kappa=1}^{\infty} \kappa(\kappa-1) \frac{(\lambda t)^\kappa}{\kappa!}$$

$$= e^{-\lambda t} (\lambda t)^2 \sum_{\kappa=2}^{\infty} \frac{(\lambda t)^{\kappa-2}}{(\kappa-2)!} = e^{-\lambda t} (\lambda t)^2 \sum_{\kappa=0}^{\infty} \frac{(\lambda t)^\kappa}{\kappa!} = (\lambda t)^2$$

οπότε:

$$\sigma_\kappa^2 = E[K(K-1)] + E[K] - (E[K])^2 = (\lambda t)^2 + \lambda t - (\lambda t)^2 \Leftrightarrow \sigma_\kappa^2 = \lambda t$$

Δηλαδή η *μέση τιμή* και η *διακύμανση* της διαδικασίας Poisson είναι ίδιες, ίσες με λt.

Ας εξετάσουμε τη σχέση μεταξύ της διαδικασίας Poisson και της εκθετικής κατανομής. Έστω η τυχαία μεταβλητή $\tilde{t}$ που είναι ο *χρόνος μεταξύ διαδοχικών αφίξεων* σε ένα σύστημα αναμονής, και της οποίας οι PDF και pdf είναι *A(t)* και *α(t)* αντίστοιχα. Τότε $\alpha(t)\Delta t + o(\Delta t)$ είναι η πιθανότητα ότι η επόμενη άφιξη θα γίνει τουλάχιστον *t* δευτερόλεπτα και το πολύ $(t + \Delta t)$ δευτερόλεπτα από την χρονική στιγμή της τελευταίας άφιξης.

Φυσικά ισχύει: $A(t) = 1 - P[\tilde{t} > t]$

Όμως $P[\tilde{t} > t]$ είναι απλώς η πιθανότητα να μην έχουμε άφιξη στο *(0, t)*, δηλαδή $P_0(t)$.

Άρα: $A(t) = 1 - P_0(t)$

Η οποία λόγω της Εξ. (3.42) γίνεται (αν έχουμε αφίξεις Poisson):

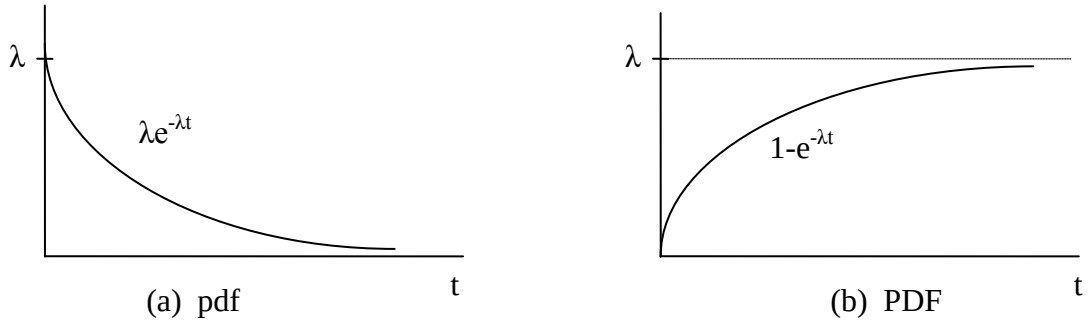
$$A(t) = 1 - e^{-\lambda t} \text{ , } t \geq 0 \qquad (3.44)$$

Παίρνοντας την παράγωγο ως προς *t*:

$$\alpha(t) = \lambda e^{-\lambda t} \text{ , } t \geq 0 \qquad (3.45)$$

Η οποία είναι η *εκθετική κατανομή*.

Η PDF και η pdf της φαίνονται στο Σχήμα 3.6.



Σχήμα 3.6. Η εκθετική κατανομή

Οι Εξ. (3.44) και (3.45) μας δείχνουν ότι για μια διαδικασία αφίξεων Poisson, οι χρόνοι μεταξύ διαδοχικών αφίξεων είναι εκθετικά κατανεμημένοι. Το πιο εντυπωσιακό χαρακτηριστικό της εκθετικής κατανομής είναι ότι έχει την ιδιότητα της *αμνησίας*. Αυτό σημαίνει ότι το παρελθόν μιας τυχαίας μεταβλητής που είναι εκθετικά κατανεμημένη, δεν παίζει κανένα ρόλο στην πρόβλεψη του μέλλοντος.

Συγκεκριμένα εννοούμε το εξής: Έστω ότι μόλις έγινε μια άφιξη τη χρονική στιγμή 0. Τώρα, έστω ότι πέρασαν t_0 δευτερόλεπτα κατά την διάρκεια των οποίων δεν έγινε άφιξη. Αν αυτή την στιγμή t_0 ρωτήσουμε «ποια είναι η πιθανότητα η επόμενη άφιξη να γίνει μετά από t δευτερόλεπτα από τώρα», η απάντηση θα είναι:

$$\mathrm{P}\left[\tilde{t} \le t + t_0 \;\middle|\; \tilde{t} > t_0\right] = \frac{\mathrm{P}\left[t_0 < \tilde{t} \le t + t_0\right]}{\mathrm{P}\left[\tilde{t} > t_0\right]} = \frac{\mathrm{P}\left[\tilde{t} \le t + t_0\right] - \mathrm{P}\left[\tilde{t} \le t_0\right]}{\mathrm{P}\left[\tilde{t} > t_0\right]}$$

η οποία λόγω της Εξ. (3.44) γίνεται:

$$\mathrm{P}\left[\tilde{t} \le t + t_0 \;\middle|\; \tilde{t} > t_0\right] = \frac{1 - e^{-\lambda(t+t_0)} - \left(1 - e^{-\lambda t_0}\right)}{1 - \left(1 - e^{-\lambda t_0}\right)}$$

ή αλλιώς

$$\mathrm{P}\left[\tilde{t} \le t + t_0 \;\middle|\; \tilde{t} > t_0\right] = 1 - e^{-\lambda t} \qquad (3.46)$$

Δηλαδή

$$\mathrm{P}\left[\tilde{t} \le t + t_0 \;\middle|\; \tilde{t} > t_0\right] = \mathrm{P}\left[\tilde{t} \le t\right]$$

Το τελευταίο αποτέλεσμα μας δείχνει ότι ο ‘χρόνος ως την επόμενη άφιξη είναι ανεξάρτητος από τον χρόνο που έχει περάσει από την τελευταία άφιξη’.

Η εκθετική κατανομή είναι η μόνη συνεχής κατανομή με αυτή την ιδιότητα της αμνησίας (ενώ η γεωμετρική κατανομή είναι η μόνη διακριτή κατανομή με αυτή την ιδιότητα).

Θα χρησιμοποιήσουμε τώρα την ιδιότητα της αμνησίας της εκθετικής κατανομής για να κλείσουμε τον κύκλο της σχέσης της διαδικασίας Poisson με την εκθετική κατανομή.

Η Εξ. (3.46) δίνει μια έκφραση για την PDF του χρόνου μεταξύ διαδοχικών αφίξεων υπό την συνθήκη ότι είναι τουλάχιστον t_0. Ας τοποθετηθούμε στη χρονική στιγμή t_0 και ας βρούμε την πιθανότητα η επόμενη άφιξη να γίνει μέσα στα επόμενα Δt δευτερόλεπτα.

Η Εξ. (3.46) θα μας δώσει:

$$\mathrm{P}\left[\tilde{t} \le t_0 + \Delta t \;\middle|\; \tilde{t} > t_0\right] = 1 - e^{-\lambda \Delta t} = 1 - \left[1 - \lambda \Delta t - \frac{(\lambda \Delta t)^2}{2!} - \cdots\right] = \lambda \Delta t + \mathrm{o}(\Delta t) \qquad (3.47)$$

Η τελευταία σχέση μας λέει ότι: δεδομένου ότι δεν έχει γίνει ακόμα άφιξη, η πιθανότητα να γίνει στο επόμενο χρονικό μήκους Δt, είναι $\lambda \Delta t + o(\Delta t)$.

Επίσης, η πιθανότητα να *μην* έχουμε άφιξη στο χρονικό διάστημα $(t_0, t_0 + \Delta t)$ είναι:

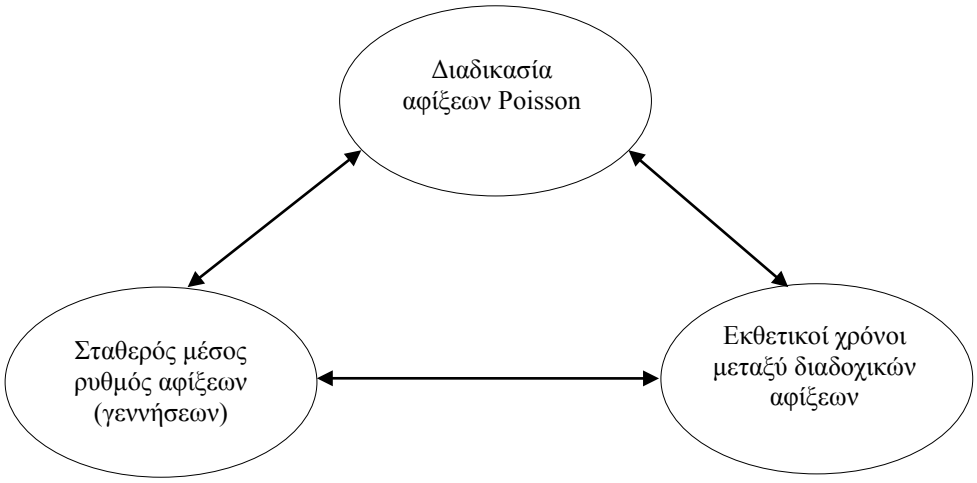
$$\mathrm{P}\left[\tilde{t} \le \Delta t + t_0 \;\middle|\; \tilde{t} > t_0\right] = 1 - \mathrm{P}\left[\tilde{t} \le \Delta t + t_0 \;\middle|\; \tilde{t} > t_0\right] = 1 - \left(1 - e^{-\lambda \Delta t}\right)$$

$$= e^{-\lambda \Delta t} = 1 - \lambda \Delta t - \frac{(\lambda \Delta t)^2}{2!} - \cdots = 1 - \lambda \Delta t + \mathrm{o}(\Delta t)$$

Επίσης:

P[2 ή περισσότερες αφίξεις στο (t_0, $t_0 + \Delta t$)]
= 1 - P[καμία άφιξη στο (t_0, $t_0 + \Delta t$)] - P[μία μόνο άφιξη στο (t_0, $t_0 + \Delta t$)]
= 1 - [1 - λΔt + o(Δt)] - [λΔt + o(Δt)] = o(Δt)

Το συμπέρασμα μας είναι λοιπόν ότι ‘εκθετικά κατανεμημένος χρόνος μεταξύ διαδοχικών αφίξεων (που είναι ανεξάρτητοι μεταξύ τους) συνεπάγονται ότι έχουμε διαδικασία Poisson, γεγονός που συνεπάγεται ότι έχουμε ένα σταθερό ρυθμό γεννήσεων (αφίξεων). Επίσης ισχύουν οι αντίστροφες συνεπαγωγές’. Οι σχέσεις αυτές φαίνονται στο Σχήμα 3.7.



Σχήμα 3.7. Ο κύκλος της ιδιότητας αμνησίας

Ας υπολογίσουμε τώρα τη μέση τιμή και τη διακύμανση της εκθετικής κατανομής.

Η μέση τιμή:

$$E\left[\tilde{t}\right] = \bar{t} = \int_0^\infty t\alpha(t)dt = \int_0^\infty t\lambda e^{-\lambda t}dt = \lambda \int_0^\infty t e^{-\lambda t}dt = \lambda \frac{1}{\lambda^2}$$

Δηλαδή:
$$\bar{t} = \frac{1}{\lambda} \tag{3.48}$$

Συνεπώς ο μέσος χρόνος μεταξύ διαδοχικών αφίξεων για μια κατανομή, είναι *1/ λ*.

Για να υπολογίσουμε την διακύμανση, βρίσκουμε πρώτα την δεύτερη ροπή του $\tilde{t}$:

$$\mathrm{E}\left[(\tilde{t})^2\right] = \int_0^\infty t^2\alpha(t)dt = \int_0^\infty t^2\lambda e^{-\lambda t}dt = \lambda \int_0^\infty t^2 e^{-\lambda t}dt = \lambda \frac{2}{\lambda^3} = \frac{2}{\lambda^2}$$

Άρα:
$$\sigma_{\tilde{t}}^2 = E\left[(\tilde{t})^2\right] - (\bar{t})^2 = \frac{2}{\lambda^2} - \left(\frac{1}{\lambda}\right)^2$$

Δηλαδή:
$$\sigma_{\tilde{t}}^2 = \frac{1}{\lambda^2} \tag{3.49}$$

Συντελεστής διασποράς:

$$C_a = \frac{\sigma_{\tilde{t}}}{\bar{t}} = 1 \qquad (3.50)$$

3.6 Η ΛΥΣΗ ΤΩΝ ΑΠΛΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΝΑΜΟΝΗΣ

3.6.1 *M/M/*1. Το Κλασικό Σύστημα Αναμονής

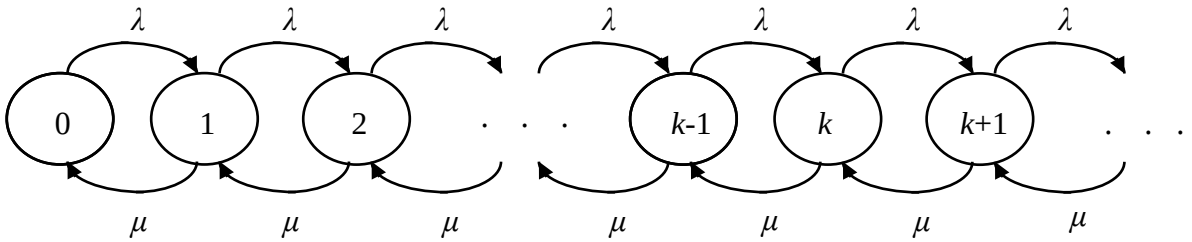
Το σύστημα αναμονής *M/M/1*, όπως φαίνεται από το συμβολισμό του, έχει εκθετική κατανομή της διαδικασίας των χρόνων μεταξύ διαδοχικών αφίξεων «πελατών», εκθετική κατανομή των χρόνων εξυπηρέτησης των «πελατών», έναν εξυπηρετητή και άπειρο μήκος ουράς. Επίσης υποθέτουμε FCFS πολιτική εξυπηρέτησης. Τα υπόλοιπα μεγέθη που περιγράφονται στην υπο-παράγραφο 2 της παραγράφου 3.1.1 θεωρούνται με τις συνήθεις τιμές τους.

Το σύστημα *M/M/1* είναι και αυτό βέβαια, μια αλυσίδα Markov η οποία είναι διαδικασία γεννήσεων – θανάτων. Με τη (συνηθισμένη) παραδοχή ότι οι *ρυθμοί* αφίξεων και εξυπηρέτησης δεν εξαρτώνται από την κατάσταση (αριθμό παρόντων πελατών) του συστήματος, ισχύει:

$$\lambda_k = \lambda \qquad \text{για} \quad k = 0, 1, 2, \ldots \qquad (3.51)$$

$$\mu_k = \mu \qquad \text{για} \quad k = 1, 2, 3, \ldots$$

Στο Σχήμα 3.8 φαίνεται το διάγραμμα καταστάσεων – ρυθμών μεταβάσεων του συστήματος *M/M/1*.



Σχήμα 3.8. Διάγραμμα καταστάσεων-ρυθμών μεταβάσεων για το σύστημα *M/M/1*.

Για σύστημα με έναν εξυπηρετητή, όπως είναι το *M/M/1*, γνωρίζουμε, από τη σχέση (3.4), ότι η *χρησιμοποίηση* είναι:

$$\rho = \lambda \cdot \bar{x} = \frac{\lambda}{\mu} \qquad (3.52)$$

Η γενική συνθήκη σταθερότητας , λόγω της ισχύος των (3.51) για το σύστημα *M/M/1*, μεταφράζεται στη γνωστή από την παράγραφο 3.1.3. *συνθήκη σταθερότητας*:

$$0 < \rho = \frac{\lambda}{\mu} < 1 \qquad (3.53)$$

Εφαρμόζοντας τη γενική λύση των σχέσεων (3.38) και (3.39), παίρνουμε τη *λύση του M/M/1 στη μόνιμη κατάσταση*, η οποία εκφράζει την κατανομή του αριθμού πελατών στο σύστημα:

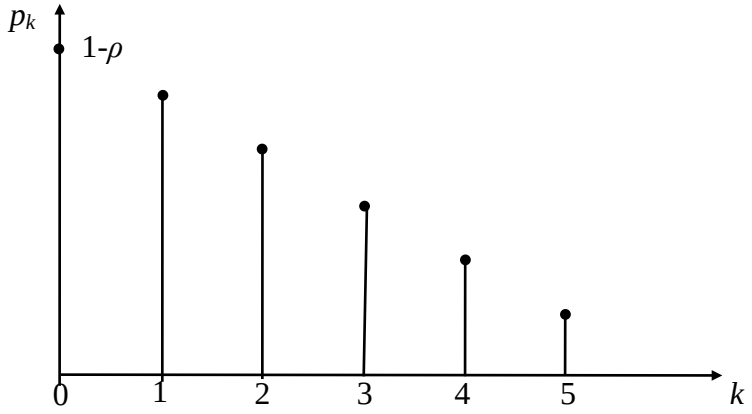
$$\boxed{p_k = (1-\rho)\cdot\rho^k \qquad \text{για} \quad k = 0, 1, 2, \ldots} \tag{3.54}$$

Η (3.54) έχει ενσωματωμένη τη σχέση που δίνει το p_0, δηλαδή την *πιθανότητα να είναι άδειο (ή άεργο) το σύστημα*:

$$p_0 = 1-\rho \tag{3.55}$$

Μπορείτε να παρατηρήσετε ότι η σχέση (3.55) είναι ουσιαστικά η σχέση (3.6), την οποία αποδείξαμε για τη γενική κατηγορία συστημάτων αναμονής *G/G/1*, στην οποία περιλαμβάνεται βέβαια, και το *M/M/1*.

Παρατηρώντας τη μορφή της λύσης (3.54), μπορούμε να δούμε καταρχήν ότι τα p_k ακολουθούν τη *γεωμετρική κατανομή*, καθώς και ότι εξαρτώνται από τα λ και μ, μόνο μέσω του λόγου τους ρ. Στο Σχήμα 3.9 φαίνεται η γραφική παράσταση των p_k.



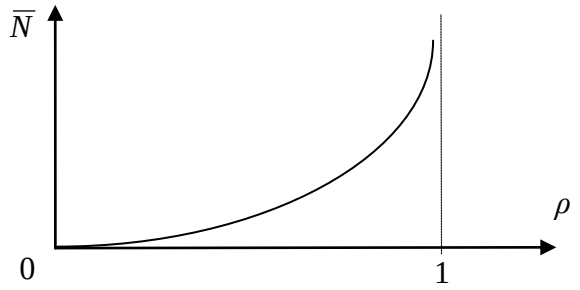
Σχήμα 3.9. Τα p_k στο σύστημα *M/M/1*.

Ορισμένες μετρικές απόδοσης που προκύπτουν εύκολα από τη λύση (3.54) είναι:

A. Μέσος αριθμός πελατών στο σύστημα $\overline{N}$:

$$\overline{N} = \sum_{k=0}^{\infty} k\,p_k = (1-\rho)\sum_{k=0}^{\infty} k\rho^k = (1-\rho)\frac{\rho}{(1-\rho)^2} = \frac{\rho}{1-\rho} \tag{3.56}$$

Στο Σχήμα 3.10 φαίνεται η συμπεριφορά του $\overline{N}$ συναρτήσει του ρ.



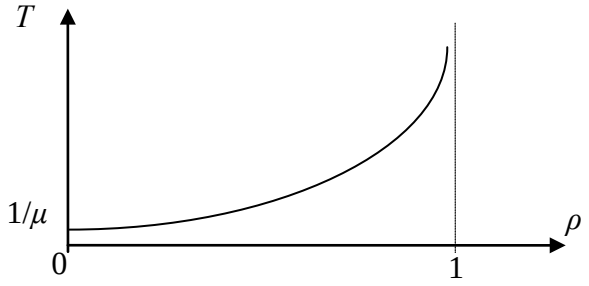
Σχήμα 3.10. Ο μέσος αριθμός πελατών στο *M/M/1*.

Β. Μέσος χρόνος ενός πελάτη στο σύστημα *T*:

Με χρήση του νόμου του Little παίρνουμε:

$$T = \frac{\overline{N}}{\lambda} = \frac{1/\mu}{1-\rho} \tag{3.57}$$

Στο Σχήμα 3.11 φαίνεται η συμπεριφορά του T συναρτήσει του ρ.



Σχήμα 3.11. Ο μέσος χρόνος συστήματος στο M/M/1.

Γ. Μέσος χρόνος αναμονής ενός πελάτη στην ουρά *W*:

$$W = T - \bar{x} = T - 1/\mu = \frac{\rho}{\mu \cdot (1-\rho)} \tag{3.58}$$

Δ. Μέσος αριθμός πελατών στην ουρά $\overline{N}_q$:

$$\overline{N}_q = \overline{N} - \rho = \frac{\rho^2}{1-\rho} \tag{3.59}$$

Ε. Πιθανότητα να υπάρχουν τουλάχιστον *n* πελάτες στο σύστημα $P_{(n)}$:

$P_{(n)} = \text{Prob}[\ n$ ή περισσότεροι πελάτες στο σύστημα$]$

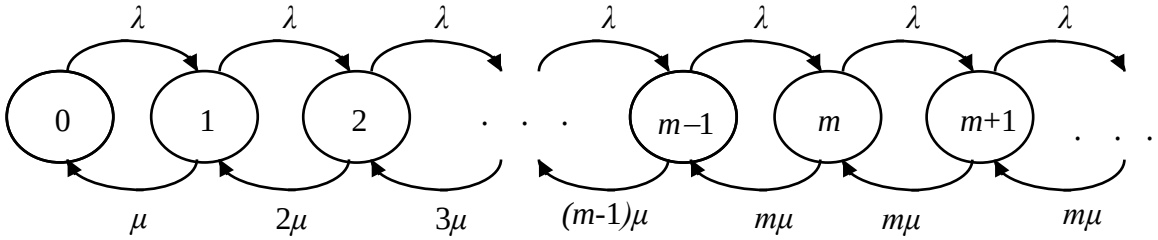
ή $$P_{(n)} = \sum_{k=n}^{\infty} p_k = (1-\rho)\sum_{k=n}^{\infty} \rho^k = (1-\rho)\rho^n \sum_{k=0}^{\infty} \rho^k = (1-\rho)\rho^n \frac{1}{1-\rho} = \rho^n \tag{3.60}$$

3.6.2 *M/M/m*. Σύστημα με *m* εξυπηρετητές

Το σύστημα αναμονής *M/M/m*, έχει και αυτό εκθετικές αφίξεις και εξυπηρετήσεις, αλλά διαθέτει *m* ίδιους εξυπηρετητές, ο καθένας από τους οποίους έχει ρυθμό εξυπηρέτησης μ. Τα υπόλοιπα χαρακτηριστικά του συστήματος είναι ίδια με αυτά του *M/M/1*. Οι ρυθμοί αφίξεων και εξυπηρέτησης του *M/M/m* είναι:

$$\lambda_k = \lambda \qquad \text{για} \quad k = 0, 1, 2, \ldots$$
$$\mu_k = \begin{cases} k\mu & \text{για} \quad 1 \le k \le m \\ m\mu & \text{για} \quad m \le k \end{cases} \tag{3.61}$$

Στο Σχήμα 3.12 φαίνεται το διάγραμμα καταστάσεων – ρυθμών μεταβάσεων του συστήματος *M/M/m*.



Σχήμα 3.12. Διάγραμμα καταστάσεων-ρυθμών μεταβάσεων για το σύστημα *M/M/m*.

Το διάγραμμα του σχήματος 3.12 αποτυπώνει τη λειτουργία του συστήματος: κάθε πελάτης που φθάνει, θα «πάρει» έναν εξυπηρετητή, όσο υπάρχουν διαθέσιμοι. Στη φάση αυτή το σύστημα λειτουργεί με όσους εξυπηρετητές είναι κατειλημμένοι (έστω *k*, αν υπάρχουν *k* πελάτες στο σύστημα), με *συνολικό* ρυθμό εξυπηρέτησης *km*. Αν βρεθούν περισσότεροι από *m* πελάτες στο σύστημα, οι *m* θα χρησιμοποιούν τους εξυπηρετητές και οι επιπλέον θα περιμένουν στην ουρά. Στη φάση αυτή το σύστημα λειτουργεί με το μέγιστο συνολικό ρυθμό εξυπηρέτησης $m\mu$.

Η *χρησιμοποίηση* του συστήματος είναι:

$$\rho = \frac{\lambda \bar{x}}{m} = \frac{\lambda}{m\mu} \tag{3.62}$$

Η *συνθήκη σταθερότητας* είναι τώρα:

$$0 < \rho = \frac{\lambda}{m\mu} < 1 \tag{3.63}$$

Εφαρμόζοντας τη γενική λύση των σχέσεων (3.38) και (3.39), με τους ρυθμούς των σχέσεων (3.61), παίρνουμε τη λύση του *M/M/m* στη μόνιμη κατάσταση:

$$p_k = \begin{cases} p_0 \dfrac{(m\rho)^k}{k!} & \text{για} \quad 1 \le k \le m \\ p_0 \dfrac{\rho^k m^m}{m!} & \text{για} \quad k \ge m \end{cases} \tag{3.64}$$

και

$$p_0 = \left[\sum_{k=0}^{m-1} \frac{(m\rho)^k}{k!} + \left(\frac{(m\rho)^m}{m!} \right) \left(\frac{1}{1-\rho} \right) \right]^{-1} \tag{3.65}$$

Ορισμένες μετρικές απόδοσης που προκύπτουν από τη λύση (3.64) είναι:

Α. Πιθανότητα να χρειαστεί να περιμένει στην ουρά ένας πελάτης, *Π*:

$\Pi =$ Prob[m ή περισσότεροι πελάτες στο σύστημα]

ή $$\Pi = \sum_{k=m}^{\infty} p_k = \sum_{k=m}^{\infty} p_0 \frac{\rho^k m^m}{m!} = p_0 \frac{m^m}{m!} \sum_{k=m}^{\infty} \rho^k = p_0 \frac{m^m}{m!} \rho^m \frac{1}{1-\rho} = p_0 \frac{(m\rho)^m}{m!(1-\rho)} \tag{3.66}$$

Β. Μέσος αριθμός πελατών στο σύστημα $\overline{N}$:

$$\overline{N} = \sum_{k=0}^{\infty} k p_k = m\rho + \frac{\rho \Pi}{1-\rho} \tag{3.67}$$

Γ. Μέσος χρόνος ενός πελάτη στο σύστημα *T*:

Με χρήση του νόμου του Little παίρνουμε:

$$T = \frac{\overline{N}}{\lambda} = \frac{1}{\mu} \left(1 + \frac{\Pi}{m(1-\rho)} \right) \tag{3.68}$$

Δ. Μέσος χρόνος αναμονής ενός πελάτη στην ουρά *W*:

$$W = T - \overline{x} = T - 1/\mu = \frac{\Pi}{m\mu(1-\rho)} \tag{3.69}$$

Ε. Μέσος αριθμός πελατών στην ουρά $\overline{N}_q$:

$$\overline{N}_q = \overline{N} - m\rho = \frac{\rho \Pi}{1-\rho} \tag{3.70}$$

3.6.3 M/M/1/K. Σύστημα με περιορισμένο μήκος ουράς

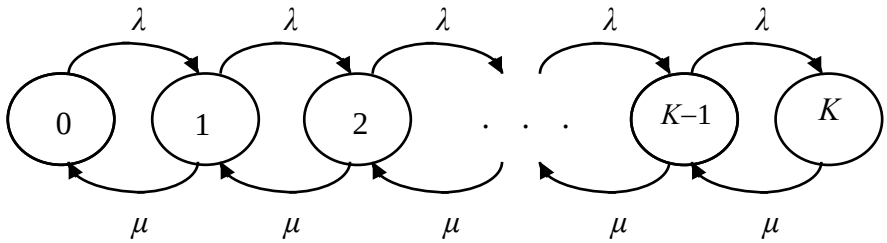
Το σύστημα αναμονής *M/M/1/K* έχει τα ίδια χαρακτηριστικά με αυτά του *M/M/1*, αλλά περιορισμένη χωρητικότητα σε πελάτες. Συγκεκριμένα, στο σύστημα μπορούν να βρίσκονται το πολύ *K* πελάτες (στην ουρά και στον εξυπηρετητή). Πελάτες που θα φθάνουν στο σύστημα και θα το βρίσκουν γεμάτο, θα «χάνονται» χωρίς να λαμβάνουν εξυπηρέτηση.

Οι ρυθμοί αφίξεων και εξυπηρέτησης του *M/M/1/K* είναι:

$$\lambda_k = \begin{cases} \lambda & \text{για} \quad 0 \le k < K \\ 0 & \text{για} \quad k \ge K \end{cases} \tag{3.71}$$

$$\mu_k = \begin{cases} \mu & \text{για} \quad 1 \le k \le K \\ 0 & \text{για} \quad k > K \end{cases}$$

Στο Σχήμα 3.13 φαίνεται το διάγραμμα καταστάσεων – ρυθμών μεταβάσεων του συστήματος *M/M/1/K*.



Σχήμα 3.13. Διάγραμμα καταστάσεων – ρυθμών μεταβάσεων για το σύστημα *M/M/1/K*.

Επειδή το σύστημα αυτό έχει περιορισμένη χωρητικότητα σε πελάτες, είναι *πάντα σταθερό*, ανεξάρτητα από τις τιμές των λ και μ, αφού δεν είναι δυνατό να δημιουργηθούν ανεξέλεγκτες ουρές.

Εφαρμόζοντας τη γενική λύση των σχέσεων (3.38) και (3.39), με τους ρυθμούς των σχέσεων (3.71), παίρνουμε τη λύση του *M/M/1/K* στη μόνιμη κατάσταση:

$$p_k = \begin{cases} \dfrac{1-\lambda/\mu}{1-(\lambda/\mu)^{K+1}} \left(\dfrac{\lambda}{\mu}\right)^k & \text{για} \quad 0 \le k \le K \\ 0 & \text{αλλιώς} \end{cases} \tag{3.72}$$

Η (3.72) περιλαμβάνει και τη λύση για το p_0 (την πιθανότητα να είναι άδειο το σύστημα), καθώς και για το p_K που εκφράζει την *πιθανότητα να είναι γεμάτο το σύστημα*.

Ορισμένες μετρικές απόδοσης που προκύπτουν από τη λύση (3.72) είναι:

Α. Μέσος αριθμός πελατών στο σύστημα $\overline{N}$:

$$\overline{N} = \sum_{k=0}^{K} k p_k = \frac{\lambda/\mu}{1-\lambda/\mu} - \frac{(K+1)(\lambda/\mu)^{K+1}}{1-(\lambda/\mu)^{K+1}} \tag{3.73}$$

Β. Μέσος αριθμός πελατών στην ουρά $\overline{N}_q$:

$$\overline{N}_q = \sum_{k=2}^{K} (k-1) p_k = \frac{\lambda/\mu}{1-\lambda/\mu} - (\lambda/\mu) \cdot \frac{1+K(\lambda/\mu)^K}{1-(\lambda/\mu)^{K+1}} \tag{3.74}$$

Παράδειγμα

Το *M/M/1/K* είναι ένα καλό μοντέλο για την τηλεφωνία, όπου όντως χάνονται κλήσεις. Το μοντέλο μιας τηλεφωνικής συσκευής μπορεί να περιγραφεί από το *M/M/1/1*. Δηλαδή έχουμε *K*=1 και από τον τύπο (3.72) προκύπτει:

$$p_k = \begin{cases} \dfrac{1}{1+\lambda/\mu} & \text{για} \quad k=0 \\ \dfrac{\lambda/\mu}{1+\lambda/\mu} & \text{για} \quad k=1 \\ 0 & \text{αλλιώς} \end{cases} \tag{3.75}$$

Στην παραπάνω σχέση (3.75), το p_0 εκφράζει την πιθανότητα κάποιος που καλεί έναν αριθμό τηλεφώνου, να μπορέσει να μιλήσει, ενώ το p_1 εκφράζει την πιθανότητα να βρει κατειλημμένη τη συσκευή. Το μέγεθος εισόδου λ εκφράζει το μέσο ρυθμό με τον οποίο γίνονται κλήσεις προς τη συσκευή, ενώ το μέγεθος εισόδου $\bar{x} = 1/\mu$ εκφράζει τη μέση χρονική διάρκεια μιας συνομιλίας στη συσκευή.

ΚΕΦΑΛΑΙΟ 4

ΔΙΚΤΥΑ ΣΥΣΤΗΜΑΤΩΝ ΑΝΑΜΟΝΗΣ

4.1 ΕΙΣΑΓΩΓΗ

Τα απλά συστήματα αναμονής που μελετήσαμε στο Κεφάλαιο 3, μπορούν να χρησιμοποιηθούν είτε ως μοντέλα μελέτης *ενός* πόρου κάποιου πληροφοριακού συστήματος, όπως η CPU, ένας δίσκος, ένα κανάλι μετάδοσης δεδομένων, ή ως μοντέλα για μια μακροσκοπική μελέτη του συστήματος θεωρώντας το έναν πόρο (π.χ. ένας Η/Υ, ένα δίκτυο υπολογιστών). Στην πρώτη περίπτωση περιοριζόμαστε σε ένα μόνο μικρό μέρος του συστήματος, ενώ στη δεύτερη χάνουμε σε λεπτομέρεια και δεν παίρνουμε πληροφορία για τις αλληλεπιδράσεις των διαφόρων τμημάτων του συστήματος.

Τα *δίκτυα συστημάτων αναμονής* (queueing networks), μπορούν να χρησιμοποιηθούν ως μοντέλα για πολύπλοκα πληροφοριακά συστήματα που περιέχουν περισσότερους από έναν πόρους και στα οποία οι «πελάτες» μετακινούνται από τον έναν πόρο στον άλλο. Κάθε πόρος έχει συγκεκριμένη μέγιστη δυνατότητα εξυπηρέτησης και τη δική του ουρά.

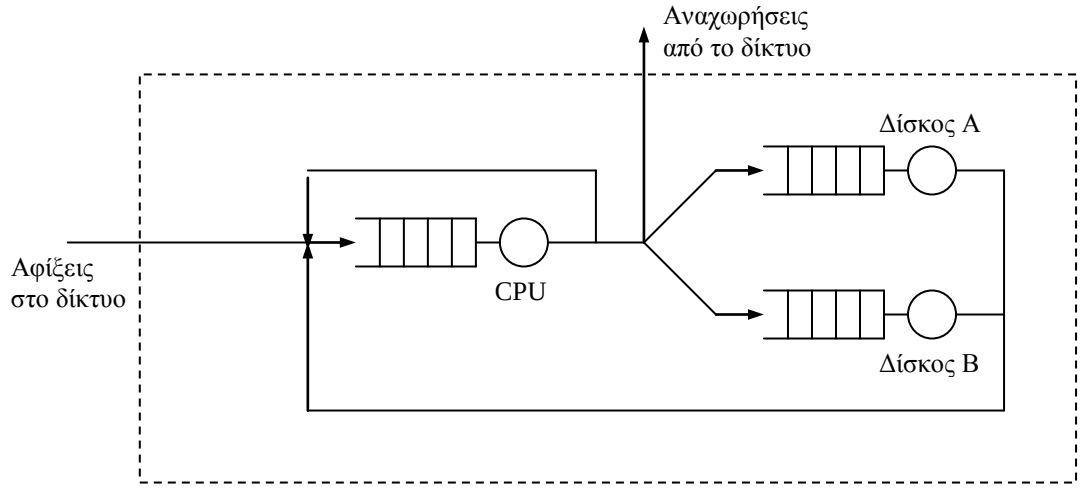
Δίκτυο συστημάτων αναμονής είναι μια συλλογή απλών συστημάτων αναμονής, στην οποία κάθε «πελάτης» που αναχωρεί από ένα απλό σύστημα, μετακινείται σε κάποιο άλλο (πιθανώς το ίδιο), ή αναχωρεί από το δίκτυο, αν αυτό επιτρέπεται.

Ανοικτό δίκτυο συστημάτων αναμονής είναι αυτό στο οποίο επιτρέπονται αφίξεις από το περιβάλλον του δικτύου και αναχωρήσεις προς το περιβάλλον.

Κλειστό δίκτυο είναι αυτό στο οποίο δεν επιτρέπονται αφίξεις και αναχωρήσεις, αλλά διατηρείται ένας σταθερός αριθμός N μετακινούμενων «πελατών».

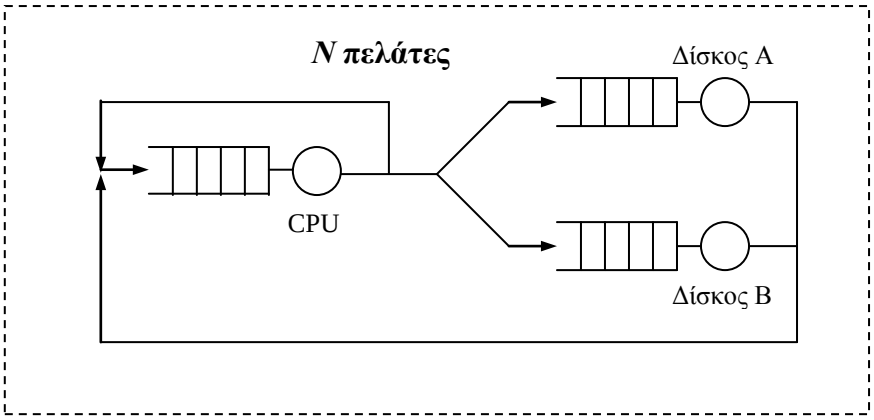
Παράδειγμα

Στο Σχήμα 4.1 φαίνεται ένα ανοικτό δίκτυο συστημάτων αναμονής ως μοντέλο για ένα απλό υπολογιστικό σύστημα, ενώ στο Σχήμα 4.2 φαίνεται ένα μοντέλο κλειστού δικτύου για το ίδιο σύστημα.



Σχήμα 4.1. Ένα μοντέλο ανοικτού δικτύου συστημάτων αναμονής.

Στο ανοικτό μοντέλο του σχήματος 4.1, οι εργασίες εισέρχονται στο σύστημα απαιτώντας εξυπηρέτηση από τη CPU. Μόλις πάρουν ένα ποσόν εξυπηρέτησης (κβάντο), είτε θα ζητήσουν δεδομένα από κάποιον από τους δύο δίσκους, είτε θα ξαναζητήσουν κβάντο εξυπηρέτησης από τη CPU, ή θα τερματίσουν τη χρήση του συστήματος. Όταν «βρίσκονται» οι αιτήσεις σε έναν από τους δίσκους, θα πάρουν τα δεδομένα που χρειάζονται και στη συνέχεια θα επανέλθουν για περαιτέρω επεξεργασία στη CPU.



Σχήμα 4.2. Ένα μοντέλο κλειστού δικτύου συστημάτων αναμονής.

Στο κλειστό μοντέλο του σχήματος 4.2, η λειτουργία μιας αίτησης είναι ίδια με αυτήν του ανοικτού δικτύου, με τη διαφορά ότι δεν έχουμε αφίξεις και αναχωρήσεις στο δίκτυο, αλλά θεωρούμε ότι κυκλοφορεί στο δίκτυο ένας σταθερός αριθμός αιτήσεων *N*. Ένα τέτοιο μοντέλο μπορεί να χρησιμοποιηθεί για τη μελέτη συστήματος που βρίσκεται σε σημείο κορεσμού, δηλαδή έχει συνεχώς το μέγιστο αριθμό αιτήσεων *N* που μπορεί συνολικά να εξυπηρετεί.

□

Η *επίλυση* των δικτύων συστημάτων αναμονής είναι στη γενική περίπτωση μια πολύπλοκη διαδικασία. Ευτυχώς όμως, υπάρχουν περιπτώσεις που η συνολική λύση προκύπτει εύκολα ως απλός συνδυασμός των λύσεων των επιμέρους απλών συστημάτων αναμονής που συγκροτούν το δίκτυο.

Ένα δίκτυο συστημάτων αναμονής, μπορεί να περιγραφεί από ένα σύνολο *M* σταθμών. Κάθε σταθμός αναπαριστά κάποιου είδους παροχή υπηρεσίας με τη βοήθεια c_i servers στον σταθμό i, $i = 1, 2, ..., M$. Στην πιο γενική περίπτωση, οι πελάτες μπορούν να αφιχθούν από το περιβάλλον στο δίκτυο σε οποιοδήποτε σταθμό και να εγκαταλείψουν το σύστημα από οποιοδήποτε σταθμό. Οι πελάτες μπορούν να εισέρχονται και να αναχωρούν από διαφορετικούς σταθμούς καθώς επίσης, εσωτερικά, μπορούν να διασχίσουν το σύστημα από διαφορετικά μονοπάτια κάθε φορά. Οι πελάτες μπορούν να επιστρέψουν σε σταθμούς που έχουν προηγουμένως επισκεφτεί, να παραλείψουν εντελώς κάποιους από αυτούς, ή ακόμα να επιλέξουν να παραμείνουν στο σύστημα για πάντα.

Θα μας απασχολήσουν κυρίως δίκτυα συστημάτων αναμονής με τα παρακάτω χαρακτηριστικά:

1. Οι αφίξεις από το περιβάλλον στο σταθμό i ακολουθούν μία διαδικασία Poisson με μέσο ρυθμό γ_i.
2. Οι χρόνοι εξυπηρέτησης είναι ίδιοι στους servers κάθε σταθμού, ανεξάρτητοι και εκθετικά κατανεμημένοι με παράμετρο μ_i.

3. Η πιθανότητα ένας πελάτης που τελείωσε την εξυπηρέτηση του στον σταθμό i, να πάει στο σταθμό j (routing probability), είναι r_{ij} (ανεξάρτητη από την κατάσταση του συστήματος) όπου, $i = 1, 2, ..., M$, $j = 1, 2, ..., M$. Η πιθανότητα r_{io} υποδεικνύει την πιθανότητα ο πελάτης να αναχωρήσει από το δίκτυο από το σταθμό i.

Δίκτυα που έχουν τις παραπάνω ιδιότητες αποκαλούνται Δίκτυα Jackson.

Περιπτώσεις όπου $\gamma_i = 0$ για όλα τα i (κανένας πελάτης δε μπορεί να εισέλθει στο δίκτυο), και $r_{io} = 0$ για όλα τα i (κανένας πελάτης δε μπορεί να αναχωρήσει από το σύστημα) αναφέρονται στα Κλειστά Jackson Δίκτυα.

4.2 ΑΝΟΙΧΤΑ JACKSON ΔΙΚΤΥΑ

Θεωρούμε ένα δίκτυο με *M* σταθμούς-κόμβους. Οι πελάτες μπορούν να αφιχθούν από το περιβάλλον σε οποιοδήποτε σταθμό του συστήματος ακολουθώντας μία Poisson διαδικασία. Αναπαριστούμε το μέσο ρυθμό εξωτερικών αφίξεων στο σταθμό i με γ_i. Όλοι οι servers στο σταθμό i δουλεύουν βάσει εκθετικής κατανομής με μέσο ρυθμό μ_i (όλοι οι servers, σε ένα κόμβο, είναι ίδιοι). Όταν ένας πελάτης ολοκληρώσει την εξυπηρέτηση του στο σταθμό i, πηγαίνει στον επόμενο, έστω j, με πιθανότητα r_{ij} (ανεξάρτητη από την κατάσταση του συστήματος). Υπάρχει, επίσης, η πιθανότητα r_{io} όπου ένας πελάτης θα εγκαταλείψει το σύστημα από τον κόμβο i, αφού έχει ολοκληρώσει την εξυπηρέτησή του. Δεν υπάρχει περιορισμός στο μέγεθος της ουράς ενός σταθμού, δηλαδή, ποτέ δεν μπορεί να υπάρξει μπλοκαρισμένο σύστημα ή σταθμός.

Δεδομένου ότι διαθέτουμε ένα σύστημα Markov μπορούμε να χρησιμοποιήσουμε τους γνωστούς τύπους ανάλυσης για να γράψουμε τις εξισώσεις *μόνιμης κατάστασης* του συστήματος. Από τη στιγμή που διάφοροι αριθμοί πελατών μπορούν να βρίσκονται σε διάφορους σταθμούς αναζητούμε τη δεσμευμένη πιθανότητα κατανομής για τον αριθμό των πελατών σε κάθε σταθμό. Πιο συγκεκριμένα, έστω N_i μία τυχαία μεταβλητή για τον αριθμό των πελατών στο σταθμό i. Στη μόνιμη κατάσταση αναζητούμε την πιθανότητα $\Pr\{N_1 = n_1, N_2 = n_2, ..., N_M = n_M\} \equiv p_{n_1, n_2, ..., n_M}$. Από αυτή τη δεσμευμένη πιθανότητα μπορούμε να λάβουμε την επί μέρους πιθανότητα κατανομής για τον αριθμό πελατών σε ένα συγκεκριμένο σταθμό αθροίζοντας κατάλληλα τις πιθανότητες.

Για να λάβουμε τις εξισώσεις μόνιμης κατάστασης για το δίκτυο αυτό, θα χρησιμοποιήσουμε το *νόμο ισορροπίας της ροής πιθανότητας* (παράγραφος 3.4.2). Προκειμένου να χρησιμοποιήσουμε ένα διάνυσμα διάστασης Μ για την περιγραφή μίας κατάστασης, $(n_1, n_2, ..., n_M)$, χρησιμοποιούμε την παρακάτω σημειογραφία:

Κατάσταση	Συμβολισμός
$n_1, n_2, ..., n_i, ..., n_j, ..., n_M$	$\bar{n}$
$n_1, n_2, ..., n_i + 1, ..., n_j, ..., n_M$	$\bar{n}; i^+$

$n_1, n_2, ..., n_i - 1, ..., n_j, ..., n_M$	$\bar{n}; i^-$
$n_1, n_2, ..., n_i + 1, ..., n_j - 1, ..., n_M$	$\bar{n}; i^+ j^-$

Πίνακας 4.1: Απλοποιημένη περιγραφή καταστάσεων

Για να μεταπέσουμε από την κατάσταση $\bar{n}; i^+$ στην κατάσταση $\bar{n}$, αρκεί μία αναχώρηση στο περιβάλλον από τον σταθμό i. Ομοίως, η κατάσταση $\bar{n}; i^-$ οδηγεί στην κατάσταση $\bar{n}$ με μία άφιξη από το περιβάλλον στο σταθμό i. Η κατάσταση $\bar{n}; i^+ j^-$ αλλάζει στην $\bar{n}$, με την μετακίνηση ενός πελάτη από το σταθμό i στο σταθμό j. Στις δύο πρώτες περιπτώσεις ο αριθμός πελατών μεταβάλλεται. Αντίθετα, στην περίπτωση όπου από την κατάσταση $\bar{n}; i^+ j^-$ περνάμε στην $\bar{n}$, ο αριθμός πελατών παραμένει σταθερός συνολικά στο σύστημα, αλλά η κατανομή του μεταβάλλεται.

Χρησιμοποιώντας στοχαστική ισορροπία για τον υπολογισμό της ροής προς την κατάσταση $\bar{n}$ και της ροής από την $\bar{n}$ προς τα έξω, και υποθέτοντας ότι $c_i = 1$ για όλους τους σταθμούς i (κάθε σταθμός έχει ένα server για την εξυπηρέτηση πελατών), λαμβάνουμε την εξίσωση στοχαστικής ροής προς – έξω από την κατάσταση $\bar{n}$:

$$\sum_{i=1}^{M} \gamma_i p_{\bar{n};i^-} + \sum_{\substack{j=1 \\ (i \neq j)}}^{M} \sum_{i=1}^{M} \mu_i r_{ij} p_{\bar{n};i^+ j^-} + \sum_{i=1}^{M} \mu_i r_{i0} p_{\bar{n};i^+} = \sum_{i=1}^{M} \mu_i (1 - r_{ii}) p_{\bar{n}} + \sum_{i=1}^{M} \gamma_i p_{\bar{n}}$$

(4.1)

Για την κατάσταση $\bar{0} = (0, 0, \ldots, 0)$, δηλαδή για την κατάσταση στην οποία όλοι οι σταθμοί είναι άδειοι, ισχύει η εξίσωση:

$$\sum_{i=1}^{M} \gamma_i p_{\bar{0}} = \sum_{i=1}^{M} \mu_i r_{i0} p_{\bar{0};i^+}$$

ενώ ισχύει και η προφανής εξίσωση

$$\sum_{\text{όλα τα } \bar{n}} p_{\bar{n}} = 1$$

Η παραπάνω ισότητες αναφέρονται στη μόνιμη κατάσταση. Υπενθυμίζεται ότι μόνο ένα γεγονός μπορεί να συμβαίνει κάθε χρονική στιγμή.

Το αριστερό μέλος της ισότητας (4.1) είναι η «ροή πιθανότητας προς την κατάσταση $\bar{n}$». Αντίστοιχα, το δεξί μέλος αντιστοιχεί στη «ροή πιθανότητας έξω από την $\bar{n}$».

Ας παρατηρήσουμε, πιο αναλυτικά, τα επί μέρους αθροίσματα της σχέσης (4.1):

- $\sum_{i=1}^{M} \gamma_i p_{\bar{n};i^-}$: Για τον όρο i του αθροίσματος (σταθμός i), παρατηρούμε ότι το σύστημα βρίσκεται στην κατάσταση $\bar{n}; i^-$ και περνά στην κατάσταση $\bar{n}$. Ουσιαστικά, λαμβάνει χώρα μία άφιξη από το περιβάλλον, στο σταθμό i. Η ροή πιθανότητας προς την $\bar{n}$ από το σταθμό i, στην περίπτωση εξωτερικών αφίξεων, είναι $\gamma_i p_{\bar{n};i^-}$.

- $\sum_{\substack{j=1 \\ (i\neq j)}}^{M}\sum_{i=1}^{M} \mu_i r_{ij} p_{\bar{n};i^+j^-}$: Για τους όρους i, j του αθροίσματος (σταθμοί i, j), το σύστημα βρίσκεται στην κατάσταση $\bar{n};i^+j^-$ και περνά στην κατάσταση $\bar{n}$. Λαμβάνει χώρα μία μετακίνηση από το σταθμό i στο σταθμό j. Η ροή πιθανότητας για μία τέτοια μετακίνηση, είναι το γινόμενο $\mu_i r_{ij} p_{\bar{n};i^+j^-}$.

- $\sum_{i=1}^{M} \mu_i r_{i0} p_{\bar{n};i^+}$: Για τον όρο i του αθροίσματος (σταθμός i), παρατηρούμε ότι το σύστημα βρίσκεται στην κατάσταση $\bar{n};i^+$ και περνά στην κατάσταση $\bar{n}$. Στην περίπτωση αυτή συμβαίνει μία αναχώρηση προς το περιβάλλον. Η ροή πιθανότητας για μία τέτοια αναχώρηση είναι $\mu_i r_{i0} p_{\bar{n};i^+}$.

- $\sum_{i=1}^{M} \mu_i (1-r_{ii}) p_{\bar{n}}$: Για τον όρο i του αθροίσματος (σταθμός i), παρατηρούμε ότι το σύστημα βρίσκεται στην κατάσταση $\bar{n}$ και περνά στην κατάσταση $\bar{n};i^-$ ή στην κατάσταση $\bar{n};i^-j^+$. Γίνεται αναφορά στις αναχωρήσεις από το σταθμό i, εφόσον αυτός εξυπηρετήσει κάποιον πελάτη. Οι αναχωρήσεις αυτές μπορεί να είναι αναχωρήσεις στο περιβάλλον, ή μετακινήσεις σε κάποιον άλλο σταθμό, εκτός του i. Πράγματι, η πιθανότητα $1-r_{ii}$ αναφέρεται στην πιθανότητα αναχώρησης από το σταθμό i χωρίς να υπάρξει ανάδραση στο σύστημα. Η ροή πιθανότητας για μία τέτοια αναχώρηση είναι $\mu_i (1-r_{ii}) p_{\bar{n}}$.

- $\sum_{i=1}^{M} \gamma_i p_{\bar{n}}$: Για τον όρο i του αθροίσματος (σταθμός i), παρατηρούμε ότι το σύστημα βρίσκεται στην κατάσταση $\bar{n}$ και περνά στην κατάσταση $\bar{n};i^+$. Στην περίπτωση αυτή εξετάζεται η περίπτωση όπου λαμβάνει χώρα μία άφιξη στο σταθμό i. Η άφιξη αυτή, έρχεται από το περιβάλλον. Η ροή πιθανότητας για την άφιξη αυτή είναι $\gamma_i p_{\bar{n}}$.

4.2.1 Η λύση μορφής γινομένου (product form solution)

Έστω λ_i ο συνολικός *ενεργός (effective) μέσος ρυθμός αφίξεων στο σταθμό* i (από το περιβάλλον και από άλλους σταθμούς). Ισχύει :

$$\lambda_i = \gamma_i + \sum_{j=1}^{M} r_{ji} \lambda_j \tag{4.2}$$

Το λ_i προκύπτει από το άθροισμα του μέσου ρυθμού αφίξεων των εξωτερικών αφίξεων με το μέσο ρυθμό αφίξεων εσωτερικά επί την πιθανότητα οι πελάτες εσωτερικά να πάνε στο σταθμό i.

Ορίζεται, $\rho_i = \frac{\lambda_i}{\mu_i}$ για $i = 1, 2, ..., M$. Ο Jackson απέδειξε ότι η λύση στη σταθερή κατάσταση στην εξίσωση στοχαστικής ροής (4.1) είναι:

$$p_{\bar{n}} \equiv p_{n_1, n_2, ..., n_M} = (1-\rho_1)\rho_1^{n_1}(1-\rho_2)\rho_2^{n_2}...(1-\rho_M)\rho_M^{n_M} \tag{4.3}$$

Παρατηρούμε ότι το δίκτυο *συμπεριφέρεται* σαν κάθε σταθμός του να είναι ένα ανεξάρτητο *M/M/1* με παραμέτρους λ_i και μ_i έτσι ώστε η δεσμευμένη πιθανότητα κατανομής να μπορεί να γραφεί σαν γινόμενο των επί μέρους *M/M/1*. Επισημαίνεται ότι το δίκτυο δεν μπορεί να αναλυθεί σε ανεξάρτητα *M/M/1* των οποίων οι αφίξεις να ακολουθούν διαδικασία Poisson με μέσο ρυθμό λ_i. Στην πραγματικότητα, αποδεικνύεται ότι η ενεργή εσωτερική ροή σε τέτοιου είδους δίκτυα *δεν* είναι Poisson, εξαιτίας των αναδράσεων που μπορεί να υπάρξουν. Παρ' όλ' αυτά, όπως αναφέρθηκε και παραπάνω, το δίκτυο συμπεριφέρεται σαν να αποτελείται από ανεξάρτητα *M/M/1*, ανεξάρτητα αν οι εσωτερικές ροές είναι Poisson ή όχι.

Θα αποδείξουμε ότι η λύση του Jackson ικανοποιεί πράγματι την εξίσωση στοχαστικής ροής του συστήματος (4.1).

Αρχικά θα δείξουμε ότι η $p_{\bar{n}} \equiv C\rho_1^{n_1}\rho_2^{n_2}...\rho_M^{n_M}$ ικανοποιεί την εξίσωση στοχαστικής ροής (4.1) και μετά ότι το C είναι ίσο με, $C = \prod_{i=1}^{M}(1-\rho_i)$. Έστω, $R_{\bar{n}} \equiv \rho_1^{n_1}\rho_2^{n_2}...\rho_M^{n_M}$. Βάζουμε την $p_{\bar{n}} = CR_{\bar{n}}$ στην εξίσωση στοχαστικής ροής (4.1) και η εξίσωση γίνεται:

$$\sum_{i=1}^{M}\gamma_i CR_{\bar{n};i^-} + \sum_{j=1}^{M}\sum_{\substack{i=1\\(i\neq j)}}^{M}\mu_i r_{ij} CR_{\bar{n};i^+j^-} + \sum_{i=1}^{M}\mu_i r_{i0} CR_{\bar{n};i^+} \overset{?}{=} \sum_{i=1}^{M}\mu_i(1-r_{ii})CR_{\bar{n}} + \sum_{i=1}^{M}\gamma_i CR_{\bar{n}} \Leftrightarrow$$

$$CR_{\bar{n}}\sum_{i=1}^{M}\frac{\gamma_i}{\rho_i} + CR_{\bar{n}}\sum_{j=1}^{M}\sum_{\substack{i=1\\(i\neq j)}}^{M}\mu_i r_{ij}\frac{\rho_i}{\rho_j} + CR_{\bar{n}}\sum_{i=1}^{M}\mu_i r_{i0}\rho_i \overset{?}{=} CR_{\bar{n}}\sum_{i=1}^{M}\mu_i(1-r_{ii}) + CR_{\bar{n}}\sum_{i=1}^{M}\gamma_i \Leftrightarrow$$

$$\sum_{i=1}^{M}\frac{\gamma_i\mu_i}{\lambda_i} + \sum_{j=1}^{M}\sum_{\substack{i=1\\(i\neq j)}}^{M}\mu_i r_{ij}\frac{\lambda_i\mu_j}{\lambda_j\mu_i} + \sum_{i=1}^{M}\mu_i r_{i0}\frac{\lambda_i}{\mu_i} \overset{?}{=} \sum_{i=1}^{M}(\mu_i - \mu_i r_{ii} + \gamma_i)$$

(4.4)

Για το λ_j έχουμε από την (4.2):

$$\lambda_j = \gamma_j + \sum_{\substack{i=1\\(i\neq j)}}^{M} r_{ij}\lambda_i + r_{jj}\lambda_j$$

ή

$$\sum_{\substack{i=1\\(i\neq j)}}^{M} r_{ij}\lambda_i = \lambda_j - \gamma_j - r_{jj}\lambda_j$$

Έτσι, με αντικατάσταση του αθροίσματος $\sum_{\substack{i=1\\(i\neq j)}}^{M} r_{ij}\lambda_i$ στην εξίσωση (4.4), η εξίσωση δίνει:

$$\sum_{i=1}^{M}\frac{\gamma_i\mu_i}{\lambda_i} + \sum_{j=1}^{M}\frac{\mu_j}{\lambda_j}\{\lambda_j - \gamma_j - r_{jj}\lambda_j\} + \sum_{i=1}^{M}\mu_i r_{i0}\frac{\lambda_i}{\mu_i} \overset{?}{=} \sum_{i=1}^{M}\{\mu_i - \mu_i r_{ii} + \gamma_i\}.$$

Με αλλαγή του δείκτη από j σε i του δεύτερου όρου του πρώτου μέλους στην παραπάνω σχέση, προκύπτει:

$$\sum_{i=1}^{M}\{\frac{\gamma_i\mu_i}{\lambda_i}+\frac{\mu_i}{\lambda_i}(\lambda_i-\gamma_i-r_{ii}\lambda_i)+\lambda_i r_{i0}\}\overset{?}{=}\sum_{i=1}^{M}\{\mu_i-\mu_i r_{ii}+\gamma_i\}\,.$$

Μετά από πράξεις, έχουμε:

$$\sum_{i=1}^{M}\{(\mu_i-\mu_i r_{ii})+\lambda_i r_{i0}\}\overset{?}{=}\sum_{i=1}^{M}\{\mu_i-\mu_i r_{ii}+\gamma_i\}\,.$$

Με απαλοιφή ομοίων όρων, καταλήγουμε στη σχέση:

$$\sum_{i=1}^{M}\lambda_i r_{i0}\overset{?}{=}\sum_{i=1}^{M}\gamma_i\,.$$

Πράγματι, η παραπάνω σχέση ισχύει. Το αριστερό μέλος αναπαριστά το συνολικό μέσο ρυθμό ροής προς το εξωτερικό περιβάλλον, ενώ, το δεξί αναπαριστά το συνολικό μέσο ρυθμό από το περιβάλλον προς το δίκτυο. Στη μόνιμη κατάσταση οι ρυθμοί αυτοί είναι ίσοι.

Για να βρούμε το C, έχουμε:

$$\sum_{\gamma\iota\alpha\ o\lambda\alpha\ \tau\alpha\ \bar{n}} p_{\bar{n}}=1\Leftrightarrow\ \sum_{n_M=0}^{\infty}\ldots\sum_{n_2=0}^{\infty}\ \sum_{n_1=0}^{\infty}C\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_{\mathrm{M}}^{n_M}=1\Leftrightarrow$$

$$C=[\sum_{n_M=0}^{\infty}\ldots\sum_{n_2=0}^{\infty}\ \sum_{n_1=0}^{\infty}\rho_1^{n_1}\rho_2^{n_2}\ldots\rho_{\mathrm{M}}^{n_M}\,]^{-1}\Leftrightarrow$$

$$C=[\sum_{n_1=0}^{\infty}\rho_1^{n_1}\sum_{n_2=0}^{\infty}\rho_2^{n_2}\ldots\sum_{n_M=0}^{\infty}\rho_{\mathrm{M}}^{n_M}\,]^{-1}\Leftrightarrow$$

$$C=[\frac{1}{1-\rho_1}\cdot\frac{1}{1-\rho_2}\cdot\ldots\cdot\frac{1}{1-\rho_M}]^{-1}$$

$(\rho_i<1,\ \ i=1,2,\ldots,M)$

Συνεπώς,

$$C=(1-\rho_1)\ldots(1-\rho_2)(1-\rho_M)=\prod_{i=1}^{M}(1-\rho_i)\qquad(\rho_i<1,\ \ i=1,2,\ldots,M)$$

Τα παραπάνω αποτελέσματα μπορούν εύκολα να γενικευτούν σε σταθμούς εργασίας με c_i >1 servers.

4.3 ΚΛΕΙΣΤΑ JACKSON ΔΙΚΤΥΑ

Θα μελετήσουμε τώρα την περίπτωση που κανένας πελάτης δεν μπορεί να εισέλθει στο σύστημα και κανένας πελάτης δεν μπορεί να εξέλθει από αυτό. Ισχύει ,ότι $\gamma_i=0$ και $r_{i0}=0$ για κάθε i και το δίκτυο αυτό είναι ένα κλειστό δίκτυο Jackson. Επίσης, έχουμε ένα πεπερασμένο αριθμό από N πελάτες, που συνεχώς ταξιδεύουν μέσα στο δίκτυο. Αν $c_i=1$ για κάθε i, παίρνουμε τις εξισώσεις της εξισορρόπησης ροής για τη μόνιμη κατάστασης από την εξίσωση (4.1), που ορίστηκε για το μοντέλο των ανοιχτών δικτύων, θέτοντας $\gamma_i= r_{i0}=0$. Έτσι, προκύπτει:

$$\sum_{\substack{j=1\\(j\neq i)}}^{M}\sum_{i=1}^{M}\mu_i r_{ij}\, p_{\bar{n};i^+j^-}=\sum_{i=1}^{M}\mu_i\left(1-r_{ii}\right)p_{\bar{n}} \tag{4.5}$$

Ο Gordon κι ο Newell (1967) απέδειξαν ότι η λύση είναι της μορφής:

$$p_{\bar{n}}=C\rho_1^{\,n_1}\rho_2^{\,n_2}\ldots\rho_M^{\,n_M}\equiv C\Re_{\bar{n}} \tag{4.6}$$

όπου το ρ_i πρέπει να ικανοποιεί τις εξισώσεις εξισορρόπησης ροής σε κάθε κόμβο i, έτσι ώστε η ροή που εισέρχεται στον κόμβο i να είναι ίση με τη ροή που εξέρχεται από τον κόμβο i. Δηλαδή,

$$\mu_i\rho_i=\sum_{j=1}^{M}\mu_j r_{ji}\rho_j \tag{4.7}$$

Για να δείξουμε ότι η (4.6) είναι μία λύση, αντικαθιστούμε την (4.6) στην (4.5) κι έχουμε

$$\sum_{\substack{j=1\\(j\neq i)}}^{M}\sum_{i=1}^{M}C\Re_{\bar{n}}\mu_i r_{ij}\frac{\rho_i}{\rho_j}=\sum_{i=1}^{M}\mu_i\left(1-r_{ii}\right)C\Re_{\bar{n}}$$

ή ισοδύναμα

$$-\sum_{i=1}^{M}\mu_i r_{ii}+\sum_{j=1}^{M}\frac{1}{\rho_j}\sum_{i=1}^{M}\mu_i r_{ij}\rho_i=\sum_{i=1}^{M}\mu_i-\sum_{i=1}^{M}\mu_i r_{ii}$$

από την οποία μετά από υπολογισμούς και με χρήση της (4.7) καταλήγουμε στη σχέση

$$\sum_{j=1}^{M}\frac{1}{\rho_j}\mu_j\rho_j=\sum_{i=1}^{M}\mu_i$$

ή απλούστερα

$$\sum_{j=1}^{M}\mu_j=\sum_{i=1}^{M}\mu_i$$

που προφανώς ισχύει.

Σ' αυτή την περίπτωση, το C πρέπει να υπολογιστεί από τη σχέση

$$\sum_{n_1+n_2+\ldots+n_M=N}C\rho_1^{\,n_1}\rho_2^{\,n_2}\ldots\rho_M^{\,n_M}=1$$

ή ισοδύναμα

$$C = \left[\sum_{n_1+n_2+\ldots+n_M=N} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_M^{n_M} \right]^{-1} \tag{4.8}$$

όπου το άθροισμα λαμβάνεται με όλους τους πιθανούς τρόπους με τους οποίους μπορούν να κατανεμηθούν τα *N* στοιχεία στους *M* κόμβους. Η λύση αυτή συχνά γράφεται ως *G(N)=1/C*, ώστε

$$p_{\bar{n}} = p_{n_1,n_2,\ldots,n_M} = \frac{1}{G(N)} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_M^{n_M} \tag{4.9}$$

όπου

$$G(N) = \sum_{n_1+n_2+\ldots+n_M=N} \rho_1^{n_1} \rho_2^{n_2} \ldots \rho_M^{n_M} \tag{4.10}$$

Γενικά, υπάρχουν πολλοί πιθανοί τρόποι να κατανεμηθούν *N* στοιχεία σε *M* κόμβους και συγκεκριμένα $\binom{N+M-1}{N}$ τρόποι, με αποτέλεσμα ο υπολογισμός του *G(N)* να γίνεται δύσκολος για μεγάλα *N* και *M*. Οπότε, θα ήταν ιδιαίτερα χρήσιμος ένα αποτελεσματικός αλγόριθμος που να υπολογίζει το *G(N)*. Ο αλγόριθμος που το επιτυγχάνει αυτό είναι *ο αλγόριθμος του Buzen* (1973), ο οποίος παρουσιάζεται παρακάτω.

4.3.1 Ο Αλγόριθμος του Buzen.

Τα ρ_i είναι οι *σχετικές χρησιμοποιήσεις* των σταθμών του δικτύου. Θέτουμε κάποιο $\rho_i = 1$ για να βρούμε τα υπόλοιπα σε σχέση με αυτό, από τις σχέσεις (4.7).

Ορίζουμε τη βοηθητική συνάρτηση

$$g_m(n) = \sum_{n_1+n_2+\ldots+n_m=n} \prod_{i=1}^{m} \rho_i^{n_i} \tag{4.11}$$

Σημειώνεται ότι η $g_m(n) = G(N)$, αν $m=M$ και $n=N$, δηλαδή $g_M(N) = G(N)$.

Είναι

$$g_m(n) = \sum_{n_1+n_2+\ldots+n_m=n} \prod_{i=1}^{m} \rho_i^{n_i} = \sum_{n_1+n_2+\ldots+n_m=n \,/\, n_m=0} \prod_{i=1}^{m} \rho_i^{n_i} + \sum_{n_1+n_2+\ldots+n_m=n \,/\, n_m>0} \prod_{i=1}^{m} \rho_i^{n_i}$$

$$= \sum_{n_1+n_2+\ldots+n_m=n \,/\, n_m=0} \prod_{i=1}^{m} \rho_i^{n_i} + \rho_m \sum_{n_1+n_2+\ldots+n_m=n-1} \prod_{i=1}^{m} \rho_i^{n_i}$$

Δηλαδή,

$$g_m(n) = g_{m-1}(n) + \rho_m \cdot g_m(n-1) \tag{4.12}$$

Οι αρχικές τιμές για την $g_m(n)$ είναι:

$$\begin{cases} g_0(n) = 0 & n = 1,2,\ldots,N \\ g_m(0) = 1 & m = 1,2,\ldots,M \end{cases}$$

Ο αλγόριθμος του Buzen είναι ως έξης:

For $m \leftarrow 1$ to M do

For $n \leftarrow 1$ to N do

$G(n) \leftarrow G(n) + \rho(m) \cdot G(n-1)$

Ο τρόπος που εφαρμόζεται ο αλγόριθμος φαίνεται στον παρακάτω πίνακα:

Σταθμοί / Πελάτες	ρ_1	ρ_2	.	ρ_{m-1}	ρ_m	.	.	ρ_M
0	$1 = g_1(0)$	$1 = g_2(0)$	.	.	$1 = g_m(0)$	.	.	$1 = g_M(0) = G(0)$
1	$\rho_1 = g_1(1)$	$g_2(1)$	.	.	.	.	.	$g_M(1) = G(1)$
2	$\rho_1^{\,2} = g_1(2)$	$g_2(2)$	.	.	.	.	.	$g_M(2) = G(2)$
3	$\rho_1^{\,3} = g_1(3)$	$g_2(3)$	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
n-1	.	.	.	.	$g_m(n-1)$ $\downarrow \times \rho_m$	.	.	.
n	$\rho_1^{\,n} = g_1(n)$	.	.	$g_{m-1}(n) \rightarrow$	$g_m(n)$	.	.	.
.	.	.	.	.	.	.	.	.
N-1	.	.	.	.	.	.	.	$g_M(N-1) = G(N-1)$
N	$\rho_1^{\,N} = g_1(N)$	.	.	.	.	.	.	$g_M(N) = G(N)$

Πίνακας 4.2: Υλοποίηση του αλγορίθμου του Buzen

Αξιοποιώντας τον αλγόριθμο του Buzen, μπορούμε να υπολογίσουμε εύκολα τις παρακάτω μετρικές, αφού τα *G(1), G(2),…, G(N)* είναι πλέον γνωστά.

- $$P(n_i \geq j) = \sum_{n_1+n_2+\cdots+n_M=N \,/\, n_i \geq j} \frac{\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_M^{n_M}}{G(N)}$$

$$= \rho_i^{\,j} \sum_{n_1+n_2+\cdots+n_M=N-j} \frac{\rho_1^{n_1}\rho_2^{n_2}\cdots\rho_M^{n_M}}{G(N)}$$

$$= \rho_i^{\,j} \frac{G(N-j)}{G(N)}$$

- $$P(n_i = j) = P(n_i \geq j) - P(n_i \geq j+1)$$

$$= \frac{\rho_i^{\,j}}{G(N)}\left[G(N-j) - \rho_i \cdot G(N-j-1)\right]$$

- $$P(n_i \geq j, n_k \geq l) = \rho_i^{\,j}\rho_k^{\,l} \frac{G(N-j-l)}{G(N)}$$

- Η χρησιμοποίηση είναι:

$$U_i = P(n_i \geq 1) = \rho_i \frac{G(N-1)}{G(N)}$$

- Ο μέσος αριθμός εργασιών σε ένα σταθμό, είναι:

$$E[n_i] = \sum_{j=1}^{N} P(n_i \geq j) = \sum_{j=1}^{N} \rho_i^{\,j} \frac{G(N-j)}{G(N)}$$

ΚΕΦΑΛΑΙΟ 5
ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΠΡΟΣΟΜΟΙΩΣΗ

5.1 ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΜΟΝΤΕΛΑ

Ένα **σύστημα** ορίζεται ως μία συλλογή οντοτήτων (π.χ. άνθρωποι ή μηχανές) που ενεργούν και αλληλεπιδρούν, με στόχο κάποιο λογικό τερματισμό.

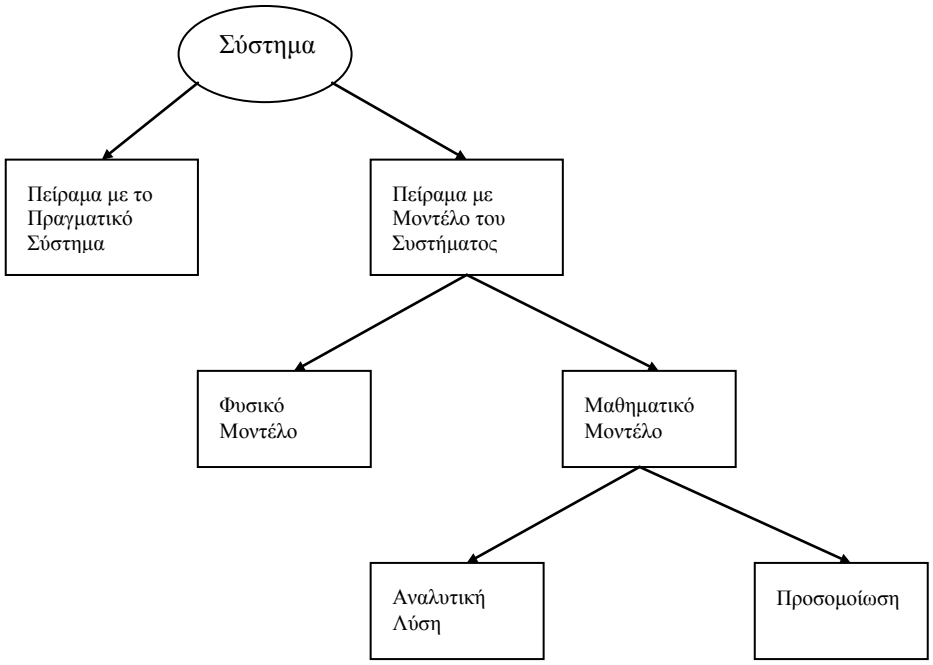
Στην πράξη, η έννοια του συστήματος εξαρτάται από τους στόχους της συγκεκριμένης μελέτης. Η συλλογή οντοτήτων που συγκροτούν ένα σύστημα για μία μελέτη, μπορεί να είναι μόνο ένα υποσύνολο του συνολικού συστήματος για μία άλλη μελέτη.

Ορίζουμε την *κατάσταση* ενός συστήματος ως τη συλλογή των μεταβλητών που είναι απαραίτητες για την περιγραφή του συστήματος σε μια χρονική στιγμή, αναφορικά με τους στόχους της μελέτης.

Τα συστήματα μπορούν να είναι διακριτά (discrete) ή συνεχή (continuous). *Διακριτό* σύστημα είναι αυτό στο οποίο οι μεταβλητές κατάστασης μπορούν να αλλάξουν σε διακεκριμένες στιγμές του χρόνου. *Συνεχές* είναι το σύστημα του οποίου οι μεταβλητές κατάστασης αλλάζουν συνεχώς στο χρόνο. Λίγα συστήματα στην πράξη είναι εξ ολοκλήρου διακριτά ή συνεχή, αλλά επειδή συνήθως μία από τις δύο ιδιότητες κυριαρχεί, μπορούμε τις περισσότερες φορές να χαρακτηρίσουμε το σύστημα ως διακριτό ή συνεχές.

Κατά τη διάρκεια της ζωής ενός συστήματος, παρουσιάζεται η ανάγκη μελέτης του, ώστε να αποκτήσουμε γνώση για τις σχέσεις μεταξύ των διαφόρων τμημάτων του, ή για να προβλέψουμε την απόδοσή του κάτω από νέες, πιθανόν, συνθήκες. Το Σχήμα 5.1 παρουσιάζει τους διαφορετικούς τρόπους με τους οποίους μπορεί να μελετηθεί ένα σύστημα.

- *Πείραμα με το Πραγματικό Σύστημα* **ή** *Πείραμα με Μοντέλο του Συστήματος:* Αν είναι δυνατόν (και οικονομικά αποδεκτό) να έχουμε φυσική πρόσβαση στο σύστημα και να το λειτουργήσουμε κάτω από τις νέες συνθήκες, ίσως είναι προτιμότερο να το κάνουμε, αφού στην περίπτωση αυτή δεν υπάρχει αμφισβήτηση για την ακρίβεια της μελέτης. Πάντως, σπάνια μας δίνεται αυτή η ευκαιρία, είτε γιατί κοστίζει, ή γιατί θα προκαλέσει προβλήματα στην κανονική λειτουργία του συστήματος. Επίσης, στη φάση σχεδιασμού, το σύστημα δεν υπάρχει, οπότε αν θέλουμε να μελετήσουμε τις διάφορες εναλλακτικές προτάσεις υλοποίησής του, είναι αναγκαίο να κατασκευάσουμε ένα *μοντέλο,* το οποίο θα αναπαριστά το σύστημα και θα το αντικαθιστά κατά τη μελέτη. Όταν χρησιμοποιούμε ένα μοντέλο, η *εγκυρότητά* του, δηλαδή ο βαθμός ακρίβειας με τον οποίο αναπαριστά το σύστημα, είναι πάντα ένα κρίσιμο ερώτημα.



Σχήμα 5.1. Τρόποι Μελέτης ενός Συστήματος.

- *Φυσικό Μοντέλο* **ή** *Μαθηματικό Μοντέλο:* Οι περισσότεροι άνθρωποι, με τη λέξη "μοντέλο" αντιλαμβάνονται ένα *φυσικό* μοντέλο, όπως για παράδειγμα τη μινιατούρα ενός πλοίου που δοκιμάζεται σε μια δεξαμενή νερού, για τη μελέτη των υδροδυναμικών ιδιοτήτων του πραγματικού συστήματος (πλοίου). Όμως, η μεγάλη πλειοψηφία των μοντέλων και ιδιαίτερα αυτά που χρησιμοποιούνται για τη μελέτη υπολογιστικών συστημάτων, είναι μαθηματικά. Ένα *μαθηματικό* μοντέλο αναπαριστά το σύστημα με βάση λογικές και ποσοτικές σχέσεις οι οποίες χειριζόμενες και μεταβαλλόμενες κατάλληλα, επιτρέπουν να δούμε πώς αντιδρά το μοντέλο και κατά συνέπεια πώς θα αντιδρούσε το σύστημα, αν το μαθηματικό μοντέλο είναι έγκυρο. Ένα παράδειγμα μαθηματικού μοντέλου είναι η γνωστή σχέση $d = rt$, όπου r είναι η ταχύτητα ταξιδίου, t είναι ο χρόνος ταξιδίου και d είναι η απόσταση που έχει διανυθεί.
- *Αναλυτική Λύση* **ή** *Προσομοίωση:* Από τη στιγμή που έχει δημιουργηθεί ένα μαθηματικό μοντέλο, θα πρέπει να εξετασθεί ο τρόπος χρήσης του ώστε να μπορεί να απαντήσει στα ερωτήματα που μας ενδιαφέρουν για το σύστημα που υποτίθεται ότι αντιπροσωπεύει. Αν το μοντέλο είναι αρκετά απλό, είναι δυνατό να πάρουμε μία ακριβή, *αναλυτική λύση*. Στο παράδειγμα της προηγούμενης παραγράφου, αν γνωρίζουμε την απόσταση που θα διανυθεί και την ταχύτητα, μπορούμε από το μοντέλο να πάρουμε $t = d/r$ για το χρόνο που θα χρειασθεί για το ταξίδι. Το μοντέλο αυτό είναι πολύ απλό, πράγμα που δεν συμβαίνει βέβαια στις περισσότερες περιπτώσεις, στις οποίες η αναλυτική λύση μπορεί να είναι αρκετά πολύπλοκη και να απαιτεί μεγάλη υπολογιστική ισχύ. Πάντως, αν υπάρχει αναλυτική λύση στο μαθηματικό μοντέλο και είναι υπολογιστικά αποδοτική, συνήθως την προτιμούμε από την προσομοίωση. Όμως πολλά συστήματα είναι πολύπλοκα, όπως και τα μαθηματικά μοντέλα που τα αναπαριστούν, έτσι ώστε να αποκλείουν κάθε πιθανότητα αναλυτικής λύσης. Στην περίπτωση αυτή το μοντέλο πρέπει να μελετηθεί με τη χρήση ***προσομοίωσης,*** δηλαδή με την εκτέλεση αριθμητικών πειραμάτων στο μοντέλο για τις εισόδους (δεδομένα) που μας ενδιαφέρουν, για να δούμε πως αυτά επηρεάζουν τις εξόδους (μέτρα απόδοσης) του συστήματος. Η προσομοίωση είναι επίσης πολύ χρήσιμη στις περιπτώσεις που μπορούμε να πάρουμε μόνο προσεγγιστική αναλυτική

λύση, οπότε θέλουμε να επιβεβαιώσουμε την προσέγγιση, να βρούμε το σχετικό λάθος κ.λ.π.

5.2 ΤΑ ΕΙΔΗ ΜΟΝΤΕΛΩΝ ΠΡΟΣΟΜΟΙΩΣΗΣ

Έχοντας ένα μαθηματικό μοντέλο που πρέπει να μελετήσουμε με προσομοίωση (δηλαδή ένα *Μοντέλο Προσομοίωσης*), θα πρέπει να αναζητήσουμε κατάλληλα εργαλεία για το σκοπό αυτό. Στην προσπάθεια αυτή, είναι χρήσιμο να ταξινομήσουμε τα Μοντέλα Προσομοίωσης με βάση τέσσερις διαφορετικές έννοιες:

1. *Στατικά* **ή** *Δυναμικά Μοντέλα Προσομοίωσης:* Ένα *στατικό* μοντέλο προσομοίωσης, αναπαριστά ένα σύστημα σε μία συγκεκριμένη χρονική στιγμή, ή αναπαριστά ένα σύστημα στο οποίο ο χρόνος δεν έχει σημασία. Αντίθετα, ένα *δυναμικό* μοντέλο προσομοίωσης αναπαριστά ένα σύστημα, όπως αυτό εξελίσσεται με την πάροδο του χρόνου.
2. *Ντετερμινιστικά* **ή** *Στοχαστικά Μοντέλα Προσομοίωσης:* Αν ένα μοντέλο προσομοίωσης δεν περιλαμβάνει πιθανοτικά (δηλαδή "τυχαία") τμήματα, ονομάζεται *ντετερμινιστικό*. Για παράδειγμα, ένα πολύπλοκο σύστημα διαφορικών εξισώσεων που περιγράφει μία χημική αντίδραση, μπορεί να είναι ένα τέτοιο μοντέλο. Στα ντετερμινιστικά μοντέλα, η έξοδος είναι καθορισμένη, με δεδομένο το σύνολο των ποσοτήτων και σχέσεων εισόδου του μοντέλου. Όμως, πολλά συστήματα πρέπει να χρησιμοποιήσουν *στοχαστικά* μοντέλα προσομοίωσης, δηλαδή μοντέλα που θα έχουν τουλάχιστον ορισμένα τμήματα με "τυχαία" είσοδο. Τα περισσότερα υπολογιστικά συστήματα, που βασίζονται στα συστήματα αναμονής (queueing systems), χρησιμοποιούν στοχαστικά μοντέλα προσομοίωσης.
3. *Αυτο-οδηγούμενα* **ή** *Ιχνο-οδηγούμενα Μοντέλα Προσομοίωσης:* Σε ένα *αυτο-οδηγούμενο* (self-driven) μοντέλο, υπάρχει μία εσωτερική πηγή τυχαίων αριθμών. Οι τυχαίοι αριθμοί οδηγούν τα τμήματα του μοντέλου, δηλαδή χρησιμοποιούνται για τον προσδιορισμό των στιγμών εμφανίσεων των γεγονότων του συστήματος. Το βασικό χαρακτηριστικό του αυτο-οδηγούμενου μοντέλου είναι ότι αποτελεί ένα αυτάρκες μοντέλο το οποίο δεν χρειάζεται εξωτερικές εισόδους (inputs) για να λειτουργήσει. Αντίθετα, ένα *ιχνο-οδηγούμενο* (trace-driven) μοντέλο καθοδηγείται από ακολουθίες εισόδου που προέρχονται από δεδομένα (trace data) που έχουν δημιουργηθεί από τη λειτουργία ενός πραγματικού συστήματος. Τέτοια δεδομένα μπορούν να παραχθούν στα περισσότερα υπολογιστικά συστήματα που διαθέτουν ενσωματωμένα προγράμματα ιχνηλάτησης (tracing programs) που παρακολουθούν και καταγράφουν τις δραστηριότητες του συστήματος. Τα ιχνο-οδηγούμενα μοντέλα έχουν ορισμένα πλεονεκτήματα, όπως το γεγονός ότι αποφεύγονται οι δυσκολίες της πιθανοτικής ανάλυσης που χρειάζεται για τη χρήση κατανομών στην περιγραφή των εισόδων του μοντέλου και επίσης το γεγονός ότι τα μοντέλα αυτά είναι εύκολο να επιβεβαιωθούν. Το πρόβλημα με τα ιχνο-οδηγούμενα μοντέλα είναι το μικρό εύρος εφαρμογών που μπορούν να αντιμετωπίσουν. Οι εφαρμογές αυτές πρακτικά περιορίζονται σε υπολογιστικά συστήματα και μάλιστα μόνο για τη μελέτη μετατροπών σε ένα σύστημα που ήδη λειτουργεί.
4. *Συνεχή* **ή** *Διακριτά Μοντέλα Προσομοίωσης:* Οι ορισμοί των *συνεχών* και *διακριτών* μοντέλων προσομοίωσης, είναι ανάλογοι με τους ορισμούς των συνεχών και διακριτών συστημάτων που αναφέρθηκαν στην παράγραφο 1.4. Πάντως, πρέπει να σημειωθεί ότι ένα διακριτό μοντέλο δεν χρησιμοποιείται μόνο για την αναπαράσταση ενός διακριτού συστήματος και ένα διακριτό σύστημα δεν αναπαριστάται μόνο από ένα διακριτό μοντέλο προσομοίωσης. Η απόφαση για τη χρήση ενός διακριτού ή ενός

συνεχούς μοντέλου για ένα συγκεκριμένο σύστημα, εξαρτάται από τους ιδιαίτερους στόχους της μελέτης. Για παράδειγμα, ένα μοντέλο της ροής πακέτων δεδομένων σε ένα WAN, θα είναι διακριτό εάν μας ενδιαφέρουν τα χαρακτηριστικά και η κίνηση των επιμέρους πακέτων και κατά συνέπεια των επιμέρους χρηστών. Αντίθετα, αν μας ενδιαφέρει μόνο η συνολική κίνηση, η ροή των πακέτων θα μπορούσε ίσως να περιγραφεί με διαφορικές εξισώσεις σε ένα συνεχές μοντέλο.

Τα μοντέλα προσομοίωσης που θα μας απασχολήσουν στη συνέχεια, θα είναι διακριτά, δυναμικά, στοχαστικά και αυτο-οδηγούμενα και θα ονομάζονται *Μοντέλα Προσομοίωσης Διακριτών Γεγονότων (discrete event simulation models).* Μάλιστα, αφού τα ντετερμινιστικά μοντέλα μπορούν να θεωρηθούν ειδικές περιπτώσεις των στοχαστικών μοντέλων, δεν θα έχουμε απώλεια της γενικότητας στη μελέτη των μοντέλων προσομοίωσης.

5.3 Ο ΜΗΧΑΝΙΣΜΟΣ ΕΞΕΛΙΞΗΣ ΤΟΥ ΧΡΟΝΟΥ

Λόγω του δυναμικού χαρακτήρα των μοντέλων προσομοίωσης διακριτών γεγονότων, πρέπει να έχουμε τη δυνατότητα αποθήκευσης της τρέχουσας τιμής του προσομοιωμένου χρόνου, ενώ χρειαζόμαστε και ένα μηχανισμό αύξησής του από μία τιμή σε μία άλλη. Η μεταβλητή του μοντέλου προσομοίωσης που μας δίνει την τρέχουσα τιμή του χρόνου, ονομάζεται *ρολόι προσομοίωσης (simulation clock).* Η μονάδα χρόνου που χρησιμοποιεί το ρολόι είναι συνήθως η ίδια με αυτή που χρησιμοποιούν οι παράμετροι εισόδου, ενώ γενικά δεν υπάρχει σχέση του χρόνου που καταγράφει το ρολόι, με το χρόνο που απαιτείται για την εκτέλεση του προσομοιωτή στον υπολογιστή.

Ιστορικά έχουν επικρατήσει δύο βασικές μέθοδοι για την εξέλιξη του ρολογιού προσομοίωσης: Η *Εξέλιξη με βάση το Χρόνο του Επομένου Γεγονότος (next-event time advance)* και η *Εξέλιξη Σταθερής Αύξησης του Χρόνου (fixed-increment time advance).* Θα χρησιμοποιήσουμε την πρώτη μέθοδο διότι είναι πιο διαδεδομένη και διότι η δεύτερη μπορεί να θεωρηθεί ειδική περίπτωση της πρώτης.

Στη μέθοδο *εξέλιξης με βάση το χρόνο του επομένου γεγονότος,* το ρολόι προσομοίωσης αρχικοποιείται στο μηδέν και καθορίζονται οι στιγμές εμφάνισης των μελλοντικών γεγονότων. Το ρολόι τότε αυξάνει στο χρόνο εμφάνισης του πιο κοντινού στο μέλλον, από τα γεγονότα αυτά. Τη στιγμή αυτή η κατάσταση του συστήματος ενημερώνεται ώστε να πάρει υπ΄ όψη της το γεγονός που εμφανίστηκε, ενώ ενημερώνεται επίσης η γνώση μας για τις χρονικές στιγμές εμφάνισης των μελλοντικών γεγονότων. Στη συνέχεια, το ρολόι αυξάνει ώστε να δείχνει τη στιγμή εμφάνισης του νέου πιο κοντινού στο μέλλον γεγονότος, η κατάσταση του συστήματος ενημερώνεται, καθορίζονται οι χρονικές στιγμές εμφάνισης των μελλοντικών γεγονότων κ.ο.κ. Η διαδικασία αυτή εξέλιξης του ρολογιού προσομοίωσης από το ένα γεγονός στο άλλο, συνεχίζεται μέχρι να ικανοποιηθεί κάποια προκαθορισμένη συνθήκη τερματισμού της προσομοίωσης. Αφού όλες οι αλλαγές κατάστασης γίνονται μόνο στις χρονικές στιγμές εμφάνισης των γεγονότων, οι ενδιάμεσες ανενεργοί περίοδοι δεν λαμβάνονται υπ΄ όψη και το ρολόι μετακινείται αυτόματα στη στιγμή εμφάνισης του επομένου γεγονότος.

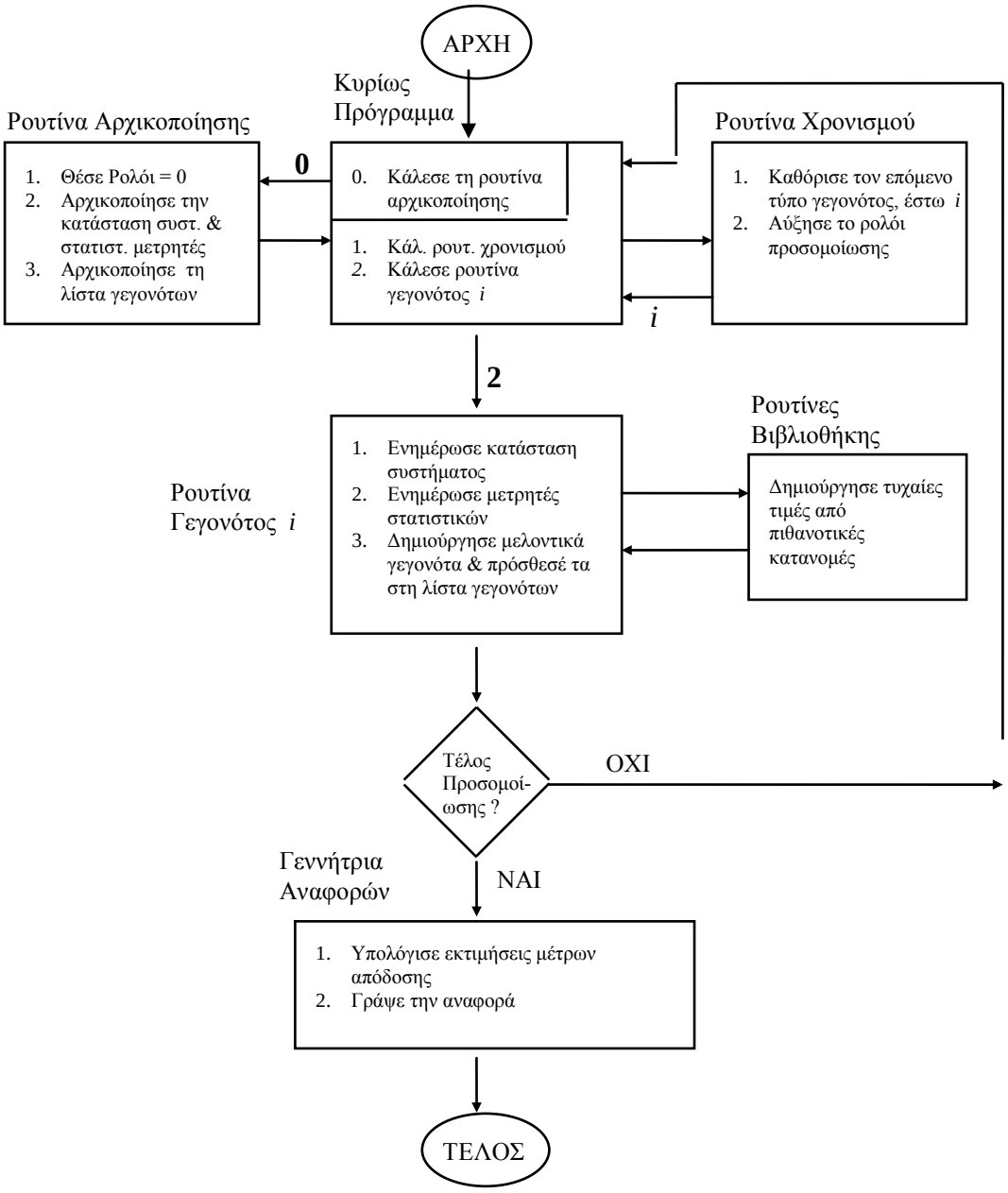
Όσον δε αφορά τη μέθοδο *εξέλιξης σταθερής αύξησης του χρόνου,* το ρολόι προσομοίωσης εξελίσσεται με σταθερές αυξήσεις ακριβώς Δt μονάδων χρόνου κάθε φορά. Μετά από κάθε ενημέρωση του ρολογιού, γίνεται ένας έλεγχος για να εξακριβωθεί εάν θα έπρεπε να έχουν εμφανισθεί κάποια γεγονότα κατά το προηγούμενο χρονικό διάστημα Δt. Αν εμφανίσθηκαν γεγονότα στο διάστημα αυτό, θεωρούμε ότι αυτά εμφανίζονται στο τέλος του χρονικού διαστήματος και η κατάσταση του συστήματος ενημερώνεται κατάλληλα.

5.4 ΣΥΣΤΑΤΙΚΑ ΚΑΙ ΟΡΓΑΝΩΣΗ ΕΝΟΣ ΜΟΝΤΕΛΟΥ ΠΡΟΣΟΜΟΙΩΣΗΣ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ

Τα περισσότερα μοντέλα προσομοίωσης διακριτών γεγονότων που χρησιμοποιούν τη μέθοδο εξέλιξης με βάση το χρόνο του επομένου γεγονότος, περιλαμβάνουν τα παρακάτω τμήματα:

- *Κατάσταση Συστήματος (system state):* Η συλλογή των μεταβλητών κατάστασης που είναι απαραίτητες για την περιγραφή του συστήματος σε μία χρονική στιγμή.
- *Ρολόι Προσομοίωσης (simulation clock):* Μία μεταβλητή που περιέχει την τρέχουσα τιμή του προσομοιωμένου χρόνου.
- *Λίστα Γεγονότων (event list):* Μία λίστα που περιέχει την επόμενη χρονική στιγμή εμφάνισης κάθε τύπου γεγονότος.
- *Μετρητές Στατιστικών (statistical counters):* Μεταβλητές που χρησιμοποιούνται για την αποθήκευση στατιστικών μετρήσεων της απόδοσης του συστήματος.
- *Ρουτίνα Αρχικοποίησης (initialization routine):* Ένα υποπρόγραμμα που αρχικοποιεί το μοντέλο προσομοίωσης τη χρονική στιγμή μηδέν.
- *Ρουτίνα Χρονισμού (timing routine):* Ένα υποπρόγραμμα που αναγνωρίζει το επόμενο γεγονός από τη λίστα γεγονότων και ακολούθως αυξάνει το ρολόι προσομοίωσης στη χρονική στιγμή που το γεγονός αυτό θα εμφανισθεί.
- *Ρουτίνες Γεγονότων (event routines):* Υποπρογράμματα που ενημερώνουν την κατάσταση συστήματος όταν εμφανίζεται ένα συγκεκριμένο είδος γεγονότος (υπάρχει μία τέτοια ρουτίνα για κάθε είδος γεγονότος).
- *Ρουτίνες Βιβλιοθήκης (library routines):* Σύνολο υποπρογραμμάτων που δημιουργούν τυχαίες εμφανίσεις τιμών από πιθανοτικές κατανομές, που έχουν ορισθεί ως μέρος του μοντέλου προσομοίωσης.
- *Γεννήτρια Αναφορών (report generator):* Υποπρόγραμμα που υπολογίζει εκτιμήσεις των επιθυμητών μέτρων απόδοσης από τους μετρητές στατιστικών και παράγει αναφορές όταν τελειώσει η εκτέλεση του προσομοιωτή.
- *Κυρίως Πρόγραμμα (main program):* Το πρόγραμμα που καλεί τη ρουτίνα χρονισμού για να καθοριστεί το επόμενο γεγονός και μετά μεταφέρει τον έλεγχο στην αντίστοιχη ρουτίνα γεγονότος για να ενημερωθεί κατάλληλα η κατάσταση του συστήματος. Ελέγχει επίσης αν πρέπει να τερματισθεί η προσομοίωση και καλεί τότε τη γεννήτρια αναφορών.

Οι λογικές σχέσεις ανάμεσα στα παραπάνω τμήματα φαίνονται στο Σχήμα 5.2.



Σχήμα 5.4. Διάγραμμα Ροής για την Εξέλιξη με βάση το Χρόνο Επομένου Γεγονότος.

Όπως αναφέρθηκε στα προηγούμενα, ένα σύστημα είναι μια καλά ορισμένη συλλογή από οντότητες. Οι οντότητες χαρακτηρίζονται από τιμές δεδομένων που καλούνται *ιδιότητες (attributes)* και οι οποίες είναι μέρος της κατάστασης του συστήματος. Οι οντότητες που έχουν ορισμένα κοινά χαρακτηριστικά, συχνά ομαδοποιούνται σε λίστες (ή αρχεία ή σύνολα). Για κάθε οντότητα υπάρχει μια εγγραφή στη λίστα που αποτελείται από τις ιδιότητες της οντότητας, ενώ η σειρά με την οποία τοποθετούνται οι εγγραφές στη λίστα, εξαρτάται από κάποιο καθορισμένο κανόνα.

Η οργάνωση και η λειτουργία ενός προγράμματος προσομοίωσης διακριτών γεγονότων που χρησιμοποιεί το μηχανισμό εξέλιξης με βάση το χρόνο του επομένου

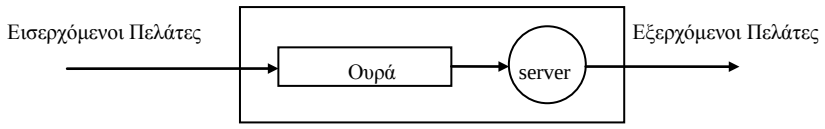
γεγονότος, όπως περιγράφηκε παραπάνω, είναι τυπική για την ανάπτυξη προσομοιωτών με γλώσσες προγραμματισμού γενικού σκοπού όπως η C, η Pascal και η FORTRAN. Ονομάζεται *Προσέγγιση Χρονοδρομολόγησης Γεγονότων (event-scheduling approach)* στη μοντελοποίηση της προσομοίωσης, διότι οι χρονικές στιγμές εμφάνισης των μελλοντικών γεγονότων κωδικοποιούνται στο μοντέλο και προγραμματίζονται να εμφανισθούν στο προσομοιωμένο μέλλον. Η εναλλακτική προσέγγιση στο θέμα, είναι η *Προσέγγιση Διαδικασίας (process approach),* η οποία "βλέπει" την προσομοίωση σε σχέση με τις ξεχωριστές οντότητες που εμπλέκονται. Ο κώδικας που γράφεται εδώ, περιγράφει την "εμπειρία" μιας "τυπικής" οντότητας καθώς αυτή μετακινείται δια μέσου του συστήματος. Η κωδικοποίηση ενός προγράμματος προσομοίωσης με την προσέγγιση διαδικασίας, γίνεται συνήθως με χρήση ειδικών γλωσσών προγραμματισμού για προσομοίωση, όπως η SIMULA, η GPSS, η SIMSCRIPT, η SIMAN, η SLAM κ.α.

ΚΕΦΑΛΑΙΟ 6
ΠΡΟΣΟΜΟΙΩΣΗ ΕΝΟΣ ΣΥΣΤΗΜΑΤΟΣ ΑΝΑΜΟΝΗΣ

6.1 ΔΙΑΤΥΠΩΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Τα συστήματα αναμονής (queueing systems), βρίσκονται πίσω από τα περισσότερα μοντέλα μελέτης της απόδοσης υπολογιστικών συστημάτων, αφού φαινόμενα καθυστερήσεων λόγω της απαίτησης χρήσης περιορισμένων πόρων από πολλούς "πελάτες", μπορούν να εντοπισθούν τόσο σε απλούς υπολογιστές (π.χ. CPU, I/O, μνήμη), όσο και σε πολύπλοκα συστήματα όπως τα δίκτυα υπολογιστών (π.χ. κανάλια μετάδοσης δεδομένων).

Έστω το σύστημα αναμονής με έναν εξυπηρετητή που φαίνεται στο Σχήμα 6.1.



Σχήμα 6.1. Σύστημα αναμονής με έναν εξυπηρετητή

Οι *χρόνοι μεταξύ διαδοχικών αφίξεων* $A_1, A_2, \ldots$ είναι ανεξάρτητες, όμοια κατανεμημένες τυχαίες μεταβλητές. Ένας πελάτης που φθάνει και βρίσκει τον εξυπηρετητή (server) άδειο, αρχίζει αμέσως την εξυπηρέτησή του, ενώ εάν τον βρει κατειλλημένο μένει στο τέλος της ουράς αναμονής. Όταν ο server ελευθερωθεί, παίρνει τον πρώτο πελάτη της ουράς (αν υπάρχει κάποιος), δηλαδή έχουμε FIFO πολιτική εξυπηρέτησης. Οι *χρόνοι εξυπηρέτησης των πελατών* $S_1, S_2, \ldots$ είναι επίσης ανεξάρτητες, όμοια κατανεμημένες τυχαίες μεταβλητές.

Η προσομοίωση θα αρχίσει από τη "μηδενική" κατάσταση του συστήματος, δηλαδή με άδειο σύστημα. Τη χρονική στιγμή 0, θα αρχίσουμε να περιμένουμε την άφιξη του πρώτου πελάτη, η οποία θα γίνει μετά από χρόνο A_1. Θέλουμε να προσομοιώσουμε το σύστημα μέχρις ότου ένας συγκεκριμένος αριθμός πελατών n περάσει από την ουρά, δηλαδή η προσομοίωση θα σταματήσει όταν ο n-στός πελάτης θα εισέλθει στον εξυπηρετητή. Όπως είναι φανερό, ο χρόνος ολοκλήρωσης της προσομοίωσης είναι επίσης μια τυχαία μεταβλητή που εξαρτάται από τις τιμές των τυχαίων μεταβλητών που περιγράφουν τους χρόνους μεταξύ διαδοχικών αφίξεων και εξυπηρέτησης των πελατών.

Προκειμένου να μετρήσουμε την απόδοση του συστήματος, θα ενδιαφερθούμε για εκτιμήσεις τριών ποσοτήτων. *Πρώτα,* θα εκτιμήσουμε την "αναμενόμενη" *μέση καθυστέρηση* στην ουρά $d(n)$, κάθε ενός από τους n πελάτες. Το $d(n)$ θα πρέπει κανονικά να βρίσκεται ως η μέση τιμή πολλών (πρακτικά άπειρων) μέσων καθυστερήσεων n πελατών. Από *μία* εκτέλεση του προσομοιωτή, στην οποία παρατηρούνται καθυστερήσεις στην ουρά $D_1, D_2, \ldots, D_n$, των n πελατών, μία προφανής εκτίμηση του $d(n)$ είναι:

$$\hat{d}(n) = \frac{\sum_{i=1}^{n} D_i}{n}$$

(Στη συνέχεια, όλες οι εκτιμήσεις μεγεθών θα τονίζονται με το σύμβολο ^).

Η έννοια της καθυστέρησης, περιλαμβάνει φυσικά και την περίπτωση ένας πελάτης να μη χρειασθεί να περιμένει. Π.χ. ο πρώτος πελάτης θα έχει οπωσδήποτε $D_1 = 0$. Πρέπει να παρατηρήσουμε εδώ, ότι το $\hat{d}(n)$ δεν είναι ο μαθηματικός μέσος όρος μιας τυχαίας μεταβλητής, αφού δεν έχουμε τυχαίες παρατηρήσεις της ίδιας τυχαίας μεταβλητής. Επίσης, εδώ το $\hat{d}(n)$ είναι μια εκτίμηση που βασίζεται σε ένα "δείγμα" μεγέθους 1, αφού βασίζεται μόνο σε μία εκτέλεση του προσομοιωτή. Φυσικά, δεν είναι αρκετή μία εκτέλεση για να πάρουμε ασφαλείς εκτιμήσεις.

Το μέγεθος $d(n)$ είναι ένα μέτρο ενδιαφέροντος χρήστη. Ένα *δεύτερο* μέγεθος που μας ενδιαφέρει, είναι ο "αναμενόμενος" *μέσος αριθμός πελατών στην ουρά* $q(n)$. Το μέγεθος αυτό είναι ένα μέτρο ενδιαφέροντος συστήματος και διαφέρει από το $d(n)$, καθότι η μέση τιμή υπολογίζεται πάνω στο (συνεχή) χρόνο, σε αντίθεση με το $d(n)$ που υπολογίζεται πάνω στους (διακριτούς) πελάτες. Συγκεκριμένα, έστω $Q(t)$ ο αριθμός των πελατών στην ουρά τη χρονική στιγμή t, για κάθε πραγματικό αριθμό $t \geq 0$ και έστω $T(n)$ ο χρόνος που απαιτείται για να παρατηρήσουμε τις n καθυστερήσεις στην ουρά. Τότε, για κάθε χρόνο t μεταξύ 0 και $T(n)$, το $Q(t)$ είναι ένας μη-αρνητικός ακέραιος. Επίσης, αν p_i είναι το ποσοστό (με τιμές μεταξύ 0 και 1) του χρόνου που το $Q(t)$ είναι ίσο με i, τότε το $q(n)$ ορίζεται ως:

$$q(n) = \sum_{i=0}^{\infty} i p_i$$

ενώ η εκτίμηση του $q(n)$ από μία εκτέλεση του προσομοιωτή, δίνεται από τις εκτιμήσεις των p_i, δηλαδή:

$$\hat{q}(n) = \sum_{i=0}^{\infty} i \hat{p}_i \qquad (6.1)$$

όπου $\hat{p}_i$ είναι τα *παρατηρούμενα* ποσοστά του χρόνου, κατά τη διάρκεια της προσομοίωσης, που υπάρχουν i πελάτες στην ουρά. Αν T_i είναι ο συνολικός χρόνος προσομοίωσης κατά τον οποίο η ουρά έχει μήκος i, τότε $\hat{p}_i = T_i / T(n)$, όπου $T(n) = T_0 + T_1 + T_2 + \cdots$ και η (1) μπορεί να γραφεί ως:

$$\hat{q}(n) = \frac{\sum_{i=0}^{\infty} i T_i}{T(n)} \qquad (6.2)$$

Το άθροισμα στον αριθμητή της (2) είναι η επιφάνεια κάτω από την "καμπύλη" του $Q(t)$, μεταξύ της αρχής και του τέλους της προσομοίωσης, δηλαδή είναι το ολοκλήρωμα:

$$\sum_{i=0}^{\infty} i T_i = \int_0^{T(n)} Q(t)\,dt$$

οπότε η εκτίμηση του $q(n)$ δίνεται από τη σχέση:

$$\hat{q}(n) = \frac{\int_0^{T(n)} Q(t)\,dt}{T(n)} \qquad (6.3)$$

Αν και οι σχέσεις (3) και (2) είναι ισοδύναμες, η (3) είναι προτιμότερη διότι το ολοκλήρωμα μπορεί να υπολογισθεί ως το άθροισμα ορθογωνίων παραλληλογράμμων, καθώς η προσομοίωση θα εξελίσσεται.

Το *τρίτο* και τελευταίο μέτρο απόδοσης που θα εξετάσουμε, είναι η *χρησιμοποίηση* (utilization), ένα ακόμη μέτρο ενδιαφέροντος συστήματος. Η "αναμενόμενη" χρησιμοποίηση του εξυπηρετητή $u(n)$, είναι το μέσο ποσοστό (με τιμές μεταξύ 0 και 1) του χρόνου προσομοίωσης [από τη στιγμή 0 έως τη στιγμή $T(n)$], που ο εξυπηρετητής είναι απασχολημένος. Η εκτίμηση της χρησιμοποίησης $\hat{u}(n)$, μπορεί να υπολογισθεί απ΄ ευθείας από την προσομοίωση, παρατηρώντας τις στιγμές κατά τις οποίες ο εξυπηρετητής αλλάζει κατάσταση (από άεργος σε απασχολημένος και αντίστροφα) και εκτελώντας τις κατάλληλες αφαιρέσεις και διαιρέσεις. Ορίζουμε τη "συνάρτηση απασχόλησης":

$$B(t) = \begin{cases} 1 & \text{αν ο εξυπηρετητης ειναι απασχολημενος τη στιγμη } t \\ 0 & \text{αν ο εξυπηρετητης ειναι αεργος τη στιγμη } t \end{cases}$$

Το $\hat{u}(n)$ μπορεί να εκφραστεί ως το ποσοστό του χρόνου που το $B(t)$ είναι ίσο με 1. Αφού στο διάγραμμα του $B(t)$, το ύψος της γραφικής παράστασής του είναι πάντοτε είτε 0 ή 1, το $\hat{u}(n)$ θα μπορεί να υπολογισθεί (κατά αναλογία προς τον τρόπο υπολογισμού του $\hat{q}(n)$ παραπάνω), από τη σχέση:

$$\hat{u}(n) = \frac{\int_0^{T(n)} B(t)dt}{T(n)} \qquad (6.4)$$

Στο σύστημα αυτό, τα γεγονότα είναι η άφιξη και η αναχώρηση ενός πελάτη. Οι μεταβλητές συστήματος που μας χρειάζονται για τις εκτιμήσεις των $d(n)$, $q(n)$ και $u(n)$, είναι: Η κατάσταση του εξυπηρετητή (0 αν είναι άεργος και 1 αν είναι απασχολημένος), ο αριθμός των πελατών στην ουρά, η χρονική στιγμή άφιξης κάθε πελάτη που βρίσκεται στην ουρά (μία λίστα) και η χρονική στιγμή εμφάνισης του πιο πρόσφατου γεγονότος. Η στιγμή εμφάνισης του πιο πρόσφατου γεγονότος, η οποία ορίζεται ως e_{i-1} εάν $e_{i-1} \le t < e_i$ (όπου t είναι ο παρών χρόνος της προσομοίωσης), μας χρειάζεται για τον υπολογισμό του πλάτους των ορθογωνίων παραλληλογράμμων, που χρησιμοποιούνται στις εκτιμήσεις των $q(n)$ και $u(n)$.

6.2 Η ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΣΟΜΟΙΩΤΗ

Προκειμένου να δούμε πώς το μοντέλο προσομοίωσης αναπαριστάται υπολογιστικά τη χρονική στιγμή $e_0 = 0$ και τις στιγμές $e_1, e_2, \ldots, e_{13}$ κατά τις οποίες εμφανίζονται τα 13 διαδοχικά γεγονότα που απαιτούνται για να παρατηρήσουμε τον επιθυμητό αριθμό $n = 6$ διελεύσεων πελατών από την ουρά, υποθέτουμε ότι οι χρόνοι μεταξύ διαδοχικών αφίξεων και εξυπηρέτησης των πελατών είναι:

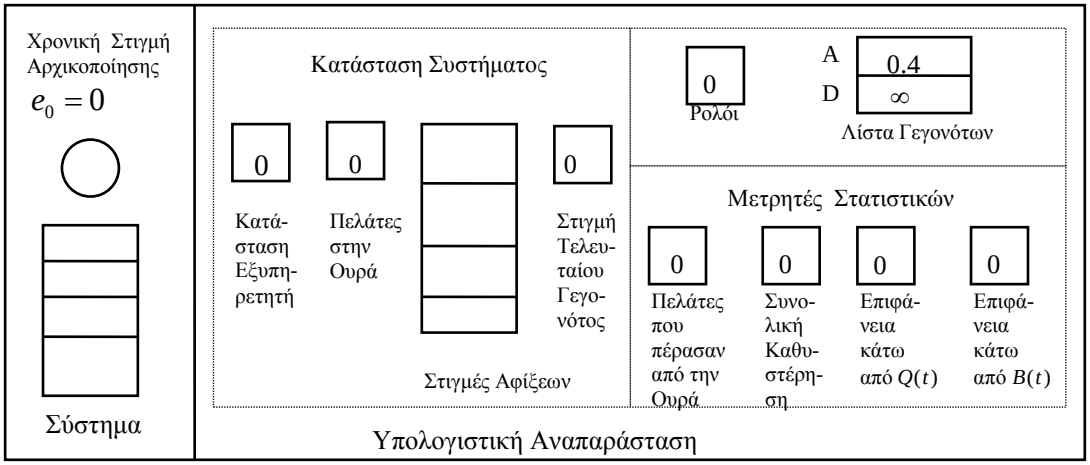
$A_1 = 0.4, A_2 = 1.2, A_3 = 0.5, A_4 = 1.7, A_5 = 0.2, A_6 = 1.6, A_7 = 0.2, A_8 = 1.4, A_9 = 1.9, \ldots$

$S_1 = 2.0, S_2 = 0.7, S_3 = 0.2, S_4 = 1.1, S_5 = 3.7, S_6 = 0.6, \ldots$

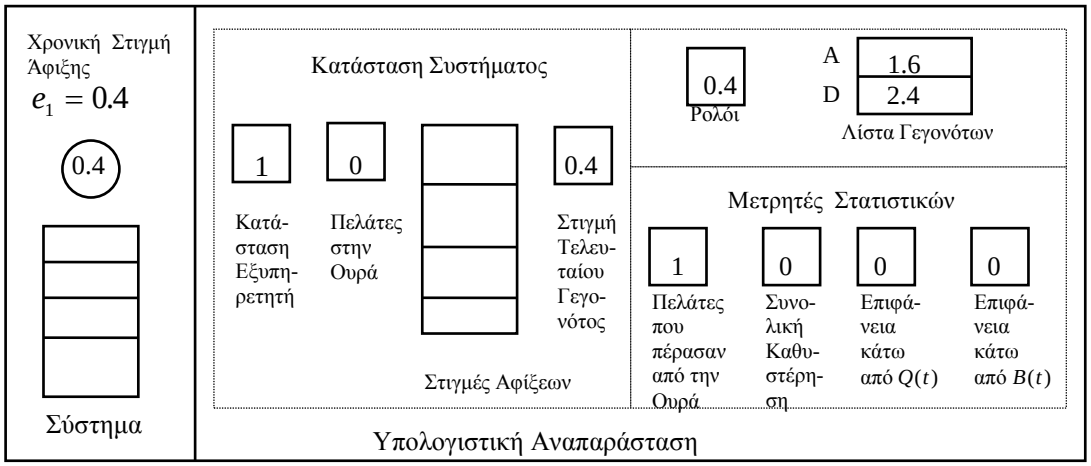
Δηλαδή, μεταξύ της στιγμής 0 και της στιγμής της πρώτης άφιξης μεσολαβεί 0.4 της μονάδας χρόνου, μεταξύ των αφίξεων του 1ου και του 2ου πελάτη μεσολαβούν 1.2 μονάδες χρόνου κ.ο.κ., ενώ ο 1ος πελάτης θα χρειασθεί 2.0 μονάδες χρόνου εξυπηρέτηση

κ.λ.π. Δεν είναι απαραίτητο να δηλωθεί η μονάδα χρόνου, αλλά απλώς να είμαστε βέβαιοι ότι όλες οι χρονικές ποσότητες εκφράζονται με την ίδια μονάδα. Σε μία πραγματική προσομοίωση (βλέπε επόμενες παραγράφους), τα A_i και S_i παράγονται από τις πιθανοτικές κατανομές που περιγράφουν τα δύο φαινόμενα αντίστοιχα.

Στο Σχήμα 5 φαίνονται το σύστημα και η υπολογιστική αναπαράστασή του τις χρονικές στιγμές $e_0 = 0, e_1 = 0.4, \ldots, e_{13} = 8.6$. Στο τμήμα που αναπαριστά το σύστημα φαίνεται ο εξυπηρετητής και η ουρά, ενώ οι αριθμοί είναι οι στιγμές άφιξης των πελατών (όταν υπάρχουν). Στο τμήμα που αντιπροσωπεύει την υπολογιστική αναπαράσταση, οι τιμές των μεταβλητών που φαίνονται, είναι μετά την ολοκλήρωση όλων των υπολογισμών που συνεπάγεται το γεγονός που έχει μόλις εμφανισθεί.

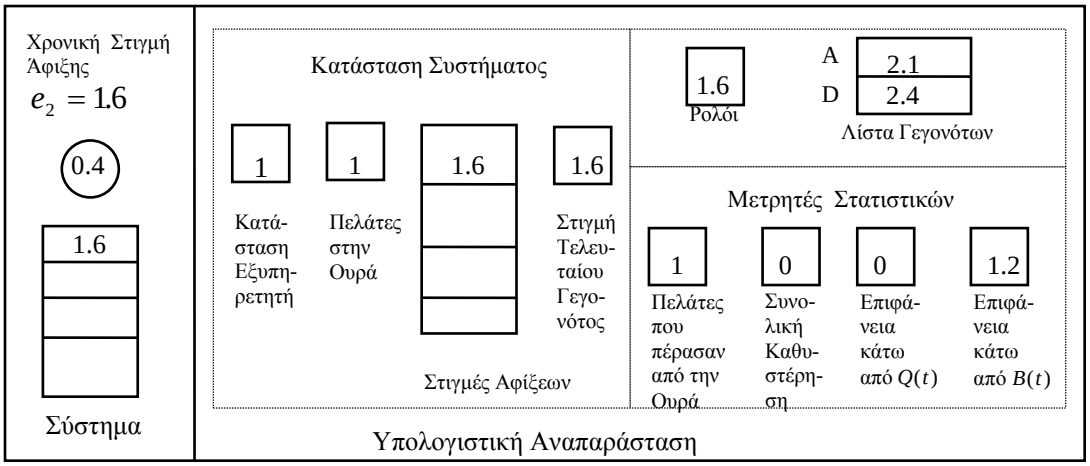


(a)

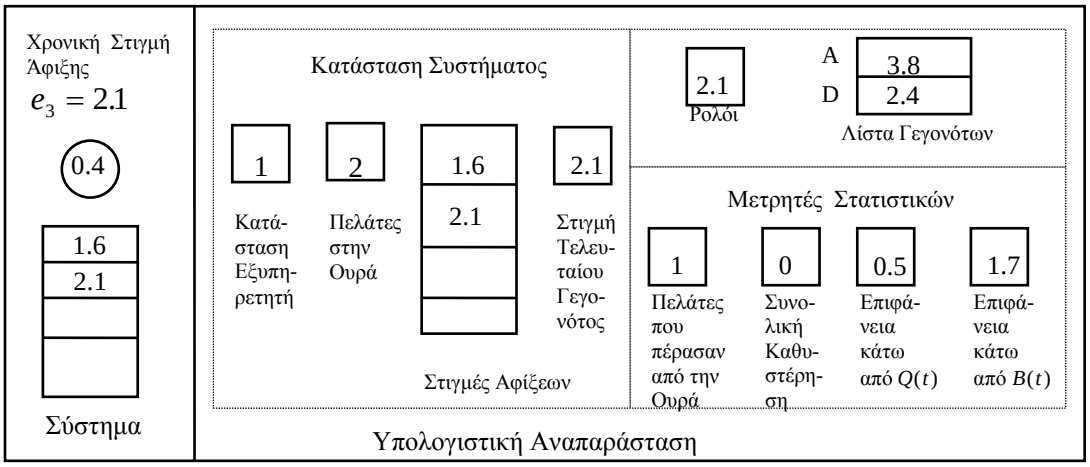


(b)

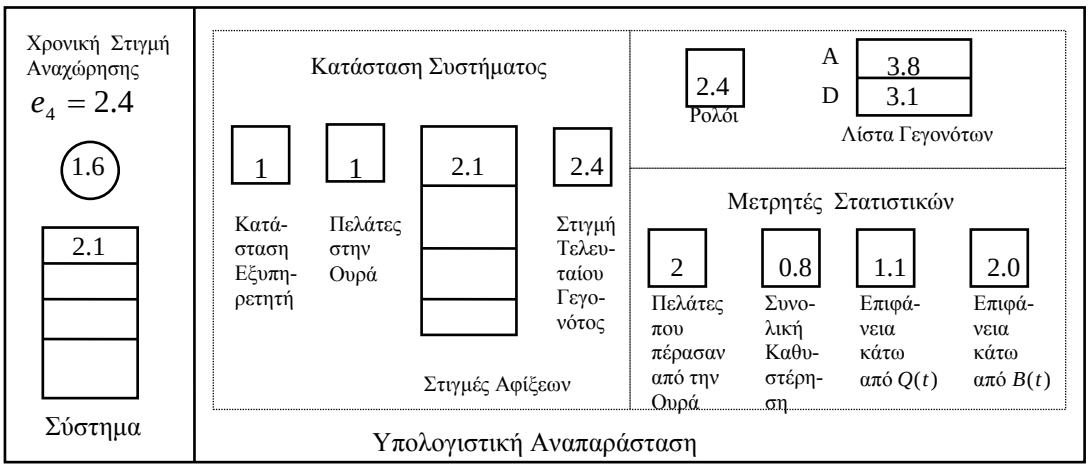
Σχήμα 6.1. Το Σύστημα και η Υπολογιστική Αναπαράστασή του κατά τις χρονικές στιγμές εμφάνισης των γεγονότων.



(c)

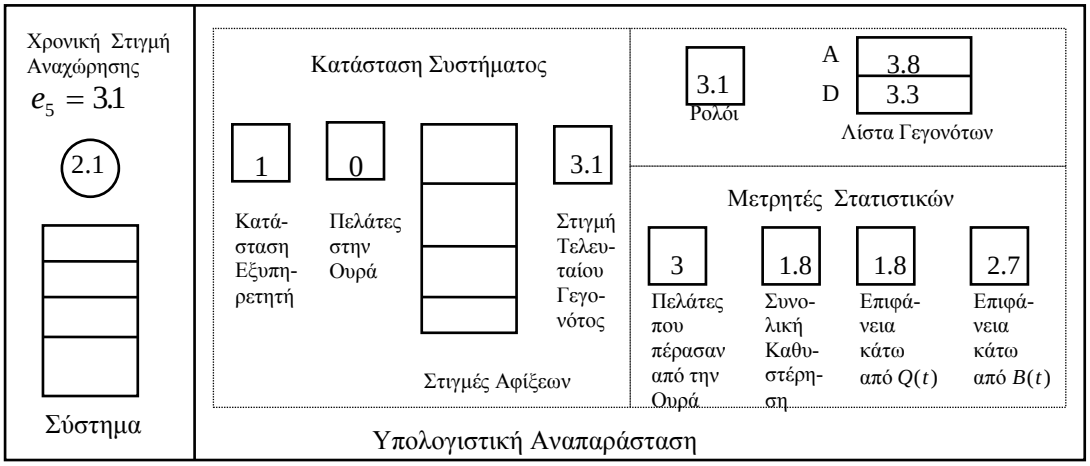


(d)

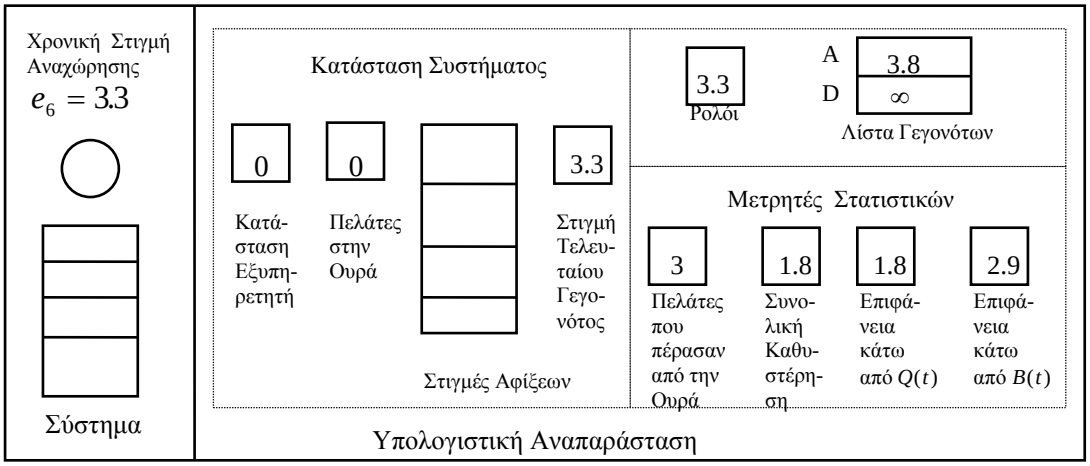


(e)

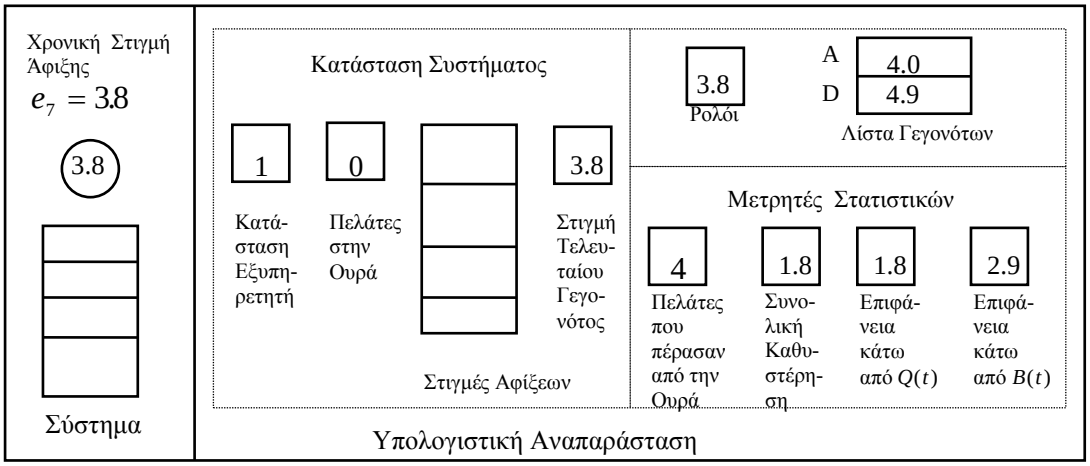
Σχήμα 6.2. (Συνέχεια)



(f)

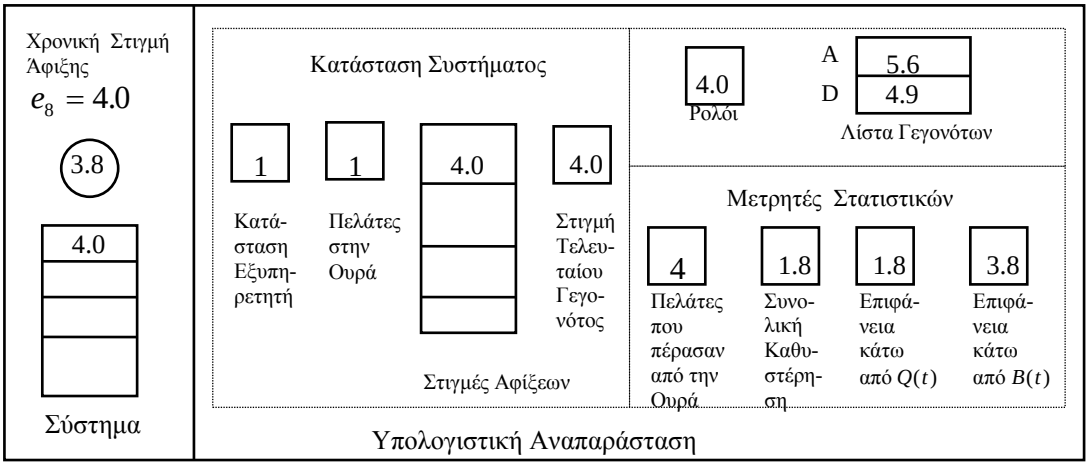


(g)

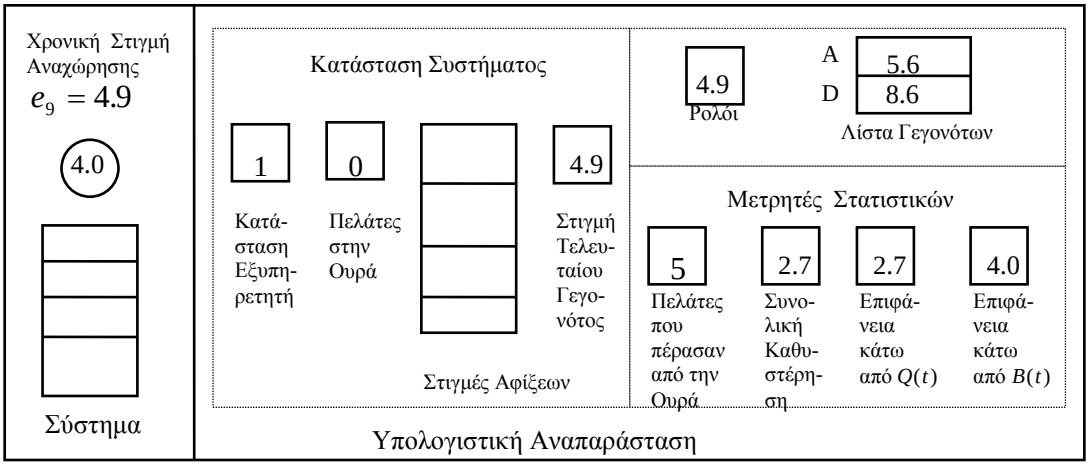


(h)

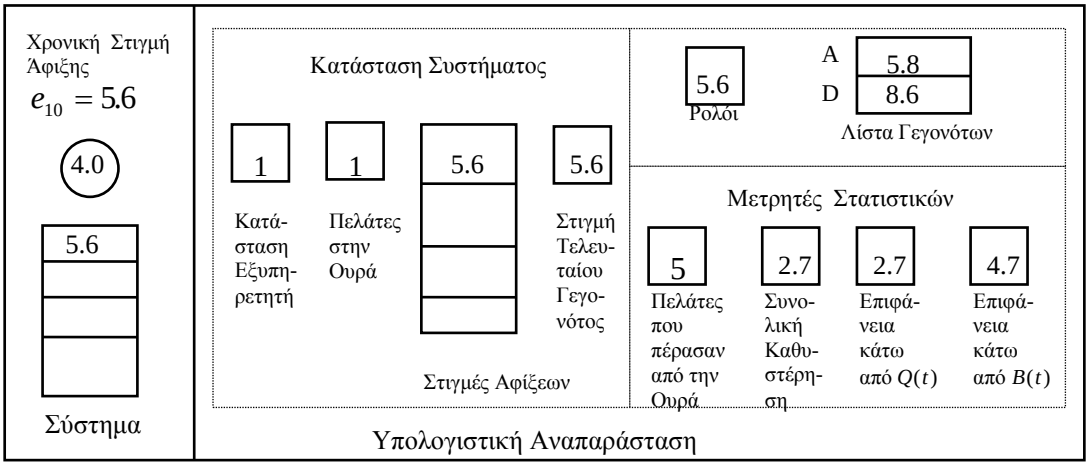
Σχήμα 6.3. (Συνέχεια)



(i)

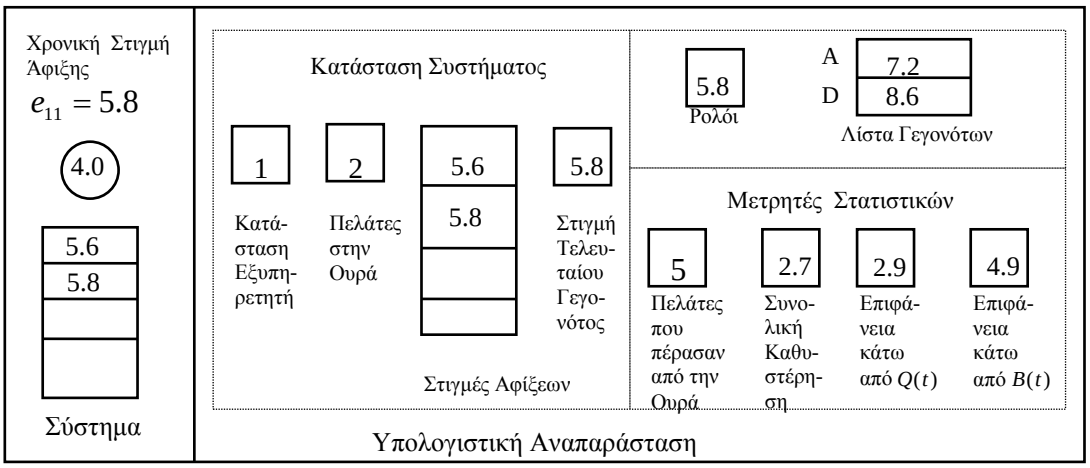


(j)

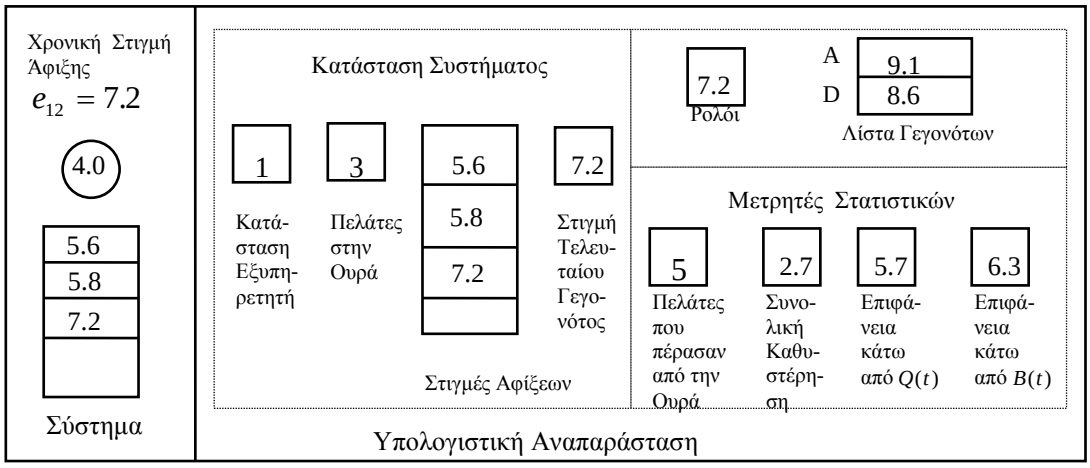


(k)

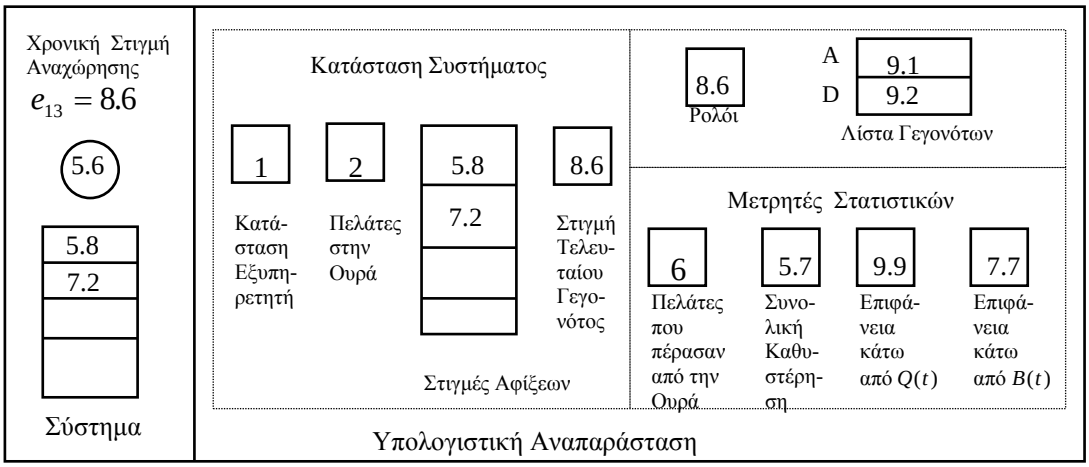
Σχήμα 6.4. (Συνέχεια)



(l)



(m)



(n)

Σχήμα 6.5. (Συνέχεια)

Μερικά σχόλια που αφορούν τη λογική της προσομοίωσης που παρουσιάζεται στο Σχήμα 5, είναι τα εξής:

- Το βασικό στοιχείο της δυναμικής εξέλιξης του προσομοιωτή είναι η αλληλεπίδραση του ρολογιού με τη λίστα γεγονότων. Η λίστα γεγονότων συντηρείται και το ρολόι μετακινείται στο επόμενο γεγονός, όπως αυτό καθορίζεται από τη λίστα γεγονότων.
- Κατά τη διάρκεια της επεξεργασίας ενός γεγονότος, δεν εξελίσσεται ο "προσομοιωμένος" χρόνος. Όμως πρέπει να ενημερώνονται οι μεταβλητές κατάστασης και οι μετρητές στατιστικών. Για παράδειγμα, θα ήταν λάθος να ενημερωθεί ο αριθμός πελατών στην ουρά πριν ενημερωθεί ο μετρητής της επιφάνειας κάτω από το $Q(t)$ αφού το ύψος του ορθογωνίου που πρέπει να χρησιμοποιηθεί, είναι αυτό της προηγούμενης τιμής του $Q(t)$. Ένα άλλο λάθος θα ήταν η αλλαγή της λίστας της ουράς σε μία αναχώρηση, πριν υπολογισθεί η καθυστέρηση του πρώτου πελάτη στην ουρά, διότι τότε θα χανόταν η χρονική στιγμή άφιξής του στο σύστημα.
- Μερικές φορές είναι εύκολο να παραβλεφθούν οι συνέπειες γεγονότων που δεν είναι συνηθισμένα κατά τη διάρκεια της προσομοίωσης, που έχουν όμως σημαντικές επιπτώσεις. Π.χ. είναι πιθανό να ξεχάσουμε ότι ένας πελάτης που αναχωρεί, μπορεί να αφήσει πίσω του ένα άδειο σύστημα και κατά συνέπεια πρέπει να μείνει άεργος ο εξυπηρετητής και το γεγονός της αναχώρησης πρέπει να διαγραφεί από τη λίστα γεγονότων.
- Μπορεί να συμβεί δύο ή περισσότερες τιμές της λίστας γεγονότων να είναι ίδιες, οπότε πρέπει να αποφασισθεί ποιό γεγονός θα ακολουθήσει. Οι κανόνες που θα χρησιμοποιούνται στις περιπτώσεις αυτές μπορεί να επηρεάζουν τα αποτελέσματα της προσομοίωσης και θα πρέπει να επιλέγονται με βάση την επιθυμητή μοντελοποίηση του συστήματος. Πάντως, όταν τα γεγονότα περιγράφονται από συνεχείς πιθανοτικές κατανομές, η πιθανότητα εμφάνισης ενός τέτοιου γεγονότος είναι 0. Τέτοια περίπτωση είναι οι χρόνοι μεταξύ διαδοχικών αφίξεων και οι χρόνοι εξυπηρέτησης του παραπάνω παραδείγματος.

Το παράδειγμα που εξετάζουμε έχει σκοπό να δείξει τις αλλαγές και τις δομές δεδομένων που χρησιμοποιούνται σε μία προσομοίωση διακριτών γεγονότων με μηχανισμό εξέλιξης με βάση το χρόνο του επομένου γεγονότος και περιλαμβάνει τις περισσότερες από τις βασικές ιδέες που απαιτούνται για πιο πολύπλοκους προσομοιωτές αυτού του τύπου. Οι χρόνοι μεταξύ διαδοχικών αφίξεων και οι χρόνοι εξυπηρέτησης θα μπορούσαν να είχαν επιλεγεί από κάποιο πίνακα τυχαίων τιμών ο οποίος θα είχε δημιουργηθεί με βάση τις χρησιμοποιούμενες πιθανοτικές κατανομές που περιγράφουν τα δύο μεγέθη.

6.3 Η ΟΡΓΑΝΩΣΗ ΚΑΙ Η ΛΟΓΙΚΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Το σύστημα αναμονής με έναν εξυπηρετητή που θα προσομοιώσουμε στις επόμενες παραγράφους, διαφέρει σε δύο σημεία με το μοντέλο που χρησιμοποιήσαμε παραπάνω:

1. Η προσομοίωση θα σταματήσει όταν θα έχουν περάσει $n = 1000$ πελάτες από την ουρά, αντί για $n = 6$, έτσι ώστε να συλλεγούν περισσότερα δεδομένα. Πρέπει να σημειώσουμε ότι η αλλαγή αυτή στον κανόνα τερματισμού της προσομοίωσης, αλλάζει το ίδιο το μοντέλο, δεδομένου ότι οι μετρήσεις εξόδου ορίζονται σε σχέση με τον κανόνα αυτό, όπως φαίνεται από την παρουσία του *"n"* στις εκφράσεις των ποσοτήτων $d(n)$, $q(n)$ και $u(n)$ που υπολογίζονται.

2. Οι χρόνοι μεταξύ διαδοχικών αφίξεων και οι χρόνοι εξυπηρέτησης, θα μοντελοποιηθούν ως ανεξάρτητες τυχαίες μεταβλητές που περιγράφονται από εκθετικές κατανομές, με μέση τιμή 1 λεπτό για τους χρόνους μεταξύ αφίξεων και 0.5 για τους χρόνους εξυπηρέτησης. Η εκθετική κατανομή με μέση τιμή β (οποιοσδήποτε θετικός πραγματικός αριθμός) είναι συνεχής, με συνάρτηση πυκνότητας πιθανότητας

$$f(x) = \frac{1}{\beta} e^{-x/\beta} \qquad \text{για} \quad x \geq 0$$

Η αλλαγή αυτή γίνεται διότι είναι πιο συνηθισμένο οι ποσότητες εισόδου που "οδηγούν" την προσομοίωση, να δημιουργούνται από συγκεκριμένες κατανομές, παρά να θεωρούμε ότι είναι γνωστές, όπως έγινε στην προηγούμενη παράγραφο. Η επιλογή της εκθετικής κατανομής με τις παραπάνω τιμές για το β είναι ουσιαστικά αυθαίρετη και έγινε γιατί είναι εύκολο να δημιουργηθούν εκθετικές τιμές από το πρόγραμμα της προσομοίωσης. Αργότερα θα δούμε πως επιλέγουμε κατανομές και παραμέτρους για τη μοντελοποίηση των τυχαίων μεταβλητών εισόδου της προσομοίωσης. Το σύστημα αυτό αναμονής με έναν εξυπηρετητή και εκθετικούς χρόνους μεταξύ αφίξεων και εξυπηρέτησης, είναι το γνωστό σύστημα αναμονής $M/M/1$.

Για να προσομοιώσουμε αυτό το μοντέλο, πρέπει με κάποιο τρόπο να δημιουργήσουμε τυχαίες τιμές από εκθετικές κατανομές. Αρχικά καλείται μία *Γεννήτρια Τυχαίων Αριθμών* (βλέπε τα επόμενα Κεφάλαια), η οποία δημιουργεί μία τυχαία τιμή U που είναι ομοιόμορφα (συνεχώς) κατανεμημένη μεταξύ 0 και 1. Η κατανομή αυτή θα αναφέρεται ως $U(0,1)$ και έχει συνάρτηση πυκνότητας πιθανότητας

$$f(x) = \begin{cases} 1 & \text{αν} \quad 0 \leq x \leq 1 \\ 0 & \text{αλλιως} \end{cases}$$

Είναι εύκολο να δείξουμε ότι: Η πιθανότητα μία $U(0,1)$ τυχαία μεταβλητή να "πέσει" σε οποιοδήποτε υποδιάστημα $[x,\ x+\Delta x]$ που περιλαμβάνεται στο διάστημα $[0,1]$, είναι (ομοιόμορφα) Δx. Η κατανομή $U(0,1)$ είναι βασική στην προσομοίωση, διότι, όπως θα δούμε αργότερα, μία τυχαία τιμή από οποιαδήποτε κατανομή, μπορεί να δημιουργηθεί, αν πάρουμε μία ή περισσότερες τυχαίες τιμές από την $U(0,1)$ και εφαρμόσουμε κάποιο είδος μετασχηματισμού. Εδώ, που έχουμε την εκθετική κατανομή, αφού πάρουμε το U, θα βρούμε το φυσικό λογάριθμό του, θα τον πολλαπλασιάσουμε με β και τελικά θα αλλάξουμε πρόσημο, ώστε να έχουμε (όπως θα δείξουμε) μία εκθετική τυχαία τιμή με μέση τιμή β, δηλαδή την τυχαία τιμή $-\beta \ln U$.

Για να δείξουμε ότι ο αλγόριθμος αυτός δουλεύει, ας θυμηθούμε ότι η *Συνάρτηση Κατανομής Πιθανότητας* μιας τυχαίας μεταβλητής *X*, ορίζεται για κάθε πραγματικό *x*, ως $F(x) = P(X \leq x)$. Αν η X είναι εκθετική με μέση τιμή β, τότε

$$F(x) = \int_0^x \frac{1}{\beta} e^{-t/\beta} dt = 1 - e^{-x/\beta} \qquad \text{για κάθε} \quad x \geq 0$$

αφού η συνάρτηση πυκνότητας πιθανότητας της εκθετικής κατανομής με όρισμα $t \geq 0$, είναι $(1/\beta)e^{-t/\beta}$. Για να δείξουμε ότι η μέθοδος αυτή είναι σωστή, επιβεβαιώνουμε ότι η τιμή που δίνει είναι μικρότερη ή ίση με *x* (οποιοσδήποτε πραγματικός μη-αρνητικός αριθμός), με πιθανότητα $F(x)$ όπως παραπάνω:

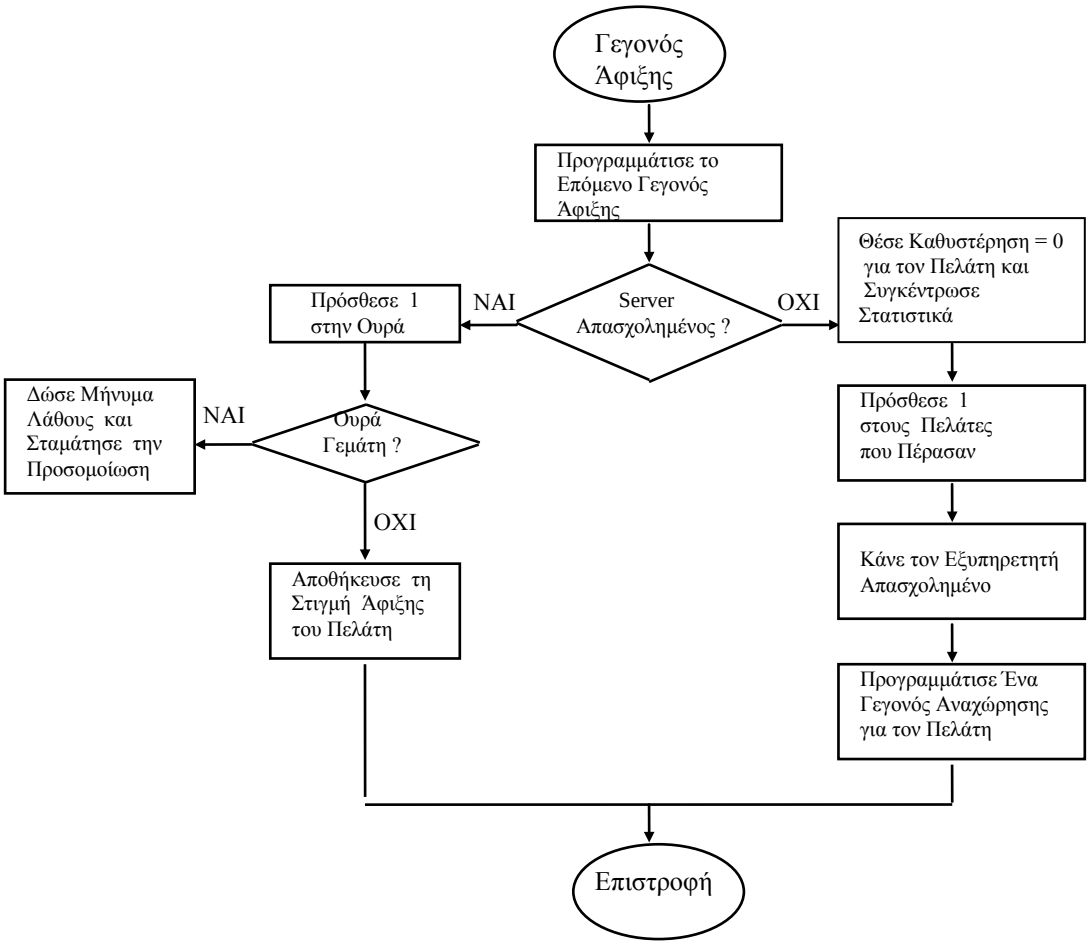
$$P(-\beta \ln U \leq x) = P(\ln U \geq -\frac{x}{\beta}) = P(U \geq e^{-x/\beta}) = P(e^{-x/\beta} \leq U \leq 1) = 1 - e^{-x/\beta}$$

Ο τελευταίος όρος είναι το $F(x)$ της εκθετικής κατανομής, οπότε έχει επιβεβαιωθεί ο αλγόριθμος. Σε επόμενο Κεφάλαιο θα ασχοληθούμε εκτενέστερα με το θέμα της

δημιουργίας τυχαίων τιμών από οποιαδήποτε κατανομή. Αν και οι περισσότεροι μεταφραστές γλωσσών προγραμματισμού έχουν ενσωματωμένες γεννήτριες τυχαίων αριθμών, αυτές είναι συνήθως κακής ποιότητας και έτσι πρέπει ο προγραμματιστής να αναπτύξει τη δική του γεννήτρια.

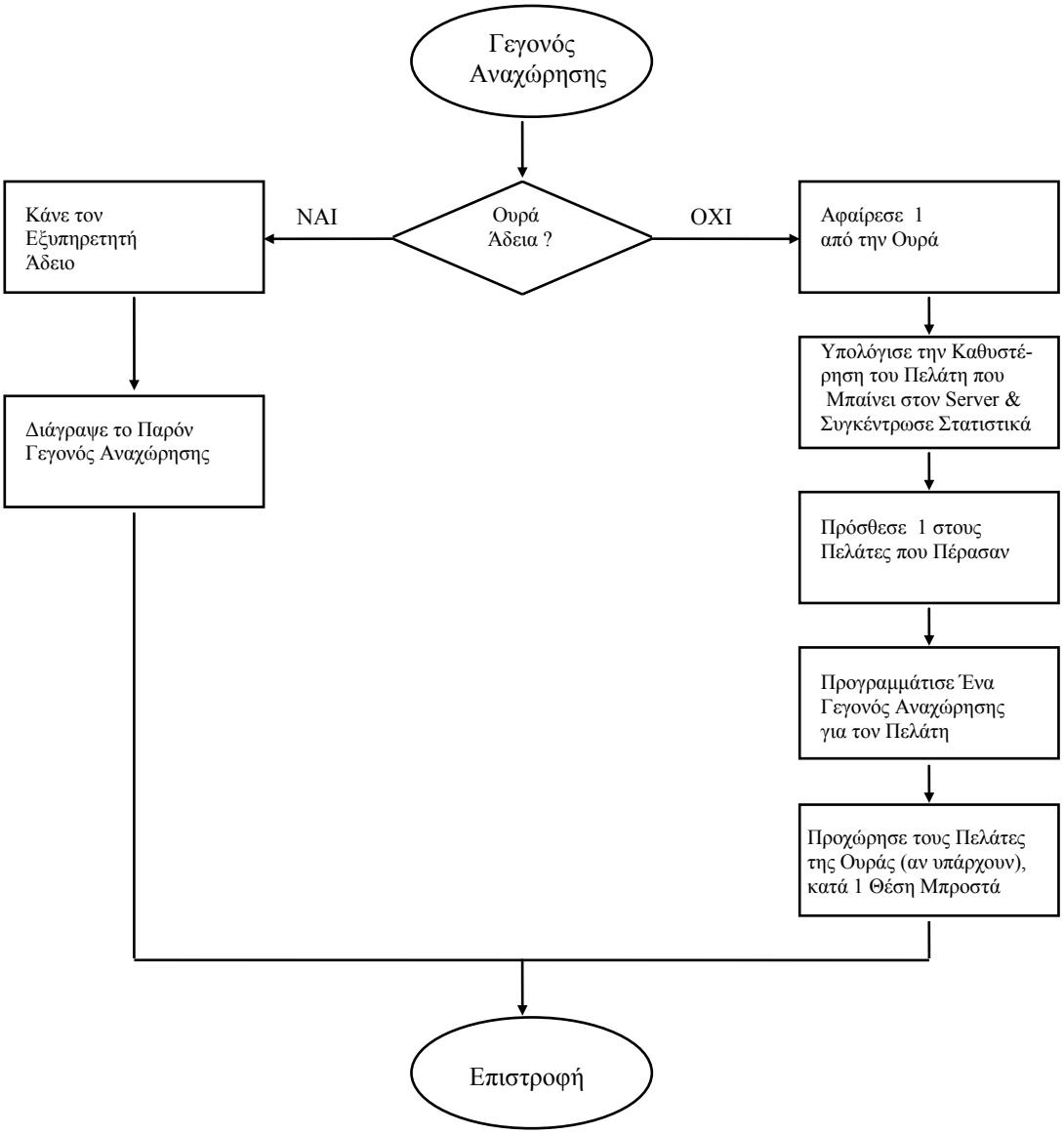
Το πρόγραμμα του προσομοιωτή πρέπει να κατασκευάζεται κατά ενότητες (modules), ώστε να ξεκαθαρίζεται η λογική και οι αλληλεπιδράσεις των τμημάτων του προγράμματος, όπως περιγράφηκαν στην Παράγραφο 2.3. Οι πιο σημαντικές ενέργειες γίνονται στις ρουτίνες γεγονότων, τα οποία αριθμούμε ως εξής:

Περιγραφή Γεγονότος	*Τύπος Γεγονότος*
• Άφιξη Πελάτη στο Σύστημα	1
• Αναχώρηση Πελάτη από το Σύστημα αφού ολοκλήρωσε την εξυπηρέτησή του	2



Σχήμα 6.6. Διάγραμμα Ροής για το Γεγονός Άφιξης

Στο Σχήμα 6.6 παρουσιάζεται το διάγραμμα ροής για το γεγονός 1, δηλαδή την άφιξη ενός πελάτη στο σύστημα. Αντίστοιχα το Σχήμα 7 εμφανίζει το διάγραμμα ροής του γεγονότος 2, της αναχώρησης ενός πελάτη από το σύστημα.

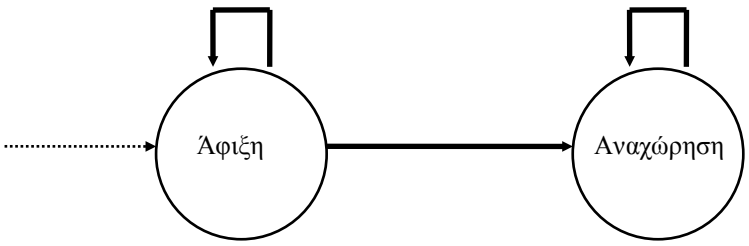


Σχήμα 6.7. Διάγραμμα Ροής για το Γεγονός Αναχώρησης

6.4 ΚΑΘΟΡΙΣΜΟΣ ΤΩΝ ΓΕΓΟΝΟΤΩΝ ΚΑΙ ΤΩΝ ΜΕΤΑΒΛΗΤΩΝ

Το *γεγονός* ορίζεται ως η στιγμιαία ενέργεια που μπορεί να αλλάξει την κατάσταση του συστήματος. Στο απλό σύστημα αναμονής που μελετήσαμε, δεν ήταν δύσκολο να προσδιορίσουμε τα γεγονότα. Όμως, σε πιο πολύπλοκα συστήματα, παραμένει το ερώτημα του συστηματικού καθορισμού του αριθμού και του τύπου των γεγονότων στο μοντέλο προσομοίωσης. Ίσως επίσης, υπάρχει δυσκολία στον προσδιορισμό των μεταβλητών κατάστασης που απαιτούνται για να συνεχίσει να εκτελείται η προσομοίωση ακολουθώντας τη σωστή ακολουθία γεγονότων, ώστε να πάρουμε τις επιθυμητές μετρήσεις. Δεν υπάρχει γενική μέθοδος που να απαντάει στα προβλήματα αυτά, όμως έχουν γίνει αποδεκτές ορισμένες βασικές αρχές και τεχνικές που βοηθούν στην απλοποίηση της δομής του μοντέλου και επιτρέπουν την αναπαράστασή του με βάση τα γεγονότα και τις μεταβλητές, έτσι ώστε να αποφεύγονται λογικά λάθη.

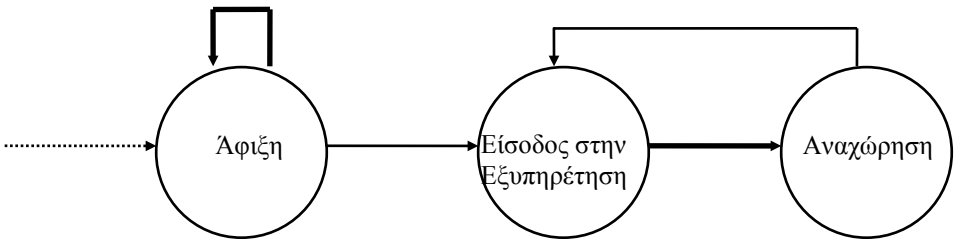
Πρόσφατα, έχει αναπτυχθεί μία μέθοδος *"γράφου-γεγονότων",* στην οποία τα προτεινόμενα γεγονότα αντιπροσωπεύονται από *κορυφές* και συνδέονται με *κατευθυνόμενες πλευρές* (βέλη), που δείχνουν πώς τα γεγονότα μπορεί να δρομολογηθούν από άλλα γεγονότα, ή από τα ίδια. Συγκεκριμένα, στο παράδειγμά μας, το γεγονός άφιξης δρομολογεί μία μελλοντική εμφάνιση του εαυτού του και πιθανόν ενός γεγονότος αναχώρησης, ενώ το γεγονός αναχώρησης μπορεί να δρομολογήσει μία μελλοντική εμφάνιση του εαυτού του. Ακόμα, το γεγονός άφιξης πρέπει να δρομολογηθεί ως αρχικό γεγονός, ώστε να μπορεί να αρχίσει η προσομοίωση. Οι γράφοι γεγονότων συνδέουν το προτεινόμενο σύνολο γεγονότων (κορυφές) μέσω πλευρών, έτσι ώστε να φαίνονται οι τρόποι δρομολόγησης γεγονότων που μπορεί να εμφανισθούν. Στο Σχήμα 6.8 φαίνεται ο γράφος γεγονότων για το παράδειγμά μας, όπου τα έντονα βέλη καθορίζουν ότι το γεγονός στο τέλος του βέλους, μπορεί να δρομολογηθεί από το γεγονός που βρίσκεται στην αρχή του βέλους μετά από χρόνο (πιθανόν) μη-μηδενικό, ενώ τα διακεκομμένα βέλη καθορίζουν το γεγονός που έχει δρομολογηθεί αρχικά. Δηλαδή, το γεγονός άφιξης επαναδρομολογεί τον εαυτό του και μπορεί να δρομολογήσει μία αναχώρηση (στην περίπτωση που η άφιξη βρίσκει τον εξυπηρετητή άεργο) και το γεγονός αναχώρησης μπορεί να επαναδρομολογήσει τον εαυτό του (αν η αναχώρηση αφήσει κάποιον άλλο πελάτη στην ουρά).



Σχήμα 6.8. Γράφος Γεγονότων, Μοντέλο Συστήματος Αναμονής

Μπορεί να ρωτήσει κάποιος, γιατί δεν παίρνουμε υπ' όψη την ενέργεια εισόδου ενός πελάτη στον εξυπηρετητή ως ένα ξεχωριστό γεγονός, αφού μπορεί να αλλάξει την κατάσταση του συστήματος, όπως η μείωση του μήκους της ουράς κατά ένα. Πράγματι μπορούμε να θεωρήσουμε και αυτό το γεγονός χωρίς να είναι λάθος το μοντέλο, οπότε θα είχαμε το γράφο γεγονότων του Σχήματος 6.9. Τα δύο λεπτά συνεχή βέλη συνδέουν ένα γεγονός που δρομολογεί ένα άλλο μετά από μηδενικό χρόνο, δηλαδή αυτόματα. Εδώ,

το αριστερό λεπτό συνεχές βέλος αναφέρεται σε έναν πελάτη που φθάνει σε άδειο σύστημα και του οποίου το γεγονός "είσοδος στην εξυπηρέτηση" δρομολογείται να εμφανισθεί αμέσως, ενώ το δεξιό λεπτό συνεχές βέλος αναφέρεται σε έναν πελάτη που αναχωρεί από το σύστημα αφήνοντας πελάτες

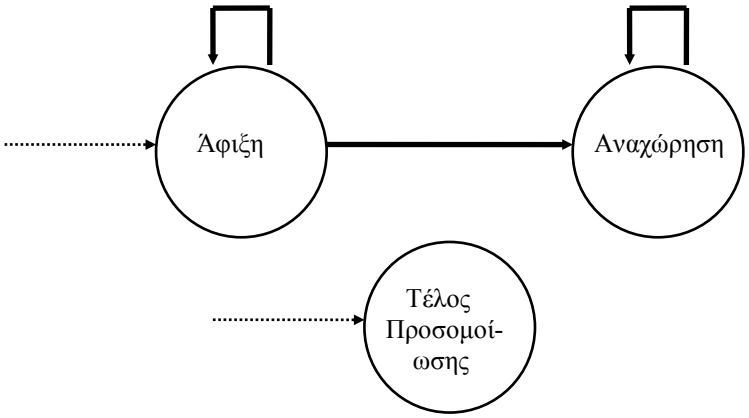


Σχήμα 6.9. Γράφος Γεγονότων, Μοντέλο Συστήματος Αναμονής με Γεγονός ''Είσοδος στην Εξυπηρέτηση''.

στην ουρά, οπότε ο πρώτος από αυτούς δρομολογείται να εισέλθει στην εξυπηρέτηση αμέσως. Ο αριθμός των τύπων γεγονότων έχει πλέον αυξηθεί κατά ένα και συνεπώς έχουμε ένα κάπως πιο πολύπλοκο μοντέλο. Μία από τις χρήσεις του γράφου γεγονότων είναι η απλοποίηση της δομής των γεγονότων της προσομοίωσης, με την απαλοιφή "άχρηστων" γεγονότων. Υπάρχουν διάφοροι κανόνες που επιτρέπουν την απλοποίηση. Ένας από αυτούς δηλώνει ότι αν ένα γεγονός στο γράφο έχει εισερχόμενα βέλη που είναι όλα λεπτά και συνεχή (δηλαδή, ο μόνος τρόπος να δρομολογηθεί το γεγονός αυτό, είναι από άλλα γεγονότα χωρίς τη μεσολάβηση χρόνου), τότε το γεγονός αυτό μπορεί να απαλοιφεί από το μοντέλο και οι ενέργειές του να ενσωματωθούν στα γεγονότα που το δρομολογούν σε μηδενικό χρόνο. Εδώ, το γεγονός "είσοδος στην εξυπηρέτηση" μπορεί να απαλοιφεί και οι ενέργειες που συνεπάγεται να ενσωματωθούν εν μέρει στο γεγονός άφιξης (όταν ένας πελάτης φθάνει σε άδειο σύστημα και αρχίζει αμέσως εξυπηρέτηση) και εν μέρει στο γεγονός αναχώρησης (όταν ένας πελάτης αναχωρεί ενώ υπάρχουν άλλοι πελάτες στην ουρά). Η απαλοιφή αυτή θα μας οδηγήσει πάλι στον απλό γράφο του Σχήματος 6.8. Συμπερασματικά, γεγονότα που μπορούν να εμφανισθούν μόνο σε συνδυασμό με άλλα γεγονότα, δεν χρειάζεται να υπάρχουν στο μοντέλο. Η μείωση του αριθμού των γεγονότων δεν απλοποιεί μόνο τη δομή του μοντέλου, αλλά μπορεί ακόμα να μειώσει το χρόνο εκτέλεσής του. Πρέπει, πάντως, να γίνεται με προσοχή η απαλοιφή, ώστε να γίνεται κατάλληλος χειρισμός τυχόν προτεραιοτήτων και ταυτόχρονων εμφανίσεων γεγονότων.

Ένας άλλος κανόνας σχετίζεται με την αρχικοποίηση. Ο γράφος γεγονότων μπορεί να αποσυντεθεί σε *ισχυρά συνδεδεμένα στοιχεία* (strongly connected components), μέσα στα οποία είναι δυνατή η μετακίνηση από κορυφή σε κορυφή ακολουθώντας τις πλευρές στην κατεύθυνση που δείχνει η καθεμιά. Ο γράφος του Σχήματος 6.8 αποσυντίθεται σε δύο ισχυρά συνδεδεμένα στοιχεία (με μία κορυφή στο καθένα), ενώ ο γράφος του Σχήματος 6.9 επίσης σε δύο ισχυρά συνδεδεμένα στοιχεία (ένα από τα οποία είναι μόνο το γεγονός άφιξης και το άλλο αποτελείται από τα δύο υπόλοιπα γεγονότα). Ο κανόνας αρχικοποίησης δηλώνει ότι σε κάθε ισχυρά συνδεδεμένο στοιχείο το οποίο δεν έχει εισερχόμενες πλευρές από άλλα γεγονότα έξω από το στοιχείο, πρέπει να υπάρχει τουλάχιστον ένα γεγονός που έχει δρομολογηθεί ως αρχικό γεγονός. Αν δεν ίσχυε ο κανόνας αυτός, δεν θα ήταν δυνατόν να εκτελεσθεί κανένα γεγονός του στοιχείου. Στα Σχήματα 6.8 και 6.9, το γεγονός άφιξης είναι ένα τέτοιο στοιχείο χωρίς εισερχόμενες πλευρές από άλλα γεγονότα και κατά συνέπεια πρέπει να δρομολογηθεί ως αρχικό

γεγονός. Το Σχήμα 6.10 δείχνει το γράφο γεγονότων για το τροποποιημένο μοντέλο, με το τρίτο γεγονός "Τέλος Προσομοίωσης". Μπορούμε να παρατηρήσουμε ότι το γεγονός αυτό είναι από μόνο του ένα ισχυρά συνδεδεμένο στοιχείο χωρίς εισερχόμενες πλευρές, οπότε πρέπει να δρομολογηθεί ως αρχικό γεγονός, δηλαδή το τέλος της προσομοίωσης θα πρέπει να δρομολογηθεί ως τμήμα της αρχικοποίησης του μοντέλου. Αν δεν γίνει αυτό, θα έχουμε λανθασμένο τερματισμό της προσομοίωσης.



Σχήμα 6.10. Γράφος Γεγονότων, Μοντέλο Συστήματος Αναμονής με Σταθερό Χρόνο Εκτέλεσης

Υπάρχουν αρκετά άλλα χαρακτηριστικά και τρόποι χρησιμοποίησης των γράφων γεγονότων, όπως: εισαγωγή σχέσεων ακύρωσης γεγονότων, συγχώνευση παρόμοιων γεγονότων σε ένα, εκλέπτυνση των πλευρών του γράφου ώστε να περιλαμβάνουν και υπό συνθήκη δρομολογήσεις κ.α., τα οποία είναι πέρα από τους σκοπούς του μαθήματος.

Όταν κατασκευάζουμε το μοντέλο ενός συστήματος, η τεχνική του γράφου γεγονότων μπορεί να χρησιμοποιηθεί για την απλοποίηση της δομής του μοντέλου και για τον εντοπισμό ορισμένων ειδών λαθών. Είναι ιδιαίτερα χρήσιμη τεχνική σε πολύπλοκα μοντέλα που περιλαμβάνουν μεγάλο αριθμό αλληλοεξαρτημένων γεγονότων. Πάντως, και άλλα θέματα πρέπει να είναι στο επίκεντρο της προσοχής μας, όπως το συνεχές ερώτημα "γιατί μία συγκεκριμένη μεταβλητή κατάστασης μας χρειάζεται", κ.λ.π.

ΚΕΦΑΛΑΙΟ 7
Η ΕΠΙΛΟΓΗ ΠΙΘΑΝΟΤΙΚΩΝ ΚΑΤΑΝΟΜΩΝ ΕΙΣΟΔΟΥ

7.1 ΕΙΣΑΓΩΓΗ

Προκειμένου να υλοποιηθεί μια προσομοίωση με τη χρήση τυχαίων εισόδων (όπως οι χρόνοι μεταξύ διαδοχικών αφίξεων και οι χρόνοι εξυπηρέτησης στο παράδειγμα που εξετάσαμε στο προηγούμενο Κεφάλαιο), πρέπει να καθορίσουμε τις πιθανοτικές κατανομές που τις περιγράφουν. Η προσομοίωση εξελίσσεται με τη δημιουργία και χρήση *τυχαίων τιμών* από τις κατανομές αυτές. Στον Πίνακα 7.1 φαίνονται ορισμένες πιθανές πηγές τυχαιότητας σε πραγματικά συστήματα.

Τύπος Συστήματος	*Πηγές Τυχαιότητας*
Υπολογιστές	Χρόνοι μεταξύ αφίξεων εργασιών, Τύποι εργασιών, Απαιτήσεις επεξεργασίας εργασιών.
Δίκτυα - Τηλεπικοινωνίες	Χρόνοι μεταξύ αφίξεων μηνυμάτων, Τύποι μηνυμάτων, Μήκη μηνυμάτων.
Συστήματα Παραγωγής	Χρόνοι επεξεργασίας, Χρόνοι λειτουργίας μηχανών μέχρι να χαλάσουν, Χρόνοι επισκευής μηχανών.

Πίνακας 7.1. Πηγές Τυχαιότητας Πραγματικών Συστημάτων

Αν είναι δυνατόν να συλλέξουμε δεδομένα από μια τυχαία μεταβλητή εισόδου που μας ενδιαφέρει, μπορούμε να τα χρησιμοποιήσουμε με τους εξής τρόπους, προκειμένου να προσδιορίσουμε την κατανομή εισόδου:

1. Τα ίδια τα δεδομένα χρησιμοποιούνται απευθείας στην προσομοίωση. Τότε θα έχουμε *ιχνο-οδηγούμενη προσομοίωση* (βλέπε Παράγραφο 2.1).
2. Τα δεδομένα χρησιμοποιούνται για να ορισθεί μία *εμπειρική* συνάρτηση κατανομής για τη μεταβλητή εισόδου.
3. Προσαρμόζουμε μια γνωστή *θεωρητική* μορφή κατανομής (π.χ. εκθετική ή Poisson) στα δεδομένα που έχουμε.

Δύο μειονεκτήματα του πρώτου τρόπου είναι, ότι η προσομοίωση θα αναπαράγει απλώς ότι έγινε στην πραγματικότητα και ότι σπανίως έχουμε αρκετά δεδομένα για να εκτελέσουμε όλες τις απαραίτητες προσομοιώσεις. Ο δεύτερος τρόπος αντιμετωπίζει τα μειονεκτήματα αυτά, αφού τουλάχιστον για συνεχή δεδομένα, μπορεί να παραχθεί οποιαδήποτε τιμή μεταξύ της ελάχιστης και μέγιστης που έχουμε στη διάθεσή μας. Πάντως, ενώ είναι προτιμητέος ο δεύτερος από τον πρώτο τρόπο, ο πρώτος συνιστάται

για τον *έλεγχο εγκυρότητας ενός μοντέλου,* συγκεκριμένα όταν θέλουμε να συγκρίνουμε τις εξόδους του μοντέλου με τις εξόδους του πραγματικού συστήματος.

Αν μπορούμε να βρούμε μια θεωρητική κατανομή που προσαρμόζεται ικανοποιητικά στα δεδομένα που έχουμε (τρίτος τρόπος), την προτιμούμε από την εμπειρική κατανομή (δεύτερος τρόπος), για τους παρακάτω λόγους:

- Μία εμπειρική κατανομή μπορεί να έχει σημεία ασυνέχειας, ή άλλες "ανωμαλίες", ιδιαίτερα αν διαθέτουμε λίγα μόνο δεδομένα για τον προσδιορισμό της. Αντίθετα, μια θεωρητική κατανομή εξομαλύνει τα δεδομένα και μπορεί να δώσει πληροφορία για τη συνολική συμπεριφορά της μεταβλητής εισόδου.
- Συνήθως δεν μπορούμε να παράγουμε τυχαίες τιμές από μια εμπειρική κατανομή έξω από τα όρια των δεδομένων που διαθέτουμε. Το γεγονός αυτό δημιουργεί μερικές φορές σημαντικό πρόβλημα, αφού η απόδοση ενός συστήματος μπορεί να εξαρτάται από την πιθανότητα εμφάνισης ενός ακραίου γεγονότος, όπως για παράδειγμα ενός πολύ μεγάλου χρόνου εξυπηρέτησης. Αντίθετα, μια θεωρητική κατανομή μπορεί να μας δώσει και τιμές έξω από το μέγιστο και ελάχιστο των δεδομένων που διαθέτουμε.

7.2 ΧΡΗΣΙΜΕΣ ΠΙΘΑΝΟΤΙΚΕΣ ΚΑΤΑΝΟΜΕΣ

Στο ΠΑΡΑΡΤΗΜΑ Α παρουσιάζονται μερικές κατανομές, που έχει δείξει η εμπειρία ότι είναι χρήσιμες στη μοντελοποίηση με προσομοίωση. Για παράδειγμα, έχουν καταγραφεί οι παρακάτω χρήσεις μερικών από αυτές:

1. *Εκθετική (exponential):* Χρησιμοποιείται για τη μοντελοποίηση εντελώς "τυχαίων" γεγονότων, όπως οι χρόνοι μεταξύ αστοχιών, ή οι χρόνοι μεταξύ αφίξεων πελατών. Συχνά χρησιμοποιείται ως η μόνη λύση όταν δεν υπάρχουν ενδείξεις για την κατανομή που πρέπει να χρησιμοποιηθεί.
2. *Βήτα (beta):* Χρησιμοποιείται για τη μοντελοποίηση τυχαίων αναλογιών, όπως το κλάσμα των πακέτων που απαιτούν επανεκπομπή σε μία γραμμή μετάδοσης δεδομένων.
3. *Γάμμα (gamma):* Χρησιμοποιείται για τη μοντελοποίηση χρόνων εξυπηρέτησης. Σε ένα σύστημα αναμονής περιγράφει το χρόνο εξυπηρέτησης, όταν έχουμε m εκθετικούς εξυπηρετητές εν σειρά.
4. *Δυωνυμική (binomial):* Χρησιμοποιείται στη μοντελοποίηση του αριθμού επιτυχιών σε μια ακολουθία *n* ανεξαρτήτων δοκιμών Bernoulli. Για παράδειγμα, ο αριθμός των bits σε ένα πακέτο που δεν έχουν επηρεασθεί από το θόρυβο.
5. *Γεωμετρική (geometric):* Χρησιμοποιείται στη μοντελοποίηση του αριθμού των εμφανίσεων γεγονότων μεταξύ άλλων σημαντικών γεγονότων. Π.χ. ο αριθμός των σωστών bits μεταξύ δύο λανθασμένων bits σε ένα πακέτο δεδομένων, ή ο αριθμός των αιτήσεων σε μια τοπική Βάση Δεδομένων (ΒΔ) μεταξύ δύο αιτήσεων σε μια απομακρυσμένη ΒΔ.
6. *Αρνητική Δυωνυμική (negative binomial):* Χρησιμοποιείται για τον αριθμό των αποτυχιών πριν τη *m*-στή επιτυχία, όπως ο αριθμός των επανεκπομπών ενός αρχείου που αποτελείται από *m* πακέτα, ή ο αριθμός των αιτήσεων σε μια τοπική ΒΔ πριν τη *m*-στή αίτηση σε μια απομακρυσμένη ΒΔ.

7. *Poisson:* Χρησιμοποιείται στη μοντελοποίηση του αριθμού εμφανίσεων γεγονότων κατά τη διάρκεια μιας συγκεκριμένης περιόδου, όπως ο αριθμός των αστοχιών εξαρτημάτων στη μονάδα του χρόνου, ή ο αριθμός των αιτήσεων σε μια ΒΔ σε χρονικό διάστημα *t.*

8. *Κανονική (normal):* Χρησιμοποιείται για τη μοντελοποίηση του αθροιστικού φαινομένου διαφόρων ανεξαρτήτων πηγών, όπως τα λάθη μετρήσεων.

9. *Weibull:* Χρησιμοποιείται για μετρήσεις αξιοπιστίας, όπως ο χρόνος ζωής εξαρτημάτων.

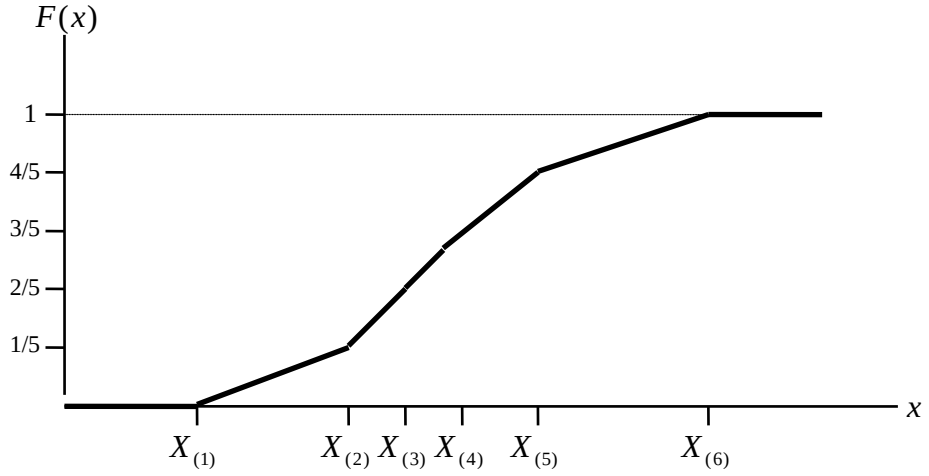
7.3 ΕΜΠΕΙΡΙΚΕΣ ΚΑΤΑΝΟΜΕΣ

Όταν δεν είναι δυνατόν να βρούμε μια θεωρητική κατανομή που να ταιριάζει στα δεδομένα που διαθέτουμε, προσπαθούμε να καθορίσουμε απευθείας μια *εμπειρική κατανομή,* από την οποία θα δημιουργούνται τυχαίες τιμές κατά τη διάρκεια της προσομοίωσης.

Για **συνεχείς** τυχαίες μεταβλητές, ο τύπος της εμπειρικής κατανομής που μπορούμε να ορίσουμε, εξαρτάται από το αν διαθέτουμε τις ίδιες τις τιμές των επιμέρους *αυθεντικών* παρατηρήσεων $X_1, X_2, \ldots, X_n$, ή μόνο τον *αριθμό* των X_i που βρίσκονται μέσα σε κάθε ένα από ορισμένα καθορισμένα διαστήματα. Στη δεύτερη περίπτωση λέμε ότι έχουμε *ομαδοποιημένα* δεδομένα, ή δεδομένα με τη μορφή *ιστογράμματος.*

Αν έχουμε στη διάθεσή μας τα αυθεντικά δεδομένα, μπορούμε να ορίσουμε μία συνεχή, τμηματικά γραμμική, συνάρτηση κατανομής *F,* έχοντας ταξινομήσει τα X_i κατά αύξουσα σειρά. Έστω $X_{(i)}$ το *i*-στό μικρότερο από τα X_j έτσι ώστε $X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}$. Τότε η *F* δίνεται από τη σχέση:

$$F(x) = \begin{cases} 0 & \alpha\nu \;\; x < X_{(1)} \\ \dfrac{i-1}{n-1} + \dfrac{x - X_{(i)}}{(n-1)(X_{(i+1)} - X_{(i)})} & \alpha\nu \;\; X_{(i)} \le x \le X_{(i+1)} \;\; \gamma\iota\alpha \;\; i = 1,2,\ldots,n-1 \\ 1 & \alpha\nu \;\; X_{(n)} \le x \end{cases}$$

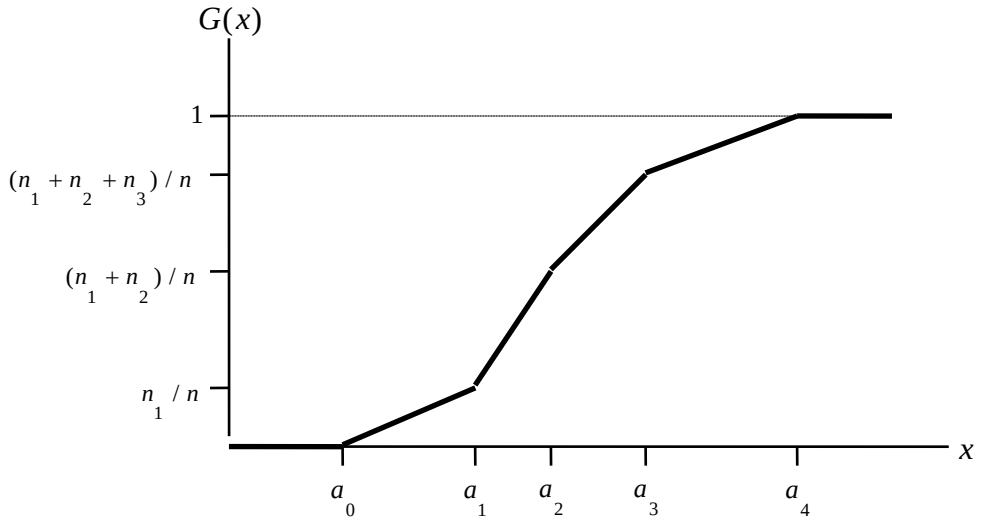


Σχήμα 7.1. Συνεχής, τμηματικά γραμμική, εμπειρική συνάρτηση κατανομής από τα αυθεντικά δεδομένα.

Το Σχήμα 7.1 δείχνει μία περίπτωση για $n = 6$. Παρατηρήστε ότι το $F(x)$ αυξάνει γρηγορότερα στις περιοχές του *x* στις οποίες τα X_i είναι κατανεμημένα πιο πυκνά, όπως επιθυμούμε. Επίσης, για κάθε *i*, $F(X_{(i)}) = (i-1)/(n-1)$, το οποίο είναι κατά προσέγγιση (για μεγάλο *n*) το ποσοστό των X_i που είναι *μικρότερα από* $X_{(i)}$, γεγονός επίσης επιθυμητό για μία *συνεχή* συνάρτηση κατανομής. Πάντως, ένα μειονέκτημα της εμπειρικής κατανομής που φτιάξαμε με τη μέθοδο αυτή, είναι ότι οι τυχαίες τιμές που δημιουργούνται από αυτήν κατά τη διάρκεια της προσομοίωσης, δεν μπορούν να είναι μικρότερες από $X_{(1)}$ ή μεγαλύτερες από $X_{(n)}$. Επίσης, η μέση τιμή της $F(x)$ δεν είναι ίση με τη δειγματοληπτική μέση τιμή $\overline{X}(n)$ των X_i.

Εάν, αντίθετα, διαθέτουμε μόνο ομαδοποιημένα δεδομένα, πρέπει να ακολουθήσουμε διαφορετική προσέγγιση, αφού δεν γνωρίζουμε τις τιμές των επιμέρους X_i. Έστω ότι τα n X_i είναι ομαδοποιημένα σε k συνεχόμενα διαστήματα $[a_0, a_1), [a_1, a_2), \ldots, [a_{k-1}, a_k)$, έτσι ώστε το *j*-στό διάστημα να περιέχει n_j παρατηρήσεις, όπου $n_1 + n_2 + \cdots + n_k = n$. Μία λογική, τμηματικά γραμμική εμπειρική συνάρτηση κατανομής *G*, θα μπορούσε να ορισθεί, αν θεωρούσαμε $G(a_0) = 0$ και $G(a_j) = (n_1 + n_2 + \cdots + n_j)/n$ για $j = 1,2,\ldots,k$. Με γραμμική παρεμβολή μεταξύ των a_j ορίζουμε:

$$G(x) = \begin{cases} 0 & \alpha\nu \quad x < a_0 \\ G(a_{j-1}) + \dfrac{x - a_{j-1}}{a_j - a_{j-1}}\left[G(a_j) - G(a_{j-1})\right] & \alpha\nu \quad a_{j-1} \le x < a_j \quad \gamma\iota\alpha \ j = 1,2,\ldots,k \\ 1 & \alpha\nu \quad a_k \le x \end{cases}$$



Σχήμα 7.2. Συνεχής, τμηματικά γραμμική, εμπειρική συνάρτηση κατανομής από τα αυθεντικά δεδομένα.

Στο Σχήμα 7.2 φαίνεται μία τέτοια κατανομή $G(x)$ για $k = 4$. Στην περίπτωση αυτή, το $G(a_j)$ είναι το ποσοστό των X_i που είναι μικρότερα από a_j και η $G(x)$ αυξάνει γρηγορότερα στα διαστήματα του *x* όπου οι παρατηρήσεις είναι πιο πυκνές. Και πάλι, οι τυχαίες τιμές που θα δημιουργηθούν από την κατανομή αυτή, θα βρίσκονται μεταξύ των a_0 και a_k.

Στην πράξη, πολλές συνεχείς κατανομές εκτείνονται μειούμενες προς τα δεξιά, με μία μορφή συνάρτησης πυκνότητας πιθανότητας $f(x)$, παρόμοια με αυτήν της κατανομής gamma(2,1) που παρατίθεται στο ΠΑΡΑΡΤΗΜΑ Α. Συνεπώς, αν το μέγεθος του δείγματος *n* δεν είναι πολύ μεγάλο, μάλλον θα έχουμε λίγες (αν έχουμε) παρατηρήσεις από τη δεξιά ουρά της πραγματικής κατανομής, αφού η πιθανότητα να βρεθούμε στην ουρά είναι συνήθως μικρή. Ακόμα, οι παραπάνω εμπειρικές κατανομές δεν επιτρέπουν τη δημιουργία τυχαίων τιμών μεγαλύτερων από τη μέγιστη παρατήρηση. Όμως, οι πολύ μεγάλες τιμές μπορεί να έχουν σημαντική επίπτωση στην προσομοίωση, όπως για παράδειγμα, ένας μεγάλος χρόνος εξυπηρέτησης ο οποίος σε ένα σύστημα αναμονής πιθανόν να προκαλέσει τη δημιουργία μεγάλων ουρών και κατά συνέπεια καθυστερήσεων. Για την αποφυγή του προβλήματος αυτού, έχει προταθεί να προστίθεται μία εκθετική κατανομή στο δεξιό τμήμα της εμπειρικής κατανομής, ώστε να επιτρέπεται η δημιουργία τυχαίων τιμών μεγαλύτερων από $X_{(n)}$.

Για **διακριτά** δεδομένα, είναι αρκετά εύκολο να ορισθεί μία εμπειρική κατανομή, αν είναι διαθέσιμες οι αυθεντικές τιμές $X_1, X_2, \ldots, X_n$. Για κάθε πιθανή τιμή *x*, μπορεί να ορισθεί μία εμπειρική συνάρτηση μάζας πιθανότητας $p(x)$ η οποία θα είναι το ποσοστό των X_i που θα είναι ίσα με *x*. Για ομαδοποιημένα διακριτά δεδομένα, θα μπορούσαμε να ορίσουμε μία συνάρτηση μάζας, έτσι ώστε το *άθροισμα* των $p(x)$ πάνω σε όλες τις πιθανές τιμές του *x* σε ένα διάστημα, να είναι ίσο με το ποσοστό των X_i στο διάστημα αυτό. Ο τρόπος που δημιουργούνται τα επιμέρους $p(x)$ για τις πιθανές τιμές του *x* μέσα σε ένα διάστημα, είναι ουσιαστικά αυθαίρετος.

7.4 ΠΡΟΣΑΡΜΟΓΗ ΘΕΩΡΗΤΙΚΗΣ ΚΑΤΑΝΟΜΗΣ

Όταν θέλουμε να βρούμε μια γνωστή κατανομή και να την προσαρμόσουμε στα δεδομένα που διαθέτουμε, ώστε να τη χρησιμοποιήσουμε στη συνέχεια για τη δημιουργία τυχαίων τιμών, ακολουθούμε γενικά τα παρακάτω βήματα:

Βήμα 1: Δημιουργούμε ένα ιστόγραμμα από τα δεδομένα που διαθέτουμε.

Βήμα 2: Επιλέγουμε μια γνωστή κατανομή, με μορφή παρόμοια με αυτήν του ιστογράμματος που έχουμε κατασκευάσει.

Βήμα 3: Υπολογίζουμε την πρώτη και δεύτερη δειγματοληπτική ροπή των δεδομένων που έχουμε.

Βήμα 4: Χρησιμοποιούμε τις ροπές για να υπολογίσουμε τις παραμέτρους της θεωρητικής κατανομής (δηλ. αντιστοιχούμε τις δειγματοληπτικές ροπές των δεδομένων, με τις ροπές της θεωρητικής κατανομής).

Βήμα 5: Κάνουμε ένα τεστ για να δούμε πόσο καλά αντιπροσωπεύει τα δεδομένα μας, η θεωρητική κατανομή που επιλέξαμε (π.χ. το χ^2 τεστ για διακριτές κατανομές και το τεστ Kolmogorov - Smirnov για συνεχείς κατανομές).

Αν δεν μπορέσουμε να βρούμε θεωρητική κατανομή με τον τρόπο αυτό, τότε ίσως να είναι δυνατόν να μετασχηματίσουμε με κάποιο τρόπο τα αρχικά δεδομένα. Δηλαδή, δημιουργούμε ένα νέο σύνολο δεδομένων Y_i από τα αρχικά δεδομένα X_i, με $Y_i = g(X_i)$, όπου $g(x)$ είναι μία συνάρτηση μετασχηματισμού. Στη συνέχεια εφαρμόζουμε και πάλι τα παραπάνω 5 βήματα, ελπίζοντας να προσδιορίσουμε μία θεωρητική κατανομή που θα προσαρμοσθεί ικανοποιητικά στα μετασχηματισμένα δεδομένα Y_i. Στην επιλογή της $g(x)$ συνιστάται να χρησιμοποιούμε απλές συναρτήσεις, όπως γραμμικούς μετασχηματισμούς, λογαρίθμους, ή δυνάμεις. Είναι πιθανό πάντως να χρειασθούμε αρκετές δοκιμές μέχρι να βρούμε μια θεωρητική κατανομή που να προσαρμόζεται ικανοποιητικά στα αρχικά δεδομένα.

ΚΕΦΑΛΑΙΟ 8
ΓΕΝΝΗΤΡΙΕΣ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ

8.1 ΕΙΣΑΓΩΓΗ

Στο Κεφάλαιο αυτό θα δούμε πώς μπορούμε να δημιουργήσουμε *τυχαίους αριθμούς* από την ομοιόμορφη κατανομή στο διάστημα [0,1]. Την κατανομή αυτή, συμβολίζουμε $U(0,1)$. Αν και η ομοιόμορφη κατανομή είναι η απλούστερη συνεχής κατανομή, είναι σημαντικό να μπορέσουμε να δημιουργήσουμε τυχαίους αριθμούς από αυτήν, αφού στη συνέχεια έχουμε τη δυνατότητα να πάρουμε *τυχαίες τιμές* από όλες τις άλλες κατανομές (εκθετική, κανονική, γάμμα, δυωνυμική κ.α.), καθώς και υλοποιήσεις διαφόρων στοχαστικών διαδικασιών (π.χ. διαδικασία Poisson), εφαρμόζοντας ένα μετασχηματισμό στους τυχαίους αριθμούς της ομοιόμορφης κατανομής. Το είδος του μετασχηματισμού εξαρτάται από την κατανομή ή τη διαδικασία από την οποία θέλουμε να πάρουμε τυχαίες τιμές. Στο επόμενο Κεφάλαιο θα ασχοληθούμε με τη δημιουργία τυχαίων τιμών από οποιαδήποτε κατανομή ή διαδικασία.

Οι μέθοδοι δημιουργίας τυχαίων αριθμών που θα δούμε στη συνέχεια (όπως και όλες οι γεννήτριες τυχαίων αριθμών που έχουν υλοποιηθεί σε υπολογιστές), είναι *αριθμητικές* και εντελώς *ντετερμινιστικές,* αφού κάθε νέος τυχαίος αριθμός καθορίζεται από έναν ή περισσότερους από τους προηγούμενους αριθμούς σύμφωνα με μια σταθερή μαθηματική φόρμουλα. Το γεγονός αυτό μας οδηγεί στην παρατήρηση ότι ουσιαστικά δεν έχουμε "τυχαίους" αριθμούς, αλλά *ψευδο-τυχαίους*, όπως εξάλλου πολλές φορές ονομάζονται. Παρ' όλα αυτά, μία σωστά σχεδιασμένη γεννήτρια ψευδο-τυχαίων αριθμών, μπορεί να παράγει ακολουθίες αριθμών με "τυχαία" στατιστικά χαρακτηριστικά, όπως μπορεί να αποδειχθεί αν οι ακολουθίες περάσουν με επιτυχία μια σειρά από στατιστικά τεστ, όπως το χ^2 τεστ, το τεστ *συσχέτισης,* τεστ *εκτελέσεων,* το τεστ *φάσματος* κ.α. Για το λόγο αυτό θα συνεχίσουμε να χρησιμοποιούμε τον όρο "τυχαίοι αριθμοί".

Μία "καλή" γεννήτρια τυχαίων αριθμών πρέπει να έχει ορισμένες ιδιότητες:

Πάνω απ' όλα, οι παραγόμενοι αριθμοί πρέπει να φαίνονται ομοιόμορφα κατανεμημένοι στο διάστημα [0,1] και δεν θα πρέπει να εμφανίζουν καμμιά συσχέτιση μεταξύ τους.

Η γεννήτρια θα πρέπει να είναι γρήγορη και να μην απαιτεί πολύ χώρο αποθήκευσης στον υπολογιστή.

Είναι επιθυμητό να μπορούμε να ξαναπάρουμε μια δεδομένη ακολουθία τυχαίων αριθμών, με την ίδια ακριβώς μορφή. Οι λόγοι είναι δύο: Πρώτον, με τη χρήση ίδιας ακολουθίας, μπορεί να γίνει ευκολότερη η επιβεβαίωση και η διόρθωση λαθών του προσομοιωτή. Δεύτερον, μπορεί να γίνει καλύτερη σύγκριση της προσομοίωσης διαφορετικών συστημάτων.

Η γεννήτρια θα πρέπει να μπορεί να δημιουργεί αρκετά διαφορετικά "ρεύματα" τυχαίων αριθμών. Μπορούμε να θεωρήσουμε τα διαφορετικά ρεύματα ως ξεχωριστές και ανεξάρτητες γεννήτριες, οι οποίες τροφοδοτούν με τυχαίους αριθμούς διαφορετικές πηγές τυχαιότητας του προσομοιωτή, όπως οι χρόνοι μεταξύ διαδοχικών αφίξεων και οι χρόνοι εξυπηρέτησης στο παράδειγμα που μελετήσαμε στο Κεφάλαιο 3.

8.2 ΓΡΑΜΜΙΚΕΣ ΑΝΑΛΟΓΙΚΕΣ ΓΕΝΝΗΤΡΙΕΣ

Η μεγάλη πλειονότητα των γεννητριών τυχαίων αριθμών που χρησιμοποιούνται σήμερα, είναι *Γραμμικές Αναλογικές Γεννήτριες* - Linear Congruential Generators (LCG). Στις γεννήτριες αυτές, ορίζεται μία ακολουθία ακεραίων $Z_1, Z_2, \ldots$ από την αναδρομική σχέση:

$$Z_i = (aZ_{i-1} + c) (\mathrm{mod}\, m) \qquad (8.1)$$

όπου m (ο *διαιρέτης*), a (ο *πολλαπλασιαστής*), c (η *αύξηση*) και Z_0 (ο *σπόρος* ή *αρχική τιμή*), είναι όλα μη-αρνητικοί ακέραιοι. Δηλαδή, η (8.1) ορίζει ότι για να πάρουμε το Z_i, διαιρούμε το $aZ_{i-1} + c$ με m και κρατάμε το *υπόλοιπο* της διαίρεσης. Συνεπώς, $0 \le Z_i \le m-1$ και για να πάρουμε τους *τυχαίους αριθμούς* U_i (για $i = 1, 2, \ldots$) στο διάστημα [0,1], θέτουμε:

$$U_i = \frac{Z_i}{m}$$

Θα επικεντρώσουμε την προσοχή μας κυρίως στα Z_i, αν και έχει σημασία η φύση της διαίρεσης των Z_i με το m, λόγω του τρόπου χειρισμού της από τους διάφορους υπολογιστές και γλώσσες προγραμματισμού. Για τους ακέραιους m, a, c και Z_0, εκτός από τη μη-αρνητικότητα, πρέπει να ισχύουν οι σχέσεις: $0 < m$, $a < m$, $c < m$ και $Z_0 < m$.

Με μια πρώτη ματιά, μπορούν να εγερθούν δύο ενστάσεις εναντίον των LCG. Η πρώτη είναι κοινή για όλες τις γεννήτριες (ψευδο) τυχαίων αριθμών: τα Z_i όπως ορίζονται στην (8.1), δεν είναι αληθινοί τυχαίοι αριθμοί. Στην πραγμα-τικότητα, μπορεί να αποδειχθεί με επαγωγή, ότι για $i = 1, 2, \ldots$,

$$Z_i = \left[a^i Z_o + \frac{c(a^i - 1)}{a - 1} \right] (\mathrm{mod}\, m)$$

έτσι ώστε κάθε Z_i είναι απολύτως ορισμένο από τα m, a, c και Z_0. Πάντως, με προσεκτική επιλογή των τεσσάρων αυτών παραμέτρων, προσπαθούμε να έχουμε τέτοια Z_i, ώστε τα αντίστοιχα U_i να *φαίνονται* ότι είναι ανεξάρτητες, όμοια κατανεμημένες $U(0,1)$ τυχαίες μεταβλητές, όταν υποβληθούν στα κατάλληλα τεστ.

Η δεύτερη ένσταση είναι ότι τα U_i μπορούν να πάρουν μόνο τις τιμές 0, 1/m, 2/m, . . . , (m-1)/m. Πράγματι, τα U_i μπορούν να πάρουν μόνο ένα τμήμα από τις τιμές αυτές, το οποίο εξαρτάται από τον ορισμό των σταθερών m, a, c και Z_0 και από τη φύση της διαίρεσης με m. Δηλαδή, δεν υπάρχει δυνατότητα να πάρουμε τιμή για τα U_i μεταξύ, ας πούμε, 0.1/m και 0.9/m, παρότι αυτό θα έπρεπε να εμφανίζεται με πιθανότητα 0.8/m>0. Αυτό που γίνεται, είναι να επιλέγουμε συνήθως το m πολύ μεγάλο, π.χ. 10^9 και μεγαλύτερο, έτσι ώστε τα σημεία του [0,1] στα οποία μπορεί να πέσουν τα U_i να είναι πάρα πολλά. Για $m \ge 10^9$, υπάρχουν τουλάχιστον ένα δισεκατομμύριο πιθανές τιμές.

Στο Σχήμα 8.1 φαίνονται οι τιμές των Z_i και U_i μίας LCG με m = 16, a = 5, c = 3 και Z_0=7, για $i = 1, 2, \ldots, 19$. Μπορούμε να παρατηρήσουμε ότι $Z_{17} = Z_1 = 6$, $Z_{18} = Z_2 = 1$ κ.ο.κ. Δηλαδή, για $i = 17$ ως 32, θα παίρνουμε *ακριβώς* τις ίδιες τιμές των Z_i (και συνεπώς των U_i), που έχουμε πάρει για $i = 1$ μέχρι 16 και με *ακριβώς* την ίδια σειρά. Φυσικά δεν θα χρησιμοποιήσουμε ποτέ μια γεννήτρια με τόσο μικρό m, αλλά μπορούμε να δούμε εδώ τον τρόπο λειτουργίας μιας LCG.

i	Z_i	U_i	i	Z_i	U_i	i	Z_i	U_i	i	Z_i	U_i
0	7	-	5	10	0.625	10	9	0.563	15	4	0.250
1	6	0.375	6	5	0.313	11	0	0.000	16	7	0.438
2	1	0.063	7	12	0.750	12	3	0.188	17	6	0.375
3	8	0.500	8	15	0.938	13	2	0.125	18	1	0.063
4	11	0.688	9	14	0.875	14	13	0.813	19	8	0.500

Σχήμα 8.0.1. Η Γραμμική Αναλογική Γεννήτρια $Z_i = (5Z_{i-1} + 3)(\mathrm{mod}\,16)$, με $Z_0 = 7$.

Η επαναληπτική αυτή συμπεριφορά της γεννήτριας, είναι φυσικά αναπόφευκτη, λόγω της μορφής της (8.1), σύμφωνα με την οποία όποτε το Z_i παίρνει μια τιμή που είχε πάρει και προηγουμένως, δημιουργείται στη συνέχεια η ίδια ακριβώς ακολουθία τιμών. Ο κύκλος αυτός επαναλαμβάνεται συνεχώς. Το μήκος του κύκλου ονομάζεται *περίοδος* της γεννήτριας. Στις LCG, το Z_i εξαρτάται μόνο από την αμέσως προηγούμενη τιμή Z_{i-1} και όχι από παλιότερες τιμές. Αφού, λοιπόν, $0 \le Z_i \le m-1$, είναι σαφές ότι η μέγιστη τιμή της περιόδου είναι m. Αν μάλιστα, είναι ίση με m, τότε λέμε ότι η LCG έχει *πλήρη περίοδο*, όπως η γεννήτρια του Σχήματος 29. Αν μία γεννήτρια έχει πλήρη περίοδο, οποιαδήποτε επιλογή αρχικής τιμής Z_0 από το σύνολο των ακεραίων $\{0,1,\ldots,m-1\}$ θα παράγει τον πλήρη κύκλο με κάποια σειρά. Αν όμως μία γεννήτρια έχει περίοδο μικρότερη της πλήρους, το μήκος της θα εξαρτάται από τη συγκεκριμένη τιμή του Z_0 που έχει επιλεγεί, οπότε ουσιαστικά θα μιλάμε για την περίοδο της *αρχικής τιμής* της γεννήτριας αυτής.

Μία προσομοίωση μπορεί να χρειασθεί εκατοντάδες χιλιάδων τυχαίους αριθμούς και κατά συνέπεια είναι επιθυμητό να έχουμε LCGs με μεγάλες περιόδους. Επίσης, είναι βολικό να έχουμε LCGs με πλήρη περίοδο, διότι έτσι είμαστε σίγουροι ότι κάθε ακέραιος μεταξύ 0 και $m-1$ θα εμφανισθεί ακριβώς μία φορά σε κάθε κύκλο, γεγονός που βοηθάει στην ομοιομορφία της κατανομής των U_i. Το παρακάτω Θεώρημα 3 μας δίνει τους περιορισμούς που πρέπει να ισχύουν στην επιλογή των m, a και c, έτσι ώστε η LCG που προσδιορίζουν, να έχει πλήρη περίοδο:

Θεώρημα 3.

Η Γραμμική Αναλογική Γεννήτρια της Σχέσης (8.1), έχει πλήρη περίοδο αν και μόνο αν ισχύουν οι παρακάτω τρεις συνθήκες:

Ο μόνος θετικός ακέραιος που διαιρεί ακριβώς και το m και το c, είναι το 1.

Αν το q είναι ένας πρώτος αριθμός που διαιρεί ακριβώς το m, τότε το q διαιρεί ακριβώς και το $a-1$.

Αν το 4 διαιρεί ακριβώς το m, τότε το 4 διαιρεί ακριβώς και το $a-1$.

□

Η ύπαρξη πλήρους ή τουλάχιστον μεγάλης περιόδου, είναι μόνο μία από τις επιθυμητές ιδιότητες μιας καλής LCG. Όπως αναφέρθηκε στην προηγούμενη Παράγραφο, θέλουμε η γεννήτρια να έχει και καλές στατιστικές ιδιότητες, υπολογιστική και αποθηκευτική αποδοτικότητα, δυνατότητα αναπαραγωγής της ίδιας ακολουθίας και επίσης ξεχωριστών ρευμάτων τυχαίων αριθμών. Η αναπαραγωγή της ίδιας ακολουθίας είναι εύκολη, αφού χρειάζεται να θυμόμαστε μόνο την αρχική τιμή Z_0 και να

ξαναρχίσουμε τη γεννήτρια με την τιμή αυτή, οπότε θα πάρουμε την ίδια ακολουθία τυχαίων αριθμών U_i. Επίσης, μπορούμε εύκολα να επανέλθουμε στη δημιουργία των Z_i σε οποιοδήποτε σημείο της ακολουθίας, αποθηκεύοντας το τελευταίο Z_i που δημιουργήθηκε και χρησιμοποιώντας ως τη νέα αρχική τιμή. Έτσι έχουμε έναν εύκολο τρόπο για να πάρουμε μη-επικαλυπτόμενες, "ανεξάρτητες" ακολουθίες τυχαίων αριθμών.

Τα διαφορετικά ρεύματα σε μια LCG, ορίζονται συνήθως με τον καθορισμό της αρχικής τιμής για κάθε ρεύμα. Για παράδειγμα, αν θέλουμε ρεύματα μήκους 100,000 το καθένα, θα δώσουμε κάποια τιμή στο Z_0 για το πρώτο ρεύμα, μετά θα χρησιμοποιήσουμε το $Z_{100,000}$ ως αρχική τιμή για το δεύτερο ρεύμα, το $Z_{200,000}$ ως αρχική τιμή για το τρίτο ρεύμα κ.ο.κ. Παρατηρούμε ότι έτσι, τα ρεύματα είναι ουσιαστικά μη-επικαλυπτόμενες, γειτονικές, υπο-ακολουθίες μιας απλής ακολουθίας τυχαίων αριθμών. Αν, πάντως, χρειασθούν περισσότεροι από 100,000 τυχαίοι αριθμοί σε κάποιο ρεύμα, θα χρησιμοποιηθούν αριθμοί από την αρχή του επομένου ρεύματος, που μπορεί να έχουν ήδη χρησιμοποιηθεί αλλού, έτσι ώστε να καταλήξουμε σε απαράδεκτη συσχέτιση των ρευμάτων μεταξύ τους.

Λόγω της συνθήκης *(a)* του Θεωρήματος 1, οι LCGs τείνουν να συμπεριφέρονται διαφορετικά για $c > 0$ (οπότε ονομάζονται *μικτές* LCGs) και διαφορετικά για $c = 0$ (οπότε ονομάζονται *πολλαπλασιαστικές* LCGs). Στις επόμενες δύο παραγράφους θα εξετάσουμε σε συντομία την επιλογή καταλλήλων παραμέτρων, ώστε να έχουμε καλές LCGs, για τις δύο αυτές κατηγορίες γεννητριών.

8.3 ΜΕΙΚΤΕΣ ΓΡΑΜΜΙΚΕΣ ΑΝΑΛΟΓΙΚΕΣ ΓΕΝΝΗΤΡΙΕΣ

Για $c > 0$, η συνθήκη *(a)* του Θεωρήματος 1, είναι πιθανόν να ικανοποιείται, οπότε έχουμε τη δυνατότητα να πάρουμε την πλήρη περίοδο *m*.

Κατ' αρχήν, όσον αφορά το *m*. Για να έχουμε μεγάλη περίοδο και υψηλή πυκνότητα των U_i στο [0,1], το *m* πρέπει να είναι μεγάλο. Ακόμα, η διαίρεση με *m* για να πάρουμε το υπόλοιπο από τη Σχέση (8.1), είναι μια σχετικά αργή αριθμητική λειτουργία στον υπολογιστή, οπότε θα ήταν επιθυμητό να αποφεύγαμε την πλήρη εκτέλεσή της. Μία επιλογή του *m* που ικανοποιεί τις παραπάνω προϋποθέσεις, είναι το $m = 2^b$, όπου *b* είναι ο αριθμός των bits της λέξης που χρησιμοποιεί το σύστημα στο οποίο θα υλοποιηθεί η προσομοίωση. Π.χ. για 32-bit υπολογιστή ή γλώσσα προγραμματισμού, όπου το πιο αριστερό bit χρησιμοποιείται για το πρόσημο, θα πάρουμε $b = 31$. Αν το *b* είναι αρκετά μεγάλο, ας πούμε $b \geq 31$, τότε $m \geq 2^{31} > 2.1 \cdot 10^9$. Επίσης, η επιλογή $m = 2^b$, μας επιτρέπει να αποφύγουμε την πλήρη διαίρεση με *m* στους περισσότερους υπολογιστές, εκμεταλλευόμενοι την *υπερχείλιση ακεραίου*. Ο μεγαλύτερος ακέραιος που μπορεί να αναπαρασταθεί είναι ο $2^b - 1$ και κάθε απόπειρα να αποθηκευθεί ένας μεγαλύτερος ακέραιος *W* (με, έστω, $h > b$ δυαδικά ψηφία), θα έχει σαν αποτέλεσμα την απώλεια των $h - b$ αριστερών (πιο σημαντικών) bits του ακεραίου. Τα υπόλοιπα *b* bits, θα έχουν τιμή ακριβώς $W \pmod{2^b}$.

Συνεπώς, το $m = 2^b$ φαίνεται να είναι μια καλή επιλογή για το *m*. Με την επιλογή αυτή, το Θεώρημα 3 δηλώνει ότι θα πάρουμε πλήρη περίοδο αν το c είναι περιττός και το $a - 1$ διαιρείται ακριβώς από το 4. Στην περίπτωση αυτή, το Z_0 μπορεί να είναι οποιοσδήποτε ακέραιος μεταξύ 0 και $m - 1$, χωρίς να υπάρχει επίπτωση στην περίοδο της γεννήτριας.

Σε σχέση τώρα με την επιλογή του πολλαπλασιαστή *a*. Υπήρχε παλιότερα η εκτίμηση, ότι είναι αποδοτικό να επηρεάζουμε τον πολλαπλασιασμό του Z_{i-1} με το *a*. Η εκτίμηση αυτή οδήγησε σε πολλαπλασιαστές της μορφής $a = 2^l + 1$, για κάποιο θετικό ακέραιο *l*. Τότε, $aZ_{i-1} = 2^l Z_{i-1} + Z_{i-1}$, έτσι ώστε το aZ_{i-1} μπορεί να δημιουργηθεί με τη μετατόπιση της δυαδικής αναπαράστασης του Z_{i-1} κατά l bits αριστερά και προσθέτοντας Z_{i-1}. Συνεπώς, ο πολλαπλασιασμός μπορεί να αντικατασταθεί από λειτουργίες μετατόπισης και πρόσθεσης. Όμως πιο πρόσφατες ερευνητικές εργασίες έχουν δείξει ότι πρέπει να αποφεύγονται πολλαπλασιαστές της παραπάνω μορφής, διότι δίνουν γεννήτριες με φτωχά στατιστικά χαρακτηριστικά.

Τότε, πώς θα έπρεπε να επιλεγούν τα *a* και *c*, όταν $m = 2^b$, ώστε να πάρουμε μία καλή μικτή LCG; Η πραγματική απάντηση είναι ότι πρέπει μάλλον να προτιμήσουμε μία πολλαπλασιαστική γεννήτρια (βλέπε επόμενη Παράγραφο), η οποία είναι απλούστερη, πιο κατανοητή, συμπεριφέρεται γενικά τόσο καλά όσο και μία μικτή LCG και για τους λόγους αυτούς χρησιμοποιείται πιο πολύ στην πράξη. Πάντως, δύο μικτές LCGs με $m = 2^b$, που έχει αποδειχθεί ότι συμπεριφέρονται καλά, είναι οι παρακάτω:

Για $b = 35$, παίρνουμε $a = 5^{15}$ και $c = 1$.

Για $b = 31$, παίρνουμε $a = 314{,}159{,}269$ και $c = 453{,}806{,}245$.

8.4 ΠΟΛΛΑΠΛΑΣΙΑΣΤΙΚΕΣ ΓΡΑΜΜΙΚΕΣ ΑΝΑΛΟΓΙΚΕΣ ΓΕΝΝΗΤΡΙΕΣ

Οι πολλαπλασιαστικές LCGs έχουν το πλεονέκτημα ότι δεν χρειάζεται η πρόσθεση του *c*, αλλά όμως δεν μπορούν να έχουν πλήρη περίοδο, αφού δεν είναι δυνατόν να ικανοποιηθεί η συνθήκη *(a)* του Θεωρήματος 1. Όμως, μπορούμε να πάρουμε περίοδο ίση με $m-1$, αν τα *m* και *a* επιλεγούν με προσοχή.

Όπως και στις μικτές γεννήτριες, είναι υπολογιστικά αποδοτικό να επιλέξουμε $m = 2^b$. Στην περίπτωση αυτή, έχει αποδειχθεί ότι η περίοδος είναι το πολύ 2^{b-2}, δηλαδή μπορούν να δημιουργηθούν μόνο το 1/4 των ακεραίων από 0 έως $m-1$, ως τιμές των Z_i. (Στην πραγματικότητα, η περίοδος είναι 2^{b-2}, αν το Z_0 είναι περιττός και το *a* είναι της μορφής $8k+3$ ή $8k+5$, για κάποιο $k = 0,1,\ldots$). Ακόμα, δεν θα μπορούμε γενικά να ξέρουμε πού θα πέσουν αυτοί οι $m/4$ ακέραιοι. Δηλαδή, μπορεί να υπάρχουν απαράδεκτα μεγάλα κενά στις τιμές των Z_i. Επίσης, αν επιλέξουμε *a* της μορφής $2^l + j$ (έτσι ώστε ο πολλαπλασιασμός του Z_{i-1} με το *a* να αντικαθίσταται από μια μετατόπιση και *j* προσθέσεις), θα έχουμε φτωχές στατιστικές ιδιότητες της γεννήτριας. Η γεννήτρια που είναι γνωστή με το όνομα RANDU και είναι αυτής της μορφής ($m = 2^{31}$, $a = 2^{16} + 3 = 65{,}539$, $c = 0$), έχει αποδειχθεί ότι έχει ανεπιθύμητες στατιστικές ιδιότητες και κατά συνέπεια πρέπει να αποφεύγεται η χρήση της.

Λόγω των δυσκολιών αυτών που σχετίζονται με την επιλογή $m = 2^b$ στις πολλαπλασιαστικές LCGs, έχει γίνει προσπάθεια να βρεθούν άλλοι τρόποι προσδιορισμού του *m*. Μία τέτοια μέθοδος προτείνει το *m* να είναι ο μεγαλύτερος πρώτος αριθμός που είναι μικρότερος από 2^b. Για παράδειγμα, στην περίπτωση $b = 31$, ο μεγαλύτερος πρώτος αριθμός μικρότερος από 2^{31} είναι ο $2^{31} - 1 = 2{,}147{,}483{,}647$. Όταν το *m* είναι πρώτος, μπορεί να αποδειχθεί ότι η περίοδος είναι $m-1$ αν το *a* είναι "primitive element modulo" *m*. Δηλαδή, ο μικρότερος ακέραιος *l* για τον οποίο το $a^l - 1$ διαιρείται ακριβώς από το *m*, είναι ο $l = m-1$. Με τα *m* και *a* επιλεγμένα με τον τρόπο αυτό, θα πάρουμε τους ακεραίους $1,2,\ldots,m-1$ ακριβώς μία φορά σε κάθε κύκλο, έτσι

ώστε το Z_0 μπορεί να είναι οποιοσδήποτε ακέραιος από 1 έως $m-1$, ενώ θα έχουμε πάντα περίοδο ίση με $m-1$. Οι γεννήτριες αυτές ονομάζονται *Prime Modulus Multiplicative LCGs (PMMLCGs).*

Δύο θέματα που αφορούν τις PMMLCGs είναι τα εξής: (1) Πώς μπορούμε να βρούμε ένα primitive element modulo *m*; Η διαδικασία αυτή είναι πολύπλοκη υπολογιστικά και δεν θα ασχοληθούμε μαζί της. Θα ξεπεράσουμε το σημείο αυτό, χρησιμοποιώντας μερικές PMMLCGs που έχουν δοκιμασθεί και έχει αποδειχθεί ότι συμπεριφέρονται καλά. Τις παρουσιάζουμε στο τέλος της παραγράφου. (2) Από τη στιγμή που δεν έχουμε επιλέξει $m=2^b$, δεν μπορούμε να χρησιμοποιήσουμε απευθείας το μηχανισμό υπερχείλισης για να επηρεάσουμε τη διαίρεση με το *m*. Μία τεχνική που έχει αναπτυχθεί για την αποφυγή της εξαντλητικής διαίρεσης στην περίπτωση αυτή, η οποία χρησιμοποιεί μία μορφή υπερχείλισης, ονομάζεται *προσομοιωμένη διαίρεση*. Η γεννήτρια της οποίας παρουσιάζουμε τον κώδικα στο Κεφάλαιο αυτό, χρησιμοποιεί την τεχνική της προσομοιωμένης διαίρεσης.

Ολοκληρώνουμε την παράγραφο αυτή, με παραδείγματα δοκιμασμένων PMMLCGs, οι οποίες έχουν διαιρέτη *m* ίσο με $m^*=2^{31}-1$ και πολλαπλασιαστές *a* οι οποίοι είναι primitive element modulo m^* και γνωρίζουμε ότι συμπεριφέρονται καλά. Η περίοδος των γεννητριών αυτών είναι $m^*-1=2^{31}-2$ και είναι αρκετά μεγάλη ώστε με ένα ρυθμό δημιουργίας 1,000 τυχαίων αριθμών ανά δευτερόλεπτο, να χρειάζεται χρόνος περισσότερος από 3 εβδομάδες για να ολοκληρωθεί ένας κύκλος.

Δύο συγκεκριμένες τιμές του πολλαπλασιαστή *a* που χρησιμοποιούνται ευρέως με το διαιρέτη m^*, είναι οι $a_1=7^5=16{,}807$ και $a_2=630{,}360{,}016$. Και οι δύο είναι primitive element modulo m^*. Έχει αποδειχθεί ότι το a_2 συμπεριφέρεται πολύ καλύτερα στατιστικά από το a_1 και κατά συνέπεια χρησιμοποιείται περισσότερο.

Η PMMLCG που παρουσιάζεται στη συνέχεια του Κεφαλαίου, έχει $m=m^*=2^{31}-1$ και $a=a_2=630{,}360{,}016$ και είναι μια γεννήτρια τυχαίων αριθμών που μπορεί να χρησιμοποιηθεί σε μεγάλα πειράματα προσομοίωσης.

8.5 Η ΥΛΟΠΟΙΗΣΗ ΜΙΑΣ ΓΕΝΝΗΤΡΙΑΣ ΤΥΧΑΙΩΝ ΑΡΙΘΜΩΝ

Στο Σχήμα 8.2 παρουσιάζεται μια PMMLCG με $m=m^*=2^{31}-1=2{,}147{,}483{,}647$ και $a=a_2=630{,}360{,}016$. Ο κώδικας προϋποθέτει ότι οι ακέραιοι μεταξύ $-m^*$ και m^* μπορούν να αναπαρασταθούν και να χρησιμοποιούνται σε υπολογισμούς χωρίς προβλήματα, στην υλοποίηση της γλώσσας C η οποία θα χρησιμοποιηθεί από τον προγραμματιστή. Αν δεν είναι διαθέσιμο αυτό το εύρος ακεραίων, θα πρέπει να τροποποιηθεί η γεννήτρια, ώστε να χρησιμοποιεί τον ίδιο διαιρέτη m^*, αλλά τον πολλαπλασιαστή $a=a_1=16{,}807$ και φυσικά να υλοποιείται με αριθμητική floating - point και όχι ακεραίων, γεγονός που θα την κάνει πιο αργή.

Η γεννήτρια του Σχήματος 30, υποστηρίζει πολλαπλά (100) ρεύματα τυχαίων αριθμών, με αρχικές τιμές που διαφέρουν τουλάχιστον κατά 100,000 μεταξύ τους. Η παράμετρος εισόδου "stream" πρέπει να είναι ένας int που θα δίνει τον επιθυμητό αριθμό του ρεύματος. Το αρχείο επικεφαλίδας rand.h του Σχήματος 31, πρέπει να περιληφθεί στο πρόγραμμα που καλεί τις συναρτήσεις του Σχήματος 30 (#include "rand.h"), πριν τη χρήση τους.

Ο τρόπος χρήσης των τριών συναρτήσεων της γεννήτριας, είναι:

Για να πάρουμε τον επόμενο $U(0,1)$ τυχαίο αριθμό από το ρεύμα "stream", εκτελούμε u = rand(stream); όπου rand είναι μια συνάρτηση float. Η float μεταβλητή u θα περιέχει τον επόμενο τυχαίο αριθμό.

Για να θέσουμε την αρχική τιμή του ρεύματος "stream" ίση με μια τιμή zset, διαφορετική από την default, εκτελούμε randst(zset, stream); όπου randst είναι μια void συνάρτηση και το zset πρέπει να είναι ένας long με τιμή ίση με την αρχική τιμή που θέλουμε, δηλαδή ένας αριθμός μεταξύ 1 και 2,147,483,646 (περιλαμβανομένου). Οι default τιμές και για τα 100 ρεύματα, δίνονται στον κώδικα.

Για να πάρουμε την τρέχουσα αρχική τιμή (και πιο πρόσφατα δημιουργημένο τυχαίο ακέραιο) του ρεύματος "stream" στην long μεταβλητή zget, εκτελούμε zget = randgt(stream); όπου randgt είναι μια long συνάρτηση.

```
/*  Define the constants.  */
#define MODLUS  2147483647
#define MULTI           24112
#define MULT2           26143

/*  Set the default seeds for all 100 streams.  */
static long zrng[] =
{   0,
 1973272912,  281629770,    20006270, 1280689831, 2096730329, 1933576050,
  913566091,  246780520, 1363774876,   604901985, 1511192140, 1259851944,
  824064364,  150493284,   242708531,    75253171, 1964472944, 1202299975,
  233217322, 1911216000,   726370533,   403498145,   993232223, 1103205531,
  762430696, 1922803170, 1385516923,    76271663,   413682397,   726466604,
  336157058, 1432650381, 1120463904,   595778810,   877722890, 1046574445,
   68911991, 2088367019,   748545416,   622401386, 2122378830,   640690903,
1774806513, 2132545692, 2079249579,    78130110,   852776735, 1187867272,
1351423507, 1645973084, 1997049139,   922510944, 2045512870,   898585771,
  243649545, 1004818771,   773686062,   403188473,   372279877, 1901633463,
  498067494, 2087759558,   493157915,   597104727, 1530940798, 1814496276,
  536444882, 1663153658,   855503735,    67784357, 1432404475,   619691088,
  119025595,   880802310,   176192644, 1116780070,   277854671, 1366580350,
1142483975,  2026948561, 1053920743,  786262391, 1792203830, 1494667770,
1923011392, 1433700034, 1244184613, 1147297105,   539712780, 1545929719,
  190641742, 1645390429,   264907697,   620389253, 1502074852,   927711160,
  364849192, 2049576050,   638580085,   547070247   } ;

/*  Generate the next random number.  */
float rand(int stream)
{
    long zi, lowprd, hi31;
          zi       = zrng[stream];
          lowprd = (zi & 65535) * MULTI;
          hi31    = (zi >> 16) * MULTI + (lowprd >> 16);
          zi       = ((lowprd & 65535) - MODLUS) +
                     ((hi31 & 32767) << 16) + (hi31 >> 15);
          if (zi < 0)  zi += MODLUS;
          lowprd = (zi & 65535) * MULT2;
          hi31    = (zi >> 16) * MULT2 + (lowprd >> 16);
          zi       = ((lowprd & 65535) - MODLUS) +
                     ((hi31 & 32767) << 16) + (hi31 >> 15);
          if (zi < 0)  zi += MODLUS;
          zrng[stream] = zi;
          return ((zi >> 7 | 1) + 1) / 16777216.0;
}

/*  Set the current zrng for stream “stream” to zset.  */
void randst (long zset, int stream)
{
          zrng[stream] = zset;
}

/*  Return the current zrng for stream “stream”.  */
long randgt (int stream)
{
          return zrng[stream];
}
```

Σχήμα 8.0.2. Ο Κώδικας C για την PMMLCG με $m = 2^{31} - 1$ και $a = 630{,}360{,}016$.

Οι 3 δηλώσεις που φαίνονται στο Σχήμα 8.3, χρησιμοποιούνται από τη γεννήτρια τυχαίων αριθμών rand και τις συναρτήσεις randst και randgt του Σχήματος 30. Ο κώδικας του Σχήματος 8.3 αποτελεί το αρχείο rand.h που πρέπει να περιλαμβάνεται σε κάθε πρόγραμμα που χρησιμοποιεί τις τρεις αυτές συναρτήσεις, με την εκτέλεση της εντολής #include "rand.h" πριν αναφερθούν οι συναρτήσεις.

```
/*  The header file rand.h  */

float rand(int stream);
void randst(long zset, int stream);
long randgt(int stream);
```

Σχήμα 8.0.3. Το Αρχείο Επικεφαλίδας rand.h

ΚΕΦΑΛΑΙΟ 9
ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΤΙΜΩΝ ΑΠΟ ΠΙΘΑΝΟΤΙΚΕΣ ΚΑΤΑΝΟΜΕΣ

9.1 ΤΕΧΝΙΚΕΣ ΔΗΜΙΟΥΡΓΙΑΣ ΤΥΧΑΙΩΝ ΤΙΜΩΝ

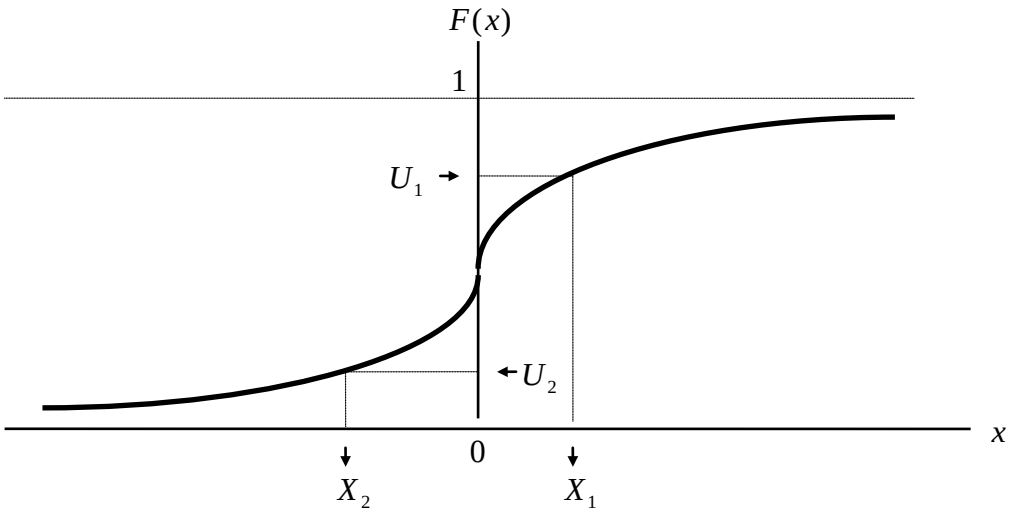
Στο Κεφάλαιο αυτό θεωρούμε ότι έχουμε στη διάθεσή μας την κατανομή από την οποία θέλουμε να πάρουμε τυχαίες τιμές (μαζί με τις τιμές όλων των παραμέτρων της), καθώς και μια καλή γεννήτρια τυχαίων αριθμών $U(0,1)$, σύμφωνα με τα αναφερθέντα στο προηγούμενο Κεφάλαιο.

9.1.1 Η Μέθοδος του Αντίστροφου Μετασχηματισμού

Έστω ότι θέλουμε να δημιουργήσουμε μια *τυχαία τιμή X* από μία τυχαία μεταβλητή, η οποία είναι συνεχής και έχει συνάρτηση κατανομής *F* συνεχή, αυστηρά αύξουσα όταν $0 < F(x) < 1$. [Αυτό σημαίνει ότι αν $x_1 < x_2$ και $0 < F(x_1) \le F(x_2) < 1$, τότε $F(x_1) < F(x_2)$.] Έστω F^{-1} η αντίστροφη συνάρτηση της *F*. Τότε, ένας αλγόριθμος δημιουργίας της τυχαίας τιμής *X* είναι ο εξής (το σύμβολο ~ διαβάζεται "κατανεμημένο σύμφωνα με την"):

1. Δημιούργησε ένα $U \sim U(0,1)$.
2. Επέστρεψε το $X = F^{-1}(U)$.

Σημειώστε ότι το $F^{-1}(U)$ θα ορίζεται πάντα, αφού $0 \le U \le 1$ και η *F* έχει τιμές στο [0,1]. Το Σχήμα 9.1 παρουσιάζει τον αλγόριθμο γραφικά.



Σχήμα 9.0.1. Η Μέθοδος του Αντίστροφου Μετασχηματισμού για συνεχείς τυχαίες μεταβλητές.

Στο παράδειγμα του Σχήματος 9.1, η τυχαία μεταβλητή μπορεί να πάρει και θετικές και αρνητικές τιμές. Η συγκεκριμένη τιμή του U καθορίζει αν η τυχαία τιμή που θα πάρουμε, θα είναι θετική ή αρνητική. Εδώ, ο τυχαίος αριθμός U_1 θα δώσει τη θετική τυχαία τιμή X_1, ενώ ο τυχαίος αριθμός U_2 θα δώσει την αρνητική τυχαία τιμή X_2.

Για να δείξουμε ότι η τιμή X που μας επιστρέφεται από τον παραπάνω αλγόριθμο, έχει ως συνάρτηση κατανομής την F, πρέπει να αποδείξουμε ότι για κάθε πραγματικό αριθμό x, $P(X \le x) = F(x)$. Αφού η F είναι αντι-στρέψιμη, θα έχουμε:

$$P(X \le x) = P(F^{-1}(U) \le x) = P(U \le F(x)) = F(x)$$

όπου η τελευταία ισότητα ισχύει διότι $U \sim U(0,1)$ και $0 \le F(x) \le 1$.

Παράδειγμα 9.1

Έστω ότι η X ακολουθεί την εκθετική κατανομή με μέση τιμή β. Η συνάρτηση κατανομής είναι

$$F(x) = \begin{cases} 1 - e^{-x/\beta} & \alpha\nu \ \ x \ge 0 \\ 0 & \alpha\lambda\lambda\iota\omega\varsigma \end{cases}$$

ώστε για να βρούμε το F^{-1}, θέτουμε $u = F(x)$ και λύνουμε ως προς x για να πάρουμε

$$F^{-1}(u) = -\beta \ln(1-u)$$

Συνεπώς, για να δημιουργήσουμε την επιθυμητή τυχαία τιμή, πρώτα δημιουργούμε έναν τυχαίο αριθμό $U \sim U(0,1)$ και στη συνέχεια παίρνουμε $X = -\beta \ln U$. Δηλαδή, αντικαθιστούμε το 1-U με το U, αφού και οι δύο ποσότητες ακολουθούν την ίδια $U(0,1)$ κατανομή. Πάντως, δεν μπορούμε σε όλες τις περιπτώσεις να κάνουμε την αλλαγή αυτή, διότι μετατρέπεται από θετική σε αρνητική η συσχέτιση των X με τα U, γεγονός που δεν είναι πάντα επιτρεπτό.

Η μέθοδος του αντίστροφου μετασχηματισμού, μπορεί επίσης να χρησιμοποιηθεί και όταν το X είναι διακριτό. Τώρα, η συνάρτηση κατανομής είναι

$$F(x) = P(X \le x) = \sum_{x_i \le x} p(x_i)$$

όπου $p(x_i)$ είναι η συνάρτηση μάζας πιθανότητας $p(x_i) = P(X = x_i)$. Υποθέτουμε ότι το X μπορεί να πάρει μόνο τις τιμές $x_1, x_2, \ldots$ όπου $x_1 < x_2 < \cdots$ Ο αλγόριθμος είναι τότε:

1. Δημιούργησε ένα $U \sim U(0,1)$.

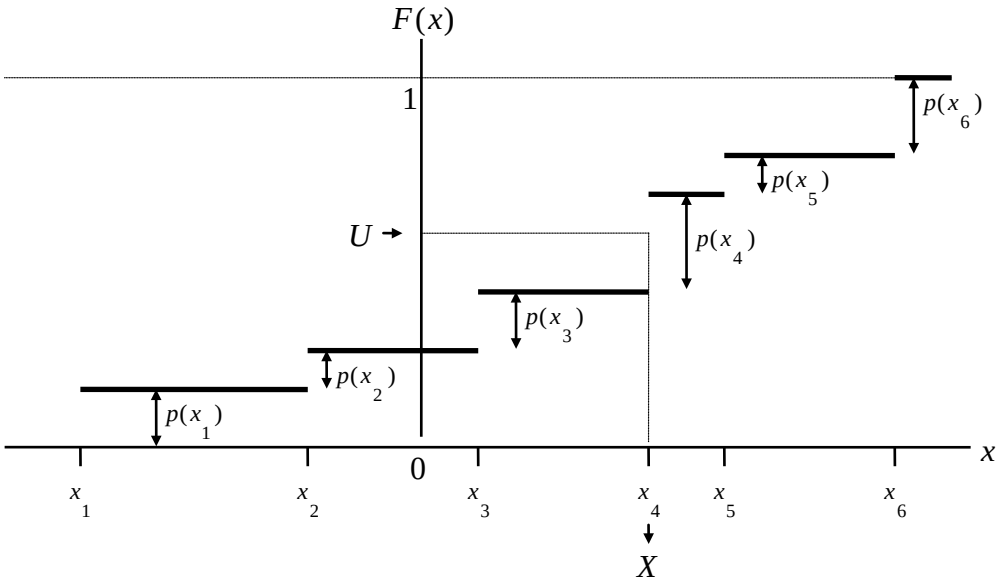
2. Βρες τον μικρότερο θετικό ακέραιο I τέτοιον ώστε $U \le F(x_I)$, και επέστρεψε το $X = x_I$.

Στο Σχήμα 33 φαίνεται ένα παράδειγμα της μεθόδου, στο οποίο δημιουργούμε το $X = x_4$. Αν και ο παραπάνω αλγόριθμος δεν φαίνεται να έχει σχέση με τον αντίστοιχο αλγόριθμο για συνεχείς τυχαίες τιμές, οι ομοιότητες των Σχημάτων 32 και 33 είναι εμφανείς.

Για να αποδείξουμε ότι ο αλγόριθμος είναι σωστός, πρέπει να δείξουμε ότι $P(X = x_i) = p(x_i)$ για όλα τα i. Για $i = 1$, θα έχουμε $X = x_1$ αν και μόνο αν $U \le F(x_1) = p(x_1)$, αφού έχουμε ταξινομήσει τα x_i κατά αύξουσα σειρά. Αφού

$U \sim U(0,1)$, $P(X = x_1) = p(x_1)$, όπως επιθυμούμε. Για $i \geq 2$, ο αλγόριθμος θέτει $X = x_i$ αν και μόνο αν $F(x_{i-1}) < U \leq F(x_i)$, διότι το i που έχει επιλέξει ο αλγόριθμος, είναι ο μικρότερος θετικός ακέραιος τέτοιος ώστε $U \leq F(x_i)$. Ακόμα, αφού $U \sim U(0,1)$ και $0 \leq F(x_{i-1}) < F(x_i) \leq 1$, ισχύει:

$$P(X = x_i) = P[F(x_{i-1}) < U \leq F(x_i)] = F(x_i) - F(x_{i-1}) = p(x_i)$$



Σχήμα 9.2. Η Μέθοδος του Αντίστροφου Μετασχηματισμού για διακριτές τυχαίες μεταβλητές.

Αν και στο Σχήμα 9.2 φαίνεται η μέθοδος να χρησιμοποιείται για τυχαίες μεταβλητές που μπορούν να πάρουν μόνο πεπερασμένο αριθμό τιμών, ο αντίστροφος μετασχηματισμός μπορεί να χρησιμοποιηθεί κανονικά και για τη δημιουργία τυχαίων τιμών από κατανομές τυχαίων μεταβλητών με άπειρο πεδίο τιμών, όπως η Poisson, η γεωμετρική, η αρνητική δυωνυμική κ.α.

9.1.2 Τεχνική της Σύνθεσης

Η τεχνική της *σύνθεσης* (composition) χρησιμοποιείται όταν η συνάρτηση κατανομής F από την οποία θέλουμε να πάρουμε τυχαίες τιμές, μπορεί να εκφραστεί ως ένας κυρτός συνδυασμός άλλων συναρτήσεων κατανομής $F_1, F_2, \ldots$ Η τεχνική έχει νόημα όταν μπορούμε να πάρουμε δεδομένα από τις F_i πιο εύκολα από ότι θα παίρναμε από τη αρχική F.

Συγκεκριμένα, υποθέτουμε ότι για όλα τα x, η $F(x)$ μπορεί να γραφεί ως:

$$F(x) = \sum_{j=1}^{\infty} p_j F_j(x)$$

όπου $p_j \geq 0$, $\Sigma_{j=1}^{\infty} p_j = 1$ και κάθε F_j είναι μια συνάρτηση κατανομής. Ισοδύναμα, αν το X έχει συνάρτηση πυκνότητας f η οποία μπορεί να γραφεί

$$f(x) = \sum_{j=1}^{\infty} p_j f_j(x)$$

όπου τα f_j είναι άλλες συναρτήσεις πυκνότητας, η μέθοδος της σύνθεσης μπορεί επίσης να εφαρμοσθεί. Η περίπτωση διακριτών τυχαίων μεταβλητών αντιμετωπίζεται ανάλογα. Ο γενικός αλγόριθμος της τεχνικής της σύνθεσης, είναι ο ακόλουθος:

1. Δημιούργησε έναν θετικό τυχαίο ακέραιο J τέτοιον ώστε
$P(J = j) = p_j$ για $j = 1,2,\ldots$

2. Επέστρεψε ένα X με συνάρτηση κατανομής F_J .

Το Βήμα 1 μπορεί να θεωρηθεί ως η επιλογή της συνάρτησης κατανομής F_j με πιθανότητα p_j και μπορεί για παράδειγμα, να γίνει μέσω της διακριτής μεθόδου του αντίστροφου μετασχηματισμού. Δεδομένου του $J = j$, η δημιουργία του X στο Βήμα 2, θα πρέπει να γίνεται ανεξάρτητα του J. Μπορούμε εύκολα να δείξουμε ότι το X που επιστρέφει ο αλγόριθμος, θα έχει συνάρτηση κατανομής F:

$$P(X \le x) = \sum_{j=1}^{\infty} P(X \le x \mid J = j)P(J = j) = \sum_{j=1}^{\infty} F_j(x)p_j = F(x)$$

9.1.3 Η Μέθοδος της Συνέλιξης

Έστω ότι η τυχαία μεταβλητή X μπορεί να εκφραστεί ως το άθροισμα m άλλων, *ανεξαρτήτων, όμοια κατανεμημένων* τυχαίων μεταβλητών, από τις οποίες μπορούμε πιο εύκολα να δημιουργήσουμε τυχαίες τιμές. Δηλαδή, ισχύει:

$$X = Y_1 + Y_2 + \cdots + Y_m$$

Αν F είναι η συνάρτηση κατανομής της X και G είναι η συνάρτηση κατανομής των Y_j, ο αλγόριθμος δημιουργίας τυχαίων τιμών από την X, είναι:

1. Δημιούργησε ανεξάρτητες τυχαίες τιμές $Y_1, Y_2, \ldots, Y_m$ από τη συνάρτηση κατανομής G.

2. Επέστρεψε το $X = Y_1 + Y_2 + \cdots + Y_m$.

Η ισχύς του αλγορίθμου αποδεικνύεται αν έχουμε πάντα υπ' όψη ότι το X και το $Y_1 + Y_2 + \cdots + Y_m$ έχουν την ίδια συνάρτηση κατανομής F. Τότε:

$$P(X \le x) = P(Y_1 + Y_2 + \cdots + Y_m \le x) = F(x)$$

9.1.4 Μέθοδος της Αποδοχής - Απόρριψης

Οι τρεις προηγούμενες μέθοδοι δημιουργίας τυχαίων τιμών, μπορούν να χαρακτηρισθούν ως *άμεσες*, με την έννοια ότι χρησιμοποιούν απευθείας την κατανομή της τυχαίας μεταβλητής. Η *μέθοδος της αποδοχής - απόρριψης* (acceptance - rejection method) είναι λιγότερο άμεση και μπορεί να αποδειχθεί χρήσιμη όταν οι άμεσες μέθοδοι αποδεικνύονται αναποτελεσματικές. Θα ασχοληθούμε με την περίπτωση συνεχών τυχαίων μεταβλητών, αφού η περίπτωση των διακριτών αντιμετωπίζεται ανάλογα.

Η μέθοδος απαιτεί τον προσδιορισμό μιας συνάρτησης t η οποία είναι πάντα μεγαλύτερη από τη *συνάρτηση πυκνότητας* f της τυχαίας μεταβλητής που μας ενδιαφέρει.

Δηλαδή, $t(x) \geq f(x)$ για όλα τα *x*. Η συνάρτηση *t* δεν μπορεί να είναι γενικά, συνάρτηση πυκνότητας, διότι:

$$c = \int_{-\infty}^{\infty} t(x)\,dx \geq \int_{-\infty}^{\infty} f(x)\,dx = 1$$

όμως, η συνάρτηση $r(x) = t(x)/c$ είναι μια συνάρτηση πυκνότητας. (Υποθέτουμε ότι το *t* είναι τέτοιο ώστε $c < \infty$). Ελπίζοντας ότι θα μπορέσουμε να δημιουργήσουμε εύκολα και γρήγορα μία τυχαία τιμή *Y* με συνάρτηση πυκνότητας *r*, ο αλγόριθμος είναι:

1. Δημιούργησε ένα *Y* με συνάρτηση πυκνότητας *r*.
2. Δημιούργησε ένα $U \sim U(0,1)$ ανεξάρτητο από το *Y*.
3. Αν $U \leq f(Y)/t(Y)$, επέστρεψε το $X = Y$. Αλλιώς, πήγαινε πίσω στο 1 και προσπάθησε πάλι.

Ο αλγόριθμος επαναλαμβάνει το βρόχο από το Βήμα 1 συνεχώς, μέχρι να δημιουργηθεί ένα ζευγάρι (Y,U) στα βήματα 1 και 2, για το οποίο $U \leq f(Y)/t(Y)$, οπότε αποδεχόμαστε την τιμή *Y* για το *X*. Η απόδειξη της ισχύος του αλγορίθμου παραλείπεται, διότι είναι πολύπλοκη και δεν προσφέρει κάτι στην κατανόηση της μεθόδου.

9.1.5 Αξιοποίηση των Ιδιαιτεροτήτων της Κατανομής

Αν και οι περισσότερες μέθοδοι δημιουργίας τυχαίων τιμών μπορούν να ενταχθούν σε κάποια από τις τεχνικές που εξετάσαμε μέχρι τώρα, μερικές μέθοδοι απλώς αξιοποιούν κάποια ιδιότητα της συνάρτησης κατανομής που μας ενδιαφέρει. Συχνά, η ιδιότητα αναφέρεται στη δυνατότητα αναπαρά-στασης της τυχαίας μεταβλητής *X* συναρτήσει άλλων τυχαίων μεταβλητών που μπορούν να υπολογιστούν ευκολότερα. Επειδή, βέβαια, δεν υπάρχει γενική μέθοδος της τεχνικής, μπορούμε να τη δούμε μόνο με παραδείγματα.

Παράδειγμα 9.2

Αν $Y \sim N(0,1)$ (standard κανονική κατανομή), τότε το Y^2 ακολουθεί την χ^2 κατανομή με 1 df. [Γράφουμε $X \sim \chi^2_k$ εννοώντας ότι το *X* ακολουθεί την χ^2 κατανομή με *k* df, η οποία είναι η κατανομή gamma(*k*/2,2)]. Συνεπώς, προκειμένου να δημιουργήσουμε ένα $X \sim \chi^2_1$, δημιουργούμε ένα $Y \sim N(0,1)$ και επιστρέφουμε $X = Y^2$.

9.2 ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΤΙΜΩΝ ΑΠΟ ΣΥΝΕΧΕΙΣ ΚΑΤΑΝΟΜΕΣ

Στην Παράγραφο αυτή, παρουσιάζεται για κάθε συνεχή κατανομή, ένας αλγόριθμος δημιουργίας τυχαίων τιμών. Έχει επιλεγεί για κάθε κατανομή, η πιο εύκολη, ακριβής και αποδοτική μέθοδος από τις περιγραφόμενες στην προηγούμενη παράγραφο, χωρίς αυτό να σημαίνει ότι δεν μπορούν να εφαρμοσθούν άλλες μέθοδοι. Στην επόμενη Παράγραφο, ακολουθείται η ίδια μεθοδολογία για διακριτές κατανομές. Η περιγραφή κάθε κατανομής, βρίσκεται στο ΠΑΡΑΡΤΗΜΑ Α.

9.2.1 Η Ομοιόμορφη κατανομή

Η συνάρτηση κατανομής μιας $U(a,b)$ τυχαίας μεταβλητής, μπορεί να αντιστραφεί εύκολα, επιλύοντας την $u = F(x)$ για x, ώστε να πάρουμε για $0 \le u \le 1$:

$$x = F^{-1}(u) = a + (b-a)\,u$$

Συνεπώς, με χρήση της μεθόδου του αντίστροφου μετασχηματισμού, ο αλγόριθμος δημιουργίας μιας τυχαίας τιμής X, είναι:

1. Δημιούργησε ένα $U \sim U(0,1)$.
2. Επέστρεψε το $X = a + (b-a)U$.

9.2.2 Η Εκθετική κατανομή

Την εκθετική κατανομή με μέση τιμή $\beta > 0$, μελετήσαμε στο Παράδειγμα 9.1, όπου βρήκαμε τον ακόλουθο αλγόριθμο αντίστροφου μετασχηματισμού:

1. Δημιούργησε ένα $U \sim U(0,1)$.
2. Επέστρεψε το $X = -\beta \ln U$.

Θυμηθείτε, ότι κανονικά πρέπει να έχουμε το $1-U$ αντί για U στο Βήμα 2, ώστε να έχουμε θετική συσχέτιση των X με τα U.

9.2.3 Εμπειρικές κατανομές

Στην Παράγραφο αυτή θα δούμε αλγορίθμους για τη δημιουργία υχαίων τιμών από τις συνεχείς συναρτήσεις εμπειρικών κατανομών F και G που ορίσθηκαν στην Παράγραφο 7.3. Και στις δύο περιπτώσεις, χρησιμοποιείται η μέθοδος του αντίστροφου μετασχηματισμού.

Πρώτα θα εξετάσουμε την περίπτωση στην οποία έχουμε τις αυθεντικές παρατηρήσεις, με βάση τις οποίες ορίσαμε την εμπειρική συνάρτηση κατανομής $F(x)$. Αν και ένας αλγόριθμος αντίστροφου μετασχηματισμού φαίνεται ότι θα χρειαζόταν να χρησιμοποιήσει κάποιου είδους αναζήτηση, το γεγονός ότι οι "γωνίες" της F βρίσκονται ακριβώς στα επίπεδα 0, 1/(n-1), 2/(n-1), . . . , (n-2)/(n-1) και 1, μας επιτρέπει να αποφύγουμε την εξαντλητική αναζήτηση. Ο αλγόριθμος αντίστροφου μετασχηματισμού είναι ο εξής:

1. Δημιούργησε ένα $U \sim U(0,1)$. Θέσε $P = (n-1)U$ και $I = \lfloor P \rfloor + 1$.
2. Επέστρεψε το $X = X_{(I)} + (P - I + 1)(X_{(I+1)} - X_{(I)})$.

Όσον αφορά την υπολογιστική υλοποίηση του αλγορίθμου, μπορούμε να παρατηρήσουμε ότι πρέπει να έχουμε αποθηκευμένα τα $X_{(i)}$, ενώ αν αποθηκεύουμε σε ένα ξεχωριστό array τις τιμές των $X_{(I+1)} - X_{(I)}$, αποφεύγουμε μία αφαίρεση στο Βήμα 2. Παρατηρούμε επίσης, ότι οι τιμές που δημιουργούμε είναι πάντα μεταξύ $X_{(1)}$ και $X_{(n)}$.

Η αποφυγή της αναζήτησης στον αλγόριθμο, έχει σαν αποτέλεσμα ένα χρόνο εκτέλεσης ουσιαστικά ανεξάρτητο του n, αν και μεγάλο n απαιτεί μεγαλύτερο αποθηκευτικό χώρο και χρόνο αρχικοποίησης για την ταξινόμηση των X_i.

Έστω, τώρα ότι έχουμε ομαδοποιημένα δεδομένα, σύμφωνα με την περιγραφή της Παραγράφου 7.3. Ο παρακάτω αλγόριθμος αντίστροφου μετασχηματισμού, επιστρέφει μία τυχαία τιμή από την αντίστοιχη συνάρτηση κατανομής $G(x)$:

1. Δημιούργησε ένα $U \sim U(0,1)$.

2. Βρες τον ακέραιο J, $(0 \le J \le k-1)$, τέτοιον ώστε $G(a_J) \le U < G(a_{J+1})$ και επέστρεψε το $X = a_J + (a_{J+1} - a_J)[U - G(a_J)] / [G(a_{J+1}) - G(a_J)]$.

Παρατηρήστε ότι το J που βρίσκετε στο Βήμα 2, ικανοποιεί τη σχέση $G(a_J) < G(a_{J+1})$, οπότε δεν μπορεί να δημιουργηθεί κανένα X σε διάστημα για το οποίο $n_j = 0$. Επίσης, ισχύει $a_0 \le X \le a_k$. Ο προσδιορισμός του J στο Βήμα 2, μπορεί να γίνει με μία απ' ευθείας αναζήτηση από αριστερά προς τα δεξιά, ή με μια αναζήτηση που ξεκινάει με την τιμή του j για την οποία το $G(a_{j+1}) - G(a_j)$ είναι μέγιστο, μετά για το αμέσως μικρότερο κ.ο.κ.

9.3 ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΩΝ ΤΙΜΩΝ ΑΠΟ ΔΙΑΚΡΙΤΕΣ ΚΑΤΑΝΟΜΕΣ

9.3.1 Η κατανομή Bernoulli

Ο αλγόριθμος που ακολουθεί είναι ισοδύναμος με τη μέθοδο αντίστροφου μετασχηματισμού, αν αντιστραφούν οι ρόλοι των U και $U-1$.

1. Δημιούργησε ένα $U \sim U(0,1)$.

2. Αν $U \le p$, επέστρεψε το $X = 1$. Αλλιώς επέστρεψε το $X = 0$.

9.3.2 Η διακριτή Ομοιόμορφη κατανομή

Ο αλγόριθμος εδώ, είναι ακριβώς η μέθοδος αντίστροφου μετασχηματισμού:

1. Δημιούργησε ένα $U \sim U(0,1)$.

2. Επέστρεψε το $X = i + \lfloor (j - i + 1)U \rfloor$.

9.3.3 Η Δυωνυμική κατανομή

Για να δημιουργήσουμε μία τυχαία τιμή $X \sim \text{bin}(t, p)$ [δυωνυμική κατανομή], σημειώνουμε ότι το άθροισμα t ανεξαρτήτων τυχαίων μεταβλητών, όμοια κατανεμημένων σύμφωνα με την κατανομή Bernoulli(p), ακολουθεί την $\text{bin}(t, p)$ κατανομή. Η σχέση αυτή μας οδηγεί στον ακόλουθο αλγόριθμο συνέλιξης:

1. Δημιούργησε $Y_1, Y_2, \ldots, Y_t$ ως ανεξάρτητες, όμοια κατανεμημένες Bernoulli(p) τυχαίες τιμές.

2. Επέστρεψε το $X = Y_1 + Y_2 + \cdots + Y_t$.

9.3.4 Η Γεωμετρική κατανομή

Ο αλγόριθμος που ακολουθεί είναι ισοδύναμος με τη μέθοδο αντίστροφου μετασχηματισμού, αν αντικατασταθεί το U με το $U - 1$ στο Βήμα 2:

1. Δημιούργησε ένα $U \sim U(0,1)$.

2. Επέστρεψε το $X = \lfloor \ln U / \ln(1-p) \rfloor$.

9.3.5 Η κατανομή Poisson

Ο αλγόριθμος που ακολουθεί για τη δημιουργία τυχαίων τιμών από την κατανομή Poisson(λ), βασίζεται στη γνωστή σχέση μεταξύ των κατανομών Poisson(λ) και expo($1/\lambda$) [εκθετική κατανομή]:

1. Θέσε $a = e^{-\lambda}$, $b = 1$, και $i = 0$.

2. Δημιούργησε ένα $U_{i+1} \sim U(0,1)$ και αντικατέστησε το b με το bU_{i+1}. Αν $b < a$, επέστρεψε το $X = i$ και τερμάτισε. Αλλιώς πήγαινε στο Βήμα 3.

3. Αντικατέστησε το i με $i+1$ και πήγαινε πίσω στο Βήμα 2.

Η ορθότητα του αλγορίθμου αποδεικνύεται, παρατηρώντας ότι $X = i$, αν και μόνο αν:

$$\sum_{j=1}^{i} Y_j \le 1 < \sum_{j=1}^{i+1} Y_j$$

όπου $Y_j = (-1/\lambda) \ln U_j \sim \text{expo}(1/\lambda)$ και τα Y_j είναι ανεξάρτητα μεταξύ τους. Δηλαδή, $X = \max\left\{i : \sum_{j=1}^{i} Y_j \le 1\right\}$, έτσι ώστε η $X \sim \text{Poisson}(\lambda)$ [βλέπε το πρώτο σχόλιο στην περιγραφή της Poisson στο ΠΑΡΑΡΤΗΜΑ Α].

9.4 ΔΗΜΙΟΥΡΓΙΑ ΤΥΧΑΙΝ ΤΙΜΩΝ ΑΠΟ ΣΤΟΧΑΣΤΙΚΕΣ ΔΙΑΔΙΚΑΣΙΕΣ

Στην Παράγραφο αυτή θα δούμε τρόπους δημιουργίας των χρονικών στιγμών εμφάνισης γεγονότων (π.χ. αφίξεις) $t_1, t_2, \ldots$ για ορισμένες στοχαστικές διαδικασίες.

9.4.1 Η περίπτωση της διαδικασίας Poisson και γενίκευσή της

Η στάσιμη (stationary) διαδικασία Poisson με μέσο ρυθμό $\lambda > 0$, έχει την ιδιότητα οι χρόνοι μεταξύ διαδοχικών γεγονότων (π.χ. μεταξύ διαδοχικών αφίξεων) $A_i = t_i - t_{i-1}$, $i = 1,2,\ldots$ να είναι ανεξάρτητες, όμοια *εκθετικά* κατανεμημένες τυχαίες μεταβλητές, με ίδια μέση τιμή $1/\lambda$. Κατά συνέπεια, μπορούμε να δημιουργήσουμε τα t_i με επαναληπτικό τρόπο, όπως φαίνεται στον παρακάτω αλγόριθμο, στον οποίο

υποθέτουμε ότι το t_{i-1} έχει υπολογισθεί και θέλουμε να δημιουργήσουμε την επόμενη χρονική στιγμή εμφάνισης γεγονότος, t_i :

1. Δημιούργησε ένα $U \sim U(0,1)$ ανεξάρτητο από τις προηγούμενες τυχαίες τιμές.
2. Επέστρεψε το $t_i = t_{i-1} - (1/\lambda)\ln U$.

Η επανάληψη αρχίζει με τον υπολογισμό του t_1 και με την υπόθεση ότι $t_0 = 0$.

Ο αλγόριθμος αυτός μπορεί εύκολα να μετατραπεί, έτσι ώστε να δημιουργούνται τυχαίες τιμές από οποιαδήποτε στοχαστική διαδικασία, της οποίας οι χρόνοι μεταξύ διαδοχικών γεγονότων, είναι ανεξάρτητες, όμοια κατανεμημένες (σύμφωνα με οποιαδήποτε κατανομή) τυχαίες μεταβλητές. Απλώς, κατά το Βήμα 2 θα προστίθεται ένας ανεξάρτητα δημιουργημένος χρόνος μεταξύ διαδοχικών γεγονότων, στο t_{i-1} προκειμένου να πάρουμε το t_i . Στον παραπάνω αλγόριθμο, ο χρόνος αυτός είναι $-(1/\lambda)\ln U$ [εκθετική κατανομή].

9.4.2 Στοχαστικές Διαδικασίες με Ομαδικές εμφανίσεις γεγονότων

Έστω μια στοχαστική διαδικασία, π.χ. αφίξεων, στην οποία η *i*-στή ομάδα (batch) πελατών φθάνει τη χρονική στιγμή t_i και ο αριθμός των πελατών της ομάδας είναι μια διακριτή τυχαία μεταβλητή B_i . Υποθέτουμε ότι τα B_i είναι ανεξάρτητες, όμοια κατανεμημένες τυχαίες μεταβλητές, ενώ είναι και ανεξάρτητα από τα t_i . Ένας γενικός επαναληπτικός αλγόριθμος για τη δημιουργία της στοχαστικής διαδικασίας (t_i και B_i), είναι ο ακόλουθος:

1. Δημιούργησε την επόμενη χρονική στιγμή άφιξης t_i (π.χ. με τον αλγόριθμο της προηγούμενης παραγράφου 6.4.1.).
2. Δημιούργησε την διακριτή τυχαία τιμή B_i ανεξάρτητη από τα προηγούμενα B_i και από τα $t_1, t_2, \ldots, t_i$.
3. Επέστρεψε το t_i και το B_i .

ΚΕΦΑΛΑΙΟ 10
ΑΝΑΛΥΣΗ ΤΩΝ ΑΠΟΤΕΛΕΣΜΑΤΩΝ ΤΗΣ ΠΡΟΣΟΜΟΙΩΣΗΣ

10.1 ΕΙΣΑΓΩΓΗ

Όπως έχει αναφερθεί, τα αποτελέσματα μιας εκτέλεσης ενός προσομοιωτή, πρέπει να αντιμετωπίζονται απλά ως μία συγκεκριμένη εμφάνιση τιμών των τυχαίων μεταβλητών που αντιπροσωπεύουν, με πιθανή μεγάλη διακύμανση και που διαφέρουν ίσως αρκετά από τα πραγματικά χαρακτηριστικά των τυχαίων μεταβλητών. Ένα άλλο πρόβλημα με τα αποτελέσματα μιας προσομοίωσης, είναι το γεγονός ότι, συνήθως δεν αποτελούν ένα σύνολο ανεξαρτήτων τιμών, αλλά είναι μή - στάσιμα και αυτο - συσχετισμένα μεγέθη, ώστε να δυσκολεύ-εται η εφαρμογή κλασικών στατιστικών τεχνικών που προϋποθέτουν ανεξαρτησία των μεγεθών.

Ας δούμε πιο αναλυτικά την πιθανοτική - στατιστική φύση των αποτελεσμά-των μιας προσομοίωσης. Έστω $Y_1, Y_2, \ldots$ μία στοχαστική διαδικασία που αντιπροσωπεύει ένα μέτρο απόδοσης εξόδου *μίας* εκτέλεσης ενός προσομοιωτή. Για παράδειγμα, το Y_i θα μπορούσε να είναι το μέσο μέγεθος μιας ουράς σε ένα σύστημα αναμονής, κατά την *i*-στή ώρα λειτουργίας της. Τα Y_i είναι τυχαίες μεταβλητές οι οποίες γενικά, δεν είναι ούτε ανεξάρτητες μεταξύ τους, ούτε όμοια κατανεμημένες.

Έστω $y_{11}, y_{12}, \ldots, y_{1m}$ ένα σύνολο εμφανίσεων τιμών των τυχαίων μεταβλητών $Y_1, Y_2, \ldots, Y_m$ ως αποτέλεσμα μιας εκτέλεσης του προσομοιωτή μήκους *m* παρατηρήσεων, με τη χρήση των τυχαίων αριθμών $u_{11}, u_{12}, \ldots$ (Ο *i*-στός τυχαίος αριθμός που χρησιμοποιείται στην *j*-στή εκτέλεση του προσομοιωτή, συμβολίζεται u_{ji}). Αν εκτελέσουμε τον προσομοιωτή με ένα διαφορετικό σύνολο τυχαίων τιμών $u_{21}, u_{22}, \ldots$ τότε θα πάρουμε ένα διαφορετικό σύνολο εμφανίσεων τιμών $y_{21}, y_{22}, \ldots, y_{2m}$ των τυχαίων μεταβλητών $Y_1, Y_2, \ldots, Y_m$. Τα δύο σύνολα εμφανίσεων τιμών δεν είναι ίδια, αφού τα διαφορετικά σύνολα τυχαίων αριθμών που χρησιμοποιούνται στις δύο εκτελέσεις, παράγουν διαφορετικές τυχαίες τιμές των πιθανοτικών κατανομών εισόδου του προσομοιωτή. Ας υποθέσουμε ότι γενικά κάνουμε *n* ανεξάρτητες εκτελέσεις μήκους *m* παρατηρήσεων και παίρνουμε τα παρακάτω αποτελέσματα:

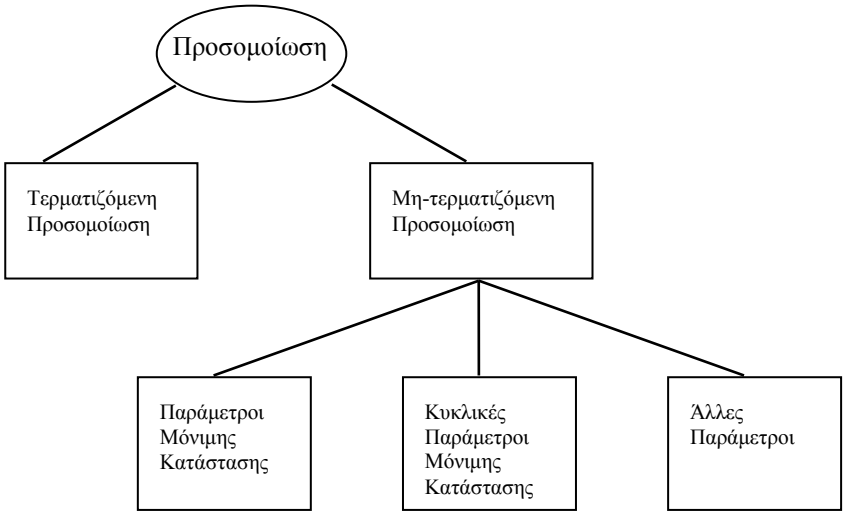
$$\begin{array}{ccccc} y_{11}, & \ldots, & y_{1i}, & \ldots, & y_{1m} \\ y_{21}, & \ldots, & y_{2i}, & \ldots, & y_{2m} \\ \vdots & & \vdots & & \vdots \\ y_{n1}, & \ldots, & y_{ni}, & \ldots, & y_{nm} \end{array}$$

Οι παρατηρήσεις μιας συγκεκριμένης εκτέλεσης (μίας γραμμής του παραπάνω Πίνακα), δεν είναι ανεξάρτητες και όμοια κατανεμημένες. Όμως, οι τιμές $y_{1i}, y_{2i}, \ldots, y_{ni}$ (της *i*-στής στήλης), είναι ανεξάρτητες, όμοια κατανεμημένες παρατηρήσεις της τυχαίας μεταβλητής Y_i για $i = 1,2,\ldots,m$. Αυτή η *"ανεξαρτησία διαμέσου διαφορετικών εκτελέσεων"*, είναι η βάση για τις σχετικά απλές τεχνικές ανάλυσης που θα δούμε στη συνέχεια. Ουσιαστικά, η προσπάθειά μας θα είναι να χρησιμοποιήσουμε τις

παρατηρήσεις y_{ji} $(i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$ για να βγάλουμε συμπεράσματα όσον αφορά τις κατανομές των τυχαίων μεταβλητών $Y_1, Y_2, \ldots, Y_m$. Για παράδειγμα, το μέγεθος $\bar{y}_i(n) = \Sigma_{j=1}^{n} y_{ji} / n$ είναι μια αμερόληπτη εκτιμήτρια του $E(Y_i)$.

10.2 ΤΥΠΟ ΠΡΟΣΟΜΟΙΩΣΗΣ ΑΝΑΦΟΡΙΚΑ ΜΕ ΤΗΝ ΑΝΑΛΥΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Οι προσομοιώσεις μπορούν να είναι *τερματιζόμενες* ή *μη - τερματιζόμενες*, ανάλογα με το αν υπάρχει ή όχι ένας προφανής τρόπος προσδιορισμού της διάρκειας της εκτέλεσης. Επίσης, τα μέτρα απόδοσης ή οι παράμετροι των μη - τερματιζομένων προσομοιώσεων, μπορούν να είναι διαφόρων τύπων, όπως φαίνεται στο Σχήμα 10.1.



Σχήμα 10.1. Τύποι Προσομοίωσης αναφορικά με την Ανάλυση Αποτελεσμάτων.

Σε μία τερματιζόμενη προσομοίωση, υπάρχει ένα "φυσικό" γεγονός *E*, το οποίο προσδιορίζει τη διάρκεια κάθε εκτέλεσης του προσομοιωτή. Με δεδομένο ότι οι διαφορετικές εκτελέσεις χρησιμοποιούν ανεξάρτητους τυχαίους αριθμούς, μπορούμε να συμπεράνουμε ότι συγκρίσιμες τυχαίες μεταβλητές από τις διαφορετικές εκτελέσεις, είναι ανεξάρτητες και όμοια κατανεμημένες. Το γεγονός *E* συχνά εμφανίζεται σε ένα χρονικό σημείο, μετά από το οποίο δεν παίρνουμε άλλη χρήσιμη πληροφορία, ή σε ένα χρονικό σημείο στο οποίο το σύστημα "αδειάζει". Προσδιορίζεται πριν αρχίσουν οι εκτελέσεις του προσομοιωτή και η στιγμή εμφάνισής του σε μία συγκεκριμένη εκτέλεση, μπορεί να είναι μια τυχαία μεταβλητή. Αφού οι αρχικές συνθήκες μιας τερματιζόμενης προσομοίωσης, γενικά επηρεάζουν τα επιθυμητά μέτρα απόδοσης, οι συνθήκες αυτές θα πρέπει να είναι αντιπροσωπευτικές των αντίστοιχων αρχικών συνθηκών του πραγματικού συστήματος.

Σε μία μη - τερματιζόμενη προσομοίωση, δεν υπάρχει φυσικό γεγονός *E* να προσδιορίζει τη διάρκεια μιας εκτέλεσης. Ένα μέτρο απόδοσης μιας τέτοιας

προσομοίωσης, ονομάζεται *παράμετρος μόνιμης κατάστασης*, αν είναι χαρακτηριστικό μέγεθος της κατανομής μόνιμης κατάστασης μιας στοχαστικής διαδικασίας εξόδου $Y_1, Y_2, \ldots$ Μία προσομοίωση ενός συγκεκριμένου συστήματος, μπορεί να είναι είτε τερματιζόμενη ή μη - τερματιζόμενη, ανάλογα με τους αντικειμενικούς στόχους της μελέτης. Ακόμα, πρέπει να σημειώσουμε ότι οι στοχαστικές διαδικασίες που περιγράφουν τα περισσότερα πραγματικά συστήματα, δεν έχουν κατανομές μόνιμης κατάστασης, αφού συνήθως τα χαρακτηριστικά του συστήματος μεταβάλλονται με το χρόνο. Από την άλλη πλευρά όμως, το μοντέλο προσομοίωσης (που είναι μια αφαίρεση της πραγματικότητας), μπορεί να έχει κατανομές μόνιμης κατάστασης, αφού συχνά θεωρούμε ότι τα χαρακτηριστικά του δεν μεταβάλλονται με την πάροδο του χρόνου.

Έστω μια στοχαστική διαδικασία $Y_1, Y_2, \ldots$ μίας μη - τερματιζόμενης προσομοίωσης, η οποία δεν έχει κατανομή μόνιμης κατάστασης. Υποθέτουμε ότι χωρίζουμε το χρονικό άξονα σε ισομεγέθη, συνεχόμενα χρονικά διαστήματα, τα οποία ονομάζουμε *κύκλους*. (Για παράδειγμα, σε ένα σύστημα παραγωγής, ένας κύκλος θα μπορούσε να είναι ένα 8-ωρο.) Έστω Y_i^C μία τυχαία μεταβλητή οριζόμενη στον *i*-στό κύκλο. Υποθέτουμε ότι τα $Y_1^C, Y_2^C, \ldots$ είναι συγκρίσιμα και επίσης ότι η διαδικασία $Y_1^C, Y_2^C, \ldots$ έχει μία κατανομή μόνιμης κατάστασης F^C, ενώ η $Y^C \sim F^C$. Τότε, ένα μέτρο απόδοσης καλείται *κυκλική παράμετρος μόνιμης κατάστασης*, αν είναι χαρακτηριστικό μέγεθος του Y^C, όπως για παράδειγμα η μέση τιμή $\nu^C = E(Y^C)$. Δηλαδή, μία κυκλική παράμετρος μόνιμης κατάστασης, είναι απλώς μία παράμετρος μόνιμης κατάστασης της κατάλληλης κυκλικής διαδικασίας $Y_1^C, Y_2^C, \ldots$

Σε μία μη - τερματιζόμενη προσομοίωση, ας υποθέσουμε ότι η στοχαστική διαδικασία $Y_1, Y_2, \ldots$ δεν έχει κατανομή μόνιμης κατάστασης και ότι δεν υπάρχει κατάλληλος ορισμός κύκλου, τέτοιος ώστε η αντίστοιχη διαδικασία $Y_1^C, Y_2^C, \ldots$ να έχει κατανομή μόνιμης κατάστασης. Αυτό μπορεί, για παράδειγμα, να συμβεί, αν οι παράμετροι του μοντέλου συνεχίζουν να μεταβάλλονται με το χρόνο. Πάντως, στις περιπτώσεις αυτές, συνήθως υπάρχει μία σταθερή ποσότητα δεδομένων που περιγράφει τον τρόπο με τον οποίο μεταβάλλονται οι παράμετροι εισόδου με το χρόνο. Το γεγονός αυτό, ουσιαστικά μας προσφέρει ένα γεγονός τερματισμού *E* της προσομοίωσης, οπότε μπορούμε να χρησιμοποιήσουμε τις τεχνικές ανάλυσης αποτελεσμάτων για τερματιζόμενες προσομοιώσεις. Τα μέτρα απόδοσης τέτοιων προσομοιώσεων, συνήθως μεταβάλλονται με το χρόνο και περιλαμβάνονται στην κατηγορία *"Άλλες Παράμετροι"* του Σχήματος 34.

10.3 ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ ΤΕΡΜΑΤΙΖΟΜΕΝΩΝ ΠΡΟΣΟΜΟΙΩΣΕΩΝ

Υποθέτουμε ότι διενεργούμε *n* ανεξάρτητες εκτελέσεις μιας τερματιζόμενης προσομοίωσης, όπου κάθε εκτέλεση τερματίζεται με το γεγονός *E* και αρχίζει με τις ίδιες αρχικές συνθήκες. Η ανεξαρτησία των εκτελέσεων επιτυγχάνεται με τη χρήση διαφορετικών τυχαίων αριθμών για κάθε εκτέλεση. Ας υποθέσουμε για λόγους απλότητας, ότι έχουμε μόνο ένα μέτρο απόδοσης που μας ενδιαφέρει να μελετήσουμε. Έστω X_j μία τυχαία μεταβλητή, οριζόμενη στην *j*-στή εκτέλεση, για $j = 1,2,\ldots,n$. Υποθέτουμε επίσης ότι τα X_j είναι συγκρίσιμα για διαφορετικές εκτελέσεις. Τότε, τα X_j είναι ανεξάρτητες, όμοια κατανεμημένες τυχαίες μεταβλητές.

10.3.1 Εκτίμηση Μέσων Τιμών

Έστω ότι θέλουμε να βρούμε μια *σημειακή εκτίμηση* και ένα *διάστημα εμπιστοσύνης* για τη μέση τιμή $\mu = E(X)$, όπου X είναι μια τυχαία μεταβλητή οριζόμενη σε μια εκτέλεση, όπως περιγράφηκε παραπάνω. Κάνουμε n ανεξάρτητες εκτελέσεις της προσομοίωσης και έστωσαν $X_1, X_2, \ldots, X_n$ οι ανεξάρτητες, όμοια κατανεμημένες τυχαίες τιμές, που παίρνουμε σαν αποτέλεσμα των εκτελέσεων. Τότε, η *δειγματική μέση τιμή*

$$\overline{X}(n) = \frac{\sum_{i=1}^{n} X_i}{n} \tag{10.1}$$

είναι μια *αμερόληπτη* σημειακή εκτίμηση του μ, δηλαδή $E[\overline{X}(n)] = \mu$. Επίσης, η *δειγματική διασπορά*

$$S^2(n) = \frac{\sum_{i=1}^{n} [X_i - \overline{X}(n)]^2}{n-1} \tag{10.2}$$

είναι μια αμερόληπτη σημειακή εκτίμηση της διασποράς σ^2 της τυχαίας μεταβλητής *X*. Δηλαδή $E[S^2(n)] = \sigma^2$.

Ένα *προσεγγιστικό* $100(1-a)\%,\ 0 < a < 1$ *διάστημα εμπιστοσύνης* για το μ, είναι το εξής:

$$\overline{X}(n) \pm t_{n-1,1-a/2} \sqrt{\frac{S^2(n)}{n}} \tag{10.3}$$

όπου $t_{n-1,1-a/2}$ είναι το *άνω* $1-a/2$ *κρίσιμο σημείο* της κατανομής *t* με *n*-1 df (βαθμούς ελευθερίας). Τα κρίσιμα αυτά σημεία μπορείτε να τα βρείτε στο ΠΑΡΑΡΤΗΜΑ Β.

Το παραπάνω διάστημα εμπιστοσύνης είναι *ακριβές*, αν οι τυχαίες μεταβλητές $X_1, X_2, \ldots, X_n$ ακολουθούν την *κανονική* κατανομή. Επειδή αυτό είναι σχετικά σπάνια περίπτωση, η (9) μπορεί να θεωρηθεί ως ένα προσεγγιστικό διάστημα εμπιστοσύνης, λόγω της ισχύος του *κεντρικού οριακού θεωρήματος*:

Θεώρημα 4. (Κεντρικό Οριακό Θεώρημα).

Έστω η τυχαία μεταβλητή $Z_n = [\overline{X}(n) - \mu] / \sqrt{S^2(n)/n}$ *και έστω* $F_n(z)$ *η συνάρτηση κατανομής της* Z_n *για ένα δείγμα* $X_1, X_2, \ldots, X_n$ *μεγέθους n. Δηλαδή,* $F_n(z) = P(Z_n \le z)$. *Τότε, η* $F_n(z) \to \Phi(z)$ *για* $n \to \infty$, *όπου* $\Phi(z)$ *είναι η συνάρτηση κατανομής της standard κανονικής κατανομής* $N(0,1)$.

□

Ουσιαστικά, το θεώρημα δηλώνει ότι, αν το *n* είναι "αρκετά μεγάλο", η τυχαία μεταβλητή Z_n θα είναι προσεγγιστικά κατανεμημένη σύμφωνα με την standard κανονική κατανομή $N(0,1)$, ανεξάρτητα από την κατανομή που ακολουθούν τα X_i. Μία άλλη διατύπωση του θεωρήματος αναφέρει ότι: "*για μεγάλο n, η δειγματική μέση τιμή*

$\overline{X}(n)$ *είναι προσεγγιστικά κατανεμημένη σύμφωνα με την κανονική κατανομή* $N(\mu, \sigma^2 / n)$.”

10.3.2 Δημιουργία επιθυμητής Ακρίβειας

Ένα μειονέκτημα της διαδικασίας που μας οδήγησε στην (9), είναι ότι δεν έχουμε τον έλεγχο του ημι - διαστήματος εμπιστοσύνης [*ακρίβεια* του $\overline{X}(n)$]. Για δεδομένο *n*, η ακρίβεια θα εξαρτάται από τη διακύμανση $\mathrm{Var}(X)$ του πληθυσμού των X_i. Θα εξετάσουμε στη συνέχεια, τον τρόπο προσδιορισμού του αριθμού των εκτελέσεων που απαιτούνται για την εκτίμηση της μέσης τιμής $\mu = E(X)$, ώστε να έχουμε ένα συγκεκριμένο σφάλμα ή ακρίβεια.

Για να υπολογίσουμε το σφάλμα της εκτιμήτριας $\overline{X}$, δεν χρησιμοποιούμε πλέον [και στο συμβολισμό του $\overline{X}(n)$], την εξάρτηση από το *n*, αφού ο αριθμός των εκτελέσεων μπορεί πλέον να είναι μια τυχαία μεταβλητή. Αν η εκτιμήτρια $\overline{X}$ είναι τέτοια ώστε $\left|\overline{X} - \mu\right| = \beta$, τότε λέμε ότι η $\overline{X}$ έχει <u>απόλυτο σφάλμα</u> ίσο με β. Αν κάνουμε συνεχείς εκτελέσεις μιας προσομοίωσης, μέχρι το ημι - διάστημα του $100(1-a)\%$ διαστήματος εμπιστοσύνης της (9), να γίνει μικρότερο ή ίσο από β (με $\beta > 0$), τότε:

$$\begin{aligned} 1-a &\approx P(\overline{X} - \text{ημι-μήκος} \le \mu \le \overline{X} + \text{ημι-μήκος}) \\ &= P(\left|\overline{X} - \mu\right| \le \text{ημι-μήκος}) \\ &\le P(\left|\overline{X} - \mu\right| \le \beta) \end{aligned}$$

Δηλαδή, το $\overline{X}$ έχει απόλυτο σφάλμα το πολύ β, με πιθανότητα περίπου $1-a$. Με άλλα λόγια, αν κατασκευάσουμε 100 ανεξάρτητα διαστήματα εμπιστοσύνης 90%, χρησιμοποιώντας τον παραπάνω κανόνα τερματισμού, περιμένουμε το $\overline{X}$ να έχει απόλυτο σφάλμα το πολύ β, σε περίπου 90 από τις 100 περιπτώσεις. Σε περίπου 10 περιπτώσεις, το απόλυτο σφάλμα θα είναι μεγαλύτερο από β.

Έστω ότι έχουμε κατασκευάσει ένα διάστημα εμπιστοσύνης για το μ, βασισμένοι σε ένα δεδομένο αριθμό εκτελέσεων *n*. Αν υποθέσουμε ότι η εκτίμησή μας $S^2(n)$ για τη διακύμανση του πληθυσμού, δεν θα έχει αξιοσημείωτη μεταβολή, όσο αυξάνει ο αριθμός των εκτελέσεων, μία *προσεγγιστική* έκφραση για το συνολικό αριθμό εκτελέσεων $n_a^*(\beta)$ που απαιτούνται για να πάρουμε ένα απόλυτο σφάλμα ίσο με β, είναι η εξής:

$$n_a^*(\beta) = \min\left\{ i \ge n : t_{i-1,1-a/2}\sqrt{\frac{S^2(n)}{i}} \le \beta \right\} \qquad (10.4)$$

Μπορούμε να προσδιορίσουμε το $n_a^*(\beta)$, αυξάνοντας επαναληπτικά το *i* κατά 1, μέχρι να βρούμε μια τιμή του i για την οποία $t_{i-1,1-a/2}\sqrt{S^2(n)/i} \le \beta$. Αν $n_a^*(\beta) > n$ και κάνουμε $n_a^*(\beta) - n$ επιπλέον εκτελέσεις της προσομοίωσης, τότε η εκτίμηση $\overline{X}$ βασισμένη σε όλες τις $n_a^*(\beta)$ εκτελέσεις, θα πρέπει να έχει ένα απόλυτο σφάλμα περίπου β. Η ακρίβεια της σχέσης (10) εξαρτάται από το πόσο καλή εκτίμηση του $\mathrm{Var}(X)$ είναι το $S^2(n)$.

Αν ο εκτιμητής $\overline{X}$ είναι τέτοιος ώστε $\left|\overline{X}-\mu\right|/\left|\mu\right|=\gamma$, τότε λέμε ότι το $\overline{X}$ έχει *σχετικό σφάλμα* ίσο με γ, ή ότι το *ποσοστιαίο σφάλμα* του $\overline{X}$ είναι 100γ %. Υποθέτουμε ότι επαναλαμβάνουμε την εκτέλεση μιας προσομοίωσης, μέχρι το ημι - διάστημα εμπιστοσύνης της (9), διαιρεμένο με $\left|\overline{X}\right|$, να γίνει μικρότερο ή ίσο με γ $(0<\gamma<1)$. Ο λόγος αυτός είναι μια εκτίμηση του πραγματικού σχετικού σφάλματος. Τότε

$$\begin{aligned}1-a &\approx P(\left|\overline{X}-\mu\right|/\left|\overline{X}\right| \le \text{ημι-μήκος}/\left|\overline{X}\right|)\\ &\le P(\left|\overline{X}-\mu\right| \le \gamma\left|\overline{X}\right|)\\ &= P(\left|\overline{X}-\mu\right| \le \gamma\left|\overline{X}-\mu+\mu\right|)\\ &\le P(\left|\overline{X}-\mu\right| \le \gamma(\left|\overline{X}-\mu\right|+\left|\mu\right|))\\ &= P((1-\gamma)\left|\overline{X}-\mu\right| \le \gamma\left|\mu\right|)\\ &= P(\left|\overline{X}-\mu\right|/\left|\mu\right| \le \gamma/(1-\gamma))\end{aligned}$$

Δηλαδή, το $\overline{X}$ έχει ένα σχετικό σφάλμα το πολύ $\gamma/(1-\gamma)$, με πιθανότητα περίπου $1-a$. Ή αλλιώς, αν κατασκευάσουμε 100 ανεξάρτητα διαστήματα εμπιστοσύνης 90%, χρησιμοποιώντας τον παραπάνω κανόνα τερματισμού, περιμένουμε το $\overline{X}$ να έχει σχετικό σφάλμα το πολύ $\gamma/(1-\gamma)$ σε περίπου 90 από τις 100 περιπτώσεις. Σε περίπου 10 περιπτώσεις, το σχετικό σφάλμα θα είναι μεγαλύτερο από $\gamma/(1-\gamma)$.

Έστω πάλι, ότι έχουμε κατασκευάσει ένα διάστημα εμπιστοσύνης για το μ, βασισμένοι σε ένα δεδομένο αριθμό εκτελέσεων *n*. Αν υποθέσουμε ότι οι εκτιμήσεις μας για τη μέση τιμή και τη διακύμανση του πληθυσμού, δεν θα έχουν αξιοσημείωτη μεταβολή, όσο αυξάνει ο αριθμός των εκτελέσεων, μία *προσεγγιστική* έκφραση για το συνολικό αριθμό εκτελέσεων $n_r^*(\gamma)$ που απαιτούνται για να πάρουμε ένα σχετικό σφάλμα ίσο με γ, είναι η εξής:

$$n_r^*(\gamma)=\min\left\{i\ge n:\frac{t_{i-1,1-a/2}\sqrt{S^2(n)/i}}{\left|\overline{X}(n)\right|}\le\gamma'\right\} \qquad (10.5)$$

όπου $\gamma'=\gamma/(1+\gamma)$ είναι το "τροποποιημένο" σχετικό σφάλμα που απαιτείται για να πάρουμε ένα *πραγματικό* σχετικό σφάλμα ίσο με γ [αλλάζουμε το συμβολισμό από γ σε γ' και λύνουμε την $\gamma=\gamma'/(1-\gamma')$]. Αν $n_r^*(\gamma)>n$ και κάνουμε $n_r^*(\gamma)-n$ επιπλέον εκτελέσεις της προσομοίωσης, τότε η εκτίμηση $\overline{X}$ βασισμένη σε όλες τις $n_r^*(\gamma)$ εκτελέσεις, θα πρέπει να έχει ένα σχετικό σφάλμα περίπου γ.

Η δυσκολία με τη χρήση της (11), βρίσκεται στο γεγονός ότι τα $\overline{X}(n)$ και $S^2(n)$ μπορεί να μην είναι ακριβείς εκτιμήσεις των αντίστοιχων παραμέτρων του πληθυσμού. Αν το $n_r^*(\gamma)$ βγει μεγαλύτερο από τον αριθμό εκτελέσεων που πραγματικά χρειάζονται, τότε μπορεί να κάνουμε ένα σημαντικό αριθμό άχρηστων επαναλήψεων, οι οποίες κοστίζουν σε χρόνο και υπολογιστική ισχύ. Αντίθετα, αν το $n_r^*(\gamma)$ βγει πολύ μικρό, μπορεί να πάρουμε εκτίμηση $\overline{X}$ όχι όσο ακριβή πιστεύουμε ότι είναι. Στη συνέχεια παρουσιάζεται μια *ακολουθιακή* διαδικασία (νέες εκτελέσεις προστίθενται, μία τη φορά), με την οποία παίρνουμε μια εκτίμηση του μ με επιθυμητό σχετικό σφάλμα,

προσδιορίζοντας τόσες εκτελέσεις, όσες πραγματικά χρειάζονται. Η διαδικασία προϋποθέτει ότι τα $X_1, X_2, \ldots$ είναι μια ακολουθία ανεξαρτήτων, όμοια κατανεμημένων τυχαίων μεταβλητών, όχι απαραίτητα σύμφωνα με την κανονική κατανομή.

Ο στόχος της διαδικασίας είναι να προσδιορίσει μια εκτίμηση του μ με σχετικό σφάλμα γ $(0 < \gamma < 1)$ και επίπεδο εμπιστοσύνης $100(1-a)\%$. Επιλέγουμε έναν αρχικό αριθμό εκτελέσεων $n_0 \geq 2$. Έστω

$$\delta(n,a) = t_{n-1,1-a/2}\sqrt{\frac{S^2(n)}{n}}$$

το σύνηθες ημι - διάστημα εμπιστοσύνης. Τότε, η ακολουθιακή διαδικασία είναι η ακόλουθη:

0. Κάνε n_0 εκτελέσεις της προσομοίωσης και θέσε $n = n_0$.

1. Υπολόγισε τα $\overline{X}(n)$ και $\delta(n,a)$ από τα $X_1, X_2, \ldots, X_n$.

2. Αν $\delta(n,a) / \left|\overline{X}(n)\right| \leq \gamma'$, χρησιμοποίησε το $\overline{X}(n)$ ως τη σημειακή εκτίμηση του μ και σταμάτησε τις εκτελέσεις του προσομοιωτή. Τότε, το

$$I(a,\gamma) = [\overline{X}(n) - \delta(n,a),\ \overline{X}(n) + \delta(n,a)]$$

(12)

είναι ένα προσεγγιστικό $100(1-a)\%$ διάστημα εμπιστοσύνης του μ με την επιθυμητή ακρίβεια. Αλλιώς, αντικατέστησε το n με το $n+1$, κάνε μία επιπλέον εκτέλεση της προσομοίωσης και πήγαινε στο Βήμα 1.

Παρατηρήστε, ότι η διαδικασία υπολογίζει μια νέα εκτίμηση του $\mathrm{Var}(X)$ μετά από κάθε εκτέλεση, ενώ ο συνολικός αριθμός των εκτελέσεων που απαιτούνται, είναι μια τυχαία μεταβλητή.

10.4 ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ ΠΑΡΑΜΕΤΡΩΝ ΜΟΝΙΜΗΣ ΚΑΤΑΣΤΑΣΗΣ

Έστω $Y_1, Y_2, \ldots$ μία στοχαστική διαδικασία εξόδου, από *μία* εκτέλεση μιας μη - τερματιζόμενης προσομοίωσης. Έστω ότι $P(Y_i \leq y) = F_i(y) \to F(y) = P(Y \leq y)$ όταν $i \to \infty$, όπου Y είναι η τυχαία μεταβλητή στη μόνιμη κατάσταση που μας ενδιαφέρει, με συνάρτηση κατανομής F. (Δεν εμφανίζουμε στο συμβολισμό μας, εξάρτηση των F_i από τις *αρχικές συνθήκες* I). Τότε το ϕ είναι μία παράμετρος μόνιμης κατάστασης, αν είναι χαρακτηριστικό μέγεθος του Y, όπως το $E(Y)$, το $P(Y \leq y)$ κ.α. Μια δυσκολία στην εκτίμηση του ϕ οφείλεται στο γεγονός ότι η συνάρτηση κατανομής των Y_i $(i = 1,2,\ldots)$ είναι διαφορετική από την F, αφού γενικά δεν μπορούμε να επιλέξουμε αρχικές συνθήκες I, αντιπροσωπευτικές της "συμπεριφοράς μόνιμης κατάστασης" της διαδικασίας. Αυτό σημαίνει ότι εκτιμήσεις του ϕ βασισμένες στις παρατηρήσεις $Y_1, Y_2, \ldots, Y_m$ δεν θα είναι αντι-προσωπευτικές. Για παράδειγμα, η δειγματική μέση τιμή $\overline{Y}(m)$ δεν θα είναι αμερόληπτη εκτιμήτρια του $\nu = E(Y)$ για όλες τις πεπερασμένες τιμές του m. Το πρόβλημα αυτό, είναι το γνωστό *πρόβλημα της αρχικής μεταβατικής περιόδου* σε μια μη - τερματιζόμενη προσομοίωση.

10.4.1. Το Πρόβλημα της Αρχικής Μεταβατικής Περιόδου

Ας υποθέσουμε ότι θέλουμε να εκτιμήσουμε τη μέση τιμή στη μόνιμη κατάσταση $\nu = E(Y)$, η οποία επίσης ορίζεται γενικά από τη σχέση:

$$\nu = \lim_{i\to\infty} E(Y_i)$$

Δηλαδή, οι μεταβατικές μέσες τιμές συγκλίνουν στη μέση τιμή της μόνιμης κατάστασης. Η πιο σοβαρή συνέπεια του προβλήματος της αρχικής μεταβατικής περιόδου, είναι μάλλον το γεγονός ότι $E[\overline{Y}(m)] \neq \nu$ για κάθε *m*. Για την αντιμετώπιση της συνέπειας αυτής, χρησιμοποιείται η τεχνική της *"διαγραφής των αρχικών παρατηρήσεων"*. Η ιδέα της τεχνικής αυτής, είναι να διαγράφεται ένας αριθμός παρατηρήσεων από την αρχή της εκτέλεσης και να χρησιμοποιούνται μόνο οι υπόλοιπες παρατηρήσεις για την εκτίμηση του ν. Για παράδειγμα, άν έχουμε τις παρατηρήσεις $Y_1, Y_2, \ldots, Y_m$, θα χρησιμοποιήσουμε το

$$\overline{Y}(m,l) = \frac{\sum_{i=l+1}^{m} Y_i}{m-l}$$

$(1 \le l \le m-1)$, αντί για το $\overline{Y}(m)$ ως εκτιμητή του ν. Γενικά περιμένουμε το $\overline{Y}(m,l)$ να είναι περισσότερο αμερόληπτος εκτιμητής από το $\overline{Y}(m)$, αφού οι παρατηρήσεις κοντά στην αρχή της προσομοίωσης, μπορεί να μην είναι αντι-προσωπευτικές της συμπεριφοράς μόνιμης κατάστασης, λόγω της επιλογής των αρχικών συνθηκών.

Το ερώτημα είναι πώς υπολογίζουμε το μήκος *l* της αρχικής μεταβατικής περιόδου. Θα θέλαμε να βρούμε ένα *l* (και ένα *m*), έτσι ώστε $E[\overline{Y}(m,l)] \approx \nu$. Αν επιλέξουμε *l* και *m* πολύ μικρά, τότε το $E[\overline{Y}(m,l)]$ μπορεί να είναι σημαντικά διαφορετικό από το ν. Αντίθετα, αν το *l* επιλεγεί μεγαλύτερο από ότι χρειάζεται, τότε το $\overline{Y}(m,l)$ θα έχει πιθανότατα απαράδεκτα μεγάλη διακύμανση.

Μία μέθοδος που αντιμετωπίζει το πρόβλημα του προσδιορισμού του *l* παρουσιάζεται στον αλγόριθμο που ακολουθεί. Ο στόχος του αλγορίθμου, είναι να προσδιορισθεί ένας χρονικός δείκτης *l*, τέτοιος ώστε $E(Y_i) \approx \nu$ για $i > l$. Γενικά είναι πολύ δύσκολο να προσδιορισθεί το *l* από μία μόνο εκτέλεση, λόγω της μεταβλητότητας της διαδικασίας $Y_1, Y_2, \ldots$ Έτσι, ο αλγόριθμος βασίζεται σε *n* ανεξάρτητες εκτελέσεις της προσομοίωσης. Έχει τα παρακάτω 4 Βήματα:

1. Κάνε *n* εκτελέσεις της προσομοίωσης $(n \ge 5)$, μήκους *m* παρατηρήσεων η κάθε μία (το *m* είναι μεγάλο). Έστω Y_{ji} η *i*-στή παρατήρηση από την *j*-στή εκτέλεση $(j = 1,2,\ldots,n;\ i = 1,2,\ldots,m)$.

2. Θέσε $\overline{Y}_i = \Sigma_{j=1}^{n} Y_{ji} / n$ για $i = 1,2,\ldots,m$. Η διαδικασία μέσων τιμών $\overline{Y}_1, \overline{Y}_2, \ldots$ έχει μέσες τιμές $E(\overline{Y}_i) = E(Y_i)$ και διακυμάνσεις $\mathrm{Var}(\overline{Y}_i) = \mathrm{Var}(Y_i) / n$.

3. Ορίζουμε τη *μετακινούμενη μέση τιμή* $\overline{Y}_i(w)$, (όπου *w* είναι το *παράθυρο* και είναι ένας θετικός ακέραιος, τέτοιος ώστε $w \le \lfloor m/2 \rfloor$) ως εξής:

$$\overline{Y}_i(w) = \begin{cases} \dfrac{\sum_{s=-w}^{w} \overline{Y}_{i+s}}{2w+1} & \alpha\nu \;\; i = w+1,\ldots,m-w \\ \dfrac{\sum_{s=-(i-1)}^{i-1} \overline{Y}_{i+s}}{2i-1} & \alpha\nu \;\; i = 1,\ldots,w \end{cases}$$

Δηλαδή, αν το i δεν είναι πολύ κοντά στην αρχή των εκτελέσεων, τότε το $\overline{Y}_i(w)$ είναι απλώς η μέση τιμή $2w+1$ παρατηρήσεων της διαδικασίας μέσων τιμών, γύρω από την παρατήρηση i . Ονομάζεται μετακινούμενη μέση τιμή, διότι το i μετακινείται με την πάροδο του χρόνου.

4. Κατασκεύασε τη γραφική παράσταση του $\overline{Y}_i(w)$ για $i = 1,2,\ldots,m-w$ και επέλεξε ένα l ίσο με την τιμή του i, κάτω από την οποία η ακολουθία $\overline{Y}_1(w), \overline{Y}_2(w),\ldots$ φαίνεται να έχει συγκλίνει.

10.4.1 Εκτίμηση της Μέσης Τιμής

Θα χρησιμοποιήσουμε τη *μέθοδο της επανάληψης - διαγραφής*, για τον προσδιορισμό μιας σημειακής εκτίμησης και ενός διαστήματος εμπιστοσύνης για το ν. Η ανάλυση είναι παρόμοια με αυτή των τερματιζομένων προσομοιώσεων, εκτός του γεγονότος ότι χρησιμοποιούμε μόνο τις παρατηρήσεις που παίρνουμε μετά την αρχική μεταβατική περίοδο *l*. Συγκεκριμένα, έστω ότι κάνουμε n' εκτελέσεις της προσομοίωσης, μήκους m' παρατηρήσεων η κάθε μία, με m' αρκετά μεγαλύτερο από το *l* που προσδιορίζουμε σύμφωνα με την προηγούμενη Παράγραφο. Έστω Y_{ji} η *i*-στή παρατήρηση από την *j*-στή εκτέλεση και έστω ότι τα X_j δίνονται από την:

$$X_j = \frac{\sum_{i=l+1}^{m'} Y_{ji}}{m'-l} \qquad \gamma\iota\alpha \;\; j = 1,2,\ldots,n'$$

Δηλαδή, τα X_j χρησιμοποιούν μόνο τις παρατηρήσεις $Y_{j,l+1}, Y_{j,l+2},\ldots,Y_{j,m'}$ της *j*-στής εκτέλεσης που αντιστοιχούν στη "μόνιμη κατάσταση". Τα X_j είναι ανεξάρτητες, όμοια κατανεμημένες τυχαίες μεταβλητές με $E(X_j) \approx \nu$, το $\overline{X}(n')$ είναι ένας προσεγγιστικά αμερόληπτος εκτιμητής του ν, ενώ ένα προσεγγιστικό διάστημα εμπιστοσύνης $100(1-a)\%$ του ν είναι το:

$$\overline{X}(n') \pm t_{n'-1,1-a/2}\sqrt{\frac{S^2(n')}{n'}} \qquad (10.6)$$

όπου τα $\overline{X}(n')$ και $S^2(n')$ υπολογίζονται με βάση τις σχέσεις (10.1) και (10.2) αντίστοιχα.

10.5 ΣΤΑΤΙΣΤΙΚΗ ΑΝΑΛΥΣΗ ΚΥΚΛΙΚΩΝ ΠΑΡΑΜΕΤΡΩΝ ΜΟΝΙΜΗΣ ΚΑΤΑΣΤΑΣΗΣ

Έστω ότι η διαδικασία εξόδου $Y_1, Y_2, \ldots$ δεν έχει κατανομή μόνιμης κατάστασης. Υποθέτουμε όμως, ότι υπάρχει κατάλληλος ορισμός "κύκλου", έτσι ώστε η διαδικασία $Y_1^C, Y_2^C, \ldots$ να έχει κατανομή μόνιμης κατάστασης F^C, όπου Y_i^C είναι η τυχαία μεταβλητή ορισμένη στον i-στό κύκλο. Αν η $Y^C \sim F^C$, τότε ενδιαφερόμαστε να εκτιμήσουμε κάποιο χαρακτηριστικό μέγεθος της Y^C, όπως τη μέση τιμή $\nu^C = E(Y^C)$, ή την πιθανότητα $P(Y^C \le y)$. Προφανώς, η εκτίμηση μιας κυκλικής παραμέτρου μόνιμης κατάστασης, είναι απλώς μια ειδική περίπτωση εκτίμησης μιας παραμέτρου μόνιμης κατάστασης, ώστε μπορούμε να χρησιμοποιήσουμε τις τεχνικές της προηγούμενης Παραγράφου 7.4, αλλά με τις κυκλικές τυχαίες μεταβλητές Y_i^C, αντί των αρχικών Y_i.