ΣΥΝΘΕΤΑ ΣΥΣΤΗΜΑΤΑ ΔΙΑΚΡΙΤΩΝ ΓΕΓΟΝΟΤΩΝ

3.1 ΕΙΣΑΓΩΓΗ

Στο Κεφάλαιο 1 είδαμε τη γενική δομή ενός αλγόριθμου διακριτών γεγονότων και μερικά απλά παραδείγματα. Στην πράξη, τα συστήματα διακριτών γεγονότων είναι αρκετά περίπλοκα και η ανάπτυξη αλγόριθμων προσομοίωσης είναι επίπονη. Στο εμπόριο διατίθενται γλώσσες προγραμματισμού εξειδικευμένες στην προσομοίωση καθώς και έτοιμα προγράμματα προσομοίωσης ειδικών συστημάτων (π.χ. συστήματα αναμονής, γραμμές και δίκτυα παραγωγής, αποθεματικά συστήματα). Το πλεονέκτημά τους είναι ότι για μικρά συστήματα απαιτούν λίγες γραμμές κώδικα ή απλώς τη σχηματική αναπαράσταση του συστήματος με έτοιμα εικονίδια. Διαθέτουν έτοιμα υποπρογράμματα-βιβλιοθήκες που περιγράφουν συνιστώσες του συστήματος και τις διασυνδέσεις τους (μηχανές ή εξυπηρετητές, αποθήκες, παλέτες ή μεταφορικές ταινίες, οχήματα κτλ.), χωρίς να χρειάζεται προγραμματισμός στον υπολογιστή και έχουν δυνατότητα γραφικής αναπαράστασης της κυκλοφορίας των οντοτήτων του συστήματος. Ωστόσο επειδή τα συνήθη ή μεγάλα συστήματα έχουν ιδιαιτερότητες που απαιτούν προγραμματισμό, παρέχεται η δυνατότητα στον χρήστη να αναπτύσσει δικά του υποπρογράμματα και να τα ενσωματώνει στο μοντέλο.

Στο κεφάλαιο αυτό θα εξετάσουμε συστήματα διακριτών γεγονότων αναπτύσσοντας αλγόριθμους προσομοίωσης εκ του μηδενός. Θα πρέπει να σημειωθεί ότι στους αλγορίθμους που παρουσιάζονται δίνεται περισσότερη έμφαση στην ευκολία κατανόησης ενώ η ανάπτυξη γρήγορων αλγορίθμων με οικονομία στη χρήση μνήμης του υπολογιστή αποτελεί δευτερεύουσα προτεραιότητα. Κατά συνέπειαν, ο αναγνώστης μπορεί να επιτύχει οικονομία, επιτάχυνση *αλλά και γενίκευση* των αλγορίθμων για πιο περίπλοκα συστήματα κάνοντας κατάλληλες τροποποιήσεις ή αλλάζοντας τη λογική των μοντέλων.

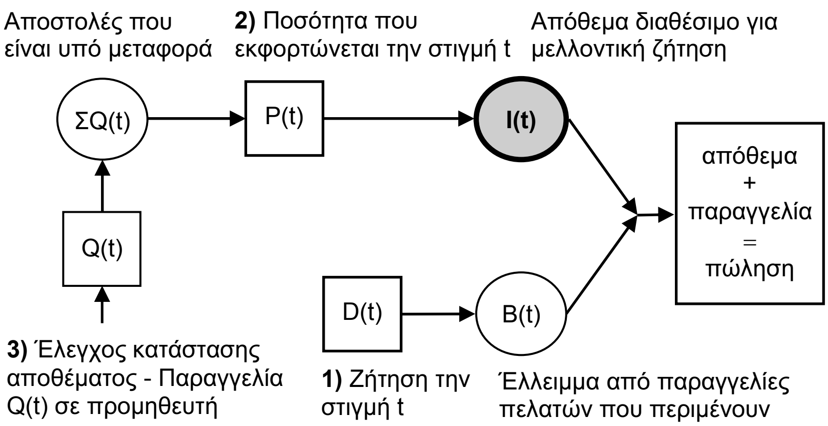
Για κάθε σύστημα που εξετάζεται στο εξής, όλες οι τ.μ. οι οποίες περιγράφουν τυχαία φαινόμενα, όπως για παράδειγμα οι χρόνοι μεταξύ αφίξεων πελατών ή οι διάρκειες εξυπηρέτησης ή κατεργασίας κομματιών, υποτίθεται ότι είναι μεταξύ τους *ανεξάρτητες*, εκτός αν ρητά αναφέρεται το αντίθετο.

3.2 ΕΝΑ ΑΠΟΘΕΜΑΤΙΚΟ ΣΥΣΤΗΜΑ

3.2.1 Περιγραφή

Κάποια εταιρεία προμηθεύεται ένα προϊόν από προμηθευτή, το αποθηκεύει και κατόπιν το πωλεί σε πελάτες που το ζητούν σε τυχαίους χρόνους. Το απόθεμα είναι θετικό αν υπάρχει προϊόν διαθέσιμο προς πώληση, ή αρνητικό όταν υπάρχει έλλειμμα λόγω ανικανοποίητων παραγγελιών που είχαν τεθεί από πελάτες όταν δεν υπήρχε απόθεμα. Στο Σχήμα 3.1 φαίνεται μία αναπαράσταση του αποθεματικού συστήματος. Τα τετράγωνα συμβολίζουν ποσότητες προϊόντος που παραγγέλλονται, αποθηκεύονται ή πωλούνται. Οι κύκλοι συμβολίζουν

- είτε τον φυσικό χώρο αποθήκευσης υλικών (γκρι κύκλος).
- είτε εικονικές «αποθήκες» συγκέντρωσης παραγγελιών που εκκρεμούν: αποστολές από τον προμηθευτή (φορτηγά καθ' οδόν για να φέρουν προϊόντα στην αποθήκη) ή παραγγελίες πελατών σε εκκρεμότητα (τρέχον έλλειμμα αποθέματος).



Σχήμα 3.1: Η κυκλοφορία υλικών σε ένα αποθεματικό σύστημα καθορίζεται από 3 γεγονότα: **(1)** ζήτηση D(t) κομματιών, **(2)** άφιξη φορτίου P(t) κομματιών για την αναπλήρωση αποθέματος και **(3)** έλεγχος της αποθεματικής θέσης του συστήματος που πιθανόν να οδηγήσει σε νέα παραγγελία αναπλήρωσης αποθέματος με Q(t) μονάδες προϊόντος. Αν x είναι ο χρόνος μεταφοράς, τότε η παραγγελία Q(t) θα φθάσει αργότερα ως P(t + x) και αν, επί πλέον, παραδίδεται τμηματικά, τότε εν γένει έχουμε $P(t + x) \leq Q(t)$.

Το σύστημα (εταιρεία) δέχεται παραγγελίες από πελάτες. Όταν ένας πελάτης ζητεί D(t) μονάδες προϊόντος, η παραγγελία του ικανοποιείται αμέσως από το *διαθέσιμο απόθεμα* I(t), αν I(t) > 0, και αν δεν επαρκεί τότε το απόθεμα μηδενίζεται και το υπόλοιπο μέρος της παραγγελίας καταγράφεται ως εκκρεμής και λογίζεται ως έλλειμμα B(t) ή ισοδύναμα ως *αρνητικό* καθαρό απόθεμα –B(t).

Ένας προμηθευτής (ή περισσότεροι από έναν) φροντίζει για την αποστολή προϊόντων κάθε φορά που λαμβάνει μία παραγγελία αναπλήρωσης αποθέματος από το σύστημα. Όταν μία αποστολή φθάνει, ένα μέρος της διοχετεύεται στους πελάτες που περιμένουν, αν B(t) > 0, και η υπόλοιπη αποθηκεύεται και υπολογίζεται στο διαθέσιμο απόθεμα I(t).

Ιδιότητα: Δεν είναι δυνατόν να έχουμε συγχρόνως και διαθέσιμο απόθεμα I(t) > 0 και έλειμμα B(t) > 0. Πρέπει ένα τουλάχιστον από τα δύο να είναι 0. Συνεπώς, I(t)B(t) = 0.

Η διαφορά

$$x(t) = I(t) - B(t)$$

ονομάζεται *καθαρό απόθεμα* (net inventory, inventory level). Από την ιδιότητα I(t)B(t) = 0, προκύπτει ότι

$$x(t) = \begin{cases} I(t) & \text{όταν } B(t) = 0 \\ -B(t) & \text{όταν } I(t) = 0. \end{cases}$$

Ακολουθούν ορισμένες υποθέσεις αναφορικά με την ζήτηση, τις παραγγελίες αναπλήρωσης αποθέματος και τις αποστολές τους και η περιγραφή κάποιων οικονομικών μεγεθών που επηρεάζουν το κόστος λειτουργίας του συστήματος. Για λόγους απλότητας, παραλείπουμε το t από τον συμβολισμό των μεταβλητών, εκτός εάν είναι απαραίτητο.

Οι χρόνοι μεταξύ δύο διαδοχικών παραγγελιών προς πώληση είναι εκθετικά κατανεμημένοι με μέση τιμή 1/μ μήνες. Η ζήτηση D ενός αφικνούμενου πελάτη υποθέτουμε ότι είναι τ.μ. που λαμβάνει τιμές d_i (κομμάτια) με γνωστές πιθανότητες p_i:

$$D = \begin{cases} d_1 & \text{μ.π. } p_1 \\ d_2 & \text{μ.π. } p_2 \\ \vdots & \\ d_r & \text{μ.π. } p_r \end{cases}$$

Στην *αρχή κάθε μήνα επιθεωρείται* το απόθεμα και αποφασίζεται πόσα κομμάτια θα παραγγελθούν στον προμηθευτή. Το κόστος παραγγελίας Q κομματιών είναι A + CQ, όπου A το σταθερό κόστος παραγγελίας. Αν Q = 0 τότε το κόστος είναι 0.

Το χρονικό διάστημα από τη στιγμή που δίνεται μία παραγγελία στον προμηθευτή μέχρις ότου η αποστολή της ολοκληρωθεί και η αντίστοιχη ποσότητα μεταφερθεί στην αποθήκη του συστήματος είναι τ.μ. $\sim U(\alpha, \beta)$. Το α είναι ο ελάχιστος χρόνος που μεσολαβεί από την παραγγελία μέχρι την άφιξή της και β ο μέγιστος χρόνος. Μονάδα χρόνου είναι οι μήνες.

Η εταιρεία εφαρμόζει μία παραλλαγή της πολιτικής αναπλήρωσης αποθεμάτων (s, S). Σύμφωνα με την πολιτική (s, S), στην αρχή του μήνα δεν γίνεται παραγγελία αν το απόθεμα είναι s ή μεγαλύτερο, ενώ αν το απόθεμα είναι μικρότερο του s παραγγέλλονται τόσα κομμάτια όσα χρειάζονται για να γεμίσουν την αποθήκη. Οι παράμετροι s και S είναι ακέραιες και $s \leq S$. Το S παίζει το ρόλο του *μέγιστου αποθέματος* και το s είναι το *απόθεμα ασφαλείας*. Η πολιτική (s, S) είναι εφαρμόσιμη όταν έχουμε άμεση παράδοση ($\alpha = \beta = 0$) ή ο χρόνος παράδοσης δεν ξεπερνά τον χρόνο επόμενης επιθεώρησης, ήτοι $\beta \leq 1$ μήνας. Αν οι χρόνοι παράδοσης είναι μεγαλύτεροι, τότε η εφαρμογή της πολιτικής αυτής μπορεί να οδηγήσει σε πλεονασματικές παραγγελίες (ενώ υπάρχουν αποστολές υπό μεταφορά) και υπέρβαση του S. Στη συνέχεια εξετάζουμε μία παρόμοια πολιτική αναπλήρωσης αποθεμάτων, η οποία συμμορφώνεται με τον περιορισμό μέγιστου αποθέματος S, ανεξαρτήτως καθυστερήσεων στην παράδοση.

Έστω x το καθαρό απόθεμα και ΣQ το άθροισμα προϊόντων που αναμένονται από *προηγούμενες παραγγελίες* στον προμηθευτή οι οποίες δεν έχουν αφιχθεί ακόμη. Η ποσότητα x + ΣQ είναι γνωστή ως *αποθεματική θέση* (inventory position) του συστήματος.

Συμβολίζουμε [s, S] τη νέα πολιτική παραγγελιών που θα εφαρμοσθεί στο σύστημα. Όπως στην πολιτική (s, S), το S παίζει το ρόλο του μέγιστου αποθέματος αλλα το s είναι η *αποθεματική θέση ασφαλείας*. Στο τέλος κάθε μήνα γίνεται καταμέτρηση του αποθέματος και παραγγέλλονται

$$Q = \begin{cases} S - x - \Sigma Q & x + \Sigma Q < s \\ 0 & x + \Sigma Q \geq s. \end{cases}$$

Δηλαδή, αν κατά την επιθεώρηση η αποθεματική θέση x + ΣQ ευρεθεί κάτω από την τιμή s, τότε παραγγέλλονται τόσα κομμάτια όσα θα γέμιζαν την αποθήκη εκείνη τη στιγμή υποθέτοντας ότι θα έφθαναν τότε και οι προηγούμενες αποστολές (ΣQ) που πιθανόν να εκκρεμούν.

Μία μονάδα αποθηκευμένου προϊόντος για έναν μήνα κοστίζει h. Το h περιλαμβάνει ένα κόστος συντήρησης αποθέματος και ένα κόστος δέσμευσης κεφαλαίου που, αν δεν αγοράζαμε το προϊόν, θα μπορούσε να επενδυθεί με κάποιο ασφαλές επιτόκιο για έναν μήνα. Το κόστος έλλειψης αγαθού ανεξάρτητο από το μέγεθος και τη διάρκεια της έλλειψης είναι Π (κόστος δυσφήμισης). Αυτό το κόστος επιβαρύνει την εταιρεία κάθε φορά που η παραγγελία κάποιου πελάτη *δεν καλύπτεται πλήρως.* Το κόστος έλλειψης μίας μονάδας για ένα μήνα είναι b. Το b έχει μία χρηματοοικονομική συνισώσα: αν υπήρχε προϊόν διαθέσιμο και πωλούνταν, τότε το κέρδος από την πώληση θα μπορούσε να επενδυθεί με ασφαλές επιτόκιο για έναν μήνα. Μία άλλη συνιστώσα του b είναι η έκπτωση ανά μήνα καθυστέρησης που μπορεί να προσφέρεται στον πελάτη για κάθε προϊόν που δεν είναι ετοιμοπαράδοτο. Η χωρητικότητα της αποθήκης είναι S. Το κόστος ενοικίασης ή/και συντήρησης αποθηκευτικού χώρου είναι $u_1S + u_2$ ανά μήνα. Στο κόστος αυτό μπορεί αντί για ενοικίαση να περιλαμβάνονται οι μηνιαίες δόσεις αποπληρωμής του δανείου που χρησιμοποιήθηκε για την κατασκευή αποθήκης. Κάνουμε τη λογική υπόθεση ότι $u_1 > 0$ δηλαδή οι μικρές αποθήκες έχουν μικρό κόστος.

Για τη λειτουργία του συστήματος είναι υποψήφιες m πολιτικές ελέγχου,

$$[s_1, S_1], [s_2, S_2], \dots , [s_m, S_m]$$

με $s_m < S_m$. Η εταιρεία θέλει να εύρει τη βέλτιστη πολιτική ώστε, ξεκινώντας με ένα αρχικό καθαρό απόθεμα x_0 (θετικό, αρνητικό, ή μηδέν) και καμμία παραγγελία αναπλήρωσης υπό μεταφορά από τον προμηθευτή (ΣQ = 0), να ελαχιστοποιήσει το συνολικό κόστος λειτουργίας για μια αρκετά μεγάλη περίοδο N μηνών. Το κόστος αυτό ισούται με το άθροισμα του κόστους προμήθειας προϊόντων, αποθήκευσης, έλλειψης αγαθών, και ενοικίασης εγκαταστάσεων.

Υποθέτουμε ότι το αρχικό απόθεμα x_0 χωρά σε όλες τις αποθήκες, δηλαδή

$$x_0 \leq \min_i S_i.$$

Αντιθέτως, το x_0 δεν περιορίζεται από κάτω όρια. Στην αρχή επιτρέπεται να έχουμε ακόμη και

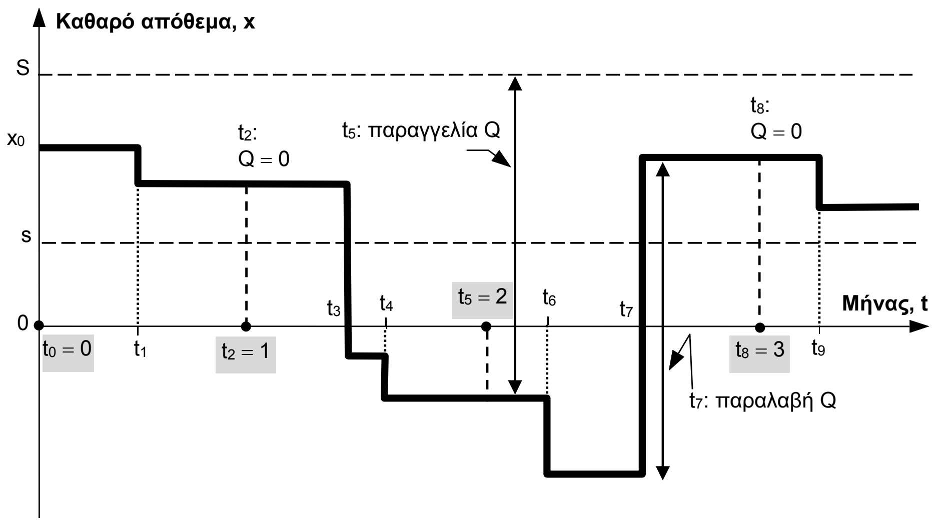
$$x_0 < s_i \text{ για κάποια } i = 1, \ldots, m$$

χωρίς προηγούμενες παραγγελίες αναπλήρωσης αποθέματος. Στις περιπτώσεις αυτές, με την πρώτη επιθεώρηση θα γίνει οπωσδήποτε μία παραγγελία ανεξάρτητα από το αν συμβαίνει ή όχι ζήτηση στη διάρκεια το πρώτου μήνα λειτουργίας.

Αν προσομοιώσουμε το σύστημα για μία περίοδο N μηνών και για κάθε πολιτική $[s_i, S_i]$ θα εύρουμε το συνολικό κόστος λειτουργίας. Το μικρότερο κόστος θα αντιστοιχεί στη βέλτιστη πολιτική.

3.2.2 Μοντέλο διακριτών γεγονότων

Στο Σχήμα 3.2 φαίνεται η εξέλιξη του καθαρού αποθέματος x για ένα σενάριο λειτουργίας του αποθεματικού συστήματος.



Σχήμα 3.2: Το απόθεμα συναρτήσει του χρόνου: μειώνεται όταν υπάρχει ζήτηση και αυξάνεται όταν γίνεται παραλαβή. Το απόθεμα δεν αλλάζει όταν γίνεται επιθεώρηση (χρόνοι t_2, t_5, t_8).

Η εξέλιξη του αποθέματος καθορίζεται από την εμφάνιση των εξής γεγονότων:

γεγονός 1: ζήτηση προϊόντος από τους καταναλωτές. Τα γεγονότα αυτά συνεπάγονται απότομες μειώσεις στο απόθεμα (χρόνοι t_1, t_3, t_4, t_6, t_9 στο σχήμα).

γεγονός 2: παραλαβή φορτίου προϊόντων από τον προμηθευτή. Συνεπάγεται απότομες αυξήσεις στο απόθεμα (χρόνος t_7).
γεγονός 3: καταμέτρηση αποθέματος στο τέλος κάθε μήνα και πιθανή παραγγελία στον προμηθευτή (χρόνοι t_2, t_5, t_8).

Η κατάσταση του συστήματος κάποια στιγμή t περιγράφεται από τις εξής μεταβλητές:

- Στάθμη αποθέματος x και λειτουργικό κόστος COST.
- Λίστα ΛQ των παραγγελιών Q που έχουν γίνει στον προμηθευτή και δεν έχουν αφιχθεί ακόμη καθώς και λίστα ΛX με τους χρόνους άφιξής των. Έστω Ε το πλήθος των αποστολών (φορτία με παραγγελίες αναπλήρωσης στον προμηθευτή) που είναι καθ' οδόν. Για κάθε εκκρεμή παραλαβή, $\varepsilon = 1, 2, \ldots, E$, το στοιχείο της λίστας $ΛX_\varepsilon$ περιέχει τον τυχαίο *χρόνο αφίξεώς* της και το στοιχείο $ΛQ_\varepsilon$ ισούται με την ποσότητα εμπορεύματος που θα παραληφθεί (ποσότητα που είχε παραγγελθεί στον προμηθευτή). Η παραγγελία με τον μικρότερο χρόνο είναι εκείνη που θα φθάσει πρώτη και αντιστοιχεί στο επόμενο γεγονός 2. Λόγω του ότι οι χρόνοι άφιξης είναι τυχαίοι, είναι πιθανόν κάποια παραγγελία που κάναμε πρόσφατα να αφιχθεί ενωρίτερα από κάποια άλλη που έγινε πιο παλιά παρόλο που η αποστολή της είχε ξεκινήσει ενωρίτερα. Με e θα συμβολίζουμε τον δείκτη της παραγγελίας που θα φθάσει νωρίτερα, δηλαδή

$$ΛX_e = \min \{ΛX_\varepsilon, \varepsilon = 1, \ldots, E\}.$$

- Μέγεθος παραγγελιών που εκκρεμούν

$$ΣQ = \sum_{\varepsilon=1}^{E} ΛQ_\varepsilon \ .$$

- Η λίστα με τους χρόνους επόμενων γεγονότων 1, 2 και 3: T_1, T_2, T_3. Ως T(2) λαμβάνουμε τον ενωρίτερο χρόνο άφιξης $ΛX_e = \min \{ΛX_\varepsilon, \varepsilon = 1, \ldots, E\}$.
- Ο χρόνος που συνέβη το προηγούμενο γεγονός στο σύστημα, τ.
- Η χρονική στιγμή N + 1 είναι 1 μονάδα μεγαλύτερη από τον ορίζοντα της λειτουργίας του συστήματος. Θα τη χρησιμοποιούμε ως χρόνο επόμενου γεγονότος, όταν κάποιο γεγονός είναι προσωρινά σε αναστολή και μη προγραμματισμένο να συμβεί. Λόγω του ότι το ρολόι t της προσομοίωσης δεν θα υπερβεί ποτέ το N, το N + 1 πρακτικά ισοδυναμεί με άπειρο χρόνο.

Στα διαστήματα (t_{k-1}, t_k) δεν μεταβάλλεται το απόθεμα. Αυτά τα διαστήματα είναι οι βάσεις ορθογωνίων παραλληλογράμμων τα οποία συνθέτουν το εμβαδό αποθέματος (που έχει μοναδιαίο κόστος h) και το εμβαδό ελλείμματος (με μοναδιαίο κόστος b), ανάλογα με το αν το απόθεμα είναι θετικό ή αρνητικό. Στον υπολογιστή δεν χρειάζεται να αποθηκεύουμε τους δείκτες k. Οι βάσεις των ορθογωνίων υπολογίζονται από τη διαφορά t – τ, όπου t είναι ο τρέχων χρόνος t_k και τ είναι ο χρόνος που έλαβε χώρα το *προηγούμενο* γεγονός, t_{k-1}.

Αλγόριθμος: *Προσομοίωση πολιτικής [s_i, S_i]*

1. Ανάγνωση δεδομένων

- πολιτική $s = s_i$, $S = S_i$ (στο εξής το i θα χρησιμοποιείται ως δείκτης γεγονότος)
- N = χρόνος (μήνες) προσομοίωσης του συστήματος, A = σταθερό κόστος παραγγελίας, C = κόστος μίας μονάδας προϊόντος και λοιπές παράμετροι: h, Π, b, u_1, u_2, x_0, α, β, μ, r, {(d_j, p_j) , j = 1, …, r}

2. Αρχικές τιμές

t = 0 = τρέχων χρόνος

$x = x_0$

COST = $N(u_1S+u_2)$ = κόστος χώρου αποθήκης για N μήνες.

Η λίστα παραγγελιών οι οποίες εκκρεμούν είναι κενή αφού στην αρχή δεν περιμένουμε άφιξη εμπορεύματος:

E = 0

ΣQ = 0

e = 0

Χρόνοι επόμενων γεγονότων:

Επόμενη άφιξη πελάτη θα γίνει τη στιγμή $T_1 = t - (\ln U)/\mu$

Στην αρχή δεν εκκρεμούν αποστολές: $T_2 = N + 1$

Επόμενη επιθεώρηση θα γίνει τη στιγμή $T_3 = 1$.

Επαναλαμβάνουμε τα βήματα 3, 4:

3. Εύρεση επόμενου γεγονότος ή περάτωση της προσομοίωσης

τ = t (ο χρόνος προηγούμενου γεγονότος)

Αναζητούμε τον χρόνο επόμενου γεγονότος: αρχικά υποθέτουμε ότι δεν θα συμβεί τίποτε ως το τέλος της προσομοίωσης και στη συνέχεια ψάχνουμε για γεγονότα που θα συμβούν πιο σύντομα:

t = N + 1

Για j = 1, 2, 3

Αν T(j) < t τότε

t = T(j)

Επόμενο γεγονός: i = j

ΤέλοςΑν

Επόμενο j

Αν $t > N$ τότε (το επόμενο γεγονός ξεπερνά τον ορίζοντα προσομοίωσης)

Φέρνουμε τον χρόνο στη χρονική στιγμή $t = N$

Εκτελούμε το υποπρόγραμμα ενημέρωσης **4α**.

Αποθηκεύουμε ή εκτυπώνουμε τα s, S, COST

Περατώνουμε την εκτέλεση του προγράμματος (διακοπή επαναλήψεων 3, 4)

ΤέλοςΑν

4. Εκτέλεση επόμενου γεγονότος i τη στιγμή t

α. Ενημέρωση

Αν $x > 0$ τότε

$COST = COST + (t – τ) x h$

Αλλιώς αν $x < 0$ τότε

$COST = COST + (t – τ) (–x) b$

ΤέλοςΑν

β. Μεταβολές

Αν $i = 1$ (ζήτηση) τότε

Γεννήτρια τυχαίας ποσότητας που ζητεί ο πελάτης: D

Το καθαρό απόθεμα μειώνεται κατά D: $x = x – D$

Αν $(x – D) < 0$ τότε

Επιβάρυνση με κόστος δυσφήμισης: $COST = COST + Π$

ΤέλοςΑν

Αλλιώς αν $i = 2$ (παραλαβή της αποστολής φορτίου e με τον ενωρίτερο χρόνο άφιξης)

$x = x + ΛQ_e$

Αφαιρούμε το στοιχείο e από τα διανύσματα ΛX και ΛQ και ανεβάζουμε τα e + 1, e + 2, …, μία θέση (αν e + 1 > E ο μεταγλωττιστής-compiler παρακάμπτει αυτομάτως τον βρόχο επαναλήψεων $j = e + 1, …, E$):

Για $j = e + 1, e + 2, …, E$

$ΛX_{j–1} = ΛX_j$

$ΛQ_{j–1} = ΛQ_j$

Επόμενο j

$E = E – 1$

Αν $E > 0$ τότε

$\Sigma Q = \Sigma Q - \Lambda Q_e$

αλλιώς (λίστα κενή)

$\Sigma Q = 0$ (ισοδύναμο του $\Sigma Q = \Sigma Q - \Lambda Q_e$, όμως επαναφέρουμε την αρχική συνθήκη γιατί οι προηγούμενες προσθαφαιρέσεις στο ΣQ μπορεί να έχουν μικρές αποκλίσεις λόγω των στρογγυλοποιήσεων που κάνει ο υπολογιστής)

ΤέλοςΑν

Αλλιώς (αν $i = 3$, τότε έχουμε επιθεώρηση και πιθανώς και παραγγελία, αλλά το απόθεμα δεν μεταβάλλεται)

Έλεγχος αν θα γίνει νέα παραγγελία (ο δείκτης order = 0, 1 δείχνει αν θα γίνει):

order = 0

Αν $x + \Sigma Q < s$ τότε

Νέα παραγγελία:

order = 1

$E = E + 1$

$\Lambda Q_E = S - x - \Sigma Q$

Στο εξής εκκρεμούν συνολικά: $\Sigma Q = \Sigma Q + \Lambda Q_E = S - x$

Γεννήτρια τυχαίου χρόνου άφιξης: $\Lambda X_E = t + \alpha + (\beta - \alpha)U$

Νέο κόστος: $COST = COST + A + C\ \Lambda Q_E$

ΤέλοςΑν

γ. Προγραμματισμός

Αν $i = 1$ τότε

Επόμενος πελάτης θα έλθει τη στιγμή $T(1) = t - (\ln U)/\mu$

Αλλιώς αν $i = 2$ τότε

Από τις παραγγελίες που εκκρεμούν ευρίσκουμε εκείνη με το μικρότερο χρόνο άφιξης

$T_2 = N + 1$

Αν $E > 0$ τότε

Για $j = 1, \ldots, E$

Αν $\Lambda X_j < T_2$ τότε

$T_2 = \Lambda X_j$

Επόμενη παραλαβή: e = j

ΤέλοςΑν

Επόμενο j

ΤέλοςΑν

Αλλιώς (έγινε καταγραφή αποθέματος στο τέλος του μήνα)

Η επόμενη καταγραφή θα γίνει τον μήνα $T_3 = t + 1$

Αν order = 1 τότε

Ελέγχουμε αν η νέα παραγγελία Ε θα φθάσει πριν τον χρόνο T_2

Αν $ΛX_E < T_2$ τότε

Η νέα e είναι τώρα η Ε και προγραμματίζεται νέος χρόνος T_2

e = E

$T_2 = ΛX_E$

ΤέλοςΑν

ΤέλοςΑν

ΤέλοςΑν

5. Επανάληψη από το βήμα 3.

Ο ανωτέρω αλγόριθμος επαναλαμβάνεται για όλες τις εναλλακτικές πολιτικές ελέγχου $[s, S] = [s_1, S_1], [s_2, S_2], \ldots, [s_m, S_m]$. Σε κάθε επανάληψη, συγκρίνεται το κόστος COST της πολιτικής [s, S] που προσομοιώνεται κάθε φορά με το κόστος της οικονομικότερης (καλύτερης) πολιτικής ως εκείνη την επανάληψη, έστω $COST^*$. Αν το COST ευρεθεί μικρότερο, τότε η βέλτιστη πολιτική αποθηκεύεται ως $COST^* = COST$, $s^* = s$, $S^* = S$. Όταν ολοκληρωθεί η προσομοίωση και σύγκριση όλων των πολιτικών αναπλήρωσης, η πολιτική [s, S] είναι εκείνη με το ελάχιστο κόστος.

3.3 ΣΕΙΡΙΑΚΗ ΔΙΑΧΕΙΡΙΣΗ ΛΙΣΤΩΝ

3.3.1 Αναζήτηση, αφαίρεση και πρόσθεση στοιχείων

Στην προηγούμενη παράγραφο χρησιμοποιήσαμε *λίστες* (διατεταγμένα σύνολα, διανύσματα) των χρόνων άφιξης και των αντίστοιχων ποσοτήτων παραγγελιών ΛΧ και ΛQ. Θα ιδούμε πώς κάνουμε αναζήτηση στοιχείων σε τέτοιες λίστες και πώς προσθέτουμε ή αφαιρούμε στοιχεία.

Έχουμε δύο διατεταγμένα σύνολα αριθμών (λίστες)

$$K = (K_1, \ldots, K_m)$$

$$L = (L_1, \ldots, L_n)$$

και δύο αριθμούς x και y. Υποθέτουμε ότι κάποιο $K_i = x$ και κάποιο $L_j = y$ αλλά αρχικά δεν γνωρίζουμε τις θέσεις τους i και j. Ζητούμε τις νέες λίστες που θα προκύψουν αν *αφαιρέσουμε* από την K το πρώτο στοιχείο ίσο με x που θα εύρουμε σε αυτήν και ακολούθως το *τοποθετήσουμε* στη λίστα L *ακριβώς πριν* από το πρώτο στοιχείο ίσο με y που θα εύρουμε στην L.

Αλγόριθμος: *Αναζήτηση, αφαίρεση και πρόσθεση στοιχείων*

1. Εύρεση του στοιχείου της K που ισούται με x, αφαίρεση από K, αναρρίχηση επόμενων στοιχείων της K μία θέση επάνω:

ifound = 0

i = 0

Όσο ifound = 0 επαναλαμβάνουμε τα ακόλουθα

    i = i + 1

    Αν $K_i = x$ τότε

        ifound = 1

        αναρρίχηση μία θέση:

        Για p = i + 1, …, m

            $K_{p-1} = K_p$

        Επόμενο p

        m = m – 1

    ΤέλοςΑν

Συνέχιση επαναλήψεων

2. Εύρεση του στοιχείου της L που ισούται με y, προσθήκη του x πριν από το y, κατάβαση επόμενων στοιχείων της L μία θέση κάτω:

jfound = 0

j = 0

Όσο jfound = 0 επαναλαμβάνουμε τα ακόλουθα

    j = j + 1

    Αν $K_j = y$ τότε

        jfound = 1