

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ
ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΤΟΜΕΑ «ΘΕΩΡΗΤΙΚΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΘΕΩΡΙΑΣ
ΣΥΣΤΗΜΑΤΩΝ ΚΑΙ ΕΛΕΓΧΟΥ»

ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΚΑΙ ΕΛΕΓΧΟΣ
ΕΦΟΔΙΑΣΤΙΚΩΝ ΑΛΥΣΙΔΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΔΙΚΤΥΩΝ
PETRI

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Όνοματεπώνυμο: Μπαλατσού Σπυριδούλα

Επιβλέπων: Τζιώνας Παναγιώτης

ΘΕΣΣΑΛΟΝΙΚΗ 2009

ΠΕΡΙΕΧΟΜΕΝΑ

0. Εισαγωγή.....	5
1. Κεφάλαιο 1: Εφοδιαστική αλυσίδα.....	6
1.1 Εισαγωγή στην εφοδιαστική αλυσίδα.....	6
1.2 Χρησιμότητα και παραδείγματα εφοδιαστικής αλυσίδας.....	7
1.2.1 Beer game.....	9
1.2.2 Εφοδιαστική αλυσίδα με χρήση συναρτήσεων.....	12
1.3 Bullwhip Effect.....	15
2. Κεφάλαιο 2: Petri Nets και Colour PN.....	18
2.1 Petri Nets.....	18
2.1.1 Εισαγωγή στα Petri Nets.....	18
2.1.2 Φορμαλισμός των PN.....	19
2.1.3 Επέκταση των PN.....	20
2.1.3.1 Μαρκάρισμα.....	20
2.1.3.2 Εκτέλεση.....	22
2.1.3.3 State Space.....	24
2.1.4 Ιδιότητες των PN.....	25
2.1.4.1 Εκτελεσιμότητα.....	25
2.1.4.2 Περιοριστικότητα (boundedness).....	26
2.1.5 Μοντελοποίηση των PN.....	27
2.1.5.1 Finite state machine.....	27
2.1.5.2 Παραγωγός- πελάτης σε προτεραιότητα.....	29
2.1.6 Ανάλυση- άλλοι τρόποι έκφρασης των PN.....	30
2.2 Colour Petri Nets.....	32
2.2.1 Φορμαλισμός των CPN.....	32
2.2.2 Προσομοίωση και καταγραφή δεδομένων στα CPN.....	43
2.2.2.1 Χειροκίνητη προσομοίωση.....	43

2.2.2.2	Ανάλυση προσομοίωσης (simulation report).....	44
2.2.2.3	Διαγράμματα.....	46
2.2.2.4	State Space και occurrence graphs.....	50
2.2.3	Timed CPN.....	65
2.2.4	Ιεραρχικά Χρωματισμένα δίκτυα Petri.....	68
3.	Κεφάλαιο 3: Εφοδιαστική αλυσίδα σχεδιασμένη από CPN.....	74
3.1	Εισαγωγή – περιγραφή μοντέλου.....	74
3.2	Υπολογισμός κόστους – Συνεχής ανατροφοδότηση.....	83
3.2.1	Aggressive ordering.....	83
3.2.2	Moving average.....	87
3.2.3	Exponential smoothing.....	89
3.3	Υπολογισμός κόστους – Στοχαστική ανατροφοδότηση.....	92
4.	Κεφάλαιο 4: Ανάλυση δεδομένων.....	95
4.1	Συνεχής ανατροφοδότηση.....	95
4.1.1	Κατάσταση αποθεμάτων.....	95
4.1.2	Μελέτη ποσοτήτων back order.....	100
4.1.3	Ανάλυση κόστους.....	102
4.2	Στοχαστική και καθορισμένου χρόνου ανατροφοδότηση.....	107
4.2.1	Κατάσταση αποθεμάτων.....	107
4.2.2	Μελέτη ποσοτήτων back orders.....	111
4.2.3	Ανάλυση κόστους.....	113
5.	Κεφάλαιο 5: Ανταγωνιστικό μοντέλο.....	118
5.1	Σχεδίαση ανταγωνιστικού μοντέλου.....	118
5.2	Ανταγωνιστικό μοντέλο με συνάρτηση κόστους.....	123
5.2.1	Εισαγωγή της συνάρτησης κόστους.....	123
5.2.2	Εναλλακτικές πολιτικές.....	126
6.	Επίλογος.....	129
7.	Παράρτημα.....	130
8.	Βιβλιογραφία.....	137

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια παρατηρείται ραγδαία ανάπτυξη της τεχνολογίας η οποία δημιουργεί νέα δεδομένα σε πολλούς τομείς.

Οι τομείς που δέχτηκαν τις μεγαλύτερες αλλαγές προκειμένου να προσαρμοστούν στις υπάρχουσες συνθήκες είναι αυτοί που έχουν να κάνουν με τον ίδιο τον άνθρωπο. Η συνεχώς αναπτυσσόμενη τεχνολογία, είχε σαν αποτέλεσμα την μεγάλη αύξηση της παραγωγής σε ελάχιστο χρόνο. Σήμερα, ο ανταγωνισμός διογκώνεται ενώ έχουν δημιουργηθεί νέα οικονομικά δεδομένα. Ταυτόχρονα, η ταχεία διάδοση της πληροφορίας λειτούργησε ανασταλτικά για τον καταναλωτή, μιας και μπορεί πλέον να ενημερώνεται και να επιλέγει με το πάτημα ενός κουμπιού.

Όλη αυτή η εξέλιξη, έφερε τις διάφορες μονάδες παραγωγής αντιμέτωπες με προβλήματα, όπως η προώθηση ανταγωνιστικών προϊόντων μαζικής παραγωγής, σε συνδυασμό με την ελαχιστοποίηση του κόστους κατασκευής τους και την διακίνηση αυτών σε δελεαστική τιμή. Δημιουργήθηκαν όμως πολύπλοκα ζητήματα, όπως η σωστή αξιολόγηση της ποιότητας-ποσότητας σε σχέση με το χρόνο κατασκευής του και την τιμή προώθησή του. Αναγκαίο είναι λοιπόν, η διερεύνηση και η δημιουργία κατάλληλων μαθηματικών μοντέλων, ώστε μία μονάδα παραγωγής να μπορέσει να βελτιστοποιήσει τους συντελεστές της ποιότητας-ποσότητας σε συνδυασμό με την ελαχιστοποίηση του κόστους.

ΚΕΦΑΛΑΙΟ 1^ο

Εφοδιαστική αλυσίδα

1.1 Εισαγωγή στην εφοδιαστική αλυσίδα

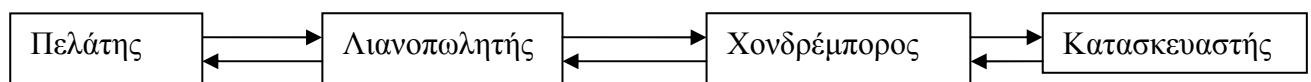
Ο όρος "εφοδιαστική αλυσίδα" (supply chain) περιγράφει το πλέγμα διαδικασιών που απαιτούνται ώστε ένα προϊόν να περάσει από τη φάση της παραγωγής στη φάση της κατανάλωσης. Μεταξύ των διεργασιών αυτών ξεχωρίζει η παραγωγή, η αποθήκευση, η διακίνηση και η διάθεση του προϊόντος.

Η παραγωγή είναι διαδικασία που πραγματοποιείται μόνο από τον κατασκευαστή, η αποθήκευση από όλα τα μέλη της αλυσίδας εκτός από τον πελάτη, ενώ η διακίνηση και η διάθεση από τα μεσαία της μέλη. Γι' αυτό το λόγω το πρόβλημα της αποθήκευσης των αποθεμάτων είναι πιο πολύπλοκο και διατίθεται για βαθύτερη έρευνα [6].

Η λειτουργικότητα μιας εφοδιαστικής αλυσίδας, στηρίζεται στο σωστό σχεδιασμό, την εκτέλεση και στον έλεγχο όλων των δραστηριοτήτων της. Παλιότερα, οι εφοδιαστικές αλυσίδες προσέδιδαν αξία μέσω της αποδοτικότητάς (efficiency) τους και της δυνατότητας επίτευξης χαμηλών τιμών. Στην σημερινή εποχή, πρέπει να προβάλουν μια νέα αξία μέσω της ελαστικότητάς τους (flexibility). Ο σχεδιασμός της θα πρέπει να είναι ευέλικτος ώστε να αντιδρά σε απρόσμενες αλλαγές όσο αφορά την διανομή, την διαθεσιμότητα και την αξία του υλικού και σε χρονοκαθυστερήσεις. Επίσης θα πρέπει να είναι ικανή ώστε να χρησιμοποιηθεί από τις σύγχρονες τεχνολογίες δικτύων και επικοινωνιών με σκοπό τον προγραμματισμό και τον άμεσο έλεγχο απόδοσής της. Απώτερος σκοπός της χρήσης της εφοδιαστικής αλυσίδας είναι η μεγιστοποίηση της αποδοτικότητας των διεργασιών κατά μήκος της.

Το πρόβλημα της βελτιστοποίησης πλέκεται ακόμα περισσότερο, μιας και τα στάδια από τα οποία περνάει ένα προϊόν (πελάτης-λιανέμπορος-χονδρέμπορος-κατασκευαστής) είναι αλληλοεξαρτώμενα και αλληλεπιδρώμενα.

Για παράδειγμα, ο λιανέμπορος θα πρέπει να έχει υπόψη του το αίτημα του πελάτη ως προς το πλήθος την ποιότητα και την ποσότητα, την κατάσταση των αποθεμάτων του, τις προσφορές που θα δέχεται από τους μεσολαβητές, οι οποίοι με τη σειρά τους θα πρέπει να προβλέπουν, να επιλέγουν ανάλογα με τις απαιτήσεις των μελών που βρίσκονται στα χαμηλότερα στρώματα της εφοδιαστικής αλυσίδας (Σχήμα 1).



Σχήμα 1. Εφοδιαστική αλυσίδα παραγωγής.

Γραφικά, μία εφοδιαστική αλυσίδα, δεν διαφέρει από ένα κατευθυνόμενο γράφημα, όπου τα βέλη με την δεξιά φορά δηλώνουν τις παραγγελίες του κάθε μέλους στο επόμενο της ενώ τα βέλη με αντίθετη φορά δείχνουν τις παραλαβές και την διακίνηση του προϊόντος. Στο σύστημά μας λοιπόν, δεν υπάρχει γενικός συντονιστής αφού κάθε κόμβος παίρνει αποφάσεις μεμονωμένα. Οι αποφάσεις όμως αυτές είναι άμεσα συνδεδεμένες από τις πληροφορίες των γειτονικών κόμβων. Επίσης υπάρχει είσοδος και έξοδος (αρχική παραγγελία και τελική ποσότητα του προϊόντος) οι οποίες βρίσκονται στην κατάσταση του πελάτη. Δεν υπάρχει αμφιβολία λοιπόν ότι πρόκειται για ένα πλήρως οργανωμένο δίκτυο.

1.2 Χρησιμότητα και παραδείγματα εφοδιαστικής αλυσίδας

Η εφοδιαστική αλυσίδα αποτελεί το παλιότερο κομμάτι της Διοικητικής Επιστήμης. Τα τελευταία χρόνια κυκλοφορούν πολυάριθμα άρθρα που αφορούν την μοντελοποίηση κάποιου προβλήματος ή παιχνιδιού με την βοήθεια των supply chains.

Ο Van der Vorst εκμεταλλεύτηκε τη δυναμική των SC και σχεδίασε αλυσίδες οι οποίες εξυπηρετούσαν ποικίλα σενάρια ως προς την οργάνωση και τη διοίκηση μιας μονάδος. Επιπρόσθετα, υπήρχε η δυνατότητα να επαναπροσδιοριστούν και να εισαχθούν κάποια δεδομένα χωρίς να χρειαστεί να σχεδιαστεί η αλυσίδα από την αρχή. Η μεγάλη τους ισχύ είχε να κάνει κυρίως με την ανάδειξη του καλύτερου σεναρίου με τη βοήθεια των εργαλείων Petri Nets όπου θα αναλυθούν λεπτομερώς στη συνέχεια. Η δυνατότητα συνεργασίας και αλληλεπίδρασης τέτοιων αλυσίδων, συνέβαλλε στην αποτελεσματική συνένωση πολλών μονάδων μαζί.

Στη συνέχεια, η χρήση των εφοδιαστικών αλυσίδων διευρύνθηκε και σε προβλήματα διαχείρισης χρόνου και καλύτερης εξυπηρέτησης σε ουρές. Επίσης βοήθησαν στη μελέτη και ανάλυση διάφορων εργαλείων και λογισμικών όπως PN, Stochastic PN, CIMOCA χάρις τον απλό σχεδιασμό τους από μία μηχανή και την εύκολη μοντελοποίηση τους [7].

Για τον σκοπό αυτό, χρησιμοποιήθηκαν κάποια λογιστικά παιχνίδια όπου προσομοίωναν ένα πραγματικό πρόβλημα με προσαρμοσμένα δεδομένα ώστε να μπορέσουν να αναλυθούν πιο εύκολα από τον χρήστη. Χαρακτηριστικά παραδείγματα τέτοιων παιχνιδιών είναι το kanban game, distribution game και το beer game. Στο πρώτο, ο παίχτης ελέγχει ένα κέντρο παραγωγής τεσσάρων προϊόντων όπου ανάλογα με τη ζήτηση, θα χρειαστεί να ρυθμίσει κάποιες παραμέτρους (προσωπικό για το κάθε προϊόν, χρόνο κατασκευής, σωστή διαχείριση πρώτων υλών) ώστε να αυξήσει το κέρδος του. Στο παιχνίδι του διανομέα, (distribution game) ο παίχτης είναι ένα κομμάτι της εφοδιαστικής του σχήματος 1, με τη διαφορά ότι υπάρχουν δύο λιανέμποροι (retailer) και ότι υπάρχει ανώτερο όριο στη παροχή προϊόντων. Για αυτό θα πρέπει να αποφασίσει, πότε και πόση ποσότητα θα παραγγείλει από τον προμηθευτή και θα παραδώσει στους λιανέμπορους. Τα δύο αυτά παιχνίδια αποτελούν διαφορετικά σενάρια μιας απλής εφοδιαστικής αλυσίδας η οποία αναλύεται εκτενέστερα στην επόμενη παράγραφο με τη βοήθεια του beer game [8].

1.2.1 Beer game

Ένα αρκετά δημοφιλές παιχνίδι που χρησιμοποιήθηκε η εφοδιαστική αλυσίδα είναι το beer game. Κάθε ομάδα του παιχνιδιού χωρίζεται σε τέσσερα μέρη: Τον λιανέμπορο (retailer), τον χονδρέμπορο (wholesaler), τον διανομέα (distributor) και τον κατασκευαστή (factory). Η αλυσίδα περιέχει ένα μέλος παραπάνω από αυτήν του σχήματος 1. Ο πελάτης δίνει μια παραγγελία στον retailer και αυτός πουλά από το απόθεμά του. Έτσι δημιουργείται ένα έλλειμμα στις αποθήκες το οποίο θα αποτελεί το stock του retailer για την επόμενη περίοδο. Αν το απόθεμα είναι μικρότερο από την παραγγελία, τότε παραδίδει άμεσα τη ποσότητα από τις αποθήκες και η υπόλοιπη ποσότητα θα παραδοθεί όταν παραλάβει και ο ίδιος την παραγγελία του από το ανώτερο μέλος της αλυσίδας. Η διαφορά της παραγγελίας που πήρε με αυτή που παραδόθηκε, όταν είναι θετική (συνήθως λόγω ελλείμματος στις αποθήκες) αποτελεί το backorder. Ο λιανέμπορος εκτελεί την παραγγελία του ανάλογα με το απόθεμά του στο επόμενο μέλος της αλυσίδας και η διαδικασία συνεχίζεται μέχρι τον κατασκευαστή. Ο κατασκευαστής με τη σειρά του παίρνει πρώτες ύλες από τον προμηθευτή (supplier). Το κόστος της διαδικασίας για το κάθε μέλος της αλυσίδας ισούται με το άθροισμα του κόστους αποθήκευσης των αποθεμάτων και του κόστους της ποσότητας του backorder αν φυσικά υπάρχει. Στην πιο απλή εκδοχή του παιχνιδιού δεν υπολογίζεται το κόστος παραγγελίας και μεταφοράς. Σκοπός του παιχνιδιού είναι κάθε ομάδα να ελαχιστοποιήσει το συνολικό κόστος της αλυσίδας στην οποία συμμετέχουν. Όλα τα μέλη της κάθε ομάδας μπαίνουν στο παιχνίδι με τα ίδια χρήματα. Κατά την διάρκεια του κάθε γύρου, ο κάθε παίχτης δεν μπορεί να γνωρίζει για την κατάσταση των αποθεμάτων και τα backorder των άλλων μελών αλλά και τις παραγγελίες των επόμενων μελών παρά μονάχα τα δικά του και τις παραγγελίες που δέχεται από το αμέσως προηγούμενο μέλος. Μόνο ο retailer γνωρίζει την επόμενη παραγγελία του πελάτη.

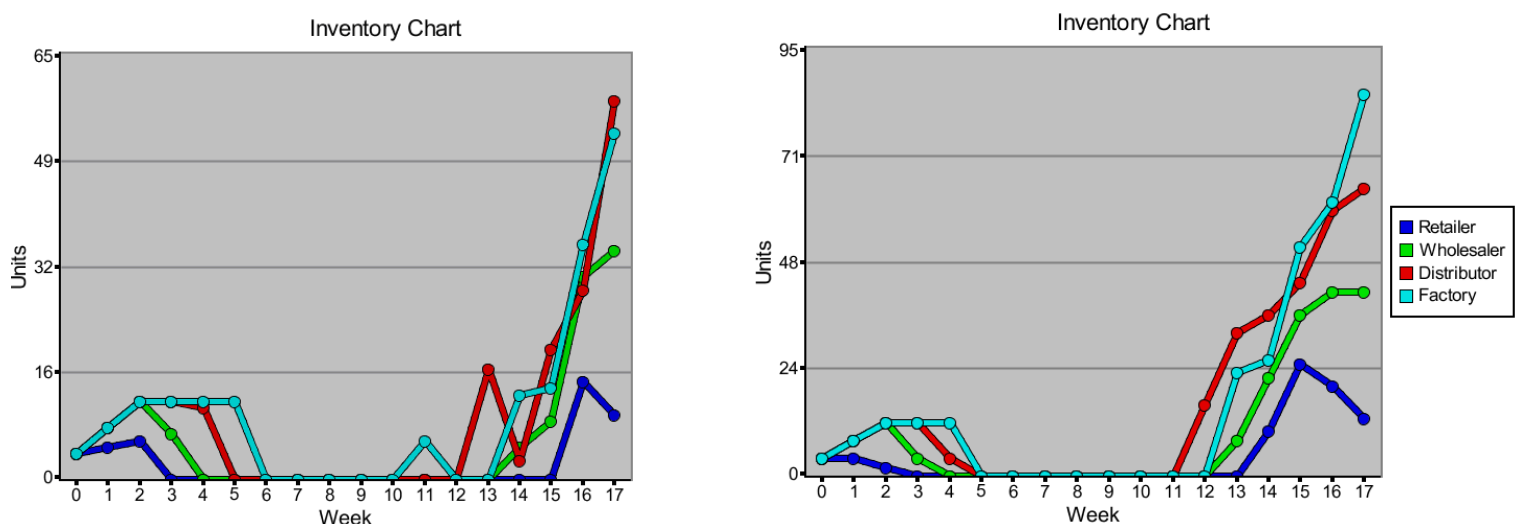
Αρχικά, όλα τα μέλη της αλυσίδας έχουν 12 κομμάτια στις αποθήκες και η αρχική παραγγελία του πελάτη θα είναι πάντα 4 όπως και τις επόμενες τρεις εβδομάδες. Κάθε βήμα συμβολίζει και το πέρασμα μιας βδομάδας. Στις επόμενες βδομάδες το αίτημα του πελάτη διαφέρει. Το κάθε μέλος θα πρέπει να προβλέπει τις ανάγκες του για τουλάχιστον τρεις βδομάδες αλλά και τις πιθανές παραγγελίες του προηγούμενου μέλους. Ένας έμπειρος παίχτης θα καθορίσει την παραγγελία του ανάλογα με μία εκτίμηση που μπορεί να κάνει στο backorder του επόμενου μέλους [7].

Στο παρακάτω παράδειγμα συμμετέχουν δύο ομάδες. Το κόστος αποθήκευσης είναι 0,5\$ το κομμάτι και το κόστος αποθήκευσης είναι 1\$ το κομμάτι ανά βδομάδα. Τα beergame που κυκλοφορούν στο διαδίκτυο είναι σχεδιασμένα έτσι ώστε ο πελάτης να δίνει παραγγελίες μέχρι το πολύ 52 βδομάδες. Στο παράδειγμά μας, ο χρόνος τελειώνει στις 17 βδομάδες. Παρακάτω περιγράφονται με πίνακες και ιστογράμματα οι κινήσεις δύο ομάδων.

Retailer Report	Wholesaler Report	Distributor Report	Factory Report	System Report	
Role	Inventory Mean	Inventory STD	Order Mean	Order STD	Cost
Retailer	2.12	4.4	5.53	4.9	156
Wholesaler	6.29	10.83	7	8.49	299.5
Distributor	10	15.16	9.71	12.65	392
Factory	10.47	14.32	13.94	13.67	729.5
				SYSTEM COST	1,577

Retailer Report	Wholesaler Report	Distributor Report	Factory Report	System Report	
Role	Inventory Mean	Inventory STD	Order Mean	Order STD	Cost
Retailer	4.35	7.87	5.53	4.9	123
Wholesaler	10.12	15.19	6.88	8.59	356
Distributor	16.82	21.73	9.94	11.99	496
Factory	17.06	25.37	14.35	13.74	835
				SYSTEM COST	1,810

Σχήμα 2: Συγκριτικός πίνακας των παραγγελιών, των αποθεμάτων και του συνολικού κόστους για το κάθε μέλος της ομάδας 1 και 2 αντίστοιχα.



Σχήμα 3: Ιστογράμματα των αποθεμάτων των μελών της αλυσίδας της ομάδας 1 και 2 αντίστοιχα.

Από τους πίνακες 2 φαίνεται η διαφορά στο κόστος του κάθε μέλους των δύο ομάδων.

Έτσι θα έχουμε:

Για την ομάδα 1: Retailer 156\$, wholesaler 299,5 \$ distributor 322\$ factory 729,5\$.

Για την ομάδα 2: Retailer 123\$, wholesaler 356 \$ distributor 496\$ factory 835\$.

Οι παραγγελίες έχουν ως εξής:

Για την ομάδα 1: Retailer (0,5,12,0,8,5,6,4,12,10,7,16,6,3,0,0,0)

Wholesaler(0,0,1,30,7,8,3,5,3,16,11,7,22,4,2,0,0)

Distributor(0,0,0,0,30,30,30,5,1,0,19,13,8,29,0,0,0)

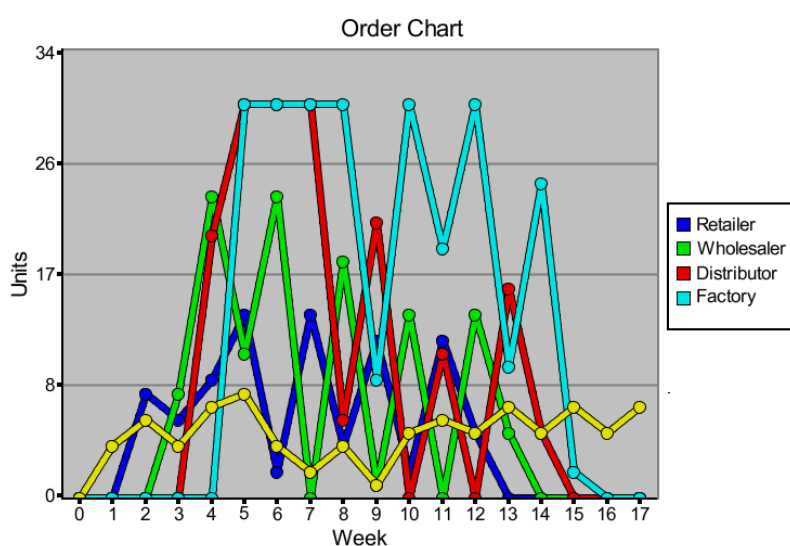
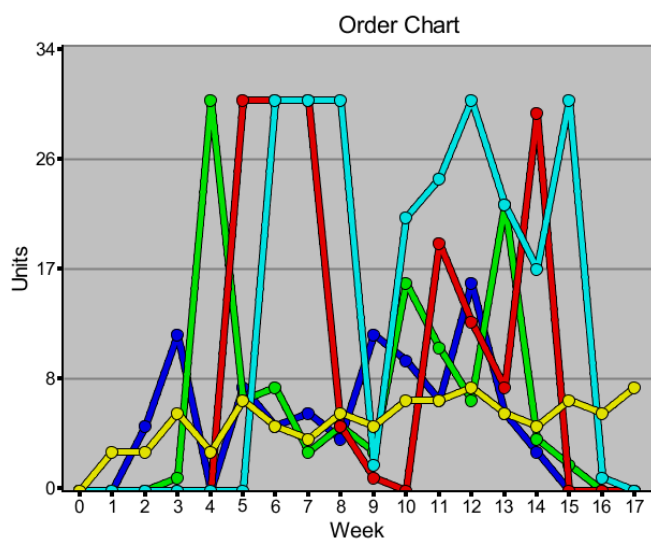
factory (0,0,0,0,0,30,30,30,2,21,24,30,22,17,30,1)

Για την ομάδα 2: Retailer (0,8,6,9,14,2,14,4,12,2,12,5,0,0,0,0,0)

Wholesaler(0,0,8,23,11,23,0,18,1,14,0,14,5,0,0,0)

Distributor(0,0,0,20,30,30,30,6,21,0,11,0,16,5,0,0,0)

factory (0,0,0,0,0,30,30,30,30,9,30,19,30,10,24,2,0,0)



Σχήμα 4: Ιστογράμμο παραγγελιών των μελών της ομάδας 1 και της 2 αντίστοιχα.

Από τις παραγγελίες φαίνεται ότι όλα τα μέλη της δεύτερης ομάδας έκαναν μεγαλύτερες παραγγελίες από ότι η πρώτη, γι' αυτό και τα αποθέματά τους είναι και μεγαλύτερα. Έτσι, λογικό είναι στη πρώτη ομάδα να χρειαστούν λιγότερα χρήματα για να ικανοποιήσουν το αίτημα του πελάτη. Νικήτρια είναι η πρώτη ομάδα με συνολικό κόστος 1.577\$ απέναντι από την ομάδα 2 η οποία ξόδεψε 1.810\$.

Για να μελετηθεί μία εφοδιαστική αλυσίδα μέσω του beer game θα πρέπει να είναι διαθέσιμα τουλάχιστον οχτώ άτομα για να γίνουν δύο ομάδες των τεσσάρων. Επειδή δεν είναι πολλές φορές κάτι τέτοιο εφικτό, τα beer game μέσω διαδικτύου παρέχουν την δυνατότητα στον παίχτη να πάρει μία από τις τέσσερις θέσεις και τις υπόλοιπες ο ίδιος ο υπολογιστής. Το κάθε παιχνίδι μπορεί να αποθηκευτεί και στη συνέχεια ο χρήστης να ξαναπαίξει από άλλη θέση ή δίνοντας διαφοροποιημένες παραγγελίες. Έτσι μπορεί να συγκρίνει τα αποτελέσματα των δύο παιχνιδιών. Το ζητούμενο εδώ είναι με πιο τρόπο επιλέγει ο υπολογιστής να παραγγείλει [9]. Σε μερικές εκδόσεις του παιχνιδιού, ο χρήστης είναι αυτός που θα επιλέξει τη στρατηγική με την οποία θα παίξει ο υπολογιστής. Έτσι δημιουργούνται ποικίλα και ενδιαφέροντα σενάρια.

1.2.2 Εφοδιαστική αλυσίδα με τη βοήθεια συναρτήσεων

Ένας αποτελεσματικός τρόπος για να μπορέσει να μοντελοποιηθεί μία εφοδιαστική αλυσίδα., είναι να δίνει κάθε μέλος τις παραγγελίες με τη βοήθεια κατάλληλων συναρτήσεων. Με αυτό τον τρόπο θα μπορέσουμε να σταθεροποιήσουμε το κόστος μιας και η επιλογή της ποσότητας των παραγγελιών δεν θα είναι τυχαία, όπως στο beer game. Επίσης η πρόβλεψη των επόμενων παραγγελιών θα είναι δυνατή σε μεγαλύτερο βαθμό και έτσι θα αποφύγουμε τις μεγάλες διαφορές των αποθεμάτων και της δημιουργίας backorder σε κάθε βήμα της προσομοίωσης.

Κατά την εκτεταμένη διάρκεια της έρευνας και αποτίμησης επιχειρηματικών εφαρμογών σε πληθώρα τεχνικών μοντελοποίησης προγνώσεων, ένα σύνολο αυτών έχουν αναδειχθεί ως οι πλέον χρήσιμες και ευρύτερα χρησιμοποιούμενες.

Μεταξύ αυτών η **aggressive order**. Ο χρήστης εισάγει στους κόμβους επικοινωνίας των μελών γραμμικές συναρτήσεις. Για παράδειγμα όλα τα μέλη της αλυσίδας εκτός τον

κατασκευαστή παραγγέλνουν p ποσότητα (ίδια με το αίτημα του προηγούμενου μέλους) αν δεν υπερβαίνει το απόθεμά του (n) και $2 \cdot p - n$ αν το υπερβαίνει. Έτσι το κάθε μέλος παραγγέλνει ακριβώς ό,τι έχει ζητηθεί από αυτόν, ή την ποσότητα αυτήν μαζί με τη διαφορά της από το απόθεμα ($p + p - n$). Ο *manufacturer* θα κατασκευάσει $n + p + k$ (όπου k η παραλαβή από τον προμηθευτή) αν η παραγγελία του *distributor* p είναι μεγαλύτερη από k ή $n + k$ αν $p < k$. Οι ποσότητες που θα στείλουν στο προηγούμενο μέλος θα είναι p αν το απόθεμα είναι μεγαλύτερο από το p και n αν $p > n$. Σε αυτό το παράδειγμα οι συναρτήσεις έχουν δύο κλάδους, για να καλύψουν τα πιθανά ενδεχόμενα.

Ιδιαίτερο ενδιαφέρον παρουσιάζει μία ε.α. όταν οι παραγγελίες της καθορίζονται με συναρτήσεις **κινητού μέσου όρου (moving average)**. Με τη χρήση μέσου όρου προσπαθούμε να μην αποκλίνουμε από τις παραγγελίες που συνήθως φτάνουν. Επειδή τα αιτήματα του πελάτη δεν είναι γνωστά εκ των προτέρων, σε κάθε βήμα υπολογίζουμε τον μέσο όρο των παραγγελιών που προηγήθηκαν [10].

Αν θεωρήσουμε σειρά n δεδομένων και $\hat{x} = a$ η εκτίμηση πρόβλεψης για την επόμενη παραγγελία θα προσπαθήσουμε να ελαχιστοποιήσουμε το μέσο τετραγωνικό σφάλμα

$$\varepsilon^2 = \sum_{i=1}^n (x_i - \hat{x})^2 = \sum_{i=1}^n (x_i - a)^2,$$

$$\frac{d\varepsilon^2}{da} = -2 \sum_i (x_i - a) = 0 \Rightarrow a = \frac{1}{n} \sum_{i=1}^n x_i.$$

Αν θεωρήσουμε το n σταθερό, τότε $a = \frac{1}{n} \sum_{i=k+1}^{k+n} x_i$.

Για παράδειγμα αν οι παραγγελίες για 10 βδομάδες είναι 100,101,99,98,103,101,100,120,136,140 τότε η εκτίμηση της επόμενης παραγγελίας θα είναι

$$\hat{x}_{11} = a = \frac{1}{10} \sum_{i=1}^{10} x_i = 109,8. \text{ Αν η επόμενη πραγματική παραγγελία είναι } x_{11} = 108 \text{ τότε } \hat{x}_{12} = 109,63.$$

Προβλέψεις επόμενων παραγγελιών πραγματοποιούνται και με τη βοήθεια συνάρτησης **εκθετικής εξομάλυνσης (exponential smoothing)**.

Στη περίπτωση αυτή, προσπαθούμε να ελαχιστοποιήσουμε το άθροισμα

$$e = \sum_{i=1}^n a^{n-i} (x_i - \beta)^2, \text{ όπου } 0 < a < 1. \text{ Με άλλα λόγια επισυνάπτουμε βάρη στο κάθε σφάλμα}$$

κατά τη σειρά $\alpha^{n-1}, \alpha^{n-2}, \dots, \alpha, 1$ που σημαίνει ότι οι πλέον πρόσφατες μετρήσεις έχουν μεγάλο βάρος ενώ οι παλιές εξασθενούν. Σκοπός μας είναι να ελαχιστοποιήσουμε το σφάλμα:

$$\frac{de}{d\beta} = 0 \Rightarrow -2 \sum_{i=1}^n \alpha^{n-i} (x_i - \beta)^2 = 0 \Rightarrow -2 \sum_{i=1}^n \alpha^{n-i} = \sum_{i=1}^n \alpha^{n-i} x_i$$

$$\text{και επειδή } \sum_{i=1}^n \alpha^{n-i} = 1 + \alpha + \dots + \alpha^{n-1} = \frac{1 - \alpha^n}{1 - \alpha}$$

$$\beta = \frac{1 - \alpha}{1 - \alpha^n} \sum_{i=1}^n \alpha^{n-i} x_i = D(n)$$

$$\text{Οπότε } D(n) = \frac{1 - \alpha}{1 - \alpha^n} \sum_{i=1}^n \alpha^{n-i} x_i + \frac{1 - \alpha}{1 - \alpha^n} x_n$$

$$= \frac{1 - \alpha^{n-1}}{1 - \alpha^n} \frac{1 - \alpha}{1 - \alpha^{n-1}} \alpha \sum_{i=1}^{n-1} \alpha^{n-1-i} x_i + \frac{1 - \alpha}{1 - \alpha^n} x_n$$

$$= \frac{1 - \alpha^{n-1}}{1 - \alpha^n} \alpha D(n-1) + \frac{1 - \alpha}{1 - \alpha^n} x_n.$$

Για $n \rightarrow \infty$ και $\alpha^n \rightarrow 0$ και άρα

$$D(n) = \alpha D(n-1) + (1 - \alpha)x_n$$

Για τη συνάρτηση αυτή το α ονομάζεται συντελεστής εξομάλυνσης.

Για παράδειγμα, αν οι παραγγελίες για τις 10 πρώτες βδομάδες είναι 100,101,99,98,103,101,100,120,136,140 (ίδιες με τις προηγούμενες) και έστω $\alpha=0.9$ (που σημαίνει ότι η πρόβλεψη της επόμενης παραγγελίας θα στηριχθεί κυρίως στις ποσότητες των τελευταίων βδομάδων), η μελλοντική ζήτηση θα είναι $D(n) = 0.9(n-1) + 0.1x_n$. Για αρχική

εκτίμηση $D(0)$ εκτιμούμε μια τιμή και έστω $D(0) = \frac{1}{10} \sum_{i=1}^{10} x_i = 109.8 = D(10)$, που σημαίνει ότι

$$D(11) = 0.9D(10) + 0.1x_{11}. \text{ Δεδομένου } x_{11} = 93 \text{ τότε } D(11) = 108.12.$$

Μολονότι η τιμή x_{11} είναι αρκετά χαμηλή, η πρόβλεψη της επόμενης παραγγελίας είναι πολύ κοντά στην προηγούμενη $D(10)$.

Πολλές φορές κατά τη διάρκεια ενός κύκλου παραγγελιών ο συντελεστής εξομάλυνσης αλλάζει ανάλογα με τη περίοδο. Στην προσαρμοστική εξομάλυνση (adaptive smoothing) ο συντελεστής δίνεται με μορφή διανύσματος εξυπηρετώντας τέτοιου είδους προβλήματα.

1.3 Bullwhip effect

Στο παράδειγμα του Beer game από τα σχεδιαγράμματα των αποθεμάτων γίνεται αισθητή η αύξησή τους στο τέλος της διαδικασίας σε σχέση με τα αρχικά τους κατάσταση και ιδίως στα υψηλότερα στρώματα της αλυσίδας. Αντίθετα στις ενδιάμεσες βδομάδες τα αποθέματα όλων των μελών μηδενίστηκαν. Κάτι τέτοιο έχει ως αποτέλεσμα να αυξάνονται τόσο οι παραγγελίες όσο και ο χρόνος διεκπεραίωσής τους. Στο ιστόγραμμα του σχήματος 4 φαίνονται οι γραμμές παραγγελιών σε αρκετή απόσταση από την αρχική ζήτηση. Και στις δύο ομάδες οι μεγαλύτερες παραγγελίες πραγματοποιούνταν από το τελευταίο μέλος της αλυσίδας. Είναι πιθανό λοιπόν, να δημιουργούνται backorders, δηλαδή ποσότητες οι οποίες αναμένεται να αποσταλούν. Οι παραγγελίες που καθυστερούν να φτάσουν είναι δυνατόν να δημιουργήσουν μηδενικά αποθέματα στις αποθήκες των μελών της αλυσίδας, αλλά και διόγκωση στη ζήτηση. Το φαινόμενο αυτό ονομάζεται **Bullwhip effect**.

Μια παραγωγή βασισμένη στην πρόβλεψη της ζήτησης είναι ίσως η σημαντικότερη αιτία του φαινομένου bullwhip. Τα αλληλεξαρτώμενα επίπεδα της εφοδιαστικής αλυσίδας στηρίζονται στις προβλέψεις, καθώς και στα ιστορικά στοιχεία της ζήτησης των πελατών τους. Το γεγονός αυτό, έχει αρνητικές επιπτώσεις στην σταθερότητα της εφοδιαστικής αλυσίδας. Η μεταβλητότητα στις παραγγελίες, καθώς ανεβαίνουμε στην εφοδιαστική αλυσίδα, προκαλεί στρεβλώσεις της πραγματικής ζήτησης, με αποτέλεσμα η εφοδιαστική αλυσίδα να αποσταθεροποιείται. Οι παραγγελίες προς προμηθευτές τείνουν να έχουν μεγαλύτερη διακύμανση απ' ό,τι οι πωλήσεις προς τους πελάτες (δηλ. παραμόρφωση της ζήτησης) και το φαινόμενο μεταφέρεται προς τα πίσω στην αλυσίδα τροφοδοσίας. Ο προμηθευτής που καθοδηγείται από δεδομένα πρόβλεψης παραγγελιών αντί πραγματικής ζήτησης παραπλανάτε από τη μεγαλύτερη διακύμανση με σημαντικές επιπτώσεις κόστους.

Σε διάφορες μελέτες φαίνεται ότι η επίδραση του φαινομένου bullwhip έχει αρνητικά αποτελέσματα και προκαλεί σημαντικά προβλήματα, όπως:

- Η κακή εξυπηρέτηση πελατών σε κάποιο μέρος της εφοδιαστικής αλυσίδας λόγω της μεταβλητότητας .
- Χαμένα κέρδη λόγω των ελλείψεων, οι οποίες έχουν προκληθεί αντίστοιχα από τις μεταβολές .

- Η παραγωγικότητα του επενδυμένου κεφαλαίου σε διαδικασίες είναι κάτω του μετρίου, καθώς τα εισοδήματα χάνονται.
- Αύξηση του κόστους της μονάδος μιας και το απόθεμα στο οποίο καταλήγει είναι σαφώς μεγαλύτερο από το αρχικό. Ακόμα μεγαλύτερο θα είναι το κόστος όταν οι αποθήκες μείνουν κενές για κάποιο διάστημα.
- Αύξηση των παραγγελιών με κίνδυνο να μην υπάρχει διαθέσιμος χώρος για αποθήκευση. Έτσι αυξάνονται και τα έξοδα αποστολής.
- Εξαιτίας της χρονοκαθυστερήσης οι παραγγελίες και οι παραλαβές δεν μπορούν να γίνουν στον ίδιο χρόνο δημιουργώντας μη προβλεπόμενα έξοδα αλλά και δυσκολία στην οργάνωση [12].

Το φαινόμενο bullwhip είναι ο λόγος της μη αναμενόμενης αύξησης του κόστους παραγωγής. Φυσικά το φαινόμενο παρατηρείται σε όλες τις εφοδιαστικές αλυσίδες, αλλά με τη χρήση κατάλληλων μεθόδων, οι επιπτώσεις του δεν γίνονται μεγάλο εμπόδιο στην ανάλυση του μοντέλου μας.

Δύο από τις αρχές που θα έπρεπε να ενσωματωθούν σε αυτήν την αποτελεσματική αντιμετώπιση είναι ότι η παραγγελία σε ανταπόκριση με την παραγγελία εξουδετερώνει το φαινόμενο μαστιγίου (bullwhip effect) αλλά όχι της αποθήκευσης και επίσης ότι οι επιχειρήσεις πρέπει να ανταποκρίνονται μόνο μια φορά σε κάθε αλλαγή κατανάλωσης της αγοράς διότι οι πολλαπλές ανταποκρίσεις μεγαλώνουν την εντροπία και συνεπώς τις τριβές του συστήματος.

Όταν η εφοδιαστική αλυσίδα είναι ομοιογενής, πράγμα το οποίο αποτελεί μια ιδανική περίπτωση, τότε οι τυπικές στρατηγικές αν είναι αποτελεσματικές σε αυτό το επίπεδο μπορούν να γενικευτούν ομαλά σε μια ολική στρατηγική η οποία θα είναι αποτελεσματική. Η γενική περίπτωση όμως, και είναι βέβαια αυτή που συναντάμε πρακτικά στη διαχείριση μιας εφοδιαστικής αλυσίδας, εμπεριέχει παράγοντες που διαφέρουν μεταξύ τους και ως οντότητες και ως λειτουργικά συστήματα. Τότε το συνολικό παίγνιο μπορεί να εξεταστεί μόνο μέσω της θεωρίας παιγνίων διότι ακόμη και οι αλγόριθμοι δεν μπορούν να εξετάσουν εξ ορισμού όλες τις περιπτώσεις.

Αυτή η παρατήρηση ισχύει και για τον αλγόριθμο σάρωσης που προτάθηκε αρχικά από τους Gillett και Miller για την επίλυση του προβλήματος διανομής από μία αποθήκη και επεκτάθηκε από τους Gillett και Johnson για την περίπτωση πολλαπλών κέντρων διανομής, αλλά και για τους αλγόριθμους ομαδοποίησης-δρομολόγησης που δημιουργούν αρχικά ένα μεγάλο, συνήθως μη εφικτό δρομολόγιο, το οποίο περιλαμβάνει όλους τους πελάτες και το οποίο στη συνέχεια σπάει σε αριθμό μικρότερων και βεβαίως εφικτών δρομολογίων. Πιο

γενικά, το πρόβλημα εμπεριέχει στρατηγικές εφαρμογές οι οποίες χρησιμοποιούνται για σχεδιαστικούς σκοπούς, τακτικές εφαρμογές οι οποίες χρησιμεύουν στον σχεδιασμό των δρομολογίων εντός μιας σχετικά βραχείας χρονικής περιόδου και λειτουργικές εφαρμογές οι οποίες χρησιμοποιούνται για τον λεπτομερή σχεδιασμό των δρομολογίων σε ημερήσια βάση [12].

ΚΕΦΑΛΑΙΟ 2^ο

Petri Nets και Colour PN

2.1 Petri Nets

2.1.1 Εισαγωγή στα Petri Nets

Ένα από τα ποια διαδεδομένα εργαλεία προσομοίωσης συστημάτων , είναι τα **Petri δίκτυα (Petri nets PN)**. Η χρήση τους αναπτύχθηκε κυρίως στα μέσα της δεκαετίας του '60, όπου αναγνωρίστηκε ως ένα ικανοποιητικό μέσο για την περιγραφή και την ανάλυση αλληλοεπιδρώντων διεργασιών. Αρχικά τα δίκτυα Petri χρησιμοποιήθηκαν για την μελέτη ποιοτικών χαρακτηριστικών συστημάτων όπου βασιζόταν στον συγχρονισμό και στη λήψη αποφάσεων στους ίδιους χρόνους. Γι' αυτό και σε καταναμημένα και ταυτόχρονα συστήματα ο χρόνος παίζει σημαντικό ρόλο. Η απόδοση αυτών, εξαρτάται άμεσα από το χρόνο που διαρκούν οι συγκεκριμένες δράσεις και από τις διαφορετικές αποφάσεις που λαμβάνονται κατά την σχεδιάσή τους. Έτσι, για να μπορούν τα δίκτυα Petri να μελετούν θέματα απόδοσης και να εξετάζουν τόσο την ποιότητα (πχ, τη χρονική καθυστέρηση), όσο και την ποσότητα μια υπηρεσίας (πχ, τη ρυθμοαπόδοση) που παρέχεται από το σύστημα, εισήχθη η έννοια του χρόνου στο μοντέλο των δικτύων Petri.

Αποκαλύφθηκαν όμως δύο μειονεκτήματα: Το πρώτο ήταν ότι δεν υπήρχαν κατάλληλες δομές για την αναπαράσταση δεδομένων κι έτσι τα μοντέλα πολλές φορές γινόντουσαν υπερβολικά μεγάλα, γιατί η διαχείριση όλων των δεδομένων έπρεπε να αναπαραχθεί απ' ευθείας μέσω του δικτύου και το δεύτερο ήταν πως ο χειριστής δεν είχε την δυνατότητα να δημιουργήσει ιεραρχικά μοντέλα ή υπομοντέλα. Έτσι δεν υπήρχε διασύνδεση μεταξύ μοντέλων αλλά μόνο μεταξύ των κόμβων [13].

Τα προβλήματα αυτά λύθηκαν ή αντιμετωπίστηκαν σε μεγάλο βαθμό με την βελτίωση νεότερων εκδόσεων και με την βοήθεια των «**χρωματισμένων Petri Nets**» (**colour petri nets CPN**).

2.1.2 Φορμαλισμός των PN

Ένα Petri δίκτυο μπορεί να δομηθεί σαν μία μηχανή $C=(P, T, I, O)$ όπου η σύνταξή της απαιτεί την εισαγωγή δύο συνόλων και δύο συναρτήσεων:

P: το σύνολο των καταστάσεων (p)(places)

T: Το σύνολο των βελών-μεταβάσεων (t)(transition)

I: Συνάρτηση εισόδου (input function) με $I: T \rightarrow P$

O: Συνάρτηση εξόδου (output function) με $O: T \rightarrow P$

όπου

στην $I(t_i)=p_i$ οι p_i είναι οι καταστάσεις που ξεκινούν από την μετάβαση t_i .

στην $O(t_j)=p_j$ οι p_j είναι οι καταστάσεις που θα καταλήξει η t_j .

Για παράδειγμα ένα PN1 μπορεί να οριστεί ως $C=(P, T, I, O)$ με $P=\{p_1, p_2, p_3, p_4, p_5\}$ και $T=\{t_1, t_2, t_3, t_4\}$ όπου

$$I(t_1)=\{p_1\}$$

$$O(t_1)=\{p_2, p_3, p_5\}$$

$$I(t_2)=\{p_2, p_3, p_5\}$$

$$O(t_2)=\{p_5\}$$

$$I(t_3)=\{p_3\}$$

$$O(t_3)=\{p_4\}$$

$$I(t_4)=\{p_4\}$$

$$O(t_4)=\{p_2, p_3\}$$

Η πολυπλοκότητα μιας κατάστασης εισόδου είναι ο αριθμός των ακμών που ξεκινούν από αυτήν ($p_i, I(t_i)$) ενώ η πολυπλοκότητα μιας κατάστασης εξόδου είναι ο αριθμός ακμών που καταλήγει στην $p_j(p_i, O(t_j))$. Στο παράδειγμά μας θα έχουμε

$$\#(p_1, I(t_1)) = \#(p_1, p_1) \text{ η πολλαπλότητα της } p_1 \text{ είναι } 1$$

$$\#(p_4, O(t_4)) = \#(p_4, p_2, p_3) \text{ η πολλαπλότητα της } p_4 \text{ είναι } 2.$$

Ένα τέτοιο δίκτυο μπορεί να αναπαρασταθεί από ένα γράφημα $G=(V, A)$ όπου $V=\{v_1, v_2, \dots, v_n\}$ το σύνολο των κορυφών (places) και $A=\{a_1, a_2, \dots, a_n\}$ οι ακμές όπου $a_i=(v_i, v_j)$ οι κορυφές (transition). Το σύνολο V μπορούμε να πούμε ότι δομείται από τα P, T ενός PN μόνο όταν $V=P \cup T$ και $P \cap T = \emptyset$ Έτσι, όταν $a_i=(v_i, v_j)$ το $v_i \in P$ και $v_k \in T$.

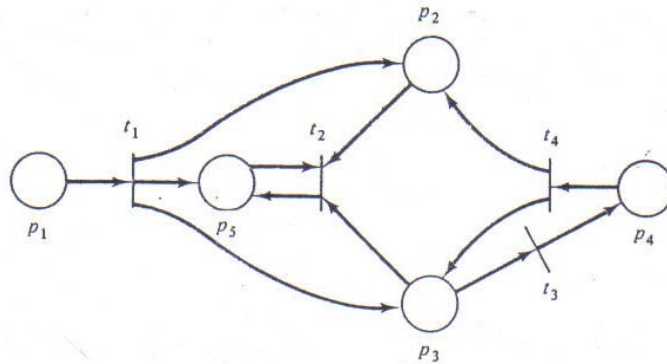
Αν θεωρήσουμε την αντιστοιχία

$$\#((p_i, t_j), A) = \#(p_i, I(t_j))$$

$$\#((t_j, p_i), A) = \#(p_i, O(t_j))$$

τότε ο γράφος G θα είναι ισοδύναμος με το δίκτυο C.

Το σχήμα 5 είναι το γράφημα που αντιστοιχεί το PN1 [1].



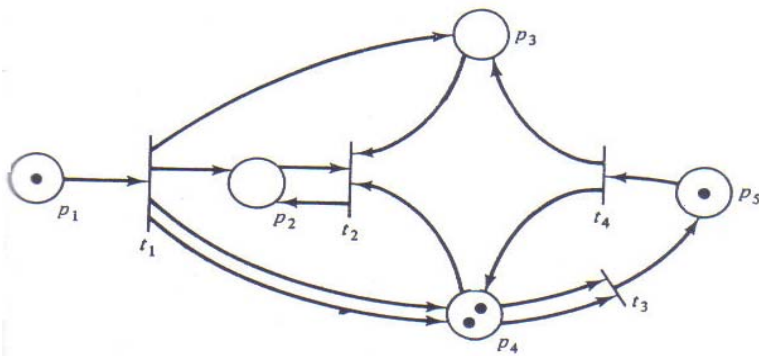
Σχήμα 5. Το γράφημα του PN1.

2.1.3 Επέκταση των PN

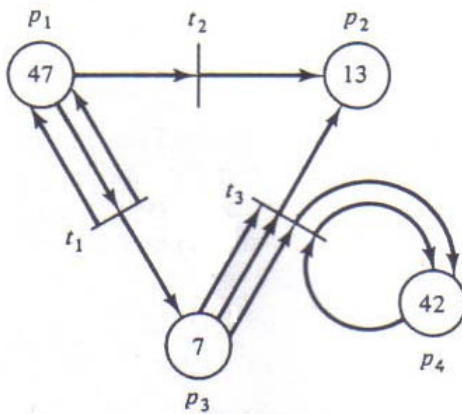
2.1.3.1 Μαρκάρισμα

Προκειμένου να εφαρμοστεί η διακίνηση προϊόντων σε ένα PN μπορούμε να εισάγουμε την έννοια του μαρκαρίσματος. Θα λέμε ότι μία εντολή θα βρίσκεται σε μία κατάσταση, όταν στο γράφημα ενός PN ένα place επισημαίνεται (μαρκαρίζεται) με μία μάρκα (token).

Θεωρούμε συνάρτηση $\mu: P \rightarrow \mathbb{N}$ με $\mu(p_i) = \mu_i$ η οποία δηλώνει πόσες μάρκες υπάρχουν σε κάθε place. Επίσης η έννοια του μαρκαρίσματος μπορεί να δοθεί και σαν ένα n -διάστατο διάνυσμα $\mu = (\mu_1, \mu_2, \dots, \mu_n)$ όπου $n = |P|$. Το μ συμβολίζει τον αριθμό των tokens που υπάρχει στα p_1, p_2, \dots, p_n αντίστοιχα. Πολλές φορές ένα PN μαρκαρισμένο δηλώνεται και ως $M = (P, T, I, O, \mu)$.



Σχήμα 6. Μαρκάρισμα του PN1.



Σχήμα 7. Γράφημα ενός PN όπου το πλήθος των μαρκών αναγράφονται με αριθμούς.

Για παράδειγμα στο σχήμα 6 θα έχουμε $\mu(p_1)=1$, $\mu(p_2)=2$, ή ποιο συνοπτικά $\mu=(1,0,0,2,1)$ ενώ στο σχήμα 7 θα έχουμε $\mu=(47,13,7,42)$.

2.1.3.2 Εκτέλεση

Η έννοια της μάρκας δεν θα είχε καμία αξία, αν δεν θα μπορούσε να μεταφερθεί (αναπηδήσει) σε μια άλλη κατάσταση ανάλογα με την συνδεσιμότητα της. Μία μετάβαση ενεργοποιείται, όταν συμβαίνει ένα γεγονός.

Το πόσο είναι δυνατή ή όχι η εκτέλεση ενός PN ελέγχεται από τον αριθμό των μαρκών της κάθε κατάστασης. Έτσι, το κάθε token μεταπηδά από μία κατάσταση σε μία άλλη, αν και εφόσον υπάρχει το πολύ ο ίδιος αριθμός ακμών που να ξεκινάει από αυτήν. Σε αυτή τη περίπτωση χαρακτηρίζεται ως **ικανό** (enabled token).

Οι κανόνες εκτέλεσης των δικτύων Petri είναι οι εξής :

i) Μία κατάσταση $t_j \in T$ ενεργοποιείται αν και μόνο αν

$$\mu(p) \geq I(p,t), \forall p \in P,$$

όπου $\mu(p)$ είναι το μαρκάρισμα της θέσης p .

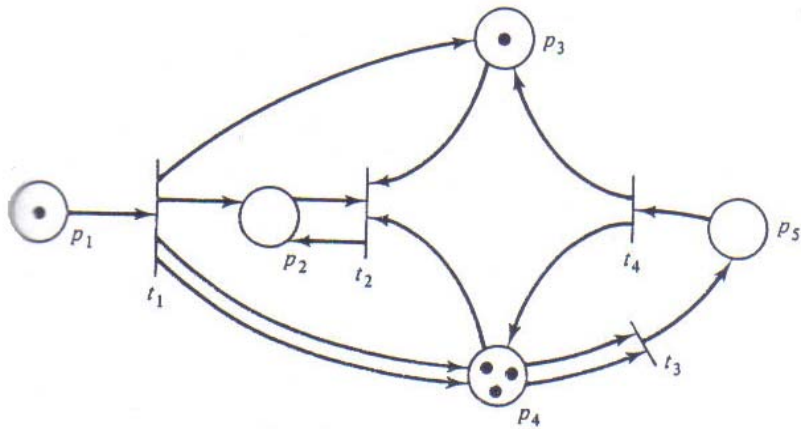
ii) Η μετάβαση t που ενεργοποιείται σε ένα μαρκάρισμα μ , πυροδοτεί και ως αποτέλεσμα προκύπτει ένα νέο μαρκάρισμα μ' , όπου

$$\mu'(p) = \mu(p) - I(p,t) + O(p,t), \forall p \in P.$$

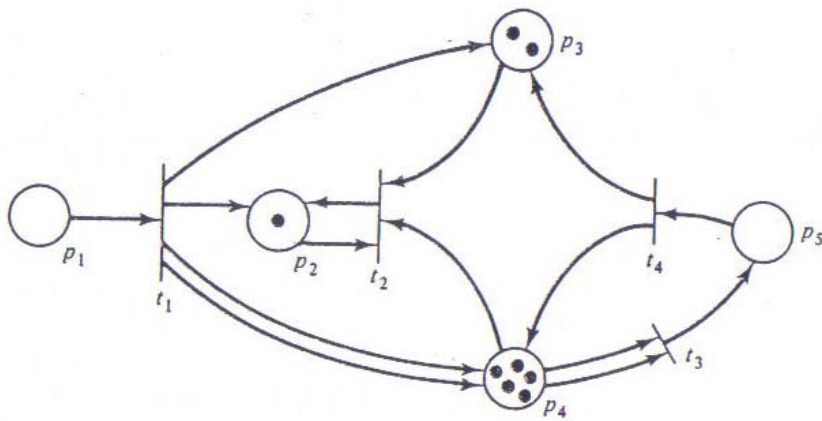
Ο πρώτος από τους κανόνες εκτέλεσης είναι ο κανόνας ενεργοποίησης, ενώ ο δεύτερος είναι ο κανόνας πυροδότησης.

Στο παράδειγμα (σχήμα 6), τρεις μεταβάσεις είναι ενεργές, οι t_1, t_3 και t_4 . Αν η t_4 πυροδοτήσει, τότε η κατάσταση εισόδου (με ένα token) θα έχει μία μάρκα λιγότερη ενώ οι καταστάσεις εξόδου (p_3 και p_4) της t_4 θα έχουν μία επιπλέον (σχήμα 8). Τώρα είναι μόνο οι t_1 και t_3 ενεργές. Αν πυροδοτήσει η t_1 θα χάσει το ένα token και θα μεταφερθεί μία μάρκα στην p_3 μία στην p_2 δύο και στην p_4 αφού σε αυτήν αντιστοιχούν δύο βέλη (σχήμα 9).

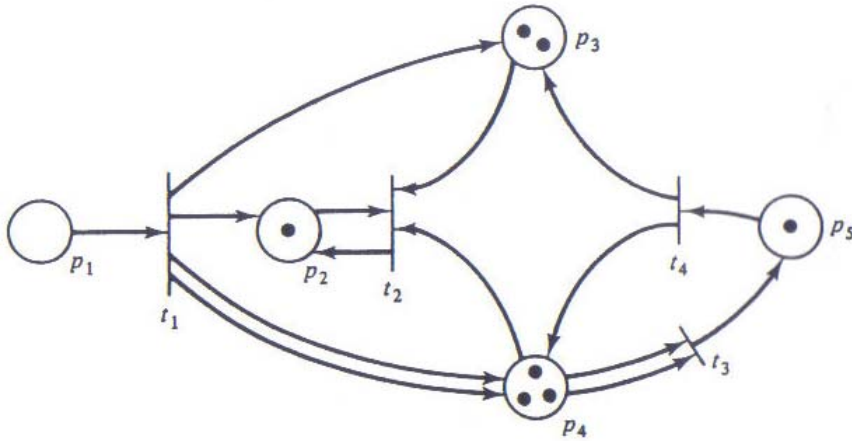
Ενεργοποιημένες μεταβάσεις σε αυτήν τη φάση θα είναι οι t_2 και οι t_3 . Αν πυροδοτήσει η t_3 θα χαθούν 2 μάρκες από την p_4 και η p_5 θα έχει πλέον μία μάρκα. Η διαδικασία της πυροδότησης θα συνεχίζεται εωσότου υπάρχουν ενεργά transitions και ονομάζεται token game (το παιχνίδι των μαρκών). Αν δεν συμβαίνει αυτό τότε το σύστημα σταματάει την διαδικασία.



Σχήμα 8. Τα μαρκαρίσματα του PN του σχήματος 6 μετά την πυροδότηση της t_4 .



Σχήμα 9. Τα μαρκαρίσματα του PN του σχ. 8 μετά την πυροδότηση της t_1 .



Σχήμα 10. Τα μαρκαρίσματα του PN του σχ. 9 μετά την πυροδότηση της t_3 .

2.1.3.3 State Space

Όταν υπάρχουν περισσότερες από μία ενεργοποιημένες μεταβάσεις, υπάρχουν αρκετές ενδεχόμενα μαρκαρίσματα. Η συνάρτηση ‘next state space’ εξάγει όλες τις πιθανές καταστάσεις ανάλογα με ποια φάση του token game βρισκόμαστε.

Για το $C=(P, T, I, O)$ η συνάρτηση $\delta: N^n \times T \rightarrow N^n$ ορίζεται αναδρομικά ως

$$\delta(\mu^i, t_{j_i}) = \mu^{i+1} \quad i=0,1,2,\dots \text{ δοθέντος του } \mu^0$$

όπου μ^{i+1} τα επόμενα μαρκαρίσματα μετά την πυροδότηση της t_{j_i} .

Η συνάρτηση δ ορίζεται καλά, όταν η t_j είναι ενεργοποιημένη.

Από τις τιμές της δ προκύπτουν δύο ακολουθίες οι οποίες μπορούν να περιγράψουν με ακρίβεια τη διακίνηση των μαρκών.

Η ακολουθία των μαρκαρισμάτων $(\mu^0, \mu^1, \mu^2, \dots)$

Η ακολουθία των ενεργών καταστάσεων $(t_{j_0}, t_{j_1}, t_{j_2}, \dots)$.

Στο PN1 του σχήματος 6 οι δ παίρνει τις τιμές:

$$\delta((1,0,0,2,1), t_4) = (1,0,1,3,0)$$

$$\delta((1,0,1,3,0), t_1) = (0,1,2,5,0)$$

$$\delta((0,1,2,5,0), t_3) = (0,1,2,3,1)$$

2.1.4 Ιδιότητες ενός PN

2.1.4.1 Εκτελεσιμότητα

Για το $C=(P, T, I, O)$ με μ το διάνυσμα των μαρκαρισμάτων, σε μία φάση με ενεργά transition το μ' θα ονομάζεται **άμεσα εκτελέσιμο** από το μ αν υπάρχει $t_j \in T$ ώστε

$$\delta(\mu, t_j) = \mu'.$$

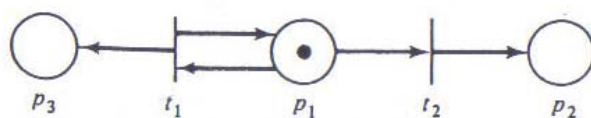
Ως επέκταση αυτού του ορισμού μπορούμε να πούμε ότι αν μ' είναι άμεσα εκτελέσιμο ως προς το μ και μ'' άμεσα εκτελέσιμο ως προς το μ' τότε και το μ'' είναι άμεσα εκτελέσιμο ως προς το μ . Ισχύει δηλαδή η προσεταιριστική ιδιότητα για αυτήν την έννοια.

Το μ' θα λέμε ότι θα είναι απλά εκτελέσιμο ή **εκτελέσιμο** από το μ αν και μόνο αν υπάρχει ακολουθία ενεργών μεταβάσεων η οποία θα αλλάξει την θέση των μαρκαρισμάτων του μ στο μ' . Το σύνολο $R(C, \mu)$ το ονομάζουμε σύνολο εκτελεσιμότητας και περιέχει όλα τα διανύσματα μαρκαρισμάτων τα οποία είναι **εκτελέσιμα** με το μ .

Αποδεικνύεται ότι το $R(C, \mu)$ είναι το μικρότερο υποσύνολο του $C'=(C, \mu)$ το οποίο έχει τις εξής ιδιότητες:

i) $\mu \in R(C, \mu)$

ii) Αν $\mu' \in R(C, \mu)$ και $\mu'' = \delta(\mu', t_j)$ για κάποιο $t_j \in T$ τότε $\mu'' \in R(C, \mu)$



Σχήμα 11. Το γράφημα του PN2

Εφαρμογή της πρότασης αποτελεί το παρακάτω παράδειγμα για το PN2 του σχήματος 11:

Με $\mu^0 = (1, 0, 0)$ θα έχουμε:

$$\delta(\mu^0, t_1) = (1, 0, 1) = \mu^1$$

$$\delta(\mu^0, t_2) = (0, 1, 0) = \mu^2.$$

Για την μ^0 οι μ^1, μ^2 είναι άμεσα εκτελέσιμες. Για την μ^2 δεν υπάρχει αντίστοιχο εκτελέσιμο μ . Αντίθετα για το μ^1 τα $(0,1,1), (1,0,2)$ είναι άμεσα εκτελέσιμα. Αναγωγικά, μπορούμε να πούμε ότι το σύνολο εκτελέσιμων ακολουθιών ως προς το αρχικό μ^0 είναι

$$R(C, \mu) = \{(1,0,n), (0,1,n) \mid n \geq 0\}.$$

Επεκτείνουμε την συνάρτηση επόμενης κατάστασης έτσι ώστε να δίνει τα εκτελέσιμα μ' δεδομένου μ και όχι μόνο τα άμεσα εκτελέσιμα. Για

$$\mu^1 \xrightarrow{t_1} \mu^2 \xrightarrow{t_2} \mu^3 \xrightarrow{t_3} \dots \xrightarrow{t_k} \mu'$$

Σχήμα 12. Ακολουθία μαρκαρισμάτων του PN2

θα έχουμε $\mu' = \delta(\mu, t_1, t_2, \dots, t_k)$ ή αν σ ακολουθία ενεργών καταστάσεων τότε

$$\delta(\mu, t_j, \sigma) = \delta(\delta(\mu, t_j), \sigma) \text{ με}$$

$$\delta(\mu, \lambda) = \mu$$

2.1.4.2 Περιοριστικότητα (boundedness)

Μία κατάσταση ενός PN θα ονομάζεται **κ-περιορισμένη** αν ο αριθμός των tokens που έχει είναι μικρότερος ή ίσος με κ. Ένα PN θα είναι κ-πεπερασμένο (simply bounded) αν όλες οι καταστάσεις του δικτύου είναι κ-περιορισμένες δηλαδή για κάθε p να ισχύει $\mu(p) \leq \kappa$, $\kappa \in \mathbb{N}^*$. Για $\kappa=1$ το PN ονομάζεται ασφαλές.

Αυτή η ιδιότητα είναι ιδιαίτερα χρήσιμη σε προβλήματα όπου δεν θέλουμε να έχουμε υπερχειλίση από tokens σε μία κατάσταση (σε hardware υπολογιστών) ή όταν θέλουμε να εκτελούνται πολλές παραγγελίες στην μονάδα του χρόνου ή ακόμα για να αποφύγουμε την παραβίαση των χρονικών ορίων.

Περισσότερες ιδιότητες των PN αναλύονται στο παράρτημα στην ενότητα A.

2.1.5 Μοντελοποίηση των PN

Στην ενότητα αυτήν θα εξεταστούν κάποια σημαντικά συστήματα τα οποία μπορούν να μοντελοποιηθούν με χρήση των PN. Από αυτή την διαδικασία μπορούμε να δούμε διάφορες τεχνικές μοντελοποίησης που θα μας βοηθήσουν στη συνέχεια της εργασίας.

2.1.5.1 Finite state machine

Η συμπεριφορά των μηχανών πεπερασμένης κατάστασης εξαρτάται από όλες τις προηγούμενες καταστάσεις και όχι μόνο από το σύνολο μαρκών εισόδου μιας δεδομένης μετάβασης. Τέτοιες μηχανές είναι τα ακολουθιακά κυκλώματα, όπου όταν έχουν n ενεργά transitions στο επόμενο βήμα θα υπάρχουν 2^n ενεργά. Επίσης σε μία πεπερασμένη μηχανή δεν έχουμε εισαγωγή και εξαγωγή μαρκών, αλλά συμβόλων [14].

Φορμαλιστικά, μία τέτοια μηχανή ορίζεται ως

$S=(Q, \Sigma, \Delta, \delta, \Gamma)$ με

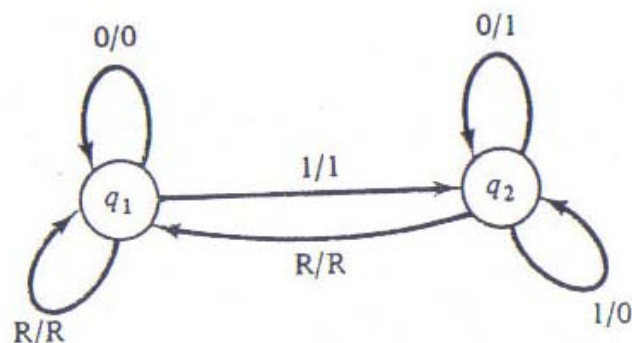
Q : το σύνολο των πεπερασμένων καταστάσεων (q_1, q_2, \dots, q_k)

Σ : Τα σύμβολα εισόδου.

Δ : Τα σύμβολα εξόδου.

$\delta: Q \times \Sigma \rightarrow Q$ η συνάρτηση της επόμενης μετάβασης

$\Gamma: Q \times \Sigma \rightarrow \Delta$ η συνάρτηση εξόδου.



Σχήμα 13. Παράδειγμα μηχανής state S.

Ένα βέλος που συνδέει τις καταστάσεις q_1, q_2 με μάρκα a/b σημαίνει ότι αν στην q_1 εισάγουμε το a τότε θα έχουμε αλλαγή στην κατάσταση q_2 με εξαγόμενο στοιχείο το b . Στο παράδειγμα του σχήματος 13, έχουμε $Q = \{q_1, q_2\}$ $\Sigma = \{0, 1, R\}$ $\Delta = \{0, 1, R\}$

Και οι τιμές των συναρτήσεων θα είναι:

Για την δ :	$\delta(q_1, 0) = q_1$	Για την Γ :	$\Gamma(q_1, 0) = 0$
	$\delta(q_1, 1) = q_2$		$\Gamma(q_1, 1) = 1$
	$\delta(q_1, R) = q_1$		$\Gamma(q_1, R) = R$
	$\delta(q_2, 0) = q_2$		$\Gamma(q_2, 0) = 1$
	$\delta(q_2, 1) = q_2$		$\Gamma(q_2, 1) = 0$
	$\delta(q_2, R) = q_1$		$\Gamma(q_2, R) = R$

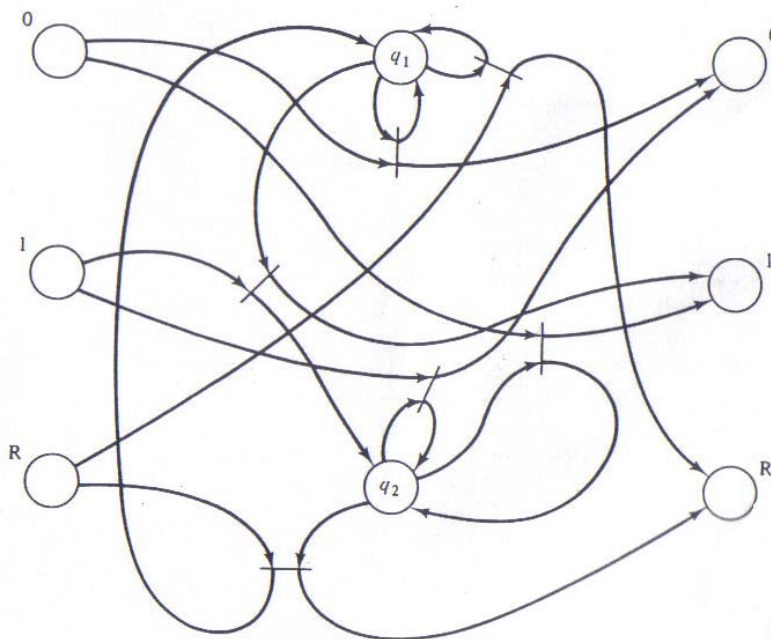
Μία πεπερασμένη state μηχανή $S = (Q, \Sigma, \Delta, \delta, \Gamma)$ μπορεί να μοντελοποιηθεί από ένα PN $C = (P, T, I, O)$ με

$$T = \{t_{q,\sigma} / q \in Q \text{ και } \sigma \in \Sigma\}$$

$$I(t_{q,\sigma}) = \{q, \sigma\}$$

$$O(t_{q,\sigma}) = \{\delta(q, \sigma), \Gamma(q, \sigma)\}$$

με συνδεσιμότητα όπως φαίνεται στο σχήμα 14.

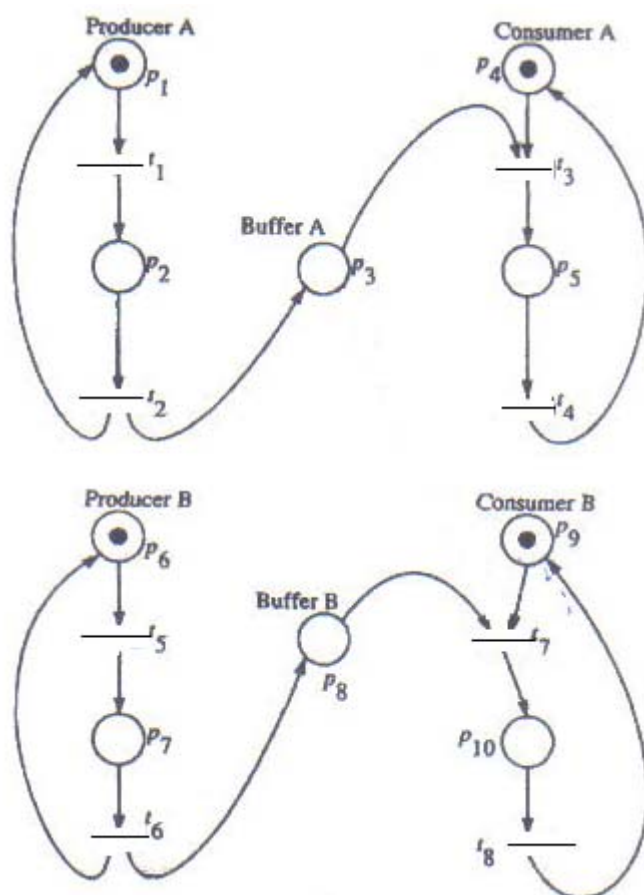


Σχήμα 14. Παράδειγμα μοντελοποίησης μηχανής state S με τη βοήθεια Petri Net.

2.1.5.2 Παραγωγός-Πελάτης σε προτεραιότητα

Σε προβλήματα διακίνησης προϊόντων μεταξύ παραγωγού και πελάτη πολλές φορές μας ενδιαφέρει να δίνουμε προτεραιότητα σε κάποια διεργασία ώστε να υπάρχει μία λογική σειρά στις εντολές.

Το δίκτυο του σχήματος 15 δείχνει τις φάσεις που περνάει ένα προϊόν (μάρκα) από τον παραγωγό A (producer A) στον πελάτη (customer A). Ενώ και οι δύο καταστάσεις p_1 p_4 είναι μαρκαρισμένες, μόνο η μετάβαση t_1 είναι ενεργή, μιας και για να πυροδοτήσει η t_3 θα πρέπει η p_3 να είναι στη φάση μαρκαρίσματος. Για να γίνει αυτό, θα πρέπει να πυροδοτήσει και η t_2 . Δηλαδή το σύστημα θα ξεκινήσει οπωσδήποτε από την κατάσταση p_1 όπως είναι και λογικό.



Σχήμα 15. Δίκτυο με δύο παραγωγούς και δύο πελάτες.

Στο σχήμα δίνονται δύο πανομοιότυπες περιπτώσεις διακίνησης υλικού ανάμεσα σε δύο μέλη της εφοδιαστικής αλυσίδας, προκειμένου να επιτευχθεί είτε συνεργασία είτε ανταγωνισμός.

Αν συνδέσουμε την p_3 με την t_7 ο customer B θα έχει την δυνατότητα να εφοδιαστεί από δύο παραγωγούς. Ορίζοντας κατάλληλα κριτήρια και με συγκεκριμένο έλεγχο ο πελάτης B θα επιλέξει την καλύτερη προσφορά. Αν συνδέσουμε τα t_2 με p_7 θα μπορέσουν οι δύο παραγωγοί να συνεργαστούν για την αποδοτικότερη εξυπηρέτηση του εκάστοτε πελάτη [14].

2.1.6 Ανάλυση-άλλοι τρόποι έκφρασης των PN

Για συντομότερη και αναλυτικότερη περιγραφή ενός μοντέλου με τη βοήθεια ενός PN, ενσωματώνοντας όλες τις ιδιότητές του, χρησιμοποιούμε εναλλακτικούς τρόπους έκφρασής τους.

Ορίζω το $PN=(P,T,F,W,M_0)$ όπου

$P=(p_1, p_2, \dots, p_k)$ το πεπερασμένο σύνολο καταστάσεων.

$T=(t_1, t_2, \dots, t_n)$ το πεπερασμένο σύνολο των μεταβάσεων.

$F \subseteq (P \times T) \cup (T \times P)$ το σύνολο των ακμών.

$W: F \rightarrow \{1, 2, 3, \dots\}$ τα βάρη των ακμών

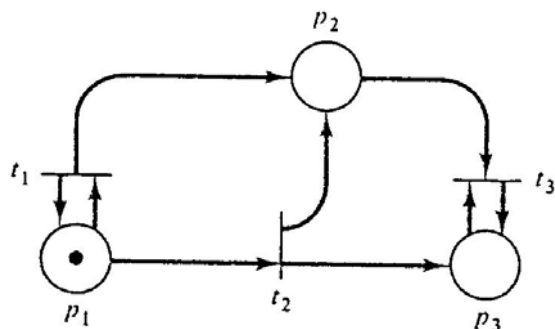
$M_0: P \rightarrow \{0, 1, 2, 3, \dots\}$ η αρχική κατάσταση μαρκαρισμάτων

με τον περιορισμό $P \cap T = \emptyset$ και $P \cup T \neq \emptyset$.

Η συνάρτηση βάρους δίνει τον αριθμό των βελών που ξεκινούν από ένα place ή transition και καταλήγουν σε transition ή place. Για παράδειγμα στο σχήμα 7 θα έχουμε: $W(t_1, p_1)=2$ $W(p_3, t_3)=3$. Ο αριθμός των βαρών πολλές φορές σημειώνεται επάνω στο αντίστοιχο βέλος. Όταν $W=1$ το βάρος παραλείπεται.

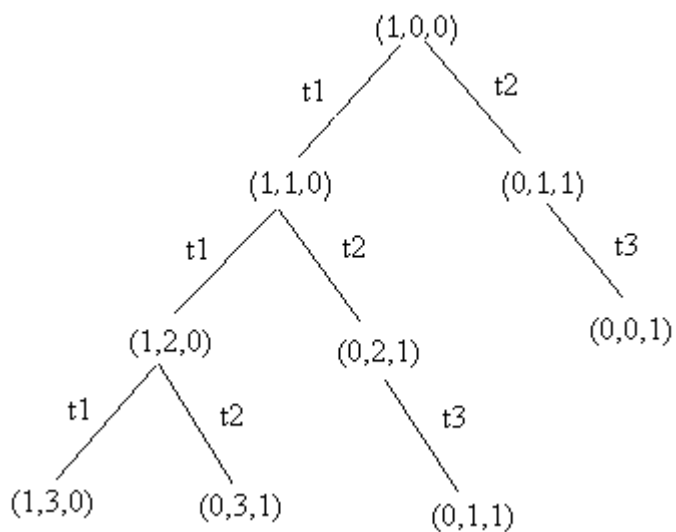
$M_0(p_1)=47$ $M_0(p_2)=13$ $M_0(p_3)=7$ $M_0(p_4)=42$.

Προκειμένου να καταγραφούν οι πιθανές μεταβάσεις και όλες οι δυνατές πυροδοτήσεις ενός δικτύου χρησιμοποιούμε τα δέντρα προσιτότητας.



Σχήμα 16. Το petri net 2

Η αρχική κατάσταση μαρκαρισμάτων του δικτύου του σχήματος 16 καταγράφεται ως διάνυσμα και τοποθετείται στην κορυφή. (1^ο επίπεδο)



Σχήμα 17. Δέντρο προσιτότητας του PN του petri net 2.

Τα ενδεχόμενα μαρκαρίσματα κατά την επόμενη πυροδότηση t_1 ή t_2 φαίνονται στο επίπεδο 2. Αν είμαστε στην κατάσταση $(1,1,0)$ υπάρχουν δύο περιπτώσεις: Να πυροδοτήσει πάλι η t_1 ή t_2 . (3^ο επίπεδο).

Η αναπαράσταση της λειτουργίας ενός PN με τη βοήθεια του δέντρου, μας δίνει πληροφορίες για το σύνολο των καταστάσεων, των μεταβάσεων, των βαρών των ακμών ακόμα και για την συνδεσιμότητα του.

Η ανάλυση του δέντρου προσιτότητας περιγράφει ένα PN με τον ίδιο τρόπο που περιγράφουν τα occurrence graphs ένα colour Petri net όπως θα δούμε παρακάτω.

2.2 Coloured Petri Nets

Τα Χρωματισμένα δίκτυα Petri ενσωματώνουν και τη δόμηση δεδομένων και την ιεραρχική σύνθεση, χωρίς να αλλοιώνουν τις αξίες των πρωτότυπων δικτύων Petri.

Η ανάπτυξη των Χρωματισμένων δικτύων Petri οφείλεται στην επιθυμία να δημιουργηθεί μια γλώσσα μοντελοποίησης βιομηχανικής ισχύος, που να είναι ταυτόχρονα, καλά ορισμένη σε θεωρητικό επίπεδο και πολύπλευρη, ώστε να μπορεί να χρησιμοποιηθεί πρακτικά σε συστήματα του μεγέθους και της πολυπλοκότητας, που συναντώνται στα τυπικά βιομηχανικά έργα. Για να επιτευχθεί αυτό στα CPN, συνδυάστηκε η δύναμη των δικτύων Petri με τη δύναμη των γλωσσών προγραμματισμού. Τα δίκτυα Petri παρέχουν τις κατάλληλες δομές για την περιγραφή του συγχρονισμού των ταυτόχρονων διεργασιών, ενώ μια γλώσσα προγραμματισμού παρέχει τις κατάλληλες δομές για τον ορισμό των τύπων δεδομένων και τη διαχείριση των τιμών τους. Τέλος, το μοντέλο ενός CPN ενός συστήματος περιγράφει τις καταστάσεις στις οποίες μπορεί να βρίσκεται ένα σύστημα και τις μεταβάσεις μεταξύ των διαφόρων καταστάσεων [81].

Τα Χρωματισμένα δίκτυα Petri έχουν χρησιμοποιηθεί σε μια μεγάλη ποικιλία εφαρμογών, όπως θα αναφερθεί εκτενέστερα παρακάτω.

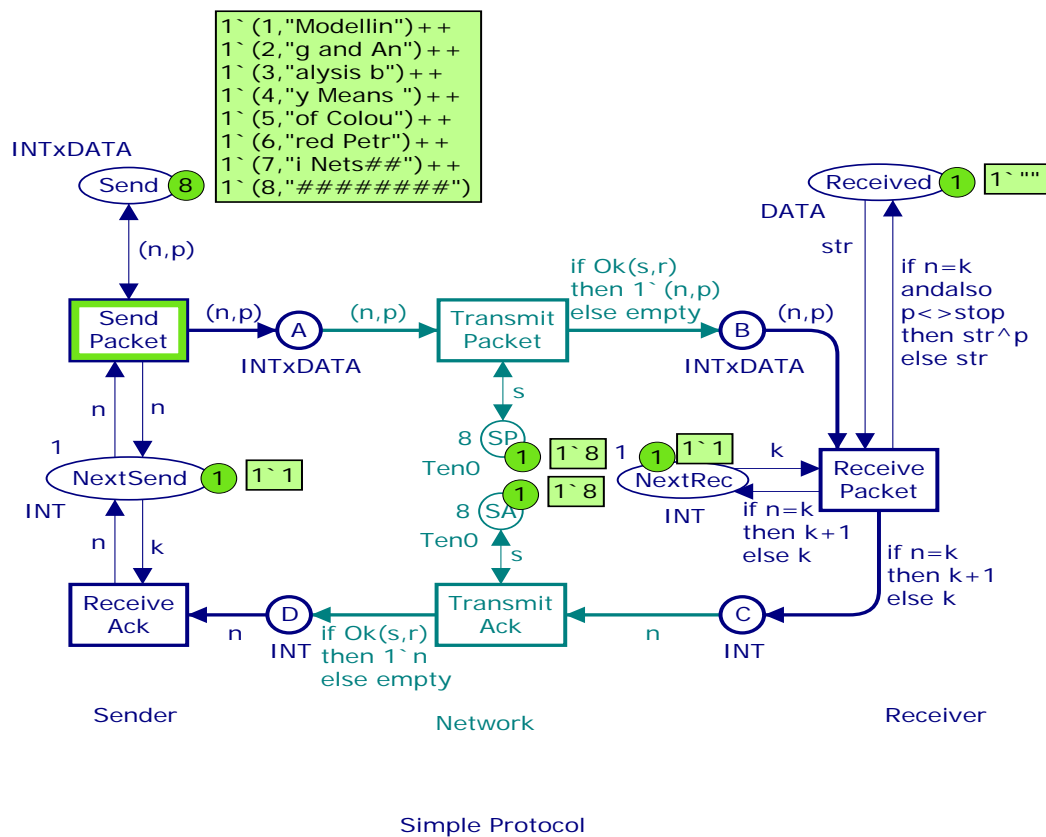
2.2.1 Φορμαλισμός των CPN

Η εξέλιξη των CPN μπορεί να συγκριθεί σαν την μετάβαση συναρμολογούμενων γλωσσών σε γλώσσες προγραμματισμού νέας τεχνολογίας. Και αυτό γιατί ενώ στα PN υπήρχε η δυνατότητα μιας απλής περιγραφής του μοντέλου, στα CPN με χρήση κατάλληλων μέσων ελέγχου ο χρήστης επιλέγει την διαδρομή που θα ακολουθήσει μία αρχική μετάβαση (σχήμα 17),

τον αριθμό και τον τύπο των tokens, τη χρονική έκταση του μοντέλου, την ανάλυση συμπερασμάτων και τη σύνθεσή τους.

Το παράδειγμα που ακολουθεί δίνει μία γενική ιδέα για την μοντελοποίηση με χρήση ενός CPN (σχήμα 18).

Το δίκτυο περιγράφει την μεταφορά πακέτων από τον αποστολέα στον παραλήπτη όπου μπορεί ανάλογα με τη διαδρομή, το πακέτο να παραδοθεί στην ώρα του, να αργήσει ή και να χαθεί. Το CPN του σχήματος αποτελείται από 3 υποδίκτυα. Από το sender, το network και το receiver.

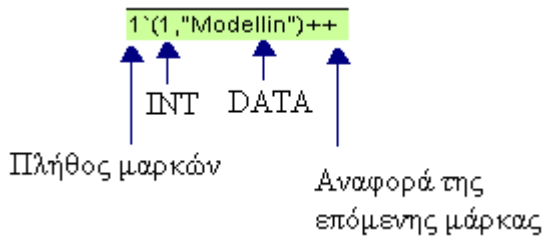


Σχήμα 18. Η έναρξη προσομοίωσης του μοντέλου

Το γραφικό περιβάλλον ενός CPN μοιάζει αρκετά με ενός PN με κάποιες βέβαια προσθήσεις. Οι καταστάσεις (places) συμβολίζονται με ελλείψεις, ενώ οι μεταβάσεις (transitions) με ορθογώνια που συνδέονται μεταξύ τους με βέλη. Τα ονόματα που δίνονται σε κάθε κατάσταση ή μετάβαση βοηθούν στην μετέπειτα ανάλυσή του.

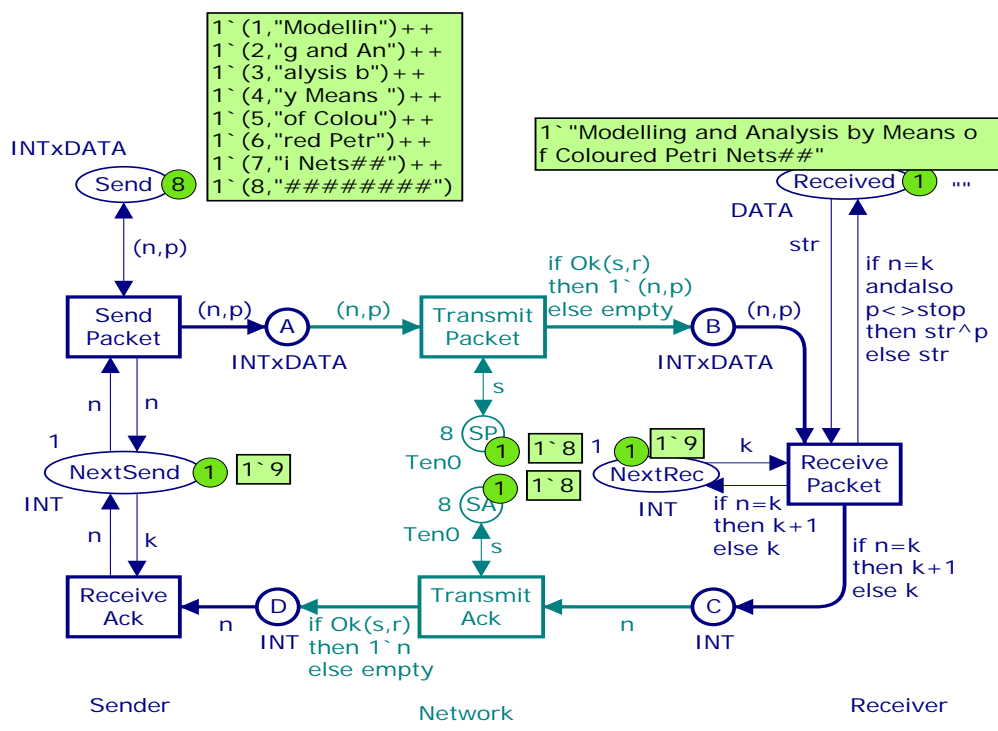
Οι μάρκες των διαφόρων places στην αρχική φάση (initial mark) συμβολίζονται με πράσινο χρώμα. Σε κάθε κατάσταση πρέπει εξαρχής να δηλώνεται ο τύπος της μάρκας.

Για παράδειγμα η κατάσταση send δέχεται μόνο μάρκα με τη μορφή διατεταγμένου ζεύγους INT×DATA δηλαδή φυσικού αριθμού και κειμένου και δηλώνουν τα πακέτα που θα πρέπει να σταλούν στην received



Στις καταστάσεις A και B η δήλωση είναι ίδια και αναπαριστούν τα πακέτα που έχουν αποσταλεί από το δίκτυο sender αλλά δεν έχουν φτάσει στο network και τα πακέτα που στέλνονται από το network και βρίσκονται σε αναμονή από τον receiver αντίστοιχα.

Στη κατάσταση received το token είναι μορφής DATA. Το $1^{<str>}$ σημαίνει ότι κάθε φορά που θα πυροδοτεί η receive packet στη received θα εισάγεται ένα κομμάτι κειμένου. Στο τέλος της διαδικασίας η μετάβαση θα τυπωθεί με το κείμενο “Modelling and analysis by Means of Coloured Petri Nets ##”



Simple Protocol

Σχήμα 19. Η τελική φάση προσομοίωσης του μοντέλου Top.

Οι καταστάσεις C και D δέχονται μάρκα, μόνο όταν είναι τύπου INT και αποτελούν βεβαιώσεις λήψης πακέτων που αποστέλλονται μεταξύ των μελών. Κάθε φορά που ένα πακέτο φτάνει στο προορισμό του στη C τυπώνεται συμβολικός αριθμός που φέρει κάθε πακέτο, ενώ στην D πραγματοποιείται η δήλωση αυτή, έτσι ώστε να ενημερωθεί ο next send και να μην σταλθεί ξανά το αυτό πακέτο.

Οι θέσεις NextSend και NextRec παριστάνουν μετρητές, που κρατάνε τον αριθμό του επόμενου πακέτου που είναι να αποσταλεί ή να παραληφθεί. Έχουν τύπο INT και κάθε μία από αυτές έχουν αρχικά μία μάρκα με τιμή 1. Οι δύο τελευταίες θέσεις SP και SA έχουν τον τύπο Ten0, ο οποίος περιέχει όλους τους ακέραιους αριθμούς από το μηδέν ως το δέκα [15].

Οι δράσεις σε ένα Χρωματισμένο δίκτυο Petri παριστάνονται, μέσω των μεταβάσεων. Σε μια μετάβαση, ένα εισερχόμενο τόξο υποδεικνύει ότι η μετάβαση μπορεί να αφαιρέσει μάρκες από την αντίστοιχη θέση, ενώ ένα εξερχόμενο τόξο δείχνει ότι μπορεί να προσθέσει μάρκες στην αντίστοιχη θέση. Ο ακριβής αριθμός των μαρκών, καθώς και οι τιμές τους, ορίζονται από τις επιγραφές τόξου (arc expressions), οι οποίες τοποθετούνται δίπλα από τα τόξα. Η μετάβαση SendPacket έχει τρία τόξα που την περιβάλλουν, με δύο διαφορετικές επιγραφές τόξου : (n, p) και n. Για να μιλήσουμε για την εκτέλεση (occurrence) της μετάβασης SendPacket, πρέπει να καταχωρήσουμε (bind) στη μεταβλητή n μία τιμή τύπου INT (δηλαδή μία τιμή από το σύνολο των ακεραίων αριθμών) και στη μεταβλητή p μία τιμή τύπου DATA. Αλλιώς δεν μπορούμε να υπολογίσουμε τις επιγραφές τόξου (n, p) και n. Μετά την πυροδότηση της μετάβασης οι μεταβλητές θα πάρουν τις εξής τιμές: η n=1 και p="Modellin".

Ταυτόχρονα, τρεις μάρκες θα προστεθούν στις θέσεις εξόδου : οι θέσεις Send και A θα πάρουν μία μάρκα με τιμή (1, "Modellin"), ενώ η θέση NextSend θα πάρει μία μάρκα με τιμή 1. Έτσι το συνολικό αποτέλεσμα της εκτελούμενης μετάβασης είναι ότι πρόσθεσε στη θέση A, μία μάρκα που αναπαριστά το πακέτο νούμερο ένα. Αυτό σημαίνει ότι ο αποστολέας Sender μεταφέρει ένα αντίγραφο του πακέτου νούμερο ένα στο buffer εισόδου του δικτύου Network. Το πακέτο δεν απομακρύνεται από τη θέση Send, γιατί μπορεί να χρειαστεί να αναμεταδοθεί. Επίσης, ούτε ο μετρητής NextSend αυξάνεται διότι θα συνεχίζει να αναμεταδίδει ένα πακέτο, μέχρι να λάβει μία θετική βεβαίωση που να επιβεβαιώνει, ότι το πακέτο έχει παραληφθεί από τον αποστολέα Receiver.

Η θέση SP έχει μία μάρκα με τιμή 8. Από εδώ προκύπτει ότι στη μεταβλητή s πρέπει να καταχωρηθεί η τιμή 8. Η μεταβλητή r εμφανίζεται μόνο σε ένα εξερχόμενο τόξο, κι επομένως μπορεί να καταχωρηθεί στη μεταβλητή r οποιαδήποτε τιμή που ανήκει στον τύπο Ten1, χωρίς να

επηρεαστεί η ενεργοποίηση της μετάβασης TransmitPacket. Αυτό σημαίνει ότι έχουμε δέκα διαφορετικές ενεργοποιημένες καταχωρήσεις :

< n = 1, p = "Modellin", s = 8, r = 1 >
< n = 1, p = "Modellin", s = 8, r = 2 >
< n = 1, p = "Modellin", s = 8, r = 3 >
.....
.....
< n = 1, p = "Modellin", s = 8, r = 9 >
< n = 1, p = "Modellin", s = 8, r = 10 >

Η επιγραφή του εξερχόμενου τόξου από τη θέση TransmitPacket :

if Ok(s, r) then 1` (n, p) else empty

είναι μια πρόταση if, η οποία καλεί τη συνάρτηση Ok(s, r). Η συνάρτηση Ok(s, r) συγκρίνει τις τιμές των μεταβλητών s και r, και επιστρέφει την τιμή true, αν και μόνο αν $r \leq s$. Δηλαδή, στην προκειμένη περίπτωση, η συνάρτηση Ok(s, r) θα επιστρέφει την τιμή true για τις πρώτες οχτώ καταχωρήσεις, ενώ για τις τελευταίες δύο καταχωρήσεις θα επιστρέφει την τιμή false. Όταν η συνάρτηση Ok(s, r) είναι αληθής, η μετάβαση προσθέτει μία μάρκα με τιμή (1, "Modellin") στη θέση B. Διαφορετικά δεν προστίθεται καμία μάρκα, όπως δηλώνει και η θέση empty η οποία συμβολίζει το κενό πολλαπλό σύνολο. Η επιλογή μεταξύ των δέκα ενεργοποιημένων στοιχείων καταχώρησης είναι δίκαια κι έτσι, η πιθανότητα για επιτυχή μετάδοση είναι 80%, ενώ η πιθανότητα να χαθεί ένα πακέτο είναι 20%. Μεταβάλλοντας την τιμή της μάρκας στη θέση SP, μπορούμε να αλλάξουμε τις πιθανότητες. Αν η τιμή της μάρκας είναι 10, δεν χάνονται ποτέ πακέτα, ενώ αν είναι 0, χάνονται όλα τα πακέτα. Δηλαδή η θέση SP παριστάνει το ρυθμό επιτυχίας μετάδοσης πακέτων (Success rate for Packets).

Έστω τώρα, ότι η μετάβαση TransmitPacket εκτελείται με μία από τις οχτώ πρώτες καταχωρήσεις.

Η κατάσταση next rec χρησιμοποιείται σαν μετρητής με αρχική τιμή k=1 και κάθε φορά που πρόκειται να παραληφθεί ένα πακέτο (να ενεργοποιηθεί η receive packet) ελέγχει αν ο κωδικός του πακέτου είναι ίσος με τον μετρητή k. Αν είναι, τότε στέλνει την τιμή k στην άμεση μετάβαση και επιστρέφει με αυξημένη την τιμή του k κατά 1. Αλλιώς το πακέτο δε θα παραδοθεί μέχρι k=n.

Υπάρχουν πάλι δύο ενεργοποιημένες μεταβάσεις, η SendPacket η οποία θα αναμεταδώσει το πακέτο 1 μέσω της καταχώρησης (SendPacket, $\langle n = 1, p = \text{"Modellin"} \rangle$) και η TransmitPacket με το (ReceivePacket, $\langle n = 1, p = \text{"Modellin"}, k = 1, str = \text{""} \rangle$).

Στη μεταβλητή str καταγράφεται η συμβολοσειρά που έχει τυπωθεί στη received με αρχική τιμή "" δηλαδή η κενή. Στη φάση αυτή, γίνεται ο εξής έλεγχος:

```
if n=k andalso p<>stop then str^p else str
```

Αν ο μετρητής k συμφωνεί με τον κωδικό n και ταυτόχρονα η συμβολοσειρά p δεν παίρνει την τιμή της stop (η οποία έχει καταχωρηθεί ως '#####') τότε θα γίνει η παράθεση της str με την p. Αφού πλέον θα γίνει η παράδοση του πακέτου, στην received θα τυπωθεί "Modellin" και η str θα πάρει αυτή τη τιμή. Οπότε στην επόμενη παράδοση η εντολή str^p θα δώσει "Modelling and An".

Στη θέση C στέλνει βεβαίωση ότι έχει παραληφθεί το πακέτο, (αν $n=k$) ενώ αν $n \neq k$ δίνεται η πληροφορία ότι δεν έχει γίνει παράδοση, διότι το πακέτο έχει επαναληφθεί.

Η μετάβαση Transmit Acknow δέχεται δύο βέλη εισόδου με μία μεταβλητή το καθένα. Το n ή θα πάρει την τιμή του επόμενου πακέτου που θα πρέπει να σταλεί, είτε αν έχει πραγματοποιηθεί με επιτυχία η παράδοση ($n=k+1$) είτε αν δηλώθηκε να σταλεί πακέτο που ήδη έχει στην κατοχή του ο παραλήπτης (επανάληψη) και θα πρέπει να παραδοθεί το επόμενο ($n=k$). Η μεταβλητή s παίζει παρόμοιο ρόλο με την αντίστοιχη στην μετάβαση transmit packet. Αν η r (η οποία έχει καταχωρηθεί ως φυσικός αριθμός με τυχαίες τιμές από το 1 ως το 10) πάρει τιμή μικρότερη από την s τότε ενεργοποιείται η D με την μάρκα n. Αλλιώς αν $r > s$ δεν καταφτάνει καμία πληροφορία στην D που σημαίνει ότι η βεβαίωση λήψης πακέτου έχει χαθεί. Αυτό βέβαια συμβαίνει όταν το r πάρει τιμές 9 ή 10 από τις $\{1,2,3,\dots,10\}$ με πιθανότητα δηλαδή 20%.

Η receive acknow στέλνοντας την n στην next send ενημερώνει για την επόμενη παράδοση. Έτσι ο send ενημερώνεται για την επόμενη παραγγελία και η διαδικασία επαναλαμβάνεται.

Το πλήθος των επαναλήψεων της διαδικασίας ελέγχεται από τον χρήστη. Στο σύστημά μας δεν υπάρχει εγγύηση ότι οι μάρκες θα απομακρυνθούν από μια θέση με τη σειρά με την οποία προστέθηκαν. Κατά τη διάρκεια μιας προσομοίωσης του Χρωματισμένου δικτύου Petri, η θέση A μπορεί να περιέχει αρκετές μάρκες και κάθε μία από αυτές μπορεί να επιλεγεί ως η επόμενη για να μεταδοθεί στη θέση B. Το ίδιο ισχύει και για τις θέσεις B, C και D. Έτσι τα πακέτα μπορούν να προσπεράσουν το ένα το άλλο στις θέσεις A και B, ενώ οι βεβαιώσεις λήψης πακέτων μπορούν να προσπεράσουν η μία την άλλη στις θέσεις C και D. Εάν είναι επιθυμητό, μπορεί εύκολα να οριστεί μια πειθαρχία ουράς (queuing discipline). Για να γίνει αυτό, εφοδιάζουμε τις θέσεις A, B, C και D με ένα τύπο λίστας που παίρνει τιμές του προηγούμενου

τύπου της θέσης. Κάθε μία από αυτές τις θέσεις έχει πάντα μία μάρκα, της οποίας η αρχική τιμή είναι η άδεια λίστα. Στο ένα άκρο της λίστας εισάγουμε πακέτα, ενώ από το άλλο άκρο αφαιρούμε πακέτα. Μια τέτοια θέση ορίζεται ως θέση ουράς (queuing place).

Στο τελευταίο βήμα η p παίρνει την τιμή “#####” με αποτέλεσμα στην receive packet η p να μην παίρνει πλέον τιμές, η str έχει σταθερή τιμή “Modelling and Analysis by Means of Coloured Petri Nets” το $k=9$ η οποία τιμή μεταδίδεται μέχρι την next send ως $n=9$ και η send packet δεν ενεργοποιείται. Έτσι, μετά από 265 βήματα το δίκτυο παίρνει την τελική μορφή του σχήματος 19, όπου όλα τα πακέτα έχουν παραδοθεί και όλες οι μεταβάσεις είναι νεκρές [15].

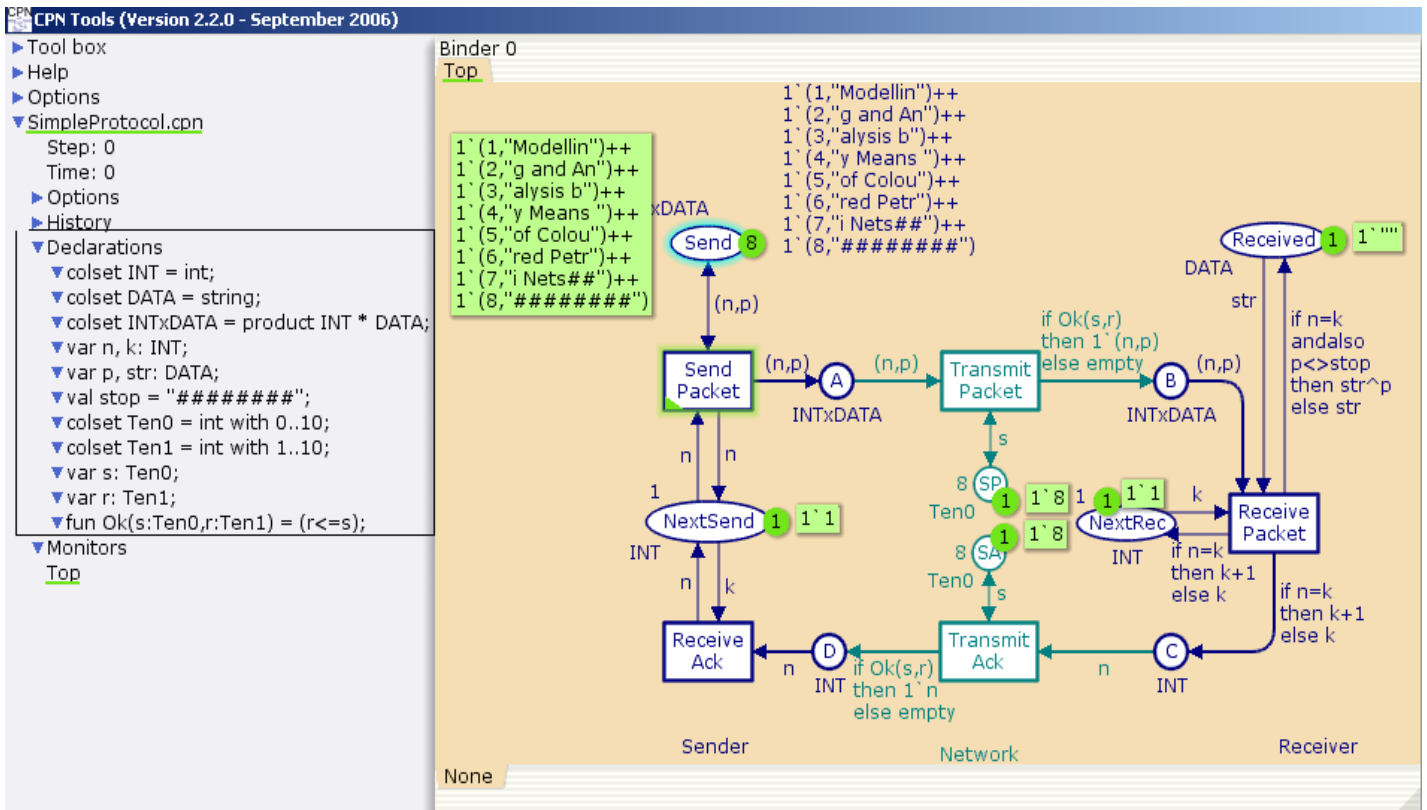
Επιπρόσθετα με τις επιγραφές τόξων, είναι πιθανό να επισυνάψουμε μια έκφραση boolean με μεταβλητές, σε κάθε μετάβαση. Η έκφραση boolean ονομάζεται φύλακας (guard) και ορίζει ότι δεχόμαστε μόνο καταχωρήσεις για τις οποίες η έκφραση boolean είναι αληθής. Παραδείγματος χάριν, θα μπορούσαμε να προσθέσουμε ένα φύλακα $n \leq 100$ στη μετάβαση SendPacket, ώστε να αποτρέπεται η αποστολή μηνυμάτων που αποτελούνται από περισσότερα από εκατό πακέτα.

Όπως είδαμε παραπάνω, σε ένα μαρκάρισμα μπορεί να υπάρχουν αρκετά ενεργοποιημένα στοιχεία καταχώρησης. Για την ακρίβεια, αυτά τα δύο στοιχεία καταχώρησης είναι ταυτόχρονα ενεργοποιημένα, που σημαίνει ότι οι μεταβάσεις ReceivePacket και SendPacket μπορούν να εκτελεστούν ταυτόχρονα (occur concurrently). Ο κανόνας για τον ταυτοχρονισμό είναι πολύ απλός. Ένα σύνολο από στοιχεία καταχώρησης είναι ταυτόχρονα ενεργοποιημένο, αν υπάρχουν τόσες πολλές μάρκες, έτσι ώστε κάθε στοιχείο καταχώρησης να μπορεί να πάρει τόσες μάρκες, όσες χρειάζεται, δηλαδή όσες προσδιορίζονται από τις επιγραφές των εισερχόμενων τόξων, χωρίς να μοιράζεται τις μάρκες με άλλα στοιχεία καταχώρησης. Έτσι, ένα βήμα (step) από ένα μαρκάρισμα στο επόμενο, μπορεί να εμπλέκει ένα πολλαπλό σύνολο από στοιχεία καταχωρήσεων. Το πολλαπλό σύνολο πρέπει να είναι πεπερασμένο και μη-κενό. Γενικά, μπορεί μία μετάβαση να είναι ταυτόχρονα ενεργοποιημένη με τον εαυτό της, χρησιμοποιώντας δύο διαφορετικές καταχωρήσεις ή την ίδια καταχώρηση δύο φορές.

Στο παράδειγμα του πρωτοκόλλου, κάθε μετάβαση έχει τουλάχιστον μία θέση εισόδου με μόνο μία μάρκα. Δύο στοιχεία καταχώρησης που αφορούν την ίδια μετάβαση, θα χρειαστούν και οι δύο αυτή τη μοναδική μάρκα. Άρα τα στοιχεία καταχώρησης βρίσκονται σε σύγκρουση μεταξύ τους και καμία μετάβαση δεν μπορεί να εκτελεστεί ταυτόχρονα με τον εαυτό της.

Η εκτέλεση ενός Χρωματισμένου δικτύου Petri περιγράφεται με τη βοήθεια μιας ακολουθίας εκτέλεσης (occurrence sequence), η οποία κάνει μια λίστα των προσιτών μαρκαρισμάτων και των βημάτων που εκτελούνται. Υπάρχουν πολλές ακολουθίες εκτέλεσης. Όταν ένα μαρκάρισμα έχει αρκετά ενεργοποιημένα στοιχεία καταχώρησης, οποιοδήποτε μη-κενό και χωρίς συγκρούσεις υποσύνολο τους, μπορεί να επιλεγεί για το επόμενο βήμα. Αυτό σημαίνει ότι ένα Χρωματισμένο δίκτυο Petri παρουσιάζει μη-ντετερμινιστική συμπεριφορά. Για παράδειγμα, το Χρωματισμένο δίκτυο Petri που μοντελοποιεί το απλό πρωτόκολλο επικοινωνίας, ορίζει ότι μπορούν να γίνουν αναμεταδόσεις πακέτων, αλλά δεν παρέχει λεπτομέρειες σχετικά με το πότε και το πόσο συχνά μπορεί να γίνει κάποια αναμετάδοση.

Η μετάβαση SendPacket είναι ενεργοποιημένη σε όλα τα προσιτά μαρκαρίσματα, εκτός από εκείνα στα οποία η τιμή της μάρκας στη θέση NextSend είναι μεγαλύτερη του αριθμού του τελευταίου πακέτου. Άρα οι αναμεταδόσεις πακέτων μπορούν να λάβουν χώρα οποιαδήποτε στιγμή και με οποιαδήποτε συχνότητα κι έτσι έχουμε ακολουθίες εκτέλεσης χωρίς αναμεταδόσεις (όπως η ακολουθία εκτέλεσης που θεωρήσαμε παραπάνω) και ακολουθίες εκτέλεσης με πολλές αναμεταδόσεις. Κατά την μοντελοποίηση μέσω των CPN δεν καθορίζονται οι συνθήκες κάτω από τις οποίες θα συμβεί μια αναμετάδοση. Για πολλούς λόγους όμως αυτό δεν είναι απαραίτητο. Τα περισσότερα Χρωματισμένα δίκτυα Petri χρησιμοποιούνται για να εξεταστεί η λογική και λειτουργική ορθότητα ενός συστήματος. Για αυτό το λόγο, είναι συνήθως αρκετό να ορίζουμε ότι οι αναμεταδόσεις πακέτων είναι επιτρεπτές, γιατί, παραδείγματος χάριν, το δίκτυο είναι αργό. Δεν είναι απαραίτητο ούτε και ωφέλιμο να καθοριστεί πόσο συχνά συμβαίνουν αναμεταδόσεις, γιατί το πρωτόκολλο πρέπει να ικανοποιεί όλα τα είδη των δικτύων, και αυτά που λειτουργούν τόσο καλά, ώστε να μην υπάρχουν αναμεταδόσεις και αυτά, στα οποία οι αναμεταδόσεις είναι συχνές.



Σχήμα 20. Το δίκτυο Top σχεδιασμένο με CPN Tool. Αριστερά στο πλαίσιο συντάσσονται οι εντολές προσομοίωσης.

Κατά την περιγραφή του συστήματος δώσαμε κάποια χαρακτηριστικά στα tokens ως προς το είδος, το χρόνο ή και στα βέλη ως προς τον τρόπο με τον οποίο διανέμουν της μάρκες, αλλά και περιορισμούς (συναρτήσεις, μεταβλητές εισόδου, εξόδου). Η διαδικασία αυτή αποτελεί το προγραμματιστικό μέρος της σχεδίασης ενός δικτύου CPN. Στο αριστερό μέρος ενός φύλλου CPN, επιλέγοντας “declaration”, μπορεί ο χρήστης να δώσει τις εντολές στο μοντέλο στο οποίο κατασκευάζει. Η σύνταξη των στοιχείων καταχώρησης είναι απαραίτητη διαδικασία μια και αλλιώς το δίκτυο παραμένει νεκρό` πραγματοποιούνται από τα CPN με την χρήση γλώσσας “Standard ML”, έναν συνδυασμό με βασικά χαρακτηριστικά των C++, fortran και mathematica. Το ίδιο το εργαλείο διευκολύνει το χρήστη όσο αφορά στο κομμάτι της σύνταξης των εντολών μιας και κάθε φορά που επισημαίνεται ένα λάθος αυτόματα κοκκινίζετε η γραμμή και βγαίνει κάτω αριστερά της οθόνης ένα μήνυμα λάθους.

Στη συνέχεια ακολουθεί πίνακας με τις βασικότερες εντολές τις γλώσσας ML με επεξήγηση για την κάθε εντολή- τύπο και κάποιο απλό παράδειγμα. (Πίνακας 1).

Είδος	Τύπος	Επεξήγηση	Παράδειγμα
	$\sim i$	Άρνηση του i	
	abs i	Απόλυτη τιμή του i	
	int.min(i,j) ή int.max(i,j)	Μικρότερη ή μεγαλύτερη τιμή των ακεραίων i,j .	
	s^t	Παράθεση συμβολοσειρών s και t	
	$\#j(i_1,i_2,\dots,i_n)$	Τυπώνει το j -οστό στοιχείο i_j	$\#(2,"abc")$ τυπώνει "abc"
	rev l	Επιστρέφει την ανάστροφη λίστα της l	
	map $f l$	Χρησιμοποιεί ως πεδίο ορισμού τα στοιχεία της λίστας l και εξάγει σε λίστα της τιμές της f	
	$l_1 \wedge l_2$	Παραθέτει δύο λίστες	
Colset	With	Η K δέχεται τιμές μόνο man ή woman ή child.	colset $K=$ with man woman child
	Index	car(3),car(4),car(5),..., car(8)	colset $L=$ index car with 3..8
	int	Ακέραιος	int with 1..10
	Timed	Χρονική μεταβλητή	colset $T=$ int timed
	bool	boulean	bool with {no,yes}
	String	Συμβολοσειρά	string with "a".. "z"
	product	Γινόμενο όρων	product $U*I$
	Record	Δήλωση τύπων μεταβλητών	record se:SITES*re:SITES*no:INT
Fun		Τυπώνει true αν ο i	fun even $i=(i \bmod$

		είναι πολ- λαπλάσιο του 2,αλλιώς false	2)=0;
		Τυπώνει τον μικρότερο αριθμό	fun ok(i,j)= if i<=j then I else j
		Αναδρομική συνάρτηση: όταν ο ακέραιος n>1 τυπώνει τον παραγοντικό του n!, αλλιώς 1	fun fac(n:int)=if n>1 then n*fac(n-1) else 1
Val		Ανάθεση τιμής	val a=3
		Ή μάρκας	val M=1`2++2`8
		Ή τύπο συνάρτησης	val ad=fn x => x+3
		Εισαγωγή λίστας	val list=[1,2,5]
		Εισαγωγή κειμένου	val last=DATA("#####")
	Bernoulli(p:real):int	Κατανομή bernoulli	
	Normal(n:real,y:real): real	Κανονική κατανομή	
	Poisson(m:real):int	Κατανομή Poisson	
	Exponential(r:real):re al	Εκθετική κατανομή	
	Discrete(a:int,b:int):i nt	Διακριτή κατανομή	

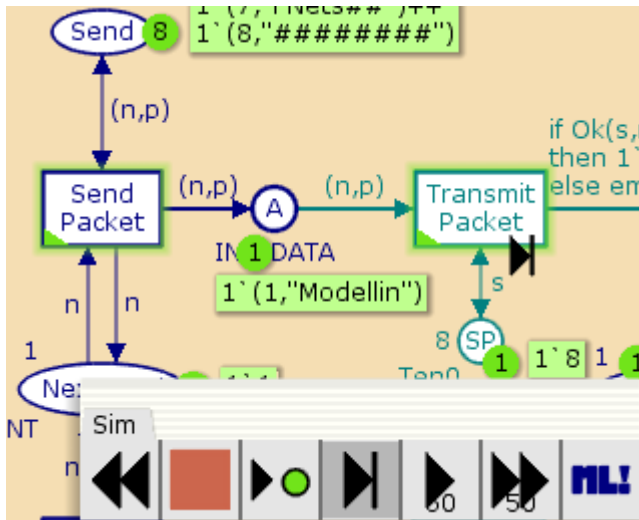
Πίνακας 1. Βασικές εντολές της γλώσσας *Standard ML*

2.2.2 Προσομοίωση και καταγραφή δεδομένων στα CPN

Προκειμένου να συγκεντρωθούν και να αναλυθούν τα δεδομένα μετά το τέλος της διαδικασίας αλλά και να διατυπωθούν χρήσιμα συμπεράσματα για το μοντέλο και τον τρόπο λειτουργίας του, το CPN είναι εφοδιασμένο με κατάλληλες εντολές. Με τη βοήθεια αυτών των επιλογών γίνεται δυνατή η καταγραφή των ενδιάμεσων σταθμών μιας μάρκας σε ένα δίκτυο, η γραφική αναπαράσταση των προσιτών μαρκαρισμάτων και η εξαγωγή χρήσιμων σχεδιαγραμμάτων. Η **αναφορά προσομοίωσης (simulation report)**, η **ανάλυση δεδομένων (state space)** και τα **occurrence graphs ή δέντρα προσιτότητας** αποτελούν πολύτιμα εργαλεία τόσο για την ερμηνεία της λειτουργίας ενός μοντέλου όσο και για την διατύπωση συμπερασμάτων και την βελτίωση στη πολυπλοκότητα και τον χρόνο περάτωσής του.

2.2.2.1 Χειροκίνητη προσομοίωση

Καθώς δομούνται τα τμήματα ενός χρωματισμένου δικτύου Petri ελέγχονται με την βοήθεια ενός προσομοιωτή (ταμπλέτα simulator). Πλέον αυτός που μοντελοποιεί, είναι δυνατόν να ελέγχει και να επιθεωρεί την διαδικασία χρησιμοποιώντας της επιλογές της ταμπλέτας. Έτσι μπορεί η λειτουργία να σταματήσει, να μεταπηδήσει απ' ευθείας είτε στο πρώτο είτε στο τελευταίο βήμα της προσομοίωσης. Η πιο χρήσιμη και σημαντική λειτουργία είναι να επιλέξει την μετάβαση η οποία θα πυροδοτήσει. Στο σχήμα 21 υπάρχουν δύο ενεργοποιημένες μεταβάσεις και ο χρήστης επιλέγει να ενεργοποιηθεί η "transmit packet". Αν είχε επιλεγθεί «play» η επόμενη μετάβαση θα ορίζονταν από το σύστημα. Η διαδικασία αυτή αποτελεί την πιο απλή μορφή ελέγχου στο σύστημα που σχεδιάστηκε μια και είναι χειροκίνητη και πιο διαδραστική (interactive) αφού είναι πολύ αργή και πραγματοποιείται βήμα-βήμα. Συνήθως χρησιμοποιείται για να εξακριβωθεί αν το δίκτυο δουλεύει κανονικά. Δεν γίνεται δηλαδή ο χρήστης να παρέμβει σε μια μεταγενέστερη πιθανή πυροδότηση αλλά ούτε να προβλέψει τις επόμενες επιλογές μαρκαρισμάτων, εάν δεν φτάσει στο αμέσως προηγούμενο βήμα.



Σχήμα 21. Παλέτα προσομοίωσης

2.2.2.2 Ανάλυση προσομοίωσης (simulation report)

Για να αποφευχθεί η χειροκίνητη καταγραφή δεδομένων δόθηκε από το εργαλείο η δυνατότητα να τυπώνονται σε μορφή σημειωματάριου όλες οι μεταβάσεις και οι τιμές των μεταβλητών που βρίσκονται καταχωρημένες σε αυτές, ενώ το δίκτυο βρίσκεται σε λειτουργία (Πίνακας 2). Δημιουργείτε έτσι ένας φάκελος output με το σημειωματάριο αυτό και ό, τι άλλο εξάγουμε στη συνέχεια .

```

1      0      Send_Packet @ (1:Top)
- p = "Modellin"
- n = 1
2      0      Transmit_Packet @ (1:Top)
- n = 1
- p = "Modellin"
- s = 8
- r = 7
3      0      Send_Packet @ (1:Top)

```

```

- p = "Modellin"
- n = 1
4      0      Receive_Packet @ (1:Top)
- str = ""
- k = 1
- n = 1
- p = "Modellin"
5      0      Transmit_Ack @ (1:Top)
- n = 2
- s = 8
- r = 4
6      0      Receive_Ack @ (1:Top)
- k = 1
- n = 2
7      0      Send_Packet @ (1:Top)
- p = "g and An"
- n = 2
8      0      Send_Packet @ (1:Top)
- p = "g and An"
- n = 2
9      0      Transmit_Packet @ (1:Top)
- n = 1
- p = "Modellin"
- s = 8
- r = 10
10     0      Transmit_Packet @ (1:Top)
- n = 2
- p = "g and An" - s = 8 - r = 10

```

Πίνακας 2. *Simulation report: Αναφορά προσομοίωσης του μοντέλου «Top».*

Στον πίνακα 2 συναντάμε την αναφορά προσομοίωσης του αρχικού μας μοντέλου για τα 10 πρώτα βήματα για χάριν απλούστευσης. Η λειτουργία του είναι ρυθμισμένη να διακόπτεται στα 50 βήματα ανεξαρτήτως αν υπάρχουν ζωντανές μεταβάσεις. Επιλέγοντας

χειροκίνητα να συνεχιστεί η προσομοίωση, τα επόμενα 50-100 βήματα επισυνάπτονται αυτόματα στο σημειωματάριο και έτσι στο τέλος συγκεντρώνονται όλα τα αποτελέσματα μαζί.

- Κάθε βήμα δηλώνεται με έναν αριθμό (χρωματισμένο με μπλε), ώστε να γίνεται αντιληπτή η σειρά με την οποία πραγματοποιήθηκαν τα μαρκαρίσματα.
- Μετά τον αριθμό αυτό βλέπουμε ότι τυπώνεται το 0. Αυτό σημαίνει ότι τα στοιχεία καταχώρησης δηλώθηκαν με αυτόματο τρόπο.
- Ακολουθεί το όνομα της μετάβασης που μαρκαρίστηκε. Μετά το @ αναγράφεται το φύλλο προσομοίωσης (χρησιμοποιείται στα hierarchical CPN).
- Στη συνέχεια καταγράφονται οι τιμές που παίρνουν οι μεταβλητές στην αντίστοιχη μετάβαση. Αν μία μεταβλητή δεν είναι καταχωρημένη ούτε στην είσοδο ούτε στην έξοδο της μετάβασης, τότε απλά παραλείπεται.

2.2.2.3 Διαγράμματα

Ένας δεύτερος τρόπος καταγραφής των αποτελεσμάτων μιας προσομοίωσης είναι η χρήση ενός **διαγράμματος διαδοχής μηνυμάτων (message sequence chart)**, το οποίο παρέχει γραφικά, μια συνολική εικόνα των δράσεων σε ένα Χρωματισμένο δίκτυο Petri. Τα διαγράμματα διαδοχής μηνυμάτων δημιουργούνται με τη χρήση τμημάτων κώδικα (**code segments**), τα οποία επισυνάπτονται στις μεταβάσεις. Αυτά τα τμήματα κώδικα καλούν συναρτήσεις μιας βιβλιοθήκης (predefined function), η οποία παρέχεται μαζί με τον προσομοιωτή των Χρωματισμένων δικτύων Petri. Όταν εκτελείται μια μετάβαση, εκτελείται και το αντίστοιχο τμήμα κώδικα [15].

Για την ενημέρωση του διαγράμματος του σχήματος 23 χρησιμοποιήθηκαν τα τμήματα κωδικών στις μεταβάσεις send packet και transmit packet του σχήματος 22.

Το MSC diagram είναι ένας δείκτης προς το διάγραμμα διαδοχής μηνυμάτων, ενώ το mkst_col_INTxDATA είναι μια συνάρτηση που παριστάνει την τιμή (n,p) ως συμβολοσειρά. Τα τμήματα κώδικα των υπόλοιπων τριών μεταβάσεων είναι παρόμοια.

```

input (n,p);
action
  MSC.Message (!MSCdiagram)
    {sender = "sender",
      receiver = "sendernet",
      label = "SendPack:"^NEWLINE^
              (mkst_col'INTxDATA(n,p))};

```

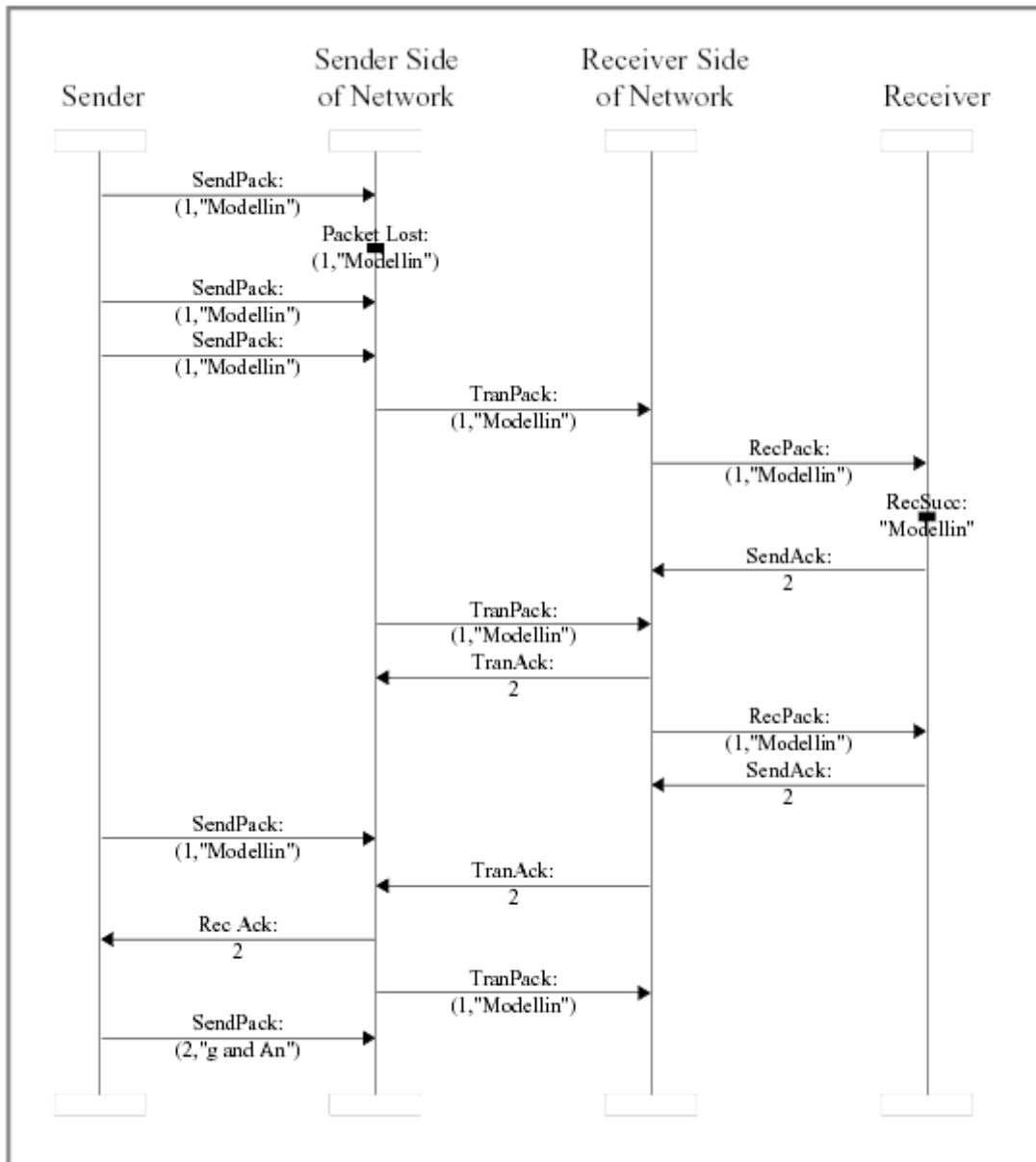
```

input (n,p,s,r);
action
  if Ok(s,r) then
    MSC.Message (!MSCdiagram)
      {sender = "sendernet",
        receiver = "receivernet",
        label = "TranPack:"^NEWLINE^
                (mkst_col'INTxDATA(n,p))}
  else
    MSC.Processmark (!MSCdiagram)
      {process = "sendernet",
        label = "Packet Lost:"^NEWLINE^
                (mkst_col'INTxDATA(n,p))};

```

Σχήμα 22. Τα code segments των μεταβάσεων send packet και transmit packet αντίστοιχα

Μια εκτέλεση της μετάβασης SendPacket υποδεικνύεται από ένα οριζόντιο βέλος μεταξύ της πρώτης και της δεύτερης κατακόρυφης γραμμής. Το βέλος έχει ως επιγραφή το πακέτο που μεταδίδεται στο δίκτυο Network. Ανάλογα, μια επιτυχής εκτέλεση της μετάβασης TransmitPacket υποδεικνύεται από ένα οριζόντιο βέλος μεταξύ της δεύτερης και της τρίτης κατακόρυφης γραμμής. Αν όμως το πακέτο χαθεί, υπάρχει στη δεύτερη γραμμή μόνο ένα μικρό ορθογώνιο. Μια εκτέλεση της μετάβασης ReceivePacket υποδεικνύεται από δύο βέλη : Ένα βέλος για το πακέτο που παραλήφθηκε και ένα βέλος για τη βεβαίωση λήψης πακέτου που αποστέλλεται. Αν το πακέτο που παραλήφθηκε είναι το σωστό, τότε υπάρχει επίσης, ένα μικρό ορθογώνιο μεταξύ των δύο βελών. Οι εκτελέσεις της μετάβασης TransmitAcknowledgement υποδεικνύονται παρόμοια με τις εκτελέσεις της μετάβασης TransmitPacket, με τη διαφορά ότι τα βέλη σχεδιάζονται με διεύθυνση από δεξιά προς τα αριστερά και τα ορθογώνια είναι τοποθετημένα στη Τρίτη κατακόρυφη γραμμή. Τέλος οι εκτελέσεις της μετάβασης ReceiveAcknowledgement υποδεικνύονται παρόμοια με τις εκτελέσεις της μετάβασης SendPacket, με τα βέλη όμως να δείχνουν από δεξιά προς τα αριστερά [15].



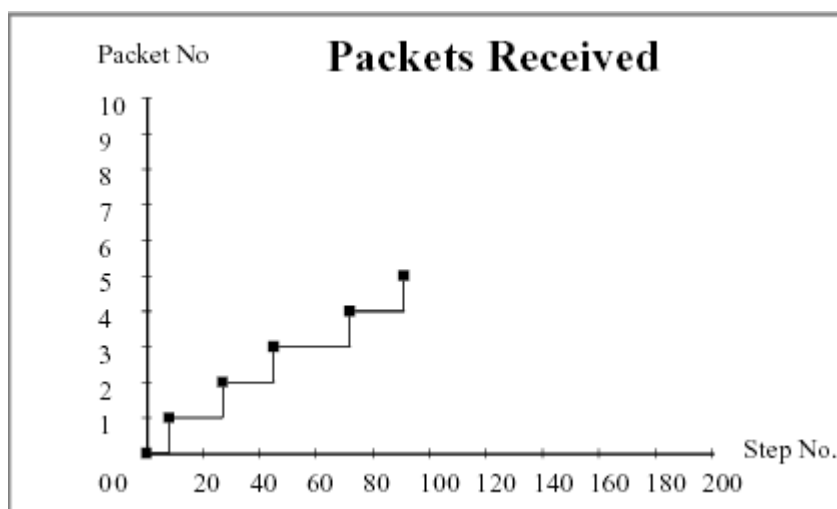
Σχήμα 23. Διάγραμμα διαδοχής μηνυμάτων του μοντέλου Top.

Τα τμήματα κώδικα μπορούν επίσης να χρησιμοποιηθούν για την ενημέρωση και άλλων ειδών διαγραμμάτων.

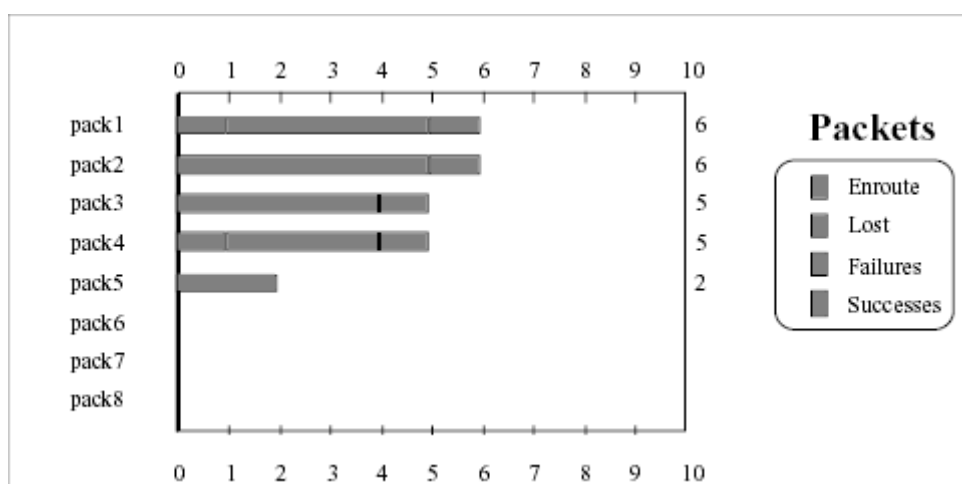
Στο σχήμα 24 έχουμε διάγραμμα γραμμής. Στο κάθετο άξονα είναι αριθμημένα τα πακέτα που διανέμονται ενώ στον οριζόντιο φαίνεται ο αριθμός των βημάτων που χρειάστηκαν για να πραγματοποιηθεί με επιτυχία η μεταφορά τους. Για παράδειγμα το πακέτο No1 παραλήφθηκε στο 10^ο βήμα, το No2 πακέτο στο 25^ο βήμα κτλ. Το τμήμα κώδικα προσαρμόζεται στη μετάβαση receive για να έχουμε το παρακάτω αποτέλεσμα. Η προσομοίωση πραγματοποιήθηκε στα 100 βήματα.

Το οριζόντιο ραβδόγραμμα μας ενημερώνει για το πόσες φορές έχει αποσταλεί ένα πακέτο και με τι αποτέλεσμα. Για παράδειγμα, το πακέτο Νο1 έχει χαθεί μία φορά, η παράδοσή του κρίθηκε αποτυχημένη 4 φορές δηλαδή στάλθηκε με λάθος σειρά και μόνο την τελευταία φορά παραλήφθηκε με επιτυχία. Συνολικά έχει αποσταλεί 6 φορές, όπως και το Νο2. Τέλος το πακέτο Νο5 έχει αποσταλεί μόλις 2 φορές όπου και τις δύο είναι σε κατάσταση enroute δηλαδή σε κατάσταση αναμονής στις θέσεις A και B του απλού πρωτοκόλλου.

Το τελευταίο διάγραμμα ενημερώνεται περιοδικά σε τακτά χρονικά διαστήματα ανάλογα με επιλογές του χρήστη.



Σχήμα 24. Διάγραμμα γραμμής της μετάβασης Receive.

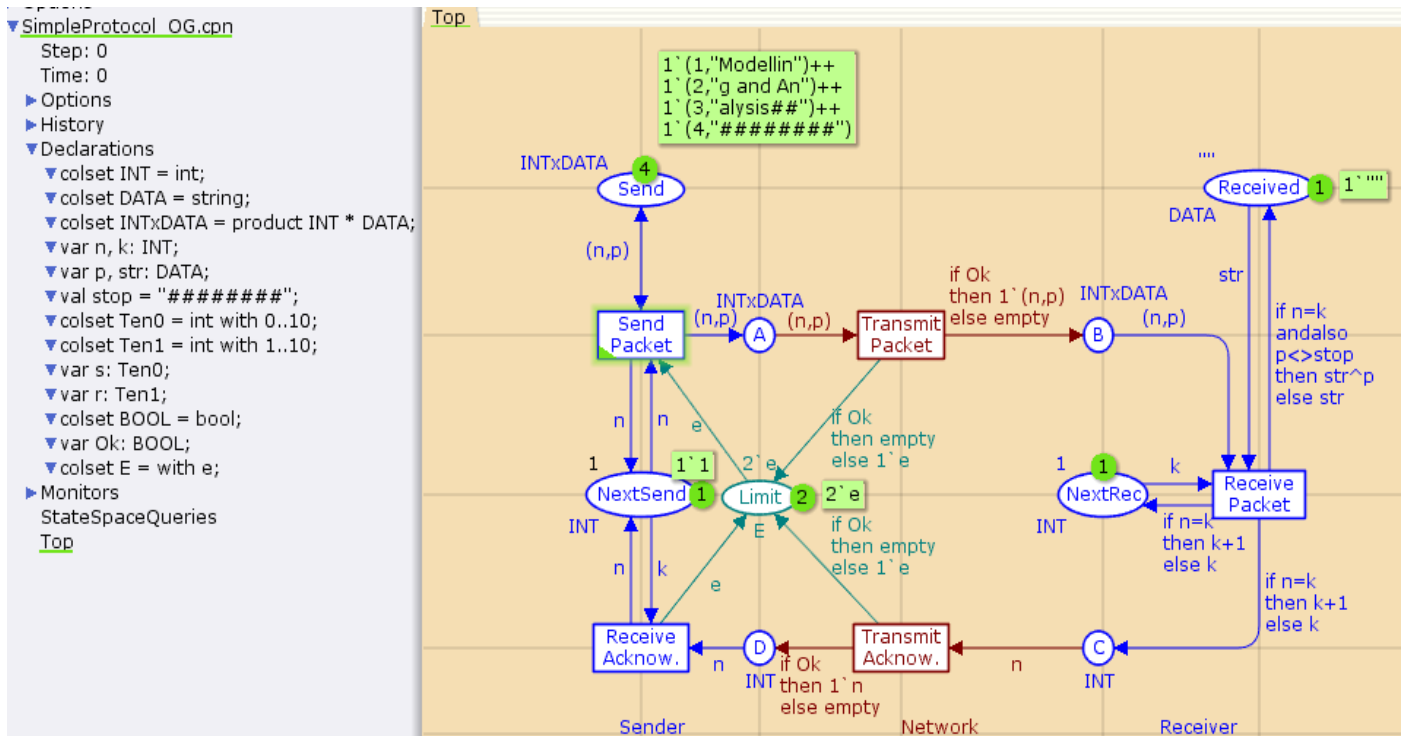


Σχήμα 25. Αναφορά κατάστασης των πέντε πρώτων πακέτων.

2.2.2.4 *State Space και occurrence graphs*

Κατά την περιγραφή των δέντρων προσιτότητας των petri nets του σχήματος 17 είχε γίνει αναφορά στα occurrence graphs. Πρόκειται για γραφήματα όπου αναπαριστούν βήμα-βήμα την πρόοδο της προσομοίωσης. Αναγράφεται σε κάθε αριθμό βήματος οι ενεργοποιημένες μεταβάσεις, οι τιμές των μεταβλητών σε αυτές και οι επόμενες μεταβάσεις που πρόκειται να μαρκαριστούν μετά την πυροδότηση αυτών. Τα διαδοχικά βήματα συνδέονται μεταξύ τους με βέλη.

Πριν τη δημιουργία και περιγραφή αυτών των διαγραμμάτων (ή διαγράμματα προσιτότητας) προηγείται η ενημέρωση του χώρου καταστάσεων (state space). Από την κατάλληλη παλέτα ο χρήστης επιλέγει να καταχωρήσει τον χώρο καταστάσεων στην μνήμη. Στο αρχείο CPN Tools.Ink που ανοίγει ταυτόχρονα με το CPN-Tool αναγράφεται αυτή η καταχώρηση. Στη συνέχεια «υπολογίζεται» ο χώρος καταστάσεων (calculate state machine). Στη φάση αυτή απομνημονεύεται από το σύστημα το state space με τα ονόματα των κόμβων, το σύνολο των εισερχόμενων και εξερχόμενων βελών αλλά και η συνδεσιμότητα τους. Αφού υπολογίσουμε και το SCC graph (πίνακας που θα αναλυθεί στη συνέχεια) σώζουμε μία αναφορά (save report) αποθηκεύοντας το αρχείο σε φάκελο. Έτσι ολοκληρώνεται η διαδικασία σχεδίασης και ανάλυσης ενός occurrence graph. Πολλές φορές βέβαια η πολυπλοκότητα του γράφου προσιτότητας (occurrence graph) είναι τόσο μεγάλη ώστε να είναι αδύνατον να διαβαστεί ολόκληρο το γράφημα, πόσο μάλλον να ερμηνευτεί και αυτό εξαιτίας του αριθμού των τόξων και το σύνολο των πιθανών μαρκαρισμένων καταστάσεων. Γι' αυτόν το λόγο οι τροποποιήσεις του δικτύου με σκοπό τη μείωση μιας πολύ μεγάλης πολυπλοκότητας κρίνεται απαραίτητη [15].



Σχήμα 26. Το βοηθητικό μοντέλο του Top που χρησιμοποιήθηκε για την εκτύπωση των διαγραμμάτων occurrence.

Είναι αυτονόητο ότι στο απλό πρωτόκολλο, οι πιθανές πυροδοτήσεις είναι αρκετές και οι ακολουθίες των μαρκαρισμάτων αυξάνονται εκθετικά. Είναι αναγκαίο λοιπόν να περιορίσουμε τις επιλογές στο μοντέλο μας. Για τον σκοπό αυτό, δημιουργούμε ένα παραπλήσιο μοντέλο με το αρχικό (σχήμα 26) με τη διαφορά ότι λιγοστεύουμε τα πακέτα από οχτώ σε τέσσερα, για να μην έχουμε πολλά βήματα (αντίστοιχα στα δέντρα προσιτότητας επίπεδα). Επίσης προσθέτουμε μια καινούρια κατάσταση την Limit για να περιορίσουμε τον αριθμό των πακέτων και των βεβαιώσεων λήψης πακέτων που μπορούν να υπάρχουν ταυτόχρονα στο δίκτυο Network, δηλαδή στις θέσεις A, B, C και D. Η νέα κατάσταση έχει τύπο E με μοναδική πιθανή τιμή e που σημαίνει ότι οι μάρκες σε αυτή τη θέση δεν μπορεί να είναι σε μορφή κειμένου. Τέλος στο νέο μας μοντέλο απλουστεύεται ο μηχανισμός απόφασης, για την μετάδοση πακέτων και λήψη βεβαιώσεων. Κάτι τέτοιο είναι σημαντικό γιατί στα συγκεκριμένου τύπου διαγράμματα δεν έχει νόημα όταν τα πακέτα παραδίδονται ή χάνονται με μια συγκεκριμένη πιθανότητα. Έτσι, αντικαθιστούμε την συνάρτηση ok με την μεταβλητή ok τύπου BOOL, όπου δέχεται δύο τιμές: Την true και την false. Η τιμή που θα πάρει η συγκεκριμένη μεταβλητή ορίζεται τυχαία.

Η αρχική μάρκα της limit είναι 2'e. Ουσιαστικά η επιτυχή παράδοση ενός πακέτου μέχρι να καθοριστεί το επόμενο, ισοδυναμεί με την διατήρηση του σταθερού αθροίσματος των

μαρκών σε αυτή τη θέση, δηλαδή 2. Όταν πυροδοτεί η send packet μία μάρκα αφαιρείται από την limit (σύνολο 1'ε). Στην μετάβαση transmit packet όταν (1)ok=true (σωστή σειρά πακέτου) και στην transmit acknow. η (2)ok=true (δεν χάθηκε αλλά παραδόθηκε) τότε ενεργοποιείται η receive acknow. και επιστρέφει ένα 1'ε στην limit (σύνολο 2'ε). Αν (1)ok=true και (2)ok=false τότε το πακέτο ενώ μεταφέρεται με τη σωστή σειρά, χάνεται στη πορεία και μεταφέρεται 1'ε στην limit (σύνολο 2'ε) και η receive acknow δεν ενεργοποιείται. Αν (1)ok= false το πακέτο που έρχεται δεν είναι στη σωστή σειρά οπότε 1'ε μεταφέρεται στη limit (σύνολο 1'ε) και η προσομοίωση συνεχίζει με την επιλογή του επόμενου πακέτου.

Κατά την προσομοίωσή του, γίνεται αντιληπτό ότι σχεδόν σε όλα τα βήματα παραπάνω από μία μεταβάσεις είναι ταυτόχρονα ενεργοποιημένες. Ο συγχρονισμός αυτός θα έχει ως αποτέλεσμα οι ακολουθίες των πιθανών μαρκαρισμάτων και να μεγαλώνουν αλλά και να πληθαίνουν με αποτέλεσμα ο γράφος να γίνεται ιδιαίτερα πολύπλοκος. Αντίθετα, αν δώσουμε αρχική μάρκα στην limit 1'ε τότε το σχεδιάγραμμα απλουστεύεται κατά πολύ. Μια τέτοια επιλογή δηλώνει εκτός των άλλων ότι ένα πακέτο δεν μπορεί να προσπεράσει κάποιο άλλο κατά την διακίνησή τους.

Τα σχήματα 27 και 28 είναι το occurrence graph του μοντέλου TopOG. Για ευκολία ανάγνωσης δίνονται τα 11 πρώτα και 9 τελευταία βήματα. Στην πραγματικότητα το διάγραμμα είναι ενιαίο και μεσολαβούν άλλα 13 βήματα.

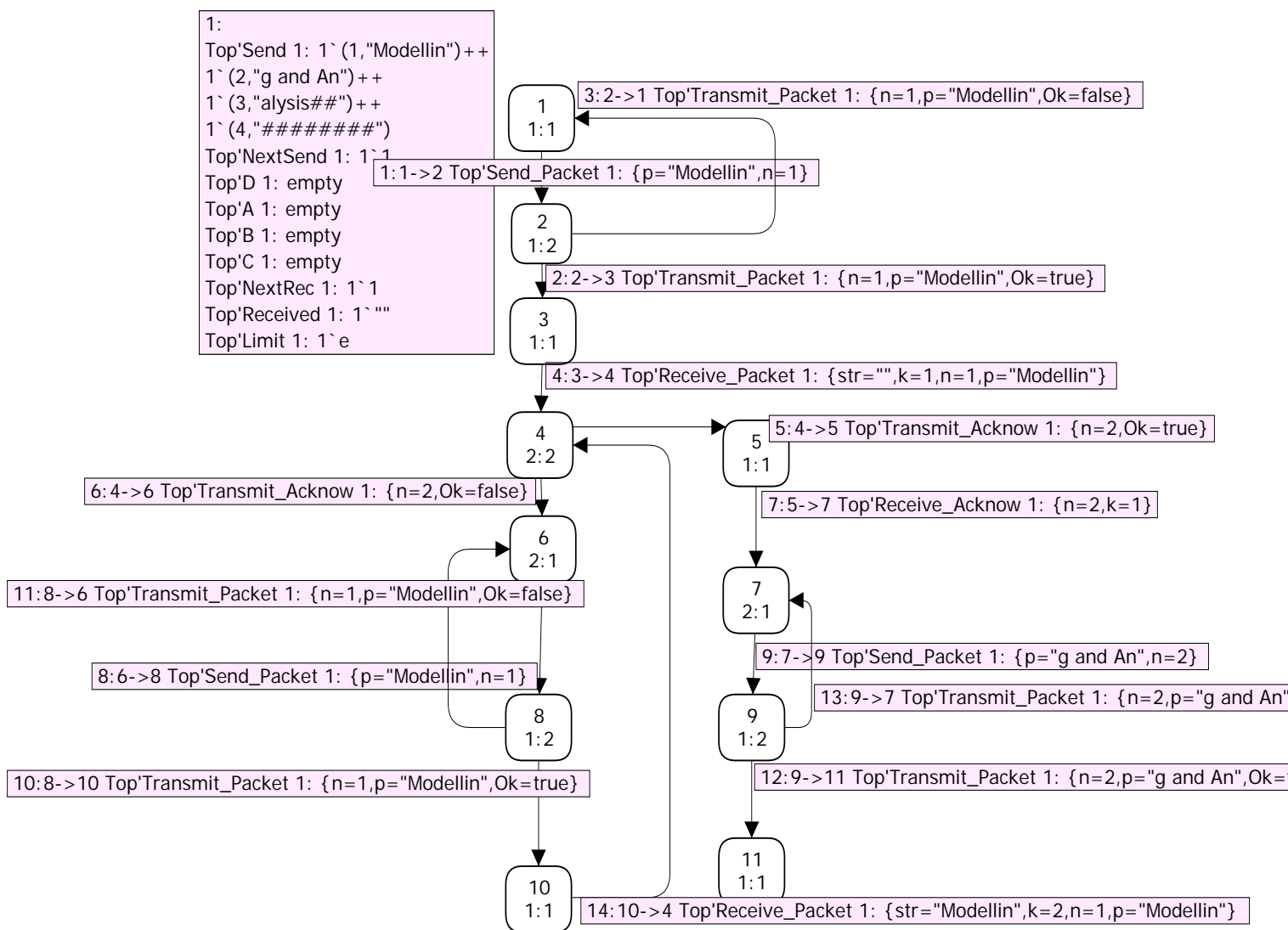
Κάθε ορθογώνιο αντιστοιχεί με μία πιθανή ενεργοποιημένη μετάβαση, όπου αναγράφεται ο αριθμός του βήματος και το σύνολο των προγόνων και απογόνων. (π.χ. για το 2^ο βήμα έχουμε 1:2 προγόνους και απογόνους αντίστοιχα). Έτσι το εξερχόμενο βέλος οδηγεί στις επόμενες πιθανές μαρκαρισμένες καταστάσεις (απόγονοι) ενώ το βέλος που εισέρχεται σημαίνει ότι στην επόμενη πυροδότηση πιθανόν να μαρκαριστεί το συγκεκριμένο place (πρόγονος). Το κομμάτι αυτό καταγράφεται με τη βοήθεια του state space.

Σε κάθε βέλος τα χρωματισμένα πλαίσια ενημερώνουν για κάποια χαρακτηριστικά αυτού του βήματος.

2:2->3 Top'Transmit_Packet 1: {n=1 ,p="Modellin",Ok=true}

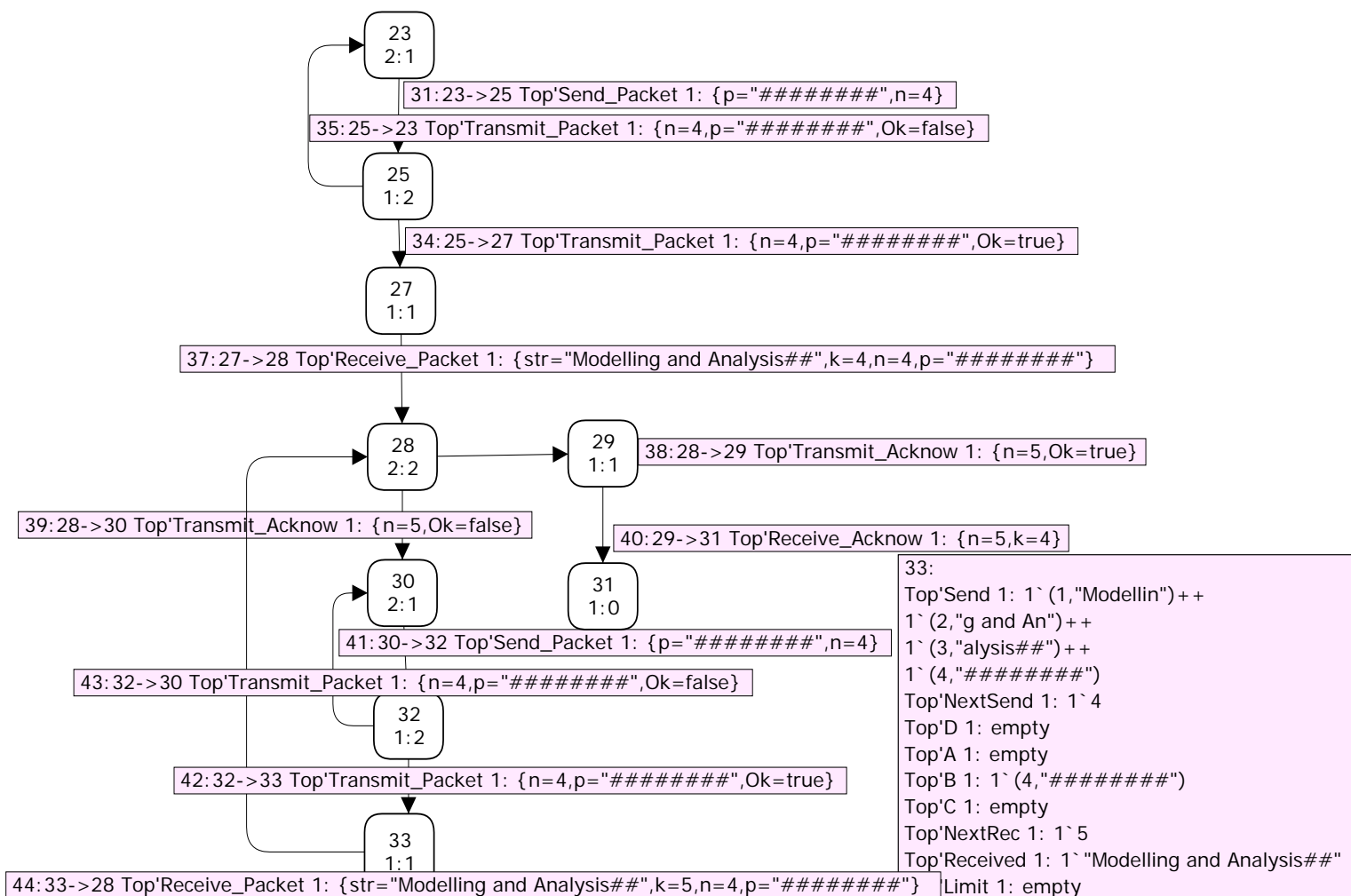
3:2->1 Top'Transmit_Packet 1: {n=1 ,p="Modellin",Ok=false}

Για παράδειγμα, στο δεύτερο βήμα η transmit packet έχει δύο απογόνους. Η πυροδότησή της δηλαδή συνεπάγεται την ενεργοποίηση δύο μεταβάσεων αυτήν του βήματος 1 (Send Packet) και του βήματος 3 (Receive Packet). Ανάλογα με την τιμή που θα πάρει η ok θα επιλεγθεί και η επόμενη ακολουθία μαρκαρισμάτων. Αν ok=false ακολουθούμε το 3^ο βέλος το οποίο συνδέει το 2^ο και 1^ο βήμα, όπου οι μεταβλητές παίρνουν τις τιμές n=1 p="Modellin" ok=false και ενεργοποιείτε η Send Packet. Αν ok=true τότε περιμένουμε να ενεργοποιηθεί η επόμενη μετάβαση Receive Packet η οποία με τη σειρά της έχει έναν μόνο απόγονο. Δηλαδή στο βήμα 3 μόνο αυτό το transition είναι έτοιμο για πυροδότηση.



Σχήμα 27. Οι 11 πρώτες μαρκαρισμένες καταστάσεις κατά την προσομοίωση του μοντέλου όταν η limit είναι μαρκαρισμένη με 1`e.

Τα μεγαλύτερα πλαίσια δίπλα στο πρώτο και τελευταίο βήμα ονομάζονται toggle descriptors και ενημερώνουν για τις μάρκες των places στο συγκεκριμένο στιγμιότυπο. Η επιλογή αυτή αφορά όλα τα βήματα. Στο τελευταίο στάδιο της διαδικασίας (No33) το αντίστοιχο toggle descriptor δίνει ότι στην Send υπάρχουν ακόμα τα πακέτα, 1`1("Modellin")++ 1`2("g and An")++ 1`3("alysis##")++ 1`4("#####"), η NextSend διαθέτει 1`4 μάρκες ενώ θα έπρεπε να έχει 1`5 (4 πακέτα σύν 1 από τη Limit για επιβεβαίωση). Κάτι τέτοιο σημαίνει ότι το τελευταίο πακέτο ή δεν θα παραδόθηκε με τη σειρά του, πράγμα αδύνατον γιατί μόνο αυτό απέμεινε, ή θα χάθηκε. Όμως ενώ οι A και D είναι άδειες, η B έχει την μάρκα (4, "#####") και η επόμενη μετάβαση απαγορεύει την διακίνηση αυτού του πακέτου. Στη Received έχουν φτάσει όλα τα πακέτα (1`"Modelling and Analysis##") ενώ η Limit είναι άδεια. Αυτό συμβαίνει γιατί το τελευταίο πακέτο παραμένει σε κατάσταση αναμονής στη θέση B και δεν έχει ενεργοποιηθεί η Transmit Acknow. ώστε να γίνει η επιβεβαίωση αποστολής.



Σχήμα 28. Οι 9 τελευταίες μαρκαρισμένες καταστάσεις κατά την προσομοίωση του μοντέλου όταν η limit είναι μαρκαρισμένη με 1'ε.

Αν το δίκτυο τεθεί πάλι σε λειτουργία το πιο πιθανό είναι να έχουμε διαφορετικές ακολουθίες μαρκαρισμάτων άρα και διαφορετικό διάγραμμα προσιτότητας.

Στα σχήματα 27 και 28 φαίνεται ότι ο γράφος αναπτύσσει αλγοριθμική συμπεριφορά μιας και κάποια κομμάτια του επαναλαμβάνονται. Οι υπογράφοι {4,5,7,9,11} και {29,28,30,32,33} έχουν ίδια δομή. Η μόνη διαφορά είναι ότι οι καταστάσεις του δεύτερου υπογράφου διαθέτουν 3 παραπάνω πακέτα από αυτές του πρώτου. Αν σχεδιάσουμε και το μεσαίο τμήμα του χώρου καταστάσεων, θα βρούμε ακόμα δύο υπογράφους με την ίδια μορφή.

Στην πιο σύνθετη περίπτωση, όπου ένα πακέτο μπορεί να προσπεράσει ένα άλλο θα χρειαστεί να κατασκευάσουμε ένα χώρο καταστάσεων όταν η θέση Limit έχει 2'ε. Όπως είπαμε

σε αυτή τη περίπτωση το διάγραμμα γίνεται αρκετά μεγάλο, οπότε θα περιοριστούμε στη περιγραφή μόνο του state space report η οποία παρέχει χρήσιμες πληροφορίες σχετικά με τη συμπεριφορά του Χρωματισμένου δικτύου Petri.

Η αναφορά του χώρου καταστάσεων αποτελείται από τέσσερα μέρη:

Περιέχει **στατιστικές (statistical)** πληροφορίες σχετικά με το μέγεθος του χώρου καταστάσεων (πίνακας 3_1). Βλέπουμε ότι ο συγκεκριμένος χώρος καταστάσεων έχει 428 κόμβους και 1130 τόξα και ότι υπολογίστηκε ο πλήρης χώρος καταστάσεων (Status: Full) μέσα σε ένα δευτερόλεπτο (Secs: 1). Το στατιστικό μέρος περιέχει επίσης πληροφορίες σχετικά με το SCC- γράφο (SCC-graph), δηλαδή τον αριθμό των ισχυρά συνδεδεμένων συνιστωσών (strongly connected components) που περιέχει ο χώρος καταστάσεων, καθώς και τον αριθμό των τόξων που ξεκινούν από μια ισχυρά συνδεδεμένη συνιστώσα και καταλήγουν σε μία άλλη.

Σύμφωνα με τη θεωρία γράφων μία συνιστώσα είναι οποιοσδήποτε υπογράφος συνδεδεμένων κορυφών. Μία ισχυρά συνδεδεμένη συνιστώσα είναι ένας μέγιστος υπογράφος, στον οποίο είναι δυνατό να βρεθεί ένα μονοπάτι από ένα κόμβο σε έναν οποιονδήποτε άλλο κόμβο. Τα ισχυρά συνδεδεμένα τμήματα είναι πολύ χρήσιμα για τον προσδιορισμό συγκεκριμένων ιδιοτήτων συμπεριφοράς και μπορούν να υπολογιστούν από αλγόριθμους οι οποίοι είναι γραμμικοί στον χώρο και στον χρόνο. Τέτοιες ιδιότητες μπορεί να είναι η δήλωση τερματισμού, η καταγραφή του συντομότερου ή του μεγαλύτερου μονοπατιού ώστε να τερματιστεί το σύστημα ή ακόμα και ο προσδιορισμός διαδρομών προς τον τερματισμό που αποτελούνται από τον αριθμό κορυφών. Επίσης μπορεί να επιλεγθούν οι διαδρομές οι οποίες διέρχονται ή αποφεύγουν συγκεκριμένους κόμβους. Έτσι αν όλοι οι κόμβοι οδηγούσαν σε αυστηρά συνδεδεμένες συνιστώσες, θα ήμασταν σίγουροι ότι όλες οι ακολουθίες μαρκαρισμάτων θα κατέληγαν σε μετάβαση τερματικής κατάστασης και το πλήθος των μελών αυτών των ακολουθιών θα ήταν αν όχι μετρήσιμο, πεπερασμένο.

Στο SCC graph του στατιστικού μέρους του πίνακα δείχνει ότι υπάρχουν 182 αυστηρά συνδεδεμένα τμήματα και 673 τόξα τα οποία συνδέουν ισχυρά συνδεδεμένα τμήματα. Άρα οι ισχυρά συνδεδεμένες συνιστώσες είναι λιγότερες από τους κόμβους που είναι μη αυστηρά συνδεδεμένοι. Αυτό σημαίνει ότι το σύστημα έχει τουλάχιστον ένα ισχυρά συνδεδεμένο τμήμα που περιέχει περισσότερους από έναν κόμβους και άρα υπάρχουν στο σύστημα μη-πεπερασμένες ακολουθίες εκτέλεσης. Επομένως δεν μπορούμε να είμαστε σίγουροι ότι το πρωτόκολλο τερματίζει. Για να πετύχουμε τερματισμό του πρωτοκόλλου, συνήθως περιορίζουμε τον αριθμό των επαναμεταδόσεων [81].

CPN Tools state space report for:

C:\Samples\SimpleProtocol\SimpleProtocol_OG.cpn

Report generated:

Statistics

State Space

Nodes: 428

Arcs: 1130

Secs: 1

Status: Full

Sec Graph

Nodes: 182

Arcs: 673

Secs: 0

Πίνακας 3_1. Στατιστικά δεδομένα του state space report των διαγραμμάτων OG και SCC αντίστοιχα.

Το δεύτερο κομμάτι του SCC γραφήματος, καταγράφει το ανώτερο και κατώτερο όριο της τιμής κάθε μεταβλητής, κατηγοριοποιώντας αυτές ανάλογα με το είδος, όπου στο παράδειγμά μας είναι ακέραιοι (**integer bounds**) και πολλαπλά σύνολα (**multi-set bounds**). Στο πάνω μέρος του πίνακα 3_2 φαίνονται τα άνω και τα κάτω όρια ακεραίων, δηλαδή ο μέγιστος και ο ελάχιστος αριθμός των μαρκών που μπορεί να έχει κάθε θέση. Έτσι, φαίνεται ότι οι τις θέσεις A, B, C, D και Limit έχουν πάντα από μηδέν έως δύο μάρκες, ενώ οι θέσεις NextSend, NextRec και Received έχουν πάντα μία μάρκα η καθεμιά. Τέλος, η θέση Send έχει πάντα τέσσερις μάρκες [81]. Τα αποτελέσματα του πίνακα 3_2 επιβεβαιώνουν ότι το σύστημα λειτουργεί όπως είχε προβλεφθεί. Στα κατώτερα μέρη του πίνακα 3_2 φαίνονται τα όρια πολλαπλών συνόλων. Εξ ορισμού, το άνω όριο των πολλαπλών συνόλων μιας θέσης, είναι το μικρότερο πολλαπλό σύνολο που είναι μεγαλύτερο από όλα τα προσιτά μαρκαρίσματα αυτής της

θέσης. Αντίστοιχα, το κάτω όριο των πολλαπλών συνόλων μιας θέσης, είναι το μεγαλύτερο πολλαπλό σύνολο που είναι μικρότερο από όλα τα προσιτά μαρκαρίσματα αυτής της θέσης. Τα όρια ακεραίων μας δίνουν πληροφορίες για τον αριθμό των μαρκών, ενώ τα όρια πολλαπλών συνόλων μας δίνουν πληροφορίες για τις τιμές που μπορούν να έχουν οι μάρκες. Από τα όρια πολλαπλών συνόλων, φαίνεται ότι οι θέσεις A και B μπορούν να περιέχουν και τα τέσσερα διαφορετικά πακέτα, ενώ οι θέσεις C και D μπορούν να περιέχουν και τις τέσσερις πιθανές βεβαιώσεις λήψης πακέτων. Οι καταστάσεις C και D ενημερώνουν για την λήψη πακέτου και στη συνέχεια ο αριθμός του επόμενου πακέτου που θα αποσταλεί. Για αυτό και δεν υπάρχει για καμία μετάβαση βεβαίωση λήψης για το πακέτο 1. Για αυτές τις τέσσερις θέσεις δεν είναι δυνατές άλλες τιμές μαρκών.

Παρατηρούμε, ότι το άνω όριο πολλαπλών συνόλων για την θέση A είναι 8 μαρκαρίσματα σε αντίθεση με το άνω όριο των ακεραίων στην αντίστοιχη θέση που είναι 2. Αυτό συμβαίνει διότι το κομμάτι του πίνακα που αναγράφεται ο μέγιστος όρος πολλαπλών συνόλων αναφέρονται στο μέγιστο αριθμό επιλογών που έχουμε. Για παράδειγμα, η θέση A μπορεί να μαρκαριστεί με $1^1(1, \text{"Modellin"})$ ή $1^1(1, \text{"Modellin"}) + 1^1(3, \text{"alysis"})$ ή $2^1(3, \text{"alysis"})$.

Επομένως το ελάχιστο πολλαπλό σύνολο που είναι μεγαλύτερο από όλα τα πιθανά μαρκαρίσματα στη θέση A θα είναι το

$$2^1(1, \text{"Modellin"}) + 2^1(2, \text{"g and An"}) + 2^1(3, \text{"alysis"}) + 2^1(4, \text{"###"}).$$

Με παρόμοιο τρόπο προκύπτουν και τα όρια πολλαπλών συνόλων των μεταβάσεων B, C, D, NextRec, NextSend και Received.

Οι κόμβοι NextSend και NextRec μπορούν να πάρουν όλες τις μάρκες από το 1 μέχρι το 5 το οποίο είναι και φυσιολογικό μιας και οι θέσεις αυτές αποτελούν δηλώσεις και βεβαιώσεις λήψης πακέτων. Βέβαια σε κάθε στιγμιότυπο μαρκάρονται ακριβώς από μία μάρκα, κάτι που φαίνεται από το Best integer bounds όπου ο μέγιστος και ελάχιστος αριθμός μαρκαρισμάτων είναι 1. Τέλος η θέση Send διαθέτει σταθερό σύνολο μαρκών ίσο με 4. Αφού στο πίνακα πολλαπλού συνόλου καταγράφονται οι

$$1^1(1, \text{"Modellin"}) + 1^1(2, \text{"g and An"}) + 1^1(3, \text{"alysis###"}) + 1^1(4, \text{"#####"})$$

σημαίνει ότι πάντα θα τα μαρκαρίσματα θα είναι τα ίδια.

Boundedness Properties

Best Integer Bounds

	<u>Upper</u>	<u>Lower</u>
Top'A 1	2	0
Top'B 1	2	0
Top'C 1	2	0
Top'D 1	2	0
Top'Limit 1	2	0
Top'NextRec 1	1	1
Top'NextSend 1	1	1
Top'Received 1	1	1
Top'Send 1	4	4

Best Upper Multi-set Bounds

Top'A 1 2^{(1,"Modellin")++}
2^{(2,"g and An")++}
2^{(3,"alysis##")++}
2^(4,"#####")

Top'B 1 2^{(1,"Modellin")++}
2^{(2,"g and An")++}
2^{(3,"alysis##")++}
2^(4,"#####")

Top'C 1 2²⁺⁺
2³⁺⁺
2⁴⁺⁺
2⁵

Top'D 1 2²⁺⁺
2³⁺⁺
2⁴⁺⁺
2⁵

Top'Limit 1 2^e

Top'NextRec 1 1¹⁺⁺
1²⁺⁺
1³⁺⁺

```

1`4++
1`5
  Top'NextSend 1    1`1++
1`2++
1`3++
1`4++
1`5
  Top'Received 1    1`""++
1`"Modellin"++
1`"Modelling and An"++
1`"Modelling and Analysis###"
  Top'Send 1      1`(1,"Modellin")+
1`(2,"g and An")+
1`(3,"alysis###")+
1`(4,"#####")

```

Best Lower Multi-set Bounds

```

Top'A 1    empty
Top'B 1    empty
Top'C 1    empty
Top'D 1    empty
Top'Limit 1  empty
Top'NextRec 1  empty
Top'NextSend 1  empty
Top'Received 1  empty
Top'Send 1    1`(1,"Modellin")+
1`(2,"g and An")+
1`(3,"alysis###")+
1`(4,"#####")

```

Πίνακας 3_2. Πίνακας ορίων ακεραίων και πολλαπλών συνόλων του *state space report*.

Το τρίτο μέρος της αναφοράς του χώρου καταστάσεων (πίνακας 3_3) δίνει πληροφορίες για τις ιδιότητες της **οικείας κατάστασης** (Home Properties) και της **διάρκειας** (Liveness Properties).

Το οικείο μαρκάρισμα αφορά τα μαρκάρια που είναι προσιτά από όλα τα άλλα, δηλαδή μπορούμε να φτάσουμε σε αυτό το μαρκάρισμα ανεξάρτητα με τη διαδρομή που ακολουθούμε και τις προηγούμενες καταστάσεις μαρκαρισμάτων. Με άλλα λόγια σε συγκεκριμένης μορφής μαρκαρίσματος μπορούμε να καταλήξουμε από όλες τις ακολουθίες πιθανών πυροδοτήσεων. Στο παράδειγμά μας μόνο ένα μαρκάρισμα είναι οικείο και βρίσκεται στο 235^ο βήμα στο διάγραμμα προσιτότητας (σχήμα 29).

Ένα νεκρό μαρκάρισμα (Dead Markings) είναι ένα μαρκάρισμα που δεν έχει καθόλου ενεργοποιημένες μεταβάσεις. Παρατηρούμε ότι το μοναδικό νεκρό μαρκάρισμα είναι το 235^ο και ότι ταυτίζεται με το οικείο [15].

Home Properties

Home Markings

[235]

Liveness Properties

Dead Markings

[235]

Dead Transition Instances

None

Live Transition Instances

None

Πίνακας 3_3. Οικεία, νεκρά μαρκάρια και νεκρές και διαρκείς μεταβάσεις του συστήματος.

Το μαρκάρισμα στη θέση 235 όπως φαίνεται και στο σχήμα 29 έχει καταχωρηθεί τη στιγμή που έχουν παραδοθεί επιτυχώς και τα τέσσερα τα πακέτα και έχουν δοθεί οι βεβαιώσεις λήψης και παράδοσης. Το ότι το μαρκάρισμα είναι νεκρό σημαίνει ότι το πρωτόκολλο είναι μερικώς σωστό, δηλαδή αν τερματίσει θα έχουμε σωστό αποτέλεσμα. Από την άλλη μεριά, το γεγονός ότι το μαρκάρισμα 235 είναι οικείο, υποδεικνύει ότι το πρωτόκολλο έχει την ιδιότητα, να μην φτάσει ποτέ σε μια κατάσταση από την οποία είναι αδύνατο να τερματίσει με το σωστό αποτέλεσμα.

```

235:
Top'Send 1: 1` (1,"Modellin")++
1` (2,"g and An")++
1` (3,"alysis##")++
1` (4,"#####")
Top'NextSend 1: 1` 5
Top'D 1: empty
Top'A 1: empty
Top'B 1: empty
Top'C 1: empty
Top'NextRec 1: 1` 5
Top'Received 1: 1` "Modelling and Analysis##"
Top'Limit 1: 2` e

```

Σχήμα 29. Το μοναδικό νεκρό και ταυτόχρονα οικείο μαρκάρισμα του συστήματος μέσα στο διάγραμμα προσιτότητας μαζί με το αντίστοιχο toggle descriptor.

Από τις ιδιότητες του πίνακα 3_3 φαίνεται ότι δεν υπάρχουν νεκρές μεταβάσεις (Dead Transition). Αυτό σημαίνει ότι κάθε μετάβαση είναι ενεργοποιημένη σε τουλάχιστον ένα προσιτό μαρκάρισμα. Φαίνεται επίσης ότι δεν υπάρχουν **διαρκείς μεταβάσεις (live transitions)**. Μια διαρκής μετάβαση είναι μια μετάβαση η οποία πάντα, ότι κι αν γίνει, μπορεί να ενεργοποιηθεί άλλη μια φορά. Όταν υπάρχουν νεκρά μαρκάρια, όπως στο πρωτόκολλο επικοινωνίας, δεν μπορούν να υπάρχουν διαρκείς μεταβάσεις. Με μια πρώτη ματιά θα μπορούσε κανείς να σκεφτεί ότι η ύπαρξη νεκρών μαρκαρισμάτων θα απέτρεπε την ύπαρξη οικείων μαρκαρισμάτων. Όμως, εξ ορισμού, ένα μαρκάρισμα είναι πάντα προσιτό από τον εαυτό του, κι έτσι ένα νεκρό μαρκάρισμα μπορεί να είναι ένα οικείο μαρκάρισμα, αν είναι το μοναδικό νεκρό μαρκάρισμα. Αυτή η περίπτωση ισχύει και για το πρωτόκολλο επικοινωνίας.

Στο τέταρτο τμήμα της αναφοράς του χώρου καταστάσεων περιγράφεται ο χαρακτήρας δικαιοσύνης ή αμεροληψίας. Οι βρόγχοι Send_Packet, Transmit_Packet κρίνονται αμερόληπτη, που σημαίνει ότι πάντα, σε οποιαδήποτε μη-πεπερασμένη ακολουθία μαρκαρισμάτων θα

μπορούν να ενεργοποιηθούν. Με άλλα λόγια, αν κάποιες από αυτές τις μεταβάσεις πάνε να εκτελείται, το σύστημα θα πρέπει οπωσδήποτε μετά από κάποια βήματα να τερματίσει. Υπάρχουν άλλοι δύο τύποι αμεροληψίας, η δίκαιη και ακριβοδίκαιη αλλά επειδή η διαφορά μεταξύ τους είναι ασθενής, χαρακτηρίζονται και στις δύο περιπτώσεις αμερόληπτη. Τα υπόλοιπα τρία transition δηλώνονται μεροληπτικά, ότι δηλαδή μπορεί σε κάποια φάση να μην ενεργοποιηθούν ξανά, παρότι το σύστημα συνεχίζει να τρέχει [15].

Fairness Properties

Top'Receive_Acknow 1 No Fairness

Top'Receive_Packet 1 No Fairness

Top'Send_Packet 1 Impartial

Top'Transmit_Acknow 1 No Fairness

Top'Transmit_Packet 1 Impartial

Πίνακας 3_4. Πίνακας αμεροληψίας των μεταβάσεων.

Το πλεονέκτημα των Occurrence graph έναντι της αναφοράς του χώρου κατάστασης, είναι ότι μπορούμε σχηματικά να αντιληφθούμε την πρόοδο της προσομοίωσης και να προσκομίσουμε χρήσιμες πληροφορίες για τις μεταβλητές και σε ποια χρονική στιγμή από την έναρξη της διαδικασίας είχαν τις συγκεκριμένες τιμές. Είναι αντιληπτό ότι ένα τέτοιο διάγραμμα χρησιμεύει σε πολύ απλά μοντέλα με λίγες επιλογές μιας και η πολυπλοκότητα του αυξάνεται ραγδαία και χρειάζεται χρόνος για την κατασκευή του. Με τον χώρο καταστάσεων παίρνουμε πληροφορίες για τις τιμές των μεταβλητών που αλλιώς θα μπορούσαμε να πάρουμε μόνο με την χειροκίνητη προσομοίωση σε χρόνο το λιγότερο που τρέχει το σύστημα. Τα άνω και κάτω όρια δεν μπορούν να γίνουν αντιληπτά ούτε με την σταδιακή παρακολούθηση του μοντέλου, μιας και δεν είναι δυνατόν ο απλός παρατηρητής να αφομοιώσει τα πιθανά μαρκαρίσματα. Επίσης η αναφορά του χώρου δίνει μία ιδέα σε πρώτο επίπεδο αν το σύστημα λειτουργεί σωστά και με τις προϋποθέσεις που θέλαμε.

Ένα από τα βασικά μειονεκτήματα του χώρου καταστάσεων είναι το λεγόμενο πρόβλημα της έκρηξης. Τότε γίνεται τόσο μεγάλος με αποτέλεσμα να μη μπορεί να κατασκευαστεί

ολόκληρος και να λείπουν κάποια δεδομένα, ή τα αποτελέσματα να αφορούν μεμονωμένα βήματα. Ένα δεύτερο μειονέκτημα, είναι ότι η αναφορά του χώρου κατασκευάζετε για ένα συγκεκριμένο αρχικό μαρκάρισμα, το οποίο συχνά αντιστοιχεί μόνο σε μία από τις πολλές πιθανές αρχικές καταστάσεις του μοντέλου [15].

Προκειμένου να έχουμε μια καλύτερη άποψη για το σύστημα ο χρήστης μπορεί να υποβάλλει τα δικά του ερωτήματα (Queries)τα οποία διατυπώνονται στο ίδιο μοντέλο σε άλλο φύλλο και πάντα με τη μορφή συναρτήσεων. Ένα από τα συχνότερα ερωτήματα είναι να βρεθεί το μήκος της συντομότερη διαδρομής μέχρι μια συγκεκριμένη μετάβαση ή ερωτήματα που απαντώνται με true ή false. Ακολουθούν παραδείγματα ερωτημάτων που διατυπώθηκαν στο απλό πρωτόκολλο.

Length of Shortest Path

```
length(ArcsInPath(1,235));
```

Is Next Send Ever Decreased?

```
Text
```

```
let
```

```
  fun Mark_NextSend n =
```

```
    ms_to_col(Mark.Top'NextSend 1 n)
```

```
in
```

```
  PredAllArcs
```

```
    (fn a => Mark_NextSend(DestNode a) <Mark_NextSend(SourceNode a)) end
```

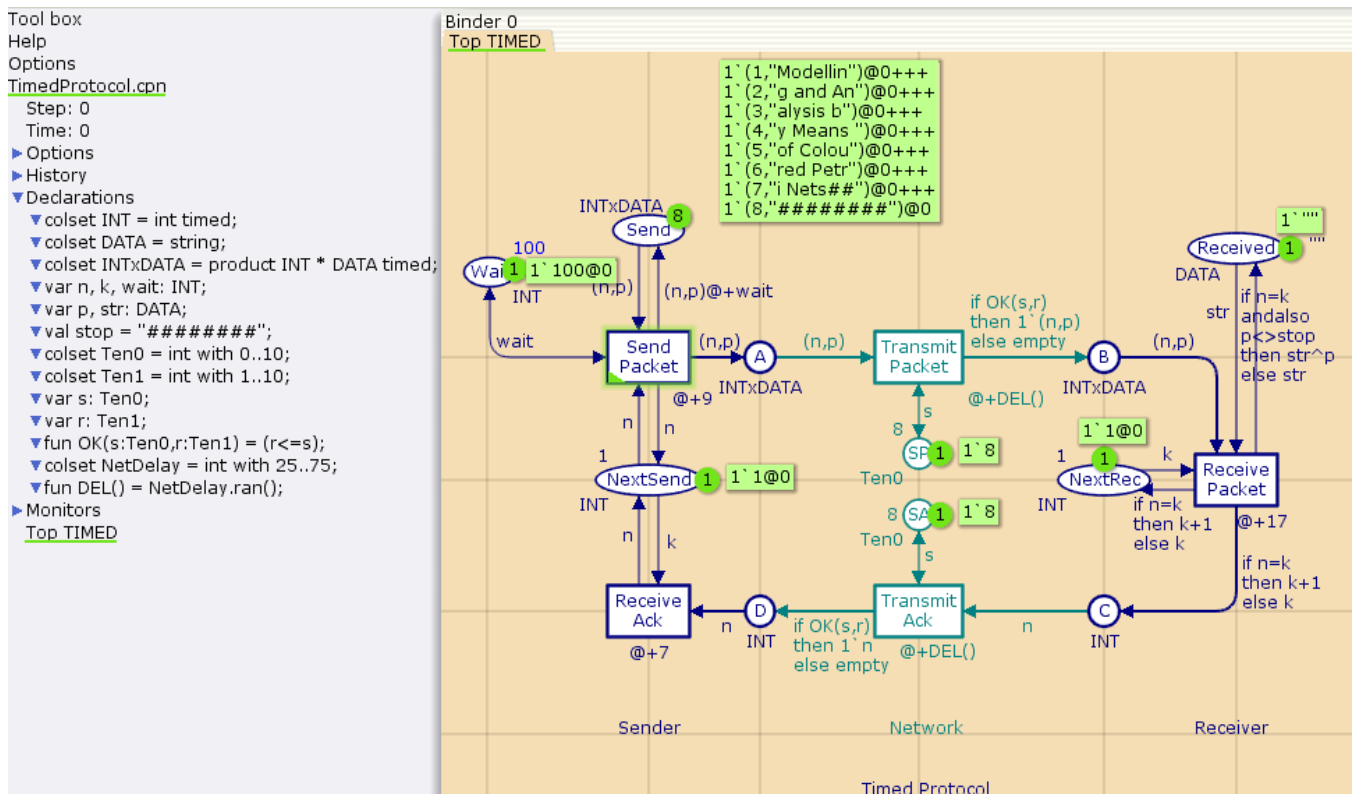
Πίνακας 4. Ερωτήματα του απλού πρωτοκόλλου.

2.2.3 Timed CPN

Σε αυτή την ενότητα εξετάζουμε τον τρόπο με τον οποίο μπορούμε να χειριστούμε ένα χρονικό σήμα ή μία συνάρτηση που περιορίζει τη διακίνηση ως προς τον χρόνο. Στα συστήματα πραγματικού χρόνου (real time system) τα αποτελέσματα μπορούν να χρησιμοποιηθούν για έρευνα και διαχείριση και για αυτό το λόγο είναι και πιο εύχρηστα.

Για την δημιουργία ενός τέτοιου δικτύου επεκτείνουμε το πρωτόκολλο σε **δίκτυο χρόνου (Timed Coloured Petri nets – CPNs)**. Εισάγουμε ένα καθολικό ρολόι του οποίου οι τιμές παριστάνουν χρόνο μοντελοποίησης και είναι είτε ακέραιοι αριθμοί, διακριτές τιμές δηλαδή, είτε πραγματικοί (συνεχείς τιμές). Κάθε μάρκα εκτός από την τιμή της διαθέτει και μια τιμή χρόνου η οποία ονομάζεται χρονική σφραγίδα (**time stamp**). Η χρονική σφραγίδα συμβολίζεται με «@+» και υποδεικνύει τον μικρότερο επιτρεπτό χρόνο στον οποίο απομακρύνεται από το στοιχείο καταχώρησης.

Όταν σε έναν κόμβο t καταχωρηθεί μάρκα χρόνου a αυτό σημαίνει πως από τη στιγμή που θα ενεργοποιηθεί μέχρι να πυροδοτήσει θα μεσολαβήσουν a χρόνοι (**times**). Δηλαδή θα βρίσκεται σε αναμονή για a χρόνους. Αν ένας άλλος κόμβος είναι επίσης ενεργοποιημένος και μαρκαρισμένος με χρόνο $a-k$ τότε θα πυροδοτήσει k χρόνους νωρίτερα. Με αυτό τον τρόπο εφαρμόζεται στα πακέτα μία σειρά προτεραιότητας. Έτσι ο παρατηρητής μπορεί να επέμβει στην ακολουθία μαρκαρισμάτων και να ορίσει αυτός τη διαδοχή των μαρκών, σε αντίθεση με τα απλά CPN όπου η επιλογή της διαδρομής γινόταν τυχαία.



Σχήμα 30. Η επέκταση του απλού πρωτοκόλλου για τη δημιουργία Time CPN.

Με της εισαγωγή της χρονικής παραμέτρου η εκτέλεση ενός γεγονότος παραπέμπει στην εκτέλεση ενός μέλους σε ουρά γεγονότων. Τη θεωρία ουρών τη συναντάμε σε αρκετές γλώσσες προσομοίωσης συστημάτων διακριτών γεγονότων. Κάθε μαρκάρισμα υπάρχει για ένα πεπερασμένο διάστημα του χρόνου μοντελοποίησης, ενώ η εκτέλεση ενός στοιχείου καταχώρησης είναι στιγμιαία. Όταν σε διαδοχικές μεταβάσεις καταχωρείται διακριτό σήμα χρόνου αυξημένο κατά ένα σε κάθε μέλος, η εκτέλεση μοιάζει σαν την εξυπηρέτηση ταμείου-πελατών μιας τράπεζας με νούμερα. Αν υπήρχαν περισσότερες από μία μεταβάσεις ενεργοποιημένες σε κάθε βήμα μπορεί η ταχύτητα μετάδοσης να βελτιωνόταν (περισσότερα ταμεία) ενώ η πολυπλοκότητα να αυξανόταν.

Στο παράδειγμα η `SendPacket` έχει μάρκα `@+9`. Αυτό σημαίνει πως οι μάρκες που δημιουργούνται στις θέσεις `A` και `Next Sent`, οι συνδεδεμένες με την `SendPacket`, παίρνουν χρονικές σφραγίδες κατά 9 χρόνους μεγαλύτερες, δηλαδή από τη στιγμή που ενεργοποιούνται οι μεταβάσεις η αντίστοιχη μάρκα θα μετακινηθεί μετά από 9 χρόνους. Όταν πυροδοτήσει η `SendPacket` στη και ενεργοποιηθεί η `send`, η μάρκα θα είναι αυξημένη κατά `@+9+100` όπου 100 η τιμή της `wait`. Το 9 συμβολίζει το χρόνο που απαιτείται μέχρι την αναμετάδοση του πακέτου ενώ το 100 τον χρόνο που χρειάζεται μέχρι να ολοκληρωθεί μία μετάδοση και πριν γίνει η

επιλογή για το επόμενο πακέτο. Για να περάσει ένα πακέτο από την TransmitPacket στο B θα πρέπει να μεσολαβήσουν «Delay()» χρόνοι δηλαδή από 25 ως 75. Αυτό σημαίνει ότι η διάρκεια για να βεβαιωθεί το σύστημα ότι το πακέτο που έφτασε είναι στη σωστή σειρά σε σχέση με τα άλλα, θα μεσολαβήσουν από 25 ως 75 χρόνοι. Στη συνέχεια ο ReceivePacket για να στείλει το πακέτο στον δέκτη, χρειάζεται 17 χρόνους.

Αλλάζοντας τις τιμές της Wait μπορούμε να εξετάσουμε κατά κάποιον τρόπο την απόδοση του συστήματος. Μπορεί να επιβαρυνθεί η ταχύτητα αναμετάδοσης, ώστε να σημειωθεί επανάληψη πακέτων, να προσπεράσει ένα πακέτο κάποιο άλλο, μέχρι και να μεγαλώσει η πιθανότητα να αναβληθεί μία βεβαίωση λήψης. Επίσης είναι δυνατόν να τερματίσει το σύστημα προτού παραδοθούν όλα τα πακέτα. Για αυτό το λόγο πολλές φορές αλλάζουμε τις τιμές των μαρκών (ως προς την ποσότητα ή τον χρόνο) με σκοπό τη βελτιστοποίηση λειτουργίας του δικτύου.

Το εργαλείο δίνει την δυνατότητα να καλέσουμε συναρτήσεις από τη βιβλιοθήκη ή ακόμα να συντάξουμε πιο πολύπλοκες στατιστικές συναρτήσεις ώστε να δώσουμε στις καθυστερήσεις σήμα που ακολουθεί κάποια κατανομή.

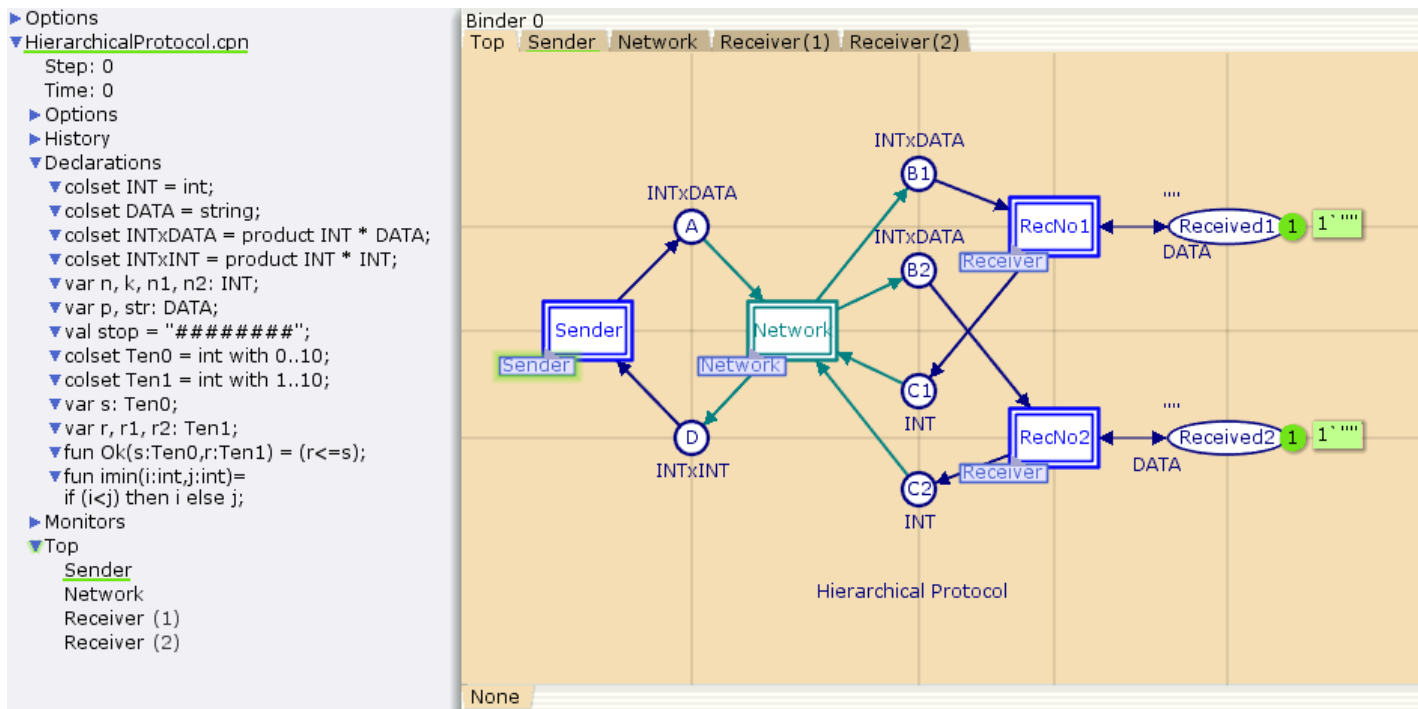
2.2.4 Ιεραρχικά Χρωματισμένα δίκτυα Petri

Σε πολλές γλώσσες προγραμματισμού προκειμένου ο χειριστής να μειώσει την έκταση των εντολών συντάσσει κάποια κομμάτια προγραμμάτων, οι λεγόμενες υπορουτίνες, οι οποίες στη συνέχεια καλούνται να εισάγουν τα δεδομένα τους στο βασικό πρόγραμμα. Μπορεί επίσης να ανοίξει και να τρέξει μέσα σε ένα πρόγραμμα και άλλα αρχεία του ίδιου τύπου.

Η ανάγκη απλούστευσης και μιας πιο απλής περιγραφής στη μοντελοποίηση συστημάτων οδήγησε στη δημιουργία κατάλληλων εργαλείων τα οποία θα μιμούταν τη χρήση των υπορουτίνων. Για αυτό το σκοπό τα CPN σχεδίασαν τα **ιεραρχικά χρωματισμένα δίκτυα Petri (Hierarchical CPN-HCPN)**.

Στα διαγράμματα ροής δεδομένων (**data flow charts**) που υπάρχουν σε αρκετές περιγραφικές γλώσσες, καθιερώνεται μία παραπλήσια δομή ιεραρχίας με αυτή των HCPN. Στις πιο σύγχρονες γλώσσες προγραμματισμού, μια ανάλογη δράση συναντάμε στις αυτοτελείς μονάδες (**modules**).

Για τη δημιουργία ενός HCPN θα χρειαστεί να κατασκευάσουμε το σκελετό του μοντέλου το οποίο θα είναι ένα **βασικό δίκτυο (δίκτυο πηγής)**. Στο παράδειγμα του απλού πρωτοκόλλου το κεντρικό αυτό δίκτυο φαίνεται στο σχήμα 31 (**super page**). Στις μεταβάσεις που θέλουμε να μεσολαβήσει ένα κομμάτι του δικτύου (**καταστάσεις αντικατάστασης**), από την παλέτα “hier” επιλέγουμε την επιλογή «move the transition to the subpage». Με αυτή τη κίνηση, εισάγουμε μία ετικέτα «tag» με το ίδιο όνομα της μετάβασης (μετάβαση αντικατάστασης) η οποία μας οδηγεί σε μία καινούρια σελίδα (**subpage**). Η καινούρια σελίδα έχει το όνομα της ετικέτας και είναι καταχωρημένες σε αυτή μόνο η μετάβαση και οι καταστάσεις εισόδου και εξόδου αυτής. Η σελίδα που άνοιξε είναι έτοιμη για σχεδίαση. Στο χώρο αυτό θα πρέπει να σχεδιαστεί το υπομοντέλο. Η super page δίνει μία γενικότερη εποπτεία ως προς την λειτουργία του συστήματος ενώ τα υποσέλιδα (subpage) καταγράφουν μια πιο λεπτομερή περιγραφή του μοντέλου.



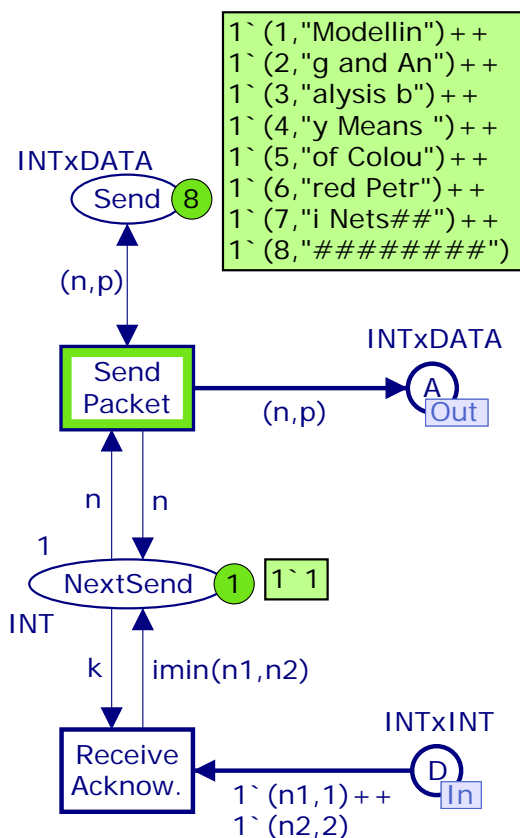
Σχήμα 31. Η κεντρική σελίδα του ιεραρχικού μοντέλου του πρωτοκόλλου συναπτόμενο με τις υποσελίδες.

Επιπλέον, θέλουμε να συνδέσουμε την απλή περιγραφή με την πιο λεπτομερή, με τέτοιο τρόπο ώστε να μπορούμε να μεταφράσουμε τη συνολική συμπεριφορά του συστήματος και να μπορούν να καταγραφούν τα δεδομένα με λογική σειρά.

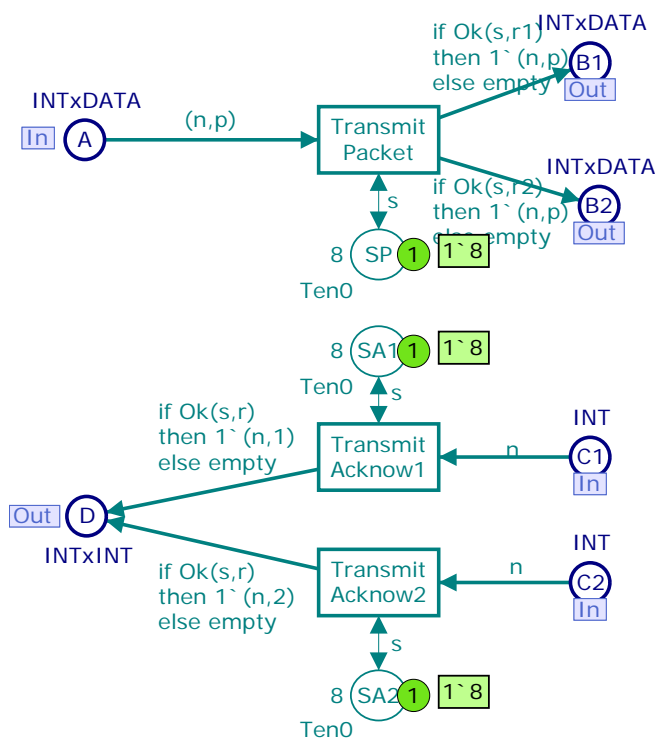
Για να γίνει όμως σωστά η σύνδεση των υποσελίδων με το δίκτυο πηγής θα πρέπει να γίνει διαχωρισμός ποιες θα είναι οι καταστάσεις εισόδου και ποιες εξόδου της κάθε υποσελίδας ξεχωριστά. Ανάλογα με τον αριθμό απογόνων και προγόνων της μετάβασης με την ετικέτα του υπομοντέλου, θα πρέπει ισάριθμες καταστάσεις στο υποσέλιδο να πάρουν τον χαρακτήρα εξόδου ή εισόδου.

Οι θέσεις αυτές ονομάζονται θέσεις πύλης (port place) και μέσω αυτής η υποσελίδα επικοινωνεί με το super page. Οι θέσεις εισόδου ονομάζονται πύλες εισόδου (input port) ενώ οι θέσεις εξόδου είναι οι πύλες εξόδου (output port). Μία θέση πύλης με ετικέτα I/O είναι και πύλη εισόδου και πύλη εξόδου. Μέσω των πυλών εισόδου το subpage λαμβάνει μάρκες από την αντίστοιχη μετάβαση του superpage, ενώ μέσω των πυλών εξόδου στέλνει μάρκες στο περιβάλλον της super page.

Στο ιεραρχικό του σχήματος 31 η super page είναι η Top. Οι μεταβάσεις sender, network RecNo1, RecNo2 είναι μαρκαρισμένες με το tag της ιεραρχίας, που οδηγεί στις ομώνυμες υποσελίδες. Η μετάβαση sender, συνδέεται με τα place A (απόγονος) και D (πρόγονος). Αυτό σημαίνει πώς η πύλη εισόδου της sender θα είναι η A και η πύλη εξόδου η D (Σχήμα 32).



Σχήμα 32. Το υποσέλιδο sender.

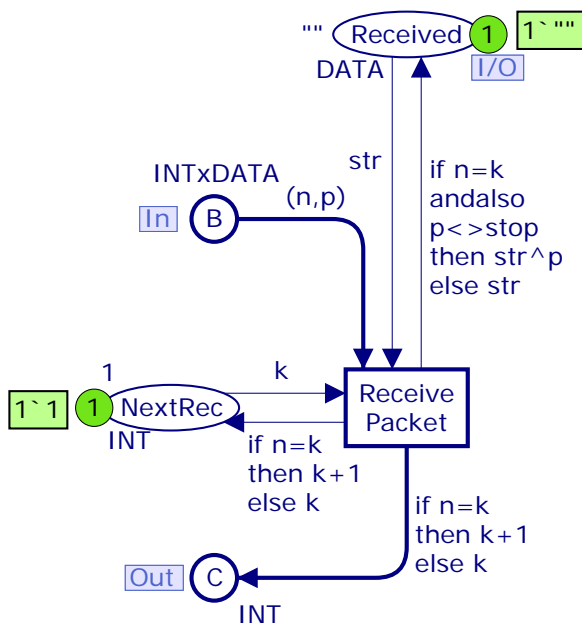


Σχήμα 33. Το υποσέλιδο Network.

Όταν θα πυροδοτήσει η send packet στο sub page sender μαρκάρεται η A η οποία συνδεδεμένη με την A του κεντρικού δικτύου ενεργοποιεί την network, εκτός και αν η send packet πυροδοτήσει ξανά. Η Receive Acknow. βρίσκεται σε κατάσταση αναμονής μιας και είναι πύλη εξόδου. Τώρα ενεργοποιείται η Transmit Packet του υποσέλιδου network (σχήμα 33). Οι καταστάσεις B1 και B2 μαρκαρίζονται είτε η μία είτε και οι δύο ταυτόχρονα. Στην περίπτωση του συγχρονισμού, επιλέγεται σε τυχαία σειρά ιεραρχίας ένα από τα receiver(1) και receiver(2) ώστε να επιτρέψουν να μαρκαριστεί η πύλη εισόδου. Επειδή τα δύο υποσέλιδα έχουν

την ίδια ακριβώς δομή το σχήμα 34 περιγράφει και τις δύο καταστάσεις. Στην περίπτωση που υπάρχουν δύο πύλες εισόδου επιλέγεται τυχαία ποια θα μαρκαριστεί πρώτη. Τότε το μοντέλο θα βρίσκεται σε κατάσταση αναμονής μέχρι να μαρκαριστούν όλες οι πύλες εισόδου. Στο σχήμα 34 κάτι τέτοιο σίγουρα θα συμβεί μιας και για να πυροδοτήσει η Receive Packet θα πρέπει να διαβαστεί και η μεταβλητή str. Επειδή οι Received1, Received2 του super page επιστέφουν τιμές στις προαναφερόμενες μεταβάσεις για αυτό το λόγο στα αντίστοιχα υποσέλιδα ενεργοποιούνται και οι πύλες εξόδου. Στη συνέχεια οι πυροδοτήσεις πραγματοποιούνται σε τυχαία σειρά, μια και πολλά πιθανά μαρκαρίσματα μπήκαν σε αναμονή. Σε μία τυχαία προσομοίωση του συστήματος, στο 551ο βήμα οι μεταβάσεις Received1 και Received2 καταχωρήθηκαν με την μάρκα 1 "Modelling and Analysis by Means of Coloured Petri Nets###".

Η δημιουργία ιεραρχικού μοντέλου χρησιμεύει στο να δοθούν περισσότερες λεπτομέρειες αλλά και περιορισμοί στη λειτουργία του. Η διαφορά με το απλό μοντέλο πρωτοκόλλου είναι ότι εδώ έχουμε δύο παραλήπτες. Για αυτό και διαφοροποιούνται οι βεβαιώσεις αποστολής και λήψης. Το Network του σχήματος 33 αποτελείται από δύο ανεξάρτητους γράφους. Ο ένας εκτελεί παραδόσεις πακέτων και ο άλλος βεβαιώσεις λήψης. Στην πύλη εισόδου D του σχήματος 32 μπορούν να καταφτάνουν πληροφορίες λήψης και για τους δύο παραλήπτες. Η επόμενη όμως παραγγελία θα περιέχει την μικρότερη αυτών.



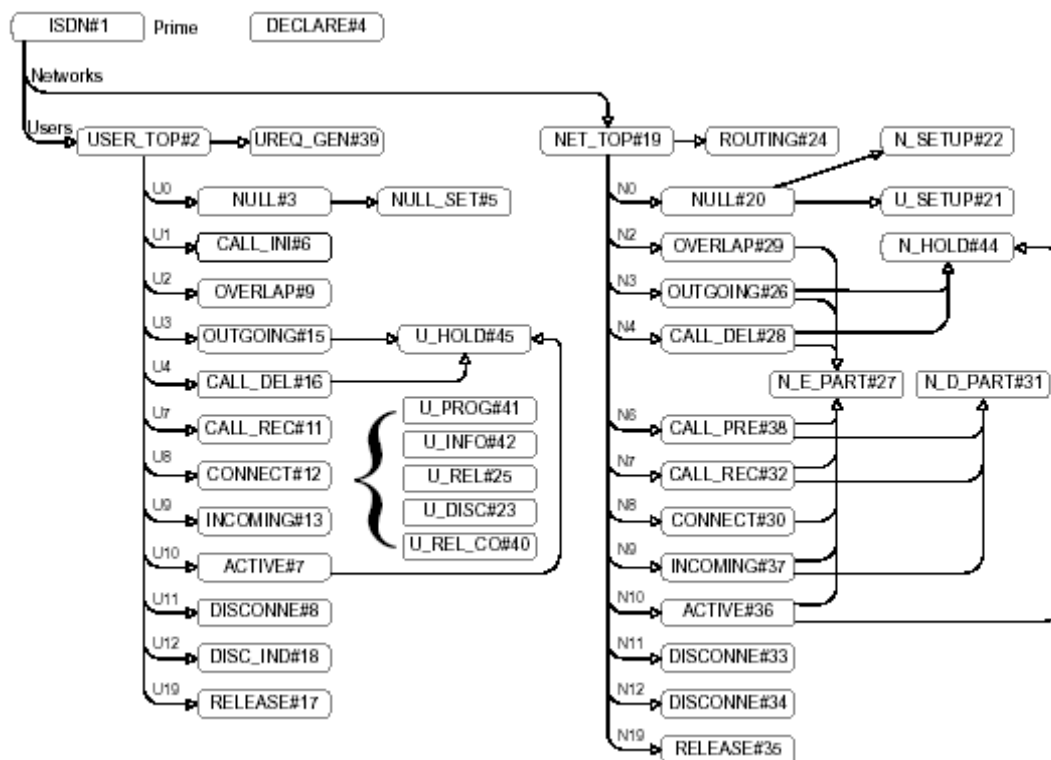
Σχήμα 34. Το υποσέλιδο Receiver(1) και Receiver(2).

Είχαμε αναφέρει πως οι σωστές αναθέσεις των πυλών θέτουν το σύστημα σε λειτουργία. Αλλιώς η προσομοίωση σταματάει στο transition με το πρώτο tag ιεραρχίας. Αυτό γίνεται με τη βοήθεια της ανάθεσης πύλης. (port assignment). Η πύλη εξόδου A στο σχήμα 32 συνδέεται με την υποδοχή εισόδου A του σχήματος 31, η πύλη εισόδου D στο σχήμα 32 συνδέεται με την υποδοχή εξόδου D του 31. Ομοίως συνεχίζουμε για όλες τις πύλες των υποσελίδων [15].

Κάθε ιεραρχικό δίκτυο μπορεί να μετατραπεί σε απλό CPN. Το αντίστροφο δεν ισχύει πάντα. Κάτι τέτοιο εξαρτάται από τον βαθμό πολλαπλότητας του δικτύου αλλά και από τον αριθμό των ακμών που διασταυρώνεται. Είναι αναμενόμενο, ότι κάθε δίκτυο χωρίς διασταυρωμένα βέλη έχει ισοδύναμο ιεραρχικό μοντέλο.

Στην περίπτωση του απλού πρωτοκόλλου, καμία υποσελίδα δεν διαθέτει μετάβαση αντικατάστασης, δηλαδή δεν οδηγούσε σε δευτερεύον ιεραρχικό μοντέλο. Τότε λέγεται ότι το βάθος ιεραρχίας είναι 1. Αν κάποια μετάβαση μιας υποσελίδας αποτελεί μετάβαση αντικατάστασης για ένα καινούριο ιεραρχικό δίκτυο τότε το βάθος ιεραρχίας θα ήταν 2. Σε πολλές περιπτώσεις συναντάμε ιεραρχικά μοντέλα με βάθος μεγαλύτερο του ένα. Έχει αποδειχθεί ότι σε αυτή τη περίπτωση η πολυπλοκότητα του γράφου αυξάνεται εκθετικά. Κάτι τέτοιο δίνει ιδιαίτερη σημασία στη χρήση ιεραρχικής δομής τόσο γιατί δίνει τη δυνατότητα γρήγορου και απλού σχεδιασμού τόσο γιατί μειώνει την πολυπλοκότητα των επιμέρους δικτύων σε σχέση με το απλό.

Για να δώσουμε μια συνολική εικόνα της σχέσης μεταξύ των διαφορετικών σελίδων σε ένα ιεραρχικό Χρωματισμένο δίκτυο Petri, χρησιμοποιούμε ένα γράφο ιεραρχίας σελίδων (page hierarchy graph) (σχήμα 35).



Σχήμα 35. Παράδειγμα γράφου ιεραρχίας σελίδων.

Ένας τέτοιος γράφος, συμβολίζει κάθε κόμβο με μία σελίδα. Το κάθε βέλος μεταξύ δύο σελίδων δηλώνει ότι η μία είναι υποσελίδα της άλλης, ανάλογα με την κατεύθυνσή του. Δηλαδή ότι η μετάβαση από την οποία ξεκινάει το βέλος είναι η μετάβαση αντικατάστασης της σελίδας της πηγής που οδηγεί στην αντίστοιχη υποσελίδα. Το κείμενο που είναι γραμμένο σε κάθε κόμβο προσδιορίζει το όνομα και τον αριθμό της σελίδας. Με τη βοήθεια του γράφου, το βάθος ιεραρχίας μπορεί να γίνει πιο αντιληπτό.

Στα ιεραρχικά μοντέλα, υπάρχουν συνήθως και οι **θέσεις συγχώνευσης (fusion place)**. Μπορεί να θεωρηθεί, ότι ένα σύνολο από θέσεις είναι ίδιες, δηλαδή αναπαριστούν μία ιδεατή θέση, ακόμα και αν σχεδιάζονται ως διαφορετικές. Όταν μία μάρκα προστίθεται ή αφαιρείται από μία τέτοια θέση, μία ίδια μάρκα δείχνει την ίδια συμπεριφορά και σε μία άλλη θέση του ίδιου συνόλου συγχώνευσης. Η σχέση που έχουν δύο θέσεις στο ίδιο σύνολο συγχώνευσης ίδια με τη σχέση θέσεων που ανατίθεται η μία στην άλλη από μία πύλη ανάθεσης. Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι θέσεις του ίδιου συνόλου σε διαφορετικές σελίδες. Τότε ένα στιγμιότυπο μπορεί να εμφανίζεται σε πολλά παράθυρα ανάλογα με το πόσες σελίδες διαθέτουν θέσεις του ίδιου συνόλου.

ΚΕΦΑΛΑΙΟ 3^ο

Εφοδιαστική αλυσίδα σχεδιασμένη από CPN

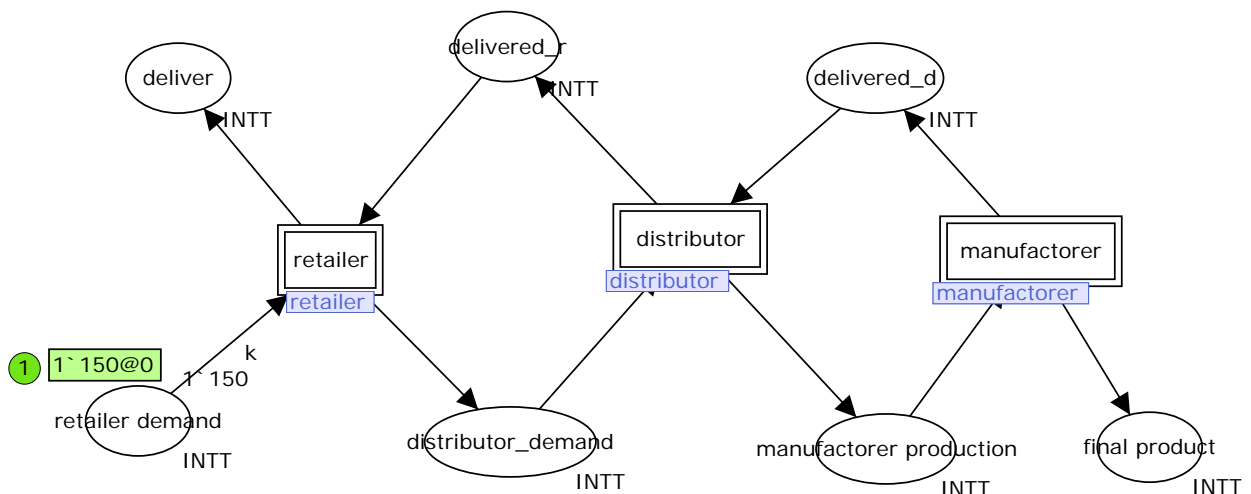
3.1 Εισαγωγή- Περιγραφή μοντέλου

Στο σχήμα 1 είχε δοθεί το πιο απλό και συνηθισμένο παράδειγμα εφοδιαστικής αλυσίδας που χρησιμοποιείται. Ουσιαστικά αποτελεί μια μακροσκοπική ματιά της παραγωγικής διαδικασίας στις διάφορες μονάδες παραγωγής. Το κάθε μέλος της αλυσίδας, προκειμένου να φτάσει στη φάση μετάδοση της πληροφορίας ή αποστολή εμπορευμάτων περνάει από διάφορα στάδια όπως κατασκευή, διαχείριση αποθεμάτων, επεξεργασία δεδομένων. Κάθε μία από αυτές τις διεργασίες προϋποθέτει κατάλληλο προγραμματισμό και μελέτη ώστε να αποφέρει το ζητούμενο αποτέλεσμα που είναι καλύτερη εξυπηρέτηση και μέγιστο κέρδος.

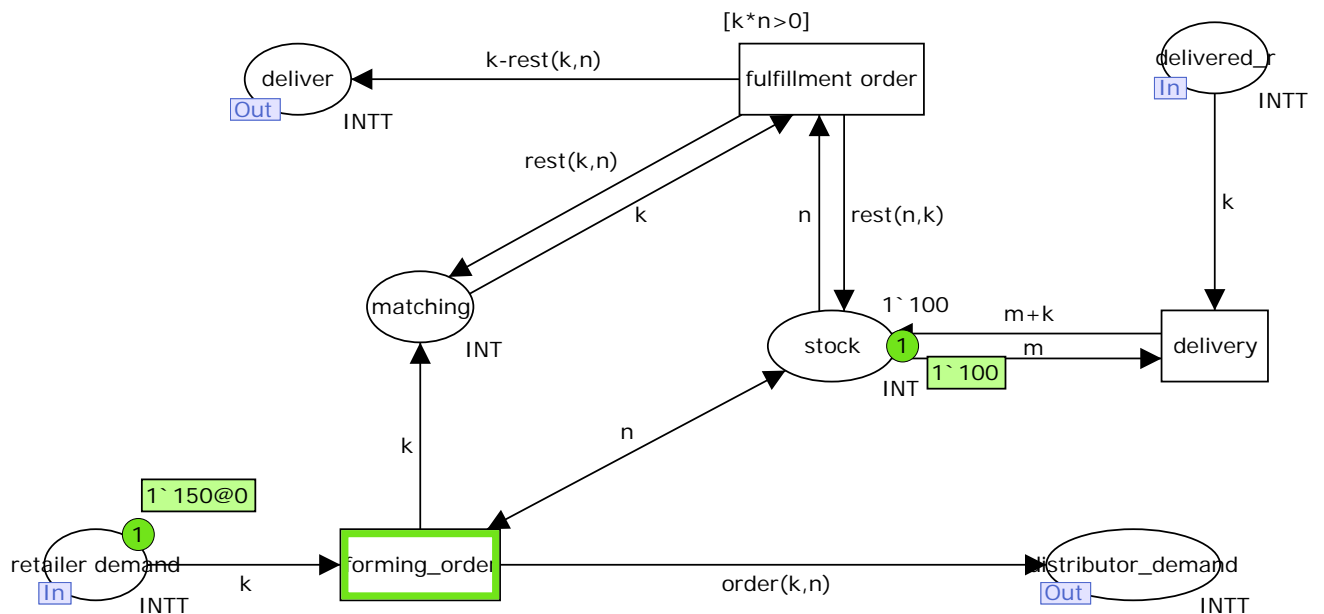
Η αλυσίδα του σχήματος 1 περιγράφεται στο σχήμα 36 (δίκτυο πηγής) με τη βοήθεια των HCPN. Οι μεταβάσεις *retailer*, *distributor* και *manufacturer* (μεταβάσεις αντικατάστασης) συμβολίζουν τα ενδιάμεσα μέλη της αλυσίδας λιανέμπορος, χονδρέμπορος και κατασκευαστής αντίστοιχα. Σε κάθε μία από αυτές τις μεταβάσεις αντιστοιχεί μία υποσελίδα στις οποίες περιγράφεται με περισσότερη λεπτομέρεια ο τρόπος λειτουργίας του συνολικού δικτύου στα επιμέρους κομμάτια. Οι συνέχεια δίνεται μια σύντομη ανάλυση του ρόλου κάθε κατάστασης του κεντρικού δικτύου:

- “*retailer demand*”: Οι παραγγελίες που δίνονται στον λιανέμπορο. Το αίτημα του πελάτη.
- “*distributor demand*” : Οι παραγγελίες που θα δοθούν στον χονδρέμπορο από το προηγούμενο μέλος της αλυσίδας, ανάλογα με την κατάσταση των αποθεμάτων του.
“*manufacturer production*”: Η τελική παραγγελία που θα δοθεί στον κατασκευαστή.
- “*final product*”: Η ποσότητα που θα χρειαστεί τελικά να κατασκευαστεί ύστερα από διαχείριση των αποθεμάτων του και των υλικών του.
- “*delivered_d*”: Η ποσότητα προϊόντος που θα δοθούν στον χονδρέμπορο ανάλογα με την παραγγελία που πήρε και τα αποθέματά του.

- “delivered_r”: Η ποσότητα που θα φτάσει στον λιανέμπορο.
- “deliver”: Η τελική ποσότητα που θα δοθεί στον πελάτη η οποία θα πρέπει να είναι ίδια με αυτή του retailer demand στο άθροισμά της.



Σχήμα 36. Το δίκτυο πηγής $hier0$ του ομώνυμου μοντέλου.



Σχήμα 37. Η υποσελίδα $retailer0$, $distributor0$ με μεταβάσεις σύνδεσης την $retailer$ και του $distributor$ αντίστοιχα.

```

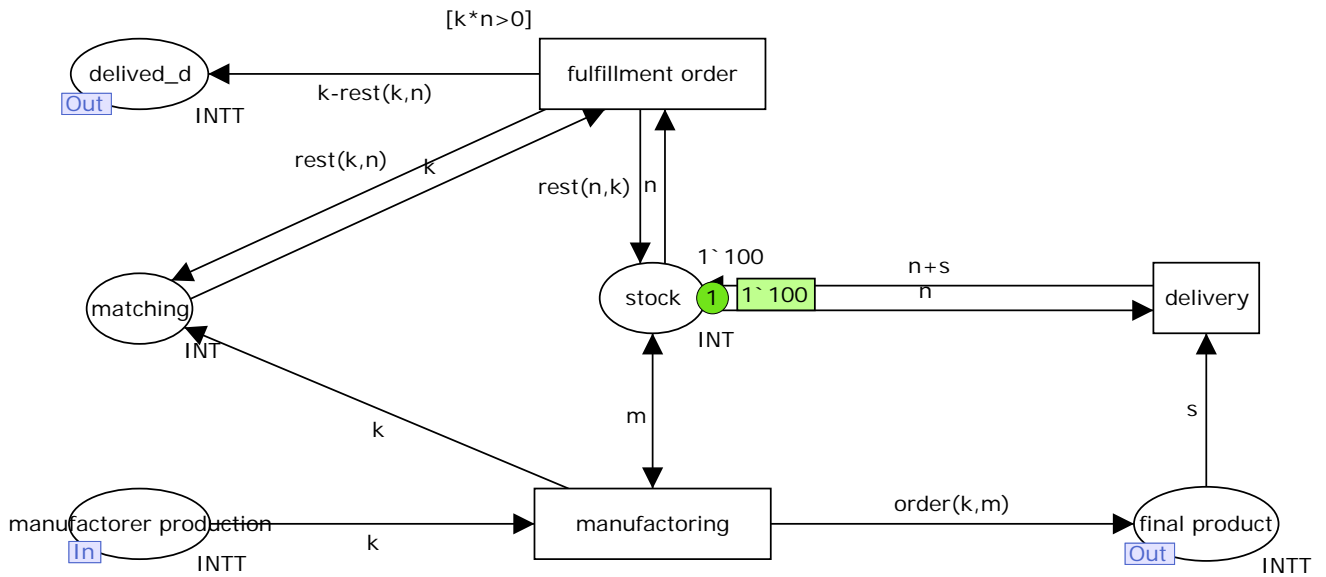
Declarations
▼ val se=33.0;
▼ val me=100.0;
▼ colset INT=int;
▼ colset INTT=INT timed;
▼ var k,m,n,s,a:INTT;
▼ normal(me,se): real;
▼ fun disp(me,se)=floor(normal(me,se));
▼ fun rest(i,j)=if (i>=j) then i-j else 0;
▼ fun order(i,j)=if i<=j then i else i+i-j;

```

Σχήμα 38. Οι εντολές των μοντέλων *hier0* και *hier*.

Το σενάριο του προβλήματος διακίνησης προϊόντων είναι το εξής: Ο πελάτης δίνει παραγγελία στον λιανέμπορο 150 μονάδες προϊόντος. Τα μέλη της αλυσίδας διαθέτουν από 100 μονάδες προϊόντος ο καθένας στις αποθήκες του. Σκοπός τους είναι να ικανοποιήσουν το αίτημα του αμέσως προηγούμενου μέλους της αλυσίδας το δυνατόν συντομότερο αλλά και να ξαναγεμίσουν τις αποθήκες τους με ελάχιστη ποσότητα την αρχική. Μία μεγάλη απόκλιση της τελικής τιμής των αποθεμάτων θα προκαλούσε το φαινόμενο του μαστιγίου και το σύστημά μας θα κρινόταν ως μη αποδοτικό.

Η μελέτη και διαχείριση αποθεμάτων του λιανέμπορου αναλύεται στην αντίστοιχη υποσελίδα ιεραρχίας του σχήματος 37: Η κατάσταση forming order συλλέγει τις παραγγελίες του retailer k και τη ποσότητα n της αποθήκης (stock). Στον σχήμα 38 δίνονται οι εντολές του δικτύου. Η συνάρτηση $rest(i,j)$ δέχεται δύο φυσικές μεταβλητές και εξάγει την διαφορά $i-j$ αν $i \geq j$ αλλιώς μηδέν. Η $order(i,j)$ δίνει i αν $i \leq j$ αλλιώς $i+(i-j)$. Η distributor demand θα μαρκαριστεί με $order(k,n)$. Αυτό σημαίνει ότι αν η παραγγελία είναι μεγαλύτερη από την ποσότητα στις αποθήκες τότε θα στείλει $k+k-n=150+50=200$ δηλαδή την παραγγελία μαζί με τη διαφορά της από τη ποσότητα του stock. Η λογική αυτής της ενέργειας από μεριάς του retailer στηρίζεται στην ανάγκη για ποιο γρήγορη εξυπηρέτηση του πελάτη. Έτσι μέσω της fulfillment order στέλνει στον πελάτη $k-(k-n)=n=100$ (αφού $k < n$) δηλαδή όλο το stock.



Σχήμα 39. Η υποσελίδα *manufacturer0* που αντιστοιχεί στην μετάβαση αντικατάστασης *manufacturer*.

Η κατάσταση *fulfillment order* επιστρέφει $rest(n,k)$ στο *stock* δηλαδή 0 που σημαίνει ότι οι αποθήκη άδειασε και $rest(k,n)=k-n=50$ στη *matching* δηλαδή μένουν άλλες 50 μονάδες για να ολοκληρωθεί η παραγγελία που πήρε από το προηγούμενο στρώμα της αλυσίδας που είναι ο πελάτης.

Τώρα στον *distributor* θα έχουμε παραγγελία $k=200, n=100$ και θα ζητήσει από τον *manufacturer* $order(k-n) = k+(k-n)=200+100=300$ μονάδες. Το απόθεμά του θα αδειάσει, η διαφορά του αιτήματος του προηγούμενου μέλους με τη ποσότητα στις αποθήκες του είναι $rest(k,n)=200-100=100$ μονάδες (μαρκάρισμα στη θέση *matching*).

Το αίτημα προς τον κατασκευαστή είναι να παραδώσει 300 μονάδες προϊόντος. Συνολικά καλείται να κατασκευάσει $order(k,n)=300+200=500$ μονάδες (*final product*) για να καλύψει και τις δικές του ανάγκες στους αποθηκευτικούς χώρους μιας και στη συνέχεια θα αδειάσουν. Το *stock* εφοδιάζεται με $n+s=600$ προϊόντα μέσω της *delivery* και στέλνει στο προηγούμενο μέλος ποσότητα $k-rest(k,n)=k-0=300$ (αφού τώρα $k=300, n=600$ άρα $k < n$). Στις αποθήκες του κατασκευαστή επιστρέφουν $600-300=300$ κομμάτια.

Η *delivered_d* στέλνει στις αποθήκες του χονδρέμπορο $n+k=0+300$ μονάδες (σχήμα 37). Τώρα η αποθήκη έχει μεγαλύτερη ποσότητα από τα 100 στη *matching*, άρα θα στείλει $k-rest(k,n)=k-0=100$ μονάδες. Συνολικά ο χονδρέμπορος στέλνει στον λιανέμπορο $100+100=200$

μαζί με το stock που είχε στείλει σε προηγούμενα βήματα. Στην αποθήκη του επιστρέφουν 200 κομμάτια.

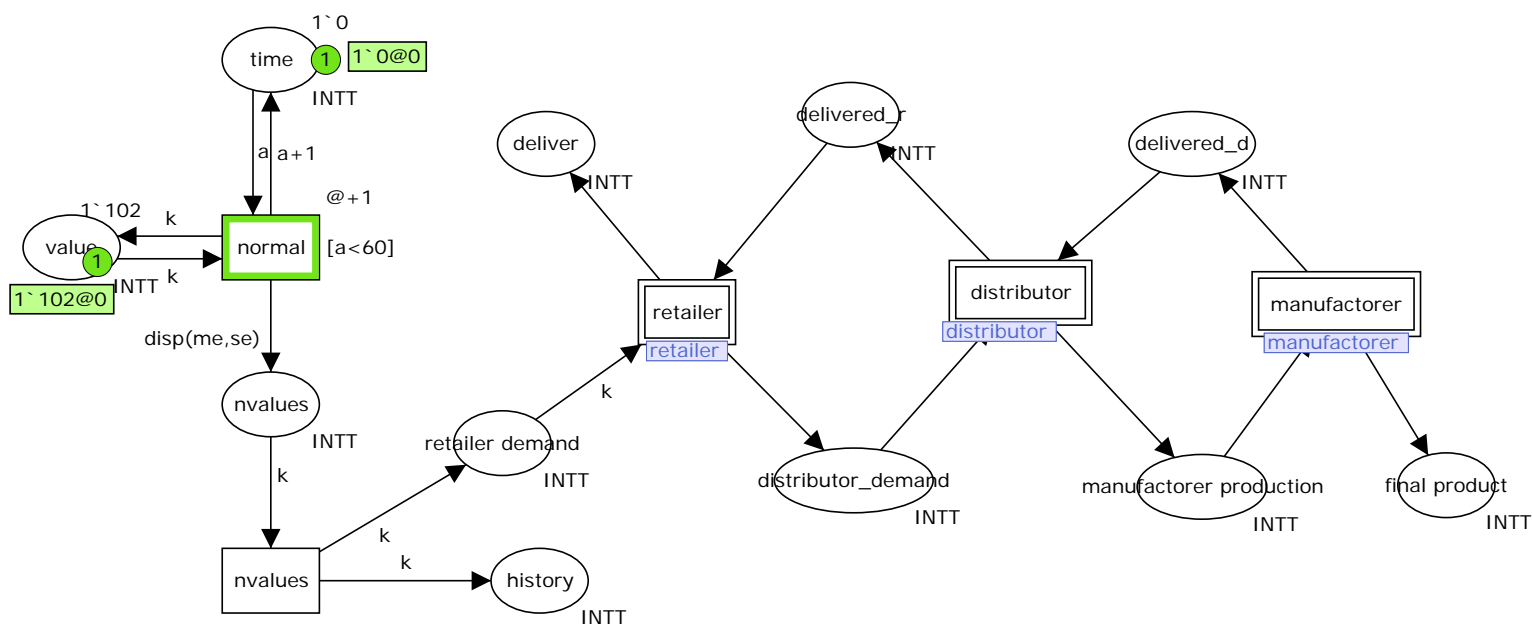
Η `delivered_r` είναι μαρκαρισμένη με 2 μάρκες των 100. Στο επόμενο βήμα οι αποθήκες θα έχουν 100 προϊόντα και θα στείλει στον πελάτη (`deliver`) τελικά $k\text{-rest}(k,n)=k=50$ κομμάτια, ενώ στο stock του λιανέμπορου προστίθεται και η δεύτερη μάρκα των 100 προϊόντων. Σε αυτή τη φάση ολοκληρώνεται η παραγγελία.

Το μοντέλο καταγράφει με ακρίβεια όλες τις διεργασίες των στρωμάτων της αλυσίδας ανεξαρτήτως μεγέθους και τύπου του αιτήματος του πελάτη. Υπάρχουν όμως τρία μειονεκτήματα. Πρώτον δίνεται λεπτομερή περιγραφή της ολοκλήρωσης ενός αιτήματος και όχι πολλών παραγγελιών σε βάθος χρόνου. Έτσι δεν δίνονται πληροφορίες για την απόδοση του συστήματος περιοδικά, για παράδειγμα για το πόσο καλά εξυπηρετεί τους πελάτες κατά την διάρκεια ενός έτους. Δεύτερον, από τη στιγμή που θα φύγει η παραγγελία από τον λιανέμπορο, παρατηρείται το φαινόμενο του συγχρονισμού. Δηλαδή κάποιες μεταβάσεις είναι ταυτόχρονα μαρκαρισμένες σε διαφορετικές υποσελίδες. Έτσι δεν γίνεται ξεκάθαρη η προτεραιότητα των διεργασιών. Τρίτον δεν ορίζεται πουθενά ο πραγματικός χρόνος που διεξάγεται κάποια διεργασία. Δηλαδή σε πόσο χρόνο θα μεταδοθεί μία πληροφορία, από τη στιγμή που θα γίνει γνωστό το αίτημα του πελάτη.

Για τον σκοπό αυτό τροποποιούμε το μοντέλο μας έτσι ώστε να δίνεται μία διαδοχή παραγγελιών. Για να συμβεί αυτό θα πρέπει στο δίκτυο πηγής να προστεθεί ένας υπογράφος ο οποίος θα παράγει συνεχώς τιμές. Οι τιμές αυτές βέβαια δεν θα είναι τελείως τυχαίες και ανεξάρτητες μεταξύ τους. Ανάλογα με τη σχέση των παραγγελιών που θέλουμε να υπάρχει εισάγουμε και την ανάλογη συνάρτηση. Στο παράδειγμα του σχήματος 40 δίνεται σήμα κανονικής κατανομής με μέση τιμή 100 (ίδια με τη ποσότητα των αποθηκών των μελών) και διασπορά 33. Επίσης για να αποφύγουμε την ατέρμονη συμπεριφορά του, εισάγουμε ως μέγιστο αριθμό βημάτων το 60. Οι υποσελίδες του δικτύου `hier` είναι ίδιες με αυτές του `hier0` με τη διαφορά ότι οι μεταβάσεις `fulfillment order` των `retailer` και `distributor` μαρκαρίστηκαν με `@+1`, η `fulfillment order` του `manufacturer` με `@+1` και η `delivery` του `manufacturer` με `@+1`. Κάτι τέτοιο σημαίνει ότι για να σταλεί μια παραγγελία στο προηγούμενο μέλος μεσολαβεί 1 χρονική στιγμή (1 μέρα για το παράδειγμά) και για την κατασκευή στο τελευταίο μέλος απαιτείται μία μέρα.

Στην έναρξη της προσομοίωσης η μετάβαση `normal` είναι μαρκαρισμένη με `@+1` για να θεωρηθεί ότι η πρώτη παραγγελία δίνεται στην πρώτη μέρα. Η κατάσταση `time` καταγράφει

τη χρονική στιγμή που δίνεται μία παραγγελία από τον πελάτη, στη nvalues δίνεται η τιμή της συνάρτησης $\text{disp}(me,se)$ της κανονικής κατανομής που θα αποτελέσει την παραγγελία που πελάτη και στη history καταχωρούνται και οι 60 παραγγελίες του πελάτη και σε ποιον χρόνο σημειώθηκαν (σχήμα 40).



Σχήμα 40. Η κεντρική σελίδα του δικτύου hier. Οι αρχικές παραγγελίες ακολουθούν κανονική κατανομή.

Μετά το τέλος της προσομοίωσης καταγράφονται τα αποθέματα και οι παραγγελίες που δεν διεκπεραιώθηκαν στον χρόνο τους (backorders) στα παρακάτω σχήματα.

Για τον λιανέμπορο, παρατηρούμε ότι για τις πρώτες 5 μέρες οι αποθήκες του είχαν μικρότερη ποσότητα από την αρχική (100). Αυτό συμβαίνει όταν οι παραγγελίες που δέχεται από πελάτες είναι μεγαλύτερες από τα αποθέματα. Την 6^η μέρα ο λιανέμπορος δέχεται ποσότητα από τον μεσολαβητή και το απόθεμά του φτάνει στα 378.

Τα αποθέματα του μεσολαβητή δεν διαφέρουν και πολύ. Την 5^η μέρα δέχεται ποσότητα από τον κατασκευαστή ενώ ποτέ δεν είχε μηδενικό απόθεμα. Την 60^η μέρα διαθέτει 350 κομμάτια στις αποθήκες του.

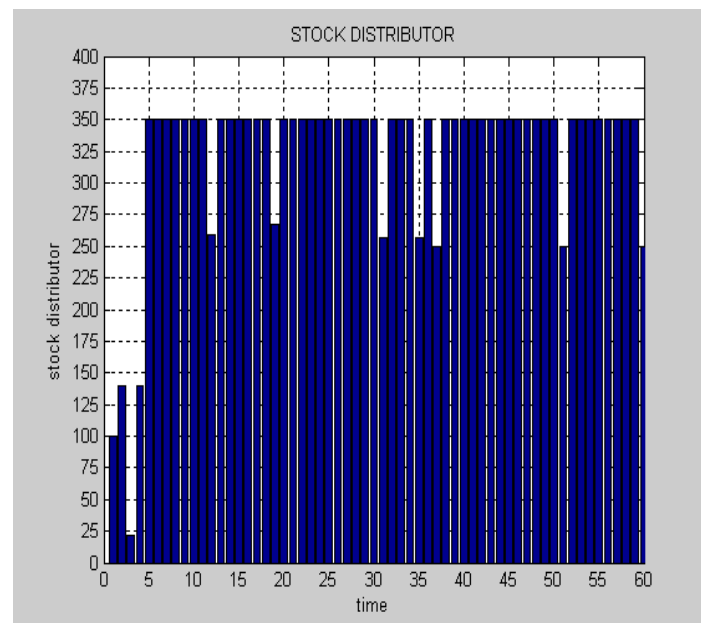
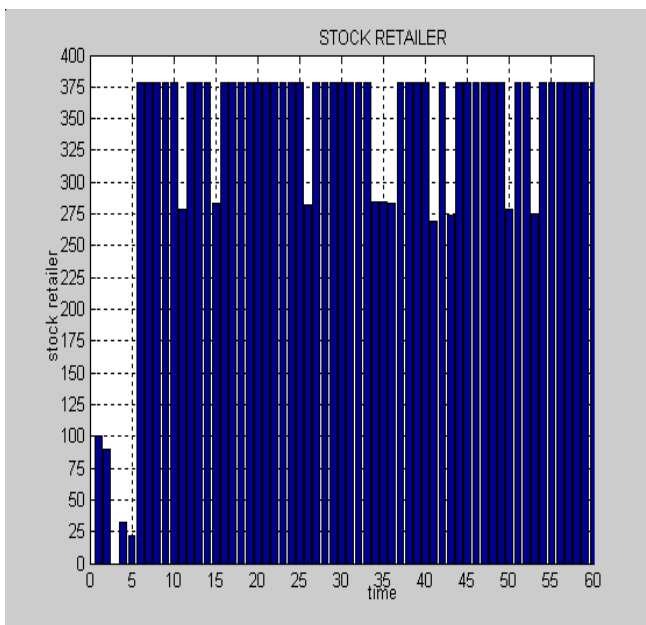
Ο κατασκευαστής από την έναρξη των παραγγελιών διαθέτει απόθεμα μεγαλύτερο από το αρχικό, ενώ από την 4^η κιόλας μέρα έχει γεμίσει τις αποθήκες του με 392 κομμάτια χωρίς καμία μεταβολή μέχρι και την τελευταία μέρα (σχήμα 41).

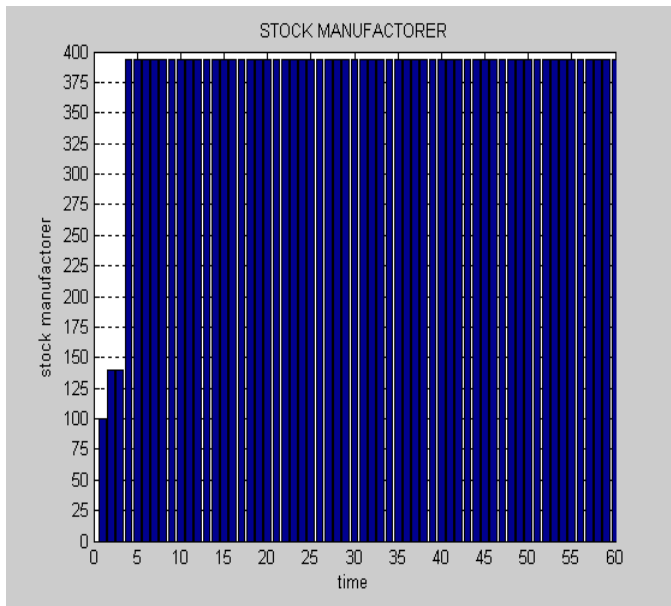
Ο λιανέμπορος εμφανίζει μέχρι και την 6^η μέρα backorder ενώ στη συνέχεια είναι μηδενικό. Αυτό συμβαίνει διότι εκείνη την ημέρα δέχτηκε την παραγγελία από τον μεσολαβητή, γέμισαν οι αποθήκες του και δεν δημιουργήθηκε ξανά έλλειμμα στο απόθεμα.

Ο μεσολαβητής αν και την 3^η μέρα εμφάνισε ένα μεγάλο backorder (180) μία μέρα νωρίτερα από τον λιανέμπορο εκπληρώνει την παραγγελία του.

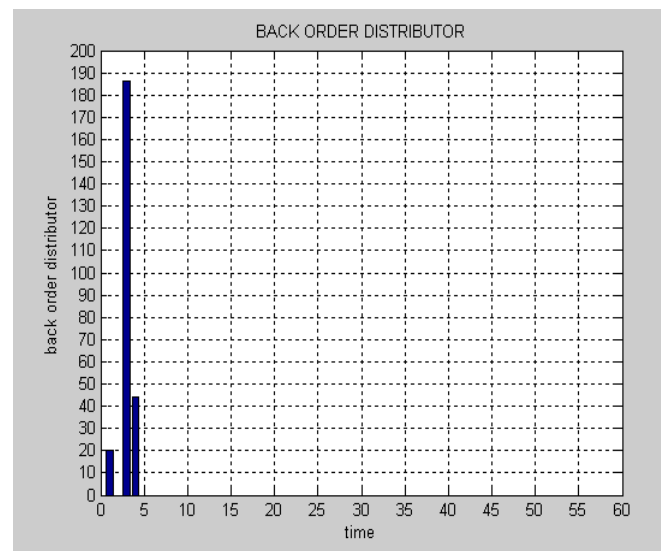
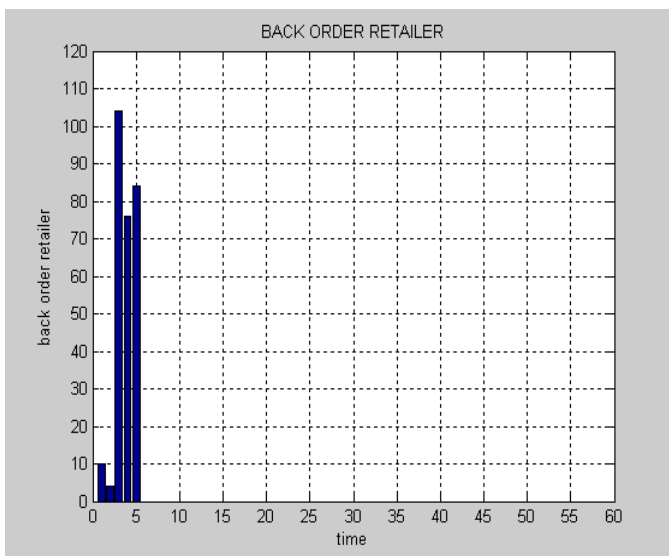
Τέλος ο κατασκευαστής την 3^η μέρα παρουσιάζει το μέγιστο υπόλοιπο παραγγελίας και από την επόμενη μέρα το απόθεμά του ήταν αρκετό για να ικανοποιήσει το αίτημα του προηγούμενου μέλους.

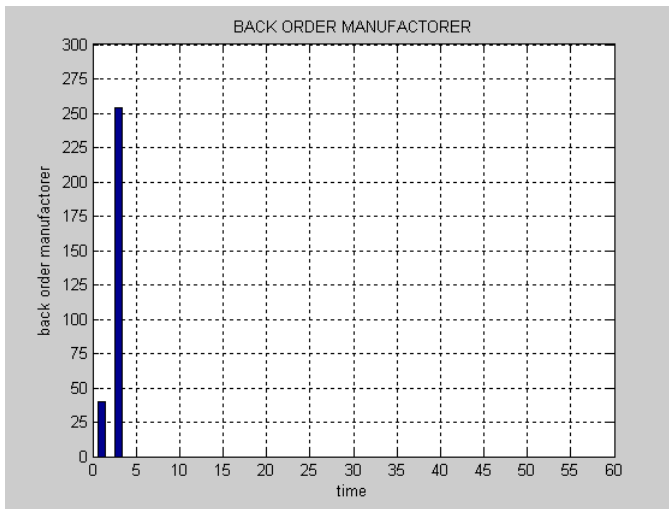
Θα μπορούσαμε να πούμε ότι το σύστημά μας χρησιμοποίησε την aggressive ordering ως τεχνική παραγγελιών αφού οι συμμετέχοντες έδιναν τις εντολές τους με τη βοήθεια της γραμμικής συνάρτησης $order(i,j)=2i-j$.





Σχήματα 41. Τα αποθέματα των retailer, distributor και manufacturer μετά από 60 μέρες αντίστοιχα.





Σχήματα 42. Τα back orders των retailer, distributor και manufacturer μετά από 60 μέρες αντίστοιχα.

Το φαινόμενο bullwhip στην αλυσίδα μας γίνεται ιδιαίτερα αισθητό, μιας και τα αποθέματα όλων των μελών έχουν αυξηθεί πάνω από 250% σε σχέση με το αρχικό. Η δημιουργία υψηλών backorders και η εξάλειψή τους μετά την 5^η μέρα δηλώνουν ότι τα αποθέματά τους είχαν αυξηθεί τόσο πολύ με αποτέλεσμα να ικανοποιούν τις παραγγελίες μόνο από τα αποθέματά τους.

Το γεγονός αυτό σε συνδυασμό με την δημιουργία σε κάποια χρονική στιγμή μηδενικού αποθέματος του λιανέμπορου, πολλαπλασιάζει το κόστος των μελών.

Επιπρόσθετα, κάτι τέτοιο μπορεί να αποφέρει σύγχυση στις επόμενες παραγγελίες μιας και όσο μεγαλύτερη είναι η παραγγελία του πελάτη τόσο η διαφορά της ποσότητας της αποθήκης από την αρχική αυξάνεται.

Κρίνεται λοιπόν αναγκαίο η επέκταση του μοντέλου μας σε ένα σύστημα που να μπορεί να υπολογίζει το κόστος διακίνησης των συμμετεχόντων αλλά και τη χρήση κατάλληλων μεθόδων πρόβλεψης ώστε το φαινόμενο bullwhip να είναι ελεγχόμενο.

3.2 Υπολογισμός κόστους- Συνεχής ανατροφοδότηση

Οι σημαντικότεροι παράγοντες που επηρεάζουν κόστος μιας αλυσίδας, είναι η ποσότητα των αποθεμάτων, η μεταφορά, η κατασκευή, οι ποσότητες που θα παραγγείλει το κάθε μέλος αλλά και το έλλειμμα των παραγγελιών που δημιουργείται. Στο μοντέλο μας εισάγονται μεταβλητές κόστους και μία συνάρτηση η οποία υπολογίζει τις δαπάνες αθροιστικά κατά την διάρκεια της προσομοίωσης. Στη συνέχεια θα χρησιμοποιηθούν τεχνικές πρόβλεψης παραγγελιών προκειμένου να εξομαλυνθούν τα ακραία αποτελέσματα που διαπιστώθηκαν.

3.2.1 Aggressive ordering

Στο κομμάτι των εντολών (σχήμα 43) προσθέτουμε τις εξής μεταβλητές:

cs=12: Το κόστος αποθήκευσης ανά μονάδα προϊόντος.

cbo=50: Το κόστος ελλείμματος της αποθήκης (backorder) ανά μονάδα προϊόντος.

co=9: Το κόστος παραγγελίας ανά μονάδα προϊόντος

cd=20: Το κόστος μεταφοράς ανά μονάδα προϊόντος.

cman=9: Το κόστος κατασκευής ανά μονάδα προϊόντος.

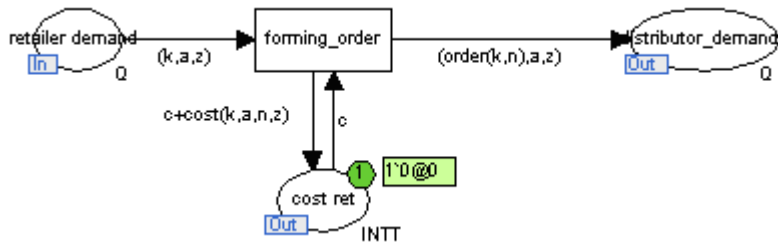
Στα σχήματα 43 και 44 διακρίνονται οι αλλαγές που απαιτούταν ώστε το ίδιο το σύστημα να υπολογίζει το κόστος. Η δομή του distributor είναι ίδια με την αρχική τροποποιημένη κατά τον ίδιο τρόπο όπως στον κατασκευαστή.

Η συνάρτηση $cost(i,j,l,x)=cs*I+cbo*rest(i,l)+orman(x)*order(i,j)$ δέχεται ως είσοδο 4 μεταβλητές στο δίκτυο $cost(k,a,n,z)$ (εκτός από το τελευταίο μέλος όπου το stock συμβολίζεται με m).

Η δαπάνη ελλειμμάτων στο απόθεμα δίνεται από $cbo*rest(k,n)$ όπου rest η συνάρτηση που δίνει τη ποσότητα $backorder=k-n$ αν $k-n>0$ και 0 σε άλλη περίπτωση.

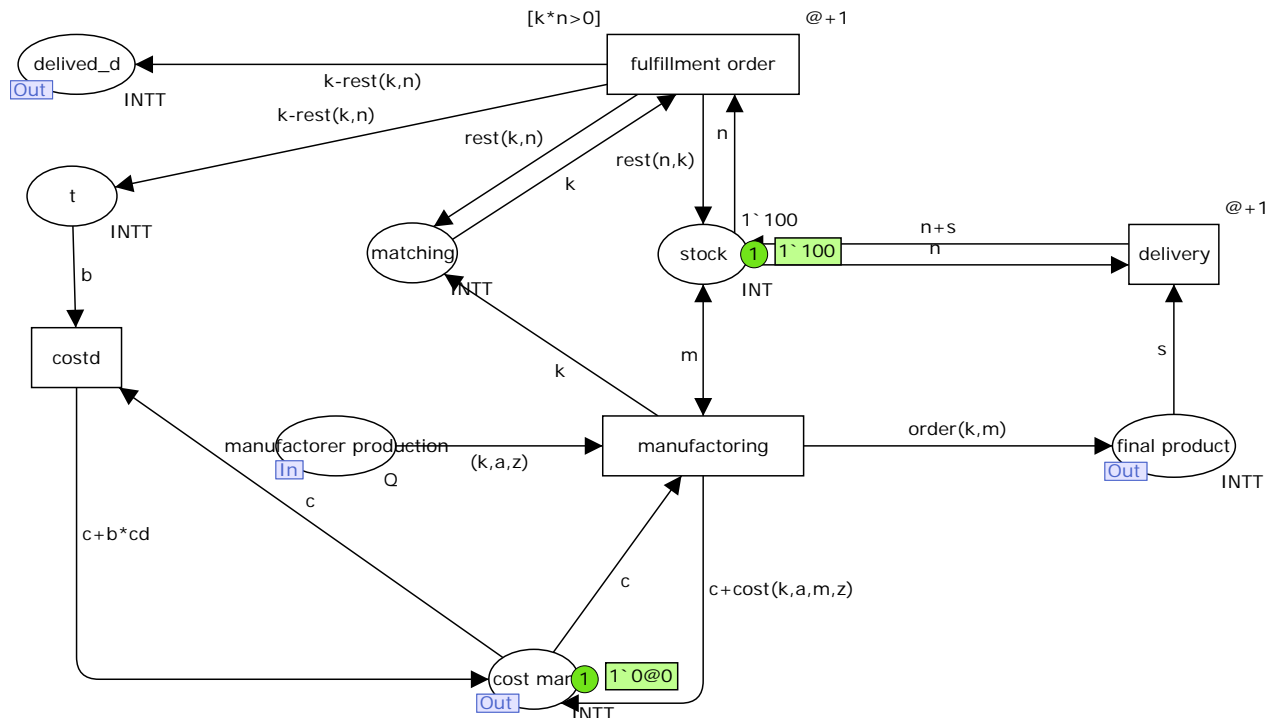
Με τη βοήθεια της συνάρτησης $orman(z)$ βρίσκουμε το κόστος παραγγελίας= $orman(z)*order(k,n)$. Η $orman$ είναι διμερής και επιστρέφει co όταν $z=0$ και cman όταν $z=1$. Στο δίκτυο, στα ιεραρχικές σελίδες η μεταβλητή z παίρνει την τιμή 1 μόνο στον κατασκευαστή. Άρα η συνάρτηση υπολογίζει το κόστος παραγγελίας στους retailer και distributor ανά μονάδα προϊόντος (όταν $z=0$) και το κόστος κατασκευής ανά μονάδα προϊόντος (όταν $z=1$). Στο συγκεκριμένο βέβαια παράδειγμα τυγχάνει $co=cman$.

Το κόστος υπολογίζεται αθροιστικά με $c=c+cost(k,a,n,z)$ όπου c το κόστος του μέλους όταν δώσει παραγγελία (αφού πυροδοτήσει η forming order) c με αρχική τιμή ίση 0. Στην πρώτη παραγγελία, όταν ενεργοποιηθεί η κατάσταση, το κόστος θα είναι $0+cost(k,1,n,z)$ ενώ στο επόμενο βήμα θα είναι $0+cost(k,1,n,z)+cost(k',2,n',z)$ όπου k',n' η νέα παραγγελία και απόθεμα αντίστοιχα. Έτσι μετά από 120 χρόνους στη μετάβαση «cost ter» αναγράφεται το σύνολο των δαπανών του λιανέμπορου.



Σχήμα 43. Ο υπογράφος που προστέθηκε στη σελίδα του retailer προκειμένου να υπολογιστεί το κόστος.

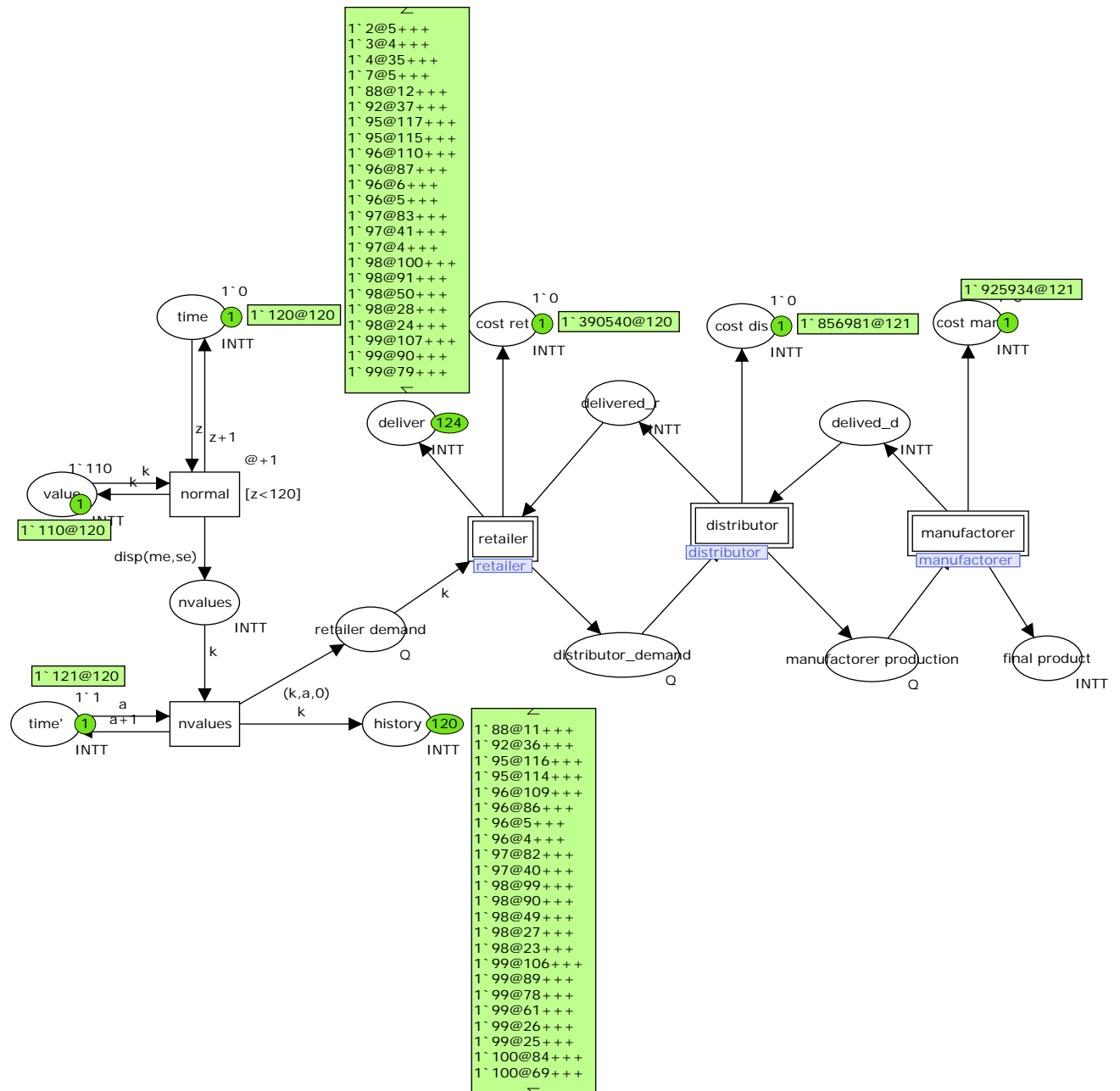
Στο σχεδιασμό των άλλων δύο συμμετεχόντων υπάρχει μία διαφοροποίηση και αυτό γιατί θα πρέπει στο κόστος να συνυπολογιστεί και το κόστος μεταφοράς, μια παράμετρος που δεν αφορά τον λιανέμπορο. Έτσι, από τη στιγμή που θα αποφασιστεί από τον μεσολαβητή ή τον κατασκευαστή πόση ποσότητα θα στείλει στο προηγούμενο του μέλος (ενεργοποίηση fulfillment order) και ετοιμάζεται να τα στείλει (πυροδότηση), ταυτόχρονα ενεργοποιείται η «costd» με $d=k-rest(k,n)$. Μετά την πυροδότησή της στέλνει $c+b*cd$ στη μετάβαση cost man (ή cost dis του μεσολαβητή). Ουσιαστικά στο ήδη υπάρχον κόστος c προσθέτουμε το κόστος μεταφοράς το οποίο υπολογίζεται αφού φύγει η παραγγελία από το μέλος. Από τη στιγμή που σταλεί η παραγγελία στην μετάβαση cost man θα είναι καταχωρημένη η συνολική δαπάνη μέχρι το συγκεκριμένο βήμα.



Σχήμα 44. Η τροποποίηση της σελίδας *manufacturer* προκειμένου να υπολογίζει και το συνολικό κόστος διακίνησης.

Οι μεταβάσεις κόστους και στις τρεις ιεραρχικές σελίδες αποτελούν πύλες εξόδου. Έτσι σε κάθε subpage του κεντρικού δικτύου hier επισυνάπτονται οι μεταβάσεις κόστους, ώστε να γίνεται εμφανή η αύξηση των δαπανών των μελών ακόμα και όταν η παρατήρηση της διακίνησης πραγματοποιείται μόνο από την κεντρική σελίδα.

Στο σχήμα 45 μόλις έχει ολοκληρωθεί και το 120^ο βήμα. Ο πελάτης παραγγέλνει ποσότητες τυχαίες βάση κανονικής κατανομής. Διαπιστώνουμε ότι οι δαπάνες για τους συμμετέχοντες είναι 390.540 χρηματικές μονάδες για τον λιανέμπορο, 856.981 χ. μονάδες για τον μεσολαβητή και 925.934 χ.μ. Αν θέλουμε να δούμε το κόστος κάθε βήματος ξεχωριστά θα μπορούσαμε να προσθέσουμε άλλη μία μετάβαση στα place κόστους με μία μόνο θέση εισόδου την μεταβλητή c , όπως γίνεται στην μετάβαση *history* όπου καταγράφονται όλες οι παραγγελίες του πελάτη. Επίσης, με εργαλεία το *simulation report* και το *state space* μπορούμε να έχουμε μια πιο ακριβή εικόνα για τις ποσότητες που διακινούνται σε κάθε βήμα. Επισημαίνεται ότι η επιλογή των ποσοτήτων των παραγγελιών είναι τυχαία.



Σχήμα 45. Η κεντρική σελίδα του δικτύου hier μετά το τέλος της προσομοίωσης.

```

▼ Declarations
▼ val se=33.0;
▼ val me=105.0;
▼ val cs=12;
▼ val cbo=50;
▼ val co=9;
▼ val cd=20;
▼ val cman=9;
▼ colset INT=int;
▼ colset INTT=INT timed;
▼ colset U=product INTT*INTT;
▼ colset P=product INTT*INTT*INTT*INTT;
▼ colset Q=product INTT*INTT*INTT;
▼ var k,m,n,s,a,b,z,v,c:INTT;
▼ normal(me,se): real;
▼ fun disp(me,se)=floor(normal(me,se));
▼ fun rest(i,j)=if (i>=j) then i-j else 0;
▼ fun order(i,j)=if i<=j then i else i+i-j;
▼ fun orman(i)=if i=0 then co else cman;
▼ fun cost(i,j,l,x)=cs*I+cbo*rest(i,l)+orman(x)*order(i,l);

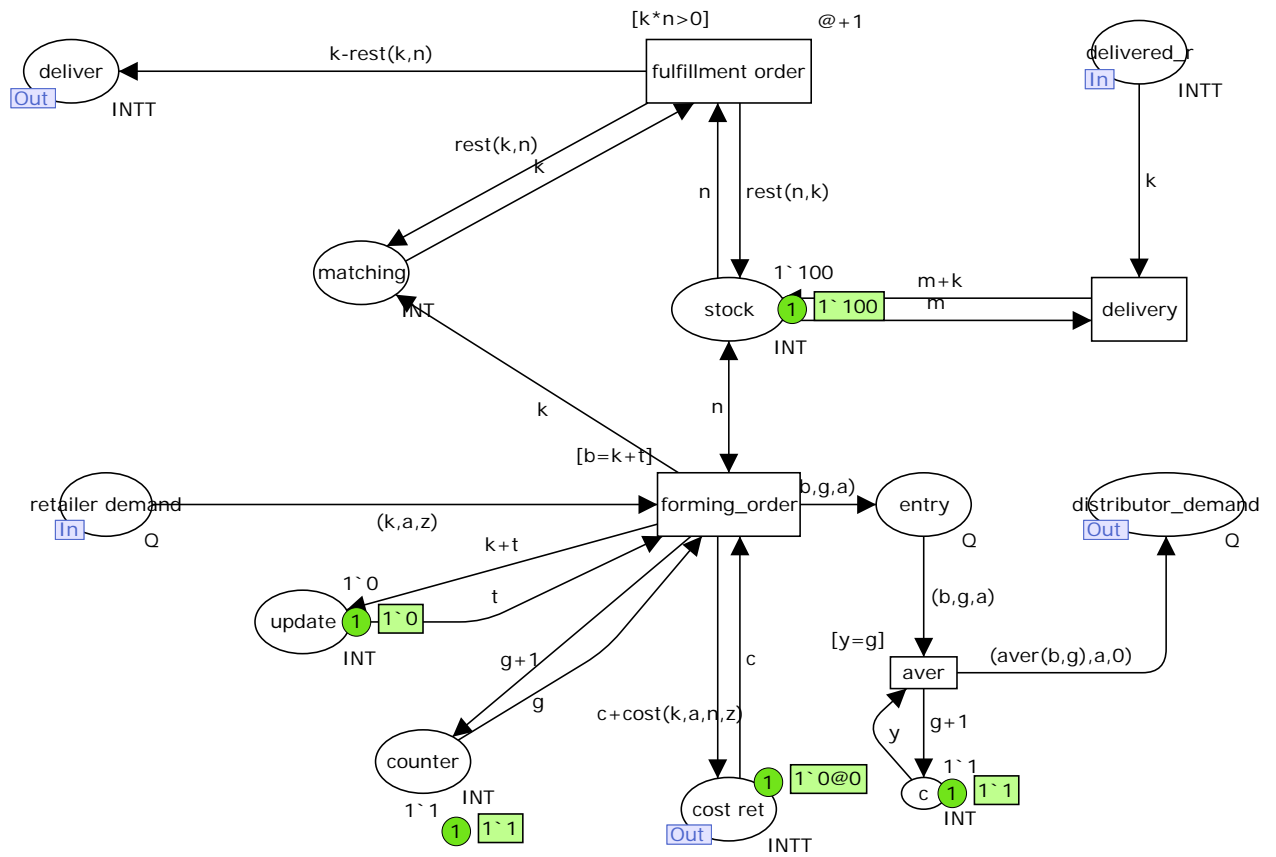
```

Σχήμα 46. Η εντολές του δικτύου.

3.2.2 Moving average

Μία εναλλακτική πολιτική που μπορεί να χρησιμοποιηθεί ώστε να προβλεφθούν οι εισερχόμενες παραγγελίες πραγματοποιείται με τη χρήση συναρτήσεων κινητού μέσου (moving average). Επιλέγουμε, η συνάρτηση να χρησιμοποιηθεί μόνο στον λιανέμπορο για να μπορέσουμε στο τέλος να επιλέξουμε την πιο συμφέρουσα πολιτική για αυτόν μιας και ο ίδιος στην πραγματικότητα δεν γνωρίζει τη πολιτική και τη διαχείριση των άλλων μελών. Όπως στο beer game όπου ο παίχτης έπαιρνε τη θέση του ενός, ενώ δεν ήξερε τις κινήσεις των υπολοίπων. Ο μεσολαβητής και ο κατασκευαστής δίνουν παραγγελίες με τον ίδιο ακριβώς τρόπο με τη βοήθεια της συνάρτησης order.

Στο σχήμα 47 φαίνεται πώς ο retailer δίνει τις παραγγελίες του στον distributor με βάση την $aver(b,g)$.



Σχήμα 47. Η σελίδα του λιανέμπορου ο οποίος δίνει με τη βοήθεια συνάρτησης κινητού μέσου.

Χρησιμοποιούμε τον μετρητή βημάτων «counter» g με αρχική τιμή 1, έναν αθροιστή των εισερχόμενων παραγγελιών t και μία μεταβλητή $b=k+t$ που υπολογίζει αθροιστικά τις παραγγελίες μαζί με την πιο πρόσφατη. Ο κόστος υπολογίζεται όπως και στο προηγούμενο δίκτυο. Επίσης η μεταβλητή y όταν γίνεται ίση με την g τότε επιστρέφει στην «aver» κατάσταση η οποία θα στείλει $aver(b,g)$ παραγγελίες. Η σύνταξη της συνάρτησης $aver$ δίνεται παρακάτω.

```

fun conv(i)=Real.fromInt((i));
fun aver(i,j)= round(conv(i)/conv(j));

```

Με τη βοήθεια της συνάρτησης $conv$ η οποία μετατρέπει τον i από φυσικό σε

πραγματικό η $aver$ υπολογίζει το πηλίκο $\frac{1}{n} \sum_{i=1}^n x_i$.

Μολονότι ο υπολογισμός κινητού μέσου ανήκει στις μεθόδους συνεχόμενης παραγγελιοληψίας (continuous replacement policy), δεν φαίνεται να ανταποκρίνεται άμεσα σε εναλλαγές των δεδομένων και των κριτηρίων. Έτσι δεν μπορεί να αποτελέσει ισχυρό εργαλείο για να προβλέψει διακυμάνσεις στις παραγγελίες, αλλά βοηθάει τον αναλυτή να αποφασίσει τις επόμενες κινήσεις του ανάλογα με τη προηγούμενη συμπεριφορά του μοντέλου. Παρόλο αυτά μακροσκοπικά και για μεγάλο αριθμό βημάτων έχει διαπιστωθεί ότι είναι ένα από τα καταλληλότερα προγνωστικά εργαλεία. Για αυτό το λόγο σε παραγγελίες που τους χαρακτηρίζει η περιοδικότητα και η επανάληψη βρίσκει μεγάλη εφαρμογή. Ιδιαίτερη η χρησιμότητα του σε συστήματα που είναι βασισμένα στη συνεργασία.

Σε αντίθεση με τα γραμμικά μοντέλα, όπου η πρόβλεψή τους στηρίζεται αποκλειστικά στην αμέσως προηγούμενη παραγγελία, η συνάρτηση κινητού μέσου, κυρίως σε μεγάλους χρόνους παραμένει ανθεκτική στις απρόσμενες διακυμάνσεις. Όταν η ποσότητα μιας παραγγελίας αυξάνεται απότομα, η συνάρτηση θα αυξηθεί ανάλογα με τον αριθμό των βημάτων έχουν προηγηθεί.

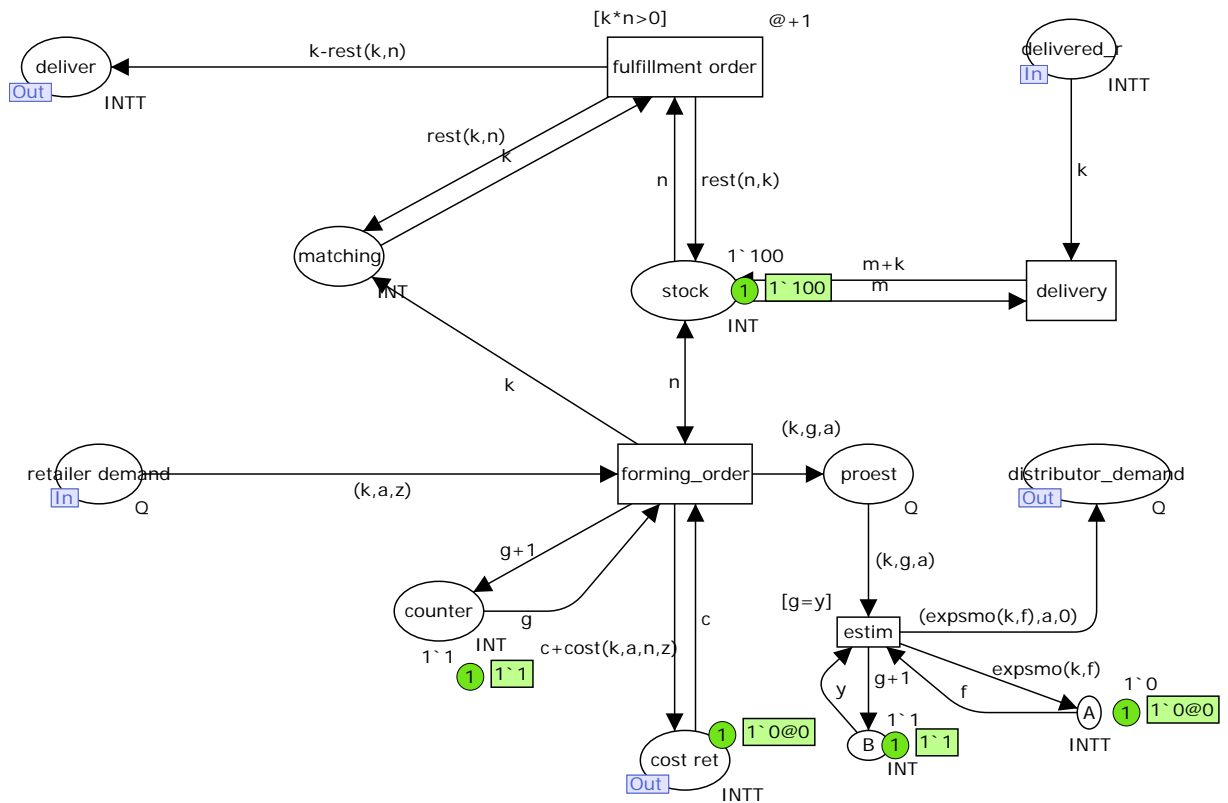
Συμπεραίνουμε ότι η σωστή χρήση της συνάρτησης κινητού μέσου εξαρτάται πρώτον από τα χρονικά πλαίσια που θα ορίσει ο αναλυτής, μιας και σε άπειρους χρόνους καταγράφεται ασυμπτωτική συμπεριφορά και δεύτερον από την σταθερότητα ή μη των παραγγελιών.

3.2.3 Exponential smoothing

Όπως έχει αναλυθεί και στο κεφάλαιο 1 της εφοδιαστικής αλυσίδας, μια συνάρτηση εκθετικής εξομάλυνσης (exponential smoothing) ορίζεται και ως σταθμικός μέσος όρος, δίνοντας στις πιο πρόσφατες παραγγελίες μεγαλύτερο συντελεστή βαρύτητας από τις παλιότερες και ο τύπος της δίνεται από τη σχέση

$$D(n) = aD(n-1) + (1-a)x_n \quad (1)$$

όπου a ο συντελεστής βαρύτητας και $D(n)$ η ποσότητα των παραγγελιών.



Σχήμα 48. Η σελίδα του λιανέμπορου ο οποίος δίνει παραγγελίες με τη βοήθεια εκθετικής εξομάλυνσης.

Για την σωστή λειτουργία του δικτύου, στη σελίδα “retailer” απαιτείται η εισαγωγή του μετρητή χρόνου g με αρχική τιμή 1, του κόστους c , η είσοδος y της μετάβασης «estim» η οποία την ενεργοποιεί μόνο όταν $y=g$ με αρχική τιμή 1 και η f με αρχική τιμή 0. Η f παίζει το ρόλο της $D(n-1)$ της προηγούμενης τιμής της συνάρτησης (1). Η σύνταξη της καινούριας συνάρτησης είναι

$$\text{fun expsmo}(i,j)=\text{round}((\alpha*\text{conv}(i)+(1.0-\alpha)*\text{conv}(j)));$$

όπου conv συνάρτηση που μετατρέπει έναν φυσικό αριθμό σε πραγματικό [10].

Η μέθοδος της εκθετικής εξομάλυνσης χρησιμοποιείται αποτελεσματικότερα στις παρακάτω περιπτώσεις:

- Όταν ο χρόνος στον οποίο αναφέρεται είναι σχετικά μικρός, για παράδειγμα βδομαδιαίος ή μηνιαίος.
- Εξαιτίας του μεγάλου συντελεστή σημαντικότητας στις τελευταίες παραγγελίες δεν είναι τόσο απαραίτητο ο συμμετέχων να γνωρίζει τις κινήσεις των υπολοίπων.
- Όταν απαιτείται η πρόβλεψη παραγγελιών που δεν ακολουθούν κάποια συγκεκριμένη κατανομή.
- Σε απότομες εναλλαγές τις διακυμάνσης των ποσοτήτων που χαρακτηρίζονται εποχιακά.

Οι δύο βασικές γνώσεις που απαιτούνται για την καλύτερη πρόβλεψη αυτής της μεθόδου είναι η γνώση των ισχυρών παρατηρήσεων κατά την διάρκεια μιας περιόδου και για πόσο και πότε παρατηρείται η εμφάνιση των ασθενών παρατηρήσεων. Η αυτοματοποίηση των δεδομένων και τα αποτελέσματα πρόβλεψης απλουστεύονται αν γίνουν γνωστές οι επικρατέστερες παρατηρήσεις (με την μεγαλύτερη συχνότητα) και η αμέσως προηγούμενη παραγγελία.

Επίσης ο αναλυτής θα χρειαστεί να προβλέψει την επόμενη περίοδο όπου θα παρατηρηθούν οι ακραίες τιμές. Αν το χρονικό πλαίσιο είναι σχετικά μικρό, τότε οι αποκλίσεις αυτών των περιόδων προβλέπονται με τη βοήθεια της exponential με άλλους τώρα παράγοντες. Σε χρονικά μεγέθη έτους ή και μεγαλύτερα η χρήση κινητού μέσου κρίνεται αναγκαία.

Η βασική ιδέα στη χρήση αυτής της συνάρτησης ήταν να βρεθεί ένας τρόπος ώστε να πετύχουμε μία ικανοποιητική πρόβλεψη σε μία παλίνδρομη ζήτηση. Όσο πιο μεγάλη είναι η τιμή του α , σημαίνει ότι η συνάρτηση πρόβλεψης αντιδρά ταχύτερα στις αλλαγές, αλλά γίνεται πιο ευαίσθητη σε στοχαστικές αποκλίσεις. Για αυτό το σκοπό σε μικρούς χρόνους προτείνονται οι μεγάλες τιμές του α . Αντίθετα όταν η τιμή του α πλησιάζει στο 0,5 η συνάρτηση εκθετικής εξομάλυνσης εκφυλίζεται σε συνάρτηση κινητού μέσου. Προτείνεται σε συστήματα όπου οι διακυμάνσεις στη ζήτηση παρατηρούνται σε βάθος χρόνου.

Συμπερασματικά, η πρόβλεψη στη ζήτηση εξαρτάται άμεσα από τον χρόνο στον οποίο μελετάτε το σύστημα και από την καταγραφή των προηγούμενων παρατηρήσεων. Όταν το μοντέλο μας μελετάται βραχυπρόθεσμα με έντονες διακυμάνσεις, τότε η εκθετική εξομάλυνση είναι καταλληλότερη. Αντίθετα, για μεγάλους χρόνους με ισχυρές παρατηρήσεις η συνάρτηση κινητού μέσου προσεγγίζει σε μεγάλο ποσοστό τη γραμμική πραγματικής ζήτησης. Ας

υποθέσουμε ότι σε μία εκτίμηση με τη χρήση των δύο γνωστών συναρτήσεων ορίζουμε ένα διάστημα εμπιστοσύνης μέσα στο οποίο βρίσκεται η τιμή της πραγματικής ζήτησης με κάποιο σφάλμα κ (ή αλλιώς η πιθανότητα η πραγματική τιμή να βρίσκεται μέσα σε αυτό το διάστημα εμπιστοσύνης είναι $1-\kappa$). Γνωρίζουμε ότι υπάρχουν και άλλοι μέθοδοι πρόβλεψης οι οποίοι στηρίζονται στη χρήση συναρτήσεων που συνδυάζουν τα θετικά των δύο προηγούμενων. Αν χρησιμοποιήσουμε τις συναρτήσεις αυτές, το σφάλμα στην εύρεση του διαστήματος εμπιστοσύνης θα είναι μεγαλύτερο, ή για να αποφύγουμε κάτι τέτοιο χρησιμοποιούμε διάστημα εμπιστοσύνης με μεγαλύτερο πλάτος. Τότε βέβαια η πραγματική τιμή της παρατήρησης προσεγγίζεται πιο δύσκολα. Για αυτό το λόγο οι δύο μέθοδοι που αναλύθηκαν παραπάνω, είναι και οι αποδοτικότεροι.

Και στις τρεις προηγούμενες περιπτώσεις που μελετήθηκαν (μαζί με aggressive order), κάθε φορά που κάποιο μέλος δεχόταν μία παραγγελία, ανάλογα με την πολιτική που ακολουθούσε και την εκτίμηση που έκανε έδινε την δικιά του παραγγελία στο επόμενο μέλος της αλυσίδας. Δηλαδή ο συμμετέχων εκτελούσε παραγγελία ανεξαρτήτως του αποθέματος ή του backorder που ενδεχομένως δημιουργούνταν. Τέτοιες μέθοδοι παραγγελιών ανήκουν στις «continuous replacement policy».

3.3 Υπολογισμός κόστους- Στοχαστική ανατροφοδότηση

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι πολιτικές παραγγελιών που δεν πραγματοποιούνται σε κάθε βήμα της συναλλαγής αλλά ανάλογα με το επίπεδο των αποθεμάτων, ή άλλων παραγόντων. Τέτοιες πολιτικές είναι οι λεγόμενες «stochastic replacement policies».

Στη συνέχεια θα μελετηθούν οι μέθοδοι παραγγελιών ανάλογα με το ανώτερο ή το κατώτερο όριο των αποθεμάτων.

Η διαφοροποίηση αυτών των μεθόδων γίνεται με βάση δύο κριτηρίων: Τους μηχανισμούς που χρησιμοποιούνται και το μέγεθος της ποσότητας των αποθεμάτων που πρέπει να φτάσουν για να ενεργοποιηθούν οι μηχανισμοί αυτοί. Υποθέτουμε ότι η μεταβλητή Q συμβολίζει τη ποσότητα της παραγγελίας (order quantity), η s το επίπεδο των αποθεμάτων κάτω από το οποίο πραγματοποιείται παραγγελία (reorder point) και S την ανώτερη ποσότητα που θα φτάσει στις αποθήκες μετά από μία παραγγελία (order level). Οι (s,Q) policy (s,S) policy θα αναλυθούν εκτενέστερα στη συνέχεια.

- **s-Q**

Ο μηχανισμός s-Q, ενεργοποιείται, (ένα μέλος της αλυσίδας εκτελεί μια παραγγελία) όταν το απόθεμα γίνει μικρότερο από s. Τότε ο συμμετέχων θα παραγγείλει ακριβώς ποσότητα Q-s. Για τον λόγο αυτό η κατάσταση των αποθεμάτων παρακολουθείτε συνέχεια. Για παράδειγμα, στην περίπτωση ενός (s,Q) μηχανισμού με s=100 Q=200 αν το απόθεμα πέσει στα 95 θα εκτελεστεί παραγγελία ίση με 200-100=100. Όσο είναι από 100 και πάνω, δεν υπάρχουν καινούριες παραγγελίες.

- **s-S**

Η (s,S) είναι μία πολιτική ανεφοδιασμού, κατά την οποία όταν το επίπεδο των αποθεμάτων πέσει κάτω από έναν ελάχιστο αριθμό s τότε δίνεται παραγγελία έτσι ώστε οι αποθήκες να γεμίσουν με ποσότητα S. Για παράδειγμα, αν s=50 και S=100 και στις αποθήκες υπάρχουν 48 κομμάτια, τότε η παραγγελία θα είναι 100-48=52.

Στο δίκτυο hier που μελετάμε, ο λιανέμπορος παραγγέλνει από τον μεσολαβητή με τη βοήθεια της συνάρτησης ss:

fun ss(i,j)=if i<j then i else i-j;

Θεωρούμε ότι η κατώτερη τιμή του αποθέματος είναι 0. Για να πέσει η ποσότητα του stock κάτω από αυτό το όριο σημαίνει ότι θα δημιουργηθεί έλλειμμα στις αποθήκες (backorders).

Η κύρια διαφορά μεταξύ της s, Q και s, S είναι ότι στη δεύτερη περίπτωση λαμβάνεται υπόψη πόσο κάτω του επιπέδου αποθεμάτων βρισκόμαστε όταν δίνεται η παραγγελία ενώ στην πρώτη περίπτωση όσο κάτω από το επίπεδο αυτό και να πέσουν η παραγγελία παραμένει σταθερή.

Η συμπεριφορά του συστήματος εξαρτάται άμεσα από τους εξής παράγοντες:

- Πρώτα από όλα από την πολιτική που θα ακολουθήσουμε ανάλογα πάντα με τις συνθήκες της δοσοληψίας.

- Από το όριο s που θα ορίσουμε. Όταν το επίπεδο αποθεμάτων είναι ιδιαίτερα υψηλό οι παραγγελίες είναι πιο συχνές και αυξάνεται το κόστος μεταφοράς. Αντίθετα χαμηλή τιμή της s μπορεί να σημαίνει έλλειψη ποσότητας στις αποθήκες, δημιουργίας backorders και αύξηση του κόστους που αφορά την καθυστέρηση παράδοσης παραγγελίας.

- Από τις τιμές του Q ή S ανάλογα με την πολιτική που θα χρησιμοποιηθεί. Όταν η τιμή Q είναι μεγάλη τότε αυξάνεται δυσανάλογα το απόθεμα και κατά συνέπεια το αντίστοιχο κόστος. Κάτι ανάλογο συμβαίνει και με την τιμή του S .

- Από το πόσο συχνά ελέγχονται οι αποθήκες. Για να είναι όλη η διαδικασία πιο οργανωτική και αποδοτική θα πρέπει να επιλεγεί κατάλληλη περίοδος ανασκόπησης των αποθηκών (review Period). Το καλύτερο σενάριο, είναι να γίνεται έλεγχος κάθε φορά που παραδίδεται μία παραγγελία. Αν κάτι τέτοιο δεν είναι εφικτό, ανάλογα με την συχνότητα των εισερχόμενων παραγγελιών, η συντομότερη καταγραφή των αποθεμάτων σίγουρα διευκόλυνε την διαδικασία.

Ιδιαίτερο ενδιαφέρον παρουσιάζουν οι **ανατροφοδοτήσεις καθορισμένου χρόνου (replenishment lead time)**. Αν θεωρήσουμε ότι r είναι το διάστημα του χρόνου μέσα στο οποίο αναμένεται η ποσότητα των αποθεμάτων να φτάσει σε S . Αν μετά από r χρόνο το απόθεμα n είναι μικρότερο από S τότε παραγγέλνεται ποσότητα $S-n$ τόση ώστε να φτάσει στην επιθυμητή. Αν ξεπεράσει την S τότε αναμένουμε διάστημα r χωρίς να γίνει παραγγελία για να γίνει στη συνέχεια ο σχετικός έλεγχος. Η μέθοδος έχει καταγραφεί ως **(r,S)** και διαφέρει από τις προηγούμενες, μιας και το σημείο στο οποίο γίνεται έλεγχος για ανατροφοδότηση δεν είναι ποσότητα στο απόθεμα αλλά χρόνος.

ΚΕΦΑΛΑΙΟ 4

Ανάλυση δεδομένων

4.1 Συνεχής ανατροφοδότηση

4.1.1 Κατάσταση αποθεμάτων

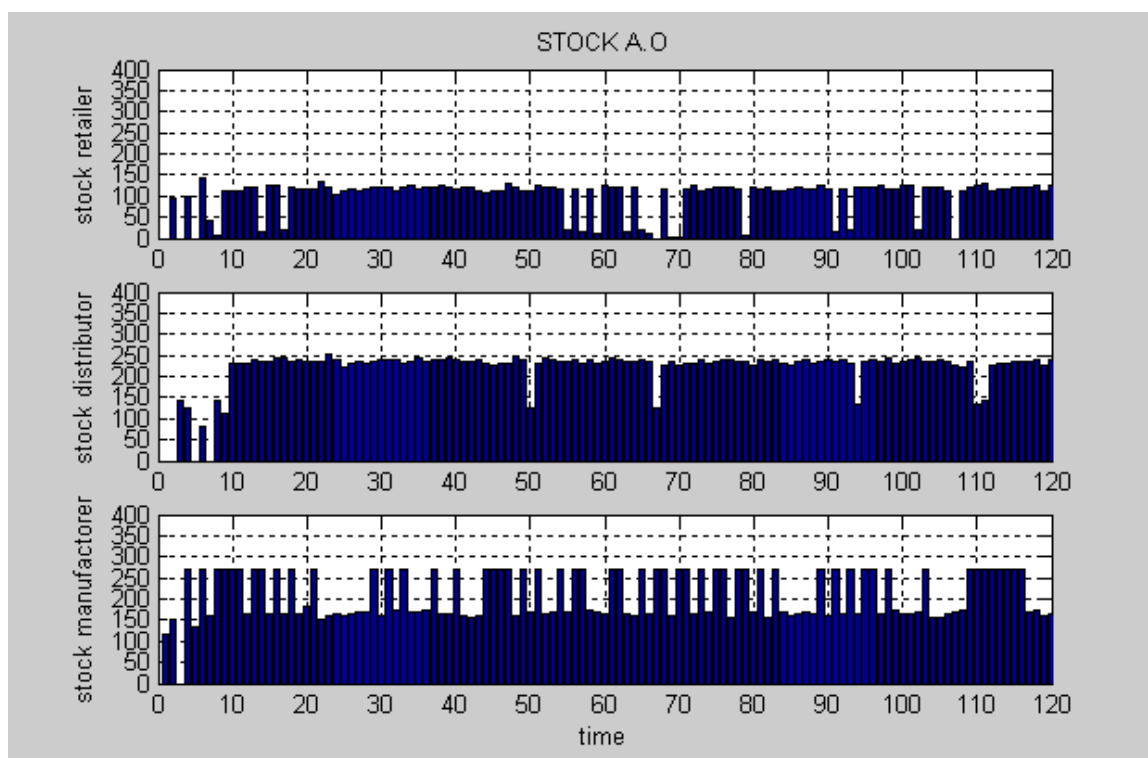
Στα σχήματα 49-52 καταγράφονται οι διακυμάνσεις των αποθεμάτων των συμμετεχόντων κατά τη διάρκεια των 120 χρόνων, με τις τρεις μεθόδους συνεχόμενης ανατροφοδότησης (continuous replacement).

Για την κάθε περίπτωση μεμονωμένα, παρατηρούμε ότι η ποσότητα των αποθεμάτων αυξάνεται όσο περνάμε στα ανώτερα στρώματα της αλυσίδας. Ο λιανέμπορος εμφανίζεται με το μικρότερο stock, αμέσως μετά ο μεσολαβητής και στη συνέχεια ο κατασκευαστής με τη μεγαλύτερη ποσότητα στις αποθήκες. Ένας λόγος που παρατηρείται το φαινόμενο, είναι εξαιτίας της χρήσης της γραμμικής συνάρτησης ως τρόπος παραγγελίας. Η εντολή order θα έδινε ποσότητα ίση με την εισερχόμενη παραγγελία (k) αν ήταν μικρότερη από το απόθεμα, ή ίση με την k μαζί με την διαφορά της από το απόθεμα. Άρα ή η παραγγελία θα έμενε ίδια ή θα αυξανόταν. Αφού μεγαλώνουν οι παραγγελίες και το κάθε μέλος είναι αναγκασμένο να στείλει την ποσότητα της εισερχόμενης παραγγελίας, τα υπόλοιπα κομμάτια καταχωρούνται στο απόθεμα.

Επίσης, παρατηρείται σε όλα τα μέλη της αλυσίδας η ποσότητα των αποθεμάτων να είναι σχεδόν πάντα μεγαλύτερη από την αρχική (100) εκτός από κάποιες φορές που αδειάζει τελείως. Το ανώτερο όριο του λιανέμπορου είναι 150, του μεσολαβητή 250 και του κατασκευαστή κοντά στο 270. Παρόλ' αυτά διαπιστώνονται έντονες διακυμάνσεις κυρίως στα ακραία μέλη της αλυσίδας. Η ποσότητα στον μεσολαβητή είναι πιο σταθερή, λόγω της θέσης της. Ταυτόχρονα, δέχεται και παίρνει παραγγελίες από τα υπόλοιπα μέλη και έτσι δεν

προλαβαίνει να αδειάσει, σε αντίθεση με τον λιανέμπορο που περιμένει την παραγγελία του πελάτη και του κατασκευαστή που δεν παραλαμβάνει παραγγελίες. Η υπερφόρτωση των αποθεμάτων του μεσολαβητή θα έχει ως αποτέλεσμα την αύξηση του αντίστοιχου κόστους.

Το «bullwhip effect» είναι έντονο σε αυτή τη περίπτωση μιας και παρατηρείται μια σταθερή αύξηση του stock σε σχέση με το αρχικό και έντονες αυξομειώσεις.



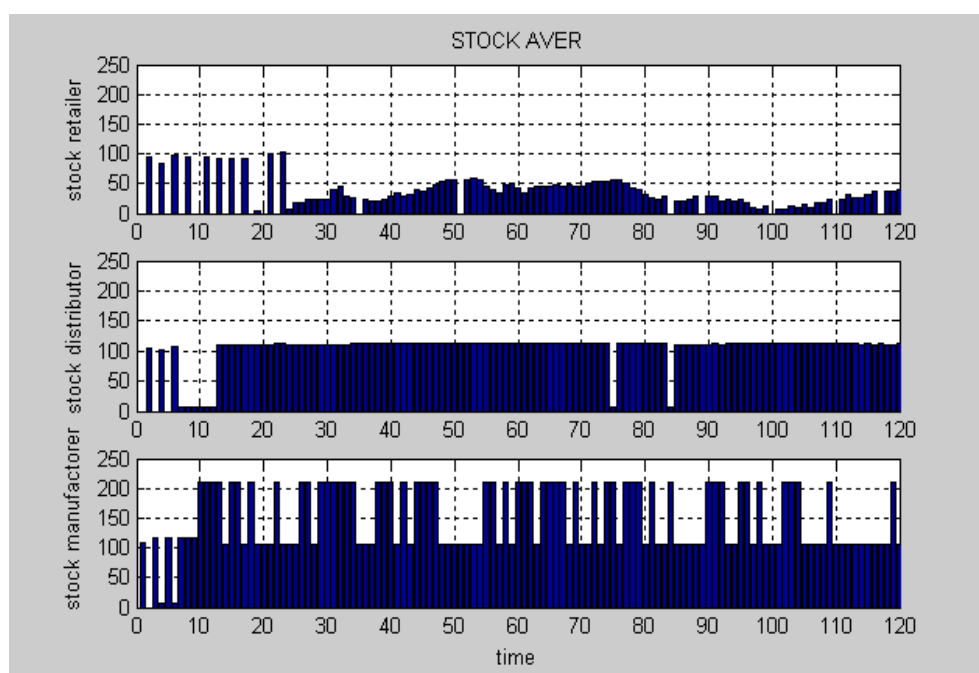
Σχήμα 49. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “aggressive ordering”.

Με την χρήση της συνάρτησης κινητού μέσου όρου, εξομαλύνονται οι επιπτώσεις του φαινομένου bullwhip, μιας και κάθε μέλος εμφανίζεται με χαμηλότερη στάθμη αποθεμάτων σε σχέση με του προηγούμενου παραδείγματος. Στον λιανέμπορο η ανώτερη στάθμη των περισσευούμενων κομματιών είναι 100 (ίδια με την αρχική) και κυρίως τις 22 πρώτες βδομάδες. Το υπόλοιπο διάστημα μπορούμε να πούμε ότι το stock κυμαινόταν από 50 και κάτω. Οι αυξομειώσεις είναι πολύ πιο σπάνιες. Αυτό που παρατηρείται πιο έντονα σε αυτή τη περίπτωση είναι ότι σε πολλές βδομάδες το απόθεμα είναι μηδενικό, κάτι που επιβαρύνει τον λιανέμπορο με επιπλέον κόστος. Σε συνδυασμό όμως με την πτώση του κόστους αποθεμάτων μπορούμε να πούμε ότι το σύστημα είναι πιο λειτουργικό από το προηγούμενο (το κόστος θα αναλυθεί και στη συνέχεια).

Ο μεσολαβητής χαρακτηρίζεται από μία σταθερότητα, μιας και μετά την 13^η βδομάδα το απόθεμα παραμένει σταθερό στα 100 εκτός από λίγες περιπτώσεις (ανώτερη τιμή 112). Παρόλο που οι αποθήκες έμεναν πριν από την 13^η βδομάδα συχνά άδειες δεν έχουμε το φαινόμενο απότομης αύξησης, όπως παραπάνω. Μπορούμε να πούμε ότι η εξομάλυνση του φαινομένου πραγματοποιήθηκε με μεγάλη επιτυχία στη θέση του μεσολαβητή.

Το απόθεμα του κατασκευαστή είναι περίπου 210 (ανώτερη τιμή), μικρότερο από το αντίστοιχο της πρώτης μεθόδου. Τις περισσότερες βδομάδες είναι λίγο μεγαλύτερο του 100 και οι αυξομειώσεις εμφανίζονται σε σταθερούς χρόνους. Κάτι τέτοιο συμβαίνει σε συστήματα που παρουσιάζουν σταθερότητα στη συχνότητα παραγγελιών και μεταφορών.

Η συνάρτηση κινητού μέσου συνέβαλε στην εξάλειψη του φαινομένου bullwhip μιας και μειώθηκε το απόθεμα και οι εναλλαγές του είναι περιοδικές.



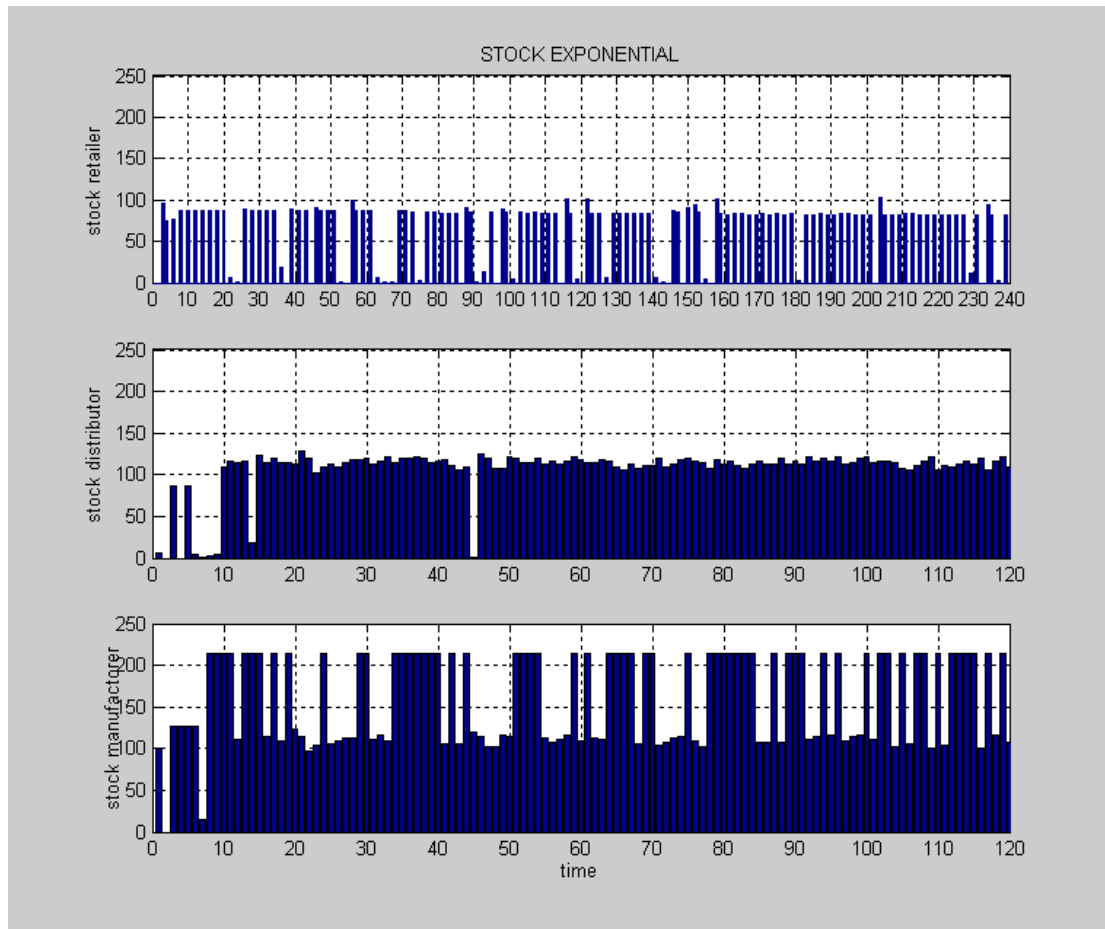
Σχήμα 50. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “moving average”.

Στην μέθοδο εκθετικής εξομάλυνσης, η διαφορά στο απόθεμα στα δύο πρώτα μέλη είναι σχετικά μικρή. Το τελευταίο μέλος παρουσιάζει μεγαλύτερη απόκλιση όπως και στις άλλες περιπτώσεις που μελετήθηκαν. Συγκριτικά με τα σχήματα 49 και 50 ο λιανέμπορος εμφανίζει τη μεγαλύτερη διαφορά. Ανά 10 βδομάδες αδειάζουν τελείως οι αποθήκες του αλλά στο επόμενο βήμα θα επανέλθει σχεδόν στην αρχική του ποσότητα. Αυτό μπορεί να επιβαρύνει το κόστος του

μιας και σε ενδεχόμενη παραγγελία θα σημειωθεί μια σχετική καθυστέρηση. Κάτι τέτοιο μπορεί να οφείλεται στη ποσότητα που παραγγέλλεται κάθε φορά από το μέλος. Όπως έχει αναλυθεί παραπάνω, η συνάρτηση εκθετικής εξομάλυνσης δίνει μεγαλύτερη βαρύτητα στις πιο πρόσφατες παραγγελίες. Όταν αδειάσει η αποθήκη, σημαίνει πως οι τελευταίες παραγγελίες είναι μεγαλύτερες από τις προηγούμενες ενώ όταν επανέρχεται ο αριθμός στο 100, οι εισερχόμενες παραγγελίες είναι μικρότερες. Οι σταθερότητα στις εναλλαγές οφείλεται στο αίτημα του πελάτη, όπου παραγγέλλει με τη βοήθεια κανονικής κατανομής.

Στο απόθεμα του μεσολαβητή διαπιστώνεται μια μικρή αστάθεια σε σχέση με την προηγούμενη μέθοδο (ανώτερη τιμή 129). Έτσι το κόστος επιβαρύνεται. Όταν χρησιμοποιούμε συνάρτηση κινητού μέσου, μετά την 13^η βδομάδα δεν αδειάζει σχεδόν ποτέ η αποθήκη. Στην περίπτωση αυτή, ο χρόνος μετατοπίζεται κατά 3 βδομάδες νωρίτερα. Αυτό μπορεί να μικραίνει το κόστος του μέλους. Ίσως σε βάθος χρόνου οι παραγγελίες έχουν μία μικρή ανοδική τάση. Επειδή με μια απλή καταγραφή του αποθέματος, δεν μπορεί να βγει ασφαλές συμπέρασμα για το πια μέθοδος αποφέρει περισσότερο κέρδος, θα χρειαστεί και η μελέτη του κόστους.

Στο σχήμα 51 παρατηρούμε ότι ο χαρακτήρας του αποθέματος του κατασκευαστή είναι παρόμοιος. Η μέγιστη τιμή είναι 214. Η απόφαση της καλύτερης πολιτικής δεν θα εξαρτηθεί από το απόθεμα του τελευταίου μέλους.



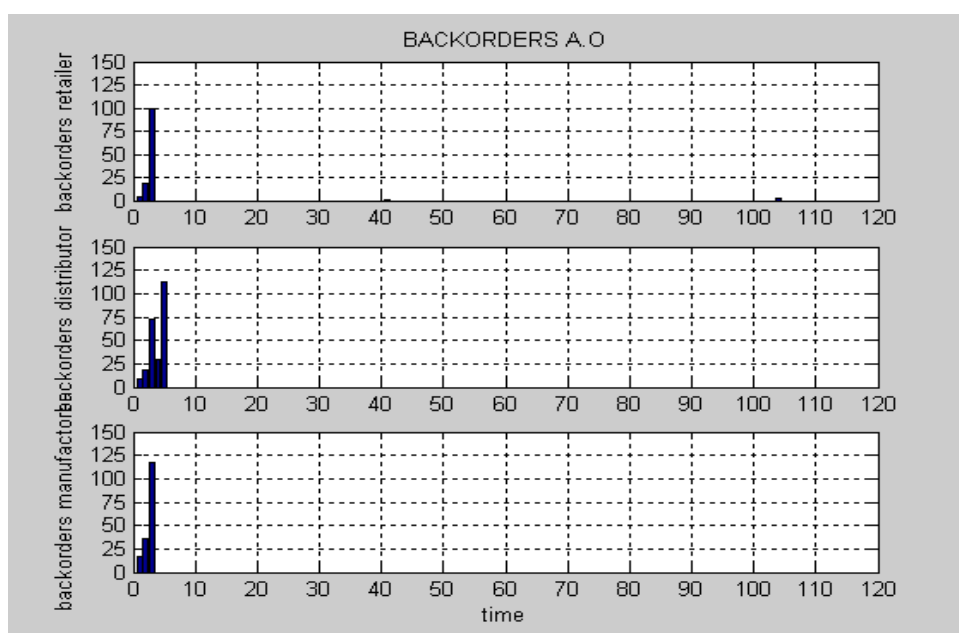
Σχήμα 51. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “exponential smoothing”.

Αν θέλουμε να μελετήσουμε το σύστημα πιο μικροσκοπικά, ανεξαρτήτως του κόστους και με κύρια παράμετρο τις διακυμάνσεις του αποθέματος, θα μπορούσαμε με ασφάλεια να πούμε ότι όσο αφορά τον μεσολαβητή, η μέθοδος «moving average» είναι αποδοτικότερη. Για τη περίπτωση των άλλων δύο μελών δεν μπορούμε να βγάλουμε ασφαλή συμπεράσματα μόνο με τη μελέτη του αποθέματος.

4.1.2 Μελέτη ποσοτήτων back orders

Όπως έχει προαναφερθεί, η ποσότητα των παραγγελιών που δεν έχουν εκπληρωθεί (back orders) επηρεάζουν άμεσα το κόστος. Γι' αυτό και η αναφορά των μη-σταλμένων παραγγελιών είναι ιδιαίτερα σημαντική. Στα σχήματα 52-54 περιγράφεται η κατάσταση αυτών των προϊόντων για το κάθε μέλος της αλυσίδας με τη βοήθεια ραβδογραμμάτων.

Γενικά, μετά το πέρασμα των 5 βδομάδων δεν υπάρχουν ελλειμματικές παραγγελίες με τη μέθοδο που χρησιμοποιήθηκε αρχικά. Η τελευταία παρατήρηση των back orders είναι λίγο μεγαλύτερη από το αρχικό απόθεμα. Αυτό συμβαίνει, διότι ο 1^{ος} κύκλος παραγγελίας πραγματοποιείται σε 3 βδομάδες. Μέχρι τότε το σύστημα δεν δέχεται άλλη παραγγελία από τον πελάτη. Αν η επόμενη παραγγελία είναι πάλι μεγαλύτερη από το απόθεμα, back orders εξακολουθούν να υπάρχουν. Το ζητούμενο είναι να μελετηθεί η κατάσταση των ελλειμματικών παραγγελιών με τις επόμενες μεθόδους που χρησιμοποιήθηκαν.



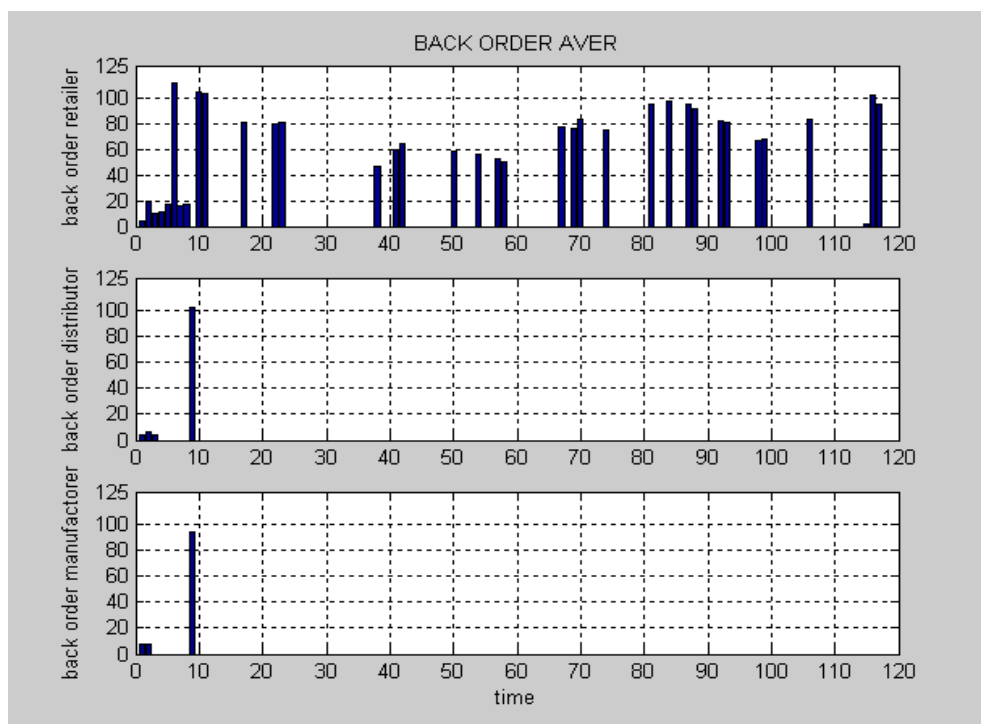
Σχήμα 52. Back orders των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “aggressive ordering”.

Το απόθεμα του λιανέμπορου του σχήματος 50, μέχρι την 22^η βδομάδα είναι περιοδικά μηδενικό. Αυτό έχει επίπτωση και στη δημιουργία ελλειμματικών παραγγελιών και μάλιστα μέχρι το τέλος της διαδικασίας. Αυτό συμβαίνει, διότι οι παραγγελίες του λιανέμπορου είναι πάντα ο μέσος όρος αυτών που έχουν προηγηθεί. Αν ο πελάτης δίνει μεγαλύτερες παραγγελίες από το μέσο όρο, η παραγγελία του λιανέμπορου προς τον μεσολαβητή θα είναι πάντα

μικρότερες από αυτή την ποσότητα. Έτσι, είναι φυσικό να δημιουργούνται back orders στη θέση του λιανέμπορου.

Αντίθετα, τα άλλα δύο μέλη όχι μόνο δεν έχουν επιπλέον ελλειμματικές παραγγελίες αλλά έχουν μειώσει την ποσότητα τους αθροιστικά. Η εικόνα των back orders των τελευταίων μελών είναι σχεδόν η ίδια. Η εξήγηση είναι απλή αν αναλογιστούμε την εικόνα των αποθεμάτων με χρήση της “moving average”(από μέρος του λιανέμπορου). Παρόμοια κατάσταση αποθεμάτων δίνει παρόμοια εικόνα στις ελλειμματικές παραγγελίες.

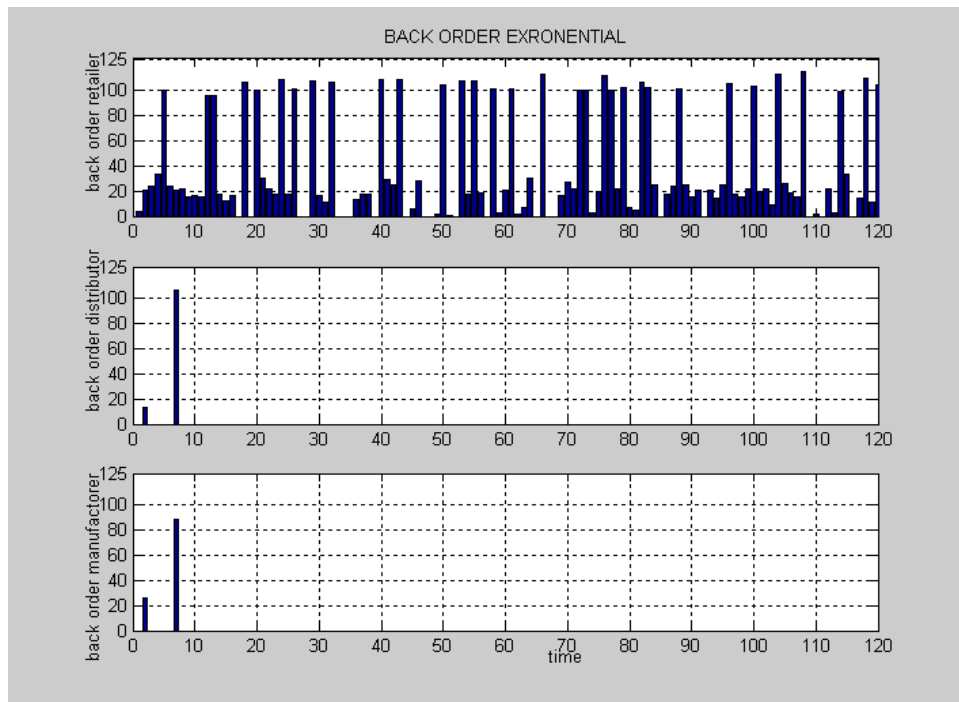
Θα μπορούσαμε να πούμε ότι με τη μέθοδο “moving average” το κόστος που αφορά την δημιουργία back orders επιβάρυνε μόνο τον λιανέμπορο, ενώ μειώθηκε κατά πολύ λίγο για τα άλλα μέλη.



Σχήμα 53. Back orders των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “moving average”.

Η κατάσταση των back orders του λιανέμπορου χειροτερεύει όταν χρησιμοποιεί συνάρτηση εκθετικής εξομάλυνσης. Η εμφάνισή τους πραγματοποιείται σε πιο τακτά χρονικά διαστήματα. Οι περισσότερες παρατηρήσεις είναι κάτω του 40, ενώ αρκετές ξεπερνούν τις 100. Η αύξηση αυτών των ποσοτήτων μπορεί να ερμηνευτεί με τη βοήθεια των αποθεμάτων του λιανέμπορου στο σχήμα 51. Το άδειασμα των αποθηκών του λιανέμπορου ανά δύο βδομάδες, συνεπάγεται και καθυστέρηση παράδοσης της παραγγελίας του πελάτη μιας και περιμένει μέχρι να του δοθεί ποσότητα από τον μεσολαβητή.

Λίγο διαφορετική εικόνα παρουσιάζουν τα άλλα δύο μέλη, μιας και το σύνολο των ελλειμματικών παραγγελιών μειώνεται ακόμα περισσότερο.



Σχήμα 54. *Back orders* των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “*exponential smoothing*”.

Γενικό συμπέρασμα για ότι αφορά τα back orders είναι ότι αυξάνονται κατά πολύ στη περίπτωση εκθετικής εξομάλυνσης και λιγότερο στην moving average.

4.1.3 Ανάλυση κόστους

Τα συμπεράσματα για την βελτιστοποίηση του μοντέλου ολοκληρώνονται με τη μελέτη του κόστους των μελών της αλυσίδας. Στα σχήματα 55-57 το κόστος υπολογίζεται αθροιστικά για τις 120 βδομάδες όπου ολοκληρώθηκε η διαδικασία ενώ στο παρακάτω πίνακα αναγράφονται οι ακριβείς τιμές του κόστους για τις τρεις μεθόδους σε χρηματικές μονάδες.

	Λιαν έμπορος	Μεσο λαβητής	Καταστ ευαστής
Aggressive ordering	387.1 35	834.6 31	765.749
Moving average	399.3 99	658.1 65	663.635
Exponential smoothing	468.9 55	664.0 04	671.408

Πίνακας 5. Το κόστος του κάθε μέλους με τη βοήθεια των τριών μεθόδων συνεχόμενης παραγγελιοληψίας.

	Λιαν έμπορος	Μεσο λαβητής	Καταστ ευαστής
Aggressive ordering	0%	0%	0%
Moving average	- 3,32%	21,14 %	13,34%
Exponential smoothing	- 21,13%	20,44 %	12,32%

Πίνακας 6. Το ποσοστό μείωσης του κόστους των μεθόδων “aver” “exp” σε σχέση με την “a.o”

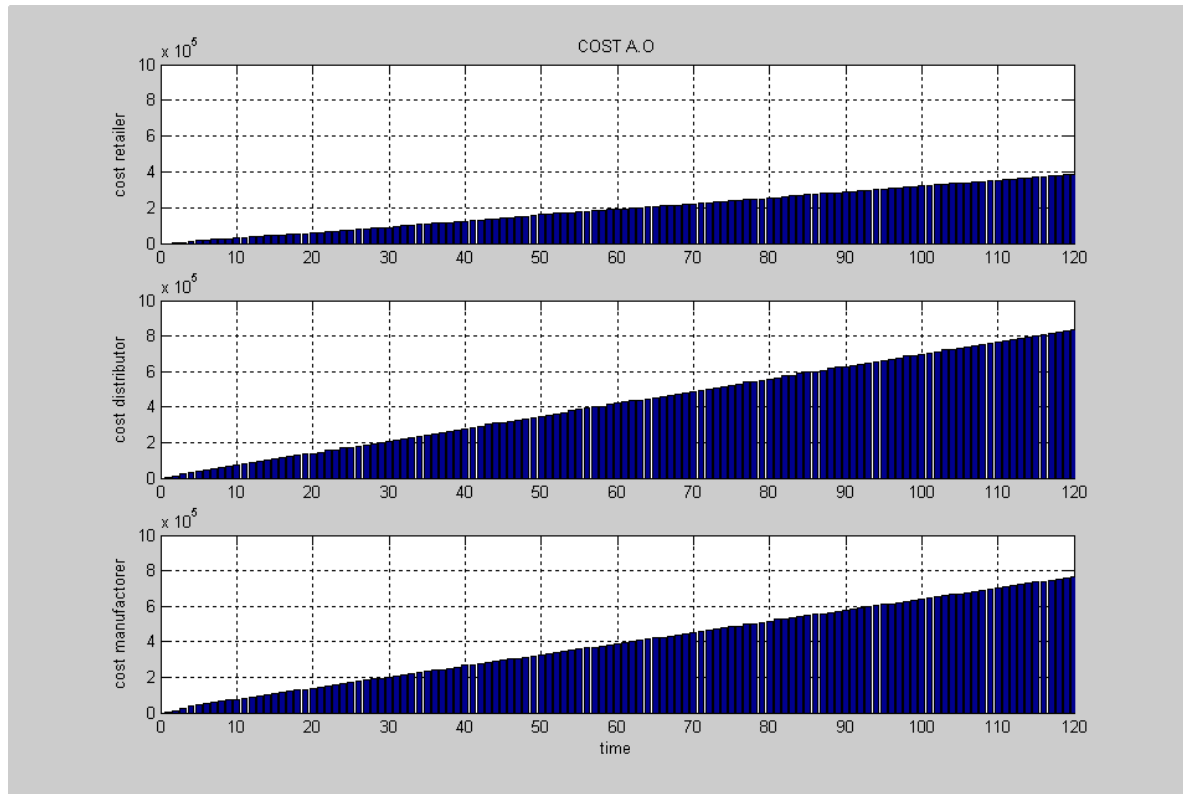
Στην πρώτη περίπτωση, όπως ήταν αναμενόμενο, το κόστος του **λιανέμπορου** είναι πολύ μικρότερο από αυτό του μεσολαβητή και του κατασκευαστή. Το συνολικό κόστος ανέρχεται μόλις στις 387.135 χρηματικές μονάδες ενώ των άλλων δύο στις 834.631 και 765.749 χ. μονάδες αντίστοιχα. Η περιορισμένη ποσότητα στο απόθεμα σε συνδυασμό με την απουσία ποσοτήτων back orders και ότι ο συγκεκριμένος δεν επιβαρύνεται από το κόστος μεταφοράς, οφείλονται για την μεγάλη διαφορά στο κόστος σε σχέση με τους υπόλοιπους.

Ενώ για τα άλλα δύο μέλη θα περιμέναμε τα νούμερα που εκφράζουν το κόστος να ήταν αντίστροφα, εντούτοις ο **μεσολαβητής** σε αυτή τη μέθοδο επιβαρύνεται περισσότερο από τον **κατασκευαστή**. Το φαινόμενο θα μπορούσε να ερμηνευτεί ως εξής:

- Εξαιτίας της αυξημένης ποσότητας backorders του μεσολαβητή σε σχέση με τον κατασκευαστή.
- Αυτό έχει σαν αποτέλεσμα στα πρώτα βήματα ο μεσολαβητής να στέλνει πολύ μικρές ποσότητες στον λιανέμπορο. Έτσι, για την ολοκλήρωση μιας παραγγελίας

να εκτελεί πολλά δρομολόγια και να ανεβαίνει το κόστος μεταφοράς με τη πάροδο των εβδομάδων.

- Το φαινόμενο bullwhip οφείλεται συνήθως σε τέτοιου είδους ανωμαλίες.



Σχήμα 55. Το κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “aggressive ordering”.

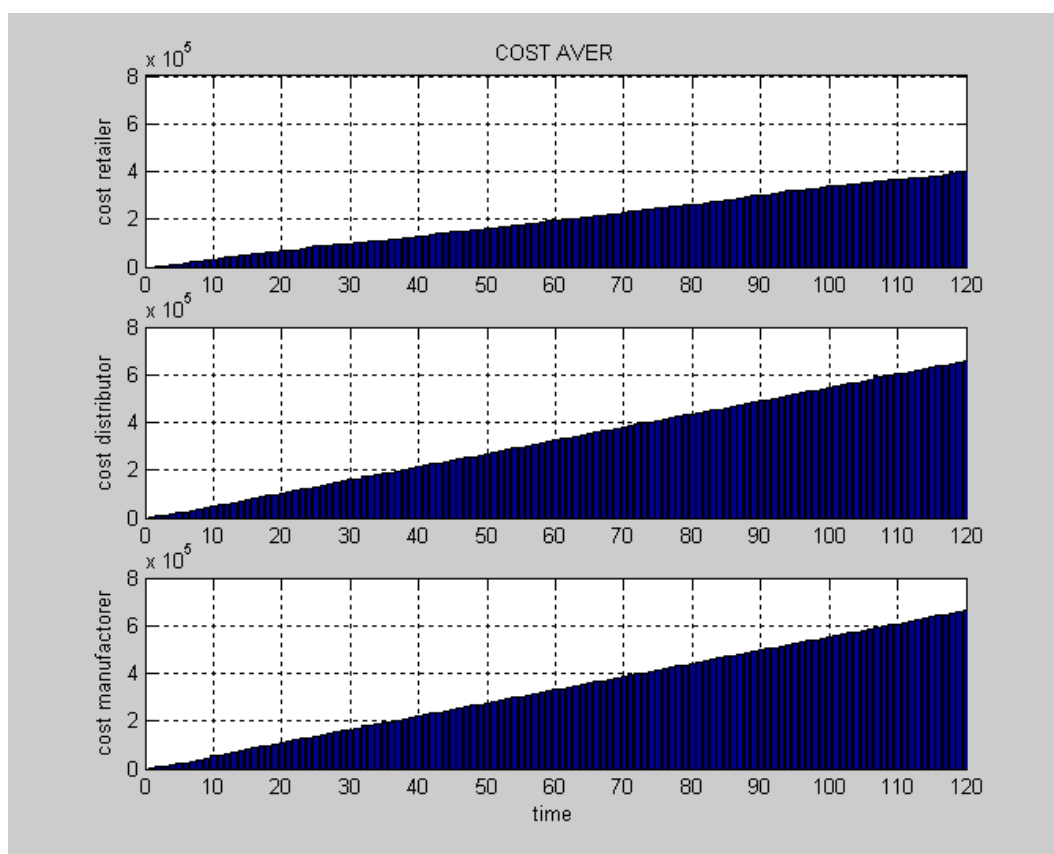
Στη συνέχεια θα εξεταστεί κατά πόσο έχει επιτευχθεί η εξομάλυνση του συστήματος με σκοπό την ελαχιστοποίηση του κόστους με τις άλλες δύο μεθόδους.

Οι χρηματικές μονάδες που καταναλώθηκαν για την διεκπεραίωση των ίδιων παραγγελιών με τη μέθοδο **κινητού μέσου** είναι 399.399 658.165 663.635 αντίστοιχα. Η μείωση του κόστους με τη βοήθεια της συγκεκριμένης συνάρτησης είναι -3,32%,21,14%,13,34% αντίστοιχα.

Μπορούμε να πούμε πως το κόστος εκ μέρους του **λιανέμπορου** αυξήθηκε κατά 3,32%. Το φαινόμενο οφείλεται στην δημιουργία δυσανάλογων ποσοτήτων back orders στη

θέση του λιανέμπορου σε σχέση με τους υπόλοιπους. Η μείωση είναι μικρή μιας και η ποσότητα στις αποθήκες ήταν σχετικά λίγη και μερικές φορές μηδενική και έτσι μειώνονταν το κόστος αποθήκευσης.

Το κόστος του **μεσολαβητή** μειώθηκε περισσότερο από ότι του **κατασκευαστή** με αποτέλεσμα το συνολικό κόστος του πρώτου (658.165 χ.μ) να είναι τώρα μικρότερο από τον δεύτερο (663.635 χ.μ.) όπως είναι και φυσικό. Μπορούμε να πούμε ότι η εξομάλυνση πέτυχε σε αρκετά μεγάλο ποσοστό. Η διαφορά αυτή αποδίδεται στο μεγαλύτερο απόθεμα που είχε δημιουργηθεί στη θέση του κατασκευαστή μιας και οι ποσότητες των ελλειμματικών παραγγελιών ήταν περίπου ίδιες.



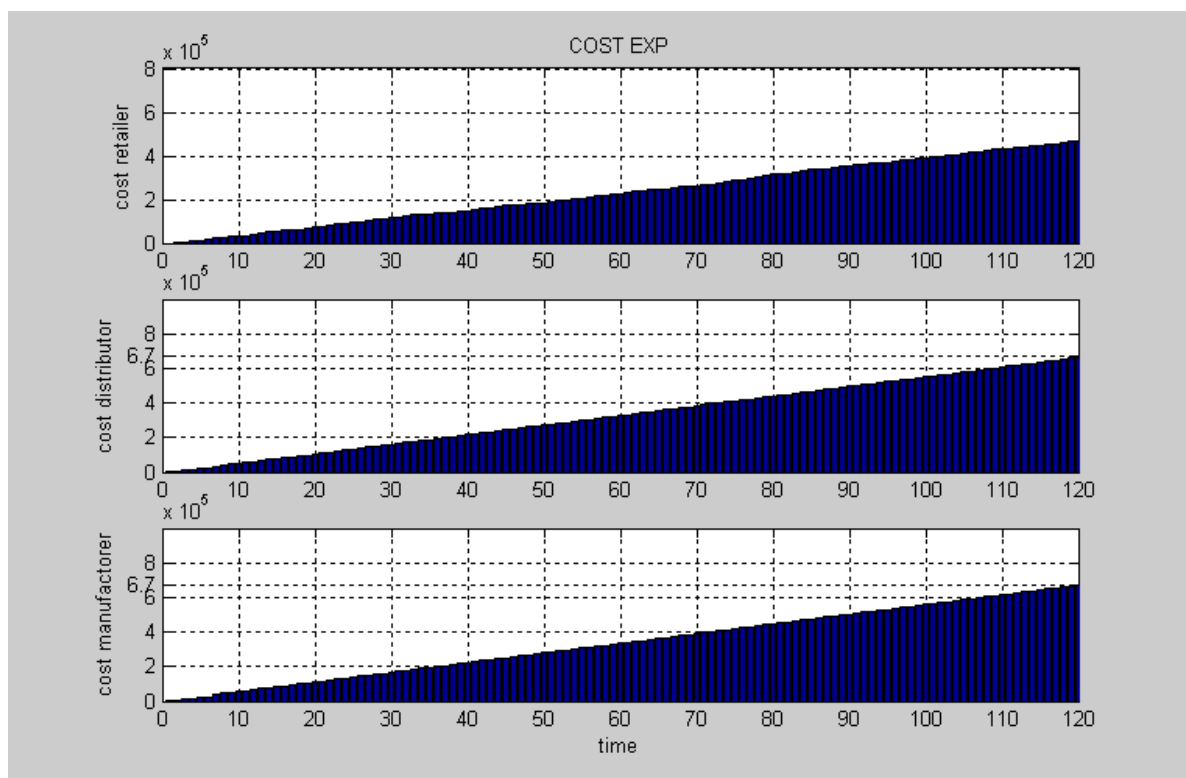
Σχήμα 56. Το κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “moving average”.

Τα μέλη της αλυσίδας με τη βοήθεια της **εκθετικής εξομάλυνσης** χρειάστηκαν συνολικά 468.955, 664.004, 671.408 χ.μ. για να ικανοποιήσουν το αίτημα του πελάτη για τις 120 βδομάδες. Παρατηρείται μια ελαφρώς ανοδική τάση του κόστους σε σχέση με την μέθοδο κινητού μέσου. Πιο συγκεκριμένα:

Το κόστος του **λιανέμπορου** αυξάνεται κατά 21,13%, μιας και οι συχνές και αυξημένες ποσότητες back orders είναι αισθητές. Για τη θέση αυτή η μέθοδος εκθετικής εξομάλυνσης κρίνεται ασύμφορη.

Παρόλ' αυτά, ο **μεσολαβητής** μείωσε το κόστος του κατά 20,44% σε σχέση με την πρώτη μέθοδο, μεγάλο ποσοστό, το οποίο οφείλεται στη μείωση του κόστους αποθεμάτων και στην διατήρηση των ελλειμματικών παραγγελιών στα ίδια επίπεδα.

Με τη σειρά του ο **κατασκευαστής** κατάφερε να μειώσει το κόστος του κατά 12,32% σε σχέση με τη πρώτη μέθοδο για τους ίδιους ακριβώς λόγους. Επίσης παρατηρούμε ότι ο κατασκευαστής έχει μεγαλύτερη επιβάρυνση από τον μεσολαβητή. Η εξομάλυνση πέτυχε σε ικανοποιητικό βαθμό.



Σχήμα 57. Το κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “*exponential smoothing*”.

Μπορούμε να βγάλουμε κάποια γενικά συμπεράσματα για το πώς αντέδρασε το σύστημα με τη χρήση των συναρτήσεων συνεχούς παραγγελιοληψίας:

- Ελαχιστοποίηση του κόστους για το σύνολο της αλυσίδας πετυχαίνουμε με τη βοήθεια συνάρτησης κινητού μέσου. Επομένως ο λιανέμπορος θα πρέπει να παραγγέλνει σύμφωνα με το μέσο όρο των προηγούμενων παραγγελιών για βέλτιστη πρόβλεψη του αιτήματος του πελάτη.
- Η δεύτερη συμφερότερη πολιτική είναι η εκθετική εξομάλυνση αν κρίνουμε από το συνολικό κόστος.
- Αν θέλουμε να μελετήσουμε το μέγιστο κέρδος για το κάθε μέλος ξεχωριστά, τότε τα συμπεράσματα δεν θα ήταν τα ίδια. Για παράδειγμα αν σκοπεύουμε στην μείωση του κόστους μόνο του λιανέμπορου, θα μέναμε στη πρώτη μέθοδο.
- Αρκετές φορές ζητείται η μείωση των back orders ή μόνο η καλύτερη διαχείριση των αποθεμάτων για να μειωθεί το αντίστοιχο κόστος. Το κόστος των back orders είναι περιορισμένο στην περίπτωση “aggressive ordering” ενώ το απόθεμα έχει πιο σταθερή και μικρή τιμή στην μέθοδο κινητού μέσου.

4.2 Στοχαστική και καθορισμένου χρόνου ανατροφοδότηση

4.2.1 Κατάσταση αποθεμάτων

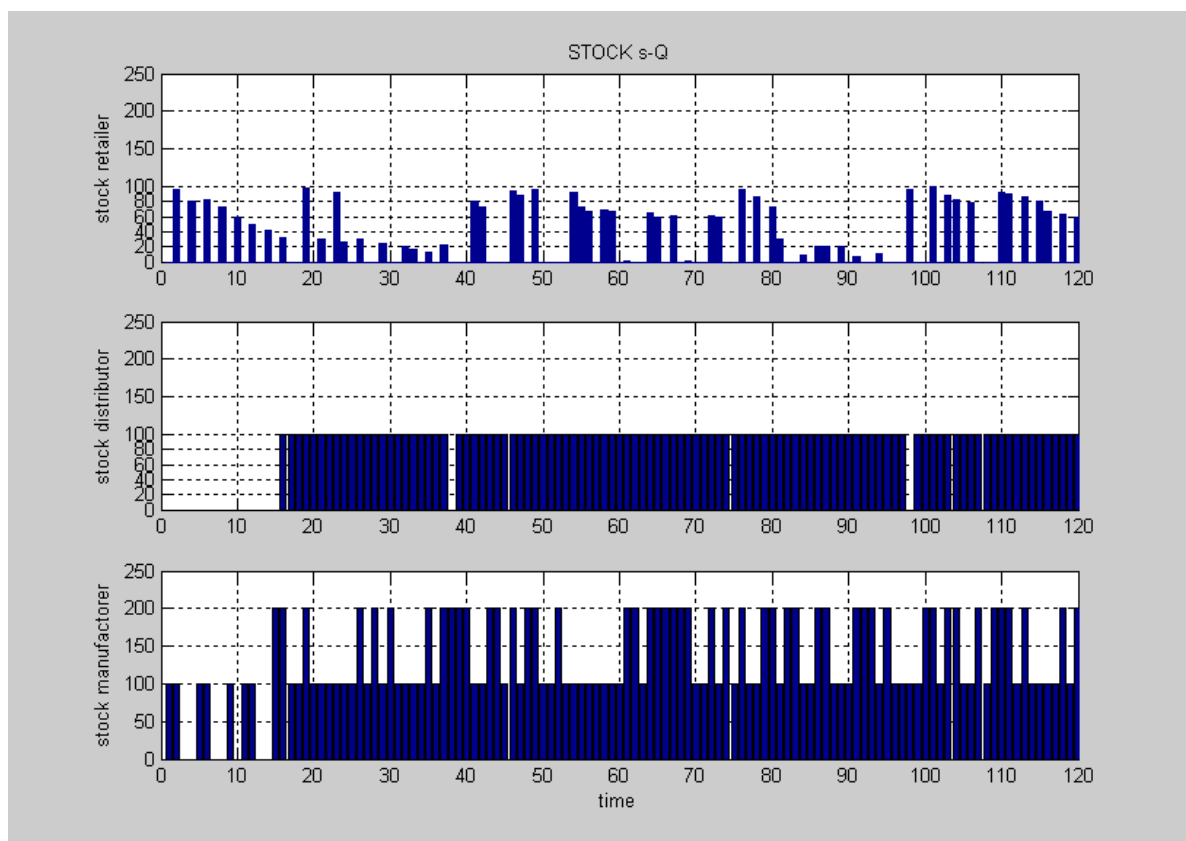
Στα επόμενα τρία ραβδογράμματα έχουν καταγραφεί τα αποθέματα των μελών όταν ο λιανέμπορος χρησιμοποιεί τις s-Q, s-S, r-S μεθόδους ανατροφοδότησης αντίστοιχα. Στην τρίτη μέθοδο θεωρήθηκε ότι $r=1$. Αυτό σημαίνει ότι ο έλεγχος των αποθεμάτων πραγματοποιείται σε

κάθε βήμα και αν είναι μικρότερο από το S θα παραγγείλει τόση ποσότητα όση να φτάσει στο S . Η παραδοχή αυτή, έγινε ώστε τα δύο είδη ανατροφοδότησης να είναι συγκρίσιμα.

Στο παρακάτω σχήμα φαίνονται οι διακυμάνσεις των αποθεμάτων χρησιμοποιώντας την $s-Q$. Οι αποθήκες του λιανέμπορου μένουν πολλές φορές άδειες ενώ η ποσότητα δεν ξεπερνάει τα 100. Αυτό συμβαίνει, διότι ανεξαρτήτου εισερχόμενης παραγγελίας, αν είναι κάτω από s ο λιανέμπορος θα παραγγείλει πάντα $Q-s=100$ στη προκειμένη περίπτωση. Έτσι, ενώ η αρχική ποσότητα στο απόθεμα είναι 100, στην πρώτη εισερχόμενη παραγγελία ίση με 104 δίνει τα 100, μένουν 4 κομμάτια backorders και αφού $0 < s = 100$ παραγγέλνει άλλα 100. Στην επόμενη παραγγελία (115) δίνει τα 100 που του έχουν έρθει, αδειάζει πάλι εντελώς οπότε παραγγέλνει πάλι 100. Τώρα τα back orders θα είναι $4+115-100=19$. Έτσι ποτέ δεν πρόκειται το απόθεμα να ξεπεράσει το 100.

Ο μεσολαβητής, πάντα θα έχει στις αποθήκες του 100 προϊόντα εκτός από λίγες φορές όπου αδειάζει τελείως. Προφανώς τότε θα έχει παραδώσει μια παραγγελία χωρίς να έχει προλάβει να παραλάβει.

Τα πλήθος των προϊόντων του κατασκευαστή είναι σταθερό σε όλη τη διάρκεια της διαδικασίας και ίσο με 100 ή 200. Αυτό συμβαίνει γιατί πάντα θα δέχεται παραγγελίες ίσες με 100 κομμάτια και ο ίδιος θα παρασκευάζει 100 προϊόντα.

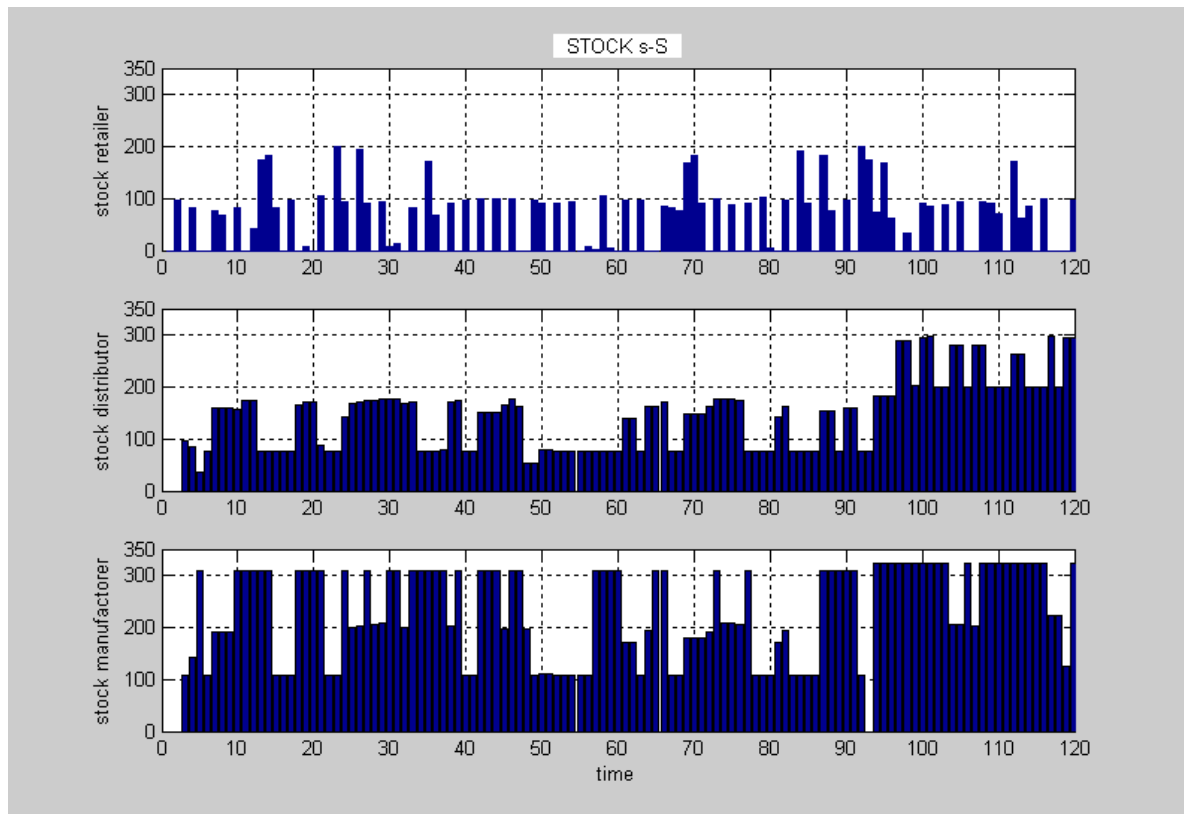


Σχήμα 58. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με χρήση της $s-Q$.

Τα διαγράμματα των αποθεμάτων τις μεθόδου $s-S$ διαφοροποιούνται κατά πολύ:

Οι αυξομειώσεις των αποθεμάτων είναι μεγαλύτερες και φτάνουν ως και 200. Καθώς ο λιανέμπορος δέχεται παραγγελία 104 δίνει τα 100 και 4 κομμάτια αναμένεται να τα παραδώσει στην επόμενη παραγγελία. Αδειάζει η αποθήκη και έτσι παραγγέλνει $S-n=200-100=100$ όπου n η ποσότητα των αποθεμάτων. Μόλις παραλαμβάνει την παραγγελία παραδίδει και τα υπόλοιπα 4 κομμάτια και μένουν 96. Στην επόμενη παραγγελία (115) επειδή $96 < 100$ παραγγέλνει $S-n=200-96=104$, τα 96 παραδίδονται και υπολείπονται $115-96=19$ κομμάτια. Όσο οι εισερχόμενες παραγγελίες μεγαλώνουν τόσο θα μεγαλώνουν και οι απαιτήσεις του λιανέμπορου προκειμένου να καλύψει τα αυξημένα back orders.

Στα επόμενα δύο μέλη διακρίνουμε υψηλά επίπεδα αποθεμάτων σε όλη τη διάρκεια της διαδικασίας που κυμαίνονται από 100 ως 300. Κάτι τέτοιο οφείλεται στις συνεχώς αυξανόμενες απαιτήσεις του πρώτου μέλους, ώστε να καλύψει τις ποσότητες back orders. Επίσης, στη περίπτωση που ο λιανέμπορος έχει άδεια αποθήκη και δεχτεί παραγγελία από τον πελάτη, τότε αναγκαστικά θα παραγγείλει $S-n=200-0=200$ και έτσι μεγαλώνει η ποσότητα παραγγελίας και άρα των αποθεμάτων.



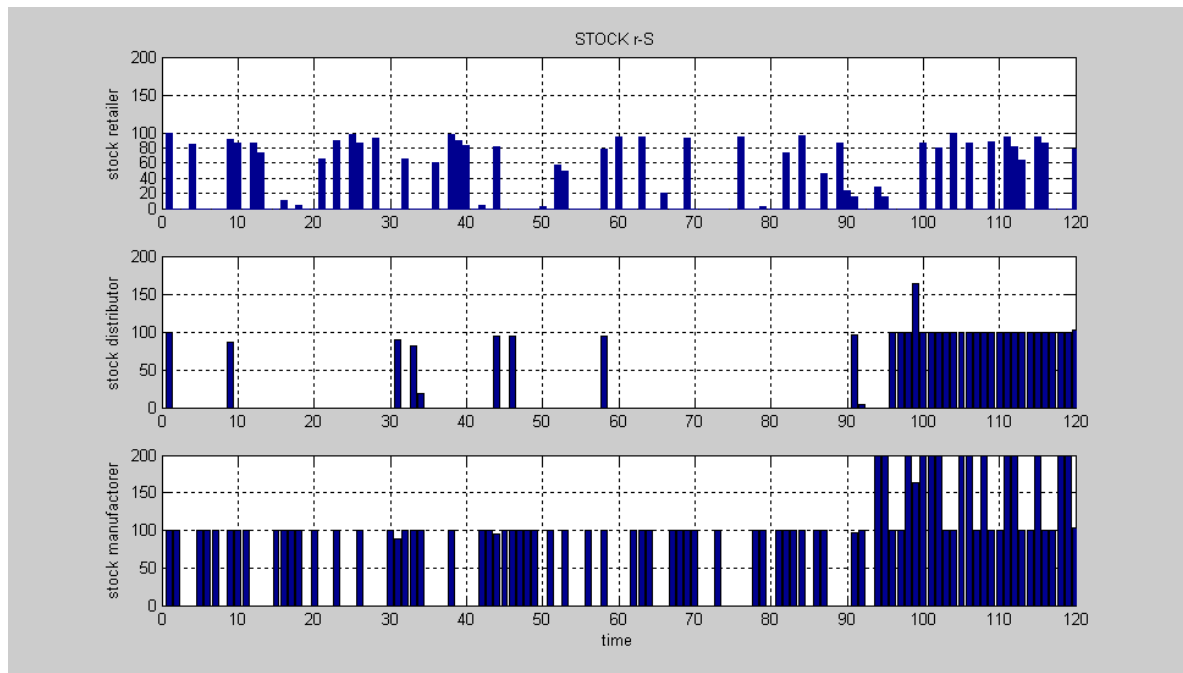
Σχήμα 59. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “s-S”.

Η εικόνα των αποθεμάτων με τη μέθοδο καθορισμένου χρόνου r-S (με $r=1$) αλλάζει αρκετά. Οι αποθήκες του λιανέμπορου είναι τις περισσότερες βδομάδες άδειες όπου συνεπάγεται την εμφάνιση μεγάλου back order. Κάτι τέτοιο συμβαίνει, διότι, για παράδειγμα στην πρώτη παραγγελία (104) ο λιανέμπορος στέλνει τα 100 και μένουν 4 ως ανεκπλήρωτη παραγγελία. Έτσι θα παραγγείλει $S-n=100-100=0$. Το back order του ανέρχεται στα 96 κομμάτια. Στο επόμενο αίτημα του πελάτη (115) ο λιανέμπορος θα παραγγείλει από τον μεσολαβητή $S-n=100-0=100$ μόλις κομμάτια, μεγάλη διαφορά με το άθροισμα της εισερχόμενης παραγγελίας και του back order που χρειάζεται για να καλύψει το αίτημα του πελάτη $115+96=216$ κομμάτια.

Αυτή η συμπεριφορά επηρεάζει και τα επόμενα μέλη μιας και δέχονται χαμηλές παραγγελίες και δεν μπορούν εύκολα να γεμίσουν το απόθεμά τους. Ο λιανέμπορος μέχρι την 95^η βδομάδα μόνο τις 10 έχει γεμάτες τις αποθήκες. Μέχρι το τέλος της 120^{ης} βδομάδας η κατάσταση σταθεροποιείται.

Ο κατασκευαστής μέχρι την 95^η βδομάδα λιγότερες φορές μένει χωρίς εμπόρευμα στις αποθήκες του με ποσότητα που δεν ξεπερνά τα 100 κομμάτια. Στις επόμενες βδομάδες η

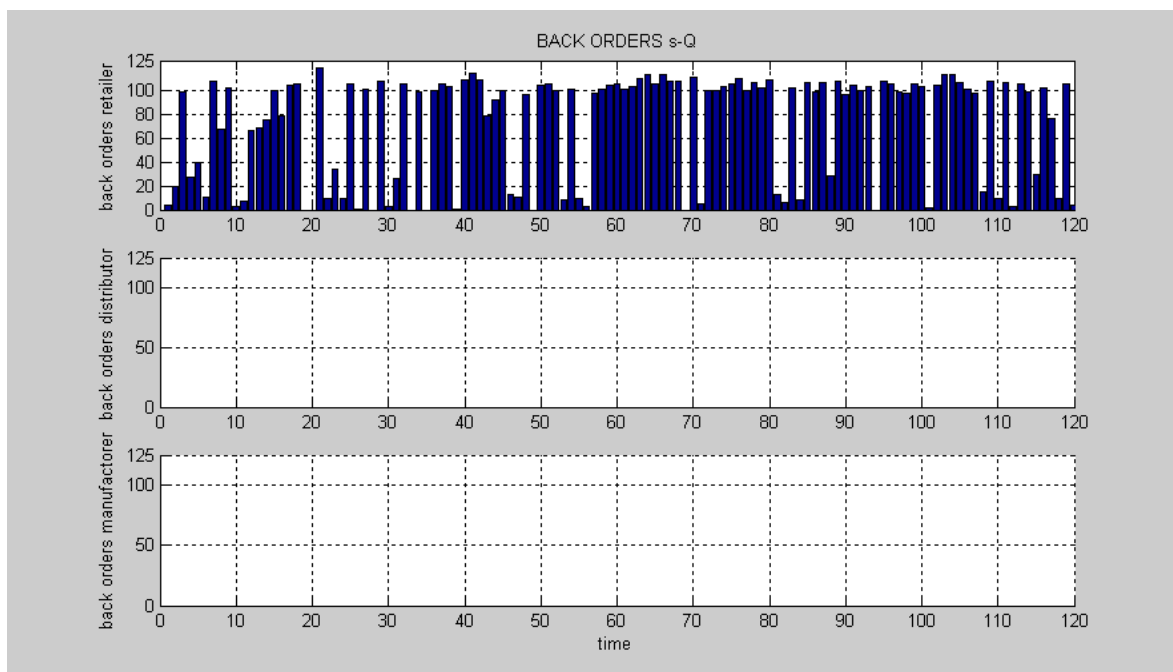
ποσότητα φτάνει μέχρι και 200, κάτι που οφείλεται στην αύξηση ζήτησης από μέρους του μεσολαβητή και έτσι κατασκευάστηκαν μεγαλύτερες ποσότητες προϊόντος.



Σχήμα 60. Τα αποθέματα των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “r-S”.

4.2.2 Μελέτη ποσοτήτων back orders

Εξαιτίας της σταθερής εικόνας που παρουσιάζουν τα αποθέματα των δύο τελευταίων μελών χρησιμοποιώντας την s-Q δεν υπάρχουν στη συγκεκριμένη εκτέλεση του προβλήματος παραγγελίες που δεν έχουν πραγματοποιηθεί. Η ζήτηση των μελών είναι σταθερή και όχι υψηλή. Αντίθετα στο πρώτο μέλος οι διακυμάνσεις των back orders είναι ιδιαίτερα έντονες. Η ζήτηση του λιανέμπορου δεν πρόκειται να ξεπεράσει τα 100 κομμάτια με αποτέλεσμα από την πρώτη κιάλα παραγγελία (104) να υπάρχει έλλειμμα στις αποθήκες. Όταν μία παραγγελία εκπληρώνεται η ποσότητα των back orders πέφτει κοντά στο 0.



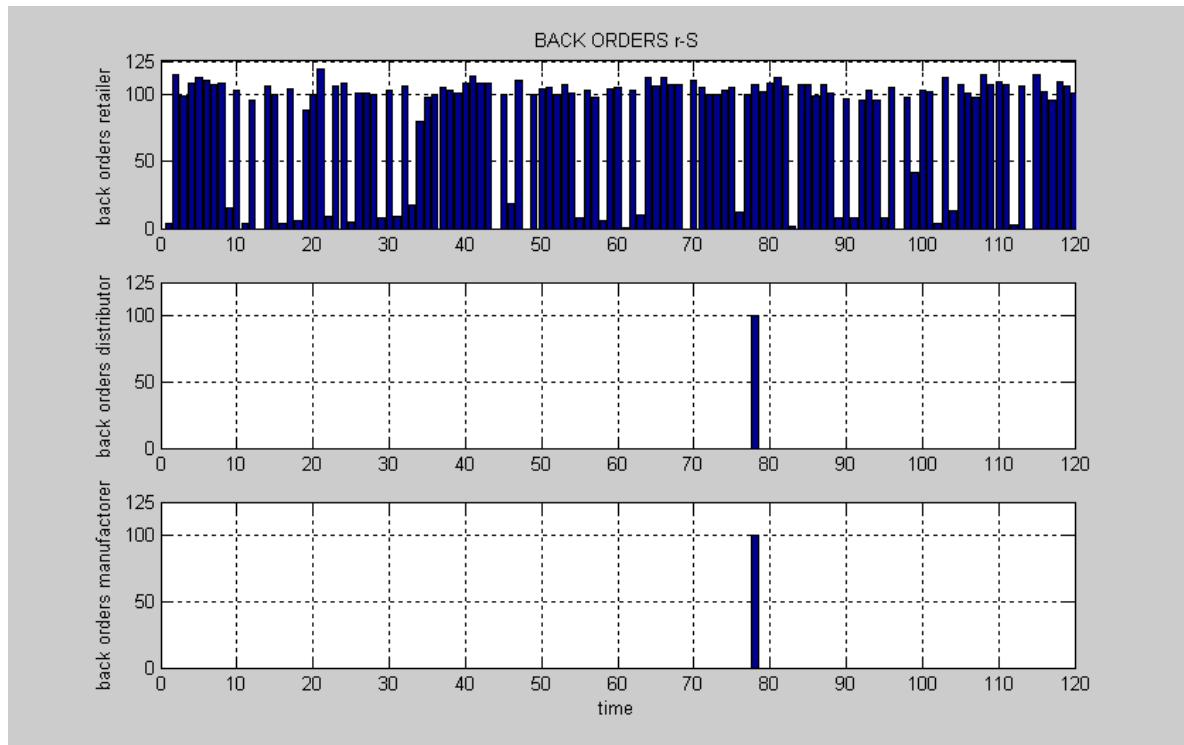
Σχήμα 61. *Back orders των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “s-Q”.*

Η ζήτηση των μελών στην s-S σίγουρα είναι μεγαλύτερη με αποτέλεσμα οι εισερχόμενες ποσότητες να μειώνουν τις ανεκπλήρωτες παραγγελίες. Ο λιανέμπορος περίπου στο 1/3 του συνολικού χρόνου δεν έχει καθόλου back orders και στα υπόλοιπα 2/3 η ποσότητα κυμαίνεται μεταξύ του 10 με 100 κομμάτια με ελάχιστες φορές να ξεφεύγει άνω του 100, όταν προηγείται ασυνήθιστα αυξημένο το αίτημα του πελάτη. Στα υπόλοιπα μέλη παρατηρούμε μία άνοδο των back orders στις 6 πρώτες εβδομάδες είτε διότι τα μέλη δεν πρόλαβαν να παραλάβουν το εμπόρευμα και δέχτηκαν άλλη παραγγελία, είτε διότι οι πρώτες παραγγελίες ήταν από τις μεγαλύτερες.



Σχήμα 62. *Back orders των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “s-S”.*

Το παρακάτω σχήμα μοιάζει με τη κατάσταση των back orders του σχήματος 62 με ελαφρώς μεγαλύτερη πυκνότητα. Αυτό σημαίνει ότι η χρήση της r-S με κριτήριο αυτού του είδους τις ποσότητες επιβαρύνει πιο πολύ από τις άλλες μεθόδους την διαδικασία της εξυπηρέτησης του πελάτη. Ο λιανέμπορος θα παραγγείλει μόνο όταν η αποθήκη θα αδειάσει με αποτέλεσμα, σχεδόν σε κάθε βήμα οι αποθήκες του να είναι άδειες προκειμένου να εξυπηρετήσει προηγούμενες παραγγελίες. Η κατάσταση των υπολοίπων είναι ίδια με του σχήματος 62 με εξαίρεση την 78^η βδομάδα, γεγονός που δεν επηρεάζει πολύ την εξέλιξη της διαδικασίας.



Σχήμα 63. *Back orders των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “r-S”.*

4.2.3 Ανάλυση κόστους

Στους πίνακες που ακολουθούν, έχει καταχωρηθεί η συνολική τιμή του κόστους μέχρι και την 120^η βδομάδα.

	Λιαν έμπορος	Μεσο λαβητής	Καταστ ευαστής
-Q	583.4 98	609.90 0	624.800
-S	385.4 44	667.29 6	686.116
-S	560.3 69	407.50 6	411.606

Πίνακας 7. Το κόστος του κάθε μέλους με τη βοήθεια των τριών μεθόδων σταδιακής παραγγελιοληψίας.

	Λιαν έμπορος	Μεσο λαβητής	Καταστ ευαστής
-Q	0%	0%	0%
-S	33,9 %	-9,4%	-9,8%
-S	3,9%	33,2%	34,1%

Πίνακας 8. Το ποσοστό μείωσης του κόστους των μεθόδων “r-S” “s-S” σε σχέση με την “s-Q”

Παρατηρούμε ότι στην **s-Q** το κόστος των μελών είναι ελαφρώς αυξημένο όσο προχωράμε στα κατώτερα στρώματα της αλυσίδας φαινόμενο που παρατηρείται στις περισσότερες μεθόδους και είναι αναμενόμενο.

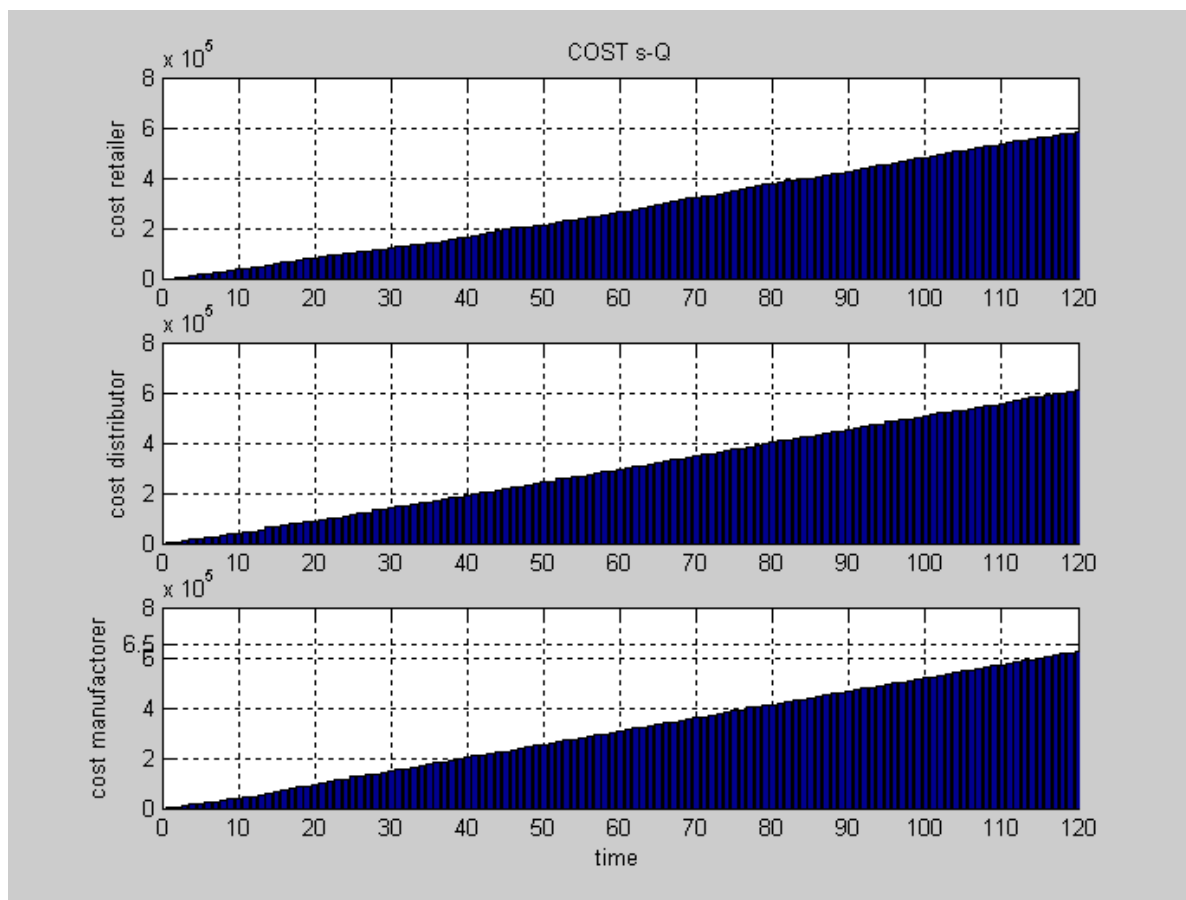
Μεγάλη διαφορά παρατηρείται στο κόστος των μελών της **s-S** όπου από το δεύτερο μέλος και κάτω σχεδόν διπλασιάζεται. Αυτό μπορεί να εξηγηθεί αν αναλογιστούμε ότι ο λιανέμπορος παραγγέλνει ποσότητα S-n κάθε φορά που η ποσότητα στις αποθήκες πέσει κάτω από 100. Έτσι ενώ ο λιανέμπορος δεν αφήνει ανεκπλήρωτες παραγγελίες, τα άλλα μέλη δέχονται υψηλές παραγγελίες με αποτέλεσμα να αυξάνεται το κόστος παραγωγής αλλά και αποθήκευσης τους.

Μια ασυνήθιστη εικόνα παρατηρείται στην επόμενη μέθοδο **r-S**. Το κόστος του λιανέμπορου το μεγαλύτερο κόστος μέλους στην αλυσίδα. Ο σημαντικότερος λόγος για τον οποίο συμβαίνει κάτι τέτοιο, είναι εξαιτίας του μεγάλου back order που δημιουργείται μιας και παραγγέλνει μόνο όταν αδειάσουν οι αποθήκες του. Η ποσότητα που παραγγέλνει είναι η r-n σχεδόν 100 λιγότερα κομμάτια από την s-S. Έτσι τα υπόλοιπα μέλη δεν καταναλώνουν χρηματικές μονάδες για μεταφορά μεγάλου εμπορεύματος ούτε πρώτη ύλη, αλλά ούτε και δημιουργείται back orders με αποτέλεσμα το κόστος τους να είναι πολύ μικρό.

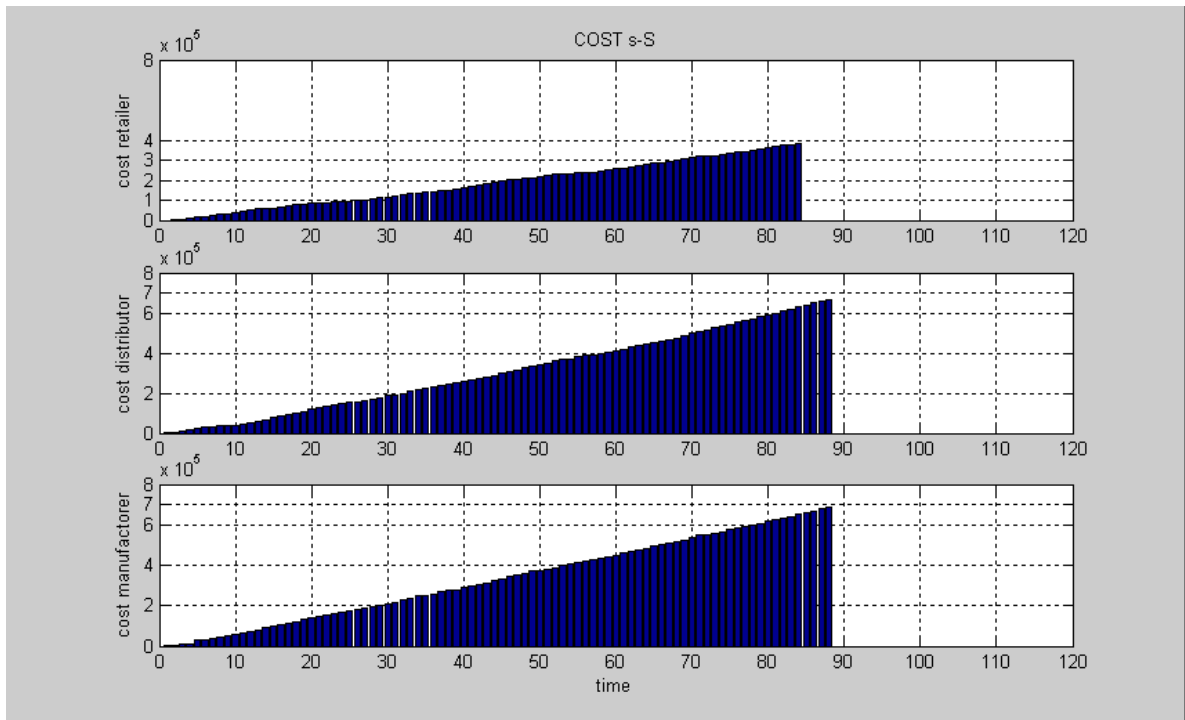
Η πιο συμφέρουσα πολιτική από άποψη κόστους όσο αφορά τον **λιανέμπορο**, σίγουρα είναι η s-S μιας και μειώνει το κόστος κατά 33,9% έναντι του 3,9% όταν χρησιμοποιούμε την r-S.

Τέτοια μεγάλη διαφορά στη μείωση του κόστους οφείλεται τόσο στις ομαλές μεταβολές των back orders τόσο και στις ιδιαίτερα χαμηλές τιμές του. Οι αποθήκες του σπάνια μένουν άδειες στην s-S μιας και δεν περιμένει να αδειάσει για να παραγγείλει όπως στην r-S και δεν παραγγέλνει πάντα σταθερή ποσότητα s-Q αλλά όση χρειάζεται για να μην πέσει κάτω από τα $s=100$ η αρχική ποσότητα των αποθεμάτων. Οι παραγγελίες εκτελούνται πιο γρήγορα και δεν υπάρχει χρονοκαθυστέρηση.

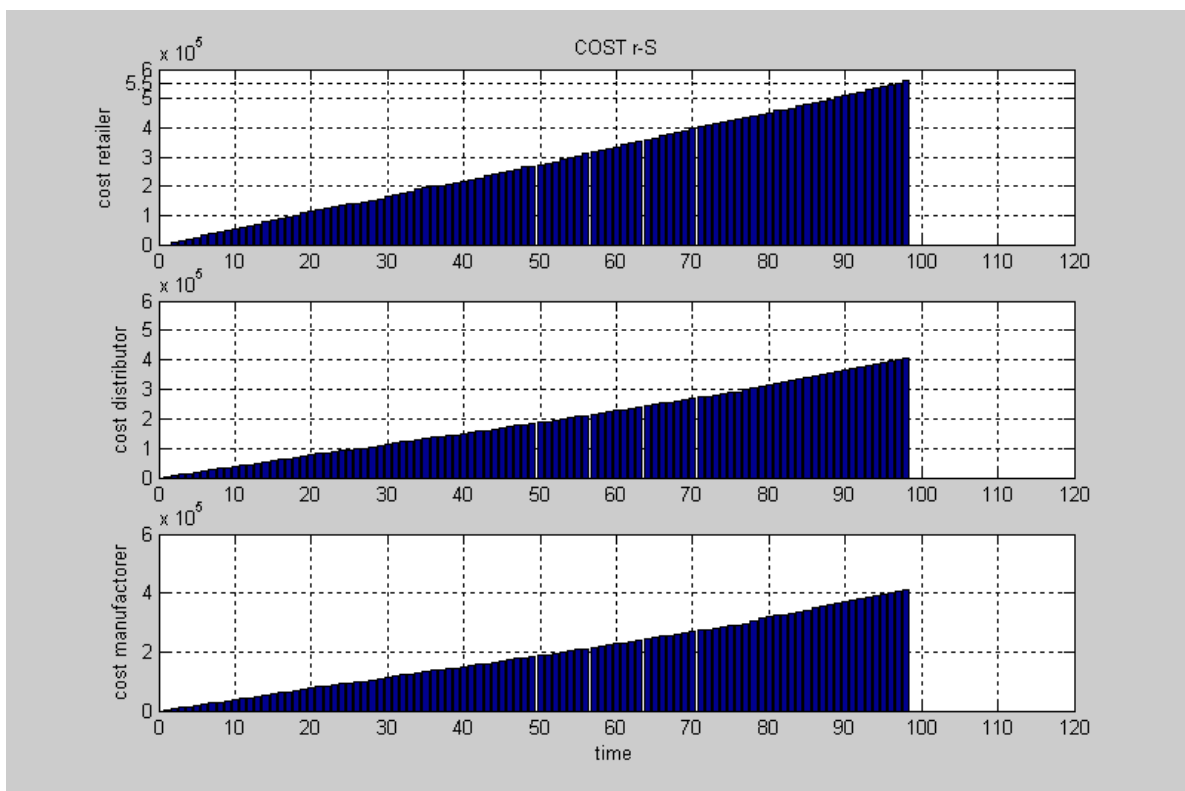
Κατά 33,2% μειωμένη είναι η τιμή του κόστους για τον **μεσολαβητή** στην 3^η μέθοδο. Οι παραγγελίες που δέχεται είναι περιορισμένες και το απόθεμα συνήθως είναι αρκετό. Κάτι που παίζει σημαντικό ρόλο σε αυτή τη διαφορά, είναι ότι ο λιανέμπορος παραγγέλνει μόνο με άδειες αποθήκες, άρα σε αραιότερα χρονικά διαστήματα. Ίδιο φαινόμενο παρατηρείται και στον **κατασκευαστή**.



Σχήμα 64: Το κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “s-Q”.



Σχήμα 65: Το κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “s-S”.



Σχήμα 66: Κόστος των τριών μελών της αλυσίδας στη περίπτωση όπου ο λιανέμπορος δίνει παραγγελίες με την τεχνική “r-S”.

Στη συνέχεια διατυπώνονται χρήσιμα συμπεράσματα για ότι αφορά τις 3 μεθόδους κατά συνθήκη παραγγελιοληψίας:

- Από άποψη κόστους πιο συμφέρουσα επιλογή στο σύνολο της αλυσίδας κρίνεται η r-S μιας και τα μέλη της υποχρεούνται να παραγγέλνουν σε μεγαλύτερα χρονικά διαστήματα. Οι έλεγχοι θα είναι πιο σπάνιοι και πιο εύκολοι μιας και δεν υπάρχει στάθμη ανεφοδιασμού και γι' αυτό ούτε καταμέτρηση. Το τίμημα αυτής της επιλογής είναι η καθυστέρηση της παραλαβής του εμπορεύματος από μεριάς του πελάτη μιας και αρκετές φορές αδειάζουν οι αποθήκες και δεν υπάρχει διαθέσιμο εμπόρευμα. Αν προσαρμόζαμε το μοντέλο ώστε να μειώνεται η απόδοσή του με τις καθυστερήσεις των παραγγελιών σίγουρα η s-S θα ήταν η κατάλληλη επιλογή.

- Αν το κόστος των ανεκπλήρωτων παραγγελιών ήταν μεγαλύτερο τότε η καλύτερη επιλογή θα ήταν η s-S. Σε αυτή τη περίπτωση τις περισσότερες φορές υπάρχουν διαθέσιμα κομμάτια στο απόθεμα και ο πελάτης έχει στη διάθεσή του συντομότερα μέρος της παραγγελίας.

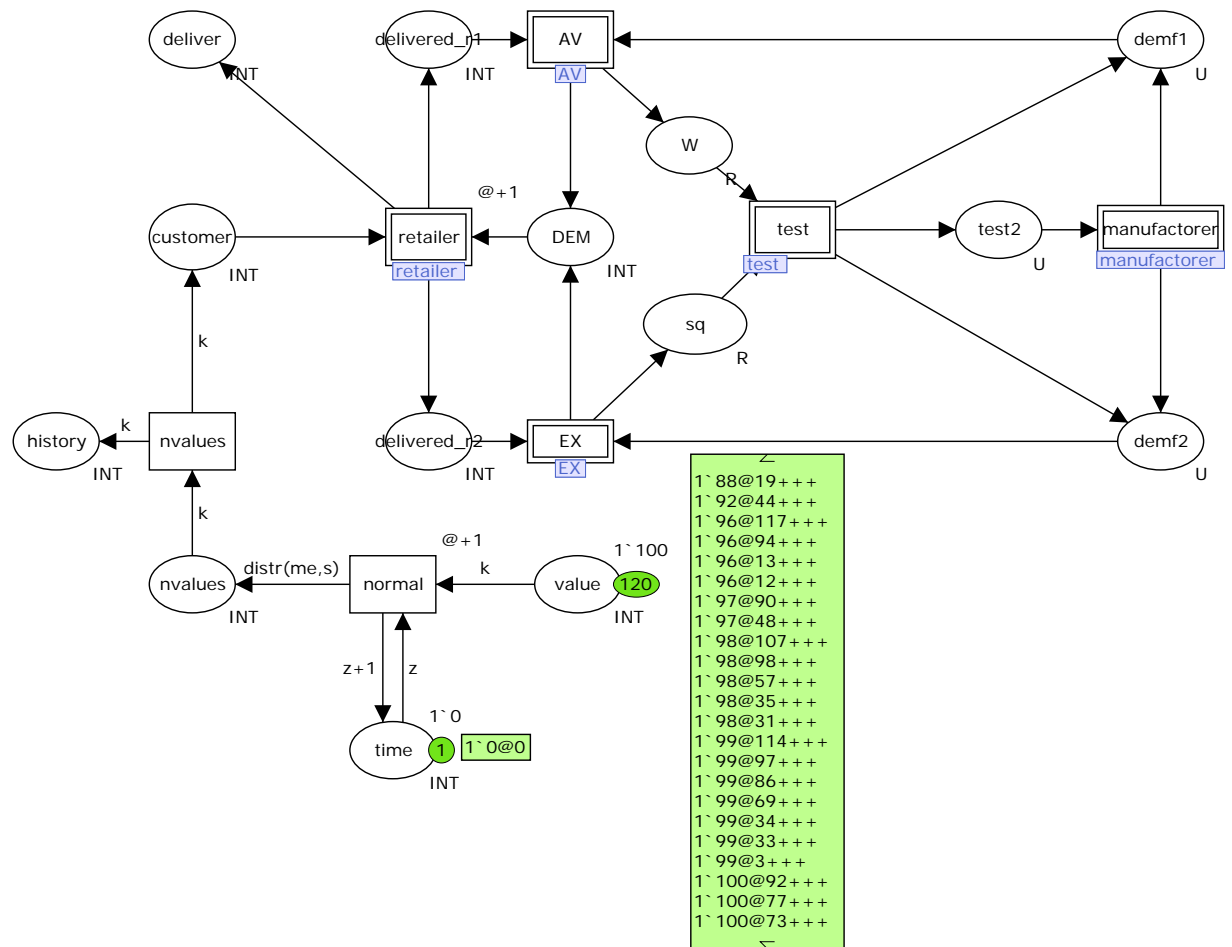
ΚΕΦΑΛΑΙΟ 5

Ανταγωνιστικό μοντέλο

Στο προηγούμενο κεφάλαιο, στη συνεχή παραγγελιοληψία, στη προσπάθεια να μειωθεί το κόστος της διαδικασίας δημιουργήθηκαν καινούρια δίκτυα τα οποία εκτελούσαν την παραγγελία με τη βοήθεια συναρτήσεων κινητού μέσου και εκθετικής εξομάλυνσης. Τα συγκεκριμένα δίκτυα μελετήθηκαν ξεχωριστά με τις ίδιες πάντα εισερχόμενες παραγγελίες. Σε αυτό το κεφάλαιο τα δίκτυα αυτά συμμετέχουν στο ίδιο μοντέλο. Έτσι θα προσπαθήσουμε να ελέγξουμε τυχόν αλληλεπιδράσεις μεταξύ τους και τελικώς ποια πολιτική είναι η πιο συμφέρουσα. Τέτοιου είδους μοντέλα ονομάζονται ανταγωνιστικά.

5.1 Σχεδιασμός ανταγωνιστικού μοντέλου

Στο σχήμα 67 φαίνεται η αρχική σελίδα του ανταγωνιστικού μοντέλου. Σε αυτή τη περίπτωση η κατάσταση του λιανέμπορου επικοινωνεί με δύο μεσολαβητές (distributors) τους ανταγωνιστές “av”, “ex” οι οποίοι χρησιμοποιούν τις συναρτήσεις παραγγελίας “aver” “exponential smoothing” αντίστοιχα. Οι δύο καταστάσεις συνδέονται με το place “test”, στο οποίο δίνονται τα ζητούμενα κριτήρια ώστε να γίνει η επιλογή της καλύτερης προσφοράς. Στη συνέχεια μεταφέρεται εντολή στον κατασκευαστή για την παραγγελία του «νικητή» ο οποίος με τη σειρά του εφοδιάζεται με την απαιτούμενη ποσότητα. Μετά από επεξεργασία των ποσοτήτων που έλαβε και των αποθεμάτων, στέλνει στον retailer την ποσότητα που είχε ενημερώσει ότι θα παραδώσει.



Σχήμα 67: Η αρχική σελίδα του ανταγωνιστικού μοντέλου. Στη κατάσταση value είναι ενσωματωμένες οι αρχικές τιμές.

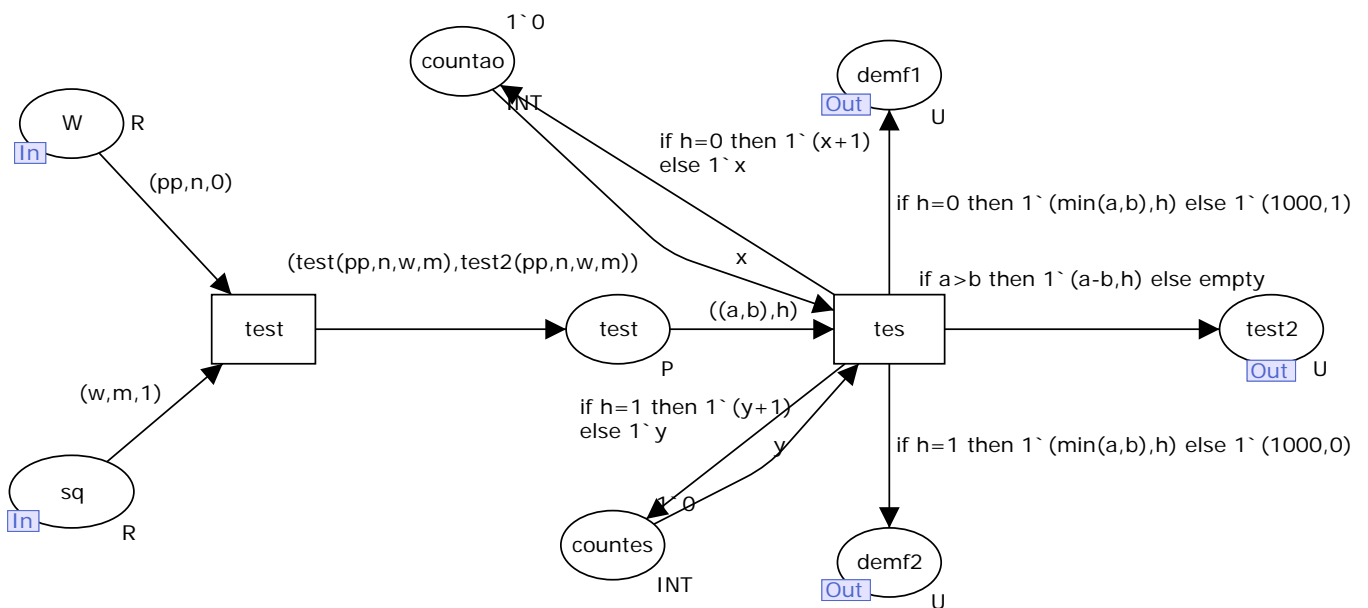
Στο δίκτυο AV (σχήμα 68) χρησιμοποιούνται δύο μετρητές update και counter. Με τη βοήθεια του update καταχωρούνται τα προηγούμενα αιτήματα του retailer και αθροίζονται στη μεταβλητή $d=p+t$. Ο counter αριθμεί το πλήθος των εισερχόμενων παραγγελιών. Η συνάρτηση aver υπολογίζει το πηλίκο αυτών.

Στο δίκτυο EX (σχήμα 69) χρησιμοποιείται ένας μετρητής παρόμοιος με τον counter.

Με τη βοήθεια των συναρτήσεων «aver» και «exp» ανακοινώνονται οι ποσότητες που διατίθεται να στείλουν για το εκάστοτε αίτημα του λιανέμπορου. Οι εξερχόμενες ποσότητες εισάγονται στο δίκτυο test μαζί με τα αποθέματά τους. Στη μετάβαση test γίνεται η επεξεργασία των δεδομένων και ανάλογα με τα κριτήρια που έχει ορίσει ο χειριστής, επιλέγεται ο κατάλληλος μεσολαβητής.

Στο συγκεκριμένο πρόβλημα ο νικητής θα είναι αυτός ο οποίος θα μπορέσει στο συντομότερο χρονικό διάστημα να στείλει και την μεγαλύτερη ποσότητα. Πρακτικά αυτό σημαίνει, πως ο μεσολαβητής που θα προτιμηθεί θα είναι αυτός με τη μεγαλύτερη ποσότητα στις αποθήκες του. Σε περίπτωση που έχουν ακριβώς ίδια ποσότητα επιλέγεται αυτός με την μεγαλύτερη εξερχόμενη παραγγελία.

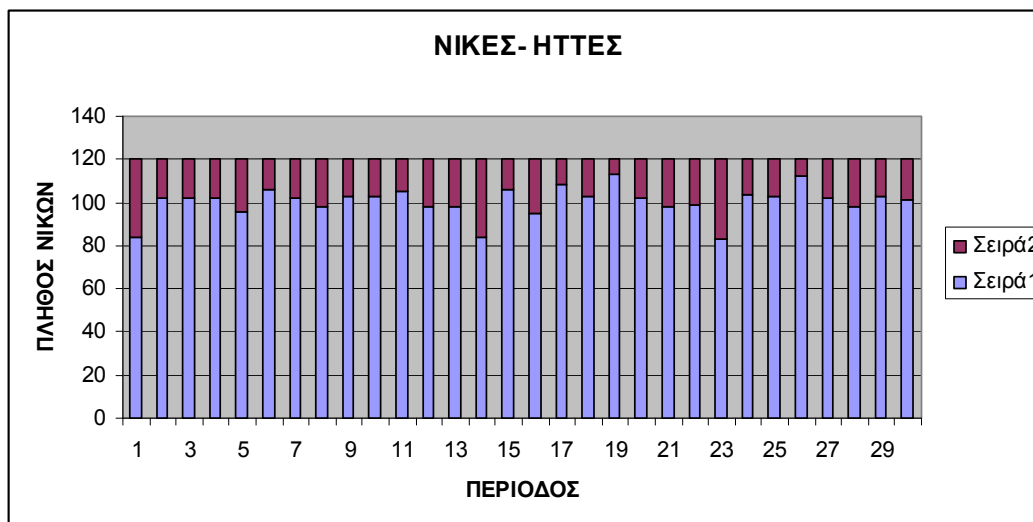
Οι καταστάσεις countao και countes μετράνε τις «νίκες» των μελών.



Σχήμα 70: Το δίκτυο test.

Το πείραμα εκτελέστηκε 30 φορές. Στη συνέχεια καταγράφονται οι φορές που επιλέγεται το κάθε μέλος κατά την διάρκεια των 120 βδομάδων (AV-EXP):

84-36, 102-18, 102-18, 102-18, 96-24, 106-14, 102-18, 98-22, 103-17, 103-17, 105-15, 98-22, 98-22, 84-36, 106-14, 95-25, 108-12, 103-17, 113-7, 102-18, 98-22, 99-21, 83-37, 104-16, 103-17, 112-8, 102-18, 98-22, 103-17, 101-19.



Σχήμα 71: Πίνακες νικών και διαφορών των παραγγελιών για τις 30 περιόδους.

Παρατηρούμε ότι σε όλες τις δοκιμές ο λιανέμπορος επιλέγει τον μεσολαβητή που χρησιμοποιεί τη συνάρτηση παραγγελίας “moving average” αφού σε κάθε εισερχόμενη παραγγελία, το μέγεθος των αποθεμάτων του είναι μεγαλύτερο από το αντίστοιχο του δεύτερου μέλους. Το φαινόμενο αυτό οφείλεται κυρίως στο ανεβασμένο επίπεδο των εκτιμώμενων εξερχόμενων τιμών στις παραγγελίες που δίνει ο «AV». Όταν τα δεδομένα περνάνε από τον έλεγχο, επιλέγεται το μέλος με τη μεγαλύτερη διαθέσιμη ποσότητα, δηλαδή το μεγαλύτερο απόθεμα. Στις πρώτες προσφορές των μελών όπου τα αποθέματα είναι σχεδόν ίδια, επιλέγεται αυτός με τη μεγαλύτερη προσφορά. Αυτό έχει ως αποτέλεσμα ο κατασκευαστής να στείλει περισσότερα κομμάτια στον επιλεγόμενο παίχτη, ο οποίος με τη σειρά του τα προϊόντα που

περισσεύουν τα τοποθετεί στο απόθεμά του. Έτσι αυξάνεται το επίπεδο των αποθεμάτων του δημιουργώντας πλεονέκτημα στο επόμενο αίτημα του λιανέμπορου.

Στο σχήμα 71 καταγράφονται οι διαφορές των παραγγελιών ανάμεσα στους ανταγωνιστές. Ο μεσολαβητής «aver» επιλέγεται 40 με 103 φορές περισσότερες από τον αντίπαλό του.

Το συμπέρασμα στο οποίο καταλήξαμε με χρήση ανταγωνιστικού μοντέλου επιβεβαιώνει τα αποτελέσματα των δοκιμών σε ξεχωριστά δίκτυα. Ενώ το κόστος του λιανέμπορου αυξάνεται κατά 3% περίπου με την συνάρτηση «aver», με χρήση της «exp» αυξάνεται κατακόρυφα κατά 21% περίπου.

Η επιλογή του κατάλληλου τρόπου παραγγελιών έχοντας ως κριτήριο την μεγαλύτερη διαθεσιμότητα, δεν περιγράφει το μοντέλο με μεγάλη ακρίβεια, για τον λόγο ότι το μεγάλο απόθεμα αυξάνει το κόστος αποθέματος αλλά και το κόστος μεταφοράς και άρα το συνολικό. Στο σημείο αυτό κρίνεται αναγκαία η δημιουργία κατάλληλου συστήματος το οποίο θα επιλέγει τον νικητή με τη βοήθεια συνάρτησης κόστους.

5.2 Ανταγωνιστικό μοντέλο με συνάρτηση κόστους

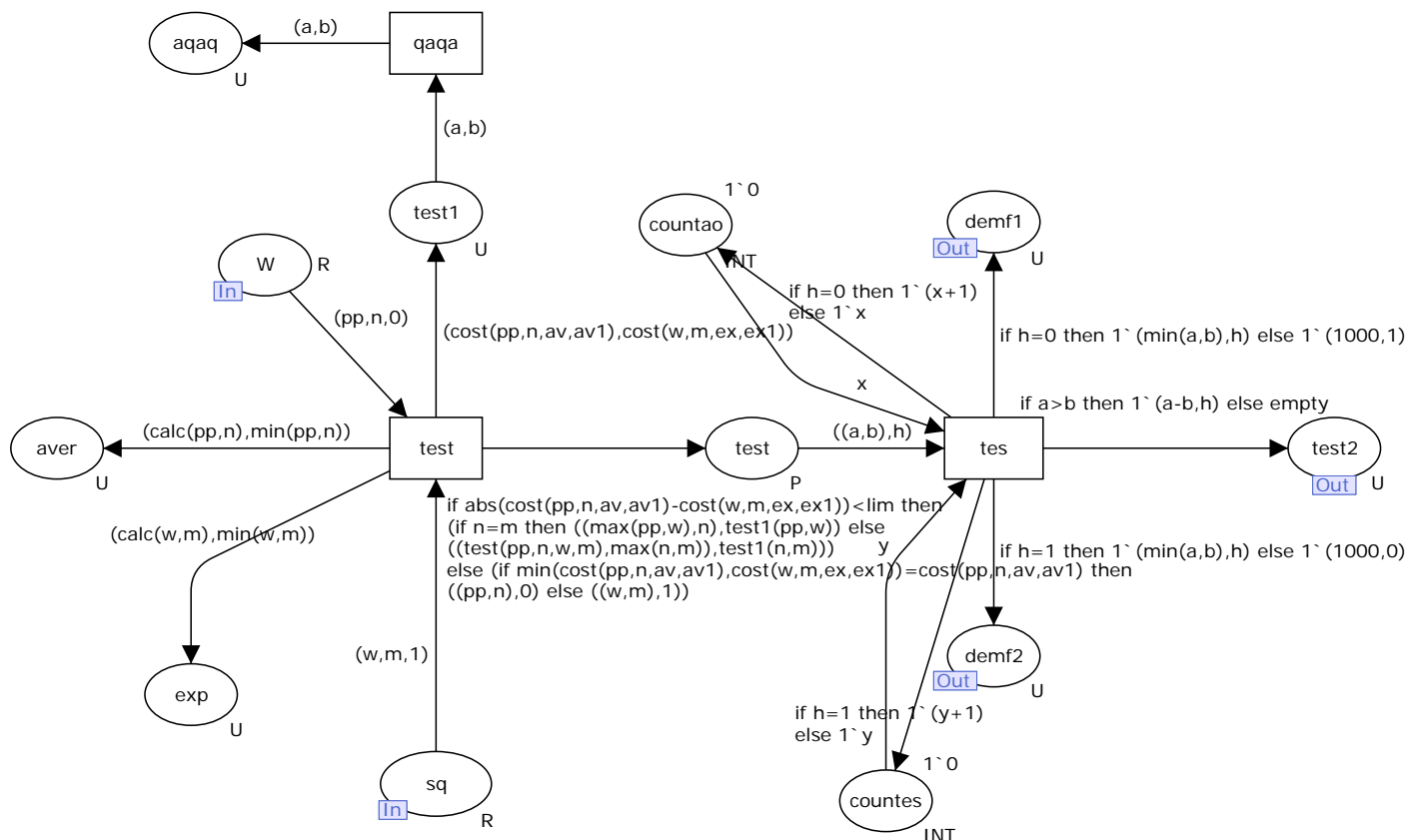
5.2.1 Εισαγωγή της συνάρτησης κόστους

Στη παράγραφο αυτή τροποποιείται το προηγούμενο μοντέλο, ώστε στα κριτήρια απόφασης να συμπεριλαμβάνεται και αυτό του κόστους.

Επειδή ο όγκος του αποθέματος είναι καθοριστικός για την τελική επιλογή, στο καινούριο μοντέλο τα κόστη μεταφοράς, παραγγελίας και backorder υπολογίζονται σε μία μεταβλητή ενώ το κόστος αποθεμάτων σε μία δεύτερη για χάρη απλούστευσης του δικτύου. Από τις προσομοιώσεις των προηγούμενων μοντέλων κόστους, υπολογίστηκε προσεγγιστικά ότι το κόστος της ποσότητας που θα αποστείλει ο μεσολαβητής (μεταβλητή k ανά τεμάχιο) και το

κόστος των κομματιών που μπορεί να χρειαστεί να παραγγείλει (μεταβλητή l ανά τεμάχιο) υπολογίζεται από την γραμμική συνάρτηση

$$\text{fun cost}(i,j,k,l)=l*\text{calc}(i,j)+k*\text{min}(i,j);$$



Σχήμα 72: Τροποποιημένο το δίκτυο test ώστε να επιλέγει τον «νικητή» με τη συνάρτηση κόστους.

Για να βρεθεί η τιμή της l συνυπολογίστηκε το κόστος μεταφοράς, παραγγελίας και back order με συντελεστές σημαντικότητας 9, 20 και 50 αντίστοιχα. Ύστερα από 120 προσομοιώσεις του μοντέλου με τα ίδια δεδομένα, βρέθηκε πως για $l=30$ (κοντά στο μέσο όρο των παραπάνω συντελεστών) το κόστος μεταφοράς, παραγγελίας και ελλειμματικών παραγγελιών των μεσολαβητών στο ανταγωνιστικό μοντέλο πλησιάζει κατά πολύ το αντίστοιχο κόστος στα ανεξάρτητα μοντέλα. Ομοίως και για το κόστος αποθεμάτων, η μεταβλητή k παίρνει την τιμή 12.

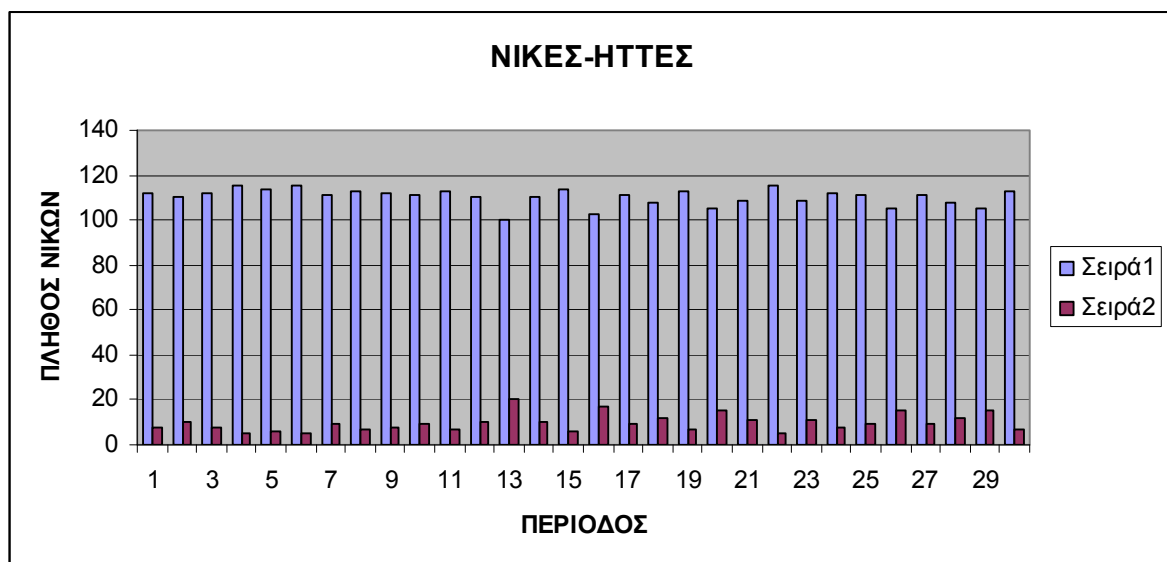
Η ποσότητα που θα στείλει ο μεσολαβητής στον λιανέμπορο από τη στιγμή που θα επιλεγεί ($\min(i,j)$), θα είναι ίση με το απόθεμά του αν το αίτημα του λιανέμπορου ήταν μεγαλύτερο από αυτό ή ίσο με το αίτημα αυτό αν δεν ξεπερνά το μέγεθος του αποθέματος.

Η ποσότητα που θα παραγγείλει από τον κατασκευαστή ($\text{calc}(i,j)$) θα είναι ίση με τη διαφορά της παραγγελίας του retailer και του αποθέματος, εφόσον είναι μεγαλύτερη, αλλιώς 0.

Ο μεσολαβητής ο οποίος θα διαθέσει άμεσα μεγαλύτερη ποσότητα θα είναι αυτός με το μεγαλύτερο κόστος. Βέβαια, όταν το κόστος αυτό γίνει αρκετά μεγάλο δεν θα είναι πλέον συμφέρουσα επιλογή για τον λιανέμπορο. Γι' αυτό και θα πρέπει να τεθεί ένα άνω όριο διαφοράς των κοστών των συμμετεχόντων έτσι ώστε όταν το ξεπερνούν να επιλέγεται η οικονομικότερη προσφορά.

Στο παράδειγμά μας η μεταβλητή $\text{lim}=800$ είναι το άνω όριο της διαφοράς των κοστών των συμμετεχόντων. Πάνω από το νούμερο θα επιλέγεται ο οικονομικότερος. Με βάση τα κριτήρια αυτά, τα αποτελέσματα 30 δοκιμών δίνονται παρακάτω:

112-8, 110-10, 112-8, 115-5, 114-6, 115-5, 111-9, 113-7, 112-8, 111-9, 113-7, 110-10, 100-20, 110-10, 114-6, 103-17, 111-9, 108-12, 113-7, 105-15, 109-11, 115-5, 109-11, 112-8, 111-9, 105-15, 111-9, 108-12, 105-15, 113-7.



Σχήμα 73: Πλήθος των νικών στις 30 περιόδους.

Έτσι και με τη βοήθεια της συνάρτησης κόστους, ακόμα και όταν οι μεσολαβητές συμμετέχουν σε ανταγωνιστικό μοντέλο, κατά μέσο όρο 110 από τις 120 βδομάδες νικητής είναι

ο «aver». Το αποτέλεσμα ήταν αναμενόμενο, μιας και όταν μελετούσαμε τη συνάρτηση κόστους στο μοντέλο με τους 3 διαφορετικούς τρόπους παραγωγείας, (πίνακας 5) το κόστος όσο του μεσολαβητή τόσο και του λιανέμπορου όταν χρησιμοποιούσαν τη συνάρτηση aver ήταν μικρότερο.

5.2.2 Εναλλακτικές πολιτικές

Σε ένα ανταγωνιστικό μοντέλο θα είχε ιδιαίτερο ενδιαφέρον η εύρεση μιας εναλλακτικής πολιτικής εκ μέρους του «ηττημένου» έτσι ώστε να μπορέσει να ανατρέψει το αποτέλεσμα. Για τον λόγο αυτό στη συνέχεια θα εξεταστεί με ποιες τεχνικές θα μπορέσει ο «expr» να γίνει πιο ανταγωνιστικός ώστε με τα συγκεκριμένα κριτήρια να είναι ο νικητής.

Σκοπός του είναι με χρήση της ίδιας συνάρτησης «exponential smoothing» να μπορέσει να μειώσει το συνολικό του κόστους, έτσι ώστε η διαφορά των ποσών αυτών να ανέβει άνω του $\lim (800)$ και να επιλεγεί ο ίδιος σαν οικονομικότερη λύση. Για να γίνει εφικτό αυτό, θα πρέπει να αλλάξει τις μεταβλητές l και k της συνάρτησης κόστους.

Θέτουμε

$$c_1 = lx_{11} + kx_{12} \text{ και } c_2 = l'x_{21} + k'x_{22} \text{ (} l' = 30 \text{ και } k' = 12 \text{)}$$

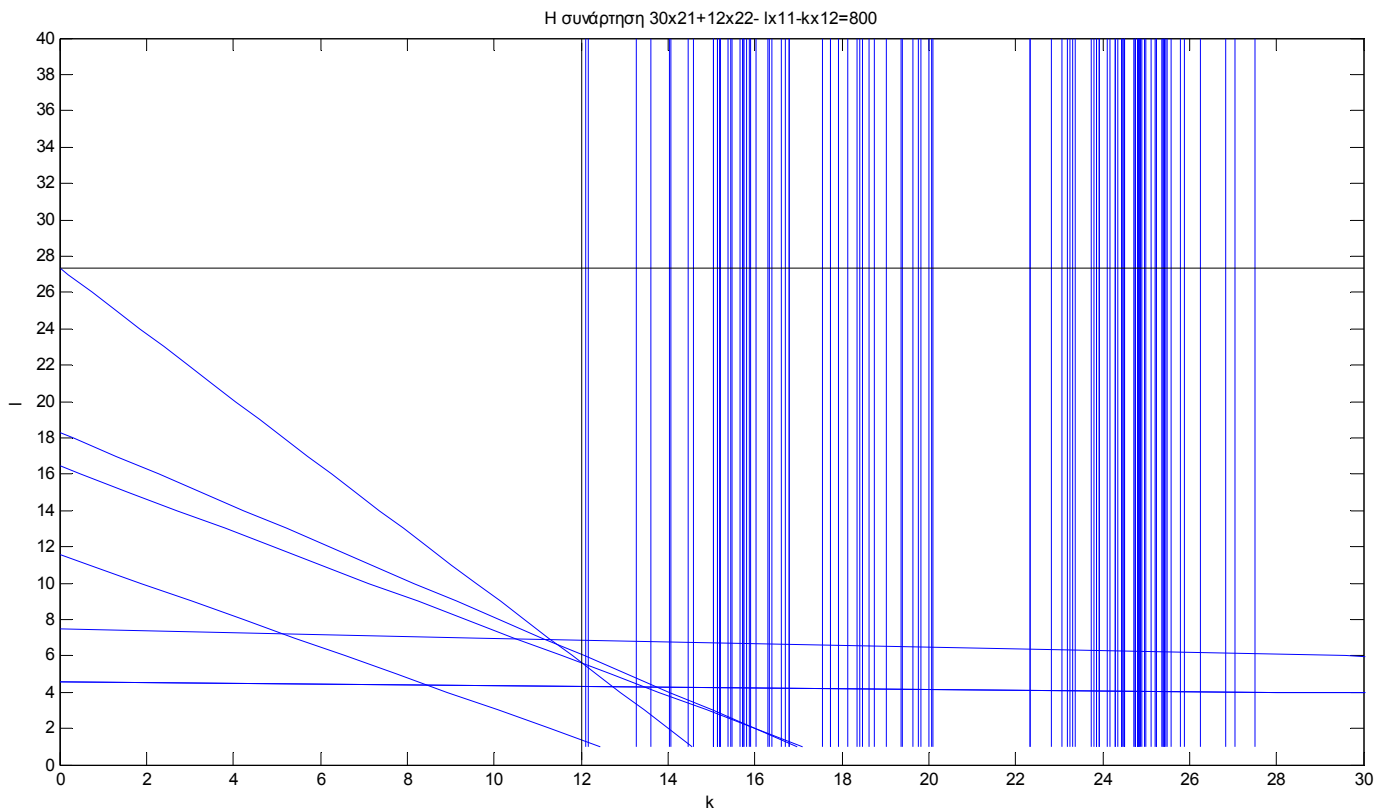
οι συναρτήσεις κόστους των «expr» «aver» αντίστοιχα, με x_{11} , x_{21} οι ποσότητες των αντίστοιχων μεσολαβητών που πρόκειται να προμηθευτούν από τον κατασκευαστή (αν χρειάζεται) και x_{12} , x_{22} οι ποσότητες που θα αποστείλουν άμεσα (είτε είναι το αποθεματικό μέρος είτε όλη η παραγωγή) τα μέλη αντίστοιχα. Σκοπός του προβλήματος είναι να μεγιστοποιηθεί η συνάρτηση c_1 με την προϋπόθεση ότι θα ισχύει $c_2 - c_1 > 800$ (1) $c_1 > 0$ (2). Δηλαδή το κόστος του “aver” θα εξακολουθεί να είναι μεγαλύτερο, αλλά εξαιτίας της μείωσης που θα πετύχουμε στο c_1 η διαφορά θα είναι μεγαλύτερη από 800 και έτσι θα επιλεγεί ο «expr» ως οικονομικότερη λύση.

Πρόκειται για ένα απλό πρόβλημα γραμμικού προγραμματισμού μεγιστοποίησης της συνάρτησης $c_1 = lx_{11} + kx_{12}$ υπό τους περιορισμούς (1) και (2).

Με τη βοήθεια του matlab σχεδιάστηκαν όλες οι γραμμικές συναρτήσεις τις μορφής

$$30x_{21} + 12x_{22} - lx_{11} - kx_{12} = 800 \quad (\text{σχήμα 74})$$

με δεδομένα τα $x_{11}, x_{12}, x_{21}, x_{22}$ (οι αριθμητικές τιμές των τεσσάρων μεταβλητών όσο και οι εντολές που δόθηκαν δίνονται αναλυτικά στο παράρτημα στη παράγραφο Γ).



Σχήμα 74: Η συνάρτηση $30x_{21}+12x_{22}-lx_{11}+kx_{12}=800$ για όλες τις τιμές των μεταβλητών.

Ο κάθε μεσολαβητής δεν θα μπορούσε να μειώσει το κόστος του χωρίς αυτό να επηρέαζε το κέρδος του. Γι' αυτό και θα προσπαθήσουμε να βρούμε τις μεγαλύτερες τιμές των k, l ώστε $l < 28$, $k \leq 12$ και να ισχύουν οι συνθήκες (1) και (2).

Από την επιτρεπόμενη περιοχή του σχήματος 74, το διάστημα στο οποίο ανήκει η l είναι $[27, 28)$ ενώ $k \leq 12$. Το ζεύγος τιμών που ικανοποιεί τις συνθήκες και μεγιστοποιεί την c_1 στα όρια που τέθηκαν είναι $(k, l) = (12, 27)$. Τα αποτελέσματα 30 προσομοιώσεων του συστήματος για αυτές τις τιμές αναγράφονται παρακάτω:

39-81, 42-78, 36-84, 36-84, 42-78, 38-82, 36-84, 51-69, 37-83, 36-84,
 25-95, 38-82, 37-83, 23-97, 34-86, 37-83, 36-84, 25-95, 38-82, 37-83,
 23-97, 34-86, 37-83, 36-84, 25-95, 38-82, 37-83, 23-97, 34-86, 37-83.

Τα αποτελέσματα είναι φυσικά υπέρ του “exp” όπου σε όλες τις δοκιμές επιλέχτηκε τις περισσότερες φορές ως οικονομικότερη λύση.

Παρατηρούμε ότι μία μείωση του συντελεστή l κατά 3 μονάδες, ήταν αρκετή για να αλλάξει το αποτέλεσμα. Ωστόσο, θα μπορούσε να μειώσει και άλλο τις μεταβλητές για μεγαλύτερη ασφάλεια στα αποτελέσματα. Όμως κάτι τέτοιο μπορεί να μηδένιζε το κέρδος του μεσολαβητή και για αυτό τα όρια των μεταβλητών δίνονται με τη μορφή κλειστού διαστήματος. Κάτι ανάλογο θα μπορούσε να συμβεί αν μειωνόταν μέχρι και μία μονάδα η μεταβλητή k . Το κόστος όμως των αποθεμάτων παίζει καθοριστικό ρόλο στην διαμόρφωση της συνάρτησης κόστους μιας και ανάλογα με την ποσότητα των αποθεμάτων επιλέγεται η πολιτική της παραγωγείας και γι’ αυτό είναι προτιμότερο η τιμή της k να παραμείνει σταθερή.

Αν η μεταβλητή l μειωνόταν λιγότερο (1 ή 2 μονάδες) το πιο πιθανό θα ήταν να είχαμε εναλλαγές στις επιλογές του λιανέμπορου, ανάλογα με το αίτημα του πελάτη.

Μπορεί να γίνει αντιληπτό ότι τα διαστήματα ασφαλείας μέσα στα οποία κινούνται οι δύο μεταβλητές καθορίζονται από διάφορους παράγοντες. Ο πιο σημαντικός είναι το αίτημα του πελάτη, γιατί από αυτό εξαρτώνται οι τιμές των συναρτήσεων “aver” και “exp”. Επίσης σημαντικό ρόλο παίζει και η αρχική ποσότητα των αποθεμάτων. Στο παράδειγμά μας είναι και για τους δύο μεσολαβητές 100 το ίδιο με τις αρχικές τιμές των παραπάνω συναρτήσεων. Μπορεί όμως να συμμετείχαν στο πρόβλημα με διαφορετικές αρχικές ποσότητες στα αποθέματα. Κάτι τέτοιο θα άλλαζε αρκετά τα αποτελέσματα του προβλήματος. Σημαντικό ρόλο παίζει και η τιμή της μεταβλητής lim . Αν υπήρχε το περιθώριο από τα υπόλοιπα μέλη της αλυσίδας η διαφορά των κοστών να ξεπερνά τη τιμή 800, τότε αναμένεται οι νίκες του “aver” να είναι ακόμα περισσότερες. Σε αυτή τη περίπτωση ο ανταγωνιστής θα έπρεπε να μειώσει περισσότερες μονάδες τις σταθερές ώστε να ελαττωθεί το κόστος του ακόμα πιο πολύ. Σίγουρα κάθε φορά που πραγματοποιείται η διαδικασία τα αποτελέσματα δύσκολα επαναλαμβάνονται.

Επίλογος

Αναμφίβολα, τελικό συμπέρασμα του τελευταίου κεφαλαίου είναι ότι ακόμα και σε ένα ανταγωνιστικό μοντέλο η χρήση της συνάρτησης κινητού μέσου είναι αποτελεσματικότερη και αποδοτικότερη από τη στιγμή που οι παραγγελίες δίνονται με τη βοήθεια κανονικής κατανομής. Βέβαια στην πραγματικότητα οι παραγγελίες δεν έχουν πάντα την ιδανική κατανομή στη διάρκεια του χρόνου, έτσι ώστε το μοντέλο που αναλύθηκε να προσαρμόζεται σε κάθε περίπτωση. Σε περιόδους όπου οι παραγγελίες πυκνώνουν (π.χ σε εποχιακά προϊόντα) το μοντέλο ανταποκρίνεται με μία απλή αλλαγή στη συνάρτηση, στο βέλος των αρχικών παραγγελιών στην αρχική σελίδα. Ανάλογα μπορούμε να το προσαρμόσουμε σε μία αλλαγή στα αποθέματα, στο πλήθος των μελών ακόμα και στη μεταξύ τους σύνδεση.

Κατά την μελέτη μιας εφοδιαστικής αλυσίδας δημιουργούνται ποικίλα και ενδιαφέροντα σενάρια. Ανάλογα με τις συνθήκες και τους κανόνες με βάση τους οποίους δουλεύει μία τέτοια αλυσίδα, με μελέτη και καταγραφή των δεδομένων κατά την διάρκεια ενός διαστήματος μπορούμε να καταλήξουμε στην κατάλληλη πολιτική. Η επιλογή της κατάλληλης πολιτικής είναι συνισταμένη πολλών και αλληλεπιδρώντων παραγόντων: Το επίπεδο των αποθεμάτων, ο ρυθμός εισερχόμενων παραγγελιών του πελάτη, ο χρόνος ανταπόκρισης όλων των μελών, το πλήθος των μελών της αλυσίδας, ο χρόνος καθυστέρησης παραλαβής, η ποσότητα των ελλειμματικών παραγγελιών είναι μερικοί από τους παράγοντες που επηρεάζουν την διαδικασία.

Γι' αυτό και χρησιμοποιήθηκε ένα απαιτητικό εργαλείο όπως είναι το CPN το οποίο υποστηρίζει την αλληλεξάρτηση των μελών (με χρήση των ιεραρχικών σελίδων), την εναλλαγή των πολιτικών που χρησιμοποιούν τα μέλη (με τη βοήθεια του προγραμματιστικού περιβάλλοντος), την επεξεργασία αρκετά μεγάλου όγκου δεδομένων (με τη βοήθεια των αναφορών και των state space και την εύκολη μεταφορά τους σε άλλα προγράμματα επεξεργασίας δεδομένων όπως Excel- Matlab).

ΠΑΡΑΡΤΗΜΑ

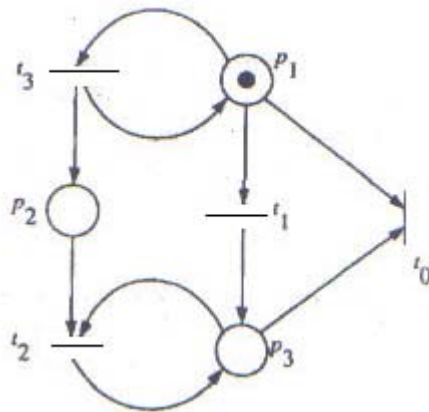
A. Ιδιότητες PN

1. Διάρκεια

Ένα δίκτυο θα λέμε ότι είναι ζωντανό, όταν υπάρχει ενεργή κατάσταση στο σύστημα. Η t_j θα είναι ζωντανή ως προς το μ αν για κάθε $\mu' \in R(C, \mu)$ υπάρχει μία ακολουθία σ τέτοια ώστε η t_j να είναι ενεργή στην $\delta(\mu', \sigma)$.

Επειδή δεν γνωρίζουμε εκ των προτέρων την περατότητα κάθε transition ορίζουμε κάποια επίπεδα διάρκειας:

- i) νεκρό (L0): δεν είναι ενεργό.
- ii) (L1) : αν μπορεί να πυροδοτήσει το λιγότερο 1 φορά σε ενεργή κατάσταση.
- iii) (L2) : αν μπορεί να πυροδοτήσει το λιγότερο k φορές σε ενεργή κατάσταση.
- iv) (L3) : αν t εμφανίζεται άπειρες φορές σε δεδομένη ακολουθία..
- v) (L3) : αν η t είναι L1 ζωντανή για κάθε $\mu \in \delta(\mu, t)$.



Σχήμα 1. Το δίκτυο PN.

Η μετάβαση του PN του σχήματος 10 t_0 είναι νεκρή, γιατί δεν υπάρχει καμία ακολουθία σ για την οποία η t_0 να πυροδοτήσει, η t_1 είναι L1 ζωντανή, t_2 L2, ενώ η t_3 είναι L3 ζωντανή.

2. Συντήρηση (conservation)

Ένα PN $C=(P, T, I, O)$ με αρχική κατάσταση μαρκαρισμάτων μ είναι αυστηρά διατηρημένο αν για κάθε $\mu' \in R(C, \mu)$ ισχύει

$$\sum_{p_i \in P} \mu'(p_i) = \sum_{p_i \in P} \mu(p_i)$$

Δηλαδή, για την ίδια κατάσταση το άθροισμα των μαρκαρισμάτων αυτής θα είναι το ίδιο και μετά την πυροδότηση μιας μετάβασης t τέτοιας ώστε $\delta(\mu, t) = \mu'$. Για να συμβεί αυτό, θα πρέπει ο αριθμός των καταστάσεων εισόδου ενός transition θα είναι ίδιος με τον αριθμό εξόδου, $|I(t_j)| = |O(t_j)|$.

Η ιδιότητα αυτή βρίσκει εφαρμογή σε συστήματα όπου θέλουμε τόσο η διανομή υλικού όσο και οι προηγούμενες παραγγελίες να γίνονται ισόποσα και ταυτόχρονα.

Άλλες ιδιότητες είναι η αντιστρεψιμότητα, η κάλυψη και η επιμονή.

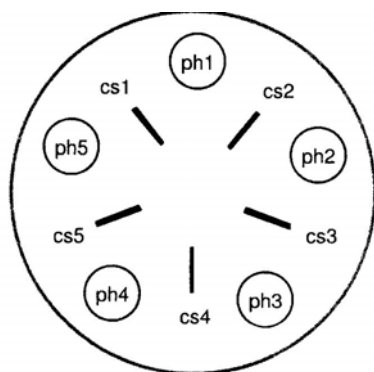
Ένα PN θα είναι αντιστρέψιμο όταν για κάθε $\mu \in R(C, \mu_0)$ είναι δυνατόν να επιστρέψουμε στην μ_0 για κάθε ακολουθία πυροδοτήσεων. Η ιδιότητα αυτή μας βοηθάει σε συστήματα όπου μετά την ολοκλήρωση μιας διαδικασίας χωρίς να προσδιορίσουμε να ενδιάμεσα στάδια να επιστρέψουμε στην αρχική κατάσταση μαρκαρισμάτων.

Στην κάλυψη μπορούμε να πούμε ότι ένα μ στο C καλύπτεται από το $\mu' \in R(C, \mu)$ αν και μόνο αν $\mu'(p) \geq \mu(p)$.

Εμμονή θα έχουμε όταν ανάμεσα σε δύο ενεργά transition η πυροδότηση του ενός δεν επηρεάζει την εκτελεσιμότητα του άλλου. Τέτοια PN χρησιμοποιούνται σε μοντέλα όπου δεν θέλουμε να υπάρχει αποκλεισμός από τους διαμοιρασμένους πόρους του συστήματος.

3. “Το δείπνο των φιλοσόφων”

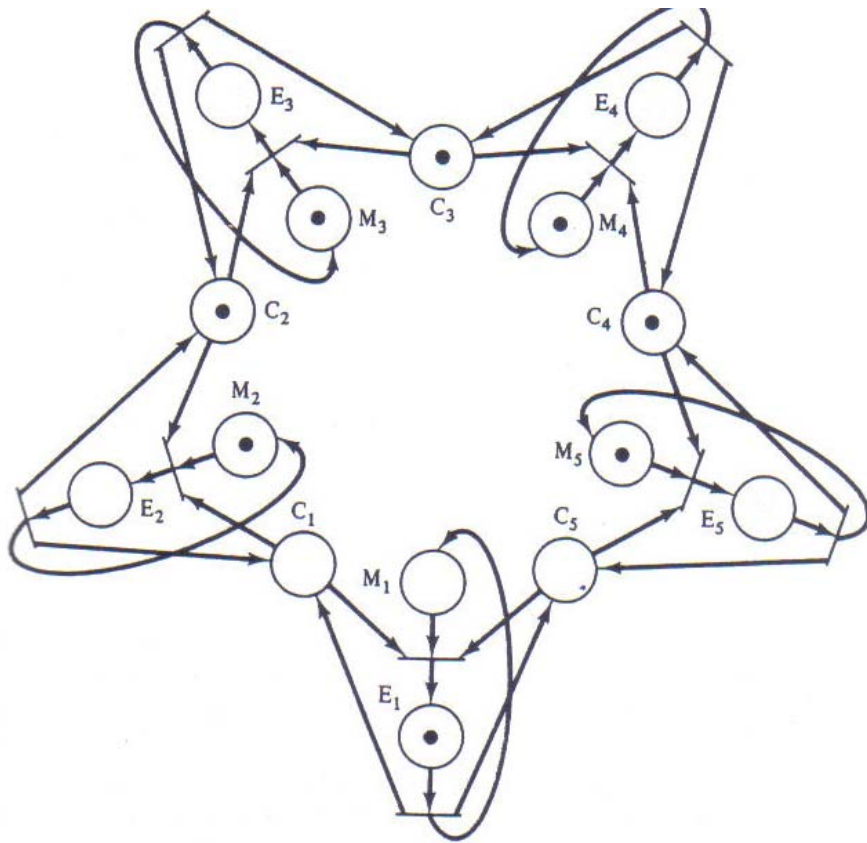
Ένα αρκετά γνωστό πρόβλημα μοντελοποίησης με την βοήθεια των PN είναι το δείπνο των φιλοσόφων. Πέντε Κινέζοι φιλόσοφοι κάθονται σε ένα στρογγυλό τραπέζι όπου στη μέση υπάρχει ένα πιάτο με ρύζι και ανάμεσά τους από ένα μόνο chopstick (σχήμα 2).



Σχήμα 2. Το τραπέζι των φιλοσόφων.

Για να φάει κάθε φιλόσοφος χρειάζεται δύο chopstick ενώ επιτρέπεται να χρησιμοποιήσει μόνο όσα βρίσκονται δίπλα του. Φυσικά δεν μπορούν να τρώνε όλοι ταυτόχρονα, αλλά όταν τρώει ο ph_i οι διπλανοί του ph_{i-1} ph_{i+1} βρίσκονται σε αναμονή.

Οι καταστάσεις ch_i συμβολίζουν το i -στό chopstick. Στο σχήμα 3 τα c_i είναι τα chopstick τα E_i τους φιλοσόφους ενώ τα M_i το μέλος που βρίσκεται σε αναμονή.



Σχήμα 3. Το petri net του δείπνου των φιλοσόφων.

Στην αρχική κατάσταση του συστήματος τρώει μόνο ο E_1 γι' αυτό και τα c_1 c_5 δεν είναι μαρκαρισμένα (δεν είναι διαθέσιμα). Όταν μαρκαριστούν τα c_1 c_5 και M_1 σημαίνει πως οι E_2 E_5 μπαίνουν σε σειρά ενώ ο E_1 μπαίνει σε αναμονή. Όταν οι μάρκες M_2 M_5 μεταπηδούν στις E_2 E_5 δεν είναι πλέον σε θέση αναμονής, τα $c_1c_2c_4c_5$ δεν είναι μαρκαρισμένα, άρα δεν είναι διαθέσιμα. Όποιος από τους E_2 E_5 τελειώσει πρώτος αφήνει το τα chopstick για να ενεργοποιηθούν και άλλες καταστάσεις. Επειδή δεν γίνεται στο συγκεκριμένο εργαλείο οι μεταπηδήσεις να συμβαίνουν ταυτόχρονα, υπάρχει περίπτωση ένας E_i ενώ έχει ολοκληρώσει την διαδικασία να βρίσκεται σε ουρά, για να ενεργοποιηθεί. Στο συγκεκριμένο πρόβλημα, λόγω συμμετρίας δεν υπάρχει κανόνας για το ποιο μέλος θα έχει προτεραιότητα, αλλά ορίζεται τυχαία από το σύστημα. Γι' αυτό και κάθε φορά που θα μπαίνει το δίκτυο σε λειτουργία οι καταστάσεις που θα ενεργοποιούνται πιθανώς να είναι διαφορετικές.

B. Μοντέλο CPN

Παρακάτω αναγράφονται οι αρχικές τιμές του μοντέλου οι οποίες εισάγονται κάθε φορά που τρέχει η διαδικασία. Με «1» συμβολίζεται ο αριθμός των μαρκών με τιμή που δίνεται μετά το σύμβολο «`» τη χρονική στιγμή «@...».

1`88@19+++	1`103@112+++	1`108@67+++
1`92@44+++	1`103@100+++	1`108@55+++
1`96@117+++	1`103@93+++	1`108@53+++
1`96@94+++	1`103@74+++	1`108@29+++
1`96@13+++	1`103@62+++	1`108@7+++
1`96@12+++	1`103@56+++	1`109@80+++
1`97@90+++	1`103@38+++	1`109@43+++
1`97@48+++	1`103@30+++	1`109@42+++
1`98@107+++	1`103@10+++	1`109@40+++
1`98@98+++	1`104@120+++	1`109@24+++
1`98@57+++	1`104@102+++	1`109@22+++
1`98@35+++	1`104@91+++	1`109@8+++
1`98@31+++	1`104@59+++	1`109@4+++
1`99@114+++	1`104@50+++	1`110@118+++
1`99@97+++	1`104@17+++	1`110@110+++
1`99@86+++	1`104@16+++	1`110@63+++
1`99@69+++	1`104@11+++	1`111@70+++
1`99@34+++	1`104@1+++	1`111@47+++
1`99@33+++	1`105@99+++	1`111@6+++
1`99@3+++	1`105@96+++	1`112@76+++
1`100@92+++	1`105@75+++	1`113@104+++
1`100@77+++	1`105@71+++	1`113@103+++
1`100@73+++	1`105@60+++	1`113@81+++
1`100@72+++	1`105@51+++	1`113@66+++
1`100@52+++	1`105@37+++	1`113@64+++
1`100@49+++	1`105@25+++	1`113@46+++
1`100@45+++	1`106@119+++	1`113@5+++
1`100@36+++	1`106@113+++	1`114@41+++
1`100@28+++	1`106@82+++	1`115@115+++
1`100@20+++	1`106@65+++	1`115@108+++
1`100@15+++	1`106@32+++	1`115@2+++
1`101@106+++	1`106@23+++	1`119@21
1`101@88+++	1`106@18+++	
1`101@61+++	1`106@14+++	
1`101@58+++	1`107@111+++	
1`101@54+++	1`107@105+++	
1`101@39+++	1`107@87+++	
1`101@27+++	1`107@85+++	
1`101@26+++	1`107@78+++	
1`102@116+++	1`108@109+++	
1`102@101+++	1`108@95+++	
1`102@83+++	1`108@89+++	
1`102@79+++	1`108@84+++	
1`102@9+++	1`108@68+++	

Γ. Ανταγωνιστικό μοντέλο-Εναλλακτικές πολιτικές

Στον παρακάτω πίνακα σημειώνονται οι τιμές των μεταβλητών x11,x12, x21, x22 και οι εντολές που χρησιμοποιήθηκαν στο Matlab για τη δημιουργία του σχήματος 62

X11	X12	X21	X22	X11	X12	X21	X22
2	98	2	99	100	0	101	9
5	100	0	124	100	0	101	12
46	51	11	109	100	0	101	13
52	51	23	87	100	0	102	10
67	37	28	81	100	0	102	22
80	94	29	85	100	0	102	23
94	0	31	93	100	0	103	21
94	0	34	77	100	0	103	21
94	0	35	74	100	0	104	6
94	0	38	86	100	0	104	9
94	0	40	80	100	0	105	19
95	0	41	72	101	0	106	17
95	0	46	63	101	0	107	13
96	0	46	64	101	0	107	17
96	0	52	57	101	0	108	0
96	0	57	66	101	0	108	1
97	0	61	59	101	0	109	0
98	0	62	62	101	0	109	0
98	0	63	54	101	0	109	0
98	0	65	55	101	0	110	0
98	0	68	55	101	0	110	0
98	0	71	53	102	0	110	0
98	0	76	37	102	0	110	0
98	0	77	36	102	0	110	0
98	0	91	34	102	0	111	0
99	0	95	21	102	0	111	0
99	0	95	31	102	0	111	0
99	0	97	28	103	0	111	7
99	0	98	12	103	0	111	9
99	0	98	13	103	0	112	0
99	0	98	14	103	0	112	0
99	0	98	19	103	0	112	0
99	0	99	18	103	0	112	3
99	0	100	1	104	0	112	17
99	0	100	1	104	0	112	21
100	0	100	23	104	0	112	25
100	0	100	24	105	0	113	0

105	0	114	0	188	0	123	0
106	0	115	0	190	0	123	0
106	0	115	0	192	0	123	4
106	0	115	4	192	0	124	0
108	0	115	11	193	0	124	0
110	0	116	0	194	0	124	0
112	0	116	0	194	0	124	0
116	0	117	12	194	0	124	0
120	0	117	17	194	0	125	0
137	0	118	0	195	0	125	0
140	0	118	0	195	0	125	0
143	0	118	6	195	0	127	0
143	0	118	7	195	0	129	0
149	0	119	0	198	0	130	0
151	0	120	3	198	0	130	0
152	0	121	0	200	0	131	0
156	0	121	0	201	0	132	0
163	0	121	0	201	0	132	0
167	0	122	0	201	0	134	0
172	0	122	17	203	0	136	0
172	0	123	0				
173	0	123	0				
174	0	123	0				
182	0	123	0				
184	0	123	0				
184	0	123	0				

```

b=1:120;
a=(30*x21(1)+12*x22(1)-b*x12(1)-800)/x11(1);
plot(a,b)
hold on
for i=1:120
a=(30*x21(i)+12*x22(i)-b*x12(i)-800)/x11(i);
plot(a,b)
end
hold off

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Kurt Jensen, **Modelling Supply Chains with (Coloured) Petri Nets**.
2. http://www.go-online.gr/ebusiness/specials/article.html?article_id=1315
3. van der vorst, jack G.A.J., A. J.M. Beulens, P. van Beek (2000). **Modelling and Simulating Multi-Echelon Food Systems**, **European Journal of Operational Research**, Vol. 122, pp. 354-366.
4. Ρούμπος Χρήστος, **Μοντέλα Επιχειρησιακών Διαδικασιών**, σημειώσεις στο μάθημα «Τεχνολογία Διοίκησης Επιχειρησιακών Διαδικασιών».
5. Marcel Welters **building logistic games in cpn tools** <http://is.tm.tue.nl/staff/wvdaalst/LOGgames/papers/report.pdf> .
6. http://www.go-online.gr/ebusiness/specials/article.html?article_id=1315
7. Dragana Makajić-Nikolić, Biljana Panić, Mirko Vujošević , **Bullwhip effect and supply chain modelling and analysis using CPN tools**, Operations Research Laboratory »Jovan Petric«, Faculty of Organizational Sciences, University of Belgrade, http://www.daimi.au.dk/CPnets/workshop04/cpn/papers/makajic-nikolic_panic_vujosevic.pdf pp. 2-7.
8. Marcel Welters, **Building Logistic Games in CPN Tools**, Technische Universiteit Eindhoven Eindhoven, January 22nd 2005, <http://wwwis.win.tue.nl/~wvdaalst/LOGgames/papers/report.pdf> pp. 3-4, 16-18.
9. <http://beergame.mit.edu/>
10. CHRISTOS I. PAPANAGNOU, **ANALYZING DIFFERENT ORDERING POLICIES IN A SERIES SUPPLY CHAIN BY USING COLOURED PERTI NETS.** □
11. Γεωργούσης & Χατζοπουλος Χ., **Μαζική παραγωγή (Mass Production)** <http://helleanic.wordpress.com/%CE%9C%CE%B1%CE%B6%CE%B9%CE%BA%CE%AE-%CF%80%CE%B1%CF%81%CE%B1%CE%B3%CF%89%CE%B3%CE%AE/>
12. Ν. Λυγερός, **Ανάλυση του φαινομένου μαστιγίου** http://www.lygeros.org/1511-Analysh_toy_fainomenoy_mastigiouy.html
13. Αβραμίδου Ευτέρπη, **Μοντελοποίηση και Προσομοίωση με Χρωματισμένα Δίκτυα Petri** .
14. Tadao Murata, **Properties, Analysis and Applications**, **Proceedings of the IEEE**, vol. 77, No.4, April 1989, pp. 544-546.

15. Kurt Jensen, **An introduction to the Practical Use of coloured Petri Nets**, Department of Computer Science, University of Aarhus, www.daimi.aau.dk/~kjensen/ .
16. Βλασερός Βασίλειος, **Διαχείριση εφοδιαστικής αλυσίδας**, σημειώσεις παραδόσεων από το Δ.Π.Μ.Σ. στο μάθημα «Μαθηματικά της αγοράς και της παραγωγής», Οκτώβριος 2004.
17. Γεώργιος Τζιραλής, **Προγνωστικές αγορές**, σημειώσεις από το μάθημα «Εφοδιαστική & Οργάνωση Παραγωγής Ι» του Τομέα Βιομηχανικής Διοίκησης & Επιχειρησιακής Έρευνας στο Τμήμα Μηχανολόγων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, Οκτώβρης 2008.
18. Kurt Jensen, **A Brief Introduction to Coloured Petri Nets**, Computer Science Department, University of Aarhus, <http://www.daimi.aau.dk/~kjensen/> .
19. Christos Papanagnou, George Halikias, **Supply Chain Modelling and Control under proportional inventory-replenishment policies**, School of Engineering and Mathematical Sciences.
20. Γ. Φεσάκης, Κ. Χαλάτσης, Φ. Καλαβάσης, Α. Δημητρακοπούλου, **Οι περιορισμοί του μοντέλου υπολογισμού Von Neumann**, στο μάθημα «ανάπτυξη εφαρμογών σε προγραμματιστικό περιβάλλον» και τα δίκτυα Petri.
21. Christos Papanagnou and George Halikias, **A state-space approach for analysing the bullwhip effect in supply chains**, Control Engineering Research Centre School of Engineering and Mathematical Sciences.
22. Tauseef Aized, Koji Takahashi, and Ichiro Hagiwara, **Advanced Multiple Product Flexible Manufacturing System Modelling Using Coloured Petri Net**, Vol.11 No.6, 2007.
23. Anders Moen, **Introduction to Petri Nets - Part One**, March 26, 2003.
24. Lars Michael Kristensen_, Jens Bζk Jyrgensen, and Kurt Jensen, **Application of Coloured Petri Nets in System Development**, Department of Computer Science University of Aarhus.
25. Kurt Jensen, **An Introduction to the Theoretical Aspects of Coloured Petri Nets**, Computer Science Department, Aarhus University.
26. S. Bartkevičius, V. Bagdonas, M. Juraška, K. Šarkauskas, **Train Traffic Simulation with Coloured Petri Nets and Schedule Optimisation**, *ELEKTRONIKA IR ELEKTROTECHNIKA*. 2005.
27. Mogens Nielsen and Vladimiro Sassone, **Petri Nets and Other Models of Concurrency**.

28. http://www.daimi.au.dk/CPnets/intro/example_indu.html, **Colored Petri Nets Theory and Applications: Modeling and Verifications of Protocols**, Multimedia and Network Research Lab CTI, DePaul University Chicago, USA, Νοέμβρης 2002.
29. Ana Karla Alves de Medeiros and Christian W. Günther, **Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms**, /faculteit technologie management.
30. Anne Vinter Ratzer, Lisa Wells, Henry Michael Lassen, Mads Laursen, Jacob Frank, Qvortrup, Martin Stig Stissing, Michael Westergaard, Søren Christensen, Kurt Jensen, **CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets**, Department of Computer Science, University of Aarhus.
31. Michel Beaudouin-Lafon, Wendy E. Mackay, Peter Andersen, Paul Janecek, Mads Jensen, Michael Lassen, Kasper Lund, Kjeld Mortensen, Stephanie Munck, Anne Ratzer, Katrine Ravn, Søren Christensen and Kurt Jensen, **CPN/Tools: A Post-WIMP Interface for Editing and Simulating Coloured Petri Nets**, Department of Computer Science University of Aarhus.
32. Chaoyue Xiong, Tadao Murata, Jeffery Tsai, **Modeling and Simulation of Routing Protocol for Mobile Ad Hoc Networks Using Colored Petri Nets**, Department of Computer Science University of Illinois at Chicago.
33. David N. Ford, **The Application of System Dynamics to Concurrent Engineering**, 1999.
34. Andreas Glausch, Wolfgang Reisig, **Distributed Abstract State Machines and Their Expressive Power**.
35. Lars Michael Kristensen_ and Kurt Jensen, **Specification and Validation of an Edge Router Discovery Protocol for Mobile Ad Hoc Networks**, Department of Computer Science, University of Aarhus.
36. Clarence A. Ellis and Gary J. Nutt, **Modeling and Enactment of Workflow Systems**, Department of Computer Science University of Colorado, March 1993.
37. Claus Reinke, **Tool Support for Haskell-Coloured Petri Nets**.
38. Guy Helmer, Johnny Wong, Mark Slagell, Vasant Honavar, Les Miller, Yanxin Wang, Xia Wang and Natalia Stakhanova, **Software fault tree and coloured Petri net–based specification, design and implementation of agent-based intrusion detection systems**, Department of Computer Science, Iowa State University, Atanasoff Hall, Ames, Iowa.
39. J. C. Hill and P. C. Wright, **From text to Petri Nets: the difficulties of describing accident scenarios formally**, The Human Computer Interaction Group, Department of Computing Science, University of York, Heslington.

40. Χρήστος Σωτηρίου, **Χρονισμός Ψηφιακών Συστημάτων -1-**.
41. Χρήστος Σωτηρίου, **Χρονισμός Ψηφιακών Συστημάτων -9-**.
42. Πέτρος Γαβαλάκης, **Μοντελοποίηση και Ανάλυση Θεμελιωδών Στοιχείων Συστημάτων Παραγωγής με χρήση Έγχρωμων Δικτύων Petri**, Τμήμα Μηχανικών Παραγωγής και Διοίκησης Πολυτεχνείο Κρήτης, Αύγουστος 2006.
43. Robert B. Cooper, **Introduction to Queueing Theory, Second edition**.
44. University of Aarhus, **CPN Tools State Space Manual**, January 2006.
45. Christian W. Günther and Wil M.P. van der Aalst, **Modeling the Case Handling Principles with Colored Petri Nets**, Department of Technology Management, Eindhoven University of Technology.
46. George J. Tsinarakis, *Student Member*, Nikos C. Tsourveloudis, *Associate Member, IEEE*, and Kimon P. Valavanis, *Senior Member*, **Modeling, Analysis, Synthesis, and Performance Evaluation of Multioperational Production Systems With Hybrid Timed Petri Nets**.
47. G. J. Tsinarakis, K. P. Valavanis, **Modular Hybrid Petri Nets for Studying Multioperational Production Systems Where Parts Follow Multiple Alternative Processes**, Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orleans, April 2004.
48. G. J. Tsinarakis, K. P. Valavanis, N. C. Tsourveloudis, **Modular Petri Net Based Modeling, Analysis and Synthesis of Dedicated Production Systems**, Proceedings of the 2003 IEEE International Conference on Robotics & Automation Taipei, Taiwan, September 14-19, 2003.
49. G. J. Tsinarakis, N. C. Tsourveloudis, **Modular Petri Net based modeling, analysis, synthesis and performance evaluation of random topology dedicated production systems**, Journal of Intelligent Manufacturing, 16, 67–92, 2005.
50. Aphrodite Tsalgaidou, Panos Louridas, George Fesakis, Thanasis Schizas, **Multilevel Petri Nets for Modeling and Simulating Organizational Dynamic Behaviour**, University of Athens.
51. Απόστολος Μπουρνέτας, **Μοντέλα ανταγωνισμού και συνεργασίας σε εφοδιαστικές αλυσίδες**, σημειώσεις στο μάθημα «Μαθηματικά της αγοράς και της παραγωγής», Πανεπιστήμιο Αθηνών θερινό σχολείο 2006.
52. University of Aarhus, **Design/CPN Occurrence Graph Manual Version 3.0**, 1996.

53. Andreas Glausch and Wolfgang Reisig, **On the Expressive Power of Unbounded-Nondeterministic Abstract State Machines**, Humboldt-Universität zu Berlin Institut für Informatik.
54. Chap 7: **Stochastic Petri Net Models**.
55. Christophe Hirel, Bruno Tu_n and Kishor S. Trivedi, **SPNP: Stochastic Petri Nets. Version 6.0**, Center for Advanced Computing and Communication Department of Electrical and Computer Engineering Duke University, Durham U.S.A.
56. G. J. Tsinarakis, N. C. Tsourveloudis, and K. P. Valavanis, **Petri Net Modeling of Routing and Operation Flexibility in Production Systems**, Proceedings of the 13th Mediterranean Conference on Control and Automation Limassol, Cyprus, June 27-29, 2005.
57. Γιάννης Βεργινάδης, Διαχείριση διαδικασιών - Digital Enterprise Research Network, www.imu.iccs.gr .
58. Lars M. Kristensen, Soren Christensen, Kurt Jensen, **The practitioner's guide to colouredPetri nets, CPN Group**, Department of Computer Science, University of Aarhus, Denmark, 1998.
59. W.M.P. van der Aalst, **Three good reasons for using a Petri-net-Based Workflow management system**, Eindhoven University of Technology Department of Mathematics and Computing Science.
60. CIS 525 Parallel and Distributed Software Development, **Reductions of occurrence graph for colored Petri nets**.
61. Wolfgang Reisig, **The Decent Philosophers: An exercise in concurrent behaviour**, *Humboldt-Universität zu Berlin Institute of Informatics Unter den Linden 6, 10099 Berlin, Germany*.
62. Wolfgang Reisig, **The Scholten/Dijkstra Pebble Game Played Straightly, Distributedly, Online and Reversed**, Department of Computer Science, Humboldt-Universität zu Berlin.
63. W. Reisig, **The Universal Net Composition Operator**, Humboldt-Universität zu Berlin.
64. Αντώνης; Ταταράκης, Βασίλης Ζεϊμπέκης, Αναδιοργάνωση της διαχείρισης εφοδιαστικής αλυσίδας (SCM) μέσω τεχνολογιών διαδικτύου.
65. Bruno de Athayde Prata, Ernesto Ferreira Nobre Junior, Giovanni Cordeiro Barroso, **A stochastic colored Petri Net model to allocate equipments for earth moving operations**, November 2007.

66. Kurt Jensen, Lars Michael Kristensen and Lisa Wells, **Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems**, Department of Computer Science University of Aarhus.
67. June Dong and Ding Zhang, **A Supply Chain Network Equilibrium Model with Random Demands**, Department of Marketing and Management School of Business State University of New York at Oswego New York, May 2002.
68. Α. Ν. Σκοδράς, **Αλγοριθμικές μηχανές καταστάσεων**, σημειώσεις από το μάθημα ‘Ψηφιακή λογική: Ακολουθιακά κυκλώματα’.
69. Βασίλης Χ. Γερογιάννης, **Αντικειμενοστραφής μοντελοποίηση των διαδικασιών της ανάπτυξης λογισμικού**, 17^ο Συνέδριο της Ε.Ε.Ε.Ε.-«Διαχείριση Κινδύνων».
70. Β.Βεσκούκης, Δ.Παραδείσης, Δ.Αργιαλάς, Δ.Δεληκαράογλου, Β.Καραθανάση, Β.Μασσίνας, **Στοιχεία επεξεργασίας σημάτων**, ΕΜΠ - ΣΧΟΛΗ ΑΤΜ, Ακ. Έτος 2004-2005.
71. Ν. Παπανδρέου, **Στοχαστικά Σήματα & Εφαρμογές- Τυχαίες Διαδικασίες Διακριτού Χρόνου**, Πανεπιστήμιο Πατρών Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Στοχαστικά Σήματα και Εφαρμογές CEID 2007-2008.
72. Ν. Παπανδρέου, **Στοχαστικά Σήματα & Εφαρμογές- Μοντελοποίηση Σήματος**, Πανεπιστήμιο Πατρών Τμήμα Μηχανικών Η/Υ και Πληροφορικής, Στοχαστικά Σήματα και Εφαρμογές CEID 2007-2008.
73. Τμ. Επ. Υπολογιστών, Πανεπιστήμιο Κρήτης, **Μηχανές Πεπερασμένων Καταστάσεων (Finite State Machines - FSM)**, σημειώσεις από το μάθημα «Ψηφιακή Σχεδίαση», **Φθινόπωρο 2003**, http://www.csd.uoc.gr/~hy120/03f/lab10_fsm.htm .
74. **Συστήματα Πληροφορικής- Διαχείριση της Εφοδιαστικής Αλυσίδας**, Τα συστήματα Πληροφορικής στην Ελληνική αγορά- διαχείριση της εφοδιαστικής αλυσίδας.
75. **Η γλώσσα CPN ML**
76. Αφροδίτη Τσαλγατίδου, **Μοντέλα Επιχειρησιακών Διαδικασιών**, σημειώσεις στο μάθημα «Τεχνολογία Διοίκησης Επιχειρησιακών Διαδικασιών», www.di.uoa.gr/~pms541 .
77. Kurt Jensen, **Coloured Petri Nets**, Department of Computer Science University of Aarhus, Denmark, <http://www.daimi.au.dk/CPnets/slides/>.
78. **Δικτυακά Μοντέλα Διεργασιών**.
79. Ρούμπος Χρήστος, **Προγραμματισμός και έλεγχος παραγωγής**. Χρονικός προγραμματισμός, σημειώσεις στο μάθημα «Αρχές Διοίκησης και οργάνωση παραγωγής».

80. Ρούμπος Χρήστος, **Μοντέλα Επιχειρησιακών Διαδικασιών**, σημειώσεις στο μάθημα «Τεχνολογία Διοίκησης Επιχειρησιακών Διαδικασιών».
81. Αβραμίδου Ευτέρπη, **Μοντελοποίηση και προσομοίωση με χρωματισμένα δίκτυα Petri**, Πτυχιακή εργασία, επιβλέπων Παναγιώτης Κατσαρός, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Σχολή Θετικών Επιστημών Τμήμα πληροφορικής.
82. Βασίλης Οδοντίδης, Έλεγχος πρωτοκόλλων ηλεκτρονικού εμπορίου με χρωματισμένα Petri: ατομική εκτέλεση συναλλαγών, ειδικό θέμα, επιβλέπων καθηγήτρια Μ. Γουσίδου Κουτίτα, Θεσσαλονίκη 2005.
83. Ηλιάδου Ευτυχία, Προσομοίωση Δικτύων Ουρών αναμονής με χρωματισμένα δίκτυα Petri, διπλωματική εργασία με επιβλέπων Παναγιώτης Κατσαρός, Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης Σχολή Θετικών Επιστημών τμήμα Πληροφορικής, Θεσσαλονίκης 2006.
84. Γιάννης Α. Φίλης, Δίκτυα Παραγωγής C.A.M. Αναμονητικά συστήματα Γραμμές παραγωγής F.M.S., Πολυτεχνείο Κρήτης, Ιανουάριος 2003.
85. Γιάννης Α. Φίλης, Δίκτυα Παραγωγής C.A.M., Αποθέματα, Πρόβλεψη, Προγραμματισμός, Πολυτεχνείο Κρήτης, Μάρτιος 2006.