



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

**ΣΧΟΛΗ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ
ΤΟΜΕΑΣ ΜΕΤΑΦΟΡΩΝ ΚΑΙ ΣΥΓΚΟΙΝΩΝΙΑΚΗΣ ΥΠΟΔΟΜΗΣ**

Διπλωματική εργασία

**Διερεύνηση θεμάτων οργάνωσης του υποσυστήματος
διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και
χερσαίου χώρου με τη χρήση του προγράμματος AnyLogic**

ΣΤΑΘΟΠΟΥΛΟΣ ΣΠΥΡΙΔΩΝ

Επιβλέπων: Μπαλλής Αθανάσιος, Αναπληρωτής Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΙΟΥΛΙΟΣ 2017

ΠΕΡΙΛΗΨΗ

Διερεύνηση θεμάτων οργάνωσης του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου με τη χρήση του προγράμματος AnyLogic

ΣΤΑΘΟΠΟΥΛΟΣ ΣΠΥΡΙΔΩΝ

Κάθε σύγχρονος τερματικός σταθμός φορτοεκφόρτωσης εμπορευματοκιβωτίων είναι ένα πολύπλοκο σύστημα που απαιτεί συνεχή προσπάθεια βελτίωσης καθώς υπάρχουν πολλές διαφορετικές συνιστώσες που αλληλεπιδρούν μεταξύ τους. Στην παρούσα διπλωματική εργασία, διερευνήθηκαν θέματα οργάνωσης του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου με τη χρήση του προγράμματος AnyLogic. Αναπτύχθηκαν απλά μοντέλα προσομοίωσης (σε αντιστοιχία με τα μοντέλα M/M/n, E2/M/n, M/E2/n, E2/E2/1, E2/E2/2 και (M/M/2) : (GD/N/∞), της θεωρίας αναμονής) προκειμένου να γίνει έλεγχος εγκυρότητας των μοντέλων προσομοίωσης του AnyLogic. Στη συνέχεια αναπτύχθηκε και επιλύθηκε ένα μοντέλο προσομοίωσης με 2 (ή περισσότερες) θέσεις παραβολής, κάθε μια από τις οποίες έχει διαφορετικό ρυθμό εξυπηρέτησης. Πραγματοποιήθηκε, επίσης, διερεύνηση των εναλλακτικών τεχνικών οργάνωσης του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου, μέσω μοντέλου προσομοίωσης που αναπαριστά τις διαδικασίες κατανομής εργασιών μεταξύ των οχημάτων που απαρτίζουν την ομάδα φόρτωσης/εκφόρτωσης και μεταφοράς των εμπορευματοκιβωτίων. Τέλος, κατασκευάστηκε ένα σύνθετο μοντέλο προσομοίωσης του επιλιμένιου σιδηροδρομικού σταθμού στον λιμένα του Πειραιά και διατυπώθηκαν σενάρια ώστε να μελετηθεί η λειτουργία του υποσυστήματος «κρηπιδώματος - χερσαίου χώρου» σε σχέση με τον απαιτούμενο αριθμό των οχημάτων μεταφοράς εμπορευματοκιβωτίων.

ABSTRACT

STATHOPOYLOS SPYRIDON

Investigation of issues relating to the organization of the container handling subsystem between platforms and terrains using the program

Every modern container loading / unloading terminal is a complex system that requires constant improvement, as there are many different components interacting with each other. In this diploma thesis, issues of organization of container handling subsystem between platforms and terrestrial space were investigated by using the AnyLogic program. Simple simulation models of queuing (in correspondence with the M/M/n, E2/M/n, M/E2/n, E2/E2/1, E2/E2/2 and (M/M/2) : (GD/N/∞), models of the queue theory) were developed in order to validate the simulating models of AnyLogic. Then, a simulator model had been formulated and resolved by using 2 (or more) berth positions, each of which has a different service rate. An exploration of the alternative technical arrangements of the container handling subsystem between the platform and land space was also explored through a simulation representing the workflow processes between the vehicles constituting the loading / unloading transporting containers. Finally, a complex simulation model was constructed, which approach organizationally the existing railway station in port of Piraeus and scenarios were prepared to study the operation of the “platforms and terrestrial space” subsystem in accordance with the required number of container transport vehicles.

ΠΕΡΙΧΟΜΕΝΑ

1	ΕΙΣΑΓΩΓΗ	17
1.1	Το σύστημα διακίνησης εμπορευματοκιβωτίων	17
1.2	Αντικείμενο και στόχος της διπλωματικής εργασίας.....	18
1.3	Δομή διπλωματικής εργασίας.....	19
2	ΠΛΟΙΑ ΜΕΤΑΦΟΡΑΣ ΚΑΙ ΛΙΜΕΝΕΣ ΔΙΑΚΙΝΗΣΗΣ ΕΜΠΟΡΕΥΜΑΤΟΚΙΒΩΤΙΩΝ	21
2.1	Πλοία μεταφοράς εμπορευματοκιβωτίων	21
2.2	Μηχανολογικός εξοπλισμός λιμένα	23
2.2.1	Εξοπλισμός θαλάσσιας πλευράς	23
2.2.2	Εξοπλισμός μεταφοράς εμπορευματοκιβωτίων μεταξύ θαλάσσιας και χερσαίας πλευράς.....	24
2.2.3	Εξοπλισμός χερσαίας πλευράς	27
3	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ.....	31
3.1	Περιγραφή των βασικών μεθόδων και αλγορίθμων επίλυσης προβλημάτων στις λιμενικές εγκαταστάσεις.....	31
3.2	Προβλήματα που καλούνται να επιλυθούν σε ένα λιμενικό εμπορευματικό σταθμό	38
3.2.1	Γενικά.....	38
3.2.2	Γενική επισκόπηση των προβλημάτων ενός εμπορευματικού λιμένα	39
3.3	Χώρος παραβολής	41
3.3.1	Τοποθεσία θέσης παραβολής (BAP).....	41
3.3.2	Ανάθεση των γερανών κρηπιδώματος (QCs)	42
3.3.3	Προγραμματισμός των γερανών κρηπιδώματος.....	42
3.4	Χώρος μεταφοράς εντός του λιμένα	44
3.5	Χώρος στοιβασίας.....	45
3.6	Σύνθεση των επιμέρους υποσυστημάτων ενός λιμένα	47
4	ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΛΙΜΕΝΙΚΩΝ ΕΓΚΑΤΑΣΤΑΣΕΩΝ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ANYLOGIC	50
4.1	Το πρόγραμμα προσομοίωσης Anylogic	50
4.1.1	Περιβάλλον μοντελοποίησης με δυνατότητες χρήσης πολλών μεθόδων	50
4.1.2	Αναπαράσταση και οπτικοποίηση.....	52
4.1.3	Εξειδικευμένες βιβλιοθήκες εργαλείων	53

4.1.4	Ενσωμάτωση χάρτη GIS	54
4.1.5	Δυνατότητα αυτοματοποιημένης λειτουργίας.....	55
4.1.6	Διαλειτουργικότητα δεδομένων.....	56
4.1.7	Επεκτάσιμη και προσαρμόσιμη πλατφόρμα	56
4.1.8	Βασικά βήματα στην διαμόρφωση ενός μοντέλου Anylogic	57
4.1.9	Πλεονεκτήματα και μειονεκτήματα του Anylogic	59
4.2	Ανάπτυξη και έλεγχος εγκυρότητας μοντέλων λιμενικών εγκαταστάσεων	60
4.2.1	Το μοντέλο M/M/1	60
4.2.2	Το μοντέλο M/M/2	63
4.2.3	Το μοντέλο E2/M/1.....	66
4.2.4	Το μοντέλο M/E2/2.....	68
4.2.5	Το μοντέλο E2/E2/1	70
4.2.6	Το μοντέλο E2/E2/2	72
4.2.7	Το μοντέλο (M/M/2) : (GD/N/∞). (Περιορισμένος χώρος αναμονής).....	74
4.2.8	Ανάπτυξη μοντέλου M/M*/2 με διαφορετικό ρυθμό εξυπηρέτησης σε κάθε θέση παραβολής	77

5 ΠΡΟΣΟΜΟΙΩΣΗ ΣΤΡΑΤΗΓΙΚΩΝ ΑΝΑΘΕΣΗΣ ΕΡΓΑΣΙΩΝ ΣΤΑ ΟΧΗΜΑΤΑ ΜΕΤΑΦΟΡΑΣ 83

5.1	Σταθερή ανάθεση οχημάτων μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων.....	84
5.2	Ανάθεση ομάδας οχημάτων μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων.....	85
5.2.1	Επιλογή ανάθεσης οχημάτων μεταφοράς με κυκλικό κανόνα	86
5.2.2	Επιλογή ανάθεσης οχημάτων μεταφοράς με τυχαίο κανόνα	87
5.2.3	Πρώτο διαθέσιμο όχημα μεταφοράς	87
5.2.4	Μέγιστη απόσταση από το εμπορευματοκιβώτιο	88
5.2.5	Ελάχιστη απόσταση από το εμπορευματοκιβώτιο	89
5.3	Συγκριτικά αποτελέσματα	90
5.4	Σταθερή ανάθεση οχημάτων μεταφοράς στις γερανογέφυρες σταθερού κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων.....	90
5.5	Ανάθεση ομάδας οχημάτων μεταφοράς στις γερανογέφυρες σταθερού κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων.....	94
5.6	Συγκριτικά Αποτελέσματα σεναρίων	98
5.7	Προσομοίωση στρατηγικών ανάθεσης εργασιών στα οχήματα μεταφοράς με μεταβλητούς χρόνους κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων.....	99
5.8	Σταθερή ανάθεση με αποκλίσεις στο χρόνο κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς	99
5.9	Ανάθεση ομάδας με αποκλίσεις στο χρόνο κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς	102
5.10	Συγκριτικά αποτελέσματα σεναρίων	106

6	ΠΡΟΣΟΜΟΙΩΣΗ ΕΠΙΛΙΜΕΝΙΟΥ ΣΙΔΗΡΟΔΡΟΜΙΚΟΥ ΣΤΑΘΜΟΥ ΠΕΙΡΑΙΑ ...	108
6.1	Πρώτη εικόνα του μοντέλου και διαδραστικότητα	109
6.2	Γενική επισκόπηση λιμένα	110
6.3	Θαλάσσια περιοχή	112
6.4	Περιοχής κρηπιδώματος	113
6.4.1	Διαδικασία εκφόρτωσης πλοίων	113
6.4.2	Διαδικασία φόρτωσης πλοίων	114
6.5	Ζώνη κυκλοφορίας μεταξύ κρηπιδώματος και χώρων στοιβασίας	115
6.6	Χώρος στοιβασίας εμπορευματοκιβωτίων	116
6.7	Επιλιμένος σιδηροδρομικός σταθμός	117
6.7.1	Η λειτουργία των εμπορευματικών τραίνων	117
6.7.2	Η λειτουργία των γερανογεφυρών που εξυπηρετούν τα τραίνα	117
6.8	Πύλες εισόδου- εξόδου και χώροι αναμονής των φορτηγών	118
6.8.1	Η λειτουργία των εξωτερικών φορτηγών	118
6.9	Η λειτουργία των ελκυστήρων εντός του λιμένα	119
6.10	Αποτελέσματα μοντέλου	120
6.11	Διατύπωση και επίλυση σεναρίων φορτοεκφόρτωσης πλοίων	121
7	ΣΥΜΠΕΡΑΣΜΑΤΑ	134
8	ΕΙΣΗΓΗΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ	135
9	ΒΙΒΛΙΟΓΡΑΦΙΑ	137
	ΠΑΡΑΡΤΗΜΑ Α ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ	143
	ΠΑΡΑΡΤΗΜΑ Β: ΑΝΑΛΥΣΗ ΕΝΔΕΙΚΤΙΚΟΥ ΜΟΝΤΕΛΟΥ	175
	ΠΑΡΑΡΤΗΜΑ Γ ΑΝΑΛΥΣΗ ΜΟΝΤΕΛΟΥ ΠΡΟΣΟΜΟΙΩΣΗΣ ΛΙΜΕΝΑ	196
	ΠΑΡΑΡΤΗΜΑ Δ ΕΠΕΞΗΓΗΣΗ ΤΟΥ ΜΕΤΑΒΛΗΤΟΥ ΧΡΟΝΟΥ ΚΥΚΛΟΥ	210
	ΠΑΡΑΡΤΗΜΑ Ε ΘΕΩΡΙΑ ΑΝΑΜΟΝΗΣ	211

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 1 Υποσυστήματα διακίνησης εμπορευματοκιβωτίων σε έναν επιλιμένιο τερματικό σταθμό εμπορευματοκιβωτίων	18
Εικόνα 2 Η εξέλιξη των πλοίων μεταφοράς εμπορευματοκιβωτίων [Πηγή: Μαλιάγκα, Ε. (2014)]	22
Εικόνα 3 Εξέλιξη γερανών θαλάσσιας πλευράς [Πηγή: Edmonds, R. (2010)]	23
Εικόνα 4 Γερανογέφυρα με μονό βαγονέτο (αριστερό τμήμα) και Γερανογέφυρα με διπλό βαγονέτο (δεξιό τμήμα) [Πηγή: Bartošek, A., Marek, O. (2013)].....	24
Εικόνα 5 Ελκυστήρες με συρόμενες βάσεις [Πηγή: Kalmar, http://www.kalmarglobal.com/equipment/ , τελευταία πρόσβαση 10/9/2015.].....	25
Εικόνα 6 Ελκυστήρες με δυνατότητα ρυμούλκησης πολλών τρέιλερ [Πηγή: Buiscar, Cargo Solutions, http://www.buiscar.com/mts.htm , τελευταία πρόσβαση 10/9/2015.].....	25
Εικόνα 7 Οχήματα πλαίσια [Πηγή: Kalmar, http://www.kalmarglobal.com/equipment .].....	26
Εικόνα 8 Αυτομάτως καθοδηγούμενα οχήματα μεταφοράς [Πηγή: Terex, http://www.terex.com/port-solutions/en/ .].....	27
Εικόνα 9 Αυτοματοποιημένα οχήματα με δυνατότητα ανύψωσης [Πηγή: Terex, http://www.terex.com/port-solutions/en/].....	27
Εικόνα 10 Περονοφόρα ανυψωτικά οχήματα [Πηγή: Kalmar, http://www.kalmarglobal.com/equipment].....	28
Εικόνα 11 Οχήματα εμπρόσθιας στοιβάσεως [Πηγή: Kalmar, http://www.kalmarglobal.com/equipment .].....	28
Εικόνα 12 Γερανογέφυρες επί σιδηροτροχιών [Πηγή: KoneCranes, http://www.konecranes.com	29
Εικόνα 13 Γερανογέφυρες επί ελαστικών τροχών [Πηγή: KoneCranes, http://www.konecranes.com]	30
Εικόνα 14 Προσδιοριστικά & στοχαστικά μοντέλα [Πηγή: Καρλαύτης Μ.Γ., Λαγαρός Ν.Δ. (2010),]	31
Εικόνα 15 Το σύστημα αναμονής [Πηγή: Ξηροκώστας, Δ. (1990), Επιχειρησιακή έρευνα – Εφαρμοσμένη θεωρία αναμονής]	34
Εικόνα 16 Περιοχές του λιμανιού [Πηγή: springer.com , Chapter 2 Maritime Terminal Operational Problems]	39
Εικόνα 17 Agent-based προσέγγιση [πηγή: anylogic/ features [http://www.anylogic.com/features/]].....	50
Εικόνα 18 System Dynamics προσέγγιση [πηγή: anylogic/ features [http://www.anylogic.com/features/]	51
Εικόνα 19 Discrete Event προσέγγιση [Πηγή: anylogic/ features [http://www.anylogic.com/features/]].....	51

Εικόνα 20	Συνδυασμός προσεγγίσεων [πηγή: anylogic/ features [http://www.anylogic.com/features/]	51
Εικόνα 21 4.1.2	Αναπαράσταση και οπτικοποίηση [πηγή: anylogic/ features [http://www.anylogic.com/features/]	52
Εικόνα 22	Εξειδικευμένες βιβλιοθήκες εργαλείων [πηγή: anylogic/ features [http://www.anylogic.com/features/]	53
Εικόνα 23	Ενσωμάτωση GIS χάρτη [πηγή: anylogic/ features [http://www.anylogic.com/features/]	54
Εικόνα 24	Δυνατότητα αυτοματοποιημένης λειτουργίας [Πηγή: anylogic/ features http://www.anylogic.com/features/]	55
Εικόνα 25	Διαλειτουργικότητα των δεδομένων [Πηγή: anylogic/ features http://www.anylogic.com/features/]	56
Εικόνα 26	Επεκτάσιμη και προσαρμόσιμη πλατφόρμα [Πηγή: anylogic/ features [http://www.anylogic.com/features/]	56
Εικόνα 27	διάγραμμα ροής απλού μοντέλου	58
Εικόνα 28	Η έννοια του πράκτορα [Πηγή: anylogic/ features http://www.anylogic.com/features/	58
Εικόνα 29 4.1.10	Πλεονεκτήματα και μειονεκτήματα του Anylogic [Πηγή: Critical Manufacturing S.A. (2014)]	59
Εικόνα 30	Η διαμόρφωση του μοντέλου M/M/1 με το λογισμικό AnyLogic	60
Εικόνα 31	Χρόνοι αναμονής M/M/1	61
Εικόνα 32	Απόκλιση μεθόδων	62
Εικόνα 33	Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/M/1	63
Εικόνα 34	Η διαμόρφωση του μοντέλου M/M/2 με το λογισμικό AnyLogic	64
Εικόνα 35	Χρόνοι αναμονής M/M/2	64
Εικόνα 36	Διαγράμματα επαναλήψεων	65
Εικόνα 37	Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/M/2	66
Εικόνα 38	Η διαμόρφωση του μοντέλου E2/M/1 με το λογισμικό AnyLogic	67
Εικόνα 39	Χρόνοι αναμονής E2/M/1	67
Εικόνα 40	Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα E2/M/1	68
Εικόνα 41	Η διαμόρφωση του μοντέλου E2/M/2 με το λογισμικό AnyLogic	69
Εικόνα 42	Χρόνοι αναμονής M/E2/2	69

Εικόνα 43 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/E2/2	70
Εικόνα 44 Η διαμόρφωση του μοντέλου E2/E2/1 με το λογισμικό AnyLogic	71
Εικόνα 45 Χρόνοι αναμονής E2/E2/1.....	71
Εικόνα 46 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα E2/E2/1.....	72
Εικόνα 47 Η διαμόρφωση του μοντέλου E2/E2/2 με το λογισμικό AnyLogic	72
Εικόνα 48 Χρόνοι αναμονής E2/E2/2.....	73
Εικόνα 49 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα E2/E2/2.....	74
Εικόνα 50 Η διαμόρφωση του μοντέλου (M/M/2) : (GD/N/∞) με το λογισμικό AnyLogic	74
Εικόνα 51 Χρόνοι αναμονής (M/M/2) : (GD/N/∞).....	75
Εικόνα 52 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα (M/M/2) : (GD/N/∞).....	76
Εικόνα 53 Η διαμόρφωση του μοντέλου M/M*/2 με το λογισμικό AnyLogic	77
Εικόνα 54 Διαγράμματα M/M*/2 : $\mu/\mu^*=1$	78
Εικόνα 55 Διαγράμματα M/M*/2 : $\mu/\mu^*=0,8333$	79
Εικόνα 56 Διαγράμματα M/M*/2 : $\mu/\mu^*=0,66667$	79
Εικόνα 57 Διαγράμματα M/M*/2 : $\mu/\mu^*=0,5$	80
Εικόνα 58 Διαγράμματα M/M*/2: $\mu/\mu^*=0,333$	81
Εικόνα 59 Διαγράμματα M/M*/2: $\mu/\mu^*=0,1667$	81
Εικόνα 60 Αναλυτικά αποτελέσματα M/M*/2: $\mu/\mu^*=0$	82
Εικόνα 61 Διαγράμματα M/M*/2: $\mu/\mu^*=0$	82
Εικόνα 62 Συγκριτικά αποτελέσματα.....	82
Εικόνα 63 Γράφημα εκφόρτωσης τριών πλοίων	83
Εικόνα 64 Η διαμόρφωση του μοντέλου εκφόρτωσης τριών πλοίων με το λογισμικό AnyLogic (σταθερή ανάθεση).....	84
Εικόνα 65 Η διαμόρφωση του μοντέλου εκφόρτωσης τριών πλοίων με το λογισμικό AnyLogic (ανάθεση ομάδας)	86
Εικόνα 66 Κώδικας κυκλικού κανόνα	86
Εικόνα 67 Κώδικας τυχαίου κανόνα	87
Εικόνα 68 Κώδικας κανόνα πρώτου διαθέσιμου.....	87
Εικόνα 69 Κώδικας κανόνα μέγιστης απόστασης.....	88
Εικόνα 70 Κώδικας κανόνα ελάχιστης απόστασης.....	89

Εικόνα 71 Αποτελέσματα εκφόρτωσης τριών πλοίων.....	90
Εικόνα 72 Γράφημα φόρτωσης-εκφόρτωσης 2 πλοίων (σταθερού κύκλου/σταθερή ανάθεση) ...	91
Εικόνα 73 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (σταθερού κύκλου/σταθερή ανάθεση)	92
Εικόνα 74 Σενάριο 2.1 χρόνοι αναμονής	94
Εικόνα 75 Γράφημα φόρτωσης-εκφόρτωσης 2 πλοίων (σταθερού κύκλου/ανάθεση ομάδας).....	95
Εικόνα 76 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (σταθερού κύκλου/ ανάθεση ομάδας).....	95
Εικόνα 77 Σενάριο 2.2 χρόνοι αναμονής	97
Εικόνα 78 Συγκριτικά αποτελέσματα σεναρίων 2.1, 2.2	98
Εικόνα 79 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (αποκλίσεις στον χρόνο κύκλου/ σταθερή ανάθεση)	100
Εικόνα 80 Σενάριο 2.3 χρόνοι αναμονής	102
Εικόνα 81 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (αποκλίσεις στον χρόνο κύκλου / ανάθεση ομάδας).....	103
Εικόνα 82 Χρόνοι αναμονής.....	105
Εικόνα 83 Διάγραμμα αποτελεσμάτων σεναρίου 2.3	107
Εικόνα 84 Διάγραμμα αποτελεσμάτων σεναρίου 2.4	107
Εικόνα 85 Συγκριτικό διάγραμμα αποτελεσμάτων σεναρίων 2.3, 2.4	107
Εικόνα 86 Πρώτη εικόνα	109
Εικόνα 87 Γενική επισκόπηση λιμένα 3D.....	110
Εικόνα 88 Γενική επισκόπηση λιμένα /διάγραμμα ροής.....	110
Εικόνα 89 Κάτοψη λιμένα	111
Εικόνα 90 Θαλάσσια περιοχή διάγραμμα ροής	112
Εικόνα 91 Διαδικασία εκφόρτωσης πλοίων 3D	113
Εικόνα 92 Διαδικασία εκφόρτωσης πλοίων διάγραμμα ροής	113
Εικόνα 93 Διαδικασία φόρτωσης πλοίων 3D.....	114
Εικόνα 94 Διαδικασία φόρτωσης πλοίων διάγραμμα ροής	114
Εικόνα 95 Διαδικασία φόρτωσης πλοίων (Ζώνη κυκλοφορίας μεταξύ κρηπιδώματος και χώρων στοιβάσεως) 3D	115
Εικόνα 96 Διαδικασία φόρτωσης πλοίων διάγραμμα ροής	115
Εικόνα 97 Χώρος στοιβάσεως 3D.....	116
Εικόνα 98 Χώρος στοιβάσεως διάγραμμα ροής	116
Εικόνα 99 Επιλιμένιος σιδηροδρομικός σταθμός 3D	117

Εικόνα 100 Η λειτουργία των εμπορικών τραίνων διάγραμμα ροής	117
Εικόνα 101 Η λειτουργία των γερανογεφυρών που απασχολούνται στα τραίνα διάγραμμα ροής	117
Εικόνα 102 Η λειτουργία των εξωτερικών φορτηγών	118
Εικόνα 103 Η λειτουργία των εξωτερικών φορτηγών διάγραμμα ροής	118
Εικόνα 104 Η λειτουργία των ελκυστήρων εντός του λιμένα 3D	119
Εικόνα 105 Η λειτουργία των ελκυστήρων εντός του λιμένα διάγραμμα ροής.....	119
Εικόνα 106 Αποτελέσματα μοντέλου	121
Εικόνα 107 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	123
Εικόνα 108 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	123
Εικόνα 109 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	124
Εικόνα 110 Σενάριο 3.1 μέσοι χρόνοι εξυπηρέτησης πλοίων	124
Εικόνα 111 Σενάριο 3.1 μέσοι χρόνοι εξυπηρέτησης πλοίων	124
Εικόνα 112 Σενάριο 3.2 μέσοι χρόνοι εξυπηρέτησης πλοίων	126
Εικόνα 113 Σενάριο 3.2 μέσοι χρόνοι εξυπηρέτησης πλοίων	126
Εικόνα 114 Σενάριο 3.3 μέσοι χρόνοι εξυπηρέτησης πλοίων	127
Εικόνα 115 Σενάριο 3.4 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	128
Εικόνα 116 Σενάριο 3.4 μέσοι χρόνοι εξυπηρέτησης πλοίων	128
Εικόνα 117 Σενάριο 3.5 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	129
Εικόνα 118 Σενάριο 3.5 μέσοι χρόνοι εξυπηρέτησης πλοίων	130
Εικόνα 119 Σενάριο 3.6 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	131
Εικόνα 120 Σενάριο 3.6 μέσοι χρόνοι εξυπηρέτησης πλοίων	131
Εικόνα 121 Σενάριο 3.7 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται.....	132
Εικόνα 122 Σενάριο 3.7 μέσοι χρόνοι εξυπηρέτησης πλοίων	132
Εικόνα 123 Πλεονεκτήματα και μειονεκτήματα του Anylogic [Πηγή: Critical Manufacturing S.A. (2014)]	144
Εικόνα 124 Ανοίγοντας το Anylogic	144
Εικόνα 125 New model	145

Εικόνα 126 Interface	145
Εικόνα 127 Agent	146
Εικόνα 128 Δημιουργία agent 1.....	147
Εικόνα 129 Δημιουργία agent 2.....	148
Εικόνα 130 Δημιουργία agent 3.....	148
Εικόνα 131 Agent properties.....	149
Εικόνα 132 Resource Pool.....	150
Εικόνα 133 Resource Pool 2	151
Εικόνα 134 Παράδειγμα κώδικα.....	153
Εικόνα 135 Μεταβλητή	153
Εικόνα 136 Μεταβλητή	153
Εικόνα 137 Παράμετρος	154
Εικόνα 138 Παράμετρος	155
Εικόνα 139 Συνάρτηση 1	156
Εικόνα 140 Συνάρτηση 2	156
Εικόνα 141 Συνάρτηση 3	157
Εικόνα 142 Διάγραμμα ροής.....	157
Εικόνα 143 Scale.....	158
Εικόνα 144 GIS 1.....	163
Εικόνα 145 GIS 2.....	164
Εικόνα 146 GIS 3.....	165
Εικόνα 147 GIS 4.....	165
Εικόνα 148 RoutProvider	166
Εικόνα 149 GIS-Agent.....	167
Εικόνα 150 Εισαγωγή κώδικα	168
Εικόνα 151 Εισαγωγή κώδικα 2	168
Εικόνα 152 Εισαγωγή κώδικα 3	169
Εικόνα 153 Επανάληψη στον κώδικα	169
Εικόνα 154 Εισαγωγή σημείου	171
Εικόνα 155 Ιδιότητες source	171
Εικόνα 156 Block μετακίνησης.....	172
Εικόνα 157 Block αναμονής	173

Εικόνα 158 Block sink.....	174
Εικόνα 159 Animation ενδεικτικού μοντέλου.....	175
Εικόνα 160 Διάγραμμα ροής ενδεικτικού μοντέλου	176
Εικόνα 161 Source.....	176
Εικόνα 162 Source1.....	177
Εικόνα 163 Source-main	177
Εικόνα 164 SeizeQC1.....	178
Εικόνα 165 SeizeQC1 1.....	178
Εικόνα 166 DelayQC1	179
Εικόνα 167 DelayQC1 1	180
Εικόνα 168 ReleaseQC1.....	180
Εικόνα 169 RreleaseQC1 1	181
Εικόνα 170 QueueQC1	182
Εικόνα 171 QueueQC1 1	182
Εικόνα 172 Hold	182
Εικόνα 173 Hold 1	183
Εικόνα 174 TaskEnd	183
Εικόνα 175 TaskEnd1	184
Εικόνα 176 SeizePMs1	184
Εικόνα 177 SeizePMs1 1.....	185
Εικόνα 178 DelayLoading1	186
Εικόνα 179 DelayLoading1 1	186
Εικόνα 180 MoveTo1	187
Εικόνα 181 MoveTo1 1.....	187
Εικόνα 182 DelayUnloading1	188
Εικόνα 183 DelayUnloading1 1	188
Εικόνα 184 releasePMs1	189
Εικόνα 185 ReleasePMs1 1	189
Εικόνα 186 Queue1	190
Εικόνα 187 Queue 2	190
Εικόνα 188 MoveTo3	191
Εικόνα 189 MoveTo3 1.....	192

Εικόνα 190 TaskEndPMs1	192
Εικόνα 191 TaskEndPMs1 1.....	193
Εικόνα 192 ResourceTaskEndPMs1	193
Εικόνα 193 resourceTaskStartPMs1.....	193
Εικόνα 194 ResourceTaskEndPMs1	194
Εικόνα 195 ResourcePool.....	194
Εικόνα 196 ResourcePool1.....	195
Εικόνα 197 Μεταβλητές μοντέλου προσομοίωσης λιμένα	196
Εικόνα 198 Μεταβλητές μοντέλου προσομοίωσης λιμένα1	197
Εικόνα 199 Παράμετροι agent	198
Εικόνα 200 SelectOutputByShipColour	198
Εικόνα 201 ShipGreenQueue	199
Εικόνα 202 ShipContainerSource	199
Εικόνα 203 ShipsContainerExit	200
Εικόνα 204 MoveToQC.....	201
Εικόνα 205 GreenShipCollection.....	201
Εικόνα 206 Startup.....	202
Εικόνα 207 Move To Storage	203
Εικόνα 208 move To Storage With YC.....	204
Εικόνα 209 Select Output 6.....	204
Εικόνα 210 YC Seize 2.....	205
Εικόνα 211 Move To Train Unloading Position	206
Εικόνα 212 Select Output 11.....	206
Εικόνα 213 Wait To Unload 1.....	207
Εικόνα 214 move To Train Unloading Position	207
Εικόνα 215 Train Container Source.....	207
Εικόνα 216 Trucks Queue.....	208
Εικόνα 217 SelectUnloadPosition	208
Εικόνα 218 Unloading Queue.....	209
Εικόνα 219 Το σύστημα αναμονής [πηγή: Ξηροκώστας, Δ. (1990)].....	211
Εικόνα 220 Σύστημα αναμονής πολλών παράλληλων σταθμών εξυπηρέτησης πηγή: [πηγή: Ξηροκώστας, Δ. (1990)].....	212

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1 Έλεγχος απαιτούμενου αριθμού επαναλήψεων	61
Πίνακας 2 Αποκλίσεις M/M/1.....	62
Πίνακας 3 Έλεγχος αριθμού επαναλήψεων	65
Πίνακας 4 M/M/2 Αναλυτικά αποτελέσματα.....	66
Πίνακας 5 Αναλυτικά αποτελέσματα E2/M/1	68
Πίνακας 6 αναλυτικά αποτελέσματα M/E2/2	70
Πίνακας 7 Αναλυτικά αποτελέσματα E2/E2/1.....	71
Πίνακας 8 Αναλυτικά αποτελέσματα E2/E2/2.....	73
Πίνακας 9 Αναλυτικά αποτελέσματα(M/M/2) : (GD/N/∞)	75
Πίνακας 10 Προτεραιότητες M/M*/2 αναλυτικά αποτελέσματα	77
Πίνακας 11 Αναλυτικά αποτελέσματα M/M*/2 : $\mu/\mu^*=1$	78
Πίνακας 12 Αναλυτικά αποτελέσματα M/M*/2 : $\mu/\mu^*=0,8333$	78
Πίνακας 13 Αναλυτικά αποτελέσματα M/ M*/2 : $\mu/\mu^*=0,66667$	79
Πίνακας 14 Αναλυτικά αποτελέσματα M/ M*/2 : $\mu/\mu^*=0,5$	80
Πίνακας 15 Αναλυτικά αποτελέσματα M/M*/2: $\mu/\mu^*=0,333$	80
Πίνακας 16 Αναλυτικά αποτελέσματα M/M*/2: $\mu/\mu^*=0,1667$	81
Πίνακας 17 Συγκριτικά αποτελέσματα σεναρίων 2.1, 2.2.....	98
Πίνακας 18 Αποτελέσματα σεναρίου 2.3	106
Πίνακας 19 Αποτελέσματα σεναρίου 2.4	106
Πίνακας 20 Σενάριο 3.1 αναλυτικά αποτελέσματα	122
Πίνακας 21 Σενάριο 3.1 αναλυτικά αποτελέσματα	123
Πίνακας 22 Σενάριο 3.1 αναλυτικά αποτελέσματα	123
Πίνακας 23 Σενάριο 3.2 αναλυτικά αποτελέσματα	125
Πίνακας 24 Σενάριο 3.3 αναλυτικά αποτελέσματα	126
Πίνακας 25 Σενάριο 3.4 αναλυτικά αποτελέσματα	127
Πίνακας 26 Σενάριο 3.5 αναλυτικά αποτελέσματα	129
Πίνακας 27 Σενάριο 3.6 αναλυτικά αποτελέσματα	130
Πίνακας 28 Σενάριο 3.7 αναλυτικά αποτελέσματα	132
Πίνακας 29 Αναλυτικά αποτελέσματα σύγκρισης σεναρίων 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 και 3.7....	133
Πίνακας 30 Συμβολισμός των συστημάτων αναμονής. [πηγή: Kendal, D.G., (1964)].....	214
Πίνακας 31 Τύποι μοντέλων Poisson και μη Poisson [πηγή: Θάνου Ε. (2015)].....	216

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ θερμά τον επιβλέποντα Αναπληρωτή Καθηγητή Αθανάσιο Μπαλλή, της Σχολής Πολιτικών Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, για την ευκαιρία που μου έδωσε να εκπονήσω την παρούσα διπλωματική εργασία, τη συνεχή ενασχόληση, την καθοδήγηση και την υποστήριξη του σε όλη τη διάρκεια της και την κ. Τατιάνα Μοσχόβου για τις πολύτιμες συμβουλές της.

Επίσης ευχαριστώ πολύ τον φίλο μου, Ιωάννη Νικολάου, για την βοήθειά του σε θέματα που αφορούν το προγραμματιστικό κομμάτι της διπλωματικής, όπως και την Άννα Χαβιατζή για την πολύτιμη βοήθειά της.

1 ΕΙΣΑΓΩΓΗ

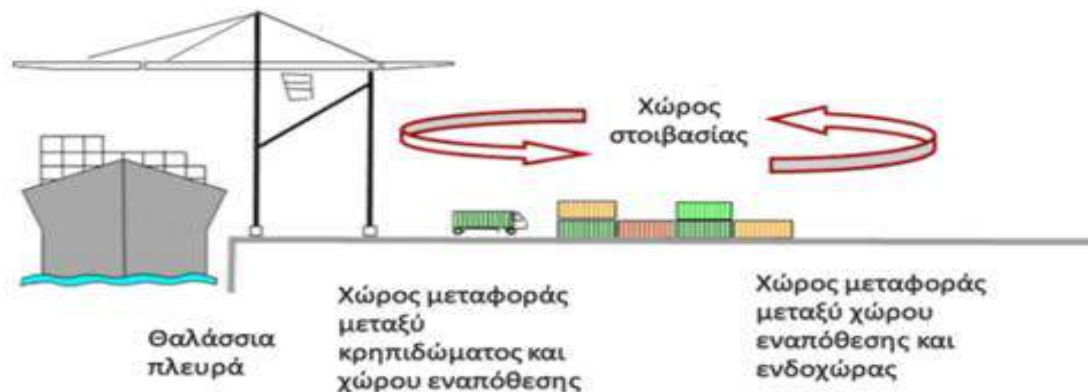
1.1 Το σύστημα διακίνησης εμπορευματοκιβωτίων

Εμπορευματοκιβώτιο είναι ένα μεγάλο κιβώτιο, συνήθως μεταλλικό, το οποίο κλείνει ερμητικά, έχει μια ή περισσότερες πόρτες προκειμένου να διευκολύνει την πρόσβαση στα εμπορεύματα και χρησιμοποιείται επανειλημμένα για τη μεταφορά στερεών, υγρών ή αέριων φορτίων. Είναι κατασκευασμένο έτσι ώστε να διευκολύνει τον χειρισμό, την φόρτωση, τη στοιβάση πάνω στο πλοίο, την εκφόρτωση και την ταυτόχρονη μεταφορά μεγάλου αριθμού φορτίων ή μεγάλης ποσότητας με την χρήση μηχανικών μέσων [Simon, 1974].

Ένας από τους πλέον μοντέρνους τύπους φορτηγών πλοίων είναι το πλοίο μεταφοράς εμπορευματοκιβωτίων. Επίσημη ονομασία στην ελληνική είναι εμπορευματοκιβωτιοφόρο αλλά λόγω της μεγάλης σε μήκος σύνθετης αυτής ονομασίας χρησιμοποιείται πιο εύχρηστα ο εκ της αγγλικής *containership*. Τα πλοία αυτού του τύπου έχουν ναυπηγηθεί έτσι ώστε τόσο τα κύπη τους (αμπάρια) όσο και το κύριο κατάστρωμα αυτών να μπορούν να δέχονται ένα μεγάλο αριθμό εμπορευματοκιβωτίων με ανάλογες διαρρυθμίσεις αλλά και ενισχύσεις. Φέρουν δε ειδικές υποδοχές σύμφωνα με διεθνή πρότυπα διαστάσεων, στις οποίες και στερεώνονται αυτά με ασφάλεια. Τα πλοία αυτά, με το είδος αυτό μεταφοράς των φορτίων περιορίζουν χρονικά στο ελάχιστο τη παραμονή τους στους λιμένες εκφορτώνοντας πλήρη και παραλαμβάνοντας κενά ή αντίστροφα, χωρίς να υφίστανται περισσότερες χρεώσεις. Στη ναυτιλιακή οικονομία ο τρόπος αυτός διακίνησης συσκευασμένων προϊόντων επέφερε μεγάλες αλλαγές και ραγδαία αύξηση.

Συνέπεια δε αυτών υπήρξε νέα επίσης πολιτική οικονομικής εκμετάλλευσης των λιμένων με τη δημιουργία ειδικών τερματικών σταθμών, σημείων φορτοεκφόρτωσής των, και με ενίσχυση μέσων και ειδικών γερανών σε μοντέρνες πλέον τεχνικές υποδομές. Βέβαια πολλά εμπορευματοκιβώτιο πλοία διαθέτουν επίσης δικά τους μέσα φορτοεκφόρτωσης εμπορευματοκιβώτιο με συνέπεια να παρέχεται σε αυτά η δυνατότητα της προσέγγισης και σε λιμένες με ελλιπή παρόμοια μέσα. Γενικά με τον όρο λιμένας νοείται ο ασφαλής εκείνος όρμος στον οποίο μπορούν να προσεγγίζουν με ασφάλεια τα πλοία προκειμένου να προβούν σε εμπορικές πράξεις. Διακρίνονται δε σε φυσικούς λιμένες και σε τεχνητούς λιμένες), και ακόμη σε θαλάσσιους ή παραθαλάσσιους, ποτάμιους ή παραποτάμιους και σε λιμναίους ή παραλίμνιους. Οι λιμένες που προσεγγίζουν αποκλειστικά πολεμικά πλοία και εξυπηρετούνται αποκλειστικά αυτά ονομάζονται ναύσταθμοι ή ναυτικές βάσεις

Στην Εικόνα 1 παρουσιάζονται τα διάφορα υποσυστήματα διακίνησης εμπορευματοκιβωτίων σε έναν επιλιμένο τερματικό σταθμό εμπορευματοκιβωτίων.



Εικόνα 1 Υποσυστήματα διακίνησης εμπορευματοκιβωτίων σε έναν επιλιμένο τερματικό σταθμό εμπορευματοκιβωτίων

Το υποσύστημα διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου μπορεί να οργανωθεί με περισσότερους από έναν τρόπους ανάλογα και με τον τύπο του χρησιμοποιούμενου εξοπλισμού. Υπάρχουν ποικίλοι τύποι εξοπλισμού και ποικίλοι τρόποι διαμόρφωσης που μπορούν να εφαρμοστούν.

Οι τεχνικές οργάνωσης που μελετώνται αφορούν στη σταθερή ανάθεση εξυπηρετητών σε γερανογέφυρες του ίδιου πλοίου (fixed allocation), στην ανάθεση ομάδας εξυπηρετητών σε γερανογέφυρες του ίδιου πλοίου (pooled allocation to one ship) και στην ανάθεση ομάδας εξυπηρετητών σε γερανογέφυρες διαφορετικών πλοίων (pooled allocation to many ships).

Η σημαντικότητα των διαδικασιών μεταφοράς στο υποσύστημα κρηπίδωμα – χώρος στοιβάσεως πηγάζει από το γεγονός ότι τυχόν ανεπάρκειες και καθυστερήσεις στην εξυπηρέτηση των γερανογεφυρών στο κρηπίδωμα επηρεάζουν αρνητικά την απόδοση όλου του συστήματος στη θαλάσσια πλευρά.

1.2 Αντικείμενο και στόχος της διπλωματικής εργασίας

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι η διερεύνηση εναλλακτικών τεχνικών οργάνωσης στο υποσύστημα διακίνησης εμπορευματοκιβωτίων εντός λιμενικών τερματικών σταθμών. Επικεντρώνεται στις τεχνικές οργάνωσης του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου. Χρησιμοποιείται η μέθοδος της προσομοίωσης και συμπεριλαμβάνεται μια βιβλιογραφική επισκόπηση πάνω σε μοντέλα που έχουν αναπτυχθεί στο παρελθόν και έχουν συμβάλει στη διερεύνηση των τεχνικών οργάνωσης του υπό μελέτη υποσυστήματος. Για την ανάπτυξη και την εφαρμογή μοντέλων προσομοίωσης χρησιμοποιείται το λογισμικό προσομοίωσης Anylogic.

Σκοπός της παρούσας διπλωματικής εργασίας είναι να διερευνηθούν εναλλακτικές τεχνικές οργάνωσης στο υποσύστημα κρηπίδωμα – χώρος στοιβάσεως με την βοήθεια μοντέλων προσομοίωσης και συγκεκριμένα του λογισμικού Anylogic. Ακόλουθο της διερεύνησης είναι ανάπτυξη μοντέλων προσομοίωσης ώστε να αξιολογούνται οι εναλλακτικές τεχνικές οργάνωσης και να είναι δυνατό να αποφασιστεί η αποτελεσματικότερη στρατηγική ανάθεσης εργασιών που

πρέπει να εφαρμοστεί υπό προϋποθέσεις. Επίσης, σκοπός της παρούσας διπλωματικής εργασίας είναι η κατασκευή ενός σύνθετου μοντέλου προσομοίωσης του επιλιμένιου σιδηροδρομικού σταθμού στον λιμένα του Πειραιά και η διατύπωση σεναρίων ώστε να μελετηθεί η λειτουργία του υποσυστήματος «κρηπιδώματος - χερσαίου χώρου» σε σχέση με τον απαιτούμενο αριθμό των οχημάτων μεταφοράς εμπορευματοκιβωτίων.

1.3 Δομή διπλωματικής εργασίας

Το κεφάλαιο 1 είναι εισαγωγικό και έχει ως σκοπό να δώσει στον αναγνώστη τη δυνατότητα να σχηματίσει μια εικόνα για το αντικείμενο της παρούσας διπλωματικής εργασίας. Στο υποκεφάλαιο 1.1 παρουσιάζονται εισαγωγικά στοιχεία σχετικά με το σύστημα διακίνησης των εμπορευματοκιβωτίων. Στο υποκεφάλαιο 1.2 γίνεται αναφορά στο αντικείμενο της διπλωματικής εργασίας και παρουσιάζεται ο σκοπός της. Τέλος στο υποκεφάλαιο αυτό ολοκληρώνεται η εισαγωγή με την παρουσίαση της δομής της διπλωματικής εργασίας.

Στο κεφάλαιο 2 γίνεται αναφορά στα πλοία μεταφοράς εμπορευματοκιβωτίων και στους λιμένες διακίνησης. Στο υποκεφάλαιο 2.1 παρουσιάζονται αναλυτικά τα πλοία μεταφοράς και η εξέλιξη τους, ενώ στο 2.2 υπάρχει αναφορά στον λιμένα και στο εξοπλισμό του.

Στο κεφάλαιο 3 λαμβάνει μέρος η βιβλιογραφική επισκόπηση. Αρχικά στο υποκεφάλαιο 3.1 παρουσιάζονται οι μέθοδοι επίλυσης που χρησιμοποιούνται κατά βάση για την επίλυση προβλημάτων που αφορούν τις λιμενικές εγκαταστάσεις, προκειμένου ο αναγνώστης να έχει το απαιτούμενο θεωρητικό υπόβαθρο για την κατανόηση των εννοιών που περιγράφονται στο υπόλοιπο κεφάλαιο. Στη συνέχεια, κάνοντας ένα χωρικό διαχωρισμό του λιμένα, έγινε καταμερισμός των υποπροβλημάτων, προκειμένου να επιτευχτεί μεγαλύτερη κατανόηση. Τέλος στο κεφάλαιο 3.6, έγινε μια συνολική προσαρμογή των υποπροβλημάτων αυτός στο συνολικό πρόβλημα προκειμένου να βρεθεί η βέλτιστη λύση.

Στο κεφάλαιο 4 πραγματοποιείται ανάπτυξη μοντέλων προσομοίωσης λιμενικών εγκαταστάσεων στο λογισμικό Anylogic. Αρχικά στο υποκεφάλαιο 4.1 παρουσιάζεται η μέθοδος της προσομοίωσης και αναλύεται το λογισμικό προσομοίωσης Anylogic. Στο υποκεφάλαιο 4.2 αναπτύσσονται τα μοντέλα λιμενικών εγκαταστάσεων και πραγματοποιείται έλεγχος εγκυρότητάς τους σε σχέση με αντίστοιχα μοντέλα της θεωρίας αναμονής.

Στο κεφάλαιο 5 παρουσιάζεται η προσομοίωση των στρατηγικών ανάθεσης εργασιών στα οχήματα μεταφοράς κατά τις διαδικασίες φόρτωσης/εκφόρτωσης πλοίων. Στο υποκεφάλαιο 5.1 γίνεται προσομοίωση στρατηγικών ανάθεσης εργασιών στα οχήματα μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων. Στο υποκεφάλαιο 5.2 γίνεται προσομοίωση στρατηγικών ανάθεσης εργασιών στα οχήματα μεταφοράς με σταθερούς χρόνους κύκλου, κατά τις διαδικασίες φόρτωσης και εκφόρτωσης πλοίων, ενώ στο υποκεφάλαιο 5.3 παρουσιάζονται τα αποτελέσματα των σεναρίων αυτών. Στο υποκεφάλαιο 5.4 και 5.5 αναπτύσσονται δύο νέα σενάρια, το σενάριο τρία και το τέσσερα, προκειμένου να γίνει η σύγκρισή τους στο υποκεφάλαιο 5.6. Στα σενάρια αυτά πραγματοποιείται προσομοίωση που λαμβάνει υπόψη αποκλίσεις στους χρόνους κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς. Η σύγκριση αφορά τη σταθερή ανάθεση (σενάριο τρία) με την κυκλική ανάθεση (σενάριο τέσσερα) κατά την αλλαγή της απόστασης μεταξύ των γερανογεφυρών. Στο υποκεφάλαιο 5.7 παρουσιάζεται μέθοδος που εκτιμά την απόκλιση του χρόνου κύκλου των οχημάτων μεταφοράς

που πρέπει να ληφθεί υπόψη σε κάθε περίπτωση, ενώ στη συνέχεια ακολουθούν το υποκεφάλαιο 5.8, όπου περιλαμβάνει το σενάριο πέντε και εφαρμόζεται σταθερή ανάθεση με μεταβλητούς χρόνους κύκλου των μηχανημάτων, το υποκεφάλαιο 5.9, όπου περιλαμβάνεται το σενάριο έξι και εφαρμόζεται ανάθεση ομάδας με μεταβλητούς χρόνους κύκλου των μηχανημάτων και το υποκεφάλαιο 5.10 όπου αναφέρονται τα αποτελέσματα της σύγκρισης αυτής.

Στο κεφάλαιο 6, γίνεται αρχικά η παρουσίαση ενός μοντέλου που αναπαριστά το σύνολο των λειτουργιών ενός λιμένα, ενώ ακολούθως εφαρμόζονται σενάρια προκειμένου να γίνει διερεύνηση ως προς τον χρόνο εξυπηρέτησης των πλοίων υπό συγκεκριμένες συνθήκες.

2 ΠΛΟΙΑ ΜΕΤΑΦΟΡΑΣ ΚΑΙ ΛΙΜΕΝΕΣ ΔΙΑΚΙΝΗΣΗΣ ΕΜΜΠΟΡΕΥΜΑΤΟΚΙΒΩΤΙΩΝ

2.1 Πλοία μεταφοράς εμπορευματοκιβωτίων

Η τεράστια βιομηχανική ανάπτυξη οδήγησε σε μια γενικότερη εξέλιξη των πλοίων ώστε να εξυπηρετούνται καλύτερα. Ανάλογα λοιπόν με την εξέλιξη τους κατηγοριοποιούνται ως εξής σύμφωνα με την Μαλιάγκα, Ε. 2014.

Πλοία 1ης γενιάς (Early Containerships)

Η πρώτη γενιά των πλοίων μεταφοράς εμπορευματοκιβωτίων αποτελούνταν από τροποποιημένα πλοία μεταφοράς ξηρού φορτίου χύδην ή δεξαμενόπλοια και μετέφεραν έως 1000 TEU. Το πρώτο πλοίο μεταφοράς εμπορευματοκιβωτίων ήταν το Ideal – X που ήταν ένα κατάλληλα τροποποιημένο δεξαμενόπλοιο του Β' Παγκοσμίου Πολέμου. Στις αρχές της δεκαετίας του 1960 το πλοίο μεταφοράς εμπορευματοκιβωτίων δεν αποτελούσε δοκιμασμένο μέσο μεταφορών και έτσι η τροποποίηση των υπάρχοντων πλοίων ήταν η λιγότερο δαπανηρή λύση καθώς τα πλοία αυτά είχαν ίδια μέσα φορτοεκφόρτωσης ενώ οι περισσότεροι λιμενικοί τερματικοί σταθμοί δεν ήταν κατάλληλα εξοπλισμένοι για να χειριστούν τα εμπορευματοκιβώτια. Όμως, είχαν ένα μεγάλο μειονέκτημα, που ήταν η δυνατότητα μεταφοράς εμπορευματοκιβωτίων μόνο πάνω στο κατάλληλα διαμορφωμένο κατάστρωμα τους και όχι στους χώρους κυτών.

Πλοία 2ης γενιάς (Fully Cellular)

Στις αρχές της δεκαετίας του 1970, όταν το πλοίο μεταφοράς εμπορευματοκιβωτίων άρχισε να χρησιμοποιείται περισσότερο στις θαλάσσιες μεταφορές και η μεταφορική τους ικανότητα ήταν της τάξης των 2000 TEU κατασκευάστηκαν τα πρώτα πλοία μεταφοράς εμπορευματοκιβωτίων πλήρως κυψελοειδούς δομής (Fully Cellular Containerships, FCC). Τα πλοία αυτού του τύπου διαθέτουν κυψέλες στις οποίες τοποθετούνται τα εμπορευματοκιβώτια σε στοίβες διαφορετικού ύψους, ανάλογα με τη χωρητικότητα του πλοίου, και υπάρχει το πλεονέκτημα φόρτωσης του πλοίου με εμπορευματοκιβώτια πάνω και κάτω από το κύριο κατάστρωμα. Παράλληλα, τα πλοία αυτά δεν έχουν γερανούς, έτσι ώστε να υπάρχει η δυνατότητα τοποθέτησης περισσότερων εμπορευματοκιβωτίων πάνω στο κύριο κατάστρωμα. Αυτό είχε ως συνέπεια τη δημιουργία εξειδικευμένων τερματικών σταθμών εμπορευματοκιβωτίων σε όλο τον κόσμο.

Πλοία 3ης γενιάς (Κατηγορία Β) (Panamax)

Οι οικονομίες κλίμακας οδήγησαν στη ναυπήγηση μεγαλύτερων πλοίων μεταφοράς εμπορευματοκιβωτίων τη δεκαετία του 1980. Έτσι, όσο μεγαλύτερος είναι ο αριθμός των εμπορευματοκιβωτίων που μεταφέρονται τόσο χαμηλότερο είναι το κόστος ανά TEU γεγονός που οδήγησε στην σχεδίαση πλοίων με χωρητικότητες της τάξης των 3000 TEU.

Πλοία 4ης γενιάς (Κατηγορία Β) (Panamax Max)

Το όριο μεγέθους της Διώρυγας του Παναμά οδήγησε στην κατασκευή πλοίων μεταφοράς εμπορευματοκιβωτίων χωρητικότητας περίπου 4000 TEU το 1985. Ο σχεδιασμός των πλοίων μεταφοράς εμπορευματοκιβωτίων ώστε να ικανοποιούν τους περιορισμούς αυτούς, έχει ως αποτέλεσμα ένα στενό και μακρύ σχεδιασμό του πλοίου.

Πλοία 5ης γενιάς (Κατηγορία C) (Post Panamax)

Το 1988 κατασκευάστηκαν τα πλοία μεταφοράς εμπορευματοκιβωτίων APL C10 που ήταν η πρώτη κατηγορία πλοίων που υπερέβη το όριο των 32,2m, πλάτος της Διώρυγας του Παναμά. Μέχρι το 1996 τα πλοία μεταφοράς εμπορευματοκιβωτίων Post Panamax είχαν μεταφορική ικανότητα που έφτανε τα 6.600 TEU.

Πλοία 6ης γενιάς (Κατηγορία C) (Post Panamax Plus)

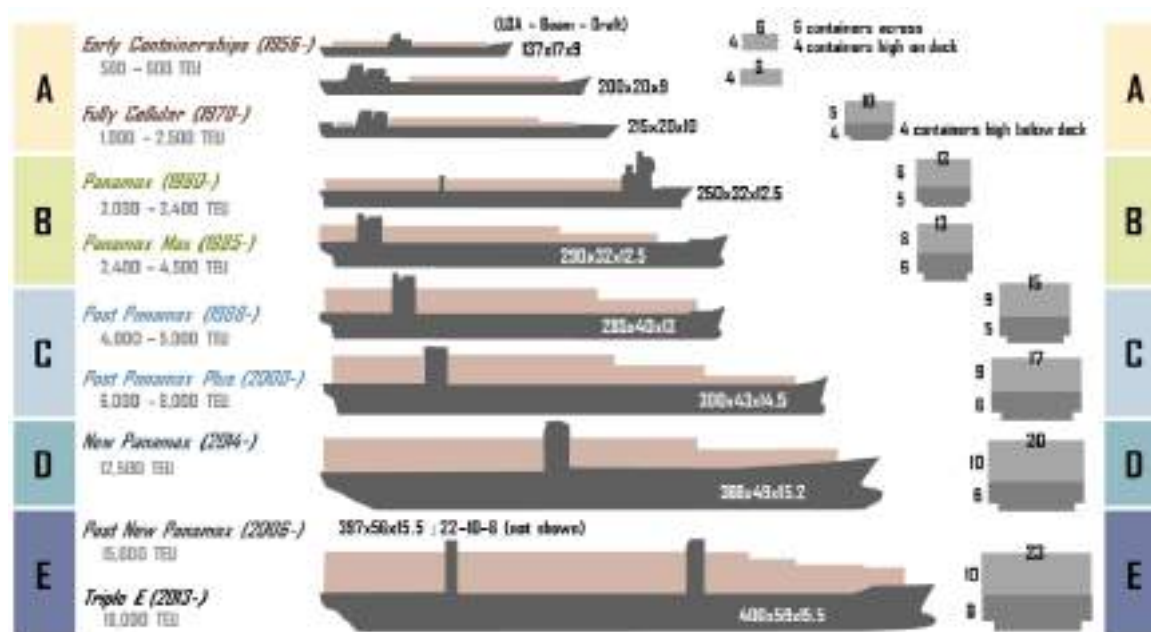
Το μέγεθος των πλοίων 6ης Γενιάς αφορά χωρητικότητες που φτάνουν τα 8.000 TEU. Τα πλοία αυτού του τύπου απαιτούν λιμάνια με μεγάλο βάθος νερού και κατάλληλο εξοπλισμό φορτοεκφόρτωσης και ξεπερνούν τους περιορισμούς της Διώρυγας του Παναμά.

New Panamax (Κατηγορία D)

Τα πλοία αυτά έχουν σχεδιαστεί για να ικανοποιούν τους περιορισμούς της διευρυμένης διώρυγας του Παναμά, και έχουν χωρητικότητες της τάξης των 12500 TEU. Όπως τα πλοία Panamax έτσι και τα πλοία New Panamax θα συνθέτουν μια συγκεκριμένη κατηγορία πλοίου που θα εξυπηρετεί αποτελεσματικά την Αμερική και την Καραϊβική, είτε από την Ευρώπη είτε από την Ασία.

Post New Panamax (Κατηγορία E)

Το 2006 παρουσιάστηκε μια νέα γενιά πλοίων μεταφοράς εμπορευματοκιβωτίων. όταν η ναυτιλιακή εταιρεία Maersk παρουσίασε τα πλοία Emma Maersk E Class με χωρητικότητα που κυμαίνεται από 11000 έως 14500 TEU. Τα πλοία αυτού του τύπου είναι μεγαλύτερα σε διαστάσεις σε σχέση με τις προδιαγραφές της διευρυμένης Διώρυγας του Παναμά και μπορούν να μεταφέρουν μέχρι και 18.000 TEU (Triple E Class).



Εικόνα 2 Η εξέλιξη των πλοίων μεταφοράς εμπορευματοκιβωτίων [Πηγή: Μαλιάγκα, Ε. (2014)]

2.2 Μηχανολογικός εξοπλισμός λιμένα

Ο μηχανολογικός εξοπλισμός που χρησιμοποιείται στις λιμενικές εγκαταστάσεις μπορεί να διαχωριστεί στον εξοπλισμό που υπάρχει στη θαλάσσια πλευρά, στον εξοπλισμό που χρησιμοποιείται για τη μεταφορά των εμπορευματοκιβωτίων από τη θαλάσσια στη χερσαία πλευρά και στον εξοπλισμό που χρησιμοποιείται στη χερσαία πλευρά σύμφωνα με την *Θάνου, Ε. 2015*.

2.2.1 Εξοπλισμός θαλάσσιας πλευράς

Στη θαλάσσια πλευρά χρησιμοποιούνται οι γερανογέφυρες κρηπιδώματος (quay or gantry cranes, QC) για την φόρτωση και την εκφόρτωση του πλοίων. Οι γερανογέφυρες κρηπιδώματος είναι μεταλλικές πλασιωτές κατασκευές που κινούνται πάνω σε σιδηροτροχιές και παρατάσσονται ή μια δίπλα στην άλλη σε απόσταση περίπου 15 μέτρα, εγκάρσια στο πλοίο. Οι βασικοί τύποι που έχουν διαμορφωθεί με βάση τα πλοία που είναι σε θέση να εξυπηρετήσουν (Εικόνα 3) είναι οι εξής:

Μήκος έκτασης του βραχίονα προς τη θάλασσα:

Small Panamax κάτω από 36 μέτρα,

Standard Panamax 36-44 μέτρα,

Post Panamax 44-48 μέτρα,

Super Post Panamax 48 μέτρα και πάνω.

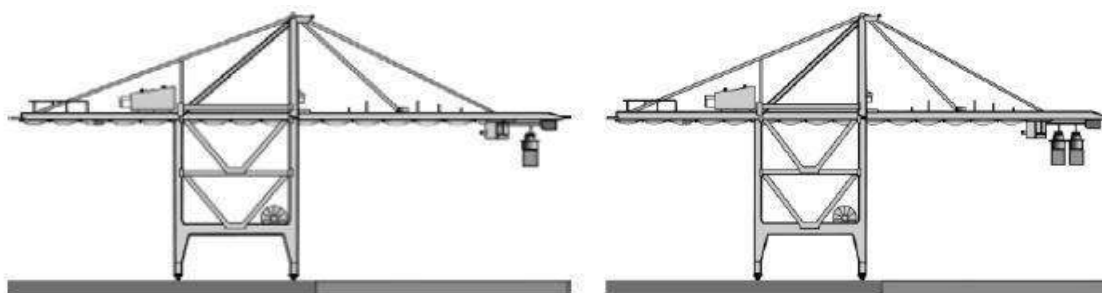


Εικόνα 3 Εξέλιξη γερανών θαλάσσιας πλευράς [Πηγή: Edmonds, R. (2010)]

Μπορούν να διακριθούν δύο είδη γερανογεφυρών κρηπιδώματος, οι γερανογέφυρες με μονό βαγονέτο (single-trolley cranes) και οι γερανογέφυρες με διπλό βαγονέτο (dual-trolley cranes)

(Εικόνα 4). Τα βαγονέτα κινούνται μαζί με τις γερανογέφυρες και είναι εξοπλισμένα με ειδικές κλίμακες για τη συλλογή των ε/κ. Οι σύγχρονες κλίμακες επιτρέπουν την κίνηση 2 εμπορευματοκιβωτίων 20 ποδών ταυτόχρονα. Συνήθως στους ΣΕΜΠΟ υπάρχουν γερανογέφυρες μονού βαγονέτου. Μεταφέρουν τα εμπορευματοκιβώτια από το πλοίο και τα τοποθετούν στην αποβάθρα ή σε κάποιο όχημα (αντίστροφη διαδικασία γίνεται στον κύκλο φόρτωσης του πλοίου). Οι γερανογέφυρες με ένα βαγονέτο είναι χειροκίνητες. Οι γερανογέφυρες διπλού βαγονέτου αποτελούν καινοτομία και χρησιμοποιούνται σε ορισμένους σταθμούς. Στις σύγχρονες γερανογέφυρες και στους δύο τύπους, ο οδηγός υποστηρίζεται με ένα ημιαυτόματο οδηγητικό σύστημα.

Η επίδοση των γερανογεφυρών είναι συνάρτηση των χειρισμών που πραγματοποιούν σε μία ώρα. Οι συμβατικού τύπου γερανογέφυρες εκτελούν περίπου 20 κινήσεις/ώρα, δηλαδή χρειάζονται τρία λεπτά για κάθε εμπορευματοκιβώτιο. Οι εξειδικευμένες γερανογέφυρες εκτελούν περίπου 30 κινήσεις/ώρα, δηλαδή 2 λεπτά για κάθε εμπορευματοκιβώτιο και τέλος οι γερανογέφυρες υψηλής τεχνολογίας οι οποίες διαθέτουν διπλό βαγονέτο εκτελούν περίπου 60 κινήσεις/ώρα, δηλαδή ένα λεπτό για κάθε εμπορευματοκιβώτιο.



Εικόνα 4 Γερανογέφυρα με μονό βαγονέτο (αριστερό τμήμα) και Γερανογέφυρα με διπλό βαγονέτο (δεξιό τμήμα)
[Πηγή: Bartošek, A., Marek, O. (2013)]

2.2.2 Εξοπλισμός μεταφοράς εμπορευματοκιβωτίων μεταξύ θαλάσσιας και χερσαίας πλευράς

Ο εξοπλισμός που χρησιμοποιείται για τη μεταφορά των εμπορευματοκιβωτίων, εκτελεί τη διαδρομή από το κρηπίδωμα στους χώρους προσωρινής αποθήκευσης όταν πρόκειται για εκφόρτωση πλοίου και την αντίστροφη διαδρομή όταν πρόκειται για φόρτωση πλοίου. Ο εξοπλισμός αυτός μπορεί να είναι:

Ελκυστήρες με συρόμενες βάσεις

Οι ελκυστήρες με συρόμενες βάσεις (terminal tractors) (Εικόνα 5) είναι γνωστά και ως φορτηγά[14] (yard trucks) και μεταφέρουν τα εμπορευματοκιβώτια χωρίς όμως να μπορούν να τα ανυψώσουν. Αποτελούνται από την καμπίνα για τον οδηγό και από την βάση στην οποία τοποθετούνται απευθείας τα εμπορευματοκιβώτια. Χρησιμοποιούνται σε Lo-Lo (Lift on – Lift off) μετακινήσεις εντός των τερματικών σταθμών και σε Ro-Ro (Roll on – Roll off) διαδικασίες φορτοεκφόρτωσης αφού μπορούν να συνδέσουν στη βάση τους και κινητά αμαξώματα. Έχουν χαμηλά λειτουργικά κόστη και παρέχουν ασφάλεια και άνεση. Θεωρητικά η ταχύτητα τους

μπορεί να φτάσει τα 53km/h, ωστόσο δεν αναπτύσσουν ταχύτητες μεγαλύτερες των 25km/h στο χώρο τους λιμανιού, λόγω κανονισμού μέγιστης επιτρεπόμενης ταχύτητας.



Εικόνα 5 Ελκυστήρες με συρόμενες βάσεις [Πηγή: Kalmar, <http://www.kalmarglobal.com/equipment/>, τελευταία πρόσβαση 10/9/2015.]

Ελκυστήρες με δυνατότητα ρυμούλκησης πολλών τρέιλερ

Οι ελκυστήρες που έχουν τη δυνατότητα ρυμούλκησης πολλών τρέιλερ (multi – Trailers) (Εικόνα 6) έχουν πολλά πλεονεκτήματα έναντι των ελκυστήρων που μεταφέρουν ένα εμπορευματοκιβώτιο τη φορά. Συγκεκριμένα, μειώνουν τις συνολικές κινήσεις στον τερματικό σταθμό, μειώνουν τις καθυστερήσεις των γερανογεφυρών, μειώνουν το συνολικό κόστος επενδύσεων για εξοπλισμό, μειώνουν το κόστος εργασίας, το κόστος καυσίμων και κατ' επέκταση το κόστος συντήρησης. Αποτελούνται από τρία τμήματα όπου κάθε τμήμα μπορεί να μεταφέρει δύο εμπορευματοκιβώτια 20' ή ένα εμπορευματοκιβώτιο 40'.



Εικόνα 6 Ελκυστήρες με δυνατότητα ρυμούλκησης πολλών τρέιλερ [Πηγή: Buiscar, Cargo Solutions, <http://www.buiscar.com/mts.htm>, τελευταία πρόσβαση 10/9/2015.]

Οχήματα πλαίσια

Τα οχήματα πλαίσια (straddle carriers) (Εικόνα 7) είναι τα σημαντικότερα μέσα μεταφοράς γιατί έχουν την δυνατότητα όχι μόνο να μεταφέρουν εμπορευματοκιβώτια αλλά και να τα στοιβάζουν. Μπορούν να θεωρηθούν ως 'γερανοί' με την ευρεία έννοια, αφού έχουν ελεύθερη πρόσβαση στα εμπορευματοκιβώτια, ανεξάρτητα από τη θέση τους στο χώρο. Τα οχήματα πλαίσια επιτρέπουν τη μεταφορά εμπορευματοκιβωτίων 20' ή 40' και είναι ικανά για ταυτόχρονη μεταφορά και στοιβάση δύο εμπορευματοκιβωτίων 20'. Εξαιτίας των εξαρτημάτων τους, είναι πολύ ευέλικτα και δυναμικά και υπάρχουν σε μεγάλη ποικιλία. Συνήθως είναι χειροκίνητα και ικανά να στοιβάζουν δύο ή τρία εμπορευματοκιβώτια σε ύψος, και να μετακινήσουν ένα εμπορευματοκιβώτιο πάνω από δύο ή τρία άλλα. Η ταχύτητα που μπορούν να αναπτύξουν είναι μεταξύ 24-30km/h.



Εικόνα 7 Οχήματα πλαίσια [Πηγή: Kalmar, <http://www.kalmarglobal.com/equipment.>]

Αυτομάτως καθοδηγούμενα οχήματα μεταφοράς

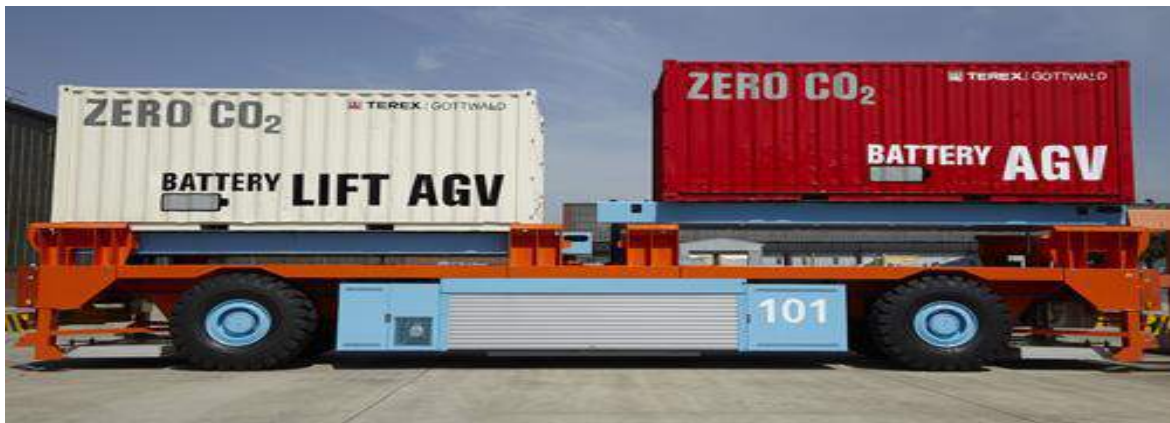
Τα αυτομάτως καθοδηγούμενα οχήματα μεταφοράς (Automatic Guided Vehicle, AGV) (Εικόνα 7) είναι ικανά να κινούνται σε οδικό δίκτυο που υπάρχουν ηλεκτρικές γραμμές ή ράγες μεταφοράς στο έδαφος για να ελέγχουν τη θέση τους. Κάθε AGV μπορεί να φορτώσει ένα 40'/45' εμπορευματοκιβώτιο ή δύο 20' εμπορευματοκιβώτια, είναι δηλαδή ικανό για πιθανές λειτουργίες πολλαπλής φόρτωσης. Έχουν μήκος περίπου 14,8m, πλάτος 3m και ύψος πλατφόρμας 1,9m περίπου. Η μέγιστη ταχύτητα που μπορούν να αναπτύξουν είναι 6m/sec στις ευθυγραμμίες, ενώ στις καμπύλες φτάνουν μέχρι τα 3m/sec. Τα συστήματα με AGV απαιτούν εγκαταστάσεις υψηλού κόστους και συνεπώς η λειτουργία τέτοιων συστημάτων είναι συμφέρουσα σε περιοχές όπου το εργατικό κόστος είναι υψηλό. Σήμερα, είναι σε λειτουργία στο Ρότερνταμ και στο Αμβούργο, σε συνδυασμό με αυτόματες γερανογέφυρες.



Εικόνα 8 Αυτομάτως καθοδηγούμενα οχήματα μεταφοράς [Πηγή: Terex, <http://www.terex.com/port-solutions/en/>.]

Αυτοματοποιημένα οχήματα με δυνατότητα ανύψωσης

Σε αντίθεση με τα συμβατικά αυτομάτως καθοδηγούμενα οχήματα μεταφοράς (AGVs), αυτοματοποιημένα οχήματα με δυνατότητα ανύψωσης (Automated Lifting Vehicle, ALV) (Εικόνα 9) έχουν δύο ενεργές πλατφόρμες ανύψωσης. Μπορούν να μεταφέρουν δύο εμπορευματοκιβώτια 20' ανεξάρτητα ή ένα εμπορευματοκιβώτιο 40'. Έχουν μήκος 14,8m, πλάτος 3m και ύψος πλατφόρμας 2,4m περίπου. Η μέγιστη ταχύτητα που μπορούν να αναπτύξουν είναι 6m/sec στις ευθυγραμμίες, ενώ στις καμπύλες φτάνουν μέχρι τα 3m/sec, όμοια με τα αυτομάτως καθοδηγούμενα οχήματα μεταφοράς.



Εικόνα 9 Αυτοματοποιημένα οχήματα με δυνατότητα ανύψωσης [Πηγή: Terex, <http://www.terex.com/port-solutions/en/>.]

2.2.3 Εξοπλισμός χερσαίας πλευράς

Ο εξοπλισμός της χερσαίας πλευράς χρησιμοποιείται για την μεταφορά και την στοιβάδα των εμπορευματοκιβωτίων στους χώρους αποθήκευσης. Ο εξοπλισμός αυτός μπορεί να είναι:

Περονοφόρα ανυψωτικά οχήματα

Τα περονοφόρα ανυψωτικά οχήματα (forklift trucks) (Εικόνα 10) είναι γρήγορα, εύχρηστα και ιδιαίτερα ευέλικτα. Είναι απλά στη λειτουργία τους και ασφαλή εξαιτίας της στιβαρής κατασκευής τους με χαμηλό κέντρο βάρους. Ο χειριστής έχει πολύ καλή εποπτεία του χώρου όμως το μεγάλο μειονέκτημα τους είναι ότι δεν μπορούν να τοποθετήσουν φορτία σε μεγάλο ύψος.



Εικόνα 10 Περονοφόρα ανυψωτικά οχήματα [Πηγή: Kalmar, <http://www.kalmarglobal.com/equipment>]

Οχήματα εμπρόσθιας στοιβασίας

Τα οχήματα εμπρόσθιας στοιβασίας (reach stackers) (Εικόνα 11) έχουν ευρεία χρήση στη μεταφορά και στοιβασία εμπορευματοκιβωτίων σε λιμενικούς τερματικούς σταθμούς, σε σιδηροδρομικούς σταθμούς και σε άλλα συστήματα μεταφορών. Είναι εύκολα στο χειρισμό, ασφαλή και έχουν υψηλή ικανότητα στο χειρισμό των φορτίων και υψηλές ταχύτητες εργασίας με μέγιστη τα 25km/hr.



Εικόνα 11 Οχήματα εμπρόσθιας στοιβασίας [Πηγή: Kalmar, <http://www.kalmarglobal.com/equipment>.]

Γερανογέφυρες επί σιδηροτροχιών

Οι γερανογέφυρες επί σιδηροτροχιών (Rail Mounted Gantry Cranes, RMG) (Εικόνα 12) είναι πλαισιωτές μεταλλικές κατασκευές εξοπλισμένες με ανυψωτικό μηχανισμό και αρπάγη που χρησιμοποιούνται για την στοιβάση των εμπορευματοκιβωτίων. Έχουν πλάτος που φτάνει τα 50m και μπορούν να στοιβάζουν εμπορευματοκιβώτια μέχρι και σε 10 σειρές κατά τον άξονα τους.

Συνήθως στοιβάζουν από 3 έως 9 εμπορευματοκιβώτια καθ' ύψος. Η κατά μήκος ταχύτητα τους φτάνει μέχρι τα 100m/min όταν είναι σε έμφορτη κατάσταση ενώ μπορεί να φτάσει τα 120m/min όταν είναι κενές. Η ταχύτητα ανύψωσης των εμπορευματοκιβωτίων φτάνει μέχρι τα 40m/min. Τέλος έχουν ανυψωτική ικανότητα μέχρι 65 τόνους.



Εικόνα 12 Γερανογέφυρες επί σιδηροτροχιών [Πηγή: KoneCranes, <http://www.konecranes.com>]

Γερανογέφυρες επί ελαστικών τροχών

Οι ελαστικοφόρες γερανογέφυρες (Rail Tired Gantries, RTG) (Εικόνα 13) είναι πιο ευέλικτες λειτουργικά ενώ οι γερανογέφυρες επί σιδηροτροχιών είναι πιο σταθερές. Οι γερανογέφυρες αυτές έχουν σχεδιαστεί για εξοικονόμηση χώρου αποθήκευσης εμπορευματοκιβωτίων και χρησιμοποιούνται μεσαίου μεγέθους τερματικούς σταθμούς. Έχουν πλάτος που φτάνουν τα 36,50m και μπορούν να στοιβάζουν εμπορευματοκιβώτια μέχρι και σε 8 σειρές κατά τον άξονα τους.

Συνήθως στοιβάζουν από 3 έως 7 εμπορευματοκιβώτια καθ' ύψος. Η κατά μήκος ταχύτητα τους φτάνει μέχρι τα 100m/min όταν είναι σε έμφορτη κατάσταση ενώ μπορεί να φτάσει τα 120m/min όταν είναι κενές. Η ταχύτητα ανύψωσης των εμπορευματοκιβωτίων φτάνει μέχρι τα 30m/min. Τέλος έχουν ανυψωτική ικανότητα μέχρι 65 τόνους.



Εικόνα 13 Γερανογέφυρες επί ελαστικών τροχών [Πηγή: KoneCranes, <http://www.konecranes.com>]

3 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ

3.1 Περιγραφή των βασικών μεθόδων και αλγορίθμων επίλυσης προβλημάτων στις λιμενικές εγκαταστάσεις

Για την επίλυση των προβλημάτων στις λιμενικές εγκαταστάσεις χρησιμοποιούνται τεχνικές που έχουν ως βάση μαθηματικά μοντέλα. Οι τεχνικές αυτές αποτελούν αντικείμενο του τομέα της Επιχειρησιακής Έρευνας (Operation Research) και λειτουργούν ως εργαλείο υποβοήθησης για τη λήψη ορθολογικότερων αποφάσεων τόσο κατά τη διάρκεια του σχεδιασμού και της κατασκευής όσο και της λειτουργίας.

Οι συνήθεις τύποι μαθηματικών μοντέλων που χρησιμοποιούνται μπορούν να διαχωριστούν στα περιγραφικά μοντέλα (descriptive models), που προσφέρουν για ένα συγκεκριμένο αριθμό δεδομένων και αρχικών συνθηκών, μια περιγραφή του φαινομένου που μελετάται και στα καθοριστικά μοντέλα (prescriptive models), στα οποία η στρατηγική και ο σχεδιασμός που προτείνουν, πετυχαίνουν τη βέλτιστη τιμή ενός αντικειμένου. Επίσης με βάση το είδος των δεδομένων που χρησιμοποιούν διαχωρίζονται στα προσδιοριστικά μοντέλα (deterministic models) και στα στοχαστικά μοντέλα (stochastic models). Η Εικόνα 14 παρουσιάζει συνοπτικά τις κατηγορίες αυτές, με τα βασικά μαθηματικά μοντέλα που χρησιμοποιούνται στις μέρες μας.

	Προσδιοριστικά μοντέλα (Deterministic models)	Στοχαστικά μοντέλα (Stochastic models)
Καθοριστικά μοντέλα (Prescriptive models)	▪ Γραμμικός προγραμματισμός ▪ Ακέραιος προγραμματισμός ▪ Πολυστοχικός προγραμματισμός ▪ Προβλήματα δικτύων ▪ Δυναμικός προγραμματισμός	▪ Στοχαστικός προγραμματισμός
Περιγραφικά μοντέλα (Descriptive models)	▪ Διαφορικός λογισμός ▪ Διαφορικές εξισώσεις	▪ Στοχαστικές διαφορικές εξισώσεις ▪ Θεωρία αναμονής ▪ Προσομοίωση

Εικόνα 14 Προσδιοριστικά & στοχαστικά μοντέλα [Πηγή: Καρλαύτης Μ.Γ., Λαγαρός Ν.Δ. (2010),]

Στα προσδιοριστικά μοντέλα (deterministic models) οι τιμές των δεδομένων είναι σταθερές ποσότητες, έχουν δηλαδή καθορισμένες τιμές και στα στοχαστικά μοντέλα (stochastic models) οι τιμές των δεδομένων ακολουθούν κάποια στατιστική κατανομή. Σε ένα πραγματικό σύστημα είναι πρακτικά αδύνατο όλα τα δεδομένα να έχουν σταθερά προκαθορισμένη τιμή. Γι αυτό η προσέγγιση μιας στατιστικής κατανομής με μέση τιμή και τυπική απόκλιση μπορεί να προσφέρει ένα ικανοποιητικό αποτέλεσμα σε αρκετές περιπτώσεις.

Στα πλαίσια λοιπόν της Επιχειρησιακής Έρευνας, σχεδιάζονται και επιλύονται συστήματα με τεχνικές τέτοιες ώστε να δίνουν το βέλτιστο αποτέλεσμα. Οι τεχνικές που χρησιμοποιούνται στην βιβλιογραφία και βασίζονται στις αρχές του μαθηματικού προγραμματισμού (mathematical programming) ταξινομούνται γενικά σε πέντε μεγάλες κατηγορίες:

1. Γραμμικός προγραμματισμός (Linear Programming – LP)

Το μαθηματικό πρότυπο του γραμμικού προγραμματισμού επιλύει το πρόβλημα βέλτιστης κατανομής πόρων κάτω από περιορισμούς (constraints) που είναι γραμμικοί όροι. Ο γραμμικός προγραμματισμός αποτελεί ένα χρήσιμο εργαλείο για την αποτελεσματική αντιμετώπιση διαφόρων πολύπλοκων προβλημάτων αποφάσεων. Η ποιότητα των αποφάσεων εξαρτάται από την ακρίβεια της περιγραφής της κατάστασης που μελετάται, καθώς και από την καταλληλότητα των προϋποθέσεων ή των απλουστεύσεων που επιβάλλονται. Εξαρτάται δηλαδή από την ακρίβεια της διατύπωσης του προβλήματος.

2. Μη-Γραμμικός προγραμματισμός (Non-Linear Programming – NLP)

Είναι οι πιο διαδεδομένες τεχνικές μαθηματικού προγραμματισμού. Αντιμετωπίζουν γενικώς όλες τις περιπτώσεις όπου η αντικειμενική συνάρτηση ή μερικοί από τους περιορισμούς είναι μη γραμμικές συναρτήσεις.

3. Ακέραιος προγραμματισμός (Integer Programming – IP)

Τα προβλήματα ακέραιου προγραμματισμού αποτελούν ειδική περίπτωση των προβλημάτων γραμμικού προγραμματισμού. Πρόκειται για προβλήματα γραμμικού προγραμματισμού, στα οποία υπάρχει ο περιορισμός ότι όλες ή κάποιες μεταβλητές είναι ακέραιες. Οι περιπτώσεις που δεν είναι όλες οι μεταβλητές ακέραιες οδηγούν στα προβλήματα «Μικτού Ακέραιου προγραμματισμού» (Mixed Integer Programming – MIP). Όταν σε προβλήματα ακέραιου προγραμματισμού δεν είναι δυνατόν να εντοπιστεί λύση με χρήση μεθόδων γραμμικού προγραμματισμού, τότε γίνεται επίλυση με μεθόδους όπως η «Μέθοδος κλάδου και φράγματος» (branch and bound), η «Μέθοδος της απαρίθμησης» (enumeration) και οι ευρεστικές-προσεγγιστικές μέθοδοι που παρουσιάζονται σε επόμενη παράγραφο.

4. Γεωμετρικός προγραμματισμός (Geometric Programming – GP)

Αναφέρεται σε μια ειδική κατηγορία προβλημάτων όπου οι συναρτήσεις περιορισμού αλλά και η αντικειμενική συνάρτηση είναι πολυωνυμικής μορφής συναρτήσεις.

5. Δυναμικός προγραμματισμός (Dynamic Programming – DP)

Κύριος στόχος αυτών των μεθόδων είναι να διασπασθεί ένα σχετικά μεγάλο και πολύπλοκο πρόβλημα σε μικρότερα τα οποία μπορούν να αντιμετωπισθούν ως ξεχωριστά προβλήματα βέλτιστου σχεδιασμού.

Εκτός όμως των μαθηματικών μεθόδων υπάρχουν και οι ευρεστικές μέθοδοι (heuristics), οι οποίες είναι προσεγγιστικές και εφαρμογή τους αποδεικνύεται ιδιαίτερα αποτελεσματική σε ένα ευρύ πεδίο προβλημάτων. Οι μέθοδοι αυτοί μπορεί να είναι:

- η μέθοδος των στρατηγικών εξέλιξης (evolution strategies)
- η μέθοδος των γενετικών αλγορίθμων (genetic algorithms)
- η μέθοδος της προσομοίωσης (simulation)

Ο συνδυασμός των τεχνικών που αναφέρθηκαν αλλά και η κάθε μία τεχνική ξεχωριστά, μπορούν να δώσουν λύση σε πολλά προβλήματα μεταφορών. Για παράδειγμα, τόσο ο γραμμικός όσο και ο ακέραιος και ο δυναμικός προγραμματισμός εφαρμόζονται στα προβλήματα δικτύων. Τα προβλήματα δικτύων αφορούν «οποιοδήποτε συνδυασμό ανθρώπων, δραστηριοτήτων αλλά και αντικειμένων, τα οποία συνδέονται μεταξύ τους με νοητές ή απτές διασυνδέσεις για την επίτευξη ενός έργου ή μεταφοράς». Σε πολλές περιπτώσεις όπου η επίλυση τους αποτελεί μια δύσκολη και πολύπλοκη διαδικασία, η επίλυση μπορεί να γίνει με χρήση των προσεγγιστικών μεθόδων.

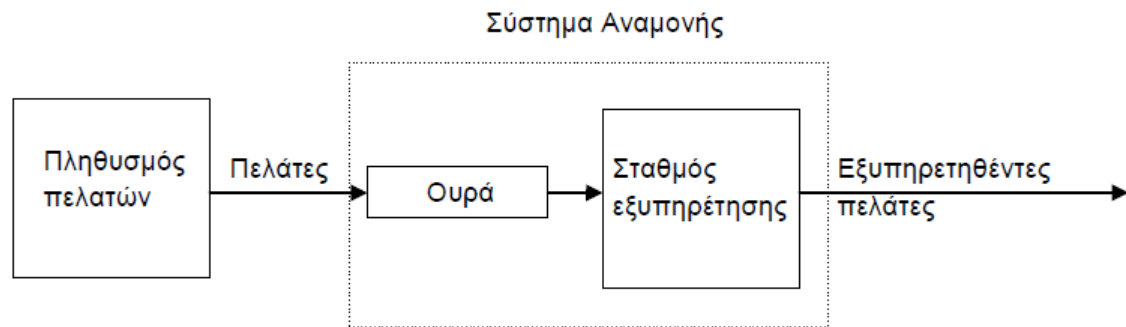
Λόγω της συνεχής αύξησης με έντονο ρυθμό των προκλήσεων στον τομέα προγραμματισμού και διαχείρισης ενός στόλου οχημάτων, που θα πραγματοποιούν μεταφορές, έχει γίνει μια προτυποποίηση προβλημάτων μεταφοράς. Τα κυριότερα πρότυπα προβλήματα της μορφής δικτύου είναι τα εξής:

- Πρόβλημα Διακίνησης (The Transshipment Problem)
- Πρόβλημα του Περιοδεύοντος Πωλητή (The Travelling Salesman Problem)
- Προβλήματα προγραμματισμού και δρομολόγησης στόλου οχημάτων (Vehicle Routing and Scheduling Problems – VRP's)
- Πιθανοτικά προβλήματα προγραμματισμού και δρομολόγησης στόλου οχημάτων (Probabilistic Vehicle Routing and Scheduling Problems)

Όταν ολοκληρωθούν οι διαδικασίες ανάπτυξης του προβλήματος και επίλυσης του, θα πρέπει να ακολουθήσει η εφαρμογή και η αξιολόγηση της λύσης. Η λύση λοιπόν του προβλήματος συνοδεύεται από ανάλυση ευαισθησίας (sensitivity analysis) των τιμών της. Εντοπίζονται με αυτό τον τρόπο, οι παράμετροι που είναι κρίσιμοι για την επίλυση του προβλήματος και η ευαισθησία της λύσης σε μεταβολή των παραμέτρων.

Από όλες τις μεθόδους που παρουσιάστηκαν, η θεωρία αναμονής, ο μικτός ακέραιος προγραμματισμός και η προσομοίωση συγκεντρώνουν τη μεγαλύτερη προτίμηση στη βιβλιογραφία, για επίλυση προβλημάτων στις λιμενικές εγκαταστάσεις.

Η θεωρία αναμονής (queueing theory) παρέχει ένα μεγάλο αριθμό μαθηματικών προτύπων για την περιγραφή των καταστάσεων των γραμμών αναμονής. Ένα βασικό σύστημα αναμονής μπορεί να περιγραφεί ως εξής: οι πελάτες φτάνουν για εξυπηρέτηση και στην περίπτωση που δεν είναι άμεσα διαθέσιμη περιμένουν για αυτήν στην ουρά αναμονής. Στην συνέχεια μετά την αναμονή τους στην ουρά φτάνουν σε έναν ή περισσότερους σταθμούς εξυπηρέτησης και μετά το πέρας της εξυπηρέτησης αυτής, αποχωρούν από το σύστημα (Εικόνα 14). Ο όρος πελάτης χρησιμοποιείται σε ευρύτερη έννοια και δεν εννοεί απαραίτητα τον ανθρώπινο πελάτη. Τα πιο συνηθισμένα στοχαστικά μοντέλα ουρών που χρησιμοποιούνται στη βιβλιογραφία για τη διατύπωση προβλημάτων στις λιμενικές εγκαταστάσεις, υποθέτουν χρόνους άφιξης και εξυπηρέτησης των πελατών που ακολουθούν τις κατανομές Poisson και Erlang.



Εικόνα 15 Το σύστημα αναμονής [Πηγή: Ξηροκώστας, Δ. (1990), Επιχειρησιακή έρευνα – Εφαρμοσμένη θεωρία αναμονής]

Οι Edmond και Mags, 1978, εξέτασαν κατά πόσο τα αποτελέσματα των μοντέλων αναμονής μπορούν να χρησιμοποιηθούν στη λήψη αποφάσεων, για πραγματοποίηση επενδύσεων σε εγκαταστάσεις διακίνησης εμπορευματοκιβωτίων. Χρησιμοποίησαν λοιπόν τα μοντέλα $M/M/s$, $D/M/s$ και $M/E2/s$ και συμπέραναν ότι μερικά απλά μοντέλα αναμονής δίνουν μια καλή προσέγγιση σε ορισμένα μεγέθη σχεδιασμού.

Για τον υπολογισμό των δεικτών απόδοσης ενός συστήματος με χρήση θεωρίας αναμονής είναι δυνατή η χρήση λογισμικών για γρήγορα και αξιόπιστα αποτελέσματα. Ωστόσο θα πρέπει να γίνεται κάθε φορά επιλογή κατάλληλου λογισμικού, το οποίο να μπορεί να περιγράψει το σύστημα και να μπορεί να δώσει τα αποτελέσματα που επιθυμεί ο χρήστης. Παρακάτω δίνεται μια λίστα με λογισμικά τα οποία δύναται να χρησιμοποιηθούν:

- **Java Modelling Tools (<http://jmt.sourceforge.net/>)**

Πρόκειται για μια σειρά εφαρμογών που αναπτύχθηκαν από το Πολυτεχνείο του Μιλάνο και το Imperial College του Λονδίνου και διατίθενται βάσει της άδειας GPL. Περιλαμβάνονται έξι εφαρμογές σε περιβάλλον Java οι οποίες εκτελούν προσομοιώσεις της θεωρίας αναμονής.

- **Solutions to Queueing Systems (<http://queueing-systems.ens-lyon.fr/>)**

Παρουσιάζεται η αριθμητική επίλυση συστημάτων με ουρές αναμονής που περιγράφονται από τα μοντέλα $G/M/c$, $M/G/1$, $M/G/c$ και $G/G/c$.

- **Queueing Theory Calculator (<http://www.supositorio.com/rcalc/rcallite.htm>)**

Παρουσιάζεται η αριθμητική επίλυση συστημάτων με ουρές αναμονής που περιγράφονται από τα μοντέλα $M/M/c$, $M/M/\inf$, $M/M/c/K$ και $M/M/c/*M$.

- **Simple Queueing Demo (http://gswce.net/?page_id=358#SimpleQueueing)**

Πρόκειται για ένα online λογισμικό το οποίο προσομοιώνει απλά συστήματα με χρήση θεωρίας αναμονής και εξάγει αποτελέσματα. Γίνεται επιλογή των κατανομών άφιξης και εξυπηρέτησης που επιθυμεί ο χρήστης να χρησιμοποιήσει.

- **M/M/1 Queue Demo (<http://web.cs.dal.ca/~grundke/3171.2004.09/lectures/mm1/>)**

Πρόκειται για μια εφαρμογή που προσομοιώνει ένα πρότυπο $M/M/1$.

- **M/M/1Queues**(http://homepages.inf.ed.ac.uk/jeh/Simjava/queueing/mm1_q/mm1_q.html)

Πρόκειται για μια εφαρμογή που προσομοιώνει ένα πρότυπο M/M/1.

- **Queueing Networks**

(<http://homepages.inf.ed.ac.uk/jeh/Simjava/queueing/Networks/networks.html>)

Αυτή η εφαρμογή επιτρέπει τον σχεδιασμό, την επίλυση και την προσομοίωση ενός συστήματος με χρήση της θεωρίας αναμονής.

- **Expokit** (<http://www.maths.uq.edu.au/expokit/>)

Πρόκειται για ένα λογισμικό που χρησιμοποιεί τις γλώσσες προγραμματισμού Matlab και Fortran και επιλύει μοντέλα με εκθετική κατανομή.

- **Qtsplus-xcel** (ftp://ftp.wiley.com/public/sci_tech_med/queueing_theory)

Πρόκειται για ένα λογισμικό το οποίο με χρήση του προγράμματος Excel μπορεί να επιλύσει τους περισσότερους τύπους μοντέλων της θεωρίας αναμονής. Για την επίλυση των μοντέλων χρησιμοποιεί τους αναλυτικούς τύπους της θεωρίας αναμονής.

Η μέθοδος της προσομοίωσης (simulation) έχει χρησιμοποιηθεί εκτενώς στη βιβλιογραφία για την ανάλυση σύνθετων συστημάτων. Ένα λιμενικό σύστημα εμπορευματοκιβωτίων είναι σύνθετο λόγω των διαφορετικών χρόνων άφιξης των πλοίων, των διαφορετικών διαστάσεων των πλοίων, των πολλαπλών γερανογεφυρών και θέσεων παραβολής, των διαφορετικών ικανοτήτων και της παραγωγικότητας των γερανογεφυρών, της προνομιακής εξυπηρέτησης των πλοίων κτλ. Το βασικότερο πλεονέκτημα της μεθόδου είναι η ευελιξία που προσφέρει στο μελετητή ως προς τη διαμόρφωση του μοντέλου. Τα μοντέλα προσομοίωσης διακρίνονται ως εξής κατά, Kelton, W. D., Sadowski, R. P.:

- Στατικά/Δυναμικά (Static/ Dynamic): Η μεταξύ τους διαφορά είναι η επίδραση του χρόνου. Στα στατικά μοντέλα ο χρόνος είναι ανύπαρκτος. Τα μοντέλα σε μελέτες επιχειρησιακής έρευνας (και κατ' επέκταση προσομοίωσης) είναι κατά κανόνα δυναμικά.
- Συνεχή/Διακριτά (Continuous/Discrete): Σε ένα μοντέλο συνεχούς χρόνου η κατάσταση του συστήματος μπορεί να αλλάζει συνεχώς σε βάθος χρόνου. Αντιθέτως στα μοντέλα διακριτού χρόνου οι αλλαγές καταστάσεων συμβαίνουν μόνο σε διακριτές χρονικές στιγμές. Η συντριπτική πλειοψηφία των συστημάτων απεικονίζονται σε μοντέλου διακριτού χρόνου.
- Προσδιοριστικά/Στοχαστικά (Deterministic/Stochastic): Όπως αναφέρθηκε και παραπάνω, τα μοντέλα δίχως τυχαιότητα στα δεδομένα τους είναι προσδιοριστικά (ή ντετερμινιστικά), ενώ όταν τα δεδομένα τους ακολουθούν κάποια κατανομή είναι στοχαστικά.

Η προσομοίωση μπορεί να υλοποιηθεί με το πείραμα του Monte Carlo, με κοινές γλώσσες προγραμματισμού (Fortran, Pascal, C/C++) και με γλώσσες προσομοίωσης (GPSS, Simscript, SLAM, SIMAN, AweSim).

Άλλοι μέθοδοι που χρησιμοποιήθηκαν στην βιβλιογραφία είναι:

Η μέθοδος Cutting Plane όπου μειώνεται ο χώρος των εφικτών λύσεων κάνοντας χρήση γραμμικών ανισοτήτων. Μια άλλη πολύ διαδεδομένη μέθοδος είναι η μέθοδος κλάδου και φραγής (Branch & Bound) που λειτουργεί σε δύο φάσεις, στην πρώτη φάση πραγματοποιείται τμηματοποίηση του χώρου των υποψήφιών – εφικτών λύσεων ενώ στη δεύτερη υπολογίζονται τα ανώτατα και κατώτατα όρια των τμημάτων αυτών. Η επίλυση βασίζεται στην παραδοχή ότι αν το κατώτερο όριο ενός συνόλου είναι μεγαλύτερο από το ανώτερο όριο του αλλού τότε μπορεί κανείς με ασφάλεια να αγνοήσει το πρώτο σύνολο. Η φάση αυτή καλείται «κλάδεμα». Η μέθοδος αυτή αποτέλεσε και τη βάση για πολλές ευρετικές διαδικασίες που έχουν προταθεί. Επεκτάσεις αυτών είναι και η Branch & Cut καθώς και η Branch & Price, αλλά και άλλες η αναλυτική παρουσίαση των οποίων ξεφεύγει από τους στόχους αυτού του κείμενου. Για περεταίρω έρευνα προτείνεται η ανασκόπηση του συγγράμματος του Laporte (1992). Το μεγάλο μειονέκτημα των αλγορίθμων αυτών είναι ότι ανταποκρίνονται σε προβλήματα μικρού μεγέθους και δεν είναι εφαρμόσιμα για δίκτυα που αποτελούνται από περισσότερους από 50 κόμβους. Από την άλλη το πρόβλημα της δρομολόγησης έχει εφαρμογή σε μεγάλη γκάμα προβλημάτων και γι αυτό η επιστημονική κοινότητα έστρεψε το ενδιαφέρον της στο να αναπτύξει μεθόδους που θα προσεγγίζουν τη βέλτιστη λύση σε πολύ μικρό χρόνο, τις λεγόμενες ευρετικές μεθόδους.

Οι αλγόριθμοι που χρησιμοποιούνται ευρέως στην βιβλιογραφία είναι:

Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι αποτελούν ένα σύστημα επίλυσης προβλημάτων βασισμένες στις αρχές της Φυσικής. Μαζί με τον Εξελικτικό Προγραμματισμό (Evolutionary Programming) τις Στρατηγικές Εξέλιξης (Evolution Strategies), τα Συστήματα Ταξινόμησης (Classifier Systems) και τον Γενετικό Προγραμματισμό (Genetic Programming) αποτελούν μια κατηγορία συστημάτων που είναι ευρύτερα γνωστή με τον όρο Εξελικτικοί Αλγόριθμοι (Evolutionary Algorithms).

Οι Γενετικοί αλγόριθμοι πρωτοεμφανίστηκαν το 1950, όταν διάφοροι επιστήμονες από τον χώρο της βιολογίας αποφάσισαν να χρησιμοποιήσουν υπολογιστές στην προσπάθεια τους να προσομοιώσουν πολύπλοκα βιολογικά συστήματα. Η ανάπτυξη των Γενετικών Αλγορίθμων πραγματοποιήθηκε στις αρχές του 1970 από τον John Holland (1975) και τους συνεργάτες του στο πανεπιστήμιο του Michigan.

Πιο συγκεκριμένα ένας Γενετικός Αλγόριθμος πρέπει να αποτελείται από κάποια συστατικά για ένα συγκεκριμένο πρόβλημα. Το πρώτο αφορά μια γενετική αναπαράσταση των πιθανών λύσεων ενός προβλήματος, το δεύτερο έναν τρόπο δημιουργίας ενός αρχικού πληθυσμού από πιθανές λύσεις, το τρίτο μι αντικειμενική συνάρτηση αξιολόγησης των μελών του πληθυσμού που παίζει ρόλο το περιβάλλον, το τέταρτο τους γενετικούς τελεστές για τη δημιουργία των λύσεων και τέλος το πέμπτο τις τιμές για τις διάφορες παραμέτρους που χρησιμοποιεί ο Γ.Α δηλαδή το μέγεθος του πληθυσμού, τις πιθανότητες εφαρμογής των γενετικών τελεστών κ.τ.λ..

Ωστόσο αξίζει να αναφερθούμε σε μερικά από τα πλεονεκτήματα που παρουσιάζουν οι Γενετικοί Αλγόριθμοι που είναι ικανά για την επίλυση προβλημάτων :

1. Μπορούν να επιλύσουν γρήγορα και αξιόπιστα δύσκολά προβλήματα.
2. Μπορούν εύκολα να συνεργαστούν με τα υπάρχοντα μοντέλα και συστήματα.
3. Είναι επεκτάσιμοι και εξελίξιμοι.

4. Συμμετέχουν σε υβριδικές μορφές με άλλες μεθόδους.
5. Έχουν εφαρμογή σε πιο πολλά πεδία από ότι άλλοι μέθοδοι.
6. Δεν απαιτούν περιορισμούς στις συναρτήσεις που επεξεργάζονται.
7. Δεν ενδιαφέρει η σημασία της υπό εξέταση πληροφορίας.
8. Έχουν από τη φύση τους το στοιχείο του παραλληλισμού.
9. Αποτελεί μια μέθοδο που κάνει ταυτόχρονα εξερεύνηση του χώρου αναζήτησης και εκμετάλλευση της ήδη επεξεργασμένης πληροφορίας.
10. Επιδέχονται παράλληλη υλοποίηση.

Ευρετικοί Αλγόριθμοι

Οι ευρετικοί αλγόριθμοι είναι προσεγγιστικές μέθοδοι οι οποίες περιορίζονται στη σάρωση «περιοχών λύσεων» ικανοποιητικής ποιότητας. Διαχωρίζονται σε δύο τμήματα, στο κατασκευαστικό και στο βελτιωτικό (τοπικής αναζήτησης). Στο κατασκευαστικό κομμάτι παράγουν μια αρχική λύση που στη συνέχεια στο βελτιωτικό χρησιμοποιώντας εναλλακτικές τεχνικές ανταλλαγής τμημάτων μεταξύ των λύσεων δημιουργούν νέα βελτιωμένα αποτελέσματα. Παρουσιάζουν ιδιαίτερο ενδιαφέρον από μόνες τους επειδή μπορούν πολύ γρήγορα να παράγουν καλές λύσεις αλλά και επειδή αποτελούν συστατικά μεθευριστικών αλγορίθμων. Τα πιο διαδεδομένα κριτήρια αξιολόγησης των λύσεων είναι:

- η απόκλιση από την βέλτιστη
- η ευκολία απόκτησης της
- η λογική πάνω στην οποία στηρίζεται και
- οι κανόνες του ευρετικού αλγορίθμου

Όταν είναι εφικτό, τα αποτελέσματα τους συγκρίνονται με αυτά που παράγονται με ενόν ακριβή αλγόριθμο.

Ακριβείς Αλγόριθμοι

Οι ακριβείς αλγόριθμοι, δίνουν τη δυνατότητα να εντοπιστεί η βέλτιστη λύση του προβλήματος κάτι που δεν εξασφαλίζεται με τις άλλες κατηγορίες αλγορίθμων. Στην ουσία εξετάζουν το σύνολο των εφικτών λύσεων και γίνεται ο υπολογισμός της αντικειμενικής συνάρτησης ώστε να εντοπιστεί η βέλτιστη λύση. Τεχνικές μείωσης του χώρου των εφικτών λύσεων χρησιμοποιούνται ώστε να επιταχύνουν και να ολοκληρώσουν τη διαδικασία.

Κατασκευαστικοί Αλγόριθμοι

Στην κατηγορία των κατασκευαστικών αλγορίθμων κατατάσσονται οι αλγόριθμοι απληστίας (Greedy algorithms) οι οποίοι χαρακτηρίζονται ως μυωπικοί, δηλαδή βλέπουν μόνο μπροστά. Επίσης οι προσεγγιστικοί κατατάσσονται στους κατασκευαστικούς οι οποίοι προσπαθούν να λύσουν το πρόβλημα χρησιμοποιώντας επιπλέον πληροφορία που καθορίζεται από τον σχεδιαστή του. Η διαδρομή που κατασκευάζουν προκύπτει από την ελαχιστοποίηση ενός κριτηρίου, το οποίο μπορούν να το κάνουν για κάθε όχημα διαδοχικά είτε παράλληλα.

Αλγόριθμοι Άμεσης Απόκρισης –Online Algorithm

Οι αλγόριθμοι αυτοί έχουν εφαρμογή σε προβλήματα όπου η είσοδος δεν είναι γνωστή εκ των προτέρων αλλά εμφανίζεται σταδιακά. Ένας αλγόριθμος άμεσης απόκρισης καλείται να πάρει αποφάσεις για τη λύση ενός προβλήματος με βάση μια μερική πληροφορία για τα δεδομένα του προβλήματος και χωρίς να μπορεί να προβλέψει το μέλλον. Τα σχετικά προβλήματα είναι είτε προβλήματα μεγιστοποίησης κέρδους είτε προβλήματα ελαχιστοποίησης κόστους. Γενικά, η είσοδος του προβλήματος γίνεται γνωστή με τη μορφή μιας ακολουθίας αιτήσεων. Όταν εμφανίζεται μια αίτηση, ένας αλγόριθμος άμεσης απόκρισης καλείται να πάρει μια τελεσίδικη απόφαση για τη συγκεκριμένη αίτηση. Για την ανάλυση αλγορίθμων άμεσης απόκρισης χρησιμοποιούμε την έννοια της ανταγωνιστικής ανάλυσης (competitive analysis), με βάση την οποία, η απόδοση του αλγορίθμου συγκρίνεται με την απόδοση ενός βέλτιστου (υποθετικού) αλγορίθμου που έχει πρόσβαση σε ολόκληρη την ακολουθία αιτήσεων. Η μετρική που χρησιμοποιούμε για την ποσοτικοποίηση της απόδοσης ενός αλγορίθμου άμεσης απόκρισης είναι ο λόγος απόδοσης (competitive ratio). Ας υποθέσουμε ότι ο αλγόριθμος ALG λειτουργεί σε ακολουθίες αιτήσεων για το πρόβλημα μεγιστοποίησης κέρδους Π. Τότε, ο λόγος απόδοσής του ορίζεται ως η ελάχιστη τιμή της ποσότητας C για την οποία ισχύει η σχέση

$$OPT(\sigma) - C \cdot ALG(\sigma) \leq \alpha$$

για κάθε δυνατή ακολουθία αιτήσεων για το πρόβλημα Π. Οι ποσότητες $OPT(\sigma)$ και $ALG(\sigma)$ δηλώνουν το βέλτιστο (μέγιστο) κέρδος για την ακολουθία αιτήσεων σ και το κέρδος του αλγορίθμου ALG κατά την εκτέλεσή του στην ακολουθία σ , ενώ η παράμετρος α είναι μια οποιαδήποτε σταθερά που είναι ανεξάρτητη της ακολουθίας. [S. Albers, 1996, Y. Azar, 1996]

3.2 Προβλήματα που καλούνται να επιλυθούν σε ένα λιμενικό εμπορευματικό σταθμό

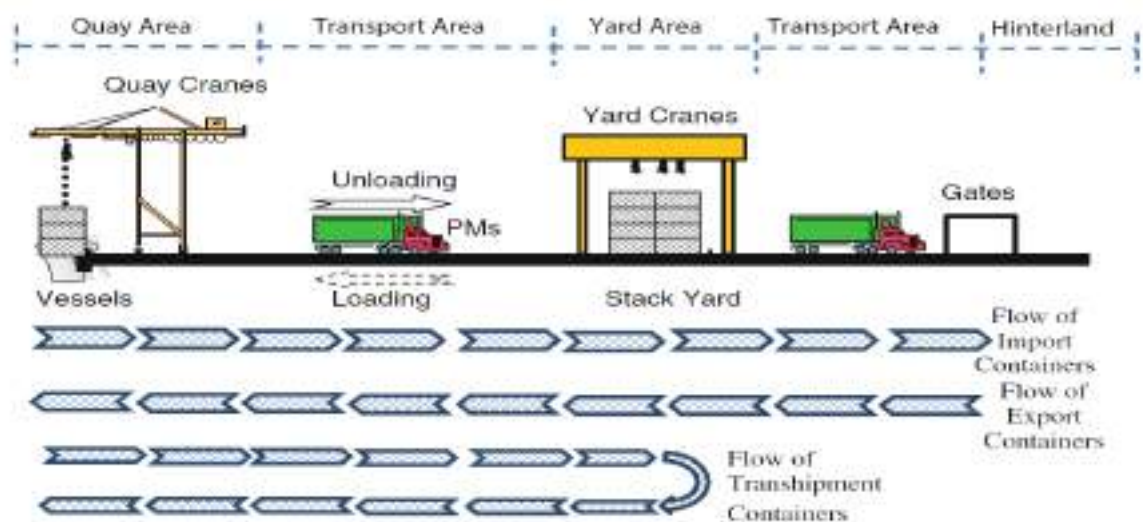
3.2.1 Γενικά

Κάθε σταθμός εμπορευματοκιβωτίων αποτελεί ένα σημαντικό στρατηγικό σημείο στην διεθνή μεταφορική αλυσίδα καθώς είναι το σημείο όπου διενεργούνται πολλές διαχειριστικές διεργασίες προκειμένου να επιτευχθεί στον συντομότερο δυνατό χρόνο, αξιόπιστα και με ασφάλεια, η απαραίτητη ανακατεύθυνση των εμπορευματοκιβωτίων προκειμένου να φτάσουν στον προορισμό τους. Εξαιτίας της πολυπλοκότητας των εισροών στο σταθμό εμπορευματοκιβωτίων, με την έννοια των πολλών πληροφοριών που συνοδεύουν το κάθε εμπορευματοκιβώτιο (πχ. Βάρος, προορισμός, χρόνος εξόδου από τον σταθμό εμπορευματοκιβωτίων, τρόπος εξόδου από σταθμό εμπορευματοκιβωτίων, κ.α.) απαιτούνται σύνθετες λειτουργίες προκειμένου να επιτευχθεί μια υψηλή συνολική απόδοση που θα έχει ως αποτέλεσμα τη βιωσιμότητα του εκάστοτε σταθμού εμπορευματοκιβωτίων έναντι των άλλων ανταγωνιστών. Ένας παράγοντας που επηρεάζει άμεσα τη βιωσιμότητα των σταθμών εμπορευματοκιβωτίων είναι η προσαρμοστικότητα τους στην νέες τεχνολογικές και οικονομοτεχνικές εξελίξεις που τα τελευταία χρόνια είναι ραγδαίες. Χαρακτηριστικό παράδειγμα της έκρηξης αυτής είναι το γεγονός ότι η εμπορευματική μεταφορικότητα των πλοίων μεταφοράς εμπορευματοκιβωτίων των τελευταίων δέκα χρόνων έχει αυξηθεί κατά 75% (11,000 TEUs [PSA International 2006] → 19,224 TEUs σύμφωνα με Stromberg, (2015), κάτι που

αντανακλά άμεσα και στις απαιτήσεις από τους σταθμούς εμπορευματοκιβωτίων. Πολλές φορές παρατηρείται το φαινόμενο τα πλοία να μην μπορούν να προσεγγίσουν τους σταθμούς εμπορευματοκιβωτίων καθώς δεν επαρκεί το βάθος, ή οι ΓΕΡΑΝΟΪ ΚΡΥΠΙΔΩΜΑΤΟΣ δεν έχουν το απαιτούμενο ύψος, ή δεν υπάρχει ο απαιτούμενος χώρος στην παραβολή κ.α. με αποτέλεσμα ο σταθμός εμπορευματοκιβωτίων να γίνεται αυτομάτως λιγότερο ανταγωνιστικό χάνοντας σημαντικούς πελάτες, οι οποίοι τελικά θα οδηγηθούν σε κάποιον ανταγωνιστή με ότι αυτό μπορεί να συνεπάγεται για μελλοντικές συνεργασίες. Παρόλα αυτά όμως, το εκάστοτε σταθμός εμπορευματοκιβωτίων οφείλει να αξιοποιεί στο μέγιστο όλες του τις δυνατότητες, προκειμένου οι επενδύσεις που έχουν γίνει να είναι αποδοτικές και να επιφέρουν ορατά αποτελέσματα. Η αποτελεσματικότητα του εκάστοτε σταθμού εμπορευματοκιβωτίων εκτός από την βιωσιμότητα του ίδιου, έχει άμεσες επιρροές και στο περιβάλλον γύρω του. Ένας ισχυρός σταθμός εμπορευματοκιβωτίων αυξάνει την εμπορευματικότητα σε όλο το χερσαίο δίκτυο που καταλήγει σε αυτό με αποτέλεσμα να αυξάνεται ο φόρτος εργασίας σε μια πληθώρα από κλάδους υποστήριξης και μεταφοράς με πολλά κοινωνικά, πολιτικά και οικονομικά οφέλη. Επίσης, πολλά οφέλη έχει και ο βιομηχανικός κλάδος, καθώς επιτυγχάνονται οι απαιτούμενες μεταφορές των προϊόντων σε μικρότερο χρόνο και κόστος καθώς η χερσαία μεταφορά είναι μικρότερη και σαφώς πιο αξιόπιστα καθώς κάθε χερσαία μεταφορά επιφυλάσσει αρκετούς κινδύνους που μπορεί να επιφέρουν μεγάλες καθυστερήσεις (μπλόκα, γραφειοκρατικά προβλήματα, πολλαπλοί έλεγχοι σε σύνορα, πολιτικές αστάθειες που επιφέρουν απαγορεύσεις προσπέλασης χωρών κ.α.).

3.2.2 Γενική επισκόπηση των προβλημάτων ενός εμπορευματικού λιμένα

Τα προβλήματα σε ένα σταθμό εμπορευματοκιβωτίων εισάγονται από τις διάφορες διαδικασίες που καλείται να διεκπεραιώσει. Οι βασικότερες αυτών είναι, η διαδικασία αγκυροβόλησης των πλοίων, η φορτοεκφόρτωσή τους, η μεταφορά από και προς τον χώρο στοιβασίας και η φορτοεκφόρτωση των χερσαίων μέσων προκειμένου τα εμπορευματοκιβώτια να φτάσουν στον τελικό τους προορισμό (Εικόνα 16).



Εικόνα 16 Περιοχές του λιμανιού [Πηγή: [springer.com](https://www.springer.com) , Chapter 2 Maritime Terminal Operational Problems]

Για τα προβλήματα αυτά, έχουν γίνει προσπάθειες από πολλούς ερευνητές προκειμένου να κατηγοριοποιηθούν.

Ο Böse (2011) τα κατηγοριοποίησε σε 3 επίπεδα, όπου το τελευταίο περιέχει άλλες τρεις κατηγορίες. Τα επίπεδα αυτά είναι:

1. Σχεδιασμός υποδομής του σταθμού εμπορευματοκιβωτίων

Αφορά τις αποφάσεις που σχετίζονται με την προετοιμασία και τον σχεδιασμό του σταθμού εμπορευματοκιβωτίων (περιοχή, μήκος θαλάσσιας πλευράς κ.α.) αλλά και τις συνδέσεις του με τα εξωτερικά δίκτυα μεταφοράς.

2. Σχεδιασμός επιδομής του σταθμού εμπορευματοκιβωτίων

Αφορά τις αποφάσεις που σχετίζονται με το είδος των υποστηρικτικών μηχανημάτων και τον αριθμό τους, την διάταξη του χώρου και το ανθρώπινο δυναμικό.

3. Σχεδιασμός των διεργασιών (operations) του σταθμού εμπορευματοκιβωτίων

Αφορά τις αποφάσεις που σχετίζονται με τον τρόπο λειτουργίας των υποστηρικτικών μηχανημάτων σε βραχυπρόθεσμο, μεσοπρόθεσμο και μακροπρόθεσμο πλάνο.

Ανάμεσα σε αυτά τα επίπεδα κατηγοριοποίησης υπάρχει έντονη αλληλεξάρτηση. Τα χαμηλότερα επίπεδα παρέχουν υποστήριξη αλλά και περιορισμούς στα υψηλότερα, ενώ τα υψηλότερα καθορίζουν τις απαιτήσεις για τα χαμηλότερα. Το πρώτο επίπεδο αφορά τις τοπικές κοινωνίες και τις αρχές, κάτι που το καθιστά εκτός της ακαδημαϊκής έρευνας.

Ο Günther et al. (2005) κάνοντας μια άλλη κατηγοριοποίηση τριών επιπέδων επίσης ασχολήθηκε μόνο με τα αντίστοιχα υψηλότερα επίπεδα του (Böse 2011), δηλαδή το δεύτερο και το τρίτο. Τα επίπεδα σύμφωνα με τον Günther et al. (2005) είναι:

1. Σχεδιασμός

Αφορά τις αποφάσεις που σχετίζονται με το είδος των υποστηρικτικών μηχανημάτων και τον αριθμό τους, την διάταξη του χώρου και το ανθρώπινο δυναμικό.

2. Προγραμματισμός των διεργασιών

Αφορά τις αποφάσεις που σχετίζονται με την μεγιστοποίηση της χρησιμοποίησης και απόδοσης των υποστηρικτικών μηχανημάτων και του ανθρώπινου δυναμικού

3. Έλεγχος σε πραγματικό χρόνο

Αφορά τις αποφάσεις που σχετίζονται με την μεγιστοποίηση της χρησιμοποίησης και απόδοσης των υποστηρικτικών μηχανημάτων και του ανθρώπινου δυναμικού σε πραγματικό χρόνο, καθώς ο προγραμματισμός πρακτικά σε ορισμένες περιπτώσεις δεν μπορεί να ξεπερνά ένα πολύ μικρό χρονικό διάστημα της τάξης των μερικών λεπτών. Σε αυτό το επίπεδο απαιτείται χρήση γρήγορων αλγορίθμων που δεν δίνουν έμφαση στην βέλτιστη λύση.

Στην παρούσα διπλωματική εργασία έχει γίνει επικέντρωση στα επίπεδα ένα και δύο όπως τα έχει ορίσει ο Günther et al. (2005) με την χρήση του προγράμματος Anylogic. Εξαιτίας βέβαια της φύσης του προγράμματος, εφόσον αποτελεί πρόγραμμα προσομοίωσης, εν μέρει γίνεται και μια προσέγγιση και του τρίτου επιπέδου. Αναλύοντας τα προβλήματα αυτά παρακάτω αναλυτικότερα, έχει γίνει μια δεύτερη παράλληλη κατηγοριοποίηση τριών σημείων που σχετίζονται με τον χωρικό προσδιορισμό, δηλαδή τον χώρο παραβολής, τον χώρο μεταφοράς

εντός του λιμένα και τον χώρο στοιβασίας. Τέλος, έγινε ενσωμάτωση των υποπροβλημάτων αυτών στο συνολικό σταθμό εμπορευματοκιβωτίων.

3.3 Χώρος παραβολής

Οι αποφάσεις που σχετίζονται με τον χώρο παραβολής παίζουν σημαντικό ρόλο για την συνολική λειτουργία του λιμένα καθώς αποτελεί το κρίσιμο σημείο της λειτουργίας. Σε αυτή την περιοχή οι αποφάσεις που πρέπει να ληφθούν είναι:

- Η τοποθεσία της θέσης παραβολής (BAP)
- Η ανάθεση των γερανών κρηπιδώματος
- Ο προγραμματισμός των γερανών κρηπιδώματος
- Ο προγραμματισμός της στοιβασίας του πλοίου

Σε αυτή τη διπλωματική δεν θα γίνει εκτενής αναφορά στον προγραμματισμό της στοιβασίας του πλοίου.

3.3.1 Τοποθεσία θέσης παραβολής (BAP)

Μια συγκεκριμένη θέση παραβολής για κάθε πλοίο που καταφθάνει καθορίζεται επακριβώς από τους αρμόδιους του λιμένος. Το BAP σχετίζεται με την επιλογή της καταλληλότερης θέσης με γνώμονα τον αναμενόμενο χρόνο που θα παραβάλει το εκάστοτε πλοίο, το μήκος του αλλά και το βάθος της παραβολής. Οι παράμετροι εισόδου που λαμβάνονται είναι τα τεχνικά χαρακτηριστικά του πλοίου (μήκος, βύθισμα κ.α.), των γερανών κρηπιδώματος (μήκος του βραχίονα), δεδομένα που σχετίζονται με τον τύπο της θέσης παραβολής (μήκος, διάταξη της θέσης παραβολής κ.α.), ο προβλεπόμενος χειρισμός του πλοίου, τα time window του ελλιμενισμού, η προτεραιότητες των πλοίων κ.α. Οι στόχοι του BAP είναι η μεγιστοποίηση του customer service, η ελαχιστοποίηση του χρόνου που παραβάλει το εκάστοτε πλοίο και η ελαχιστοποίηση του κόστους συνολικότερα. Έχουν γίνει αρκετές έρευνες πάνω στο BAP με σημαντικότερες τις παρακάτω:

- Lim (1998): έγινε προσπάθεια αποτύπωσης του BAP αλλά αποδείχθηκε ως πολύ πολύπλοκο πρόβλημα, με αποτέλεσμα να γίνει μετατροπή του προβλήματος της φορτοεκφόρτωσης σε δύο διαστάσεων προκειμένου να λυθεί.
- Nishimura et al. (2001): πρότεινε τον χειρισμό του BAP σαν ένα public berth system και εφάρμοσε μια γεννήτρια αλγορίθμων για την απόκτηση αποδοτικές λύσεις με την χρήση υπολογιστή
- Guan and Cheung (2004): έγινε θεώρηση του BAP με στόχο την ελαχιστοποίηση του συνολικού χρόνου ροής. Προτάθηκαν σύνθετες έρευνες για μεγάλου μεγέθους προβλήματα
- Imai et al. (2003): έγινε τροποποίηση της μεθόδου προσέγγισης του BAP προκειμένου να γίνει εναρμονισμός με της προτεραιότητες των πλοίων για την παραβολή
- Golias et al. (2009): αναπτύχθηκε ένα πολλαπλών στόχων BAP αντί των καθορισμένων συντελεστών προτεραιότητας. Έγινε εισαγωγή των πλοίων σε

συγκεκριμένες ομάδες που χρησιμοποιούν διαφορετικές συναρτήσεις για να γίνει καλύτερη προσέγγιση των πραγματικών συμβολαίων προτεραιότητας.

Άλλες αξιοσημείωτες έρευνες στο BAP είναι: Hendriks et al. (2010), Buhrkal et al. (2011), Du et al. (2011), Hendriks et al. (2012), and Xu et al. (2012).

3.3.2 Ανάθεση των γερανών κρηπιδώματος (QCs)

Μετά την παραβολή των πλοίων στην παραβολή, σειρά παίρνει η τοποθέτηση των γερανών κρηπιδώματος προκειμένου να ξεκινήσει η διαδικασία της φορτοεκφόρτωσης η οποία γίνεται με την ανάθεση των γερανών κρηπιδώματος (QCA). Το QCA συχνά αναφέρεται και ως crane split (Stahlbock and Voß 2008). Σχετίζεται με την τοποθέτηση των γερανών κρηπιδώματος στα πλοία και στις θέσεις παραβολής. Αναθέτοντας τους γερανούς στα πλοία για την διαδικασία της φορτοεκφόρτωσης, πρέπει να εξασφαλίζεται η μέγιστη δυνατόν αποτελεσματικότητα μειώνοντας στα μέγιστα των χρόνων. Το QCA έχει σχέση αλληλεξάρτησης με το BAP καθώς το QCA λαμβάνει το BAP ως είσοδο, ενώ το QCA έχει άμεση επιρροή στο BAP. Έχουν γίνει αρκετές έρευνες πάνω στο QCA με σημαντικότερες τις παρακάτω:

- Peterkofsky and Daganzo (1990): σε αυτή την έρευνα τονίστηκε η μεγάλη σημασία του QCA και της επιρροής του στην συνολική απόδοση του λιμένα. Επίσης αναπτύχθηκε μια λύση υποκατάστασης και δέσμευσης των γερανών κρηπιδώματος προκειμένου να επιταχυνθεί η διαδικασία της φορτοεκφόρτωσης του πλοίου.
- Park and Kim (2003): διατυπώθηκε ένα προγραμματιστικό μοντέλο για το QCA με ποικίλα περιορισμών και σχεδιάστηκε μια λύση για το μαθηματικό μοντέλο που προέκυψε.
- Chang et al. (2010): εφαρμόστηκε ένα δυναμικό μοντέλο για την τοποθέτηση των γερανών κρηπιδώματος και του QCA. Εξαιτίας της πολυπλοκότητας, μια παράλληλη υβριδική γεννήτρια αλγορίθμου αναπτύχθηκε για να λυθεί το πρόβλημα και να καθοριστεί η επιρροή του χρησιμοποιώντας προγράμματα προσομοίωσης.
- Han et al. (2010): με χρήση προσομοίωσης καθορίστηκε το πρόβλημα του προγράμματος λαμβάνοντας υπόψη τον χρόνο άφιξης των πλοίων και τον χρόνο που απαιτείται για να γίνουν διαδικασίες που αφορούν τα εμπορευματοκιβώτια. Προτάθηκε ένα μοντέλο και μια γεννήτρια αλγορίθμων βασισμένη σε πρόγραμμα προσομοίωσης προκειμένου να αναδειχθεί η απαιτούμενη πρόληψη στο πρόγραμμα των γερανών κρηπιδώματος και της θέσης παραβολής.

3.3.3 Προγραμματισμός των γερανών κρηπιδώματος

Το QCA συσχετίζεται με την τοποθέτηση των γερανών κρηπιδώματος στις θέσεις παραβολής. Η επόμενη απόφαση που πρέπει να παρθεί είναι ο προγραμματισμός καθορίζοντας τον τρόπο λειτουργίας των γερανών κρηπιδώματος, κάτι που μελετάει το QCS. Το QCS ασχολείται με την φορτοεκφόρτωση των πλοίων σε συνεργασία με τα οχήματα που μεταφέρουν τα εμπορευματοκιβώτια από και προς τον χώρο στοίβασης. Ένα μοντέλο QCS έχει δημιουργηθεί με ορισμένους κανόνες που διέπουν αυτήν την διαδικασία. Αυτά είναι: η διαδικασία της εκφόρτωσης πάντα πρέπει να προηγείται της φόρτωσης, αλληλεξάρτηση μεταξύ γειτονικών

γερανών κρηπιδώματος, ένας γερανός κρηπιδώματος μπορεί να εκτελέσει μια διαδικασία την φορά και η σωστή εκτίμηση της χαμηλής ταχύτητας του γερανού. Στόχος του QCS είναι η ελαχιστοποίηση του χρόνου διεκπεραίωσης όλων αυτών των αναθέσεων. Έχουν γίνει αρκετές έρευνες πάνω στο QCS με σημαντικότερες τις παρακάτω:

- Lim et al. (2007): αναπτύχθηκε μια προγραμματιστική διαδικασία η οποία αναθέτει τα εμπορευματοκιβώτια σε μη διέλευσης (ο ένας γερανός δεν μπορεί να ξεπεράσει τον άλλο) γερανών κρηπιδώματος με στόχο την ελαχιστοποίηση το χρόνο ολοκλήρωσης του συνόλου των διαδικασιών. Το πρόβλημα αναλύθηκε σε 2 μέρη, αρχική τοποθεσία των γερανών κρηπιδώματος και η απόσταση του γερανού από την τοποθεσία που θα λάβει χώρα το task. Δυναμικός προγραμματισμός, αλγόριθμοι ακριβείς αναστροφής και προσομοίωση ανόπτησης χρησιμοποιήθηκαν προκειμένου να επιλυθεί το πρόβλημα.
- Kim and Park (2004): έγινε μελέτη του QCS με περιοριστικούς παραμέτρους την απόσταση ασφαλείας μεταξύ των γερανών κρηπιδώματος, το γεγονός ότι οι γερανοί κρηπιδώματος είναι μη προσπελάσιμοι ενώ εξετάστηκε και η εργασία σε ομάδες. Για την επίλυση του QCS προτάθηκε η χρήση ενός σύνθετου μοντέλου, ενός κλάδου και φράγματος αλγόριθμου και μια άπληστης τυχαίας προσαρμοστικής ευρετικής διαδικασίας (GRASP) προκειμένου να ελαχιστοποιηθεί ο συνολικός χρόνος. Πάνω σε αυτή την έρευνα, διάφορες μελέτες προστέθηκαν προκειμένου να την αναπτύξουν όπως αυτή του Moccia et al. (2006), όπου ανέπτυξε έναν αλγόριθμο branch and cut προκειμένου να προσεγγίσει καλύτερα από τον αλγόριθμο κλάδου και φράγματος που αρχικά είχε προταθεί., του Sammarra et al. (2007) όπου χρησιμοποιήθηκε tabu έρευνα προκειμένου να μελετηθεί από μια άλλη οπτική γωνία ενώ παράλληλα αξιολογήθηκε και έγινε σύγκριση με τον branch and cut αλγόριθμο και με το GRASP. Οι Bierwirth and Meisel (2009) έλυσαν το πρόβλημα των Kim and Park (2004) χρησιμοποιώντας μια περιορισμένη λύση χρησιμοποιώντας ένα πρόγραμμα μονής κατεύθυνσης (οι γερανοί κρηπιδώματος επιτρέπονταν να κινούνται μόνο προς μια κατεύθυνση), ενώ ο Kaveshgar et al.(2012) χρησιμοποίησε γενετικούς αλγόριθμους προκειμένου να επιλύσει το πρόβλημα της μονής κατεύθυνσης όπως είχε οριστεί από Bierwirth and Meisel (2009).
- Zhu and Lim (2006): προσεγγίστηκε το QCS με ένα προγραμματιστικό μοντέλο με περιορισμούς το γεγονός ότι οι γερανοί κρηπιδώματος είναι μη προσπελάσιμοι και ότι απαιτείται όλοι οι γερανοί κρηπιδώματος να παραμείνουν στην ίδια παραβολή μέχρι να τελειώσει η διαδικασία φορτοεκφόρτωσης της με στόχο την ελαχιστοποίηση του χρόνου.
- Lee et al. (2008b): έγινε επικέντρωση στο μοναδικό πλοίο χρησιμοποιώντας προτεραιότητες για το εκάστοτε πλοίο και μη προσπελάσιμες διεργασίες. Με χρήση προβλήματος μικτού ακέραιου προγραμματισμού μοντέλου και γενετικούς αλγόριθμους ελαχιστοποιήθηκε ο χρόνος διεκπεραίωσης των εργασιών σε κάθε πλοίο.
- Tavakkoli-Moghaddam et al. (2009): έγινε προσπάθεια επίλυσης του QCA και του QCS προσομοιωτικά. Με χρήση προβλήματος μικτού ακέραιου προγραμματιστικού μοντέλου και γενετικούς αλγόριθμους αναλύθηκε στο πραγματικό του μέγεθος το πρόβλημα και οπτικοποιήθηκε ο συνολικός χρόνος για τα πλοία και τους γερανούς κρηπιδώματος. Μετά από σύγκριση των δύο μεθόδων αποδείχθηκε ότι οι γενετικοί αλγόριθμοι είχαν ακριβέστερα αποτελέσματα σε σχέση με το πρόβλημα μικτού ακέραιου προγραμματισμού μοντέλου μοντέλο.

- Meisel (2011): με ιδιαίτερη έμφαση στον περιορισμό των γερανών κρηπιδώματος σχετικά με το χρονικό παράθυρο, χρησιμοποιώντας ένα κατώτερο όριο και μια μονής διαδρομής ευρεστικής έρευνας. Μια μαθηματική μέθοδος εφαρμόστηκε προκειμένου να επιτευχθεί ελαχιστοποίηση του χρόνου παραμονής του πλοίου στην παραβολή.
- Legato et al. (2012): δημιουργήθηκε ένα μοντέλο που συμπεριλαμβάνει ποικίλους χρόνους για τις διαδικασίες των γερανών κρηπιδώματος για μη ταυτόσημους και μη προσπελάσιμους γερανούς κρηπιδώματος, χρονικά παράθυρα και ξεχωριστά προγράμματα. Αναπτύχθηκε ένα MILP, μια Lagrangian χαλάρωσης προσέγγιση για να βρεθεί το κατώτερο όριο, και ένας κλάδου και φράγματος αλγόριθμος βασισμένος στο Bierwirth and Meisel (2009) προκειμένου να λυθεί το πρόβλημα του προγράμματος

3.4 Χώρος μεταφοράς εντός του λιμένα

Ο χώρος μεταφοράς εντός του λιμένα περιλαμβάνει τους χώρους μεταξύ τον χώρο παραβολής και τον χώρο εναπόθεσης όπως και τους χώρους μεταξύ τον χώρο εναπόθεσης και την ενδοχώρα. Οι αποφάσεις που πρέπει να ληφθούν στον χώρο μεταφοράς αφορούν την μείωση του χρόνου μεταφοράς προκειμένου να εξασφαλιστεί ο παραγωγικότητα των γερανών κρηπιδώματος και γερανογεφυρών στοιβασίας. Σε αυτό το σημείο είναι σημαντικό να αναφερθεί η μεγάλη σημασία του καλού συγχρονισμού στον λιμένα καθώς η συνολική παραγωγικότητα του λειτουργεί ως αλυσίδα, κάτι που σημαίνει, η συνολική παραγωγικότητα εξισώνεται με την παραγωγικότητα του πιο αδύναμου κρίκου εκ των χώρο παραβολής, χώρο μεταφοράς και τον χώρο εναπόθεσης. Τα προβλήματα που καλούνται να επιλυθούν στην περιοχή του χώρου μεταφοράς είναι η κατανομή των οχημάτων μεταφοράς στους γερανούς κρηπιδώματος, η κατανομή και ο προγραμματισμός των οχημάτων μεταφοράς και ο καθορισμός της διαδρομής των οχημάτων μεταφοράς. Οι στόχοι που μπαίνουν σε αυτόν τον τομέα είναι η ελαχιστοποίηση του χρόνου μεταφοράς, η ελαχιστοποίηση των περιπτώσεων που οι γερανοί κρηπιδώματος παραμένουν αδρανείς ελαχιστοποιώντας τον στόλο των οχημάτων μεταφοράς. Έχουν γίνει αρκετές έρευνες πάνω στον χώρο μεταφοράς με σημαντικότερες τις παρακάτω:

- Nishimura et al. (2005): έγινε επικέντρωση στον έλεγχο των οχημάτων μεταφοράς και μελετήθηκε ο καθορισμός της διαδρομής τους. Η μελέτη έγινε πάνω σε ένα στατικό περιβάλλον όπου τα οχήματα μεταφοράς κινούνταν κατά μήκος μια κυκλικής διαδρομής με περιορισμένες δυνατότητες επαναπροσδιορισμού της διαδρομής.
- van der Meer (2000): έγινε μια πρώτη ολοκληρωμένη εισαγωγή των AGV συστημάτων που αφορούν την αποστολή τους.
- Levitin and Abezgauz(2003): μελετήθηκαν AGVs πολλαπλών φορτώσεων χρησιμοποιώντας περιορισμούς φόρτωσης LIFO. Η μελέτη ασχολήθηκε μόνο με ένα όχημα μεταφοράς σε απομόνωση.
- Chan (2001): μελετήθηκαν AGV πολλαπλών φορτώσεων με χρήση προσομοιωτή για να γίνει σύγκριση δύο εναλλακτικών στρατηγικών για την περίπτωση που υπάρχουν μονής φόρτωσης και πολλαπλών φορτώσεων AGVs και εμπορευματοκιβώτια που περιμένουν για να μετακινηθούν. Παρόλα αυτά η θεώρηση που έγινε ότι όλα τα οχήματα μεταφοράς είναι σε ετοιμότητα για να μετακινηθούν σε χρόνο μηδέν δεν αντιπροσωπεύει πλήρως της διεργασίες σε ένα σταθμό εμπορευματοκιβωτίων.

- Grunow et al. (2004): μελετήθηκε η ανάθεση σε μονής και διπλής φόρτωσης AGVs σε ένα αυτοματοποιημένο σταθμό εμπορευματοκιβωτίων παρόμοιο με τα ECT Rotterdam και CTA Hamburg. Τα AGVs μπορούν να παραλάβουν είτε ένα 40'/45' εμπορευματοκιβώτιο είτε δύο 20' εμπορευματοκιβώτια. Σε αυτή τη μελέτη αγνοήθηκαν οι διαδρομές και ο έλεγχος κυκλοφορίας μελετώντας μόνο την διαδικασία της ανάθεσης. Αφού πρώτα εισάχθηκαν η ιδέες της πλήρους και μερικής διαθεσιμότητας των AGVs, οι μελετητές ανέπτυξαν ένα ευρεστικό επίσπευσης αλγόριθμο. Αυτός ο αλγόριθμος εισάγεται όταν μια νέα ανάθεση γεννιέται στο πλαίσιο της πρόβλεψης του χρονοπαραθύρου. Για κάθε μερικώς διαθέσιμο AGV, ο αλγόριθμος παράγει τρεις πιθανές διαδρομές για το AGV, όπου η κάθε μια αντιστοιχεί σε μια διαφορετική εντολή για διεργασίες παραλαβής και εναπόθεσης του νέου εμπορευματοκιβωτίου σε συνάρτηση με την διεργασία εναπόθεσης του ήδη φορτωμένου εμπορευματοκιβωτίου στο AGV. Ένα αξιοσημείωτο εύρημα σε αυτή την μελέτη ήταν τα μεγάλα πλεονεκτήματα που προσφέρει η διπλή φόρτωση των AGV, ειδικά σε μεγάλους λιμένες.
- Nguyen and Kim (2009): μελετήθηκε η ανάθεση των ALVs (AGVs με δυνατότητα ανύψωσης), αναπτύσσοντας ένα προβλήματος μικτού ακέραιου προγραμματισμού μοντέλου όπου αναθέτει τις εργασίες μεταφοράς στα ALVs με τον βέλτιστο τρόπο
- Petering (2010): με χρήση προσομοιωτή έγινε έλεγχος σε πραγματικό χρόνο των διπλής φόρτωσης ελκυστήρα σε ένα τερματικό σταθμό

3.5 Χώρος στοιβάσις

Ο χώρος εναπόθεσης πολύ συχνά αποτελεί τον χώρο συμφόρησης σε ένα σταθμό εμπορευματοκιβωτίων (Li et al. 2009). Η αποτελεσματικότητα σε αυτή την περιοχή επηρεάζει σε μεγάλο βαθμό την συνολική αποτελεσματικότητα ολόκληρου του σταθμού εμπορευματοκιβωτίων. Στην περιοχή εναπόθεσης τα προβλήματα απόφασης που καλούνται να επιλυθούν για την εύρυθμη λειτουργία συμπεριλαμβάνουν την ανάθεση των blocks του χώρου στοιβάσις για τα πλοία που καταφτάνουν στην παραβολή, τον καθορισμό της ακριβούς τοποθεσίας του κάθε εμπορευματοκιβωτίου, την επανατοποθέτηση των εμπορευματοκιβωτίων όταν το απαιτούν οι περιστάσεις (remarshaling) και ο προγραμματισμός των γερανογεφυρών στοιβάσις (YCS). Το YCS μπορεί να χωριστεί σε δύο υποπροβλήματα, το πρόβλημα που αφορά την τοποθέτησης των γερανογεφυρών στοιβάσις στα διάφορα block (YCS-I), και στο πρόβλημα του ατομικού προγραμματισμού της κάθε γερανογέφυρας στοιβάσις σχετικά με τις διαδικασίες αποθήκευσης και επανάκτησης (YCS-II). Έχουν γίνει αρκετές έρευνες πάνω στον χώρο στοιβάσις με σημαντικότερες τις παρακάτω:

Μελέτες σχετικά με το YCS-I

- Linn et al. (2003): εισήχθηκε ένα πρόβλημα μικτού ακέραιου προγραμματισμού μοντέλου για να βρεθεί και να βελτιστοποιηθεί η παράταξη των γερανογεφυρών στοιβάσις κατά μήκος της περιοχής εναπόθεσης περιλαμβάνοντας προσπελάσιμες κινήσεις. Στόχος αυτής της παράταξης ήταν να κινούνται οι αδρανείς γερανογέφυρες στοιβάσις σε άλλα block στην αρχή του επόμενου προγραμματισμένου ορίζοντα προκειμένου να ισομοιράζονται οι φόρτοι εργασίας
- Linn and Zhang (2003): μελετήθηκε το πρόβλημα επιλογής του βέλτιστου αριθμού γερανογεφυρών στοιβάσις, με προσπελάσιμες κινήσεις γερανογεφυρών, σε κάθε block

του χώρου στοιβασίας για προγραμματισμένο ορίζοντα, με χρήση μικτού ακέραιου προγραμματισμού μοντέλου και ενός ελαχίστου κόστους ευρεστικό αλγόριθμο για να εξισορροπηθεί ο φόρτος εργασίας μεταξύ των γερανογεφυρών στοιβασίας.

- Legato et al. (2009): μελετήθηκε η αλληλεπίδραση μεταξύ των γερανογεφυρών στοιβασίας σε ένα block του χώρου στοιβασίας με σημείο αναφοράς τις γραμμικές και προσπελάσιμες κινήσεις των γερανογεφυρών στοιβασίας. Ένα ακέραιο προγραμματιστικό μοντέλο και μια μέθοδος προσομοίωσης διακριτικού γεγονότος (discrete event simulation) προτάθηκε προκειμένου να κατανεμηθούν οι γερανογέφυρες στοιβασίας στα κατάλληλα block, και να αξιολογηθούν οι επιδόσεις από πέντε διαφορετικές πολιτικές για τους προσπελάσιμους γερανούς. Τα αριθμητικά δεδομένα από τα αποτελέσματα έδειξαν ότι η μεταφορά πρέπει να γίνεται στο πλησιέστερο block του χώρου στοιβασίας καθώς υπήρξαν πολύ μεγάλες διαφορές από τις άλλες πολιτικές και κανόνες που εφαρμόστηκαν.
- He et al. (2010): η μελέτη είχε σκοπό την μεγιστοποίηση της παραγωγικότητας των γερανογεφυρών στοιβασίας. Προτάθηκε ένα μοντέλο που ελαχιστοποιούσε τις καθυστερήσεις και τον συνολικό χρόνο των κινήσεων των γερανογεφυρών στοιβασίας μέσα στο block με χρήση ενός υβριδικού αλγορίθμου βασισμένο σε επτά ευρεστικούς και έναν παράλληλο γενετικό αλγόριθμο. Ο Yan et al. (2011) βασιζόμενος σε γνώσεις γύρο από το YCS βελτίωσε τα αποτελέσματα της μελέτης αυτής.
- Chang et al. (2011): παρουσιάστηκε ένα μοντέλο για το YCS το οποίο περιελάμβανε δύο γερανογέφυρες στοιβασίας ανά block με προσπελάσιμους γερανούς και ομάδες εμπορευματοκιβωτίων με σκοπό την ελαχιστοποίηση της συνολικής καθυστέρησης στα blocks. Αναπτύχθηκε μια δυναμική κυλιόμενου ορίζοντα διαδικασία απόφασης και ένα ακέραιο προγραμματιστικό μοντέλο για να επιλυθεί αυτό το πρόβλημα. Ένας ολοκληρωμένος ευρεστικός αλγόριθμος και ένα προσομοιωτικό μοντέλο, σε συνδυασμό με ένα γενετικό αλγόριθμο που λάμβανε τιμές από τον ευρεστικό αλγόριθμο, χρησιμοποιήθηκαν για να επιλυθεί το πρόβλημα του προγραμματισμού. Τα υπολογιστικά αποτελέσματα όπως και τα αντίστοιχα από την προσομοίωση ήταν πολύ καλύτερα από αυτά που είχε προτείνει η μελέτη του He et al. (2010) .
- Chen and Langevin (2011): έγινε μελέτη με στόχο την ελαχιστοποίηση του χρόνου χειρισμού όλων των εξερχόμενων εμπορευματοκιβωτίων. Αναπτύχθηκε ένα μικτό ακέραιο προγραμματιστικό μοντέλο, με περιορισμούς αποστάσεων πέντε yard-bays, παρεμβάσεις μεταξύ γειτονικών γερανογεφυρών στοιβασίας στο ίδιο block και μέγιστο δύο γερανογέφυρες στοιβασίας στο ίδιο block, ένας γενετικός αλγόριθμος και μια tabu έρευνα. Τα αποτελέσματα από την έρευνα αυτή ήταν κοντά στα βέλτιστα.

Μελέτες σχετικά με το YCS-II

- Vis and Carlo (2010): κατασκευάστηκε ένα μαθηματικό μοντέλο για την ελαχιστοποίηση του χρόνου εξυπηρέτησης και των δύο γερανογεφυρών στοιβασίας. Επίσης εισήχθηκε ένας αλγόριθμος ο οποίος αντλεί ένα κατώτερο όριο για το χρόνο εξυπηρέτησης, ενώ ένας ανόπτησης ευρεστικός προσομοιωτής προτάθηκε για την επίλυση
- Stahlbock and Voß (2010): διερευνήθηκε η επιρροή των γερανογεφυρών επί σιδηροτροχιών (DMRG) με χρήση λογισμικού προσομοίωσης. Αξιολογήθηκαν διαφορετικοί online αλγόριθμοι για τον προσδιορισμό της αλληλουχίας και τον προγραμματισμό των εργασιών για τους αυτοματοποιημένους DMRG σε ένα block αποθήκευσης.

- Park et al. (2010): έγινε χρήση μίας ευρεστικής και μίας τοπικής έρευνας σε πραγματικό χρόνο προγραμματισμού για τους DRMG που δουλεύουν σε ένα block ενός αυτοματοποιημένου σταθμού εμπορευματοκιβωτίων. Χρησιμοποιήθηκε κυλιόμενου προγραμματιστικού ορίζοντα για τις περιπτώσεις που ένας γερανός τελειώνει την εργασία του, κάτι που είχε ως αποτέλεσμα να προσομοιώνει καλύτερα τις συνθήκες πραγματικού χρόνου. Πιο συγκεκριμένα εντοπίζονταν τα εμπορευματοκιβώτια που απαιτούσαν άμεση μεταφορά, κάτι το οποίο εντάσσει στην μέθοδο την έννοια της συνεργασία μεταξύ των γερανών κρηπιδώματος ενώ παράλληλα ισομοιράζονταν οι εργασίες μεταξύ των γερανών.
- Cao et al. (2008): μελετήθηκε η στρατηγική φόρτωσης των εξερχόμενων εμπορευματοκιβωτίων από την περιοχή εναπόθεσης. Ένα ολοκληρωμένο προγραμματιστικό μοντέλο, ένας ευρεστικός και ένας προσομοίωσης ανόπτησης αλγόριθμος σε συνδυασμό με ένα ευρεστικό συνδυαστικό προγραμματισμό προτάθηκαν για την αντιμετώπιση του προβλήματος.
- Dorndorf and Schneider (2010): προγραμματίστηκαν τριπλής προσέλασης γερανοί (TCSCs) σε ένα αυτοματοποιημένο σταθμό εμπορευματοκιβωτίων με ασύγχρονες παραδόσεις στον χώρο μεταφοράς. Αντιμετωπίστηκε το πρόβλημα ως πρόβλημα απευθείας βελτιστοποίησης και κατασκευάστηκε ένα πρόγραμμα για συγκεκριμένο ορίζοντα για το πότε μια νέα ανάθεση ξεκινάει ή τελειώνει
- Gharehgozli et al. (2014b): μοντελοποιήθηκε το προγραμματιστικό πρόβλημα της μιας γερανογέφυρας στοιβάδας, για εργασίες τόσο αποθήκευσης, όσο και ανάκτησης σε ένα block σαν ένα συνεχόμενου χρόνου ακεραίων αριθμών προγραμματιστικό πρόβλημα. Η δύο φάσεων μέθοδος επίλυσης, η οποία χρησιμοποίησε εγγενής ιδιότητες από το μοντέλο, υιοθετήθηκε προκειμένου να επιλυθεί το πρόβλημα βέλτιστα. Η πρώτη φάση περιελάμβανε έναν συγχωνευτικό αλγόριθμο ο οποίος προσπάθησε να περιορίσει τις μετακινήσεις των γερανών λαμβάνοντας υπόψη την χαμηλή τους ταχύτητα, ενώ στη δεύτερη φάση χρησιμοποιήθηκε ένας κλάδου και φράγματος αλγόριθμος για να επιτευχθεί το βέλτιστο αποτέλεσμα.

3.6 Σύνθεση των επιμέρους υποσυστημάτων ενός λιμένα

Το πραγματικό πρόβλημα που απαιτεί επίλυση είναι αυτό που αφορά το συνολικό της λειτουργίας του σταθμού εμπορευματοκιβωτίων καθώς αυτό θα καθορίσει την αποτελεσματικότητά του σταθμού εμπορευματοκιβωτίων. Τα παραπάνω υποπροβλήματα δημιουργήθηκαν με στόχο την απλούστευση του συνολικού προκειμένου να είναι πιο εύκολα διαχειρίσιμο επικεντρώνοντας σε συγκεκριμένα σημεία και περιοχές. Δηλαδή, αν ένα υποπρόβλημα επιλυθεί στον βέλτιστο βαθμό, για παράδειγμα του χώρο παραβολής, δεν θα έχει κανένα αποτέλεσμα αν ο χώρος μεταφοράς δεν λειτουργεί σωστά και δεν είναι απόλυτα συντονισμένη, καθώς οι γερανοί κρηπιδώματος θα αναγκάζονται να περιμένουν, με αποτέλεσμα κάθε προσπάθεια ελαχιστοποίησης του χρόνου να πηγαίνει στράφι αφού όπως προαναφέρθηκε ένας σταθμός εμπορευματοκιβωτίων λειτουργεί σαν αλυσίδα καθορίζοντας την αποτελεσματικότητα του από το κρίσιμο σημείο του. Σκεπτόμενοι λοιπόν, ότι το ίδιο πρόβλημα θα δημιουργούνταν και στην περίπτωση που και η περιοχή εναπόθεσης λειτουργούσε βέλτιστα, καταλήγουμε στο συμπέρασμα ότι η συνολική παραγωγικότητα του σταθμού

εμπορευματοκιβωτίων καθορίζεται από την περιοχή μεταφοράς που όμως εξαρτάται άμεσα από τον χώρο παραβολής και την περιοχή εναπόθεσης όπως και από την δικιά της λειτουργία. Έτσι ο καλός συγχρονισμός των τριών αυτών κρίσιμων σημείων για την λειτουργία του σταθμού εμπορευματοκιβωτίων καθίσταται μείζονος σημασίας. Μια δυσλειτουργία στον συγχρονισμό αυτό είναι ορατή σε ένας σταθμός εμπορευματοκιβωτίων βλέποντας, ουρές από εμπορευματοκιβώτια που περιμένουν να μετακινηθούν, οι γερανοί κρηπιδώματος ή οι γερανογέφυρες στοιβάσις να παραμένουν αδρανείς καθώς η περιοχή μεταφοράς δεν μπορεί να διαχειριστεί τον όγκο εργασίας που μπορεί να επιτευχθεί από τις άλλες δύο περιοχές και τέλος να υπάρχουν αδρανείς οχήματα μεταφοράς, κάτι που σημαίνει ότι έγιναν ανώφελες επενδύσεις χωρίς πραγματικά να επιφέρουν κάποια αύξηση στην απόδοση του λιμανιού. Έχουν γίνει αρκετές έρευνες πάνω στο συνολικό πρόβλημα με σημαντικότερες τις παρακάτω:

- Meersmans (2002): μελετήθηκε ο ολοκληρωμένος προγραμματισμός των αυτόματων γερανών στοιβάσις (ASCs), AGVs και των γερανών κρηπιδώματος σε ένα σταθμό εμπορευματοκιβωτίων χρησιμοποιώντας κλάδου και φράγματος, και δεσμευτικούς ευρεστικούς αλγόριθμους.
- Bish (2003): μελετήθηκαν τα συνδυαστικά προβλήματα της φορτοεκφόρτωσης των εμπορευματοκιβωτίων από τους γερανούς κρηπιδώματος, της ανάθεσης των οχημάτων μεταφοράς και της ακριβής τοποθεσίας των εμπορευματοκιβωτίων στην περιοχή εναπόθεσης. Στόχος της μελέτης αυτής ήταν η ελαχιστοποίηση του μέγιστου χρόνου εξυπηρέτησης ενός συνόλου πλοίων χρησιμοποιώντας έναν ευρεστικό αλγόριθμο.
- Chen (2007): προτάθηκε ένα ακέραια μοντέλο για τον προγραμματισμό διαφόρων τύπων μηχανημάτων την ίδια στιγμή με χρήση προσομοιωτή. Το πρόβλημα μεθοδεύτηκε σαν ένα προγραμματιστικό πρόβλημα και επιλύθηκε με χρήση ενός tabu έρευνας αλγορίθμου
- Lee et al. (2008a): μελετήθηκε το ολοκληρωμένο πρόγραμμα των γερανών και των οχημάτων μεταφοράς. Το πρόβλημα μεθοδεύτηκε με χρήση ενός προβλήματος μικτού ακέραiou προγραμματισμού μοντέλου και επιλύθηκε με χρήση γενετικών αλγορίθμων με στόχο να ελαχιστοποιηθεί ο χρόνος εξυπηρέτησης των γερανών.
- Yuan et al. (2011): προτάθηκε ένα ολοκληρωμένο σχέδιο διαδρομών για τα οχήματα πλαίσια και η αρχική τοποθεσία τους λαμβάνοντας υπόψη και της διεργασίες των γερανών κρηπιδώματος και των οχημάτων μεταφοράς. Προτάθηκε ένας συνολικός ομαδικός προγραμματισμός για την επίλυση του προβλήματος.
- Chen et al. (2012): μελετήθηκαν τα BAP, QCA και QCS σε ένα ολοκληρωμένο μοντέλο και εφαρμόστηκαν Benders Cut αλγόριθμοι για την επίλυση του προβλήματος.
- Homayouni et al. (2013): μεθοδεύτηκε ένα μοντέλο για την βελτιστοποίηση του συντονισμένου προγραμματισμού των γερανών και οχημάτων στο σταθμό εμπορευματοκιβωτίων. Οι στόχοι που καθορίστηκαν ήταν η ελαχιστοποίηση του χρόνου μετακινήσεων των φορτηγών και τις αναμονής των γερανών.
- Chen et al. (2013): μελετήθηκε με την χρήση προσομοιωτή ο προγραμματισμός των γερανών κρηπιδώματος, οχημάτων μεταφοράς και γερανογεφυρών στοιβάσις έτσι ώστε ο χρόνος εξυπηρέτησης των πλοίων να ελαχιστοποιηθεί. Το πρόβλημα αντιμετωπίστηκε με ένα περιοριστικό προγραμματιστικό μοντέλο. Μια τριών επιπέδων διαδικασία αναπτύχθηκε προκειμένου να επιλυθούν τα διαφορετικά υποπροβλήματα

- Tierney et al. (2014): αναπτύχθηκε ένα ολοκληρωμένο προγραμματιστικό μοντέλο προκειμένου να αναλυθούν οι μεταφορές των εμπορευματοκιβωτίων εντός του λιμένα. Για την επίλυση του προβλήματος προτάθηκε μια δύο επιπέδων διαδικασία.
- Homayouni et al. (2014): χρησιμοποιήθηκαν γενετικοί αλγόριθμοι προκειμένου να επιλυθεί ολοκληρωτικά το πρόβλημα του προγραμματισμού των γερανών κρηπιδώματος, AGVs και γερανογεφυρών στοιβάσεως. Αποδείχθηκε ότι ο γενετικός αλγόριθμος έχει καλύτερα αποτελέσματα από τον προσομοίωσης ανόπτησης αλγόριθμο

4 ΑΝΑΠΤΥΞΗ ΜΟΝΤΕΛΟΥ ΛΙΜΕΝΙΚΩΝ ΕΓΚΑΤΑΣΤΑΣΕΩΝ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ ΛΟΓΙΣΜΙΚΟΥ ANYLOGIC

4.1 Το πρόγραμμα προσομοίωσης Anylogic

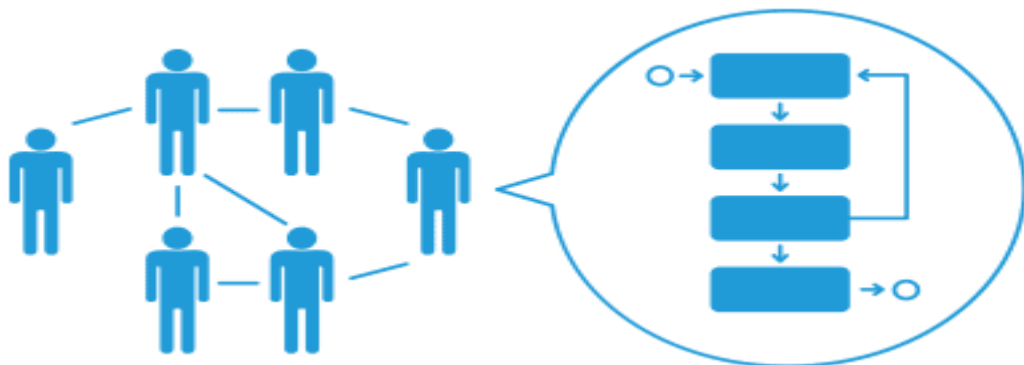
4.1.1 Περιβάλλον μοντελοποίησης με δυνατότητες χρήσης πολλών μεθόδων

Υπάρχουν τρεις βασικές μεθοδολογίες που χρησιμοποιούνται για την κατασκευή ενός δυναμικού επιχειρησιακού προσομοιωτικού μοντέλου, το Discrete Event, το System Dynamics και το Agent based. Τα πρώτα δύο αναπτύχθηκαν τις δεκαετίες του 1950 και 1960 υιοθετώντας ένα σύστημα πολλών επιπέδων προκειμένου να γίνεται η επίβλεψη. Η προσέγγιση του Agent Based αναπτύχθηκε τελευταία, είναι μια “από κάτω προς τα πάνω” (bottom up) προσέγγιση όπου ο ερευνητής εστιάζει ξεχωριστά στην συμπεριφορά του κάθε “πράκτορα”.

- System Dynamics: Υποθέτει ένα υψηλό επίπεδο αφάιρεσης και κατά κύριο λόγο χρησιμοποιείται για την επίλυση πολλών επίπεδων στρατηγικών προβλημάτων
- Discrete Event: Κατά βάση χρησιμοποιείται για την προσομοίωση προβλημάτων που σχετίζονται με την διεκπεραίωση διεργασιών και τακτικών επιπέδων
- Agent based: Χρησιμοποιείται σε όλα τα επίπεδα, ένας agent μπορεί να προσομοιώνει ένα όχημα, ένα ρομπότ, έναν άνθρωπο και άλλα.

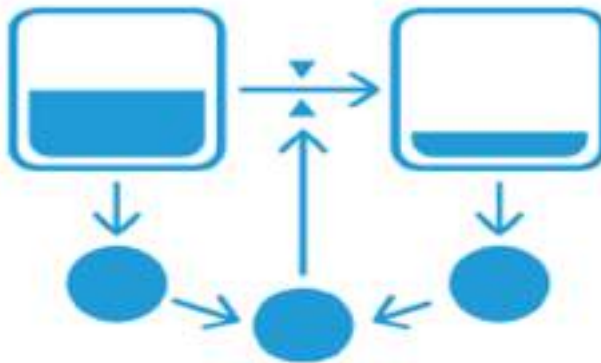
Επιλογή της κατάλληλης μεθοδολογίας για την μοντελοποίηση

- Εάν υπάρχουν λεπτομερείς πληροφορίες για κάθε “πράκτορα”, προτείνεται η εφαρμογή της τεχνικής agent-based.



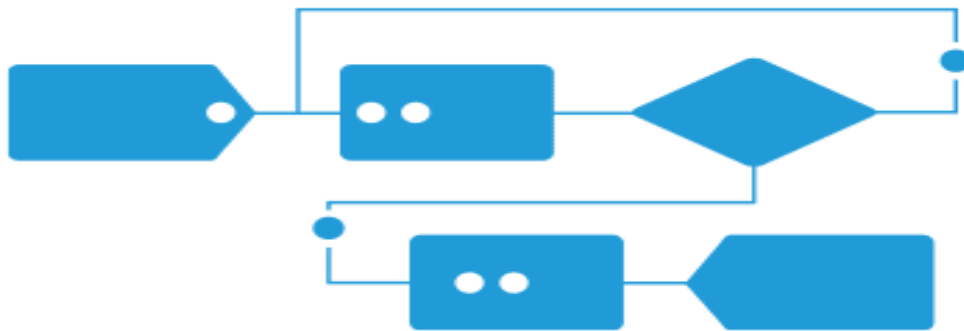
Εικόνα 17 Agent-based προσέγγιση [πηγή: anylogic/ features [http://www.anylogic.com/features/]]

- Εάν υπάρχουν μόνο αλληλεξαρτήσεις, προτείνεται η μια System Dynamics προσέγγιση



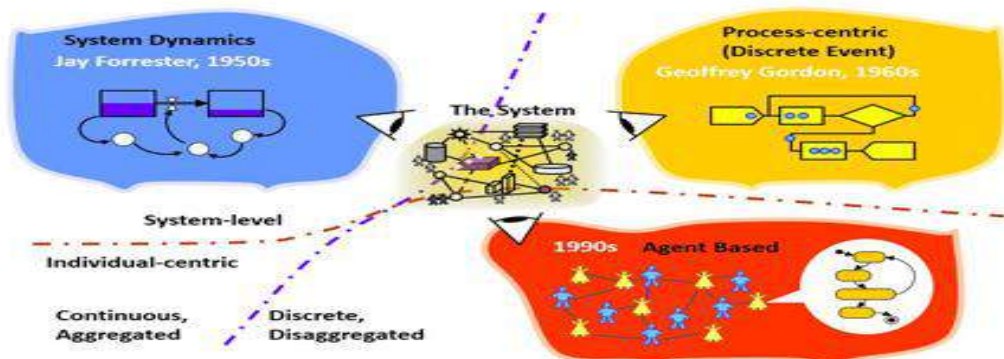
Εικόνα 18 System Dynamics προσέγγιση [πηγή: anylogic/ features [http://www.anylogic.com/features/]

- Εάν το σύστημα μπορεί να περιγραφεί σαν μια διαδικασία, προτείνεται μια Discrete Event προσέγγιση



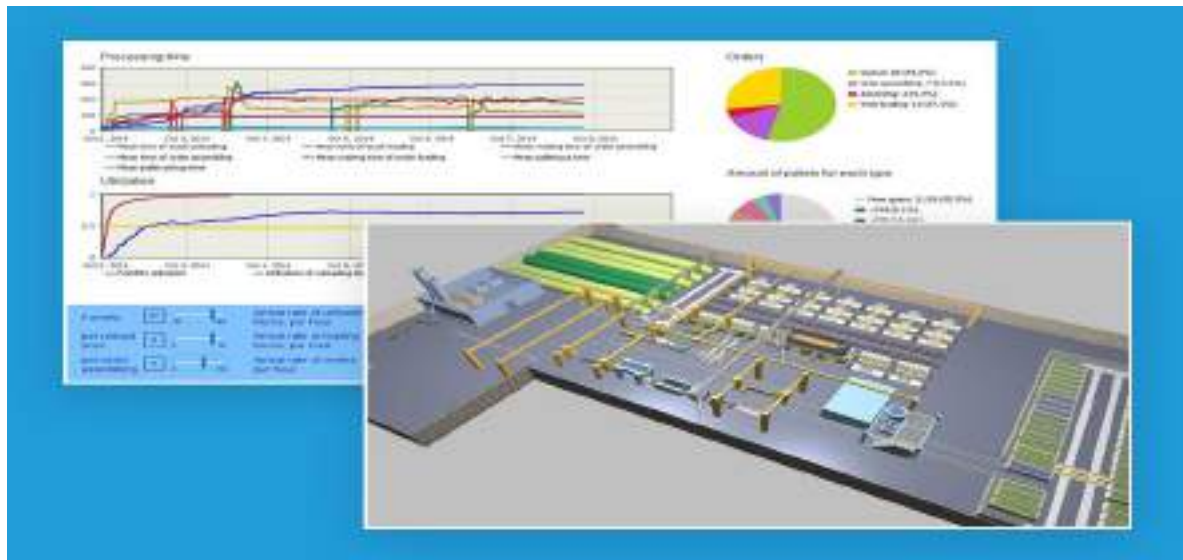
Εικόνα 19 Discrete Event προσέγγιση [Πηγή: anylogic/ features [http://www.anylogic.com/features/]

- Εάν το σύστημα είναι πιο περίπλοκο μπορεί να γίνει ένας συνδυασμός των παραπάνω προσεγγίσεων



Εικόνα 20 Συνδυασμός προσεγγίσεων [πηγή: anylogic/ features [http://www.anylogic.com/features/]

4.1.2 Αναπαράσταση και οπτικοποίηση



Εικόνα 21 4.1.2 Αναπαράσταση και οπτικοποίηση [πηγή: anylogic/features/ <http://www.anylogic.com/features/>]

- Παρέχεται η δυνατότητα να γίνει μετατροπή του flowchart σε μια διαδραστική ταινία με υψηλής ποιότητας γραφικών 3D ή 2D ανάλογα με τις απαιτήσεις
- Παρουσίαση των μοντέλων στους ενδιαφερόμενους με έναν οπτικά ελκυστικό και πιο κατανοητό τρόπο.
- Παροχή μία πληθώρας γραφικών αντικειμένων για την οπτικοποίηση οχημάτων, κτηρίων, υπαλλήλων και άλλα πολλά προκειμένου να επιτευχθεί η καλύτερη προσομοίωσης των ατομικών αναγκών του χρήστη
- Παροχή δυνατότητας εισαγωγής 3D μοντέλων, εικόνων, σχεδίων τύπου CAD και shape files στην προσομοίωση
- Δυνατότητα να γίνει το μοντέλο διαδραστικό παρέχοντας στον χρήστη να κάνει διαισθητική πλοήγηση, αλλά και να αλλάζει διάφορα στοιχεία κατά την διαδικασία την προσομοίωσης
- Παροχή δυνατότητας για συμπλήρωση περιεκτικών πινάκων ελέγχου διαχείρισης στις προσομοιώσεις

4.1.3 Εξειδικευμένες βιβλιοθήκες εργαλείων



Εικόνα 22 Εξειδικευμένες βιβλιοθήκες εργαλείων [πηγή: anylogic/ features [http://www.anylogic.com/features/]

Το πρόγραμμα Anylogic παρέχει μια μεγάλη γκάμα από εξειδικευμένα εργαλεία προσομοίωσης προκειμένου να γίνεται ευκολότερη η διεκπεραίωση του μοντέλου. Πιο συγκεκριμένα τα εργαλεία αυτά είναι:

- **Βιβλιοθήκη για διαδικαστική μοντελοποίηση (Process Modeling Library):** αφορά γενικές επιχειρησιακές διαδικασίες ή διάφορες εργασίες σε σειρά (workflows).
- **Βιβλιοθήκη για προσομοίωση υγρών (fluid library):** αφορά την προσομοίωση υγρού χύδην φορτίου, μεταφορές υγρών στην βιομηχανία όπως πετρέλαιο, αερίου ή ακόμα και προϊόντα εξόρυξης.
- **Βιβλιοθήκη σιδηροδρόμων (rail library):** αφορά τις μεταφορές στους σιδηροδρόμους και τερματικούς σταθμούς
- **Βιβλιοθήκη με βάση το ανθρώπινο στοιχείο (αναπαράσταση πεζών) (pedestrian library):** αφορά τις ροές ανθρώπων σε αεροδρόμια, στάδια, σταθμούς, ή ακόμα και σε εμπορικά κέντρα.
- **Βιβλιοθήκη συγκοινωνιών (road traffic library):** αφορά την κίνηση στους δρόμους των αυτοκινήτων, φορτηγών και λεωφορείων και χώρους στάθμευσης
- **Βιβλιοθήκη διαχείρισης υλικών (material handling library):** (θα εισαχθεί το 2017) αφορά βιομηχανικές διαδικασίες όπως και αποθήκευσης.

4.1.4 Ενσωμάτωση χάρτη GIS



Εικόνα 23 Ενσωμάτωση GIS χάρτη [πηγή: anylogic/ features [http://www.anylogic.com/features/]

Το Anylogic παρέχει πολλές δυνατότητες που σχετίζονται με τον GIS χάρτη, όπως:

- Οι χάρτες μπορούν να χρησιμοποιηθούν σε συστήματα όπως αλυσίδες εφοδιασμού (supply chain), logistics και άλλες περιπτώσεις που πρέπει να ληφθούν πληροφορίες τοποθεσίας, δρόμων, διαδρομών και περιοχών.
- Γίνεται χρήση του χάρτη με παρόμοιο τρόπο με τους Google Maps, κάτι που κάνει ευκολότερη την χρήση του, καθώς εύκολα λαμβάνονται πληροφορίες από τον χάρτη για την τοποθεσία περιοχών, πόλεων, δρόμων, νοσοκομείων κ.α. Η λήψη αυτών των δεδομένων μπορεί να γίνει και κατά την διάρκεια της μοντελοποίησης αλλά και κατά την διάρκεια της προσομοίωσης.
- Τα εργαλεία που χρησιμοποιούνται για την μοντελοποίηση εύκολα μπορούν να τοποθετηθούν στον χάρτη (π.χ. ένας agent) και να κινηθούν κατά μήκος υπαρχων δρόμων ή άλλων πραγματικών δεδομένων (π.χ. θαλάσσιων διαδρομών).
- Παρέχεται η δυνατότητα χρήσης ελεύθερων χαρτών που παρέχει το Anylogic, ή ακόμα και χρήση άλλων χαρτών που παρέχονται από άλλους παρόχους σε μορφή shapfile ή tile.

4.1.5 Δυνατότητα αυτοματοποιημένης λειτουργίας



Εικόνα 24 Δυνατότητα αυτοματοποιημένης λειτουργίας [Πηγή: [anylogic/features](http://www.anylogic.com/features/)]

- Παρέχεται η δυνατότητα για ανάπτυξη μοντέλων τύπου Monte Carlo, για ανάλυση ευαισθησίας (Sensitivity Analysis) και αυτόματη μεταβολή τιμών σε παραμέτρους του μοντέλου (π.χ. αριθμός οχημάτων μεταφοράς) προκειμένου να εξεταστεί το κατά πόσο η τυχαιότητα και η αλλαγή των παραμέτρων επηρεάζουν την συμπεριφορά του μοντέλου
- Παρέχεται η δυνατότητα βελτιστοποίησης, κάτι που επιτρέπει την εύρεση των καλύτερων λύσεων για επιχειρηματικές προκλήσεις που βασίζονται σε περιορισμούς και απαιτήσεις.

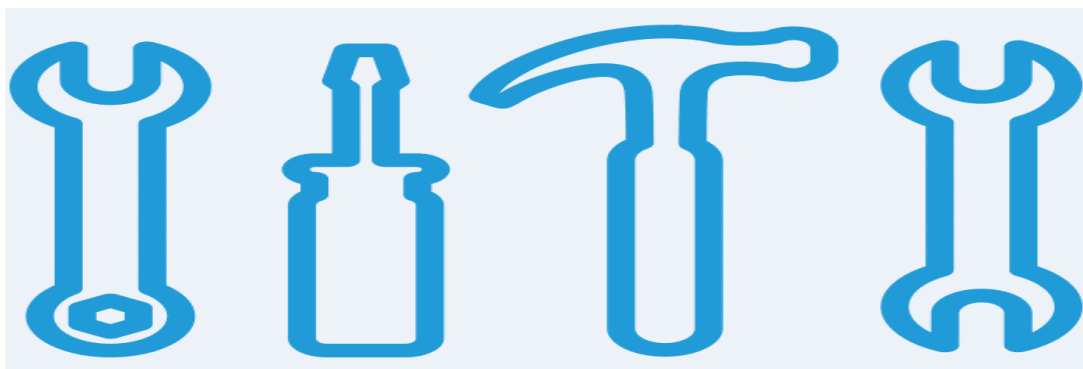
4.1.6 Διαλειτουργικότητα δεδομένων



Εικόνα 25 Διαλειτουργικότητα των δεδομένων [Πηγή: anylogic/ features <http://www.anylogic.com/features/>]

- Δυνατότητα εισαγωγής δεδομένων από οποιαδήποτε βάση δεδομένων, συμπεριλαμβάνοντας Oracle, MS SQL, MySQL, PostgreSQL, MS Access, Excel, and text files.
- Παροχή δυνατότητα (fast-build) προκειμένου να διαβάζονται εύκολα οι βάσεις δεδομένων και να γίνεται παραμετροποίηση των μοντέλων, εξαγωγή στατιστικών, στοιχείων της προσομοίωσης καθώς και αρχείων με τις ενδιάμεσες τιμές των μεταβλητών κατά την εκτέλεση του προγράμματος.

4.1.7 Επεκτάσιμη και προσαρμόσιμη πλατφόρμα



Εικόνα 26 Επεκτάσιμη και προσαρμόσιμη πλατφόρμα [Πηγή: anylogic/ features <http://www.anylogic.com/features/>]

- Τα μοντέλα του Anylogic είναι επεκτάσιμα σε επίπεδο Java, παρέχοντας στον χρήστη απεριόριστες δυνατότητες μοντελοποίησης.
- Παρέχεται η δυνατότητα σχεδιασμού προσαρμοσμένων πειραμάτων για ειδικές ανάγκες ή να γίνει χρήση προσωπικών αλγόριθμων και μηχανισμών βελτιστοποίησης. Επίσης, παρέχεται η δυνατότητα δημιουργίας προσαρμοσμένων ροών εργασίας πειραματισμού,

για παράδειγμα, συνδυασμός της ανάλυσης ευαισθησίας και της βελτιστοποίησης, με τα αποτελέσματα του πρώτου πειράματος να χρησιμεύουν ως είσοδος για το δεύτερο.

- Παρέχεται η δυνατότητα δημιουργίας προσαρμοσμένων αντικείμενων και επαναχρησιμοποιήσιμων βιβλιοθηκών αντικείμενων για τις προσωπικές περιοχές εφαρμογής.
- Ένα μοντέλο μπορεί να ρυθμιστεί από μια εξωτερική πηγή δεδομένων κατά την εκτέλεση του, κάτι που σημαίνει ότι μπορεί να αλλάξει ολόκληρη τη δομή του μοντέλου απλά αλλάζοντας τα δεδομένα εισόδου. Αυτό καθιστά τα μοντέλα επαναχρησιμοποιήσιμα, βοηθώντας στην επίλυση πολλών παρόμοιων προβλημάτων χωρίς να χρειάζεται να ξανά στηθεί το μοντέλο από την αρχή.

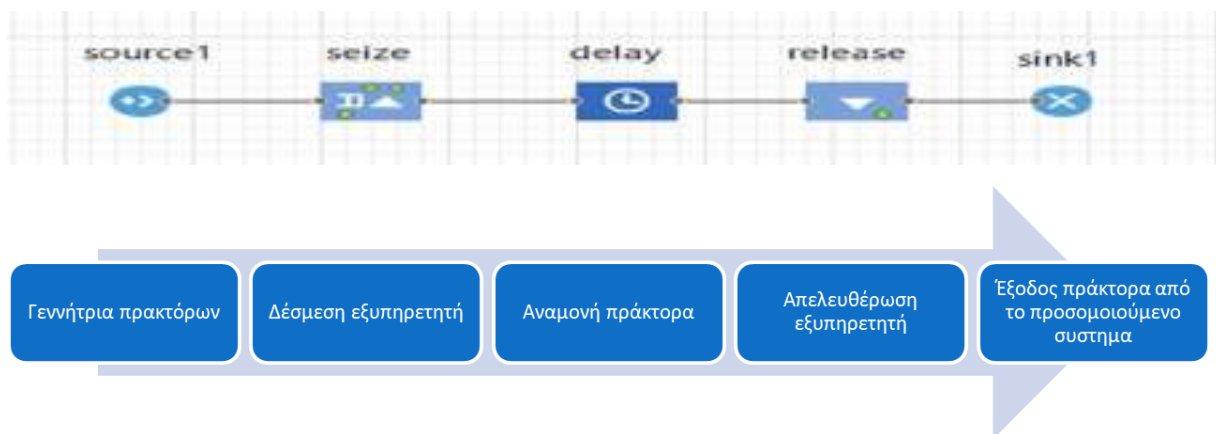
4.1.8 Βασικά βήματα στην διαμόρφωση ενός μοντέλου Anylogic

Τα Anylogic είναι ένα πρόγραμμα προσομοίωσης για την δημιουργία μοντέλων με την χρήση διαφόρων μεθόδων και 3D κινούμενων γραφικών. Στην παρούσα διπλωματική εργασία εξετάζεται η χρήση του Anylogic σε εφαρμογές μεταφοράς και διαχείρισης προϊόντων (transportation applications).

Η λογική με την οποία το Anylogic παράγει μοντέλα είναι η σύνδεση μεταξύ ενός διαγράμματος ροής (flow chart), το οποίο κατά βάση περιέχει τις διάφορες ενέργειες και εντολές που δίνονται στο πράκτορα (agent), και τα κινούμενα γραφικά (animation), στο οποίο γενικά καθορίζονται οι θέσεις και οι αποστάσεις για την κίνηση του πράκτορα.

Προκειμένου να γίνει πιο κατανοητό αναφέρεται το παρακάτω απλό παράδειγμα: ένας agent εισέρχεται στο σύστημα στην θέση Α, κατευθύνεται στην θέση Β, όπου και περιμένει για ένα χρονικό διάστημα και τέλος κατευθύνεται στην θέση Γ, σημείο στο οποίο βγαίνει από το σύστημα. Ο τρόπος εισαγωγής του μοντέλου στο Anylogic είναι (βλ. Εικόνα 27) :

1. Τοποθέτηση του σημείου Α (*animation*)
2. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> εισαγωγής πρακτόρων στο σημείο Α (*flow chart*)
3. Τοποθέτηση του σημείου Β (*animation*)
4. Τοποθέτηση της διαδρομής που ενώνει τα σημεία Α και Β (*animation*)
5. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας μετακίνησης των agent στο σημείο Β (*flow chart*)
6. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας αναμονής των agent στο σημείο Β (*flow chart*)
7. Τοποθέτηση του σημείου Γ (*animation*)
8. Τοποθέτηση του path που ενώνει τα σημεία Β και Γ (*animation*)
9. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας μετακίνησης των agent στο σημείο Γ (*flow chart*)
10. της κατάλληλης <<προγραμματιστικής μονάδας εξαγωγής agent από το σύστημα (*flow chart*)



Εικόνα 27 διάγραμμα ροής απλού μοντέλου

Η έννοια του πράκτορα στο Anylogic

Ένας πράκτορας στο Anylogic μπορεί να αντιπροσωπεύει ποικίλα πράγματα όπως οχήματα, εξοπλισμό, μηχανήματα, προϊόντα, μια ιδέα, έναν οργανισμό, ανθρώπου με ποικίλους ρόλους κ.α.



Εικόνα 28 Η έννοια του πράκτορα [Πηγή: anylogic/ features <http://www.anylogic.com/features/>

Οι πράκτορες είναι ένα από τα βασικά δομικά στοιχεία του AnyLogic. Ο πράκτορας είναι μια μονάδα σχεδιασμού όπου μπορεί να έχει δικιά του συμπεριφορά, μνήμη, επαφές, συγχρονισμό κ.α.

Σε έναν πράκτορα μπορούν να εισαχθούν διάφορες έννοιες που θα εξηγηθούν αναλυτικότερα στην συνέχεια όπως μεταβλητές, μία αλληλουχία ενεργειών ως διαγράμματα ροής κ.α.

Ο σχεδιασμός ενός πράκτορα ξεκινάει από τον καθορισμό των παραμέτρων που μπορεί να έχει, της συμπεριφοράς του και την πιθανή διεπαφή του με τους υπόλοιπους πράκτορες του μοντέλου.

4.1.9 Πλεονεκτήματα και μειονεκτήματα του Anylogic

Τα πλεονεκτήματα και τα μειονεκτήματα του προγράμματος Anylogic σε σύγκριση με άλλα λογισμικά προσομοίωσης σύμφωνα με το Critical Manufacturing S.A. (2014) είναι:

+	-
Υποστηρίζει τρεις τύπους μοντελοποίησης (Agent Based, Discrete Event and System Dynamics)	Δεν είναι ευέλικτο πρόγραμμα
Υποστηρίζει όλους του τύπους προσομοίωσης (Discrete/Continuous, Micro/Macro Level, ...)	Είναι γραμμένο σε Java, προϋποθέτει γνώση προγραμματισμού
Αποτελεί το πρόγραμμα προσομοίωσης με τους πιο γνωστούς πελάτες (Toyota, Panasonic, Rolls-Royce, IBM, Louvre, Intel, Delloite, UEFA, General Motors, Johnson & Johnson, US Navy, Volvo, HP, Red Cross, NASA, ...)	Δύσκολο σε αρχάριο επίπεδο
Υποστηρίζει επεκτάσεις των μοντέλων με την χρήση κώδικα Java	

Εικόνα 29 4.1.10 Πλεονεκτήματα και μειονεκτήματα του Anylogic
[Πηγή: Critical Manufacturing S.A. (2014)]

Τα πλεονεκτήματα και τα μειονεκτήματα του προγράμματος Anylogic όπως παρουσιάστηκαν στην παρούσα διπλωματική εργασία:

Το λογισμικό Anylogic προσφέρει σημαντικές δυνατότητες προσομοίωσης ενός συστήματος καθώς μπορεί να προσομοιώσει από απλά συστήματα όπως π.χ. M/M/1, M/M/2, E2/M/1 (Σε σύγκριση με αντίστοιχα μοντέλα από θεωρία αναμονής εντοπίστηκαν μικρές αποκλίσεις (<1%)) μέχρι και πιο σύνθετα όπως ένα σύστημα τερματικού σταθμού εμπορευματοκιβωτίων που περιλαμβάνει πολλά υποσυστήματα (π.χ. εκφόρτωση εμπορευματοκιβωτίων από τραίνα, παραλαβή εμπορευματοκιβωτίων από φορτηγά, φορτοεκφόρτωση πλοίων και στοιβασία σε χώρο εναπόθεσης) που συνδέονται μεταξύ τους. Ως μειονεκτήματα του προγράμματος αναφέρονται: α) Η δυσκολία στην εκμάθηση για έναν νέο χρήστη β) προϋποθέτει καλές γνώσεις προγραμματισμού για την προσομοίωση πολύπλοκων κανόνων λειτουργίας και την κατασκευή αυτοματοποιημένων μοντέλων (π.χ. μοντέλο Monte Carlo).

4.2 Ανάπτυξη και έλεγχος εγκυρότητας μοντέλων λιμενικών εγκαταστάσεων

Αναπτύχθηκαν τα μοντέλα λιμενικών εγκαταστάσεων M/M/1, M/M/2, E2/M/1, M/E2/2, E2/E2/1, E2/E2/2, M/M/1:GD/N/∞, M/M/2:GD/N/∞, με μοντέλα προσομοίωσης AnyLogic και θεωρία αναμονής και πραγματοποιήθηκαν έλεγχοι εγκυρότητας τους. Η τεχνική της θεωρίας αναμονής παρουσιάζεται αναλυτικά στο Παράρτημα Ε.

4.2.1 Το μοντέλο M/M/1

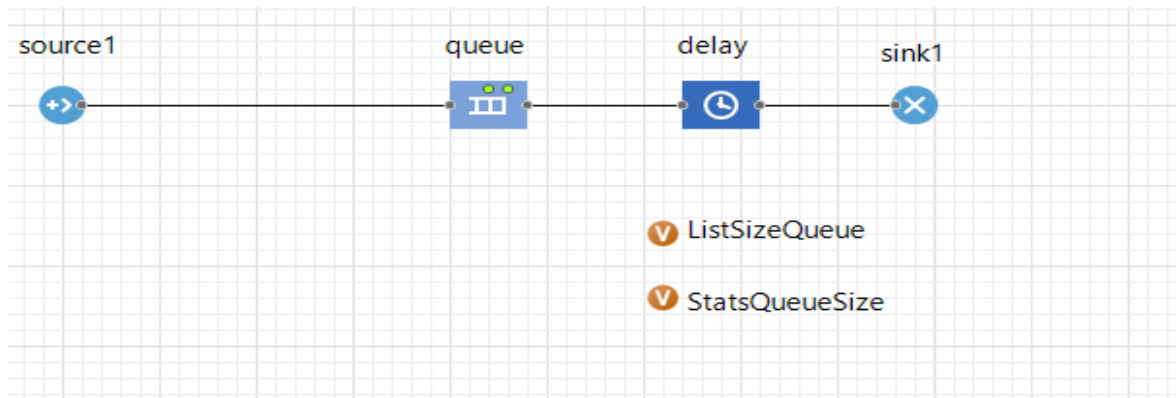
Στο μοντέλο M/M/1 οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (M). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (M). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

Βαθμός χρησιμοποίησης της θέσης παραβολής:
$$U = \frac{\lambda}{\mu}$$

Μέσος αριθμός πλοίων στην ουρά:
$$Lq = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

Μέσος χρόνος αναμονής στην ουρά:
$$Wq = \frac{\lambda}{\mu(\mu - \lambda)} = \frac{Lq}{\lambda}$$

Η διαμόρφωση του μοντέλου M/M/1 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 30



Εικόνα 30 Η διαμόρφωση του μοντέλου M/M/1 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 31 πεδίο «arrivals defined by».και πεδίο «Delay time». Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράρτημα Α.

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time

Interarrival time:

exponential(lambda)

⌚

delay - Delay

Name:

delay

☒ Show name ☐ Ignore

Type:

☒ Specified time

☐ Until stopDelay() is called

Delay time:

exponential(lambda)

minutes

Capacity:

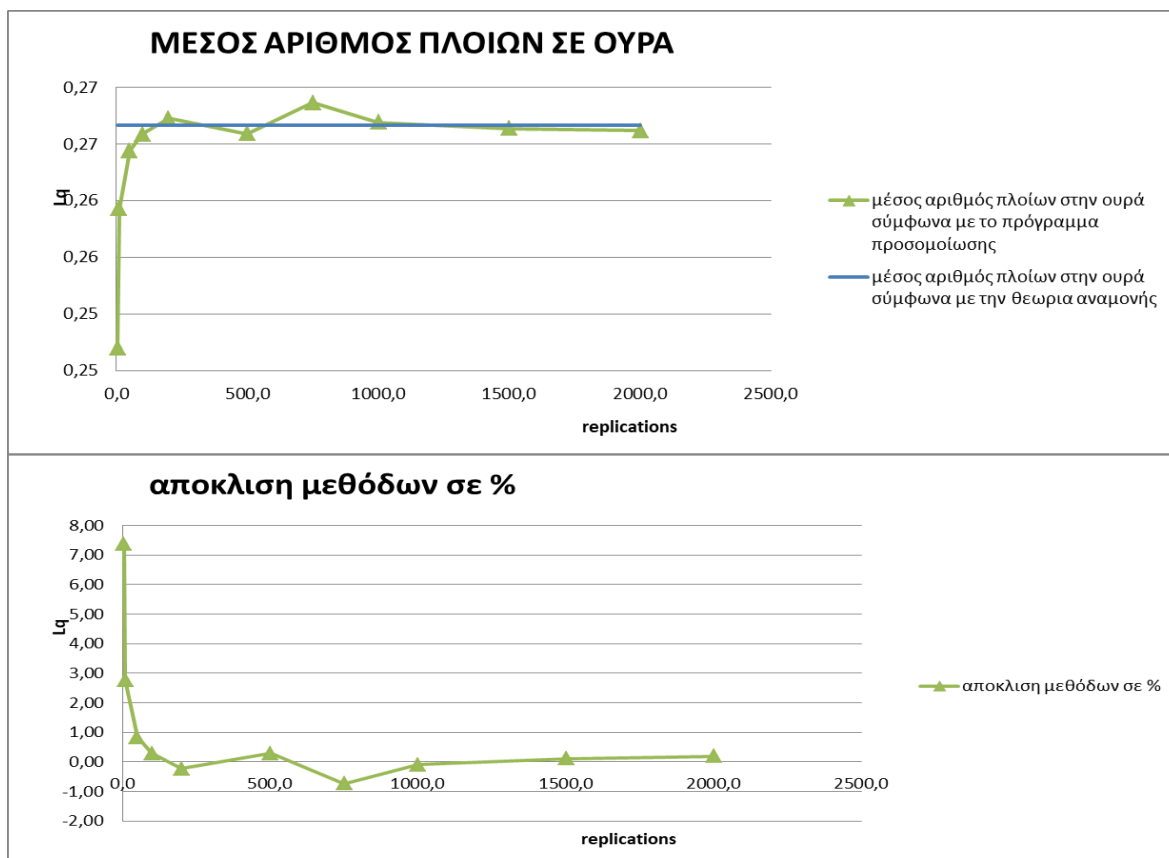
1

Εικόνα 31 Χρόνοι αναμονής M/M/1

Ο Πίνακας 1 παρουσιάζει τον έλεγχο για τον ελάχιστο αριθμό επαναλήψεων που πρέπει να γίνουν προκειμένου να εξαχθούν αποτελέσματα με αποδεκτό σφάλμα (μικρότερο του 1%).

M/M/1 λ=6 μ=15			
Αριθμός επαναλήψεων	Μέσος αριθμός πλοίων στην ουρά σύμφωνα με το πρόγραμμα προσομοίωσης	Μέσος αριθμός πλοίων στην ουρά σύμφωνα με την θεωρία αναμονής	Αποκλιση μεθόδων σε %
5	0,247	0,267	7,368
10	0,259	0,267	2,766
50	0,264	0,267	0,839
100	0,266	0,267	0,291
200	0,267	0,267	-0,224
500	0,266	0,267	0,285
750	0,269	0,267	-0,737
1000	0,267	0,267	-0,098
1500	0,266	0,267	0,114
2000	0,266	0,267	0,188

Πίνακας 1 Έλεγχος απαιτούμενου αριθμού επαναλήψεων



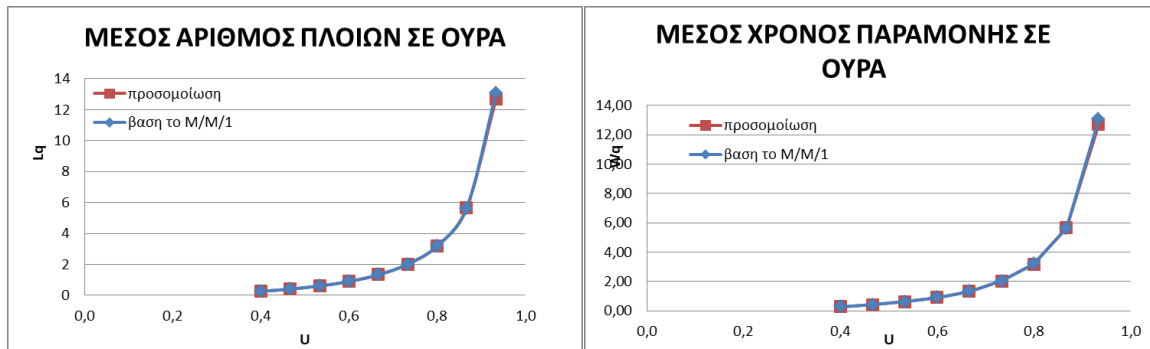
Εικόνα 32 Απόκλιση μεθόδων

Οι 1000 επαναλήψεις μας δίνουν ένα αποτέλεσμα πολύ κοντά στο απόλυτο, οπότε ο έλεγχος εγκυρότητας θα γίνει για αυτόν τον αριθμό επαναλήψεων.

M/M/1								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βάση το μοντέλο M/M/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βάση το μοντέλο M/M/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
6	15	0,400	0,267	0,267	-0,098	0,044	0,044	-0,098
7	15	0,467	0,408	0,408	0,124	0,058	0,058	0,124
8	15	0,533	0,610	0,612	-0,423	0,076	0,077	-0,423
9	15	0,600	0,900	0,895	0,573	0,100	0,099	0,573
10	15	0,667	1,333	1,335	-0,141	0,133	0,134	-0,141
11	15	0,733	2,017	2,020	-0,142	0,183	0,184	-0,142
12	15	0,800	3,200	3,189	0,337	0,267	0,266	0,337
13	15	0,867	5,633	5,652	-0,324	0,433	0,435	-0,324
14	15	0,933	13,067	12,695	2,842	0,933	0,907	2,842

Πίνακας 2 Αποκλίσεις M/M/1

Ο Πίνακας 2 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U), το μέσο αριθμό πλοίων που περιμένουν στην ουρά (Lq) και το μέσο χρόνο αναμονής τους (Wq).



Εικόνα 33 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/M/1

Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 9 για $\lambda=14$ του **Error! Reference source not found.**, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένο γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

4.2.2 Το μοντέλο M/M/2

Στο μοντέλο M/M/2 οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (M). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (M). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

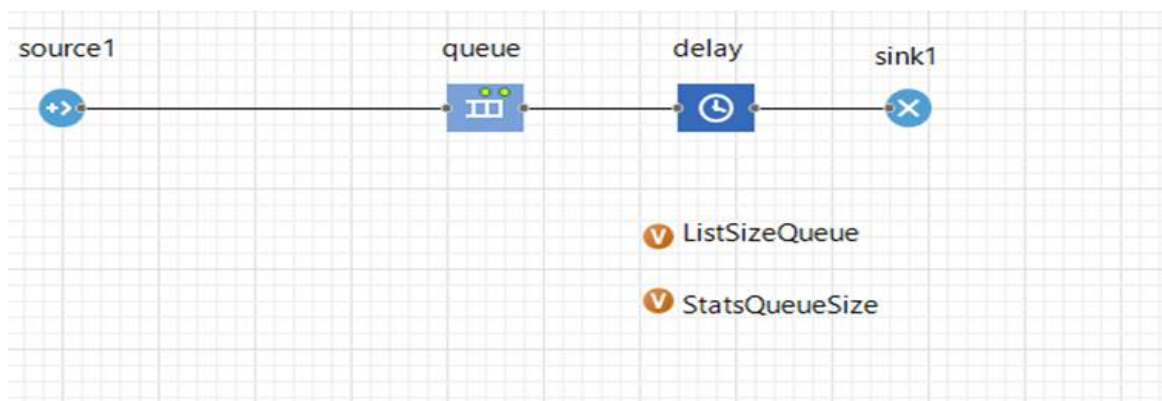
Βαθμός χρησιμοποίησης των θέσεων παραβολής:
$$U = \frac{\lambda}{s * \mu}$$

Μέσος αριθμός πλοίων στην ουρά:
$$Lq = L - \frac{\lambda}{\mu}$$

όπου,
$$L = \frac{\lambda \mu \left(\frac{\lambda}{\mu} \right)^s}{(s-1)!(s\mu - \lambda)^2} P_0 + \frac{\lambda}{\mu}$$
 και
$$P_0 = \frac{1}{\left[\sum \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n \right] + \frac{1}{s!} \left(\frac{\lambda}{\mu} \right)^s \left(\frac{s\mu}{s\mu - \lambda} \right)}$$

Μέσος χρόνος αναμονής στην ουρά:
$$Wq = \frac{Lq}{\lambda}$$

Η διαμόρφωση του μοντέλου M/M/2 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 34



Εικόνα 34 Η διαμόρφωση του μοντέλου M/M/2 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 35 πεδίο «arrivals defined by» και πεδίο «Delay time». Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράρτημα Α.

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

⇒

 Interarrival time

▼

Interarrival time:

↺

 exponential(lambda)

🕒 delay - Delay

Name:

delay

☒ Show name ☐ Ignore

Type:

⇒

☒ Specified time ☐ Until stopDelay() is called

Delay time:

↺

 exponential(lambda)

minutes

▼

Capacity:

⇒

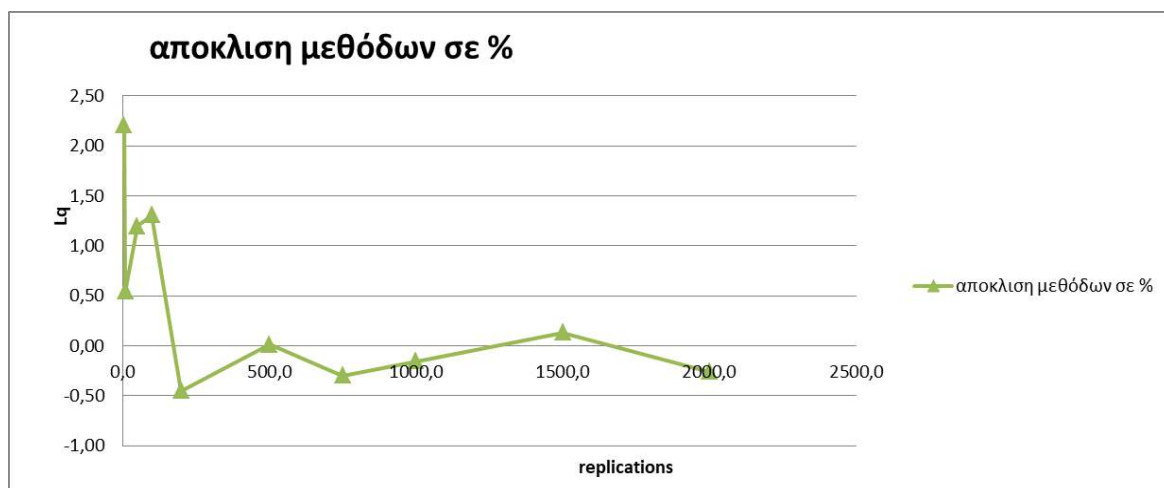
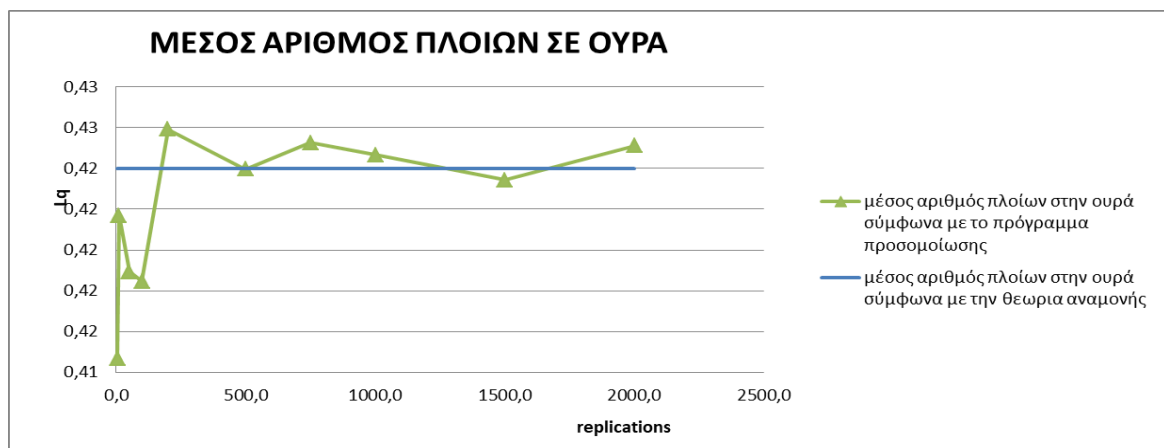
 2

Εικόνα 35 Χρόνοι αναμονής M/M/2

Ο Πίνακας 3 παρουσιάζει τον έλεγχο για τον ελάχιστο αριθμό επαναλήψεων που πρέπει να γίνουν προκειμένου να εξαχθούν αποτελέσματα με αποδεκτό σφάλμα (μικρότερο του 1%).

M/M/2 λ=16 μ=15			
Αριθμός επαναλήψεων	Μέσος αριθμός πλοίων στην ουρά σύμφωνα με το πρόγραμμα προσομοίωσης	Μέσος αριθμός πλοίων στην ουρά σύμφωνα με την θεωρία αναμονής	αποκλιση μεθόδων σε %
5	0,415	0,424	2,202
10	0,422	0,424	0,542
50	0,419	0,424	1,196
100	0,418	0,424	1,303
200	0,426	0,424	-0,452
500	0,424	0,424	0,010
750	0,425	0,424	-0,298
1000	0,425	0,424	-0,155
1500	0,423	0,424	0,133
2000	0,425	0,424	-0,263

Πίνακας 3 Έλεγχος αριθμού επαναλήψεων



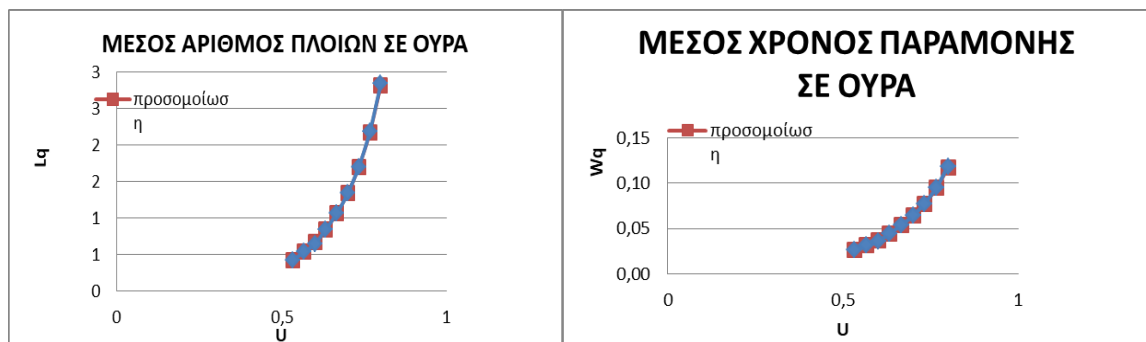
Εικόνα 36 Διαγράμματα επαναλήψεων

Οι 1000 επαναλήψεις μας δίνουν ένα αποτέλεσμα πολύ κοντά στο απόλυτο, οπότε ο έλεγχος εγκυρότητας θα γίνει για αυτόν τον αριθμό επαναλήψεων

M/M/2								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βάση το μοντέλο M/M/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βάση το μοντέλο M/M/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
16	15	0,533	0,424	0,425	-0,155	0,027	0,027	-0,155
17	15	0,567	0,536	0,537	-0,218	0,032	0,032	-0,218
18	15	0,600	0,645	0,673	-4,392	0,036	0,037	-4,392
19	15	0,633	0,848	0,849	-0,129	0,045	0,045	-0,129
20	15	0,667	1,067	1,067	-0,055	0,053	0,053	-0,055
21	15	0,700	1,345	1,344	0,082	0,064	0,064	0,082
22	15	0,733	1,706	1,703	0,227	0,078	0,077	0,227
23	15	0,767	2,186	2,176	0,461	0,095	0,095	0,461
24	15	0,800	2,844	2,825	0,685	0,119	0,118	0,685

Πίνακας 4 M/M/2 Αναλυτικά αποτελέσματα

Ο Πίνακας 4 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U), το μέσο αριθμό πλοίων που περιμένουν στην ουρά (Lq) και το μέσο χρόνο αναμονής τους (Wq).



Εικόνα 37 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/M/2

Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 3 για λ=18 του Πίνακας 4, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένο γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

4.2.3 Το μοντέλο E2/M/1

Στο μοντέλο E2/M/1 οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (M). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (M). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

$$\text{Βαθμός χρησιμοποίησης των θέσεων παραβολής: } U = \rho = \frac{\lambda}{\mu}$$

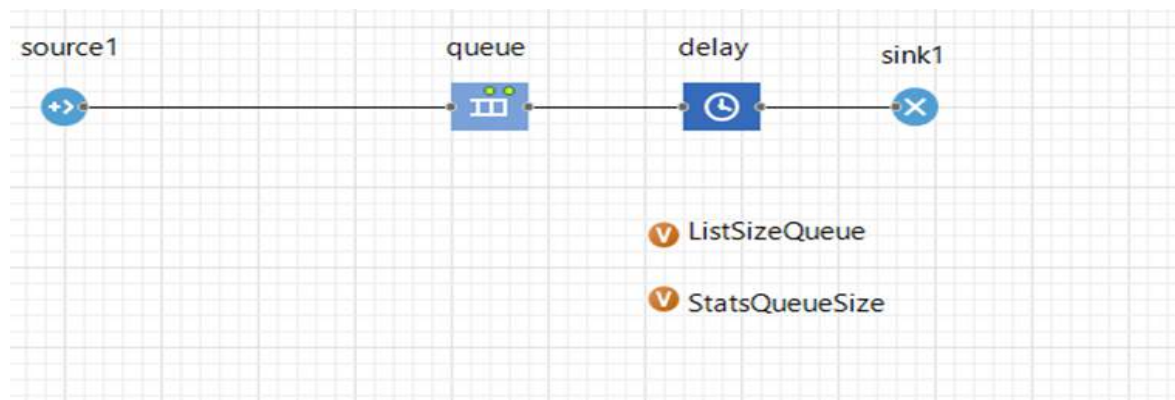
$$\text{Μέσος αριθμός πλοίων στην ουρά: } Lq = \sum_{n=1}^{\infty} n P_n - \rho$$

$$P_n = \begin{cases} 1 - \rho, & n=0 \\ \frac{z_0^k - 1}{z_0^{kn}} \rho & n \geq 1 \end{cases}$$

όπου,

Μέσος χρόνος αναμονής στην ουρά: $Wq = \frac{Lq}{\lambda}$

Η διαμόρφωση του μοντέλου M/M/2 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 38



Εικόνα 38 Η διαμόρφωση του μοντέλου E2/M/1 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 39 πεδίο «arrivals defined by». και πεδίο «Delay time». Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράρτημα Α

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time

Interarrival time:

erlang(beta, m)

⌚

delay - Delay

Name:

delay

☒ Show name ☐ Ignore

Type:

☒ Specified time

☐ Until stopDelay() is called

Delay time:

exponential(lambda)

minutes

Capacity:

1

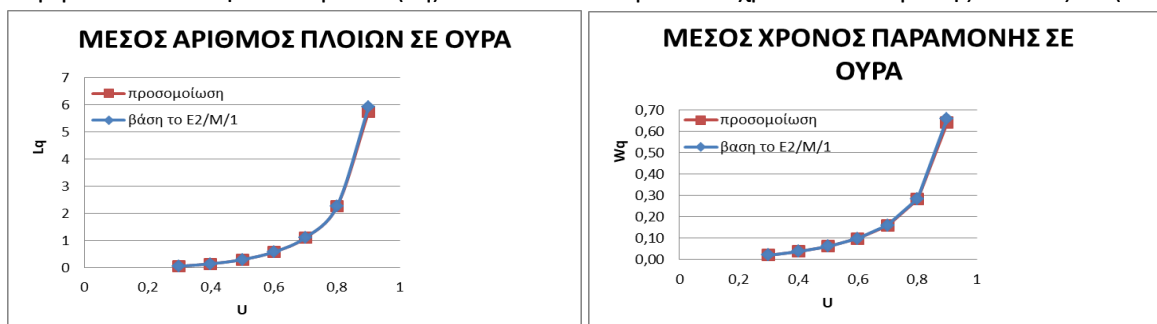
Εικόνα 39 Χρόνοι αναμονής E2/M/1

Οι 1000 επαναλήψεις μας δίνουν ένα αποτέλεσμα πολύ κοντά στο απόλυτο, οπότε ο έλεγχος εγκυρότητας θα γίνει για αυτόν τον αριθμό επαναλήψεων

Ε2/Μ/1								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βάση το μοντέλο Ε2/Μ/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βάση το μοντέλο Ε2/Μ/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
3	10	0,300	0,065	0,065	0,295	0,022	0,022	0,295
4	10	0,400	0,152	0,153	-0,357	0,038	0,038	-0,357
5	10	0,500	0,309	0,309	0,006	0,062	0,062	0,006
6	10	0,600	0,590	0,588	0,330	0,098	0,098	0,330
7	10	0,700	1,120	1,113	0,676	0,160	0,159	0,676
8	10	0,800	2,275	2,263	0,530	0,284	0,283	0,530
9	10	0,900	5,930	5,759	2,876	0,659	0,640	2,876

Πίνακας 5 Αναλυτικά αποτελέσματα Ε2/Μ/1

Ο Πίνακας 5 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης της θέσης παραβολής (U), το μέσο αριθμό πλοίων που περιμένουν στην ουρά (Lq) και το μέσο χρόνο αναμονής τους (Wq).



Εικόνα 40 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα Ε2/Μ/1

Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 7 για λ=9 του Πίνακας 5, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένη γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

4.2.4 Το μοντέλο Μ/Ε2/2

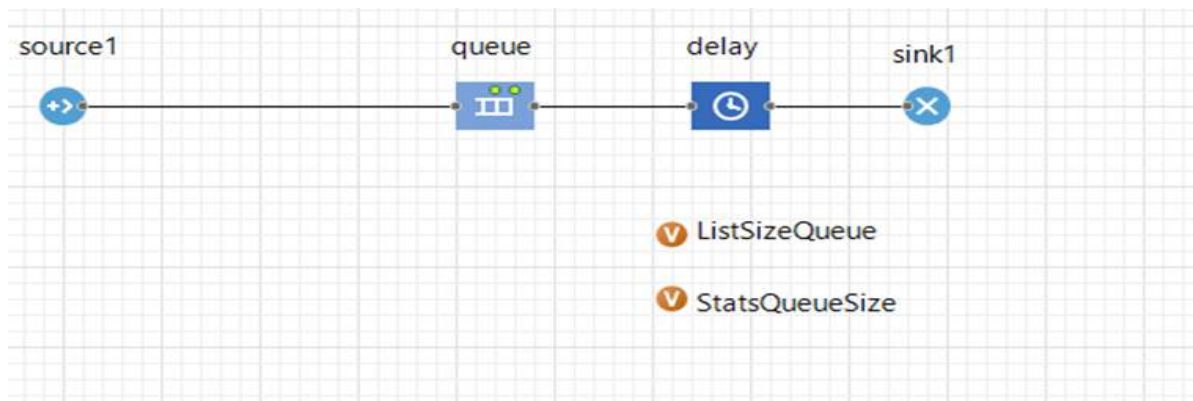
Στο μοντέλο Μ/Ε2/1 οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (Μ). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (Μ). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

$$\text{Βαθμός χρησιμοποίησης των θέσεων παραβολής: } U = \frac{\lambda}{s * \mu}$$

Μέσος αριθμός πλοίων στην ουρά: $Lq = \frac{1+k}{2k} \times + \frac{\lambda^2}{\mu(\mu-\lambda)}$

Μέσος χρόνος αναμονής στην ουρά: $Wq = \frac{Lq}{\lambda}$

Η διαμόρφωση του μοντέλου M/M/2 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 41



Εικόνα 41 Η διαμόρφωση του μοντέλου E2/M/2 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 42 πεδίο «arrivals defined by».και πεδίο «Delay time». Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράτημα Α

source1 - Source

Name: source1 ☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time:

delay - Delay

☐ Until stopDelay() is called

Delay time:

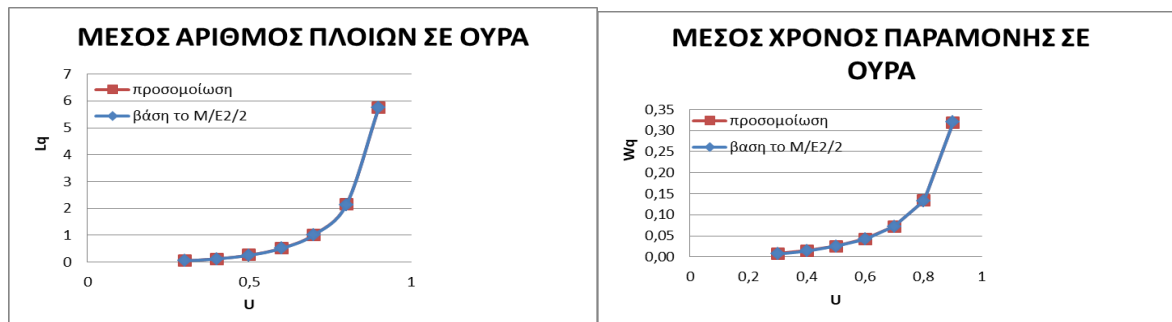
Capacity:

Εικόνα 42 Χρόνοι αναμονής M/E2/2

M/E2/2								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βάση το μοντέλο M/E2/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βάση το μοντέλο M/E2/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
6	10	0,3	0,045	0,047	-4,689	0,007	0,008	-4,689
8	10	0,4	0,114	0,118	-3,131	0,014	0,015	-3,131
10	10	0,5	0,250	0,256	-2,396	0,025	0,026	-2,396
12	10	0,6	0,506	0,513	-1,343	0,042	0,043	-1,343
14	10	0,7	1,009	1,018	-0,926	0,072	0,073	-0,926
16	10	0,8	2,133	2,140	-0,318	0,133	0,134	-0,318
18	10	0,9	5,755	5,742	0,223	0,320	0,319	0,223

Πίνακας 6 αναλυτικά αποτελέσματα M/E2/2

Ο Πίνακας 6 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U), το μέσο αριθμό πλοίων που περιμένουν στην ουρά (Lq) και το μέσο χρόνο αναμονής τους (Wq). Οι υπολογισμοί έγιναν με το πρόγραμμα Qts (αναλυτική επίλυση) και με μοντέλο προσομοίωσης στο πρόγραμμα Anylogic.



Εικόνα 43 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα M/E2/2

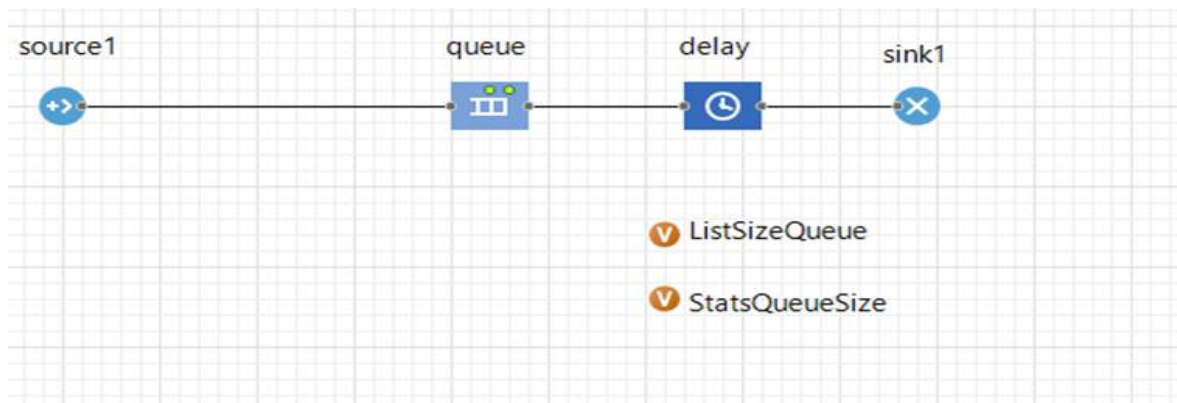
Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 7 για $\lambda=18$ του Πίνακας 6, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένο γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

4.2.5 Το μοντέλο E2/E2/1

Στο μοντέλο E2/E2/1 οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (M). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (M). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

Η διαμόρφωση του μοντέλου E2/E2/1 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 44



Εικόνα 44 Η διαμόρφωση του μοντέλου E2/E2/1 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 45 πεδίο «arrivals defined by».και πεδίο «Delay time». Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδοντα στο Παράτημα Α

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time

Interarrival time:

erlang(beta, m)

⌚

delay - Delay

☐ Until stopDelay() is called

Delay time:

erlang(beta, m)

minutes

Capacity:

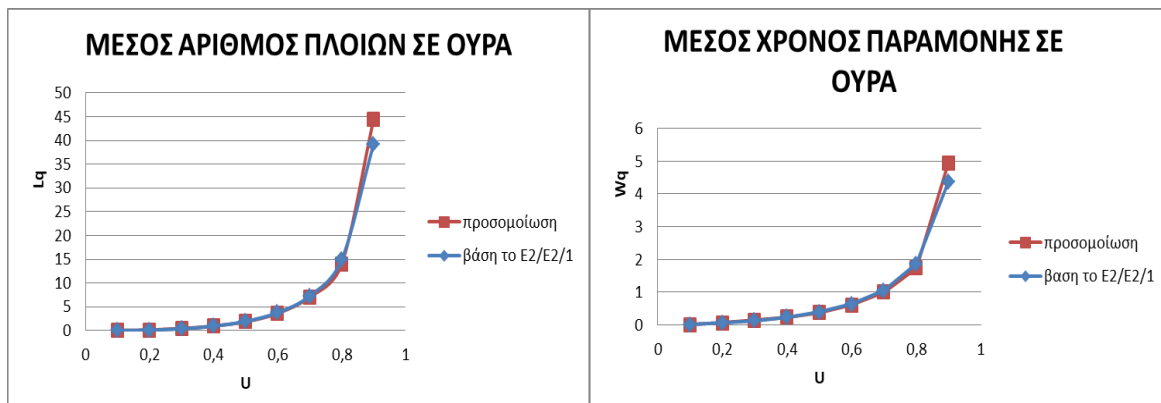
1

Εικόνα 45 Χρόνοι αναμονής E2/E2/1

E2/E2/1								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βάση το μοντέλο E2/E2/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βάση το μοντέλο E2/E2/1	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
1	10	0,1	0,166	0,183	-10,221	0,166	0,183	-10,221
2	10	0,2	1,208	1,221	-1,095	0,604	0,611	-1,095
3	10	0,3	3,930	3,673	6,541	1,310	1,224	6,541
4	10	0,4	9,420	9,400	0,213	2,355	2,350	0,213
5	10	0,5	19,520	19,520	0,002	3,904	3,904	0,002
6	10	0,6	18,360	18,393	-0,180	3,060	3,066	-0,180
7	10	0,7	72,737	72,452	0,392	10,391	10,350	0,392
8	10	0,8	149,224	150,004	-0,523	18,653	18,750	-0,523
9	10	0,9	392,310	389,495	0,718	43,590	43,277	0,718

Πίνακας 7 Αναλυτικά αποτελέσματα E2/E2/1

Ο Πίνακας 7 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U), το μέσο αριθμό πλοίων που περιμένουν στην ουρά (Lq) και το μέσο χρόνο αναμονής τους (Wq).



Εικόνα 46 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα E2/E2/1

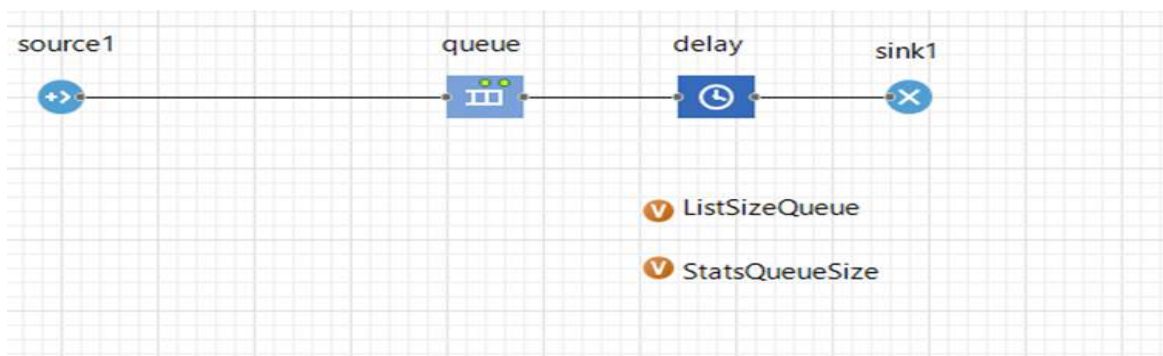
Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 3 και 9 για $\lambda=3$ και $\lambda=9$ αντίστοιχα του Πίνακας 7, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελούν μεμονωμένα γεγονότα.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

4.2.6 Το μοντέλο E2/E2/2

Στο μοντέλο E2/E2/2, οι χρόνοι μεταξύ αφίξεων πλοίων στον λιμένα ακολουθούν την εκθετική κατανομή (M). Οι χρόνοι εξυπηρέτησης τους ακολουθεί επίσης την εκθετική κατανομή (M). Ο λιμένας διαθέτει μία θέση παραβολής. Για το μοντέλο αυτό ισχύει (Ξηροκώστας, Δ. 1990, σελ 199):

Η διαμόρφωση του μοντέλου E2/E2/2 με το λογισμικό AnyLogic πραγματοποιείται με τρεις «προγραμματιστικές μονάδες» που συνδέονται εν σειρά σύμφωνα με την Εικόνα 47



Εικόνα 47 Η διαμόρφωση του μοντέλου E2/E2/2 με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 48 πεδίο «arrivals defined by» και πεδίο «Delay time»). . Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράρτημα Α

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time

Interarrival time:

erlang(beta, m)

🕒 delay - Delay

☐ Until stopDelay() is called

Delay time:

erlang(beta, m)

minutes

Capacity:

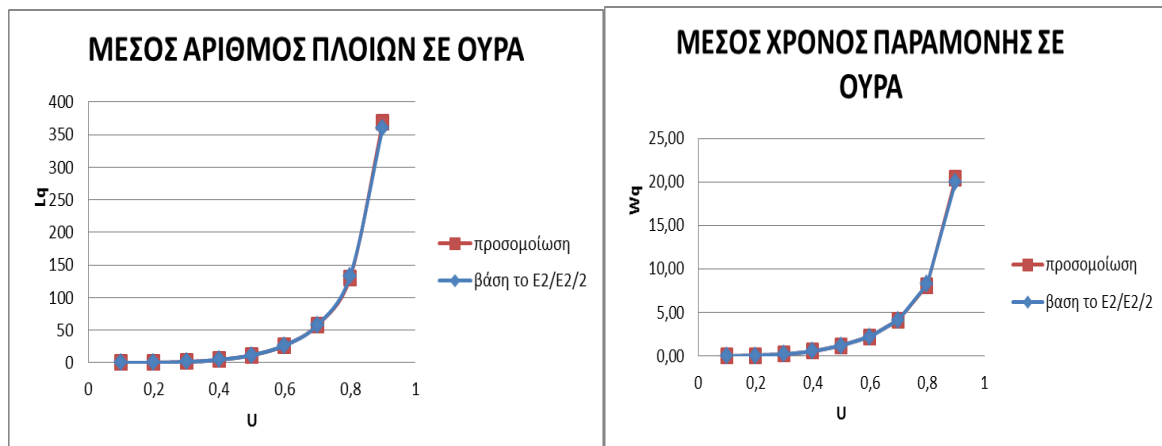
2

Εικόνα 48 Χρόνοι αναμονής E2/E2/2

E2/E2/2								
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	μέσος αριθμός πλοίων στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βαση το μοντέλο E2/E2/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βαση το μοντέλο E2/E2/2	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Lq	Lq		Wq	Wq	
2	10	0,1	0,012	0,013	-6,782	0,006	0,006	-6,782
4	10	0,2	0,260	0,261	-0,278	0,065	0,065	-0,278
6	10	0,3	1,410	1,412	-0,129	0,235	0,235	-0,129
8	10	0,4	4,608	4,619	-0,230	0,576	0,577	-0,230
10	10	0,5	11,810	11,809	0,006	1,181	1,181	0,006
12	10	0,6	26,664	26,614	0,189	2,222	2,218	0,189
14	10	0,7	57,750	57,879	-0,224	4,125	4,134	-0,224
16	10	0,8	132,800	129,759	2,290	8,300	8,110	2,290
18	10	0,9	360,000	368,822	-2,451	20,000	20,490	-2,451

Πίνακας 8 Αναλυτικά αποτελέσματα E2/E2/2

Ο Πίνακας 8 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U) και το μέσο χρόνο αναμονής των πλοίων (Wq)



Εικόνα 49 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα E2/E2/2

Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 7 για $\lambda=14$ του Πίνακα 8, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένο γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

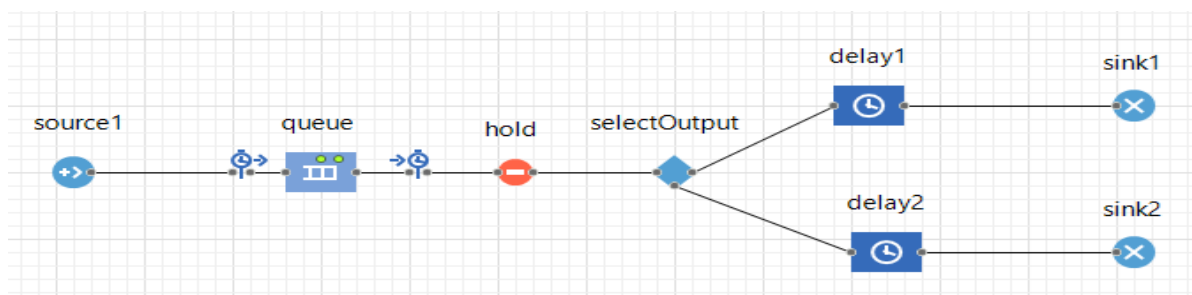
4.2.7 Το μοντέλο (M/M/2) : (GD/N/∞). (Περιορισμένος χώρος αναμονής)

Στο μοντέλο (M/M/2) : (GD/N/∞), πλοία φθάνουν σε λιμάνι και οι αφίξεις τους ακολουθούν την εκθετική κατανομή (M). Η εξυπηρέτηση τους ακολουθεί επίσης την εκθετική κατανομή (M) και στο λιμάνι υπάρχουν δύο θέσεις παραβολής. Ο μέγιστος χώρος αναμονής είναι (N).

Οι υπολογισμοί έγιναν με μοντέλο προσομοίωσης στο πρόγραμμα Anylogic.

Η σύγκριση έγινε με βάση τους πίνακες του βιβλίου Ξηροκώστας, Δ. (1990), σελ 199

Η διαμόρφωση του μοντέλου (M/M/2) : (GD/N/∞), με το λογισμικό AnyLogic πραγματοποιείται με «προγραμματιστικές μονάδες» που συνδέονται εν σειρά και εν παραλλήλω σύμφωνα με την Εικόνα 50



Εικόνα 50 Η διαμόρφωση του μοντέλου (M/M/2) : (GD/N/∞) με το λογισμικό AnyLogic

Στην συνέχεια προσδιορίζονται ο χρόνος μεταξύ διαδοχικών αφίξεων πλοίων και ο χρόνος εξυπηρέτησης τους (βλ. Εικόνα 51 πεδίο «arrivals defined by» και πεδίο «Delay time»).

Αναλυτικές οδηγίες για την εισαγωγή στοιχείων και όλες τις διαθέσιμες επιλογές δίδονται στο Παράρτημα Α

+

source1 - Source

Name:

source1

☒ Show name ☐ Ignore

Arrivals defined by:

Interarrival time

Interarrival time:

exponential (lambda)

⌚

delay - Delay

Name:

delay

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

exponential (lambda)

minutes

Capacity:

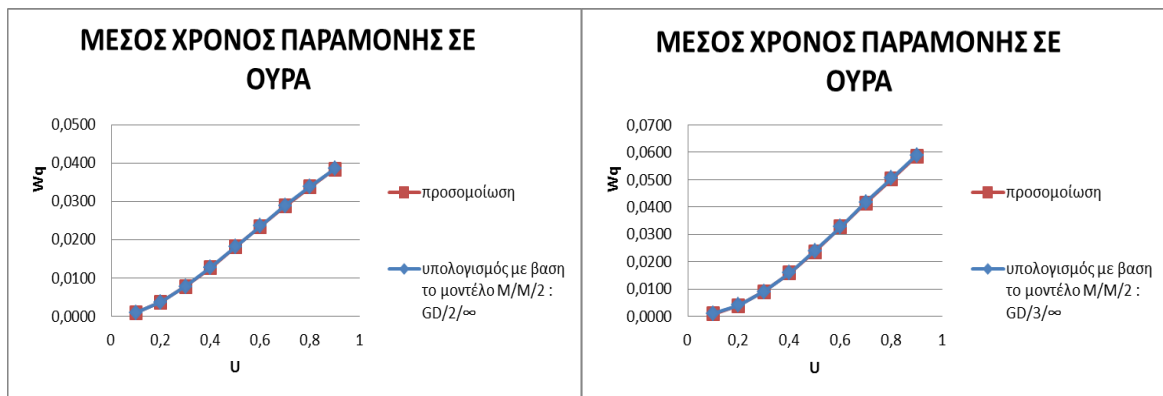
2

Εικόνα 51 Χρόνοι αναμονής (M/M/2) : (GD/N/∞)

ρυθμός άφιξης	ρυθμός εξυπηρέτησης	βαθμός χρησιμοποίησης	M/M/2 : GD/2/∞			M/M/2 : GD/3/∞		
			μέσος χρόνος παραμονής στην ουρά			μέσος χρόνος παραμονής στην ουρά		
			Υπολογισμός με βαση το μοντέλο M/M/2 : GD/2/∞	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %	Υπολογισμός με βαση το μοντέλο M/M/2 : GD/3/∞	Υπολογισμός με πρόγραμμα προσομοίωσης	Αποκλιση μεθόδων σε %
λ	μ	U	Wq	Wq		Wq	Wq	
2	10	0,1	0,001	0,001	2,411	0,001	0,001	5,941
4	10	0,2	0,004	0,004	1,526	0,004	0,004	1,424
6	10	0,3	0,008	0,008	0,317	0,009	0,009	0,624
8	10	0,4	0,013	0,013	0,934	0,016	0,016	0,240
10	10	0,5	0,018	0,018	0,396	0,024	0,024	0,862
12	10	0,6	0,024	0,023	0,700	0,033	0,033	0,159
14	10	0,7	0,029	0,029	0,241	0,042	0,042	0,467
16	10	0,8	0,034	0,034	0,200	0,051	0,050	0,559
18	10	0,9	0,039	0,039	0,183	0,059	0,059	0,398

Πίνακας 9 Αναλυτικά αποτελέσματα(M/M/2) : (GD/N/∞)

Ο Πίνακας 9 παρουσιάζει, για διάφορους ρυθμούς άφιξης (λ) και σταθερό ρυθμό εξυπηρέτησης (μ), το βαθμό χρησιμοποίησης των θέσεων παραβολής (U) και το μέσο χρόνο αναμονής των πλοίων (Wq).



Εικόνα 52 Μέσος αριθμός πλοίων & μέσος χρόνος αναμονής πλοίων συγκριτικά διαγράμματα ($M/M/2$) : ($GD/N/\infty$)

Παρατηρείται ότι τα δύο γραφήματα ταυτίζονται. Η μέγιστη διαφορά παρουσιάζεται στην γραμμή 1 για $\lambda=2$ και χώρο αναμονής ίσο με 3 του Πίνακα 9, ωστόσο δεν αποτελεί αιτία μη εγκυρότητας του προγράμματος καθώς αποτελεί μεμονωμένο γεγονός.

Από τα ανωτέρω προκύπτει το συμπέρασμα ότι το πρόγραμμα είναι απολύτως έγκυρο, με αποκλίσεις που είναι στα αποδεκτά όρια λόγω της τυχαιότητας.

Συμπεράσματα & Παρατηρήσεις:

Σε όλες τις περιπτώσεις έχοντας το μ ως σταθερή μεταβλητή, το λ ως μεταβλητή και τέλος το U ως εξαρτημένη, έγινε προσπάθεια αναπαράστασης των μοντέλων που περιγράφονται στην θεωρία αναμονής με επιτυχία, καθώς τα αποτελέσματα είναι πολύ κοντά με αποκλίσεις κατά μέσο όρο κάτω του 1%.

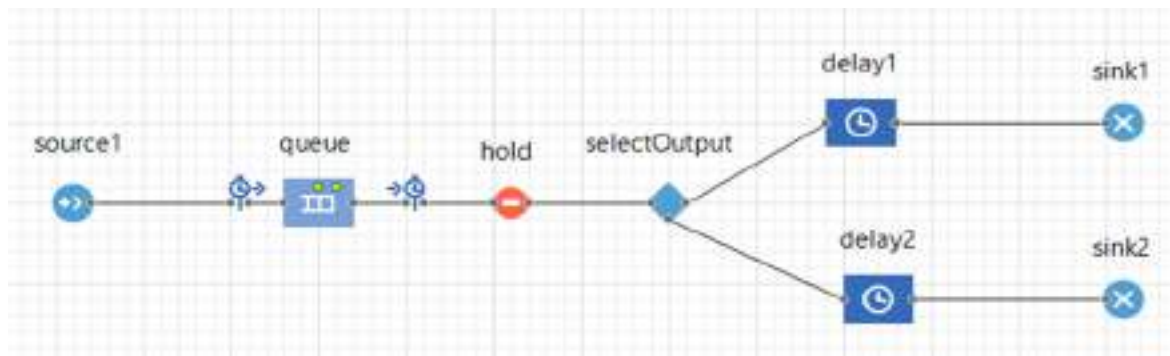
Κατά την διάρκεια των πειραμάτων αλλά και στο τέλος βλέποντας συνολικά τα αποτελέσματα των πειραμάτων αυτών, έγιναν ορισμένες παρατηρήσεις. Η πρώτη παρατήρηση σχετίζεται με τον απαιτούμενο αριθμό επαναλήψεων που απαιτούνται προκειμένου το αποτέλεσμα να είναι ασφαλές. Παρατηρήθηκε ότι καθώς αυξανόταν ο αριθμός τους, τα αποτελέσματα γίνονταν όλο και πιο έγκυρα. Έτσι προκειμένου να αποφασιστεί ο ακριβής αριθμός των επαναλήψεων διαμορφώθηκαν για 3 διαφορετικά σενάρια, πίνακες και γραφήματα προκειμένου να γίνει η επιλογή. Το κριτήριο με το οποίο έγινε η επιλογή ήταν να βρεθεί ο μικρότερος δυνατός αριθμός επαναλήψεων με αποδεκτά όρια στις αποκλίσεις των αποτελεσμάτων (<1%). Ο αριθμός αυτός επιλέχθηκε να είναι οι 1000 επαναλήψεις. Μια δεύτερη παρατήρηση είναι το γεγονός ότι κατά βάση, καθώς αυξανόταν η χρησιμοποίηση (U), αυξάνονταν και οι αποκλίσεις. Αυτό οφείλεται στο γεγονός ότι όταν πιέζεται το σύστημα, φτάνοντας σε οριακές συνθήκες, η τυχαιότητα έχει μεγαλύτερο βαρόμετρο, με αποτέλεσμα να υπάρχουν μεγαλύτερες αποκλίσεις. Τέλος, λόγω του ότι τα συστήματα $E2/E2/1$ και $E2/E2/2$, λόγω του ότι δεν είναι εφικτό να γίνει ακριβείς μαθηματική επίλυση, αλλά προσεγγιστική, οι αποκλίσεις που έχουν καταγραφεί δεν αποτελούν σύγκριση με το βέλτιστο αποτέλεσμα αλλά με την μαθηματικά προσεγγιστική λύση.

4.2.8 Ανάπτυξη μοντέλου M/M*/2 με διαφορετικό ρυθμό εξυπηρέτησης σε κάθε θέση παραβολής

Στο μοντέλο M/M*/2, πλοία φθάνουν σε λιμάνι και οι αφίξεις τους ακολουθούν την εκθετική κατανομή (M). Η εξυπηρέτηση τους ακολουθεί επίσης την εκθετική κατανομή (μ) και (μ^*) ενώ στο λιμάνι υπάρχουν δύο θέσεις παραβολής.

Το μοντέλο αυτό αποτελεί έναν διαφορετικό τρόπο προσέγγισης της θεωρίας αναμονής, έχοντας ως λογική την πιθανή διαφορετικότητα των εξυπηρετητών ως προς τον ρυθμό εξυπηρέτησης. Μαθηματικά αποδεδειγμένες λύσεις δεν υπάρχουν, αλλά λόγω τις εγκυρότητας των υπολοίπων προσεγγίσεων, θεωρείται επαρκή και απολύτως ακριβή και η συγκεκριμένη που ακολουθεί.

Λόγο του γεγονότος ότι διαμορφώθηκε μία νέα ακολουθία «προγραμματιστικές μονάδες» εν σειρά και εν παραλλήλω, έγινε ξανά έλεγχος εγκυρότητας, διαμορφώνοντας με τον διαφορετικό αυτό τρόπο που παρουσιάζεται στην Εικόνα 53, ένα μοντέλο M/M/2 όπου τα αποτελέσματα του είναι γνωστά. Στον έλεγχο αυτό υπήρξαν ταυτόσημα αποτελέσματα.



Εικόνα 53 Η διαμόρφωση του μοντέλου M/M*/2 με το λογισμικό AnyLogic

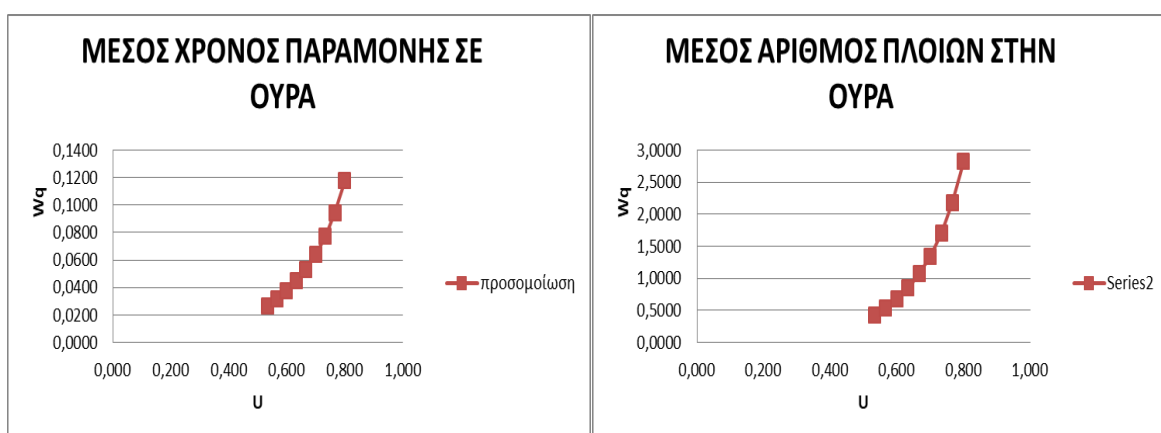
M/M*/2: μ/μ^*							
Ρυθμός αφίξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	προτεραιότητα στο γρηγορότερο(2)		προτεραιότητα στο πιο αργό (1)		Αποκλιση μεθόδων σε %
			Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης	
λ	μ	μ	Wq	Lq	Wq	Lq	
16	10	15	0,055	0,879	0,056	0,911	3,388
17	10	15	0,067	1,154	0,069	1,191	2,905
18	10	15	0,084	1,541	0,087	1,576	3,365
19	10	15	0,107	2,060	0,109	2,108	1,976
20	10	15	0,141	2,844	0,142	2,873	0,554
21	10	15	0,186	3,998	0,190	4,066	2,061
22	10	15	0,262	5,966	0,266	6,033	1,471
23	10	15	0,414	10,129	0,408	9,893	-1,509
24	10	15	0,770	21,167	0,780	21,153	1,289

Πίνακας 10 Προτεραιότητες M/M*/2 αναλυτικά αποτελέσματα

Οι πίνακες των αναλυτικών αποτελεσμάτων και τα αντίστοιχα διαγράμματα που δημιουργήθηκαν μετά από τα διάφορα μ/μ^* δίνοντας προτεραιότητα στον γρηγορότερο εξυπηρετητή είναι τα εξής:

M/M*/2: $\mu/\mu^*=1$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
16	15	15	0,533	0,027	0,425
17	15	15	0,567	0,032	0,537
18	15	15	0,600	0,037	0,673
19	15	15	0,633	0,045	0,849
20	15	15	0,667	0,053	1,067
21	15	15	0,700	0,064	1,344
22	15	15	0,733	0,077	1,703
23	15	15	0,767	0,095	2,176
24	15	15	0,800	0,118	2,825

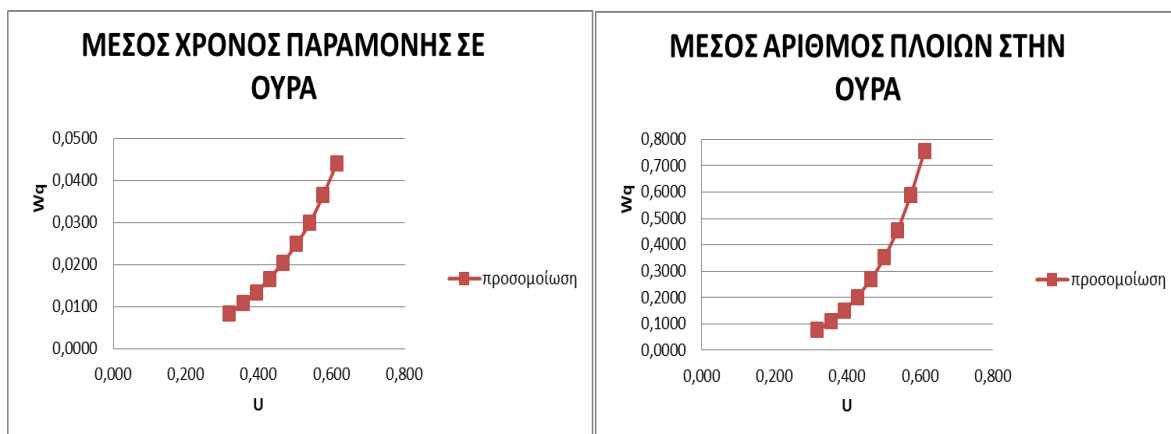
Πίνακας 11 Αναλυτικά αποτελέσματα M/M*/2 : $\mu/\mu^*=1$



Εικόνα 54 Διαγράμματα M/M*/2 : $\mu/\mu^*=1$

M/M*/2: $\mu/\mu^*=0,833$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
9	12,5	15	0,319	0,008	0,077
10	12,5	15	0,356	0,011	0,109
11	12,5	15	0,392	0,013	0,149
12	12,5	15	0,429	0,017	0,202
13	12,5	15	0,466	0,021	0,269
14	12,5	15	0,503	0,025	0,352
15	12,5	15	0,539	0,030	0,455
16	12,5	15	0,576	0,037	0,591
17	12,5	15	0,613	0,044	0,756

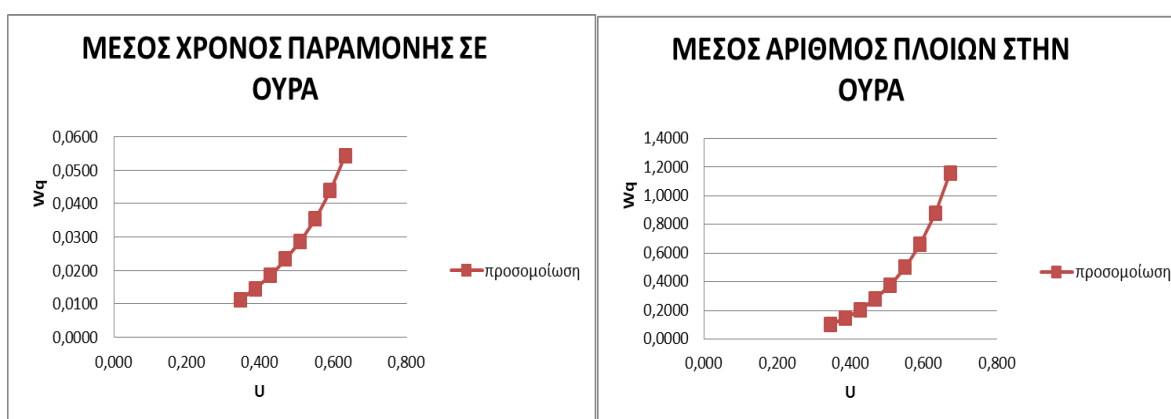
Πίνακας 12 Αναλυτικά αποτελέσματα M/M*/2 : $\mu/\mu^*=0,8333$



Εικόνα 55 Διαγράμματα $M/M/2$: $\mu/\mu^*=0,8333$

$M/M/2$: $\mu/\mu^*=0,667$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
9	10	15	0,346	0,011	0,103
10	10	15	0,386	0,015	0,147
11	10	15	0,428	0,018	0,205
12	10	15	0,469	0,023	0,282
13	10	15	0,510	0,029	0,377
14	10	15	0,550	0,035	0,502
15	10	15	0,591	0,044	0,663
16	10	15	0,633	0,054	0,878
17	10	15	0,674	0,067	1,160

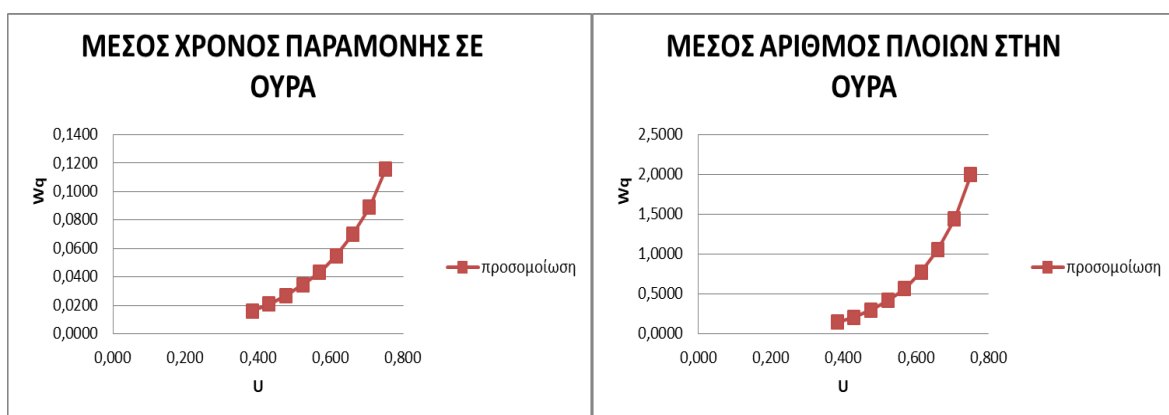
Πίνακας 13 Αναλυτικά αποτελέσματα $M/M/2$: $\mu/\mu^*=0,66667$



Εικόνα 56 Διαγράμματα $M/M/2$: $\mu/\mu^*=0,66667$

M/M*/2: $\mu/\mu^*=0,5$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
9	7,5	15	0,385	0,016	0,146
10	7,5	15	0,431	0,021	0,212
11	7,5	15	0,477	0,027	0,299
12	7,5	15	0,523	0,034	0,417
13	7,5	15	0,569	0,043	0,567
14	7,5	15	0,616	0,055	0,777
15	7,5	15	0,661	0,070	1,057
16	7,5	15	0,707	0,089	1,442
17	7,5	15	0,752	0,116	1,996

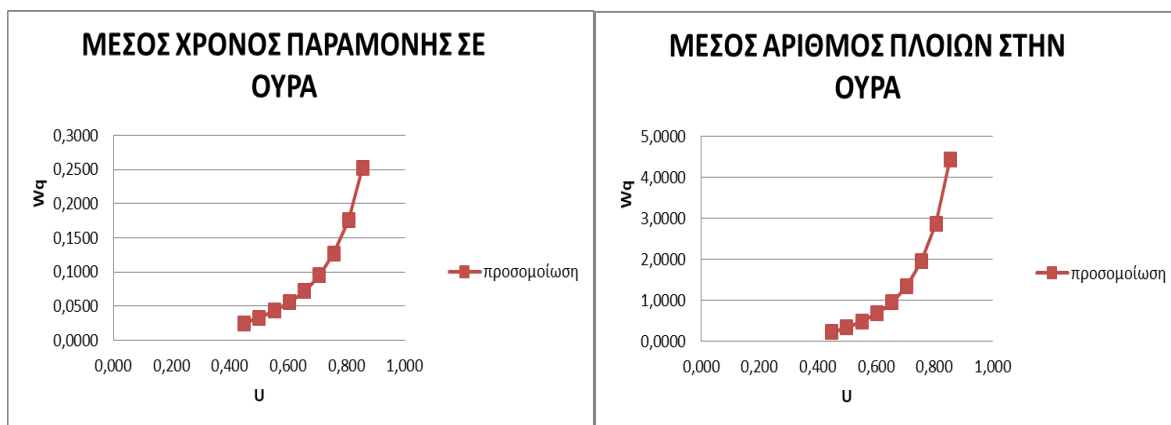
Πίνακας 14 Αναλυτικά αποτελέσματα M/ M*/2 : $\mu/\mu^*=0,5$



Εικόνα 57 Διαγράμματα M/M*/2 : $\mu/\mu^*=0,5$

M/M*/2: $\mu/\mu^*=0,333$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
9	5	15	0,448	0,025	0,228
10	5	15	0,500	0,033	0,334
11	5	15	0,552	0,043	0,480
12	5	15	0,604	0,056	0,676
13	5	15	0,656	0,072	0,955
14	5	15	0,706	0,095	1,350
15	5	15	0,756	0,128	1,948
16	5	15	0,806	0,175	2,867
17	5	15	0,855	0,253	4,431

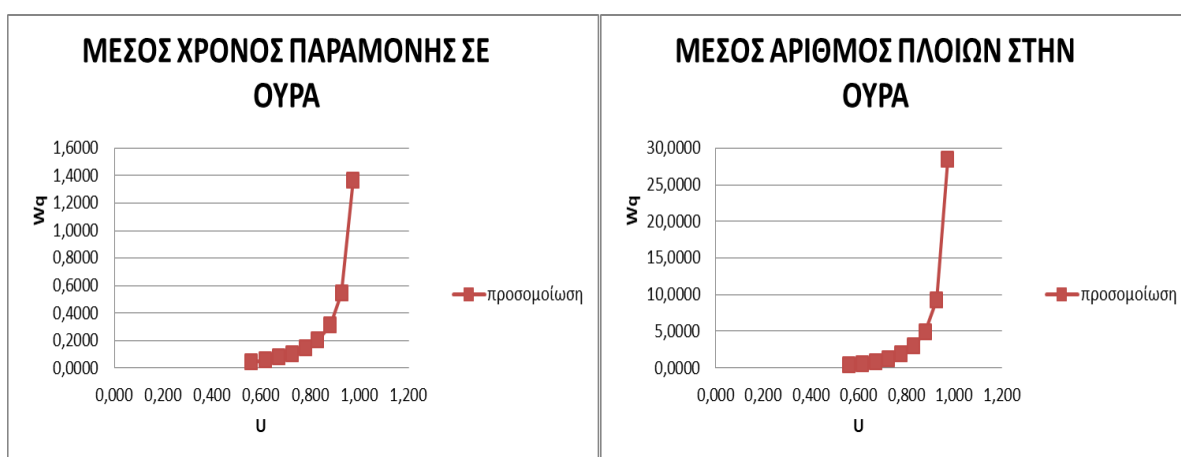
Πίνακας 15 Αναλυτικά αποτελέσματα M/M*/2: $\mu/\mu^*=0,333$



Εικόνα 58 Διαγράμματα $M/M^*/2$: $\mu/\mu^*=0,333$

M/M*/2: $\mu/\mu^*=0,167$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης στον σταθμό 1	Ρυθμός εξυπηρέτησης στον σταθμό 2	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα προσομοίωσης	Υπολογισμός με πρόγραμμα προσομοίωσης
λ	μ	μ	U	Wq	Lq
9	2,5	15	0,558	0,044	0,402
10	2,5	15	0,616	0,059	0,597
11	2,5	15	0,672	0,079	0,875
12	2,5	15	0,725	0,106	1,287
13	2,5	15	0,778	0,145	1,913
14	2,5	15	0,829	0,204	2,948
15	2,5	15	0,880	0,312	4,877
16	2,5	15	0,927	0,543	9,303
17	2,5	15	0,973	1,369	28,468

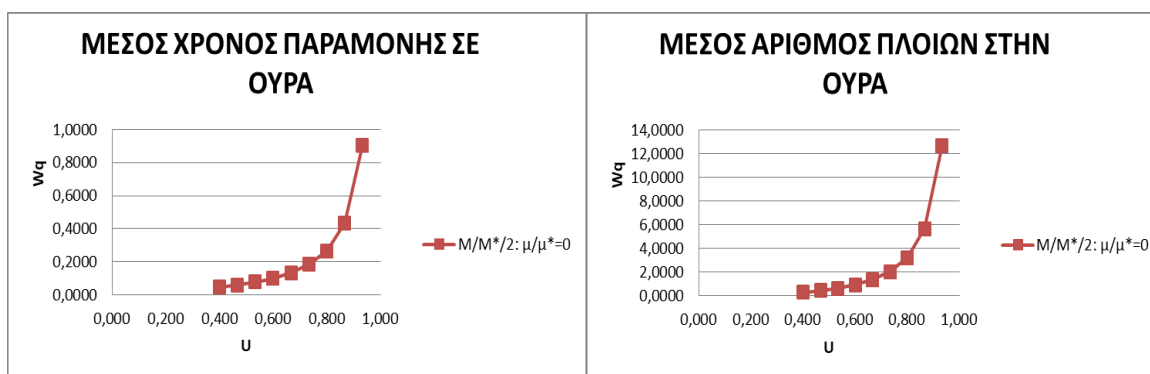
Πίνακας 16 Αναλυτικά αποτελέσματα $M/M^*/2$: $\mu/\mu^*=0,1667$



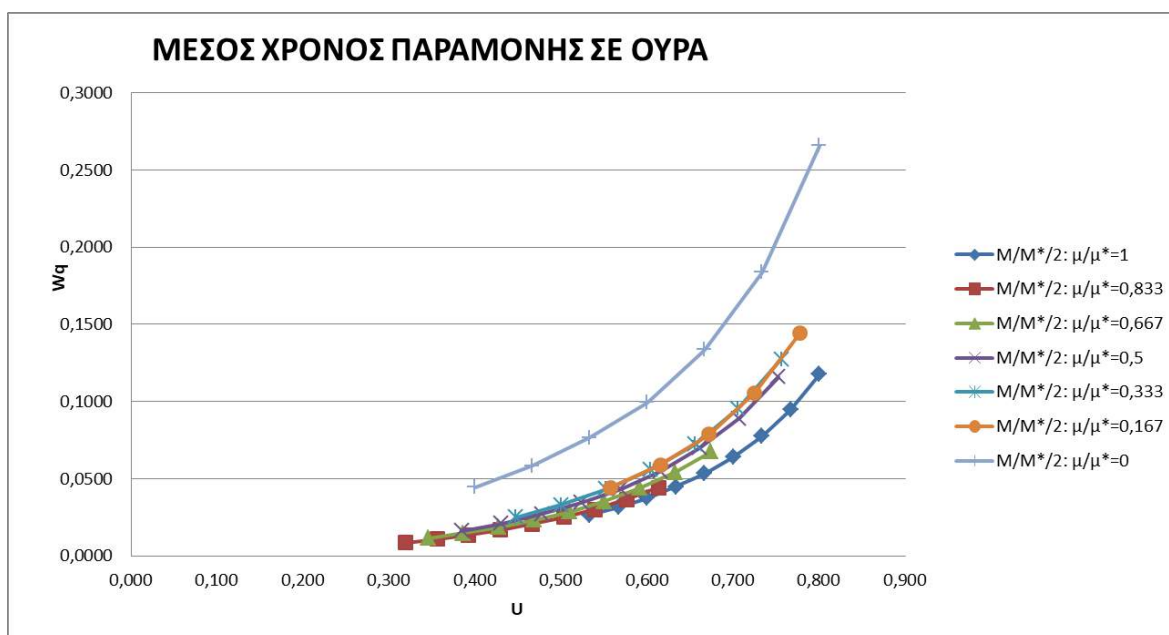
Εικόνα 59 Διαγράμματα $M/M^*/2$: $\mu/\mu^*=0,1667$

M/M*/2: $\mu/\mu^*=0$					
Ρυθμός άφιξης	Ρυθμός εξυπηρέτησης	Ρυθμός εξυπηρέτησης	Βαθμός χρησιμοποίησης	Υπολογισμός με πρόγραμμα	Υπολογισμός με πρόγραμμα
λ	μ	μ	U	Wq	Lq
6	0	15	0,400	0,044	0,267
7	0	15	0,467	0,058	0,408
8	0	15	0,533	0,077	0,612
9	0	15	0,600	0,099	0,895
10	0	15	0,667	0,134	1,335
11	0	15	0,733	0,184	2,020
12	0	15	0,800	0,266	3,189
13	0	15	0,867	0,435	5,652
14	0	15	0,933	0,907	12,695

Εικόνα 60 Αναλυτικά αποτελέσματα M/M*/2: $\mu/\mu^*=0$



Εικόνα 61 Διαγράμματα M/M*/2: $\mu/\mu^*=0$



Εικόνα 62 Συγκριτικά αποτελέσματα

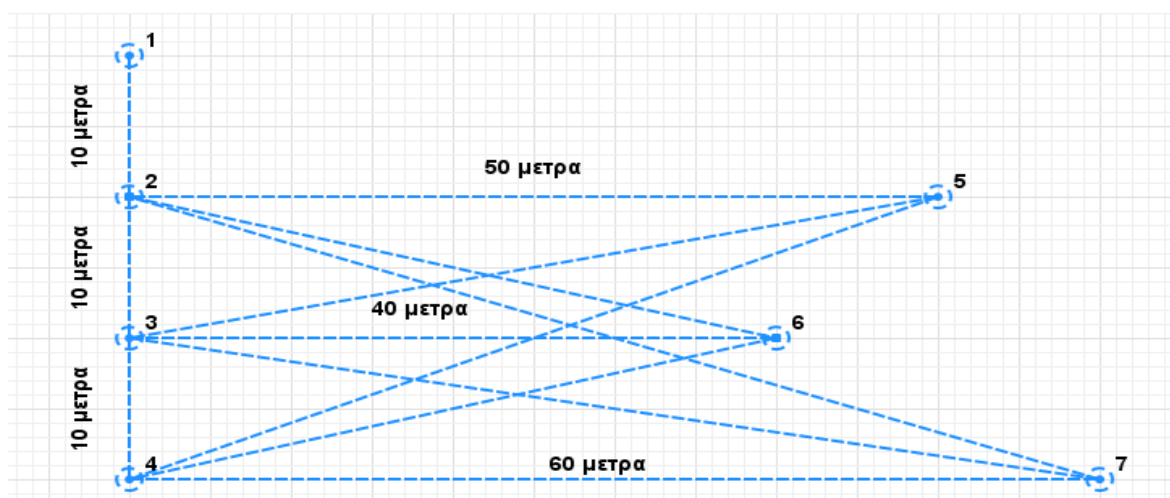
5 ΠΡΟΣΟΜΟΙΩΣΗ ΣΤΡΑΤΗΓΙΚΩΝ ΑΝΑΘΕΣΗΣ ΕΡΓΑΣΙΩΝ ΣΤΑ ΟΧΗΜΑΤΑ ΜΕΤΑΦΟΡΑΣ

Σε αυτό το κεφάλαιο θα παρουσιαστούν διάφορες στρατηγικές ανάθεσης των οχημάτων μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων προκειμένου να αποφευχθεί μια στρατηγική έχει τα βέλτιστα αποτελέσματα, όσον αφορά τον χρόνο ολοκλήρωσης μιας διαδικασίας, με γνώμονα τις διαφορετικές συνθήκες και καταστάσεις που θα μπορούσαν να δημιουργηθούν σε ένα λιμένα. Οι στρατηγικές αυτές αφορούν την απόφαση σε πιο μέσο μεταφοράς θα δοθεί η εντολή ανάθεσης μεταφοράς του εκάστοτε εμπορευματοκιβωτίου, με γνώμονα την σχετική τους απόσταση.

Τα σενάρια που έχουν φτιαχτεί για αυτό το λόγο είναι τα εξής:

- **Σταθερή Ανάθεση:** σε κάθε γερανογέφυρα αντιστοιχεί ένας συγκεκριμένος αριθμός οχημάτων μεταφοράς
- **Ανάθεση στο Κοντινότερο Διαθέσιμο όχημα μεταφοράς :** ελέγχονται οι αποστάσεις όλων των διαθέσιμων οχημάτων μεταφοράς σε σχέση με το εμπορευματοκιβώτιο που πρέπει να μεταφερθεί, και επιλέγεται το κοντινότερο.
- **Ανάθεση στο Πιο Απομακρυσμένο Διαθέσιμο όχημα μεταφοράς :** ελέγχονται οι αποστάσεις όλων των διαθέσιμων οχημάτων μεταφοράς σε σχέση με το εμπορευματοκιβώτιο που πρέπει να μεταφερθεί, και επιλέγεται το πιο απομακρυσμένο.
- **Κυκλικός Κανόνας:** γίνεται επιλογή των οχημάτων μεταφοράς με βάση μιας σειράς, κυκλικά.
- **Πρώτο Διαθέσιμο:** γίνεται επιλογή των οχημάτων μεταφοράς με βάση μιας σειράς, κυκλικά αλλά και με γνώμονα την διαθεσιμότητα του οχήματος μεταφοράς
- **Τυχαίος κανόνας:** γίνεται τυχαία επιλογή ενός οχήματος μεταφοράς

Οι αποστάσεις ανάμεσα στον χώρο στοιβασίας και στην θέση παραβολής φαίνονται στην ακόλουθη φωτογραφία (Εικόνα 63), όπου στο σημείο 1 είναι η αρχική θέση των οχημάτων μεταφοράς, στα σημεία 2 ,3 ,4 οι θέσεις παραβολής, και τέλος στα σημεία 5, 6, 7 ο χώρος στοιβασίας.

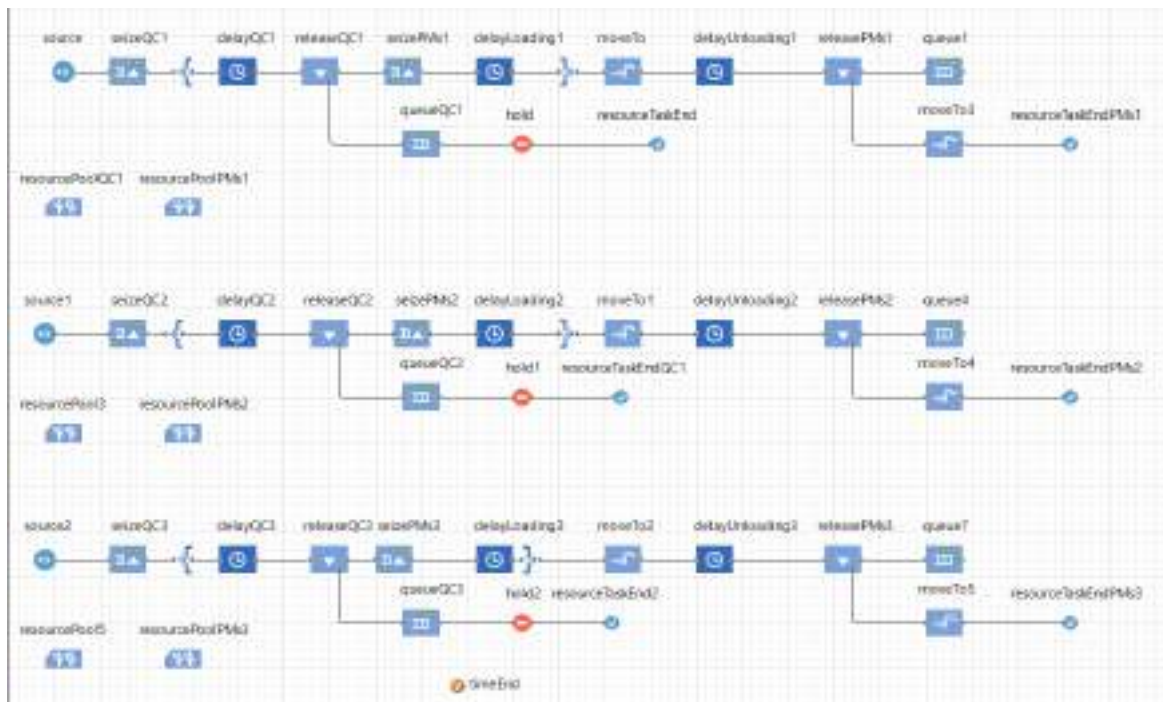


Εικόνα 63 Γράφημα εκφόρτωσης τριών πλοίων

5.1 Σταθερή ανάθεση οχημάτων μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων

Το 1ο σενάριο αφορά στη στρατηγική της σταθερής ανάθεσης των οχημάτων μεταφοράς σε γερανογέφυρες που είναι σε λειτουργία μονού κύκλου και περιγράφει τις διαδικασίες εκφόρτωσης τριών πλοίων που έχουν παραβάσει σε ένα λιμάνι .

Από κάθε πλοίο πρόκειται να εκφορτωθούν εκατό εμπορευματοκιβώτια. Διατίθεται μία γερανογέφυρα στο κρηπίδωμα για να εξυπηρετήσει το κάθε πλοίο. Έχουν εκχωρηθεί δύο οχήματα μεταφοράς σε κάθε γερανογέφυρα. Υπάρχουν τρεις χώροι εναπόθεσης των εμπορευματοκιβωτίων που αντιστοιχούν σε κάθε πλοίο. Κάθε γερανογέφυρα χρειάζεται 50sec για να παραλάβει ένα εμπορευματοκιβώτιο και 10sec για να το φορτώσει στο όχημα μεταφοράς. Τα οχήματα που έχουν εκχωρηθεί στη πρώτη γερανογέφυρα διανύουν απόσταση 50m μέχρι το χώρο στοιβασίας και 50m για να επιστρέψουν πάλι στο κρηπίδωμα. Τα οχήματα που έχουν εκχωρηθεί στη δεύτερη γερανογέφυρα διανύουν απόσταση 40m μέχρι το χώρο στοιβασίας και 40m για να επιστρέψουν πάλι στο κρηπίδωμα. Τα οχήματα που έχουν εκχωρηθεί στη τρίτη γερανογέφυρα διανύουν απόσταση 60m μέχρι το χώρο στοιβασίας και 60m για να επιστρέψουν πάλι στο κρηπίδωμα. Όλα τα οχήματα χρειάζονται 17sec για να εκφορτώσουν τα εμπορευματοκιβώτια στο χώρο στοιβασίας τους και κινούνται με ταχύτητα 15m/min ανεξάρτητα από το αν είναι έμφορτα ή κενά. Επίσης όλα τα οχήματα βρίσκονται σε μια αρχική θέση πριν ξεκινήσει οι διαδικασίες. Η θέση αυτή είναι σε απόσταση 10m από τη πρώτη γερανογέφυρα, 20m από τη δεύτερη γερανογέφυρα και 30m από την τρίτη γερανογέφυρα.



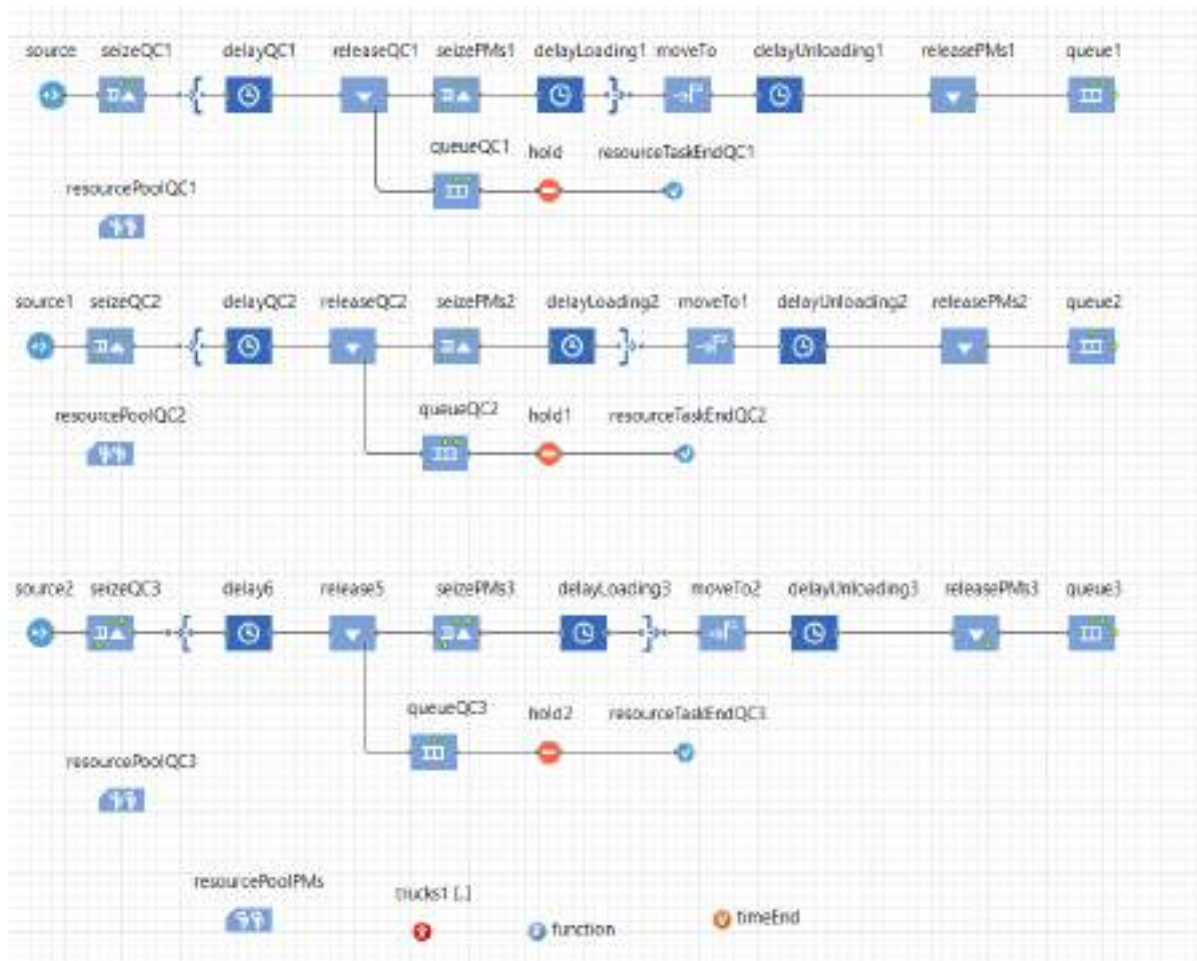
Εικόνα 64 Η διαμόρφωση του μοντέλου εκφόρτωσης τριών πλοίων με το λογισμικό AnyLogic (σταθερή ανάθεση)

5.2 Ανάθεση ομάδας οχημάτων μεταφοράς κατά τις διαδικασίες εκφόρτωσης πλοίων

Το 2ο σενάριο αφορά στη στρατηγική της ανάθεσης ομάδας οχημάτων μεταφοράς σε γερανογέφυρες που είναι σε λειτουργία μονού κύκλου και περιγράφει τις διαδικασίες εκφόρτωσης τριών πλοίων που έχουν παραβάλει σε ένα λιμάνι.

Από κάθε πλοίο πρόκειται να εκφορτωθούν εκατό εμπορευματοκιβώτια όπως και στο προηγούμενο σενάριο. Η διαφορά με το πρώτο σενάριο είναι ότι σε αυτό υπάρχει μια ομάδα από έξι οχήματα μεταφοράς τα οποία θα εκχωρούνται στις γερανογέφυρες ανάλογα με κάποιους κανόνες. Οι κανόνες αυτοί είναι οι εξής:

- **Κυκλικός κανόνας:** Τα οχήματα μεταφοράς θα επιλέγονται με κυκλική σειρά. Δηλαδή θα επιλέγεται το 1ο, μετά το 2ο, το 3ο, το 4ο, το 5ο, το 6ο και ύστερα πάλι το 1ο κτλ. Όμως αν είναι σειρά να επιλεγεί το 3ο και αυτό δεν είναι διαθέσιμο τότε επιλέγεται το 4ο κλπ. Αν έχει επιλεγεί το 4ο και ο εξυπηρετητής ολοκληρώσει την ενέργεια τότε η επόμενη ζήτηση του εκκινεί από το 5ο φορτηγό.
- **Τυχαίος κανόνας:** Τα οχήματα μεταφοράς επιλέγονται με τυχαίο τρόπο από έναν αριθμό διαθέσιμων φορητών.
- **Πρώτο διαθέσιμο:** Θα επιλέγεται πάντα το πρώτο διαθέσιμο όχημα μεταφοράς. Δηλαδή θα ελέγχεται πρώτα αν είναι διαθέσιμο το 1ο. Αν ναι θα επιλέγεται αυτό, αλλιώς θα ελέγχεται το 2ο όχημα μεταφοράς και αν είναι διαθέσιμο θα επιλέγεται αλλιώς θα γίνεται έλεγχος στο 3ο κτλ. Όταν η ενέργεια ολοκληρωθεί, η επιλογή ξεκινά πάλι από το 1ο φορτηγό και αν αυτό δεν είναι διαθέσιμο συνεχίζει στο επόμενο.
- **Μέγιστη απόσταση από το εμπορευματοκιβώτιο:** Με τον κανόνα αυτό, όταν υπάρχει ένας αριθμός οχημάτων μεταφοράς που είναι διαθέσιμα τότε θα επιλέγεται αυτό που βρίσκεται σε μεγαλύτερη απόσταση από το εμπορευματοκιβώτιο.
- **Ελάχιστη απόσταση από το εμπορευματοκιβώτιο:** Με τον κανόνα αυτό, όταν υπάρχει ένας αριθμός οχημάτων μεταφοράς που είναι διαθέσιμα τότε θα επιλέγεται αυτό που βρίσκεται σε μικρότερη απόσταση από το εμπορευματοκιβώτιο.



Εικόνα 65 Η διαμόρφωση του μοντέλου εκφόρτωσης τριών πλοίων με το λογισμικό AnyLogic (ανάθεση ομάδας)

Για όλα τα παραπάνω χρησιμοποιήθηκε το παραπάνω μοντέλο με αλλαγές στο **function** με το οποίο καθορίζεται η διαδικασία επιλογής του οχήματος μεταφοράς εκτός από τον κυκλικό κανόνα που χρησιμοποιήθηκαν και άλλες δύο βοηθητικές μεταβλητές.

5.2.1 Επιλογή ανάθεσης οχημάτων μεταφοράς με κυκλικό κανόνα

```
variable1=trucks1.get(variable2);
if (variable2<=4)
variable2=variable2+1;
else variable2=0;
```

Εικόνα 66 Κώδικας κυκλικού κανόνα

Με αυτόν το κώδικα η τιμή variable1 αποτελεί το όχημα μεταφοράς που έχει σειρά, ενώ η variable2 αυξάνει κατά 1 τον αριθμό κάθε φορά, μέχρι να φτάσει τον συνολικό αριθμό των οχημάτων μεταφοράς, όπου και ξανά μηδενίζεται.

5.2.2 Επιλογή ανάθεσης οχημάτων μεταφοράς με τυχαίο κανόνα

```
variable1=trucks1.random();
```

Με αυτόν τον κώδικα, η τιμή variable1 αποτελεί ένα τυχαίο όχημα μεταφοράς, ενώ η επιλογή γίνεται με χρήση του κώδικα

```
return variable1.isIdle();
```

Εικόνα 67 Κώδικας τυχαίου κανόνα

Όπου ελέγχεται αν το συγκεκριμένο τυχαίο φορτηγό είναι διαθέσιμο, και αν είναι επιστρέφεται η τιμή true προκειμένου να δοθεί η εντολή ανάθεσης.

5.2.3 Πρώτο διαθέσιμο όχημα μεταφοράς

function - Function

Name: function ☒ Show name ☐ Ignore
Visible: ☒ yes
☐ Just action (returns nothing)
☒ Returns value
Type: boolean

Arguments

Name	Type
agent	container
unit	trucks

Function body

```
variable1=unit;  
for (trucks t : trucks1) {  
    if( t.isIdle() ) {  
        if (unit==null){  
            unit=t;}  
    }  
}  
return unit.isIdle();
```

Εικόνα 68 Κώδικας κανόνα πρώτου διαθέσιμου

Με αυτόν τον κώδικα η `variable1` αποτελεί ένα όχημα μεταφοράς, όπου αρχικά δεν έχει κάποια τιμή. Με την `for` γίνεται προσπέλαση όλων των οχημάτων μεταφοράς. Όταν βρεθεί το πρώτο διαθέσιμο με την `if` το `unit` παίρνει την τιμή αυτού, και τέλος επιστρέφεται `true` προκειμένου να γίνει η ανάθεση.

5.2.4 Μέγιστη απόσταση από το εμπορευματοκιβώτιο

function - Function

Name: function ☒ Show name ☐ Ignore

Visible: ☒ yes

☐ Just action (returns nothing)

☒ Returns value

Type: boolean

Arguments

Name	Type
agent	container
unit	trucks

Function body

```
double dist = 0;
for (trucks t : trucks1) {
    if( t.isIdle() && agent.distanceTo(t) >= dist ) {
        dist = agent.distanceTo(t);
    }
}

return agent.distanceTo(unit) == dist;
```

Εικόνα 69 Κώδικας κανόνα μέγιστης απόστασης

Με αυτόν τον κώδικα αρχικοποιείται η μεταβλητή `dist` σε ένα ελάχιστο νούμερο. Στην συνέχεια γίνεται προσπέλαση όλων των οχημάτων μεταφοράς. Με την εντολή `if` γίνεται επιλογή των διαθέσιμων και αν η απόσταση του οχήματος μεταφοράς από τον **πράκτορα**, δηλαδή του εμπορευματοκιβωτίου, η μεταβλητή `dist` λαμβάνει την νέα μεγαλύτερη τιμή. Στο τέλος επιστρέφεται `true` για το όχημα μεταφοράς που έχει την απόσταση αυτή και γίνεται η ανάθεση.

5.2.5 Ελάχιστη απόσταση από το εμπορευματοκιβώτιο

function - Function

Name: function

☒ Show name ☐ Ignore

Visible: ☒ yes

☐ Just action (returns nothing)

☒ Returns value

Type:

boolean

Arguments

Name	Type
agent	container
unit	trucks

Function body

```
double dist = 1000;
for (trucks t : trucks1) {
    if( t.isIdle() && agent.distanceTo(t) <= dist ) {
        dist = agent.distanceTo(t);
    }
}

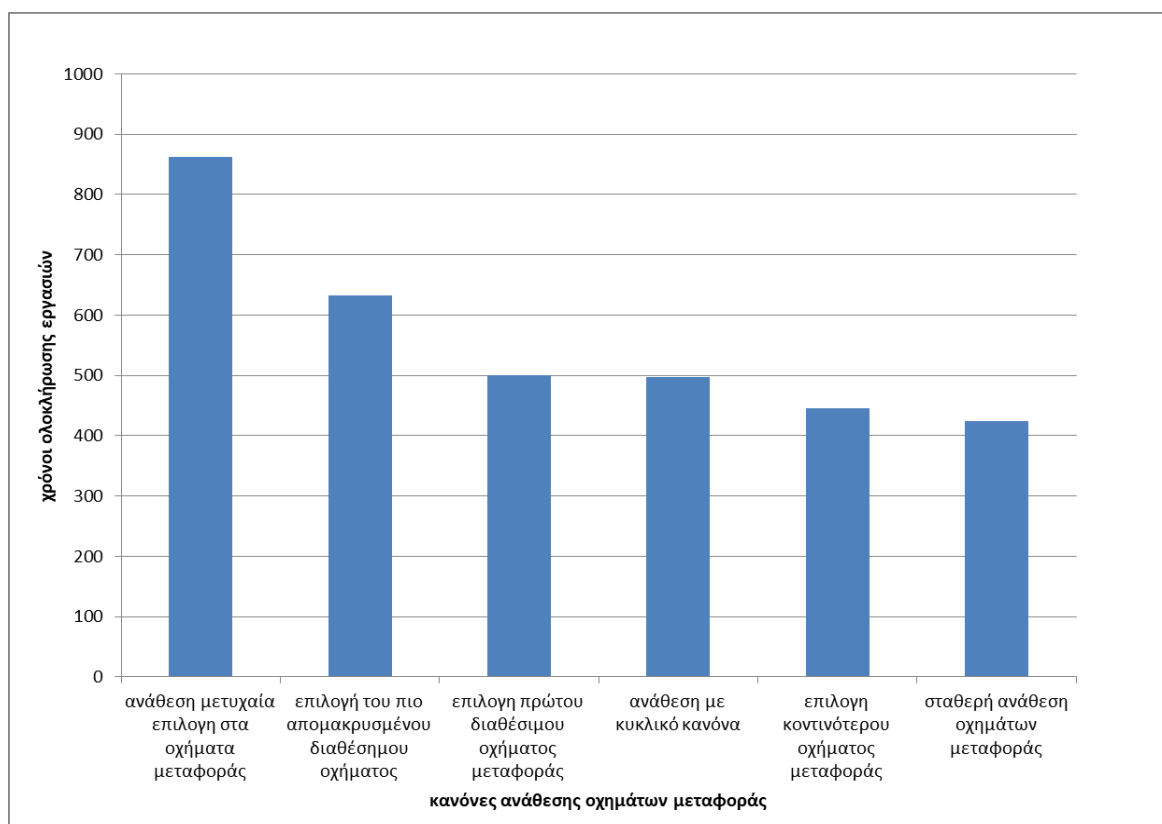
return agent.distanceTo(unit) == dist;
```

Εικόνα 70 Κώδικας κανόνα ελάχιστης απόστασης

Με αυτόν τον κώδικα αρχικοποιείται η μεταβλητή **dist** σε ένα μέγιστο νούμερο. Στην συνέχεια γίνεται προσπάθεια όλων των οχημάτων μεταφοράς. Με την εντολή **if** γίνεται επιλογή των διαθέσιμων και αν η απόσταση του όχημα μεταφοράς από τον **πράκτορα**, δηλαδή του εμπορευματοκιβωτίου, η μεταβλητή **dist** λαμβάνει την νέα μικρότερη τιμή. Στο τέλος επιστρέφεται **true** για το όχημα μεταφοράς που έχει την απόσταση αυτή και γίνεται η ανάθεση.

5.3 Συγκριτικά αποτελέσματα

Μετά το τρέξιμο των μοντέλων δημιουργήθηκε το παρακάτω διάγραμμα



Εικόνα 71 Αποτελέσματα εκφόρτωσης τριών πλοίων

Όπως είναι φανερό, την χειρότερη απόδοση είχε ο τυχαίος κανόνας, με 861,96 λεπτά για την ολοκλήρωση, στην συνέχεια ακολουθεί η μεγαλύτερη απόσταση με 632,46 λεπτά. Έπεται στην σειρά η διαδικασία επιλογής του πρώτου διαθέσιμου (499,8 λεπτά), με μικρή διαφορά από το κυκλικό, όπου χρειάστηκαν 497,62 λεπτά. Τις καλύτερες αποδόσεις είχαν η επιλογή της κοντινότερου δυνατού φορτηγού με 444,98 λεπτά και η σταθερή ανάθεση με 424,33 λεπτά.

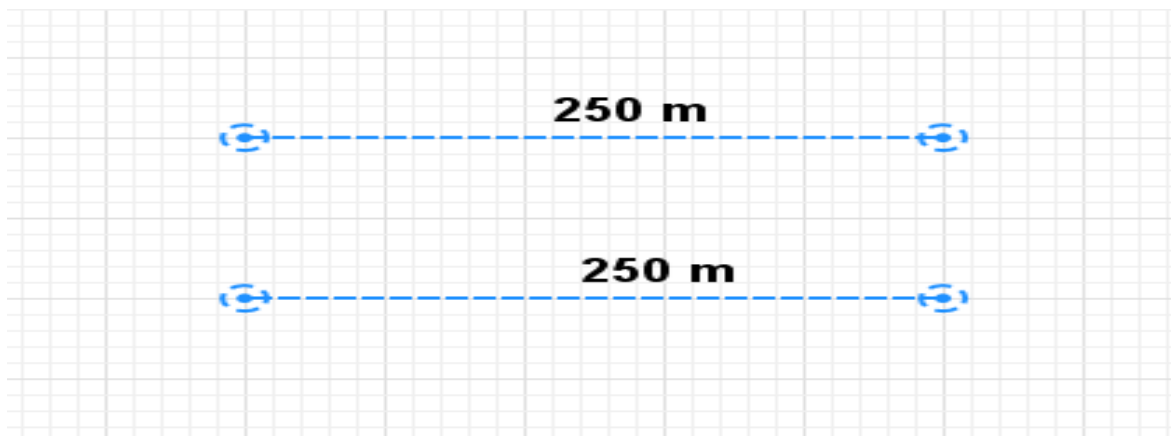
5.4 Σταθερή ανάθεση οχημάτων μεταφοράς στις γερανογέφυρες σταθερού κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων.

Το σενάριο αφορά στη στρατηγική της σταθερής ανάθεσης των οχημάτων μεταφοράς σε γερανογέφυρες που είναι σε λειτουργία μονού κύκλου και περιγράφει τις διαδικασίες φόρτωσης και εκφόρτωσης δύο πλοίων που έχουν παραβάλει σε ένα λιμάνι.

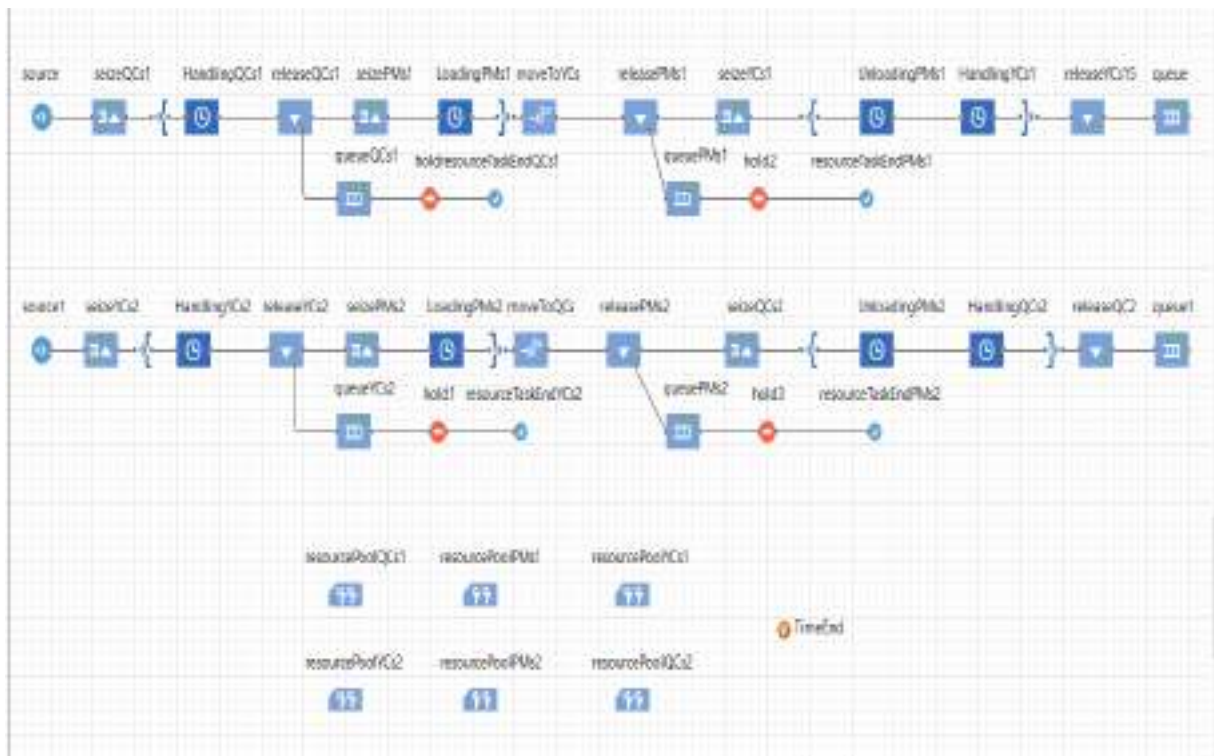
Από το πρώτο πλοίο πρόκειται να εκφορτωθούν εκατό (100) εμπορευματοκιβώτια. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να το εξυπηρετήσει και ένα όχημα μεταφοράς που θα εκτελεί τη διαδρομή από το κρηπίδωμα στο χώρο στοιβάσεως και αντίστροφα. Το όχημα μεταφοράς κινείται με ταχύτητα 20km/h και η απόσταση που πρέπει να διανύσει από το κρηπίδωμα στο χώρο στοιβάσεως είναι 250m. Ομοίως η απόσταση από το χώρο στοιβάσεως στο

κρηπίδωμα είναι 250m. Η γερανογέφυρα έχει ικανότητα 20 εμπορευματοκιβωτίων/ώρα, δηλαδή χρειάζεται περίπου 3min για να παραλάβει κάθε εμπορευματοκιβώτιο και άλλο 1min για να το φορτώσει στο όχημα μεταφοράς. Όταν το όχημα μεταφοράς φτάνει στο χώρο στοιβασίας, απαιτείται 1min για να εκφορτωθεί.

Στο δεύτερο πλοίο πρόκειται να φορτωθούν εκατό (100) εμπορευματοκιβώτια. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να το εξυπηρετήσει και ένα όχημα μεταφοράς που θα εκτελεί τη διαδρομή από το κρηπίδωμα στο χώρο στοιβασίας και αντίστροφα. Το όχημα μεταφοράς κινείται με ταχύτητα 20km/h και η απόσταση που πρέπει να διανύσει από το κρηπίδωμα στο χώρο στοιβασίας είναι 250m. Ομοίως η απόσταση από το χώρο στοιβασίας στο κρηπίδωμα είναι 250m. Το όχημα μεταφοράς χρειάζεται 1min ώστε να φορτωθεί με ένα εμπορευματοκιβώτιο στο χώρο στοιβασίας και άλλο 1min ώστε να εκφορτωθεί στο κρηπίδωμα από τη γερανογέφυρα. Η γερανογέφυρα έχει ικανότητα 20 εμπορευματοκιβωτίων/ώρα, δηλαδή χρειάζεται περίπου 3min για να παραλάβει ένα εμπορευματοκιβώτιο από το όχημα μεταφοράς και να το εναποθέσει στο πλοίο.



Εικόνα 72 Γράφημα φόρτωσης-εκφόρτωσης 2 πλοίων (σταθερού κύκλου/σταθερή ανάθεση)



Εικόνα 73 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (σταθερού κύκλου/σταθερή ανάθεση)

HandlingQCs1 - Delay

Name:
HandlingQCs1
☒ Show name
☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

HandlingYCs2 - Delay

Name:
HandlingYCs2
☒ Show name
☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

⌚ LoadingPMs1 - Delay

Name:

LoadingPMs1

☒ Show name ☐ Ignore

Type:

⇒

☒ Specified time

☐ Until stopDelay() is called

Delay time:

↺

1

minutes

▼

Capacity:

⇒

1

⌚ LoadingPMs2 - Delay

Name:

LoadingPMs2

☒ Show name ☐ Ignore

Type:

⇒

☒ Specified time

☐ Until stopDelay() is called

Delay time:

↺

1

minutes

▼

Capacity:

⇒

1

⌚ UnloadingPMs1 - Delay

Name:

UnloadingPMs1

☒ Show name ☐ Ignore

Type:

⇒

☒ Specified time

☐ Until stopDelay() is called

Delay time:

↺

1

minutes

▼

Capacity:

⇒

1

⌚ UnloadingPMs2 - Delay

Name:

UnloadingPMs2

☒ Show name ☐ Ignore

Type:

⇒

☒ Specified time

☐ Until stopDelay() is called

Delay time:

↺

1

minutes

▼

Capacity:

⇒

1

HandlingYCs1 - Delay

Name: HandlingYCs1 ☒ Show name ☐ Ignore
Type: ☒ Specified time
☐ Until stopDelay() is called
Delay time:
Capacity:

HandlingQCs2 - Delay

Name: HandlingQCs2 ☒ Show name ☐ Ignore
Type: ☒ Specified time
☐ Until stopDelay() is called
Delay time:
Capacity:

Εικόνα 74 Σενάριο 2.1 χρόνοι αναμονής

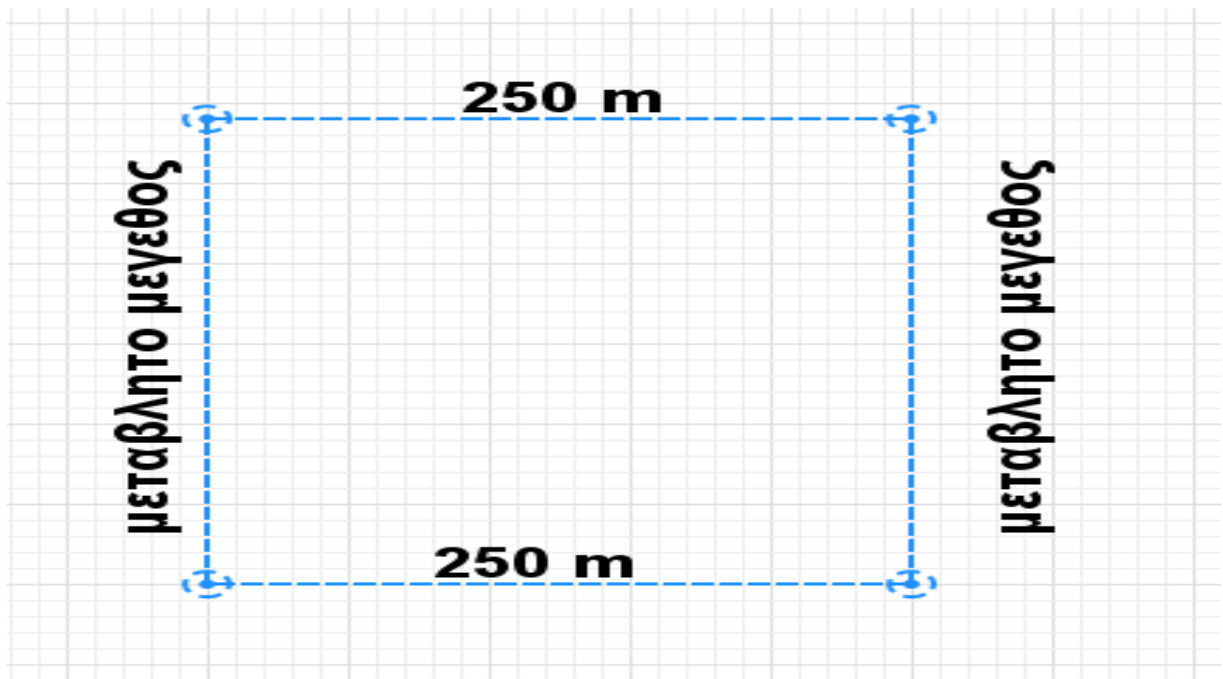
5.5 Ανάθεση ομάδας οχημάτων μεταφοράς στις γερανογέφυρες σταθερού κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων

Το σενάριο αφορά στη στρατηγική της ανάθεσης ομάδας οχημάτων μεταφοράς σε γερανογέφυρες διαφορετικών πλοίων, που είναι σε λειτουργία μονού κύκλου. Περιγράφει τις διαδικασίες φόρτωσης και εκφόρτωσης δύο πλοίων που έχουν παραβάλει σε ένα λιμάνι .

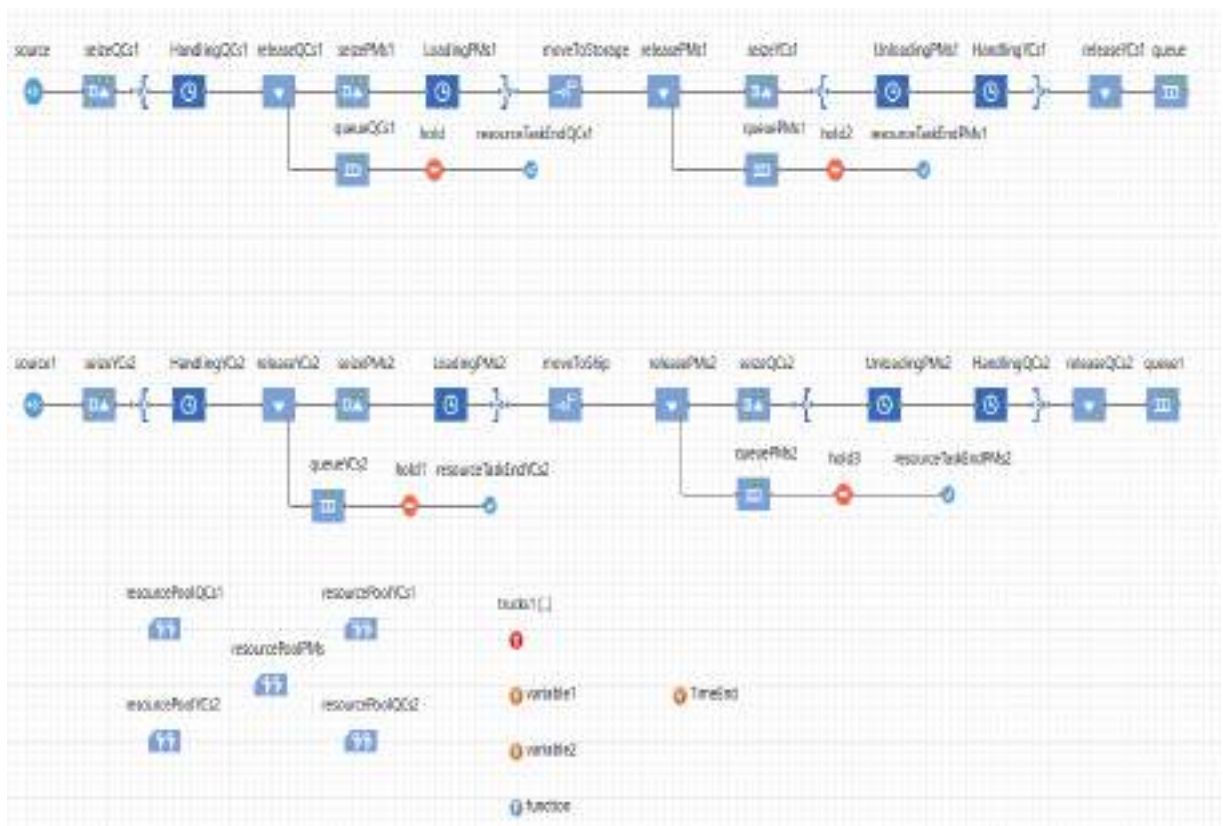
Από το πρώτο πλοίο πρόκειται να εκφορτωθούν 100 εμπορευματοκιβώτια και να εναποτεθούν σε ένα χώρο στοιβασίας. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να εκτελεί διαδικασίες εκφόρτωσης. Στο δεύτερο πλοίο πρόκειται να φορτωθούν 100 εμπορευματοκιβώτια από ένα άλλο χώρο στοιβασίας και διατίθεται μία γερανογέφυρα στο κρηπίδωμα να εκτελεί διαδικασίες φόρτωσης.

Για την πραγματοποίηση αυτών των διαδικασιών ανατίθεται μια ομάδα από δύο οχήματα μεταφοράς να εξυπηρετήσουν τις δύο γερανογέφυρες. Τα οχήματα μεταφοράς κινούνται με ταχύτητα 20km/h. Ο χρόνος φόρτωσης ή εκφόρτωσης εμπορευματοκιβωτίων από τα οχήματα είναι 1 λεπτό και ο χρόνος κύκλου των γερανογεφυρών είναι 3 λεπτά. Τα οχήματα ξεκινούν από το πρώτο πλοίο, όπου η γερανογέφυρα τους φορτώνει ένα εμπορευματοκιβώτιο. Ύστερα το μεταφέρουν στον πρώτο χώρο στοιβασίας και το εναποθέτουν εκεί. Σε κενή κατάσταση μεταφέρονται στο δεύτερο χώρο στοιβασίας όπου παραλαμβάνουν ένα εξαγόμενο εμπορευματοκιβώτιο και το μεταφέρουν στη δεύτερη γερανογέφυρα για να το παραλάβει και να το φορτώσει στο δεύτερο πλοίο. Τελικά επιστρέφουν σε κενή κατάσταση στο πρώτο πλοίο για να παραλάβουν επόμενο εισαγόμενο εμπορευματοκιβώτιο. Οι αποστάσεις μεταξύ “γερανογέφυρα – χώρο στοιβασίας” και αντίστροφα είναι 250m όπως και στο προηγούμενο σενάριο. Στη στρατηγική αυτή όμως απαιτείται και ένα επιπλέον δεδομένο που είναι η απόσταση μεταξύ των χώρων στοιβασίας. Το σενάριο αυτό λοιπόν, θα εξετάσει διάφορες περιπτώσεις για διάφορες

τιμές αποστάσεων μεταξύ των χώρων στοιβασίας. Το μοντέλο που ακολουθεί λαμβάνει μια αρχική απόσταση των χώρων 100m.



Εικόνα 75 Γράφημα φόρτωσης-εκφόρτωσης 2 πλοίων (σταθερού κύκλου/ανάθεση ομάδας)



Εικόνα 76 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (σταθερού κύκλου/ ανάθεση ομάδας)

⌚

HandlingQCs1 - Delay

Name:

HandlingQCs1

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time

☐ Until stopDelay() is called

Delay time:

⌵

3

minutes

▼

Capacity:

⌵

1

⌚

HandlingYCs2 - Delay

Name:

HandlingYCs2

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time

☐ Until stopDelay() is called

Delay time:

⌵

3

minutes

▼

Capacity:

⌵

1

⌚

LoadingPMs1 - Delay

Name:

LoadingPMs1

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time

☐ Until stopDelay() is called

Delay time:

⌵

1

minutes

▼

Capacity:

⌵

1

⌚

LoadingPMs2 - Delay

Name:

LoadingPMs2

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time

☐ Until stopDelay() is called

Delay time:

⌵

1

minutes

▼

Capacity:

⌵

1

⌚

UnloadingPMs1 - Delay

Name:

UnloadingPMs1

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time

☐ Until stopDelay() is called

Delay time:

⌵

1

minutes

▼

Capacity:

⌵

1

⌚ UnloadingPMs2 - Delay

Name:

UnloadingPMs2

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

⌚ HandlingYCs1 - Delay

Name:

HandlingYCs1

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

⌚ HandlingQCs2 - Delay

Name:

HandlingQCs2

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

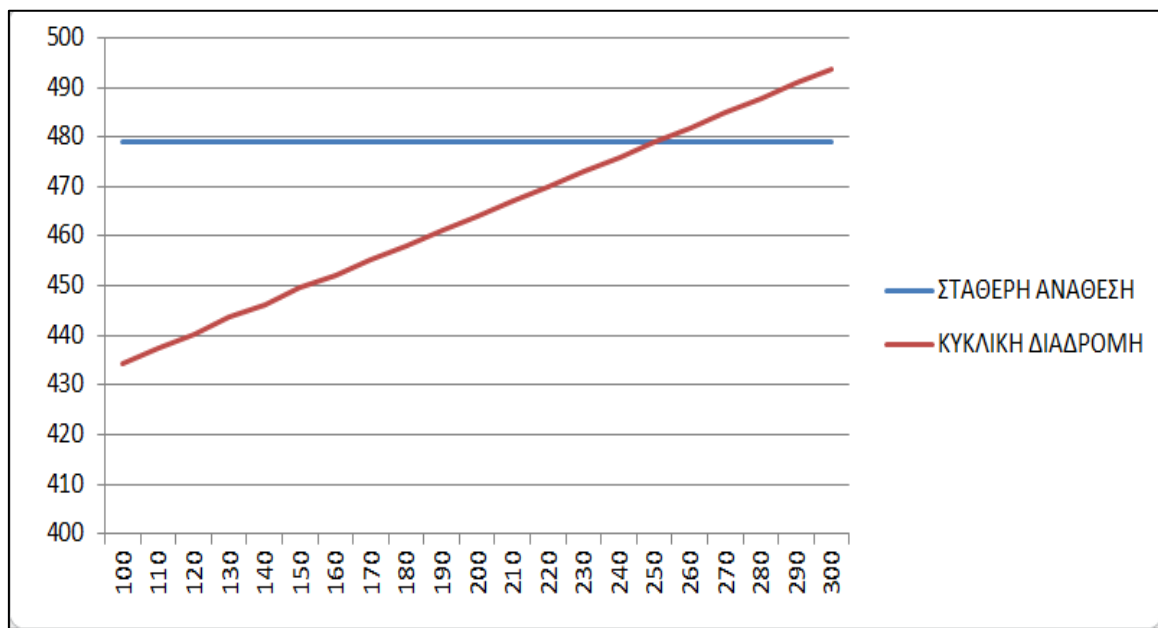
Capacity:

Εικόνα 77 Σενάριο 2.2 χρόνοι αναμονής

5.6 Συγκριτικά Αποτελέσματα σεναρίων

	ΣΤΑΘΕΡΗ ΑΝΑΘΕΣΗ	ΚΥΚΛΙΚΗ ΔΙΑΔΡΟΜΗ
ΜΕΤΡΑ	min	min
100	479 (σε όλες τις περιπτώσεις)	434,45
110		437,42
120		440,38
130		443,75
140		446,32
150		449,75
160		452,27
170		455,23
180		458,2
190		461,18
200		464,15
210		467,12
220		470,08
230		473,05
240		476,03
250		479
260		481,97
270		484,93
280		487,92
290		490,88
300		493,85

Πίνακας 17 Συγκριτικά αποτελέσματα σεναρίων 2.1, 2.2



Εικόνα 78 Συγκριτικά αποτελέσματα σεναρίων 2.1, 2.2

5.7 Προσομοίωση στρατηγικών ανάθεσης εργασιών στα οχήματα μεταφοράς με μεταβλητούς χρόνους κύκλου κατά τις διαδικασίες φόρτωσης – εκφόρτωσης πλοίων

Όπως αποδείχθηκε και από τον Frankel, E.G. (1980), οι αποκλίσεις που παρουσιάζουν διάφοροι τύποι εξοπλισμού δεν επηρεάζουν μόνο την επίδοση και το χρόνο κύκλου του συγκεκριμένου εξοπλισμού επηρεάζουν και το χρόνο κύκλου άλλων τύπων εξοπλισμού με τους οποίους αλληλεπιδρά. Για το λόγο αυτό είναι σημαντικό να εξετάζονται οι αποκλίσεις που έχουν οι χρόνοι κύκλου του εξοπλισμού που χρησιμοποιείται, για να εκτιμάται η πραγματική ικανότητα του συστήματος η οποία είναι τελικά μικρότερη από την αναμενόμενη. Η επεξήγηση αυτού του φαινομένου βρίσκεται στο Παράρτημα Δ.

Τα μοντέλα που επιλύθηκαν στην προηγούμενη ενότητα και αφορούσαν τις στρατηγικές σταθερής ανάθεσης και ανάθεσης ομάδας, επιλύθηκαν με την παραδοχή ότι οι χρόνοι κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς είναι σταθεροί. Στην πραγματικότητα όμως και οι δύο χρόνοι κύκλου έχουν αποκλίσεις. Ο χρόνος κύκλος της γερανογέφυρας παρουσιάζει αποκλίσεις λόγω πιθανόν αναδιατάξεων που θα χρειαστεί να εκτελέσει η γερανογέφυρα σε ένα πλοίο, είτε σε διαδικασίες φόρτωσης είτε εκφόρτωσης. Οι λόγοι αποκλίσεων στο χρόνο κύκλου των οχημάτων μεταφοράς και η μέθοδος εκτίμησης τους παρουσιάζονται στην επόμενη ενότητα και αφορούν κυρίως πιθανές αναδιατάξεις στους χώρους στοιβασίας. Πραγματοποιήθηκε λοιπόν επίλυση των παραπάνω μοντέλων λαμβάνοντας υπόψη αποκλίσεις στο χρόνο κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς.

5.8 Σταθερή ανάθεση με αποκλίσεις στο χρόνο κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς

Το σενάριο αφορά στη στρατηγική της σταθερής ανάθεσης των οχημάτων μεταφοράς σε γερανογέφυρες που είναι σε λειτουργία μονού κύκλου και περιγράφει τις διαδικασίες φόρτωσης και εκφόρτωσης δύο πλοίων που έχουν παραβάλει σε ένα λιμάνι.

Από το πρώτο πλοίο πρόκειται να εκφορτωθούν εκατό (100) εμπορευματοκιβώτια. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να το εξυπηρετήσει και ένα όχημα μεταφοράς που θα εκτελεί τη διαδρομή από το κρηπίδωμα στο χώρο στοιβασίας και αντίστροφα. Το όχημα μεταφοράς κινείται με ταχύτητα 20km/h και η απόσταση που πρέπει να διανύσει από το κρηπίδωμα στο χώρο στοιβασίας είναι 250m. Ομοίως η απόσταση από το χώρο στοιβασίας στο κρηπίδωμα είναι 250m. Η γερανογέφυρα χρειάζεται κατά μέσο όρο 3min ακολουθώντας κανονική κατανομή για να παραλάβει κάθε εμπορευματοκιβώτιο και άλλο 1min κατά μέσο όρο ακολουθώντας κανονική κατανομή για να το φορτώσει στο όχημα μεταφοράς. Όταν το όχημα μεταφοράς φτάνει στο χώρο στοιβασίας, απαιτείται επίσης 1min κατά μέσο όρο ακολουθώντας κανονική κατανομή για να εκφορτωθεί.

Στο δεύτερο πλοίο πρόκειται να φορτωθούν εκατό (100) εμπορευματοκιβώτια. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να το εξυπηρετήσει και ένα όχημα μεταφοράς που θα εκτελεί τη διαδρομή από το κρηπίδωμα στο χώρο στοιβασίας και αντίστροφα. Το όχημα μεταφοράς κινείται με ταχύτητα 20km/h και η απόσταση που πρέπει να διανύσει από το κρηπίδωμα στο χώρο στοιβασίας είναι 250m. Ομοίως η απόσταση από το χώρο στοιβασίας στο κρηπίδωμα είναι

The figure displays two Petri nets, one for process P1 (top) and one for process P2 (bottom). Each Petri net consists of places (circles) and transitions (squares). The places represent the state of the system, including the availability of resources (e.g., resourcePoolQC1, resourcePoolPM1, resourcePoolPC1) and the state of the processes (e.g., source, wait, hold, move, release, queue). The transitions represent the actions performed by the processes, such as acquiring resources, holding them, moving them, and releasing them. The Petri nets illustrate the sequence of operations and the associated resource states for each process.

⌚ HandlingQCs1 - Delay

Name: HandlingQCs1 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ HandlingYCs2 - Delay

Name: HandlingYCs2 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ LoadingPMs1 - Delay

Name: LoadingPMs1 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ LoadingPMs2 - Delay

Name: LoadingPMs2 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ UnloadingPMs1 - Delay

Name: UnloadingPMs1 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ UnloadingPMs2 - Delay

Name: UnloadingPMs2 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚

HandlingYCs1 - Delay

Name:

HandlingYCs1

☒ Show name

☐ Ignore

Type:

⇒

☒ Specified time
☐ Until stopDelay() is called

Delay time:

↺

normal(sigma, 3)

minutes

▼

Capacity:

⇒

1

⌚

HandlingQCs2 - Delay

Name:

HandlingQCs2

☒ Show name

☐ Ignore

Type:

⇒

☒ Specified time
☐ Until stopDelay() is called

Delay time:

↺

normal(sigma, 3)

minutes

▼

Capacity:

⇒

1

Εικόνα 80 Σενάριο 2.3 χρόνοι αναμονής

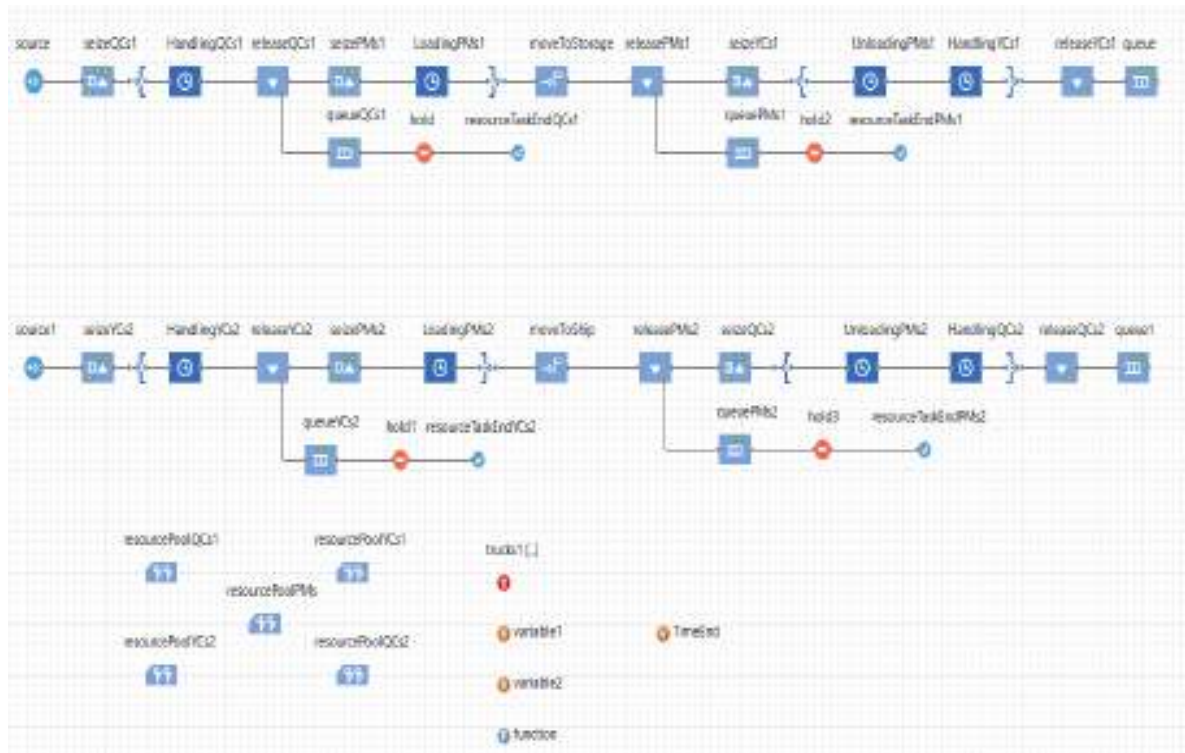
5.9 Ανάθεση ομάδας με αποκλίσεις στο χρόνο κύκλου των γερανογεφυρών και των οχημάτων μεταφοράς

Το σενάριο αφορά στη στρατηγική της ανάθεσης ομάδας οχημάτων μεταφοράς σε γερανογέφυρες διαφορετικών πλοίων, που είναι σε λειτουργία μονού κύκλου. Περιγράφει τις διαδικασίες φόρτωσης και εκφόρτωσης δύο πλοίων που έχουν παραβάλει σε ένα λιμάνι.

Από το πρώτο πλοίο πρόκειται να εκφορτωθούν 100 εμπορευματοκιβώτια και να εναποτεθούν σε ένα χώρο στοιβασίας. Διατίθεται μια γερανογέφυρα στο κρηπίδωμα για να εκτελεί διαδικασίες εκφόρτωσης. Στο δεύτερο πλοίο πρόκειται να φορτωθούν 100 εμπορευματοκιβώτια από ένα άλλο χώρο στοιβασίας και διατίθεται μία γερανογέφυρα στο κρηπίδωμα να εκτελεί διαδικασίες φόρτωσης.

Για την πραγματοποίηση αυτών των διαδικασιών ανατίθεται μια ομάδα από δύο οχήματα μεταφοράς να εξυπηρετήσουν τις δύο γερανογέφυρες. Τα οχήματα μεταφοράς κινούνται με ταχύτητα 20km/h. Ο χρόνος φόρτωσης ή εκφόρτωσης εμπορευματοκιβωτίων από τα οχήματα είναι 1 λεπτό και ο χρόνος κύκλου των γερανογεφυρών είναι 3 λεπτά. Τα οχήματα ξεκινούν από το πρώτο πλοίο, όπου η γερανογέφυρα τους φορτώνει ένα εμπορευματοκιβώτιο. Ύστερα το μεταφέρουν στον πρώτο χώρο στοιβασίας και το εναποθέτουν εκεί. Σε κενή κατάσταση μεταφέρονται στο δεύτερο χώρο στοιβασίας όπου παραλαμβάνουν ένα εξαγόμενο εμπορευματοκιβώτιο και το μεταφέρουν στη δεύτερη γερανογέφυρα για να το παραλάβει και να το φορτώσει στο δεύτερο πλοίο. Τελικά επιστρέφουν σε κενή κατάσταση στο πρώτο πλοίο για να παραλάβουν επόμενο εισαγόμενο εμπορευματοκιβώτιο. Οι αποστάσεις μεταξύ “γερανογέφυρα – χώρο στοιβασίας” και αντίστροφα είναι 250m όπως και στο προηγούμενο σενάριο. Στη στρατηγική αυτή όμως απαιτείται και ένα επιπλέον δεδομένο που είναι η απόσταση μεταξύ των χώρων στοιβασίας. Το σενάριο αυτό λοιπόν, θα εξετάσει διάφορες περιπτώσεις για διάφορες

τιμές αποστάσεων μεταξύ των χώρων στοιβασίας. Το μοντέλο που ακολουθεί λαμβάνει μια αρχική απόσταση των χώρων 100m.



Εικόνα 81 Η διαμόρφωση του μοντέλου φόρτωσης-εκφόρτωσης δύο πλοίων με το λογισμικό AnyLogic (αποκλίσεις στον χρόνο κύκλου / ανάθεση ομάδας)

⌚

HandlingQCs1 - Delay

Name:

HandlingQCs1

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

⌚

HandlingYCs2 - Delay

Name:

HandlingYCs2

☒ Show name ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

minutes

Capacity:

⌚ LoadingPMs1 - Delay

Name: LoadingPMs1 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ LoadingPMs2 - Delay

Name: LoadingPMs2 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ UnloadingPMs1 - Delay

Name: UnloadingPMs1 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ UnloadingPMs2 - Delay

Name: UnloadingPMs2 ☒ Show name ☐ Ignore

Type: ☒ Specified time
☐ Until stopDelay() is called

Delay time:

Capacity:

⌚ HandlingYCs1 - Delay

Name:

HandlingYCs1

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time
☐ Until stopDelay() is called

Delay time:

↺

Capacity:

⌵

⌚ HandlingQCs2 - Delay

Name:

HandlingQCs2

☒ Show name ☐ Ignore

Type:

⌵

☒ Specified time
☐ Until stopDelay() is called

Delay time:

↺

Capacity:

⌵

Εικόνα 82 Χρόνοι αναμονής

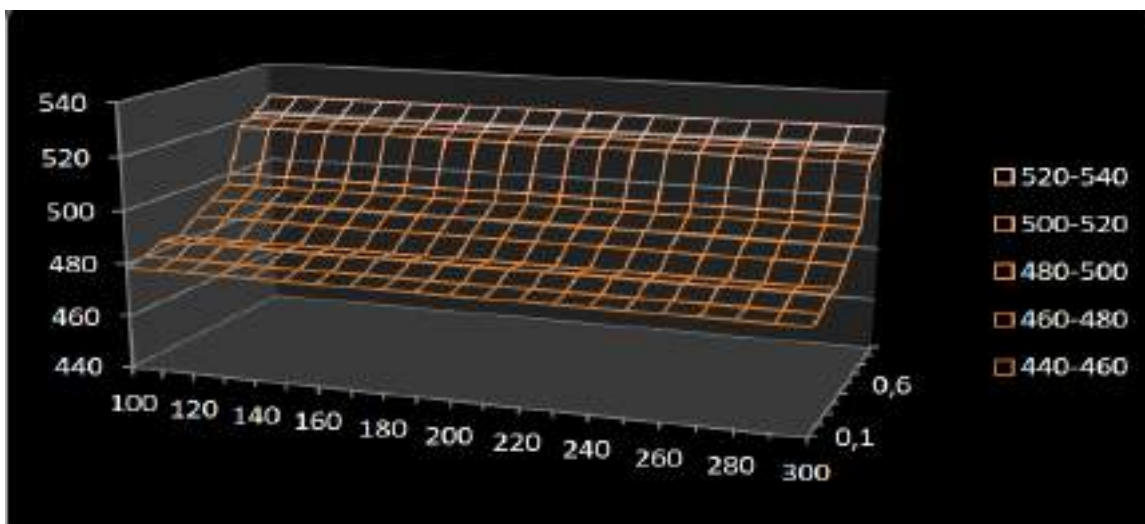
5.10 Συγκριτικά αποτελέσματα σεναρίων

	ΣΤΑΘΕΡΗ ΑΝΑΘΕΣΗ									
sigma/meters	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
100	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
110	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
120	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
130	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
140	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
150	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
160	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
170	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
180	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
190	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
200	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
210	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
220	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
230	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
240	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
250	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
260	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
270	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
280	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
290	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76
300	478,65	479,93	483,43	482,48	487,96	492,29	496,26	517,58	521,32	526,76

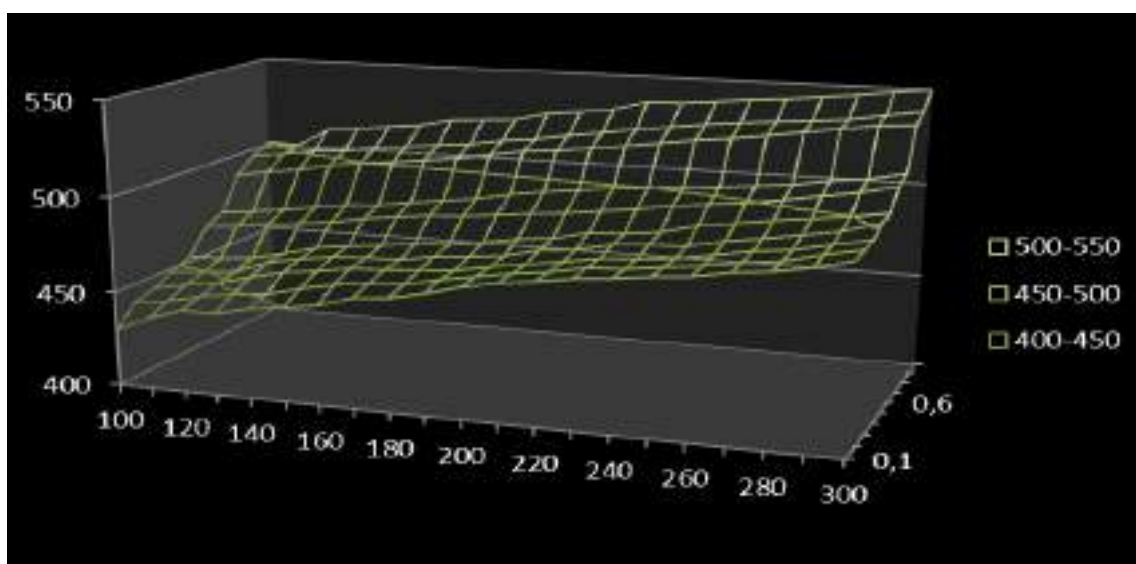
Πίνακας 18 Αποτελέσματα σεναρίου 2.3

	ΚΥΚΛΙΚΗ ΑΝΑΘΕΣΗ									
sigma/meters	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
100	430	440,32	443,56	448,32	452,85	465,87	470,33	488,49	496,42	501,44
110	437,87	442,32	446,96	449,65	453,23	467,35	472,21	491,32	498,45	500,69
120	443,58	446,56	449,35	451,32	454,95	469,67	474,34	493,65	501,35	510,48
130	443,45	449,23	454,08	457,98	458,63	471,65	476,98	497,35	502,46	512,35
140	447,52	450,86	458,23	461,23	463,23	473,49	479,65	499,79	505,16	514,46
150	449,78	455,23	460,35	463,78	465,56	476,89	480,63	501,23	507,16	516,34
160	452,43	458,78	463,13	464,78	467,32	479,16	483,35	503,49	508,94	520,19
170	457,89	461,12	463,89	465,23	469,65	481,19	485,67	505,67	510,56	521,31
180	459,36	462,16	465,25	468,14	470,32	482,16	488	508,64	513,73	522,49
190	463,22	465,89	466,4	469,56	472,98	484,16	491,56	511,79	515,48	526,84
200	468,07	468,98	469,78	471,35	476,23	486,19	494,19	513,14	518,64	528,64
210	471,66	472,12	472,6	475,81	477,99	487,56	496,16	515,49	521,13	531,26
220	473,92	474,29	475,23	477,35	482,13	489,97	498,64	517,94	524	536,15
230	476,73	477,84	478,46	481,36	483,16	492,16	499,65	520,49	526,16	536,49
240	478,35	478,65	479,65	482,65	485,19	495,12	501,52	523,19	528,64	538,1
250	480,63	480,98	481,23	483,23	489,91	497,32	502,79	525,16	530,46	540,15
260	482,23	483,12	484,32	487,23	492,35	499,16	504,61	527,64	532,16	541,32
270	485,65	486,12	487,65	489,78	495,16	501,89	506,31	530,49	533,49	543,25
280	487,56	488,35	489,6	492,98	497,37	505,23	509,64	533,18	535,16	545,15
290	490,87	492,78	493,65	496,23	499,73	508,16	511,49	535,82	538,19	547,62
300	495,56	496,47	497,12	501,45	502,79	512,35	515,49	537,16	541,2	549,56

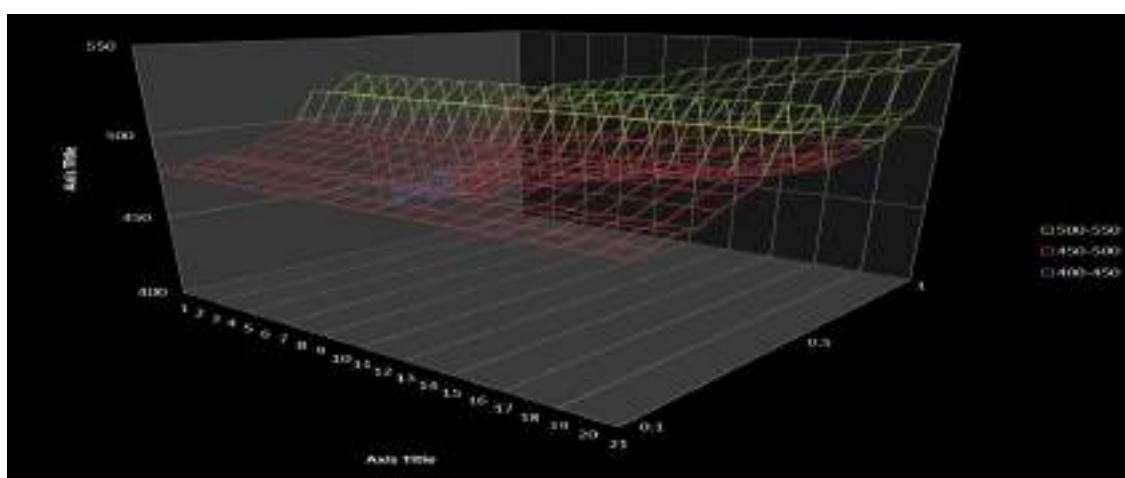
Πίνακας 19 Αποτελέσματα σεναρίου 2.4



Εικόνα 83 Διάγραμμα αποτελεσμάτων σεναρίου 2.3



Εικόνα 84 Διάγραμμα αποτελεσμάτων σεναρίου 2.4



Εικόνα 85 Συγκριτικό διάγραμμα αποτελεσμάτων σεναρίων 2.3, 2.4

6 ΠΡΟΣΟΜΟΙΩΣΗ ΕΠΙΛΙΜΕΝΙΟΥ ΣΙΔΗΡΟΔΡΟΜΙΚΟΥ ΣΤΑΘΜΟΥ ΠΕΙΡΑΙΑ

Σκοπός της δημιουργίας αυτού του μοντέλου είναι η δημιουργία ενός μοντέλου το οποίο θα μπορεί να αναπαριστά τις λειτουργίες του επιλιμένιου σιδηροδρομικού σταθμού στον Πειραιά. Παρέχοντας την δυνατότητα στον χρήστη να αρχικοποιεί διάφορους κομβικούς παραμέτρους, όπως ο αριθμός των μηχανημάτων, το καθιστά ένα διαδραστικό και εύκολα προσαρμόσιμο στις εκάστοτε ανάγκες και καταστάσεις περιβάλλον. Στόχος του μοντέλου αυτού είναι να δίνει στον χρήστη οπτικά και αριθμητικά αποτελέσματα για το περιβάλλον που μελετάει, κάτι που βοηθάει στην λήψη των απαραίτητων αποφάσεων που πρέπει να λάβει. Επίσης σε ερευνητικό επίπεδο δείχνει την αλληλεξάρτηση διάφορων παραμέτρων και πως αυτές επηρεάζουν την συνολική αποδοτικότητα ενός εμπορευματικού λιμένα. Οι λειτουργίες του λιμένα που έχουν μοντελοποιηθεί και αναλύονται παρακάτω, είναι:

1. **Θαλάσσια περιοχή:** αποτελείται από τον χώρο αναμονής των πλοίων μέχρι να υπάρξει διαθεσιμότητα στον χώρο παραβολής, το δίκτυο στο οποίο γίνεται η κίνηση των πλοίων και τέλος από τις μεταβλητές εισόδου.
2. **Περιοχή κρηπιδώματος (QA):** αποτελείται από τον χώρο παραβολής και τις γερανογέφυρες όπου γίνεται η φορτοεκφόρτωση των πλοίων.
3. **Χώρο μεταφοράς (TA):** αποτελείται από τα οχήματα μεταφοράς των εμπορευματοκιβωτίων εντός του λιμένα και του δικτύου μεταξύ της περιοχής κρηπιδώματος και του χώρου στοιβασίας, όπως και του χώρου στοιβασίας και της ενδοχώρας.
4. **Χώρο στοιβασίας (YA):** αποτελείται από τον χώρο εναπόθεσης των εμπορευματοκιβωτίων και τους γερανούς που τα στοιβάζουν
5. **Χώρο εισόδου και εξόδου των τρενών:** αποτελείται από τις σιδηροδρομικές γραμμές και τους αντίστοιχους γερανούς φορτοεκφόρτωσής τους. Επίσης περιλαμβάνει και τις μεταβλητές εισόδου των τρενών.
6. **Χώρο εισόδου και εξόδου των φορτηγών:** αποτελείται από τον χώρο αναμονής εκτός του λιμένα των φορτηγών, τις θέσεις εκφόρτωσης τους όπως και τις αντίστοιχες μεταβλητές εισόδου.

Για την ευκολότερη κατανόηση της δομής του μοντέλου άλλα και της ροής του, αρχικά θα γίνει μια γενική επισκόπηση επ' αυτού και στη συνέχεια θα παρουσιαστούν αναλυτικότερα οι λειτουργίες του, όπως έχουν παρατεθεί ανωτέρω. Τέλος, τα κρίσιμότερα και πιο πολύπλοκα σημεία των ιδιοτήτων (properties) των διαφόρων <<προγραμματιστικών μονάδων>>, όπως και ο κώδικας που χρησιμοποιήθηκε, παρατίθενται στο **ΠΑΡΑΡΤΗΜΑ Γ**.

6.1 Πρώτη εικόνα του μοντέλου και διαδραστικότητα

CT unloading trains trucks ships, loading ships

Run

Train Interarrival Rate: 10.0 min

Truck Interarrival Rate: 10.0 min

Ship Interarrival Rate: 10.0 min

Number Of Containers To Unload: 10

Number Of Containers To Load: 10

Number Of QCs: 10

Number Of YCs: 10

Number Of Train Cranes: 10

Number Of AGVs: 10

Number Of Trucks: 10

Speed Of QCs: 10 km/h

Speed Of YCs: 10 km/h

Speed Of Train Cranes: 10 km/h

Speed Of AGVs: 10 km/h

Speed Of Trucks: 10 km/h

Time for Train Crane to wrap container from Train: 10.0

Time for Truck to wrap container from Train Crane: 10.0

Time for YCs to wrap container from Truck: 10.0

Time for AGVs to wrap container from YCs: 10.0

Time for QCs to wrap container from AGVs: 10.0

Time for AGVs to wrap container from QCs: 10.0

Time for QC To Put Container into Ship: 10.0

Time for QC to wrap container from Ship: 10.0

Speed Of Moving Container By Train Crane: 10.0 m/s

Speed Of Moving Container By YC: 10.0 m/s

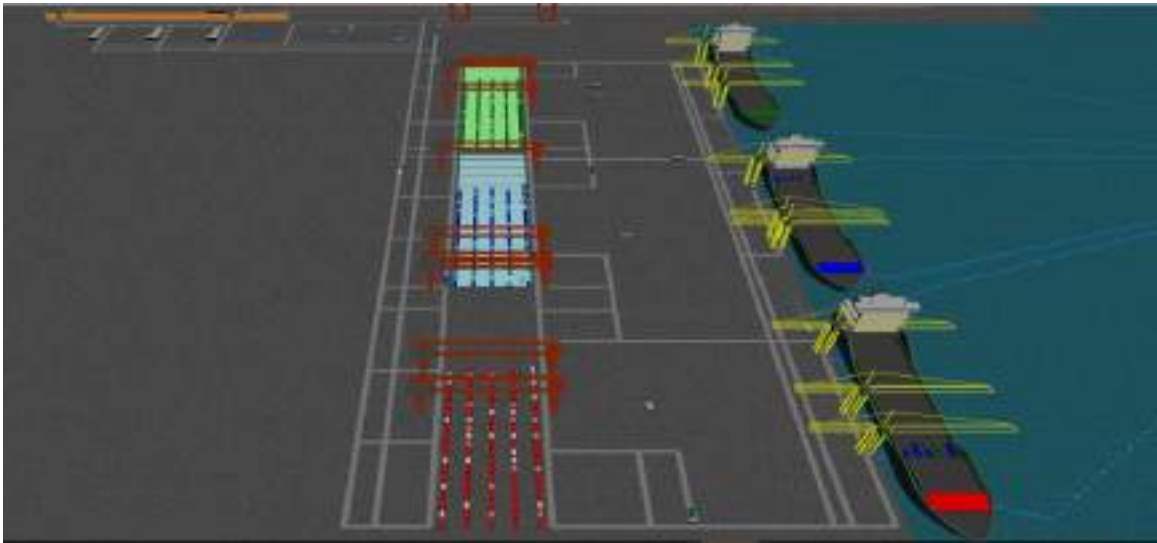
Speed Of Moving Container By QC: 10.0 m/s

Εικόνα 86 Πρώτη εικόνα

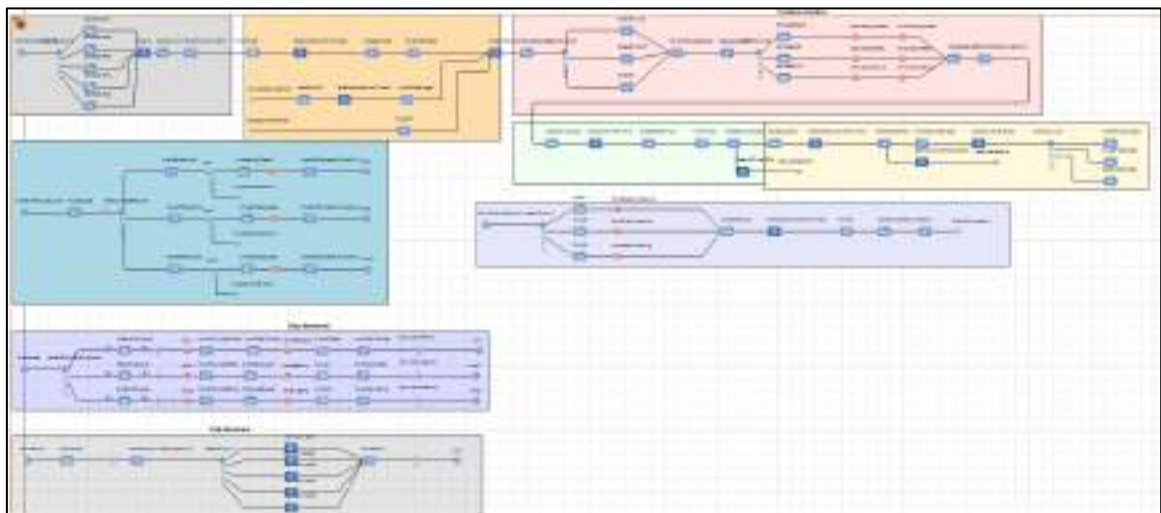
Όπως φαίνεται και στην Εικόνα 86 Πρώτη εικόνα χρήστης έχει την δυνατότητα να ορίσει εκείνος τις μεταβλητές εισόδου του λυμένα. Αυτές είναι:

- Οι χρόνοι άφιξης των τρενών, φορτηγών και πλοίων
- τον αριθμό των εμπορευματοκιβωτίων που θα ξεφορτώσει το πλοίο αλλά και θα φορτώσει
- Τον αριθμό των διαφόρων μηχανημάτων
- Τις ταχύτητες κίνησης των διαφόρων μηχανημάτων
- Τους χρόνους διαχείρισης των εμπορευματοκιβωτίων από τα διάφορα μηχανήματα
- Τις ταχύτητες κίνησης των εμπορευματοκιβωτίων ενώ τα έχουν παραλάβει οι γερανοί

6.2 Γενική επισκόπηση λιμένα



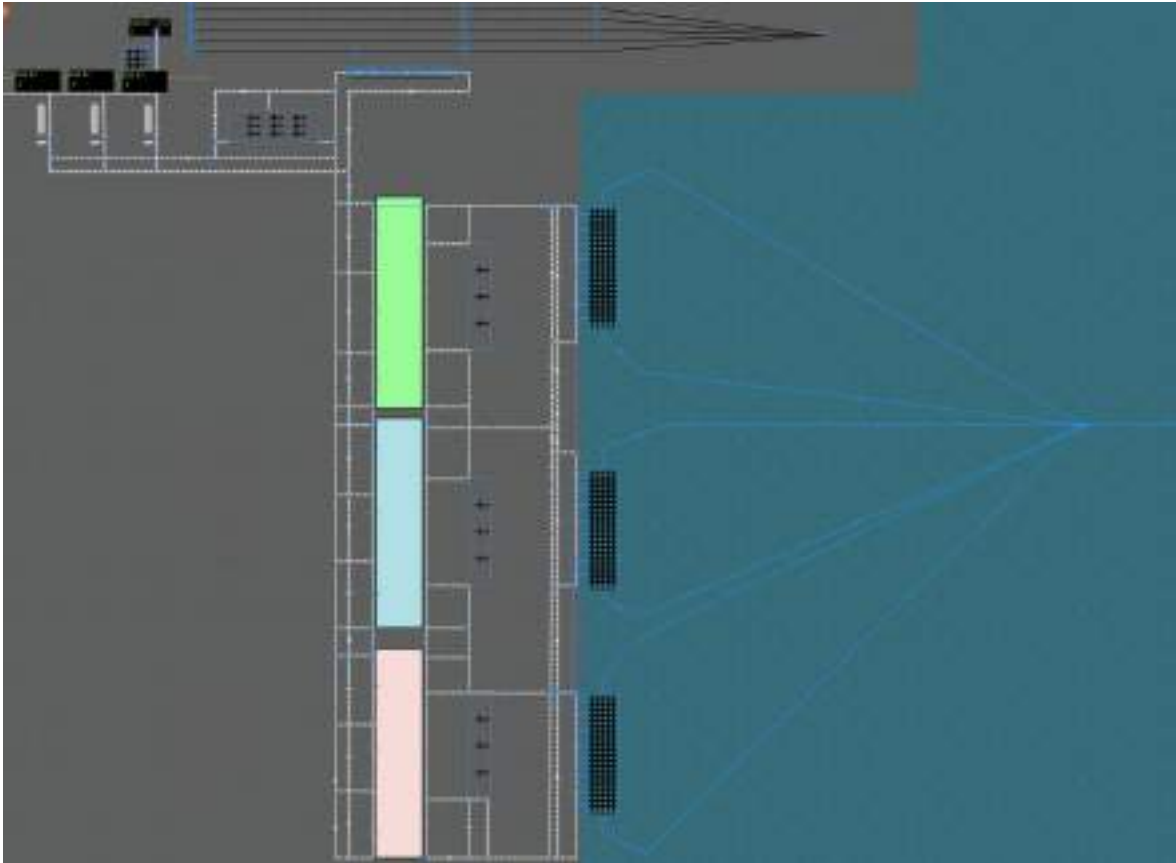
Εικόνα 87 Γενική επισκόπηση λιμένα 3D



Εικόνα 88 Γενική επισκόπηση λιμένα /διάγραμμα ροής

1. Στην εικόνα Εικόνα 88 με μπλε χρώμα (κάτω αριστερά) φαίνεται η λειτουργία των πλοίων μεταφοράς.
2. Στην Εικόνα 88 με μοβ χρώμα (στα δεξιά) φαίνεται η λειτουργία των παραθαλάσσιων γερανών κατά την διαδικασία εκφόρτωσης των πλοίων .
3. Στην εικόνα Εικόνα 88 με κίτρινο χρώμα φαίνεται η λειτουργία των παραθαλάσσιων γερανών κατά την διαδικασία της φόρτωσης των πλοίων
4. Στην εικόνα Εικόνα 88 με πράσινο χρώμα φαίνεται η λειτουργία των οχημάτων μεταφοράς μεταξύ του χώρου εναπόθεσης και του χώρου παραβολής των πλοίων (AGVs)
5. Στην εικόνα Εικόνα 88 με κόκκινο χρώμα οι λειτουργίες των γερανογεφυρών που απασχολούνται στον χώρο εναπόθεσης
6. Στην εικόνα Εικόνα 88 με γκρι χρώμα (κάτω αριστερά) φαίνεται η λειτουργία των εμπορικών τρενών.

7. Στην εικόνα Εικόνα 88 Εικόνα 88 Γενική επισκόπηση λιμένα /διάγραμμα ροής με γκρι χρώμα (πάνω αριστερά) φαίνεται η λειτουργία των γερανών που απασχολούνται στο τραίνο.
8. Στην Εικόνα 88 με ανοιχτό μπλε χρώμα φαίνεται η λειτουργία των εξωτερικών φορτηγών.
9. Στην Εικόνα 88 με πορτοκαλί χρώμα φαίνεται η λειτουργία των φορτηγών εντός του λιμένα



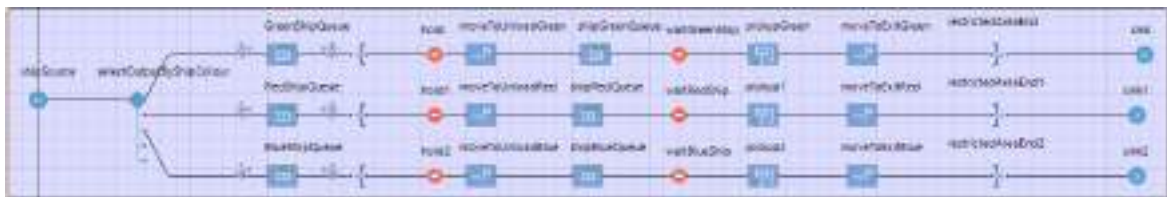
Εικόνα 89 Κάτοψη λιμένα

Αντίστοιχα οι λειτουργίες αυτές χωρικά βρίσκονται σε:

1. **Θαλάσσια περιοχή:** βρίσκεται στα δεξιά της Εικόνα 89 Κάτοψη λιμένα μπλε χρώμα. Τα κουτιά που φαίνονται στα δεξιά αποτελούν τον χώρο αναμονής των πλοίων. Οι γαλάζιες γραμμές δείχνουν τις διάφορες πορείες που μπορούν να χρησιμοποιήσουν τα πλοία. Τέλος τα μαύρα κουτιά εντός της μπλε ζώνης (θάλασσας) αποτελούν τον χώρο παραβολής των πλοίων
2. **Περιοχή κρηπιδώματος (QA):** περιλαμβάνει τον χώρο παραβολής και την θαλάσσια πλευρά του λιμένα όπου βρίσκονται οι γερανοί
3. **Χώρο μεταφοράς (TA):** περιλαμβάνει το δίκτυο των δρόμων εντός του λιμένα όπου απεικονίζονται με λευκές γραμμές (δρόμους)
4. **Χώρο στοιβασίας (YA):** περιλαμβάνει το ορθογώνιο παραλληλόγραμμο αριστερά από τους χώρους παραβολής

5. **Χώρο εισόδου και εξόδου των τραϊνών** : περιλαμβάνει τις σιδηροδρομικές γραμμές στο πάνω μέρος του λιμένα, όπως και τον χώρο ακριβώς από κάτω τους όπου βρίσκονται οι γερανοί
6. **Χώρο εισόδου και εξόδου των φορτηγών**: περιλαμβάνει τον χώρο πάνω αριστερά όπου τα φορτηγά αναμένουν να αδειάσει κάποιος χώρος εκφόρτωσης, ο οποίος βρίσκεται ακριβώς από κάτω.

6.3 Θαλάσσια περιοχή

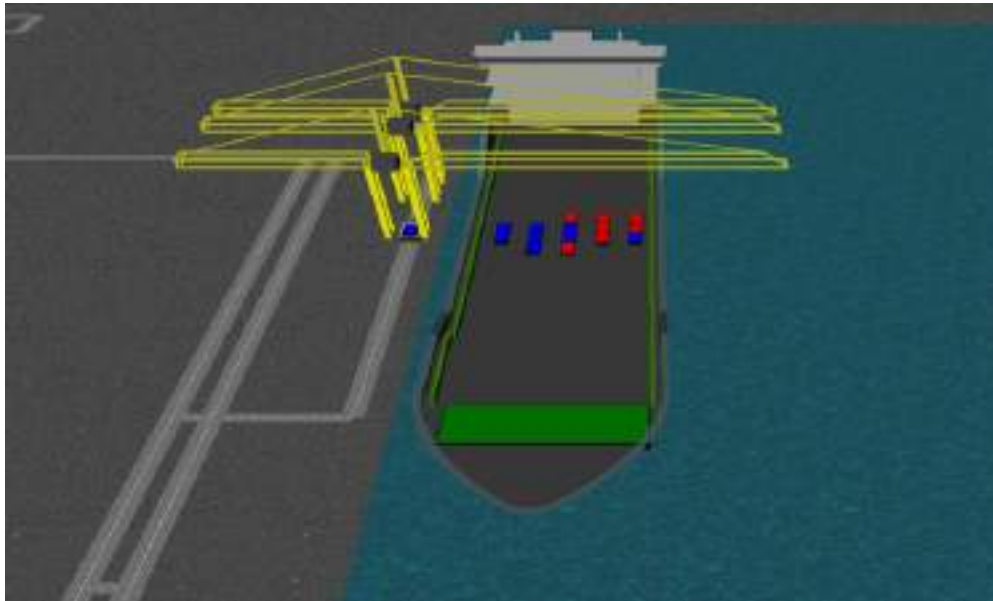


Εικόνα 90 Θαλάσσια περιοχή διάγραμμα ροής

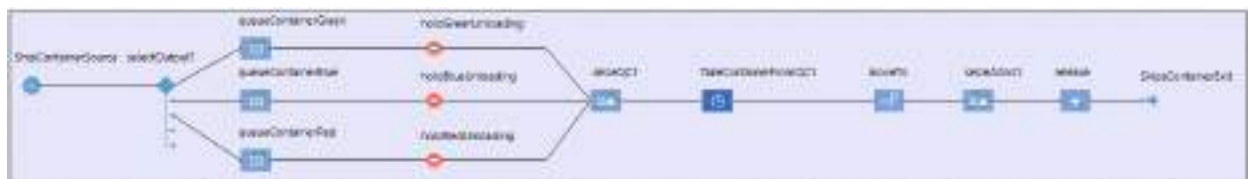
1. Αρχικά με το **shipSource** δίνεται η εντολή εισόδου των πλοίων στο μοντέλο και καθορίζεται και ο ρυθμός εισαγωγής τους.
2. Στην συνέχεια ανάλογα με το χρώμα του πλοίου, κατευθύνεται ανάλογα στο χώρο αναμονής (**GreenShipQueue**, **RedShipQueue**, **BlueShipQueue**)
3. Στην συνέχεια με τα **hold** καθορίζεται το αν το πλοίο θα περιμένει στην θέση αναμονής ή όχι ανάλογα με την διαθεσιμότητα της θέσης παραβολής.
4. Αν υπάρχει διαθεσιμότητα, με τα **moveTo**, κατευθύνεται το πλοίο στον χώρο παραβολής (**shipGreenQueue**, **shipRedQueue**, **shipBlueQueue**)
5. Με το που μπει το πλοίο στον χώρο παραβολής, δίνεται εντολή για την εισαγωγή των εμπορευματοκιβωτίων στο μοντέλο, και πιο συγκεκριμένα σε μια ουρά που βρίσκεται στην περιοχή κρηπιδώματος (**queueContainerGreen**, **queueContainerBlue**, **queueContainerRed**).
6. Ακολουθούν οι διαδικασίες φορτοεκφόρτωσης, οι οποίες γίνονται σε άλλες περιοχές, και μόλις ολοκληρωθούν τα **waitShip** απελευθερώνουν το πλοίο προκειμένου να παραλάβει τα εμπορευματοκιβώτια (**pickup**) και να φύγει από το σύστημα (**moveTo & sink**)

6.4 Περιοχής κρηπιδώματος

6.4.1 Διαδικασία εκφόρτωσης πλοίων



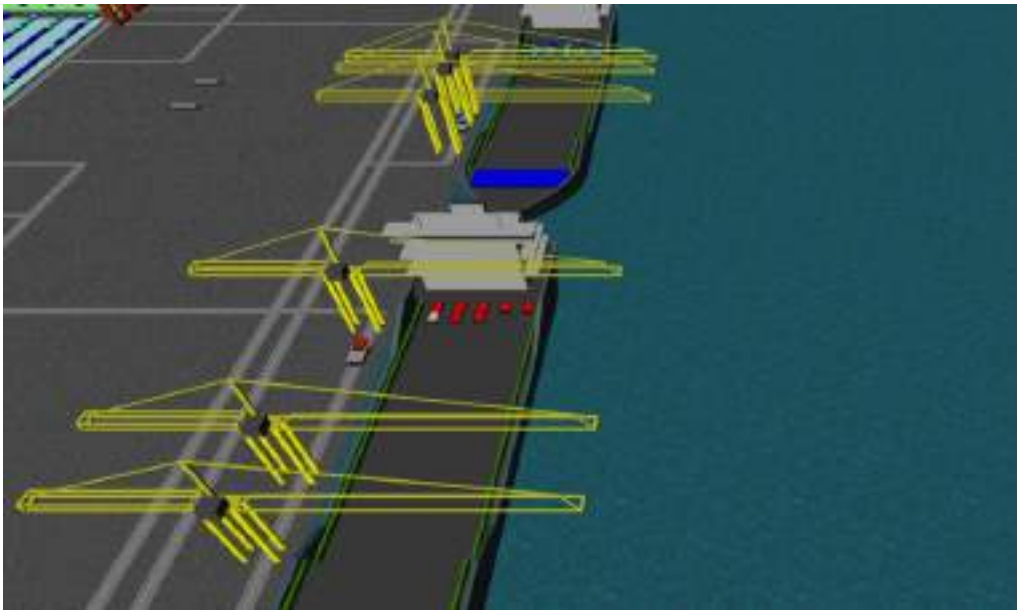
Εικόνα 91 Διαδικασία εκφόρτωσης πλοίων 3D



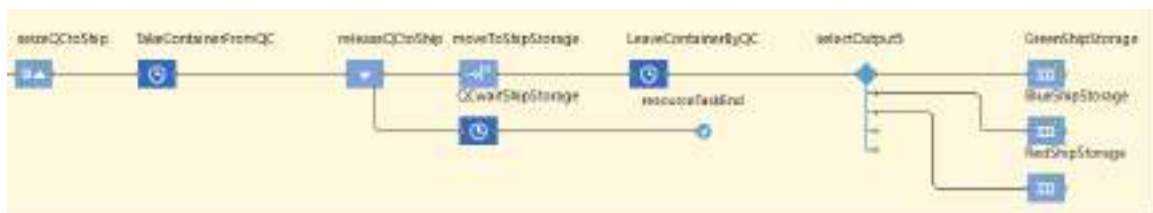
Εικόνα 92 Διαδικασία εκφόρτωσης πλοίων διάγραμμα ροής

1. Με το **shipContainerSource** εισάγονται τα εμπορευματοκιβώτια, που είναι φορτωμένα στο πλοίο, στο σύστημα.
2. Στην συνέχεια ανάλογα το πιο πλοίο παραβάλει, κατευθύνονται στην αντίστοιχη ουρά, όπου και αναμένουν
3. Στην συνέχεια τα αναλαμβάνουν οι γερανοί κρηπιδώματος (**seizeQC1**), όπου τα μεταφέρουν στα οχήματα μεταφοράς (**moveTo**) τα οποία τα παραλαμβάνουν (**seizeAGV1**) ενώ με το (**release**) απελευθερώνεται το εμπορευματοκιβώτιο από τον γερανό.
4. Τέλος με το (**ShipContainerExit**) τα εμπορευματοκιβώτια οδηγούνται στον χώρο μεταφοράς, και πιο συγκεκριμένα στο (**shipsContainerEnter**)

6.4.2 Διαδικασία φόρτωσης πλοίων



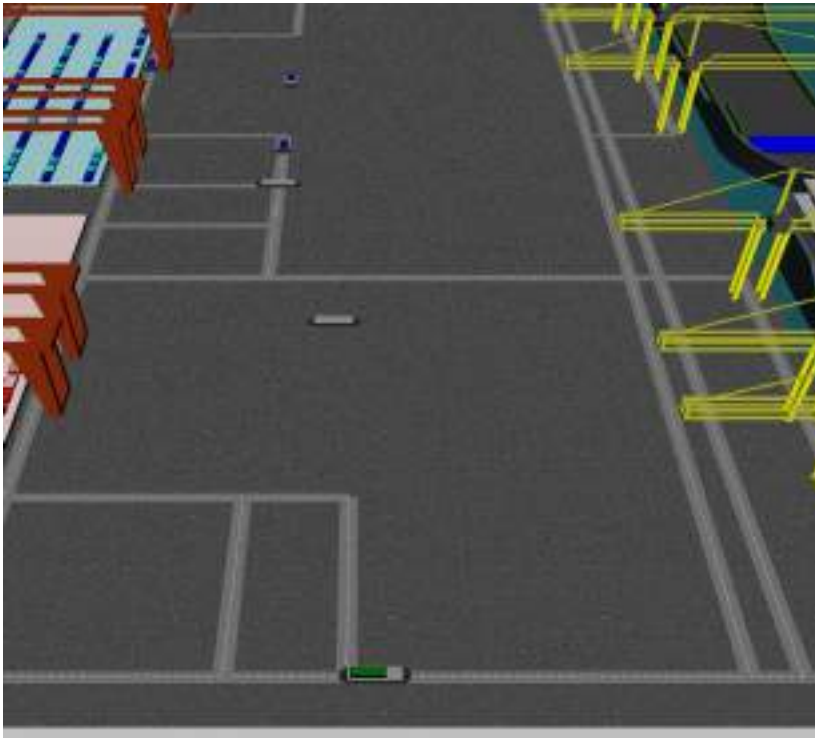
Εικόνα 93 Διαδικασία φόρτωσης πλοίων 3D



Εικόνα 94 Διαδικασία φόρτωσης πλοίων διάγραμμα ροής

1. Αρχικά με το **(seizeQCtoShip)** ο γερανός αναλαμβάνει το εμπορευματοκιβώτιο που πρόκειται να φορτωθεί στο πλοίο
2. Το **(TakeContainerFromQC)** αποτελεί μια καθυστέρηση η οποία προσομοιώνει τον χρόνο που απαιτείται προκειμένου να γίνει η μεταφορά του εμπορευματοκιβωτίου από το όχημα μεταφοράς στον γερανό.
3. Στην συνέχεια με το **(releaseQCtoShip)** απελευθερώνεται το εμπορευματοκιβώτιο προκειμένου να μεταφερθεί **(moveToShipStorage)** στο χώρο εναπόθεσης εντός του πλοίου (**GreenShipStorage**, **BlueShipStorage**, **RedShipStorage**), ενώ λόγω του ότι ο γερανός δεν μετακινείται κατά την διαδικασία αυτή συνεχίζει την δεύτερη (κάτω) πορεία, όπου και αναμένει **(QCwaitShipStorage)** μέχρις ότου το εμπορευματοκιβώτιο τοποθετηθεί στο πλοίο.

6.5 Ζώνη κυκλοφορίας μεταξύ κρηπιδώματος και χώρων στοιβασίας



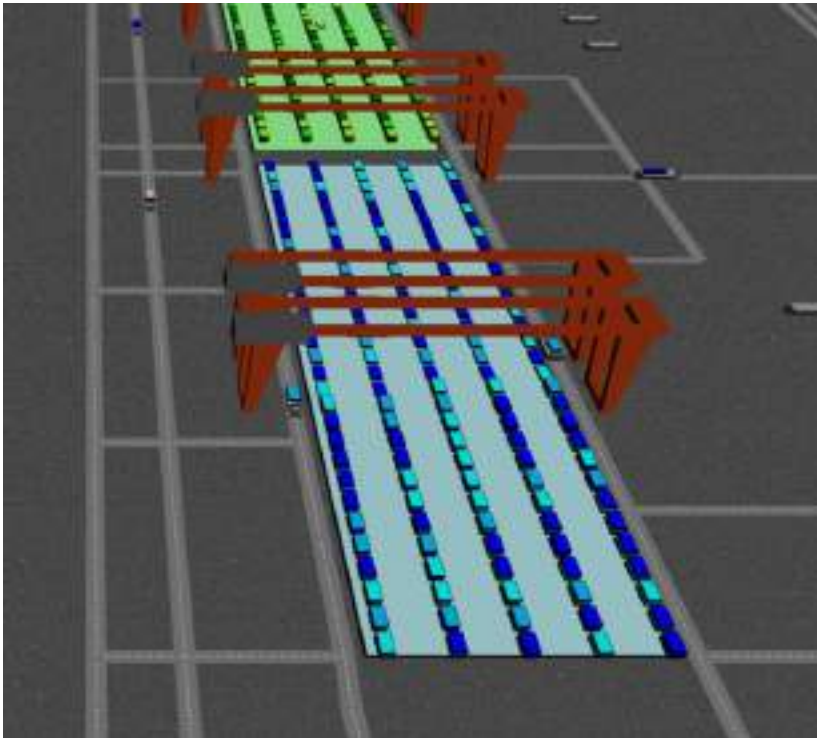
Εικόνα 95 Διαδικασία φόρτωσης πλοίων (Ζώνη κυκλοφορίας μεταξύ κρηπιδώματος και χώρων στοιβασίας) 3D



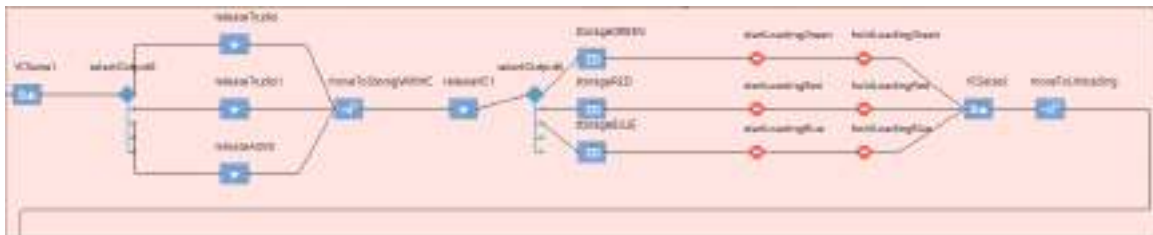
Εικόνα 96 Διαδικασία φόρτωσης πλοίων διάγραμμα ροής

1. Αρχικά με το **seizeAGVsToQCs** τα οχήματα μεταφοράς παραλαμβάνουν τα εμπορευματοκιβώτια. Ο χρόνος που απαιτείται προκειμένου να γίνει η παραλαβή προσομοιώνεται με το **takeContainerFromYC**.
2. Το **releaseYC2** αναφέρεται στην απελευθέρωση του γερανού εφόσον το έχει παραλάβει το AGV.
3. Στην συνέχεια με το **moveToQC** οδηγείται το όχημα μεταφοράς στον γερανό ο οποίος θα παραλάβει το εμπορευματοκιβώτιο.
4. Τέλος με το **releaseAGVsToQCs** απελευθερώνεται το εμπορευματοκιβώτιο, ενώ το AGV συνεχίζει στο **AGVwaitQCs** έως ότου ο γερανός παραλάβει το εμπορευματοκιβώτιο

6.6 Χώρος στοιβασίας εμπορευματοκιβωτίων



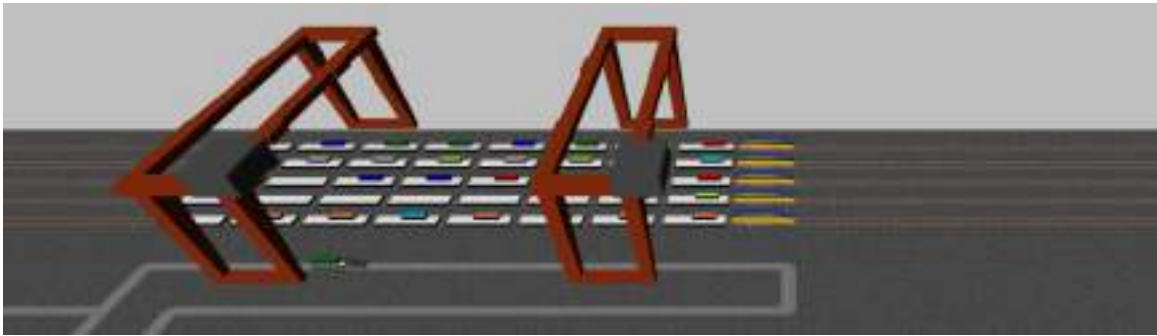
Εικόνα 97 Χώρος στοιβασίας 3D



Εικόνα 98 Χώρος στοιβασίας διάγραμμα ροής

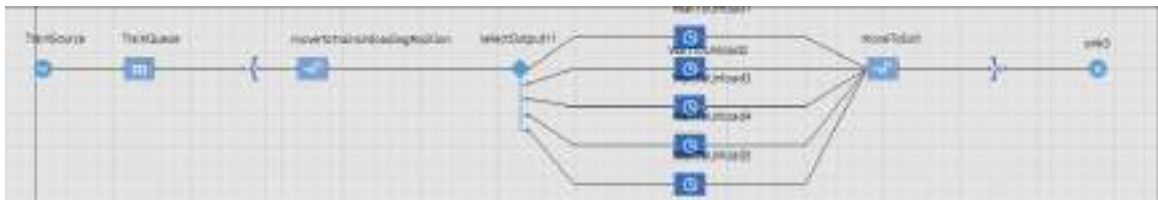
1. Αρχικά, με το **YCSeize1** οι γερανοί που απασχολούνται στον χώρο στοιβασίας, παραλαμβάνουν το εμπορευματοκιβώτιο.
2. Στην συνέχεια με το **selectOutput9** ανάλογα από πού προέρχεται (τραίνο, φορτηγό ή πλοίο), απελευθερώνεται και το αντίστοιχο όχημα μεταφοράς.
3. Ακολούθως με το **moveToStorageWithYC** μεταφέρεται το εμπορευματοκιβώτιο στην αντίστοιχη θέση που έχει προκαθοριστεί και στην συνέχεια απελευθερώνεται ο γερανός.
4. Με το **selectOutput6** γίνεται καταμερισμός των εμπορευματοκιβωτίων ανάλογα με το χρώμα τους στις διάφορες αποθήκες.
5. Μετέπειτα η ύπαρξη του **startLoading** σηματοδοτεί το γεγονός ότι υπάρχει πλοίο στην παραβολή, ενώ το **holdLoadingGreen** σηματοδοτεί το γεγονός ότι το πλοίο έχει ξεφορτώσει πλήρως. Ένα είναι αληθείς και τα δύο η διαδικασία φόρτωσης του πλοίου ξεκινάει.
6. Τέλος, ο γερανός αναλαμβάνει με το **YCSeize2** προκειμένου να γίνει η μεταφορά του εμπορευματοκιβώτιου (**moveToUnloading**) στον χώρο όπου θα το παραλάβει το όχημα μεταφοράς

6.7 Επιλιμένος σιδηροδρομικός σταθμός



Εικόνα 99 Επιλιμένος σιδηροδρομικός σταθμός 3D

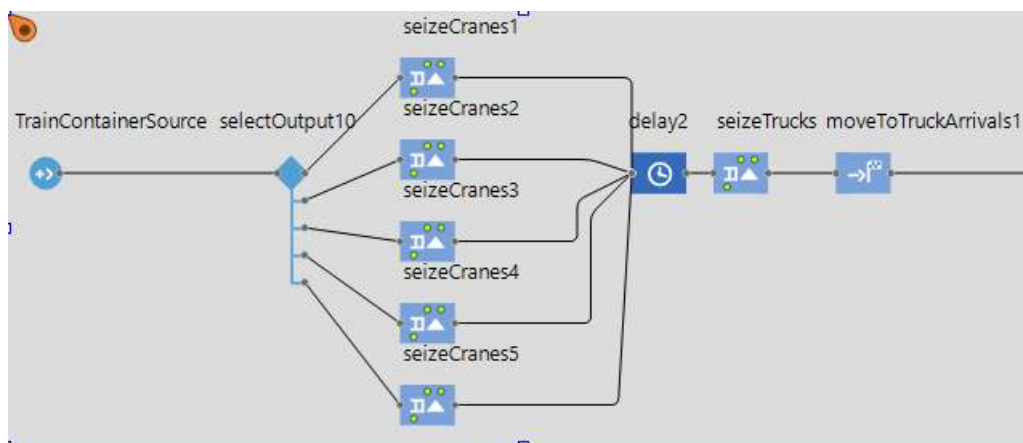
6.7.1 Η λειτουργία των εμπορευματικών τρένων



Εικόνα 100 Η λειτουργία των εμπορικών τρένων διάγραμμα ροής

1. Με την χρήση του **TrainSource**¹ γίνεται η εισαγωγή των τρένων στο σύστημα, ενώ το **TrainQueue** αποτελεί ένα χώρο αναμονής εκτός του λιμένα, στην περίπτωση που και οι πέντε θέσεις στάθμευσης των τρένων είναι κατειλημμένες.
2. Στην συνέχεια με το **moveToTrainUnloadingPosition** τα τρένα κατευθύνονται στην πρώτη διαθέσιμη θέση στάθμευσης.
3. Με το **selectOutput11** κατευθύνεται το τρένο στο αντίστοιχο **waitToUnload** μέχρις ότου ξεφορτώσει πλήρως.
4. Τέλος, όταν ξεφορτώσει κατευθύνεται προς την έξοδο με το **moveToExit**.

6.7.2 Η λειτουργία των γερανογεφυρών που εξυπηρετούν τα τρένα

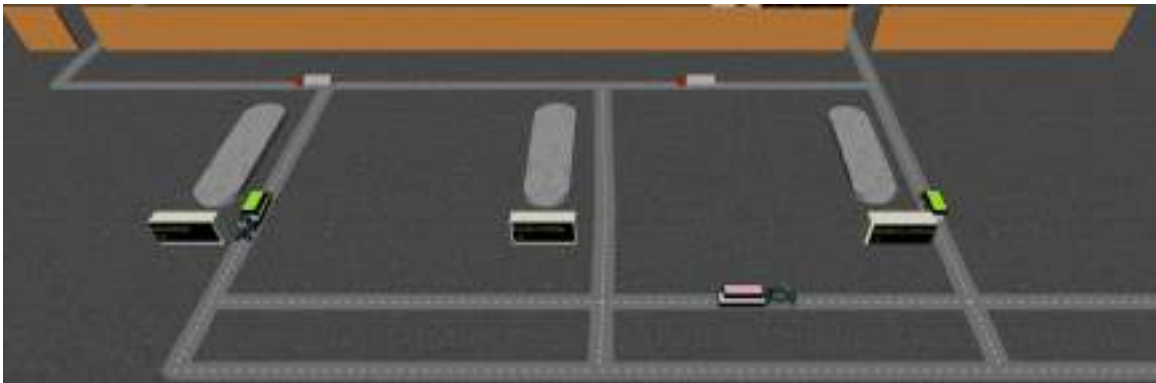


Εικόνα 101 Η λειτουργία των γερανογεφυρών που απασχολούνται στα τρένα διάγραμμα ροής

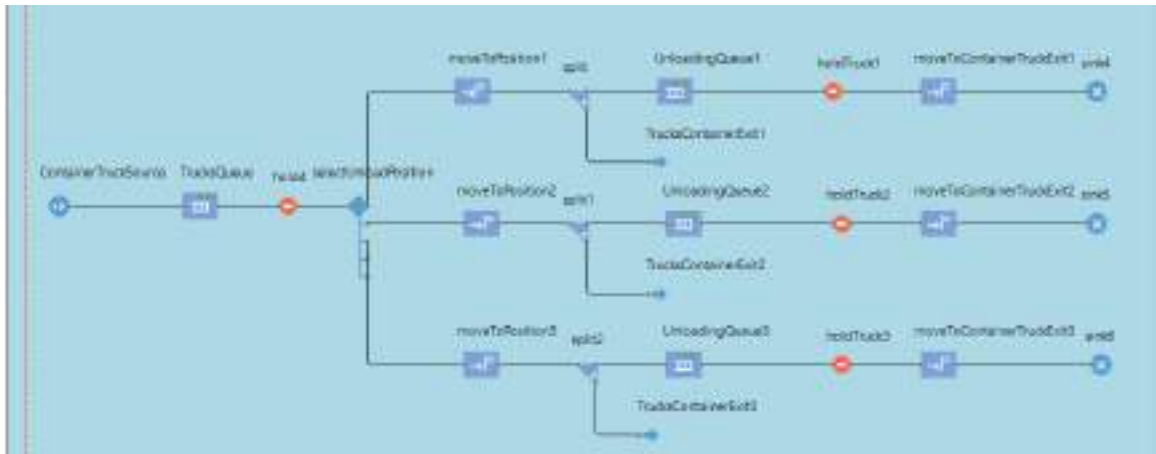
1. Αρχικά, με την χρήση του **TrainContainerSource** γίνεται η εισαγωγή των εμπορευματοκιβωτίων που μεταφέρουν τα τραίνα στο σύστημα.
2. Με το **selectOutput10** τα εμπορευματοκιβώτια κατευθύνονται στο αντίστοιχο **seizeCranes** προκειμένου να τα αναλάβουν οι γερανοί.
3. Με το **delay2** προσομοιώνεται ο χρόνος που απαιτείται για να γίνει η παραλαβή αυτή.
4. Με το που γίνει η προηγούμενη διαδικασία, καλούνται τα φορτηγά να κατευθυνθούν στον χώρο όπου πρόκειται να παραλάβουν το εμπορευματοκιβώτιο.
5. Τέλος με το **moveToTruckArrivals1**, τα εμπορευματοκιβώτια κατευθύνονται στον χώρο παραλαβής τους.

6.8 Πύλες εισόδου- εξόδου και χώροι αναμονής των φορτηγών

6.8.1 Η λειτουργία των εξωτερικών φορτηγών



Εικόνα 102 Η λειτουργία των εξωτερικών φορτηγών



Εικόνα 103 Η λειτουργία των εξωτερικών φορτηγών διάγραμμα ροής

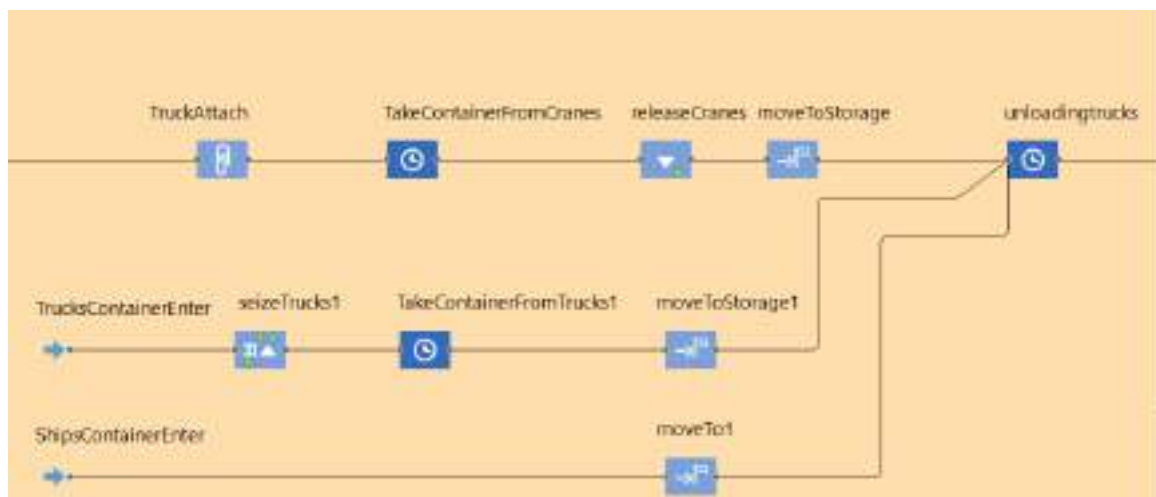
1. Αρχικά, με το **ContainerTruckSource** γίνεται η εισαγωγή των φορτηγών στο σύστημα.
2. Στην συνέχεια με την χρήση του **TrucksQueue** δημιουργήθηκε ένας χώρος αναμονής εκτός του λιμένα για την περίπτωση που και οι τρεις θέσεις εκφόρτωσης των φορτηγών είναι καταλυμένες. Για τον έλεγχο αυτό υπεύθυνο είναι το **hold4**.

3. Με τη χρήση του **selectUnloadPosition** κατευθύνονται τα φορτηγά στο πρώτο διαθέσιμο σημείο εκφόρτωσης φορτηγών, ενώ με το αντίστοιχο **moveToPosition** δίνεται η εντολή μετακίνησής τους.
4. Ακολούθως, με την χρήση του **split**, εισάγεται στο σύστημα το εμπορευματοκιβώτιο που είναι φορτωμένο στο εκάστοτε φορτηγό, ενώ με το **TrucksContainerExit** στέλνεται στην διαδικασία όπου τα εσωτερικά πλέον φορτηγά το κατευθύνουν στον χώρο στοιβασίας.
5. Αμέσως μόλις σταλεί σε αυτή την λειτουργία το εμπορευματοκιβώτιο, μέχρι να γίνει η παραλαβή του, το εξωτερικό φορτηγό μπαίνει σε μια ουρά αναμονής.
6. Τέλος, μόλις ολοκληρωθεί η διαδικασία εκφόρτωσης, το **holdTruck** απελευθερώνει το φορτηγό, το οποίο μέσω του **moveToContainerTruckExit** βγαίνει ξανά από τον λιμένα και το σύστημα.

6.9 Η λειτουργία των ελκυστήρων εντός του λιμένα



Εικόνα 104 Η λειτουργία των ελκυστήρων εντός του λιμένα 3D



Εικόνα 105 Η λειτουργία των ελκυστήρων εντός του λιμένα διάγραμμα ροής

Α γραμμή:

1. Τα εμπορευματοκιβώτια προέρχονται από το τραίνο, όπου κατά την διαδικασία την παραλαβής του από τους γερανούς καλέστηκε να πάει το φορτηγό στον χώρο

παραλαβής. Σε αυτή τη φάση, με το **truckAttach** το φορτηγό παραλαμβάνει το εμπορευματοκιβώτιο.

2. Στην συνέχεια, με το **TakeContainerFromCranes**, προσομοιώνεται ο χρόνος ο οποίος απαιτείται προκειμένου να τοποθετηθεί σωστά πάνω στο φορτηγό.
3. Ακολούθως, απελευθερώνονται οι γερανοί με το **releaseCranes**, και πλέον τα φορτωμένα φορτηγά οδηγούνται στην θέση όπου θα ξεφορτώσουν στον χώρο στοιβασίας (**move to Storage**).
4. Τέλος, το **unloadingtrucks** προσομοιώνει τον χρόνο αναμονής προκειμένου να σταθμεύσει το φορτηγό στην ακριβή θέση και να έρθει ο γερανός να παραλάβει το εμπορευματοκιβώτιο.

Β γραμμή:

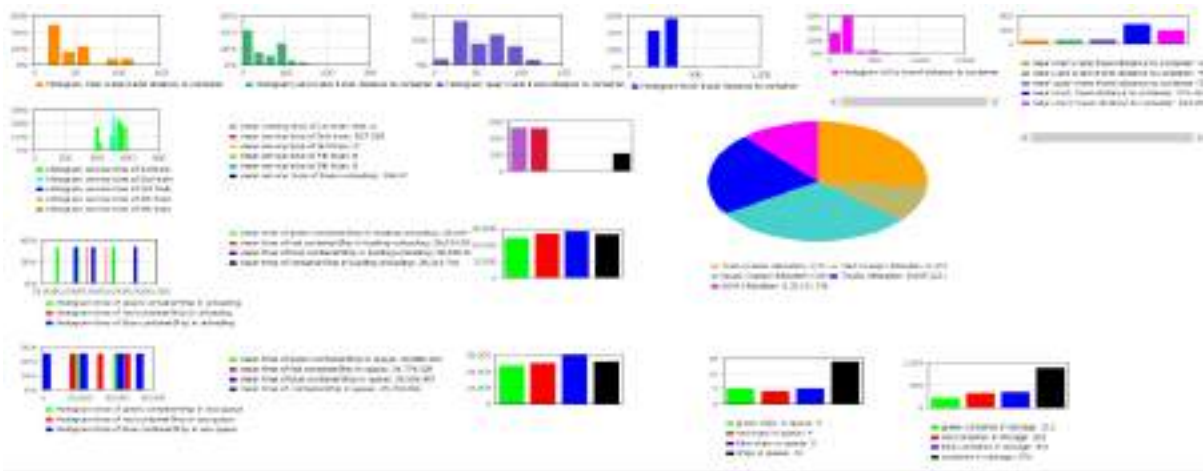
1. Τα εμπορευματοκιβώτια προέρχονται από τα εξωτερικά φορτηγά. Η είσοδος τους σε αυτή την λειτουργία γίνεται με το **TrucksContainerEnter**.
2. Στην συνέχεια καλούνται τα εσωτερικά φορτηγά να παραλάβουν το εμπορευματοκιβώτιο με την χρήση του **seizeTrucks1**, ενώ με το **TakeContainerFromTrucks1** προσομοιώνεται ο χρόνος που απαιτείται προκειμένου να γίνει η παραλαβή αυτή.
3. Πλέον τα φορτωμένα φορτηγά οδηγούνται στην θέση όπου θα ξεφορτώσουν στον χώρο στοιβασίας (**move to Storage1**)
4. Τέλος, το **unloadingtrucks** προσομοιώνει τον χρόνο αναμονής προκειμένου να σταθμεύσει το φορτηγό στην ακριβή θέση και να έρθει ο γερανός να παραλάβει το εμπορευματοκιβώτιο.

Γ γραμμή:

1. Τα εμπορευματοκιβώτια προέρχονται από τα πλοία. Η είσοδος τους σε αυτή την λειτουργία γίνεται με το **ShipsContainerEnter**.
2. Με την χρήση του **moveTo1** τα φορτηγά μετακινούνται προς τον χώρο στοιβασίας.
5. Τέλος, το **unloadingtrucks** προσομοιώνει τον χρόνο αναμονής προκειμένου να σταθμεύσει το φορτηγό στην ακριβή θέση και να έρθει ο γερανός να παραλάβει το εμπορευματοκιβώτιο.

6.10 Αποτελέσματα μοντέλου

Κατά την διαδικασία που τρέχει το μοντέλο, παρέχεται στον χρήστη οπτικοποίηση διαφόρων δεδομένων. Αυτά φαίνονται όπως στην Εικόνα 106 Αποτελέσματα μοντέλου:



Εικόνα 106 Αποτελέσματα μοντέλου

Τα αποτελέσματα αυτά είναι:

- Ιστογράμμο απόστασης εμπορευματοκιβωτίου από τον γερανό επί του τραίνου
- Ιστογράμμο απόστασης εμπορευματοκιβωτίου από τον ΥC
- Ιστογράμμο απόστασης εμπορευματοκιβωτίου από τον γερανό κρηπιδώματος (QC)
- Ιστογράμμο απόστασης εμπορευματοκιβωτίου από το φορτηγό
- Ιστογράμμο απόστασης εμπορευματοκιβωτίου από το AGV
- Ραβδόγραμμα μέσης απόστασης του εκάστοτε μηχανήματος από το εμπορευματοκιβώτιο
- Ιστογράμμο του χρόνου εξυπηρέτησης της κάθε θέσης του τραίνου
- Ραβδόγραμμα του μέσου χρόνου εξυπηρέτησης της κάθε θέσης του τραίνου
- Ιστογράμμο του χρόνου εξυπηρέτησης της κάθε θέσης παραβολής
- Ραβδόγραμμα του μέσου χρόνου εξυπηρέτησης της κάθε θέσης παραβολής
- Ιστογράμμο του χρόνου αναμονής έξω εκτός του λιμένα για κάθε θέσης παραβολής
- Ραβδόγραμμα του χρόνου αναμονής εκτός του λιμένα για κάθε θέσης παραβολής
- Ραβδόγραμμα αριθμού πλοίων εκτός του λιμένα για κάθε θέσης παραβολής
- Ραβδόγραμμα αριθμού αποθηκευμένων εμπορευματοκιβωτίων στην στοιβασία
- Πίτα που παρουσιάζει την χρησιμοποίηση του εκάστοτε μηχανήματος

6.11 Διατύπωση και επίλυση σεναρίων φορτοεκφόρτωσης πλοίων

Προκειμένου να βελτιστοποιηθεί η λειτουργία του λιμανιού, ως προς τον χρόνο εξυπηρέτησης των πλοίων, έχουν γίνει ορισμένα σενάρια, τα οποία περιγράφονται λεπτομερέστερα παρακάτω, όπου σταδιακά καταλήγουν στην βέλτιστο αριθμό μηχανημάτων που πρέπει να έχει ο λιμένας αλλά και στον τρόπο λειτουργίας των ΥCs. Τα σενάρια αυτά έχουν σταθερές τις μεταβλητές εισόδου όπως και τις επιμέρους ταχύτητες και χρόνους των μηχανημάτων.

Τα σταθερά αυτά στοιχεία είναι:

- χρόνος μεταξύ των αφίξεων των τραίνων : 10min
- χρόνος μεταξύ των αφίξεων των φορτηγών : 20min
- χρόνος μεταξύ των αφίξεων των πλοίων : 60min

- αριθμός εμπορευματοκιβωτίων που θα ξεφορτωθούν από το πλοίο :50
- αριθμός εμπορευματοκιβωτίων που θα φορτωθούν από το πλοίο :80
- ταχύτητα κίνησης των QCs : 2 km/h
- ταχύτητα κίνησης των YCs : 2 km/h
- ταχύτητα κίνησης των γερανών επί των τρενών: 2 km/h
- ταχύτητα κίνησης των AGVs : 15 km/h
- ταχύτητα κίνησης των φορτηγών : 15 km/h
- χρόνος για να πιάσει ο γερανός επί του τρενού το εμπορευματοκιβώτιο :30sec
- χρόνος για να παραλάβει το φορτηγό το εμπορευματοκιβώτιο :30sec
- χρόνος για να παραλάβει ο YC το εμπορευματοκιβώτιο από το φορτηγό :30sec
- χρόνος για να παραλάβει το AGV το εμπορευματοκιβώτιο από τον YC :30sec
- χρόνος για να παραλάβει ο QC το εμπορευματοκιβώτιο από το AGV :30sec
- χρόνος για να παραλάβει το AGV το εμπορευματοκιβώτιο από τον QC :30sec
- χρόνος για να τοποθετήσει ο QC το εμπορευματοκιβώτιο στο πλοίο :30sec
- χρόνος για να παραλάβει ο QC το εμπορευματοκιβώτιο από το πλοίο :30sec
- ταχύτητα κίνησης φορείου TC: 2m/s
- ταχύτητα κίνησης φορείου YC: 2m/s
- ταχύτητα κίνησης φορείου QC: 2m/s
- αριθμός φορτηγών: 9

και μεταβλητά στοιχεία αποτελούν τα:

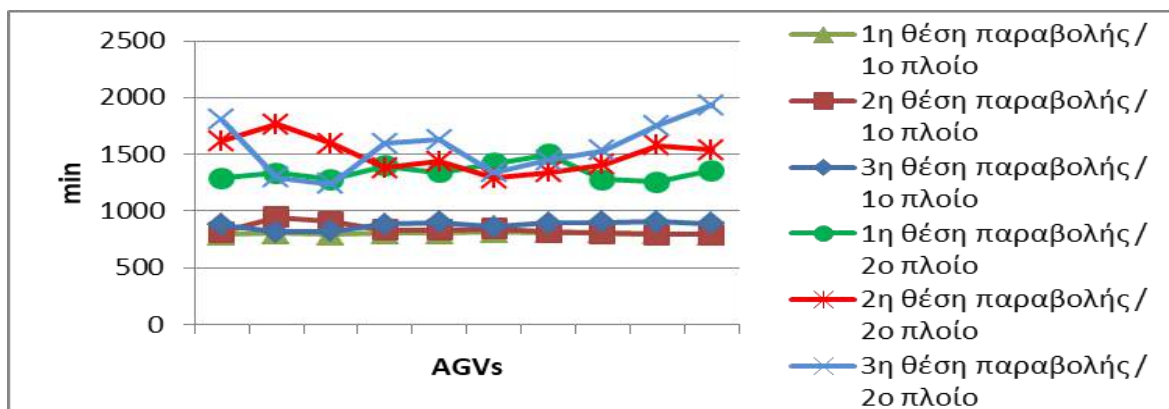
- αριθμός AGVs
- αριθμός YCs
- αριθμός QCs
- κανόνες λειτουργίας των YCs
- κανόνες λειτουργίας των AGVs

Σενάριο 3.1: 6 YC, 6QC, μεταβλητός αριθμός AGV και κανόνες λειτουργίας των YCs και AGVs

Στο σενάριο αυτό έρχονται 6 πλοία, τα οποία παραβάλουν σε 3 θέσεις προκειμένου αρχικά να εκφορτώσουν και στην συνέχεια να φορτώσουν, μετρώντας τον χρόνο που παραμένουν στον χώρο παραβολής, για τις διάφορες μεταβολές του αριθμού των AGVs και των κανόνων προτεραιότητας εξυπηρέτησης των AGVs και των YCs. Επίσης υπολογίζεται ο μέσος χρόνος παραμονής των πρώτων τριών πλοίων όπου εξυπηρετούνται ταυτόχρονα, όπως και της δεύτερης τριάδας προκειμένου να βγουν συμπεράσματα.

AGVs stay at position and the nearest YC take Container										
AGVs/time	6	7	8	9	10	11	12	13	14	15
1η θέση παραβολής / 1ο πλοίο	790	803	795	804	802	813	812	808	801	792
2η θέση παραβολής / 1ο πλοίο	813	943	905	834	834	846	815	806	795	793
3η θέση παραβολής / 1ο πλοίο	878	814	821	884	901	861	897	899	904	889
μέσος χρόνος 1ων πλοίων	827	853	841	840	846	840	841	838	833	825
1η θέση παραβολής / 2ο πλοίο	1292	1336	1277	1399	1341	1422	1500	1278	1257	1355
2η θέση παραβολής / 2ο πλοίο	1619	1763	1596	1378	1435	1289	1343	1408	1579	1537
3η θέση παραβολής / 2ο πλοίο	1803	1297	1239	1598	1627	1341	1448	1535	1751	1934
μέσος χρόνος 2ων πλοίων	744	612	530	618	622	511	589	570	696	784

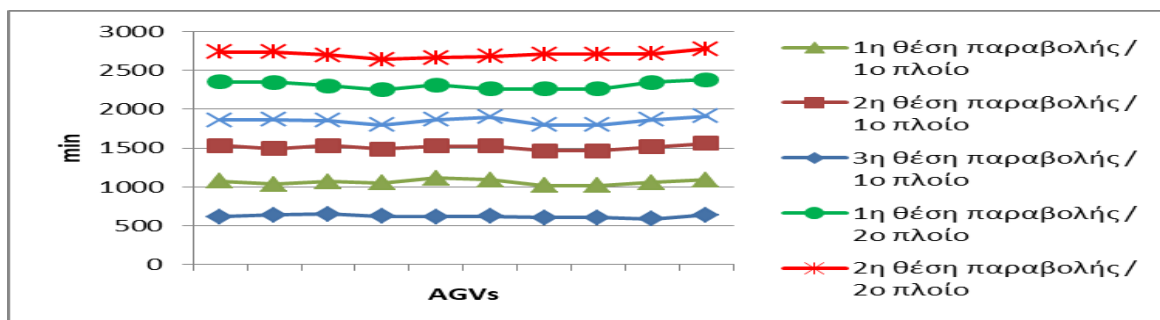
Πίνακας 20 Σενάριο 3.1 αναλυτικά αποτελέσματα



Εικόνα 107 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται

AGVs return at base and the nearest and idle YC take Container										
AGVs/time	6	7	8	9	10	11	12	13	14	15
1η θέση παραβολής / 1ο πλοίο	1072	1037	1068	1053	1114	1092	1015	1015	1059	1092
2η θέση παραβολής / 1ο πλοίο	1531	1491	1530	1484	1526	1524	1461	1461	1516	1559
3η θέση παραβολής / 1ο πλοίο	614	637	650	623	616	624	607	607	586	640
μέσος χρόνος 1ων πλοίων	1072	1055	1083	1053	1085	1080	1028	1028	1053	1097
1η θέση παραβολής / 2ο πλοίο	2347	2343	2299	2246	2312	2258	2259	2259	2343	2377
2η θέση παραβολής / 2ο πλοίο	2735	2732	2694	2635	2662	2677	2706	2706	2712	2773
3η θέση παραβολής / 2ο πλοίο	1859	1866	1854	1795	1865	1897	1796	1796	1864	1907
μέσος χρόνος 2ων πλοίων	1242	1259	1200	1172	1194	1198	1226	1226	1253	1255

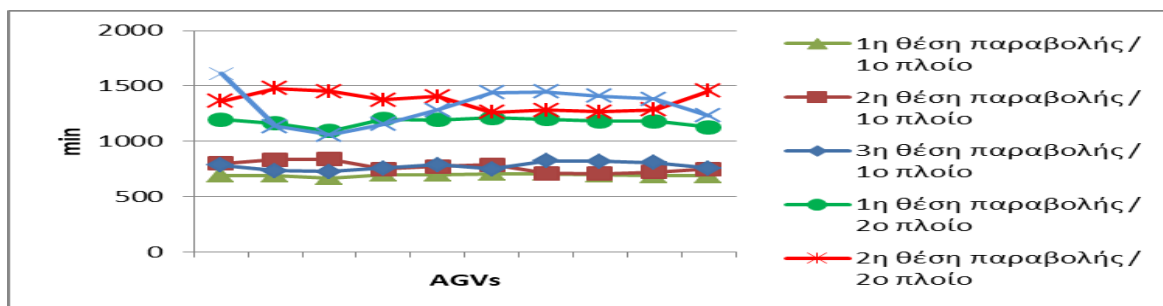
Πίνακας 21 Σενάριο 3.1 αναλυτικά αποτελέσματα



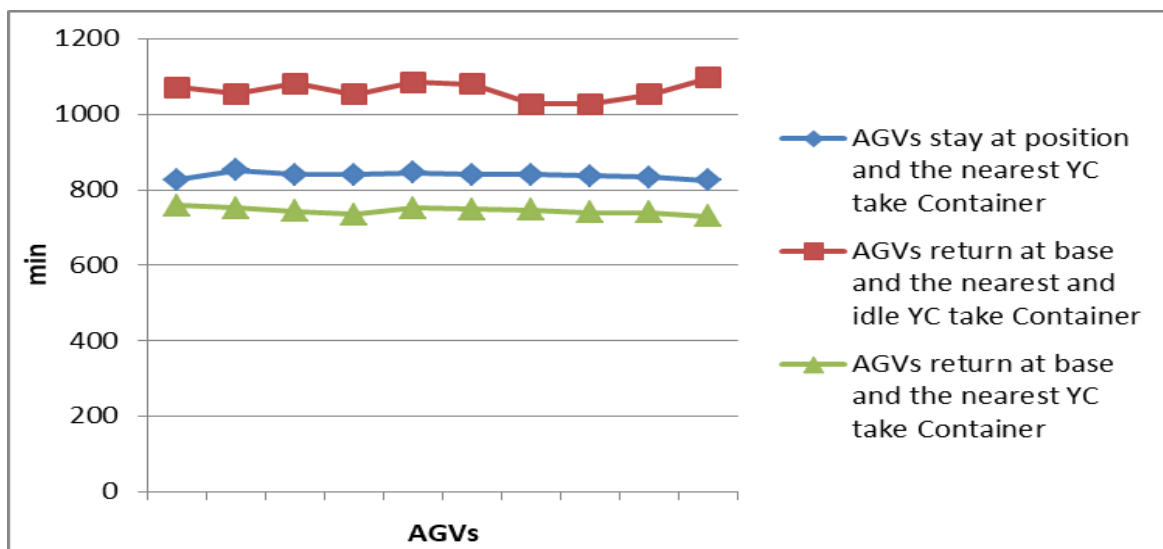
Εικόνα 108 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται

AGVs return at base and the nearest YC take Container										
AGVs/time	6	7	8	9	10	11	12	13	14	15
1η θέση παραβολής / 1ο πλοίο	693	689	669	699	697	707	707	693	688	686
2η θέση παραβολής / 1ο πλοίο	799	834	837	748	772	787	712	706	722	748
3η θέση παραβολής / 1ο πλοίο	785	734	724	758	789	750	824	818	808	756
μέσος χρόνος 1ων πλοίων	759	752	744	735	753	748	748	739	739	730
1η θέση παραβολής / 2ο πλοίο	1196	1161	1093	1201	1192	1213	1201	1180	1180	1127
2η θέση παραβολής / 2ο πλοίο	1362	1480	1452	1374	1404	1262	1280	1267	1287	1455
3η θέση παραβολής / 2ο πλοίο	1613	1135	1059	1156	1283	1438	1446	1410	1383	1232
μέσος χρόνος 2ων πλοίων	631	506	458	509	540	556	561	546	544	542

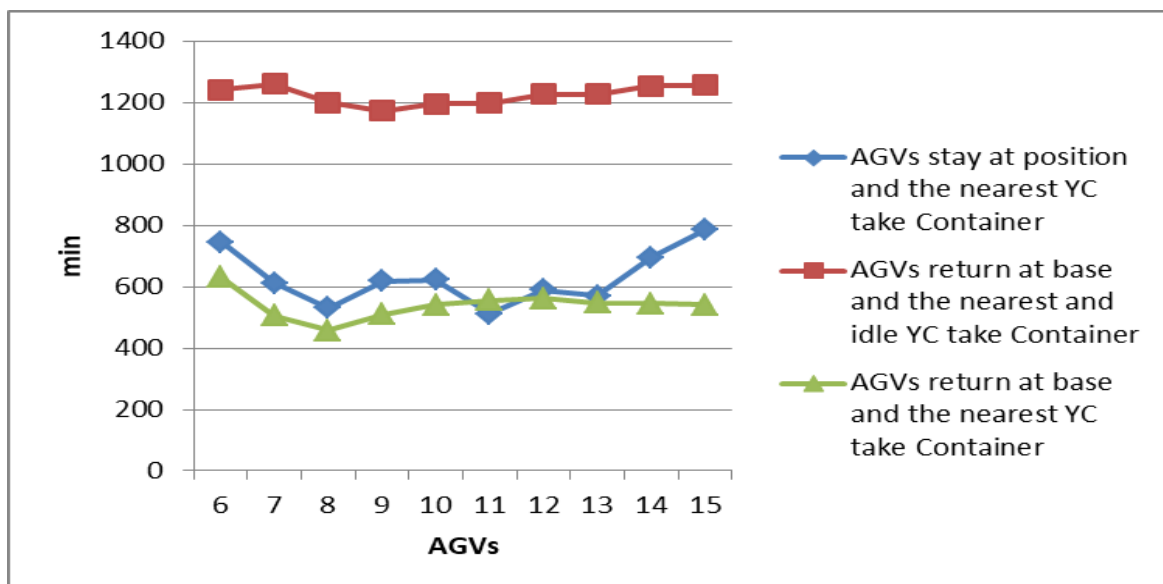
Πίνακας 22 Σενάριο 3.1 αναλυτικά αποτελέσματα



Εικόνα 109 Σενάριο 3.1 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται



Εικόνα 110 Σενάριο 3.1 μέσοι χρόνοι εξυπηρέτησης πλοίων



Εικόνα 111 Σενάριο 3.1 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

Οι χρόνοι από τους διάφορους αριθμούς των AGVs δεν αποτελεί στοιχείο το οποίο θα μπορούσε να επηρεάσει την αποδοτικότητα του λιμένα σε αυτό το σενάριο καθώς τα αποτελέσματα δείχνουν μια τυχαιότητα.

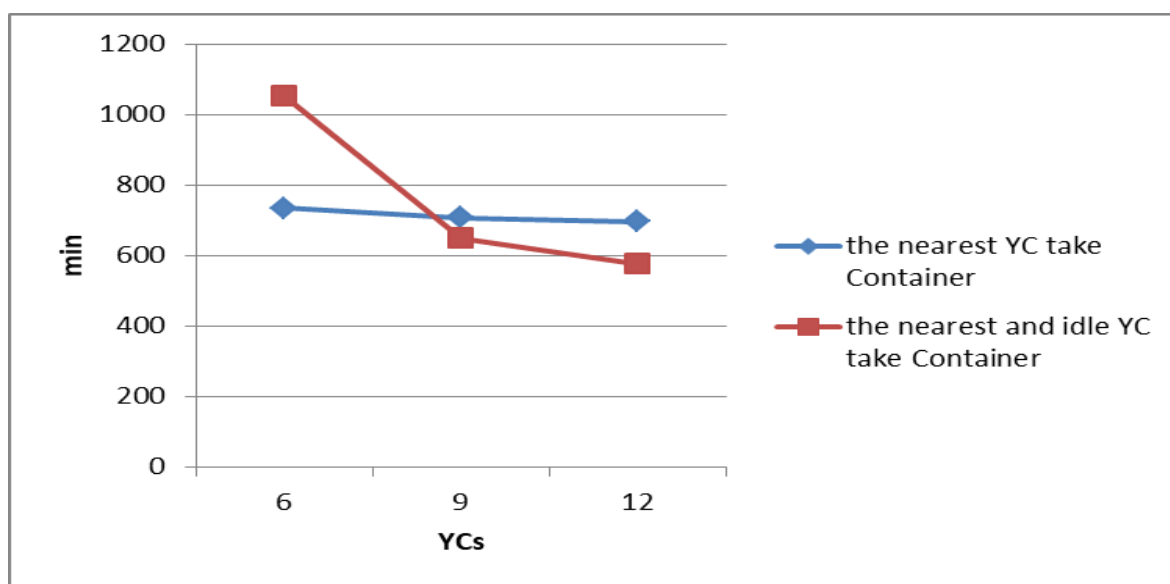
Παρόλα αυτά, οι διαφορές στους κανόνες λειτουργίας των AGVs επηρεάζουν σε μεγάλο βαθμό την λειτουργία αυτή. Από τους παραπάνω τρεις τρόπους όπου έγινε η εφαρμογή, το πιο αποδοτικό είναι τα AGVs να επιστρέφουν στην βάση τους αν δεν υπάρχει άλλη εργασία για ανάθεση και οι YCs να επιλέγονται ανάλογα με το ποιος είναι πιο κοντά, ανεξάρτητα το αν είναι διαθέσιμοι.

Σενάριο 3.2: 9 AGV, 6QC, μεταβλητός αριθμός YC και κανόνες λειτουργίας των YCs

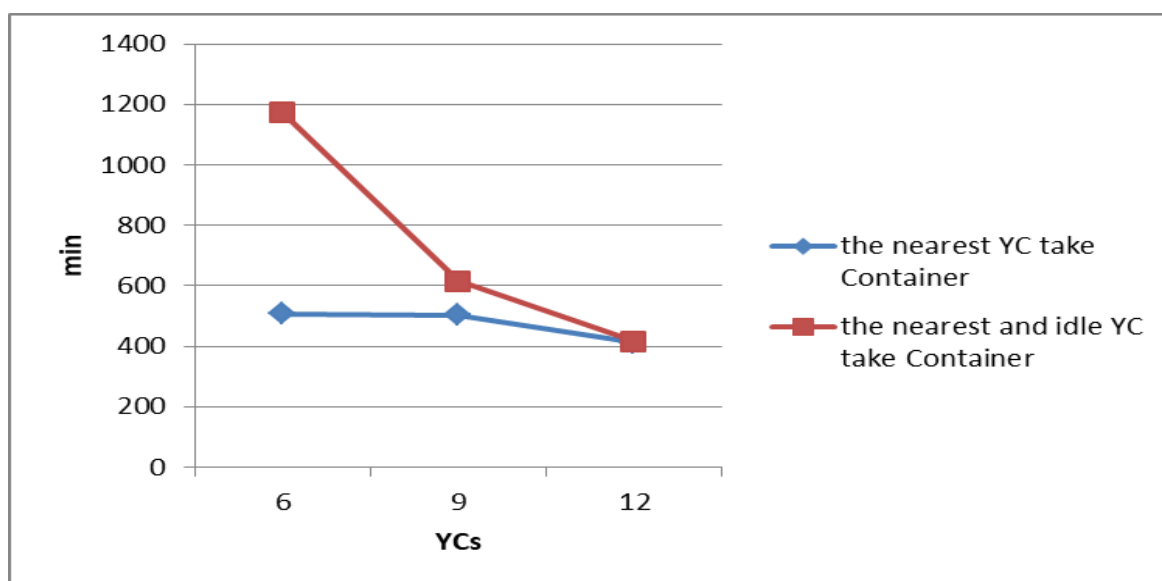
Στο σενάριο αυτό έρχονται 6 πλοία, τα οποία παραβάλουν σε 3 θέσεις προκειμένου αρχικά να εκφορτώσουν και στην συνέχεια να φορτώσουν, μετρώντας τον χρόνο που παραμένουν στον χώρο παραβολής και μετράται ο χρόνος που παραμένουν στον χώρο παραβολής για τις διάφορες μεταβολές του αριθμού των YCs και των κανόνων προτεραιότητας εξυπηρέτησης των YCs. Επίσης υπολογίζεται ο μέσος χρόνος παραμονής των πρώτων τριών πλοίων όπου εξυπηρετούνται ταυτόχρονα, όπως και της δεύτερης τριάδας προκειμένου να βγουν συμπεράσματα.

	the nearest YC take Container			the nearest & idle YC take Container		
YCs/time	6	9	12	6	9	12
1η θέση παραβολής / 1ο πλοίο	699	703,5167	677,8167	1052,9833	629,433333	582,366667
1ο κόκκινο πλοίο	747,9833	710,7167	744,2333	1483,9	845,55	700,716667
3η θέση παραβολής / 1ο πλοίο	758,05	709,9667	668,7	622,51667	471,95	450,033333
μέσος χρόνος 1ων πλοίων	735,0111	708,0667	696,9167	1053,1333	648,977778	577,705556
1η θέση παραβολής / 2ο πλοίο	1200,533	1174,767	1120,967	2246,15	1233,71667	957,416667
2η θέση παραβολής / 2ο πλοίο	1374,067	1311,017	1157,483	2634,6167	1589,45	1212,23333
3η θέση παραβολής / 2ο πλοίο	1155,983	1149,433	1051,333	1794,65	970,216667	816,816667
μέσος χρόνος 2ων πλοίων	508,5167	503,6722	413,0111	1172,0056	615,483333	417,783333

Πίνακας 23 Σενάριο 3.2 αναλυτικά αποτελέσματα



Εικόνα 112 Σενάριο 3.2 μέσοι χρόνοι εξυπηρέτησης πλοίων



Εικόνα 113 Σενάριο 3.2 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

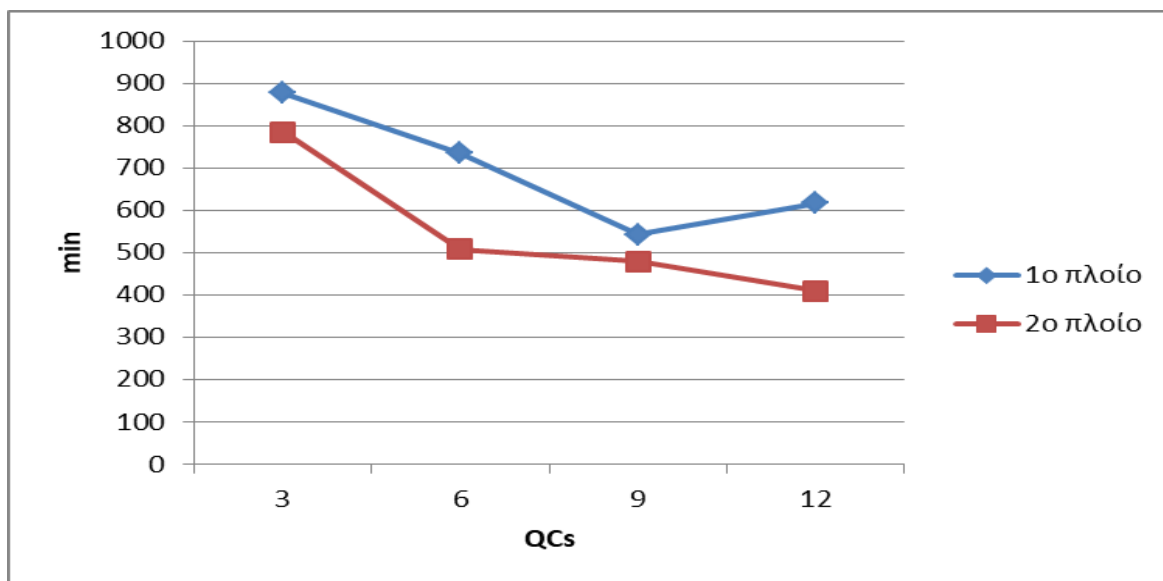
Παρατηρείται ότι ο βέλτιστος κανόνας προτεραιότητας εξυπηρέτησης των γερανών είναι να αναλαμβάνουν το κοντινότερο εμπορευματοκιβώτιο ανεξάρτητα αν είναι διαθέσιμος ή όχι. Σχετικά με τον αριθμό των γερανών, φαίνεται να βελτιώνεται σε μικρό βαθμό ο χρόνος εξυπηρέτησης των πλοίων (5 λεπτά ανά πλοίο μεταξύ 6 και 9 γερανών) κατά την αύξηση του.

Σενάριο 3.3: 9 AGV, 6YC, μεταβλητός αριθμός QC

Στο σενάριο αυτό έρχονται 6 πλοία, τα οποία παραβάλουν σε 3 θέσεις προκειμένου αρχικά να εκφορτώσουν και στην συνέχεια να φορτώσουν, μετρώντας τον χρόνο που παραμένουν στον χώρο παραβολής για τις διάφορες μεταβολές του αριθμού των γερανών κρηπιδώματος. Επίσης υπολογίζεται ο μέσος χρόνος παραμονής των πρώτων τριών πλοίων όπου εξυπηρετούνται ταυτόχρονα, όπως και της δεύτερης τριάδας προκειμένου να βγουν συμπεράσματα.

the nearest QC take Container				
QCs/time	3	6	9	12
1η θέση παραβολής / 1ο πλοίο	859	699	513	528
2η θέση παραβολής / 1ο πλοίο	948	748	659	679
3η θέση παραβολής / 1ο πλοίο	824	758	456	644
	877	735	542	617
1η θέση παραβολής / 2ο πλοίο	1501	1201	996	835
2η θέση παραβολής / 2ο πλοίο	1733	1374	989	1094
3η θέση παραβολής / 2ο πλοίο	1752	1156	1081	1152
	785	509	480	410

Πίνακας 24 Σενάριο 3.3 αναλυτικά αποτελέσματα



Εικόνα 114 Σενάριο 3.3 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

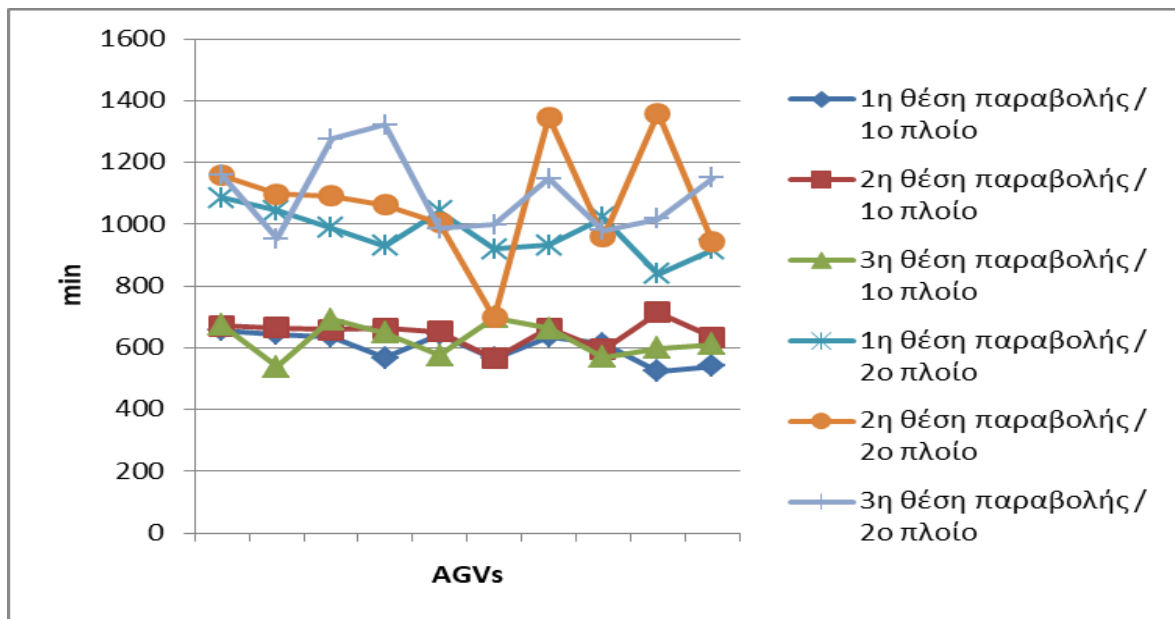
Παρατηρείται μείωση του χρόνου αναμονής των πλοίων κατά την αύξηση του αριθμού των γερανών κρηπιδώματος. Κατά την αύξηση από 6 σε 9 γερανούς παρατηρείται μια μείωση του μέσου χρόνου ανά πλοίο κατά 29 λεπτά. Η επιλογή των 12 γερανών (4 γερανοί ανά πλοίο) φαίνεται η καλύτερη επιλογή καθώς παρατηρείται μείωση του χρόνου κατά 70 λεπτά.

Σενάριο 3.4: 9 QC's, 6YC, μεταβλητός αριθμός AGVs

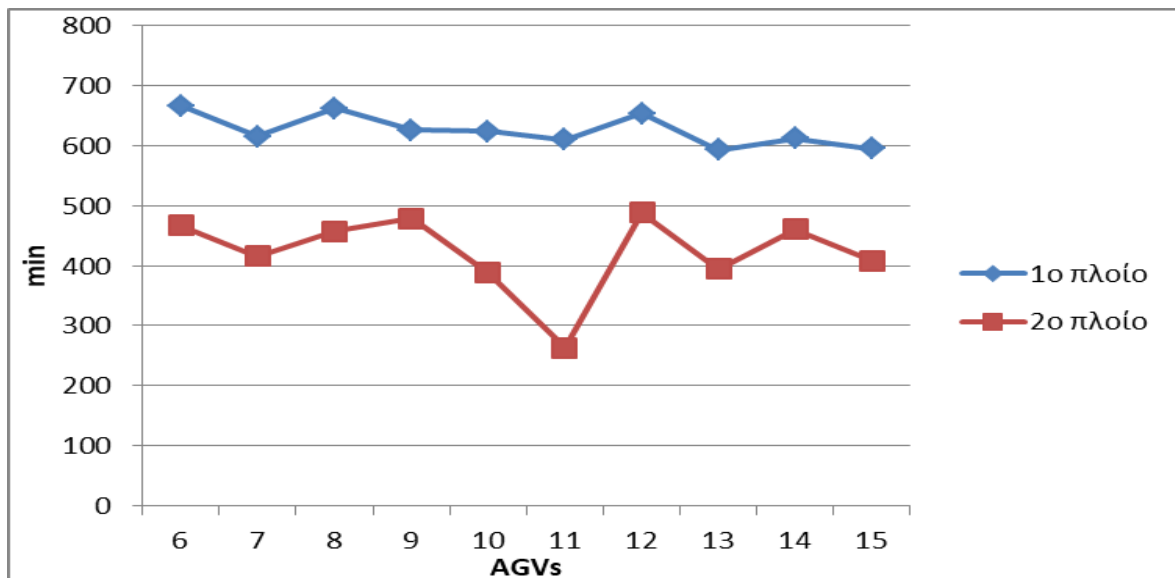
Μετά την εφαρμογή των σεναρίων 7,8 και 9 θεωρείται κρίσιμο μηχανήμα για το λιμένα οι γερανοί κρηπιδώματος. Το πιο αποδοτικό σενάριο είναι να υπάρχουν 12 γερανοί κρηπιδώματος, δηλαδή 4 γερανοί ανά πλοίο, αλλά λόγω πρακτικών προβλημάτων στην συνύπαρξη τόσο πολλών γερανών σε μικρό χώρο επιλέχτηκε ως το πιο αποδοτικό οι 9 γερανοί κρηπιδώματος, δηλαδή 3 γερανοί ανά πλοίο. Στην συνέχεια μελετάται ο βέλτιστος αριθμός των AGVs.

AGVs return at base and the nearest YC take Container										
AGVs/time	6	7	8	9	10	11	12	13	14	15
1η θέση παραβολής / 1ο πλοίο	655	645	635	567	644	565	635	614	523	541
2η θέση παραβολής / 1ο πλοίο	671	665	658	662	651	566	663	595	715	635
3η θέση παραβολής / 1ο πλοίο	673	537	692	648	575	698	663	569	598	610
μέσος χρόνος 1ων πλοίων	667	616	662	626	623	609	654	592	612	595
1η θέση παραβολής / 2ο πλοίο	1085	1045	989	930	1042	919	931	1022	840	917
2η θέση παραβολής / 2ο πλοίο	1158	1098	1092	1062	1004	698	1347	959	1359	942
3η θέση παραβολής / 2ο πλοίο	1158	952	1277	1321	986	999	1149	979	1018	1151

Πίνακας 25 Σενάριο 3.4 αναλυτικά αποτελέσματα



Εικόνα 115 Σενάριο 3.4 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται



Εικόνα 116 Σενάριο 3.4 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

Οι χρόνοι από τους διάφορους αριθμούς των AGVs δεν αποτελεί στοιχείο το οποίο θα μπορούσε να επηρεάσει την αποδοτικότητα του λιμένα σε αυτό το σενάριο καθώς τα αποτελέσματα δείχνουν μια τυχαιότητα.

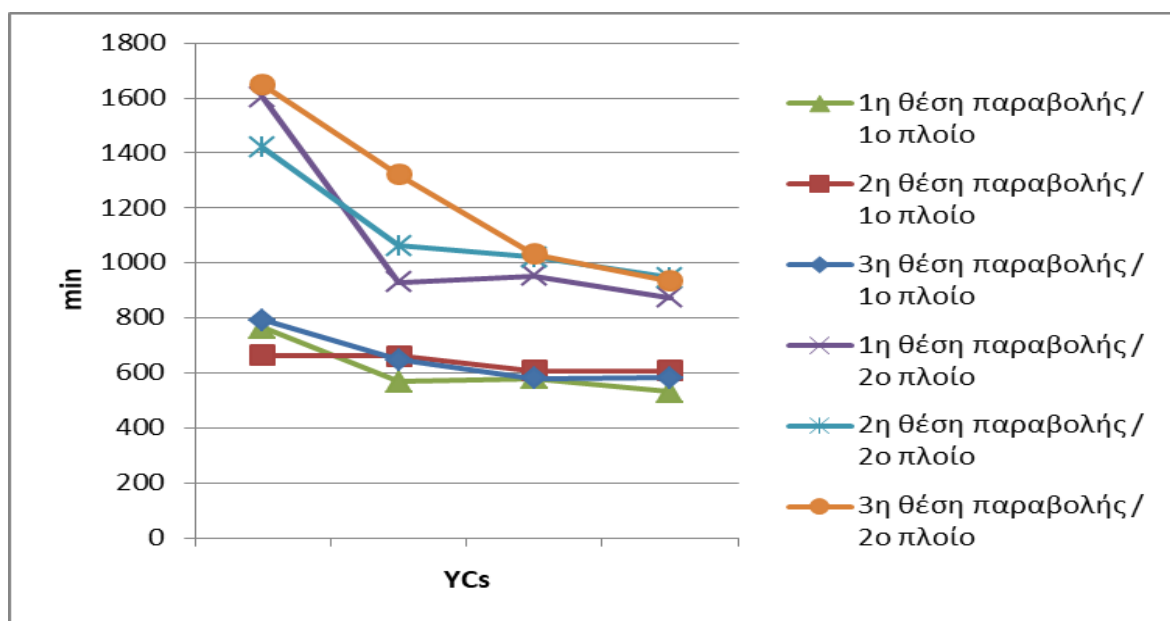
Σενάριο 3.5: 9 AGVs, 9QCs, μεταβλητός αριθμός YCs

Μετά την εφαρμογή των σεναρίων 7,8 και 9 θεωρείται κρίσιμο μηχανήμα για το λιμένα οι γερανοί κρηπιδώματος. Το πιο αποδοτικό σενάριο είναι να υπάρχουν 12 γερανοί κρηπιδώματος, δηλαδή 4 γερανοί ανά πλοίο, αλλά λόγω πρακτικών προβλημάτων στην συνύπαρξη τόσο πολλών

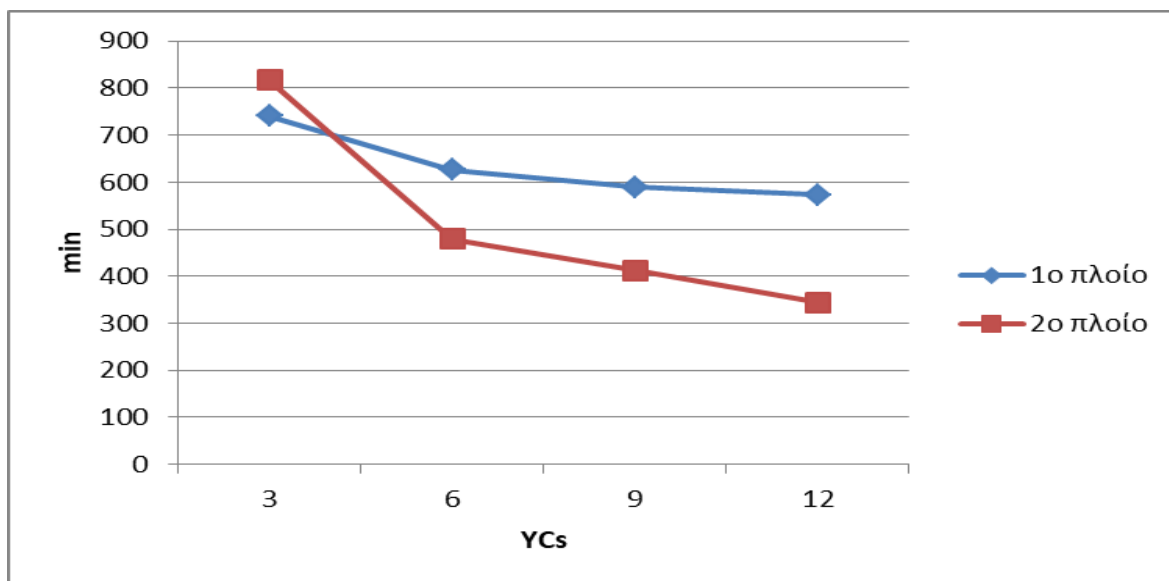
γερανών σε μικρό χώρο επιλέχτηκε ως το πιο αποδοτικό οι 9 γερανοί κρηπιδώματος, δηλαδή 3 γερανοί ανά πλοίο. Στην συνέχεια μελετάται ο βέλτιστος αριθμός των YCs.

the nearest YC take Container				
YCs/time	3	6	9	12
1η θέση παραβολής / 1ο πλοίο	764	567	580	530
2η θέση παραβολής / 1ο πλοίο	664	662	608	607
3η θέση παραβολής / 1ο πλοίο	791	648	580	583
μέσος χρόνος 1ων πλοίων	740	626	589	573
1η θέση παραβολής / 2ο πλοίο	1602	930	953	874
2η θέση παραβολής / 2ο πλοίο	1419	1062	1020	945
3η θέση παραβολής / 2ο πλοίο	1648	1321	1032	934
μέσος χρόνος 2ων πλοίων	817	479	412	344

Πίνακας 26 Σενάριο 3.5 αναλυτικά αποτελέσματα



Εικόνα 117 Σενάριο 3.5 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται



Εικόνα 118 Σενάριο 3.5 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

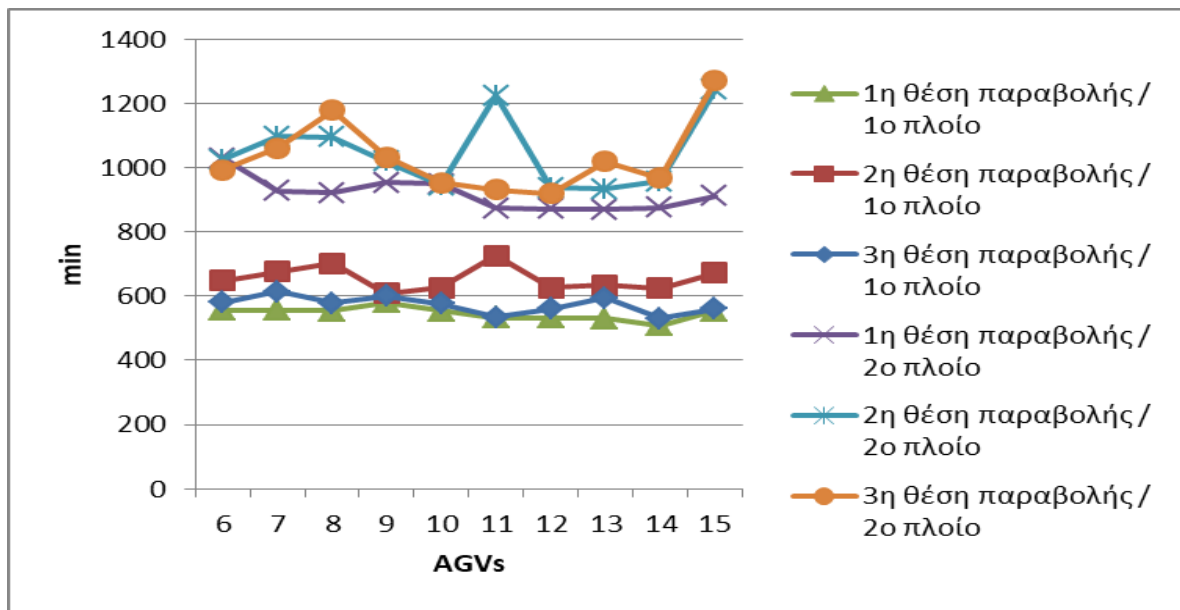
Παρατηρείται μείωση του χρόνου αναμονής των πλοίων κατά την αύξηση του αριθμού των YCs. Κατά την αύξηση από 6 σε 9 γερανούς παρατηρείται μια μείωση του μέσου χρόνου ανά πλοίο κατά 67 λεπτά. Η επιλογή των 12 γερανών φαίνεται η καλύτερη επιλογή καθώς παρατηρείται μείωση του χρόνου κατά 78 λεπτά. Παρόλα αυτά για χωρικούς περιορισμούς επιλέγεται ως η βέλτιστη λύση οι 9 γερανοί.

Σενάριο 3.6: 9 YCs, 9QCs, μεταβλητός αριθμός AGVs

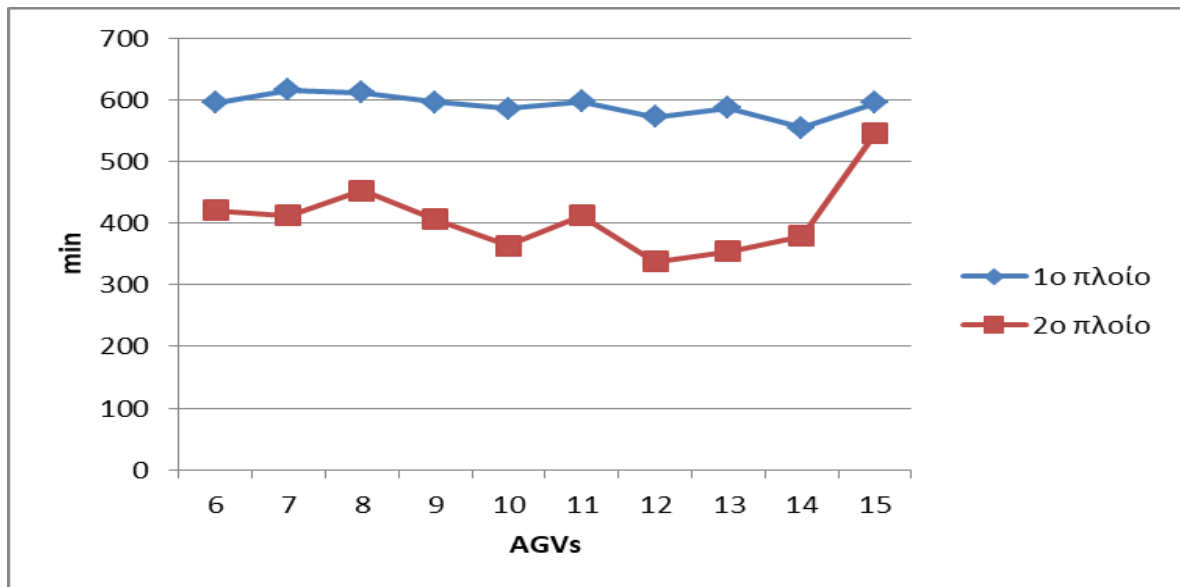
Μετά την εφαρμογή των σεναρίων 7,8,9 και 10, από τα συμπεράσματα καταλήγουμε στην ύπαρξη 9 YCs και 9 QCs για την βέλτιστη λειτουργία του λιμένα. Σε αυτό το σενάριο μελετάται ο αριθμός των απαιτούμενων AGVs.

AGVs return at base and the nearest YC take Container										
AGVs/time	6	7	8	9	10	11	12	13	14	15
1η θέση παραβολής/ 1ο πλοίο	556	556	555	580	554	532	532	530	508	553
2η θέση παραβολής/ 1ο πλοίο	648	675	702	608	625	726	626	634	623	673
3η θέση παραβολής/ 1ο πλοίο	581	615	578	599	577	533	560	594	531	559
μέσος χρόνος 1ων πλοίων	595	615	612	596	585	597	572	586	554	595
1η θέση παραβολής/ 2ο πλοίο	1028	927	922	953	949	873	872	869	876	912
2η θέση παραβολής/ 2ο πλοίο	1026	1097	1094	1020	945	1224	938	932	957	1241
3η θέση παραβολής/ 2ο πλοίο	992	1060	1179	1032	953	932	919	1018	966	1270
μέσος χρόνος 2ων πλοίων	420	412	453	406	364	413	337	354	379	546

Πίνακας 27 Σενάριο 3.6 αναλυτικά αποτελέσματα



Εικόνα 119 Σενάριο 3.6 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται



Εικόνα 120 Σενάριο 3.6 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

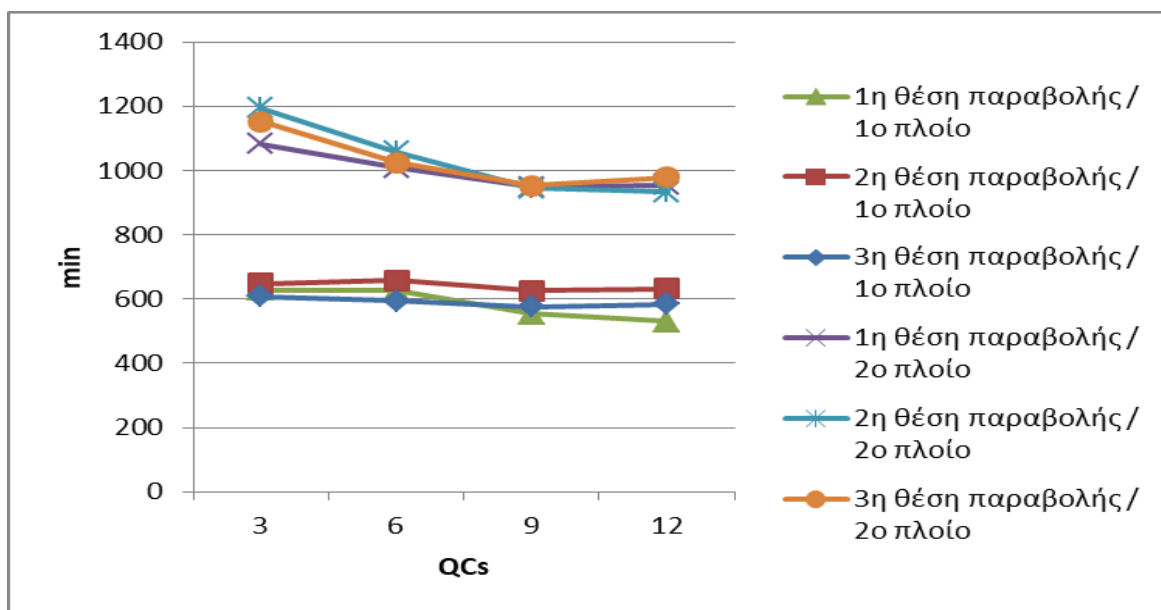
Παρατηρείται ότι ο ελάχιστος χρόνος παραμονής για τα πλοία είναι για 10 AGVs.

Σενάριο 3.7: 9 YCs, 10 AGVs, μεταβλητός αριθμός QCs

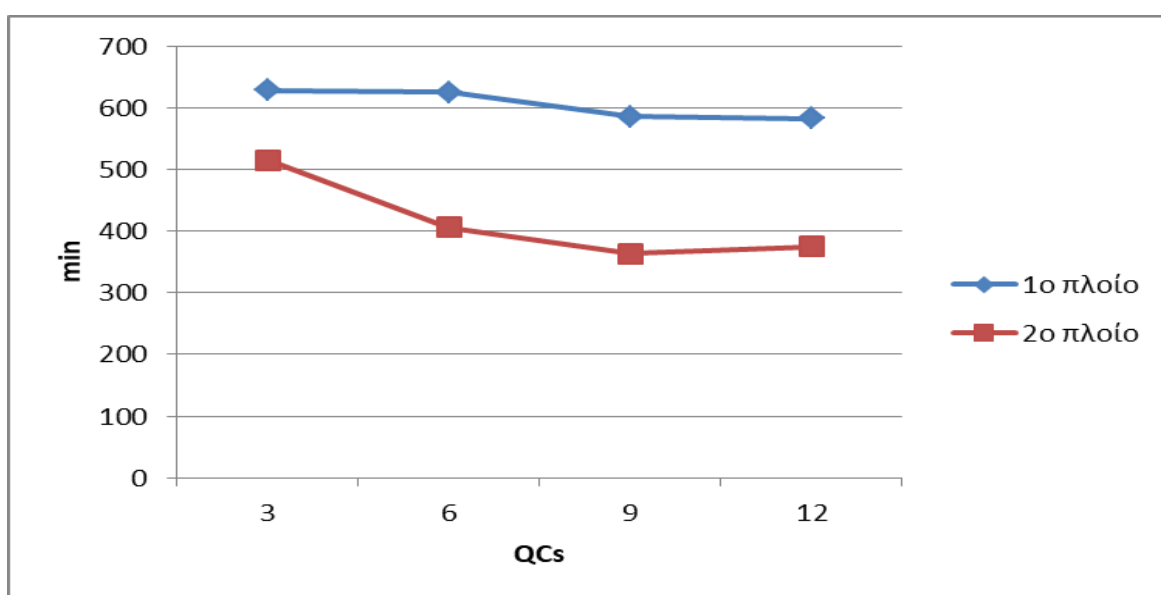
Στο σενάριο αυτό γίνεται επανέλεγχος για το αν αυτές οι μεταβολές επηρέασαν τον βέλτιστο αριθμό γερανών κρηπιδώματος που είχαν επιλεγεί (9).

the nearest QC take Container				
QCs/time	3	6	9	12
1η θέση παραβολής / 1ο πλοίο	628	625	554	531
2η θέση παραβολής / 1ο πλοίο	648	657	625	631
3η θέση παραβολής / 1ο πλοίο	608	593	577	584
μέσος χρόνος 1ων πλοίων	628	625	585	582
1η θέση παραβολής / 2ο πλοίο	1083	1009	949	956
2η θέση παραβολής / 2ο πλοίο	1195	1058	945	934
3η θέση παραβολής / 2ο πλοίο	1152	1025	953	979
μέσος χρόνος 2ων πλοίων	515	405	364	374

Πίνακας 28 Σενάριο 3.7 αναλυτικά αποτελέσματα



Εικόνα 121 Σενάριο 3.7 αναλυτικό διάγραμμα χρόνων ολοκλήρωσης για το κάθε πλοίο που εξυπηρετείται



Εικόνα 122 Σενάριο 3.7 μέσοι χρόνοι εξυπηρέτησης πλοίων

Συμπεράσματα:

Δεν παρατηρείται κάποια διαφορά με την έρευνα που έγινε στο σενάριο 9. Ο βέλτιστος αριθμός των γερανών κρηπιδώματος παραμένει 9.

Συμπεράσματα σεναρίων

Συνοπτικά τα σενάρια με τα αποτελέσματά τους:

	Αριθμός YCs	Κανόνες προτεραιότητας YC	Αριθμός AGVs	Κανόνες προτεραιότητας AGV	Αριθμός QCs	Τελική επιλογή παραμέτρων	Χρόνος ολοκλήρωσης διεργασιών σε λεπτά	Σχόλια	Βελτίωση χρόνου %
Σενάριο 3.0	6	Κοντινότερο	9	Κοντινότερο διαθέσιμο & παραμονή στη θέση	6	-	618	Αρχικές τιμές	-
Σενάριο 3.1	6	Κοντινότερο	6 ~ 15	Παραμονή στη θέση / Επιστροφή στην βάση	6	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	508	Η μεταβολή του αριθμού των AGVs δεν επηρεάζει το αποτέλεσμα	-17,7994
Σενάριο 3.2	6~12	Κοντινότερο/ Κοντινότερο διαθέσιμο	9	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	6	Κοντινότερο	508	Η μεταβολή του αριθμού των YCs βελτιώνει αμελητέα το αποτέλεσμα	0
Σενάριο 3.3	6	Κοντινότερο	9	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	6 ~ 12	Αριθμός QCs=9	480	-	-5,51181
Σενάριο 3.4	6	Κοντινότερο	6 ~ 15	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	9	-	-	Ο αριθμός των AGVs δεν επηρεάζει	0
Σενάριο 3.5	6~12	Κοντινότερο	9	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	9	Αριθμός YCs = 9	412	-	-14,1667
Σενάριο 3.6	9	Κοντινότερο	6 ~ 15	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	9	Αριθμός AGVs = 10	364	-	-11,6505
Σενάριο 3.7	9	Κοντινότερο	11	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	6 ~ 12	Αριθμός QCs=9	364	Επιτυχής επαλήθευση	0
Τελικά αποτελέσματα	9	Κοντινότερο	11	Κοντινότερο διαθέσιμο & επιστροφή στην βάση	9	-	364	Συνολική βελτίωση	-41,1003

Πίνακας 29 Αναλυτικά αποτελέσματα σύγκρισης σεναρίων 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 και 3.7

Μετά την εφαρμογή των σεναρίων 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 και 3.7 φαίνεται πώς ο βέλτιστος συνδυασμός του αριθμού μηχανημάτων στο συγκεκριμένο λιμένα με αυτές τις σταθερές είναι 10 AGVs, 9 YCs και 9 γερανοί κρηπιδώματος. Ο μέσος χρόνος εξυπηρέτησης του κάθε πλοίου με αυτές τις συνθήκες είναι 364 λεπτά σε συνεχή λειτουργία του λιμένα. Ο χρόνος εξυπηρέτησης των πρώτων πλοίων που παραβάλουν στην παραβολή μετά την διακοπή λειτουργίας του λιμένα είναι 585 λεπτά.

7 ΣΥΜΠΕΡΑΣΜΑΤΑ

Κάθε σύγχρονος τερματικός σταθμός φορτοεκφόρτωσης εμπορευματοκιβωτίων είναι ένα πολύπλοκο σύστημα που απαιτεί συνεχή προσπάθεια βελτίωσης καθώς υπάρχουν πολλές διαφορετικές συνιστώσες που αλληλεπιδρούν μεταξύ τους. Η διερεύνηση των θεμάτων αυτών μπορεί να διευκολυνθεί με την χρήση μοντέλων.

Η προσομοίωση είναι μια μέθοδος που μπορεί να χρησιμοποιηθεί τόσο για την αναπαράσταση απλών συστημάτων αναμονής, όσο και για την αναπαράσταση σύνθετων συστημάτων με πολλούς και διαφορετικούς πελάτες, εξυπηρετητές με διαφορετικά χαρακτηριστικά και κανόνες οργάνωσης και λειτουργίας (π.χ. ένα υποσύστημα μεταφοράς εμπορευματοκιβωτίων μεταξύ θέσεων παραβολής και επιλιμένιου σιδηροδρομικού σταθμού).

Το λογισμικό Anylogic προσφέρει σημαντικές δυνατότητες προσομοίωσης ενός συστήματος καθώς μπορεί να προσομοιώσει από απλά συστήματα όπως π.χ. M/M/1, M/M/2, E2/M/1 (Σε σύγκριση με αντίστοιχα μοντέλα από θεωρία αναμονής εντοπίστηκαν μικρές αποκλίσεις (<1%)) μέχρι και πιο σύνθετα όπως ένα σύστημα τερματικού σταθμού εμπορευματοκιβωτίων που περιλαμβάνει πολλά υποσυστήματα (π.χ. εκφόρτωση εμπορευματοκιβωτίων από τρένα, παραλαβή εμπορευματοκιβωτίων από φορτηγά, φορτοεκφόρτωση πλοίων και στοιβάση σε χώρο εναπόθεσης) που συνδέονται μεταξύ τους. Ως μειονεκτήματα του προγράμματος αναφέρονται: α) Η δυσκολία στην εκμάθηση για έναν νέο χρήστη β) προϋποθέτει καλές γνώσεις προγραμματισμού για την προσομοίωση πολύπλοκων κανόνων λειτουργίας και την κατασκευή αυτοματοποιημένων μοντέλων (π.χ. μοντέλο Monte Carlo).

Η διερεύνηση του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου με χρήση του προγράμματος AnyLogic έδειξε ότι οι στρατηγικές ανάθεσης εργασιών στα οχήματα μεταφοράς α) επηρεάζει σημαντικά τον χρόνο αναχώρησης του πλοίου β) στην περίπτωση που παραβάλουν ταυτόχρονα δύο πλοία, η προτιμητέα στρατηγική ανάθεσης εξαρτάται από την μεταξύ τους απόσταση σε σχέση με την απόσταση τους από τον χώρο στοιβάσης.

8 ΕΙΣΗΓΗΣΕΙΣ ΓΙΑ ΠΕΡΑΙΤΕΡΩ ΕΡΕΥΝΑ

- Να εξεταστούν και άλλες τεχνικές οργάνωσης του υποσυστήματος, όπως οι λειτουργίες διπλού κύκλου των γερανογεφυρών στο κρηπίδωμα, όπου θα εκφορτώνουν ένα εμπορευματοκιβώτιο από το πλοίο και στον ίδιο κύκλο θα παραλαμβάνουν ένα άλλο εμπορευματοκιβώτιο και θα το φορτώνουν στο πλοίο.
- Να εξεταστούν περιπτώσεις όπου συμβαίνουν απρόβλεπτα προβλήματα, όπως η βλάβη κάποιου οχήματος μεταφοράς ή η άφιξη επιτακτικών πελατών και η ανάγκη για εξυπηρέτηση σύμφωνα με προτεραιότητες.
- Να αναπτυχθούν οι τεχνικές οργάνωσης που εφαρμόστηκαν και σε άλλα λογισμικά προσομοίωσης ή ακόμα και με άλλες τεχνικές της επιχειρησιακής έρευνας (για παράδειγμα με Μικτό Ακέραιο Προγραμματισμό) και να συγκριθούν και να αξιολογηθούν τα αποτελέσματα.
- Να γίνει εφαρμογή των τεχνικών οργάνωσης που παρουσιάστηκαν σε πραγματικά δεδομένα λιμανιών και να αναπτυχθούν μοντέλα μεγαλύτερης κλίμακας ώστε συγκρίνοντας τα πραγματικά δεδομένα με τα αποτελέσματα των μοντέλων να πραγματοποιείται επικύρωση (validation) τους και στη συνέχεια αξιολόγηση διάφορων σεναρίων για τη βελτιστοποίηση των διαδικασιών.

9 ΒΙΒΛΙΟΓΡΑΦΙΑ

- Θάνου Ε., (2015), Διπλωματική εργασία, Διερεύνηση εναλλακτικών τεχνικών οργάνωσης του υποσυστήματος διακίνησης εμπορευματοκιβωτίων μεταξύ κρηπιδώματος και χερσαίου χώρου
- Καρλαύτης Μ.Γ., Λαγαρός Ν.Δ. (2010), Επιχειρησιακή έρευνα και βελτιστοποίηση για μηχανικούς, Εκδόσεις: Συμμετρία.
- Μαλιάγκα, Ε. (2014), Διπλωματική εργασία, Ανάπτυξη βάσης δεδομένων για ατυχήματα σύγκρουσης, επαφής και προσάραξης σε πλοία μεταφοράς εμπορευματοκιβωτίων, Εθνικό Μετσόβιο Πολυτεχνείο.
- Μπαλλής Α., Σημειώσεις μαθήματος (2014), Συνδυασμένες μεταφορές – Μοναδοποιημένα φορτία
- Ξηροκώστας, Δ. (1990), Επιχειρησιακή έρευνα – Εφαρμοσμένη θεωρία αναμονής, Εκδόσεις: Οργανισμός Εκδόσεων Διδακτικών Βιβλίων.
- Albers S. Lecture notes on competitive online algorithms. BRICS Lecture Series LS-96-2, Aarhus University, Denmark, 1996.
- Azar Y. On-line load balancing. Online Algorithms, The state of the art. LNCS 1442, Springer, pp. 178-195, 1996.
- Bartošek, A., Marek, O. (2013), Quay Cranes in Container Terminals, Transaction on Transport Sciences, vol.6(1): 1-10, doi: 10.2478/v10158-012-0027-y.
- Bierwirth, C., F. Meisel. (2009) A fast heuristic for quay crane scheduling with interference constraints. Journal of Scheduling. 12(4): 345–360.
- Bish, E.K. (2003) A multiple-crane-constrained scheduling problem in a container terminal. European Journal of Operational Research. 144(1): 83–107.
- Böse, J. (2011) Handbook of Terminal Planning, New York : Springer.
- Buhrkal, K., S. Zuglian, S. Ropke, J. Larsen, R. Lusby. (2011) Models for the discrete berth allocation problem: A computational comparison. Transportation Research Part E: Logistics and Transportation Review. 47(4): 461–473.
- Buiscar, Cargo Solutions, <http://www.buiscar.com/mts.htm>, τελευταία πρόσβαση 10/9/2015.
- Cao, Z., D. H. Lee, Q. Meng. (2008) Deployment strategies of double-rail-mounted gantry crane systems for loading outbound containers in container terminals. International Journal of Production Economics. 115(1): 221–228.
- Chang, D., Z. Jiang, W. Yan, J. He. (2010) Integrating berth allocation and quay crane assignments. Transportation Research Part E: Logistics and Transportation Review. 46(6): 975–990.
- Chang, D., Z. Jiang, W. Yan, J. He. (2011). Developing a dynamic rolling-horizon decision strategy for yard crane scheduling. Advanced Engineering Informatics 25(3): 485–494.
- Chan, S. H. (2001) Dynamic AGV-container job deployment strategy. Master of Science, National University of Singapore.

- Chen, L., A. Langevin. (2011) Multiple yard cranes scheduling for loading operations in a container terminal. *Engineering Optimization* 43(11): 1205–1221.
- Chen, L., Bostel, N., Dejax, P., Cai, J., & Xi, L. (2007) A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research*. 181(1): 40–58.
- Chen, J. H., Lee, D. H., Cao, J. X. (2012) A combinatorial benders' cuts algorithm for the quayside operation problem at container terminals. *Transportation Research Part E: Logistics and Transportation Review*. 48(1): 266–275
- Critical Manufacturing S.A. (2014)
- Dorndorf, U., F. Schneider. (2010) Scheduling automated triple cross-over stacking cranes in a container yard. *OR Spectrum*. 32(3): 617–632.
- Du, Y., Q. Chen, X. Quan, L. Long, R. Y. K. Fung. (2011) Berth allocation considering fuel consumption and vessel emissions. *Transportation Research Part E: Logistics and Transportation Review*. 47(6): 1021–1037.
- Edmond, E. and Maggs, R. (1978), How useful are queue models in Port investment decisions for container berths?, *J.Opl.Res.Soc*, vol.29 (8): 741-750.
- Edmonds, R. (2010), A new profile at the Port of Savannah, <http://www.racheledmonds.com/blog/category/shipping/>, τελευταία πρόσβαση 10/9/2015.
- Frankel, E.G. (1980), Container System Selection, The 2nd Terminal Operations Conference, Cargo Systems Publications, U.S.A.
- Gharehgozli, A. H., Yu, Yugang, Koster, R. d., Udding, J. T. (2014b) An exact method for scheduling a yard crane. *European Journal of Operational Research*. 235(2): 431–447.
- Grunow, M., Gunther, H.-O., and Lehmann, M. (2004) Dispatching multi-load AGVs in highly automated seaport container terminals. *OR Spectrum*. 26: 211–235.
- Guan Y., and Cheung, R. K. (2004) The berth allocation problem: models and solution methods, *OR Spectrum*. 26: 75–92.
- Günther, H.-O., Kim, K. H. (2005) Logistics control issues of container terminals and automated transportation systems, In: Günther, H.-O., Kim, K. H., *Container Terminals and Automated Transport Systems*.
- Han, X. L., Z. Q. Lu, L. F. Xi. (2010) A proactive approach for simultaneous berth and quay crane scheduling problem with stochastic arrival and handling time. *European Journal of Operational Research*. 207(3): 1327–1340.
- He, J., D. Chang, W. Mi, W. Yan. (2010) A hybrid parallel genetic algorithm for yard crane scheduling. *Transportation Research Part E: Logistics and Transportation Review*. 46(1): 136–155.
- Hendriks, M. P. M., D. Armbruster, M. Laumanns, E. Lefebvre, J. T. Udding. (2012) Strategic allocation of cyclically calling vessels for multi-terminal container operators. *Flexible Services and Manufacturing Journal*. 24:248–273.

- Holland J. (1975) adaptation in natural and artificial systems
- Homayouni, S. M., Tang, S. H. (2013) Multi Objective Optimization of Coordinated Scheduling of Cranes and Vehicles at Container Terminals, Mathematical Problems in Engineering, Article ID 746781, 9 pages, <http://dx.doi.org/10.1155/2013/746781>
- Homayouni, S. M., Tang, S. H., Motlagh, O. (2014) A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals, Journal of Computational and Applied Mathematics. 270: 545–556.
- Imai A., Nishimura E., Papadimitriou S. (2003) Berth allocation with service priority. Transportation Research-B. 37(5): 437–457.
- John D.C. Little, (1961), A Proof for the Queueing Formula " $L=\lambda*W$ ", Operations Research, vol.9 (3): 383-387.
- Kaveshgar, N., Huynh, N., and Rahimian, S. K. (2012) An efficient genetic algorithm for solving the quay crane scheduling problem. Expert Systems with Applications. 39(18): 13108–13117
- Kelton, W. D., Sadowski, R. P., Sturrock, D.T. (2004), Simulation with Arena, Third Edition, McGraw Hill.
- Kendal, D.G., (1964), Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of Imbedded Markov Chains, Annals of Mathematical Statistics, vol.24: 338-354.
- Kim, K. H. and Park, Y. M. (2004) A crane scheduling method for port container terminals, European Journal of Operational Research. 156: 752–768.
- KoneCranes, <http://www.konecranes.com/>, τελευταία πρόσβαση 10/9/2015.
- Laporte G. European journal of operational research 59 (3), 345-358, 1992.
- Lee, D.-H., Cao, J. X., and Shi, Q. X. (2008a) Integrated Quay crane and yard truck schedule for inbound containers, Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM '08), 1219–1223, December 2008.
- Lee, D.-H., Wang, H. Q., and Miao, L. (2008b) Quay crane scheduling with handling priority in port container terminals. Engineering Optimization. 40(2): 179–189.
- Legato, P., Truno, R., Meisel. F. (2012) Modeling and solving rich quay crane scheduling problems. Computers & Operations Research. 39(9): 2063–2078.
- Legato, P., Canonaco, P., Mazza, R. M. (2009) Yard Crane Management by Simulation and Optimisation. Maritime Economics & Logistics. 11(1):36–57.
- Levitin, G. and Abezgaouz, R. (2003) Optimal routing of multiple-load AGV subject to LIFO loading constraints. Computers & Operations Research. 30:397–410.
- Li, W., Y. Wu, M. E. H. Petering, M. Goh, R. de Souza. (2009) Discrete time model and algorithms for container yard crane scheduling. European Journal of Operational Research. 198 (1):165–172.

- Lim A (1998) The berth planning problem. *Operations Research Letters*. 22: 105–110.
- Lim, A., B. Rodrigues, X. Zhou. (2007) A m-parallel crane scheduling problem with a noncrossing constraint. *Naval Research Logistics*. 54(2): 115–127
- Linn, R., Liu, J.-y., Wan, Y.-w., Zhang, C., and Murty, K. G. (2003) Rubber tired gantry crane deployment for container yard operation. *Computers & Industrial Engineering*. 45(3): 429–442.
- Linn, R. J., and Zhang, C.-Q. (2003) A heuristic for dynamic yard crane deployment in a container terminal. *IIE Transactions*. 35(2): 161–174.
- Meersmans, P. J. M. (2002) *Optimization of Container Handling Systems*, Tinbergen Institute Research Series, No. 271, ISBN 90-5170-591-3, http://repub.eur.nl/pub/1855/fewdis200203_13164327.pdf.
- Meisel, F. (2011) The quay crane scheduling problem with time windows. *Naval Research Logistics*. 58(7): 619–636.
- Mihalis M. Golias, M. Boile, Sotirios Theofanis (2009) Berth scheduling by customer service differentiation: A multi-objective approach. *Transportation Research Part E*. 45: 878–892.
- Moccia, L., Cordeau, J. F., Gaudioso, M. and Laporte, G. (2006) A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal. *Naval Research Logistics*. 53: 45–59
- Nguyen, V. D., Kim, K. H. (2009) A dispatching method for automated lifting vehicles in automated port container terminals. *Computers & Industrial Engineering*. 56(3): 1002–1020.
- Nishimura E, Imai A, Papadimitriou S. (2001) Berth allocation planning in the public berth system by genetic algorithms. *European Journal of Operational Research*. 131: 282–292.
- Nishimura, E., Imai, A., and Papadimitriou, S. (2005) Yard trailer routing at a maritime container terminal. *Transportation Research Part E*. 41:53–76.
- Park Y-M, Kim K. H. (2003) A scheduling method for berth and quay cranes. *OR Spectrum*. 25: 1–23. control systems for seaport container transshipment terminals. *OR Spectrum*. 32: 633–661.
- Park, T., R. Choe, S. M. Ok, K. R. Ryu. (2010) Real-time scheduling for twin RMGs in an automated container yard. *OR Spectrum*. 32: 593–615.
- Peterkofsky R .I., Daganzo C. F. (1990) A branch and bound solution method for the crane scheduling problem. *Transportation Research-B*. 24B: 159–172.
- Petering, M. E. H. (2010) Development and simulation analysis of real-time, dual-load yard truck
- Sammarra, M., Cordeau, J. F., Laporte, G. and Monaco, M. F. (2007) A tabu search heuristic for the quay crane scheduling problem. *Journal of Scheduling*. 10: 327–336.
- Stahlbock R, Voß S. (2008) Operations research at container terminals: a literature update. *OR Spectrum*. 30(1):1–52.

- Stahlbock, R., S. Voß. (2010) Efficiency considerations for sequencing and scheduling of double rail-mounted gantry cranes at maritime container terminals. *International Journal of Shipping and Transport Logistics*. 2: 95–123.
- S.Simon. the law of shipping containers JMLC,1974
- Tavakkoli-Moghaddam, R., Makui, A., Salahi, S., Bazzazi, M., and Taheri, F. (2009) An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports. *Computers & Industrial Engineering*. 56(1), 241–248.
- Terex, <http://www.terex.com/port-solutions/en/>, τελευταία πρόσβαση 10/9/2015.
- Tierney, K., S. Voß, and Stahlbock, R. (2014) A mathematical model of inter-terminal transportation, *European Journal of Operational Research*. 235(2): 448–460.
- URENIO, 200 Cadell Christopher - Falk Nicholas - King Francesca, 2008
- Van der Meer, R. (2000) Operational Control of Internal Transport. TRAIL thesis series nr. t2000/5.
- Vis, I. F. A., Carlo. H. J. (2010) Sequencing two cooperating automated stacking cranes in a container terminal. *Transportation Science*. 44(2): 169–182.
- Xu, D., C. L. Li, J. Y. T. Leung. (2012) Berth allocation with time-dependent physical limitations on vessels. *European Journal of Operational Research*. 216(1) 47–56.
- Yan, W., Huang, Y., Chang, D., and He, J. (2011) An investigation into knowledge-based yard crane scheduling for container terminals. *Advanced Engineering Informatics*. 25(3): 462–471.
- Yuan, S., Skinner, B.T., Huang, S.D., Liu, D.K., Dissanayake, G., Lau, H. (2011) A job grouping approach for planning container transfers at automated seaport container terminals. *Advanced Engineering Informatics*. 25(3): 413–426.
- Zhu, Y., and Lim, A. (2006). Crane scheduling with non-crossing constraint. *The Journal of the Operational Research Society*. 57(12): 1464–1471

ΠΑΡΑΡΤΗΜΑ Α Εγχειρίδιο Χρήσης

Στο συγκεκριμένο παράρτημα γίνεται χρήση αγγλικής ορολογίας, έχοντας αμετάφραστους όρους, προκειμένου να είναι πιο φιλικό στον χρήστη και να έχει πλήρη αντιστοιχία με την ορολογία που θα συναντήσει στο πρόγραμμα.

Τρόπος λειτουργίας και λογική του Anylogic

Τα Anylogic είναι ένα πρόγραμμα προσομοίωσης για την δημιουργία μοντέλων με την χρήση διαφόρων μεθόδων και 3D κινούμενων γραφικών. Στην παρούσα διπλωματική εργασία εξετάζεται η χρήση του Anylogic σε εφαρμογές μεταφοράς και διαχείρισης προϊόντων (transportation applications).

Η λογική με την οποία το Anylogic παράγει μοντέλα είναι η σύνδεση μεταξύ ενός flow chart, το οποίο κατά βάση περιέχει τις διάφορες ενέργειες και εντολές που δίνονται στο προϊόν (πράκτορα), και το animation, στο οποίο γενικά καθορίζονται οι θέσεις και οι αποστάσεις για την κίνηση του προϊόντος.

Προκειμένου να γίνει πιο κατανοητό αναφέρεται το παρακάτω απλό παράδειγμα:

Περιγραφή μοντέλου: ένας πράκτορας εισέρχεται στο σύστημα στην θέση Α, κατευθύνεται στην θέση Β, όπου και περιμένει για ένα χρονικό διάστημα και τέλος κατευθύνεται στην θέση Γ, σημείο στο οποίο βγαίνει από το σύστημα

Τρόπος εισαγωγής του μοντέλου στο Anylogic:

11. Τοποθέτηση του σημείου Α (*animation*)
12. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> εισαγωγής πράκτορα στο σημείο Α (*flow chart*)
13. Τοποθέτηση του σημείου Β (*animation*)
14. Τοποθέτηση του path που ενώνει τα σημεία Α και Β (*animation*)
15. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> μετακίνησης των πρακτόρων στο σημείο Β (*flow chart*)
16. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> αναμονής των πρακτόρων στο σημείο Β (*flow chart*)
17. Τοποθέτηση του σημείου Γ (*animation*)
18. Τοποθέτηση του path που ενώνει τα σημεία Β και Γ (*animation*)
19. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> μετακίνησης των πρακτόρων στο σημείο Γ (*flow chart*)
20. Δημιουργία της κατάλληλης <<προγραμματιστικής μονάδας>> εξαγωγής πρακτόρων από το σύστημα (*flow chart*)

Πλεονεκτήματα και μειονεκτήματα του Anylogic

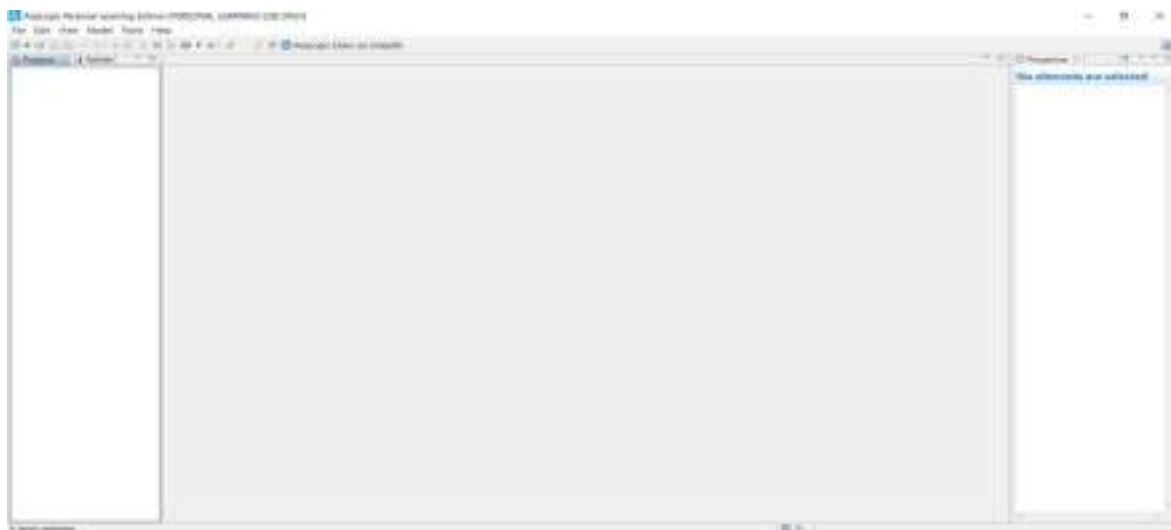
+	-
Υποστηρίζει τρεις τύπους μοντελοποίησης (Agent Based, Discrete Event and System Dynamics)	Δεν είναι ευέλικτο πρόγραμμα
Υποστηρίζει όλους του τύπους προσομοίωσης (Discrete/Continuous, Micro/Macro Level, ...)	Είναι γραμμένο σε Java, προϋποθέτει γνώση προγραμματισμού
Αποτελεί το πρόγραμμα προσομοίωσης με τους πιο γνωστούς πελάτες (Toyota, Panasonic, Rolls-Royce, IBM, Louvre, Intel, Delloite, UEFA, General Motors, Johnson & Johnson, US Navy, Volvo, HP, Red Cross, NASA, ...)	Δύσκολο σε αρχάριο επίπεδο
Υποστηρίζει επεκτάσεις των μοντέλων με την χρήση κώδικα Java	

Εικόνα 123 Πλεονεκτήματα και μειονεκτήματα του Anylogic [Πηγή: Critical Manufacturing S.A. (2014)]

Εισαγωγή στις έννοιες του προγράμματος AnyLogic και πως αυτές χρησιμοποιούνται κατά την μοντελοποίηση

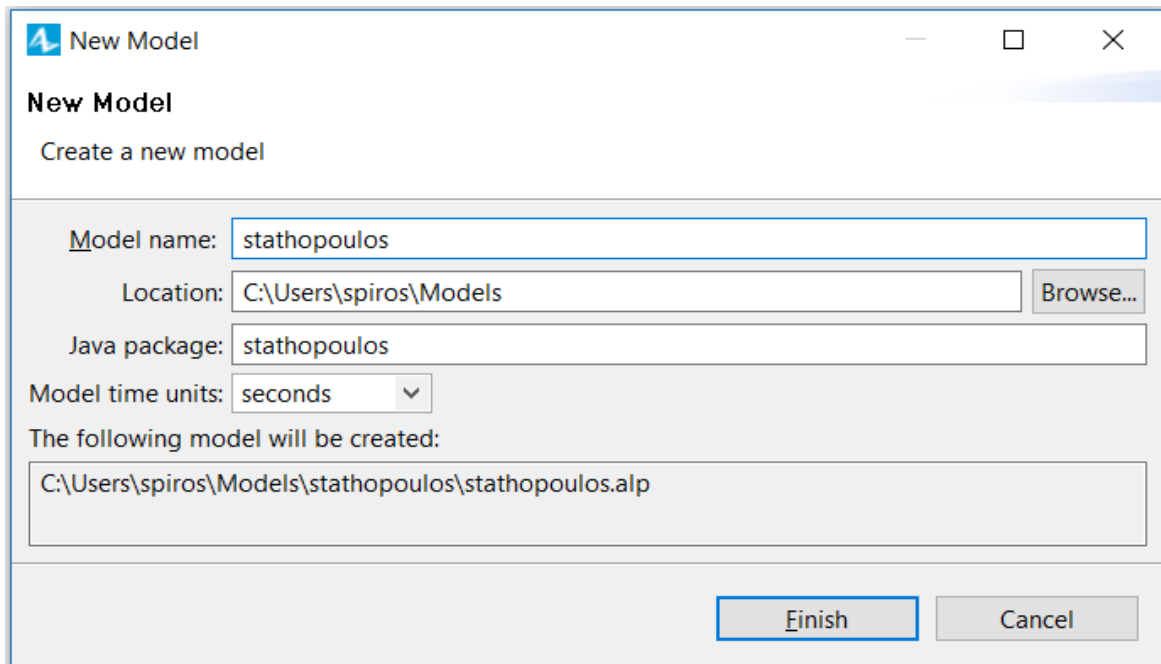
Ανοίγοντας το Anylogic

Η πρώτη εικόνα μπαίνοντας στο πρόγραμμα



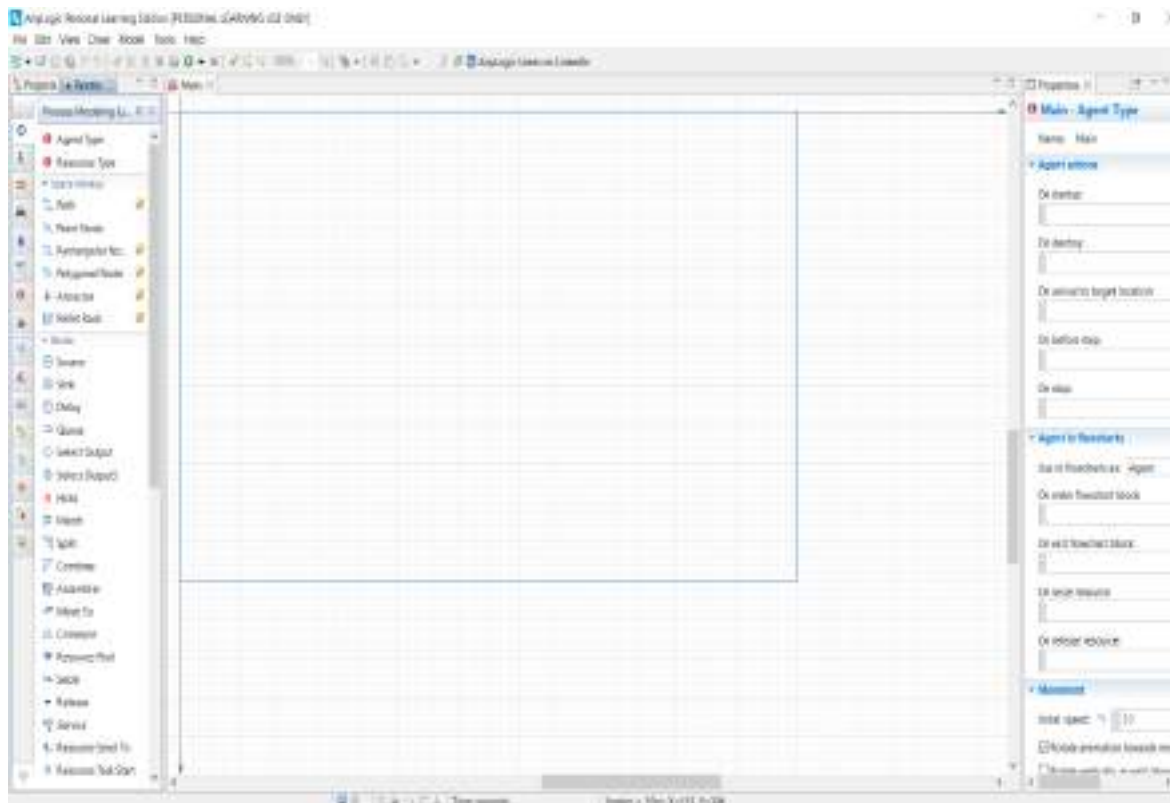
Εικόνα 124 Ανοίγοντας το Anylogic

Πατώντας **file** → **new** → **model** ανοίγει το παρακάτω παράθυρο στο οποίο συμπληρώνουμε το model name ,το location στον δίσκο, το java package και τέλος τα time units που θέλουμε να τρέχει το μοντέλο. Πατάμε **finish**



Εικόνα 125 New model

Έτσι ανοίγει το interface που θα δουλέψουμε το μοντέλο. Στα αριστερά βλέπουμε τις παλέτες όπου είναι τα βασικά εργαλεία του προγράμματος



Εικόνα 126 Interface

Με drag and drop τοποθετούμε τα εργαλεία στο main

Η έννοια του Agent

Ένας agent στο Anylogic μπορεί να αντιπροσωπεύει ποικίλα πράγματα όπως οχήματα, εξοπλισμό, μηχανήματα, προϊόντα, μια ιδέα, έναν οργανισμό, ανθρώπου με ποικίλους ρόλους κ.α.



Εικόνα 127 Agent

Οι Agent είναι ένα από τα βασικά δομικά στοιχεία του AnyLogic. Ο Agent είναι μια μονάδα σχεδιασμού όπου μπορεί να έχει δικιά του συμπεριφορά, μνήμη, επαφές, συγχρονισμό κ.α.

Σε έναν Agent μπορούν να εισαχθούν διάφορες έννοιες που θα εξηγηθούν αναλυτικότερα στην συνέχεια όπως variables, events, state charts κ.α.

Ο σχεδιασμός ενός Agent ξεκινάει από τον καθορισμό των attributes που μπορεί να έχει, της συμπεριφοράς του και την πιθανή διεπαφή του με τους υπόλοιπους Agent του μοντέλου.

Η δημιουργία ενός Agent ξεκινάει κάνοντας drag & drop το στοιχείο **Agent** από την παλέτα Agent στο διάγραμμα. Στην πρώτη σελίδα του wizard που ανοίγει υπάρχει η ερώτηση για το αν θέλεις να δημιουργήσεις έναν εντελώς καινούργιο Agent ή θέλεις να “τραβήξεις” κάποιον από ένα database που υπάρχει ήδη στον υπολογιστή.

New agent

Step 1. Creating new agent type

Agent type name:

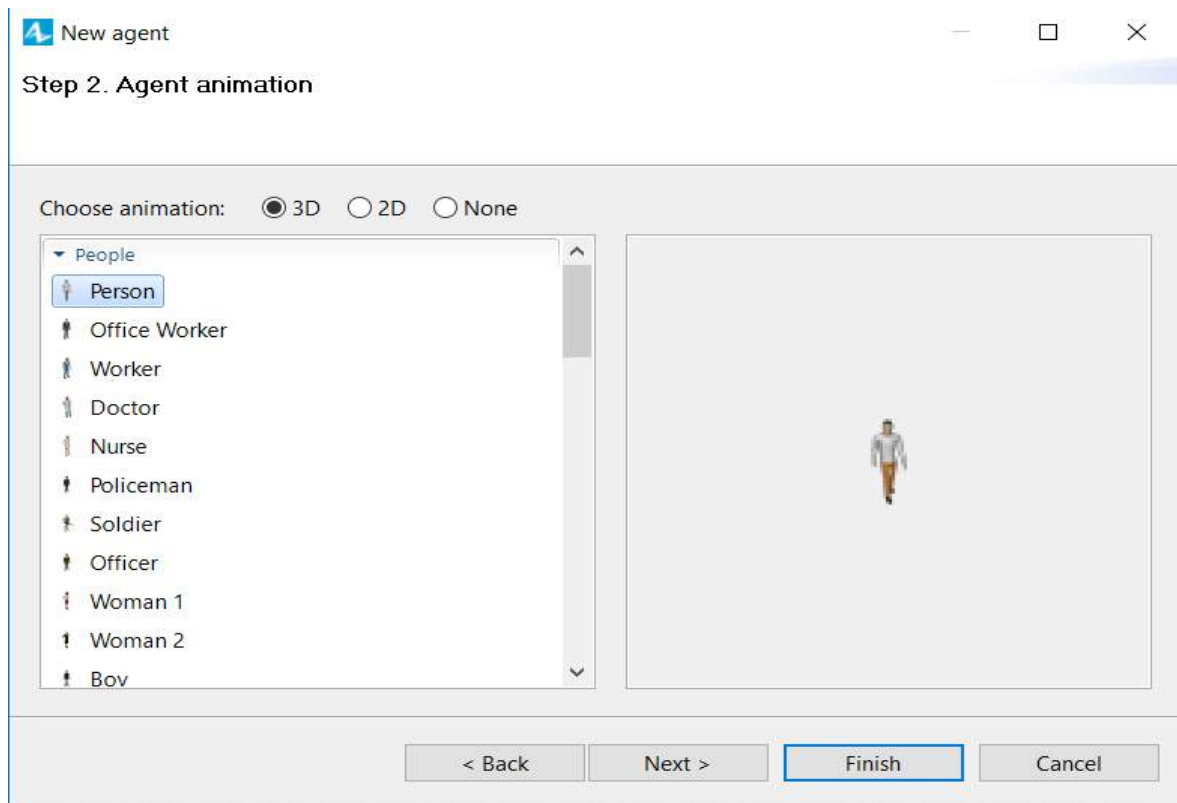
☒ Create the agent type "from scratch"

☐ Use database table
I have agent data stored in a database

< Back Next > **Finish** Cancel

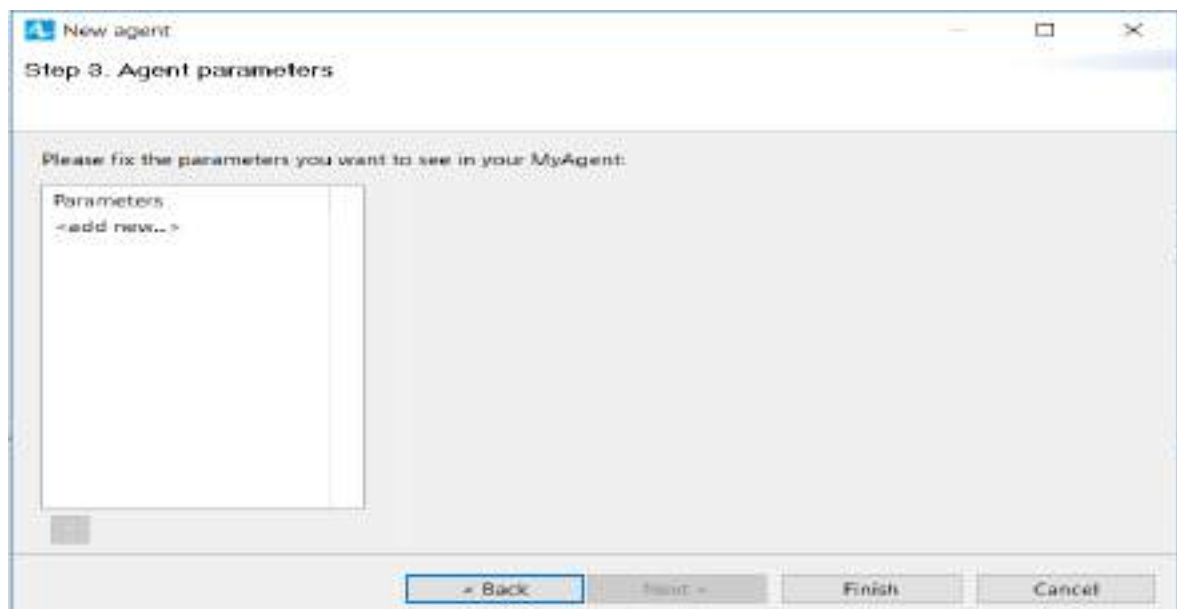
Εικόνα 128 Δημιουργία agent 1

Στην συνέχεια πατάμε το **next** προκειμένου να προχωρήσουμε στο επόμενο βήμα και καθορίσουμε το animation που θέλουμε, διαφορετικά πατάμε **finish** προκειμένου να δημιουργηθεί ένας agent χωρίς animation.



Εικόνα 129 Δημιουργία agent 2

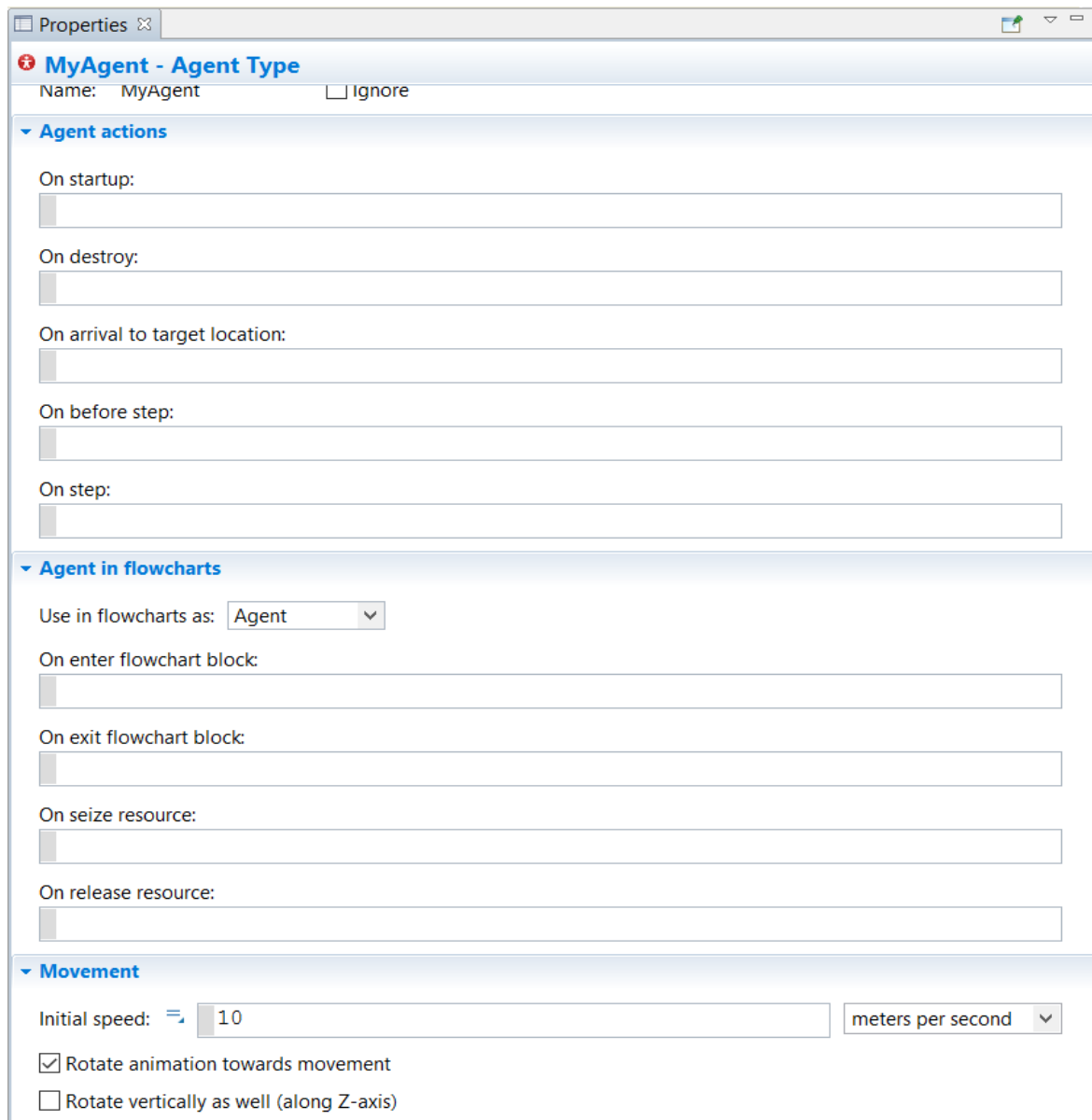
Εφόσον, έγινε η επιλογή και του animation, μπορούμε να πατήσουμε είτε το **next >** που θα μας οδηγήσει στην δημιουργία ορισμένων παραμέτρων που θα θέλαμε να έχει ο Agent ή να πατήσουμε το **Finish** για να κλείσει ο wizard και να ολοκληρωθεί ο Agent.



Εικόνα 130 Δημιουργία agent 3

Σε αυτό το σημείο προτείνεται να μην εισαχθούν από τώρα οι παράμετροι, καθώς είναι μια διεργασία που υπάρχει η δυνατότητα να γίνει και κατά την διάρκεια της μοντελοποίησης.

Έχοντας πλέον δημιουργηθεί ο Agent, κάνοντας κλικ πάνω του έρχονται στα δεξιά της οθόνης τα properties του Agent.



Properties

MyAgent - Agent Type

Name: MyAgent ☐ Ignore

Agent actions

On startup:

On destroy:

On arrival to target location:

On before step:

On step:

Agent in flowcharts

Use in flowcharts as: Agent

On enter flowchart block:

On exit flowchart block:

On seize resource:

On release resource:

Movement

Initial speed: 10 meters per second

☒ Rotate animation towards movement

☐ Rotate vertically as well (along Z-axis)

Εικόνα 131 Agent properties

Σε αυτό το σημείο, μπορούν να καθοριστούν οι ενέργειες των Agent (Agent Actions) σε διάφορες στιγμές κατά την ύπαρξή τους στο simulation. Επίσης μπορούν να δοθούν και εντολές κατά την διάρκεια που ένας Agent βρίσκεται στο flowchart (η έννοια του flowchart περιγράφεται αναλυτικότερα στην ενότητα 2.3.7). Οι εντολές αυτές δύνονται με χρήση κώδικα java που περιγράφεται αργότερα στην ενότητα 2.3.10. Στο movement καθορίζεται η ταχύτητα που κινείται ο Agent όπως και ο τρόπος που θα κάνει τις διάφορες απαιτούμενες περιστροφές κατά την διάρκεια του simulation. Κάνοντας scroll down στις properties εμφανίζονται και άλλες κατηγορίες εντολών που μπορούν να δοθούν στον Agent. Αυτές είναι η χρήση advanced java κώδικα, η κατηγορία advanced που με κλικ στο **log to database** δίνεται η δυνατότητα αποθήκευσης δεδομένων για τον Agent στο database για διάφορες χρονικές στιγμές του Agent στο simulation.

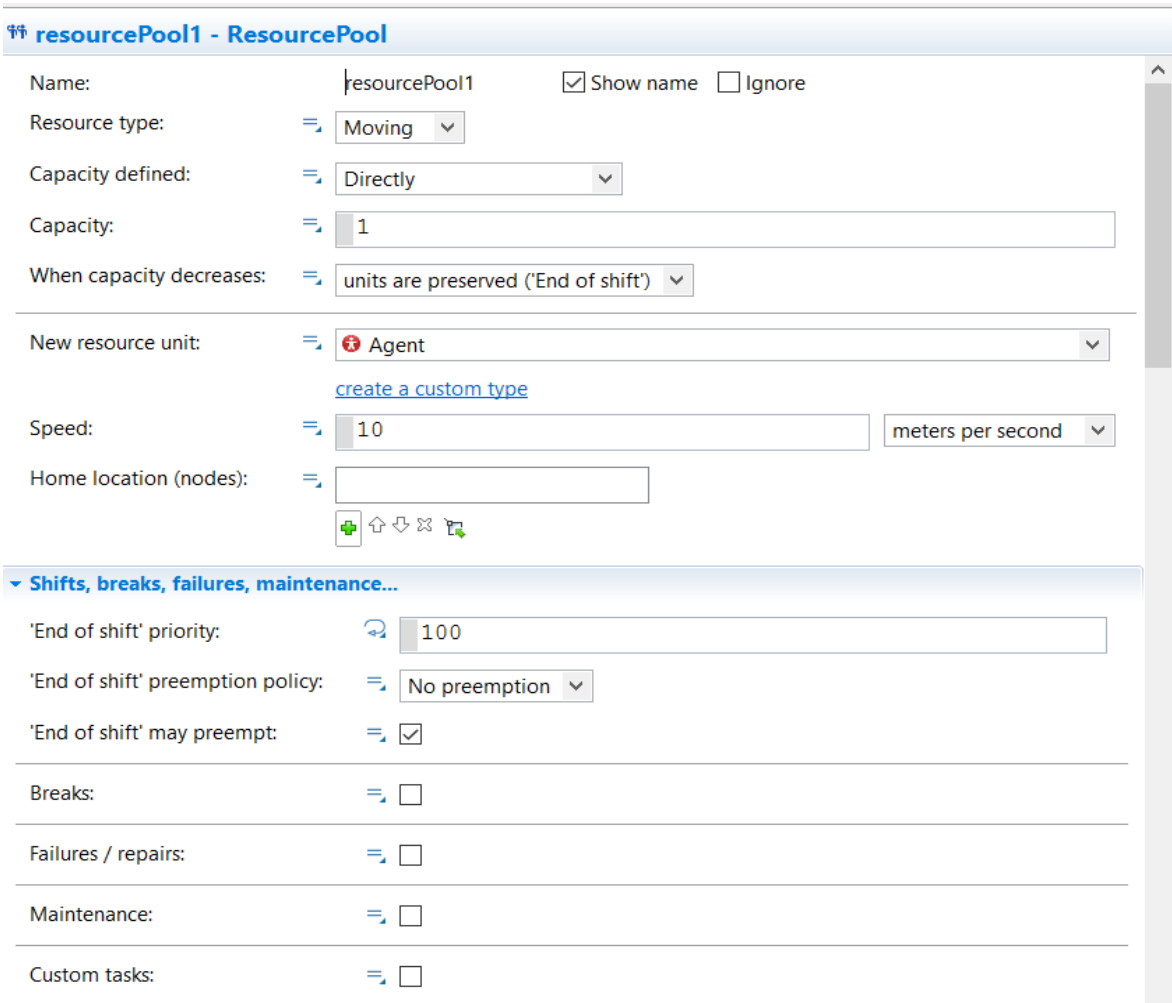
Η έννοια του Resource Pool

Το Resource Pool περιλαμβάνει Resource Units τα οποία διεκπεραιώνουν διεργασίες προκειμένου να εξυπηρετήσουν τον Agent. Παραδείγματα τέτοιων διεργασιών είναι:

- η μεταφορά ενός Agent(που προσομοιώνει για παράδειγμα ένα container) με χρήση ενός Resource Unit (που προσομοιώνει ένα φορτηγό)
- το να γίνει μια ακτινογραφία σε έναν Agent
- ένας άνθρωπος ο οποίος ελέγχει την εγκυρότητα και τα τεχνικά χαρακτηριστικά του Agent προκειμένου να τον κατευθύνει στο σωστό μέρος
- μια γέφυρα η οποία καταρρέει αν υπάρχει μεγάλο βάρος πάνω σε αυτή.

Κάθε Resource Unit μπορεί να είναι είτε σε idle κατάσταση, είτε σε busy, ενώ παράλληλα μπορούν να συλλεχθούν στατιστικά για αυτό με χρήση κώδικα.

Για να εισαχθεί το Resource Pool στο διάγραμμα εφαρμόζεται η διαδικασία Drag & drop του εικονιδίου  Resource Pool στο διάγραμμα. Κάνοντας κλικ πάνω του εμφανίζονται στο δεξί μέρος της οθόνης τα Properties.



resourcePool1 - ResourcePool

Name: resourcePool1 ☒ Show name ☐ Ignore

Resource type: Moving

Capacity defined: Directly

Capacity: 1

When capacity decreases: units are preserved ('End of shift')

New resource unit: Agent [create a custom type](#)

Speed: 10 meters per second

Home location (nodes):

Shifts, breaks, failures, maintenance...

'End of shift' priority: 100

'End of shift' preemption policy: No preemption

'End of shift' may preempt: ☒

Breaks: ☐

Failures / repairs: ☐

Maintenance: ☐

Custom tasks: ☐

Εικόνα 132 Resource Pool

- Στο πεδίο **Resource Type** καθορίζεται το είδος των Resource Units. Αυτά μπορεί να είναι είτε Moving, δηλαδή τα Resource Units να κινούνται σε διάφορα Paths στο διάγραμμα (π.χ. φορτηγά), είτε Portable, δηλαδή τα Resource Units να κινούνται από τους Agents η από άλλα Moving Resources (π.χ. μια αναπηρική καρέκλα), είτε Static, δηλαδή ακίνητα Resource Units (π.χ. ένα X-Ray μηχάνημα).
- Στο πεδίο **Capacity** καθορίζεται ο αριθμός των Resource Units
- Στο πεδίο **New resource unit** δημιουργείται ο Agent που θα αναπαριστά το Resource Unit
- Στο πεδίο **Speed** δίνεται η ταχύτητά του
- Στο πεδίο **Home location** καθορίζεται η τοποθεσία του

Στην κατηγορία **Shifts , breaks, failures, maintenance...** κάνοντας κλικ στα αντίστοιχα καθορίζεται ο κανόνας προτεραιότητας εξυπηρέτησης τους.

Κάνοντας scroll down βλέπου και τις υπόλοιπες Properties

Advanced

Add units to:

☐ default population
☒ custom population

Population:
Force statistics collection:
☐

Actions

On new unit:

On destroy unit:

On seize:

On release:

On wrap-up:

On unit state change:

Advanced

Unit type:

☒ Agent

☐ Single agent
☒ Population of agents

Population is:

☐ Initially empty
☒ Contains a given number of agents
☐ Loaded from database

Initial number of agents:
Model/library:
Process Modeling Library [\(change...\)](#)
Visible:
☒ yes
☐ Visible on upper level
Optimize for:

☒ Access by index (ArrayList)
☐ Add/remove operations (LinkedHashSet)

Εικόνα 133 Resource Pool 2

Σε αυτό το σημείο επιλέγεται το Population που θέλουμε να προσθέσουμε και τον αν χρειάζεται να κρατούνται στατιστικά για αυτόν.

Στα **Actions** με χρήση κώδικα δίνονται διάφορες εντολές. Ο συνηθέστερος κώδικας που χρησιμοποιούμε στο Resource Pool είναι:

```
double utilization() - Returns utilization of this resource pool. The returned value is the mean over all individual unit utilization, calculated from the most recent resetStats() call up to current time.

int idle() - Returns the number of currently idle resource units. (Units out of shift aren't counted).

int busy() - Returns the number of busy units, including those who are in wrap-up/maintenance/break/failure. (Units out of shift aren't counted).

boolean contains(Unit/Agent unit) - Checks whether the resource pool contains the specified unit (Returns true, if yes, false otherwise).

void setShiftGroupCapacity(int id, int capacity) - Sets new size of shift group. Doesn't kill excess resource units. Instead assigns Out-of-shift task to them (idle units are the first to be pushed to Out-of-shift, the next are units with less-priority tasks).

int size() - Returns the total number of units in this resource pool (including those that are out of shift).

int sizeActive() - Returns the total number of resource units in the current shift.

int sizeOfShiftGroup(int id) - Returns the total number of units (including those that are out of shift) in the shift group with the given id.
Parameters: id - the id of shift group (1, 2, 3, ...)

void resetStats() - resets the statistics collected for this object.
```

Η έννοια των μεταβλητών (variables)

Οι variables γενικά χρησιμοποιούνται για την αποθήκευση των αποτελεσμάτων των μοντέλων ή για την μοντελοποίηση κάποιων συγκεκριμένων δεδομένων ή για την αλλαγή κάποιων χαρακτηριστικών στην πάροδο του χρόνου.

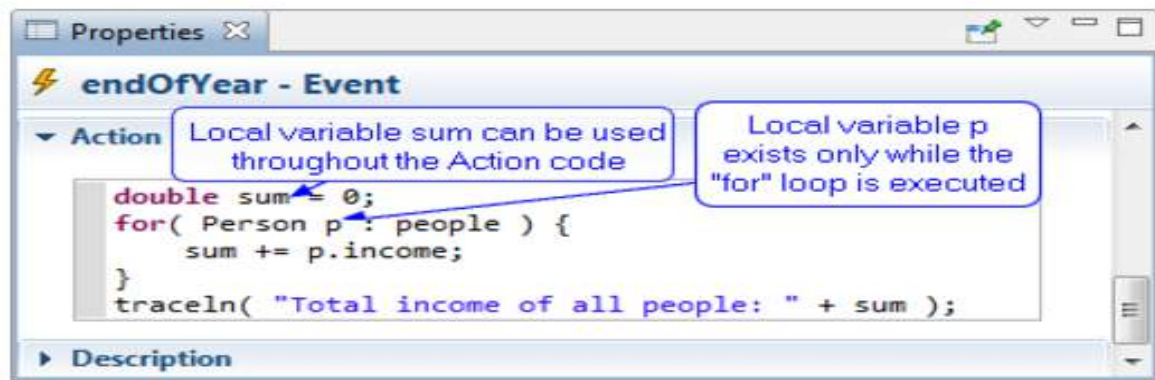
Μια variable μπορεί να είναι ένας βαθμωτός τύπος που παίρνει διάφορες τιμές εν εξελίξει του μοντέλου ή μια Java class. Επίσης πρέπει πάντα να έχει μια τιμή για αυτό είναι αναγκαίο να καθορίζεται η αρχική της τιμή (Initial value). Αν δεν έχει γίνει κάτι τέτοιο, το πρόγραμμα αρχικοποιεί μια τυχαία τιμή.

Οι variables μπορούν να είναι διαφόρων τύπων. Οι πιο συνηθισμένοι είναι:

- Double : παρέχεται η δυνατότητα εισαγωγής δεκαδικών αριθμών
- Int : παρέχεται η δυνατότητα εισαγωγής μόνο ακεραίων αριθμών
- Boolean : παρέχεται η δυνατότητα εισαγωγής μόνο TRUE ή FALSE
- String : παρέχεται η δυνατότητα εισαγωγής κειμένου σε “.....”

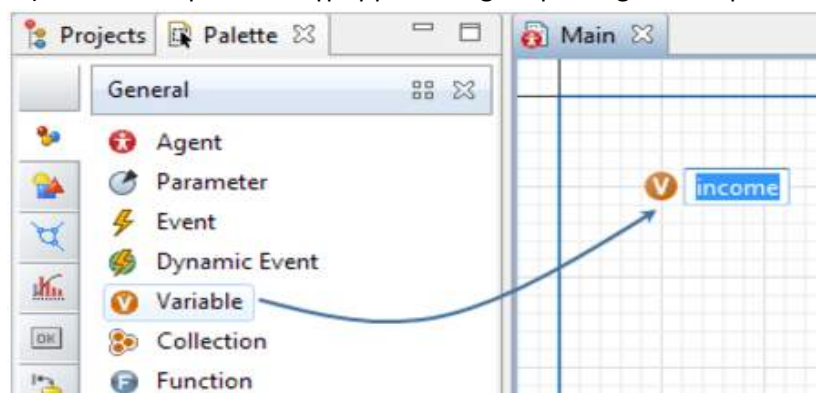
Επίσης ανάλογα με το πού βρίσκεται αυτή η variable μπορεί να είτε Local variable είτε Class variable.

- Local variable: βρίσκεται σε κομμάτια εντός του Java κώδικα όπως σε loops, functions ή κάποιο block. Πρακτικά δημιουργείται μέσα στον κώδικα, αρχικοποιείται και τέλος εξαφανίζεται κατά το τέλος του. Ένα παράδειγμα είναι:



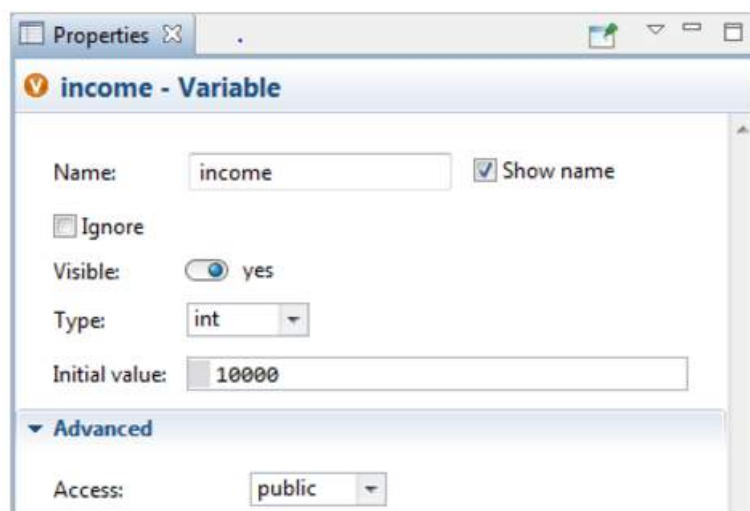
Εικόνα 134 Παράδειγμα κώδικα

- Class variable: μια class variable είναι μέρος της μνήμης ή της κατάστασης των Agents. Την τοποθετούμε στο διάγραμμα του Agent με drag and drop



Εικόνα 135 Μεταβλητή

και καθορίζουμε τα χαρακτηριστικά τις από τα properties που εμφανίζονται πατώντας πάνω της



Εικόνα 136 Μεταβλητή

Στο type καθορίζουμε τον τύπο της μεταβλητής μέσα από μια πληθώρα επιλογών που εμφανίζονται πατώντας το βελάκι στα δεξιά ενώ στο initial value συμπληρώνουμε την αρχική της τιμή.

Ο τρόπος που αυτή η μεταβλητή αλλάζει τιμές είναι με χρήση κώδικα στα διάφορα Actions των block έχουμε εισάγει από το διάγραμμα.

Η έννοια της παραμέτρου (parameter)

Συνήθως οι Agents έχουν parameters από τους οποίους καθορίζεται η συμπεριφορά τους στο μοντέλο. Συνήθως χρησιμοποιούνται προκειμένου να καθοριστεί ένα συγκεκριμένου χαρακτηριστικό τους. Η διαφορά μεταξύ μιας Parameter και μιας Variable είναι ότι, η Variable αντιπροσωπεύει μια κατάσταση (π.χ. σωστό-λάθος) και συνήθως αλλάζει κατά την διάρκεια του simulation, ενώ η Parameter χρησιμοποιείται συνήθως για να περιγράψει ένα αντικείμενο (ή ομάδα αντικειμένων π.χ. Agent) στατικά. Μια Parameter αποτελεί κατά βάση ένα στατικό μέγεθος σε ένα simulation, και αλλάζει μόνο όταν χρειάζεται να γίνει αλλαγή στην συνολική συμπεριφορά του μοντέλου. Ο τρόπος για να γίνει μία τέτοια αλλαγή είναι να γίνει εισαγωγή της σε κάποιο Action.

Για να τοποθετήσουμε μια parameter στο διάγραμμα κάνουμε Drag & Drop τι εικονίδιο στο  Parameter διάγραμμα



Εικόνα 137 Παράμετρος

και κάνοντας κλικ πάνω του βλέπουμε τις Properties.

acceleration - Parameter

Name: ☒ Show name ☐ Ignore

Visible: ☒ yes

Type:

Unit:

Default value:

☐ System dynamics array

Value editor

Label:

Control type:

Hide conditions:

Parameter	Condition	Value

Advanced

☒ Static ☐ Dynamic ☐ Action

☐ System dynamics units:

☐ Save in snapshot

On change:

Units numeric type:

Description

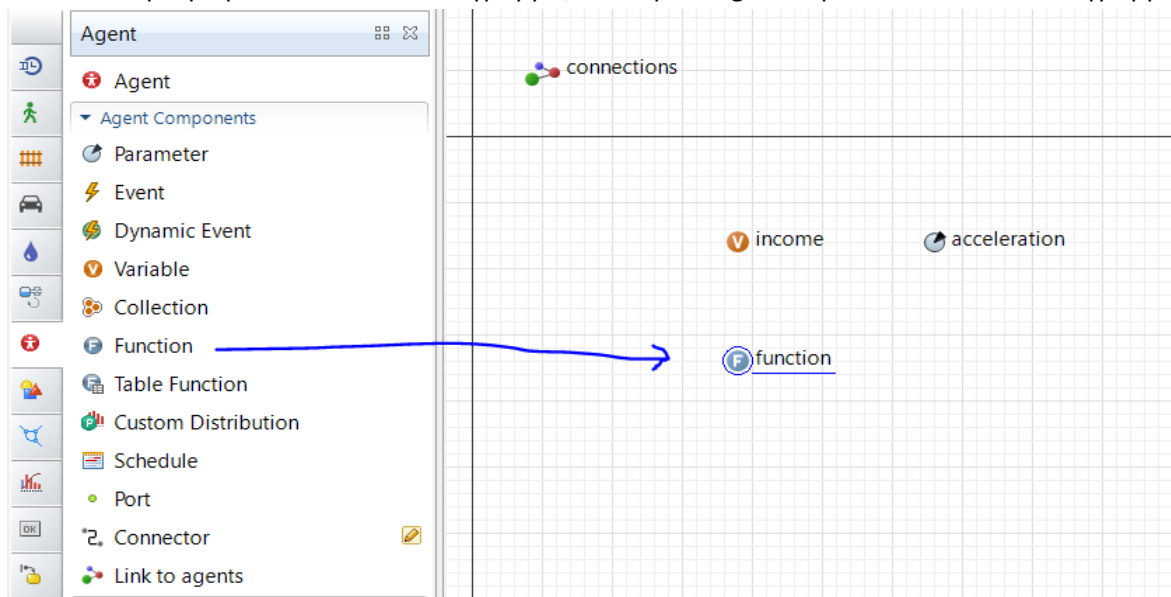
Εικόνα 138 Παράμετρος

Στο παράδειγμά βλέπουμε μια παράμετρο η οποία καθορίζει την επιτάχυνση επιθυμούμε να έχει ο Agent επιλέγοντας στο Type το Acceleration. Στο unit επιλέγεται η μονάδα μέτρησης που επιθυμείται, ενώ στο default value η τιμή της. Στο value editor δύνεται η δυνατότητα επικοινωνίας της παραμέτρου με άλλες παραμέτρους, προκειμένου να αλλάζει τιμή υπό την συνθήκη που καθορίζεται.

Η έννοια της συνάρτησης (Function)

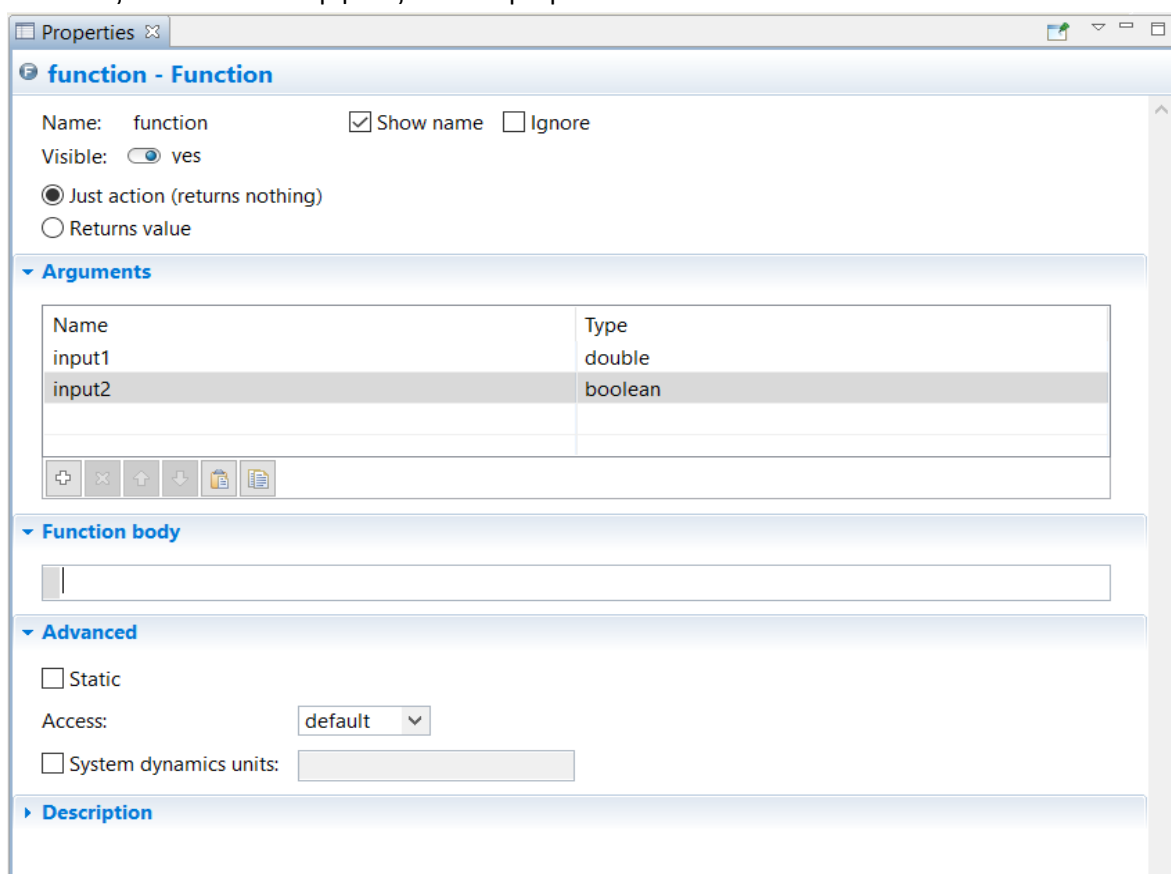
Η function επιστρέφει την τιμή μιας expression κάθε φορά που αυτή καλείται. Μέσα σε μια function γίνεται χρήση κώδικα java, κάτι που παρέχει όλα τα πλεονεκτήματα που μπορεί να παρέχει ο κώδικας, όπως χρήση conditional (συνήθως με χρήση του if), χρήση της επαναληπτικής διαδικασίας κ.α.

Για να εισάγουμε μια function στο διάγραμμα, κάνουμε drag & drop το εικονίδιο στο διάγραμμα



Εικόνα 139 Συνάρτηση 1

Κάνοντας κλικ πάνω του εμφανίζονται τα properties



Εικόνα 140 Συνάρτηση 2

Μια function μπορεί είτε να κάνει μια Action, είτε να επιστρέφει κάποια τιμή. Στην κατηγορία Arguments συμπληρώνουμε τα input που θα έχει η function και τον τύπο τους. Στο function body

γράφεται ο απαιτούμενος κώδικας σε μορφή java. Στην περίπτωση που η function επιστρέφει μια τιμή, αυτό γίνεται με την εντολή return (....)

Για να καλέσουμε μια function, γράφουμε το όνομα της function και σε παρένθεση της τιμές που θέλουμε να πάρει σαν input.

Actions

On enter:

`function(5, true);`

On at exit:

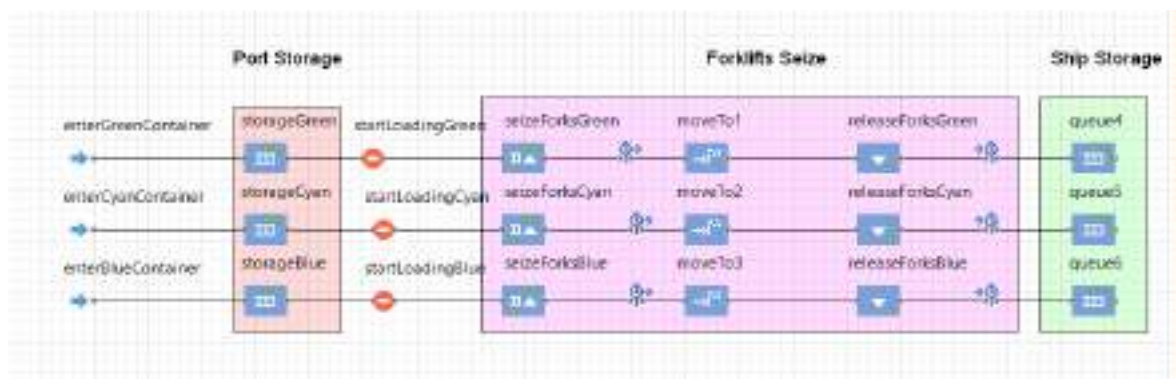
On exit:

On remove:

Εικόνα 141 Συνάρτηση 3

Η έννοια του flow chart και του animation

Το flow chart αποτελεί τον θεμέλιο λίθο ενός simulation, καθώς εκεί καθορίζεται η κίνηση ή οι actions των agent κατά την διάρκεια της εξέλιξής του. Για την διαμόρφωση του flow chart γίνεται εισαγωγή διαφόρων block από την παλέτα προκειμένου να διαμορφωθεί το μοντέλο. Τα περισσότερα block έχουν κάποια inputs, όπου ο Agent εισάγεται στο block, όπως και outputs, όπου εξέρχονται προκειμένου να συνεχίσει η πορεία του Agent. Εισάγοντας ένα block (με την διαδικασία drag & drop) και πλησιάζοντας το στα υπόλοιπα, είναι προφανές ότι αυτό συνδέεται με τα ήδη υπάρχοντα block αυτόματα με connectors, κάτι που καθορίζει την συνέχεια και την εξέλιξη του flow chart. Ένα παράδειγμα είναι

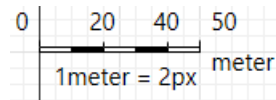


Εικόνα 142 Διάγραμμα ροής

Όπου τα containers εισάγονται με την χρήση του block enter, αποθηκεύονται στα storages, στην συνέχεια υπάρχει η ύπαρξη ενός block το οποίο σταματάει την κίνηση μέχρι μια condition (γίνεται παύση της κίνησης των agent μέχρι να παραβάλει το καράβι) να το κάνει unblock, στην συνέχεια με το block seize καλούνται τα περνοφόρα να φορτώσουν τα container προκειμένου με το block moveTo να μεταφερθούν στο πλοίο και με το release να τα αφήσουν. Τέλος τα containers αποθηκεύονται στο storage του πλοίου. Στην συνέχεια παραβάλλεται αναλυτικό παράδειγμα step by step για το πώς κατασκευάζεται ένα απλό μοντέλο.

Σχετικά με το animation, γίνεται ο καθορισμός των σημείων που βρίσκονται οι Agent αλλά και τα paths που χρησιμοποιούνται προκειμένου να φτάσει ο Agent στα προκαθορισμένα σημεία (στην

περίπτωση που δεν χρησιμοποιηθούν Paths, ο Agent πηγαίνει στα αντίστοιχα σημεία με straight movement. Πρακτικά, το animation λειτουργεί παράλληλα με το flow chart, ενώ οι αποστάσεις που πλέον αναγκάζονται να διανύσουν οι Agents λειτουργούν κατά κάποιο τρόπο σαν καθυστερήσεις προσομοιώνοντας τον χρόνο μετακίνησης από το ένα σημείο στο άλλο (στην περίπτωση που δεν επιθυμείται η χρήση animation, η μεταφορά γίνεται στιγμιαία, ενώ το Delay θα μπορεί να καθοριστεί μέσα από τα αντίστοιχα block). Οι αποστάσεις καθορίζονται από τα pixels, τα οποία μέσω της κλίμακας αναπαριστούν τα αντίστοιχα μέτρα ή χιλιόμετρα κτλ. Η κλίμακα φαίνεται από το εικονίδιο



Όπου κάνοντας κλικ πάνω ανοίγουν οι Properties όπου δίνεται η δυνατότητα καθορισμού της κλίμακας ανά pixel.

scale - Scale

Name: scale

☐ Visible on upper level ☐ Lock

Visible: ☐ no

Length, pixels: 100

Scale is: ☒ Defined graphically ☐ Specified explicitly

Ruler length corresponds to: 50 meter

Position

X: 0

Y: -150

Rotation, °: 0.0

Advanced

Show in: ☐ 2D and 3D ☒ 2D only ☐ 3D only

Description

Εικόνα 143 Scale

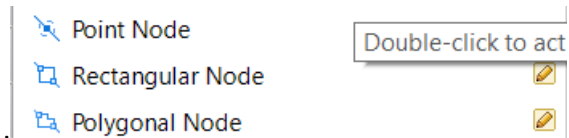
Κάνοντας κλικ στο specified explicitly. Βέβαια αυτός ο καθορισμός μπορεί να γίνει και στο γραφικό περιβάλλον, σύροντας το δεξί άκρο του εικονιδίου.

Στην επόμενη ενότητα παραβάλλονται τα συχνότερα blocks και animation εξηγώντας την βασική τους ιδέα και πώς αυτά χρησιμοποιούνται.

Επεξήγηση βασικών εργαλείων από τις παλέτες

- Process Modeling Library

1.



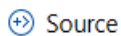
Τοποθέτηση σημείων στο σύστημα. Δίνεται η δυνατότητα για drawing mode

2.



Τοποθέτηση σημείου στοιβάσις. Συνδυάζεται με τις εντολές Rack Store και Rack pick. Παροχή δυνατότητας ελέγχου του τύπου, του αριθμού των κελιών, του βάθους, των επιπέδων, του ύψους κ.α.

3.



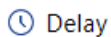
Εισαγωγή των agent στο σύστημα. Παροχή δυνατότητας ελέγχου τρόπου εισαγωγής των agent, την τοποθεσία, την ταχύτητα κ.α.

4.



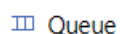
Εξαγωγή των agent από το σύστημα.

5.



Συνήθως περιγράφει διαδικασίες processing. Δυνατότητα αναμονής μέχρι να δοθεί εντολή με χρήση κώδικα ή με καθορισμένο χρόνο ο οποίος μπορεί να προέρχεται και από κατανομές. Επίσης καθορίζεται ο αριθμός των agent που βρίσκονται μέσα σε αυτό το block, η τοποθεσία τους κ.α.

6.



Παρέχεται δυνατότητα αναμονής των agent σε μια ουρά. . Παροχή δυνατότητας ελέγχου του μέγιστου αριθμού των agent που βρίσκονται σε αυτό το block, η τοποθεσία τους, ο τρόπος εξαγωγής κ.α.

7.

 Select Output

 Select Output5

Υπαρξη μιας εισόδου και δύο η περισσότερων εξόδων. Η επιλογή της εξόδου γίνεται είτε με πιθανότητες είτε με προϋπόθεση τύπου σωστό/λάθος.

8.

 Hold

Διακόπτει την κίνηση των agent αν είναι blocked. Η εναλλαγή της κατάστασής του από block σε unblock γίνεται είτε manual, είτε αυτόματα, είτε με χρήση προϋπόθεσης (condition).

9.

 Assembler

Υπαρξη πέντε εισόδων και μιας εξόδου. Παροχή δυνατότης λήψης διαφόρων ποσοτήτων ανά είσοδο, χρήση resource pool, παρέχει δυνατότητα αναμονής προσομοιάζοντας διαδικασίες processing κ.α.

10.

 Move To

 Conveyor

Χρήση για μεταφορά του agent από το ένα σημείο στο άλλο. Η χρήση το Conveyor γίνεται για να προσομοιωθεί η μεταφορά με χρήση διαδρόμου.

11.

 Resource Pool

Περιλαμβάνει αντικείμενα τα οποία βοηθούν τους agent για να γίνουν κάποιες διαδικασίες. Παρέχεται δυνατότητα ελέγχου του αριθμού τους, της ταχύτητάς του, της αρχικής τοποθεσίας κ.α.

12.

 Seize

 Release

 Service

Με την χρήση του Seize εισέρχεται στο flow chart ένα αντικείμενο του Resource Pool προκειμένου να βοηθήσει τον agent να εκτελέσει κάποια διαδικασία. Με το Release εξέρχεται και συνεχίζεται το flow chart χωρίς το βοηθητικό αντικείμενο. Η χρήση του Service αντικαθιστά τρία block, Seize Delay και Release.

13.

 Dropoff

 Pickup

Το εργαλείο Pick up έχει δύο εισόδους όπου δύο διαφορετικοί agent εισέρχονται στο block, και κατά στην έξοδο κινούνται μαζί πλέον. Με την χρήση του drop off ξεχωρίζονται.

14.

 Rack Store

 Rack Pick

Με την χρήση του block Rack Store δύναται η δυνατότητα τοποθέτησης των agent στο Pallet Rack με χρήση αντικειμένων που βρίσκονται στο Resource Pool. Με το Rack Pick αφαιρούνται οι agent από το Pallet Rack.

- Agent library

1.

 Parameter

Με την χρήση αυτού του block εισάγουμε στον agent παραμέτρους προκειμένου να αποκτήσει κάποιες ιδιότητες και κατά το flow chart να κινείται ανάλογα. Γίνεται επεξεργασία της παραμέτρου κυρίως με την χρήση κώδικα java. Παρέχεται δυνατότητα επιλογής του τύπου (double, int, string κ.α.)

2.

 Variable

Με την χρήση αυτού του block τοποθετούμε μία μεταβλητή. Παρέχεται δυνατότητα επιλογής του τύπου (double, int, string κ.α.)

3.

 Function

Με την χρήση του Function δίνεται η δυνατότητα επεξεργασίας των παραμέτρων και μεταβλητών. Απαραίτητη είναι η χρήση κώδικα java.

4.

 Statechart Entry Point

 State



 Transition



 Branch

 Final State

 Initial State Pointer

 History State

Παρέχεται η δυνατότητα χρήσης State chart για διαδικασίες που αφορούν τον agent. Καλείται εύκολα μέσω εντολών στο Action στο εκάστοτε block.

- Presentation library

1.

 Line	
 Polyline	
 Curve	
 Rectangle	
 Rounded Rectangle	
 Oval	
 Arc	
 Pixel	

Με τα παραπάνω εργαλεία εισάγονται στοιχεία σχεδιασμού. Στα line, polyline, curve, rectangle, rounded rectangle και oval παρέχεται drawing mode

2.

 Text

 Image

Δυνατότητα εισαγωγής κειμένου με την χρήση του text για επεξήγηση διαφόρων διαδικασιών και να γίνει το μοντέλο πιο κατανοητό στον απλό χρήστη. Με την χρήση του image τοποθετείται εικόνα στο background προκειμένου να γίνει πιο ρεαλιστικό το μοντέλο.

3.

 3D Window

Γίνεται χρήση του 3D Window προκειμένου κατά την διαδικασία που τρέχει το μοντέλο να υποστηρίζεται και 3D απεικόνιση.

- Space Mark up library

1.

 GIS Map

Με drag and drop εισάγεται χάρτης GIS

2.

 GIS Point

Με drag and drop μέσα στον χάρτη εισάγεται ένα σημείο

3.

Route Provider

Με την χρήση του Route Provider καθορίζεται η διαδρομή που θα ακολουθήσει ο agent στο περιβάλλον του GIS

- Analysis library

1.

 Bar Chart

 Stack Chart

 Pie Chart

Εισάγοντας τα παραπάνω Chart μπορούμε να οπτικοποιήσουμε τα δεδομένα που συλλέγονται κατά την διαδικασία της προσομοίωσης. Η συλλογή τους γίνεται με κώδικα java.

2.

 Time Plot

 Time Stack Chart

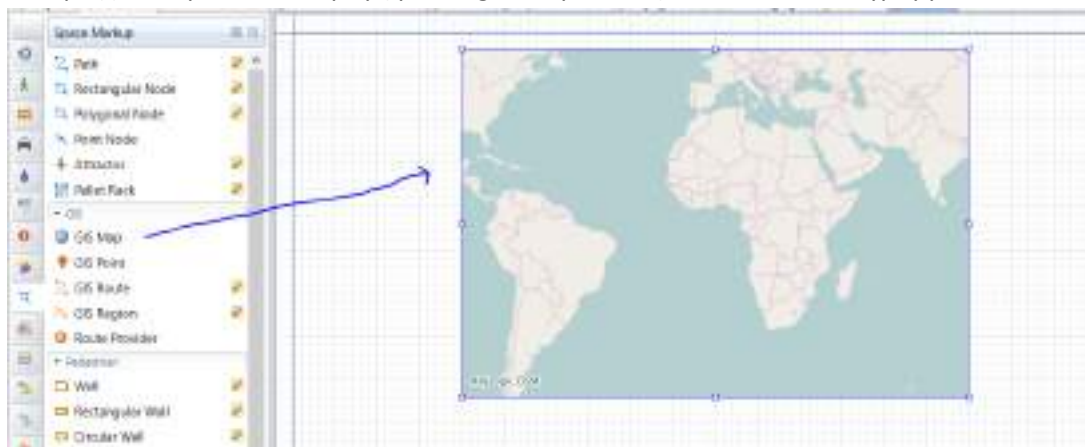
 Time Color Chart

Εισάγοντας τα παραπάνω Chart ή Plot μπορούμε να οπτικοποιήσουμε τα δεδομένα που συλλέγονται κατά την διαδικασία της προσομοίωσης εν εξελίξει του χρόνου. Η συλλογή τους γίνεται με κώδικα java.

Η έννοια του GIS στο Anylogic

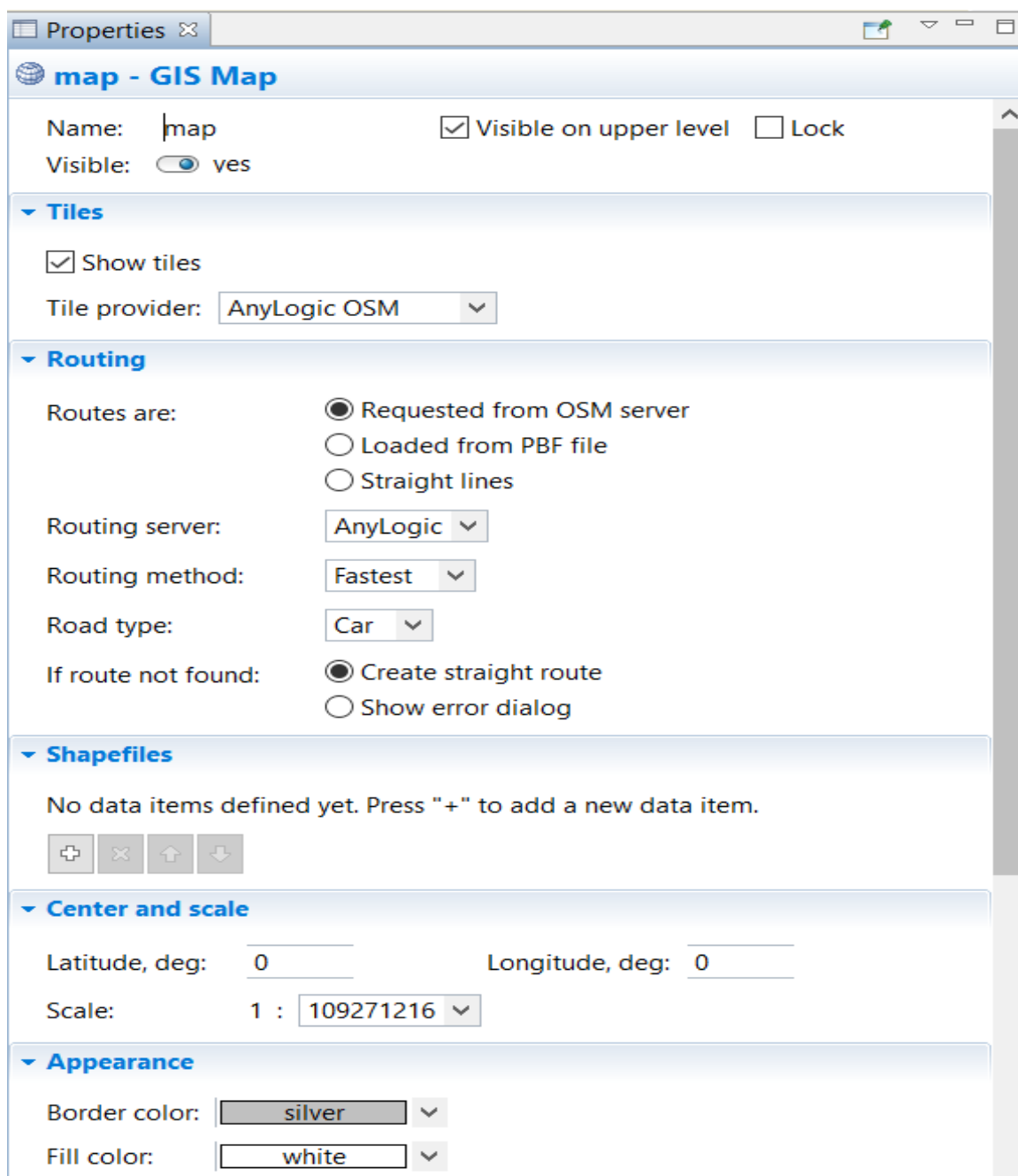
Στο AnyLogic δίνεται η δυνατότητα χρήσης χαρτών GIS προκειμένου οι Agents να μπορούν να κουνηθούν σε ένα GIS space. Η λογική είναι παρόμοια με αυτή της κίνησης στο απλό γραφικό περιβάλλον του Anylogic, με την διαφορά ότι παρέχονται και άλλες δυνατότητες τοποθετώντας τον Agent στον παγκόσμιο χάρτη. Είναι ευκολότερη η κατανόηση των δυνατοτήτων αυτών, εξηγώντας τις παροχές που μας δίνουν τα εργαλεία από τις παλέτες που αφορούν το GIS.

- Ο χάρτης GIS: γίνεται εισαγωγή με Drag &Drop από την παλέτα στο διάγραμμα



Εικόνα 144 GIS 1

Κάνοντας κλικ πάνω του εμφανίζονται στα αριστερά οι Properties



Properties ✕

map - GIS Map

Name: ☒ Visible on upper level ☐ Lock

Visible: ☒ yes

▼ Tiles

☒ Show tiles

Tile provider: ▼

▼ Routing

Routes are: ☒ Requested from OSM server
☐ Loaded from PBF file
☐ Straight lines

Routing server: ▼

Routing method: ▼

Road type: ▼

If route not found: ☒ Create straight route
☐ Show error dialog

▼ Shapefiles

No data items defined yet. Press "+" to add a new data item.

▼ Center and scale

Latitude, deg: Longitude, deg:

Scale: 1 : ▼

▼ Appearance

Border color: ▼

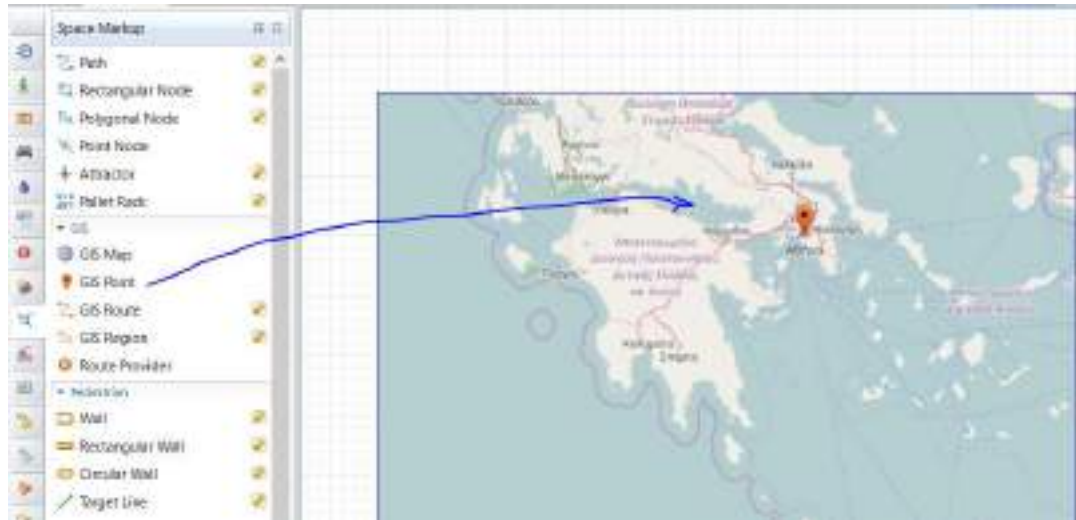
Fill color: ▼

Εικόνα 145 GIS 2

Στο tile provider παρέχεται η δυνατότητα να διαλέξουμε ανάμεσα σε πολλές επιλογές. Το AnyLogic προτείνει το AnyLogic OSM. Σχετικά με το Routing , δίνεται η δυνατότητα επιλογής εκ των Request from OSM server, Loaded from PBF file, η να δημιουργήσουμε Straight lines. Στην συνέχεια, πάλι δύνεται η δυνατότητα πολλών επιλογών server, με το AnyLogic να προτείνει τον δικό του server. Σχετικά με το Routing Method, παρέχεται η δυνατότητα επιλογής ενός εκ των fastest (γρηγορότερη διαδρομή) και shortest (συντομότερη διαδρομή). Στο road type επιλέγεται ο by default τύπος της διαδρομής, δίνοντας την δυνατότητα επιλογής ενός εκ των, car (εδώ χρησιμοποιείται το οδικό δίκτυο), rail (εδώ χρησιμοποιείται το σιδηροδρομικό δίκτυο) , bike (με χρήση ποδηλάτου) και foot (χωρίς χρήση μεταφορικού μέσου) . Στην περίπτωση που δεν είναι

δυνατό να βρεθεί η διαδρομή, παρέχεται η δυνατότητα επιλογής αν επιθυμείται η δημιουργία μιας straight route ή εμφάνισης error. Σχετικά με το Shape file, δίνεται η δυνατότητα να 'τραβήξουμε' κάποιο data το οποίο είναι αποθηκευμένο στον υπολογιστή. Στους παρεχόμενους χάρτες όλοι οι δρόμοι είναι διπλής κατεύθυνσης, χωρίς όρια ταχύτητας κτλ. Εάν επιθυμούμε να καθορίσουμε πιο συγκεκριμένα τις ιδιότητες της διαδρομής αυτό γίνεται με την χρήση του προγράμματος GIS και φορτώνοντάς αυτές στο AnyLogic.

- GIS point: με Drag & Drop τοποθετούμε το GIS point πάνω στον GIS χάρτη.



Εικόνα 146 GIS 3

- RoutProvider: για να γίνει χρήση του RoutProvider με Drag & Drop τοποθετούμε στο διάγραμμα το αντίστοιχο εργαλείο, όπως και τον Agent που θέλουμε να καθοριστεί ο τρόπος κίνησης του.



Εικόνα 147 GIS 4

- κάνοντας κλικ στο RoutProvider ανοίγουν οι Properties στα αριστερά της οθόνης

routeProvider - Route Provider

Name: routeProvider ☐ Ignore

☒ Visible on upper level

Visible: ☒ yes

Routes are: ☒ Requested from OSM server
☐ Loaded from PBF file
☐ Straight lines

Routing server: AnyLogic ▼

Routing method: Fastest ▼

Road type: Car ▼

If route not found: ☒ Create straight route
☐ Show error dialog

▶ **Advanced**

▶ **Description**

Εικόνα 148 RoutProvider

Όπου μπορούμε να αλλάξουμε τις by default ιδιότητες, όπως τις έχουμε ορίσει στον GIS map. Σε αυτό το σημείο γίνεται επαναπροσδιορισμός των ιδιοτήτων με την αντίστοιχη διαδικασία που έγινε και στον GIS map. Στην συνέχεια πρέπει να καθορίσουμε τον Agent που θα ακολουθήσει αυτές τις οδηγίες κάνοντας κλικ στον Agent που κάναμε Drag & Drop.

Properties ⓘ

myAgent - MyAgent

Name: myAgent ☒ Show name

☐ Ignore

☒ Single agent ☐ Population of agents

▼ Movement

Initial speed: 10 meters per second ▼

▼ Initial location

Place agent(s): ☒ at the agent animation location
☐ in the latitude/longitude
☐ in the node
☐ in the first result of map search

Routing: routeProvider ▼

▼ Statistics

Statistics properties are only available for population.
Switch this object to Population of agents.

▼ Advanced

Model/library: paradeigma GIS [\(change...\)](#)

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database
[Turn on model execution logging](#)

Show presentation

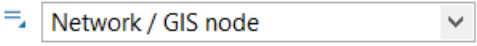
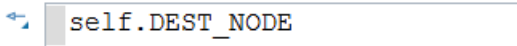
► Description

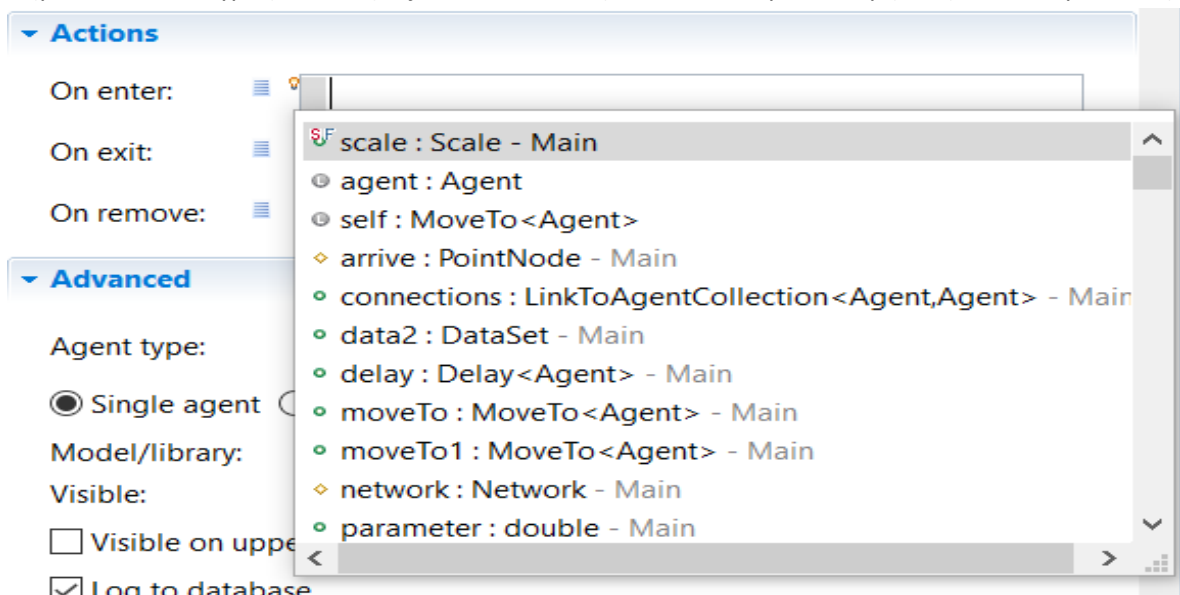
Εικόνα 149 GIS-Agent

Και επιλέγοντας στο Routing τον Route Provider που έχει δημιουργηθεί.

Η χρήση της Java στο μοντέλο

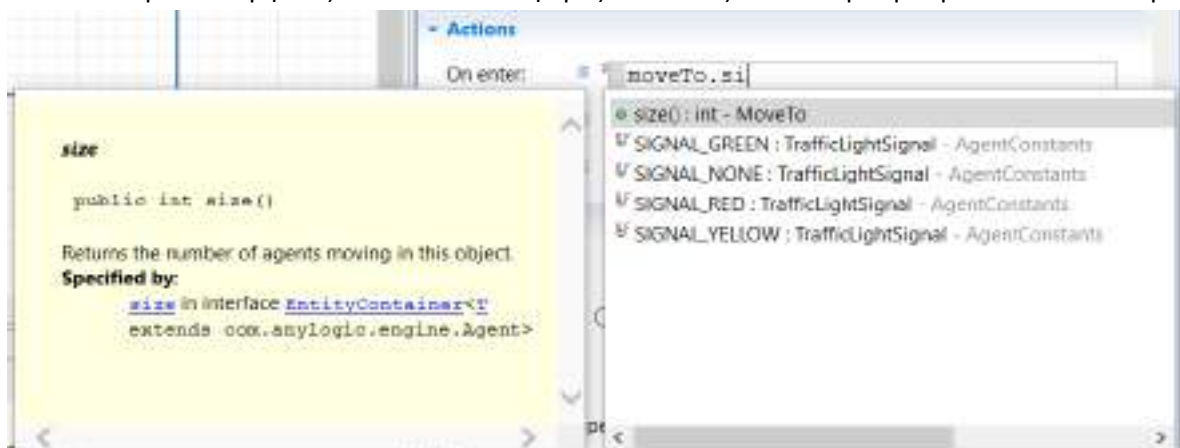
Το πρόγραμμα AnyLogic παρέχει την δυνατότητα χρήσης κώδικα Java σε κάθε πιθανή επιλογή που μπορούμε να κάνουμε. Υπάρχουν κάποια σημεία (όπως τα Action ή το Body function) το οποίο απαιτεί μόνο χρήση κώδικα αλλά και σημεία όπου παρέχεται δυνατότητα επιλογής είτε χρήση κάποιων επιλογών που παρέχονται από το πρόγραμμα, είτε μια πιο σύνθετης επιλογής με χρήση κώδικα. Αυτό μπορεί να γίνει κάνοντας κλικ στο '=' που εμφανίζεται σε κάθε επιλογή

Destination:  , μετατρέποντάς το σε
Destination:  , όπου δίνεται η
δυνατότητα χρήσης κώδικα. Ένα σημαντικό Tip σε αυτό το σημείο είναι ότι πατώντας πάνω στο
σημείο όπου γράφεται η java, κάνοντας ctrl + space εμφανίζονται προτάσεις.



Εικόνα 150 Εισαγωγή κώδικα

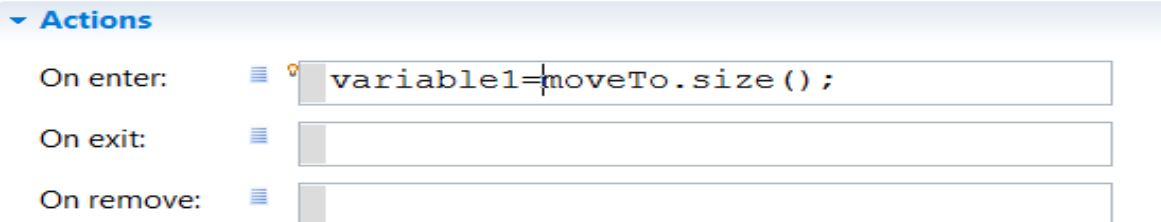
Σε αυτό το σημείο δίνεται η επιλογή να επιλέξουμε σε ποιο πράγμα αναφέρεται ο κώδικας. Αφού γίνει η επιλογή αυτή πατάμε την '.' Προκειμένου να δώσουμε την εντολή, ενώ πατώντας ξανά Ctrl + space εμφανίζονται οι διάφορες εντολές που μπορούμε να δώσουμε



Εικόνα 151 Εισαγωγή κώδικα 2

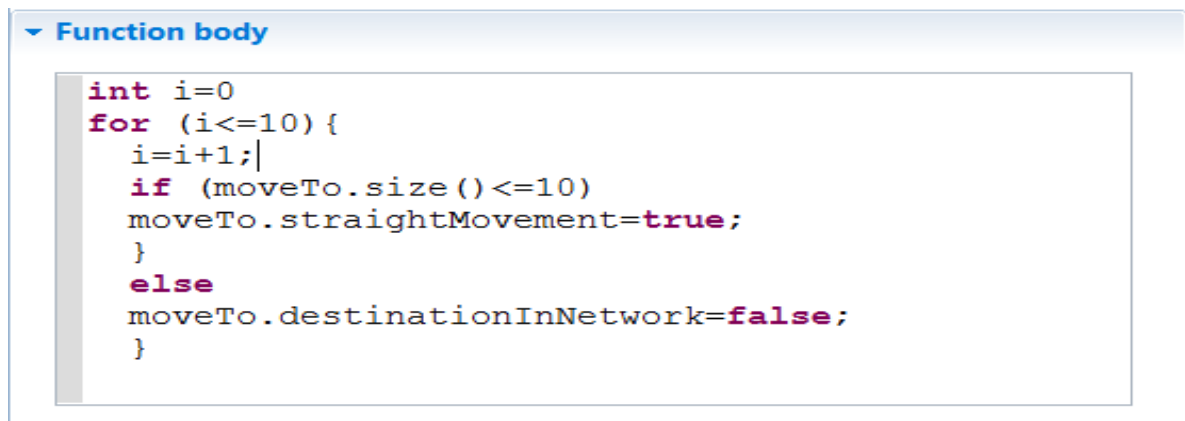
Επιλέγοντας για παράδειγμα το size() μας επιστρέφει τον αριθμό των Agent που βρίσκονται στο block moveTo.

Για να εισάγουμε αυτόν τον αριθμό σε κάποια Variable, δεν έχουμε τίποτα περισσότερο να κάνουμε από το να γράψουμε το όνομα της variable που θέλουμε να εισάγουμε την τιμή αυτή, ενώ στο τέλος της έκφρασης να συμπληρώσουμε το ';'.



Εικόνα 152 Εισαγωγή κώδικα 3

Επίσης θα μπορούσαμε να χρησιμοποιήσουμε και πιο σύνθετο κώδικα με χρήση των δομών "if", "for" κτλ. Όπως αυτές καθορίζονται από την java. Ένα απλό παράδειγμα χρήσης αυτών των μορφών είναι:



Εικόνα 153 Επανάληψη στον κώδικα

Σε αυτόν τον κώδικα έχουμε μία δομή επανάληψης όπου όσο το i είναι μικρότερο ή ίσο του 10, ελέγχει αν οι Agents που υπάρχουν στο block moveTo είναι λιγότεροι από 10. Αν είναι, τότε η μετακίνησή τους γίνεται με straight movement αλλιώς ακολουθούν τα path που έχουν φτιαχτεί στο μοντέλο.

Χρήσιμα κομμάτια του Java code:

- Κοινά Contextual Variables που χρησιμοποιούνται από αποσπάσματα του κώδικα
 1. Στην στατιστική: "item", υποδηλώνει το συγκεκριμένο Agent
 2. Στο "on Message Receiver" ενός Agent: "msg", υποδηλώνει το μήνυμα που λήφθηκε
 3. Στις Dynamic properties του Agent: "index", υποδηλώνει το index του ατόμου του Agent
 4. Στο "parameters" properties ενός Agent population: "index", υποδηλώνει το index του συγκεκριμένου Agent στο population
- Παραδείγματα κώδικα για export ενός Dataset

1. `FileOutputStream fos = new FileOutputStream("Filename");`
 2. `PrintStream p = new PrintStream(fos);`
 3. `p.println(datasetName.toString());`
- Κώδικας που χρησιμοποιείται συχνά
 1. `get_Main()` : γίνεται αναφορά στο Main object
 2. `ActiveObject.trace(str)` : εξάγει ένα string στο log
 3. `Engine.getTime()` : λαμβάνει τον χρόνο εκείνη την στιγμή
 4. `agents.size()` : παίρνει τον αριθμό των Agents
 5. `agents.item(i)`: παίρνει το item i από την agent collection
 6. `uniform()` : παράγεται ένας τυχαίος αριθμός από τα 0..1
 - Γενικές εκφράσεις (distributions)
 1. `uniform()`: παράγει έναν τυχαίο αριθμό από 0..1
 2. `uniform(x)`: παράγει έναν τυχαίο αριθμό από το 0 μέχρι το x
 3. `lognormal(double meanNormal, double sigmaStdDevNormal, double minNormal)`: παράγει έναν αριθμό από την lognormal distribution
 4. `normal(double meanNormal, double sigmaStdDevNormal)`: παράγει έναν αριθμό από την normal distribution
 5. υπάρχουν πολλές εκφράσεις κατανομών. Για λεπτομέρειες βλ. Anylogic.help
 - Μέθοδοι στο Populations of Agents (in Main class)
 1. `population.size()`: παίρνει τον αριθμό των Agents μιας population
 2. `population.item(int i)`: παίρνει το item i από το population
 3. `add_populationname()`: προσθέτει τον Agent σε αυτό το population
 4. `remove_populationname()`: αφαιρεί τον Agent από αυτό το population
 - Μέθοδοι για να καλεστεί ένας Agent
 1. `a.getConnectionsNumber()`: επιστρέφει τον αριθμό των connections ανάμεσα σε αυτόν τον Agent και στους υπόλοιπους
 2. `get_Main()`: γίνεται αναφορά στο Main object
 3. `toString()`: αποδίδει τον Agent σε string μορφή
 4. `a.connectTo(Agent b)`: συνδέει τον a με τον b
 5. `a.disconnectFrom(Agent b)`: αποσυνδέει τον a με τον b
 6. `a.disconnectFromAll()`: αποσυνδέει όλους τους agents από τον a
 7. `a.getConnectedAgent(int i)`: συνδέει τον ith agent με τον a
 8. `a.isConnectedTo(Agent b)`: ελέγχει αν ο a agent είναι συνδεδεμένος με τον b
 - Μέθοδοι στην Process Flow Diagrams
 1. `source.inject(int count)`: εισάγει έναν αριθμό εισροών στο block source
 - Μέθοδοι διαχείρισης μηνυμάτων
 1. `sender.deliver(msg, receiver)`: στέλνει ένα μήνυμα απευθείας από τον αποστολέα σε συγκεκριμένο δέκτη

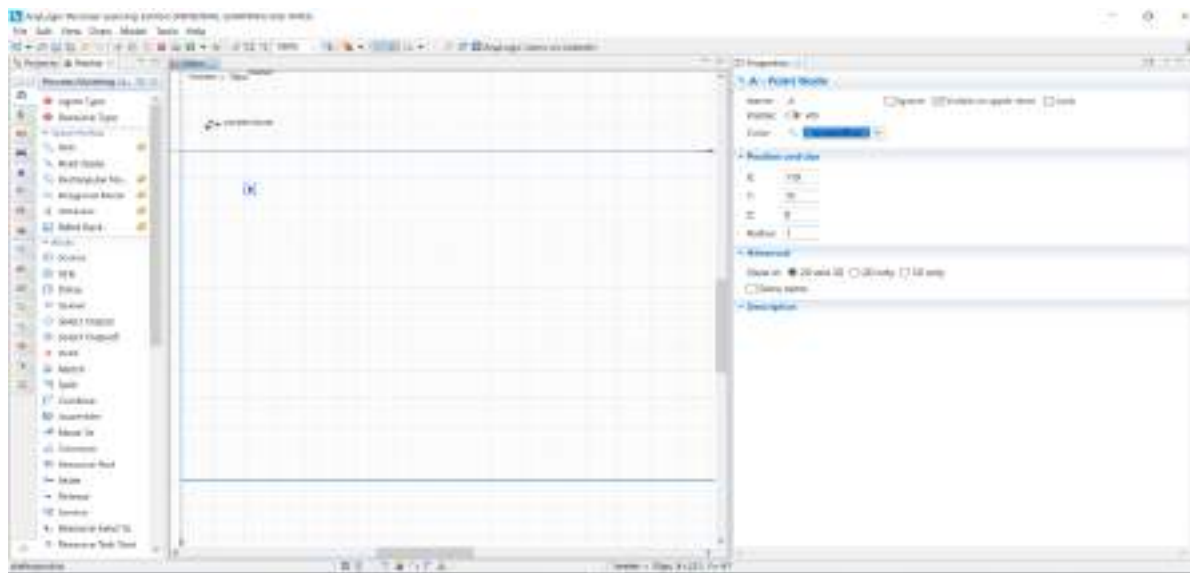
2. `environment.deliverToRandom(msg)`: στέλνει μήνυμα σε έναν τυχαίο agent σε ένα environment
3. `send( "Infection", RANDOM_CONNECTED)`: στέλνει μήνυμα σε έναν τυχαίο connected agent
4. `TBProgressionStatechart.receiveMessage (msg)`: προωθεί ένα μήνυμα που λήφθηκε από έναν agent στο statechart

Πρακτική εφαρμογή μοντέλου

Πρακτική εφαρμογή του μοντέλου που στην ενότητα 2.1:

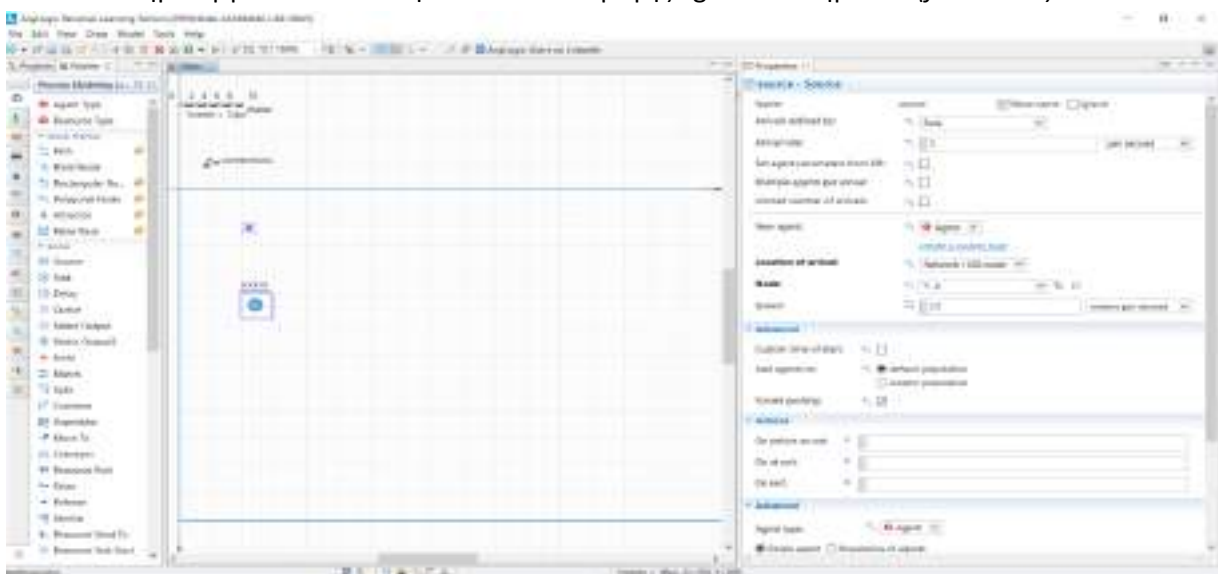
1. Τοποθέτηση του σημείου A (*animation*)

Με drag and drop τοποθετούμε το **point node** στο main και το ονομάζουμε A



Εικόνα 154 Εισαγωγή σημείου

2. Δημιουργία του κατάλληλου block εισαγωγής agent στο σημείο A (*flow chart*)



Εικόνα 155 Ιδιότητες source

Στα δεξιά ανοίγει το παράθυρο με τις ιδιότητες του. Με το **source** καθορίζουμε τον ρυθμό που θα εισέρχονται οι **agents**, πόσοι θα είναι αυτοί σε κάθε είσοδο, πόσες εισοδοι θα γίνουν, ποιοι **agent** θα έρθουν (μπορεί να έχουμε παραπάνω από έναν **agent**, π.χ. **Container** και **πλοία**, οπότε θα πρέπει να καθοριστούν) και τέλος σε πιο σημείο. Παράλληλα βλέπουμε και κάποια **action** που μπορούν να γίνουν στις διάφορες φάσεις του **agent** σε αυτό το **block**.

Στο **location of arrival** βάζουμε **A** και έτσι δόθηκε η εντολή να τοποθετούνται ένας **agent** κάθε δευτερόλεπτο στο σημείο **A**.

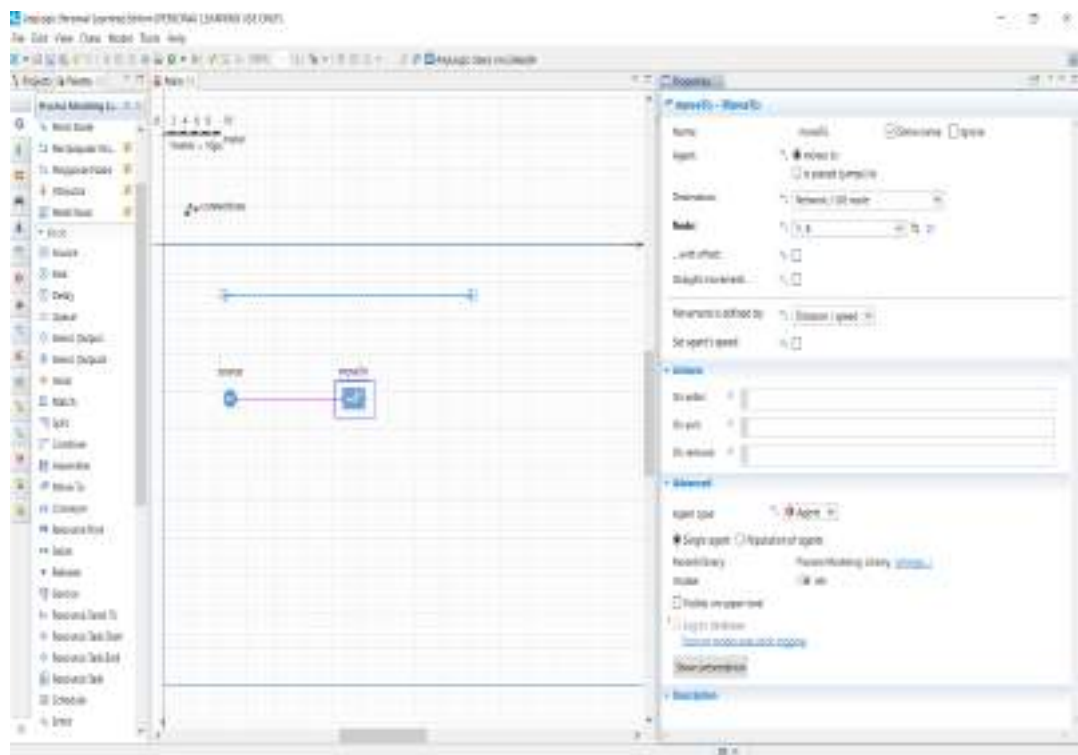
3. Τοποθέτηση του σημείου **B** (*animation*)

Ομοίως με το **A**

4. Τοποθέτηση του **path** που ενώνει τα σημεία **A** και **B** (*animation*)

κάνοντας διπλό κλικ στο **path** πάνω στην παλέτα, το εργαλείο μπαίνει σε **drawing mode** και με κλικ στο σημείο **A** και στην συνέχεια στο **B** ενώνονται τα σημεία

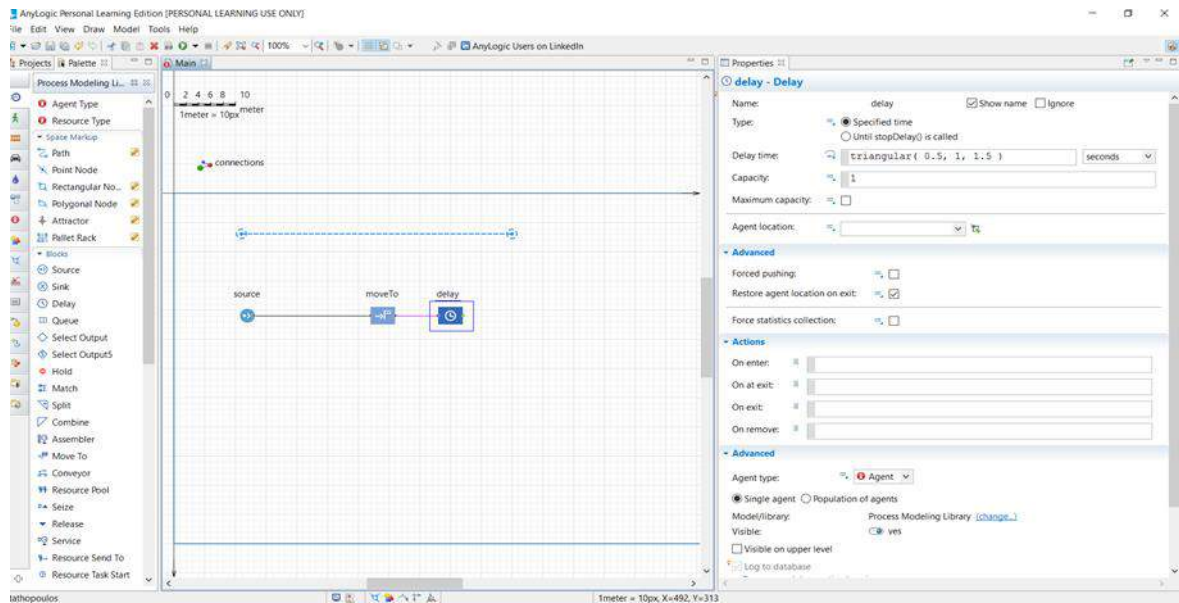
5. Δημιουργία του κατάλληλου **block** μετακίνησης των **agent** στο σημείο **B** (*flow chart*)



Εικόνα 156 Block μετακίνησης

Στις ιδιότητες του **Move to**, στο **destination** πατώντας το βελάκι επιλέγουμε **network/GIS** node και στο **node** το σημείο **B**. Με αυτόν τον δίνεται την εντολή να μετακινηθεί ο **agent** από το ένα σημείο στο άλλο.

6. Δημιουργία του κατάλληλου block αναμονής των agent στο σημείο B (flow chart)



Εικόνα 157 Block αναμονής

σε αυτό το block ρυθμίζεται το **Delay time** όπου αναφέρεται στο χρόνο αναμονής των agent σε αυτό το block προσομοιώνοντας κάποια διεργασία που γίνεται, το **capacity** όπου καθορίζεται πόσοι agents μπορούν να βρίσκονται ταυτόχρονα σε αυτό το block ενώ στο **agent location** καθορίζεται το σημείο που γίνεται αυτή η διεργασία (αν παραμένει κενό η διεργασία γίνεται στο σημείο που βρίσκεται από πριν ο agent) .

7. Τοποθέτηση του σημείου Γ (animation)

Ομοίως με προηγούμενως

8. Τοποθέτηση του path που ενώνει τα σημεία Β και Γ (animation)

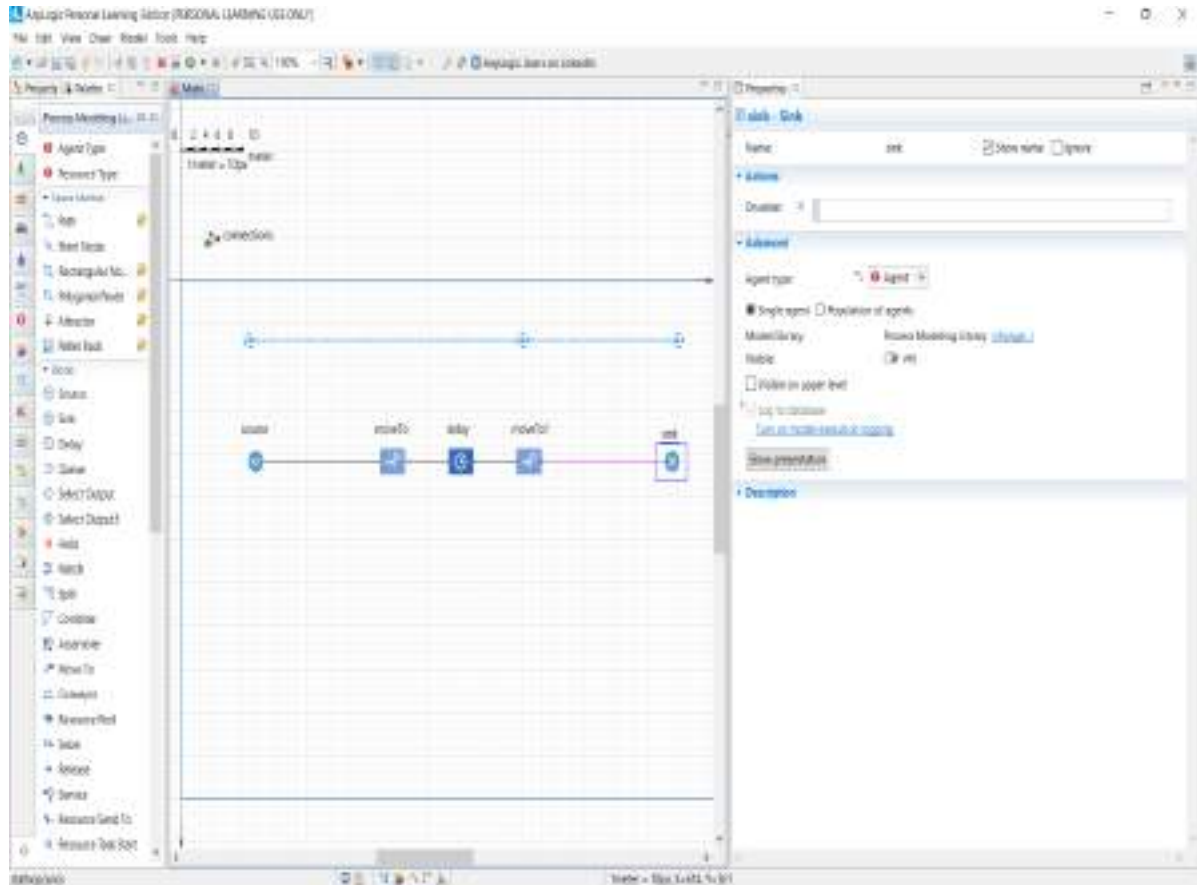
Ομοίως με προηγούμενως

9. Δημιουργία του κατάλληλου block μετακίνησης των agent στο σημείο Γ (flow chart)

Ομοίως με προηγούμενως

10. Δημιουργία του κατάλληλου block εξαγωγής agent από το σύστημα (flow chart)

Τοποθέτηση το block **sink** προκειμένου να γίνει η εξαγωγή των agent



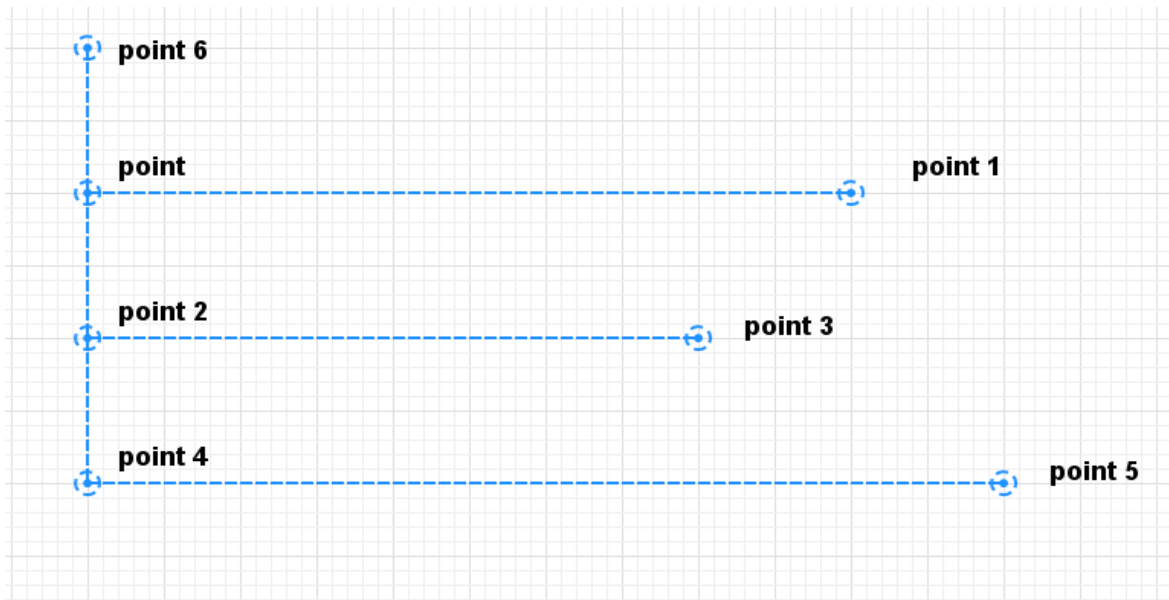
Εικόνα 158 Block sink

ΠΑΡΑΡΤΗΜΑ Β: Ανάλυση ενδεικτικού μοντέλου

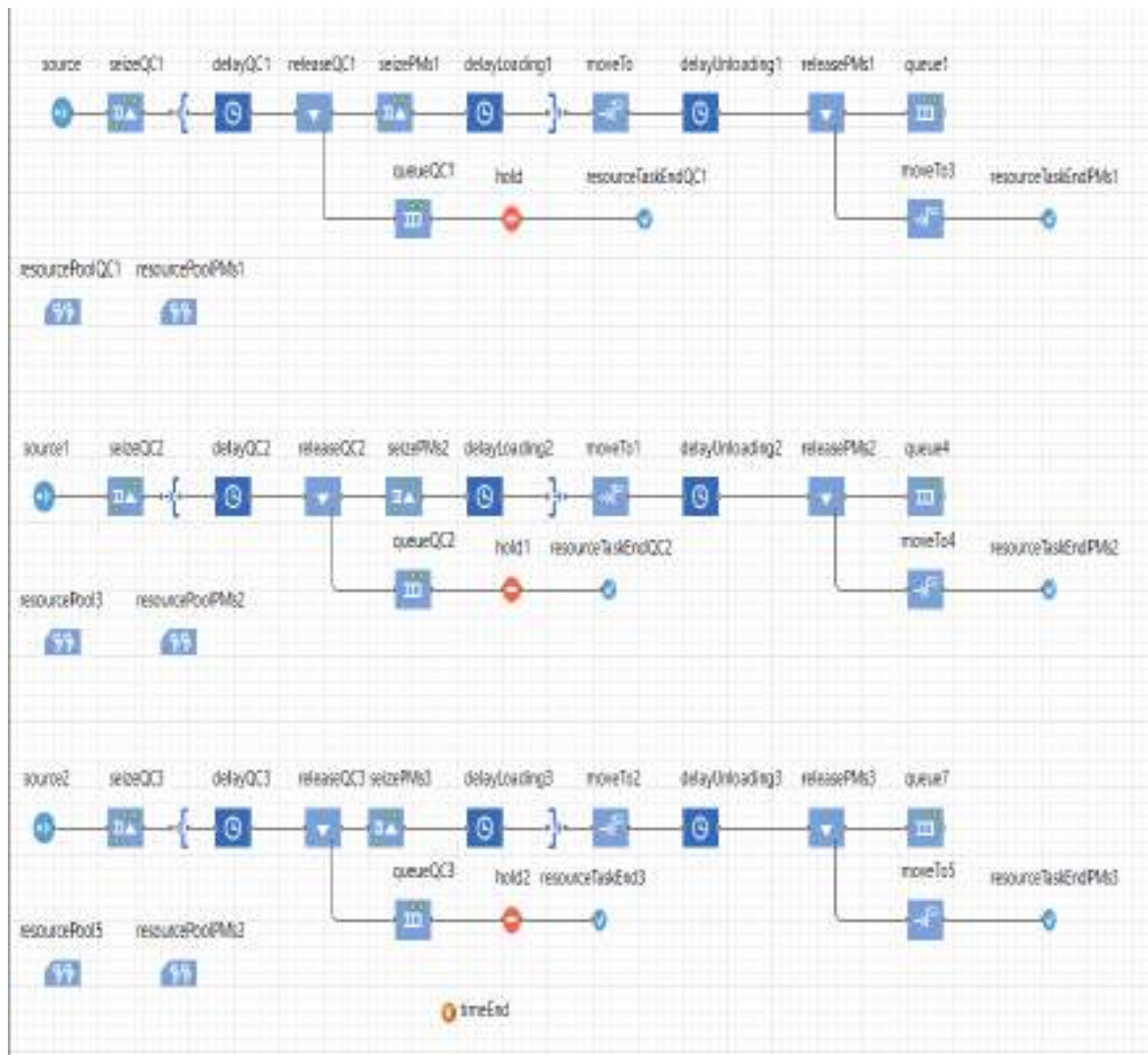
Αναλυτική επεξήγηση και επαλήθευση της σωστής λειτουργίας της σταθερής ανάθεσης.

1. Αναλυτική επεξήγηση:

Το συνολικό μοντέλο που κατασκευάστηκε για την εκφόρτωση τριών πλοίων είναι



Εικόνα 159 Animation ενδεικτικού μοντέλου



Εικόνα 160 Διάγραμμα ροής ενδεικτικού μοντέλου

Όπως είναι φανερό αναπτύχθηκαν τρεις πανομοιότυπες ροές, όπου η κάθε μία αναπαριστά την διαδρομή των εμπορευματοκιβωτίων από την γερανογέφυρα στον αντίστοιχο χώρο στοιβασίας. Η διαδικασία που ακολουθήθηκε είναι η δημιουργία των εμπορευματοκιβωτίων, η εξυπηρέτηση τους από τους QCs, η εξυπηρέτηση τους από τα Οχήματα μεταφοράς και τέλος η αποθήκευσή τους. Παρακάτω γίνεται ανάλυση της μιας ροής, ενώ οι υπόλοιπες έχουν διαμορφωθεί αντίστοιχα.

1.



Εικόνα 161 Source

source - Source

Name: ☒ Show name ☐ Ignore

Arrivals defined by:

New agent:

Location of arrival:

Node:

Speed: meters per second

Advanced

Add agents to: ☒ default population ☐ custom population

Forced pushing: ☒

Actions

On at exit:

On exit:

Advanced

Agent type:

☒ Single agent ☐ Population of agents

Model/library: [\(change...\)](#)

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database [Turn on model execution logging](#)

Εικόνα 162 Source1

Με το **source** γίνεται η εισαγωγή των εμπορευματοκιβωτίων στο σημείο εισαγωγής **point**. Η εισαγωγή αυτή γίνεται όταν δοθεί η εντολή `inject` η οποία δίνεται στην έναρξη του τρεξιματος.

Main - Agent Type

Name: ☐ Ignore

Agent actions

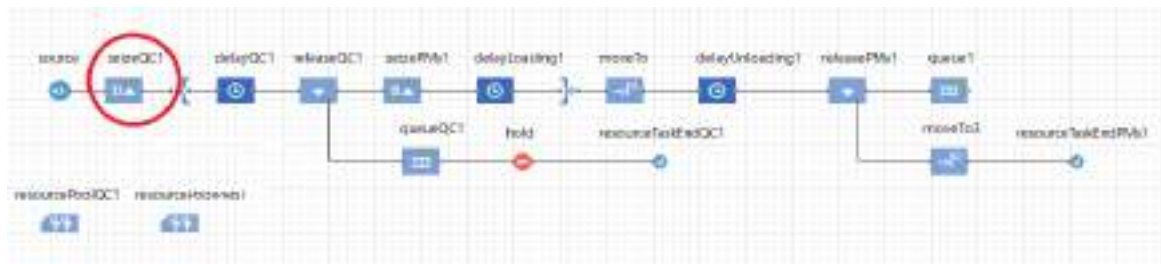
On startup:

```
source.inject(100);
source1.inject(100);
source2.inject(100);
```

Εικόνα 163 Source-main

Με αυτόν τον τρόπο γίνεται η εισαγωγή των 100 εμπορευματοκιβωτίων για κάθε **source** στο σύστημα.

2.



Εικόνα 164 SeizeQC1

Properties

seizeQC1 - Seize

Name: seizeQC1 ☒ Show name ☐ Ignore

Seize: ☒ (alternative) resource sets ☐ units of the same pool

Resource sets: ☒ resourcePoolQC1 1

Seize policy: ☒ Seize whole set at once ☐ Seize units one by one

Queue capacity:

Maximum queue capacity: ☐

Send seized resources: ☒

Destination is:

Attach seized resources: ☒

Agent location:

Priorities

Task priority:

Task may preempt: ☒

Task preemption policy:

Advanced

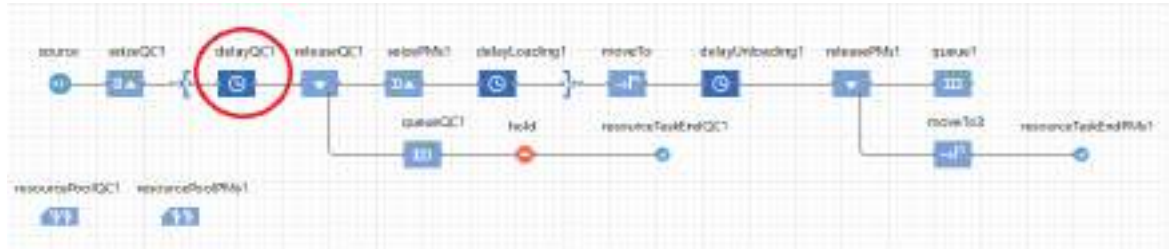
Customize resource choice: ☐

Define preparation tasks by:

Εικόνα 165 SeizeQC1 1

Με το **seize** καλείται ο QC που ανήκει στο **resourcePoolQC1**, ο οποίος αποτελεί εξυπηρετητή. Με το **send seized resources** πηγαίνει ο γερανός στο εμπορευματοκιβώτιο, ενώ με το **attach seized resources** το ``πιάνει``.

3.



Εικόνα 166 DelayQC1

Properties

delayQC1 - Delay

Name:

delayQC1

☒ Show name
☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

50

seconds

Capacity:

1

Maximum capacity:

Agent location:

Advanced

Forced pushing:

☐

Restore agent location on exit:

☒

Force statistics collection:

☐

Actions

On enter:

On at exit:

On exit:

hold.block();

On remove:

Advanced

Agent type:

container

☒ Single agent
☐ Population of agents

Model/library:

Process Modeling Library (change...)

Visible:

☒ yes

☐ Visible on upper level

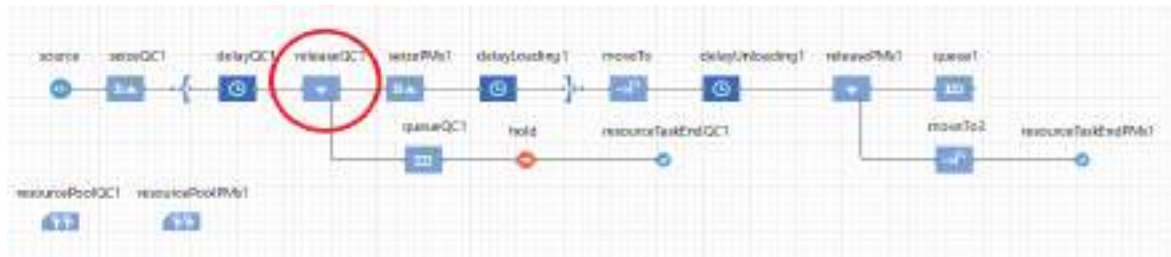
☒ Log to database

Σελίδα 179

Εικόνα 167 DelayQC1 1

Με το **delayQC1** προσομοιώνεται ο χρόνος που απαιτείται προκειμένου να μεταφερθεί το εμπορευματοκιβώτιο στο χώρο παραλαβής του από το όχημα μεταφοράς, ο οποίος ισοδυναμεί με 50 δευτερόλεπτα. Το **capacity** δηλώνει το πόσα εμπορευματοκιβώτια μπορούν ταυτόχρονα να βρίσκονται στην κατάσταση αυτή, σε αυτή την περίπτωση μόνο ένα. Το **agent location** είναι κενό, κάτι που σημαίνει ότι το εμπορευματοκιβώτιο θα παραμείνει στην θέση που είναι μετά από αυτή την αναμονή. Στα **actions** και στο **on exit** έχει χρησιμοποιηθεί ο κώδικας **hold.block()**; προκειμένου το block **hold** να παραμείνει μπλοκαρισμένο.

4.



Εικόνα 168 ReleaseQC1

Properties

releaseQC1 - Release

Name: releaseQC1 ☒ Show name ☐ Ignore

Release: All resources seized in given Seize block(s)

Seize blocks: seizeQC1

Moving resources: ☐ Return to home location ☒ Stay where they are

Wrap-up tasks

Wrap-up (e.g. move home): ☒ each time ☐ if no other tasks ☐ custom

'Wrap-up' usage statistics are: counted as 'idle'

Actions

On enter:

On release:

On exit:

On 'Wrap-up' terminated:

Advanced

Agent type: container

☒ Single agent ☐ Population of agents

Model/library: Process Modeling Library ([change...](#))

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database

[Turn on model execution logging](#)

Εικόνα 169 RreleaseQC1 1

Στο **releaseQC1** απελευθερώνεται το εμπορευματοκιβώτιο από τον γερανό. Όπως φαίνεται και στο διάγραμμα ροής, το εμπορευματοκιβώτιο συνεχίζει την κίνησή του στην δεξιά ροή, ενώ ο γερανός κινείται στο κάτω διάγραμμα ροής. Αυτό επιτυγχάνεται με την χρήση του **Wrap-up**, όπου έχοντας επιλέξει το **each time** κάθε φορά που ελευθερώνεται ένα εμπορευματοκιβώτιο δύναται η εντολή ο γερανός να κινηθεί στο άλλο διάγραμμα.

5.



Εικόνα 170 QueueQC1

Properties

queueQC1 - Queue

Name:

queueQC1

☒ Show name
☐ Ignore

Capacity:

100

Maximum capacity:

Agent location:

Advanced

Queuing:

FIFO

Enable exit on timeout:

☐

Enable preemption:

☐

Restore agent location on exit:

☒

Force statistics collection:

☐

Actions

On enter:

On at exit:

On exit:

On remove:

Εικόνα 171 QueueQC1 1

Στο **queueQC1** περιμένει ο γερανός μέχρι το **hold** να απελευθερωθεί. Αυτή η αναμονή αναπαριστά τον χρόνο που χρειάζεται για να φτάσει το όχημα μεταφοράς στον χώρο παραλαβής, και να ολοκληρωθεί η διαδικασία φόρτωσής του.

6.



Εικόνα 172 Hold

Properties ✕

hold - Hold

Name: hold ☒ Show name ☐ Ignore

Mode: ⌵ Manual (use block(), unblock()) ⌵

Initially blocked: ⌵ ☒

Actions

On enter: ⌵

Advanced

Agent type: ⌵ Agent ⌵

☒ Single agent ☐ Population of agents

Model/library: Process Modeling Library [\(change...\)](#)

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database [Turn on model execution logging](#)

Εικόνα 173 Hold 1

Στο **Hold** γίνεται το σταμάτημα της ροής αν η κατάσταση του είναι **blocked**, ενώ δεν αποτελεί εμπόδια όταν βρίσκεται στην κατάσταση **unblocked**. στην προκειμένη περίπτωση έχει γίνει η επιλογή στο **Mode**, **Manual (use block(), unblock())** έτσι ώστε να δίνεται με χρήση κώδικα η εντολή για να καθοριστεί η κατάσταση του. Στο διάγραμμα αυτό δίνεται η εντολή **hold.block()**; στο **delayQC1** κατά την έξοδο του εμπορευματοκιβωτίου, ενώ η εντολή **hold.unblock()**; δίνεται κατά την είσοδο στο **moveTo**.

7.



Εικόνα 174 TaskEnd

Properties

resourceTaskEndQC1 - ResourceTaskEnd

Name:

resourceTaskEndQC1

☒ Show name
☐ Ignore

Move resource to its home location:

=>

☒

Release resources seized by this unit:

=>

☒

Actions

On exit:

On remove:

Advanced

Unit type:

=>

Agent

☒ Single agent
☐ Population of agents

Model/library:

Process Modeling Library

(change...)

Visible:

☒ yes

☐ Visible on upper level

☒ Log to database

[Turn on model execution logging](#)

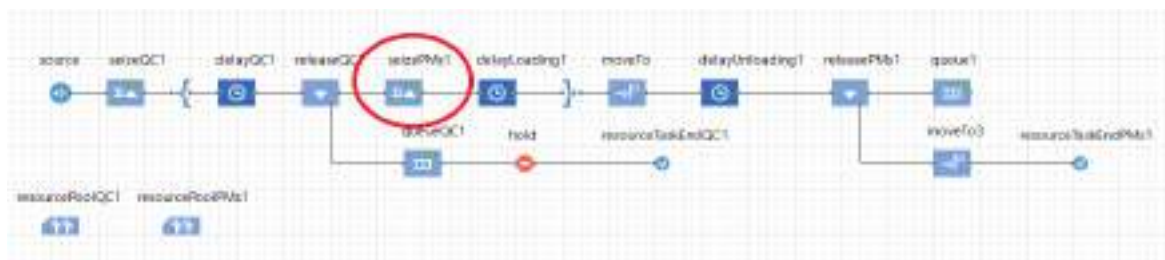
Show presentation

Description

Εικόνα 175 TaskEnd1

Σε αυτό το σημείο με την χρήση του **resourceTaskEndQC1** γίνεται η τελική απελευθέρωση των γερανών προκειμένου να γίνει η έναρξη της επόμενης εργασίας.

8.



Εικόνα 176 SeizePMs1

Properties

seizePMs1 - Seize

Name: seizePMs1 ☒ Show name ☐ Ignore

Seize: ☐ (alternative) resource sets
☒ units of the same pool

Resource pool: resourcePoolPMs1

Number of units: 1

Seize policy: ☒ Seize whole set at once
☐ Seize units one by one

Queue capacity: 100

Maximum queue capacity: ☐

Send seized resources: ☒

Destination is: Agent

Attach seized resources: ☒

Agent location:

Priorities

Task priority: 0

Task may preempt: ☒

Task preemption policy: No preemption

Advanced

Customize resource choice: ☐

Define preparation tasks by: Connecting blocks

Enable exit on timeout: ☐

Enable preemption: ☐

Εικόνα 177 SeizePMs1 1

Με την χρήση του **seizePMs1**, οι εξυπηρετητές που διαθέτει το **resourcePoolPMs1**, όπου στο συγκεκριμένο μοντέλο είναι μόνο ένας και προσομοιώνει το μεταφορικό μέσο προκειμένου να γίνει η μεταφορά του εμπορευματοκιβωτίου από τον χώρο παραβολής στον χώρο αποθήκευσης, πηγαίνουν στον **agent**, δηλαδή στο εμπορευματοκιβώτιο, επιλέγοντας στο **send seized resources** και στο **destination is** την επιλογή **Agent**. Επίσης κάνοντας κλικ στο **attach seized resources** δίνεται η εντολή ο εξυπηρετητής να πιάσει το εμπορευματοκιβώτιο.

8.



Εικόνα 178 DelayLoading1

Properties

delayLoading1 - Delay

Name:

delayLoading1

☒ Show name
 ☐ Ignore

Type:

☒ Specified time
☐ Until stopDelay() is called

Delay time:

10

seconds

Capacity:

1

Maximum capacity:

Agent location:

Advanced

Forced pushing:

☐

Restore agent location on exit:

☒

Force statistics collection:

☐

Actions

On enter:

On at exit:

On exit:

On remove:

Advanced

Agent type:

container

☒ Single agent
 ☐ Population of agents

Model/library:

Process Modeling Library [\(change...\)](#)

Visible:

☒ yes

☐ Visible on upper level

Εικόνα 179 DelayLoading1 1

Με την χρήση του **delayLoading1** προσομοιώνεται ο χρόνος που απαιτείται προκειμένου να φορτωθεί το εμπορευματοκιβώτιο στο μέσο μεταφοράς. Ο χρόνος αυτός ισοδυναμεί με 10 δευτερόλεπτα, ενώ η επιλογή στο **capacity** δηλώνει το πόσα εμπορευματοκιβώτια μπορούν ταυτόχρονα να βρίσκονται στην κατάσταση αυτή, σε αυτή την περίπτωση μόνο ένα.

Σελίδα 186

9.



Εικόνα 180 MoveTo1

Properties

moveTo - MoveTo

Name:

moveTo

☒ Show name
 ☐ Ignore

Agent:

☒ moves to
 ☐ is placed (jumps) to

Destination:

Network / GIS node

Node:

point1

... with offset:

☐

Straight movement:

☐

Movement is defined by:

Distance / speed

Set agent's speed:

☒

Speed:

0.25

meters per second

Actions

On enter:

hold.unblock();

On exit:

On remove:

Advanced

Agent type:

container

☒ Single agent
 ☐ Population of agents

Model/library:

Process Modeling Library (change...)

Visible:

☒ yes

☐ Visible on upper level

☒ Log to database

[Turn on model execution logging](#)

Show presentation

Εικόνα 181 MoveTo1 1

Με την χρήση του **MoveTo** γίνεται η μεταφορά του εμπορευματοκιβωτίου από την θέση παραβολής στον χώρο εναπόθεσης (**point1**). Η μεταφορά αυτή γίνεται με ταχύτητα 0,25 μέτρα το δευτερόλεπτο. Στα **Actions** και στο **On enter** δίνεται η εντολή **hold.unblock()**; προκειμένου να απελευθερωθεί το **hold** και κατ' επέκταση και ο γερανός.

10.



Εικόνα 182 DelayUnloading1

Properties   

 **delayUnloading1 - Delay**

Name: delayUnloading1 ☒ Show name ☐ Ignore

Type:  ☒ Specified time
☐ Until stopDelay() is called

Delay time:  17 seconds 

Capacity:  1

Maximum capacity: 

Agent location:   

Advanced

Forced pushing:  ☐

Restore agent location on exit:  ☒

Force statistics collection:  ☐

Actions

On enter:  

On at exit:  

On exit:  

On remove:  

Advanced

Agent type:   container 

☒ Single agent ☐ Population of agents

Model/library: Process Modeling Library [\(change...\)](#)

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database

Εικόνα 183 DelayUnloading1 1

Με την χρήση του **delayUnloading1** προσομοιώνεται ο χρόνος που απαιτείται προκειμένου να ξεφορτωθεί το εμπορευματοκιβώτιο από το μέσο μεταφοράς. Ο χρόνος αυτός ισοδυναμεί με 17 δευτερόλεπτα, ενώ η επιλογή στο **capacity** δηλώνει το πόσα εμπορευματοκιβώτια μπορούν ταυτόχρονα να βρίσκονται στην κατάσταση αυτή, σε αυτή την περίπτωση μόνο ένα.

11.



Εικόνα 184 releasePMs1

Properties

releasePMs1 - Release

Name:

releasePMs1

☒ Show name
 ☐ Ignore

Release:

All resources seized in given Seize block(s)

Seize blocks:

seizePMs1

Moving resources:

☐ Return to home location
☒ Stay where they are

Wrap-up tasks

Wrap-up (e.g. move home):

☒ each time
☐ if no other tasks
☐ custom

'Wrap-up' usage statistics are:

counted as 'busy'

Actions

On enter:

On release:

On exit:

On 'Wrap-up' terminated:

Advanced

Agent type:

container

☒ Single agent
 ☐ Population of agents

Model/library:

Process Modeling Library [\(change...\)](#)

Visible:

☒ yes

☐ Visible on upper level

☒ Log to database

[Turn on model execution logging](#)

Εικόνα 185 ReleasePMs1 1

Στο **releasePMs1** απελευθερώνεται το εμπορευματοκιβώτιο από το μεταφορικό μέσο. Όπως φαίνεται και στο διάγραμμα ροής, το εμπορευματοκιβώτιο συνεχίζει την κίνησή του στην δεξιά ροή, ενώ το μεταφορικό μέσο κινείται στο κάτω διάγραμμα ροής. Αυτό επιτυγχάνεται με την

Με την χρήση του **queue1** προσομοιώνεται ο χώρος εναπόθεσης των εμπορευματοκιβωτίων. Στα **Actions** και στο **On enter** γίνεται χρήση του κώδικα

```
“ if (self.size()==100)
    timeEnd=time(); ”
```

όπου αν σε στον χώρο στοιβάσις υπάρχουν 100 εμπορευματοκιβώτια, η μεταβλητή **timeEnd** λαμβάνει ως τιμή την συγκεκριμένη χρονική στιγμή.

13.



Εικόνα 188 MoveTo3

Properties

moveTo3 - MoveTo

Name:

moveTo3

☒ Show name
☐ Ignore

Agent:

☒ moves to
☐ is placed (jumps) to

Destination:

Network / GIS node

Node:

point

... with offset:

☐

Straight movement:

☐

Movement is defined by:

Distance / speed

Set agent's speed:

☐

Actions

On enter:

On exit:

On remove:

Advanced

Agent type:

Agent

☒ Single agent
☐ Population of agents

Model/library:

Process Modeling Library

(change...)

Visible:

☒ yes

☐ Visible on upper level

☒ Log to database

Turn on model execution logging

Show presentation

Εικόνα 189 MoveTo3 1

Με την χρήση του **moveTo3** δίνεται εντολή στο μεταφορικό μέσο να μετακινηθεί στον χώρο παραβολής προκειμένου να είναι σε αναμονή προκειμένου να παραλάβει το επόμενο εμπορευματοκιβώτιο.

14.



Εικόνα 190 TaskEndPMs1

Properties ✕

resourceTaskEndPMs1 - ResourceTaskEnd

Name: resourceTaskEndPMs1 ☒ Show name ☐ Ignore

Move resource to its home location: ☐

Release resources seized by this unit: ☐

Actions

On exit:

On remove:

Advanced

Unit type: Agent

☒ Single agent ☐ Population of agents

Model/library: Process Modeling Library [\(change...\)](#)

Visible: ☒ yes

☐ Visible on upper level

☒ Log to database
[Turn on model execution logging](#)

Εικόνα 191 TaskEndPMs1 1

Με την χρήση του **resourceTaskEndPMs1** απελευθερώνονται τα μεταφορικά μέσα και είναι σε ετοιμότητα προκειμένου να αναλάβουν την επόμενη εργασία που θα ανατεθεί.

15.



Εικόνα 192 ResourceTaskEndPMs1

restrictedAreaStart2 - RestrictedAreaStart

Name: restrictedAreaStart2 ☐ Show name ☐ Ignore

Capacity (max allowed):

Actions

On enter:

Εικόνα 193 resourceTaskStartPMs1

restrictedAreaEnd - RestrictedAreaEnd

Name: restrictedAreaEnd
☐ Show name
☐ Ignore

RestrictedAreaStart object:

restrictedAreaStart2

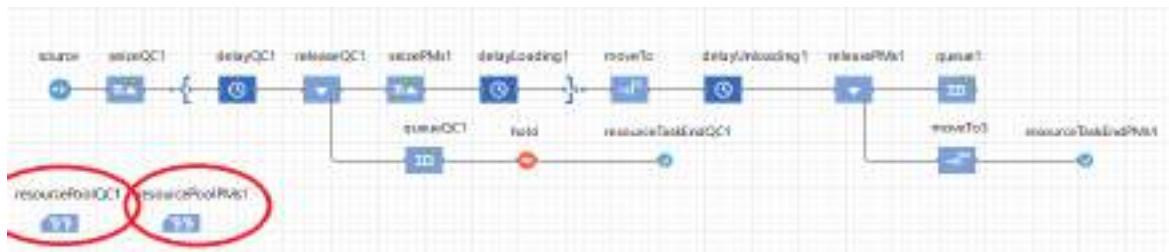
Actions

On enter:

Εικόνα 194 ResourceTaskEndPMs1

Με την χρήση του συνδυασμού restrictedAreaStart2 και restricted AreaEnd, έχοντας ορίσει την **capacity** στο 1 υπάρχει ο περιορισμός του ενός **agent**, δηλαδή εμπορευματοκιβωτίου, στην περιοχή αυτή. Αυτό γίνεται προκειμένου να μην μπορεί αν ξεκινήσει μια διεργασία στον γερανό ένα δεν τελειώσει με την προηγούμενη.

16.



Εικόνα 195 ResourcePool

Properties

resourcePoolPMs1 - ResourcePool

Name:

resourcePoolPMs1

☒ Show name

☐ Ignore

Resource type:

Moving

Capacity defined:

Directly

Capacity:

2

When capacity decreases:

units are preserved ('End of shift')

New resource unit:

trucks

Speed:

0.25

meters per second

Home location (nodes):

point6

Shifts, breaks, failures, maintenance...

'End of shift' priority:

100

'End of shift' preemption policy:

No preemption

'End of shift' may preempt:

☒

Breaks:

☐

Failures / repairs:

☐

Maintenance:

☐

Custom tasks:

☐

Advanced

Add units to:

☒ default population
☐ custom population

Force statistics collection:

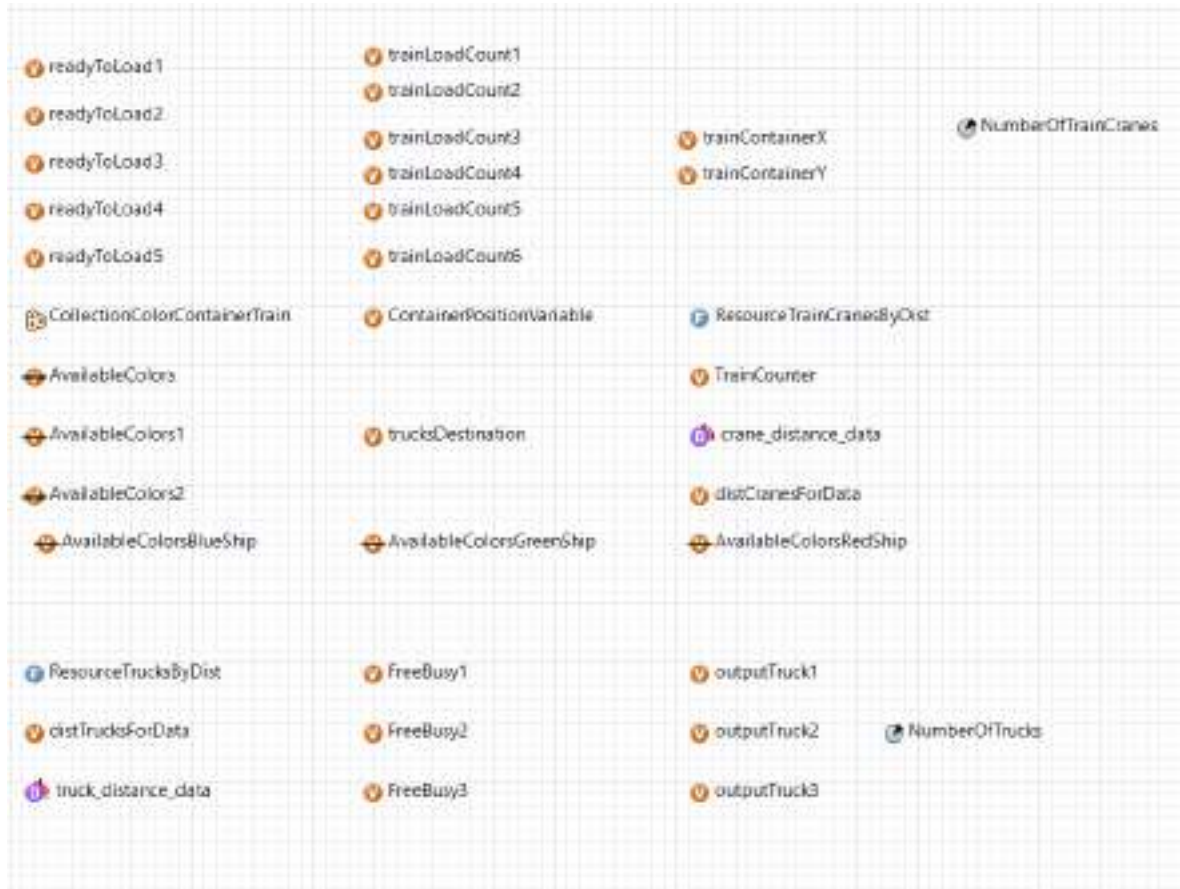
☐

Εικόνα 196 ResourcePool1

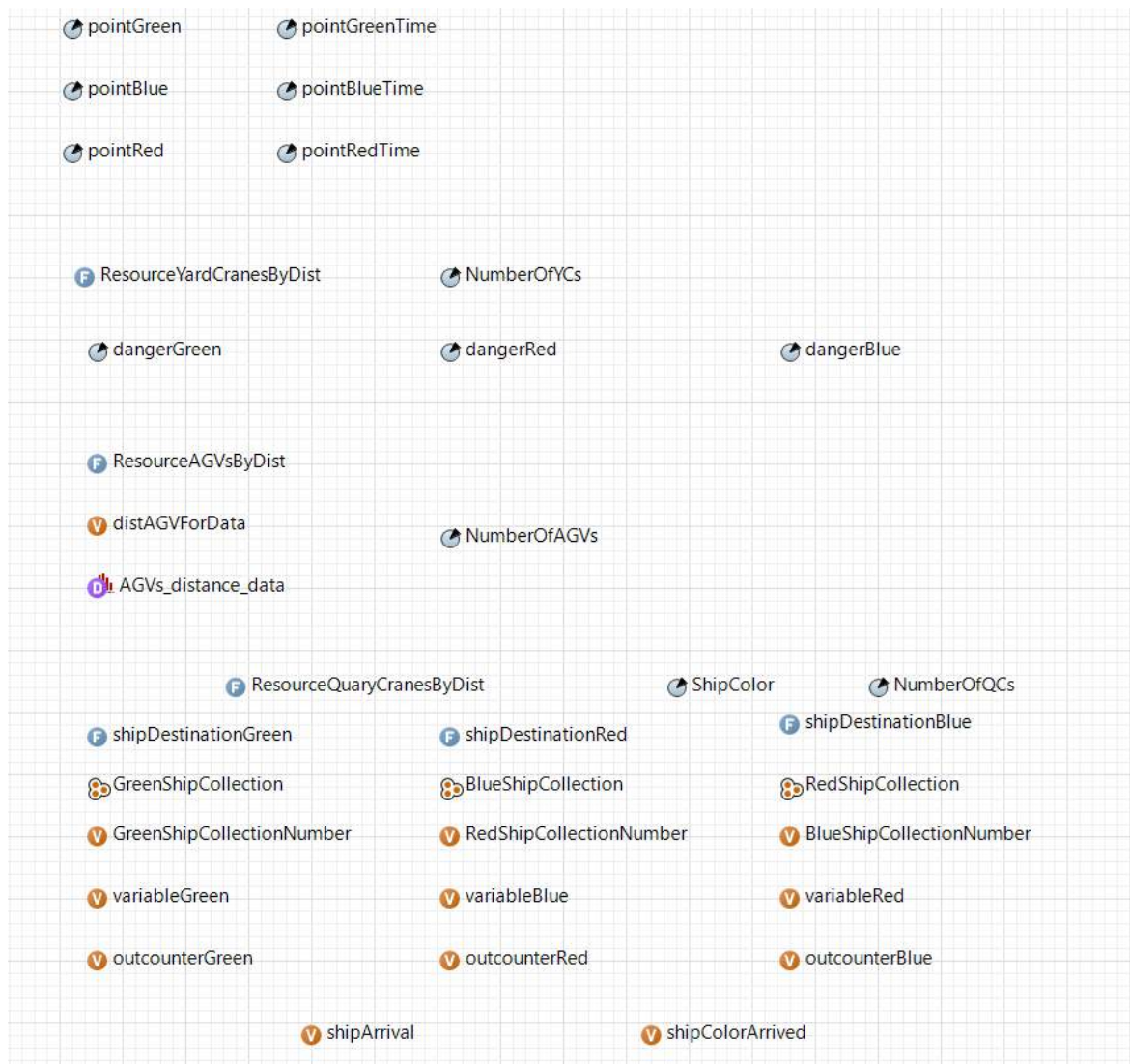
Τα 2 **resource pool** αποτελούν τους εξυπηρετητές. Το **resourcePoolQC1** περιλαμβάνει τον γερανό, ενώ το **resourcePoolPMs1** τα μεταφορικά μέσα. Σε αυτό το μοντέλο έχουμε έναν γερανό και δύο μεταφορικά μέσα όπου κινούνται με 0,25 μέτρα το δευτερόλεπτο και με αρχική τοποθεσία το **point6**.

ΠΑΡΑΡΤΗΜΑ Γ Ανάλυση μοντέλου προσομοίωσης λιμένα

Οι μεταβλητές που χρησιμοποιήθηκαν στο σύνολο του μοντέλου είναι:

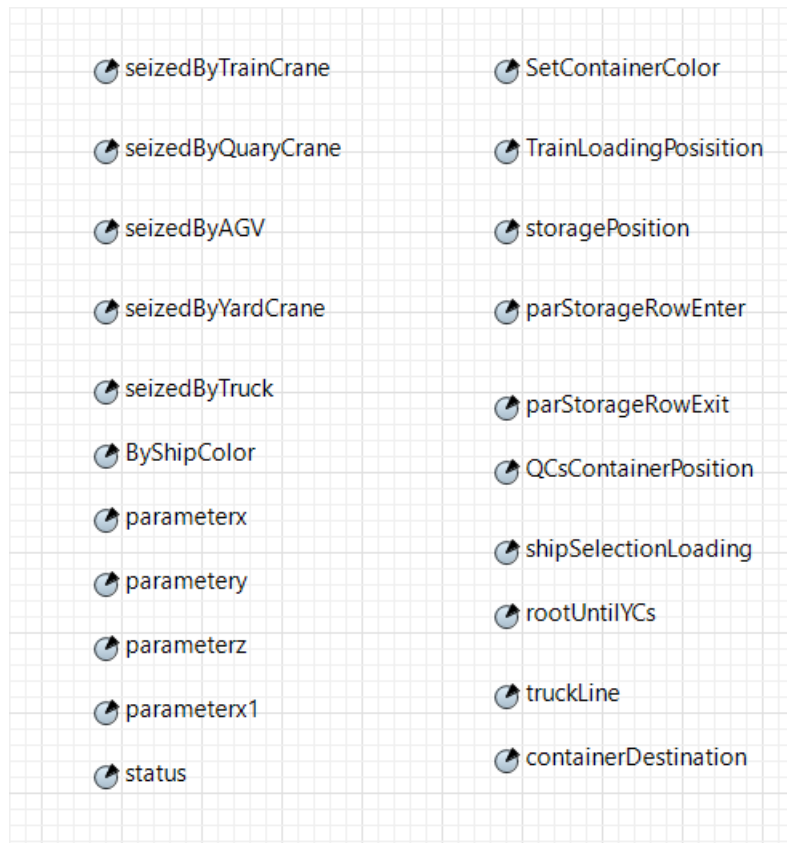


Εικόνα 197 Μεταβλητές μοντέλου προσομοίωσης λιμένα



Εικόνα 198 Μεταβλητές μοντέλου προσομοίωσης λιμένα1

Επίσης σημαντικό παράγοντα για την υλοποίηση του μοντέλου αποτελούν και οι παράμετροι που περιέχει μέσα ο agent.



Εικόνα 199 Παράμετροι agent

ΘΑΛΑΣΣΙΑ ΠΕΡΙΟΧΗ

Με την χρήση του (**selectOutputByShipColour**) ανάλογα με το χρώμα του εκάστοτε πλοίου, ανακατευθύνεται στον αντίστοιχο χώρο αναμονής και χώρο παραβολής.

selectOutputByShipColour - SelectOutput5

Name: ☒ Show name ☐ Ignore

Use: ☐ Probabilities ☒ Conditions ☐ Exit number

Condition 1:

Condition 2:

Condition 3:

Condition 4:

Εικόνα 200 SelectOutputByShipColour

Μόλις το πλοίο φτάσει και παραβάλει, τότε εισέρχονται στο σύστημα τα εμπορευματοκιβώτια που κουβαλά και πρόκειται να εκφορτωθούν. Έτσι στο **shipGreenQueue**, **shipRedQueue** και στο **shipBlueQueue**, διαμορφώνονται κάποιες μεταβλητές ως εξής. (η φωτογραφία αφορά το πλοίο με το πράσινο χρώμα)

▼ **Actions**

On enter:

```
shipArrival=nodeGreen;
startLoadingGreen.unblock();
ShipContainerSource.inject(60);
shipColorArrived=green;
```

Εικόνα 201 ShipGreenQueue

ΠΑΡΑΘΑΛΑΣΣΙΑ ΠΕΡΙΟΧΗ

Στην περιοχή κρηπιδώματος, η αρχή γίνεται με την έλευση των εμπορευματοκιβωτίων από τα πλοία, δηλαδή με το “ShipContainerSource.inject(60);”. Επίσης ανάλογα με την μεταβλητή “shipArrival” τα εμπορευματοκιβώτια χρωματίζονται ανάλογα από μια λίστα. Έτσι στο **ShipContainerSource** έχουν καθοριστεί οι εξής ιδιότητες:

⌕ **ShipContainerSource - Source**

Name: ☒ Show name ☐ Ignore

Arrivals defined by:

New agent:

Location of arrival:

Node:

Speed:

▼ **Advanced**

Add agents to: ☒ default population ☐ custom population

Forced pushing: ☒

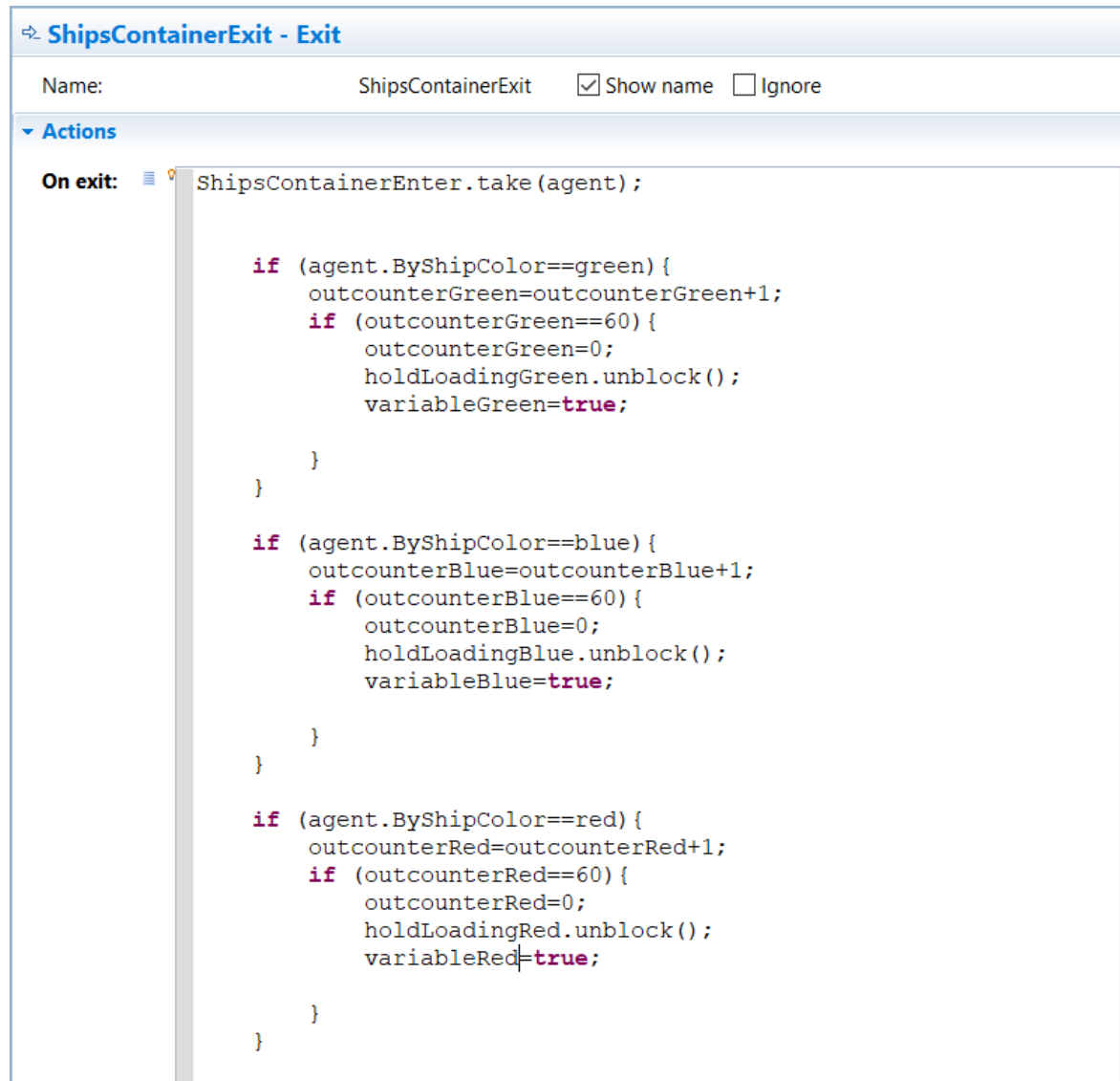
▼ **Actions**

On at exit:

```
if (shipArrival==nodeGreen){
agent.SetContainerColor=randomFrom(AvailableColorsGreenShip);}
if (shipArrival==nodeRed){
agent.SetContainerColor=randomFrom(AvailableColorsRedShip);}
if (shipArrival==nodeBlue){
agent.SetContainerColor=randomFrom(AvailableColorsBlueShip);}
```

Εικόνα 202 ShipContainerSource

Τέλος, λίγο πριν εισχωρήσουν τα εμπορευματοκιβώτια στην περιοχή μεταφοράς (**ShipsContainerExit**), έχουν δημιουργηθεί μετρητές, με την χρήση των μεταβλητών `outcounterGreen`, `outcounterBlue` και `outcounterRed`, προκειμένου να ελεγχτεί πότε ολοκληρώθηκε η διαδικασία της εκφόρτωσης. Μόλις γίνει αυτό, η μεταβλητές `variableGreen`, `variableBlue` και `variableRed` αλλάζουν κατάσταση προκειμένου να αρχίσει η διαδικασία της φόρτωσης.



Εικόνα 203 ShipsContainerExit

Για την φόρτωση, έχουν δημιουργηθεί 3 λίστες που περιέχουν σημεία, μια για κάθε παραβολή, όπου τα AGVs πηγαίνουν προκειμένου οι γερανοί να παραλάβουν τα εμπορευματοκιβώτια και τα τοποθετήσουν στα πλοία. Η περιήγηση στη λίστα αυτή γίνεται στο block (**moveToQC**) με την χρήση των μεταβλητών `GreenShipCollectionNumber`, `RedShipCollectionNumber` και `BlueShipCollectionNumber`

```

agent.setAutomaticHorizontalRotation(true);
if (agent.setContainerColor==green || agent.setContainerColor==lime || agent.setContainerColor==greenYellow){
    if (GreenShipCollectionNumber<=6){
        GreenShipCollectionNumber=GreenShipCollectionNumber+1;
    }
    else{
        GreenShipCollectionNumber=0;
        agent.shipSelectionLoading=nodeGreen;
        agent.QcContainerPosition=GreenShipCollection.get(GreenShipCollectionNumber);
    }
}

if (agent.setContainerColor==red || agent.setContainerColor==coral || agent.setContainerColor==pink){
    if (RedShipCollectionNumber<=6){
        RedShipCollectionNumber=RedShipCollectionNumber+1;
    }
    else{
        RedShipCollectionNumber=0;
        agent.shipSelectionLoading=nodeRed;
        agent.QcContainerPosition=RedShipCollection.get(RedShipCollectionNumber);
    }
}

if (agent.setContainerColor==blue || agent.setContainerColor==deepSkyBlue || agent.setContainerColor==cyan){
    if (BlueShipCollectionNumber<=6){
        BlueShipCollectionNumber=BlueShipCollectionNumber+1;
    }
    else{
        BlueShipCollectionNumber=0;
        agent.shipSelectionLoading=nodeBlue;
        agent.QcContainerPosition=BlueShipCollection.get(BlueShipCollectionNumber);
    }
}
}

```

Εικόνα 204 MoveToQC

Ένα παράδειγμα τέτοιας λίστας είναι η εξής:

GreenShipCollection - Collection

Name: GreenShipCollection
☒ Show name
☐ Ignore

Visible: ☒ yes

Collection class: ArrayList

Elements class: Other... PointNode

Advanced

Initial contents:

point65
point66
point67
point68
point184
point87
point489

+
-
↑
↓
✕

Access: public

☐ Save in snapshot
☐ Static

Description

Εικόνα 205 GreenShipCollection

ΠΕΡΙΟΧΗ ΜΕΤΑΦΟΡΑΣ

Κατά την εκκίνηση του μοντέλου, δημιουργείται μια λίστα για κάθε χρώμα εμπορευματοκιβωτίου, όπου περιέχει αντικείμενα. Το κάθε αντικείμενο έχει σαν παραμέτρους την θέση x, την θέση y, την θέση z και το status της θέσης (καινή ή καταλυμένη).

On startup:

```
TrainSource.inject(1);

for ( int z=1; z<5; z++) {
    for ( int y=0; y<30; y++) {
        for ( int x=0; x<5; x++) {
            container cg = new container();
            cg.parameterx = x;
            cg.parametery = y;
            cg.parameterz = z;
            cg.status = -1;

            container cb = new container();
            cb.parameterx = x;
            cb.parametery = y;
            cb.parameterz = z;
            cb.status = -1;

            container cr = new container();
            cr.parameterx = x;
            cr.parametery = y;
            cr.parameterz = z;
            cr.status = -1;

            pointGreen.add(cg);
            pointBlue.add(cb);
            pointRed.add(cr);
        }
    }
}
```

Εικόνα 206 Startup

Κατά την μεταφορά των εμπορευματοκιβωτίων προς τον χώρο στοιβασίας (δηλαδή στα **moveToStorage**, **moveToStorage1** και **moveTo1**) γίνεται ο καθορισμός της θέσης του κάθε εμπορευματοκιβωτίου. Κάτι που σημαίνει, έχοντας δημιουργήσει 30 θέσεις κατά τον y άξονα, 5 θέσεις κατά τον x και 5 κατά τον z κατά την εκκίνηση του μοντέλου, πολλαπλασιάζοντας και προσθέτοντας κατάλληλα δημιουργούνται οι τελικές συντεταγμένες όπου και θα γίνει η εναπόθεση. Οι συντεταγμένες αυτές 'αποθηκεύονται' στο εμπορευματοκιβώτιο (agent). Ταυτόχρονα το status της αντίστοιχης θέσης γίνεται 0, δηλαδή καταλυμένο. Ο λόγος που έχει ενταχθεί η έννοια του status είναι το γεγονός ότι ο κώδικας κάνει αναζήτηση και βρίσκει την πρώτη διαθέσιμη θέση.

```

if (agent.SetContainerColor==green || agent.SetContainerColor==lime || agent.SetCont
    for ( int i=0; i<pointGreen.size(); i++) {
        if ( pointGreen.get(i).status == -1 ){
            agent.parameterx=(810 - 20 * ( pointGreen.get(i).parameterx ));
            agent.parameterz=(765 -13*( pointGreen.get(i).parameterz ));
            agent.parameterz=(3*pointGreen.get(i).parameterz);
            pointGreen.get(i).status = 0;
            break;
        }
    }
}
if (agent.SetContainerColor==blue || agent.SetContainerColor==deepSkyBlue || agent.S

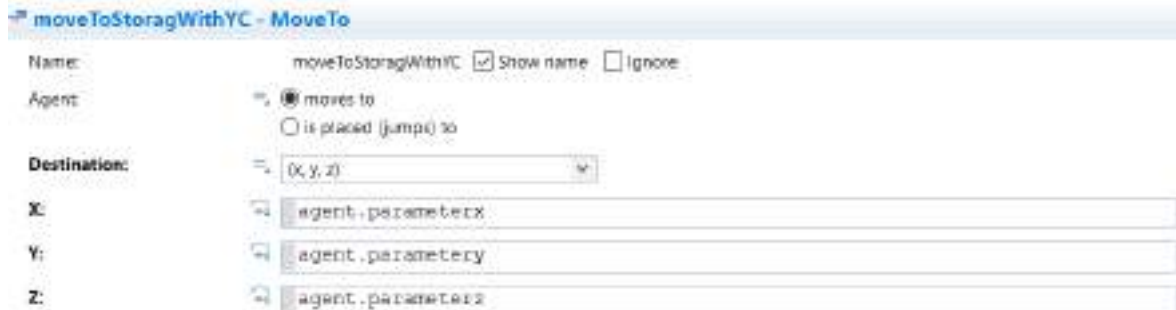
    for ( int i=0; i<pointBlue.size(); i++) {
        if ( pointBlue.get(i).status == -1 ){
            agent.parameterx=(810 - 20 * ( pointBlue.get(i).parameterx ));
            agent.parameterz=(1180 -13*( pointBlue.get(i).parameterz ));
            agent.parameterz=(3*pointBlue.get(i).parameterz);
            pointBlue.get(i).status = 0;
            break;
        }
    }
}
if (agent.SetContainerColor==red || agent.SetContainerColor==coral || agent.SetConta
    for ( int i=0; i<pointRed.size(); i++) {
        if ( pointRed.get(i).status == -1 ){
            agent.parameterx=(810 - 20 * ( pointRed.get(i).parameterx ));
            agent.parameterz=(1610 -13*( pointRed.get(i).parameterz ));
            agent.parameterz=(3*pointRed.get(i).parameterz);
            pointRed.get(i).status = 0;
            break;
        }
    }
}

```

Εικόνα 207 Move To Storage

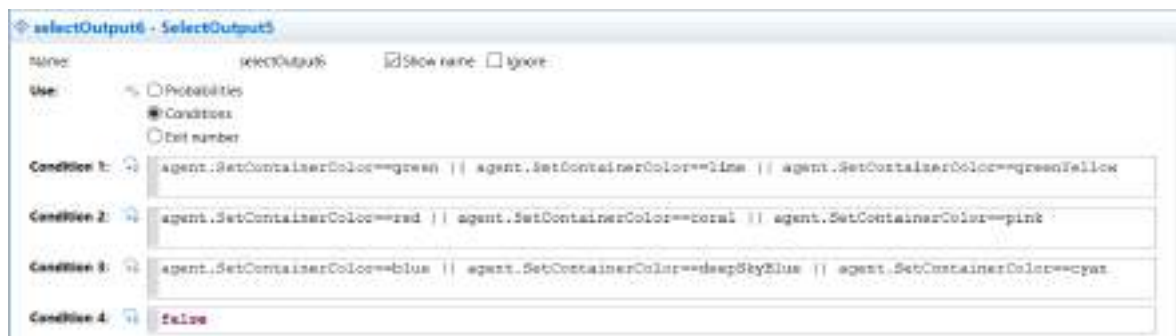
ΠΕΡΙΟΧΗ ΣΤΟΙΒΑΣΙΑΣ

Έτσι βλέπουμε ότι ο γερανός τα εναποθέτει στις θέσεις (**moveToStorageWithYC**) σύμφωνα με τις συντεταγμένες που ‘κουβαλά’ το κάθε εμπορευματοκιβώτιο (agent).



Εικόνα 208 move To Storage With YC

Η λογική που τοποθετείται το κάθε εμπορευματοκιβώτιο στον αντίστοιχο χώρο αναμονής είναι με βάση το χρώμα του. Έτσι με την χρήση του (**selectOutput6**) ανακατευθύνεται.



Εικόνα 209 Select Output 6

Κατά την έξοδο του εμπορευματοκιβωτίου από τον χώρο εναπόθεσης, δηλαδή την στιγμή που το πιάνει ο γερανός (**YCSeize2**), πρέπει να εφαρμοσθεί η αντίστροφη διαδικασία για τον εντοπισμό της θέσης μέσα στην λίστα που κατείχε το εμπορευματοκιβώτιο. Αυτό γίνεται προκειμένου να αλλάξει εκ νέου το status σε καινό.

```

if (agent.SetContainerColor==green || agent.SetContainerColor==lime || agent.SetContainerColor==greenYellow){
    for (int i=0; i<pointGreen.size(); i++) {
        if ( agent.parameterx == (810 - 20 * ( pointGreen.get(i).parameterx )) &&
            agent.parametery == (765 -13*( pointGreen.get(i).parametery )) &&
            agent.parametersz == 3*pointGreen.get(i).parametersz){

            pointGreen.get(i).status = -1;
            break;
        }
    }
}

if (agent.SetContainerColor==blue || agent.SetContainerColor==deepSkyBlue || agent.SetContainerColor==cyan ){
    for (int i=0; i<pointBlue.size(); i++) {
        if ( agent.parameterx == (810 - 20 * ( pointBlue.get(i).parameterx )) &&
            agent.parametery == (1165 -13*( pointBlue.get(i).parametery )) &&
            agent.parametersz == 3*pointBlue.get(i).parametersz){

            pointBlue.get(i).status = -1;
            break;
        }
    }
}

if (agent.SetContainerColor==red || agent.SetContainerColor==coral || agent.SetContainerColor==pink){
    for (int i=0; i<pointRed.size(); i++) {
        if ( agent.parameterx == (810 - 20 * ( pointRed.get(i).parameterx )) &&
            agent.parametery == (1615 -13*( pointRed.get(i).parametery )) &&
            agent.parametersz == 3*pointRed.get(i).parametersz){

            pointRed.get(i).status = -1;
            break;
        }
    }
}

```

Εικόνα 210 YC Seize 2

Χώρος εισόδου και εξόδου των τραίνων

Στον χώρο αυτό, κατά την άφιξη των τραίνων πρέπει να γίνει η επιλογή σε ποιόν σιδηρόδρομο από τους 5 πρέπει να πάνε. Για να επιτευχθεί αυτό στο block **moveToTrainUnloadingPosition** έγινε χρήση του παρακάτω κώδικα όπου ελέγχεται αν τα **WaitToUnload1**, **WaitToUnload2**, **WaitToUnload3**, **WaitToUnload4** και **WaitToUnload5** είναι ελεύθερα, επιλέγοντας το πρώτο διαθέσιμο. Ταυτόχρονα “αποθηκεύεται” στο τρένο (agent) σε ποια γραμμή βρίσκεται.

```

if (WaitToUnload1.size()==0) {
agent.trainParInjection=trainInject1;
agent.trainParDest=trainarrival1;
agent.trainLine=1;
}
else if (WaitToUnload2.size()==0) {
agent.trainParInjection=trainInject2;
agent.trainParDest=trainarrival2;
agent.trainLine=2;
}
else if (WaitToUnload3.size()==0) {
agent.trainParInjection=trainInject3;
agent.trainParDest=trainarrival3;
agent.trainLine=3;
}
else if (WaitToUnload4.size()==0) {
agent.trainParInjection=trainInject4;
agent.trainParDest=trainarrival4;
agent.trainLine=4;
}
else if (WaitToUnload5.size()==0) {
agent.trainParInjection=trainInject5;
agent.trainParDest=trainarrival5;
agent.trainLine=5;
}

```

Εικόνα 211 Move To Train Unloading Position

Στην συνέχεια με το **selectOutput11** κατευθύνεται στο αντίστοιχο **WaitToUnload1**

selectOutput11 - SelectOutput5

Use: ☐ Probabilities
☒ Conditions
☐ Exit number

Condition 1:

Condition 2:

Condition 3:

Condition 4:

Εικόνα 212 Select Output 11

Ακολούθως κατά την εισαγωγή στα blocks **WaitToUnload1**, **WaitToUnload2**, **WaitToUnload3**, **WaitToUnload4** και **WaitToUnload5** δίνεται η εντολή εισόδου στο σύστημα των εμπορευματοκιβωτίων. Τέλος κατά την ολοκλήρωση της διαδικασίας δίνεται εντολή μετατροπής της μεταβλητής **readyToLoad** σε true προκειμένου να έρθει το επόμενο τρέινο. Παράδειγμα τέτοιου κώδικα είναι:

On enter:

```
trainContainerY=100;

trainContainerX=685;
TrainContainerSource.inject(1);

trainContainerX=685+30;
TrainContainerSource.inject(1);

trainContainerX=685+60;
TrainContainerSource.inject(1);

trainContainerX=685+90;
TrainContainerSource.inject(1);

trainContainerX=685+120;
TrainContainerSource.inject(1);

trainContainerX=685+150;
TrainContainerSource.inject(1);

trainContainerX=685+180;
TrainContainerSource.inject(1);
```

On at exit:

```
readyToLoad1=true
```

Εικόνα 213 Wait To Unload 1

Σχετικά με το εμπορευματοκιβώτια, αρχικά κατά την εισαγωγή του στο σύστημα με την εντολή που δόθηκε παραπάνω, στο block **TrainContainerSource** τους δίνεται το χρώμα που έχουν. Τα χρώματα αυτά βρίσκονται σε 3 λίστες. Η πρώτη λίστα αφορά τα χρώματα που θα έχει το πρώτο τραίνο, η δεύτερη το δεύτερο και η τρίτη το τρίτο τραίνο. Στην συνέχεια ξανά μηδενίζεται η σειρά. Αυτή η διαδικασία γίνεται στο block **moveToTrainUnloadingPosition** με τον κώδικα

```
TrainCounter=TrainCounter+1;
if (TrainCounter==3){
TrainCounter=0;}
```

Εικόνα 214 move To Train Unloading Position

Ενώ ο χρωματισμός δίνεται στο block **TrainContainerSource** με την εντολή:

```
On at exit: agent.setContainerColor=randomFrom(collectionOfColorsContainerTrain.get(trainCounter));
```

Εικόνα 215 Train Container Source

Χώρος εισόδου και εξόδου των φορτηγών

Στο block **TrucksQueue** ελέγχεται η διαθεσιμότητα των θέσεων ξεφόρτωσης των φορτηγών μέσω των μεταβλητών **outputTruck1**, **outputTruck2** και **outputTruck3**.

Actions

On enter:

On at exit:

```
if (outputTruck1==false && outputTruck2==false && outputTruck3==false)
hold4.block();
else
hold4.unblock();
```

On exit:

On remove:

Εικόνα 216 Trucks Queue

Ακολουθώς, έχοντας εξασφαλίσει πλέον το ότι υπάρχει διαθεσιμότητα, μέσω του **selectUnloadPosition** το φορτηγό κατευθύνεται στη πρώτη διαθέσιμη θέση. Ταυτόχρονα, με το που περάσει αυτό το κομβικό σημείο το φορτηγό, η θέση γίνεται αυτομάτως μη διαθέσιμη. Τέλος, με τις μεταβλητές **FreeBusy1**, **FreeBusy2** και **FreeBusy3** αναρτάται σε μια οθόνη η κατάσταση των θέσεων για τους οδηγούς των φορτηγών.

selectUnloadPosition - SelectOutput5

Name: selectUnloadPosition ☒ Show name ☐ Ignore

Use: ☐ Probabilities ☒ Conditions ☐ Exit number

Condition 1: outputTruck1

Condition 2: outputTruck2

Condition 3: outputTruck3

Condition 4: false

Actions

On enter:

On exit 1:

```
outputTruck1=false;
FreeBusy1="busy";
```

On exit 2:

```
outputTruck2=false;
FreeBusy2="busy";
```

On exit 3:

```
outputTruck3=false;
FreeBusy3="busy";
```

Εικόνα 217 SelectUnloadPosition

Την στιγμή που το φορτηγό φεύγει από το block **UnloadingQueue** η μεταβλητή **outputTruck** γίνεται true προκειμένου να ελευθερωθεί η θέση, ενώ η μεταβλητή **FreeBusy** γίνεται free.

On exit:

```
outputTruck1=true;  
FreeBusy1="free";  
hold4.unblock();
```

Εικόνα 218 Unloading Queue

ΠΑΡΑΡΤΗΜΑ Δ Επεξήγηση του μεταβλητού χρόνου κύκλου

Ο Frankel (1980) παρουσίασε ένα τερματικό σταθμό εμπορευματοκιβωτίων σαν ένα δίκτυο λειτουργιών όπου κάθε σύνδεσμος του δικτύου αναπαριστά τη λειτουργία συγκεκριμένου τύπου εξοπλισμού. Κάθε τύπος εξοπλισμού έχει ένα συγκεκριμένο χρόνο κύκλου ο οποίος όμως δεν είναι πάντα σταθερός και παρουσιάζει αποκλίσεις. Όπως αναφέρει και αποδεικνύει, οι αποκλίσεις αυτές δεν επηρεάζουν μόνο την επίδοση και την ικανότητα του συγκεκριμένου εξοπλισμού αλλά επηρεάζουν και το χρόνο κύκλου άλλων τύπων εξοπλισμού με τους οποίους αλληλεπιδρά. Για την απόδειξη αυτής της συμπεριφοράς χρησιμοποιεί το εξής παράδειγμα:

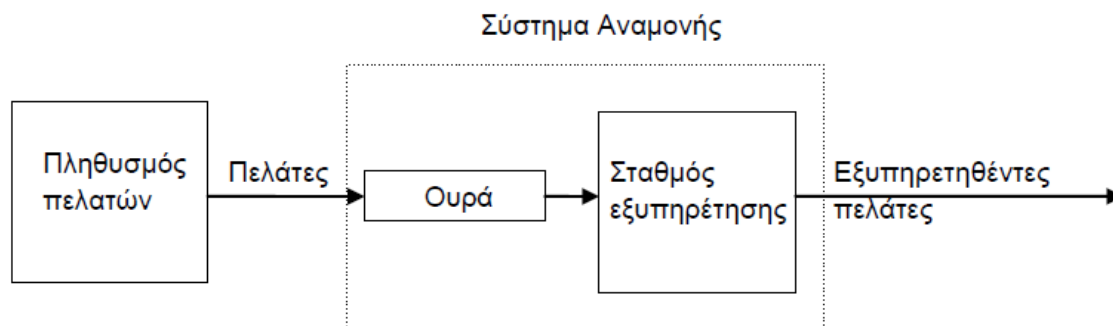
Δίνεται ότι μία γερανογέφυρα κρηπιδώματος έχει μέσο χρόνο κύκλου 120 δευτερόλεπτα και τυπική απόκλιση 10 δευτερόλεπτα. Η γερανογέφυρα εξυπηρετείται από ένα σύστημα ελκυστήρων που εκτελούν τις μεταφορές των εμπορευματοκιβωτίων από το κρηπίδωμα στους χώρους στοιβασίας. Το σύστημα των ελκυστήρων έχει μέσο χρόνο κύκλου 360 δευτερόλεπτα και εμφανίζει τυπική απόκλιση 90 δευτερόλεπτα. Θεωρώντας ότι το σύστημα αποτελείται από 3 ελκυστήρες τότε ο μέσος χρόνος κύκλου κάθε ελκυστήρα είναι 120 δευτερόλεπτα. Επομένως η ικανότητα μεταφοράς εμπορευματοκιβωτίων τόσο της γερανογέφυρας όσο και του ελκυστήρα είναι 30 εμπορευματοκιβώτια/ώρα. Ωστόσο κατά τη λειτουργία αυτού του συστήματος στο 38% του χρόνου, είτε η γερανογέφυρα θα περιμένει να την εξυπηρετήσει κάποιος ελκυστήρας είτε κάποιος ελκυστήρας θα περιμένει την γερανογέφυρα. Επομένως η πραγματική ικανότητα του συστήματος είναι 18,6 εμπορευματοκιβώτια/ώρα.

ΠΑΡΑΡΤΗΜΑ Ε Θεωρία αναμονής

Οι ουρές (queues) ή γραμμές αναμονής (waiting lines) είναι πολύ συνηθισμένο φαινόμενο, που δημιουργείται όταν η τρέχουσα ζήτηση για μια εξυπηρέτηση είναι μεγαλύτερη από την τρέχουσα δυναμικότητα του συστήματος, που παρέχει την εξυπηρέτηση. Οι αποφάσεις που έχουν σχέση με τον προσδιορισμό της δυναμικότητας είναι πολλές φορές δύσκολες, επειδή είναι συχνά αδύνατο να προβλέψουμε επακριβώς πότε θα αφιχθούν οι μονάδες στο σύστημα για να εξυπηρετηθούν και πόσος χρόνος θα χρειασθεί για την εξυπηρέτηση τους. Η ύπαρξη στο σύστημα μεγαλύτερης δυναμικότητας από αυτή που χρειάζεται, θα απαιτήσει μεγαλύτερο κόστος. Έτσι αντικειμενικός σκοπός είναι να βρεθεί μια οικονομική ισορροπία μεταξύ του κόστους εξυπηρέτησης και του κόστους που συνδέεται με την αναμονή για αυτή την εξυπηρέτηση. Η θεωρία αναμονής δεν λύνει το πρόβλημα αυτό άμεσα, όμως δίνει την πληροφόρηση που χρειάζεται για μια τέτοια απόφαση, με το να προσδιορίζει τα διάφορα χαρακτηριστικά του συστήματος σύμφωνα με το βιβλίο του Ξηροκώστα, Δ. (1990),

Η θεωρία αναμονής παρέχει ένα μεγάλο αριθμό μαθηματικών προτύπων για την περιγραφή των καταστάσεων των γραμμών αναμονής. Στη συνέχεια παρουσιάζονται μερικά από τα πιο βασικά πρότυπα.

Το απλούστερο σύστημα αναμονής είναι εκείνο που ακολουθεί την εξής διαδικασία : πελάτες που πρόκειται να εξυπηρετηθούν δημιουργούνται από ένα πληθυσμό πελατών, εισέρχονται σε ένα σύστημα αναμονής και ακολουθούν μια ουρά. Ο πελάτης εξυπηρετείται από ένα σταθμό εξυπηρέτησης και μετά φεύγει από το σύστημα αναμονής. Το σύστημα αυτό αναπαριστάται στην ακόλουθη Εικόνα 219.



Εικόνα 219 Το σύστημα αναμονής [πηγή: Ξηροκώστας, Δ. (1990)]

Ένα βασικό σύστημα αναμονής προσδιορίζεται από έξι χαρακτηριστικά μεγέθη τα οποία είναι σύμφωνα με Μπαλλής, Α (2014)

1. Κατανομή αφίξεων (arrival distribution)
2. Κατανομή εξυπηρετήσεων (service distribution)
3. Αριθμός παράλληλων σταθμών εξυπηρέτησης του συστήματος εξυπηρέτησης
4. Νόμος προτεραιότητας εξυπηρέτησης των πελατών (service discipline)
5. Μέγιστος επιτρεπόμενος αριθμός πελατών στο σύστημα αναμονής

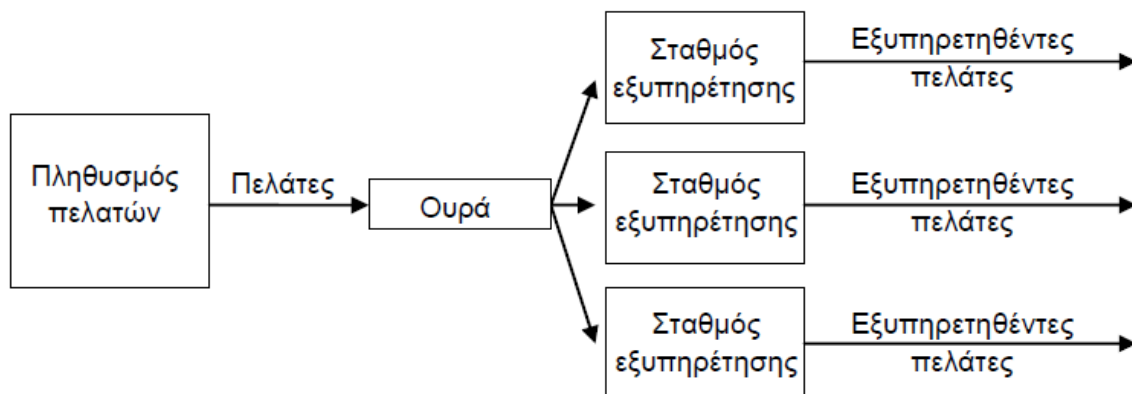
6. Μέγεθος πηγής προέλευσης των πελατών

Η κατανομή των αφίξεων μπορεί να εκφρασθεί είτε σαν κατανομή του αριθμού των αφικνούμενων πελατών στην μονάδα του χρόνου, οπότε έχουμε κατανομή διακριτής τυχαίας μεταβλητής, είτε σαν κατανομή των χρονικών διαστημάτων μεταξύ διαδοχικών αφίξεων, οπότε έχουμε κατανομή συνεχούς τυχαίας μεταβλητής. Αντίστοιχα και η κατανομή εξυπηρετήσεων μπορεί να εκφρασθεί ως διακριτή κατανομή ή ως τυχαία κατανομή. Μπορεί να γίνει η διάκριση δυο ακραίων περιπτώσεων στην κατανομή αφίξεων και εξυπηρετήσεων και είναι οι εξής :

- Σταθερές αφίξεις ή εξυπηρετήσεις, δηλαδή τα χρονικά διαστήματα μεταξύ διαδοχικών αφίξεων ή τα χρονικά διαστήματα εξυπηρετήσεων σταθερά.
- Εντελώς τυχαίες αφίξεις και εξυπηρετήσεις οι οποίες θα εκφρασθούν στη συνέχεια με την κατανομή Poisson, όταν πρόκειται για την κατανομή των αφικνούμενων ή εξυπηρετημένων πελατών στη μονάδα του χρόνου, και από την αρνητική εκθετική κατανομή, όταν πρόκειται για την κατανομή των χρονικών διαστημάτων μεταξύ αφίξεων ή των χρονικών διαστημάτων εξυπηρετήσεων. Η περίπτωση αυτή παρουσιάζει ιδιαίτερο ενδιαφέρον στη Θεωρία Αναμονής.

Μεταξύ των δύο ακραίων περιπτώσεων μπορούν να υπάρξουν αφίξεις ή εξυπηρετήσεις που ακολουθούν διάφορες κατανομές πιθανότητας, είτε εκφραζόμενες μαθηματικά, είτε όχι.

Ο αριθμός παράλληλων σταθμών εξυπηρέτησης του συστήματος εξυπηρέτησης αποτελεί βασικό χαρακτηριστικό του συστήματος. Η εξυπηρέτηση κάθε πελάτη αρχίζει και τελειώνει στον ίδιο σταθμό. Η μορφή του απεικονίζεται παρακάτω στην Εικόνα 220.



Εικόνα 220 Σύστημα αναμονής πολλών παράλληλων σταθμών εξυπηρέτησης πηγή: [πηγή: Ξηροκώστας, Δ. (1990)]

Ο νόμος προτεραιότητας εξυπηρέτησης των πελατών είναι ο τρόπος με τον οποίο επιλέγονται οι πελάτες στην ουρά για να εξυπηρετηθούν. Οι πιο συνηθισμένοι τρόποι εξυπηρέτησης είναι οι εξής :

- Εξυπηρέτηση κατά προτεραιότητα άφιξης (First In – First Out, FIFO). Είναι η πιο συνηθισμένη πειθαρχία η οποία μπορεί να παρατηρηθεί στην καθημερινότητα μας, κατά την οποία ο πρώτος πελάτης που φτάνει στο σύστημα εξυπηρετείται πρώτος.
- Εξυπηρέτηση με αντίστροφη προτεραιότητα (Last In – First Out, LIFO). Ο τρόπος αυτός εξυπηρέτησης συνηθίζεται σε βιομηχανικά προβλήματα, κατά τον οποίο ο πελάτης που φτάνει τελευταίος εξυπηρετείται πρώτος.

- Τυχαιά εξυπηρέτηση (Service In Random Order, SIRO). Ο τρόπος αυτός εξυπηρέτησης συναντάται επίσης σε βιομηχανικά προβλήματα κυρίως, κατά τον οποίο οι πελάτες επιλέγονται με τυχαιά σειρά προκειμένου να εξυπηρετηθούν.
- Ουρές με προτεραιότητες (Priority Queues). Στην εξυπηρέτηση με προτεραιότητες μια κατηγορία πελατών παίρνει προτεραιότητα από άλλη κατηγορία πελατών, που ήδη περιμένουν στην σειρά αναμονής.

Ωστόσο, μπορεί να παρουσιαστεί και μη τυποποιημένη συμπεριφορά των πελατών (GD). Η συμπεριφορά αυτή μπορεί να περιγραφεί από ιδιομορφίες όπως:

- μη προσχώρηση στην σειρά αναμονής όταν είναι μεγάλη (balking)
- προσχώρηση στην σειρά αναμονής αλλά αποχώρηση όταν ο χρόνος αναμονής υπερβεί μια τιμή (reneging)
- άφιξη των πελατών κατά ομάδες (bulk arrivals)

Ο μέγιστος επιτρεπόμενος αριθμός πελατών στο σύστημα καθώς και ο αριθμός των αφικνουμένων πελατών μπορεί να είναι απεριόριστος ή περιορισμένος και αυτό εξαρτάται από το σχεδιασμό του συστήματος.

Συμβολική παράσταση συστημάτων αναμονής

Τα συστήματα αναμονής ακολουθούν ένα συγκεκριμένο συμβολισμό, ο οποίος προτάθηκε από τον David George Kendall (1964), έτσι ώστε να υπάρξει μια κοινή “γλώσσα” αναφοράς για όλους. Προτάθηκε η μορφή $a/b/c$, όπου το a παριστάνει την κατανομή αφίξεων, το b την κατανομή εξυπηρετήσεων και το c τον αριθμό των παράλληλων σταθμών εξυπηρέτησης. Αργότερα το 1966 ο A. Lee πρόσθεσε δυο ακόμα χαρακτηριστικά τα d , e και το 1971 ο H. Taha πρόσθεσε και ένα έκτο χαρακτηριστικό το f . Το d παριστάνει το νόμο προτεραιότητας των πελατών στο σύστημα αναμονής, το e παριστάνει τον μέγιστο επιτρεπόμενο αριθμό πελατών στο σύστημα αναμονής και το f παριστάνει το μέγεθος της πηγής προέλευσης των πελατών. Η τελική μορφή του συμβολισμού είναι $(a/b/c) : (d/e/f)$ και τα σύμβολα που χρησιμοποιούνται για να το περιγράψουν δίνονται στον Πίνακα 30.

Χαρακτηριστικά	Σύμβολο	Ερμηνεία
Κατανομή αφίξεων (a)	M	Poisson
	D	Κανονική
	Ek	Erlang παραμέτρου k
	G	Γενική
Κατανομή εξυπηρέτησεων (b)	M	Εκθετική
	D	Κανονική
	Ek	Erlang παραμέτρου k
	G	Γενική
Αριθμός παράλληλων σταθμών εξυπηρέτησης (c)	1,2,...,∞	Αριθμός σταθμών εξυπηρέτησης
Νόμος προτεραιότητας στο σύστημα αναμονής (d)	FIFO	Εξυπηρέτηση κατά προτεραιότητα άφιξης
	LIFO	Εξυπηρέτηση με αντίστροφη προτεραιότητα
	SIRO	Τυχαία εξυπηρέτηση
	PRI	Ουρές με προτεραιότητα
	GD	Συμπεριφορά μη τυποποιούμενη
Μέγιστη δυνατότητα συστήματος (e) , (f)	1,2,...,∞	Αριθμός πελατών

Πίνακας 30 Συμβολισμός των συστημάτων αναμονής. [πηγή: Kendal, D.G., (1964)]

Ορολογία και Συμβολισμοί

Στην θεωρία αναμονής χρησιμοποιείται η εξής ορολογία:

- λ = μέσος ρυθμός αφίξεων (mean arrival rate)
- $1/\lambda$ = μέσος χρόνος μεταξύ διαδοχικών αφίξεων (expected or mean interarrival time)
- μ = μέσος ρυθμός εξυπηρέτησης (mean service rate)
- $1/\mu$ = μέσος χρόνος εξυπηρέτησεων (expected or mean service time)
- ρ = πυκνότητα κυκλοφορίας (traffic intensity)
- U = βαθμός χρησιμοποίησης των σταθμών εξυπηρέτησης (utilization factor)

Για να ισχύουν η θεωρία και οι τύποι που παρουσιάζονται στη συνέχεια θα πρέπει το σύστημα αναμονής να βρίσκεται σε ισορροπία. Δηλαδή θα πρέπει ο βαθμός χρησιμοποίησης να είναι μικρότερος της μονάδας ($U < 1$) ή ο ρυθμός αφίξεων να είναι μικρότερος του ρυθμού εξυπηρέτησης που έχουν οι σταθμοί εξυπηρέτησης ($\lambda < \mu$). Όταν ένα σύστημα έχει αρχίσει να λειτουργεί πρόσφατα, η κατάσταση του συστήματος θα επηρεαστεί σημαντικά από την αρχική κατάσταση και το σύστημα τότε βρίσκεται σε μεταβατική κατάσταση (transient conditions). Με την πάροδο του χρόνου όμως, το σύστημα γίνεται ανεξάρτητο της αρχικής κατάστασης και λέμε ότι έχει μεταβεί στη σταθερή κατάσταση (steady - state conditions), στην οποία οι ρυθμοί άφιξης λ και εξυπηρέτησης μ , είναι σταθεροί. Η θεωρία αναμονής εξετάζει τα συστήματα τα οποία έχουν μεταβεί σε σταθερή κατάσταση και χρησιμοποιεί τους

συμβολισμούς και τους τύπους που δίνονται στη συνέχεια. Τα μεγέθη αυτά αποτελούν τους δείκτες απόδοσης ενός συστήματος.

- N = ο αριθμός πελατών στο σύστημα αναμονής
- P_0 = η πιθανότητα να μην υπάρχουν πελάτες στο σύστημα
- P_n = η πιθανότητα να υπάρχουν n πελάτες στο σύστημα
- L = αναμενόμενος ή μέσος αριθμός πελατών στο σύστημα αναμονής, στην ουρά και στην εξυπηρέτηση (expected number of customers in the system or expected line length)
- L_q = αναμενόμενος ή μέσος αριθμός πελατών στη σειρά αναμονής (expected number of customers in the queue)
- W = αναμενόμενος ή μέσος χρόνος παραμονής ενός πελάτη στο σύστημα αναμονής (expected waiting time per customer in the system)
- W_q = αναμενόμενος ή μέσος χρόνος αναμονής ενός πελάτη στην σειρά αναμονής (expected waiting time per customer in the queue)

Στις αρχές του 1961 ο John D.C. Little απέδειξε τη σχέση που συνδέει τα μεσαία μεγέθους συστήματα σε σταθερή κατάσταση με τους σταθερούς χρόνους αναμονής των πελατών. Η σχέση αυτή είναι η εξής :

$$L = \lambda * W \quad (1)$$

Με βάση τη σχέση (1) προκύπτει και η εξής σχέση :

$$L_q = \lambda * W_q \quad (2)$$

Ακόμα, με την προϋπόθεση ότι ο μέσος χρόνος εξυπηρέτησης είναι σταθερός και ίσος με $1/\mu$ εξ' ορισμού προκύπτουν οι εξής σχέσεις :

$$W = W_q + \frac{1}{\mu} \quad (3)$$

$$L = L_q + \rho \quad (4)$$

Μοντέλα θεωρίας αναμονής

Τα βασικά μοντέλα που χρησιμοποιούνται για την περιγραφή συστημάτων διαχωρίζονται σε «Μοντέλα Πουασσόν» και «Μοντέλα μη Πουασσόν». Τα «Μοντέλα Πουασσόν» είναι τα μαθηματικά μοντέλα των συστημάτων αναμονής στα οποία τόσο οι αφίξεις όσο και οι εξυπηρετήσεις συμβαίνουν με την κατανομή Πουασσόν. Είναι απλά στην μαθηματική τους επεξεργασία γιατί η περιγραφή των καταστάσεων τους γίνεται μέσω του αριθμού $N(t) = n$ των πελατών στο σύστημα και μόνο. Αυτό συμβαίνει χάρη στη σπουδαία ιδιότητα της έλλειψης μνήμης της αρνητικής εκθετικής κατανομής, που ονομάζεται και «Μαρκοβιανή Ιδιότητα», και εξασφαλίζει ότι η μελλοντική εξέλιξη του συστήματος από μια χρονική στιγμή t και έπειτα εξαρτάται από την κατάσταση $N(t)$ και καθόλου από την κατάσταση του συστήματος πριν την χρονική στιγμή t . Στα «Μοντέλα μη Πουασσόν» δεν ισχύει αυτή η ιδιότητα καθώς η εξέλιξη του

συστήματος εκτός από το $N(t)$ εξαρτάται και από τον χρόνο που έχει παρέλθει του t γι αυτό και η επίλυση τους παρουσιάζει δυσκολίες. Ο Πίνακας 31 παρουσιάζει ορισμένους τύπους μοντέλων Πουασσόν και μη Πουασσόν.

Μοντέλα Πουασσόν	Μοντέλα μη Πουασσόν
$(M/M/1):(GD/\infty/\infty)$	$(M/G/1):(GD/\infty/\infty)$
$(M/M/s):(GD/\infty/\infty)$	$(M/D/1):(GD/\infty/\infty)$
$(M/M/1):(GD/N/\infty)$	$(M/D/s):(GD/\infty/\infty)$
$(M/M/s):(GD/N/\infty)$	$(D/M/s):(GD/\infty/\infty)$
$(M/M/1):(GD/\infty/K)$	$(E_k/M/1):(GD/\infty/\infty)$
$(M/M/s):(GD/\infty/K)$	$(M/E_l/1):(GD/\infty/\infty)$
	$(E_k/E_l/1):(GD/\infty/\infty)$
	$(E_k/E_l/s):(GD/\infty/\infty)$

Πίνακας 31 Τύποι μοντέλων Poisson και μη Poisson [πηγή: Θάνου Ε. (2015)]

Μοντέλο $(M / M / 1) : (GD / \infty / \infty)$

Το μοντέλο αυτό είναι το απλούστερο μοντέλο που μπορεί να υπάρξει και για να εφαρμοστεί πρέπει να πληρούνται οι εξής προϋποθέσεις :

- Η διαδικασία αφίξεων ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό αφίξεων λ . Ισοδύναμα ο μέσος χρόνος μεταξύ διαδοχικών αφίξεων $1/\lambda$ ακολουθεί την εκθετική κατανομή.
- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό εξυπηρέτησης μ . Ισοδύναμα ο μέσος χρόνος εξυπηρέτησεων $1/\mu$ ακολουθεί την εκθετική κατανομή.
- Υπάρχει ένας σταθμός εξυπηρέτησης ($s = 1$) και η εξυπηρέτηση γίνεται σε μια φάση.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Οι βασικοί τύποι που περιγράφουν το μοντέλο αυτό είναι οι εξής :

- Η πιθανότητα να μην υπάρχουν πελάτες στο σύστημα :

$$P_0 = (1 - \frac{\lambda}{\mu})$$

- Η πιθανότητα να υπάρχουν n πελάτες στο σύστημα :

$$P_n = \left(\frac{\lambda}{\mu}\right)^n \left(1 - \frac{\lambda}{\mu}\right)$$

- Μέσος αριθμός πελατών στο σύστημα αναμονής, στην ουρά και στην εξυπηρέτηση :

$$L = \frac{\lambda}{\mu - \lambda}$$

- Μέσος αριθμός πελατών στη σειρά αναμονής :

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

- Μέσος χρόνος παραμονής ενός πελάτη στο σύστημα αναμονής :

$$W = \frac{1}{\mu - \lambda} = \frac{L}{\lambda}$$

- Μέσος χρόνος αναμονής ενός πελάτη στην σειρά αναμονής :

$$W_q = \frac{1}{\mu(\mu - \lambda)} = \frac{L_q}{\lambda}$$

- Βαθμός χρησιμοποίησης του σταθμού εξυπηρέτησης :

$$U = \frac{\lambda}{\mu}$$

Μοντέλο $(M / M / s) : (GD / \infty / \infty)$

Το μοντέλο αυτό είναι το απλούστερο μοντέλο που μπορεί να υπάρξει και για να εφαρμοστεί πρέπει να πληρούνται οι εξής προϋποθέσεις :

- Η διαδικασία αφίξεων ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό αφίξεων λ . Ισοδύναμα ο μέσος χρόνος μεταξύ διαδοχικών αφίξεων $1/\lambda$ ακολουθεί την εκθετική κατανομή.
- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό εξυπηρέτησης μ . Ισοδύναμα ο μέσος χρόνος εξυπηρέτησεων $1/\mu$ ακολουθεί την εκθετική κατανομή.

- Υπάρχει ένας σταθμός εξυπηρέτησης ($s = 1$) και η εξυπηρέτηση γίνεται σε μια φάση.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Οι βασικοί τύποι που περιγράφουν το μοντέλο αυτό είναι οι εξής :

- Η πιθανότητα να μην υπάρχουν πελάτες στο σύστημα :

$$P_0 = \frac{1}{\left[\sum \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n \right] + \frac{1}{s!} \left(\frac{\lambda}{\mu} \right)^s \left(\frac{s\mu}{s\mu - \lambda} \right)}$$

- Η πιθανότητα να υπάρχουν πελάτες στο σύστημα :

$$P_n = \frac{1}{s! s^{n-s}} \left(\frac{\lambda}{\mu} \right)^n P_0 \quad \text{για } n > s$$

$$P_n = \frac{1}{n!} \left(\frac{\lambda}{\mu} \right)^n P_0 \quad \text{για } n \leq s$$

- Μέσος αριθμός πελατών στο σύστημα αναμονής, στην ουρά και στην εξυπηρέτηση :

$$L = \frac{\lambda \mu \left(\frac{\lambda}{\mu} \right)^s}{(s-1)!(s\mu - \lambda)^2} P_0 + \frac{\lambda}{\mu}$$

- Μέσος αριθμός πελατών στη σειρά αναμονής :

$$L_q = L - \frac{\lambda}{\mu}$$

- Μέσος χρόνος παραμονής ενός πελάτη στο σύστημα αναμονής:

$$W = \frac{L}{\lambda}$$

- Μέσος χρόνος αναμονής ενός πελάτη στην σειρά αναμονής :

$$W_q = \frac{L_q}{\lambda}$$

- Βαθμός χρησιμοποίησης των σταθμών εξυπηρέτησης :

$$U = \frac{\lambda}{s * \mu}$$

Μοντέλο $(E_k / M / 1) : (GD / \infty / \infty)$

Η διαδικασία αφίξεων ακολουθεί την κατανομή Erlang k σταδίων με ρυθμό αφίξεων λ.

- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό εξυπηρέτησης μ. Ισοδύναμα ο μέσος χρόνος εξυπηρέτησεων $1/\mu$ ακολουθεί την εκθετική κατανομή.
- Υπάρχει ένας σταθμός εξυπηρέτησης ($s = 1$) και η εξυπηρέτηση γίνεται σε μια φάση.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Οι βασικοί τύποι που περιγράφουν το μοντέλο αυτό είναι οι εξής :

- Μέσος αριθμός πελατών στο σύστημα αναμονής, στην ουρά και στην εξυπηρέτηση :

$$L = \sum_{n=1}^{\infty} n P_n$$

$$P_n = \begin{cases} 1 - \rho & \text{για } n=0 \\ \frac{z_o^k - 1}{z_o^{kn}} \rho & \text{για } n \geq 1 \end{cases}$$

- Μέσος αριθμός πελατών στη σειρά αναμονής

$$L_q = L - \frac{\lambda}{\mu}$$

- Μέσος χρόνος παραμονής ενός πελάτη στο σύστημα αναμονής :

$$W = \frac{L}{\mu}$$

- Μέσος χρόνος αναμονής ενός πελάτη στην σειρά αναμονής :

$$Wq = \frac{Lq}{\lambda}$$

- Βαθμός χρησιμοποίησης των σταθμών εξυπηρέτησης :

$$U = \frac{\lambda}{\mu}$$

Μοντέλο $(M / E_1 / 1) : (GD / \infty / \infty)$

- Η διαδικασία αφίξεων ακολουθεί την κατανομή Poisson με σταθερό μέσο ρυθμό αφίξεων λ . Ισοδύναμα ο μέσος χρόνος μεταξύ διαδοχικών αφίξεων $1/\lambda$ ακολουθεί την εκθετική κατανομή.
- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Erlang I σταδίων με ρυθμό εξυπηρέτησεων μ .
- Υπάρχει ένας σταθμός εξυπηρέτησης ($s = 1$) και η εξυπηρέτηση γίνεται σε μια φάση.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Οι βασικοί τύποι που περιγράφουν το μοντέλο αυτό είναι οι εξής :

- Μέσος αριθμός πελατών στο σύστημα αναμονής, στην ουρά και στην εξυπηρέτηση :

$$L = \frac{\lambda}{\mu} + \frac{1+k}{2k} * \frac{\lambda^2}{\mu(\mu-\lambda)}$$

- Μέσος αριθμός πελατών στη σειρά αναμονής :

$$Lq = L - \frac{\lambda}{\mu}$$

- Μέσος χρόνος παραμονής ενός πελάτη στο σύστημα αναμονής :

$$W = \frac{L}{\lambda}$$

- Μέσος χρόνος αναμονής ενός πελάτη στην σειρά αναμονής :

$$Wq = \frac{Lq}{\lambda}$$

- Βαθμός χρησιμοποίησης των σταθμών εξυπηρέτησης :

$$U = \frac{\lambda}{\mu}$$

Μοντέλο $(E_k / E_1 / 1) : (GD / \infty / \infty)$

- Η διαδικασία αφίξεων ακολουθεί την κατανομή Erlang k σταδίων με ρυθμό αφίξεων λ.
- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Erlang l με ρυθμό εξυπηρετήσεων μ.
- Υπάρχει ένας σταθμός εξυπηρέτησης (s = 1) και η εξυπηρέτηση γίνεται σε μια φάση.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Λόγω της πολυπλοκότητας της επίλυσης των τύπων του μοντέλου αυτού, η βιβλιογραφία δίνει συνήθως σε μορφή πινάκων τις τιμές του μέσου χρόνου αναμονής των πελατών στην ουρά, για διάφορους βαθμούς χρησιμοποίησης των σταθμών εξυπηρέτησης [Ξηροκώστας, Δ. (1990)].

Μοντέλο $(E_k / E_1 / s) : (GD / \infty / \infty)$

- Η διαδικασία αφίξεων ακολουθεί την κατανομή Erlang k σταδίων με ρυθμό αφίξεων λ.
- Η διαδικασία εξυπηρέτησης ακολουθεί την κατανομή Erlang l με ρυθμό εξυπηρετήσεων μ.
- Υπάρχουν περισσότεροι από έναν σταθμοί εξυπηρέτησης s.
- Υπάρχει μία μόνο ουρά στην οποία γίνεται εξυπηρέτηση κατά προτεραιότητα άφιξης (FIFO).
- Η χωρητικότητα του συστήματος αναμονής είναι απεριόριστη.
- Η πηγή των πελατών προέρχεται από ένα άπειρο πλήθος πελατών.
- Δεν παρατηρούνται μη τυποποιημένες συμπεριφορές όπως η αποχώρηση από την ουρά.
- Ο μέσος ρυθμός άφιξης είναι μικρότερος από τον μέσο ρυθμό εξυπηρέτησης ($\lambda < \mu$) ώστε το σύστημα να είναι σε ισορροπία.

Λόγω της πολυπλοκότητας της επίλυσης των τύπων του μοντέλου αυτού, η βιβλιογραφία δίνει σε μορφή πινάκων τις τιμές του μέσου χρόνου αναμονής των πελατών στην ουρά, για διάφορους βαθμούς χρησιμοποίησης των σταθμών εξυπηρέτησης [Ξηροκώστας, Δ. (1990)].
