



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ Μ/Υ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΕΦΑΡΜΟΓΗ ΤΩΝ ΜΟΝΤΕΛΩΝ ΑΝΑΜΟΝΗΣ ΣΤΗ ΝΑΥΤΙΛΙΑ

ΚΡΕΠΙΑΣ - ΒΛΑΧΟΣ ΓΕΩΡΓΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
(ΔΗΜΗΤΡΙΟΣ ΑΣΚΟΥΝΗΣ ΚΑΘΗΓΗΤΗΣ)

ΗΜΕΡΟΜΗΝΙΑ (ΙΟΥΛΙΟΣ 2020)

Περίληψη

Στη παρούσα εργασία πραγματοποιείται εκτενής βιβλιογραφική ανασκόπηση για τα μοντέλα αναμονής από το έργο του A.K. Erlang μέχρι σύγχρονες εφαρμογές στη Ναυτιλία. Όμως συνδυάζοντας τα αποτελέσματα της βιβλιογραφικής έρευνας προέκυψαν νέα μοντέλα που βελτιώνουν και αναβαθμίζουν τα υπάρχοντα και βρίσκουν εφαρμογή σε πτυχές και προβλήματα της Ναυτιλίας όπως οι αργοπορημένες αφίξεις πλοίων, που μέχρι στιγμής αντιμετωπίζονται δίχως κανόνες, και προσφέρουν τις απαραίτητες μεθόδους και αποτελέσματα έτσι ώστε να αποδοθούν σε αυτά αποτελεσματικές λύσεις. Μεγάλο τμήμα της εργασίας επικεντρώνεται στα μοντέλα αναμονής έτσι ώστε να εφαρμοστούν με τη προσήκουσα μεθοδολογία όπως απόδειξη της εγκυρότητας, επιβεβαίωση της καταλληλότητας και επικύρωση της αξιοπιστίας τους.

Αντίστοιχα μεγάλο τμήμα αφιερώνεται και στη διαμόρφωση ενός σύνθετου προβλήματος για την εφαρμογή τους. Θέτοντας το δίλλημα του διάπλου μέσα από τρικυμία για την αποφυγή καθυστερήσεων στο συμφωνηθέν ναυλοσύμφωνο προκύπτουν δύο επιλογές που απαιτούν ανάλυση για τη λήψη της απόφασης. Αντλώντας πραγματικά ωκεανογραφικά και ατμοσφαιρικά δεδομένα γίνεται προσομοίωση των θαλάσσιων καταστάσεων με έντονους κυματισμούς για τη διερεύνηση της επίδρασής τους στα πλοία. Προτείνεται έτσι το φαινόμενο της τεχνητής γήρανσης του πλοίου για να συνδέσει τις αρνητικές συνέπειες της έκθεσης του πλοίου σε έντονους κυματισμούς με τη μείωση της εμπορικής του ζωής και των οικονομικών επιπτώσεων που απορρέουν. Το υπολογιστικό σκέλος της εργασίας ολοκληρώνεται με την οικονομική σύγκριση των δύο διαφορετικών σχημάτων κόστους που αρθρώνονται

για τις δύο επιλογές με βάση των δεικτών της Καθαρής Παρούσης Αξίας (ΚΠΑ) και του Εσωτερικού Βαθμού Απόδοσης (ΕΒΑ).

Για τα ποικίλα προβλήματα που παρουσιάστηκαν και αντιμετωπίστηκαν στη πάροδο της εργασίας αξιοποιήθηκε η γλώσσα προγραμματισμού Python, η συνεισφορά της οποίας παρατίθεται κάθε φορά σε μορφή κώδικα με σχολιασμούς και αποτελεί αναπόσπαστο κομμάτι της εργασίας.

Όπως αποδεικνύεται από τα αποτελέσματα μέσω της τελικής οικονομικής σύγκρισης τα εργαλεία που δημιουργήθηκαν και σφυρηλατήθηκαν στη παρούσα εργασία δύναται να λάβουν σημαντική θέση στη λήψη μακροπρόθεσμων αποφάσεων των διαχειριστών πλοίων όπως η κατάστρωση στρατηγικής για δρομολόγηση στόλου.

Abstract

In the present work, extensive Literature review of the queuing models is carried out from the work of A.K Erlang to modern applications at Shipping. Combining the results of the Literature review, new models emerged that improve and upgrade the existing ones and find applications in aspects and problems of Shipping such as late arrivals of ships, which up until today were treated without any frame of reference. These models offer the necessary methods so as to provide more efficient solutions. A large part of this paper focuses on queuing models so they can be applied in Shipping with the proper methodologies, such as proof of validity, confirmation of suitability and validation of their reliability.

Respectively a large part of the present work formulates a complex problem to implement the findings. Posing the dilemma of crossing through storms to avoid delays in the charter party agreement, two options arise that require analysis in order to obtain a decision. Acquiring actual oceanographic data and simulating the Sea State conditions of intense weather allows to investigate their effect on ships. For the study of the above problem the phenomenon of artificial aging of the ship is proposed to link the negative effects of the ship's exposure to strong waves with the reduction of its commercial life and the resulting economic consequences. The computational part of the work is completed with the financial comparison of the two different cost schemes articulated for the two options based on the Net Present Value (NPV) and Internal Rate of Return (IRR).

The various problems presented in the present work were addressed by utilizing the programming language Python, the contribution of which is presented each time in the form of code with comments and constitutes an integral part of the work.

As evidenced by the results through the final economic comparison the mathematical tools that were created in this work can take an important role in the long-term decisions making of ship operators such as developing a fleet routing strategy.

Ευχαριστίες

Ξεκινώντας θα ήθελα να ευχαριστήσω το Δρ. Παναγιώτη Κοκκινάκο που μοιράστηκε μαζί μου τις ιδέες και τις σκέψεις του και που με βοήθησε καθ' όλη τη διάρκεια της εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω τον Καθ. Δημήτριο Ασκούνη για την βοήθεια που μου προσέφερε καθώς και για την απαραίτητη καθοδήγηση όπως και τους αναγκαίους πόρους για την εκπλήρωση των στόχων της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου για τη συνεχή υποστήριξή τους δίχως την οποία δεν θα είχα καταφέρει το ίδιο αποτέλεσμα.

Πίνακας Περιεχομένων

1	Εισαγωγή	12
2	Ανανεωτική Θεωρία & Απαριθμήτρια Ανανεωτική Διαδικασία	17
2.1	Η Αμνήμονη Ιδιότητα	20
2.2	Η Διαδικασία Poisson	26
2.3	Η Γεωμετρική Διαδικασία	33
3	Μεθοδολογία Παραγωγής Τεχνητών Δεδομένων	38
4	Προσαρμογή G.I. κατανομών σε Ιστογράμματα	56
5	Εφαρμογή των M/M/m G.I/M/m & M/G/m μοντέλων	64
5.1	Εφαρμογή του M/M/m μοντέλου αναμονής	64
5.2	Εφαρμογή του G.I/M/m μοντέλου αναμονής	71
5.3	Εφαρμογή του M/G/m μοντέλου αναμονής	101
6	Σύγκριση μεταξύ δύο διαφορετικών σχημάτων κόστους	110
6.1	Απόκτηση και Διαχείριση Ωκεανογραφικών Δεδομένων	112
6.2	Δημιουργία Χαρακτηριστικών των Sea States	122
6.3	Ανάλυση σχήματος κόστους - 1η επιλογή του Αποφασίζων	140
6.3.1	Αυξημένη Κατανάλωση Καυσίμων	141
6.3.2	Αυξημένα Έξοδα Συντήρησης	146
6.3.3	Μειωμένα Έσοδα λόγω Τεχνητής Γήρανσης	151
6.4	Ανάλυση σχήματος κόστους - 2η επιλογή του Αποφασίζων	155
6.4.1	Αυξημένη κατανάλωση των Βοηθητικών Συστημάτων	156

6.4.2	Μείωση Ναύλου	157
6.4.3	Επιπρόσθετα κόστη ελλιμενισμού	159
6.4.4	Μειωμένα Έσοδα λόγω Αδράνειας	160
7	Οικονομική Ανάλυση της Εμπορικής Ζωής του Πλοίου	161
7.1	Οικονομική Σύγκριση με βάση το δείκτη ΚΠΑ	162
7.2	Οικονομική Σύγκριση με βάση το δείκτη ΕΒΑ	163
7.3	Αξιολόγηση Αποτελεσμάτων & Συμπεράσματα	164
	Βιβλιογραφία	165

Ευρετήριο Διαγραμμάτων

1	Σύστημα Αναμονής	14
2	Η πιθανότητα της τομής μη ασυμβίβαστων ενδεχομένων	23
3	Διαμερισμός χρόνου αναμονής	25
4	Πραγματοποίηση της απαριθμήτριας διαδικασίας αφίξεων $N(t)$	28
5	Κατανομή συνολικού χρόνου παραμονής	38
6	Διάγραμμα ροής μεθόδου	41
7	Ημιτονοειδής συνάρτηση	43
8	Διάγραμμα ροής της μεθόδου	46
9	Poisson no 1	47
10	Poisson no 2	47
11	Gauss no 1	47
12	Gauss no 2	47
13	Σύγκριση διαφορετικού τύπου κατανομών	48
14	Διάγραμμα ροής της μεθόδου	52
15	Ιστόγραμμα Αφίξεων	53
16	Ιστόγραμμα Αναχωρήσεων	53
17	Ισοζύγιο Αφίξεων - Αναχωρήσεων	55
18	Σύγκριση συχνοτήτων με κατανομές	57
19	χ^2 Test	58
20	Ιστόγραμμα και Gaussian KDE	60
21	Gaussian kernels and KDE	62
22	Gaussian kernels and KDE $h=1$	63

23	Gaussian kernels and KDE $h=0.5$	63
24	Εβδομαδιαίο Ιστόγραμμα Αφίξεων	65
25	Εβδομαδιαίο Ιστόγραμμα Αναχωρήσεων	66
26	Σύγκριση συχνοτήτων	70
27	Σύγκριση συχνοτήτων	72
28	Συναρτήσεις πιθανότητας του χρόνου αναμονής	85
29	Συναρτήσεις πιθανότητας του μήκους ουράς	85
30	Συναρτήσεις πιθανότητας του χρόνου αναμονής	89
31	Συναρτήσεις πιθανότητας του μήκους ουράς	89
32	Συναρτήσεις πιθανότητας του χρόνου αναμονής	94
33	Συναρτήσεις πιθανότητας του μήκους ουράς	95
34	Συναρτήσεις πιθανότητας του χρόνου αναμονής	99
35	Συναρτήσεις πιθανότητας του μήκους ουράς	99
36	Γραμμική Παρεμβολή	103
37	Ιστόγραμμα χρόνων εξυπηρέτησης για το λιμάνι Yokkaichi	109
38	Παράδειγμα χρονοσειράς για Βόρειο Ατλαντικό	113
39	Παγκόσμιο κατανομή ανέμου στην επιφάνεια της θάλασσας	114
40	Υπολογισμός fetch για κάθε θάλασσα/Ωκεανό	115
41	JONSWAP spectrum για 10 RSS	125
42	Αφαίρεση ακραίων τιμών	125
43	Ιστόγραμμα c_i πριν την αφαίρεση ακραίων τιμών	126
44	Ιστόγραμμα c_i μετά την αφαίρεση ακραίων τιμών	126
45	JONSWAP για WCSST	127
46	Γραμμική σχέση τεχνητής γήρανσης	132

47	Ημερήσια επιπρόσθετη κατανάλωση	144
48	Ημερήσιο επιπρόσθετο κόστος καυσίμων	145
49	Ετήσιο κόστος συντήρησης	146
50	Ετήσιο κόστος συντήρησης 2	147
51	Ετήσιο επιπρόσθετο κόστος συντήρησης	147
52	Μηνιαίες τιμές Ναύλων	153
53	Ποσοστιαία μείωση ναύλου	157
54	Οικονομική Σύγκριση με δείκτη NPV	162
55	Οικονομική Σύγκριση με δείκτη IRR NPV>0	163
56	Οικονομική Σύγκριση με δείκτη IRR NPV<0	163

Ευρετήριο Πινάκων

1	Ταξινόμηση Στοχαστικών Διαδικασιών	18
2	Υπολογισμός ετήσιων αφίξεων	54
3	Κατανομή συχνοτήτων αφίξεων κάθε bin στις ημέρες του έτους	54
4	Αντιστοίχιση ελάχιστου και μέγιστου bin σε κάθε βαθμό δραστηριοποίησης	55
5	Πινακοποιημένη μορφή για εύρεση του ορίου του χ^2 test	58
6	Πίνακας δεδομένων από τη MarineTraffic	64
7	Δεδομένα για το λιμάνι Incheon	68
8	Αποτελέσματα για το λιμάνι Incheon	69
9	Δεδομένα για το λιμάνι Yokkaichi	70
10	Αποτελέσματα για το λιμάνι Yokkaichi	71
11	Σύγκριση αποτελεσμάτων για μοντέλο M/M/4 και G.I/M/4 με είσοδο Poisson	84
12	Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο Poisson	88
13	Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο KDE	94
14	Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο Γεωμετρική	98
15	Γραμμική Παρεμβολή	103
16	Αποτελέσματα για το μοντέλο αναμονής M/G.E./2	106
17	Αποτελέσματα για το μοντέλο αναμονής M/G.E./2 κατά Hokstad	108
18	χ^2 test για το μοντέλο αναμονής M/G.E./2	109
19	Μειωμένα έσοδα λόγω μείωσης ζωής πλοίου	154
20	Μειωμένα έσοδα από μείωση ναύλου	158
21	Ετήσια μειωμένα έσοδα	160

1 Εισαγωγή

Ο σκοπός αυτής της εργασίας είναι να προσφέρει ένα μέτρο για τη σημαντικότητα της επίδρασης των ουρών αναμονής όπου ένα πλοίο αντιμετωπίζει κατά τη διάρκεια των ναυλωμένων δρομολογίων του. Κατά τη διάρκεια ενός απλού δρομολογίου του ένα πλοίο αντιμετωπίζει διάφορα “Operations” που δύναται να μοντελοποιηθούν χρησιμοποιώντας τη θεωρία των ουρών αναμονής. Τα “Operations” περιλαμβάνουν διαδικασίες όπως εφοδιασμός καυσίμων, φορτοεκφορτώσεις εντός λιμένα και διελεύσεις καναλιών. Επομένως, είναι μεγίστης σημασίας στον ιδιοκτήτη και εν γένει στον διαχειριστή του πλοίου να καταλάβει τις προεκτάσεις που τα συστήματα αναμονής φέρουν για τα δρομολόγια ενός πλοίου.

Υπάρχει μεγάλο εύρος και έντονη δραστηριότητα στη βιβλιογραφία σχετικά με τη θεωρία αναμονής στη ναυτιλία, αλλά στη πλειοψηφία των περιπτώσεων χρησιμοποιείται για να αντληθούν αποτελέσματα για το διαχειριστή του συστήματος αναμονής όπως λιμάνια, σταθμοί ανεφοδιασμού καυσίμων, κανάλια κ.λπ. Παρ’ όλα αυτά, από αυτές τις βιβλιογραφικές αναφορές μπορούν να αντληθούν χρήσιμες πληροφορίες για την ανάλυση αυτής της εργασίας. Αρχικά, από [1] οι συγγραφείς προσφέρουν μια σύγκριση μεταξύ δύο συστημάτων αναμονής διαφορετικής μορφής και εφαρμόζουν μια μέθοδο βελτιστοποίησης έτσι ώστε να προσδιορίζουν τις βέλτιστες υποδομές για το λιμάνι υπό εξέταση. Ενώ, στο [2] εφαρμόζεται μια μέθοδος για τη επιβεβαίωση της καταλληλότητας των συναρτήσεων πυκνότητας πιθανότητας (π.χ. Poisson) και κατά πόσο αυτές προσαρμόζονται στα δεδομένα της ανάλυσης των συστημάτων αναμονής που αποτελεί ένα καίριο βήμα για τη μελέτη των συστημάτων αναμονής. Επιπροσθέτως, στο [3] οι ουρές αναμονής των καναλιών διαφοροποιούνται από εκείνες στα λιμάνια και στους σταθμούς ανεφοδιασμού καυσίμων λόγω του ότι τα πλοία εξυπηρετούνται σε ομάδες

των 5 ή 10 αντί για ένα τη φορά. Αρκετά που παρατηρήθηκαν στη βιβλιογραφία όπως η μέθοδος επιβεβαίωσης και οι ουρές αναμονής σε ομάδες (χύδην ουρές) ενσωματώνονται στη παρούσα εργασία διότι ολοκληρώνουν την ανάλυση της.

Όπως προαναφέρθηκε στις πλειοψηφία των περιπτώσεων η θεωρία ουρών αναμονής χρησιμοποιείται από τους διαχειριστές του συστήματος αναμονής, παρ' όλα αυτά στη περίπτωση μας θα χρησιμοποιηθεί η θεωρία ουρών αναμονής για να αποκομίσουμε αποτελέσματα για το διαχειριστή του πλοίου. Στην πρώτη ενότητα θα γίνει μια εισαγωγή στις βασικές έννοιες της θεωρίας ουρών αναμονής και θα εξαχθούν κάποια βασικά αποτελέσματα, στις περισσότερες των περιπτώσεων υπό τη μορφή εξισώσεων, που θα χρησιμοποιηθούν αργότερα για την ανάλυση των συστημάτων αναμονής.

Τα συστήματα αναμονής αποτελούνται από δύο χώρους, το χώρο αναμονής στον οποίο διαμορφώνεται η ουρά και το χώρο εξυπηρέτησης, όπως απεικονίζεται και στην παρακάτω εικόνα. Για να μην χαθούν στοιχεία από το γενικότερο ορισμό του συστήματος αναμονής και για να μην εκτρέπουμε από το σκοπό αυτής της εργασίας, στην παρακάτω εικόνα, λαμβάνεται υπόψη η περίπτωση απείρου μήκους ουράς και m εξυπηρετητών παράλληλων μεταξύ τους. Με την ολοκλήρωση την εξυπηρέτησης των πελατών, αναχωρούν από το σύστημα και οι πελάτες από τον χώρο αναμονής παίρνουν τη θέση τους.

Το πιθανό πλήθος ουρών αναμονής που δύναται να σχηματιστεί ποικίλει από μία ουρά για όλους τους εξυπηρετητές όπως απεικονίζεται στη παραπάνω εικόνα έως μία ουρά για κάθε εξυπηρετητή. Επιπροσθέτως το μήκος της ουράς στη θεωρία ουρών αναμονής δύναται

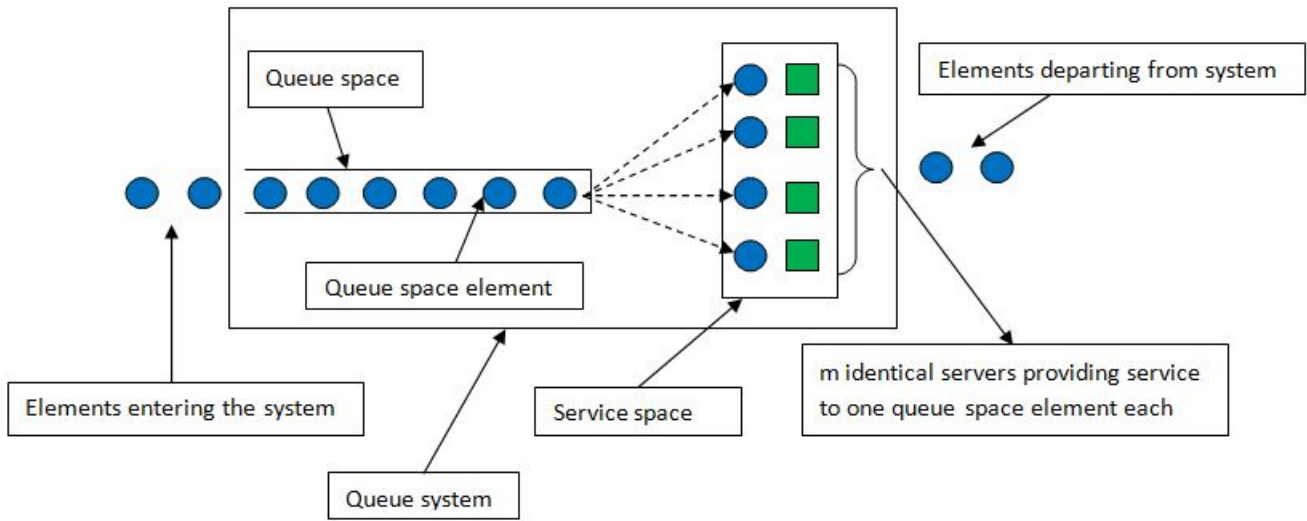


Figure 1: Σύστημα Αναμονής

να είναι πεπερασμένο ή απείρως μεγάλο. Κατά την εφαρμογή της θεωρίας των ουρών αναμονής σε αυτήν την εργασία η εστίαση θα είναι στις ουρές που δεν φέρουν ανώτατο όριο στο μήκος τους, δηλαδή το μήκος τους είναι απείρως μεγάλο, και το πλήθος ουρών που θα σχηματίζεται θα είναι μια κοινή για όλους τους εξυπηρετητές.

Επιπροσθέτως, υπάρχουν διάφοροι τρόποι με τους οποίους οι πελάτες μπορούν να εξυπηρετηθούν, με άλλα λόγια η προτεραιότητα ή η πειθαρχία της ουράς αναμονής ενδέχεται να λάβει πολλές μορφές. Για παράδειγμα, οι πελάτες μπορούν να εξυπηρετηθούν σύμφωνα με τη πειθαρχία “first come first served basis” (FCFS) ή “last come first served basis” (LCFS). Για το σκοπούς αυτής της εργασίας η πειθαρχία FCFS θα χρησιμοποιηθεί.

Σύμφωνα με τη βιβλιογραφία χρησιμοποιείται μια καθορισμένη συντομογραφία για την αναπαράσταση των διαφόρων μορφών που λαμβάνουν τα συστήματα αναμονής. Αυτή

η συντομογραφία συχνά αναφέρεται ως “Kendall’s notation” η οποία καθιερώθηκε το 1953 και γνωρίζει πλέον τη μεγαλύτερη αποδοχή. Η συντομογραφία λαμβάνει τη μορφή A/B/C/D/E στην οποία κάθε γράμμα αντιπροσωπεύει μια ιδιότητα του συστήματος αναμονής. Τα πρώτα δύο γράμματα αντιπροσωπεύουν τη συνάρτηση πυκνότητας πιθανότητας ή τη συνάρτηση κατανομής του αριθμού αφίξεων και αναχωρήσεων. Το τρίτο γράμμα αντιπροσωπεύει τον αριθμό των σημείων εξυπηρέτησης (εξυπηρετητών) που βρίσκονται εντός του χώρου εξυπηρέτησης, ενώ το τέταρτο τη μέγιστη χωρητικότητα του μήκους της ουράς αναμονής. Το τελευταίο γράμμα, αντιπροσωπεύει τη πειθαρχία της ουράς με τη έννοια που προαναφέρθηκε. Είναι κοινώς αποδεκτό να παραλείπονται τα τελευταία δύο γράμματα αφού οι τιμές τους είναι συνήθως οι ίδιες για μια μεγάλη ποικιλία εφαρμογών. Επομένως και στη περίπτωση αυτής της εργασίας οι τιμές που λαμβάνουν είναι απείρως μεγάλο μήκος ουράς και FCFS πειθαρχία.

Για παράδειγμα η ουρά αναμονής M/M/1 μοντελοποιείται ως ένα σύστημα αναμονής με συνάρτηση πυκνότητας πιθανότητας Poisson για τις αφίξεις και για τις αναχωρήσεις και με ένα μόνο σημείο εξυπηρέτησης. Όπως προαναφέρθηκε είναι ευκόλως εννοούμενα ότι η ουρά αναμονής είναι απείρως μήκους και πειθαρχίας FCFS. Το γράμμα M ενδεικνύει την αμνήμονη ιδιότητα της εκθετικής κατανομής που συνδυάζεται με την κατανομή Poisson στην διαδικασία Poisson. Ένα άλλο παράδειγμα είναι η M/G.I/m ουρά που μοντελοποιείται ως ένα σύστημα αναμονής με Poisson αφίξεις, μια γενική κατανομή για αναχωρήσεις και m παράλληλα σημεία εξυπηρέτησης. Από την βιβλιογραφία [4] οι συνήθεις κατανομές πιθανότητας που προτιμώνται είτε για τον αριθμό αφίξεων ή για τον αριθμό αναχωρήσεων είναι η Poisson (M), η Deterministic (D) για την οποία ο αριθμός αφίξεων ή αναχωρήσεων είναι σταθερός και η πιθανότητα ίση με 1 και τέλος η “General Independent” (G.I) για

την οποία η κατανομή είναι μια οποιαδήποτε συνάρτηση που τα χρονικά διαστήματα των διαδοχικών αφίξεων ή αναχωρήσεων είναι ισόνομες και ανεξάρτητες τυχαίες μεταβλητές.

2 Ανανεωτική Θεωρία & Απαριθμήτρια Ανανεωτική Διαδικασία

Στοχαστικά λέγονται τα συστήματα των οποίων οι συναρτήσεις χαρακτηρίζονται από αβεβαιότητα.

Με άλλα λόγια, στοχαστικό είναι κάθε σύστημα η μελλοντική συμπεριφορά του οποίου δεν μπορεί να προβλεφθεί επακριβώς, αλλά μπορεί να προσεγγιστεί μόνο με τα μέσα που προσφέρει η θεωρία πιθανοτήτων. Τα μέσα της θεωρίας πιθανοτήτων αποτελούνται από τις στοχαστικές διαδικασίες που αξιοποιούνται ως μαθηματικά μοντέλα έτσι ώστε να προσομοιώσουν τα στοχαστικά συστήματα.

Στοχαστική διαδικασία είναι κάθε οικογένεια τυχαίων μεταβλητών $X(t) : t \in T$ που ορίζεται από τη τριπλέτα (Ω, F, P) . Σε αυτόν τον ορισμό, T είναι ο παραμετρικός χώρος που μπορεί να είναι είτε αριθμήσιμο ή υπεραριθμήσιμο σύνολο. Στην πρώτη περίπτωση, η στοχαστική διαδικασία θεωρείται ότι είναι διακριτού χρόνου και στη δεύτερη περίπτωση συνεχούς χρόνου. Ω είναι ο δειγματικός χώρος F είναι το δυναμοσύνολο του δειγματικού χώρου Ω και P είναι η συνάρτηση πιθανότητας τέτοια ώστε αν A είναι ένα ενδεχόμενο τότε $P : F \ni A \rightarrow P(A) \in R$. Κάθε πιθανή τιμή του X λέγεται κατάσταση της στοχαστικής διαδικασίας και ο χώρος όλων των εφικτών καταστάσεων, S δύναται να είναι είτε αριθμήσιμο ή υπεραριθμήσιμο σύνολο. Επομένως συνδυάζοντας όλες τις παραπάνω περιπτώσεις, οι στοχαστικές διαδικασίες μπορούν να ταξινομηθούν σε τέσσερις κατηγορίες όπως φαίνεται στο παρακάτω πίνακα.

Ουσιαστικά κάθε στοχαστική διαδικασία $X(t) : t \in T$ που ορίζεται στο πιθανοχώρο (Ω, F, P) είναι μια συνάρτηση τέτοια ώστε

$$X : T \times \Omega \ni (t, \omega) \rightarrow X(t, \omega) \in S$$

Χώρος Καταστάσεων (S)	Παραμετρικός Χώρος (T)	
	Διακριτός	Συνεχής
Διακριτός	Διακριτή Στοχαστική Διαδικασία με διακριτό χώρο καταστάσεων	Συνεχής Στοχαστική Διαδικασία με διακριτό χώρο καταστάσεων
Συνεχής	Διακριτή Στοχαστική Διαδικασία με συνεχή χώρο καταστάσεων	Συνεχής Στοχαστική Διαδικασία με συνεχή χώρο καταστάσεων

Table 1: Ταξινόμηση Στοχαστικών Διαδικασιών

1. Για συγκεκριμένο $t = t_0$ και $\omega = \omega_0$
 Η $X(t_0, \omega_0)$ είναι η συγκεκριμένη κατάσταση στην οποία βρίσκεται η στοχαστική διαδικασία
2. Για συγκεκριμένο $t = t_0$,
 Η $X(t_0, \omega)$ είναι ακριβώς η τυχαία μεταβλητή $X(t_0)$.
3. Για συγκεκριμένο $\omega = \omega_0$,
 Η $X(t, \omega_0)$ είναι μια πραγματοποίηση της στοχαστικής διαδικασίας και περιγράφει την εξέλιξή της στο χρόνο.

Ένας άλλος τρόπος για την κατηγοριοποίηση των στοχαστικών διαδικασιών είναι σύμφωνα με τη μορφή της στοχαστικής εξάρτησης των μελών της $X(t) : t \in T$. Υιοθετώντας αυτή τη βάση κατηγοριοποίησης υπάρχουν τέσσερις κύριες κατηγορίες [5].

A Στάσιμες διαδικασίες είναι οι στοχαστικές διαδικασίες με την ιδιότητα ότι $\forall n \in \mathbb{N}$ και για $\tau > 0$, οι τυχαίες μεταβλητές $X(t_n)$ και $X(t_n + \tau)$, είναι στοχαστικά

ισοδύναμες και έτσι ως αρχή του χρόνου μπορεί να επιλεγεί οποιαδήποτε χρονική στιγμή t .

B Οι διαδικασίες με ανεξάρτητες προσαυξήσεις είναι στοχαστικές διαδικασίες με την ιδιότητα ότι $\forall n \in N$ οι τυχαίες μεταβλητές $X(t_n) - X(t_{n-1})$ είναι ανεξάρτητες.

C Ανανεωτική διαδικασία και απαριθμήτρια ανανεωτική διαδικασία είναι οι στοχαστικές διαδικασίες με οικογένειες τυχαίων μεταβλητών $\{Z_n : n \in N_0\}$ και $\{M(t) : t \geq 0\}$ αντίστοιχα, όπου Z_n και $M(t)$ δίνονται από:

$$Z_n = \sum_{i=1}^n X_i$$

$$M(t) = \max\{n : Z_n \leq t\}$$

και X_i είναι μια αλληλουχία ανεξάρτητων, ισόνομων και μη αρνητικών τυχαίων μεταβλητών. Αυτό ο τύπος διαδικασιών μαζί με τις μαρκοβιανές διαδικασίες που έπονται αντιστοιχούν στα στοχαστικά μοντέλα που θα χρησιμοποιηθούν έτσι ώστε να προσομοιωθεί η συμπεριφορά των συστημάτων αναμονής που θα εξεταστούν. Υπό αυτές τις συνθήκες, οι τυχαίες μεταβλητές X_i αντιστοιχούν στους ενδιάμεσους χρόνους αφίξεων και αναχωρήσεων του συστήματος αναμονής και $M(t)$ στον αριθμό των αφίξεων και αναχωρήσεων του συστήματος αναμονής εντός του χρονικού διαστήματος $[0,t]$.

D Οι μαρκοβιανές διαδικασίες είναι στοχαστικές διαδικασίες $X(t) : t \in T$ με την ιδιότητα ότι δεδομένης της τιμής της τυχαίας μεταβλητής $X(t)$ που συμβολίζει τη παρούσα κατάσταση, οι τυχαίες μεταβλητές $X(u) : u \geq t$ που συμβολίζουν τη μελλοντική κατάσταση είναι στοχαστικά ανεξάρτητες από τις τυχαίες μεταβλητές $X(s) : s < t$ που συμβολίζουν τη παρελθούσα κατάσταση. Ισοδύναμα, η πιθανότητα μετάβασης

σε οποιαδήποτε μελλοντική κατάσταση, δεδομένης της παρούσας κατάστασης, είναι ανεπηρέαστη από τη πλεονάζουσα πληροφορία της παρελθούσας κατάστασης. Αυτή η χαρακτηριστική ιδιότητα λέγεται μαρκοβιανή ή αμνήμονη ιδιότητα και εκφράζεται ισοδύναμα από τη σχέση :

$$P(X(t+u) = j | X(s), s < t) = P(X(t+u) = j | X(t)) \\ (t, u \geq 0, j \in S)$$

Στη περίπτωση που η μεταβλητή s είναι ίση με τη χρονική στιγμή t και θέτοντας $X(t) = i$, η δεσμευμένη πιθανότητα $P(X(t+u) = j | X(t) = i)$ αναφέρεται ως μεταβατική πιθανότητα της διαδικασίας από τη κατάσταση i τη χρονική στιγμή t στη κατάσταση j τη χρονική στιγμή $t+u$. Οι μαρκοβιανές διαδικασίες που φέρουν διακριτό χώρο καταστάσεων αναφέρονται ως μαρκοβιανές αλυσίδες που κατηγοριοποιούνται περαιτέρω σε διακριτές και συνεχείς μαρκοβιανές αλυσίδες ανάλογα με τη φύση του παραμετρικού χώρου.

2.1 Η Αμνήμονη Ιδιότητα

Είναι σημαντικό σε αυτό το σημείο να υπάρξει μια λεπτομερής ανάλυση της αμνήμονης ιδιότητας και των μαθηματικών συναρτήσεων που φέρουν αυτή την ιδιότητα. Η μοναδική τυχαία μεταβλητή που έχει την αμνήμονη ιδιότητα είναι η εκθετική που έχει την συνάρτηση πυκνότητας πιθανότητας $f(x) = \lambda \cdot e^{-\lambda x}$ που σχετίζεται με τη συνάρτηση κατανομής πιθανότητας μέσω της συνάρτησης $F(x) = \int_0^x f(x) \cdot dx$. Για κάθε συνεχή τυχαία μεταβλητή

Χ η πιθανότητα $P(x < X \leq x + h)$ μπορεί εναλλακτικά να αποδοθεί από την συνάρτηση πυκνότητας πιθανότητας της τυχαίας μεταβλητής και τον παράγοντα διόρθωσης $o(h)$.

$$f(x) = F'(x) = \lim_{h \rightarrow 0} \frac{F(x+h) - F(x)}{h}$$

Από τη βασική θεωρία πιθανοτήτων, δεδομένης μιας συνάρτησης κατανομής πιθανότητας $F(x) = P(X < x)$ και τη συνάρτηση επιβίωσης της $F^c(x) = P(X > x)$ είναι γνωστό ότι :

$$\begin{aligned} P(X > x) &= P(\{X \leq x\}^c) \\ &= 1 - P(X \leq x) \\ &= 1 - F(x) \end{aligned}$$

και

$$\begin{aligned} P(x < X \leq y) &= P(X \leq y, X > x) \\ &= P(X \leq y, (X \leq x)^c) \\ &= P(X \leq y) - P(X \leq y, X \leq x) \\ &= P(X \leq y) - P(X \leq x) \\ &= F(y) - F(x) \quad (x < y) \end{aligned}$$

Επομένως, μπορεί να εξαχθεί ότι :

$$f(x) = \lim_{h \rightarrow 0} \frac{P(x < X \leq x + h)}{h}$$

Και ισοδύναμα για κάθε απειροστό μικρό $h > 0$:

$$P(x < X \leq x + h) = f(x) \cdot h + o(h)$$

Ο παράγοντας διόρθωσης είναι μια συνάρτηση με την εξής ιδιότητα :

$$\lim_{h \rightarrow 0} \frac{o(h)}{h} = 0$$

Με άλλα λόγια, $o(h)$ συγκλίνει πιο γρήγορα στο μηδέν απ' ό τι το h και άρα η ποσότητα $o(h)$ μπορεί να θεωρηθεί αμελητέα συγκριτικά με το h .

Για να γίνει καλύτερα κατανοητή η αμνήμονη ιδιότητα στο κλάδο της ναυτιλίας προσφέρεται ένα παράδειγμα. Αν θεωρηθεί ότι η τυχαία μεταβλητή X δηλώνει το χρόνο αναμονής μέχρις ότου αφιχθεί ένα πλοίο στο σύστημα αναμονής (π.χ. λιμάνι) και $f(x)$ καθώς και $F^c(x)$ είναι οι συναρτήσεις πυκνότητας πιθανότητας και επιβίωσης αντίστοιχα, τότε ο δεσμευμένος ρυθμός άφιξης πλοίων μπορεί να προσδιοριστεί ως $\varphi(x) = \frac{f(x)}{F^c(x)}$.

Για ακόμα μια φορά είναι απαραίτητο, πριν προχωρήσουμε παρακάτω, να χρησιμοποιήσουμε κάποιες βασικές αρχές από τη θεωρία πιθανοτήτων και πιο συγκεκριμένα από τις δεσμευμένες πιθανότητες. Άρα, χρησιμοποιώντας αυτή τη γνώση λαμβάνουμε ότι η δεσμευμένη πιθανότητα ενός ενδεχομένου A δεδομένου ότι έχει ήδη συμβεί ένα ενδεχόμενο B είναι $P(A|B) = \frac{P(A \cap B)}{P(B)}$.

Η πιθανότητα άφιξης ενός πλοίου εντός του απειροστού χρονικού διαστήματος $(x, x+h]$ δεδομένου ότι κανένα πλοίο δεν έχει αφιχθεί εντός του χρονικού διαστήματος $(0,x]$ ή ισοδύναμα ότι ο χρόνος αναμονής είναι x , είναι:

$$P(x < X \leq x + h | X > x) = \frac{P(x < X \leq x + h, X > x)}{P(X > x)}$$

Από τη παρακάτω εικόνα εφόσον η τομή των δύο ενδεχομένων είναι η απεικονιζόμενη κοινή περιοχή, μπορεί εύκολα να εξαχθεί ότι :

$$P(x < X \leq x + h, X > x) = P(x < X \leq x + h).$$

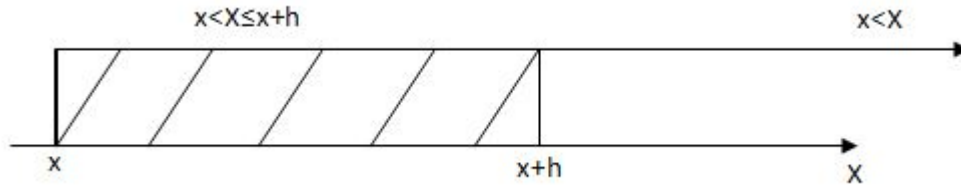


Figure 2: Η πιθανότητα της τομής μη ασυμβίβαστων ενδεχομένων

Συνεχίζοντας με την απόδειξη προκύπτουν τα παρακάτω :

$$\begin{aligned} \frac{P(x < X \leq x + h)}{P(X > x)} &= \frac{F(x + h) - F(x)}{F^c(x)} \\ &= \frac{f(x) \cdot h + o(h)}{F^c(x)} \\ &= \varphi(x) \cdot h + o(h) \end{aligned}$$

Είναι ευδιάκριτο από το τύπο που προκύπτει, μέσω της απόδειξης, ότι εν γένει η παραπάνω πιθανότητα είναι συνάρτηση του χρόνου αναμονής x . Παρ' όλα αυτά στη περίπτωση που η τυχαία μεταβλητή X ακολουθεί την εκθετική κατανομή, τότε :

$$\varphi(x) = \frac{\lambda \cdot e^{-\lambda \cdot x}}{e^{-\lambda \cdot x}} = \lambda$$

και

$$P(x < X \leq x + h | X > x) = \lambda \cdot h + o(h)$$

Το γεγονός ότι ο δεσμευμένος ρυθμός είναι ανεξάρτητος του x δηλώνει την αμνήμονη ιδιότητα της εκθετικής κατανομής. Συμπληρωματικά, γίνεται ευκόλως αντιληπτό ότι η πιθανότητα άφιξης ενός πλοίου εντός του απειροστού χρονικού διαστήματος $(x, x+h]$ είναι επίσης ανεξάρτητη του χρόνου αναμονής x .

Ένας δεύτερος τρόπος απόδειξης της αμνήμονης ιδιότητας είναι :

$$\begin{aligned} P(X > x + t | X > x) &= \frac{P(X > x + t)}{P(X > x)} \\ &= \frac{e^{-\lambda \cdot (x+t)}}{e^{-\lambda \cdot x}} \\ &= e^{-\lambda \cdot t} \\ &= P(X > t) \end{aligned}$$

Αυτή η μαθηματική έκφραση δηλώνει το ότι αν ο χρόνος αναμονής για την άφιξη του πρώτου πλοίου ακολουθεί την εκθετική κατανομή τότε αν διαμερίσουμε το χρόνο αναμονής μέχρι την άφιξη του πρώτου πλοίου σε δύο διακριτά χρονικά τμήματα, με οποιαδήποτε δυνατή αναλογία, το δεύτερο χρονικό τμήμα θα ακολουθεί επίσης την εκθετική κατανομή.

Αυτό σημαίνει ότι ξεκινώντας από οποιοδήποτε χρονικό σημείο t_1 εντός του χρονικού διαστήματος $(0, T)$, ανεξάρτητα από το χρονικό διάστημα μέχρις ότου φτάσουμε σε αυτό το χρονικό σημείο από το αρχικό χρονικό σημείο 0, το υπολειπόμενο χρονικό διάστημα $(t_1, T]$ μέχρι την άφιξη του πρώτου πλοίου θα ακολουθεί την ίδια κατανομή πιθανότητας με το χρόνο αναμονής μέχρι την άφιξη του πρώτου πλοίου.

Το t_1 μπορεί να λάβει οποιαδήποτε τιμή στο $(0, T)$ όπου T είναι το χρονικό σημείο μέχρι την άφιξη του πρώτου πλοίου.

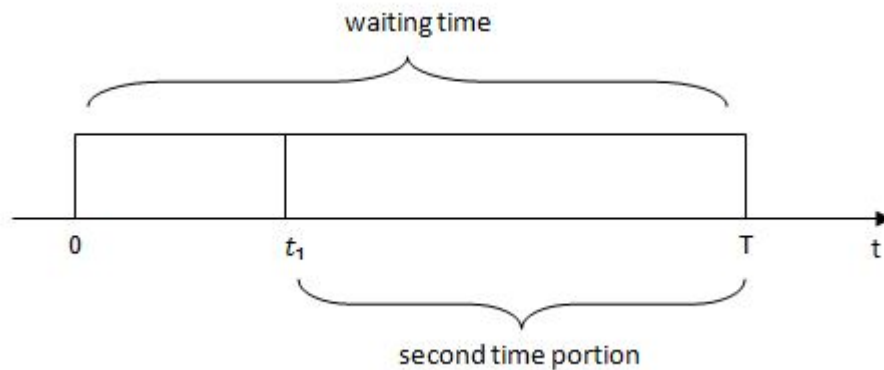


Figure 3: Διαμερισμός χρόνου αναμονής

Όσο αφορά της διακριτές τυχαίες μεταβλητές, η μόνη συνάρτηση που κατέχει την αμνήμονη ιδιότητα είναι η γεωμετρική κατανομή, το οποίο ήταν αναμενόμενο εφόσον είναι το διακριτό ανάλογο της εκθετικής κατανομής. Η συνάρτηση μάζας πιθανότητας, το ισοδύναμο της συνάρτησης πυκνότητας πιθανότητας για συνεχείς τυχαίες μεταβλητές, δίνεται από $f(x) = (1 - p)^x \cdot p$ και η αντίστοιχη συνάρτηση επιβίωσης είναι $F_c(x) = (1 - p)^x$, όπου p είναι η πιθανότητα επιτυχίας ενός πειράματος “Bernoulli” και n ο αριθμός των επιτυχιών. Για την απόδειξη της αμνήμονης ιδιότητας, η δεύτερη από τις προαναφερθέντες αποδείξεις χρησιμοποιείται παρακάτω :

$$\begin{aligned}
 P(X > x + t | X > x) &= \frac{P(X > x + t)}{P(X > x)} \\
 &= \frac{(1 - p)^{x+t}}{(1 - p)^x} \\
 &= (1 - p)^t \\
 &= P(X > t)
 \end{aligned}$$

2.2 Η Διαδικασία Poisson

Μια ειδική περίπτωση των ανανεωτικών και μαρκοβιανων διαδικασιών είναι η διαδικασία Poisson, που είναι σημαντικό να εστιάσουμε με ένα παράδειγμα. Θεωρούμαι ως παράδειγμα ένα λιμένα που εξυπηρετεί πλοία με τη φορτοεκφόρτωση των εμπορευμάτων τους. Αν Z_i είναι τα διαδοχικά χρονικά σημεία άφιξης πλοίων εντός του χρονικού διαστήματος $(0, \infty)$ και $N(t)$ ο συνολικός αριθμός αφίξεων πλοίων εντός του χρονικού διαστήματος $[0, t]$, τότε οι στοχαστικές διαδικασίες :

$$\{N(t) : t \geq 0\}$$

$$N(t) = \max\{n \in N_0 : Z_n \leq t\}, N(0) = 0$$

και

$$\{Z_n : n \in N_0\}, Z_0 = 0$$

λέγονται απαριθμήτρια διαδικασία αφίξεων και διαδικασία αφίξεων αντίστοιχα. Όταν οι ενδιάμεσοι χρόνοι αφίξεων ή ισοδύναμα τα χρονικά διαστήματα μεταξύ δύο διαδοχικών αφίξεων πλοίων $T_n = Z_n - Z_{n-1}, n \in N$ είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές τότε η διαδικασία αφίξεων είναι μια ανανεωτική διαδικασία. Αλλά, κάθε διαδικασία αφίξεων $\{Z_n : n \in N_0\}$ με την ιδιότητα ότι οι ενδιάμεσοι χρόνοι αφίξεων είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές που ακολουθούν την εκθετική κατανομή είναι διαδικασία Poisson. Επομένως, η διαδικασία Poisson είναι μια ανανεωτική διαδικασία.

Συμπληρωματικά στο προαναφερθέν ορισμό της διαδικασίας Poisson, κάθε απαριθμήτρια διαδικασία αφίξεων $\{N(t) : t \geq 0\}$ για την οποία για κάθε απειροστό χρονικό διάστημα $(t, t+h]$ ισχύει η παρακάτω ιδιότητα είναι επίσης διαδικασία Poisson :

$$P(N(t, t+h) = k | N(s), 0 \leq s \leq t) = \begin{cases} \lambda \cdot h + o(h) & \text{if } k = 1 \\ 1 - \lambda \cdot h + o(h) & \text{if } k = 0 \\ o(h) & \text{if } k \geq 2 \end{cases} \quad (1)$$

Αυτή η σχέση δηλώνει ότι για κάθε απειροστό χρονικό διάστημα $(t, t+h]$ υπάρχουν δύο δυνατές εκβάσεις. Είτε να υπάρξει μια μοναδική άφιξη πλοίου ($k=1$) ή να μην υπάρξει άφιξη πλοίου ($k=0$). Είναι σημαντικό να σημειωθεί ότι οι προαναφερθέν εκβάσεις είναι σταθερές και ανεξάρτητες από τις αφίξεις $\{N(s), 0 \leq s \leq t\}$ που έλαβαν χώρα στο παρελθοντικό χρονικό διάστημα $(0, t]$. Άρα, ο αριθμός των αφίξεων σε διαφορετικά και μη επικαλυπτόμενα χρονικά διαστήματα είναι ανεξάρτητες τυχαίες μεταβλητές.

Η παρακάτω εικόνα απεικονίζει μια πραγματοποίηση της στοχαστικής διαδικασίας $\{N(t) : t \geq 0\}$. Ο άξονας x δηλώνει το διανυόμενο χρόνο και ο άξονας y τον αριθμό των αφιχθέντων πλοίων.

Θεωρώντας ότι ο αριθμός των αφίξεων εντός του χρονικού διαστήματος $(t, t+h]$ είναι ίσος με τη διαφορά μεταξύ των αφιχθέντων πλοίων στο χρονικό διάστημα $(0, t+h]$ και $(0, t]$ λαμβάνουμε τον εξής μαθηματικά τύπο :

$$N(t, t+h) = N(t+h) - N(t)$$

Μια εναλλακτική μορφή της εξίσωσης (1) είναι :

$$P(N(t+h) = N(t) + k | N(s), 0 \leq s \leq t)$$

θέτοντας τη μεταβλητή $s = t$ και αφήνοντας $N(t) = i, N(t) + k = j$ όπου $i \in N_0$, η παραπάνω εξίσωση αναδιαμορφώνεται σε :

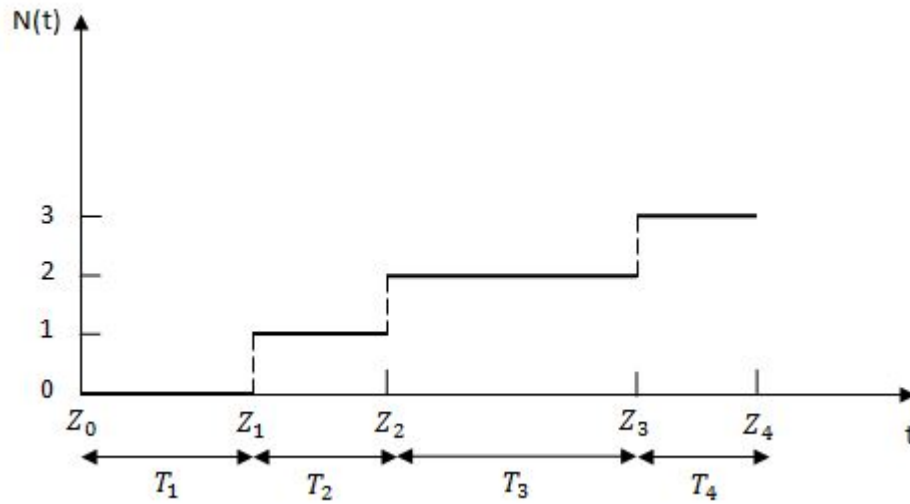


Figure 4: Πραγματοποίηση της απαριθμητριας διαδικασίας αφίξεων $N(t)$

$$P(N(t+h) = j | N(t) = i) = \begin{cases} \lambda \cdot h + o(h) & \text{if } j = i + 1 \\ 1 - \lambda \cdot h + o(h) & \text{if } j = i \\ o(h) & \text{if } j > i + 1 \\ 0 & \text{if } j < i \end{cases}$$

που φέρει την ίδια ιδιότητα με τις μαρκοβιανές αλυσίδες και άρα η απαριθμητρια διαδικασία αφίξεων είναι μια μαρκοβιανή αλυσίδα συνεχούς χρόνου με χώρο καταστάσεων το N_0 .

Στη περίπτωση μιας Poisson απαριθμητριας διαδικασίας αφίξεων $N(t) : t \geq 0$, θα αποδειχθεί ότι η τυχαία μεταβλητή $N(t)$ ακολουθεί τη κατανομή Poisson.

$$p_n(t) = P(N(t) = n) = e^{-\lambda \cdot t} \frac{(\lambda \cdot t)^n}{n!}$$

Η ανωτέρα μαθηματική σχέση δηλώνει ότι η $p_n(t)$ είναι η πιθανότητα να υπάρξουν n αφίξεις πλοίων εντός του χρονικού διαστήματος $(0, t]$. Για μια πιο ολοκληρωμένη προσέγγιση

δίνεται μια λεπτομερής απόδειξη της παραπάνω σχέσης. Διαμερίζοντας το χρονικό διάστημα $(0, t+h]$ σε $(0, t]$ και $(t, t+h]$ και παρατηρώντας ότι το ενδεχόμενο 0 αφίξεις εντός του χρονικού διαστήματος $(0, t+h]$ είναι ισοδύναμο με το ενδεχόμενο μηδενικών αφίξεων εντός του χρονικού διαστήματος $(0, t]$ και μηδενικών αφίξεων εντός του χρονικού διαστήματος $(t, t+h]$, τα οποία είναι στοχαστικά ανεξάρτητα σύμφωνα με τον ορισμό της απαριθμητριας διαδικασίας Poisson, η πιθανότητα $p_0(t+h)$ λαμβάνει τη παρακάτω μορφή :

$$\begin{aligned} p_0(t+h) &= P(N(t+h) = 0) \\ &= P(N(t) = 0, N(t, t+h) = 0) \\ &= P(N(t) = 0) \cdot P(N(t, t+h) = 0) \\ &= p_0(t) \cdot [1 - \lambda \cdot h + o(h)] \\ &= p_0(t) \cdot (1 - \lambda \cdot h) + o(h) \end{aligned}$$

Ακολουθώντας μια παραπλήσια λογική, το ενδεχόμενο n αφίξεων εντός του χρονικού διαστήματος $(0, t+h]$ είναι ισοδύναμο με τα ανεξάρτητα ενδεχόμενα $\{n-k$ αφίξεις εντός του χρονικού διαστήματος $(0, t]\}$ και $\{k$ αφίξεις εντός του χρονικού διαστήματος $(t, t+h]\}$ για κάποιο k ($k=0, 1, 2, \dots, n$). Επομένως, για $n \geq 1$:

$$\begin{aligned} p_n(t+h) &= P(N(t+h) = n) \\ &= P\left(\bigcup_{k=0}^n \{N(t) = n-k\} \cap \{N(t, t+h) = k\}\right) \\ &= \sum_{k=0}^n P(N(t) = n-k) \cdot P(N(t, t+h) = k) \\ &= p_n(t) \cdot P(N(t, t+h) = 0) + p_{n-1}(t) \cdot P(N(t, t+h) = 1) + o(h) \\ &= p_n(t) \cdot [1 - \lambda \cdot h + o(h)] + p_{n-1}(t) \cdot [\lambda \cdot h + o(h)] + o(h) \\ &= p_n(t) \cdot (1 - \lambda \cdot h) + p_{n-1}(t) \cdot \lambda \cdot h + o(h) \end{aligned}$$

Δημιουργώντας τη πρώτη παράγωγο της συνάρτησης πυκνότητας πιθανότητας προκύπτει το παρακάτω ζεύγος εξισώσεων :

$$\frac{p_0(t+h) - p_0(t)}{h} = -\lambda \cdot p_0(t) + \frac{o(h)}{h} \quad (2)$$

$$\frac{p_n(t+h) - p_n(t)}{h} = -\lambda \cdot p_n(t) + \lambda \cdot p_{n-1}(t) + \frac{o(h)}{h} \quad (3)$$

Καθώς $h \rightarrow 0$ οι εξισώσεις (2) και (3) λαμβάνουν τη μορφή :

$$p'_0(t) = -\lambda \cdot p_0(t) \quad (4)$$

$$p'_n(t) = -\lambda \cdot p_n(t) + \lambda \cdot p_{n-1}(t) \quad (5)$$

Λόγω του ότι $N(0) = 0$, ή με άλλα λόγια τη χρονική στιγμή $t = 0$ ο αριθμός των αφιχθέντων πλοίων είναι 0, οι κατάλληλες αρχικές συνθήκες είναι :

$$p_0(0) = 1 \quad p_n(0) = 0 \quad n \in N$$

Για την επίλυση του ζεύγους των διαφορικών εξισώσεων (4) και (2.3) και για τον προσδιορισμό της συνάρτησης πυκνότητας πιθανότητας $\{p_n(t) : n \in N_0\}$ χρησιμοποιήθηκε η πιθανογεννήτρια συνάρτησή της :

$$P(t, z) = E[z^{N(t)}] = \sum_{n=0}^{\infty} p_n(t) \cdot z^n \quad (t \geq 0, |z| \leq 1)$$

Πιο συγκεκριμένα πολλαπλασιάζοντας τις εξισώσεις (4) και (2.3) με z_0 και z_n αντίστοιχα

και προσθέτοντας για όλα τα $n \in N_0$ καταλήγουμε στη παρακάτω εξίσωση :

$$\begin{aligned} \sum_{n=0}^{\infty} p'_n(t) \cdot z^n &= -\lambda \cdot \sum_{n=0}^{\infty} p_n(t) \cdot z^n + \lambda \cdot \sum_{n=0}^{\infty} p_{n-1}(t) \cdot z^n \\ &= -\lambda \cdot P(t, z) + \lambda \cdot z \cdot P(t, z) \\ &= -\lambda \cdot (z - 1) \cdot P(t, z) \end{aligned}$$

Θεωρώντας το z ως σταθερά, προκύπτει η παρακάτω γενική λύση της διαφορικής εξίσωσης :

$$\begin{aligned} P'(t, z) &= -\lambda \cdot (1 - z) \cdot P(t, z) \\ \frac{P'(t, z)}{P(t, z)} &= -\lambda \cdot (1 - z) \\ [\ln P(t, z)]' &= -\lambda \cdot (1 - z) \\ P(t, z) &= c(z) \cdot e^{-\lambda \cdot t \cdot (1-z)} \end{aligned}$$

Με αρχική συνθήκη :

$$P(0, z) = \sum_{n=0}^{\infty} p_n(0) \cdot z^n = 1 \cdot z^0 = 1$$

Τελικά αντικαθιστώντας για $c(z) = 1$ προκύπτει η συγκεκριμένη λύση της διαφορικής εξίσωσης :

$$P(t, z) = e^{-\lambda \cdot t \cdot (1-z)}$$

που είναι η πιθανογεννήτρια συνάρτηση της κατανομής Poisson με παράμετρο $\lambda \cdot t$. Λόγω της 1-1 αντιστοιχίας μεταξύ πιθανογεννητριών συναρτήσεων και κατανομών η τυχαία

μεταβλητή $N(t)$ ακολουθεί τη Poisson ($\lambda \cdot t$). Αν το t τεθεί ίσο με τη μονάδα ($t=1$) τότε το λ εκφράζει το μέσο ρυθμό αφίξεων στη μονάδα του χρόνου.

2.3 Η Γεωμετρική Διαδικασία

Ακολουθώντας μια ανάλογη συλλογιστική με την απόδειξη της Διαδικασίας Poisson, θα αποδειχθεί ότι υπάρχει μια παρόμοια διαδικασία με τα ίδια χαρακτηριστικά και ιδιότητες, με τη διαφορά ότι ως βάση έχει την γεωμετρική κατανομή παρά την εκθετική. Εδώ, θεωρείται ότι ως βάση της διαδικασίας εννοείται οι ενδιάμεσοι χρόνοι αφίξεων ή οι χρόνοι εξυπηρέτησης να ακολουθούν την γεωμετρική κατανομή ανάλογα με την εφαρμογή. Οι δύο αυτές διαδικασίες ήταν αναμενόμενο να έχουν κοινά καθώς οι αντίστοιχες κατανομές τους είναι οι μόνες που μοιράζονται την αμνήμονη ιδιότητα.

Λαμβάνοντας υπ' όψη ότι στις διακριτές συναρτήσεις δεν γίνεται να χρησιμοποιηθεί διαφορικός λογισμός, το ελάχιστο διάστημα εντός του οποίου δύναται να πραγματοποιηθεί ένα γεγονός, δηλαδή μία άφιξη ή αναχώρηση, είναι το $(t, t+h]$, όπου $h=1$. Άρα, για τη νέα (απαριθμητρία) διαδικασία $\{B(t) : t \geq 0\}$ που έχει ως βάση τη γεωμετρική κατανομή ισχύει η σχέση :

$$B(N(t, t+h) = k | N(s), 0 \leq s \leq t) = \begin{cases} p & \text{if } k = 1 \\ 1 - p = q & \text{if } k = 0 \\ 0 & \text{if } k \geq 2 \end{cases}$$

Όπως γίνεται ευκόλως αντιληπτό, αντί του όρου λ της διαδικασίας Poisson, στη Γεωμετρική διαδικασία υπάρχει ο όρος p που προκύπτει με παρόμοιο τρόπο όπως ήδη έχει γίνει και στη

παράγραφο (2.1) :

$$\begin{aligned}
 \frac{P(x < X \leq x + h)}{P(X > x)} &= \frac{F(x + h) - F(x)}{F^c(x)} \\
 &= \frac{1 - q^{h+n+1} - 1 + q^{n+1}}{q^{n+1}} \\
 &= \frac{q^{n+1} \cdot (1 - q^h)}{q^{n+1}} \\
 &= 1 - q^h \\
 &= 1 - q \\
 &= p
 \end{aligned}$$

Όπου $F(x)$ είναι η συνάρτηση κατανομής πιθανότητας και η $F^c(x)$ η αντίστοιχη συνάρτηση επιβίωσης της γεωμετρικής κατανομής. Για την απόδειξη που ακολουθεί χρησιμοποιείται η συμπληρωματική πιθανότητα $q = 1 - p$. Επομένως ισχύουν οι παρακάτω εξισώσεις για τις συναρτήσεις που προαναφέρθηκαν :

$$\begin{aligned}
 F(x) &= \sum_{x=0}^n f(x) \\
 &= \sum_{x=0}^n (1 - q) \cdot q^x \\
 &= (1 - q) \cdot (q^0 + q^1 + q^2 + \dots + q^n) \\
 &= (1 - q) \cdot \frac{1 - q^{n+1}}{1 - q} \\
 &= 1 - q^{n+1}
 \end{aligned}$$

$$\begin{aligned}
 F(x + h) &= \sum_{x=0}^n f(x + h) \\
 &= 1 - q^{h+n+1}
 \end{aligned}$$

$$\begin{aligned} F^c(x) &= 1 - F(x) \\ &= q^{n+1} \end{aligned}$$

Επαναλαμβάνοντας εκ νέου τα ίδια αρχικά βήματα όπως και στη διαδικασία Poisson, προκύπτουν οι παρακάτω μαθηματικές σχέσεις :

$$\begin{aligned} p_0(t+h) &= B(N(t+h) = 0) \\ &= B(N(t) = 0, N(t, t+h) = 0) \\ &= B(N(t) = 0) \cdot B(N(t, t+h) = 0) \\ &= p_0(t) \cdot (1-p) \end{aligned}$$

Το ενδεχόμενο n αφίξεων εντός του χρονικού διαστήματος $(0, t+h]$ είναι ισοδύναμο με τα ανεξάρτητα ενδεχόμενα $\{n-k$ αφίξεις εντός του χρονικού διαστήματος $(0, t]\}$ και $\{k$ αφίξεις εντός του χρονικού διαστήματος $(t, t+h]\}$ για κάποιο k ($k=0, 1, 2, \dots, n$). Επομένως, για $n \geq 1$

$$\begin{aligned} p_n(t+h) &= B(N(t+h) = n) \\ &= B\left(\bigcup_{k=0}^n \{N(t) = n-k\} \cap \{N(t, t+h) = k\}\right) \\ &= \sum_{k=0}^n B(N(t) = n-k) \cdot B(N(t, t+h) = k) \\ &= p_n(t) \cdot B(N(t, t+h) = 0) + p_{n-1}(t) \cdot B(N(t, t+h) = 1) \\ &= p_n(t) \cdot (1-p) + p_{n-1}(t) \cdot p \end{aligned}$$

Άρα έχουμε το εξής ζεύγος μαθηματικών σχέσεων:

$$p_0(t+h) = B(N(t+h) = 0) = p_0(t) \cdot (1-p) \tag{6}$$

$$p_n(t+h) = B(N(t+h) = n) = p_n(t) \cdot (1-p) + p_{n-1}(t) \cdot p \quad (7)$$

Οι αρχικές συνθήκες είναι πάλι οι ίδιες :

$$p_0(0) = 1 \quad p_n(0) = 0 \quad n \in N$$

Χρησιμοποιώντας την ίδια μέθοδο επίλυσης που στηρίζεται στις πιθανογεννήτριες συναρτήσεις, πολλαπλασιάζοντας την (6) με z_0 και την (7) με z_n και αθροίζοντας κατά μέλη για όλα τα $n \in N_0$ καταλήγουμε στο παρακάτω μαθηματικό τύπο :

$$\begin{aligned} \sum_{n=0}^{\infty} p_n(t+h) \cdot z^n &= (1-p) \cdot \sum_{n=0}^{\infty} p_n(t) \cdot z^n + p \cdot z \cdot \sum_{n=0}^{\infty} p_{n-1}(t) \cdot z^n \\ P(t+h, z) &= (1-p) \cdot P(t, z) + p \cdot z \cdot P(t, z) \\ P(t+h, z) &= P(t, z) \cdot [(1-p) + z \cdot p] \end{aligned}$$

Κάνοντας χρήση των αρχικών συνθηκών και παρατηρώντας ότι η σχέση είναι αναδρομική ως προς h λαμβάνουμε τη τελική μορφή της πιθανογεννήτριας συνάρτησης.

$$\begin{aligned} P(t+1, z) &= P(t, z) \cdot [(1-p) + z \cdot p] \\ P(t+2, z) &= P(t+1, z) \cdot [(1-p) + z \cdot p] \\ &\dots \\ P(t, z) &= [(1-p) + z \cdot p]^n \end{aligned}$$

Όμως η συνάρτηση αυτή είναι η πιθανογεννήτρια συνάρτηση της διωνυμικής κατανομής και έτσι φτάνουμε στο συμπέρασμα ότι η τ.μ. $B(t)$ για κάθε $t > 0$ ακολουθεί τη διωνυμική

κατανομή με παραμέτρους (n,p) όπου n το πλήθος των δοκιμών και p η πιθανότητα το αποτέλεσμα της δοκιμής να είναι επιτυχία. Η σημαντικότητα αυτής της κατανομής έγκειται στη μελέτη συστημάτων αναμονής τα οποία δεν πρέπει να προσεγγίζονται από τη κατανομή Poisson. Λόγω του ότι ούτε η παράμετρος n , το πλήθος των δοκιμών αλλά ούτε και η παράμετρος p , η πιθανότητα το αποτέλεσμα της δοκιμής να είναι επιτυχία δεν πληρούν τα κριτήρια, δεν θα ήταν αποδεκτό η διωνυμική κατανομή να μετατραπεί σε Poisson. Έτσι συμπεραίνουμε ότι για συστήματα αναμονής για τα οποία το πλήθος των δοκιμών όπως π.χ. η άφιξη ενός πλοίου ή η απουσία του, είναι σχετικά μικρό και καθώς $n \rightarrow \infty$ η πιθανότητα $p \rightarrow 0$, δηλαδή παραμένει στάσιμη, συνίσταται η χρήση της διωνυμικής κατανομής παρά της κατανομής Poisson.

3 Μεθοδολογία Παραγωγής Τεχνητών Δεδομένων

Ο συνολικός χρόνος παραμονής του πλοίου στο λιμάνι, δηλαδή από τη στιγμή που διαπερνά τα όρια του και αγκυροβολεί μέχρι τη στιγμή που τα ξαναδιαπερνά για να αποχωρήσει αφού έχει εξυπηρετηθεί, μπορεί να τμηματοποιηθεί σε 3 στάδια και 4 σημεία. Το πρώτο στάδιο είναι το χρονικό διάστημα αναμονής στο αγκυροβόλι, το δεύτερο στάδιο είναι η εξυπηρέτηση στο λιμάνι και το τελευταίο στάδιο είναι η απομάκρυνση και η επιστροφή στη θέση του αγκυροβολιού. Τα 4 σημεία που πλαισιώνουν και οριοθετούν τα 3 αυτά χρονικά διαστήματα είναι η άφιξη στο αγκυροβόλι, η άφιξη στο σημείο εξυπηρέτησης στο λιμάνι, η ολοκλήρωση της εξυπηρέτησης και η επιστροφή στο αγκυροβόλι. Όσο αφορά τα 3 χρονικά διαστήματα, το τελευταίο μπορεί να θεωρηθεί αμελητέο σχετικά με τα υπόλοιπα δύο. Παρατίθεται παράδειγμα για τα πώς κατανέμεται ο χρόνος, στις περισσότερες των περιπτώσεων, για τα 3 στάδια. Είναι φανερό ότι ο δεξιός άξονας έχει αντιστραφεί για να



Figure 5: Κατανομή συνολικού χρόνου παραμονής

είναι πιο ευδιάκριτα τα σημεία της καμπύλης. Για την ανάλυση των συστημάτων αναμονής χρησιμοποιείται μόνο το δεύτερο χρονικό διάστημα, δηλαδή εκείνο της εξυπηρέτησης μόνο.

Για να απομονωθεί το χρονικό διάστημα εξυπηρέτησης στο λιμάνι χρησιμοποιήθηκε μια μέθοδος που η αποτελεσματικότητά της είναι τόσο καλή όσο και η ποιότητα των δεδομένων εισόδου που χρησιμοποιήθηκαν. Αυτή η ιδιότητα της μεθόδου φαίνεται από τα στάδια της διαδικασίας για τον υπολογισμό του τελικού αποτελέσματος. Παρατίθεται και παράδειγμα.

- 1 Εντοπισμός του λιμένα υπό εξέταση από την ιστοσελίδα <https://marinetraffic.com>.
- 2 Υπολογισμός του μέσου όρου α) Χρόνου αναμονής στο αγκυροβόλι β) Χρόνου εξυπηρέτησης στο λιμάνι, για το χρονικό διάστημα που προσφέρεται ελεύθερα από την Marinetrtraffic (τελευταίο δίμηνο με βάση εβδομαδιαία δεδομένα). Ο μέσος όρος υπολογίζεται για το διάμεσο που χρησιμοποιεί η Marinetrtraffic. Επομένως, από τους υπολογισμούς προκύπτει ο μέσος όρος των διαμέσων. Ο λόγος που η Marinetrtraffic χρησιμοποιεί τη διάμεσο αντί για το μέσο όρο είναι για να μην υπάρχει στρέβλωση των δεδομένων λόγω ακραίων τιμών . Λίγες ακραίες παρατηρήσεις σε ένα δείγμα μπορούν να προκαλέσουν αλλοίωση στο τελικό μέσο όρο και να μην είναι τόσο αντιπροσωπευτικός και κοντά στη πραγματικότητα. Από τη Marinetrtraffic προσφέρεται ελεύθερα μόνο η διάμεσος για όλους του τύπους πλοίων συγκεντρωτικά και όχι για μεμονωμένο τύπο και μέγεθος πλοίου π.χ. Panamax Bulk carrier, Suezmax Tanker κ.λπ.
- 3 Εκμεταλλευόμενοι δεδομένα από την ετήσια αναφορά του UNCTAD [6] εντοπίζουμε το μέσο όρο του συνολικού χρόνου παραμονής από δεδομένα πάλι της Marinetrtraffic έτους 2017. Αυτή τη φορά ο μέσος όρος αναφέρεται στον ετήσιο μέσο όρο των

εβδομαδιαίων διαμέσων για όλα τα λιμάνια της χώρας του λιμανιού υπό εξέταση αλλά για το συγκεκριμένο τύπο πλοίου (π.χ. LNG carrier).

- 4 Από τα δεδομένα του βήματος 2 υπολογίζουμε το ποσοστό του χρόνου που διανύει το πλοίο στην εξυπηρέτηση εντός του λιμένα. Θεωρώντας απλή αναλογία μεταξύ των δεδομένων του βήματος 2 και 4, δηλαδή του μέσου όρου των διαμέσων του τελευταίου διμήνου για όλους τους τύπους πλοίων και του ετήσιου μέσου όρου διαμέσων για όλα τα λιμάνια της χώρας του λιμένα υπό εξέταση αλλά μόνο για τον υπό εξέταση τύπο πλοίου αντίστοιχα, προκύπτει ο ετήσιος μέσος όρος των διαμέσων για το τύπο πλοίου υπό εξέταση αλλά για όλα τα λιμάνια της χώρας υπό εξέταση.
- 5 Στο τελικό στάδιο της διαδικασίας γίνεται η υπόθεση ότι οι χρόνοι εξυπηρέτησης εντός του λιμένα είναι κατανομημένοι ημιτονοειδώς στα λιμάνια της χώρας του λιμένα υπό εξέταση.
- 6 Το τελικό βήμα είναι ότι ο διάμεσος χρόνος εξυπηρέτησης εντός του λιμένα υπό εξέταση λαμβάνεται ως ο ελάχιστος δυνατός. Δηλαδή, η μικρότερη δυνατή διακριτή τιμή που λαμβάνει η ημιτονοειδής συνάρτηση στο διάστημα που ορίζεται :

$$f : \Omega \ni x \rightarrow f(x) \in Z$$

όπου

$$\Omega \equiv 5 \cdot \nu \text{ με } \nu = 1, 2, \dots, 18$$

και

$$Z \equiv 0.1 \cdot \sin x + 0.65$$

Επομένως, ο χρόνος εξυπηρέτησης εντός του λιμένα που αξιοποιείται είναι η τιμή $\min f(x)$. Η μετάβαση από το χρόνο εξυπηρέτησης στο συνολικό ετήσιο αριθμό αναχωρήσεων για το λιμάνι υπό εξέταση και το συγκεκριμένο τύπο πλοίου γίνεται λαμβάνοντας τον αντίστροφο αριθμό του χρόνου εξυπηρέτησης δηλαδή τον αριθμό πλοίων που αναχωρούν στη μονάδα του χρόνου, στη περίπτωση μας, την ημέρα, πολλαπλασιασμένο με τον αριθμό των ημερών ενός έτους.

Γενική παραδοχή για τη συγκεκριμένη μέθοδο καθώς και για τις υπόλοιπες που έπονται είναι ότι το έτος έχει 365 ημέρες, ενώ για τις αναλύσεις που αφορούν το εικονικό έτος 20/01/19 έως 21/01/20 υπάρχουν 367 ημέρες.

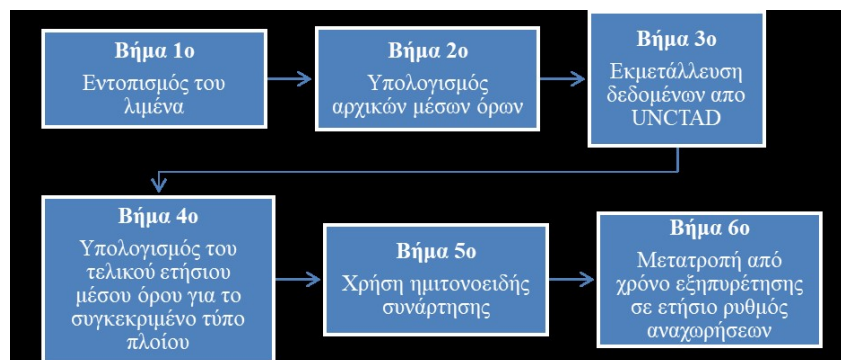


Figure 6: Διάγραμμα ροής μεθόδου

Παράδειγμα για το λιμάνι Incheon (LNG, LPG carriers) της Νότιας Κορέας

Βήμα 2ο

$$M_1 = \frac{0.3 + 0.3 + 0.4 + 0.5 + 0.5 + 0.3 + 0.3 + 0.5 + 0.5}{9} \simeq 0.4$$

$$M_2 = \frac{0.8 + 0.9 + 0.8 + 0.9 + 0.8 + 0.8 + 0.8 + 0.8 + 0.9}{9} \simeq 0.85$$

Όπου M_1 και M_2 ο μέσος χρόνος αναμονής στο αγκυροβόλι και ο μέσος χρόνος εξυπηρέτησης στο λιμάνι αντίστοιχα

Βήμα 3ο & 4ο

Μέσος συνολικός χρόνος παραμονής = 0.95 ημέρες

Ποσοστό χρόνου εξυπηρέτησης = $\frac{0.85}{0.85 + 0.4} = 0.68\%$

$M_5 = 0.68 \cdot 0.95 = 0.65$ ημέρες

Βήμα 5ο & 6ο

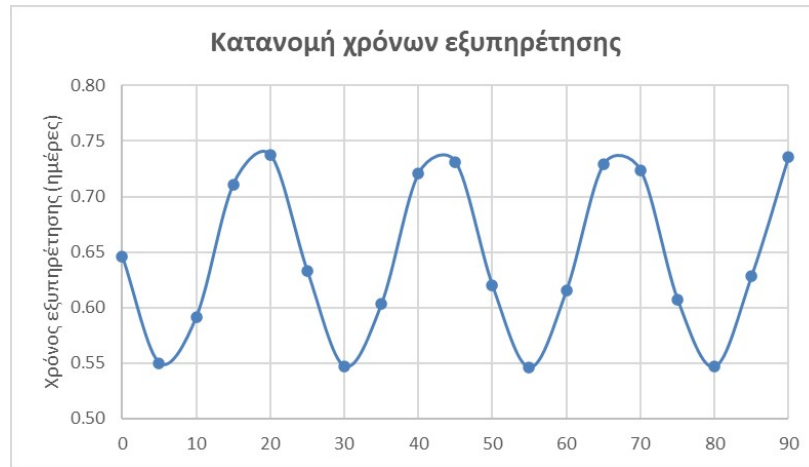


Figure 7: Ημιτονοειδής συνάρτηση

$$\min f(x) = 0.55 \text{ ημέρες}$$

$$\frac{1 \cdot 365}{0.55} = 668 \text{ Αφίξεις ετησίως}$$

Το δεύτερο στοιχείο που απαιτείται για τη τροφοδοσία του μοντέλου συστημάτων αναμονής είναι το χρονικό διάστημα μεταξύ δύο διαδοχικών αφίξεων πλοίων. Για αυτή τη περίπτωση χρησιμοποιήθηκε μια μέθοδος με λιγότερα βήματα ανά στάδιο, αλλά για να καταλήξουμε στο τελικό αποτέλεσμα έγινε σύγκριση μεταξύ δύο διαφορετικών τρόπων επίλυσης. Η προϋπόθεση η οποία στηρίζει τα αρχικά βήματα της μεθοδολογίας που εφαρμόστηκε είναι ότι μια από τις γνωστές κατανομές (π.χ. Gauss, Poisson) μπορεί να αποτυπώσει τον τρόπο που κατανέμονται οι ετήσιες αφίξεις ενός συγκεκριμένου τύπου πλοίου της χώρας του λιμένα υπό εξέταση στο λιμάνια της. Τα βήματα της μεθόδου που ακολουθήθηκε δεν ήταν πολυπληθής αλλά σύντομα. Για την καλύτερη κατανόηση της μεθόδου παρατίθεται και ένα παράδειγμα.

- 1 Εντοπισμός του λιμένα υπό εξέταση από την ιστοσελίδα <https://www.Marinetraffic.com>.
- 2 Εκμετάλλευση των δεδομένων από την ετήσια αναφορά του UNCTAD [6] και συγκεκριμένα του αριθμού των συνολικών ετήσιων αφίξεων για όλα τα λιμάνια και για ένα συγκεκριμένο τύπο πλοίου της χώρας του λιμένα υπό εξέταση.
- 3 Σύντομη ανάλυση των λιμένων της χώρας του λιμένα υπό εξέταση για την εξακρίβωση του ποσοστού των λιμένων που διαθέτουν τις απαραίτητες και κατάλληλες υποδομές για την υποδοχή του εκάστοτε τύπου πλοίου υπό εξέταση. Η κυριότερη πηγή πληροφόρησης ήταν η ιστοσελίδα <https://www.worldportsource.com> αλλά για μια εκτενέστερη μελέτη για βελτίωση της αποδοτικότητας της μεθόδου, στοιχεία από την ιστοσελίδα [urlhttps://www.findaport.com](https://www.findaport.com) θα ήταν περισσότερο ευνοϊκά.
- 4 Εκμετάλλευση των γνωστών κατανομών, Gauss και Poisson, με ελαφρύ τροποποίηση (ούτως ώστε να αποδίδουν αριθμό αφίξεων αντί για πιθανότητα) και σύγκριση αυτών σε δύο επίπεδα. Σε πρώτο επίπεδο σύγκριση της ίδιας κατανομής υπό διαφορετικές μορφές μεταβάλλοντας τις ελεύθερες παραμέτρους της. Πιο συγκεκριμένα για την κανονική κατανομή μεταβάλλεται η τυπική απόκλιση και για την κατανομή Poisson η παράμετρος λ . Κριτήριο επιλογής της πιο κατάλληλης κατανομής σε αυτό το επίπεδο είναι κατά πόσο συμφωνεί ποσοτικά με την ανάλυση που πραγματοποιήθηκε στο προηγούμενο βήμα. Σε δεύτερο επίπεδο γίνεται σύγκριση μεταξύ των διαφορετικών κατανομών και κριτήριο αποτελεί το ποσοστό συμφωνίας τόσο ποιοτικά όσο και ποσοτικά με τα προαναφερθέντα αποτελέσματα της ανάλυσης.
- 5 Στο τελικό βήμα επιλέγεται εκείνο το σημείο της κατανομής που αποτυπώνει όσο το δυνατόν πιο πιστά το βαθμό δραστηριοποίησης ($PX \times M \times PY$) του λιμένα υπό

εξέταση. Επομένως, έτσι προκύπτει ο ετήσιος αριθμός αφίξεων για το λιμάνι υπό εξέταση και το συγκεκριμένο τύπο πλοίου.

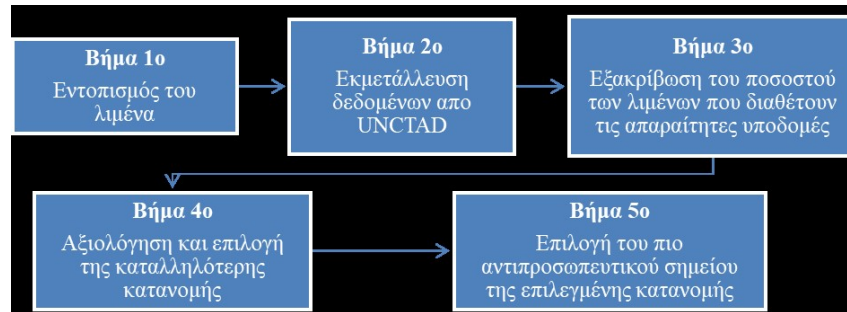


Figure 8: Διάγραμμα ροής της μεθόδου

Παράδειγμα για το λιμάνι Incheon (LNG LPG carriers) της Νότιας Κορέας

Βήμα 2ο

2827 αφίξεις για το έτος 2017.

Βήμα 3ο

1-5 από τα 17 λιμάνια διαθέτουν τερματικούς σταθμούς για φορτοεκφόρτωση LNG LPG.

Βήμα 4ο & 5ο

Η κατανομή Gauss με παραμέτρους $\mu=9$ και $\sigma=1$ είναι εκείνη που αποτυπώνει καλύτερα τη πραγματική κατάσταση και το σημείο (9,1128) είναι αυτό που επιλέγεται.

Από τη στιγμή που οι ετήσιες αφίξεις και αναχωρήσεις έχουν προσδιοριστεί υπολείπεται να εισαχθεί ως δεδομένο ο αριθμός των σημείων εξυπηρέτησης. Αυτός εντοπίζεται στην

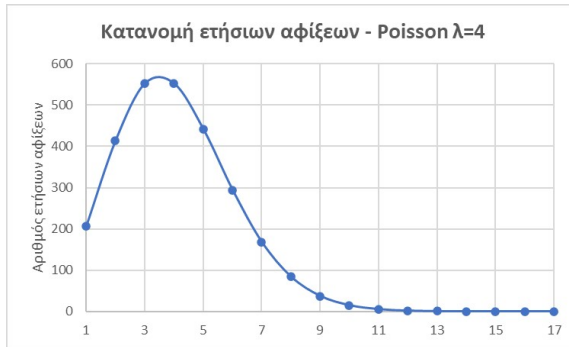


Figure 9: Poisson no 1



Figure 10: Poisson no 2

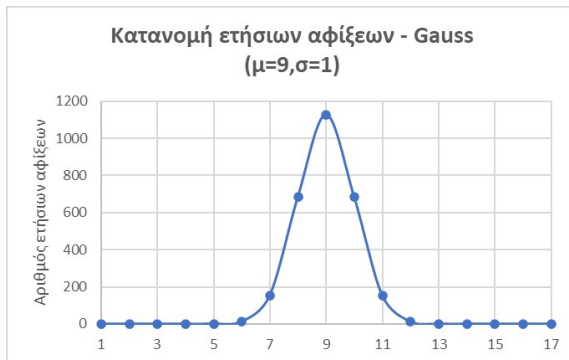


Figure 11: Gauss no 1

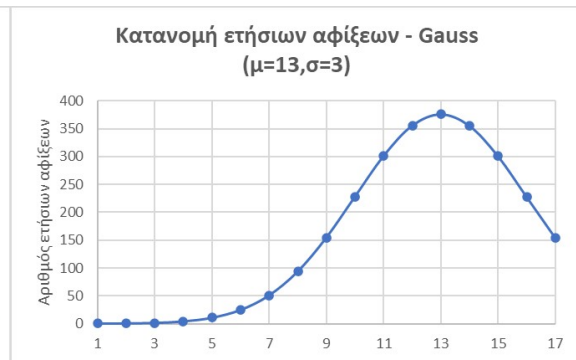


Figure 12: Gauss no 2

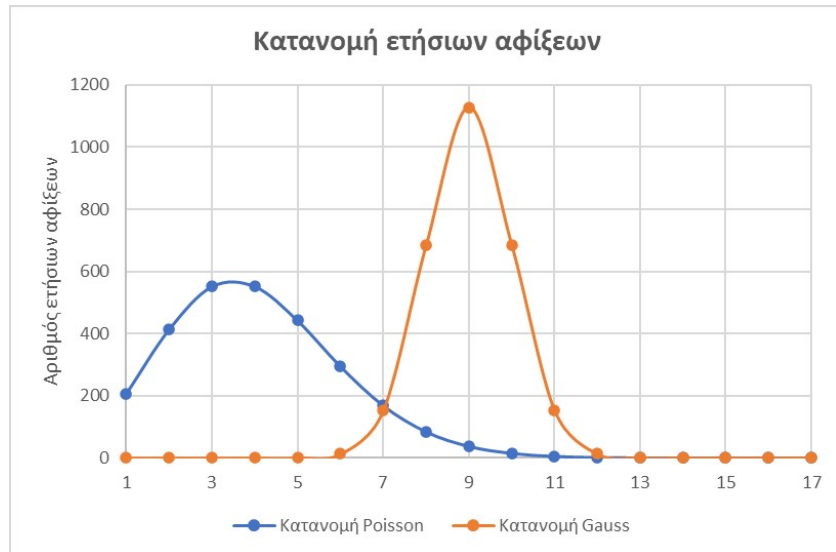


Figure 13: Σύγκριση διαφορετικού τύπου κατανομών

ιστοσελίδα <https://www.worldportsource.com> αφού πρώτα πραγματοποιηθεί μια σάρωση για το πόσα λιμάνια της χώρας του λιμένος υπό εξέταση διαθέτουν τις αναγκαίες υποδομές καθώς και πόσες από αυτές είναι κατάλληλες για την εξυπηρέτηση του τύπου πλοίου υπό εξέταση.

Από το σημείο αυτό, μόνο με μερικές μετατροπές στα δεδομένα προς είσοδο που έχουμε καταλήξει θα μπορούσε να ξεκινήσει η ανάλυση των συστημάτων αναμονής, μέσω των κατάλληλων μοντέλων, που προκύπτουν από τις κινήσεις των διαφορετικών τύπων πλοίων στους διάφορους λιμένες ανά τον κόσμο.

Παρ' όλα αυτά, στη παρούσα εργασία εφόσον τα πρωτογενή δεδομένα δεν είναι ελεύθερα

προσβάσιμα, γίνεται μια προσπάθεια να παραχθούν τα απαραίτητα δεδομένα έτσι ώστε η ανάλυση να δύναται να καλύψει όλες τις πτυχές των συστημάτων αναμονής που μπορούν να διερευνηθούν.

Ο σκελετός της μεθόδου που χρησιμοποιήθηκε για την παραγωγή των απαιτούμενων δεδομένων διαφέρει συγκριτικά με τον αντίστοιχο των άλλων δύο μεθόδων στο ότι αξιοποιεί γεννήτριες ψευδοτυχαίων αριθμών για την παραγωγή των απαραίτητων δεδομένων εισόδου για τη μέθοδο. Ακολουθούν τα αναλυτικά βήματα της μεθόδου μαζί με ένα παράδειγμα.

- 1 Η πρώτη ενέργεια είναι η επιλογή του ελάχιστου και του μέγιστου αριθμού πλοίων που μπορούν να παρουσιαστούν προς άφιξη ή αναχώρηση για το λιμάνι προς εξέταση εντός μιας (1) ημέρας. Αυτές οι δύο παράμετροι, ονομαστικά ο ελάχιστος και ο μέγιστος αριθμός πλοίων εξαρτώνται από ποικίλες μεταβλητές. Μερικές από αυτές είναι η χώρα που συνδέεται με τη νομοθεσία και το καθεστώς που επικρατεί, η γεωγραφική θέση που σχετίζεται με τον καιρό και οι υποδομές που σχετίζονται με τον τύπο και το μέγεθος πλοίου που δύνανται να εξυπηρετήσουν. Στη παρούσα όμως μέθοδο θα στρέψουμε το ενδιαφέρον και θα θεωρήσουμε ότι είναι συναρτήσεως μόνο του βαθμού δραστηριοποίησης ($PX \times M \times PY$) του λιμένα υπό εξέταση. Ως δείκτης για το βαθμό δραστηριοποίησης χρησιμοποιείται ο μεγαλύτερος από τους αριθμούς ετήσιος αριθμός αφίξεων και ετήσιος αριθμός αναχωρήσεων. Σε αυτό το βήμα γίνεται η παραδοχή ότι ο μέγιστος και ελάχιστος αριθμός πλοίων είναι κοινός για τις αφίξεις και τις αναχωρήσεις.

- 2 Για κάθε ένα από τον αριθμό πλοίων προς άφιξη ή αναχώρηση (από εδώ και στο

εξής bin) δημιουργούμε με τη χρήση γεννήτριας ψευδοτυχαίων αριθμών (συνάρτηση 'randbetween()' του excel) έναν αριθμό στο κλειστό διάστημα [20,50] για κάθε bin και που ουσιαστικά αποτελεί τη συχνότητα, σε πρώτη φάση, του κάθε bin. Για να υπάρχει ακρίβεια και οι συχνότητες να συμφωνούν τόσο με το συνολικό ετήσιο αριθμό αφίξεων ή αναχωρήσεων όσο και με τις ημέρες ενός έτους (367) θα πρέπει

- 1) Το άθροισμα των επιμέρους γινομένων των bins με την αντίστοιχη συχνότητα να ισούται με το συνολικό αριθμό αφίξεων ή αναχωρήσεων καθώς και
- 2) Το άθροισμα των συχνοτήτων να ισούται με τις ημέρες ενός έτους (367). Αυτό υλοποιήθηκε σε δύο επιμέρους βήματα.

Πρώτον, δημιουργήθηκαν οι ενδιάμεσες συχνότητες χρησιμοποιώντας το μαθηματικό

$$\text{τύπο } f_{intermediate} = \frac{f_{initial}}{\sum_{bin} f_{initial} \cdot bin} \cdot \text{annual arrivals.}$$

Δεύτερον, αξιοποιήθηκε το add in solver του excel (ένα πρόγραμμα βελτιστοποίησης)

έτσι ώστε ταυτόχρονα να εξασφαλιστούν οι ισότητες των αθροισμάτων που

προαναφέρθηκαν 1 και 2. Ως σκοπός της βελτιστοποίησης ήταν το άθροισμα

2 και ως περιορισμός ήταν το άθροισμα 1, που ήδη είχε επιτευχθεί από το

ανώτερο βήμα. Η μέθοδος επίλυσης που χρησιμοποιήθηκε ήταν η 'GRG Non-

linear' καθώς κατένεμε το βάρος κάθε μεταβλητής (των ενδιάμεσων συχνοτήτων)

με πιο ισομερή τρόπο συγκριτικά με τη μέθοδο επίλυσης 'Simplex LP' και ήταν

πιο αποδοτική από άποψη χρόνου σε σχέση με τη μέθοδο επίλυσης 'Evo-

lutionary'. Έτσι ολοκληρώνεται αυτό το βήμα και καταλήγουμε στις τελικές

συχνότητες ανά bin.

- 3 Εφόσον ολοκληρωθούν τα προηγούμενα βήματα καταλήγουμε στο επιθυμητό αποτέλεσμα που είναι ένα ιστόγραμμα αφίξεων και ένα αναχωρήσεων. Για να συνεχιστεί η

μέθοδος πρέπει να κατανεμηθούν με τυχαίο τρόπο οι συχνότητες κάθε bin στις ημέρες του έτους. Ο τρόπος που επιτεύχθηκε αυτό ήταν με τη χρήση δύο συναρτήσεων του excel. Η πρώτη 'rand()' παρήγαγε ένα τυχαίο και μοναδικό δεκαπενταψήφιο δεκαδικό αριθμό για κάθε ημέρα του έτους (από 20/01/19 έως 21/01/20, συνολικά 367 ημέρες). Η ιστοσελίδα <https://www.trumpexcel.com> υποστηρίζει ότι οι αριθμοί τείνουν να μην επαναλαμβάνονται ειδικά αν το παραγόμενο πλήθος τυχαίων αριθμών δεν υπερβαίνει ένα μεγάλο αριθμό π.χ. 367. Αμέσως μετά τη χρήση της συνάρτησης 'rand()', χρησιμοποιείται η συνάρτηση 'Rank.eq()' η οποία βρίσκει τη μοναδική (λόγω του ότι ακόμα και αν υπάρξουν διπλότυπα, δηλαδή ο ίδιος δεκαδικός δεκαπενταψήφιος αριθμός επαναληφθεί μετά τη χρήση της συνάρτησης 'rand()', ο αριθμός που βρίσκεται στο ανώτερο κελί του φύλλου στο excel επιλέγεται ως μεγαλύτερος) θέση κατάταξης για κάθε μια από τις ημέρες του έτους. Έτσι σε συνδυασμό αυτών των δύο συναρτήσεων προκύπτει ένας τυχαίος και μοναδικός αριθμός για κάθε ημέρα του έτους. Αυτό επιβεβαιώθηκε συγκρίνοντας το άθροισμα των μοναδικών αριθμών κατάταξης με το άθροισμα των αριθμών από 1 έως 367 με τη χρήση του μαθηματικού τύπου $\nu \cdot \frac{\nu + 1}{2}$.

- 4 Στο τελικό βήμα αντιστοιχίζεται, με μια απλή και σύντομη διαδικασία όπως φαίνεται στο παράδειγμα, σε κάθε ημέρα το πλήθος των αφιχθέντων ή αναχωρηθέντων πλοίων. Επομένως, μπορούμε να έχουμε μια εικόνα για τον τρόπο εξέλιξης των αφίξεων και αναχωρήσεων για οποιοδήποτε χρονικό διάστημα εντός του έτους και να υπολογιστούν αναλυτικά, όπως ακριβώς σε μια στατιστική ανάλυση ενός δείγματος από ένα λιμάνι, οι μετρικές όπως ο ρυθμός αφιχθέντων και αναχωρηθέντων πλοίων (λ, μ), η ένταση συνωστισμού (ρ) καθώς και άλλοι στατιστικοί όροι όπως η διακύμανση.

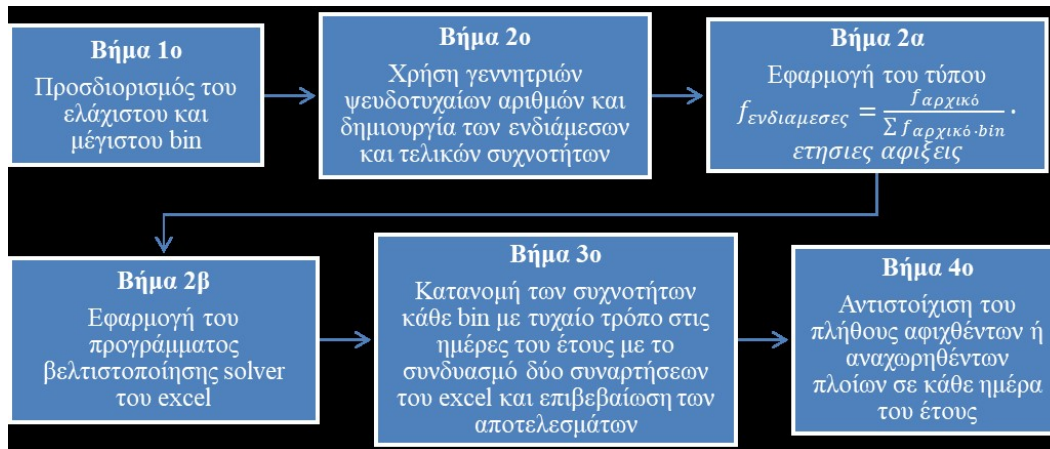


Figure 14: Διάγραμμα ροής της μεθόδου

Είναι σημαντικό προτού προχωρήσουμε στο παράδειγμα να τονίσουμε και να δώσουμε έμφαση ξανά σε αυτό το σημείο για το λόγο που χρησιμοποιήθηκαν γεννήτριες ψευδοτυχαίων αριθμών και όχι κάποια από τις γνωστές κατανομές π.χ. Gauss, Poisson κ.λπ. Η χρήση γνωστών κατανομών για την παραγωγή των απαραίτητων δεδομένων θα μείωνε την αυθεντικότητά τους, με αποτέλεσμα να αποκλίνουν από τη πραγματικότητα σε σημαντικό βαθμό και να μην αποδίδουν σε επαρκή βαθμό την αβεβαιότητα που τα διέπει.

Παράδειγμα για το λιμάνι Incheon (LNG LPG carriers) της Νότιας Κορέας

Βήμα 1α

Το λιμάνι Incheon έχει 1128 αφίξεις και 668 αναχωρήσεις άρα ο δείκτης δραστηριοποίησής του είναι 1128 και σύμφωνα με τις βαθμίδες Πολύ χαμηλή – Χαμηλή – Μεσαία –Υψηλή - Πολύ Υψηλή εντάσσεται στη κατηγορία μεσαία και άρα τα bins είναι ακέραιοι αριθμοί στο κλειστό διάστημα [0,7].

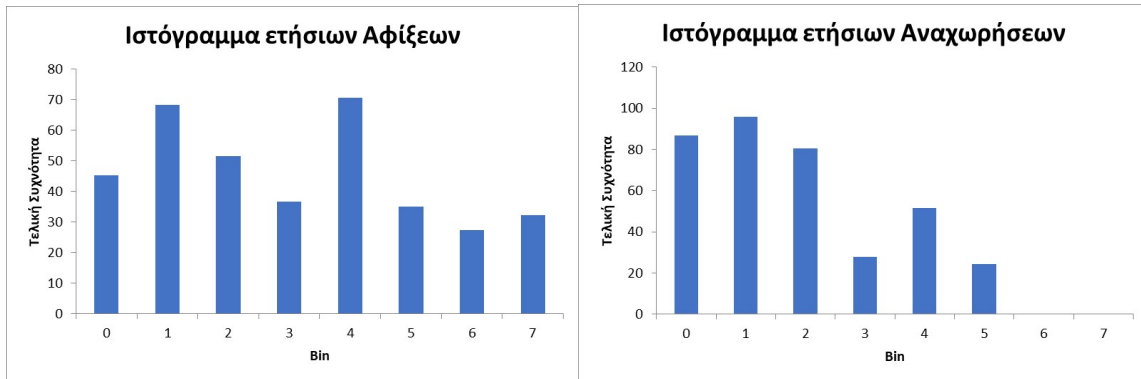


Figure 15: Ιστόγραμμα Αφίξεων

Figure 16: Ιστόγραμμα Αναχωρήσεων

Βήμα 2ο

Δημιουργία των αρχικών συχνοτήτων με τη χρήση της συνάρτησης `randbetween()`, εφαρμογή του μαθηματικού τύπου $f_{intermediate} = \frac{f_{initial}}{\sum_{bins} f_{initial} \cdot bin} \cdot \text{annual arrivals}$ για να προκύψουν οι ενδιάμεσες συχνότητες και χρήση του προγράμματος βελτιστοποίησης `solver` για την εύρεση των τελικών συχνοτήτων. Η μαθηματική έκφραση του προβλήματος για το `solver` είχε τη μορφή :

$$\text{Set objective function equal to} = \sum_{bins} f_{intermediate} = 367$$

$$\text{Subject to the constraints} = \sum_{bins} bin \cdot f_{intermediate} = 1128$$

$$\text{variables} = f_{intermediate}$$

Βήμα 3ο & 4ο

Πρώτα συμπληρώνεται η πρώτη στήλη με τη χρήση της συνάρτησης `'rand()'` και εν συνεχεία

Bin	$f_{initial}$	$f_{intermediate}$	f_{final}
0	31	33	45
1	39	42	68
2	44	47	52
3	50	53	37
4	47	50	71
5	30	32	35
6	26	28	27
7	41	44	32
$\sum_{bins} f_i$	308	328	367
$\sum_{bins} bin \cdot f_i$	1058	1128	1128

Table 2: Υπολογισμός ετήσιων αφίξεων

Τυχαίος δεκαδικός δεκαπενταψήφιος αριθμός	Μοναδική θέση κατάταξης	Bin	Πλήθος αφιχθέντων πλοίων	Ημέρα του έτους
0.720279...	104	1	0	20/01/19
0.74079...	94	1	0	21/01/19
0.380648...	218	4	0	22/01/19
0.538116...	168	3	0	23/01/19
0.15002...	311	6	0	24/01/19
0.350776...	228	4	0	25/01/19

Table 3: Κατανομή συχνοτήτων αφίξεων κάθε bin στις ημέρες του έτους

η δεύτερη στήλη με τη συνάρτηση 'rank.eq()' που έχει ως είσοδο τα στοιχεία της πρώτης στήλης. Ύστερα συμπληρώνεται η τέταρτη στήλη όπου παρατίθεται το πλήθος των αφιχθέντων πλοίων σύμφωνα με το ιστόγραμμα σε αύξουσα σειρά και τελικά η τρίτη στήλη στην οποία εκμεταλλεύεται η συνάρτηση του excel 'indirect' που χρησιμοποιεί ως είσοδο τη δεύτερη και τέταρτη στήλη και κατανέμει τυχαία σε bins το πλήθος των αφιχθέντων πλοίων στις ημέρες του έτους.

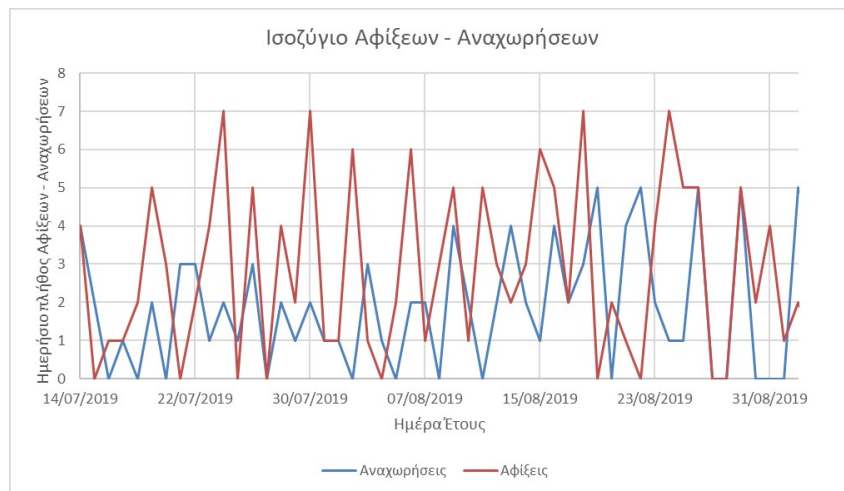


Figure 17: Ισοζύγιο Αφίξεων - Αναχωρήσεων

Δείκτης Δραστηριοποίησης	Ελάχιστο bin	Μέγιστο bin
Πολύ Χαμηλός	-	-
Χαμηλός	-	-
Μεσαίος	0	7
Υψηλός	-	-
Πολύ Υψηλός	-	-

Table 4: Αντιστοίχιση ελάχιστου και μέγιστου bin σε κάθε βαθμό δραστηριοποίησης

4 Προσαρμογή G.I. κατανομών σε Ιστογράμματα

Πέρα από τη βασική κατανομή Poisson (M) ή τους σταθερούς χρόνους εξυπηρέτησης (ή αλλιώς Deterministic - D) ή ακόμα και την συνάρτηση πυκνότητας πιθανότητας της κατανομή Erlang $\frac{(\lambda \cdot k)^k}{\Gamma(k)} \cdot u^{k-1} \cdot e^{-\lambda \cdot k \cdot u}$ [7] που χρησιμοποιούνται ευρέως στις εφαρμογές των μοντέλων αναμονής, στη βιβλιογραφία παρατηρείται ότι, από τη δεκαετία του '50 έως και την αρχή της νέας χιλιετίας γινόταν μια προσπάθεια για τη δημιουργία μαθηματικών μοντέλων που αποτυπώνουν και ενσωματώνουν την επίδραση μιας γενικής κατανομής (G.I.) που τηρεί τα κριτήρια των στοχαστικά ανεξάρτητων και ισόνομων τυχαίων μεταβλητών αλλά δεν φέρει την αμνήμονη ιδιότητα.

Η πιο γνωστή προσπάθεια που ευκόλως συναντά κανείς στη βιβλιογραφία είναι η μαθηματική σχέση του Khintchine - Pollaczek formula G.I./M/1 που αφορά συστήματα αναμονής με Poissonian είσοδο για τους ενδιάμεσους χρόνους αφίξεων, μια γενική κατανομή για τους χρόνους εξυπηρέτησης και μοναδικό εξυπηρετητή. Οι περισσότερες προσπάθειες χρησιμοποιούν ως σημείο αναφοράς τα αποτελέσματα από την έρευνα του A.K. Erlang [8] όπως το γνωστό ως Erlang C μαθηματικό τύπο ή Erlang delay formula [9]. Μια περίληψη ποικίλων μεθόδων στη μορφή προσεγγίσεων εντοπίστηκαν στο [10]. Παρ' όλα αυτά, ορισμένα άρθρα [11], [12] τείνουν να απομακρύνονται από τις προσεγγίσεις και να προσφέρουν τη δυνατότητα για μια πιο τυποποιημένη μορφή των μοντέλων αναμονής με γενικευμένη είσοδο όχι μόνο για τη μέση τιμή π.χ. του μήκους ουράς, του χρόνου αναμονής ή του πλήθους των απασχολούμενων εξυπηρετητών αλλά και για τη κατανομή του μήκους ουράς, τη κατανομή του χρόνου αναμονής καθώς και τη κατανομή του πλήθους των απασχολούμενων εξυπηρετητών.

Όλες όμως οι συμβατικές κατανομές που συναντώνται αδυνατούν να προσαρμοστούν στις υψηλής διακύμανσης δεδομένα. Ο καλύτερος τρόπος για να γίνει φανερό αυτό είναι μέσω ενός παραδείγματος όπως απεικονίζεται παρακάτω.

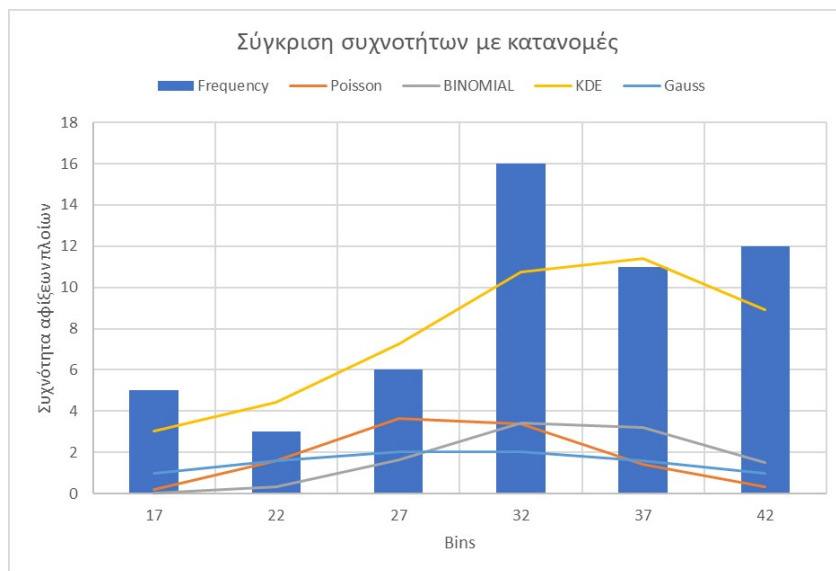


Figure 18: Σύγκριση συχνοτήτων με κατανομές

Είναι εμφανές ότι ούτε η Poisson, ούτε η Διωνυμική αλλά ούτε και η κανονική κατανομή προσαρμόζονται σε ικανοποιητικό βαθμό στις έντονες αυξομειώσεις των συχνοτήτων. Το ίδιο φαίνεται να ισχύει και να αποδεικνύεται από το χ^2 test [2] που παρατίθεται παρακάτω :

Το χ^2 test αποτελεί ένα τρόπο αξιολόγησης για το αν είναι επαρκής ο βαθμός στον οποίο προσαρμόζεται η κατανομή π.χ. Poisson ή Gauss κ.λπ. στις πραγματικές συχνότητες. Η διαδικασία που ακολουθείται για την πρακτική εφαρμογή του test είναι η εξής :

Αρχικά προσδιορίζεται η παράμετρος α που αποτελεί ένα μέτρο του κατά πόσο αυστηρό θα είναι το test. Παράλληλα, υπολογίζεται και ο βαθμός ελευθερίας της κατανομής μας.

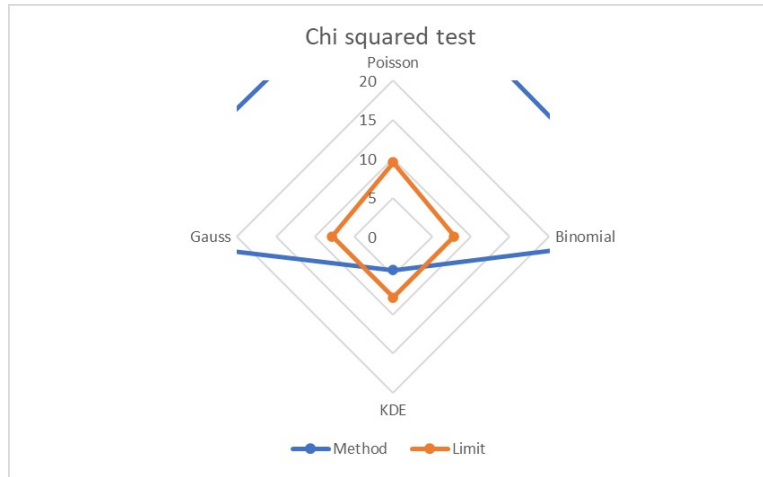


Figure 19: χ^2 Test

Αυτός προκύπτει αφαιρώντας από το πλήθος των “bins” τον αριθμό των ελεύθερων παραμέτρων της χρησιμοποιηθείσας κατανομής και τη μονάδα όπως φαίνεται παρακάτω :

$$DF = \text{number of bins} - \text{free parameters of pdf} - 1$$

Τέλος από πινακοποιημένη μορφή για διαφορετικές τιμές της παραμέτρου α και του βαθμού ελευθερίας DF επιλέγεται το όριο το οποίο δεν πρέπει να υπερβεί το χ^2 test ούτως ώστε να θεωρηθεί έγκυρη η “null hypothesis”.

DF	Probability less than the critical value				
	0.9	0.95	0.975	0.99	0.999
1	2.706	3.841	5.024	6.635	10.828

Table 5: Πινακοποιημένη μορφή για εύρεση του ορίου του χ^2 test

Η τιμή που συγκρίνεται με το όριο προκύπτει χρησιμοποιώντας το παρακάτω μαθηματικό

τύπο :

$$\sum_i \frac{(f_i - F_i)^2}{f_i}$$

Όπου f_i οι πραγματικές συχνότητες και F_i οι συχνότητες της εκάστοτε κατανομής.

Η υπόθεση που γίνεται, «null hypothesis» είναι ότι η κατανομή Poisson ή Gauss ή οποιαδήποτε άλλη χρησιμοποιηθεί έχει καλή προσαρμογή στα δεδομένα συχνοτήτων που έχουν συλλεχθεί.

Το παραπάνω διάγραμμα 19 αποτελεί ένα εύκολο τρόπο για να διαπιστώσεις άμεσα αν η τιμή που προέκυψε υπερβαίνει το όριο του χ^2 test ή όχι.

Μια λύση που βελτιώνει την αδυναμία των συμβατικών κατανομών αποτελούν οι KDEs (Kernel Density Estimators).

Ο «kernel» μπορεί να λάβει πολλές μορφές και αποτελεί το πυρήνα ή αλλιώς το δομικό στοιχείο που αξιοποιούν οι «Density Estimators» για να προσαρμόσουν τη συνάρτηση κατανομής που δημιουργούν σε ένα τυχαίο αλλά ανεξάρτητο και ισόνομα κατανεμημένο δείγμα συχνοτήτων. Οι μορφές αυτές δεν είναι τίποτα άλλο από τις σύνηθες κατανομές όπως η ομοιόμορφη κατανομή, η κατανομή Gauss, η κατανομή Epanechnikov κ.λπ [13].

Η μαθηματική σχέση που εκφράζει έναν «kernel» και η αντίστοιχη σχέση για τον KDE του, φαίνονται παρακάτω :

$$kernel = K\left(\frac{x - \text{bin value}}{h}\right)$$

$$f(x) = \frac{1}{n \cdot h} \cdot \sum_{\text{bin values}} K\left(\frac{x - \text{bin value}}{h}\right)$$

[14]. Στη περίπτωση που ο «kernel» φέρει τη κατανομή Gauss οι παραπάνω σχέσεις

γράφονται ως εξής :

$$K\left(\frac{x - \mu}{h}\right) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-0.5 \cdot \frac{(x - \text{bin value})^2}{2 \cdot \sigma^2 \cdot h^2}}$$

$$f(x) = \frac{1}{n \cdot h} \cdot \sum_{\text{bin values}} K\left(\frac{x - \text{bin value}}{h}\right)$$

Συνθέτοντας πολλούς πυρήνες σε έναν ενιαίο, δηλαδή προσθέτοντας πολλά δομικά στοιχεία που έχουν προσαρμοστεί σε ένα τυχαίο δείγμα συχνοτήτων όπως απεικονίζεται παρακάτω (ο «kernel» είναι η κατανομή Gauss με παραμέτρους $(\mu, \sigma) = (\text{bin value}, 3.4)$), προκύπτει μια τεχνητή κατανομή που προσεγγίζει με μεγάλη ακρίβεια το τυχαίο αυτό δείγμα.

Ιστόγραμμα & Gaussian KDE

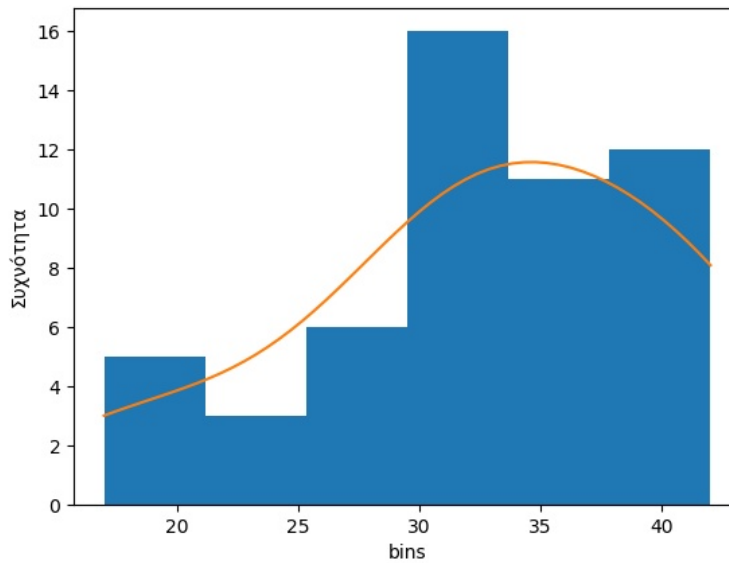


Figure 20: Ιστόγραμμα και Gaussian KDE

Η διαδικασία που ακολουθείται για να προσαρμοστεί κάθε ένας «kernel» καθώς και ο KDE είναι η εξής:

- A Δημιουργείται το ιστόγραμμα του τυχαίο δείγματος στο οποίο υπάρχει η ανάγκη να προσαρμοστεί μια τεχνητή κατανομή.
- B Δημιουργούνται «bins» για ένα εύρος τιμών, π.χ. το πρώτο «bin» του παραπάνω παραδείγματος εντάσσεται στο εύρος [15-19] με διάμεση τιμή το 17 και στο οποίο ανήκουν όλες οι παρατηρήσεις που ανήκουν στο διάστημα [15-19]. Στο συγκεκριμένο δείγμα εμφανίστηκαν 5 φορές αφίξεις πλοίων στο διάστημα [15-19] και πιο συγκεκριμένα σε αύξουσα σειρά ήταν 16,17,18,18,19. Ουσιαστικά το παραπάνω παράδειγμα αποτελεί ένα ιστόγραμμα συχνοτήτων.
- C Για κάθε μια από τις τιμές που εντάσσονται σε κάθε «bin» δημιουργείται και προσαρμόζεται ένας «kernel» με παραμέτρους (μ,σ). Το μ τίθεται για κάθε «kernel» ίσο με τη τιμή της παρατήρησης και πιο συγκεκριμένα για το πρώτο «bin» ίσο με 17. Οι συχνότητες του διαγράμματος έχουν υποστεί μερική τροποποίηση ώστε να είναι περισσότερο ευδιάκριτο το πώς γίνεται η μετάβαση (αθροίζοντας τους «kernels») από τους «kernels» στο τελικό KDE. Για λόγους απλοποίησης των υπολογισμών στην υλοποίηση των KDEs αντί των πραγματικών παρατηρήσεων χρησιμοποιήθηκε η διάμεσος. Δηλαδή αντί των τιμών 16,17,18,18,19 έγινε χρήση των τιμών 17,17,17,17,17 που είναι η διάμεσος του πρώτου «bin». Το ίδιο εφαρμόστηκε και για τα υπόλοιπα «bins». Δεν είναι ευδιάκριτο το γεγονός ότι σε κάθε «bin» υπάρχουν τόσοι «kernels» όσο και το πλήθος των τιμών του εκάστοτε «bin», επειδή στη συγκεκριμένη μέθοδο που ακολουθήθηκε συμπίπτουν, λόγω του ότι η τιμές κάθε «bin» είναι ίδιες.
- D Αθροίζοντας για κάθε «bin» τους «kernels» προκύπτει η τεχνητή κατανομή όπως

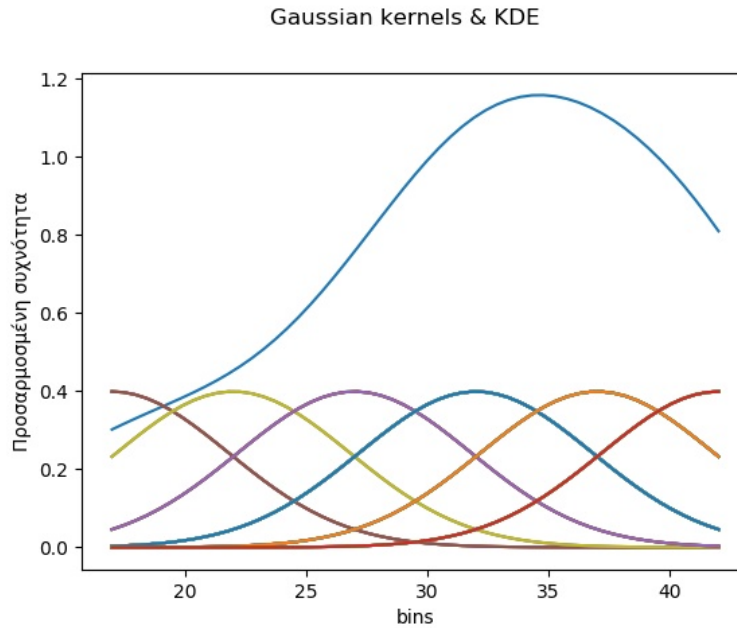


Figure 21: Gaussian kernels and KDE

φαίνεται παρακάτω.

Ε Ένα σημαντικό στοιχείο που επίσης είναι κοινό, όπως και οι παράμετροι της εκάστοτε κατανομής «kernel» που χρησιμοποιείται, για τους «kernels» είναι η παράμετρος του εύρους ζώνης h [15]. Όσο πιο μικρή η τιμή της παραμέτρου h τόσο πιο έντονα προσαρμόζεται ο KDE στο τυχαίο δείγμα.

Γίνεται εμφανές από το παραπάνω παράδειγμα ότι η προσαρμογή στις συχνότητες είναι πολύ καλύτερη συγκριτικά με τη κατανομή Poisson, Διωνυμική και Gauss. Αυτό το γεγονός επιβεβαιώνεται και με το χ^2 test όπου το αποτέλεσμα είναι για πρώτη φορά εντός των ορίων. Συμπερασματικά, τα υψηλής διακύμανσης δεδομένα προσεγγίζονται καλύτερα με τη χρήση KDE λόγω της ικανότητάς τους να αιχμαλωτίζουν τις απότομες αλλαγές των

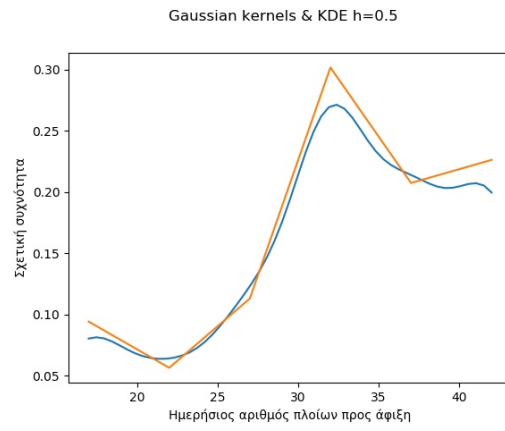
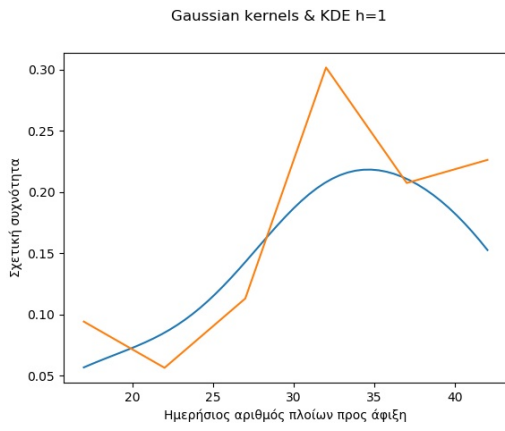


Figure 22: Gaussian kernels and KDE h=1 Figure 23: Gaussian kernels and KDE h=0.5

συχνοτήτων από “bin” σε “bin”.

5 Εφαρμογή των M/M/m G.I/M/m & M/G/m μοντέλων

5.1 Εφαρμογή του M/M/m μοντέλου αναμονής

Η εταιρεία Marinetraffic προσφέρει ποικίλα δεδομένα ελεύθερα, αλλά όχι αρκετά για συγκεκριμένο τύπο πλοίου και λιμάνι. Όμως, αποκτώντας πρόσβαση σε περισσότερα δεδομένα μέσω μιας δωρεάν συνδρομής σύντομου χρονικού διαστήματος μπορεί κανείς να εντοπίσει όλες τις απαραίτητες πληροφορίες και εισόδους που απαιτεί ένα μοντέλο αναμονής. Το λιμάνι το οποίο επιλέχθηκε για την προσομοίωση ήταν το Yokkaichi της Ιαπωνίας και ο τύπος του πλοίου LNG carrier μεγέθους 138000 – 150000 m^3 [16]. Η μηχανή αναζήτησης προσέφερε εύκολη πρόσβαση για οποιοδήποτε λιμάνι και τύπο-μέγεθος πλοίου σε δεδομένα, όπως η διάμεσος τιμή του χρόνου παραμονής στο αγκυροβόλι, η διάμεσος τιμή του χρόνου παραμονής στο λιμάνι και ο αριθμός των αφιχθέντων πλοίων. Η βάση στην οποία παρουσιάζονταν τα δεδομένα ήταν εβδομαδιαία, δηλαδή η διάμεσος τιμή του χρόνου παραμονής στο αγκυροβόλι αφορούσε την εβδομαδιαία διάμεσο τιμή για όλα τα πλοία που αγκυροβόλησαν εντός της εβδομάδας. Ταξινομώντας τη πληροφορία με βάση την εβδομάδα του έτους προέκυψε ο παρακάτω πίνακας :

Year	Week	Commercial Market	Commercial Size	Median at Anchorage	Median at Port	Weekly Arrivals
2020	1	LPG/LNG Carrier	SGC	0.2	0.2	18
2020	2	LPG/LNG Carrier	SGC	0.5	0.2	27

Table 6: Πίνακας δεδομένων από τη MarineTraffic

Συγκριτικά με τα τεχνητά δεδομένα της ενότητας 3 η εξαγωγή της πληροφορίας είναι πολύ πιο εύκολη. Ο μέσος ημερήσιος ρυθμός αφίξεων πλοίων, λ, προκύπτει προσθέτοντας τις τιμές της τελευταίας στήλης, weekly arrivals, λαμβάνοντας έτσι τις ετήσιες αφίξεις και

διαιρώντας με τις ημέρες ενός έτους 365. Όσο αφορά το μέσο ρυθμό εξυπηρέτησης, μ , προκύπτει λαμβάνοντας τη μέση τιμή της στήλης median at port. Για τη παραγωγή του ιστογράμματος αφίξεων ταξινομήθηκαν τα στοιχεία της τελευταίας στήλης με αύξουσα σειρά και τοποθετήθηκαν σε «bins» ανάλογα με τη τιμή τους. Το πλήθος των «bins» ήταν 6. Για το ιστόγραμμα αφίξεων, τα «bins» λόγω του μεγάλου αριθμού διαφορετικών τιμών παρατηρήσεων (δες τελευταία στήλη πίνακα 6) συμπεριλάμβαναν τιμές εντός ενός κλειστού διαστήματος μεγέθους 5 μονάδων, π.χ. για το τρίτο «bin» το κλειστό διάστημα είναι το [25 – 29]. Ενώ για το ιστόγραμμα αναχωρήσεων ή αλλιώς για τους χρόνους

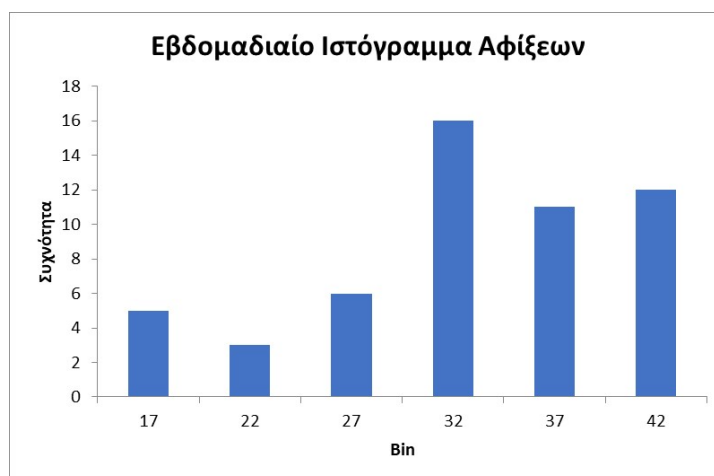


Figure 24: Εβδομαδιαίο Ιστόγραμμα Αφίξεων

εξυπηρέτησης το πλήθος των «bins» ήταν 3 και δεν είχαν εύρος αλλά διακριτές τιμές, εφόσον δεν παρουσίαζε πολλές διαφορετικές τιμές η στήλη median at port. Εφόσον οι είσοδοι για το μοντέλο αναμονής είναι έτοιμες το επόμενο στάδιο είναι η επιλογή των καταλληλότερων κατανομών καθώς και των υπόλοιπων χαρακτηριστικών του μοντέλου. Όπως έχει ήδη προαναφερθεί τα δύο πιο γενικά μοντέλα στα οποία γίνεται να εισαχθούν πολλές διαφορετικές κατανομές και να εξαχθούν τα βασικά αποτελέσματα όπως μέσο μήκος ουράς, μέσος χρόνος αναμονής κ.λπ. είναι το G.I/M/m και M/G/m. Η αρχή όμως

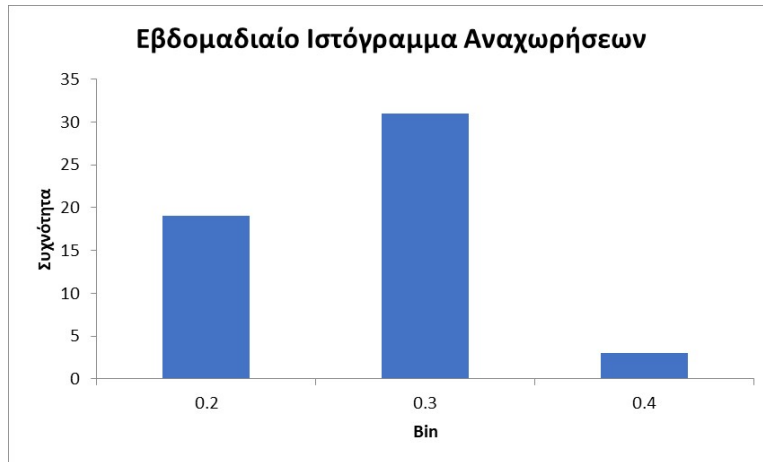


Figure 25: Εβδομαδιαίο Ιστόγραμμα Αναχωρήσεων

γίνεται με το βασικό μοντέλο M/M/m το οποίο θα χρησιμοποιηθεί ως σημείο αναφοράς για σύγκριση. Είναι γνωστό ότι οι οριακές πιθανότητες σταθερής κατάστασης (“steady-state limiting probabilities”) που εκφράζουν τη πιθανότητα το σύστημα αναμονής να εμπεριέχει k πελάτες δίνονται από τις παρακάτω σχέσεις [17]:

$$\begin{aligned}
 p_k &= p_0 \cdot \left(\frac{\lambda}{\mu}\right)^k \cdot \frac{1}{k!} \\
 &= p_0 \cdot (m \cdot \rho)^k \cdot \frac{1}{k!}, \text{ for } k < m
 \end{aligned} \tag{8}$$

$$\begin{aligned}
 p_k &= p_0 \cdot \left(\frac{\lambda}{\mu}\right)^k \cdot \frac{1}{m! \cdot m^{k-m}} \\
 &= p_0 \cdot (m \cdot \rho)^k \cdot \frac{1}{m! \cdot m^{k-m}}, \text{ for } k \geq m
 \end{aligned}$$

Έχοντας τις πιθανότητες διαθέσιμες και χρησιμοποιώντας τη σχέση $\sum_{k=0}^{\infty} p_k = 1$ λαμβάνουμε το παρακάτω για τη πιθανότητα p_0 :

$$\begin{aligned}
 p_0 &= \left[\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{k!} + \sum_{k=m}^{\infty} \frac{(m \cdot \rho)^k}{m! \cdot m^{k-m}} \right]^{-1} \\
 &= \left[\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{k!} + \frac{(m \cdot \rho)^m}{m!} \cdot \sum_{k=m}^{\infty} \rho^{k-m} \right]^{-1} \\
 &= \left[\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{k!} + \frac{(m \cdot \rho)^m}{m!} \cdot \sum_{i=0}^{\infty} \rho^i \right]^{-1} \\
 &= \left[\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{k!} + \frac{(m \cdot \rho)^m}{m!} \cdot \frac{1}{1 - \rho} \right]^{-1}
 \end{aligned}$$

Ακολουθεί η σχέση και η απόδειξη του μέσου αριθμού πελατών στο σύστημα από την οποία προκύπτουν ευκόλως ο μέσος χρόνος αναμονής στο σύστημα, ο μέσος χρόνος αναμονής στην ουρά και ο μέσος αριθμός πελατών στην ουρά. Τέλος, ορίζονται οι πιθανότητες :

A p_m : το σύστημα να περιέχει ακριβώς τόσους πελάτες όσους εξυπηρετητές διαθέτει και

B P_Q : το σύστημα να περιέχει τόσους πελάτες όσους εξυπηρετητές διαθέτει και πάνω

δηλαδή $P_Q = p_m + p_{m+1} + p_{m+2} + \dots$

$$\begin{aligned}
 E[N] &= \sum_{k=0}^{\infty} k \cdot p_k \\
 &= \sum_{k=0}^{m-1} k \cdot p_k + \sum_{k=m}^{\infty} k \cdot p_k \\
 &= p_0 \cdot (m \cdot \rho) \cdot \sum_{k=0}^{m-1} \frac{(m \cdot \rho)^{k-1}}{(k-1)!} + \frac{m^m \cdot p_0}{m!} \cdot \sum_{k=m}^{\infty} k \cdot \rho^k \\
 &= (m \cdot \rho) \cdot \left[\sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{(k)!} \right]^{-1} \cdot \sum_{k=0}^{m-1} \frac{(m \cdot \rho)^k}{(k)!} + \frac{m^m \cdot p_0}{m!} \cdot \rho \cdot \frac{\rho^m}{1 - \rho} \\
 &= m \cdot \rho + \rho \cdot \frac{(\rho \cdot m)^m \cdot p_0}{m! \cdot (1 - \rho)^2}
 \end{aligned}$$

$$E[T] = \frac{E[N]}{\lambda}$$

$$E[W] = E[T] - \frac{1}{\mu}$$

$$E[N_Q] = E[W] \cdot \lambda$$

$$p_m = p_0 \cdot (m \cdot \rho)^m \cdot \frac{1}{m!}$$

$$\begin{aligned} P_Q &= \frac{p_m}{1 - \rho} \\ &= \frac{(m \cdot \rho)^m}{m!} \cdot \frac{p_0}{1 - \rho} \end{aligned}$$

Τώρα που όλα τα δεδομένα εισόδου βρίσκονται στη θέση τους καθώς και το μοντέλο αναμονής M/M/m είναι έτοιμο προς χρήση, δύναται να γίνει η πρώτη ανάλυση και να εξαχθούν τα πρώτα αποτελέσματα. Πρώτα θα γίνει η ανάλυση για τα τεχνητά δεδομένα και το λιμάνι Incheon της Νότιας Κορέας και εν συνεχεία για τα πραγματικά δεδομένα και το λιμάνι Yokkaichi της Ιαπωνίας.

Όσο αφορά το λιμάνι Incheon της Νότιας Κορέας ο μέσος ημερήσιος ρυθμός αφίξεων, λ λαμβάνεται από το πίνακα 3 εκτελώντας μια μαθηματική πράξη μέσου όρου για τη τρίτη στήλη. Η ίδια διαδικασία ακολουθείται και για το μέσο ρυθμό εξυπηρέτησης στον αντίστοιχο πίνακα που δημιουργείται για τις ημερήσιες αναχωρήσεις. Έτσι προκύπτει ο παρακάτω πίνακας με τα δεδομένα εισόδου στο μοντέλο αναμονής M/M/4.

Δεδομένα			
λ	μ	$\rho = \frac{\lambda}{\mu \cdot m}$	πλήθος εξυπηρετητών = m
3.04	1.79	0.42	4

Table 7: Δεδομένα για το λιμάνι Incheon

Χρησιμοποιώντας την εξίσωση της σελίδας 66 καθώς και τις υπόλοιπες που τη συνοδεύουν λαμβάνονται τα εξής αποτελέσματα και χαρακτηριστικά του συστήματος αναμονής.

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	Τιμή
Πιθανότητα $\exists 0$ πελάτες	p_0	0.18
Πιθανότητα $\exists m$ πελάτες	p_m	0.06
Πιθανότητα \exists ουρά	P_Q	0.11
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	0.03
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	0.08
Μέσος χρόνος αναμονής στο σύστημα	$E[T]$	0.6
Μέσος αριθμός πελατών στο σύστημα	$E[N]$	1.8
Μέσος αριθμός απασχολημένων εξυπηρετητών	$E[M]$	1.7

Table 8: Αποτελέσματα για το λιμάνι Incheon

Τα αποτελέσματα φαίνεται να είναι ρεαλιστικά δεδομένου του ρυθμού συνωστισμού ρ , παρ' όλο που στο χ^2 test η "null hypothesis" δεν τηρείται αφού $\chi^2_{\alpha} \leq \chi^2 \Rightarrow 12.592 \leq 114.93$

Ακολουθώντας την ίδια διαδικασία και για το παράδειγμα του λιμένος Yokkaichi, ο μέσος ημερήσιος ρυθμός αφίξεων λ και ο μέσος ρυθμός εξυπηρέτησης μ για τους οποίους έχει ήδη παρατεθεί επεξήγηση στην ενότητα 4 για τον υπολογισμό τους, προσδιορίζονται και εν συνεχεία χρησιμοποιούνται μαζί με το πλήθος των εξυπηρετητών για να υπολογιστεί και ο ρυθμός συνωστισμού ρ .

Να επισημάνουμε σε αυτό το σημείο ότι για να είναι έγκυροι οι υπολογισμοί και η μεθοδολογία που ακολουθείται για τα μοντέλα αναμονής θα πρέπει ο ρυθμός συνωστισμού $\rho = \frac{\lambda}{\mu \cdot m}$ να είναι μικρότερος της μονάδας $\rho < 1$.

Τίποτα δεν αλλάζει όσο αφορά τους υπολογισμούς των χαρακτηριστικών μεγεθών του

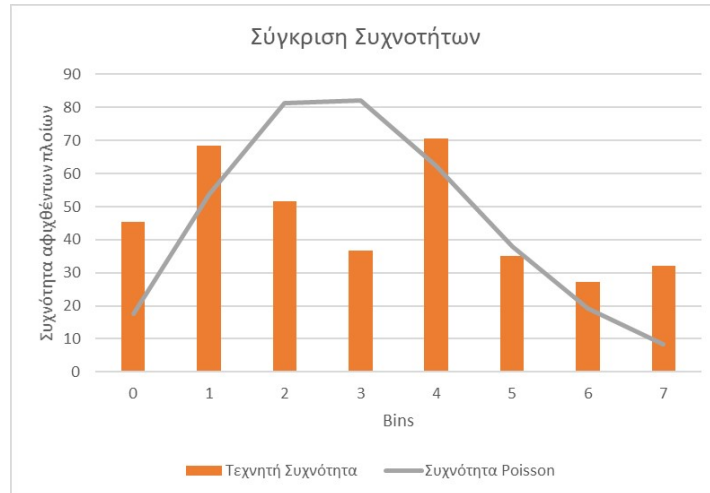


Figure 26: Σύγκριση συχνοτήτων

Δεδομένα			
λ	μ	$\rho = \frac{\lambda}{\mu \cdot m}$	πλήθος εξυπηρετητών = m
4.71	3.33	0.71	2

Table 9: Δεδομένα για το λιμάνι Yokkaichi

συστήματος αναμονής και έτσι προκύπτουν τα εξής αποτελέσματα :

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	Τιμή
Πιθανότητα $\exists 0$ πελάτες	p_0	0.2
Πιθανότητα $\exists m$ πελάτες	p_m	0.2
Πιθανότητα \exists ουρά	P_Q	0.66
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	0.34
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	1.6
Μέσος χρόνος αναμονής στο σύστημα	$E[T]$	0.64
Μέσος αριθμός πελατών στο σύστημα	$E[N]$	3
Μέσος αριθμός απασχολημένων εξυπηρετητών	$E[M]$	1.8

Table 10: Αποτελέσματα για το λιμάνι Yokkaichi

Τα αποτελέσματα φαίνεται να είναι ρεαλιστικά δεδομένου του ρυθμού συνωστισμού ρ , παρ' όλο που στο χ^2 test η "null hypothesis" δεν τηρείται για ακόμη μια φορά αφού $\chi^2_\alpha \leq \chi^2 \Rightarrow 9.488 \leq 35.84$

5.2 Εφαρμογή του G.I/M/m μοντέλου αναμονής

Όπως έχει ήδη προαναφερθεί μια μέθοδος που αξιοποιήσει οποιαδήποτε κατανομή γενικής μορφής G.I αλλά με ισόνομες και ανεξάρτητες τυχαίες μεταβλητές έχει βρεθεί στο [11]. Παρ' όλο που παραλείπεται η μαθηματική σκέψη για τον τρόπο που προκύπτουν οι τελικές μαθηματικές εξισώσεις, που όμως είναι ελεύθερα προσβάσιμες στο προαναφερθέν άρθρο, παρατίθεται ολόκληρη η διαδικασία για τον τρόπο λύσης ενός προβλήματος με μοντέλο

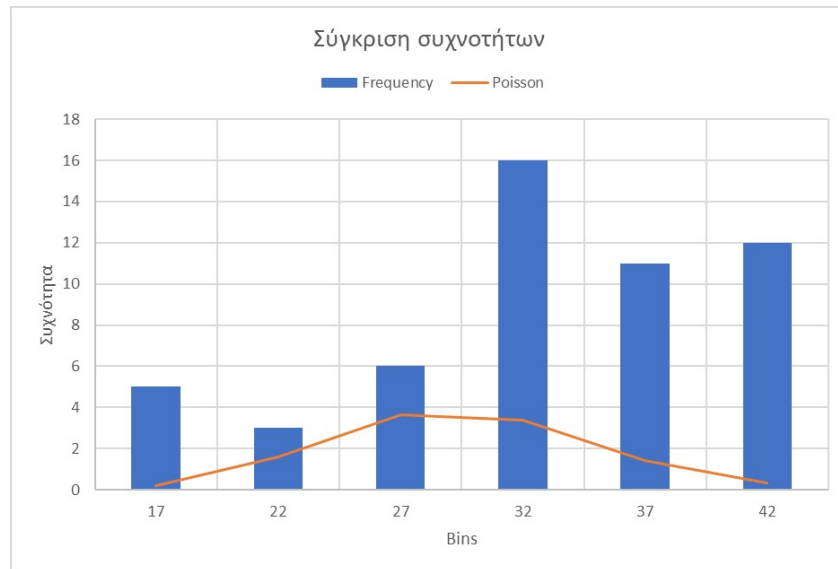


Figure 27: Σύγκριση συχνοτήτων

αναμονής της μορφής G.I/M/m.

Αρχικά δημιουργείται ένας πίνακας P που εμπεριέχει 3 υποπίνακες A,B,C οι οποίοι διαφοροποιούνται στο ότι κάθε ένας από αυτούς χρησιμοποιείται για διαφορετική κατάσταση στην οποία θα βρεθεί το σύστημα αναμονής.

$$P = \begin{bmatrix} A & & \\ & C & \\ & & B \end{bmatrix}$$

Πρώτα απ' όλα η μορφή των υποπινάκων είναι $(m \times m)$ για τον A, $(\infty \times m)$ για το B και $(\infty + m \times m)$ για το C. Γίνεται ευκόλως αντιληπτό από τον τρόπο που εκφράστηκε το μέγεθος του κάθε υποπίνακα, ότι ο αριθμός των σειρών του υποπίνακα A και B μαζί είναι ίσος με τον αριθμό των σειρών του υποπίνακα C. Κάθε στοιχείο ij του πίνακα P δηλώνει τη πιθανότητα το σύστημα αναμονής να μεταβεί από τη κατάσταση i πελάτες

στο χώρο αναμονής συμπεριλαμβανομένων και των πελατών στο χώρο εξυπηρέτησης στη κατάσταση j πελάτες.

$$p_{ij} = pr[q_2 = j | q_1 = i]$$

όπου q_2 και q_1 είναι διαδοχικές καταστάσεις του συστήματος

Αυτή η μετάβαση λαμβάνει χώρα στο χρονικό διάστημα ενός ενδιάμεσου χρόνου αφίξεως u ή αλλιώς στο χρονικό διάστημα μεταξύ δυο διαδοχικών αφίξεων. Για τα στοιχεία των υποπινάκων A,B,C ισχύουν οι παρακάτω σχέσεις :

$$\text{Για τον πίνακα A: } \begin{cases} i \leq m - 1 \\ \& \\ j \leq m - 1 \end{cases} \quad p_{ij} = [i + 1 - j | i + 1]$$

$$\text{Για τον πίνακα B: } \begin{cases} i \geq m \\ \& \\ j \leq m - 1 \end{cases} \quad p_{ij} = [m - j | m; i - m + 1]$$

$$\text{Για τον πίνακα C: } \begin{cases} j \geq m \\ p_{ij} = [i + 1 - j | m] \end{cases}$$

Αυτό που δηλώνουν οι παραπάνω σχέσεις είναι ουσιαστικά ότι στο πίνακα A εντάσσονται όλες εκείνες οι καταστάσεις στις οποίες ο νεοαφιχθέν πελάτης θα συναντήσει $m - 1$ πελάτες μπροστά του οι οποίοι όλοι θα εξυπηρετούνται και ένα σημείο εξυπηρέτησης θα είναι κενό. Επίσης, η πιθανότητα του υποπίνακα A εκφράζει τη δεσμευμένη πιθανότητα του να υπάρξουν $n = i + 1 - j$ αναχωρήσεις δεδομένου ότι $s = i + 1$ πελάτες εξυπηρετούνται στην αρχή του χρονικού διαστήματος. Για $j \leq i + 1$ η πιθανότητα παίρνει τη προαναφερθείσα μορφή, ενώ για οποιαδήποτε άλλη τιμή του j , $p_{ij} = 0$

Στο πίνακα Β ισχύει ότι $i \geq m$ και άρα στην αρχή του χρονικού διαστήματος u , δηλαδή στην αρχική κατάσταση του συστήματος θα υπάρχουν $s = i - m + 1$ επιπλέον πελάτες που θα περιμένουν μαζί με εκείνους που εξυπηρετούνται. Επίσης, επειδή στο τέλος της μετάβασης, δηλαδή μετά το πέρας του χρονικού διαστήματος μεταξύ δύο διαδοχικών αφίξεων u , ισχύει $j \leq m - 1$, δεν θα υπάρχει πελάτης στο χώρο αναμονής και έτσι $n = m - j$ εξυπηρετητές θα είναι ελεύθεροι. Επιπροσθέτως, θέτουμε ως χρόνο U το χρονικό διάστημα αναμονής του τελευταίου πελάτη από το πλήθος s .

Για τον Πίνακα Γ, ο οποίος δεν χρησιμοποιείται στους υπολογισμούς, έχουμε ότι $j \geq m$ και έτσι λαμβάνουμε υπ' όψη μόνο τις τιμές του n για τις οποίες ισχύει $0 \leq n \leq i + 1 - m$ από όπου συμπεραίνεται ότι στη τελική κατάσταση του συστήματος θα είναι όλα τα σημεία εξυπηρέτησης απασχολημένα. Για αυτόν τον υποπίνακα, $n = i + 1 - j$ και όπως στον πίνακα Α είναι ο αριθμός των πελατών που ολοκλήρωσαν την εξυπηρέτησή τους κατά τη διάρκεια του χρονικού διαστήματος u .

Οι παραπάνω πιθανότητες δίνονται από τις παρακάτω σχέσεις σε αντιστοιχία με τον υποπίνακα που ανήκουν.

Αν $ij \in A$:

$$p_{ij} = [n|s] = \int_0^{\infty} [n|s; u] dA(u)$$

$$[n|s; u] = \binom{s}{n} \cdot \left(1 - e^{-\frac{u}{b}}\right)^n \cdot e^{-(s-n) \cdot \frac{u}{b}}$$

Αν $ij \in B$:

$$p_{ij} = [n|m; s] = \int_0^\infty [n|m; s; u] dA(u)$$

$$[n|m; s; u] = \frac{1}{(s-1)!} \cdot \binom{m}{b}^s \cdot \int_0^u e^{-\frac{s \cdot U}{b}} \cdot U^{s-1} \cdot [n|m; u - U] dA(u)$$

Αν $ij \in C$:

$$p_{ij} = [n|m] = \int_0^\infty [n|m; u] dA(u)$$

$$[n|m; u] = e^{-\frac{m \cdot u}{b}} \cdot \frac{\left(\frac{m \cdot u}{b}\right)^n}{n!}$$

Στη συνέχεια κατασκευάζεται το διάνυσμα χ τα στοιχεία του οποίου προσδιορίζονται στο στάδιο των υπολογισμών και χρησιμοποιούνται για τον υπολογισμό των τελικών αποτελεσμάτων. Το διάνυσμα χ παίρνει τη παρακάτω μορφή $\chi = [\kappa_0, \kappa_1, \dots, \kappa_{m-2}, 1, \eta, \eta^2, \eta^3, \dots]$. Οι όροι κ δεν υπάρχουν όταν το σύστημα έχει μοναδικό εξυπηρετητή $m = 1$. Από αυτό το σημείο και ύστερα μένει να υπολογιστούν οι όροι του διανύσματος χ έτσι ώστε να ληφθούν τα ποσοτικά χαρακτηριστικά όπως η πιθανότητα να μην υπάρχει ουρά, η πιθανότητα να μην υπάρχει χρόνος αναμονής, το μέσο μήκος ουράς και ο μέσος χρόνος αναμονής. Ένα πολύ σημαντικό συγκριτικό πλεονέκτημα που λαμβάνεται είναι η κατανομές του μήκους ουράς και χρόνου αναμονής που προσφέρουν μια γενική εικόνα και πολύ πιο ολοκληρωμένη.

Οι όροι κ και η έχουν ρόλο κλειδί για το προσδιορισμό των τελικών αποτελεσμάτων και υπολογίζονται στη συνέχεια. Ο η όρος υπολογίζεται με βάση την παρακάτω εξίσωση και πρέπει σύμφωνα με το [11] $0 < \eta < 1$.

$$\eta = \int_0^{\infty} e^{-(1-\eta) \cdot \frac{u}{\rho}} dA(u)$$

Στη περίπτωση που οι ενδιάμεσοι χρόνοι αφίξεων ακολουθούν την εκθετική κατανομή, δηλαδή η είσοδος είναι Poissonian $dA(u) = e^{-u} du$ τότε ο όρος η γίνεται ίσος με τον ρυθμό αφίξεων λ όπως αποδεικνύεται παρακάτω.

$$\begin{aligned} \eta &= \int_0^{\infty} e^{-(1-\eta) \cdot \frac{u}{\rho}} \cdot e^{-u} du \\ &= \frac{1}{\left(1 + \frac{1}{\rho}\right)} \cdot \frac{1}{1 - \frac{\eta}{\rho \cdot \left(1 + \frac{1}{\rho}\right)}} \end{aligned}$$

Χρησιμοποιώντας λίγη άλγεβρα καταλήγουμε στη παρακάτω δευτεροβάθμια εξίσωση όπου η μοναδική ρίζα στο ανοιχτό διάστημα $(0, 1)$ είναι $\eta = \rho$.

$$\eta^2 - \eta \cdot (\rho + 1) + \rho = 0$$

Η μετάβαση από ολοκλήρωμα σε απλό γινόμενο κλασμάτων στη παραπάνω μαθηματική σχέση όπως και στη περίπτωση μιας γενικής κατανομής π.χ. KDE που θα χρησιμοποιηθεί αργότερα, αντί για Poisson πραγματοποιείται με τη χρήση αναλυτικών μαθηματικών μοντέλων από τη βιβλιοθήκη SymPy της γλώσσας προγραμματισμού Python.

```
1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
```

```

6
7
8 u,a,c=symbols('u a c',real=True)
9
10
11 Integral(exp(-u-(1-a)*u/c), (u,0,oo)).doit().evalf()

```

Για τους όρους κ πρώτα κατασκευάζονται οι υποπίνακες ανάλογα με τον αριθμό των εξυπηρετητών. Για ένα σύστημα αναμονής με 2 εξυπηρετητές οι υποπίνακες λαμβάνουν τη παρακάτω μορφή. Όπως ήδη έχει αναφερθεί, για τους υπολογισμούς απαραίτητοι είναι μόνο οι υποπίνακες A και B, άρα μόνο αυτοί παρατίθενται.

$$A = \begin{bmatrix} [1|1] & [0|1] \\ [2|2] & [1|2] \end{bmatrix}$$

$$B = \begin{bmatrix} [2|2; 1] & [1|2; 1] \\ [2|2; 2] & [1|2; 2] \\ [2|2; 3] & [1|2; 3] \\ \vdots & \vdots \end{bmatrix}$$

Ενώ για 4 εξυπηρετητές λαμβάνεται το παρακάτω ζεύγος υποπινάκων.

$$A = \begin{bmatrix} [1|1] & [0|1] & \cdot & \cdot \\ [2|2] & [1|2] & [0|2] & \cdot \\ [3|3] & [2|3] & [1|3] & [0|3] \\ [4|4] & [3|4] & [2|4] & [1|4] \end{bmatrix}$$

$$B = \begin{bmatrix} [4|4; 1] & [3|4; 1] & [2|4; 1] & [1|4; 1] \\ [4|4; 2] & [3|4; 2] & [2|4; 2] & [1|4; 1] \\ [4|4; 3] & [3|4; 3] & [2|4; 3] & [1|4; 3] \\ [4|4; 4] & [3|4; 4] & [2|4; 4] & [1|4; 4] \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Γίνεται αντιληπτό το πώς μπορεί να επεκταθεί για m εξυπηρετητές.

Δεδομένου των ιδιοτήτων του διανύσματος χ και του πίνακα P ισχύει η παρακάτω σχέση η οποία προσφέρει το μέσο για τον υπολογισμό των κ όρων. Για περισσότερες λεπτομέρειες παραπέμπεται ο αναγνώστης στο [11].

$$\kappa_j = \sum_a \kappa_a \cdot p_{aj}$$

Χρησιμοποιώντας τους πίνακες των παραπάνω παραδειγμάτων προκύπτουν τα παρακάτω σύνολα εξισώσεων που οδηγούν στο προσδιορισμό των κ όρων. Οι πιθανότητες δίνονται σύμφωνα με τον τύπος των υποπινάκων A και B .

Για σύστημα αναμονής με 2 εξυπηρετητές

$$\kappa_1 = \kappa_0 \cdot p_{01} + \kappa_1 \cdot p_{11} + \eta \cdot p_{21} + \eta^2 \cdot p_{31} + \dots$$

$$1 = \kappa_0 \cdot [0|1] + [1|2] + \sum_{a=1}^{\infty} [1|2; a] \cdot \eta^a$$

Για σύστημα αναμονής με 4 εξυπηρετητές

$$\kappa_3 = \kappa_0 \cdot p_{03} + \kappa_1 \cdot p_{13} + \kappa_2 \cdot p_{23} + \kappa_3 \cdot p_{33} + \eta \cdot p_{43} + \eta^2 \cdot p_{53} + \dots$$

$$1 = \kappa_2 \cdot [0|3] + [1|4] + \sum_{a=1}^{\infty} [1|4; a] \cdot \eta^a$$

$$\kappa_2 = \kappa_0 \cdot p_{02} + \kappa_1 \cdot p_{12} + \kappa_2 \cdot p_{22} + \kappa_3 \cdot p_{32} + \eta \cdot p_{42} + \eta^2 \cdot p_{52} + \dots$$

$$\kappa_2 = \kappa_1 \cdot [0|2] + \kappa_2 \cdot [1|3] + [2|4] + \sum_{a=1}^{\infty} [2|4; a] \cdot \eta^a$$

$$\kappa_1 = \kappa_0 \cdot p_{01} + \kappa_1 \cdot p_{11} + \kappa_2 \cdot p_{21} + \kappa_3 \cdot p_{31} + \eta \cdot p_{41} + \eta^2 \cdot p_{51} + \dots$$

$$\kappa_1 = \kappa_0 \cdot [0|1] + \kappa_1 \cdot [1|2] + \kappa_2 \cdot [2|3] + [3|4] + \sum_{a=1}^{\infty} [3|4; a] \cdot \eta^a$$

Λύνοντας το σύστημα των εξισώσεων λαμβάνοντα διαδοχικά όλες οι τιμές των κ . Οι πιθανότητες του πίνακα Α προκύπτουν άμεσα και με ευκολία ενώ του πίνακα Β το άθροισμα που εμφανίζεται σε κάθε εξίσωση του συστήματος των εξισώσεων δεν συγκλίνει αναλυτικά σε κάποιο πεπερασμένο όριο με αποτέλεσμα ο πιο αποτελεσματικός τρόπος είναι να συμπεριληφθούν όσο το δυνατόν περισσότεροι όροι του αθροίσματος. Ο συγγραφέας στο [11] έχει καταφέρει να φτάσει σε απλοποιημένες μορφές ύστερα από μετασχηματισμούς που δεν είναι φανεροί π.χ. για το σύστημα G.I/M/2 καταλήγει στη παρακάτω απλή μαθηματική έκφραση.

$$\left(\frac{2}{2 \cdot \eta - 1} - \kappa_0\right) \cdot \int_0^{\infty} e^{\frac{-u}{2 \cdot \rho}} dA(u) = \frac{1}{2 \cdot \eta - 1}$$

Στη περίπτωση που το σύστημα είναι το M/M/2, δηλαδή η είσοδος στους ενδιάμεσους χρόνους αφίξεων είναι Poisson τότε για τη σύγκριση που θα πραγματοποιηθεί αργότερα προκύπτει ότι $\kappa_0 = \frac{1}{2 \cdot \rho}$ όπου $\rho = \frac{\lambda}{m \cdot \mu}$ ο ρυθμός συνωστισμού ανά εξυπηρετητή.

Παρακάτω εμφανίζονται τα αποτελέσματα για το παράδειγμα του Incheon και του Yokkaichi με 4 και 2 εξυπηρετητές αντίστοιχα. Ως G.I. κατανομή χρησιμοποιήθηκε για το λιμάνι του Yokkaichi:

- ο KDE με “kernel” τη κατανομή Gauss με τυπική απόκλιση $\sigma=3.4$ και $h=1$

- Η δυωνυμική κατανομή (Γεωμετρική είσοδος) με παραμέτρους $(n, p) = (1727, 0.02)$. Η επιλογή των παραμέτρων έγινε με βάση τη καλύτερη προσαρμογή στο ιστόγραμμα συχνοτήτων.
- Η κατανομή Poisson για λόγους σύγκρισης

Ενώ για το λιμάνι του Incheon έγινε απλώς σύγκριση με τη κατανομή Poisson.

Στη περίπτωση που η G.I είναι ο KDE και η δυωνυμική κατανομή πρέπει πρώτα να υπολογιστεί ο όρος η . Όλη η υλοποίηση έχει γίνει σε γλώσσα Python άρα παρακάτω παρατίθεται ο κώδικας μαζί με σχόλια για τη καλύτερη κατανόηση της μεθοδολογίας βάση της οποίας προκύπτουν τα αποτελέσματα. Συμπληρωματικά στα αποτελέσματα που παρουσιάστηκαν στην υποενότητα 5.1 στους πίνακες 8 και 10, υπολογίζονται η πιθανότητα να μην υπάρξει ουρά, η πιθανότητα να μην υπάρχει χρόνος αναμονής καθώς και οι κατανομές του μήκους ουράς και χρόνου αναμονής. Οι μαθηματικοί τύποι που δίνουν αυτά τα επιπρόσθετα αποτελέσματα, οι οποίοι συγκεντρωτικά μπορούν να βρεθούν στο [11], παρατίθενται παρακάτω:

$$P_{N_Q=0} = \frac{\sum \kappa + 1 + \eta}{\kappa + \frac{1}{1 - \eta}}$$

$$P_{W=0} = \frac{\sum \kappa + 1}{\kappa + \frac{1}{1 - \eta}}$$

$$E[N_Q] = \frac{1 - P_{N_Q=0}}{1 - \eta}$$

$$E[W] = \frac{\beta \cdot (1 - P_{W=0})}{m \cdot (1 - \eta)}$$

Οι συναρτήσεις πυκνότητας πιθανότητας του μήκους ουράς και του χρόνου αναμονής δίνονται από τους ακόλουθους τύπους:

$$PDF_{N_Q} = (1 - \eta) \cdot \eta^{Q-1}, \text{ όπου } Q \text{ το πλήθος των πελατών στην ουρά και } Q \in N \quad (9)$$

$$PDF_W = e^{-\frac{w}{c}} \cdot \frac{1}{c}, \text{ όπου } c = \frac{b}{m \cdot (1 - \eta)} \text{ και } 0 < w < \infty \quad (10)$$

Ακολουθεί το κομμάτι κώδικα για επίλυση του συστήματος εξισώσεων των κ όρων για σύστημα αναμονής με 4 εξυπηρετητές με δεδομένα από το λιμάνι Incheon:

```

1 import numpy as np
2 import math
3 from scipy import optimize
4 from sympy import factorial
5 from sympy import *
6 from sympy import solveset
7 from sympy.solvers.solveset import solveset, solveset_real
8 import sys
9 sys.setrecursionlimit(1500)
10
11 #Creating the variables for solving the equations
12 #with the SymPy library
13 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
14
15 #For the Poisson input the hta term is equal to the
16 #relative traffic intensity
17 r=0.56/(4*0.33)
18
19 def factL(z1,z2):
20     return (factorial(z1)/factorial(z2)/factorial(z1-z2))
21

```

```
22 #formulating the probabilities of table A
23 def pijA(m,n,b):
24     return (Integral(factL(m,n)*\
25         ((1-exp(-u/b))**n)*exp(-(m-n)*u/b)*(exp(-u/0.33)/0.33)\
26         , (u,0,oo)).doit().evalf())
27
28 #formulating the probabilities of table B
29 def funcsup(m,n,s,b):
30     return (factL(s,n)*(1/factorial(m-1))*((s/b)**m)\
31         *Integral(Integral(exp(-s*U/b)*(U**(m-1))\
32         *((1-exp((U-u)/b))**n)*\
33         exp((s-n)*(U-u)/b),\
34         (U,0,u))*(exp(-u/0.33)/0.33), (u,0,oo)).doit().evalf())
35
36
37 #Solving for the first k term i.e. k2
38 #Only the first 15 terms are considered
39 sum1=0
40 for i in range (1,15):
41     sum1+=funcsup(i,1,4,0.56)*(r**i)
42 p03=pijA(3,0,0.56)
43 p14=pijA(4,1,0.56)
44 m2=(1-p14)/p03
45
46
47 #Solving for the second k term i.e. k1
48 #Only the first 15 terms are considered
49 sum1=0
50 for i in range (1,15):
```

```
51     sum1+=funcsup(i,2,4,0.56)*(r**i)
52 p02=pijA(2,0,0.56)
53 p13=pijA(3,1,0.56)
54 p24=pijA(4,2,0.56)
55 m1=(m2-m2*p13-p24-sum1)/p02
56
57
58 #Solving for the third k term i.e. k0
59 #Only the first 15 terms are considered
60 sum1=0
61 for i in range (1,15):
62     sum1+=funcsup(i,3,4,0.56)*(r**i)
63 p01=pijA(1,0,0.56)
64 p12=pijA(2,1,0.56)
65 p23=pijA(3,2,0.56)
66 p34=pijA(4,3,0.56)
67 m0=(m1-m1*p12-m2*p23-p34-sum1)/p01
68
69 summ=m0+m1+m2
70
71 #Obtain the mean value results and the probabilities
72 w0=(summ+1)/(summ+(1/(1-r)))
73 Q0=(summ+1+r)/(summ+(1/(1-r)))
74 Ew=0.56*(1-w0)/(4*(1-r))
75 EQ=(1-Q0)/(1-r)
76
77 print ('The probability not having to wait is, i.e. w is zero', w0)
78 print ('The probability having zero customers in queue is', Q0)
79 print ('The mean value of customers in true queue is', EQ)
```

```
80 print ('The mean value of waiting time is', Ew)
```

Αποτελέσματα				
Χαρακτηριστικό Μέγεθος	Συμβολισμός	M/M4	G.I/M/4	Ποσοστιαία Διαφορά
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	0.026	0.028	8%
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	0.078	0.087	10%
Πιθανότητα \exists ουρά	P_Q	0.11	-	-
Πιθανότητα να μην υπάρχει ουρά	$P_{N_Q=0}$	-	0.95	-

Table 11: Σύγκριση αποτελεσμάτων για μοντέλο M/M/4 και G.I/M/4 με είσοδο Poisson

Μια τελευταία επιβεβαίωση που γίνεται είναι ότι το άθροισμα όλων των πιθανοτήτων πρέπει να ισούται με τη μονάδα. Αυτό το άθροισμα μπορεί να γραφτεί ως εξής:

$$P_Q - p_m + P_{N_Q=0} = 0.11 - 0.06 + 0.95 = 1$$

Όπως εύκολα συμπεραίνεται από τα διαγράμματα και πάντα σχετικά με τα δεδομένα που είναι διαθέσιμα είναι αναμενόμενο η πιθανότητα για μήκος ουράς μεγαλύτερο από 2 πελάτες να μειώνεται απότομα αφού ο ρυθμός συνωστισμού ανά εξυπηρετητή και το μέσο μήκος ουράς έχουν μικρές τιμές ($\rho = 0.42$ και $E[N_Q] = 0.08$). Αυτό αποτελεί μια επιπρόσθετη επιβεβαίωση ότι η εφαρμογή του παρόντος μοντέλου είναι καλή. Είναι σημαντικό να σημειωθεί ότι οι κατανομές έχουν ασυνέχεια, δηλαδή μια συγκέντρωση πιθανότητας, στο σημείο 0 και αν την εξαιρέσουμε οι κατανομές όπως φαίνεται και από τις παραπάνω εξισώσεις τους, 9 και 10 είναι η εκθετική και γεωμετρική με τη σειρά που εμφανίζονται τα διαγράμματα. Αυτό το αποτέλεσμα σημειώθηκε για μοντέλα αναμονής με μοναδικό εξυπηρετητή πρώτα στο [18] και γενικεύθηκε για m εξυπηρετητές στο [11].

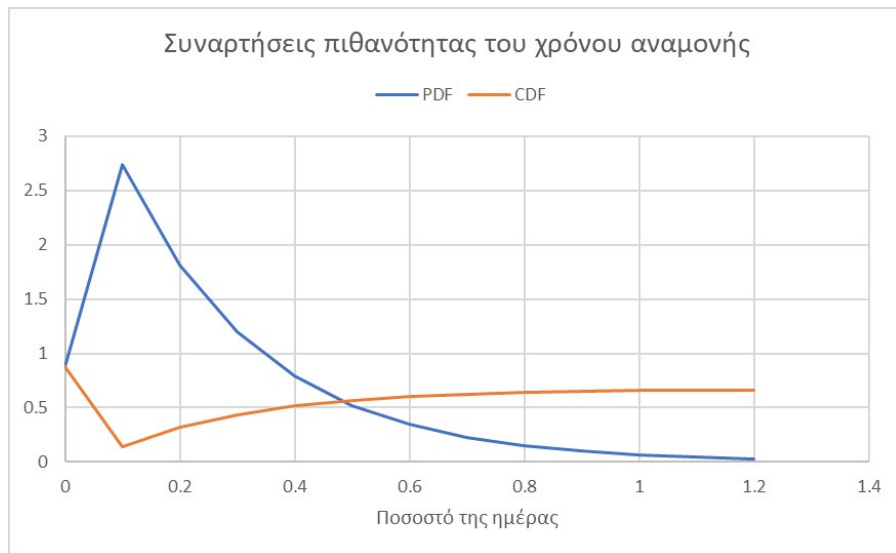


Figure 28: Συναρτήσεις πιθανότητας του χρόνου αναμονής

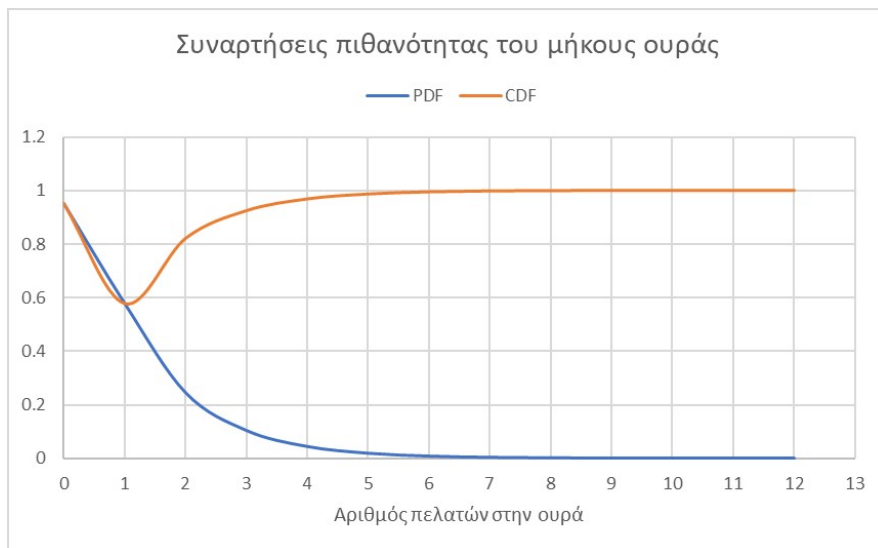


Figure 29: Συναρτήσεις πιθανότητας του μήκους ουράς

Ακολουθεί κομμάτι κώδικα για επίλυση του συστήματος εξισώσεων των k όρων για σύστημα αναμονής με 2 εξυπηρετητές με δεδομένα από το λιμάνι Yokkaichi και Poisson είσοδο για τους ενδιαμέσους χρόνους αφίξεων:

```
1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6 from sympy import factorial
7 import sys
8 sys.setrecursionlimit(1500)
9
10 #Creating the variables for solving the equations
11 #with the SymPy library
12 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
13
14 #For the Poisson input the hta term is equal to the
15 #relative traffic intensity
16 r=0.3/(2*0.21)
17
18 def factL(z1,z2):
19     return (factorial(z1)/factorial(z2)/factorial(z1-z2))
20
21 #formulating the probabilities of table A
22 def pijA(m,n,b):
23     return (Integral(factL(m,n)*\
24         ((1-exp(-u/b))**n)*exp(-(m-n)*u/b)*(exp(-u/0.21)/0.21),\
```

```

25         (u,0,oo)).doit().evalf())
26
27 #formulating the probabilities of table B
28 def funcsup(m,n,s,b):
29     return (factL(s,n)*(1/factorial(m-1))*((s/b)**m)\
30             *Integral(Integral(exp(-s*U/b)*(U**(m-1))\
31                             *((1-exp((U-u)/b))**n)*\
32                             exp((s-n)*(U-u)/b),\
33                             (U,0,u))*(exp(-u/0.21)/0.21), (u,0,oo)).doit().evalf())
34
35
36 #Solving for the k term i.e. k0
37 #Only the first 6 terms are considered
38 x1=funcsup(1,1,2,0.3)*(r**1)
39 x2=funcsup(2,1,2,0.3)*(r**2)
40 x3=funcsup(3,1,2,0.3)*(r**3)
41 x4=funcsup(4,1,2,0.3)*(r**4)
42 x5=funcsup(5,1,2,0.3)*(r**5)
43 x6=funcsup(6,1,2,0.3)*(r**6)
44 sum1=x1+x2+x3+x4+x5+x6
45 p01=pijA(1,0,0.3)
46 p12=pijA(2,1,0.3)
47 m0=(1-sum1-p12)/p01
48
49 summ=m0
50
51 #Obtain the mean value results and the probabilities
52 w0=(summ+1)/(summ+(1/(1-r)))
53 Q0=(summ+1+r)/(summ+(1/(1-r)))

```



```

54 Ew=0.3*(1-w0)/(2*(1-r))
55 EQ=(1-Q0)/(1-r)
56
57 print ('The probability not having to wait is, i.e. w is zero', w0)
58 print ('The probability having zero customers in queue is', Q0)
59 print ('The mean value of customers in true queue is', EQ)
60 print ('The mean value of waiting time is', Ew)

```

Αποτελέσματα				
Χαρακτηριστικό Μέγεθος	Συμβολισμός	M/M2	G.I/M/2	Ποσοστιαία Διαφορά
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	0.34	1.59	8%
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	0.31	1.49	6%
Πιθανότητα \exists ουρά	P_Q	0.66	-	-
Πιθανότητα να μην υπάρχει ουρά	$P_{N_Q=0}$	-	0.58	-

Table 12: Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο Poisson

Μια τελευταία επιβεβαίωση που γίνεται είναι ότι το άθροισμα όλων των πιθανοτήτων πρέπει να ισούται με τη μονάδα. Αυτό το άθροισμα μπορεί να γραφτεί ως εξής:

$$P_Q - p_m + P_{N_Q=0} = 0.66 - 0.2 + 0.58 = 1.04$$

Ακολουθεί κομμάτι κώδικα για επίλυση του συστήματος εξισώσεων των κ όρων και το προσδιορισμό του η όρου για σύστημα αναμονής με 2 εξυπηρετητές με δεδομένα από το λιμάνι Yokkaichi και KDE είσοδο για τους ενδιάμεσους χρόνους αφίξεων:

```

1 import numpy as np
2 import math

```

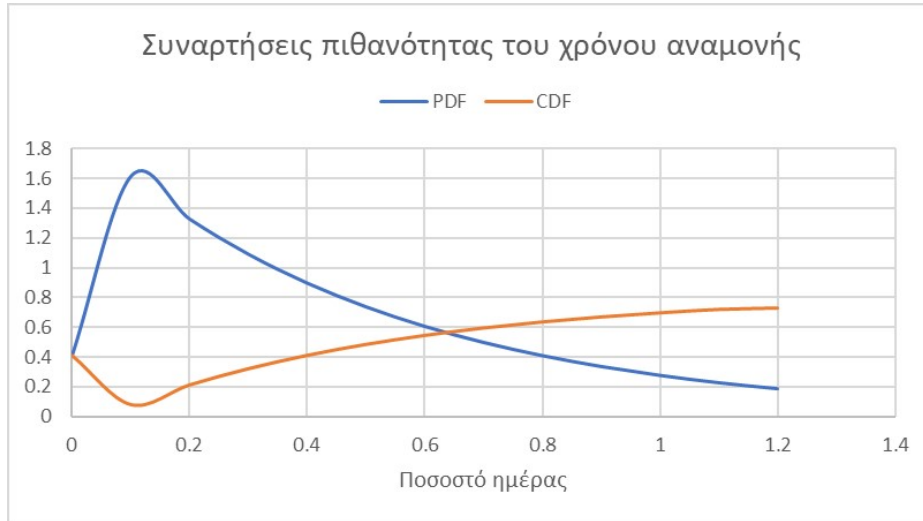


Figure 30: Συναρτήσεις πιθανότητας του χρόνου αναμονής

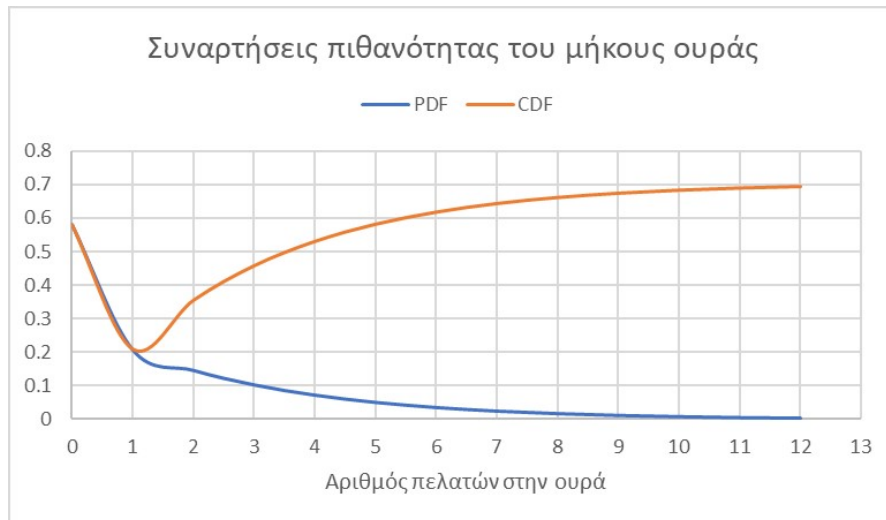


Figure 31: Συναρτήσεις πιθανότητας του μήκους ουράς

```
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6 from scipy import optimize
7 from sympy.utilities.lambdify import lambdify, implemented_function
8 from sympy.solvers.inequalities import solve_univariate_inequality
9
10 #Creating the variables for solving with SymPy library
11 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
12 init_printing(use_unicode=True)
13
14 #Setting the standard deviation and bandwidth
15 s=3.4
16 h=1
17
18 #Setting the frequency of each bin
19 cv=[5,3,6,16,11,12]
20
21 #Setting the relative traffic intensity
22 r=(0.3/(2*0.21))
23
24 #Creating the pdf of the KDE
25 Au1=(1/(53*h*sqrt(2*pi)))*(cv[0]*exp(-0.5*(((u)-17)/h)**2)/(2*s**2)+\
26         cv[1]*exp(-0.5*(((u)-22)/h)**2)/(2*s**2))+cv[2]*\
27         exp(-0.5*(((u)-27)/h)**2)/(2*s**2))
28 Au2=(1/(53*h*sqrt(2*pi)))*(cv[3]*exp(-0.5*(((u)-32)/h)**2)/(2*s**2)\
29         +cv[4]*exp(-0.5*(((u)-37)/h)**2)/(2*s**2))+cv[5]*exp(-0.5\
30         *(((u)-42)/h)**2)/(2*s**2))
31 Au=Au1+Au2
```

```
32
33 #To simplify calculations the KDE is set equal to the below
34 Au=exp(-u**2)
35
36 #Solving the equation for the hta term
37 IAx=b-integrate(exp(u*(b)/r)*exp(-(1)*u/r)*Au, (u,0,oo))
38 pprint (IAx)
39 IAx=lambdify(b,IAx)
40 pprint (IAx)
41 root=optimize.newton(IAx,x0=0.9,maxiter=4000)
42 print (optimize.newton(IAx,x0=0.9,full_output=True))

1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6 from sympy import factorial
7 import sys
8 sys.setrecursionlimit(1500)
9
10 #Setting the parameters of the Gaussian KDE i.e. standard deviation
11 #and bandwidth
12 s=3.4
13 h=1
14
15 #Setting the frequency of each bin
16 cv=[5,3,6,16,11,12]
17
```

```

18 #Creating the variables for the SymPy library
19 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
20
21 #Creating the pdf of KDE
22 Au1=(1/(53*h*sqrt(2*pi)))*(cv[0]*exp(-0.5*(((u)-17)/h)**2)/(2*s**2)+\
23      cv[1]*exp(-0.5*(((u)-22)/h)**2)/(2*s**2))+cv[2]*\
24      exp(-0.5*(((u)-27)/h)**2)/(2*s**2))
25 Au2=(1/(53*h*sqrt(2*pi)))*(cv[3]*exp(-0.5*(((u)-32)/h)**2)/(2*s**2)\
26      +cv[4]*exp(-0.5*(((u)-37)/h)**2)/(2*s**2))+cv[5]*exp(-0.5\
27      *(((u)-42)/h)**2)/(2*s**2))
28 dAu=Au1+Au2
29
30 #To simplify calculations the KDE is set equal to the below
31 dAu=exp(-u**2)
32
33 #Solution obtained for the hta term
34 l=0.9
35 r=(0.3/(2*0.21))
36
37 def factL(z1,z2):
38     return (factorial(z1)/factorial(z2)/factorial(z1-z2))
39
40 #formulating the probabilities of table A
41 def pijA(m,n,b):
42     return (Integral(factL(m,n)*\
43                    ((1-exp(-u/b))**n)*exp(-(m-n)*u/b)*dAu,\
44                    (u,0,oo)).doit().evalf())
45
46 #formulating the probabilities of table B

```

```

47 def funcsup(m,n,s,b):
48     return (factL(s,n)*(1/factorial(m-1))*((s/b)**m)\
49             *Integral(Integral(exp(-s*U/b)*(U**(m-1))\
50                 *((1-exp((U-u)/b))**n))*\
51                 exp((s-n)*(U-u)/b),\
52                 (U,0,u))*dAu, (u,0,oo)).doit().evalf())
53
54 #Solving for the k term i.e. k0
55 #Only the first 5 terms are considered
56 sum1=0
57 for i in range (1,5):
58     sum1+=funcsup(i,1,2,0.3)*(1**i)
59 p01=pijA(1,0,0.3)
60 p12=pijA(2,1,0.3)
61 m0=(1-sum1-p12)/p01
62
63 summ=m0
64
65 #Obtain the mean value results and the probabilities
66 w0=(summ+1)/(summ+(1/(1-1)))
67 Q0=(summ+1+1)/(summ+(1/(1-1)))
68 Ew=0.3*(1-w0)/(2*(1-1))
69 EQ=(1-Q0)/(1-1)
70
71 print ('The probability not having to wait is, i.e. w is zero', w0)
72 print ('The probability having zero customers in queue is', Q0)
73 print ('The mean value of customers in true queue is', EQ)
74 print ('The mean value of waiting time is', Ew)

```

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	G.I/M/2
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	1.23
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	7.4
Πιθανότητα να μην υπάρχει χρόνος αναμονής	$P_{W=0}$	0.18
Πιθανότητα να μην υπάρχει ουρά	$P_{N_Q=0}$	0.26

Table 13: Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο KDE

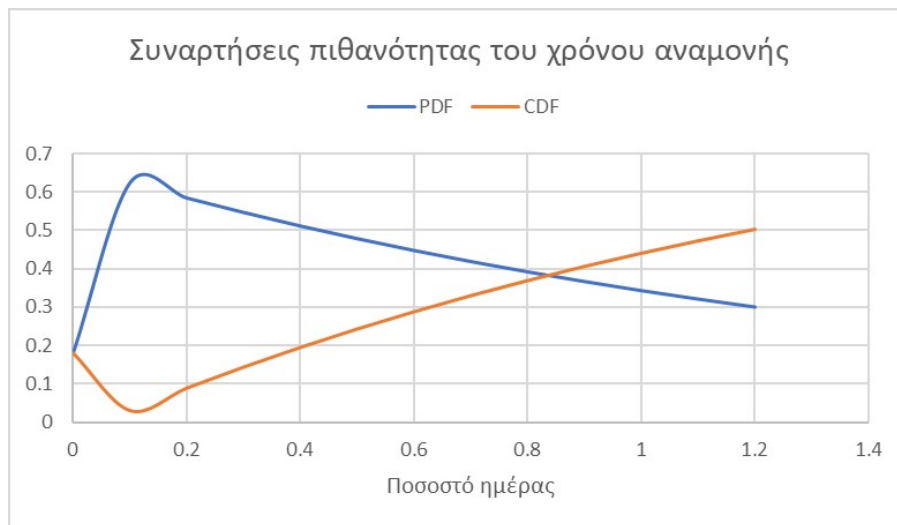


Figure 32: Συναρτήσεις πιθανότητας του χρόνου αναμονής

Τα αποτελέσματα συγκριτικά με την είσοδο Poisson διαφέρουν σημαντικά. Οι μέσες τιμές είναι σημαντικά αυξημένες (περίπου με έναν παράγοντα 3.5 – 5). Σχετικά με τις κατανομές πιθανότητας, ο ρυθμός που μειώνεται η συνάρτηση πυκνότητας πιθανότητας

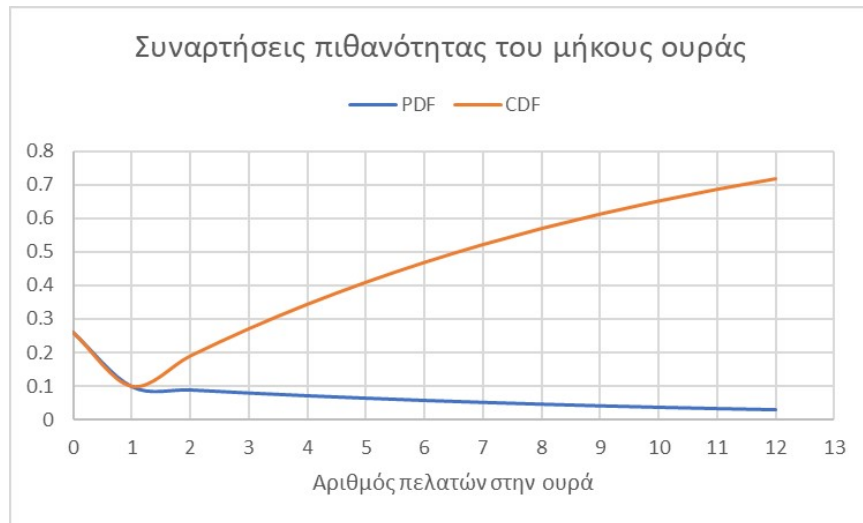


Figure 33: Συναρτήσεις πιθανότητας του μήκους ουράς

είναι σημαντικά μικρότερος ενώ η αθροιστική κατανομή πιθανότητας (CDF) αργεί να λάβει το λογαριθμικό προφίλ της, οι τιμές έχουν σχεδόν μειωθεί στο μισό και παρουσιάζει μια γραμμικότητα. Για παράδειγμα η πιθανότητα για χρόνο αναμονής μεταξύ 4.8 και 9.6 ωρών είναι 20% για είσοδο Poisson και 10% για είσοδο KDE.

Ακολουθεί κομμάτι κώδικα για επίλυση του συστήματος εξισώσεων των κ όρων και το προσδιορισμό του η όρου για σύστημα αναμονής με 2 εξυπηρετητές με δεδομένα από το λιμάνι Yokkaichi και γεωμετρική είσοδο για τους ενδιάμεσους χρόνους αφίξεων:

```

1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6 from scipy import optimize
7 from sympy.utilities.lambdify import lambdify, implemented_function

```



```
8 from sympy.solvers.inequalities import solve_univariate_inequality
9
10 #Creating the variables for solving with SymPy library
11 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
12 init_printing(use_unicode=True)
13
14 #Setting the relative traffic intensity
15 r=(0.3/(2*0.21))
16
17 #The approximation to the derivative of the geometric function
18 dAx=0.02*(0.98**u)
19
20 #The initial equation could not be solved
21 #so it was degenerated to a form that a solution
22 #could be obtained
23 IAx=b-integrate(exp(-(1)*u/r)*dAx, (u,0,oo))
24 IAx=lambdify(b,IAx)
25 root=optimize.newton(IAx,x0=0.9,maxiter=2000)

1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6 from sympy import factorial
7 import sys
8 sys.setrecursionlimit(1500)
9
10 #The probability of binomial was set equal to p=0.02 and q=1-p=0.98
```

```
11 #by utilising the data from MarineTraffic
12
13 #Creating variables for the SymPy library
14 U,u,b,c,m=symbols('U,u,b,c,m',real=True)
15
16 #Solution obtained for the hta term
17 l=0.8
18 r=(0.3/(2*0.21))
19
20 def factL(z1,z2):
21     return (factorial(z1)/factorial(z2)/factorial(z1-z2))
22
23 #formulating the probabilities of table A
24 def pijA(m,n,b):
25     return (Integral(factL(m,n)*\
26         ((1-exp(-u/b))**n)*exp(-(m-n)*u/b)*0.02*(0.98**u),\
27         (u,0,oo)).doit().evalf())
28
29 #formulating the probabilities of table B
30 def funcsup(m,n,s,b):
31     return (factL(s,n)*(1/factorial(m-1))*((s/b)**m)\
32         *Integral(Integral(exp(-s*U/b)*(U**(m-1))\
33         *((1-exp((U-u)/b))**n)*\
34         exp((s-n)*(U-u)/b),\
35         (U,0,u))*0.02*(0.98**u), (u,0,oo)).doit().evalf())
36
37 #Solving for the k term i.e. k0
38 #Only the first 6 terms are considered
39 sum1=0
```

```

40 for i in range (1,6,1):
41     sum1+=funcsup(i,1,2,0.3)*(1**i)
42 p01=pijA(1,0,0.3)
43 p12=pijA(2,1,0.3)
44 m0=(1-sum1-p12)/p01
45
46 summ=m0
47
48 #Obtain the mean value results and the probabilities
49 w0=(summ+1)/(summ+(1/(1-1)))
50 Q0=(summ+1+1)/(summ+(1/(1-1)))
51 Ew=0.3*(1-w0)/(2*(1-1))
52 EQ=(1-Q0)/(1-1)
53
54 print ('The probability not having to wait is, i.e. w is zero', w0)
55 print ('The probability having zero customers in queue is', Q0)
56 print ('The mean value of customers in true queue is', EQ)
57 print ('The mean value of waiting time is', Ew)

```

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	G.I/M/2
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	0.02
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	0.09
Πιθανότητα να μην υπάρχει χρόνος αναμονής	$P_{W=0}$	0.98
Πιθανότητα να μην υπάρχει ουρά	$P_{N_Q=0}$	0.98

Table 14: Σύγκριση αποτελεσμάτων για μοντέλο M/M/2 και G.I/M/2 με είσοδο Γεωμετρική

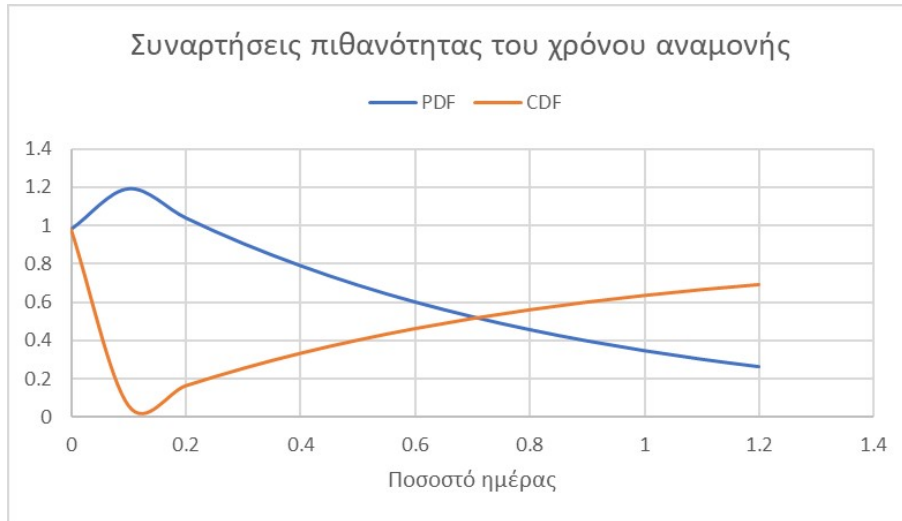


Figure 34: Συναρτήσεις πιθανότητας του χρόνου αναμονής

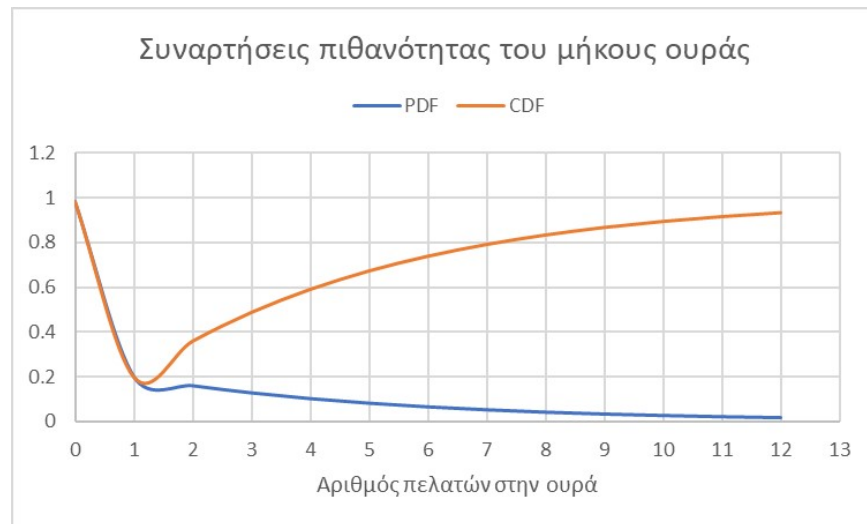


Figure 35: Συναρτήσεις πιθανότητας του μήκους ουράς

Τα αποτελέσματα διαφέρουν σημαντικά και από τις δύο προηγούμενες περιπτώσεις και λαμβάνουν τιμές που φαίνεται να μην συμφωνούν με το ρυθμό συνωστισμού ανά εξυπηρετητή.

Δηλαδή, πιθανότητα 98% για μηδενική ουρά και μέσο μήκος ουράς 0.09 ενώ $\rho = 0.71$ αποτελεί ένδειξη ότι το μοντέλο δεν λειτουργεί σωστά.

5.3 Εφαρμογή του M/G/m μοντέλου αναμονής

Σε αυτά τα μοντέλα οι χρόνοι εξυπηρέτησης λαμβάνουν μια γενική κατανομή G, ενώ οι ενδιάμεσοι χρόνοι εξυπηρέτησης δέχονται Poissonian είσοδο. Στη βιβλιογραφία υπάρχουν προσεγγίσεις για το μέσο χρόνο αναμονής στην ουρά $E[W]$ (εξαιρώντας το χρόνο εξυπηρέτησης), για τη πιθανότητα delay formula του Erlang $P(W>0)$ ή Π ή P_Q καθώς και για τη κατανομή του μήκους ουράς αλλά δίχως αποτελέσματα για τη κατανομή του χρόνου αναμονής. Στο [12] προκύπτει η εξίσωση για το P_Q για οποιοδήποτε αριθμό εξυπηρετητών m, για το μέσο χρόνο αναμονής στην ουρά και για το μέσο αριθμό πελατών στην ουρά. Στο [9] υπάρχει μια γενική λύση για το P_Q καθώς και για το μέσο χρόνο αναμονής στην ουρά, $E[W]$. Επίσης στο [19] προσφέρεται ένας εναλλακτικός τύπος για το μέσο χρόνο αναμονής. Για αυτά τα μοντέλα αναμονής M/G/m θα γίνει χρήση μόνο της γενικευμένης εκθετικής κατανομής (G.E.) αλλά οποιαδήποτε άλλη συνάρτηση που φέρει ισόνομη και ανεξάρτητη τυχαία μεταβλητή δύναται να χρησιμοποιηθεί.

$$\text{CDF of G.E.} = 1 - \gamma \cdot e^{-\gamma \cdot \mu \cdot t}$$

$$\text{PDF of G.E.} = \gamma^2 \cdot \mu \cdot e^{-\gamma \cdot \mu \cdot t}, \text{ όπου } \gamma = \frac{2}{1 + c^2}$$

Το c^2 είναι το Squared Coefficient of Variation (Variance divided by square of the Mean) και η σχέση μεταξύ γ και SCV για την G.E. αποδεικνύεται παρακάτω:

$$\begin{aligned} E[X] &= \int_0^{\infty} \gamma \cdot e^{-\gamma \cdot \mu \cdot t} dt \\ &= \frac{1}{\mu} \cdot \int_0^{\infty} \gamma \cdot \mu \cdot e^{-\gamma \cdot \mu \cdot t} dt \\ &= \frac{1}{\mu} \cdot \left(-\left[\frac{1}{e^{\infty}} - \frac{1}{e^0} \right] \right) \\ &= \frac{1}{\mu} \end{aligned}$$

$$\begin{aligned}
 E[X^2] &= \int_0^{\infty} t^2 \cdot \gamma^2 \cdot \mu \cdot e^{-\gamma \cdot \mu \cdot t} dt, \text{ αλλαγή μεταβλητής } \gamma \cdot \mu \cdot t = u \\
 &= \frac{1}{(\gamma \cdot \mu)^3} \cdot \int_0^{\infty} u^2 \cdot \gamma^2 \cdot \mu \cdot e^{-u} du \\
 &= \frac{\gamma^2 \cdot \mu}{(\gamma \cdot \mu)^3} \cdot \int_0^{\infty} u^2 \cdot e^{-u} du \\
 &= \frac{1}{\gamma \cdot \mu^2} \cdot \Gamma(3) \\
 &= \frac{2}{\gamma \cdot \mu^2}
 \end{aligned}$$

$$\begin{aligned}
 V[X] &= E[X^2] - E^2[X] = \frac{2}{\gamma \cdot \mu^2} - \frac{1}{\mu^2} \\
 &= \frac{2 - \gamma}{\gamma \cdot \mu^2}
 \end{aligned}$$

$$c^2 = \frac{V[X]}{E^2[X]} = \frac{\frac{2 - \gamma}{\gamma \cdot \mu^2}}{\frac{1}{\mu^2}}$$

$$c^2 = \frac{2 - \gamma}{\gamma}$$

Αρχικά γίνεται χρήση της G.E. για να επιβεβαιωθεί η εγκυρότητα της μεθόδου στο [9] με βάση τα δεδομένα από το λιμάνι Yokkaichi. Ακολουθώντας τα δεδομένα του παραπάνω άρθρου για διαφορετικές τιμές της τριπλέτας $(\rho, m, c^2) = (\text{ρυθμός συνωστισμού ανα εξυπηρετητή, αριθμός εξυπηρετητών και SCV})$ προκύπτει ότι για $(0.7, 2, 3)$ η πιθανότητα $\Pi(M/G/s) = 0.48$.

Η μεθοδολογία που ακολουθήθηκε για τη λήψη του αποτελέσματος ήταν να γίνει γραμμική παρεμβολή για το προσδιορισμό της πιθανότητα $\Pi(M/G/s)$ για τη τιμή του $SCV = 3$, για όλες τις διαθέσιμες τιμές εξυπηρετητών $m = 5, 10, 15, 20$ και για σταθερό $\rho = 0.7$.

Προσαρμόζοντας μια συνάρτηση εκθετικής μορφής για τις διάφορες τιμές των εξυπηρετητών $m = 5, 10, 15, 20$ λαμβάνεται και η τιμή για $m = 2$. Παρακάτω φαίνεται ο πίνακας της γραμμικής παρεμβολής και η εν λόγω συνάρτηση προσαρμογής εκθετικής μορφής.

Η διαδικασία για τον υπολογισμό των τελικών αποτελεσμάτων αποτελείται από τα εξής

Γραμμική Παρεμβολή			
$c^2 = 2.5$	$c^2 = 3$	$c^2 = 4$	Εξυπηρετητές
-	0.48	-	2
0.39	0.39	0.39	5
0.23	0.23	0.23	10
0.15	0.15	0.15	15
0.1	0.1	0.1	20

Table 15: Γραμμική Παρεμβολή

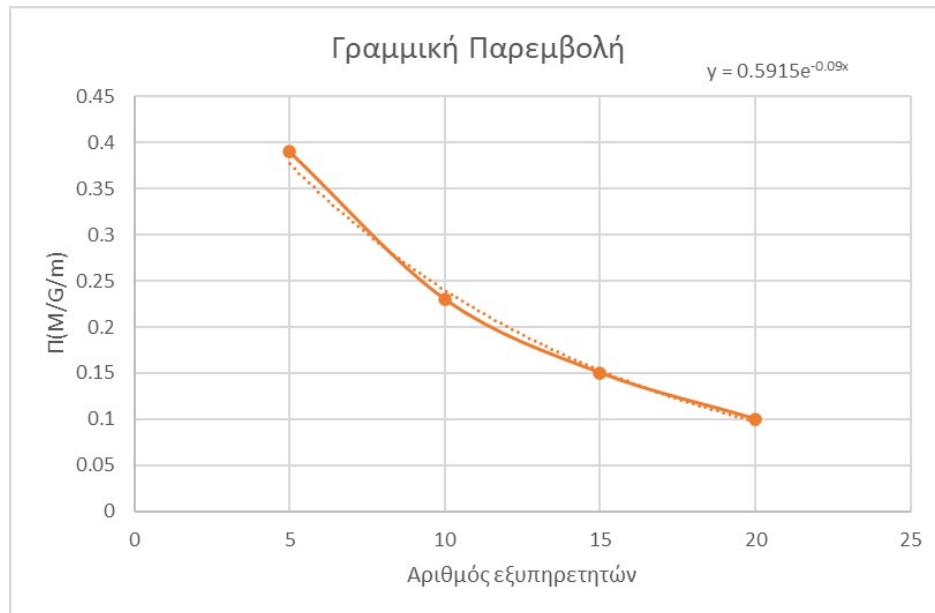


Figure 36: Γραμμική Παρεμβολή

βήματα.

- Προσδιορισμός της ποσότητας γ .
- Προσδιορισμός του ολοκληρώματος $I_{G(m)}$ σύμφωνα με τη συνάρτηση G , που έχει επιλεγεί (εδώ η $G.E.$) ως CDF για τους χρόνους εξυπηρέτησης.
- Προσδιορισμός της ποσότητας R_G^K .

Η ποσότητα γ τίθεται ίση με 0.5 και έτσι $c^2 = 3$.

Από τις τέσσερις προτεινόμενες λύσεις στο [9] επιλέγεται η "Linear Interpolation - L" και η "Harmonic Interpolation - H" διότι είναι πιο κοντά στα πραγματικά δεδομένα για $m \leq 10$ και $c^2 < 2.5$. Επομένως για τη περίπτωση L λαμβάνουμε:

$$P_Q^L = [(1 - \rho) \cdot m \cdot \mu \cdot I_{G(s)} + \rho \cdot \frac{1 + c^2}{2}]^{-1} \cdot R_G^K \cdot \Pi(M/M/m)$$

Ενώ για τη περίπτωση H:

$$P_Q^H = \left[\frac{1 - \rho}{m \cdot \mu \cdot I_{G(s)}} + \frac{2 \cdot \rho}{1 + c^2} \right] \cdot R_G^K \cdot \Pi(M/M/m)$$

Και για τις δύο περιπτώσεις όμως ο μέσος χρόνος αναμονής είναι κοινός και ίσος με:

$$E[W](M/G/m) = \frac{1 - c^2}{\frac{2 \cdot I_{G(s)}}{E[W](M/M/m)} + \frac{1 - J_{G(s)}}{E[W](M/D/m)}}$$

Για το μέσο χρόνο αναμονής για το σύστημα με σταθερούς χρόνους εξυπηρέτησης (regular input) $E[W](M/D/m)$ ο ίδιος ο συγγραφέας εκμεταλλεύεται τη προσέγγιση του Cosmetatos [20] που είναι πολύ ακριβής για $m \leq 10$ και $\rho > 0.6$ [21].

$$E[W](M/D/m) = 0.5 \cdot [1 + C(m, \rho) \cdot E[W](M/M/m)]$$

$$\text{όπου } C(m, \rho) = (1 - \rho) \cdot (1 - m) \cdot \frac{\sqrt{4 + 5 \cdot m} - 2}{16 \cdot m \cdot \rho}$$

Είναι φανερό από τις εξισώσεις ότι είναι προσεγγίσεις αφού δεν περιέχουν άμεσα αποτελέσματα για το μέσο χρόνο αναμονής και τη πιθανότητα να υπάρξει ουρά αλλά εμπεριέχουν τις αντίστοιχες ποσότητες από το σύστημα M/M/m.

Επίσης από το [21] για να ακυρωθεί η αδυναμία του Cosmetatos προτείνεται η παρακάτω σχέση για το μέσο χρόνο αναμονής με σταθερους χρόνους εξυπηρέτησης (regular είσοδο). Όλες οι επιμέρους συναρτήσεις είναι διαθέσιμες στο [21]. Συνεχίζοντας τους υπολογισμούς για την εύρεση των τελικών αποτελεσμάτων γίνεται εμφανές ότι λόγω της G.E. συνάρτησης η ποσότητα $J_{GE(m)}$ είναι ίση με τη μονάδα αφού:

$$\begin{aligned} I_{GE(m)} &= \frac{1 + c^2}{2 \cdot \mu \cdot m} \\ J_{GE(m)} &= \frac{m + 1}{m - 1} \cdot \left(\frac{1 + c^2}{(m + 1) \cdot \mu \cdot \frac{1 + c^2}{2 \cdot \mu \cdot m}} - 1 \right) \\ &= \frac{m + 1}{m - 1} \cdot \frac{2 \cdot m - m - 1}{m + 1} \\ &= 1 \end{aligned}$$

Αυτό έχει ως αποτέλεσμα το κλάσμα που επηρεάζει τη τιμή $E[W](M/D/m)$ να είναι μηδενική και επομένως να μην έχει καμία επίδραση.

$$\frac{1 - J_{GE(m)}}{E[W](M/D/m)} = 0, \text{ όταν } J_{G(m)} = 1$$

Η ποσότητα R_G^K υπολογίζεται με βάση το παρακάτω τύπο του οποίου όλες οι επιμέρους ποσότητες έχουν υπολογιστεί ήδη εκτός από τη ποσότητα R_D^K , η οποία προσδιορίζεται στο [9].

$$R_G^K = \frac{R_D^K \cdot (1 + c^2)}{(2 \cdot R_D^K - 1) \cdot J_{G(m)} + 1}$$

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	M/G.E./2
Μέσος χρόνος αναμονής στην ουρά Kimura $E[W](M/D/m)$	$E[W]$	0.68
Μέσος χρόνος αναμονής στην ουρά Cosmetatos $E[W](M/D/m)$	$E[W]$	0.68
Πιθανότητα να \exists ουρά	P_Q^L	0.66
Πιθανότητα να \exists ουρά 2	P_Q^H	0.66

Table 16: Αποτελέσματα για το μοντέλο αναμονής M/G.E./2

Τελικά, παρατηρείται ότι σε σχέση με το M/M/m σύστημα το M/G/m λαμβάνει τη διπλάσια τιμή στο μέσο χρόνο αναμονής ενώ η πιθανότητα Π είναι σχεδόν ίδια. Οι τιμές των δύο διαφορετικών προσεγγίσεων για τη πιθανότητα $P_Q = \Pi$ είναι επίσης ίδιες λόγω της ειδικής τιμής που λαμβάνει το $J_{G(m)} = 1$.

Για την ολοκλήρωση των τρόπων υπολογισμού των ποσοτικών χαρακτηριστικών των μοντέλων αναμονής M/G/m αντλείται μια προσέγγιση από το [12] για τη $\Pi(M/G/m)$, το μέσο χρόνο αναμονής και το μέσο μήκος ουράς. Η πρόταση του συγγραφέα είναι η εξής:

Ορίζοντας ως G^* το μετασχηματισμό Laplace της εκάστοτε συνάρτησης G και αποδεικνύοντας ότι μέχρι τη πλήρωση του συστήματος με $m - 1$ πελάτες, δηλαδή $k \leq m - 1$, οι εξισώσεις πιθανοτήτων p_i είναι ίσες με τις αντίστοιχες του M/M/m συστήματος, ενώ για $k \geq m$ για

τη πιθανογεννήτρια συνάρτηση $P(z)$ ισχύει η παρακάτω σχέση:

$$\sum_{k=m}^{\infty} p_k \cdot z^{k-m} = \frac{1 - G^*\left(\frac{a \cdot (1-z)}{m}\right)}{G^*\left(-z + \frac{a \cdot (1-z)}{m}\right)}$$

Καταλήγει ότι για $z = 1$ και για $G = M$:

$$P(z = 1) = \frac{\rho}{1 - \rho} \cdot p_{m-1}$$

Που είναι η Erlang Delay Formula και άρα υπάρχει μια γενική εξίσωση για τη πιθανότητα

$$\Pi(M/G/m) = P(z = 1).$$

Συνοψίζοντας τα αποτελέσματα που εμφανίζονται στο [12] και το κώδικα Python που οδήγησε στην εύρεσή τους:

```

1 import numpy as np
2 import math
3 from sympy import *
4 from sympy import solveset
5 from sympy.solvers.solveset import solveset, solveset_real
6
7 v,u,U,a,b,c=symbols('v u U a b c',real=True)
8
9 nom=Integral(3.33*(0.5**2)*exp(-3.33*0.5*u),(u,0,oo)).doit().evalf()
10 denom=Integral(exp(u)*3.33*(0.5**2)*exp(-3.33*0.5*u),(u,0,oo))\
11     .doit().evalf()
12 P=(1-nom)/denom
13 Q=0.71*(9+1)*P/(2*0.3)
14 W=Q/(3.33*2*0.7)
15

```

```

16 print ('The probability that W>0 is', P)
17 print ('The mean customers in queue', Q)
18 print ('The mean waiting time in queue', W)
    
```

Αποτελέσματα		
Χαρακτηριστικό Μέγεθος	Συμβολισμός	M/G.E./2
Μέσος χρόνος αναμονής στην ουρά	$E[W]$	1.01
Μέσος αριθμός πελατών στην ουρά	$E[N_Q]$	4.73
Πιθανότητα να \exists ουρά	P_Q	0.39

Table 17: Αποτελέσματα για το μοντέλο αναμονής M/G.E./2 κατά Hokstad

Πραγματοποιώντας την ίδια μεθοδολογία για την επιβεβαίωση της καταλληλότητας της κατανομής στα δεδομένα, δηλαδή εφαρμόζοντας το χ^2 test εξαντλούνται οι μέθοδοι για εξακρίβωση του βαθμού προσαρμοστικότητας της συνάρτησης στα δεδομένα. Σύμφωνα με τα αποτελέσματα όπως φαίνονται παρακάτω το κανονικοποιημένο τετραγωνισμένο σφάλμα που συγκρίνεται με το όριο δεν είναι αποδεκτό.

$\chi^2 test$			
PDF της G.E.	Συχνότητα της G.E.	Πραγματική Συχνότητα	κανονικοποιημένο τετραγωνισμένο σφάλμα
0.6	31.63	19	8.39
0.51	26.78	31	0.58
0.43	22.67	3	128.95
Όριο	3.814	Άθροισμα	137.91

Table 18: $\chi^2 test$ για το μοντέλο αναμονής M/G.E./2

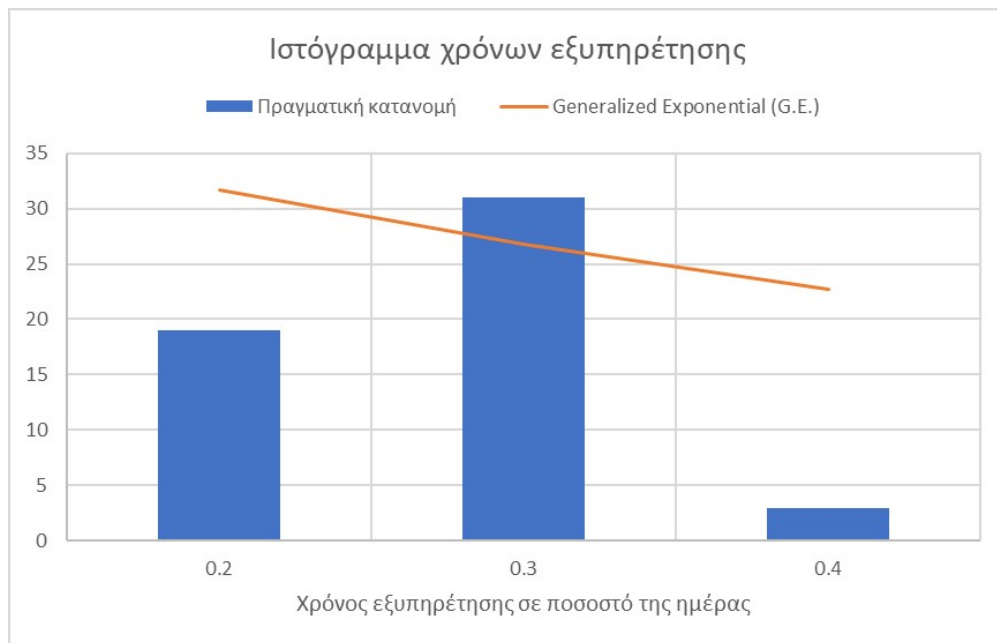


Figure 37: Ιστόγραμμα χρόνων εξυπηρέτησης για το λιμάνι Yokkaichi

6 Σύγκριση μεταξύ δύο διαφορετικών σχημάτων κόστους

Για να μπορέσουν να βρουν εφαρμογή τα ανωτέρω θα πρέπει να διαμορφωθεί ένα κατάλληλο πλαίσιο που να τηρεί τις συνθήκες που απαιτούν τα μοντέλα αναμονής. Το φαινόμενο των διαδοχικών αφίξεων πλοίων σε ένα λιμάνι δεν είναι εν γένει τυχαίο που απαιτούν τα συστήματα αναμονής αλλά σε συγκεκριμένες περιπτώσεις δύναται να γίνει. Μια τέτοια περίπτωση λαμβάνει χώρα όταν εισάγεται η έννοια της αργοπορημένης άφιξης. Στη περίπτωση που η καθορισμένη (συμβατική) διαδρομή που συνήθως είναι και η συντομότερη εξελιχθεί σε ένα δύσκολο διάπλου λόγω υψηλών εντάσεων ανέμων και θαλασσοταραχών, ο πλοίαρχος και ο εκάστοτε αποφασίζων (Decision maker) καλείται να λάβει την απόφαση για το αν θα διασχίσει αυτή τη δύσκολη θάλασσα και τον καιρό της έτσι ώστε να φτάσει εγκαίρως στο προκαθορισμένο σημείο του ναυλοσυμφώνου και τη συμφωνηθείσα ώρα ή να παρακάμψει (ή να περιμένει πλέοντας με χαμηλή ταχύτητα) τη προσχεδιασμένη πορεία, να καθυστερήσει στα όσα έχουν ήδη συμφωνηθεί εκ των προτέρων και να αποφύγει να διακινδυνεύσει και να εκθέσει το πλοίο σε κινδύνους που θα κοστίσουν μεταγενέστερα.

Αλλά υπάρχει και η άλλη όψη του νομίσματος. Δεδομένου ότι θα διασχίσει τα τρικυμία με σκοπό να τα καταφέρει στις υποχρεώσεις του ναυλοσυμφώνου θα προκαλέσει στο πλοίο δομική καταπόνηση, ενδεχόμενες βλάβες στο μηχανολογικό εξοπλισμό, αύξηση στη κατανάλωση καυσίμων κ.α. Ενώ με την εναλλακτική θα αποφύγει τα ανωτέρω και θα έχει, έστω και μια μικρή πιθανότητα, να δεχθεί μειωμένα αντίποινα οικονομικής φύσεως για τη καθυστέρηση αν εκείνη είναι εντός επιτρεπτών ορίων. Για τις δύο αυτές επιλογές όμως τα κόστη που δημιουργούνται δεν φθάνει να συγκριθούν μόνο για ένα δρομολόγιο. Μια πιο ολοκληρωμένη εικόνα θα προκύψει αν η ανάλυση για τα δύο διαφορετικά σχήματα

κόστους γίνει για ολόκληρη της διάρκεια ζωής του πλοίου υπό εξέταση. Για να γίνει όμως η ανάλυση απαιτούνται τα δεδομένα που επηρεάζουν και προκαλούν αυτά τα κόστη καθώς και ποια είναι η βαρύτητα και ο τρόπος που επιδρούν στη διαμόρφωσή τους. Από τα πιο βασικά είναι τα ωκεανογραφικά δεδομένα π.χ. ταχύτητα ανέμου, χρονοσειρές για τις τιμές των ναύλων, χρονοσειρές για το κόστος καυσίμων κ.α.

Όσο αφορά τα ωκεανογραφικά δεδομένα η πιο βασική απαίτηση ήταν η ταχύτητα του ανέμου και πιο συγκεκριμένα η συνολική ταχύτητα του ανέμου ανεξαρτήτως διεύθυνσης και οι επιμέρους συντελεστές του σε καρτεσιανό σύστημα αξόνων επί την επιφάνεια της θάλασσας. Οι ταχύτητες ήταν προσβάσιμες μέσω του ICOADS (International Comprehensive Oceanographic Atmospheric Data Set) του τμήματος PSL της υπηρεσίας NOAA του υπουργείου εμπορίου των Η.Π.Α. Το εν λόγω Data set διατηρεί δεδομένα με γεωγραφική ακρίβεια $1x1$ και $2x2$ ενώ χρονικά φέρει μηνιαίους μέσους όρους. Το ιστορικό φέρει δεδομένα για τους τελευταίους 3 αιώνες, ανανεώνεται μηνιαίως και τα αρχεία διατηρούνται σε μορφή netCDF.

6.1 Απόκτηση και Διαχείριση Ωκεανογραφικών Δεδομένων

Η μοντελοποίηση των θαλάσσιων καταστάσεων με έντονους κυματισμούς (τρικυμίες) και από εδώ και στο εξής Sea States (SS) υλοποιήθηκε με την εφαρμογή του JONSWAP Spectrum μαζί με τα αποτελέσματα που προέκυψαν από το εν λόγω project στο [22]. Η τιμή του ανέμου που χρησιμοποιήθηκε στους υπολογισμούς της δημιουργίας των Sea States ήταν το κέντρο επιφανείας των προαναφερθέντων δεδομένων για τα χρονικά διαστήματα 06/2000 έως 06/2019. Ενώ, για το προσδιορισμό της δεύτερης εισόδου, το μονόμετρο μέγεθος fetch, που απαιτείται, για μπορέσει το spectrum JONSWAP να μοντελοποιήσει το Sea State της εκάστοτε θάλασσας, επινοήθηκε η ακόλουθη μεθοδολογία. Η ποσότητα fetch που ουσιαστικά είναι η απόσταση στη οποία ο άνεμος έχει σταθερή ένταση υπολογίστηκε έχοντας ως γνώμονα τους χάρτες που απεικονίζουν τη κατεύθυνση του ανέμου. Ανάλογα με τη θάλασσα/Ωκεανό και τη γεωγραφική περιοχή των συμβατικών διαδρομών που ακολουθούνται στα πιο συνηθισμένα δρομολόγια που εκτελούν τα πλοία η απόσταση που οπτικά φαίνεται να έχει σταθερής κατεύθυνσης και έντασης άνεμο εξομοιώθηκε και τέθηκε ίση με τη ποσότητα fetch.

Όπως και στα προηγούμενα κεφάλαια η επεξεργασία των δεδομένων έγινε στη γλώσσα προγραμματισμού Python και για το συγκεκριμένο τύπο αρχείου netCDF, με τη βοήθεια της βιβλιοθήκης xarray.

Η εν λόγω βιβλιοθήκη δίνει τη δυνατότητα να δημιουργηθούν χρονοσειρές σε οποιαδήποτε μονάδα χρόνου (μηνιαία, τριμηνιαία, ετήσια αφού πρώτα εκτελεστούν οι σχετικοί υπολογισμοί όπως μέσος όρος κ.τ.λ.) και για οποιαδήποτε γεωγραφική περιοχή του πλανήτη. Παρατίθενται για την εικοσαετία 06/2000 – 06/2019 ο μέσος μηνιαίος όρος της ταχύτητας του ανέμου για

τους ωκεανούς/θάλασσες μη την πιο έντονη εμπορική δραστηριότητα. Μαζί με το χάρτη παρουσιάζεται και η εξέλιξη της έντασης του ανέμου για την ίδια περίοδο ως μέσος όρος για ολόκληρη τη γεωγραφική περιοχή του Ωκεανού/θάλασσας. Τέλος, για την παραγωγή των Sea States που έπονται βασική προϋπόθεση είναι η κατεύθυνση του ανέμου που όπως και τα υπόλοιπα δεδομένα παρατίθεται παρακάτω. Τα ανωτέρα αποτελέσματα συνοδεύει και ο κώδικας που τα παρήγαγε.

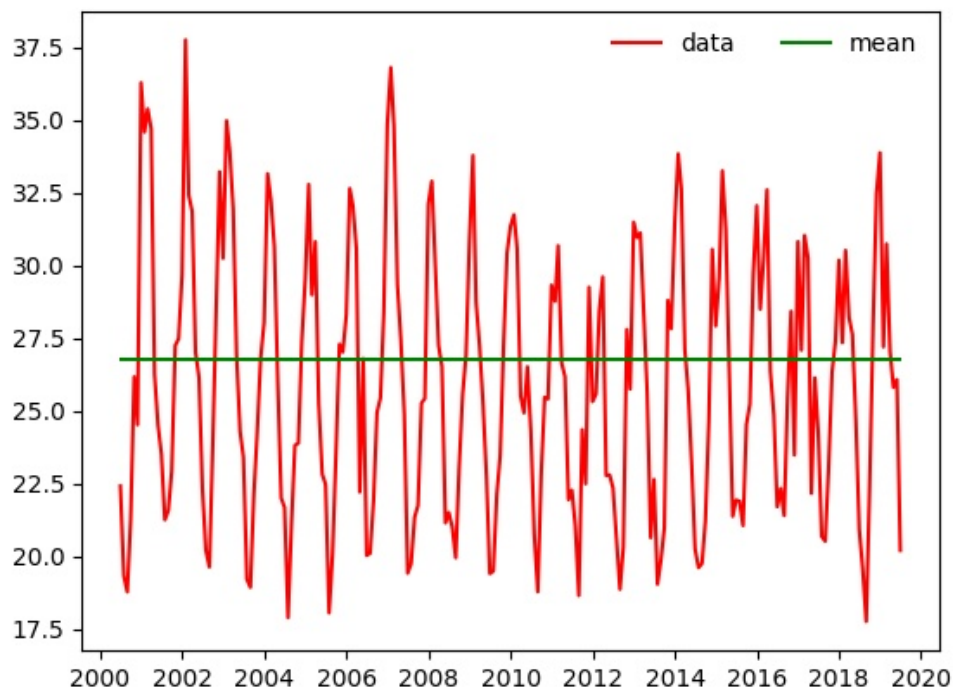


Figure 38: Παράδειγμα χρονοσειράς για Βόρειο Ατλαντικό

```
1 #!/usr/bin/python
2
3 import xarray as xr
4 import matplotlib.pyplot as plt
```

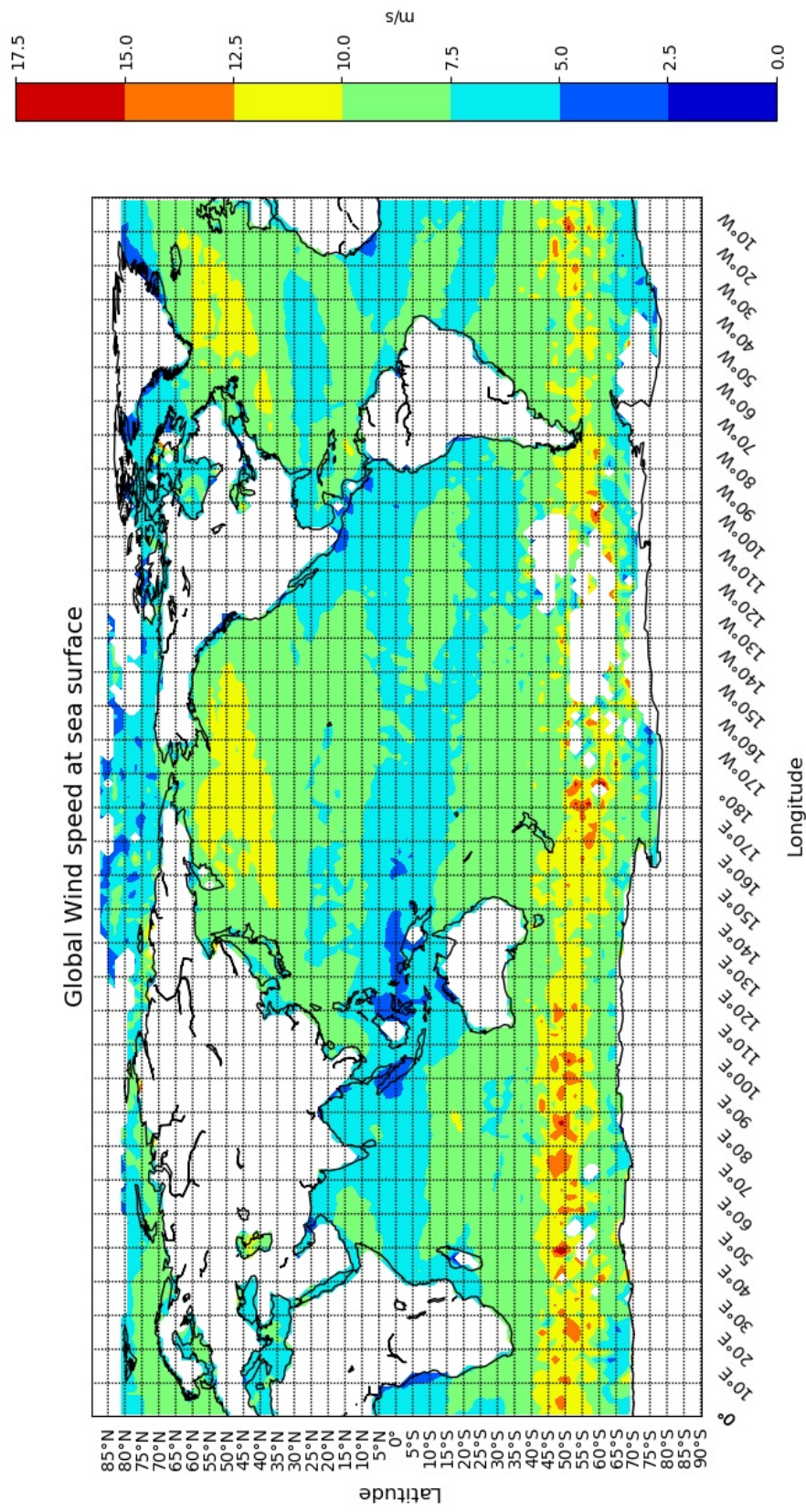


Figure 39: Παγκόσμιο κατανομή ανέμου στην επιφάνεια της θάλασσας

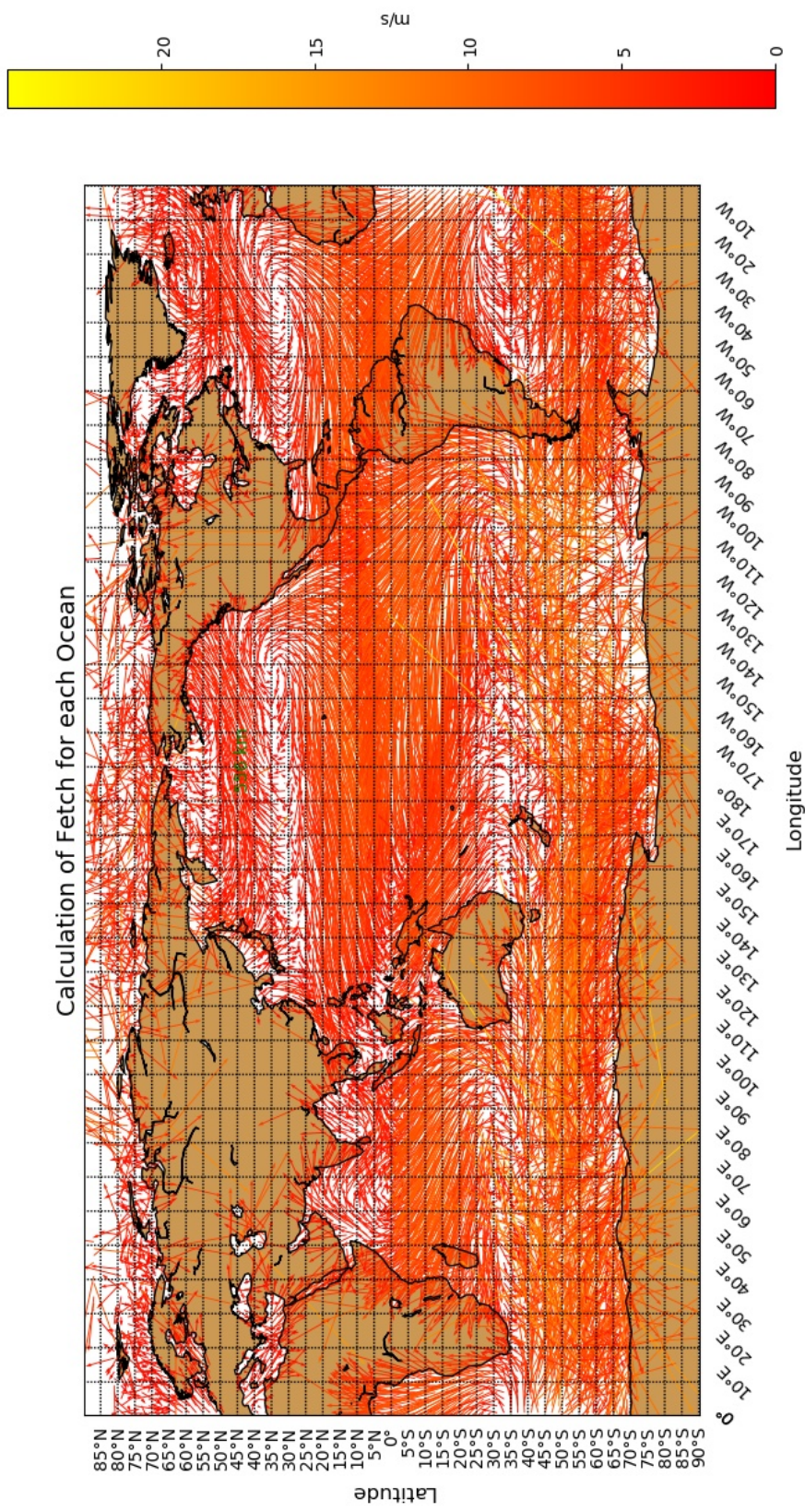


Figure 40: Υπολογισμός fetch για κάθε θάλασσα/Ωκεανό

```
5 import matplotlib
6 import numpy as np
7 import xarray.ufuncs as xu
8 from gdal import *
9 from matplotlib.patches import Polygon
10 from mpl_toolkits.basemap import Basemap
11
12 fh=xr.open_dataset('wspd.mean.nc')
13 wind=fh.wspd
14
15
16 def t_series(lat1,lat2,long1,long2):
17
18     fig, ax=plt.subplots()
19     wind2=(wind*10**0.12)*(3600/1852)
20     windA=wind2.resample(time='M').asfreq().time
21     windA2000_2019=wind2.sel(lat=slice(lat1,lat2)
22                             ,lon=slice(long1,long2))
23     ax.plot(windA,windA2000_2019.mean(['lat','lon']),'r',label='data')
24     windA2000_2019=wind2.sel(time=slice\
25                             ('2000-06','2019-06'),lat=slice(lat1,lat2)
26                             ,lon=slice(long1,long2))
27     bo2=windA2000_2019.mean(['lat','lon'])
28     bo1=xu.square(bo2)
29     acentroid=(bo1.integrate('time')/bo2.integrate('time'))
30     windA=windA.sel(time=slice('2000-06','2019-06'))
31     ll=windA.time.sel(time='2005-06').coords['time'].values
32     ul=windA.time.sel(time='2013-06').coords['time'].values
33     ax.plot(windA,windA2000_2019.mean(['lat','lon']),'r',label='data')
```

```
34     ax.plot(windA,xr.DataArray(np.array([acentroid]*\
35                                     len(windA)),dims=['time'],
36                                     coords=[windA]),'g',label='mean')
37     plt.legend(ncol=2).draw_frame(False)
38     plt.show()
39     print (acentroid)
40
41 #t_series(50,40,310,330) NA
42 #t_series(40,35,190,200) NP
43 #t_series(24,18,117,127) pg indian
44 #t_series(-20,-30,160,170) SP
45 #t_series(60,50,0,5) NS
46 #t_series(35,30,20,30) MED
```

```
1  #!/usr/bin/python
2
3  import xarray as xr
4  import matplotlib.pyplot as plt
5  import matplotlib
6  import numpy as np
7  from gdal import *
8  from mpl_toolkits.basemap import Basemap,shiftgrid
9
10 fh=xr.open_dataset('wspd.mean.nc')
11 wind=fh.wspd
12
13 def wm(lat1,lat2,long1,long2):
14
15     fig=plt.figure()
```

```
16 m=Basemap(projection='cyl',llcrnrlat=lat2,llcrnrlon=long1,
17           urcrnrlat=lat1,urcrnrlon=long2,resolution='c')
18 windA2000_2019=wind.sel(time=slice('2000','2019')\
19                          ,lat=slice(lat1,lat2)
20                          ,lon=slice(long1,long2))\
21                          .groupby('time.year').mean('time')
22 lons=windA2000_2019.coords['lon'].values[:]
23 print (lons)
24 lats=windA2000_2019.coords['lat'].values[:]
25 z=windA2000_2019.mean(dim='year').values[:]
26 print (z)
27 m.drawcoastlines()
28 parallels = np.arange(lat2,lat1,5)
29 m.drawparallels(parallels,labels=[1,0,0,0])
30 meridians = np.arange(long1,long2,10)
31 m.drawmeridians(meridians,labels=[0,0,0,1],rotation=45)
32 plt.title('Global Wind speed at sea surface', fontsize=14)
33 plt.xlabel('Longitude', labelpad=50, fontsize=12)
34 plt.ylabel('Latitude', labelpad=40, fontsize=12)
35 x,y = m(lons,lats)
36 mapy=plt.contourf(x,y,z,cmap='jet')
37 cbar=plt.colorbar(mapy)
38 cbar.ax.set_ylabel(wind.units)
39 plt.show()
40
41
42 #wm(90,-90,0,360) Global
43 #wm(50,40,310,330) NA
44 #wm(40,35,190,200) NP
```

```
45 #wm(24,18,117,127) pg indian
46 #wm(-20,-30,160,170) SP
47 #wm(60,50,0,5) NS
48 #wm(35,30,20,30) MED
```

```
1 #!/usr/bin/python
2
3 import xarray as xr
4 import matplotlib.pyplot as plt
5 import matplotlib
6 import numpy as np
7 import math as math
8 from gcd import gcd
9 from mpl_toolkits.basemap import Basemap
10
11 fh=xr.open_dataset('uwnd.mean.nc')
12 uwind=fh.uwnd
13 fh=xr.open_dataset('vwnd.mean.nc')
14 vwind=fh.vwnd
15
16 def arrows(lat1,lat2,long1,long2,flat1,flon1,flat2,flon2):
17     m=Basemap(projection='cyl',llcrnrlat=lat2,llcrnrlon=long1,
18               urcrnrlat=lat1,urcrnrlon=long2,resolution='c')
19     u=uwind.sel(time=slice('2010','2019'),lat=slice(lat1,lat2)
20               ,lon=slice(long1,long2))\
21           .groupby('time.year').mean('time').squeeze()
22     v=vwind.sel(time=slice('2010','2019'),lat=slice(lat1,lat2)
23               ,lon=slice(long1,long2))\
24           .groupby('time.year').mean('time').squeeze()
```



```
25     u=u.mean(dim='year')
26     v=v.mean(dim='year')
27     u1=u.values
28     v1=v.values
29     parallels = np.arange(lat2,lat1,5)
30     m.drawparallels(parallels,labels=[1,0,0,0])
31     meridians = np.arange(long1,long2,10)
32     m.drawmeridians(meridians,labels=[0,0,0,1],rotation=45)
33     plt.title('Calculation of Fetch for .... Ocean', fontsize=14)
34     plt.xlabel('Longitude', labelpad=50, fontsize=12)
35     plt.ylabel('Latitude', labelpad=40, fontsize=12)
36     lons=u.coords['lon'].values
37     lats=v.coords['lat'].values
38     speed=np.sqrt(u1*u1+v1*v1)
39     x,y = m(lons,lats)
40     m.fillcontinents(color='#cc9955',zorder=0)
41     m.drawcoastlines()
42     mapy=m.quiver(x, y, u1, v1,speed,scale=0.1,units='dots',\
43                 cmap=plt.cm.autumn,linewidth=2)
44     cbar=plt.colorbar(mapy)
45     cbar.ax.set_ylabel(uwind.units)
46     radius = 6371
47     dlat = math.radians(flat2-flat1)
48     dlon = math.radians(flon2-flon1)
49     a = math.sin(dlat/2) * math.sin(dlat/2) + \
50         math.cos(math.radians(lat1)) \
51         * math.cos(math.radians(lat2)) * \
52         math.sin(dlon/2) * math.sin(dlon/2)
53     c = 2 * math.atan2(math.sqrt(a), math.sqrt(1-a))
```

```
54     d = radius * c
55     plt.text(((flon1+flon2)/2)+2,(flat1+flat2)/2, '{}\
56     km'.format(round(d)),color='g')
57     plt.show()
58
59 #arrows(90,-90,0,360,40,170,45,190) Global
```

6.2 Δημιουργία Χαρακτηριστικών των Sea States

Οι θάλασσες/Ωκεανοί που μοντελοποιήθηκαν σύμφωνα με το JOWSWAP spectrum και έχουν παρατεθεί παραπάνω, οριοθετούνται γεωγραφικά από τα παρακάτω ζεύγη συντεταγμένων και ονομαστικά είναι οι εξής:

- Βόρειος Ατλαντικός (North Atlantic) : [(50,40),(310,330)]
- Βόρειος Ειρηνικός (North Pacific) : [(40,35),(190,200)]
- Νότιος Ειρηνικός (South Pacific) : [(-20,-30),(160,170)]
- Μεσόγειος θάλασσα (Mediterranean) : [(35,30),(20,30)]
- Βόρεια θάλασσα (North Sea) : [(60,50),(0,5)]
- Περσικός κόλπος - Ινδικός Ωκεανός (Persian - Indian Ocean) : [(24,18),(117,127)]

Η διαδικασία που ακολουθήθηκε για τη δημιουργία των τυχαίων Sea States από εδώ και στο εξής Random Sea States (RSS) που θα συναντήσει στη διάρκεια της ζωής του ένα πλοίο οδήγησε στη δημιουργία της βάσης πάνω στην οποία θα αξιολογηθεί πως μια ισχυρή θαλασσοταραχή ή ακόμα πιο σημαντικά πως μια ακολουθία από διάπλους σε δυνατές θάλασσες θα επιδράσει στη μείωση της διάρκειας της ζωής του.

Ακριβώς η ίδια διαδικασία μπορεί να ακολουθηθεί αν υπάρχουν πραγματικά δεδομένα από ταξίδια πλοίων για τα δρομολόγια που έκαναν, τις καιρικές συνθήκες που συνάντησαν και το χρόνο της ζωής τους μέχρι τη καταστροφή/ανακύκλωσή τους. Αυτό το φαινόμενο της μείωσης της ηλικίας του πλοίου λόγω της αλληλεπίδρασης με έντονα καιρικά φαινόμενα που συνοδεύονται από έντονους κυματισμούς και γενική καταπόνηση του πλοίου από και

στο εξής θα καλείται τεχνητή γήρανση.

Αφού δημιουργηθεί και το WCSSS (Worst Case Sea State Scenario) το μέτρο σύγκρισης αποτελεί το κέντρο επιφανείας του JONSWAP spectrum του μέσου όρου των RSS, \bar{c} .

Η σχέση η οποία ευριστικά εικάζεται μεταξύ του μέσου όρου και της χειρότερης περίπτωσης WCSSS (Worst Case Sea State Scenario) αποτυπώνεται μαθηματικά ως εξής:

$$\frac{c_E}{\bar{c}} = \frac{\theta}{\frac{y_E}{\bar{y}}}, \text{ όπου } \theta \text{ η παράμετρος τεχνητής γήρανσης} \quad (11)$$

Επίσης η παραπάνω μαθηματική σχέση δίνει άμεσα (εφόσον έχουν προσδιοριστεί οι τιμές $c_E, \bar{c}, \bar{y}, \theta$) την μείωση που θα υποστεί το υπό εξέταση πλοίο μέχρι το πέρας της ζωής του αφού η σχέση 11 είναι στατική και ανεξάρτητη του χρόνου. Αυτό όμως θα ισχύει εφόσον διασχίζει το WCSSS σε όλα τα δρομολόγια που θα εκτελέσει καθ' όλη τη διάρκεια της ζωής του. Η σχέση όμως δε δίνει πληροφορίες ούτε για το αποτέλεσμα σε ένα μόνο δρομολόγιο και ούτε για το ρυθμό.

Για να υπερβεί αυτή τη δυσκολία το μοντέλο θεωρεί ότι η σχέση μεταξύ τη πρώτης στιγμής που το πλοίο θα πλεύσει $\frac{y_E}{\bar{y}} = 1$ και της χρονικής στιγμής που θα έχουν μεσολαβήσει \bar{y} χρόνια είναι γραμμική. Αυτό αποτελεί το σύνδεσμο μεταξύ της σχέσης 11 και της δυναμικότητας που της λείπει και γεφυρώνει το γεγονός του ότι είναι ανεξάρτητη του χρόνου. Αν για το πλοίο υπήρχαν δεδομένα για όλα τα δρομολόγια που θα εκτελούσε τότε η δημιουργία μιας σχέσης $c_E(t)$ και $\theta(t)$ θα ήταν εφικτή και δεν θα ήταν απαραίτητη η παραπάνω σχέση.

Εφόσον δεν υπάρχουν πραγματικά δεδομένα για το προσδιορισμό του θ , από τα τεχνητά δεδομένα προκύπτει ότι το $\frac{C_E}{C}$ λαμβάνει τιμές εντός του ανοικτού διαστήματος (1.7 – 2.3). Άρα μια τιμή $\theta = 0.2$ θα προκαλέσει μείωση της ζωής του πλοίου ύστερα από \bar{y} χρόνια κατά 10% της μέσης ζωής \bar{y} .

Η διαδικασία που ακολουθήθηκε όπως προαναφέρθηκε για τη δημιουργία της βάσης για τους υπολογισμούς ήταν η εξής:

- Δημιουργία του JONSWAP spectrum και υπολογισμός του κέντρου επιφανείας του στον άξονα y για κάθε μια από τις 6 θάλασσες/Ωκεανούς SSB_i με $i = 1, 2, \dots, 6$ (Sea State Basis).
- Δημιουργία RSS_j με $j = 1, 2, \dots, 10$ χρησιμοποιώντας ως συστατικά τις 6 μοντελοποιημένες θάλασσες/Ωκεανούς με τις τιμές ανέμου και fetch σύμφωνα με τη σχέση $RSS_j = \sum_{i=1}^6 RSS_i \cdot a_i$ όπου $\sum a_i = 1$ και ενδεικνύουν το ποσοστό της ζωής του πλοίο υπό εξέταση στην εκάστοτε θάλασσα/Ωκεανό για διάπλους. Υπολογίζονται τα κέντρα επιφανείας c_i καθενός από τα 10 RSS.
- Ανάθεση μιας ηλικίας μεταξύ 15 και 25 ετών για τα πλοία που εκτελούν πλόες σε κάθε ένα από τα RSS_j ανάλογα με το κέντρο επιφανείας c_{i1} που προέκυψε στο προηγούμενο βήμα.
- Ορισμός ενός ανώτατου και κατώτατου ορίου στα $c_i, c_L = c_{min} + 0.1 \cdot (\frac{c_{max} + c_{min}}{2})$ και $c_U = c_{max} - 0.1 \cdot (\frac{c_{max} + c_{min}}{2})$ έτσι ώστε να παραληφθούν ακραίες τιμές. Προσδιορισμός του μέσου όρου των c_i και των y_i που τους αντιστοιχούν, \bar{c} και \bar{y} αντίστοιχα. Δημιουργία ιστογραμμάτων για πριν και μετά την απαλοιφή των ακραίων τιμών.

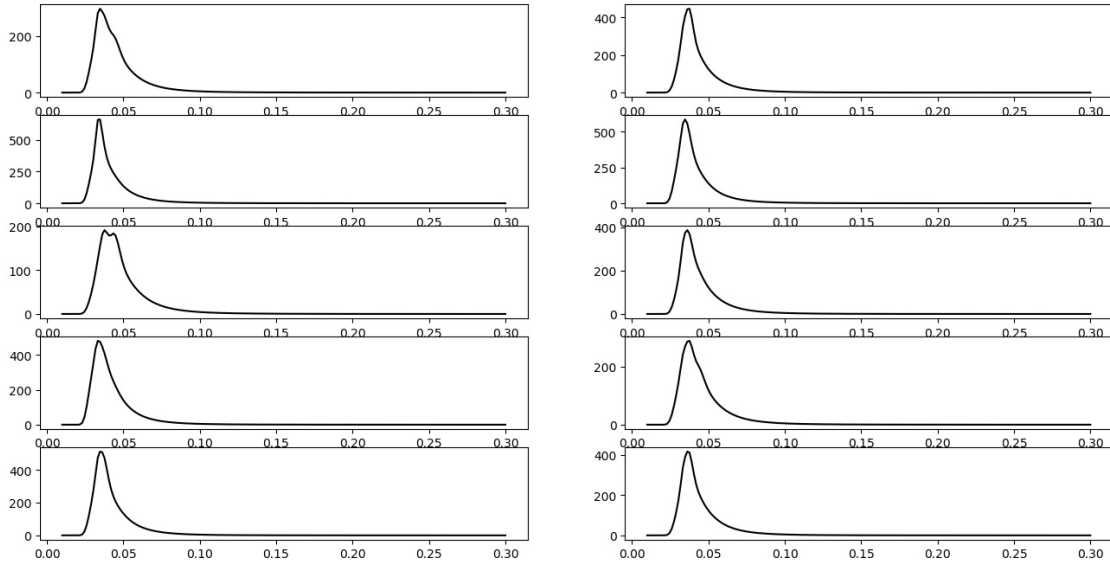


Figure 41: JONSWAP spectrum για 10 RSS

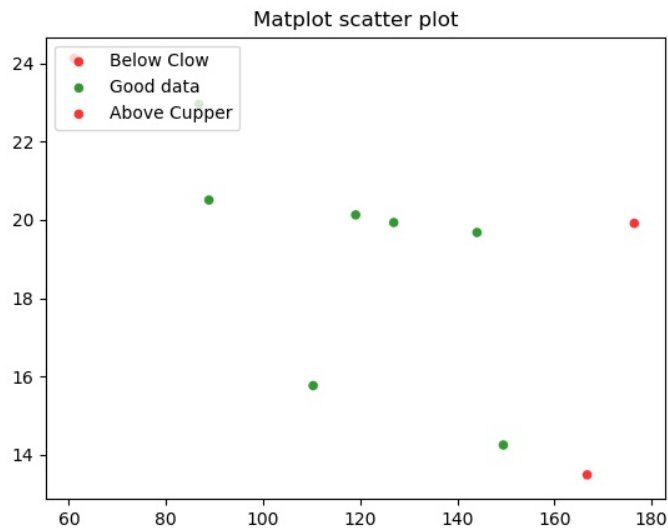


Figure 42: Αφαίρεση ακραίων τιμών

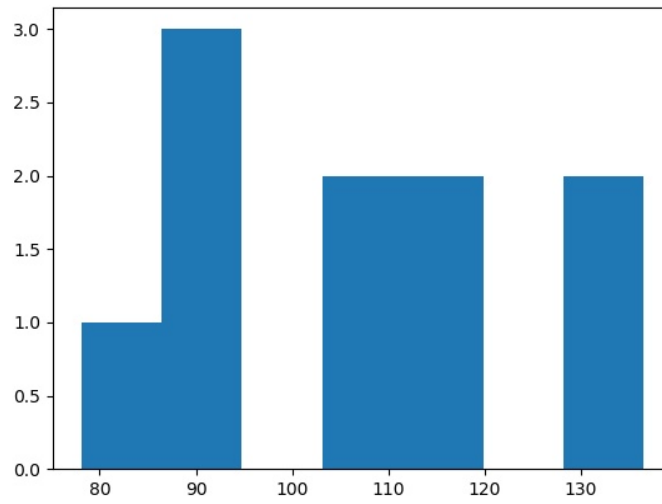


Figure 43: Ιστόγραμμα c_i πριν την αφαίρεση ακραίων τιμών

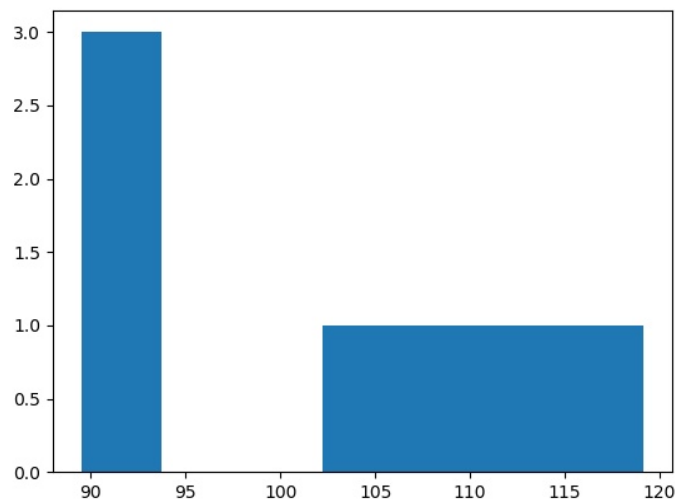


Figure 44: Ιστόγραμμα c_i μετά την αφαίρεση ακραίων τιμών

- Δημιουργία του WCSSS και προσδιορισμός της τιμής c_E . Για τον προσδιορισμό του WCSSS λήφθηκαν υπ' όψη δύο σενάρια:
 - Ελαφρώς αυξημένες τιμές fetch και έντασης ανέμου και χρήση του RSS για το προσδιορισμό του κέντρου επιφανείας ως τελική τιμή c_E .
 - Ελαφρώς αυξημένες τιμές fetch και έντασης ανέμου και χρήση του RSS για το προσδιορισμό της τιμής c_E ίση με την αντίστοιχη τιμή που θα προκύψει από το JONSWAP spectrum με $f = f_c = \frac{1}{T_c}$. Για το προσδιορισμό της τιμής T_c αξιοποιήθηκε το διάγραμμα στο [23] και προσαρμόστηκε ένα νευρωνικό δίκτυο, ο κώδικας του οποίου παρατίθεται παρακάτω.

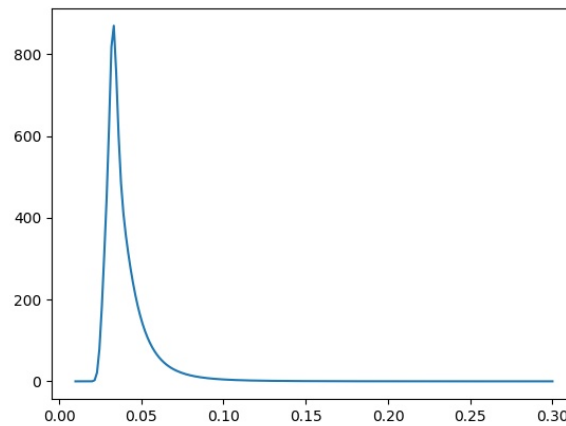


Figure 45: JONSWAP για WCSSS

```
1 #usr/bin/python
2
3 import numpy as np
4 from scipy.stats import truncnorm
5 import matplotlib.pyplot as plt
```



```
6
7 learning_rate=0.5
8
9 t=np.array([[25,1000],[25,1500],[25,2000],[20,1000],[20,1500]\
10             , [20,2000],[17.5,1000],[17.5,1500],[17.5,2000],\
11             [15,1000],[15,1500],[15,2000],[12.5,1000],[12.5,1500]\
12             , [12.5,2000],[10,1000],[10,1500],[10,2000],\
13             [7.5,1000],[7.5,1500],[7.5,2000]])
14
15 Y=np.array([[13.5],[14.5],[15],[11.6],[12.1],[12.5],[10.5]\
16             , [10.8],[10.9],[9.2],[9.4],[9.4],\
17             [7.8],[7.9],[7.9],[6.2],[6.2],[6.2],[4.7],[4.7],[4.7]])
18
19 for i in range (21):
20     t[i][0]=(t[i][0]-7.5)/(25-7.5)
21     t[i][1]=(t[i][1]-1000)/(2000-1000)
22     Y[i]=(Y[i]-4.7)/(15-4.7)
23
24 def sigm(x):
25     return 1/(1+np.e**(-x))
26
27 def truncated_normal(mean,sd,low,upp):
28     return truncnorm((low-mean)/sd,(upp-mean)/sd,loc=mean,scale=sd)
29
30 def initweights():
31
32     no_input_nodes=2
33     no_hidden_nodes=3
34     no_output_nodes=1
```

```
35
36     bias_node=1
37
38     limit1=1/(np.sqrt(no_input_nodes+bias_node))
39     limit2=1/(np.sqrt(no_hidden_nodes+bias_node))
40
41     s1=truncated_normal(2,1,-limit1,limit1)
42     X=s1.rvs(10000)
43     plt.hist(X)
44     plt.show()
45     wih=s1.rvs((no_hidden_nodes,no_input_nodes+bias_node))
46     s2=truncated_normal(2,1,-limit2,limit2)
47     X=s2.rvs(10000)
48     plt.hist(X)
49     plt.show()
50     who=s2.rvs((no_output_nodes,no_hidden_nodes+bias_node))
51
52     return wih,who
53
54 def train(wih,who,t,Y,learning_rate):
55     for j in range (1000):
56         for i in range (21):
57
58             bias_node=1
59             input_vector=np.concatenate((t[i],[bias_node]))
60
61             input_vector=np.array([input_vector]).T
62             target_vector=np.array([Y[i]]).T
63
```

```
64     output_vector1=np.dot(wih,input_vector)
65     output_vector_hidden=sigm(output_vector1)
66
67     output_vector_hidden=np.concatenate\
68         ((output_vector_hidden,[[bias_node]]))
69
70     output_vector2=np.dot(who,output_vector_hidden)
71     output_vector_network=sigm(output_vector2)
72
73     output_error=target_vector-output_vector_network
74     tmp=output_error*output_vector_network\
75         *(1-output_vector_network)
76     tmp=learning_rate*np.dot(tmp,output_vector_hidden.T)
77     who+=tmp
78
79
80     tmp=np.dot(who.T,output_error)
81     tmp=tmp*output_vector_hidden*(1-output_vector_hidden)
82     tmp=tmp[:-1]
83     tmp=learning_rate*np.dot(tmp,input_vector.T)
84     wih+=tmp
85     return wih,who
86
87 def run(wih,who,t,Y,k):
88     Total_error=0
89     for i in range (k):
90         bias_node=1
91         input_vector=np.concatenate((t[i],[bias_node]))
92
```

```
93     input_vector=np.array([input_vector]).T
94     target_vector=np.array([Y[i]]).T
95
96     output_vector1=np.dot(wih,input_vector)
97     output_vector_hidden=sigm(output_vector1)
98
99     output_vector_hidden=np.concatenate\
100         ((output_vector_hidden,[[bias_node]]))
101
102     output_vector2=np.dot(who,output_vector_hidden)
103     output_vector_network=sigm(output_vector2)
104     print (output_vector_network)
105     output_error=(target_vector-output_vector_network)**2
106
107     Total_error+=output_error
108     return Total_error
109
110 wih,who=initweights()
111
112 wih,who=train(wih,who,t,Y,learning_rate)
113
114 Total_error=run(wih,who,t,Y,21)
115
116 print (Total_error/21)
117
118 t1=np.array([[14.,1023],[19.,1500],\
119             [22.,2000],[10.,1300],[15.,1800],[13.,2000],
120             [10.,1100],[17.,1900],[14.,1750]])
121
```

```
122 Y1=np.array([[9.2],[12],[13.5],[6.3],[9.5],[7.9],
123             [6.2],[10.9],[9.4]])
124
125 for i in range(9):
126     t1[i][0]=(t1[i][0]-10)/(22-10)
127     t1[i][1]=(t1[i][1]-1023)/(2000-1023)
128     Y1[i]=(Y1[i]-6.2)/(13.5-6.2)
129
130 Total_error=run(wih,who,t1,Y1,9)
131
132 print (Total_error/9)
```

- Προσδιορισμός της παραμέτρου της τεχνητής γήρανσης θ και λήψη των αποτελεσμάτων στη μορφή ποσοστού γήρανσης ανά ημέρα, ανά έτος καθώς και ανά ταξίδι. Για το

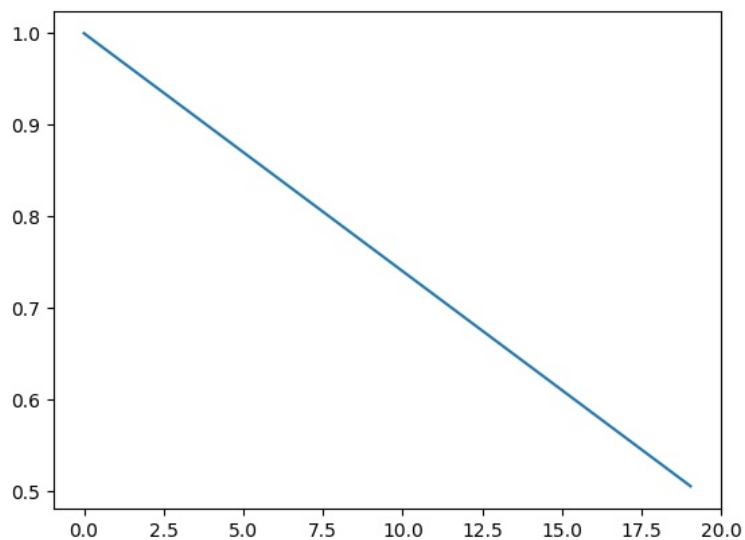


Figure 46: Γραμμική σχέση τεχνητής γήρανσης

δρομολόγιο *Qatar* → *Japan : Yokkaichi* το ποσοστό γήρανσης ανα ταξίδι είναι στο 0.2%, το ετήσιο 2% ενώ το ημερήσιο είναι σχεδόν μηδενικό.

```

1  #!/usr/bin/python
2
3  import numpy as np
4  import math
5  from math import pi
6  import matplotlib.pyplot as plt
7
8  g=9.81
9  lowerbound=0.01
10 upperbound=0.3
11
12 def RSSB(F,u10,g,Nxv):
13     a=0.076*((u10**2)/(F*g))**0.22
14     lowbound=0.01
15     upperbound=0.5
16     f=np.linspace(lowbound,upperbound,num=Nxv)
17     fp=[3.5*(g/u10)*((u10**2)/(g*F))**(0.33)]*len(f)
18     fp=np.asarray(fp)
19     JONfp=3.5*(g/u10)*((u10**2)/(g*F))**(0.33)
20     PMfp=[((4*0.74*((g/u10)**4))/(5*(2*pi)**4))**(1/4)]*len(f)
21     PMfp=np.asarray(PMfp)
22     PMfp2=((4*0.74*((g/u10)**4))/(5*(2*pi)**4))**(1/4)
23     sigma=np.zeros(len(f))
24     EPMfp=((0.0081*(g**2))/(((2*pi)**4)*PMfp2**5))*np.exp\
25         (-5/4)*((PMfp2/PMfp2)**4))
26     EJONfp=((a*(g**2))/(((2*pi)**4)*JONfp**5))*np.exp\

```

```

27         
$$(-(5/4)*((JONfp/JONfp)**4))$$

28     gama=EJONfp/EPmf
29     print (gama)
30     for i in range(len(f)):
31         if f[i]<=fp[i]:
32             sigma[i]=0.07
33         else:
34             sigma[i]=0.09
35     sigma=np.asarray(sigma)
36     r=np.exp(-((f-fp)**2)/(2*(sigma**2)*(fp**2)))
37     JONSWAP=((a*(g**2))/((2*pi)**4)*f**5)*np.exp\
38         
$$(-(5/4)*((fp/f)**4))*gama**r$$

39     PM=((0.0081*(g**2))/((2*pi)**4)*f**5)*np.exp\
40         
$$(-(5/4)*((PMfp/f)**4))$$

41     acentroidy=(1/2)*np.trapz(JONSWAP**2,\
42                             x=f)/np.trapz(JONSWAP,x=f)
43     acentroidx=np.trapz(JONSWAP*f,x=f)/\
44                 np.trapz(JONSWAP,x=f)
45     return JONSWAP
46
47
48
49 def RSSGEN(NGRSS,Nxv,lowbound,upperbound):
50     CRSS=[[ for i in range (2)]
51     zr2=((NGRSS,Nxv))
52     RSS=np.zeros(zr2)
53     f=np.linspace(lowbound,upperbound,num=Nxv)
54     NP=RSSB(1420000,27,g,Nxv)
55     SP=RSSB(1500000,23.84,g,Nxv)

```

```
56 PG=RSSB(1240000,21.73,g,Nxv)
57 NA=RSSB(1200000,20.05,g,Nxv)
58 NS=RSSB(800000,20.4,g,Nxv)
59 M=RSSB(800000,17.61,g,Nxv)
60 perc=np.random.dirichlet(np.ones(6),size=NGRSS)
61 for i in range (NGRSS):
62     RSS[i]=perc[i][0]*NP+perc[i][1]*SP+perc[i][2]*PG\
63         +perc[i][3]*NA+perc[i][4]*NS+perc[i][5]*M
64     acentroidy=(1/2)*np.trapz(RSS[i]**2,x=f)\
65         /np.trapz(RSS[i],x=f)
66     acentroidx=np.trapz(RSS[i]*f,x=f)\
67         /np.trapz(RSS[i],x=f)
68     CRSS[0].insert(0,acentroidx)
69     CRSS[1].insert(1,acentroidy)
70 fig,axs=plt.subplots(5,2)
71 av=[0,0,1,1,2,2,3,3,4,4]
72 bv=[0,1,0,1,0,1,0,1,0,1]
73 for i in range (NGRSS):
74     axs[av[i],bv[i]].plot(f,RSS[i],'k',\
75         label="JONSWAP 20.6 100")
76 plt.show()
77 return CRSS
78
79 CRSS=RSSGEN(10,200,0.01,0.3)
80
81 def YPRSS(NGRSS):
82     years=np.random.random_integers\
83         (15,25,size=(1,NGRSS))-0.01
84     Cmax=np.amax(CRSS,axis=1)
```



```
85     Cmax=Cmax [1]
86     red=0.1*(CRSS [1]/Cmax)*years
87     years -=red
88     return years
89
90 years=YPRSS (10)
91 CRSS=np.append(CRSS ,years ,axis=0)
92 Cmin=np.amin(CRSS ,axis=1)
93
94 def SPoCV():
95     Cmin=np.amin(CRSS ,axis=1)
96     Cmin=Cmin [1]
97     Cmax=np.amax(CRSS ,axis=1)
98     Cmax=Cmax [1]
99     Clow=Cmin+0.1*(Cmax+Cmin)/2
100    Cupper=Cmax-0.1*(Cmax+Cmin)/2
101    GBCL=[[ for i in range (2)]
102    GACU=[[ for i in range (2)]
103    IG=[[ for i in range (2)]
104    for i in range (len(CRSS [1])):
105        if CRSS [1] [i] <=Clow:
106            GBCL [0] .insert (0,CRSS [1] [i])
107            GBCL [1] .insert (1,CRSS [2] [i])
108        elif CRSS [1] [i] >=Cupper:
109            GACU [0] .insert (0,CRSS [1] [i])
110            GACU [1] .insert (1,CRSS [2] [i])
111        else:
112            IG [0] .insert (0,CRSS [1] [i])
113            IG [1] .insert (1,CRSS [2] [i])
```

```
114     Ymean=np.mean(IG[1])
115     Cmean=np.mean(IG[0])
116     data=(GBCL,IG,GACU)
117     colors=("red","green","red")
118     groups=("Below Clow","Good data","Above Cupper")
119     fig,ax=plt.subplots()
120     for data,colors,group in zip(data,colors,groups):
121         x,y=data
122         ax.scatter(x, y, alpha=0.8, c=colors,\
123                 edgecolors='none', s=30, label=group)
124     plt.title('Matplot scatter plot')
125     plt.legend(loc=2)
126     plt.show()
127     return IG,Cmin,Cmax,Cmean,Ymean
128
129 IG,Cmin,Cmax,Cmean,Ymean=SPoCV()
130
131 def HoCV():
132     fig,ax=plt.subplots()
133     plt.hist(CRSS[1],bins=7)
134     plt.show()
135     plt.hist(IG[0],bins=7)
136     plt.show()
137
138 HoCV()
139
140 def YCR(Nxv):
141     f=np.linspace(lowerbound,upperbound,num=Nxv)
142     WCSSS=RSSB(1540000,25.73,g,Nxv)
```

```
143 Ce=acentroidy=(1/2)*np.trapz(WCSSS**\
144         2,x=f)/np.trapz(WCSSS,x=f)
145 Cerange=np.linspace(Cmin/10,10*Cmax,num=100)
146 Y=1/(Cerange/Cmean)
147 fig,ax=plt.subplots()
148 ax.plot(f,WCSSS)
149 plt.show()
150 ax.plot(Cerange,Y)
151 ax.plot((Ce/Cmean),(1/(Ce/Cmean)),marker='x')
152 plt.show()
153 return Ce
154
155 Ce=YCR(200)
156
157 def YCR(Nxv):
158     Tc=9.5
159     wc=2*pi/Tc
160     WCSSS=RSSB(fetch,wind@10,lowbound,upperbound,Nxv)
161     Ce=WCSSS
162     Cerange=np.linspace(Cmin/10,10*Cmax,num=100)
163     Y=1/(Cerange/Cmean)
164
165 def DAAR(VD):
166     vita=1
167     lamnta=((1/(Ce/Cmean))-1)/(Ymean-0)
168     f=np.linspace(0,Ymean,10)
169     Y=lamnta*f+vita
170     AAA=lamnta
171     DAA=lamnta/365
```

```
172     print ("The cost will increase by", DAA*VD,\
173           "as a function of time")
174     fig,ax=plt.subplots()
175     ax.plot(f,Y)
176     plt.show()
```

6.3 Ανάλυση σχήματος κόστους - 1η επιλογή του Αποφασίζων

Στη περίπτωση της επιλογής του διάπλου από μια έντονη θαλασσοταραχή και κυματισμούς θα προκύψουν τα παρακάτω κόστη:

- Αυξημένα κόστη καυσίμων
- Αυξημένα κόστη συντήρησης
- Μειωμένα έσοδα λόγω λιγότερων δρομολογίων

Για κάθε ένα από αυτά δημιουργήθηκε ένας κώδικας σε Python που υπολογίζει:

- Για το πρώτο κόστος, την ημερήσια επιπρόσθετη κατανάλωση ανάλογα με το τύπο (BC, Tanker, Containership, LNG Carrier) και το μέγεθος ('000 DWT).
- Για το δεύτερο, το επιπρόσθετο κόστος συντήρησης λόγω της τεχνητής γήρανσης. Ουσιαστικά προκύπτει από τη διαφορά μεταξύ του παρόντος κόστους και του κόστους που θα είχε το πλοίο μετά το πέρας του χρονικού διαστήματος της τεχνητής γήρανσης. Η μαθηματική σχέση που διέπει το κόστος συντήρησης συναρτήσει του χρόνου είναι πολυώνυμο δευτέρου βαθμού.
- Για το τρίτο, τα έσοδα που θα χαθούν λόγω της μικρότερης διάρκειας ζωής του πλοίου αφού θα εκτελέσει λιγότερα δρομολόγια. Θεωρείται ότι το πλοίο είναι ναυλωμένο στο 65% του χρόνου ζωής του.

6.3.1 Αυξημένη Κατανάλωση Καυσίμων

Το γεγονός της αύξησης της κατανάλωσης καυσίμων οφείλεται στο ότι το πλοίο για να διατηρήσει την ίδια ταχύτητα που έχει όταν δεν είναι αντιμέτωπο με τον καιρό θα πρέπει η κύρια μηχανή να μεταφέρει περισσότερη ενέργεια στη προπέλα. Αυτό συμβαίνει διότι η ενέργεια των κυμάτων και του ανέμου εμποδίζουν το πλοίο να διατηρήσει σταθερή πορεία και πέρα από τους ελιγμούς που καλείται να κάνει ο πλοίαρχος δυσχεραίνεται η ομαλή λειτουργία της προπέλας η οποία με τη σειρά της επιζητά μεγαλύτερα ποσά ενέργειας από τη μηχανή για να διατηρήσει σταθερή τη ταχύτητα του πλοίου. Ο μαθηματικός μηχανισμός στηρίζεται στη παρακάτω σχέση [24]:

$$P = c_1 \cdot N^x, \text{ όπου } 2 \leq x \leq 4 \text{ εξαρτάται από το τύπο του πλοίου}$$

$$V = c_2 \cdot N$$

και c_1 και c_2 σταθερές που εξαρτώνται από το τύπο του πλοίου,

το μέγεθος και άλλα χαρακτηριστικά του πλοίου.

$$(PVN) = (\text{ισχύς, ταχύτητα πλοίου, αριθμός στροφών προπέλας})$$

και στην υπόθεση ότι το πλοίο για να διατηρήσει τον ίδιο αριθμό στροφών στη προπέλα θα πρέπει η μηχανή να προσφέρει μεγαλύτερη ισχύ. Το πόσο μεγαλύτερη προφανώς εξαρτάται από τη μηχανή αλλά και από το μέγεθος της πτώσης των στροφών της μηχανής που με τη σειρά του εξαρτάται από την ένταση της κατάστασης της θάλασσας και του ανέμου. Η πτώση των στροφών εδώ θεωρείται σταθερή και ίση με 10%. Παρατίθεται κομμάτι του κώδικα (λόγω μεγάλης έκτασης) και το διάγραμμα με τις επιπρόσθετες ημερήσιες καταναλώσεις ανά τύπο και μέγεθος πλοίου.

```
1  #!/usr/bin/python
2
3  from scipy import optimize
4  import matplotlib.pyplot as plt
5  import matplotlib
6  import numpy as np
7  import random
8
9  def fit1(x,a1,a2,a3):
10     return a1+a2*x+a3*x**2
11
12  ShipType=['Bulk Carrier','Tanker','Container','LPNG Carrier']
13  BCsize=['Handymax','Panamax','Capesize','VLBC']
14  Tsize=['Handymax','Panamax','Aframax','Suezmax']
15  Csize=['Panamax3.5','Post_Panamax5','Post_Panamax10','New_Panamax14']
16  LPNGsize=['Large Conventional']
17
18  s=[]
19  for i in range (100):
20     ss=random.choice(ShipType)
21     s[i].insert(0,ss)
22     if ss==ShipType[0]:
23         ss=random.choice(BCsize)    #shipsize
24         s[i].insert(1,ss)
25         if ss==BCsize[0]:
26             Pm=8400
27             Vm=14.5
28             Vchr=Pm/Vm**3.5
29             Nmax=127
```

```
30     Pmax=9960
31     Vmax=(Pmax/Vchr)**(1/3.5)
32     cV2N=Vmax/Nmax
33     Nm=((Pm/Vchr)**(1/3.5))/(cV2N)
34     c=Pm/Nm**3.5
35     Ps=(100/110)*Pm
36     Ns=(Ps/c)**(1/3.5)
37     Nlr=Ns*1.1
38     Vlr=Nlr*cV2N
39     Vclr=Pm/Vlr**3.5
40     Clr=Pm/(Nlr)**3.5
41     Plr=Clr*Nm**3.5
42     Ph=(Clr/c)*Ps
43     SeaMargin=100*((Ps/Ph)-1)
44     draught=11.6
45     breadth=30.4
46     Lengthbp=177
47     dwt=45000
48     pvgraphlr=np.array([np.linspace(Vlr-10,Vlr+10,num=200)\
49         ,Vclr*(np.linspace(Vlr-10,Vlr+10,num=200))**3.5])
50     pvgraphhr=np.array([np.linspace(Vm-10,Vm+10,num=200)\
51         ,Vchr*(np.linspace(Vm-10,Vm+10,num=200))**3.5])
52     pnggraphlr=np.array([np.linspace(Nlr-60,Nlr+10,num=200)\
53         ,Clr*(np.linspace(Nlr-60,Nlr+20,num=200))**3.5])
54     pnggraphhr=np.array([np.linspace(Nm-60,Nm+10,num=200),\
55         c*(np.linspace(Nm-60,Nm+10,num=200))**3.5])
56     BSFCgraph=np.array([[10,20,30,40,50,60,70,80,90,100],\
57         [191.1,178.1,173.1,170.7,168.5,\
58         166.9,166.1,166.6,168.1,170.5]])
```



```

59     print (BSFCgraph)
60     params,params_covariance=optimize.curve_fit\
61     (fit1,BSFCgraph[0],BSFCgraph[1])
62     plt.figure
63     plt.plot(BSFCgraph[0],fit1(BSFCgraph[0],\
64         params[0],params[1],params[2]))
65     hrcon=fit1(100*(Ps/Pmax),params[0],params[1],params[2])
66     lrcon=fit1(100*(Ph/Pmax),params[0],params[1],params[2])
67     extrafuelperday=((hrcon*(Ps/Pmax)*Pmax-lrcon*\
68         (Ph/Pmax)*Pmax)/(10**6))*24
69     s[i].insert(2,draught), s[i].insert(3,breadth),\
70         s[i].insert(4,Lengthbp),
71     s[i].insert(5,dwt), s[i].insert(6,extrafuelperday),\
72         s[i].insert(7,[Pm,Vm,Nm])
73     print (s)

```

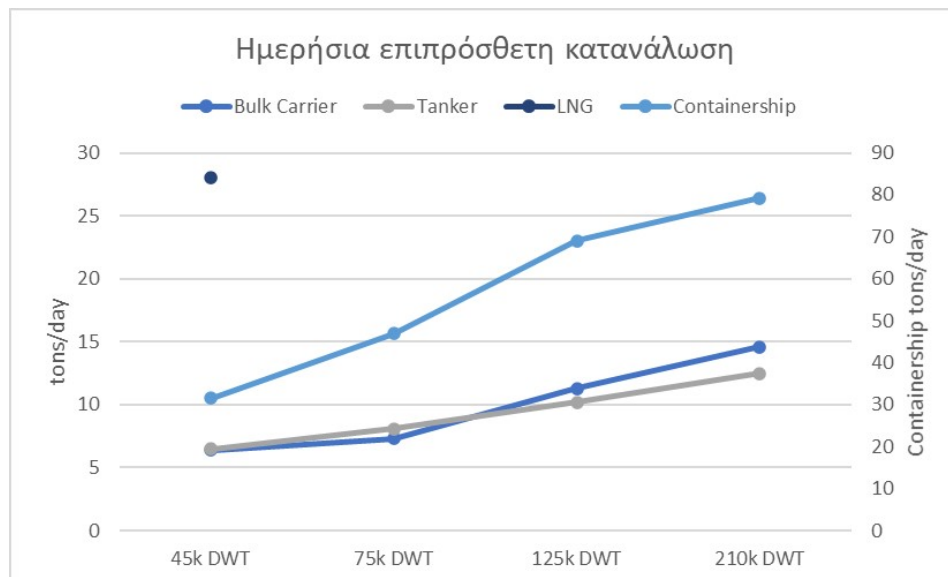


Figure 47: Ημερήσια επιπρόσθετη κατανάλωση

Το κόστος που προκύπτει εξαρτάται και από τη τιμή του καυσίμου που ποικίλει χρονικά και ανά περιοχή. Για το παράδειγμα του λιμένος Yokkaichi στη Ιαπωνία επιλέχθηκε η τιμή του καυσίμου IFO 380 στο λιμάνι της Σιγκαπούρης αφού πρώτα υπολογίστηκε ο μέσος όρος για τους τελευταίους δύο μήνες. Για 15 ταξίδια ετησίως και 2 ημέρες εντός του καιρού προκύπτει ότι το ετήσιο κόστος είναι $2000 \cdot 15 \cdot 2 = 60.000\$$. Ανανεώνοντας το παραπάνω διάγραμμα με ημερήσιο κόστος αντί για ημερήσια κατανάλωση λαμβάνουμε το παρακάτω.

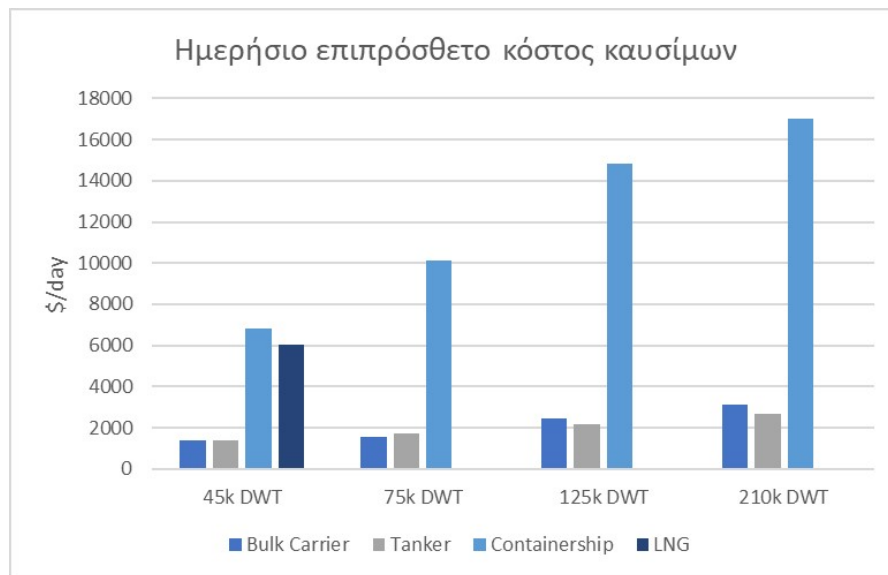


Figure 48: Ημερήσιο επιπρόσθετο κόστος καυσίμων

6.3.2 Αυξημένα Έξοδα Συντήρησης

Τα κόστη συντήρησης ενός πλοίου τείνουν να αυξάνονται όσο μεγαλώνει σε ηλικία. Από τα δεδομένα για Bulk Carrier [25] σε βάθος χρόνου 20ετίας προσαρμόστηκε πολυώνυμο δευτέρου βαθμού που παρουσιάζει το κόστος ανά έτος. Λόγω της τεχνητής γήρανσης στην

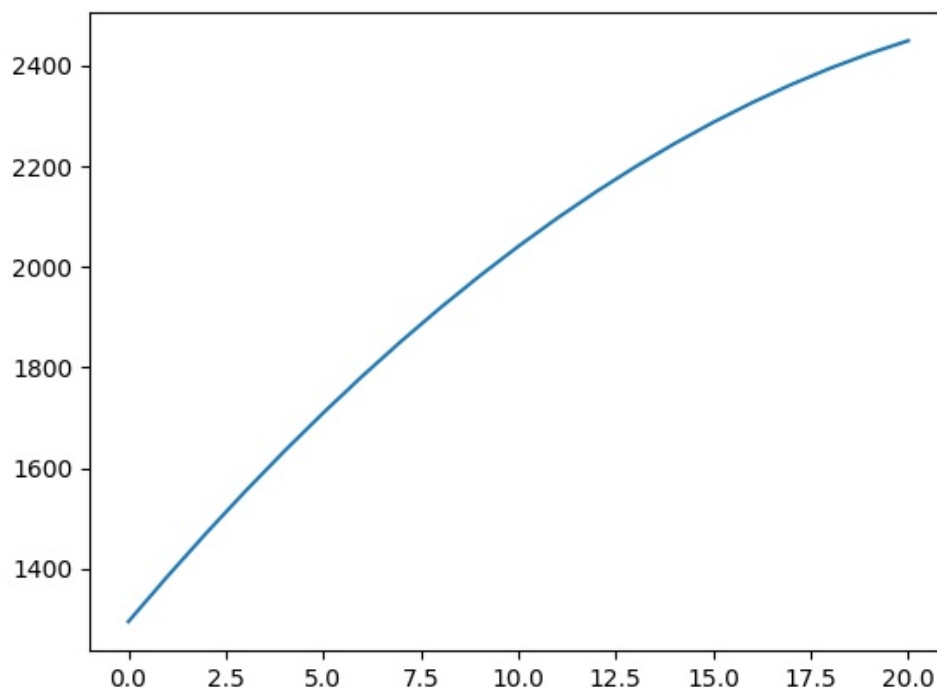


Figure 49: Ετήσιο κόστος συντήρησης

οποία υπόκεινται το πλοίο όμως τα κόστη μετατοπίζονται σε προγενέστερα χρονικά σημεία και άρα αυξάνεται το ετήσιο κόστος όπως φαίνεται στο διάγραμμα παρακάτω: Λόγω της καμπύλης προσαρμογής που ήταν πολυώνυμο δευτέρου βαθμού το επιπρόσθετο κόστος τείνει να μειώνεται όσο τείνουμε στο τέλος της ζωής του πλοίου. Να σημειώσουμε ότι η δραστηριοποίηση του πλοίου θεωρείται ότι κυμαίνεται στο 65% της ζωής του. Ο μέσος

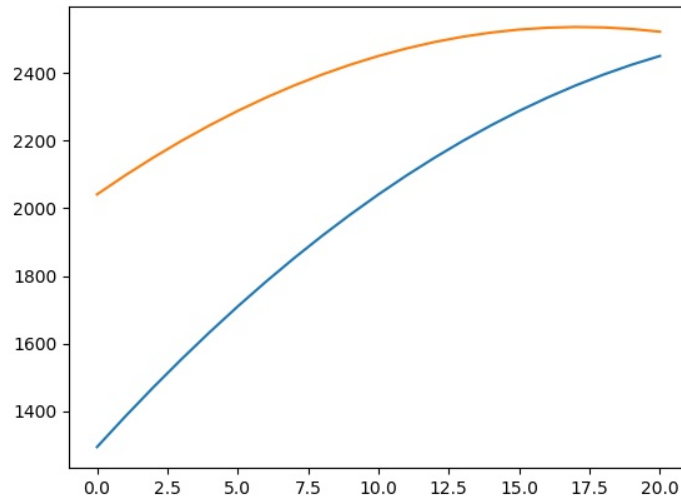


Figure 50: Ετήσιο κόστος συντήρησης 2

ετήσιο όρος είναι 550.000\$. Ο κώδικας και τα αποτελέσματα παρατίθενται παρακάτω.

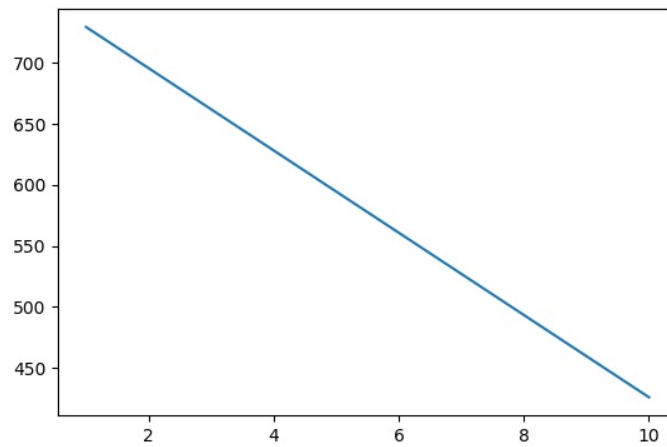


Figure 51: Ετήσιο επιπρόσθετο κόστος συντήρησης

```
1 #!/usr/bin/python  
2
```

```
3 from scipy import optimize
4 import matplotlib.pyplot as plt
5 import matplotlib
6 import numpy as np
7 import random
8
9 def func(x,a1,a2,a3):
10     return a1+a2*x+a3*x**2
11
12 xxvalues=np.linspace(0,10,10)
13 def func2(x,a1,a2,a3):
14     return a1+a2*x+a3*x**2
15
16 xvalues=np.array([[5,10,20],[1710,2041,2450]])
17 params,params_covariance=optimize.curve_fit(func,xvalues[0],xvalues[1])
18 plt.figure
19 newxvalues=np.array([-15,-14,-13,-12,-11,-10,-9,-8,-7,-6\
20                     ,-5,-4,-3,-2,-1,0\
21                     ,1,2,3,4,5,6,7,8,9,10,11,12,13,\
22                     14,15,16,17,18,19,20])
23 newxvalues=np.array([0\
24                     ,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,\
25                     16,17,18,19,20])
26 plt.plot(newxvalues,func(newxvalues,params[0],params[1],params[2]))
27 plt.show()
28 plt.plot(newxvalues,func(newxvalues,params[0],params[1],params[2]),\
29         newxvalues,func(newxvalues,params[0]+10*params[1]\
30                         +100*params[2],\
31                         params[1]+20*params[2],params[2]))
```

```
32 plt.show()
33
34 xforcost=np.linspace(0,10,100)
35 extra_cost=np.trapz(-func(xforcost,params[0],params[1],params[2])+
36                    func(xforcost,params[0]+10*params[1]+100*params[2],
37                          params[1]+20*params[2],params[2]),x=xforcost)
38 print (extra_cost)
39
40 extra_cost1=np.zeros(10)
41 x1=[1,2,3,4,5,6,7,8,9,10]
42 for i in range (10):
43     xforcost=np.linspace(i,i+1,100)
44     extra_cost1[i]=np.trapz(-func(xforcost,params[0],params[1]\
45                               ,params[2])+
46                             func(xforcost,params[0]+10*params[1]+100\
47                                   *params[2],\
48                                   params[1]+20*params[2],params[2]),x=xforcost)
49 plt.figure
50 plt.plot(x1,extra_cost1)
51 plt.show()
52 print ("T           :",extra_cost1,"           ",\
53        np.sum(extra_cost1))
54
55 xvalues=np.array([[5,10,20],[4685,5591,6712]])
56 params,params_covariance=optimize.curve_fit(func,xvalues[0],xvalues[1])
57 plt.figure
58 newxvalues=np.array([-15,-14,-13,-12,-11,-10,-9,-8,-7,-6\
59                    ,-5,-4,-3,-2,-1,0\
60                    ,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,\
```

```
61         17,18,19,20])
62 plt.plot(newxvalues,func(newxvalues,params[0],params[1],params[2]),\
63         newxvalues,func(newxvalues,params[0]+10*params[1]\
64         +100*params[2],\
65         params[1]+20*params[2],params[2]))
66 plt.show()
67
68 xforcost=np.linspace(0,10,100)
69 extra_cost=np.trapz(-func(xforcost,params[0],params[1],params[2])+\  
70         func(xforcost,params[0]+10*params[1]+100*params[2],\  
71         params[1]+20*params[2],params[2]),x=xforcost)
72 print (extra_cost*365)
```

6.3.3 Μειωμένα Έσοδα λόγω Τεχνητής Γήρανσης

Αυτό το στοιχείο κόστους όπως και το προηγούμενο οφείλεται στο φαινόμενο της τεχνητής γήρανσης. Λαμβάνοντας υπόψη τις τιμές των ναύλων για ένα πλοίο Bulk Carrier μεγέθους Aframax [26] για τη δεκαετία 2005 – 2014 μπορεί να υπάρξει μια εκτίμηση για τη μείωση των εσόδων αφού το πλοίο δεν δραστηριοποιείται και απώλειες εσόδων για το ισοδύναμο χρονικό διάστημα. Σύμφωνα με τους υπολογισμούς από το παρακάτω κώδικα και τα δεδομένα από το διάγραμμα, το εκάστοτε πλοίο θα είχε απώλειες εσόδων ίσες με τις τιμές που φαίνονται στο παρακάτω πίνακα ανά έτος. Είναι σημαντικό να σημειωθεί ότι υπάρχει μεγάλη διακύμανση στα μειωμένα έσοδα ανά έτος άρα λαμβάνεται ο μέσος ετήσιος όρος.

```
1  #!/usr/bin/python
2
3  from scipy import optimize
4  import matplotlib.pyplot as plt
5  import matplotlib
6  import numpy as np
7  import random
8
9  yvalues=np.array([40,   36,   47,   32,\
10                  38,   22,   22,   28,\
11                  57,   65,   62,   53,\
12                  40,   35,   18,   30,\
13                  28,   41,   38,   48,\
14                  38,   45,   50,   45,\
15                  45,   38,   39,   28,\
16                  28,   18,   23,   23,\
17                  28,   40,   60,   75,\
18                  60,   45,   35,   40,\
```



```
19      50,    65,    62,    62,\
20      55,    45,    36,    53,\
21      40,    30,    20,    23,\
22      18,    15,    15,    9,\
23      9,     15,    20,    30,\
24      23,    25,    20,    30,\
25      20,    15,    10,    10,\
26      15,    28,    22,    18,\
27      10,    10,    19,    15,\
28      9,     13,    15,    10,\
29      14,    14,    17,    20,\
30      20,    18,    16,    16,\
31      18,    18,    15,    15,\
32      17,    17,    20,    18,\
33      16,    16,    16,    16,\
34      16,    17,    19,    18,\
35      18,    20,    25,    35,\
36      43,    30,    18,    17,\
37      17,    18,    19,    24,\
38      30,    21,    17,    32])
39 yvalues=30*yvalues
40 xvalues=np.linspace(5,15,120)
41 plt.figure
42 plt.plot(xvalues,yvalues)
43 plt.show()
44 xvalues=np.linspace(0,120,120)
45 rev_lost=np.trapz(yvalues,x=xvalues)
46 print (0.65*rev_lost,"for 10 years")
47
```

```
48 x=xvalues[:60]
49 y=yvalues[:60]
50 rev_lost=np.trapz(y,x=x)
51 print (0.65*rev_lost,"for the first 5 years")
52
53 x=xvalues[-60:]
54 y=yvalues[-60:]
55 rev_lost=np.trapz(y,x=x)
56 print (0.65*rev_lost,"for the last 5 years")
57
58 for i in range (10):
59     x=xvalues [12*i:12*(i+1)]
60     y=yvalues [12*i:12*(i+1)]
61     rev_lost=np.trapz(y,x=x)
62     print (0.65*rev_lost,"for the",i+1, "year")
```

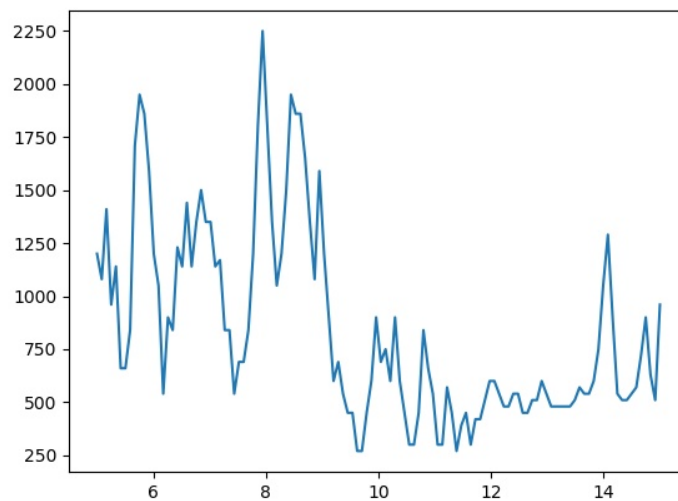


Figure 52: Μηνιαίες τιμές Ναύλων

Αποτελέσματα	
Έτος	Τιμή
2005	9M\$
2006	8.1M\$
2007	7.5M\$
2008	11M\$
2009	4.1M\$
2010	4.2M\$
2011	3M\$
2012	3.7M\$
2013	4M\$
2014	4.8M\$

Table 19: Μειωμένα έσοδα λόγω μείωσης ζωής πλοίου

6.4 Ανάλυση σχήματος κόστους - 2η επιλογή του Αποφασίζων

Στη περίπτωση της επιλογής της παράκαμψης ή πλεύσης με χαμηλή ταχύτητα θα προκύψουν τα παρακάτω κόστη:

- Αυξημένη κατανάλωση D.O. για τα βοηθητικά συστήματα όσο αναμένει το πλοίο στην ουρά για εξυπηρέτηση.
- Μείωση του ναύλου λόγω αργοπορημένης εκφόρτωσης και καθυστέρησης στην παράδοση του φορτίου.
- Εφόσον το πλοίο δεν προλάβει το προγραμματισμένο διάπλου σε κανάλι ή διώρυγα διπλοπληρώνει για τη διέλευση. Το ίδιο ισχύει και για τα έξοδα ελλιμενισμού.
- Μειωμένα έσοδα λόγω μη δραστηριοποίησης εφόσον πραγματοποιήσει παράκαμψη από το δρομολόγιο ή/και έπλευσε με χαμηλή ταχύτητα με αποτέλεσμα να αυξήσει το χρονικό διάστημα του δρομολογίου και επιπλέον χρόνος αναμονής στην ουρά από αυτόν που αρχικά είχε εκτιμηθεί.

Για κάθε ένα από τα παραπάνω δημιουργήθηκε κώδικας σε Python που εκτελεί τους αναγκαίους υπολογισμούς για τη λήψη του αποτελέσματος.

6.4.1 Αυξημένη κατανάλωση των Βοηθητικών Συστημάτων

Το πρώτο στοιχείο κόστους για την απόφαση της παράκαμψης είναι η αύξηση της κατανάλωσης Diesel Oil (D.O.) στα βοηθητικά συστήματα λόγω αναμονής στο λιμάνι. Το αποτέλεσμα προκύπτει πολύ εύκολα και δεν απαιτούνται πολλές ενέργειες. Το κόστος ισούται με το γινόμενο των αριθμών των ταξιδιών που το πλοίο θα καθυστερήσει ανά έτος, τις ημέρες (το χρονικό διάστημα του χρόνου αναμονής) που θα καθυστερήσει σε κάθε αργοπορημένο ταξίδι, την ημερήσια κατανάλωση και τη τιμή του καυσίμου ανά τόνο. Ο αριθμός των πλοίων για το δρομολόγιο *Qatar* → *Japan* : *Yokkaichi* που εκτελείται για ένα έτος αδιάκοπα είναι 15 με σταθερή ταχύτητα 20 κόμβων. Ο χρόνος αναμονής ποικίλει ανάλογα με το μοντέλο που θα χρησιμοποιηθεί. Το M/G/m του Hokstad το αποτέλεσμα είναι 1.01 ημέρες. Η ημερήσια κατανάλωση για βοηθητικά συστήματα κυμαίνεται στον 1 τόνο για τα περισσότερα πλοία ενώ η τιμή του καυσίμου που χρησιμοποιείται βρίσκεται στο επίπεδο των $400 \frac{\$}{ton}$ για το λιμάνι της Σιγκαπούρης. Άρα έτσι λαμβάνεται το εξής αποτέλεσμα:

$$\text{Ετήσιο κόστος} = 15 \cdot 1 \cdot 1 \cdot 400 = 6000\$$$

6.4.2 Μείωση Ναύλου

Είναι σύνηθες στη Ναυτιλία όταν το πλοίο δεν τηρεί τους όρους του ναυλοσυμφώνου να δέχεται κυρώσεις, από τις οποίες εκείνες που είναι οικονομικής φύσεως επηρεάζουν σημαντικά τα έσοδα. Έτσι ο ναύλος μειώνεται όταν υπάρχουν καθυστερήσεις και ο τρόπος που προσεγγίστηκε στη παρούσα εργασία είναι ότι μειώνεται ραγδαία και μάλιστα λογαριθμικά. Το παρακάτω διάγραμμα εκφράζει το ποσοστό μείωσης του ναύλου συναρτήσει του χρόνου καθυστέρησης σε ημέρες. Αξιοποιώντας τα ίδια δεδομένα αλλά και τρόπο προσέγγισης

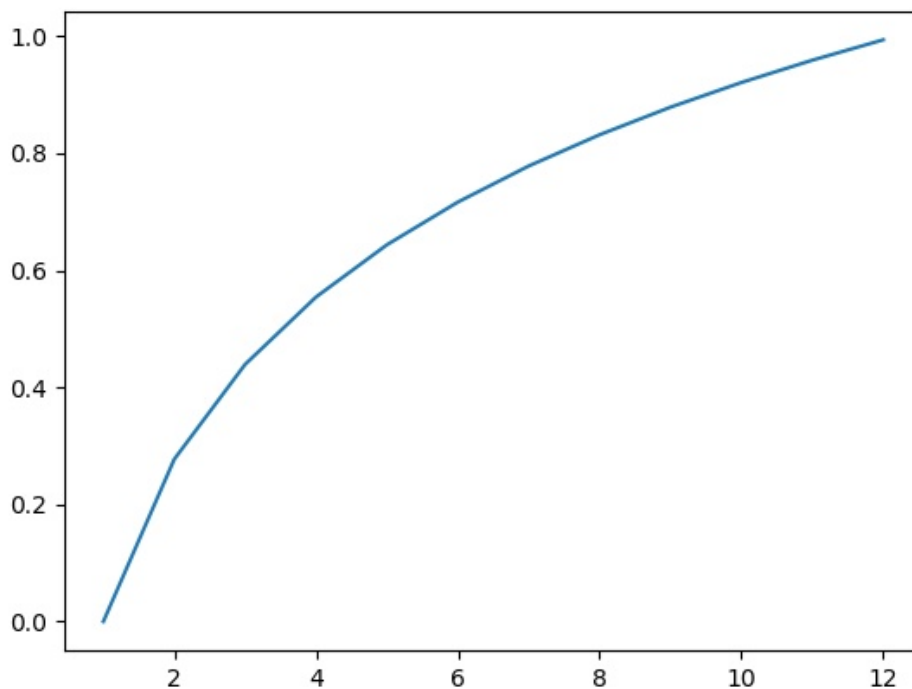


Figure 53: Ποσοστιαία μείωση ναύλου

όπως και στην υποενότητα 6.3.3, δηλαδή μειώνοντας τα έσοδα που προκύπτουν από τον μηνιαίο ναύλο σε ετήσια βάση, προέκυψαν τα παρακάτω αποτελέσματα:

Αποτελέσματα	
Έτος	Τιμή
2005	3.8M\$
2006	3.5M\$
2007	3.2M\$
2008	4.6M\$
2009	1.8M\$
2010	1.8M\$
2011	1.3M\$
2012	1.6M\$
2013	1.7M\$
2014	2.1M\$

Table 20: Μειωμένα έσοδα από μείωση ναύλου

6.4.3 Επιπρόσθετα κόστη ελλιμενισμού

Για το δρομολόγιο *Qatar* → *Japan* : *Yokkaichi* το πλοίο δεν διασχίζει κάποιο κανάλι άρα δεν θα υπάρξει η πληρωμή για δεύτερη φορά σε αυτή τη περίπτωση. Όμως η κλήση λιμένα για δεύτερη φορά λόγω της καθυστέρησης στο δρομολόγιο θα χρεωθεί για δεύτερη φορά. Οι υπολογισμοί στηρίζονται στο [27] και [28] αφορούν το λιμάνι του Rotterdam για πλοία LNG,LPG. Παρ' όλα αυτά δεν παύει να είναι μια ρεαλιστική προσέγγιση για τα έξοδα που θα πληρώνονταν για δεύτερη φορά στο λιμάνι Yokkaichi.

```
1 #!usr/bin/python
2
3 a=0.319*150000
4 b=150000*1.337*0.52
5
6 print ("The additional cost incurred due to seond port call\
7         for the ship is",a,"whereas for the cargo of 150k tons is",\
8         b,"Thus in total:",a+b)
9
10 print ("For the 15 voyages the additioanal cost is:",15*(a+b))
```

Άρα για τα 15 ταξίδια που πραγματοποιούνται ετησίως το κόστος ανέρχεται σε 2.228.040\$.

6.4.4 Μειωμένα Έσοδα λόγω Αδράνειας

Για το τελικό κόστος των μειωμένων εσόδων λόγω της μη δραστηριοποίησης δίνεται για ακόμη μια φορά κάνοντας χρήση του δεδομένου από [26]. Ο τρόπος αντιμετώπισης του προβλήματος του υπολογισμού του κόστους είναι πάλι ο ίδιος με 6.3.3 και 6.4.2. Έτσι αθροίζοντας για κάθε έτος τα έσοδα που χάνονται λόγω του χρόνου αναμονής για όλα τα ταξίδια $T_{E[W]} = 15 * 1.01 = 15$ ημέρες προκύπτει ο παρακάτω πίνακας για κάθε έτος.

Αποτελέσματα	
Έτος	Τιμή
2005	566.000\$
2006	514.000\$
2007	478.000\$
2008	685.000\$
2009	259.000\$
2010	267.000\$
2011	187.000\$
2012	234.000\$
2013	256.000\$
2014	308.000\$

Table 21: Ετήσια μειωμένα έσοδα

7 Οικονομική Ανάλυση της Εμπορικής Ζωής του Πλοίου

Για να ολοκληρωθεί η ανάλυση σε αυτή την ενότητα πραγματοποιείται μια οικονομική σύγκριση μεταξύ των δύο διαφορετικών σχημάτων κόστους με βάση δύο οικονομικών δεικτών, τη Καθαρή Παρούσα Αξία (ΚΠΑ) και τον Εσωτερικό Βαθμό Απόδοσης (ΕΒΑ). Το μέτρο σύγκρισης δεν περιλαμβάνει παράγοντες όπως την επικινδυνότητα της ανθρώπινης ζωής ή τη ρύπανση του περιβάλλοντος που είναι μεγέθη που δύσκολως μετατρέπονται σε οικονομικούς όρους και για αυτό το λόγο σπανίως θέτονται ως βάση για σύγκριση. Έτσι στις επόμενες δύο υποενότητες παρατίθενται συνοπτικά και σε μορφή πινάκων με σύντομη επεξήγηση τα αποτελέσματα που προκύπτουν.

7.1 Οικονομική Σύγκριση με βάση το δείκτη ΚΠΑ

Το κομμάτι στο οποίο αξίζει να δοθεί έμφαση πριν παρουσιαστούν οι υπολογισμοί και τα αποτελέσματα είναι η αξία που φέρει το επιτόκιο αναγωγής που θα χρησιμοποιηθεί. Για κάθε επιμέρους στοιχείου κόστους για κάθε ένα από τα δύο σχήματα κόστους πρέπει να αποδοθεί ένα επιτόκιο αναγωγής. Όσο μεγαλύτερο επιτόκιο αποδίδεται στο κόστος που θεωρείται για λόγους σύγκρισης θετικό τόσο περισσότερο ευνοείται. Για το παράδειγμα που παρατίθεται τα θετικά κόστη θεωρούνται όλα τα επιμέρους κόστη του σχήματος κόστους 1. Ενώ, όσο μεγαλύτερο επιτόκιο αποδίδεται στο κόστος που θεωρείται αρνητικό τόσο πιο σκληρά αντιμετωπίζεται το κόστος που θεωρείται θετικό. Τέλος, αποδίδοντας μεγάλο επιτόκιο σε στοιχείο κόστους αποτελεί ένδειξη ότι το κόστος θα παραμείνει σταθερό και δεν θα χάσει την αξία του διαχρονικά.

Σχήμα Κόστους	Κόστος	επιτόκιο αναγωγής	ΚΠΑ	Έτος				
				1	2	3	4	5
1	Κόστος 1	5.00%	259768.6	60000	60000	60000	60000	60000
	Κόστος 2	6.00%	2316800	550000	550000	550000	550000	550000
	Κόστος 3	10.00%	22744721	6000000	6000000	6000000	6000000	6000000
2	Κόστος 1	1.00%	29120.59	6000	6000	6000	6000	6000
	Κόστος 2	10.00%	11372360	3000000	3000000	3000000	3000000	3000000
	Κόστος 3	2.00%	9426919	2000000	2000000	2000000	2000000	2000000
	Κόστος 4	10.00%	1516315	400000	400000	400000	400000	400000
Άθροισμα			2976575					

Figure 54: Οικονομική Σύγκριση με δείκτη NPV

7.2 Οικονομική Σύγκριση με βάση το δείκτη EBA

Για τη παρούσα σύγκριση αυτό που έχει ενδιαφέρον είναι το ότι ο δείκτης EBA είναι δισδιάστατος. Αυτό σημαίνει ότι κάθε σχήμα κόστους λαμβάνει ένα ενιαίο επιτόκιο αναγωγής για όλα τα επιμέρους στοιχεία κόστους. Με γραμμική παρλινδρόμηση από τα ανωτέρω

Σχήμα Κόστους	Κόστος	επιτόκιο αναγωγής	ΚΠΑ	Έτος				
				1	2	3	4	5
1	Κόστος 1	10.00%	227447.2	60000	60000	60000	60000	60000
	Κόστος 2	10.00%	2084933	550000	550000	550000	550000	550000
	Κόστος 3	10.00%	22744721	6000000	6000000	6000000	6000000	6000000
2	Κόστος 1	5.00%	25976.86	6000	6000	6000	6000	6000
	Κόστος 2	5.00%	12988430	3000000	3000000	3000000	3000000	3000000
	Κόστος 3	5.00%	8658953	2000000	2000000	2000000	2000000	2000000
	Κόστος 4	5.00%	1731791	400000	400000	400000	400000	400000
Άθροισμα			1651950					

Figure 55: Οικονομική Σύγκριση με δείκτη IRR NPV>0

Σχήμα Κόστους	Κόστος	επιτόκιο αναγωγής	ΚΠΑ	Έτος				
				1	2	3	4	5
1	Κόστος 1	20.00%	179436.7	60000	60000	60000	60000	60000
	Κόστος 2	20.00%	1644837	550000	550000	550000	550000	550000
	Κόστος 3	20.00%	17943673	6000000	6000000	6000000	6000000	6000000
2	Κόστος 1	5.00%	25976.86	6000	6000	6000	6000	6000
	Κόστος 2	5.00%	12988430	3000000	3000000	3000000	3000000	3000000
	Κόστος 3	5.00%	8658953	2000000	2000000	2000000	2000000	2000000
	Κόστος 4	5.00%	1731791	400000	400000	400000	400000	400000
Άθροισμα			-3637205					

Figure 56: Οικονομική Σύγκριση με δείκτη IRR NPV<0

προκύπτει = (12.5, 5)

7.3 Αξιολόγηση Αποτελεσμάτων & Συμπεράσματα

Από όσα συμπεραίνονται από τα αποτελέσματα, η επιλογή του σχήματος κόστους 2, δηλαδή η παράκαμψη είναι καλύτερη. Τα μοντέλα που έχουν χρησιμοποιηθεί έχουν επιβεβαιωθεί σε ένα βαθμό με πραγματικά δεδομένα και φαίνεται ότι δύναται να αξιοποιηθούν για προβλέψεις. Από τα αποτελέσματα συμπεραίνεται ότι στο δίλλημα που έχει τεθεί είναι δύσκολο να δοθεί λύση εφόσον οι αριθμητικές τιμές των οικονομικών δεικτών που προκύπτουν ενδεικνύουν τη μικρή υπεροχή της επιλογής 2. Όπως όμως ήδη έχει αναφερθεί τα αποτελέσματα είναι μόνο οικονομικής φύσεως και δεν λαμβάνουν υπ' όψη άλλες διαστάσεις και πτυχές της Ναυτιλίας. Αρκούν όμως για να προσφέρουν μια γενική εικόνα και μια πληθώρα πληροφοριών στο άτομο που θα λάβει την απόφαση. Ο οικονομικός αντίκτυπος της απόφασης σε βάθος χρόνου φαίνεται να είναι ισχυρός και να μην φθίνει, σύμφωνα με τα αποτελέσματα. Αυτό δίνει μια αίσθηση ότι η έρευνα στη πτυχή της Ναυτιλίας που μελετήθηκε θα παραμένει ενεργή αν δεν εντατικοποιηθεί. Μια μελλοντική προτροπή είναι η αποτύπωση των μετρικών των μοντέλων αναμονής όπως πιθανότητα να υπάρχει ουρά και μέσος χρόνος αναμονής υπό τη μορφή χάρτη. Ένα σημαντικό πόρισμα που προέκυψε αλλά δεν αναφέρεται έντονα στη βιβλιογραφία εκτός από το [11] είναι ότι τα μοντέλα αναμονής στηρίζονται στο γεγονός ότι τα συμβάντα δηλαδή οι αφίξεις συμβαίνουν τυχαία και άρα αποτελούν ανεξάρτητα και ισοπίθανα ενδεχόμενα. Κάτι τέτοιο όμως υπό κανονικές συνθήκες δεν ισχύει εφόσον οι περισσότερες ημερομηνίες φορτοεκφορτώσεων καθορίζονται ημέρες πριν την άφιξη του πλοίου. Άρα τα μοντέλα δεν μπορούν να χρησιμοποιηθούν. Όμως στη περίπτωση που εξετάζεται στη παρούσα εργασία το ενδεχόμενο να υπάρχει καθυστέρηση που οφείλεται σε καιρικές συνθήκες θεωρείται ανεξάρτητο και ισοπίθανο με άλλα παρόμοιά του.

Βιβλιογραφία

- [1] Liu Dong Xu Jingjing. “Queuing Models to Improve Port Terminal Handling Service”. In: *Systems Engineering Procedia* 4 (2012), pp. 345–351.
- [2] Odd I. Larsenb Naima Saeeda. “Application of queuing methodology to analyze congestion: A case study of the Manila International Container Terminal, Philippines”. In: *Case Studies on Transport Policy* 4 (2016), pp. 143–149.
- [3] J. D. Griffiths. “Queueing at the Suez Canal”. In: *Journal of the Operational Research Society* 46:11 (1995), pp. 1299–1309.
- [4] Dimitris fakinos. *queuing theory*. symmetria, 2008.
- [5] Dimitris fakinos. *Introduction to probabilities and stochastic processes*. symmetria, 2012.
- [6] United Nations. “United Nations Conference on Trade And Development”. In: *Review of Maritime transport 2017*. 2017, pp. 61–81.
- [7] David G. Kendall. “Some Problems in the Theory of Queues”. In: *Journal of the Royal Statistical Society* 13.2 (1951), pp. 151–185.
- [8] Arne Jensen E. Brockmeyer H. L. Halstrøm. *The Life and Works of A.K. Erlang*. Copenhagen Telephone Company, 1948.
- [9] Kimura T. “Approximations for the Delay Probability in the M/G/s Queue”. In: *Mathl. Comput. Modelling* 22.10-12 (1995), pp. 157–165.
- [10] WARD WHITT. “APPROXIMATIONS FOR THE GI/G/m QUEUE”. In: *PRODUCTION AND OPERATIONS MANAGEMENT SOCIETY* 2.2 (1993).

- [11] David G. Kendall. “Stochastic processes occurring in the theory of queues and their analysis by the method of the embedded markov chain”. In: *The Annals of Mathematical Statistics* (1952).
- [12] PER HOKSTAD. “Approximations for the M/G/m Queue”. In: *OPERATIONS RESEARCH* 26.3 (1978).
- [13] Wikipedia contributors. *Kernel (statistics)*. 2020. URL: [https://en.wikipedia.org/w/index.php?title=Kernel_\(statistics\)%5C&oldid=963115754](https://en.wikipedia.org/w/index.php?title=Kernel_(statistics)%5C&oldid=963115754).
- [14] Wikipedia contributors. *Kernel density estimation*. 2020. URL: https://en.wikipedia.org/w/index.php?title=Kernel_density_estimation%5C&oldid=966082330.
- [15] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [16] MAN Diesel & Turbo. *Propulsion Trends in LNG Carriers*.
- [17] M. Veeraraghavan. *M/M/1 and M/M/m Queueing Systems*. Mar. 2004. URL: <http://www.ece.virginia.edu/~mv/edu/715/lectures/QT.pdf>.
- [18] W.L.SMITH. “On the distribution of queuing times”. In: *Cambridge University Press* 49.3 (2008).
- [19] TOSHIKAZU KIMURA. “Approximating the Mean Waiting Time in the GI/G/s Queue”. In: *Operational Research Society* 42.11 (1991), pp. 959–970.
- [20] Toshikazu Kimura. “A Two-Moment Approximation for the Mean Waiting Time in the GI/G/s Queue”. In: *MANAGEMENT SCIENCE* 42.6 (1986).
- [21] TOSHIKAZU KIMURA. “Refining Cosmetatos’ Approximation for the Mean Waiting Time in the MIDis Queue”. In: *Operational Research Society* 42.7 (1991).

- [22] K. Hasselmann. “Measurements of Wind-Wave Growth and Swell Decay during the Joint North Sea Wave Project (JONSWAP)”. In: *ANE German Bight* (1973).
- [23] WORLD METEOROLOGICAL ORGANIZATION. *GUIDE TO WAVE ANALYSIS AND FORECASTING*. WORLD METEOROLOGICAL ORGANIZATION, 1998.
- [24] MAN Diesel & Turbo. *Basic Principles of Ship Propulsion*. Tech. rep. MAN Diesel & Turbo, 2011.
- [25] Martin Stopford. *Maritime Economics*. Routledge, 2009.
- [26] Clarksons Research Shipping Intelligent Network. *Freight Rates*. Tech. rep. London Baltic Exchange, 2015.
- [27] Port of Rotterdam. *GENERAL TERMS AND CONDITIONS INCLUDING PORT TARIFFS*. Tech. rep. Havenbedrijf Rotterdam N.V, 2020.
- [28] UNITED NATIONS. *PORT PRICING*. UNITED NATIONS, 1975.