

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μελέτη Οικονομικού Ανταγωνισμού με τη χρήση τεχνολογίας Πρακτόρων.»



Του φοιτητή
Κωνσταντίνου Κοντού
Αρ. Μητρώου: 516057

Επιβλέπων
Στάθης Κασδερίδης
Βαθμίδα: Έκτακτο Διδακτικό
Προσωπικό

Ημερομηνία ...14/1/2023.....

Μελέτη Οικονομικού Ανταγωνισμού με τη χρήση τεχνολογίας Πρακτόρων.

Κωδικός Δ.Ε. 22201

Κωνσταντίνος Κοντός

19/6/2022

Ημερομηνία περάτωσης Δ.Ε. 13/1/2023

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Κωνσταντίνου Κοντού που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Αφιέρωση»

Πρόλογος

Η συγκεκριμένη διπλωματική μου κέντρισε το ενδιαφέρον διότι θεωρώ ότι η ανάλυση των συμπεριφορών και η πρόβλεψη αυτών μέσω προγραμματισμού και μαθηματικών εξισώσεων είναι πολύ ενδιαφέρον θέμα. Έπειτα, πίστευα ότι θα πάρω πολλές γνώσεις στον τομέα ανάπτυξης τέτοιων μοντέλων. Κατόπιν, οι αρχικές μου υποθέσεις αποδείχτηκαν αληθινές διότι έλαβα πολλές γνώσεις σε ένα βασικό επίπεδο για τον τρόπο κατασκευής και ανάλυσης ενός τέτοιου μοντέλου. Επίσης, ήταν πολύ ενδιαφέρον το θέμα γιατί το συγκεκριμένο μοντέλο μπορεί να προβλέψει, μέσω μαθηματικών εξισώσεων, το μέλλον ως ένα βαθμό.

Περίληψη

Η συγκεκριμένη διπλωματική έχει ως σκοπό να μελετήσει τις αντιδράσεις της αγοράς σε μικροοικονομικό επίπεδο. Αρχικά, μελετάει τις αντιδράσεις της με την υλοποίηση ενός πρώτου μοντέλου που στηρίζεται σε συγκεντρωτικές μεταβλητές (aggregated). Το πρώτο μοντέλο παίρνει υπόψη την αγορά σαν σύνολο, οι αντιδράσεις της αγοράς επηρεάζονται μόνο από την τιμή των προϊόντων. Έπειτα, ένα δεύτερο μοντέλο μελετάει τις αποφάσεις που επιλέγει κάθε καταναλωτής ξεχωριστά με βάση την προσωπικότητά του. Κοιτάζει δηλαδή το πρόβλημα της συμπεριφοράς σε ατομικό επίπεδο περιγραφής και όχι συγκεντρωτικό. Η μελέτη της συμπεριφοράς γίνεται κοιτάζοντας την επιρροή όλων των μεταβλητών στο μερίδιο που είχαν στην αγορά τα προϊόντα μετά το lock-in. Lock-in ονομάζεται το οικονομικό φαινόμενο στο οποίο τα μερίδια αγοράς ανταγωνιστικών προϊόντων παραμένουν σταθερά με την πάροδο του χρόνου και υπάρχει καθαρός κυρίαρχος παίκτης στην αγορά.

«ECONOMIC COMPETITION ANALYSIS WITH THE USE OF AGENT TECHNOLOGY»

«KONSTANTINOS KONTOS»

Abstract

This thesis' purpose is to investigate the market's reaction on microeconomic level. The first model investigates the market's reaction with aggregated variables. This model investigates the market as a whole and its reactions are only affected by the products' price. The second model investigates the decisions of each consumer separately, and the decision making is based on their personalities. So we investigate the problem on an individual level of description and not aggregated. Furthermore, the investigation of the behavior is based on the influence that the variables have on the market share after the lock-in effect occurs. Lock-in is the economic phenomenon on which the market share of competitive products remains the same over time and there is a clear dominant product on the market.

Ευχαριστίες

Ευχαριστώ πολύ τους γονείς μου που με στηρίζουν καθ' όλη την διάρκεια της ζωής μου σε ότι κάνω. Επίσης θα ήθελα να ευχαριστήσω τον κύριο Στάθη Κασδερίδη, διότι με βοήθησε πάρα πολύ σε κάθε βήμα της διπλωματικής και μου έδειχνε κάθε φορά τον σωστό δρόμο.

Περιεχόμενα

Πρόλογος.....	vi
Περίληψη.....	vii
Abstract	viii
Ευχαριστίες	ix
Περιεχόμενα	x
Συνομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή.....	1
1.1 Περιγραφή Προβλήματος.....	1
1.2 Μεθοδολογία.....	1
1.3 Σύστημα των πρακτόρων	3
1.4 Μαθηματική ανάλυση.....	3
1.5 Δομή της εργασίας	3
Κεφάλαιο 2ο: Σύστημα πρακτόρων	5
2.1 Ορισμός.....	5
2.2 Τα θετικά τους.....	5
2.2.1 Υπολογιστική μέθοδος.....	5
2.2.2 Πειράματα	5
2.2.3 Μοντέλα	5
2.2.4 Πράκτορες.....	6
2.2.5 Το περιβάλλον.....	6
2.3 Τρόπος λειτουργίας.....	6
2.3.1 Πράκτορες.....	6
2.3.2 Περιβάλλον	7
2.3.3 Τυχαιότητα	7
2.3.4 Χρόνος.....	7
2.4 Τρόποι υλοποίησης.....	7
Κεφάλαιο 3ο: Μαθηματικό μοντέλο	9
3.1 Μαθηματικοί Ορισμοί και Έννοιες.....	9
3.1.1 Σταθερό Σημείο (Fixed Point).....	9
3.1.2 Περιοδικός Κύκλος (Limit Cycle).....	9
3.1.3 Παράξενος Ελκυστής (Strange Attractor).....	10
3.2 Πρώτο μοντέλο.....	11

3.3	Τελικό μοντέλο.....	12
Κεφάλαιο 4ο: Υπολογιστικό Μοντέλο.....		16
4.1	Υλοποίηση Πρώτου Μοντέλου.....	16
4.1.1	Περιγραφή του Κώδικα.....	16
4.1.2	Testing.....	18
4.1.3	Εξαγωγή Δεδομένων.....	18
4.2	Υλοποίηση Τελικού Μοντέλου.....	19
4.2.1	Περιγραφή του κώδικα.....	20
4.2.2	Testing.....	21
4.2.3	Εξαγωγή δεδομένων.....	22
Κεφάλαιο 5ο: Αποτελέσματα.....		24
5.1	Δημιουργία Διαγραμμάτων με την R.....	24
5.2	Εύρος Τιμών Μεταβλητών.....	25
5.3	Εύρος βημάτων.....	25
5.4	Ανάλυση Αποτελεσμάτων.....	26
5.4.1	Πρώτο Μοντέλο.....	26
5.4.2	Τελικό Μοντέλο.....	32
5.5	Σύγκριση των Μοντέλων.....	44
5.6	Οπτική Απεικόνιση.....	44
Κεφάλαιο 6ο: Επίλογος.....		48
6.1	Συμπεράσματα.....	48
6.2	Σφάλματα και περιορισμοί.....	48
6.3	Βελτιώσεις.....	48
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		49
ΠΑΡΑΡΤΗΜΑ Α : ΠΡΟΓΡΑΜΜΑ ΠΡΩΤΟΥ ΜΟΝΤΕΛΟΥ.....		50
ΠΑΡΑΡΤΗΜΑ Β : ΠΡΟΓΡΑΜΜΑ ΤΕΛΙΚΟΥ ΜΟΝΤΕΛΟΥ.....		54
ΠΑΡΑΡΤΗΜΑ Γ : SCRIPT R.....		57
ΠΑΡΑΡΤΗΜΑ Δ : ΔΕΔΟΜΕΝΑ ΠΡΩΤΟΥ ΜΟΝΤΕΛΟΥ.....		58
ΠΑΡΑΡΤΗΜΑ Ε : ΔΕΔΟΜΕΝΑ ΤΕΛΙΚΟΥ ΜΟΝΤΕΛΟΥ.....		58

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία

Κεφάλαιο 1ο: Εισαγωγή

1.1 Περιγραφή προβλήματος

Όταν κάποιο προϊόν κυριαρχεί την αγορά αυτό ονομάζεται lock-in. Το φαινόμενο αυτό είναι πολύ δύσκολο να αντιστραφεί αφότου γίνει. Το βλέπουμε σε πολλές μεγάλες εταιρίες οι όποιες κυριαρχούν για πολλά χρόνια σε συγκεκριμένες αγορές παγκοσμίως (π.χ. Apple, Microsoft, Ford, Toyota). Για να καταλάβουμε πως μπορεί να αντιστραφεί το φαινόμενο αρχικά πρέπει να το κατανοήσουμε πλήρως. Αυτό μπορεί να επιτευχθεί γνωρίζοντας όλους του παράγοντες που επηρεάζουν τους καταναλωτές να επιλέξουν ένα προϊόν σε σύγκριση με ένα άλλο. Αρχικά, θα γίνει προσομοίωση με παράγοντες με βάση την διαφοροποίηση των προϊόντων (π.χ. τιμή, ποιότητα). Στην συνέχεια θα προστεθεί και ο ψυχολογικός παράγοντας του καταναλωτή ο οποίος είναι απαραίτητος για να μπορέσουμε να έχουμε μια ολοκληρωμένη εικόνα για όλους τους παράγοντες επιρροής.

1.2 Μεθοδολογία

Αρχικά, πρέπει να αποφασίσουμε με ποιον τρόπο να υλοποιήσουμε ένα τέτοιο μοντέλο. Ένας από τους καλύτερους τρόπους είναι μέσω προσομοίωσης και προγραμματισμού, διότι ότι γίνεται με προγραμματισμό πρέπει να είναι ακριβώς ορισμένο για να δουλέψει. Επίσης, είναι γρήγορο, γίνεται εύκολα η εξαγωγή δεδομένων, η μελέτη τους και μπορούμε επίσης να το υλοποιήσουμε σε πολύ μεγάλη κλίμακα (σε αντίθεση με μεθόδους (π.χ. συνέντευξη, focus groups, ερωτηματολόγια) άλλων κοινωνικών επιστημών).

Στην εργασία υλοποιήθηκαν δύο μαθηματικά-υπολογιστικά μοντέλα που προσομοιώνουν την διαδικασία του οικονομικού φαινομένου lock-in. Για το πρώτο μοντέλο [2] χρησιμοποιήθηκαν εξισώσεις με βάση συγκεντρωτικές μεταβλητές οι οποίες περιγράφουν ένα μοντέλο το οποίο παίρνει αποφάσεις με βάση την διαφοροποίηση των προϊόντων. Αυτό υλοποιήθηκε με την γλώσσα προγραμματισμού Java στην πλατφόρμα Apache NetBeans. Θα μπορούσε να χρησιμοποιηθεί οποιαδήποτε άλλη γλώσσα προγραμματισμού αλλά η Java είναι μια γρήγορη γλώσσα, με την οποία υπάρχει η δυνατότητα μέσω μιας βιβλιοθήκης (java.io.file) για εύκολη εξαγωγή των δεδομένων. Επίσης, επιλέχθηκε για λόγους ευκολίας και εξοικείωσης με τη συγκεκριμένη γλώσσα. Στην συνέχεια, δημιουργήθηκε ένας "ψευδοκώδικας" ο οποίος βοήθησε στην υλοποίηση του πραγματικού κώδικα. Όταν ο πρώτος κώδικας είχε τελειώσει ξεκίνησε η εξαγωγή των δεδομένων, τα οποία αποθηκεύονταν σε αρχεία csv. Έπειτα, κατασκευάστηκαν διαγράμματα στην εφαρμογή Microsoft Excel το οποίο δεν είναι ένα εργαλείο για απεικόνιση και σύγκριση τόσο σύνθετων συνόλων δεδομένων. Για αυτόν τον λόγο στην συνέχεια αντικαταστάθηκε με την R. Η οποία είναι μια εφαρμογή που έχει την δική της γλώσσα προγραμματισμού, αποθηκεύει πολλαπλά σύνολα δεδομένων στην μνήμη της, είναι γρήγορη και είναι το κατάλληλο εργαλείο για αναπαράσταση και επεξεργασία τέτοιων δεδομένων.

Για την υλοποίηση του δεύτερου μοντέλου ήταν απαραίτητο να χρησιμοποιηθεί ένα σύστημα πρακτόρων. Το μοντέλο αυτό μας επιτρέπει να μελετήσουμε σε ατομικό επίπεδο την διαδικασία της απόφασης με ένα σύστημα στο οποίο πράκτορες αλληλοεπιδρούν μεταξύ τους αλλά και με το περιβάλλον γύρω τους. Με αυτόν τον τρόπο καταφέραμε να αναλύσουμε το γιατί επιλέγουν το συγκεκριμένο προϊόν οι καταναλωτές την συγκεκριμένη χρονική στιγμή, με παράγοντες επιρροής να περιλαμβάνουν την αντικειμενική εκτίμηση του προϊόντος και την ψυχολογία των καταναλωτών. Το σύστημα των πρακτόρων μπορεί να υλοποιηθεί είτε σε οποιαδήποτε αντικειμενοστραφή γλώσσα προγραμματισμού, είτε σε γλώσσες που έχουν δημιουργηθεί αποκλειστικά για την προσομοίωση

τέτοιων μοντέλων όπως είναι η NetLogo. Γλώσσες όπως η C++ / Nim είναι πολύ γρήγορες γλώσσες και πολύ μεγάλα υπολογιστικά μοντέλα θα ήταν πολύ πιο αποδοτικά σε σύγκριση με κάποιο έτοιμο εργαλείο. Το αρνητικό τους όμως είναι ότι αρχικά δεν υπάρχει σχηματική απεικόνιση του μοντέλου όπως θα δούμε ότι υπάρχει στην NetLogo (δηλ. λείπει ενσωματωμένη ικανότητα γραφικών παραστάσεων). Επίσης, δεν έχουν κάποιο έτοιμο περιβάλλον το οποίο αναπαριστά το σύστημα των πρακτόρων. Οπότε πρέπει να δημιουργηθεί όλο το σύστημα από την αρχή (με βάση σχετικές βιβλιοθήκες) κάτι το οποίο είναι αρκετά χρονοβόρο. Κατόπιν, πολλά μοντέλα προσομοίωσης υλοποιούνται στην NetLogo, οπότε έχει πολύ περισσότερο υλικό που μπορεί να διαβάσει και να χρησιμοποιήσει κάποιος. Οπότε, μπορεί να διορθώσει πιο εύκολα πιθανό λάθος που θα προκύψει, θα μπορεί να βελτιστοποιήσει πιο γρήγορα το σύστημα του και να ξεκινήσει έχοντας το σύστημα πρακτόρων διαθέσιμο αλλά χρησιμοποιώντας κάποιες έτοιμες εντολές. Κατά συνέπεια, επιλέχτηκε για τους παραπάνω λόγους η NetLogo ως εργαλείο υλοποίησης.

Κατόπιν, έγινε ανάλυση των μαθηματικών εξισώσεων και εισήχθησαν μέσα στο πρόγραμμα. Στην συνέχεια, η NetLogo έχει ένα εργαλείο το οποίο ονομάζεται BehaviorSpace το οποίο βοηθάει στην εξαγωγή δεδομένων αλλά και στον πειραματισμό με διαφορετικές τιμές των μεταβλητών. Οπότε, εκμεταλλευόμενοι το συγκεκριμένο εργαλείο, επιλέγουμε μέχρι ποιο βήμα πρέπει να "τρέξουν" οι προσομοιώσεις για την αποφυγή σπατάλης χρόνου και υπολογιστικών πόρων. Δηλαδή πρέπει να δούμε σε διάφορες προσομοιώσεις ποιος είναι ο μέγιστος χρόνος που πρέπει να περάσει ώστε να γίνεται lock-in. Στην συνέχεια, πρέπει να δούμε τι τιμές θα πρέπει να δώσουμε στις μεταβλητές και τις παραμέτρους του μαθηματικού-υπολογιστικού μοντέλου έτσι ώστε τα αποτελέσματα που θα αναπαράγουμε να έχουν νόημα.. Βλέπουμε μέσω πειραμάτων σε τι εύρος τιμών επηρεάζει κάθε μεταβλητή το σύστημα και δίνουμε έναν ικανοποιητικό αριθμό ενδιάμεσων τιμών σε αυτό το διάστημα (για την κάθε μεταβλητή), στην προκειμένη περίπτωση πέντε. Έπειτα, επειδή στο σύστημα μας υπάρχουν τυχαίοι παράγοντες πρέπει με κάποιον τρόπο να χειριστούμε αυτή την τυχαιότητα. Ο συνηθής τρόπος χειρισμού είναι για δεδομένες τιμές των μεταβλητών και των παραμέτρων κάνουμε N προσομοιώσεις στις οποίες οι όποιες διαφοροποιήσεις υπάρχουν οφείλονται μόνο στους τυχαίους παράγοντες. Εάν κανείς πάρει τον μέσο όρο των N προσομοιώσεων για την κάθε μεταβλητή που μελετά, θεωρώντας τις τιμές όλων των άλλων μεταβλητών και παραμέτρων σταθερές, τότε ουσιαστικά μπορεί να εξάγει συμπέρασμα για την επιρροή της συγκεκριμένης μεταβλητής του μοντέλου πάνω στο τελικό μερίδιο αγοράς των ανταγωνιστικών προϊόντων με υψηλή εμπιστοσύνη. Αυτό υλοποιείται, στην προκειμένη περίπτωση, κάνοντας εκατό τυχαίες επαναλήψεις («πειράματα») για κάθε τιμή που θα έπαιρνε η μεταβλητή και παίρνοντας τον μέσο όρο των τελικών τιμών των μεριδίων αγοράς που προκύπτουν. Για να πάρουμε τον μέσο όρο έχουμε διάφορες εναλλακτικές. Μπορεί να γίνει στην NetLogo αλλά όχι με την χρήση του BehaviorSpace, διότι δεν μας δίνει αυτήν την δυνατότητα, οπότε θα πρέπει για όλες τις προσομοιώσεις να γράψουμε κώδικα που θα τις υλοποιεί γεγονός το οποίο δεν είναι αποδοτικό. Επίσης, μπορεί να γίνει μέσω ενός τρίτου προγράμματος, όπως η Java, που θα πρέπει να εισαχθούν τα δεδομένα σε αυτό, να τα επεξεργαστούμε γράφοντας τον κατάλληλο κώδικα, μετά να τα εξάγουμε και τέλος να τα εισάγουμε στην R. Ο πιο αποδοτικός τρόπος όμως είναι να εισάγουμε απευθείας τα δεδομένα στην R και να γίνει η επεξεργασία εκεί. Καθώς είναι αρκετά γρήγορη η γλώσσα και έχοντας όλο αυτό το πλήθος δεδομένων αποθηκευμένο στην μνήμη της μπορεί να γίνει οποιαδήποτε επεξεργασία χρειαστεί ακόμα και για τα πρωτόλεια δεδομένα. Τέλος, γίνεται γραφική απεικόνιση των τελικών μεριδίων που υπάρχουν στην αγορά από τα προϊόντα ανάλογα με τις διάφορες τιμές των μεταβλητών. Στο μοντέλο που υλοποιήσαμε εστιάσαμε στον ανταγωνισμό δύο προϊόντων ως την πλέον βασική περίπτωση, αλλά το μοντέλο μπορεί εύκολα να επεκταθεί στην γενική περίπτωση.

1.3 Σύστημα των πρακτόρων

Σε πολλές περιπτώσεις, αναλυτές μελετάνε την αγορά σαν σύνολο δηλαδή μελετάνε την αγορά μακροοικονομικά, χωρίς να λαμβάνουν υπόψη γιατί ο κάθε άνθρωπος σαν άτομο επιλέγει την συγκεκριμένη απόφαση την συγκεκριμένη χρονική στιγμή. Στην συγκεκριμένη πτυχιακή, το ζητούμενο είναι η ανάλυση από μικροοικονομικής άποψης. Αυτό για να γίνει πρέπει με κάποιον τρόπο να εισαχθεί στον προγραμματισμό ο ανθρώπινος παράγοντας. Δηλαδή τα συναισθήματα του κάθε ανθρώπου, όπως και η αβεβαιότητα, η γνώση ότι οι υπόλοιποι άνθρωποι γύρω του έχουν επιλέξει ένα συγκεκριμένο προϊόν, το γούστο του, το κατά πόσο ένα προϊόν θα μολύνει το περιβάλλον και άλλα θα αναλυθούν εκτενέστερα στα επόμενα κεφάλαια. Για να γίνει κάτι τέτοιο το σύστημα των πρακτόρων είναι ένα πολύ αξιόπιστο και εύκολο εργαλείο που μας βοηθάει να κάνουμε ακριβώς αυτό. Οι άνθρωποι είναι οι πράκτορες στους οποίους προσδίδουμε διάφορα χαρακτηριστικά, διαφορετικά στον καθένα έτσι ώστε να μπορούμε να αναλύσουμε τις συμπεριφορές τους. Έτσι, μπορούμε να φτάσουμε όσο πιο κοντά γίνεται στο να προσομοιωθεί με ακρίβεια ένα αληθινό περιβάλλον ο μόνος περιορισμός είναι η υπολογιστική δύναμη. Η υπολογιστική ισχύς θα πρέπει να είναι άπειρη για να μπορέσει να προσομοιωθεί ο αληθινός κόσμος με κάθε λεπτομέρεια, διότι οι παράγοντες που επηρεάζουν κάθε άνθρωπο να επιλέξει ένα συγκεκριμένο προϊόν είναι άπειροι στην πραγματικότητα και δεν είναι δυνατόν να τους ξέρουμε όλους, αλλά αυτό είναι το πιο κοντινό στον αληθινό κόσμο που είναι διαθέσιμο.

1.4 Μαθηματική ανάλυση

Το συγκεκριμένο μοντέλο αποτελείται από μαθηματικές εξισώσεις οι οποίες έχουν κατασκευαστεί από οικονομολόγους, ψυχολόγους, μαθηματικούς και κοινωνιολόγους [2]. Συνεργάστηκαν ώστε να αναπαραστήσουν με εξισώσεις τα αποτελέσματα που θα έχουν συγκεκριμένα ερεθίσματα στον κόσμο. Για το πρώτο μοντέλο όλες οι εξισώσεις αλληλοεπιδρούν μεταξύ τους και τα δεδομένα αλλάζουν σε κάθε βήμα. Τα αποτελέσματα τα οποία αναπαράγουν είναι μακροοικονομικά. Αυτό συμβαίνει διότι εμφανίζουν την συνολική εικόνα της αγοράς μετά από κάθε βήμα, χωρίς να προσδιορίζεται ποιος καταναλωτής επέλεξε το συγκεκριμένο προϊόν και γιατί. Για το δεύτερο μοντέλο, προστίθενται οι εξισώσεις οι οποίες προσομοιώνουν τον χαρακτήρα των πρακτόρων. Με την πρόσθεση αυτή, τα αποτελέσματα που θα αναπαραχθούν θα είναι μικροοικονομικά. Οι συγκεκριμένες εξισώσεις δεν αλληλοεπιδρούν μεταξύ τους, αλλά έχουν μια εξίσωση η οποία τις ενώνει με σκοπό να εξάγει την ικανοποίηση του καταναλωτή για να το κάθε προϊόν. Κατόπιν, τίθεται ένα κατώτατο όριο αυτής της ικανοποίησης το οποίο επηρεάζεται από την οικονομική κατάσταση στην οποία βρίσκεται ο καταναλωτής αλλά και από την οικονομική κατάσταση των γειτόνων του. Αυτό έχει ως σκοπό αν ένας καταναλωτής είναι ικανοποιημένος οριακά για ένα προϊόν πιθανότατα δεν είναι και σίγουρος για την επιλογή του, οπότε επιλέγει με βάση την επιλογή των γειτόνων του. Επίσης, και στα δυο παραπάνω μοντέλα υπάρχει και ένας στοχαστικός παράγοντας έτσι ώστε να μπορεί να συμπεριλάβει την τυχαιότητα η οποία υπάρχει στον αληθινό κόσμο.

1.5 Δομή της εργασίας

Στο πρώτο κεφάλαιο εκτίθεται το πρόβλημα, δίνεται μια περίληψη της μεθοδολογίας που ακολουθήθηκε για την επίλυση του προβλήματος, εξηγούνται κάποιες βασικές έννοιες οι οποίες είναι απαραίτητες για την κατανόηση των υπόλοιπων κεφαλαίων και εξηγείται πως έγινε η μαθηματική ανάλυση. Το δεύτερο κεφάλαιο αναφέρεται εκτενώς στα μοντέλα τα οποία είναι βασισμένα στους πράκτορες. Ποιος είναι ο ορισμός τους, πιο πρόβλημα λύνουν, πως δουλεύουν και πως μπορούν να

Κεφάλαιο 1

υλοποιηθούν. Στο τρίτο κεφάλαιο αρχικά αναφέρονται κάποιοι μαθηματικοί ορισμοί οι οποίοι είναι απαραίτητοι για την κατανόηση της μαθηματικής ανάλυσης που θα ακολουθήσει. Στην συνέχεια θα αναλυθούν εκτενώς τα δυο υλοποιημένα μοντέλα . Στο τέταρτο κεφάλαιο περιέχεται η ανάλυση του υπολογιστικού μοντέλου. Στην ουσία πως δουλεύει ο κώδικας, πως έγινε το testing και πως εξηγήθηκαν τα δεδομένα. Το πέμπτο κεφάλαιο είναι αφιερωμένο στα αποτελέσματα που λήφθηκαν , με ποιον ακριβώς τρόπο επιλέχθηκε το συγκεκριμένο εύρος τιμών καθώς περιλαμβάνεται και σύγκριση των δυο μοντέλων . Το έκτο και τελευταίο κεφάλαιο είναι ο επίλογος ο οποίος περιλαμβάνει τα συμπεράσματα, τι κερδίσαμε από την συγκεκριμένη διπλωματική εργασία, τι βελτιώσεις θα μπορούσαν να γίνουν και την βιβλιογραφία.

Κεφάλαιο 2ο: Σύστημα πρακτόρων

2.1 Ορισμός

Η μοντελοποίηση βασισμένη σε πράκτορες είναι μια υπολογιστική μέθοδος η οποία επιτρέπει τον εκάστοτε αναλυτή να δημιουργήσει, να αναλύσει, και να πειραματιστεί σε μοντέλα τα οποία αποτελούνται από πράκτορες οι οποίοι αλληλοεπιδρούν μέσα σε ένα περιβάλλον[2].

2.2 Τα θετικά του

2.2.1 Υπολογιστική μέθοδος

Η μοντελοποίηση μέσω του συστήματος των πρακτόρων είναι μια μορφή υπολογιστικής κοινωνικής επιστήμης. Δηλαδή, χρησιμοποιείται προγραμματισμός για να χτιστεί ένα μοντέλο το οποίο αναλύει κοινωνικές συμπεριφορές. Μέχρι σήμερα, ο τρόπος των κοινωνικών επιστημών να δημιουργήσουν ένα μοντέλο, είναι να ερμηνεύσουν μια απλουστευμένη έκδοση για το πως νομίζουν ότι δουλεύει ο κόσμος όσο πιο καθαρά μπορούν. Τα υπολογιστικά μοντέλα τα οποία είναι κατασκευασμένα ως προγράμματα υπολογιστών έχουν δυο βασικά θετικά στοιχεία σε σχέση με τα μοντέλα που είναι σε φυσική γλώσσα. Αρχικά, είναι απαραίτητο τέτοιου είδους μοντέλα να είναι ακριβή και όντας προγράμματα υπολογιστή είναι αναγκαστικά ακριβή έτσι ώστε να μπορούν να εκτελεστούν σε ένα υπολογιστικό σύστημα. Τέλος, είναι σχετικά εύκολο να μοντελοποιηθούν θεωρίες οι οποίες αφορούν επεξεργασίες δεδομένων διότι όλα τα προγράμματα αυτό κάνουν ουσιαστικά.

2.2.2 Πειράματα

Στις κοινωνικές επιστήμες είναι σπάνιο να διεξαχθούν πειράματα με σκοπό την εξαγωγή αποτελεσμάτων, κάτι το οποίο είναι βασική μέθοδος στις υπόλοιπες επιστήμες. Κατόπιν, ο τρόπος διεξαγωγής πειραμάτων είναι ο εξής : Αρχικά, εφαρμόζεται μια θεραπεία σε ένα απομονωμένο σύστημα και παρατηρούμε τις διαφορές μεταξύ αυτού και ενός άλλου απομονωμένου συστήματος το οποίο δεν έλαβε θεραπεία. Κάνοντας κάτι τέτοιο σε ομάδες ανθρώπων, μπορεί εύκολα να συμπεράνει κάποιος ότι θα ήταν πολλές φορές ανήθικο και πολύ δύσκολο στην υλοποίησή του. Το σύστημα πρακτόρων λύνει το συγκεκριμένο πρόβλημα, διότι μπορεί να προσομοιώσει τις ανθρώπινες συμπεριφορές σε ένα πρόγραμμα υπολογιστή. Επίσης, υπάρχει η δυνατότητα να διεξαχθούν τα πειράματα όσες φορές επιθυμούμε, μεταβάλλοντας μάλιστα τα ερεθίσματα αλλά και τις συμπεριφορές των ανθρώπων. Παρόλα αυτά, για να δουλέψει σωστά ένα τέτοιο σύστημα και τα αποτελέσματα που θα ληφθούν να έχουν κάποιο νόημα, θα πρέπει το μοντέλο να συμπεριφέρεται όπως ακριβώς ο πραγματικός κόσμος. Συμπερασματικά, η συγκεκριμένη υλοποίηση δεν είναι πανάκια.

2.2.3 Μοντέλα

Υπάρχουν διάφορων ειδών μοντέλων τα οποία έχουν διαφορετικές εφαρμογές ανάλογα με το αντικείμενο μοντελοποίησης. Αρχικά, είναι τα μοντέλα κλίμακας, τα οποία κατασκευάζουν σε μικρότερη κλίμακα το ζητούμενο μοντέλο, το μελετάνε και στην συνέχεια τα αποτελέσματα πρέπει να επεκταθούν προς τα πάνω έτσι ώστε να ταιριάζουν με το αρχικό μοντέλο. Ένα παράδειγμα ενός τέτοιου μοντέλου θα μπορούσε να είναι ο πληθυσμός μιας πόλης, στον οποίο θα παρατηρούνταν ο τρόπος με τον οποίο θα αντιδρούσε σε συγκεκριμένα ερεθίσματα. Κατόπιν, για λόγους ταχύτητας θα μελετηθεί το 1/10 του πληθυσμού της και στην συνέχεια θα επεκταθούν τα αποτελέσματα προς τα πάνω. Στην συνέχεια, ένας άλλος τύπος μοντέλου είναι το τύπου-ιδανικό μοντέλο το οποίο εξάγει κάποιο

χαρακτηριστικό από τον στόχο, το οποίο δεν επηρεάζει ουσιαστικά το σύστημα, με σκοπό την απλοποίηση του συστήματος. Επιπροσθέτως, υπάρχουν τα αναλογικά μοντέλα τα οποία χρησιμοποιούν φαινόμενα τα οποία είναι καλύτερα κατανοητά και τα αντιστοιχίζουν με τον στόχο. Τα μοντέλα αυτά για να λειτουργήσουν πρέπει η αναλογία να είναι αξιόπιστη κάτι το οποίο είναι αρκετά δύσκολο όταν μιλάμε για πολύπλοκα συστήματα. Τέλος, υπάρχουν τα μοντέλα βασισμένα σε μαθηματικές εξισώσεις, ο συγκεκριμένος τύπος μοντέλου χρησιμοποιήθηκε και στην προκειμένη εργασία. Η εγκυρότητα των μοντέλων αυτών εξαρτάται κατά μεγάλο βαθμό από το αν τα πειραματικά δεδομένα ταιριάζουν με την συγκεκριμένη εξίσωση.

2.2.4 Πράκτορες

Τα συστήματα των πρακτόρων αποτελούνται από πράκτορες οι οποίοι αλληλοεπιδρούν μέσα σε ένα περιβάλλον. Οι πράκτορες μπορεί είτε να είναι συγκεκριμένο μέρος ενός προγράμματος, είτε ξεχωριστό πρόγραμμα το οποίο αλληλοεπιδρά με το κύριο πρόγραμμα και αναπαραστά ανθρώπους, ολόκληρες εταιρίες ή ακόμα και κοινωνικά σύνολα. Επιπλέον, είναι απολύτως βασικό οι πράκτορες να μπορούν να αλληλοεπιδρούν μεταξύ τους μέσω μηνυμάτων, παρατηρώντας ο ένας τον άλλο και αυτή είναι η βασική διαφορά με τις υπόλοιπες υλοποιήσεις υπολογιστικών μοντέλων.

2.2.5 Το περιβάλλον

Το περιβάλλον είναι ο εικονικός κόσμος στον οποίο δρουν οι πράκτορες, πολλές φορές είναι πιθανό το περιβάλλον να μην επηρεάζει τους πράκτορες και την συμπεριφορά τους για το συγκεκριμένο μοντέλο το οποίο αναπαριστάται. Αντιθέτως, υπάρχουν περιπτώσεις στις οποίες το περιβάλλον μπορεί να είναι υπαρκτή γεωγραφική τοποθεσία, ιδιότητες μιας τοποθεσίας (π.χ. Το πλήθος ζώων σε μια περιοχή, το ποσοστό του πλούτου), είτε μπορεί να είναι οι συνδέσεις οι οποίες υπάρχουν ανάμεσα στους πράκτορες(π.χ. Οι πράκτορες που είναι υπέρ ενός κόμματος, οι πράκτορες που ανήκουν σε μια χώρα).

2.3 Τρόπος λειτουργίας

2.3.1 Πράκτορες

Οι πράκτορες έχουν συγκεκριμένα χαρακτηριστικά που τους επιτρέπουν να αλληλοεπιδρούν και να έχουν ξεχωριστές προσωπικότητες. Αρχικά, έχουν εργαλεία τα οποία τους επιτρέπουν να αντιληφθούν το περιβάλλον γύρω τους, την τοποθεσία άλλων πρακτόρων ή αντικειμένων, δηλαδή έχουν αντίληψη. Κατόπιν, είναι σε θέση να επιτελέσουν συγκεκριμένες πράξεις όπως κίνηση, επικοινωνία και πράξη (αλληλεπίδραση με το περιβάλλον). Επίσης, διαθέτουν μνήμη η οποία καταγράφει τις προηγούμενες πράξεις τους και τα πιστεύω τους. Κατόπιν, διαθέτουν μια συγκεκριμένη πολιτική, η οποία περιλαμβάνει ένα σετ κανόνων που πρέπει να ακολουθούν, στρατηγικές και γενικά πληροφορίες για την συμπεριφορά τους. Επιπροσθέτως, οι πράκτορες για να εισαχθούν σε προγράμματα υπολογιστών χρησιμοποιούνται αντικειμενοστραφείς γλώσσες. Οι πράκτορες δηλώνονται ως αντικείμενα μιας κλάσης τα οποία έχουν το καθένα τις δικές του μεθόδους και ιδιότητες. Κατόπιν, όπως προαναφέρθηκε χρειάζεται να τεθούν κανόνες συμπεριφοράς, που σε κάθε "τρέξιμο" του προγράμματος θα ελέγχονται μέσω μιας ή πολλαπλών συναρτήσεων if. Συμπληρωματικά, είναι απαραίτητη η ύπαρξη της μνήμης, η οποία θα είναι σε μορφή μεταβλητών. Στην συνέχεια, χρειάζεται ένας διερμηνέας κανόνων ο οποίος είναι ένα κομμάτι κώδικα το οποίο με την χρήση της μνήμης, είναι υπεύθυνο για την τήρηση των κανόνων. Τέλος, πρέπει να υπάρχει επεξεργασία εισόδου και εξόδου έτσι ώστε να είναι δυνατή η αλληλεπίδραση με το περιβάλλον αλλά και μεταξύ τους.

2.3.2 Περιβάλλον

Το περιβάλλον είναι το μέσο επικοινωνίας ανάμεσα στους πράκτορες αλλά και με άψυχες οντότητες. Κατόπιν, το περιβάλλον είναι πολύ σημαντικό σε αυτά τα συστήματα διότι αποθηκεύει προσωρινά στην μνήμη του τα μηνύματα των πρακτόρων, έτσι ώστε να μην υπάρχουν ατυχήματα λόγω της σειράς που στάλθηκαν. Επίσης, πολλές φορές περιλαμβάνει παθητικά αντικείμενα (π.χ. δρόμους, θάλασσα), τρόπους μεταφοράς, τρόπους επικοινωνίας και αποθήκευση υλικών αγαθών (π.χ. χρήματα). Κατόπιν, προγραμματίζεται με τον ίδιο τρόπο όπως οι πράκτορες αλλά πιο απλά, διότι δεν αλληλοεπιδρά με το υπόλοιπο περιβάλλον γύρω του. Τέλος, πολλές φορές αυτό παραμελείται από τους ερευνητές παρόλο που έχει μεγάλη επίδραση στις αποφάσεις που θα πάρουν οι πράκτορες που το κατοικούν.

2.3.3 Τυχαιότητα

Ένας τρόπος για να αναπαρασταθεί εύκολα ένα μέρος της πολυπλοκότητας ενός συστήματος είναι να εισαχθεί ένας τυχαίος παράγοντας. Παραδείγματος χάρη, οι καταναλωτές θα πρέπει να έχουν συγκεκριμένα χρήματα διαθέσιμα, και γι αυτό υποθέτουμε ότι κάθε καταναλωτής λαμβάνει ένα ποσό χρημάτων από μια επιλεγμένη τυχαία κατανομή εισοδήματος (π.χ. κανονική κατανομή). Επίσης, ο παράγοντας τυχαιότητας μπορεί να αναπαραστήσει την τυχαιότητα που υπάρχει στον κόσμο (π.χ. λάθη που γίνονται, άσκοπα έξοδα).

2.3.4 Χρόνος

Ένα από τα πιο δύσκολα κομμάτια της σχεδίασης μοντέλων βασισμένων στους πράκτορες είναι ο συγχρονισμός όλων των πράξεων που συμβαίνουν, χωρίς την ύπαρξη ανεπιθύμητων αλληλοεπιδράσεων. Ο τρόπος για την επίλυση του συγκεκριμένου προβλήματος που επελέχθηκε για την πτυχιακή είναι η **μέθοδος προσομοίωσης διακριτού χρόνου**. Εισήχθηκε στο πρόγραμμα η έννοια των βημάτων όπου σε κάθε βήμα όλοι οι πράκτορες έχουν την ευκαιρία να πράξουν. Επιπροσθέτως, η συγκεκριμένη προσθήκη δεν είναι αρκετή από μόνη της για να επιλυθούν όλα τα προβλήματα, οπότε υπάρχουν τρεις έννοιες οι οποίες χρειάζονται προσοχή κατά την σχεδίαση μοντέλων τέτοιου είδους. Αρχικά, πολύ σημαντική έννοια είναι ο συγχρονισμός, καθώς το αποτέλεσμα θα είναι εντελώς διαφορετικό αν για παράδειγμα, ο πράκτορας ένα στείλει μήνυμα στον πράκτορα δυο και απαντήσει ο πράκτορας δυο στον ένα και μετά στείλει ο πράκτορας τρία μήνυμα στον δυο και εντελώς διαφορετικό αν πρώτα στείλουν οι πράκτορες ένα και τρία μήνυμα στον δυο και ο δυο απαντήσει μετά στον πράκτορα ένα. Κατόπιν, αυτό συμβαίνει διότι οι πράκτορες δεν μπορούν να δράσουν όλοι μαζί. Επομένως, ένας τρόπος για να λυθεί το συγκεκριμένο πρόβλημα θα ήταν να προγραμματιστούν οι πράκτορες να δρουν με διαδοχική σειρά, το οποίο δεν είναι συχνά καλή λύση διότι μπορεί να υπάρξει σημαντική μείωση της επίδοσης του προγράμματος. Στη συνέχεια, μπορούν οι πράκτορες να καλούνται σε τυχαία σειρά σε κάθε χρονική στιγμή, το θετικό σε αυτήν την περίπτωση είναι ότι η επίδραση της διαφορετικής σειράς μπορεί να μελετηθεί συγκρίνοντας τα αποτελέσματα από πολλές προσομοιώσεις. Επίσης, ο τελευταίος τρόπος είναι οι πράκτορες να καλούνται με τυχαία σειρά, αλλά τα μηνύματα να αποθηκεύονται προσωρινά στο περιβάλλον, μέχρι να τελειώσουν την αποστολή όλοι οι πράκτορες. Ο τρόπος αυτός είναι ο καλύτερος γιατί η σειρά κατά την οποία θα πράξουν οι πράκτορες δεν επηρεάζει καθόλου το αποτέλεσμα. Επιπροσθέτως, κάτι άλλο που θα πρέπει να προσεχθεί είναι ότι ο αληθινός χρόνος δεν χρειάζεται να είναι ίδιος με τον χρόνο της προσομοίωσης, αυτό συμβαίνει γιατί οι μεταβλητές της προσομοίωσης μεταβάλλονται μόνο όταν συμβαίνει κάποια πράξη, οπότε τα διαστήματα χρόνου στα οποία δεν έχουμε γεγονότα μπορούν να παραληφθούν. Κατόπιν, πολλές φορές είναι δύσκολο να μετατραπεί ο χρόνος της προσομοίωσης στον αληθινό χρόνο, μια προσεγγιστική λύση είναι να συγκριθεί η πρόοδος της προσομοίωσης με την πρόοδο κάποιου παρόμοιου αληθινού

γεγονότος, παρόλα αυτά δεν υπάρχει κάποια ακριβή λύση σε αυτήν την περίπτωση. Έπειτα, για να υλοποιηθεί ένα μοντέλο το οποίο βασίζεται στο σύστημα των πρακτόρων τυπικά χρησιμοποιούμε ως μέσο τον προγραμματισμό. Επιπροσθέτως, οι κατάλληλες γλώσσες προγραμματισμού είναι οι αντικειμενοστραφείς διότι, όπως προαναφέρθηκε οι πράκτορες μπορούν να δηλωθούν ως αντικείμενα κλάσεων. Επίσης, υπάρχουν πλατφόρμες με τις δικές τους γλώσσες προγραμματισμού (π.χ. NetLogo) στις οποίες οι πράκτορες είναι ήδη εμφολευμένοι και έχουν και εργαλεία τα οποία βοηθούν στον πειραματισμό και στην εξαγωγή δεδομένων των μοντέλων. Αντιθέτως, στις βασικές γλώσσες προγραμματισμού όλα αυτά πρέπει να γίνουν από την αρχή, όμως για τη μελέτη πολύπλοκων μοντέλων τα οποία έχουν μεγάλες απαιτήσεις μνήμης είναι πιο γρήγορος αυτός ο τρόπος υλοποίησης. Συμπερασματικά, και οι δυο τρόποι έχουν τα θετικά και τα αρνητικά τους στην προκειμένη περίπτωση επιλέχτηκε η NetLogo ως πλατφόρμα υλοποίησης διότι δεν είναι ιδιαίτερα απαιτητικό το συγκεκριμένο μοντέλο, η γλώσσα έχει περισσότερες πηγές για εκμάθηση και επίλυση αποριών, διαθέτει οπτική αναπαράσταση των αποτελεσμάτων και τέλος, γιατί το σύνολο της διαδικασίας ανάπτυξης-προσομοίωσης-παραγωγή δεδομένων-ανάλυσης θα γινόταν πιο σύντομο.

Κεφάλαιο 3ο: Μαθηματικό μοντέλο

3.1 Μαθηματικοί Ορισμοί και Έννοιες

Ο σκοπός αυτής της υποενότητας είναι η επεξήγηση κάποιων βασικών μαθηματικών όρων, με σκοπό να καταστεί δυνατή η κατανόηση των παρακάτω κεφαλαίων. Αρχικά, η έννοια της ισορροπίας είναι η πιο βασική έννοια προς επεξήγηση. Αυτή μπορεί να είναι είτε στατική (από την στιγμή που φτάσει το σύστημα σε κατάσταση ισορροπίας μένει παγωμένο εκεί και δεν αλλάζουν καθόλου οι τιμές των μεταβλητών του), είτε δυναμική (οι τιμές των μεταβλητών μπορεί να μεταβάλλονται μέσα σε ένα πεπερασμένο διάστημα, αλλά το σύστημα ουσιαστικά δεν αλλάζει δυναμική συμπεριφορά).

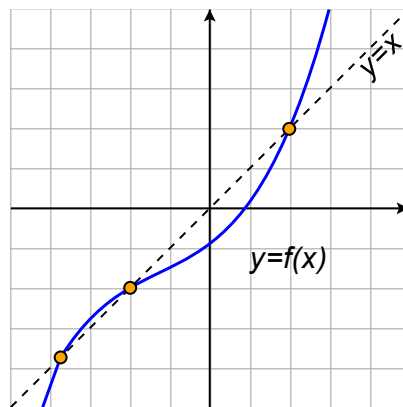
Στις δύο μεγάλες κατηγορίες ισορροπίας μπορούμε να αναφέρουμε μερικές συγκεκριμένες περιπτώσεις στην κάθε κατηγορία. Ο λόγος γι αυτό είναι ότι αυτά τα ειδικότερα είδη ισορροπίας εμφανίζονται στις εφαρμογές του πραγματικού κόσμου και ο αναγνώστης θα ήταν καλό να έχει μια αίσθηση του χαρακτήρα του κάθε είδους ισορροπίας.

3.1.1 Σταθερό Σημείο (Fixed Point)

Σε ένα σύστημα με N μεταβλητές το σύστημα αποκτά fixed point ισορροπία όταν οι τιμές των μεταβλητών λάβουν ένα συγκεκριμένο διάνυσμα τιμών που περιγράφεται από την σχέση $F(x^*)=x^*$, Όπου $F(x)$ μια διανυσματική συνάρτηση πολλών μεταβλητών που παριστάνει το μοντέλο του συστήματος που μελετούμε και x^* είναι εκείνο το διάνυσμα τιμών που από την στιγμή που πραγματοποιηθεί όλες οι επόμενες τιμές της συνάρτησης $F(x)$ θα είναι πάντα x^* .

Για $N=1$ (μια μεταβλητή) η έννοια του σταθερού σημείου μπορεί να παρουσιαστεί γεωμετρικά από το παρακάτω σχήμα:

Το fixed point (σταθερό σημείο) είναι ένα στοιχείο μιας συνάρτησης που αντιστοιχίζεται στον εαυτό του από τη συνάρτηση[4].



Σχήμα 3.1: fixed point

Η υποκατηγορία ισορροπίας fixed point ανήκει στην στατική ισορροπία.

3.1.2 Περιοδικός Κύκλος (Limit cycle)

Στο είδος αυτό της ισορροπίας οι τιμές των N μεταβλητών μεταβάλλονται με τον χρόνο αλλά μετά από κάποιο χρονικό διάστημα T οι τιμές αυτές επαναλαμβάνονται και πάλι οι ίδιες. Ουσιαστικά το σύστημα ταλαντώνεται περνώντας από καταστάσεις οι οποίες μετά από χρόνο T θα ξανα-επαναληφθούν.

Εποπτικά την έννοια της περιοδικότητας (για $N=1$) μπορούμε να την δούμε στο παρακάτω διάγραμμα.

Ένα περιοδικό φαινόμενο, π.χ. η τάση του ρεύματος, είναι μια πλήρης μεταβολή της έντασης του πεδίου ως συνάρτησης του χρόνου[5].

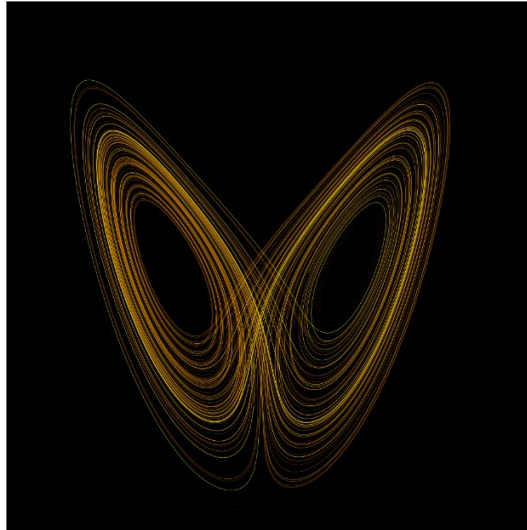


Σχήμα 3.2: περιοδικός κύκλος

Η περιοδική ισορροπία ανήκει στην ευρύτερη κατηγορία της δυναμικής ισορροπίας.

3.1.3 Παράξενος Ελκυστής (Strange Attractor)

Ο παράξενος ελκυστής (strange attractor) είναι ένα υποσύνολο του χώρου των καταστάσεων του συστήματος προς τις οποίες ένα σύστημα τείνει να εξελιχθεί, για μια μεγάλη ποικιλία συνθηκών εκκίνησης του συστήματος. Οι καταστάσεις του συστήματος που πλησιάζουν αρκετά τις τιμές του ελκυστήρα παραμένουν κοντά ακόμα και αν διαταραχθούν ελαφρώς[7]. Στην συνέχεια, θα αναφερθεί η έννοια του χάους η οποία παρατηρείται σε μη γραμμικά δυναμικά συστήματα και ορίζεται ως ιδιότητα του συστήματος ότι μικρές διαφορές στις αρχικές συνθήκες (όπως αυτές που οφείλονται σε σφάλματα στρογγυλοποίησης σε αριθμητικούς υπολογισμούς) αποδίδουν πολύ διαφορετικά αποτελέσματα για τα δυναμικά συστήματα, καθιστώντας τη μακροπρόθεσμη πρόβλεψη αδύνατη σε γενικές γραμμές. Αυτό συμβαίνει παρόλο που η μελλοντική συμπεριφορά των συστημάτων καθορίζεται πλήρως από τις αρχικές τους συνθήκες, χωρίς να εμπλέκονται τυχαίες παράμετροι[8]. Ένας ελκυστής ονομάζεται strange attractor (παράξενος ελκυστής) πολλές φορές όταν, η δυναμική του συστήματος που τον παράγει είναι χαοτική, αλλά υπάρχουν και παράξενοι ελκυστές οι οποίοι δεν έχουν χαοτική συμπεριφορά (είναι απλά «περίεργα» τοπολογικά αντικείμενα, π.χ. fractals). Στην περίπτωση όπου η δυναμική του συστήματος υπό εξέταση είναι χαοτική τότε οποιαδήποτε δυο αυθαίρετα κοντινά σημεία (δηλ. δύο κοντινές αρχικές καταστάσεις συστήματος) μετά από διάφορους αριθμούς επαναλήψεων θα καταλήξουν να απέχουν αυθαίρετα μακριά μεταξύ τους (δηλ. η τελική κατάσταση του συστήματος που προέρχεται από την πρώτη και δεύτερη αρχική κατάσταση θα είναι πολύ διαφορετική)[7].



Σχήμα 3.3: Μια γραφική παράσταση του παράξενου ελκυστήρα του Lorenz

Ο παράξενος ελκυστής είναι μια υποκατηγορία δυναμικής ισορροπίας.

3.2 Πρώτο μοντέλο

Το συγκεκριμένο μοντέλο όπως προαναφέρθηκε προσεγγίζει την αγορά μακροοικονομικά, οι αποφάσεις επηρεάζονται από την κατανάλωση βασισμένη στην τιμή και από την θεωρία της μάθησης μέσω πράξης. Επίσης, είναι μια καλή αρχή για την κατανόηση του lock-in. Αρχικά, είναι απαραίτητο να αναφερθούν οι παράγοντες οι οποίοι υπάρχουν στο μοντέλο. Οπότε, το μερίδιο της ζήτησης (D), που παρέχεται από το προϊόν i (S_i), εξαρτάται από τις τιμές των προϊόντων (P). Κατόπιν, υπάρχει η ευαισθησία της προσφοράς λόγω διαφοράς τιμής (μ) και ο χρόνος προσαρμογής της αγοράς (t_a) για να μεταβεί από τις ενδεικτικά μερίδια ($IndSh_i$) σε κανονικά μερίδια αγοράς (Sh_i). Τα ενδεικτικά μερίδια αγοράς μοντελοποιούν τις προσδοκίες των καταναλωτών ενώ τα κανονικά μοντελοποιούν τι εμφανίζεται στην πραγματικότητα κάτω από την επίδραση όλων των παραγόντων. Επιπροσθέτως, τα κόστη των προϊόντων εξαρτώνται από τα πάγια έξοδα (FC), από τα μεταβλητά έξοδα (VC), από τον αριθμό των προϊόντων που πωλούνται (πωλήσεις) και από των παράγοντα εκμάθησης (LF). Ο παράγοντας εκμάθησης αντιπροσωπεύει την εμπειρία ενός παραγωγού στην δημιουργία, παραγωγή και διάθεση του προϊόντος. Έπειτα, ο παράγοντας εκμάθησης μειώνεται όσο μεγαλώνει ο αριθμός των παραγόμενων προϊόντων (CL). Επίσης, η παράμετρος γ καθορίζει τον ρυθμό μάθησης της διαδικασίας της παραγωγής. Παρακάτω, θα αναλυθούν οι μαθηματικές εξισώσεις του μοντέλου.

Η προμήθεια του προϊόντος i (S_i) ισούται με το μερίδιο της ζήτησης :

$$S_i = Sh_i \times D \quad (3.1)$$

Το μερίδιο της ζήτησης αναπροσαρμόζεται με την μεταβολή σε ενδεικτικά μερίδια :

$$Sh_i = \frac{Dt \times (IndSh_i - Sh_{i-1})}{t_a} + Sh_{i-1} \quad (3.2)$$

Τα ενδεικτικά μερίδια στην συνέχεια καθορίζονται από την πολυωνυμική συνάρτηση logit των τιμών των προϊόντων. Αυτή η συνάρτηση εξισώνει τις τιμές των προϊόντων ενώ, η παράμετρος mu υποδηλώνει την ευαισθησία των επιλογών των καταναλωτών στις διαφοροποιήσεις των τιμών.

$$IndSh_i = \frac{e^{-mu \times P_i}}{e^{-mu \times P_1} + e^{-mu \times P_2}} \quad (3.3)$$

Παρακάτω, οι τιμές εξαρτώνται από πάγια, μεταβαλλόμενα έξοδα και από τον παράγοντα εκμάθησης.

$$P_i = \frac{LF_i \times (FC_i + S_i \times VCI)}{S_i} \quad (3.4)$$

Επιπροσθέτως, ο παράγοντας εκμάθησης μειώνεται από την μαζική παραγωγή με βάση την αρχή της μάθησης μέσω της πράξης. Οπότε, όσο περισσότερα προϊόντα παράγονται, τόσο πιο χαμηλή είναι η τιμή ανά μονάδα. Η παράμετρος gamma καθορίζει την μείωση του κόστους ανά διπλασιασμό της παραγωγής.

$$LF_i = \frac{CL_i}{CL_{i-in}} \frac{-\log_{10}(gamma)}{\log_{10}(2)} \quad (3.5)$$

$$CL_i = Dt \times S_i + CL_{i-1} \quad (3.6)$$

Τέλος, αν δυο προϊόντα έχουν ίδια οικονομικά χαρακτηριστικά και ίδια ζήτηση, η ζήτηση θα παραμείνει ίδια στο συγκεκριμένο μοντέλο. Οπότε, θα προστεθεί ένας στοχαστικός όρος ο οποίος με ευκολία θα δημιουργήσει lock-in και θα προσομοιώνει αναπάντεχα γεγονότα που συνέβησαν στην αγορά. Ο συγκεκριμένος όρος, $N(0, \sigma)$, είναι μια κανονική κατανομή με μέσο όρο μηδέν και τυπική απόκλιση ίση με σ . Επομένως, τώρα μπορεί να επιτευχθεί το lock-in εφόσον είναι αρκετά μεγάλο το mu και η αγορά επηρεάζεται από στοχαστικά γεγονότα.

$$S_1 = \min(Sh_1 \times D + N(0, \sigma), D) \quad (3.7)$$

$$S_2 = D - S_1 \quad (3.8)$$

3.3 Τελικό μοντέλο

Το τελικό μοντέλο προσεγγίζει την επιλογή των καταναλωτών από μικροοικονομική πλευρά, δηλαδή μελετάει τους λόγους τους οποίους επηρέασαν τον κάθε καταναλωτή ξεχωριστά να πάρει την κάθε απόφαση και όχι τους καταναλωτές σαν σύνολο. Αυτό επιτυγχάνεται διαχωρίζοντας τις διάφορες ανάγκες τις οποίες έχουν οι καταναλωτές και ικανοποιούνται σε κάποιον βαθμό. Ο βαθμός της

ικανοποίησης της ανάγκης i (LNS_i) αναπαριστάται από έναν δείκτη ο οποίος κυμαίνεται ανάμεσα στο μηδέν (καθόλου ικανοποιημένος) και στο ένα (εντελώς ικανοποιημένος). Στην συνέχεια, η ολική ικανοποίηση υπολογίζεται από τον σταθμισμένο μέσο όρο των επί μέρους αναγκών. Έπειτα, υπάρχουν δύο διαστάσεις αναγνωρισμένων νοητικών διαδικασιών. Η πρώτη διάσταση είναι η νοητική προσπάθεια η οποία περιλαμβάνεται στην διαδικασία. Ειδικότερα, η λογική συμπεριφορά συνδέεται με υψηλό κίνητρο της σκέψης, ενώ η αυτόματη επεξεργασία των ερεθισμάτων είναι πιο πιθανή όταν το κίνητρο αυτό είναι χαμηλό. Στην συνέχεια, η δεύτερη διάσταση σχετίζεται με τον κοινωνικό ή προσωπικό προσανατολισμό της επεξεργασίας. Η θεωρία της κοινωνικής σύγκρισης αναφέρει ότι η ορμή της σύγκρισης των απόψεων και των ικανοτήτων ενός ανθρώπου με τους υπόλοιπους, μεγαλώνει όσο πιο αβέβαιος είναι για τις απόψεις και τις ικανότητές του. Το αποτέλεσμα για την συμπεριφορά των καταναλωτών είναι ότι, όσο πιο αβέβαιος είναι για το προϊόν το οποίο έχει επιλέξει τόσο πιο εύκολα θα συγκρίνει την καταναλωτική συμπεριφορά άλλων με παρόμοιες απόψεις και ικανότητες. Αντιθέτως, όταν ο καταναλωτής είναι σίγουρος, κυριαρχεί η ατομική επεξεργασία των επιθυμιών, χωρίς ιδιαίτερη επιρροή των καταναλωτικών συμπεριφορών των υπολοίπων. Στην συνέχεια, αφότου έγινε μια γενική περιγραφή της λογικής των εξισώσεων θα αναλυθούν οι ίδιες εξισώσεις. Κατόπιν, όλες οι εξισώσεις διαχωρίζουν τον βαθμό ικανοποίησης του κάθε καταναλωτή και για το προϊόν Α και για το προϊόν Β, έτσι ώστε να καταστεί δυνατή η διάκριση του βαθμού σιγουριάς κάθε καταναλωτή. Αρχικά, ένας παράγοντας που επηρεάζει την ικανοποίηση του καταναλωτή είναι η ταυτότητά του. Το επίπεδο της ικανοποίησης για την ανάγκη της ταυτότητας εξαρτάται από τον αριθμό των γειτόνων οι οποίοι έχουν καταναλώσει το ίδιο προϊόν με τον συγκεκριμένο καταναλωτή, αυτή είναι η ικανοποίηση στην ανάγκη να ανήκουν. Επίσης, υποθέτουμε ότι το LNS_i αυξάνεται εκθετικά με τον αριθμό των γειτόνων οι οποίοι επιλέγουν το ίδιο προϊόν με τον συγκεκριμένο καταναλωτή. Συμπερασματικά, όσο περισσότεροι οι γείτονες (x_N) οι οποίοι καταναλώνουν το ίδιο προϊόν, τόσο πιο υψηλή είναι η ικανοποίηση του συγκεκριμένου καταναλωτή.

$$\text{Για το προϊόν A} \quad LNS_{1A} = 1 - \Sigma x_N \quad (3.9)$$

$$\text{Για το προϊόν B} \quad LNS_{1B} = \frac{\Sigma x_N}{8} \quad (3.10)$$

Έπειτα, ένας ακόμα παράγοντας είναι η προσωπική προτίμηση το οποίο είναι διαφορετικό χαρακτηριστικό κάθε καταναλωτή. Οπότε, το επίπεδο ικανοποίησης για την προσωπική προτίμηση LNS_2 είναι ίσο με την προσωπική προτίμηση που έχει ο κάθε καταναλωτής στο συγκεκριμένο προϊόν (β).

$$\text{Για το προϊόν A} \quad LNS_{2A} = \beta_A \quad (3.11)$$

$$\text{Για το προϊόν B} \quad LNS_{2B} = \beta_B \quad (3.12)$$

Επίσης, ο ελεύθερος χρόνος (B) συσχετίζεται με την τιμή των προϊόντων υποθέτοντας ότι, τα προϊόντα με χαμηλότερη τιμή απαιτούν λιγότερο χρόνο εργασίας για να αποκτηθούν τα χρήματα. Η αντιστοιχία αυτή είναι προσεγγιστική αλλά δημιουργεί την ευκαιρία να ισορροπήσει ο ελεύθερος χρόνος με τον χρόνο εργασίας.

$$\begin{aligned} \text{Για το προϊόν A} \quad LNS_{3A} &= \frac{B}{P_A} \end{aligned} \quad (3.13)$$

$$\begin{aligned} \text{Για το προϊόν B} \quad LNS_{3B} &= \frac{B}{P_B} \end{aligned} \quad (3.14)$$

Επιπροσθέτως, η ευαισθησία στο περιβάλλον (μόλυνση του περιβάλλοντος λόγω της παραγωγής και κατανάλωσης ενός προϊόντος) συσχετίζεται με τον βαθμό μόλυνσης του περιβάλλοντος. Επομένως, κάθε προϊόν i υποθέτεται ότι συνεισφέρει λ_i μονάδες μόλυνσης ανάλογα με τον αριθμό των γειτόνων οι οποίοι κατανάλωσαν το συγκεκριμένο προϊόν (N_i) και η συγκέντρωση της μόλυνσης (C) μειώνεται σε κάθε βήμα με ρυθμό μ .

$$C = C_{-1} \times (1 - \mu) + \lambda_A \times N_A + \lambda_B \times N_B \quad (3.15)$$

Συμπληρωματικά, η ατομική ευαισθησία στην μόλυνση (α) καθορίζει το ατομικό επίπεδο ικανοποίησης και συσχετίζεται με το επίπεδο της συγκέντρωσης (C).

$$\begin{aligned} \text{Για το προϊόν A} \quad LNS_{4A} &= 1 - e^{-\frac{\alpha}{C_{-1} \times (1-\mu) + \lambda_A}} \end{aligned} \quad (3.16)$$

$$\begin{aligned} \text{Για το προϊόν B} \quad LNS_{4B} &= 1 - e^{-\frac{\alpha}{C_{-1} \times (1-\mu) + \lambda_B}} \end{aligned} \quad (3.17)$$

Κατόπιν, το συνολικό επίπεδο ικανοποίησης ενός καταναλωτή είναι εξίσωση τύπου Cobb Douglas κατά την οποία κάθε παράγοντας ικανοποίησης σταθμίζεται από διαφορετικό γ_{need} , η επιλογή αυτού του τύπου για το LNS υποθέτει ότι οι επιμέρους βαθμοί ικανοποίησης δομούν τον συνολικό βαθμό ικανοποίησης.

Για το προϊόν A

$$\begin{aligned} LNS_A &= LNS_{1A}^{\gamma_{need1}} \times LNS_{2A}^{\gamma_{need2}} \times LNS_{3A}^{\gamma_{need3}} \\ &\quad \times LNS_{4A}^{1 - \gamma_{need1} - \gamma_{need2} - \gamma_{need3}} \end{aligned} \quad (3.18)$$

Για το προϊόν B

$$\begin{aligned} LNS_B &= LNS_{1B}^{\gamma_{need1}} \times LNS_{2B}^{\gamma_{need2}} \times LNS_{3B}^{\gamma_{need3}} \\ &\quad \times LNS_{4B}^{1 - \gamma_{need1} - \gamma_{need2} - \gamma_{need3}} \end{aligned} \quad (3.19)$$

Έπειτα, παρόλο που το επίπεδο ικανοποίησης για ένα προϊόν είναι μεγαλύτερο από το επίπεδο ικανοποίησης ένα άλλο προϊόν, μπορεί η διαφορά να είναι μικρή, με αποτέλεσμα ο καταναλωτής να μην είναι σίγουρος για την απόφαση του. Οπότε, σε περίπτωση που η διαφορά μεταξύ της ικανοποίησης ανάμεσα στα δύο προϊόντα είναι κάτω από ένα κατώφλι (LNS_{tol}) τότε οι καταναλωτές θα πράξουν με βάση την κοινωνική σύγκριση, δηλαδή θα αντιγράψουν την συμπεριφορά των γειτόνων με παρόμοια χαρακτηριστικά. Συμπληρωματικά, αυτά τα χαρακτηριστικά στην συγκεκριμένη περίπτωση είναι η οικονομική τους κατάσταση (B). Κατόπιν, οι καταναλωτές θα καταναλώσουν προϊόντα τα οποία έχουν

μεγαλύτερο αριθμό γειτόνων οι οποίοι τα κατανάλωσαν (x_N), με παρόμοια οικονομική κατάσταση, στο προηγούμενο βήμα. Επίσης, όσο μεγαλύτερο είναι το αποδεκτό οικονομικό εύρος (B_{tol}), τόσο περισσότεροι γείτονες θεωρούνται παρόμοιας οικονομικής κατάστασης. Τέλος, στην εξίσωση υπάρχει η ενδιάμεση τιμή (median) της οικονομικής κατάστασης των γειτόνων ($B_{neighbours}$), με σκοπό την σύγκριση της οικονομικής κατάστασης του καταναλωτή με τους γείτονές του.

Σε περίπτωση που $LNS_A < LNS_{tol}$ ή $LNS_B < LNS_{tol}$

θα επιλεχτεί το προϊόν A αν

$$x_{NA} \geq x_{NB}$$

Και

$$B \times (1 - B_{tol}) \leq B_{neighbours} \leq B \times (1 + B_{tol})$$

Αλλιώς θα επιλεχτεί το προϊόν B

(3.20)

Κεφάλαιο 4ο: Υπολογιστικό Μοντέλο

4.1 Υλοποίηση Πρώτου Μοντέλου

```

1. package com.mycompany.lock.in_dynamics;
2.
3. import java.io.File;
4. import java.io.FileWriter;
5. import java.io.IOException;
6. import static java.lang.Math.log;
7. import java.util.*;
8.
9. public class FirstModel {
10.     int k;
11.     float mu ;
12.     float gamma ;
13.     float sigma ;
14.     float t ;
15.     int ta ;
16.     float FCa ;
17.     float VCa ;
18.     float FCb ;
19.     float VCb ;
20.     float D ;
21.     float Sa ;
22.     float Sb ;
23.     float Sha ;
24.     float Shb ;
25.     float IndSha ;
26.     float IndShb ;
27.     float Pa ;
28.     float Pb ;
29.     float LFa ;
30.     float Lfb ;
31.     float Cla ;
32.     float Clb ;
33.     float Cllock_in ;
34.     float Exp ;
35.     float E ;
36.     float Dt ;
37.     float UN;
38.     float nxt;
39.     Random ran = new Random();
40.     String name ;
41.     float k1 ;
42.     int counter=0;
43.
44.
45.     public FirstModel (String name , float mu , float gamma , float sigma
46.     {
47.         this.name = name;
48.         this.mu = mu;
49.         this.sigma = sigma;
50.         this.gamma = gamma;
51.         this.ta = ta;
52.         this.FCa = FCa;
53.         this.VCa = VCa;
54.         this.FCb = FCb;
55.         this.VCb = VCb;
56.         this.D = D;
57.         this.Sha = Sha;
58.         this.Shb = Shb;
59.         this.Cla = Cla;
60.         this.Clb = Clb;
61.         this.Dt = Dt;
62.     }
63.
64.
65.     public void programm(float a ) throws IOException
66.     {
67.
68.     try
69.     {
70.         File myObj = new File(name + ".txt");
71.         if (myObj.createNewFile())
72.         {
73.             System.out.println("File created: " + myObj.getName());
74.         }
75.         else
76.         {
77.             System.out.println("File already exists.");
78.         }
79.     }
80.     catch (IOException e)
81.     {
82.         System.out.println("An error occurred.");
83.         e.printStackTrace();
84.     }
85.
86.     FileWriter myWriter = new FileWriter(name + ".txt");
87.
88.     for (t=1 ; t <=4.267; t+=Dt )
89.     {
90.         if(t==1)
91.         {
92.             counter=1;
93.
94.
95.
96.
97.
98.
99.
100.
101.         }
102.         else
103.         {
104.             counter ++;
105.
106.
107.
108.
109.
110.
111.
112.
113.
114.
115.
116.
117.
118.
119.
120.
121.
122.
123.
124.
125.
126.
127.
128.
129.
130.
131.
132.
133.
134.
135.
136.
137.
138.
139.
140.
141.
142.
143.
144.
145.
146.
147.
148.
149.
150.
151.
152.
153.
154.
155.
156.
157.
158.
159.
160.
161.
162.
163.
164.
165.
166.
167.
168.
169.
170.
171.
172.
173.
174.
175.
176.
177.
178.
179.
180.
181.
182.
183.
184.
185.
186.
187.
188.
189.
190.
191.
192.
193.
194.
195.
196.
197.
198.
199.
200.
201.
202.
203.
204.
205.
206.
207.
208.
209.
210.
211.
212.
213.
214.
215.
216.
217.
218.
219.
220.
221.
222.
223.
224.
225.
226.
227.
228.
229.
230.
231.
232.
233.
234.
235.
236.
237.
238.
239.
240.
241.
242.
243.
244.
245.
246.
247.
248.
249.
250.
251.
252.
253.
254.
255.
256.
257.
258.
259.
260.
261.
262.
263.
264.
265.
266.
267.
268.
269.
270.
271.
272.
273.
274.
275.
276.
277.
278.
279.
280.
281.
282.
283.
284.
285.
286.
287.
288.
289.
290.
291.
292.
293.
294.
295.
296.
297.
298.
299.
300.
301.
302.
303.
304.
305.
306.
307.
308.
309.
310.
311.
312.
313.
314.
315.
316.
317.
318.
319.
320.
321.
322.
323.
324.
325.
326.
327.
328.
329.
330.
331.
332.
333.
334.
335.
336.
337.
338.
339.
340.
341.
342.
343.
344.
345.
346.
347.
348.
349.
350.
351.
352.
353.
354.
355.
356.
357.
358.
359.
360.
361.
362.
363.
364.
365.
366.
367.
368.
369.
370.
371.
372.
373.
374.
375.
376.
377.
378.
379.
380.
381.
382.
383.
384.
385.
386.
387.
388.
389.
390.
391.
392.
393.
394.
395.
396.
397.
398.
399.
400.
401.
402.
403.
404.
405.
406.
407.
408.
409.
410.
411.
412.
413.
414.
415.
416.
417.
418.
419.
420.
421.
422.
423.
424.
425.
426.
427.
428.
429.
430.
431.
432.
433.
434.
435.
436.
437.
438.
439.
440.
441.
442.
443.
444.
445.
446.
447.
448.
449.
450.
451.
452.
453.
454.
455.
456.
457.
458.
459.
460.
461.
462.
463.
464.
465.
466.
467.
468.
469.
470.
471.
472.
473.
474.
475.
476.
477.
478.
479.
480.
481.
482.
483.
484.
485.
486.
487.
488.
489.
490.
491.
492.
493.
494.
495.
496.
497.
498.
499.
500.
501.
502.
503.
504.
505.
506.
507.
508.
509.
510.
511.
512.
513.
514.
515.
516.
517.
518.
519.
520.
521.
522.
523.
524.
525.
526.
527.
528.
529.
530.
531.
532.
533.
534.
535.
536.
537.
538.
539.
540.
541.
542.
543.
544.
545.
546.
547.
548.
549.
550.
551.
552.
553.
554.
555.
556.
557.
558.
559.
560.
561.
562.
563.
564.
565.
566.
567.
568.
569.
570.
571.
572.
573.
574.
575.
576.
577.
578.
579.
580.
581.
582.
583.
584.
585.
586.
587.
588.
589.
590.
591.
592.
593.
594.
595.
596.
597.
598.
599.
600.
601.
602.
603.
604.
605.
606.
607.
608.
609.
610.
611.
612.
613.
614.
615.
616.
617.
618.
619.
620.
621.
622.
623.
624.
625.
626.
627.
628.
629.
630.
631.
632.
633.
634.
635.
636.
637.
638.
639.
640.
641.
642.
643.
644.
645.
646.
647.
648.
649.
650.
651.
652.
653.
654.
655.
656.
657.
658.
659.
660.
661.
662.
663.
664.
665.
666.
667.
668.
669.
670.
671.
672.
673.
674.
675.
676.
677.
678.
679.
680.
681.
682.
683.
684.
685.
686.
687.
688.
689.
690.
691.
692.
693.
694.
695.
696.
697.
698.
699.
700.
701.
702.
703.
704.
705.
706.
707.
708.
709.
710.
711.
712.
713.
714.
715.
716.
717.
718.
719.
720.
721.
722.
723.
724.
725.
726.
727.
728.
729.
730.
731.
732.
733.
734.
735.
736.
737.
738.
739.
740.
741.
742.
743.
744.
745.
746.
747.
748.
749.
750.
751.
752.
753.
754.
755.
756.
757.
758.
759.
760.
761.
762.
763.
764.
765.
766.
767.
768.
769.
770.
771.
772.
773.
774.
775.
776.
777.
778.
779.
780.
781.
782.
783.
784.
785.
786.
787.
788.
789.
790.
791.
792.
793.
794.
795.
796.
797.
798.
799.
800.
801.
802.
803.
804.
805.
806.
807.
808.
809.
810.
811.
812.
813.
814.
815.
816.
817.
818.
819.
820.
821.
822.
823.
824.
825.
826.
827.
828.
829.
830.
831.
832.
833.
834.
835.
836.
837.
838.
839.
840.
841.
842.
843.
844.
845.
846.
847.
848.
849.
850.
851.
852.
853.
854.
855.
856.
857.
858.
859.
860.
861.
862.
863.
864.
865.
866.
867.
868.
869.
870.
871.
872.
873.
874.
875.
876.
877.
878.
879.
880.
881.
882.
883.
884.
885.
886.
887.
888.
889.
890.
891.
892.
893.
894.
895.
896.
897.
898.
899.
900.
901.
902.
903.
904.
905.
906.
907.
908.
909.
910.
911.
912.
913.
914.
915.
916.
917.
918.
919.
920.
921.
922.
923.
924.
925.
926.
927.
928.
929.
930.
931.
932.
933.
934.
935.
936.
937.
938.
939.
940.
941.
942.
943.
944.
945.
946.
947.
948.
949.
950.
951.
952.
953.
954.
955.
956.
957.
958.
959.
960.
961.
962.
963.
964.
965.
966.
967.
968.
969.
970.
971.
972.
973.
974.
975.
976.
977.
978.
979.
980.
981.
982.
983.
984.
985.
986.
987.
988.
989.
990.
991.
992.
993.
994.
995.
996.
997.
998.
999.
1000.

```

Σχήμα 4.1: Java code

4.1.1 Περιγραφή του κώδικα

Όπως προαναφέρθηκε το πρώτο μοντέλο είναι υλοποιημένο στην γλώσσα προγραμματισμού Java. Αυτό συνέβη διότι, δεν έχει εισαχθεί ακόμα η έννοια των πρακτόρων στο μοντέλο οπότε δεν χρειάζεται εικονική αναπαράσταση του χώρου. Επίσης, είναι μια αρκετά γρήγορη γλώσσα, που επιτρέπει εύκολη

εξαγωγή δεδομένων. Στο παράρτημα A, καταγράφεται ο κώδικας του πρώτου μοντέλου, με αριθμηση σε κάθε σειρά για να γίνει δυνατή η επεξήγησή του. Αρχικά, στις σειρές 3-7 εισάγονται οι απαραίτητες βιβλιοθήκες για την υλοποίηση του προγράμματος, οι πρώτες τρεις χρησιμεύουν για την εξαγωγή των δεδομένων και οι υπόλοιπες δύο για να γίνουν οι μαθηματικές πράξεις. Έπειτα, στην σειρά 9 δημιουργείται μια κλάση με όνομα FirstModel και στις σειρές 10-42 γίνεται η αρχικοποίηση των μεταβλητών του μοντέλου. Παρακάτω, στις σειρές 45-62 γίνεται η δήλωση ενός constructor και των μεταβλητών του, δημιουργείται μόνο ένας πλήρης constructor διότι σε κάθε προσομοίωση είναι απαραίτητες όλες οι μεταβλητές για να μπορέσει να δουλέψει το σύστημα. Επίσης, στην σειρά 66 δημιουργείται μια μέθοδος που δέχεται ως είσοδο μια μεταβλητή τύπου float και "πετάει" εξαίρεση διότι γίνεται χρήση του filewriter. Κατόπιν, στις σειρές 69-85 είναι μια ρουτίνα τύπου try-catch η οποία χρησιμεύει στο να "πιάσει" την εξαίρεση σε περίπτωση σφάλματος και κάνει και άλλες διεργασίες. Πιο συγκεκριμένα, στην σειρά 71 δημιουργείται ένα αντικείμενο της κλάσης File το οποίο έχει σαν όρισμα την μεταβλητή name προσθέτοντας το text ".txt" έτσι ώστε να δημιουργηθεί ένα txt αρχείο. Έπειτα, στις σειρές 72-76 ελέγχεται αν υπάρχει άλλο αρχείο με το ίδιο όνομα και σε περίπτωση που δεν υπάρχει εμφανίζεται στην οθόνη 'file created' και το όνομα του αρχείου, αλλιώς εμφανίζεται 'file already exists'. Στη συνέχεια, στις σειρές 81-85 αν υπάρξει κάποιο σφάλμα στις προηγούμενες σειρές εμφανίζεται στην οθόνη 'an error occurred' και το stack trace. Επίσης, στην σειρά 87 δημιουργείται ένα αντικείμενο της κλάσης FileWriter με όρισμα την μεταβλητή name συν το κείμενο ".txt". Κατόπιν, στις σειρές 90-159 υπάρχει η υλοποίηση των μαθηματικών εξισώσεων του μοντέλου και η εξαγωγή των αποτελεσμάτων. Πιο συγκεκριμένα, υπάρχει μια ρουτίνα επανάληψης η οποία κάνει 100 επαναλήψεις. Έπειτα, στις σειρές 90-99 αρχικοποιείται το counter εαν το t ισούται με ένα, ή αλλιώς θα αυξηθεί κατά ένα. Παρακάτω, στις σειρές 101-108 ελέγχεται αν το CLa είναι μεγαλύτερο η μικρότερο από το CLb, για να οριστεί η τιμή του CLock_in, το οποίο πρέπει να έχει τιμή ίση, με τη μεγαλύτερη από αυτές τις δύο τις μεταβλητές. Επιπροσθέτως, στις σειρές 111-149 εισάγονται οι μαθηματικές εξισώσεις του μοντέλου. Επίσης, πρέπει να σημειωθεί ότι για ύψωση σε εκθέτη χρησιμοποιήθηκε η κλάση Math όπως και για την υλοποίηση του στοχαστικού όρου (σειρά 137, υπολογισμός κανονικής κατανομής) χρησιμοποιήθηκε το ran.nextGaussian. Κατόπιν, στις σειρές 152,153 εμφανίζουν στην οθόνη τις τιμές των Sa, Sb σε κάθε επανάληψη, για τη διευκόλυνση του testing. Στην συνέχεια, στην σειρά 154 προστίθεται μια νέα σειρά στο αρχείο που έχει δημιουργηθεί, αναγράφοντας αρχικά την τιμή του counter, στην συνέχεια την τιμή της εκάστοτε μεταβλητής που εξετάζεται, τις τιμές των Sa, Sb και τέλος προστίθεται καινούρια σειρά. Επομένως, όλα αυτά γίνονται μέσα στην ρουτίνα for για εκατό φορές στην συγκεκριμένη περίπτωση και στην σειρά 155 κλίνει η ρουτίνα. Παρακάτω, στις σειρές 156,157 κλείνει το συγκεκριμένο αρχείο. Επιπροσθέτως, στις σειρές 162-177 υπάρχει η κλάση main η οποία είναι υπεύθυνη για την λειτουργία του προγράμματος. Αρχικά, στις σειρές 165-167 γίνονται οι δηλώσεις μεταβλητών και αρχικοποίηση του constructor FirstModel. Έπειτα, ακολουθεί μια ρουτίνα επανάληψης με σκοπό την προσομοίωση διαφόρων τιμών μιας μεταβλητής και εξαγωγή των δεδομένων. Πιο αναλυτικά, στην σειρά 170 η μεταβλητή k1 παίρνει την τιμή της εκάστοτε μεταβλητής, εξάγοντας την τιμή από τον constructor (στην συγκεκριμένη περίπτωση pro.sigma) και εισάγονται οι τιμές που είναι επιθυμητές για προσομοίωση της μεταβλητής αυτής. Επιπροσθέτως, στην σειρά 172 αλλάζει η τιμή της μεταβλητής name στο όνομα της μεταβλητής που εξετάζεται (sigmaS), συν την τιμή της μεταβλητής η οποία έχει μετατραπεί σε String, έτσι ώστε να μπορεί να εισαχθεί στο όνομα του αρχείου. Τέλος, στη σειρά 173 καλείται η μέθοδος program με μεταβλητή εισόδου την k1. Συμπερασματικά, κάθε φορά που προσομοιώνεται μια τιμή, πρέπει στην μέθοδο program η μεταβλητή εισόδου να αντικαθίσταται με την εκάστοτε μεταβλητή. Έπειτα, στην main στην σειρά 170 να αλλάζει το k1 στην τιμή της μεταβλητής, όπως και το εύρος του, στο επιθυμητό εύρος. Επιπρόσθετα, στη σειρά 172 είναι απαραίτητο να αλλάζει

το όνομα. Μόνο με όλες αυτές τις αλλαγές μπορούν να προσομοιωθούν οι διάφορες τιμές που παίρνουν οι μεταβλητές.

4.1.2 Testing

Το συγκεκριμένο πρόγραμμα μπορεί να φαίνεται να μην έχει ιδιαίτερη πολυπλοκότητα στην υλοποίηση όμως πρέπει να είναι πολύ καλά συγχρονισμένο για να δουλέψει. Οπότε, μετά την επίλυση των αρχικών σφαλμάτων, μέσω του debugging, πρέπει να ελεγχτεί και η εγκυρότητα των αποτελεσμάτων. Αρχικά, επειδή τα αποτελέσματα αναφέρονται στην πρόβλεψη της συμπεριφοράς 961 καταναλωτών, στην προκειμένη περίπτωση, δεν είναι δυνατόν η απόλυτη επιβεβαίωση αυτών. Οπότε, τα αποτελέσματα ελέγχθηκαν με βάση την κοινή λογική αν βγάζουν νόημα και αν είναι σχετικά σταθερά με ίδιες αρχικές συνθήκες. Έπειτα, συγκρίθηκαν με αποτελέσματα που εξαχθήκαν σε παρόμοιες έρευνες [2].

4.1.3 Εξαγωγή δεδομένων

Όπως έχει προαναφερθεί, τα δεδομένα εξάχθηκαν αρχικά με την χρήση των βιβλιοθηκών `java.io.File`, `java.io.FileWriter` οι οποίες έγραψαν τα δεδομένα σε αρχεία `.txt`. Έπειτα, η σύνταξη των `.txt` αρχείων έγινε έτσι ώστε να είναι εύκολα μετατρέψιμα σε αρχεία τύπου `csv`, με σκοπό να γίνουν διαγράμματα στο Microsoft Excel. Στην συνέχεια, το Excel αποδείχθηκε λάθος εργαλείο για μια τέτοια δουλειά διότι είναι αργό με τόσο μεγάλο φορτίο δεδομένων και δεν είναι καθόλου προσαρμόσιμο. Τέλος, τα αρχεία `csv` εισάχθηκαν στο πρόγραμμα R όπου και τελικά έγιναν τα διαγράμματα.

4.2 Υλοποίηση Τελικού Μοντέλου

```

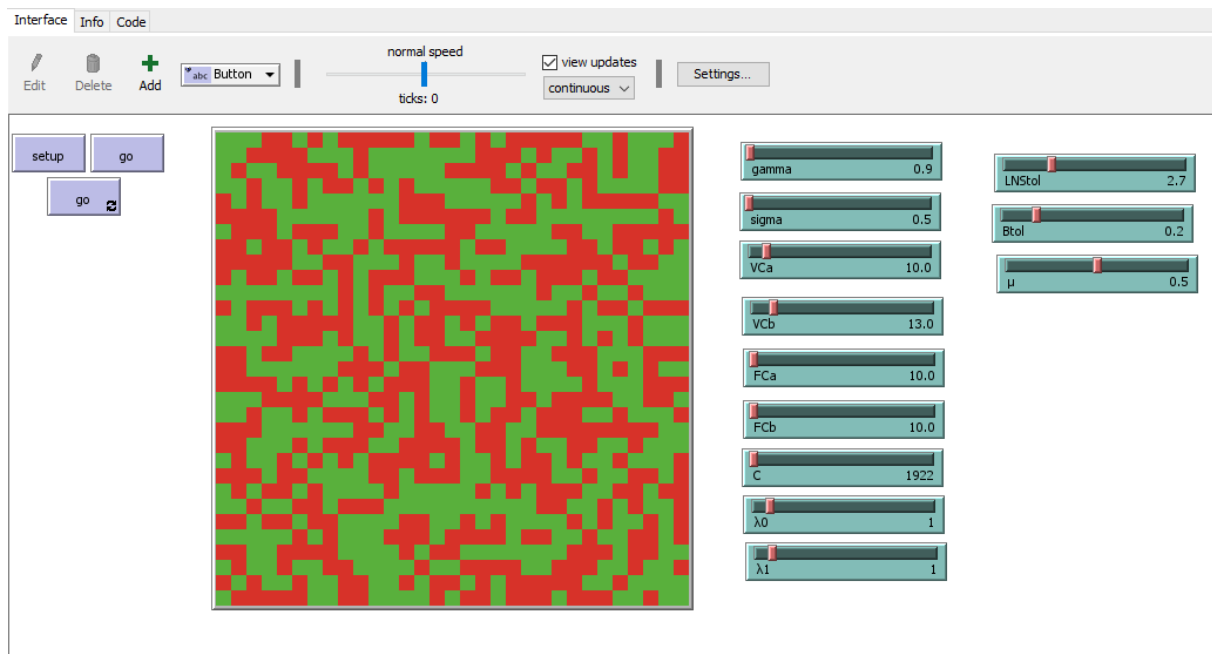
1.  globals [Dt Sha D Sa Sb Shb IndSha IndShb Pa Pb |
2.  patches-own[x LNSa LNS1a LNS2a LNS3a LNS4a LNSb LN
3.
4.  to no-op
5.  end
6.
7.
8.  to setup
9.  clear-all
10. ;global variables initialization
11. set Dt 0.1
12. set Sha 0.5
13. set Shb 1 - Sha
14. set Cla 1
15. set Clb 1
16.
17. ;red product a , green product b
18. ;patches-own variables initialization
19. ask patches [
20.
21.   ifelse ((random-float 1.0) <= Sha) [set pcolor 94.
22.   ifelse (pcolor = red)[set x 0][set x 1] 95.
23.   ;set B (40 + random-float 40) 96.
24.   set B (random-normal 50 5) 97.
25.   set a (random-normal 5 0.5) 98.
26.   set β0 (0.2 + random-float 0.1) 99.
27.   set β1 (0.2 + random-float 0.3) 100.
28.   set GAMMA1 (0.2 + random-float 0.1) 101.
29.   set GAMMA2 (0.2 + random-float 0.1) 102.
30.   set GAMMA3 (0.2 + random-float 0.1) 103.
31. ] 104.
32.  ] 105.
33.  ] 106.
34.  ] 107.
35.  ] 108.
36.  ] 109.
37.  ] 110.
38.  ] 111.
39.  ] 112.
40.  ] 113.
41.  ] 114.
42.  ] 115.
43.  ] 116.
44.  ] 117.
45.  ] 118.
46.  ] 119.
47.  ] 120.
48.  ] 121.
49.  ] 122.
50.  ] 123.
51.  ] 124.
52.  ] 125.
53.  ] 126.
54.  ] 127.
55.  ] 128.
56.  ] 129.
57.  ] 130.
58.  ] 131.
59.  ] 132.
60.  ] 133.
61.  ] 134.
62.  ] 135.
63.  ] 136.
64.  ] 137.
65.  ] 138.
66.  ] 139.
67.  ] 140.
68.  ] 141.
69.  ] 142.
70.  ] 143.
71.  ] 144.
72.  ] 145.
73.  ] 146.
74.  ] 147.
75.  ] 148.
76.  ] 149.
77.  ] 150.
78.  ]
79.  ]

```

Σχήμα 4.2: NetLogo code

4.2.1 Περιγραφή του κώδικα

Στο τελικό μοντέλο χρησιμοποιήθηκε η γλώσσα NetLogo, η οποία είναι κατάλληλη για υλοποιήσεις τέτοιου είδους μοντέλων. Στο παράρτημα Β καταγράφεται ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση του μοντέλου. Αρχικά, στην σειρά 1 γίνονται οι δηλώσεις των global μεταβλητών οι οποίες είναι μεταβλητές που είναι ίδιες για όλο το σύστημα. Έπειτα, στην σειρά 2 γίνονται οι δηλώσεις των μεταβλητών οι οποίες θα είναι μεταβλητές που θα έχουν διαφορετική τιμή για κάθε καταναλωτή. Κατόπιν, στις σειρές 4, 5 δηλώνεται ένα κενό procedure (διαδικασία), το οποίο θα βοηθήσει στην ομαλή ροή του μοντέλου, όπως θα αναλυθεί και παρακάτω. Επίσης, στις σειρές 8-38 βρίσκεται το procedure στο οποίο γίνεται η αρχικοποίηση του μοντέλου. Αρχικά, στην σειρά 9 καθαρίζονται πιθανώς παλαιότερα αποθηκευμένα μοντέλα και στις σειρές 11-15 γίνονται οι αρχικοποιήσεις των global μεταβλητών. Εδώ πρέπει να σημειωθεί ότι, η NetLogo επιτρέπει την αρχικοποίηση global μεταβλητών και μέσω sliders, το οποίο είναι απαραίτητο να αρχικοποιηθούν με αυτόν τον τρόπο, αν υπάρχει η επιθυμία να γίνει αυτόματα η προσομοίωση των διαφόρων τιμών τους. Στο σχήμα (4.1) στα δεξιά φαίνονται οι sliders οι οποίοι κάνουν την αρχικοποίηση εκτός κώδικα.



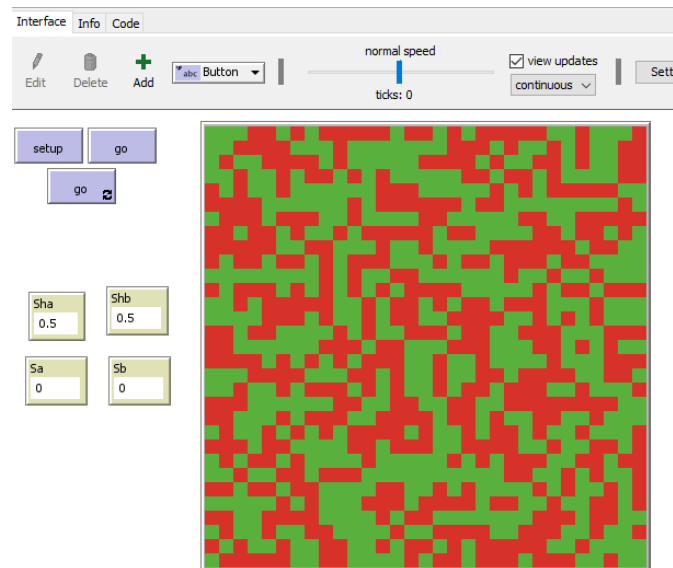
Σχήμα 4.3: Sliders

Επιπροσθέτως, στην σειρά 19 βρίσκεται μια εντολή η οποία ζητάει τα patches, τα οποία είναι στην προκειμένη περίπτωση η καταναλωτές, να κάνουν κάποιες πράξεις. Κατόπιν, στην σειρά 21 μέσω μιας ρουτίνας ifelse θέτουμε τα χρώματα των patches, τα οποία θα είναι μια οπτική απεικόνιση για το ποιο προϊόν επέλεξαν. Επίσης, στην σειρά 22 ορίζεται η μεταβλητή x , η οποία είναι η μεταβλητή που αναπαριστά την επιλογή των καταναλωτών. Κατόπιν, το χρώμα το οποίο έχει ανατεθεί σε κάθε patch και αυτό αναπαριστά την επιλογή των καταναλωτών. Πιο συγκεκριμένα, αν το χρώμα είναι κόκκινο σημαίνει ότι ο καταναλωτής επέλεξε το προϊόν a και το x ισούται με μηδέν, αν το χρώμα είναι πράσινο, επέλεξε το προϊόν b και το x ισούται με ένα. Παρακάτω, στις σειρές 23-30 έγινε αρχικοποίηση των υπόλοιπων μεταβλητών οι οποίες διαφέρουν σε κάθε καταναλωτή, μέσω τυχαίων τιμών, μέσα σε συγκεκριμένα πλαίσια τιμών, με σκοπό την διαφοροποίηση των καταναλωτών. Επιπροσθέτως, στην σειρά 35 γίνεται μηδενισμός των ticks, τα οποία ουσιαστικά είναι ο χρόνος ή τα βήματα της

προσομοίωσης. Επίσης, στις σειρές 40-90 βρίσκεται το procedure το οποίο αναπαριστάνει ουσιαστικά το κύριο πρόγραμμα. Αρχικά, στην σειρά 41 προστίθεται ένα tick κάθε φορά που "τρέχει" το procedure. Έπειτα, στην σειρά 43 θέτεται εκ νέου το Dt ίσο με το προηγούμενο, συν 0.0333, το 0,0333 προκύπτει από το χρονικό πλαίσιο που πρέπει να έχει το σύστημα έτσι ώστε να είναι ορατά τα αποτελέσματα. Επιπροσθέτως, στις σειρές 45-48 θέτονται εκ νέου πόσοι καταναλωτές υπάρχουν και τι μερίδιο έχουν τα προϊόντα στην αγορά μετά την ολοκλήρωση του προηγούμενου βήματος. Επίσης, στις σειρές 60-63 είναι μια δικλίδα ασφαλείας, η οποία ελέγχει αν η τιμή των Sa, Sb είναι μηδέν να καλείτε το procedure "no-op" το οποίο δεν κάνει τίποτα. Συμπληρωματικά, αυτό έχει γίνει για να μην "πέφτει" το σύστημα, διότι αν ένα από τα 2 είναι μηδέν θα γίνουν διαιρέσεις με το μηδέν παρακάτω. Έπειτα, στις σειρές 51-83 χρησιμοποιούνται οι εξισώσεις του πρώτου μοντέλου που εφαρμόζονται και σε αυτό. Πιο συγκεκριμένα, οι εξισώσεις αυτές βοηθούν την ερμηνεία του κόσμου στον οποίο βρίσκονται οι καταναλωτές, από την οπτική γωνία των μακροοικονομικών παραγόντων, τους οποίους πρέπει να λάβουν υπόψη για την λήψη μιας τέτοιας απόφασης. Παρακάτω, στις σειρές 85, 87 γίνεται κλήση των procedure(s) play, update. Επίσης, στις σειρές 94-124 βρίσκεται το procedure play το οποίο κάνει υπολογισμό του LNS, δηλαδή τον βαθμό ικανοποίησης των χρηστών για τα δύο προϊόντα. Αρχικά, στις σειρές 96-99 θέτεται ένας πίνακας L, προστίθεται στα περιεχόμενά του τα χρήματα που διαθέτουν οι γείτονες του εκάστοτε καταναλωτή και τέλος, θέτεται η μεταβλητή Bneighbors ίση με την ενδιάμεση τιμή των περιεχομένων το πίνακα L. Κατόπιν, στις σειρές 104, 105 υπολογίζεται ο αριθμός των γειτόνων όπου επέλεξαν το προϊόν a και το προϊόν b αντίστοιχα. Επιπροσθέτως, στις σειρές 104-122 εισάγονται οι μαθηματικές εξισώσεις οι οποίες αναφέρθηκαν στην ενότητα (3.3). Παρακάτω, στις σειρές 127-150 τοποθετήθηκε το procedure update, το οποίο είναι υπεύθυνο για την υλοποίηση των εξισώσεων (3.15), (3.20) αλλά και την ενημέρωση διάφορων μεταβλητών. Πιο αναλυτικά, στις σειρές 129-141 υλοποιείται η εξίσωση (3.20) η οποία αποτελείται από εμφολευμένες ρουτίνες ifelse μέσα σε άλλες ρουτίνες ifelse. Έπειτα, στην σειρά 143 ελέγχεται η τιμή του x, και αλλάζουν χρώμα τα patches ανάλογα με τις επιλογές των καταναλωτών. Τέλος, στις σειρές 146, 147 υπολογίζεται ο αριθμός των γειτόνων όπου επέλεξαν το προϊόν a ή το προϊόν b και στην σειρά 148 υλοποιείται η εξίσωση (3.15).

4.2.2 Testing

Το κομμάτι του debugging στην NetLogo είναι αρκετά δύσκολο διότι δεν υπάρχει κάποιο εργαλείο που να βοηθάει τον προγραμματιστή, όπως στις περισσότερες γλώσσες προγραμματισμού. Αρχικά, το πιο σημαντικό είναι ο κώδικας να γράφεται προσεκτικά και χωρίς συντομεύσεις, διότι με την βιασύνη θα γίνουν απρόσεκτα λάθη τα οποία μπορεί να χρειαστεί πολύς χρόνος για την εύρεσή και επίλυση τους. Έπειτα, ένας τρόπος για να αντιληφθεί κάποιος τι πάει λάθος στον κώδικα του είναι με την χρήση των monitors. Επιπρόσθετα, οι monitors εμφανίζουν τις τιμές των επιθυμητών global μεταβλητών σε αληθινό χρόνο. Τα monitors εμφανίζονται στο σχήμα (4.4) κάτω αριστερά με κίτρινα πλαίσια και με αναγραφόμενα τα ονόματα των μεταβλητών οι οποίες παρακολουθούνται.



Σχήμα 4.4: monitors

Κατόπιν, ένα άλλο βήμα που μπορεί να γίνει είναι η ανάλυση του κώδικα βήμα, βήμα έτσι ώστε να γίνει αντιληπτό τι ακριβώς κάνει κάθε σειρά κώδικα. Επίσης, σημαντική είναι η χρήση σχολίων έτσι ώστε να είναι πάντα αντιληπτό τι θα έπρεπε να κάνει ο κώδικας στην παρακάτω γραμμή ή procedure. Τέλος, όπως και με το πρώτο μοντέλο θα πρέπει να εξεταστεί αν τα αποτελέσματα συμπίπτουν με την κοινή λογική και έπειτα αν είναι συμπίπτουν με παρόμοιες υλοποιήσεις [2].

4.2.3 Εξαγωγή δεδομένων

Οι προσομοιώσεις αυτού του μοντέλου έγιναν με ένα εργαλείο της NetLogo το οποίο ονομάζεται BehavioSpace το οποίο φαίνεται στο σχήμα (4.5). Αρχικά, στην αρχική οθόνη στο πάνω μέρος είναι δυνατή η ρύθμιση του ονόματος του πειράματος όπου εξάγεται. Παρακάτω, δηλώνονται οι μεταβλητές και οι διάφορες τιμές οι οποίες είναι επιθυμητές για το συγκεκριμένο πείραμα. Έπειτα, στο μέρος που αναγράφει "repetitions" βρίσκεται ο αριθμός που επιθυμούμε να επαναληφθεί το πείραμα για κάθε τιμή. Στο επόμενο πεδίο, αναγράφονται οι μεταβλητές οι οποίες είναι επιθυμητές για καταγραφή κατά την διάρκεια του πειράματος. Κατόπιν, υπάρχουν τέσσερα πεδία τα οποία δεν χρησιμοποιούνται στην συγκεκριμένη περίπτωση (setup commands, go commands, stop condition, final commands) οπότε δεν χρήζουν περαιτέρω εξήγησης. Επίσης, υπάρχει στο κάτω μέρος το πεδίο time limit το οποίο είναι ο αριθμός των βημάτων, ή αλλιώς των ticks, που θα διαρκέσει το πείραμα.

Experiment

Experiment name:

Vary variables as follows (note brackets and quotation marks):

Either list values to use, for example:
 ["my-slider" 1 2 7 8]
 or specify start, increment, and end, for example:
 ["my-slider" [0 1 10]] (note additional brackets)
 to go from 0, 1 at a time, to 10.
 You may also vary max-pycor, min-pycor, max-pycor, min-pycor, random-seed.

Repetitions:
 run each combination this many times

Run combinations in sequential order
 For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
 sequential order: 1, 1, 2, 2, 3, 3
 alternating order: 1, 2, 3, 1, 2, 3

Measure runs using these reporters:

one reporter per line; you may not split a reporter across multiple lines

Measure runs at every step
 if unchecked, runs are measured only when they are over

Setup commands: Go commands:

Stop condition:
 the run stops if this reporter becomes true

Final commands:
 run at the end of each run

Time limit:
 stop after this many steps (0 = no limit)

OK Cancel

Σχήμα 4.5: BehaviorSpace

Τέλος, τα δεδομένα εξάγονται σε ένα αρχείο τύπου csv το οποίο προστίθεται στην R και γίνονται τα διαγράμματα με παρόμοιο τρόπο όπως και στο πρώτο μοντέλο.

Κεφάλαιο 5ο: Αποτελέσματα

```

1.   var_csv=read.csv("~/Users/kosta/OneDrive/Υπολογιστής/thesis19_10var.csv",sep=',',skip=6)
2.
3.   colnames(var_csv) = c('runNum','var','step','Sa','Sb')
4.
5.   var10Sa <- var_csv[var_csv$step == 100 & var_csv$var == 10 ,]
6.   var10FSa<- mean(var10Sa$Sa)
7.
8.   var10Sb <- var_csv[var_csv$step == 100 & var_csv$var == 10 ,]
9.   var10FSb<- mean(var10Sb$Sb)
10.
11.  var100Sa <- var_csv[var_csv$step == 100 & var_csv$var == 100 ,]
12.  var100FSa<- mean(var100Sa$Sa)
13.
14.  var100Sb <- var_csv[var_csv$step == 100 & var_csv$var == 100 ,]
15.  var100FSb<- mean(var100Sb$Sb)
16.
17.  var1000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 1000 ,]
18.  var1000FSa<- mean(var1000Sa$Sa)
19.
20.  var1000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 1000 ,]
21.  var1000FSb<- mean(var1000Sb$Sb)
22.
23.  var2000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 2000 ,]
24.  var2000FSa<- mean(var2000Sa$Sa)
25.
26.  var2000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 2000 ,]
27.  var2000FSb<- mean(var2000Sb$Sb)
28.
29.  var3000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 3000 ,]
30.  var3000FSa<- mean(var3000Sa$Sa)
31.
32.  var3000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 3000 ,]
33.  var3000FSb<- mean(var3000Sb$Sb)
34.
35.  varData<-data.frame(varVal=c(10,100,1000,2000,3000),
36.  Sa=(var10FSa,var100FSa,var1000FSa,var2000FSa,var3000FSa),
37.  Sb = c(var10FSb,var100FSb,var1000FSb,var2000FSb,var3000FSb))
38.
39.  library(tidyverse)
40.
41.  ggplot(varData , aes(varVal))+geom_line(aes(y = Sa , colour = "Sa"))+
42.  geom_line(aes(y = Sb , colour = "Sb"))+geom_point(aes(y = Sa , colour = "Sa"))+
43.  geom_point(aes(y = Sb , colour = "Sb")) +labs(title = "var 10-3000",y = "Agents' preference" ,x="var")

```

Σχήμα 5.1: R code

5.1 Δημιουργία Διαγραμμάτων με την R

Όπως προαναφέρθηκε τα διαγράμματα των αποτελεσμάτων έγιναν με την γλώσσα R. Στο παράρτημα Γ είναι ο σχετικός κώδικας που υλοποίησε τα διαγράμματα. Αρχικά, στο συγκεκριμένο script η μεταβλητή *var* έχει τις τιμές 10, 100, 1000, 2000, 3000 και κάθε φορά που εξετάζεται άλλη μεταβλητή αλλάζει το όνομά της και οι τιμές της στον κώδικα. Κατόπιν, στην σειρά 1 γίνεται η εισαγωγή το csv αρχείου μέσω της εντολής *read_csv*, με διαχωριστικό(*sep*) των στηλών το κόμμα και με εντολή να αγνοηθούν(*skip*) οι πρώτες 6 σειρές. Επίσης, στην σειρά 3 ορίζονται τα ονόματα των στηλών. Έπειτα, στις σειρές 5-35 γίνεται επιλογή των δεδομένων διότι, κάθε τιμή κάθε μεταβλητής έχει προσομοιωθεί εκατό φορές με σκοπό να εξαλειφθεί η τυχαιότητα. Πιο συγκεκριμένα, στην σειρά 5 όλες οι μεταβλητές στο βήμα 100 (το τελικό βήμα της προσομοίωσης) και με τιμή 10 αποθηκεύονται στην *var10Sa* και στην σειρά 6 ο μέσος όρος αυτών των τιμών αποθηκεύεται στην *var10FSa*. Η συγκεκριμένη διαδικασία επαναλαμβάνεται για *Sa*, *Sb* για κάθε τιμή που παίρνει η μεταβλητή μέχρι την σειρά 33. Επιπροσθέτως, στην σειρά 34 στην μεταβλητή *varData* αποθηκεύεται το σύνολο δεδομένων, ουσιαστικά το *varData* γίνεται ένας πίνακας, με ονόματα στηλών *varVal*(που περιέχει τις τιμές της μεταβλητής), *Sa*(που περιέχει των μέσο όρο των 100 *Sa* για τις αντίστοιχες τιμές) και *Sb*(που περιέχει των μέσο όρο των 100 *Sb* για τις αντίστοιχες τιμές). Επίσης, στην σειρά 37 γίνεται η εισαγωγή της βιβλιοθήκης που θα επιτρέψει να δημιουργηθούν τα διαγράμματα. Τέλος, στην σειρά 39 βρίσκεται η εντολή της υλοποίησης

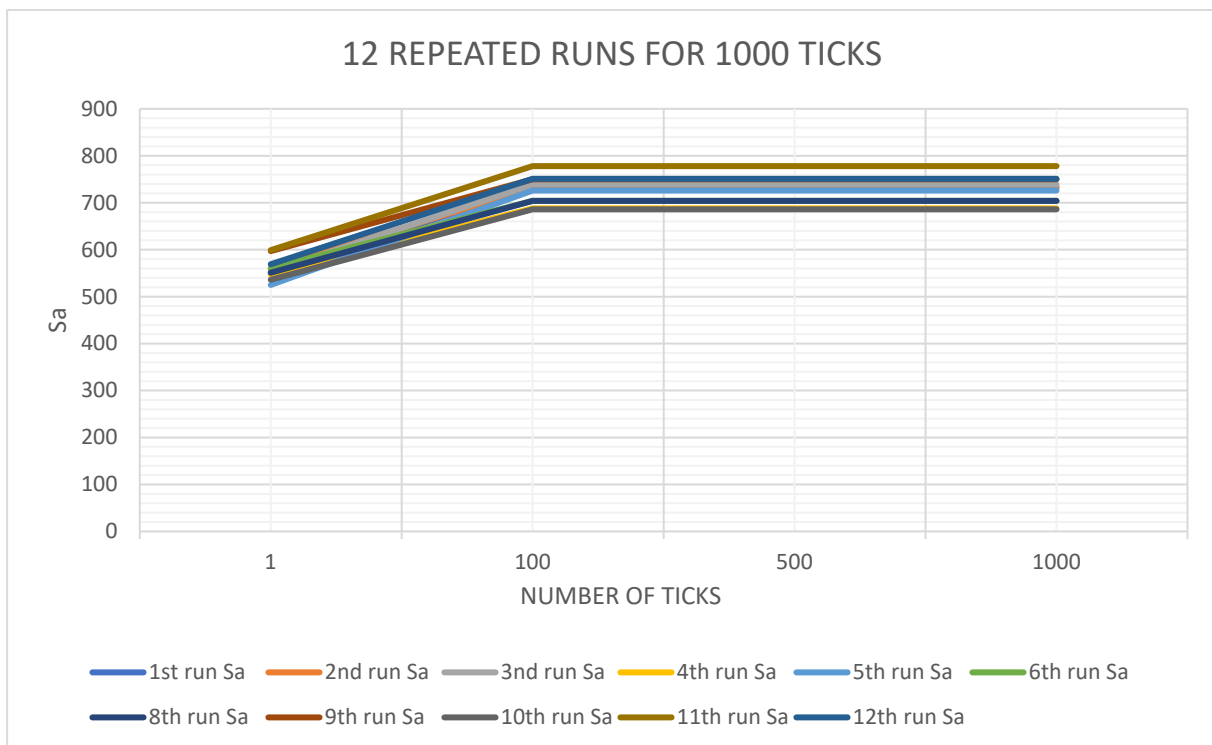
του διαγράμματος το οποίο γίνεται και με σημεία(geompoint) και με γραμμές(geomline) αξιοποιώντας το dataframe που δημιουργήθηκε νωρίτερα(varData).

5.2 Εύρος Τιμών Μεταβλητών

Το εύρος τιμών των μεταβλητών που θα εξεταστεί σε ένα πολύπλοκο σύστημα θα μπορούσε να γίνει προσομοιώνοντας όλες τις μεταβλητές σε ένα μεγάλο εύρος τιμών, εξάγοντας αυτά τα αποτελέσματα και στην συνέχεια μέσω παρατήρησης να επιλεγεί το επιθυμητό εύρος τιμών. Η συγκεκριμένη διαδικασία θα ήθελε πολύ υπολογιστικό χρόνο με τον διαθέσιμο εξοπλισμό, οπότε η επιλογή του εύρους τιμών στην συγκεκριμένη περίπτωση, έγινε θέτοντας εμπειρικά τιμές σε κάθε μεταβλητή μέχρι να σταματήσει να επηρεάζεται το σύστημα. Για παράδειγμα, θέτοντας την μεταβλητή VC_a 12,5 και 15 επηρεάζει το σύστημα αλλά θέτοντάς το 15, 100, 1000 δεν έχει επιρροή στο σύστημα γιατί σε κάθε περίπτωση τα S_a , S_b έχουν περιορισμένες τιμές (bounded), οπότε είναι λογικό να μην είναι επιθυμητό να μελετηθούν οι συγκεκριμένες τιμές.

5.3 Εύρος Βημάτων

Το εύρος βημάτων είναι απαραίτητο να υπολογιστεί διότι δεν είναι δυνατόν οι προσομοιώσεις να "τρέχουν" ατέρμονα και έπειτα, για την οικονομία χρόνου προσομοίωσης. Αρχικά, υλοποιήθηκαν δώδεκα προσομοιώσεις με τις ίδιες μεταβλητές για χίλια βήματα. Στην συνέχεια, έγινε διάγραμμα(σχήμα (5.2)) στο οποίο φαίνεται ότι τα αποτελέσματα σε όλα τα runs σταματάνε να μεταβάλλονται μετά το 100 βήμα. Συμπερασματικά, δεν υπάρχει νόημα οι προσομοιώσεις να διαρκέσουν πάνω από 100 βήματα αφού το σύστημα φτάνει σε κατάσταση ισορροπίας ήδη από το 100ο βήμα της προσομοίωσης.



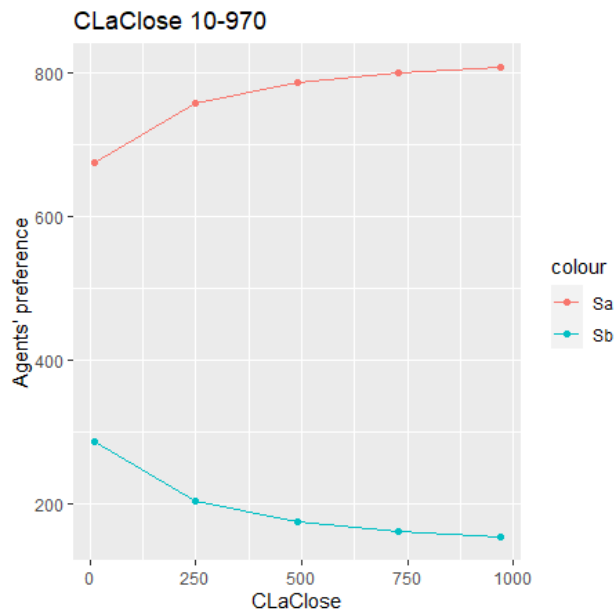
Σχήμα 5.2: 1000 TICKS RESULTS

5.4 Ανάλυση Αποτελεσμάτων

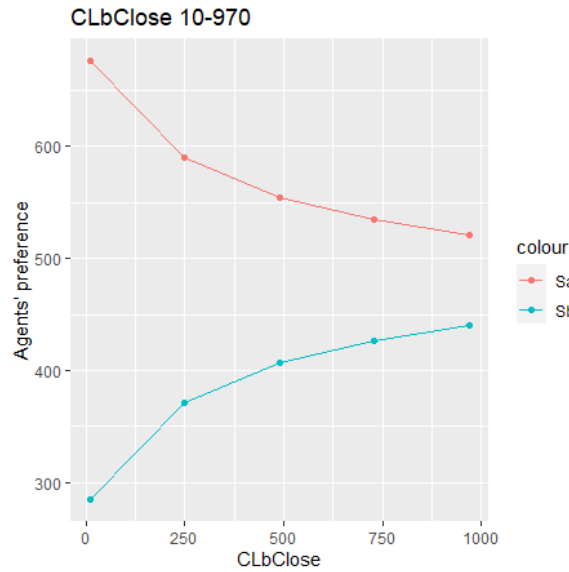
Για το πρώτο μοντέλο οι αρχικές τιμές των μεταβλητών αναγράφονται στο παράρτημα Δ και για το τελικό μοντέλο αναγράφονται στο παράρτημα Ε. Στην παρακάτω ενότητα, βρίσκονται τα διαγράμματα επιρροής, της τιμής των διάφορων μεταβλητών, στις επιλογές των καταναλωτών για τα προϊόντα (S_a , S_b) μετά από 100 βήματα. Επίσης, πρέπει να σημειωθεί ότι οι αρχικές τιμές έχουν οριστεί έτσι ώστε να ευνοείται το προϊόν a , με σκοπό να είναι πιο εύκολη η παρατήρηση των αλλαγών.

5.4.1 Πρώτο Μοντέλο

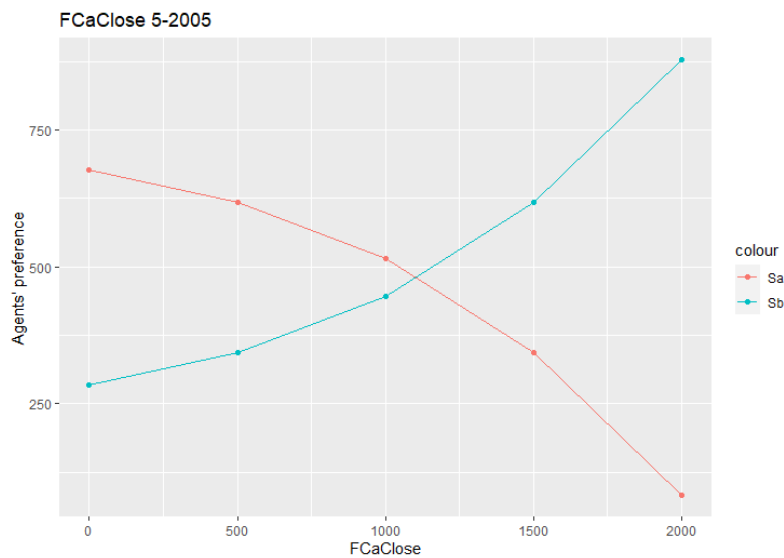
Αρχικά, στα σχήματα (5.3), (5.4) είναι τα διαγράμματα από τις μεταβλητές CL_a , CL_b αντίστοιχα, οι οποίες αναπαριστούν τον αριθμό των παραγόμενων προϊόντων. Όπως φαίνεται μεγαλώνοντας τον αριθμό παραγωγής των προϊόντων a αυξάνεται η ζήτηση στην αγορά (S_a). Αυτό συμβαίνει διότι όσο πιο μεγάλη είναι η παραγωγή τόσο πιο πολύ μειώνεται το κόστος, κατά συνέπεια και η τιμή κάτι το οποίο είναι λογικό.

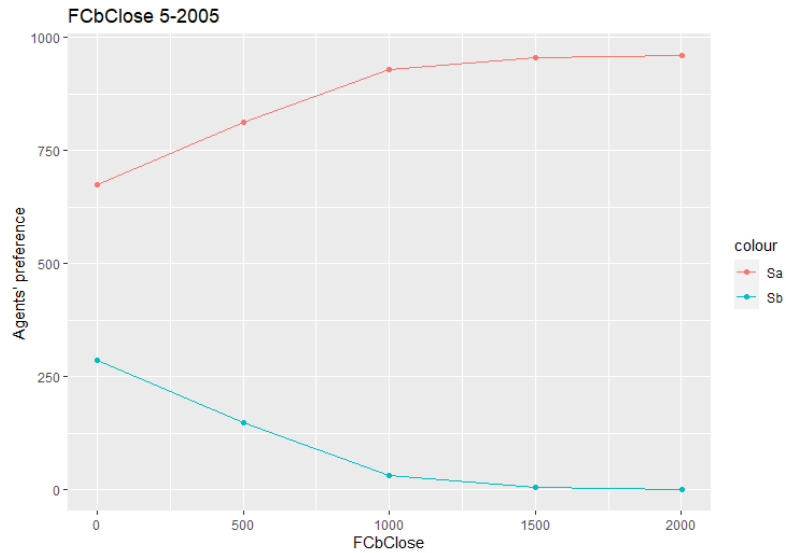


Σχήμα 5.3: CL_a 10-970

Σχήμα 5.4: CL_b 10-970

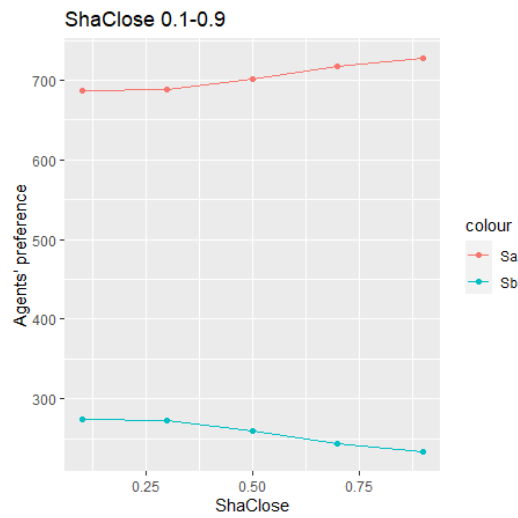
Στην συνέχεια, στα σχήματα (5.5), (5.6) βρίσκονται τα διαγράμματα για τις μεταβλητές FC_a, FC_b οι οποίες είναι τα πάγια έξοδα των προϊόντων. Έπειτα, παρατηρείται ότι ένα προϊόν δεν είναι βιώσιμο διότι με τα μεγάλα πάγια έξοδα αυξάνεται η τιμή. Οπότε παρόλο που το προϊόν a είναι ευνοούμενο φαίνεται ότι κυριαρχεί το προϊόν b όταν τα πάγια έξοδα του προϊόντος a αυξάνονται κατά πολύ.

Σχήμα 5.5: FC_a 5-2005

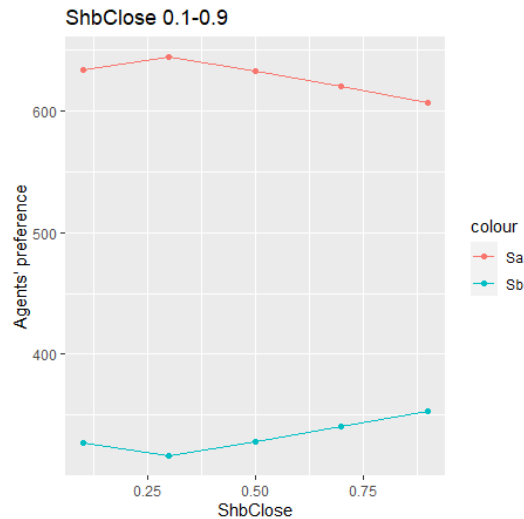


Σχήμα 5.6: FC_b 5-2005

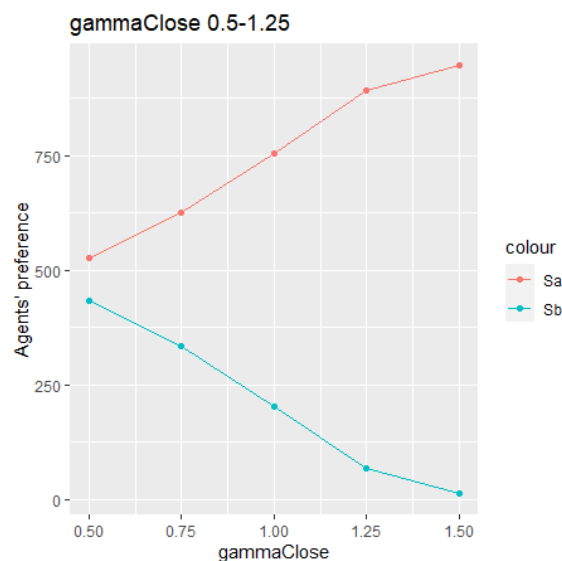
Κατόπιν, στα σχήματα (5.7), (5.8) παρατηρείται η διαφορά που κάνει το αρχικό μερίδιο στην αγορά (Sh_a , Sh_b). Φαίνεται ότι, η επιρροή στην αγορά είναι υπαρκτή αλλά όχι σημαντική. Πιο συγκεκριμένα, τα προϊόντα παρόλο που έχουν μεγάλο αρχικό μερίδιο στην αγορά δεν είναι σε κατάσταση lock-in οπότε δεν επωφελούνται από τα θετικά του.



Σχήμα 5.7: Sh_a 0.1-0.9

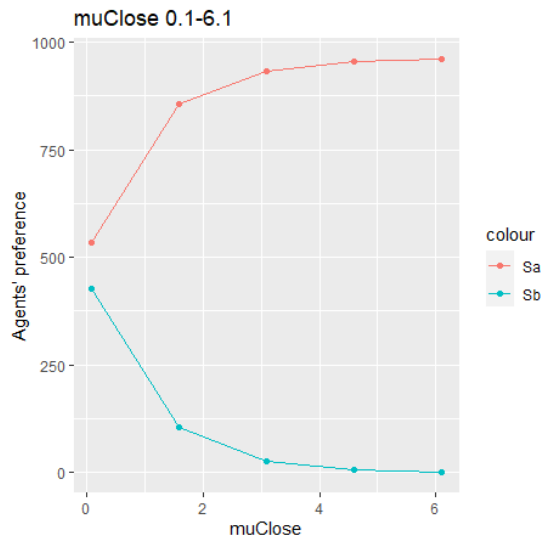
Σχήμα 5.8: Sh_b 0.1-0.9

Επίσης, στο σχήμα (5.9) έχει τοποθετηθεί η μεταβλητή gamma η οποία είναι ο παράγοντας μείωσης κόστους ανά διπλασιασμό παραγωγής. Συμπληρωματικά, όσο μεγαλύτερη είναι η τιμή της τόσο πιο μεγάλο μερίδιο έχει στην αγορά το προϊόν το οποίο είναι και ευνοούμενο. Το συγκεκριμένο αποτέλεσμα συμπίπτει με την λογική διότι το προϊόν a έχει μεγαλύτερη αρχική παραγωγή, άρα μικρότερο κόστος παραγωγής και όσο μεγαλώνει ο παράγοντας gamma τόσο πιο πολύ επηρεάζει αυτό το σύστημα.



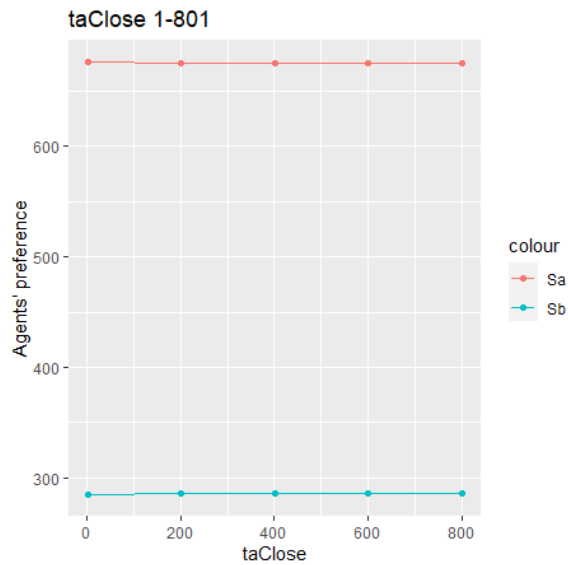
Σχήμα 5.9: gamma 0.5-1.25

Έπειτα, στο σχήμα (5.10) βρίσκεται η ευαισθησία της προσφοράς λόγω διαφοράς τιμής, δηλαδή η παράμετρος μ . Επιπροσθέτως, φαίνεται στο σχήμα ότι όσο πιο ευαίσθητη είναι η αγορά στην διαφορά τιμής τόσο πιο πολύ επιλέγεται το ευνοούμενο προϊόν. Συμπληρωματικά, αυτό γίνεται γιατί όπως προαναφέρθηκε όσο πιο μεγάλο μερίδιο στην αγορά έχει ένα προϊόν τόσο μικρότερη γίνεται η τιμή του.



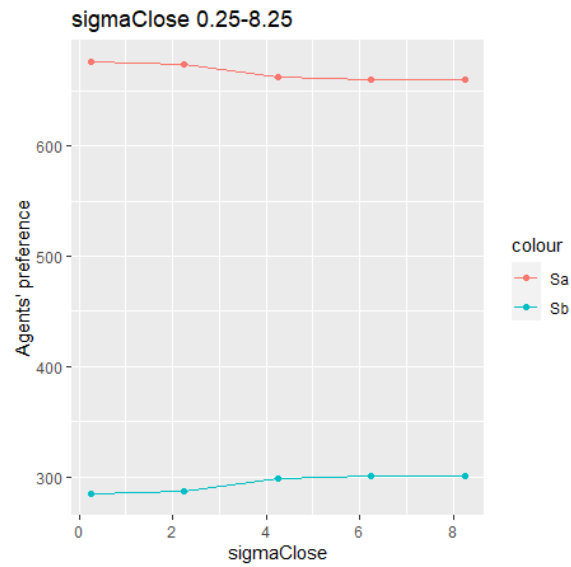
Σχήμα 5.10: μ 0.1-6.1

Παρακάτω στο σχήμα (5.11) έχει τοποθετηθεί ο χρόνος προσαρμογής της αγοράς t_a , ο οποίος δεν φαίνεται να επηρεάζει αισθητά το σύστημα.



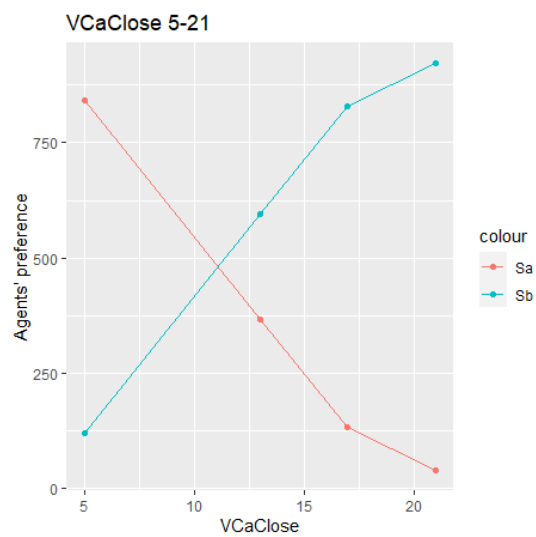
Σχήμα 5.11: t_a 0.1-800

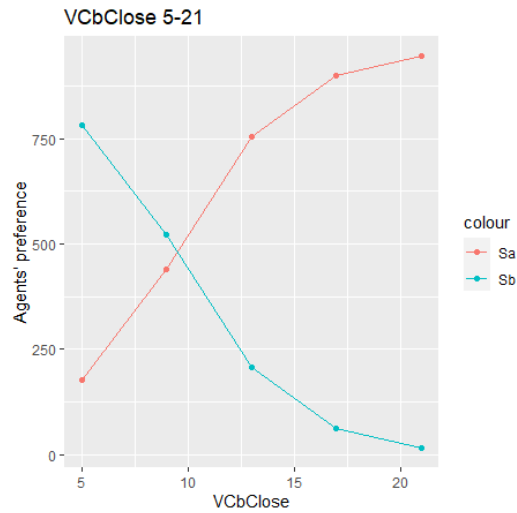
Κατόπιν, στο σχήμα (5.12) βρίσκεται το σ το οποίο είναι η τυπική απόκλιση που χρησιμοποιείται στον στοχαστικό όρο. Επιπροσθέτως, υπάρχει μια μικρή επιρροή στο σύστημα, διότι όσο μεγαλώνει τόσο πιο μεγάλη επιρροή έχουν τα τυχαία γεγονότα που επηρεάζουν την αγορά, οπότε τα προϊόντα συγκλίνουν ως προς το μερίδιο της αγοράς. Επίσης, το φαινόμενο αυτό το είδαμε να συμβαίνει και κατά την διάρκεια του κορονοϊού καθώς μεγάλες εταιρίες αποδυναμώθηκαν και μικρές εταιρίες μεγάλωσαν.



Σχήμα 5.12: ta 0.25-8.25

Τέλος, η επιρροή των μεταβαλλόμενων κοστών (VC_a , VC_b) (σχήματα (5.13), (5.14) αντίστοιχα) είναι παρόμοια με την επιρροή των πάγιων εξόδων (σχήματα (5.5), (5.6)) όπως είναι και λογικό εφόσον και τα δύο είναι κόστη για τα προϊόντα.

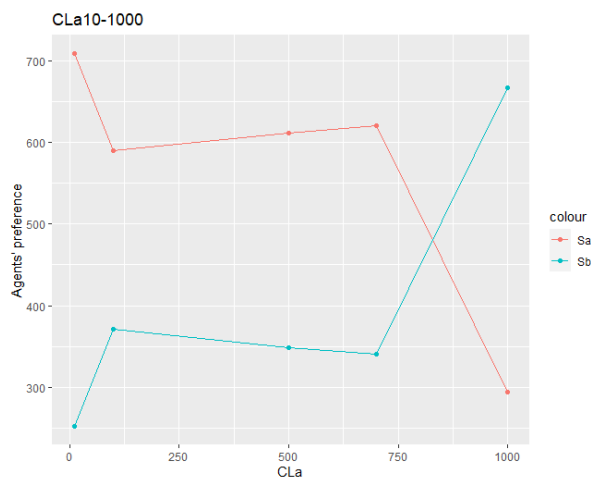
Σχήμα 5.13: VC_a 5-14.5



Σχήμα 5.14: VC_b 5-14.5

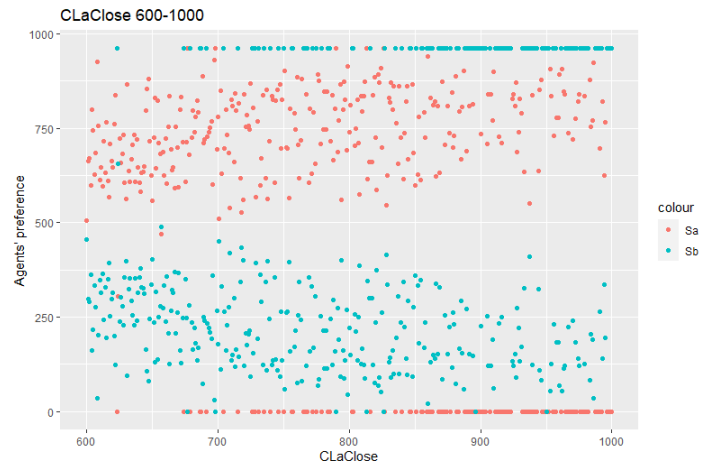
5.4.2 Τελικό Μοντέλο

Αρχικά, στο σχήμα (5.15) βρίσκεται ο αριθμός των παραγόμενων προϊόντων. Κατόπιν, το αποτέλεσμα του σχήματος είναι λογικό μέχρι την τιμή 750. Έπειτα, μετά από αυτήν την τιμή αντιστρέφεται η συμπεριφορά του μοντέλου, το οποίο χρήζει περαιτέρω μαθηματικής μελέτης.

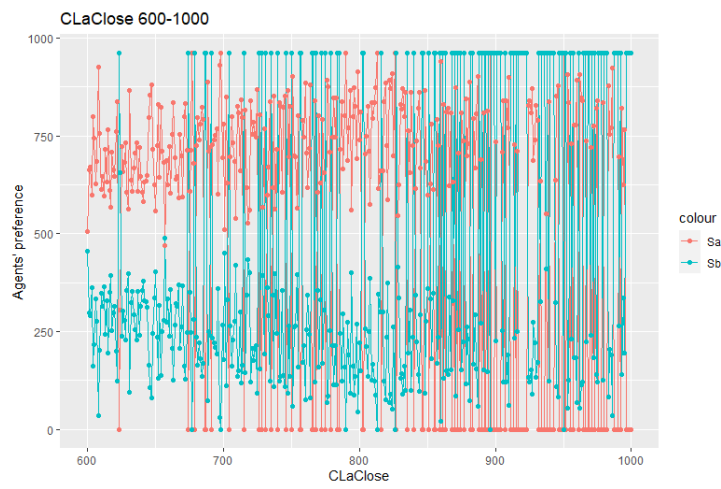


Σχήμα 5.15: CL_a 10-1000

Πιο συγκεκριμένα, μελετώντας πιο κοντά (σχήματα (5.16), (5.17)) τα αποτελέσματα, στο σημείο όπου γίνεται η τομή των δύο γραμμών φαίνεται η απρόβλεπτη συμπεριφορά του συστήματος καθώς αλλάζει η τιμή της παραμέτρου. Το φαινόμενο στο οποίο η κατάσταση ισορροπίας του συστήματος αλλάζει ανάλογα με την τιμή της παραμέτρου λέγεται «διακλάδωση» (Bifurcation) και απαιτείται περισσότερη δουλειά για να αποδειχθεί εάν η δυναμική του συστήματος ακολουθεί τον δρόμο μιας χαοτικής συμπεριφοράς ή άλλου μηχανισμού.

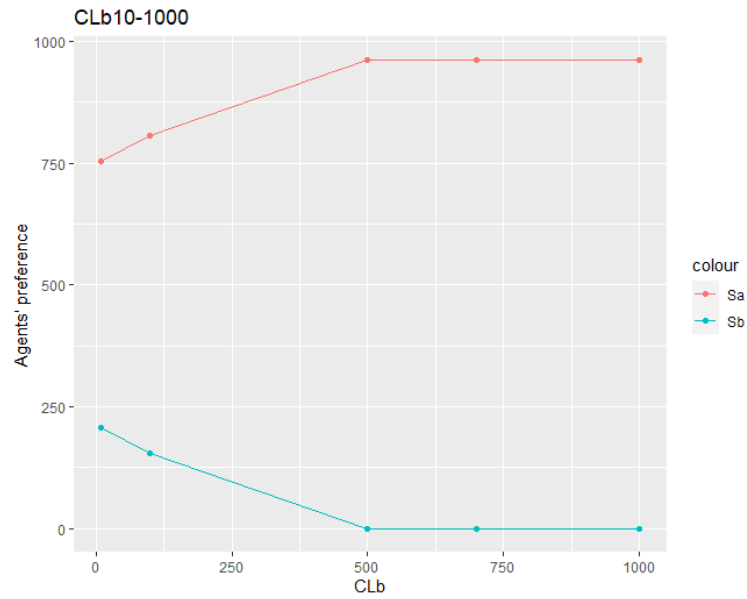


Σχήμα 5.16: CL_a 600-1000



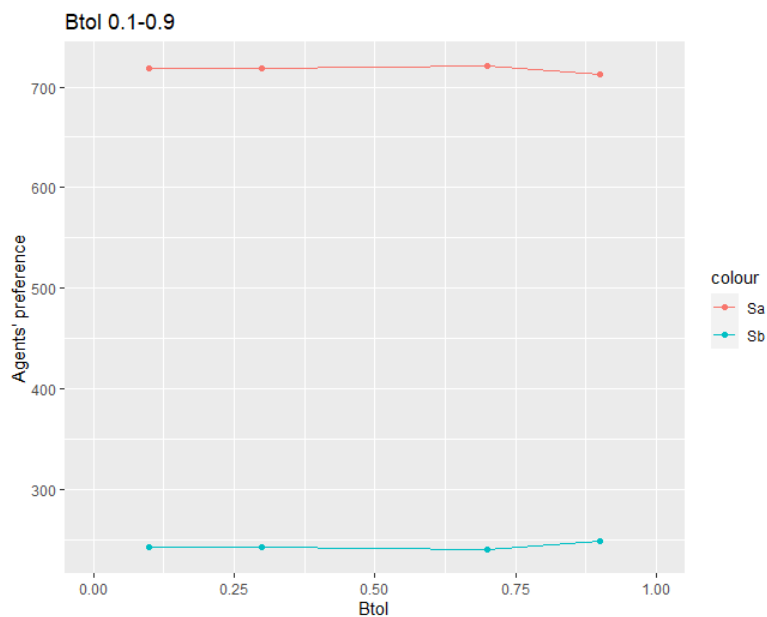
Σχήμα 5.17: CL_a 600-1000

Έπειτα, στο σχήμα (5.18) φαίνεται η επιρροή που έχει στην αγορά το CL_b .



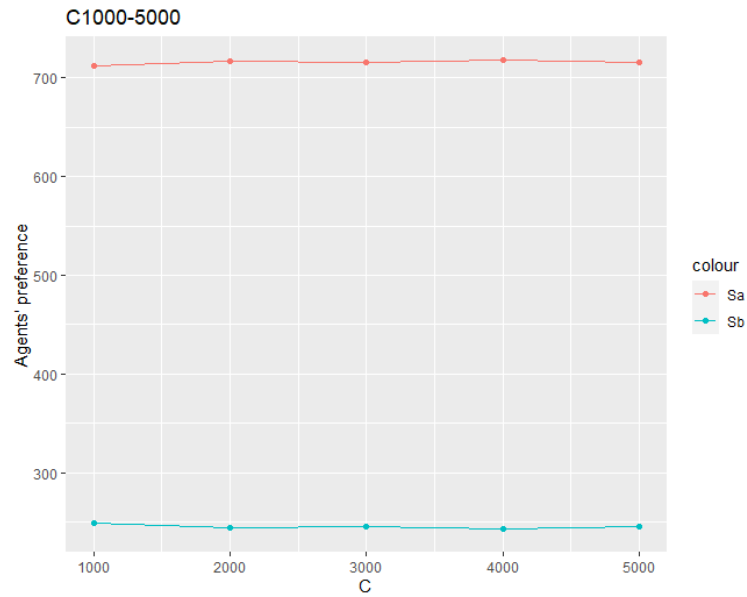
Σχήμα 5.18: CL_b 10-1000

Παρακάτω, έχει τοποθετηθεί, στο σχήμα (5.19) η ανοχή που έχουν οι καταναλωτές στα εισοδήματα των γειτόνων τους από την εξίσωση (3.20). Όπως φαίνεται δεν έχει μεγάλη επιρροή στο σύστημα όπως και είναι το λογικό διότι είναι ένας πολύ μικρός παράγοντας που θα επηρεάσει την τελική απόφαση του καταναλωτή.



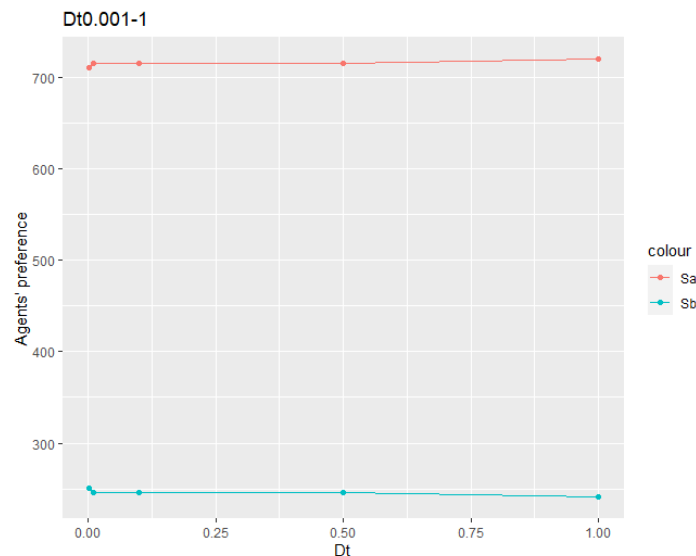
Σχήμα 5.19: B_{tol} 0.1-0.9

Κατόπιν, σύμφωνα με το σχήμα (5.20) η μεταβλητή C, η οποία είναι η συγκέντρωση της μόλυνσης, δεν επηρεάζει καθόλου το σύστημα, κάτι το οποίο είναι λανθασμένο και δεν έχει λογική. Επιπροσθέτως, το συγκεκριμένο λάθος δεν υπήρχε η δυνατότητα να επιλυθεί διότι, παρόλο που το υπολογιστικό μοντέλο και ο κώδικας είναι σωστός, η NetLogo εμφανίζει λάθος αποτελέσματα.



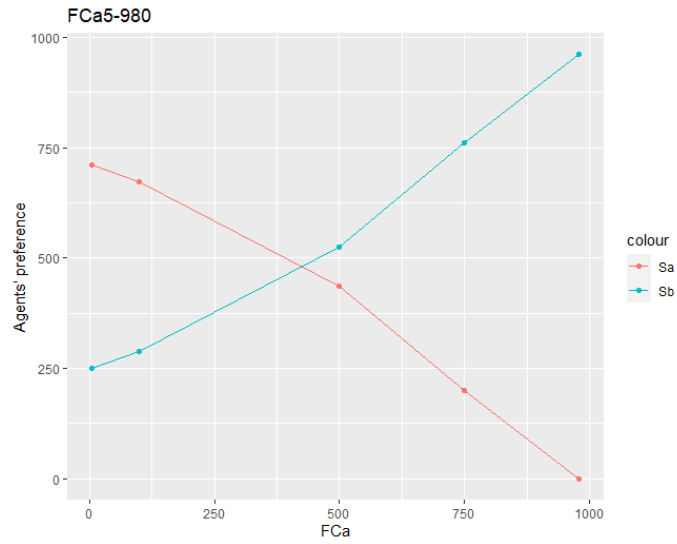
Σχήμα 5.20: C 1000-5000

Στην συνέχεια, η μεταβλητή dt δεν επηρεάζει το τελικό αποτέλεσμα του συστήματος, όπως είναι λογικό, διότι η χρησιμότητα της είναι ο έλεγχος του χρονικού πλαισίου του μοντέλου (σχήμα (5.21)). Επεξηγηματικά, η dt ορίζει το πόσο πολύ επηρεάζει κάθε βήμα το μοντέλο.

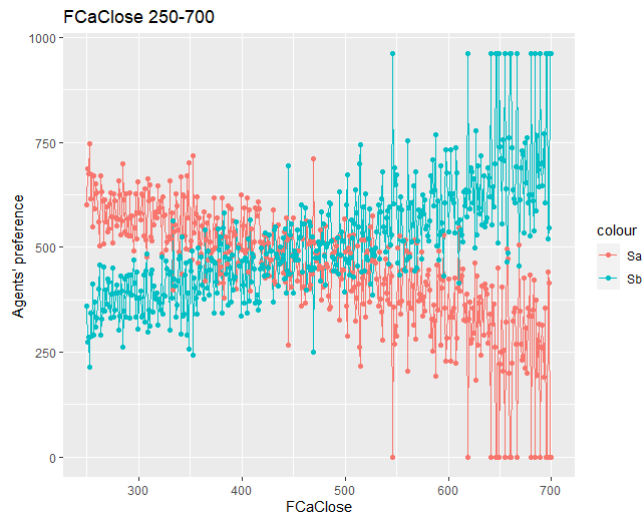


Σχήμα 5.21: dt 0.001-1

Παρακάτω, φαίνεται η επιρροή των παγίων εξόδων του προϊόντος a (σχήμα (5.22)), η οποία είναι λογική καθώς όσο αυξάνονται τα πάγια έξοδα τόσο μειώνεται το μερίδιο που έχει στην αγορά. Επιπροσθέτως στο σχήμα (5.23) είναι ένα διάγραμμα που μελετάει την δυναμική, με πολύ πυκνά σημεία τιμών της FC_a , κοντά στο σημείο που τέμνονται οι ευθείες στο σχήμα (5.22), με σκοπό να επιβεβαιωθεί ο ομαλός χαρακτήρας του συστήματος, όπως και έγινε.

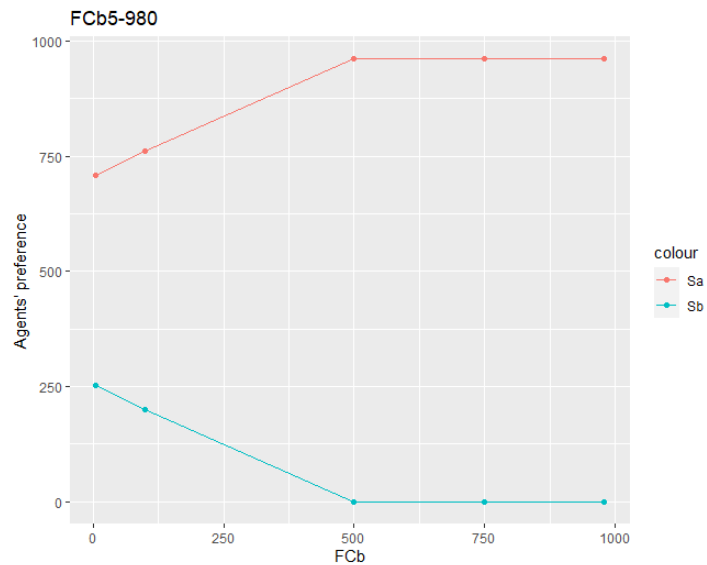


Σχήμα 5.22: FC_a 5-980

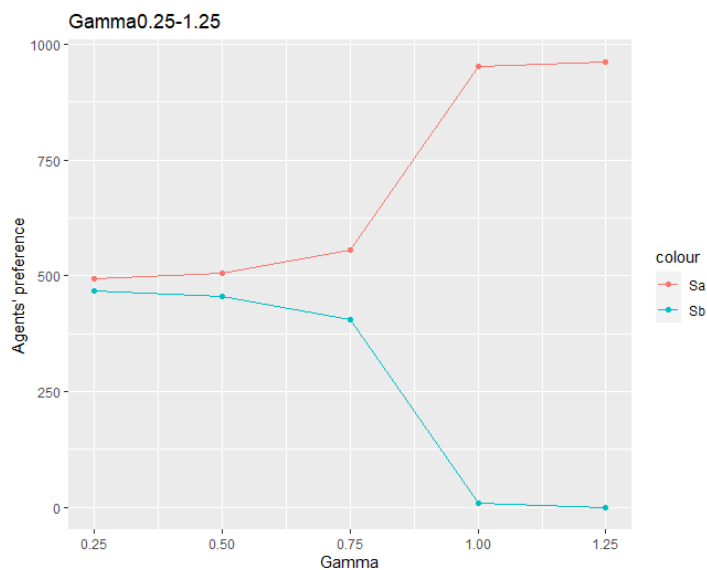


Σχήμα 5.23: FC_a 250-700

Επίσης, στο σχήμα (5.24) έχει τοποθετηθεί η επιρροή των παγίων εξόδων του προϊόντος b, η οποία έχει παρόμοια συμπεριφορά με του προϊόντος a.

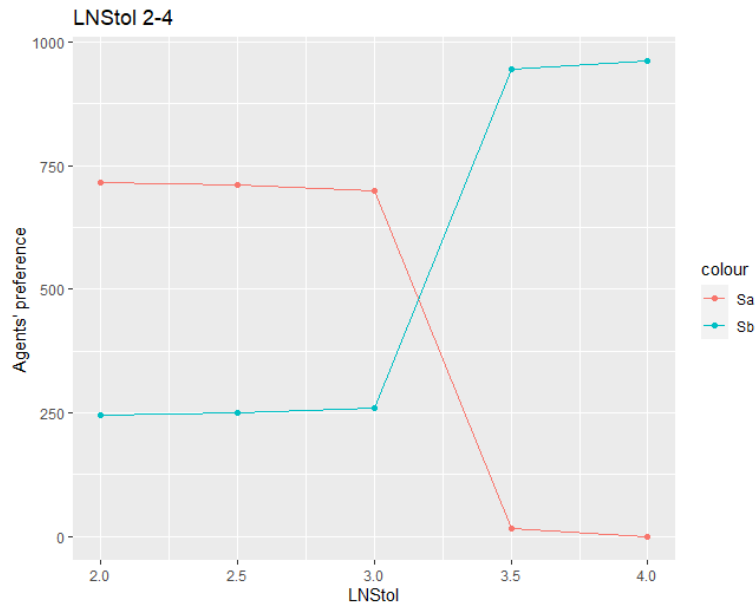
Σχήμα 5.24: FC_b 5-980

Στην συνέχεια, παρατηρείται ότι ο παράγοντας μείωσης κόστους ανά διπλασιασμό της παραγωγής επηρεάζει λογικά το σύστημα καθώς όσο αυξάνεται, τόσο πιο πολύ μερίδιο στην αγορά έχει το επωφελημένο προϊόν a (σχήμα(5.25)).



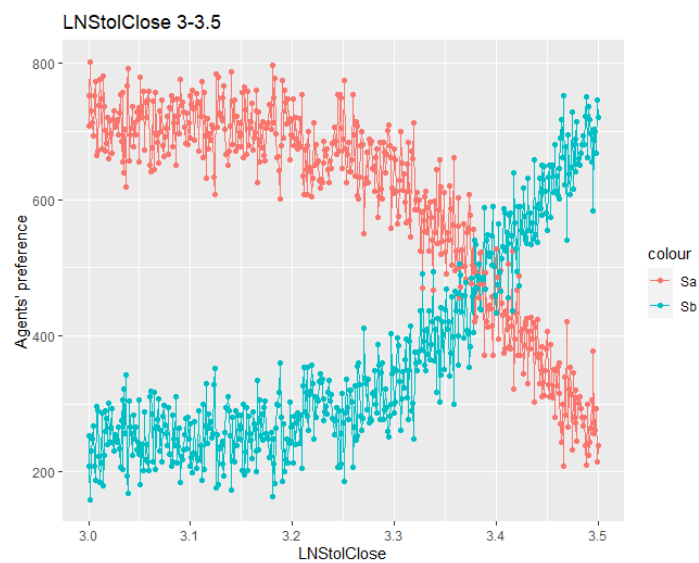
Σχήμα 5.25: gamma 0.25-1.25

Επιπροσθέτως, όταν το όριο που θέτουν οι καταναλωτές, που πρέπει να υπάρχει για να είναι ευχαριστημένοι (LNS_{i0i}) με την επιλογή τους, περάσει ένα συγκεκριμένο σημείο καταλήγουν να κάνουν πάντα την αντίθετη επιλογή. Πιο συγκεκριμένα, αυτό έχει ως αποτέλεσμα να επιλεγεί το προϊόν το οποίο έχει το μικρότερο μερίδιο στην αγορά(σχήμα(5.26)).



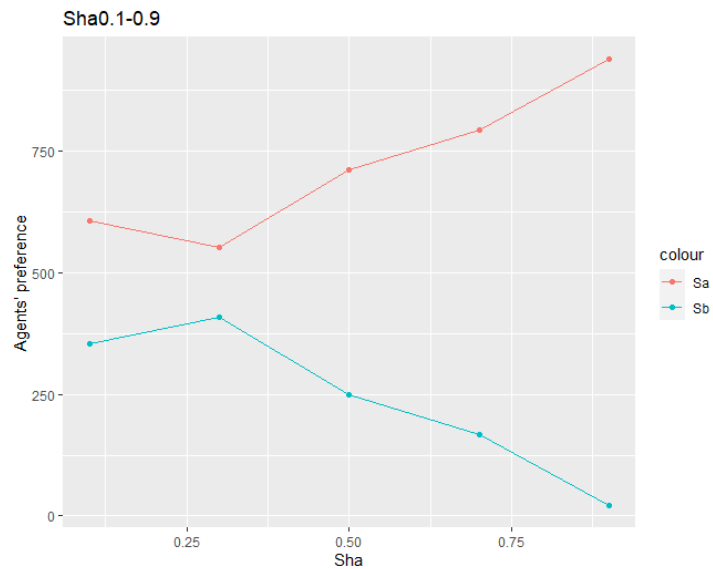
Σχήμα 5.26: LNS_{tol} 2-4

Κατόπιν, στο σχήμα(5.27) φαίνεται με πιο αναλυτικό τρόπο το σημείο που τέμνονται οι γραμμές στο σχήμα (5.26), για την επιβεβαίωση της ομαλής λειτουργίας του συστήματος.



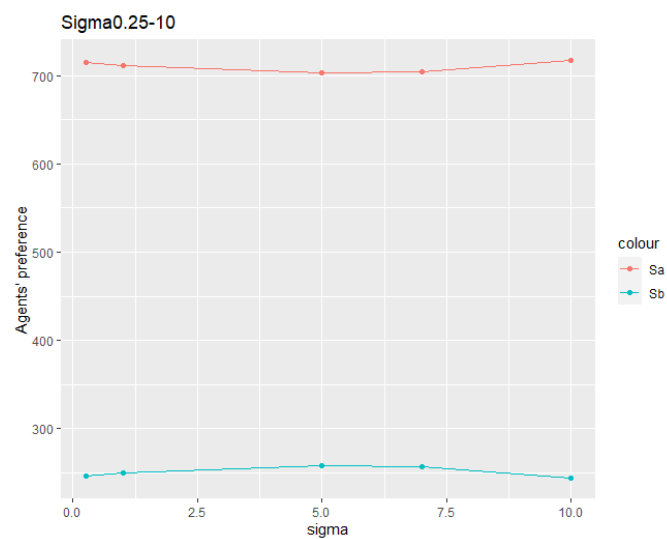
Σχήμα 5.27: LNS_{tol} 3-3.5

Επίσης, η επιρροή του αρχικού μεριδίου της αγοράς έχει αισθητές διαφορές στα τελικά αποτελέσματα του συστήματος διότι οι καταναλωτές επηρεάζονται σημαντικά από τις επιλογές των γειτόνων τους (σχήμα (5.28)).



Σχήμα 5.28: Sha 0.1-0.9

Έπειτα, η τυπική απόκλιση του τυχαίου όρου έχει μια πολύ μικρή επιρροή στο σύστημα διότι, τα στοχαστικά γεγονότα δεν αλλάζουν σε μικρό χρονικό διάστημα ριζικά το σύστημα, οπότε μόνα τους δεν έχουν την δύναμη να επηρεάσουν τις απόψεις των καταναλωτών για τα προϊόντα(σχήμα (5.29)).

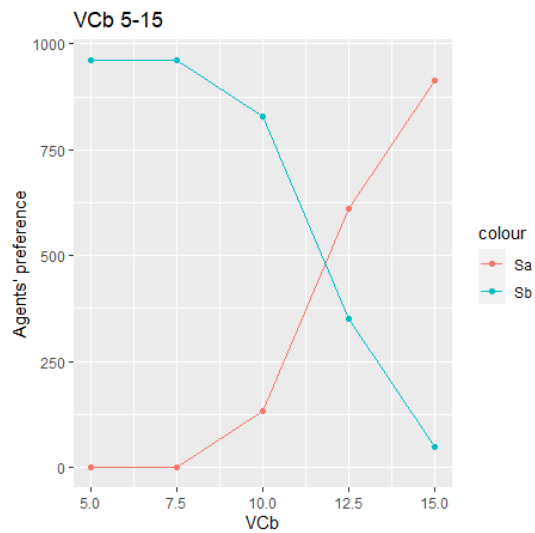


Σχήμα 5.29: sigma 0.25-10

Παρακάτω, στα σχήματα (5.30), (5.31) είναι τα μεταβλητά έξοδα για το προϊόν a, b αντίστοιχα, τα οποία έχουν παρόμοια επιρροή στο σύστημα όπως και τα FC_a , FC_b αφού πρόκειται περί εξόδων. Επιπλέον, η VC επηρεάζει πιο σημαντικά το σύστημα από την FC διότι έχει πιο σημαντική τιμή.

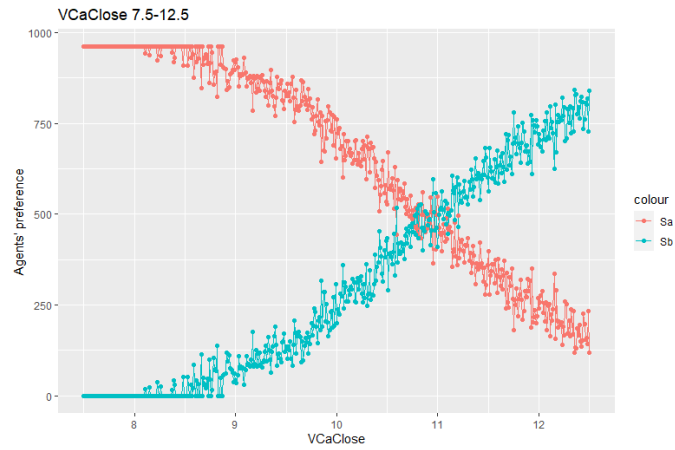
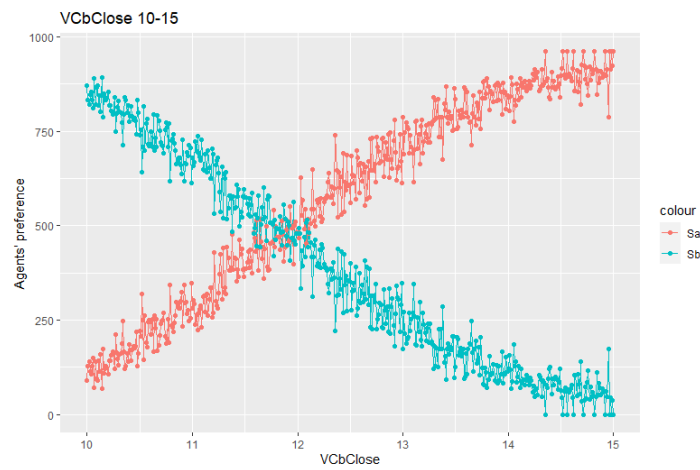


Σχήμα 5.30: VC_a 2.5-15



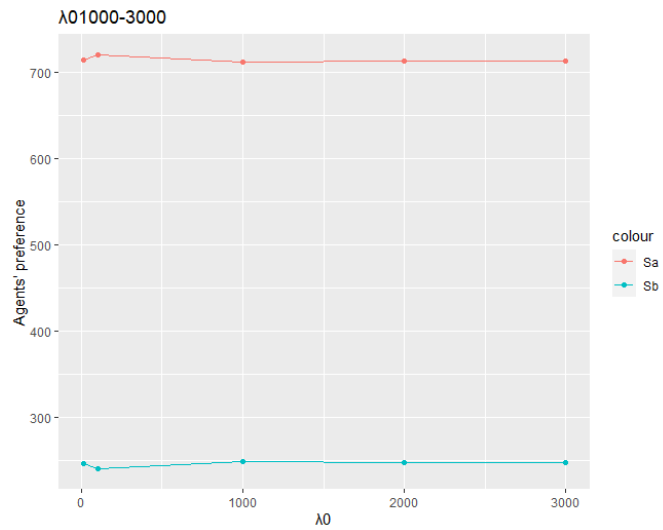
Σχήμα 5.31: VC_b 2.5-15

Επιπροσθέτως, στα σχήματα (5.32), (5.33) είναι πιο λεπτομερή διαγράμματα στις τιμές που τέμνονται οι ευθείες των σχημάτων (5.30), (5.31) για την επιβεβαίωση της ομαλής λειτουργίας του συστήματος.

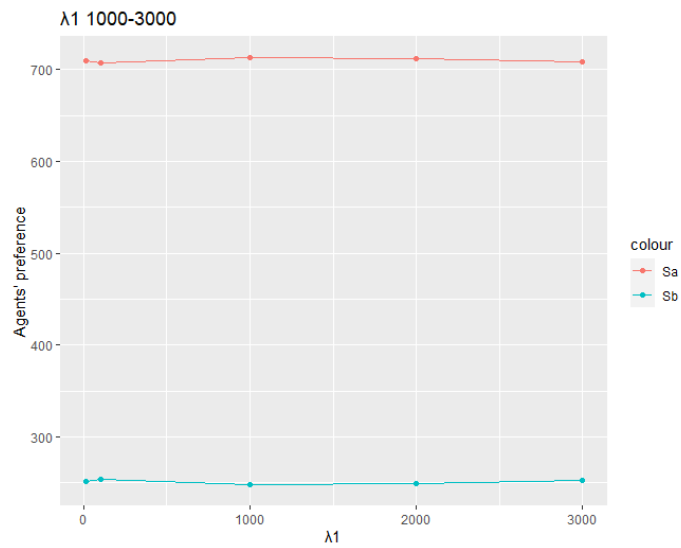
Σχήμα 5.32: VC_a 7.5-12.5Σχήμα 5.33: VC_b 10-15

Κατόπιν, οι μονάδες μόλυνσης που προσφέρει κάθε προϊόν(λ_0, λ_1) δεν επηρεάζουν καθόλου το σύστημα αφού όπως προαναφέρθηκε ούτε το C το επηρεάζει (σχήματα (5.34), (5.35) αντίστοιχα).

Κεφάλαιο 5

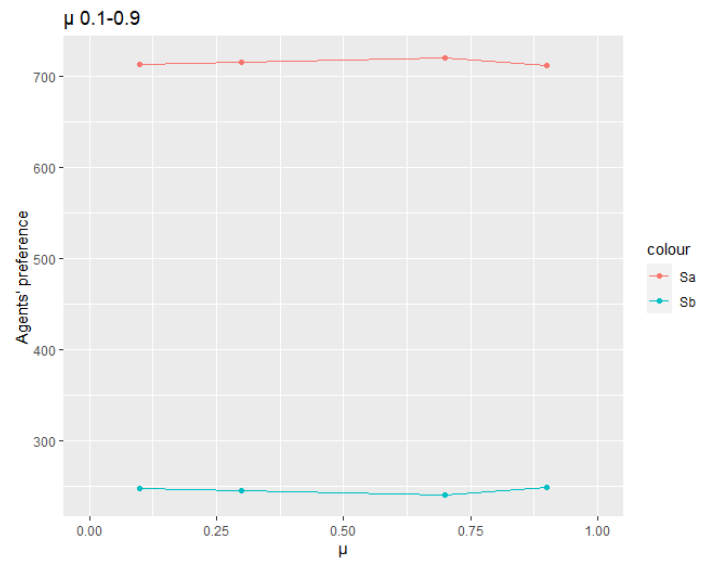


Σχήμα 5.34: λ_0 1000-3000



Σχήμα 5.35: λ_1 1000-3000

Τέλος, ο ρυθμός μείωσης μ για τον ίδιο λόγο όπως και το λ δεν επηρεάζει καθόλου το σύστημα(σχήμα (5.36)).



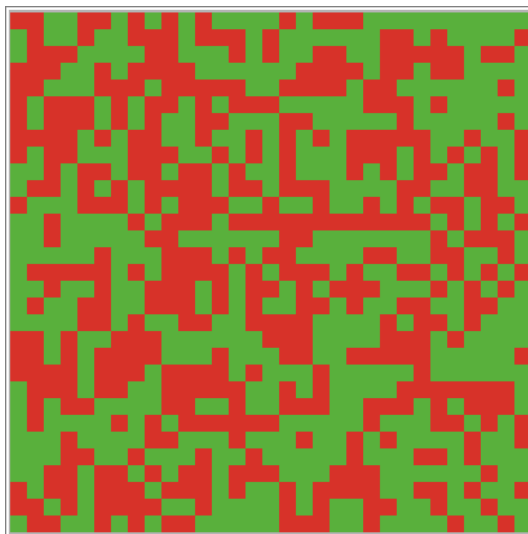
Σχήμα 5.36: μ 0.1-0.9

5.5 Σύγκριση Μοντέλων

Είναι πολύ ενδιαφέρουσα η σύγκριση διάφορων αποτελεσμάτων των δύο μοντέλων, με σκοπό την εύρεση των διαφορών που υπάρχουν στα τελικά αποτελέσματά τους, λόγω της προσθήκης καταναλωτών με δικές τους προσωπικότητες. Αρχικά, τα πάγια έξοδα φαίνεται να επηρεάζουν με διπλάσιο ρυθμό το πρώτο μοντέλο, αυτό γίνεται διότι στο δεύτερο μοντέλο υπάρχουν πολλοί περισσότεροι παράγοντες επιρροής από ότι στο πρώτο, παρόλα αυτά και τα δυο συστήματα επηρεάζονται με παρόμοια τρόπο. Στην συνέχεια, το γ φαίνεται, μέχρι την τιμή 0.75 (σχήμα(5.25)), να μην επηρεάζει σημαντικά το μοντέλο, ενώ μετά από εκεί έχει καθοριστικό ρόλο για την επικράτηση ενός προϊόντος στην αγορά. Αντιθέτως, στο πρώτο μοντέλο (σχήμα(5.8)) το γ έχει γραμμική επιρροή. Συμπερασματικά, το φαινόμενο αυτό συμβαίνει, διότι στο πρώτο μοντέλο τα κριτήρια επιλογής ενός προϊόντος είναι πλήρως αντικειμενικά, οπότε ανάλογα με την ευαισθησία στην τιμή ανεβαίνει αντίστοιχα και η ζήτηση, ενώ στο τελικό μοντέλο υπάρχουν και άλλοι παράγοντες επιρροής, οι οποίοι ελαττώνονται απότομα σε βαρύτητα, όταν γίνεται πολύ μεγάλο το γ . Παρακάτω, άλλο ένα πολύ ενδιαφέρον αποτέλεσμα είναι η επιρροή του αρχικού μεριδίου της αγοράς, που όπως φαίνεται στα σχήματα (5.6), (5.7), στο πρώτο μοντέλο η επιρροή του είναι ελάχιστη, ενώ στο δεύτερο παίζει καθοριστικό ρόλο όπως φαίνεται στο σχήμα (5.27). Επιπρόσθετα, το φαινόμενο αυτό είναι απολύτως λογικό να συμβεί, διότι στο τελικό μοντέλο η επιλογή των καταναλωτών επηρεάζεται σημαντικά από τις επιλογές των γειτόνων τους, ενώ κάτι τέτοιο δεν υφίσταται για το πρώτο.

5.6 Οπτική Απεικόνιση

Η οπτική απεικόνιση που παρέχει η NetLogo παρέχει και αυτή κάποιες πληροφορίες οι οποίες δεν μπορούν να εντοπιστούν στα παραπάνω διαγράμματα. Αρχικά, είναι πιο κατανοητό για τον μέσο χρήστη να δει τις εικόνες για το πως διαμορφώνεται η αγορά από ότι να αναλύει διαγράμματα. Έπειτα, παρέχουν την γεωγραφική τοποθεσία όπου βρίσκονται οι καταναλωτές οι οποίοι επιλέγουν τα συγκεκριμένα προϊόντα. Κατόπιν, ο κόσμος του προβλήματος στην συγκεκριμένη περίπτωση αποτελείται από 31 X 31 κουτιά (για αυτό υπάρχουν και 961 καταναλωτές), το οποίο κάθε κουτί είναι ένας καταναλωτής. Επίσης, το κουτί (patch) έχει χρώμα πράσινο αν επιλέξει το προϊόν b και χρώμα κόκκινο αν επιλέξει το προϊόν a. Στο σχήμα (5.37) είναι η στιγμή πριν αρχίσει η προσομοίωση όταν όλα τα patches έχουν επιλέξει ένα τυχαίο προϊόν, με S_a και S_b να ισούνται με 0,5.



Σχήμα 5.37: initialization with $S_a = S_b$

Παρακάτω, παρόλο που τα S_a , S_b είναι ίσα, το σύστημα είναι σε κατάσταση lock-in(σχήμα (5.38)). Κατόπιν, φαίνεται ότι το σχήμα δεν έχει καμία σχέση διότι, οι καταναλωτές επηρεάζονται από τις αποφάσεις των γειτόνων τους και για αυτόν τον λόγο οι όμοιες επιλογές είναι η μια δίπλα στην άλλη.



Σχήμα 5.38: $S_a = S_b$ lock-in

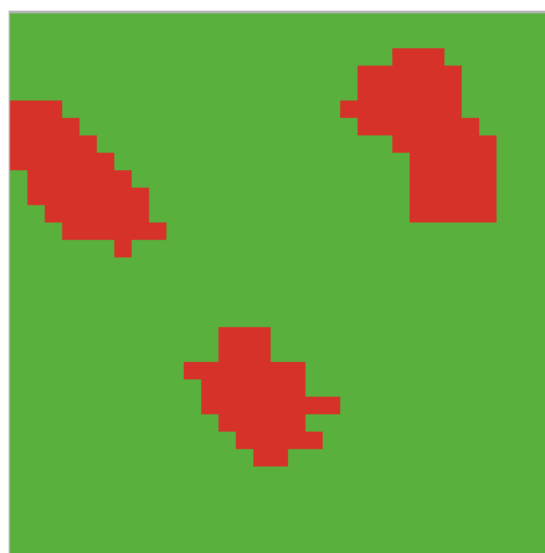
Επιπροσθέτως, παρατηρώντας όλα τα σχήματα του συστήματος μετά το lock-in, είτε για όταν κυριαρχεί προϊόν a(σχήματα (5.39), (5.40)) είτε για όταν είναι κυριαρχεί το προϊόν b(σχήματα (5.41), (5.42)), είναι όλες οι παρόμοιες επιλογές μαζί. Επίσης, αυτό δείχνει ότι το σύστημα λειτουργεί έτσι όπως θα έπρεπε να λειτουργήσει και δίνει πληροφορίες οι οποίες είναι πολύ δύσκολο να αναπαραχθούν από άλλο πρόγραμμα.



Σχήμα 5.39: $S_a > S_b$ lock-in



Σχήμα 5.40: $S_a > S_b$ lock-in



Σχήμα 5.41: $S_a < S_b$ lock-in



Σχήμα 5.42: $S_a < S_b$ lock-in

Κεφάλαιο 6ο: ΕΠΙΛΟΓΟΣ

6.1 Συμπεράσματα

Η συγκεκριμένη διπλωματική, μελέτησε αρχικά την συμπεριφορά της αγοράς σε μακροοικονομικό επίπεδο και στην συνέχεια την συμπεριφορά των καταναλωτών σε μικροοικονομικό επίπεδο. Παρακάτω, κατάφερε να εξάγει λογικά αποτελέσματα και να προβλέψει τις συμπεριφορές αυτές. Κατόπιν, τα αποτελέσματα είναι αρκετά αξιόπιστα διότι χρησιμοποιήθηκε σωστή μεθοδολογία για την υλοποίηση των μοντέλων, αξιόπιστες μαθηματικές εξισώσεις, εκτενή μελέτη των αποτελεσμάτων και τα αποτελέσματα συμπίπτουν με την κοινή λογική και με παρόμοιες μελέτες. Κατόπιν, αποκτήθηκαν πολλές γνώσεις της μεθοδολογίας κατασκευής και μελέτης τέτοιων μοντέλων. Έπειτα, λόγω αυτής τις μελέτης μπορούν να παρθούν αποφάσεις, βασισμένες στα αποτελέσματα της, από εταιρίες με ποιον τρόπο μπορούν να επιτύχουν το lock-in στην αγορά και με ποιον τρόπο μπορούν να αντιστρέψουν το lock-in των ανταγωνιστών τους.

6.2 Σφάλματα και Περιορισμοί

Παρακάτω, βρίσκονται τα γνωστά λάθη αλλά και περιορισμοί της διπλωματικής.

- Οι μεταβλητές C , λ_0 , λ_1 δεν επηρεάζουν καθόλου το σύστημα. Αυτό συνέβη λόγω της εφαρμογής υλοποίησης.
- Η υπολογιστική δύναμη είναι περιορισμένη. Στο πλαίσιο μιας διπλωματικής ο διαθέσιμος εξοπλισμός δεν είναι επαγγελματικός οπότε, δεν είναι δυνατόν να εξεταστεί ακόμα εκτενέστερα το σύστημα και πιο λεπτομερειακά.
- Περιορισμένη επικύρωση δεδομένων. Λόγω έλλειψης δεδομένων από παρόμοιες έρευνες και περιορισμένου χρόνου, δεν ήταν δυνατή η εξαγωγή δεδομένων από αυτές με σκοπό την διασταύρωση των αποτελεσμάτων.
- Περιορισμένη έρευνα σε μεταβλητές που δεν επηρεάζουν σημαντικά το σύστημα. Οι συγκεκριμένες μεταβλητές θα έπρεπε να μελετηθούν περαιτέρω με σκοπό, την επιβεβαίωση της επιρροής τους στο σύστημα, μέσω μαθηματικής διερεύνησης.

6.3 Βελτιώσεις

Παρακάτω, βρίσκονται οι βελτιώσεις που μπορούν να γίνουν στα μοντέλα για μελλοντική έρευνα.

- Διόρθωση των εξισώσεων του μοντέλου που δεν επηρεάζουν το σύστημα(C , λ_0 , λ_1) ή αλλαγή του εργαλείου υλοποίησης με σκοπό την επίλυση του συγκεκριμένου σφάλματος.
- Υλοποίηση των μοντέλων με επαγγελματικό εξοπλισμό. Με αποτέλεσμα να είναι μεγαλύτερη η υπολογιστική δύναμη, έτσι ώστε να μελετηθούν τα μοντέλα σε μεγαλύτερη κλίμακα και εκτενέστερα.
- Περισσότερη επικύρωση δεδομένων. Αυτό μπορεί να γίνει με διασταύρωση των στοιχείων με πολλαπλές έρευνες οι οποίες έχουν το ίδιο αντικείμενο μελέτης.
- Αλλαγή του εργαλείου υλοποίησης σε κάποιο πιο αποδοτικό (π.χ. Java, C++).
- Περισσότερη έρευνα σε μεταβλητές οι οποίες δεν επηρεάζουν σημαντικά το σύστημα. Πιο συγκεκριμένα, οι μεταβλητές αυτές θα μπορούσαν να είναι απλά θόρυβος στο σύστημα χωρίς να υπάρχει λογική στην επιρροή τους, κάτι το οποίο δεν μπορεί να επιβεβαιωθεί χωρίς περεταίρω μελέτη.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

[1] [Nigel Gilbert](#) , Agent-Based Models: Edition 2 ,2019.

Paper in Conference Proceedings

[2] Marco Janssen and Wander Jager , An integrated approach to simulating behavioural processes: A case study of the lock-in of consumption patterns , 1999 .

Internet Site

[3] https://www.w3schools.com/java/java_files_create.asp

[4] [https://en.wikipedia.org/wiki/Fixed_point_\(mathematics\)](https://en.wikipedia.org/wiki/Fixed_point_(mathematics))

[5]

https://www.eett.gr/opencms/opencms/EETT/Electronic_Communications/Antennas_EMR/health/EMRadiation/EMRadiation/Cycle.html

[6] <https://en.wikipedia.org/wiki/Quasiperiodicity>

[7] https://en.wikipedia.org/wiki/Attractor#Strange_attractor

[8]

https://el.wikipedia.org/wiki/%CE%98%CE%B5%CF%89%CF%81%CE%AF%CE%B1_%CF%84%CE%BF%CF%85_%CF%87%CE%AC%CE%BF%CF%85%CF%82

[9]

https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CE%BD%CE%BF%CE%BD%CE%B9%CE%BA%CE%AE_%CE%BA%CE%B1%CF%84%CE%B1%CE%BD%CE%BF%CE%BC%CE%AE

[10]

<https://www.opendemocracy.net/en/transformation/eight-essential-steps-transform-our-economy/>

ΠΑΡΑΡΤΗΜΑ Α : ΠΡΟΓΡΑΜΜΑ ΠΡΩΤΟΥ ΜΟΝΤΕΛΟΥ

```
1. package com.mycompany.lock.in_dynamics;
2.
3. import java.io.File;
4. import java.io.FileWriter;
5. import java.io.IOException;
6. import static java.lang.Math.log;
7. import java.util.*;
8.
9. public class FirstModel {
10.     int k;
11.     float mu ;
12.     float gamma ;
13.     float sigma ;
14.     float t ;
15.     int ta ;
16.     float FCa ;
17.     float VCa ;
18.     float FCb ;
19.     float VCb ;
20.     float D ;
21.     float Sa ;
22.     float Sb ;
23.     float Sha ;
24.     float Shb ;
25.     float IndSha ;
26.     float IndShb ;
27.     float Pa ;
28.     float Pb ;
29.     float LFa;
30.     float LFB ;
31.     float CLa ;
32.     float CLb ;
33.     float CLlock_in ;
34.     float Exp ;
35.     float E ;
36.     float Dt;
37.     float UN;
38.     float nxt;
39.     Random ran = new Random();
40.     String name ;
41.     float k1 ;
42.     int counter=0;
43.
44.
45.     public FirstModel (String name , float mu , float gamma , float sigma , int ta , float FCa ,
        float VCa , float FCb , float VCb , float D , float Sha , float Shb , float CLa , float CLb , float
        Dt)
46.     {
47.         this.name = name;
48.         this.mu = mu;
49.         this.sigma = sigma;
50.         this.gamma = gamma;
```

```

51.     this.ta = ta;
52.     this.FCa = FCa;
53.     this.VCa = VCa;
54.     this.FCb = FCb;
55.     this.VCb = VCb;
56.     this.D = D;
57.     this.Sha = Sha;
58.     this.Shb = Shb;
59.     this.CLa = CLa;
60.     this.CLb = CLb;
61.     this.Dt = Dt;
62. }
63.
64.
65.
66.     public void programm(float a ) throws IOException
67.     {
68.
69.         try
70.         {
71.             File myObj = new File(name + ".txt");
72.             if (myObj.createNewFile())
73.             {
74.                 System.out.println("File created: " + myObj.getName());
75.             }
76.             else
77.             {
78.                 System.out.println("File already exists.");
79.             }
80.         }
81.         catch (IOException e)
82.         {
83.             System.out.println("An error occurred.");
84.             e.printStackTrace();
85.         }
86.
87.         FileWriter myWriter = new FileWriter(name + ".txt");
88.
89.
90.         for (t=1 ; t <=4.267; t+=Dt )
91.         {
92.             if(t==1)
93.             {
94.                 counter=1;
95.             }
96.             else
97.             {
98.                 counter ++;
99.             }
100.
101.                 if(CLa > CLb)
102.                 {
103.                     CLlock_in = CLa ;
104.                 }
105.                 else

```

```

106.         {
107.             CLlock_in = CLb ;
108.         }
109.
110.
111.         Sa = Sha * D ;
112.         Sb = Shb * D ;
113.
114.
115.         Exp = (float) (-(log(gamma))/log(2)) ;
116.
117.         CLa = (float) (Dt*Sa + CLa);
118.         CLb = (float) (Dt*Sb + CLb);
119.
120.         LFa = (float) Math.pow((CLa/CLlock_in) , Exp);
121.         Lfb = (float) Math.pow((CLb/CLlock_in) , Exp);
122.
123.         Pa= (float) (LFa * (FCa + Sa * VCa ) / Sa);
124.         Pb= (float) (Lfb * (FCb + Sb * VCb ) / Sb);
125.
126.         IndSha = (float)(Math.exp(-mu * Pa)) / (float)(Math.exp(-mu * Pa) +
Math.exp(-mu * Pb));
127.         IndShb = (float)(Math.exp(-mu * Pb)) / (float)(Math.exp(-mu * Pa) +
Math.exp(-mu * Pb));
128.
129.
130.         Sha =Dt*((IndSha - Sha) / ta) + Sha ;
131.         Shb =Dt*((IndShb - Shb) / ta) + Shb ;
132.
133.
134.         Sa = Sha * D ;
135.         Sb = Shb * D ;
136.
137.         nxt = (float)(ran.nextGaussian()*a);
138.
139.         if(Sha*D + nxt < D)
140.         {
141.             UN=Sha*D + nxt;
142.             Sa = UN ;
143.             Sb= D-Sa;
144.         }
145.         else
146.         {
147.             Sa = D ;
148.             Sb= D - Sa;
149.         }
150.
151.
152.         System.out.println("Sa = " + Sa);
153.         System.out.println("Sb = " + Sb);
154.         myWriter.write(counter + ", " + a + ", " + Sa + ", " + Sb + "\n");
155.     }
156.     myWriter.write("\n" + "\n");
157.     myWriter.close();
158.

```

```
159.     }
160.
161.
162.     public static void main (String args[]) throws IOException
163.     {
164.
165.         float k1 ;
166.         float Val ;
167.         FirstModel pro = new FirstModel("", 0.5f,0.9f , 0.25f , 2 , 10f , 10f , 10f , 13f ,
168.         961 , 0.5f , 0.5f , 1f , 1f,0.033f);
169.
170.         for(k1 = pro.sigma ; k1<=8.25 ; k1 += 2)
171.         {
172.             pro.name = "sigmaS = " + String.valueOf(k1);
173.             pro.programm(k1);
174.         }
175.
176.     }
177.
178. }
```

ΠΑΡΑΡΤΗΜΑ Β : ΠΡΟΓΡΑΜΜΑ ΤΕΛΙΚΟΥ ΜΟΝΤΕΛΟΥ

```
1. globals [Dt Sha D Sa Sb Shb IndSha IndShb Pa Pb P1 P2 LFa LFb CLa CLb CLlockin EX AX
   TSa TSb p]
2. patches-own[x LNSa LNS1a LNS2a LNS3a LNS4a LNSb LNS1b LNS2b LNS3b LNS4b
   GAMMA1 GAMMA2 GAMMA3 B β0 β1 a Bneighbors XN0 XN1]
3.
4. to no-op
5. end
6.
7.
8. to setup
9. clear-all
10. ;global variables initialization
11. set Dt 0.1
12. set Sha 0.5
13. set Shb 1 - Sha
14. set CLa 1
15. set CLb 1
16.
17. ;red product a , green product b
18. ;patches-own variables initialization
19. ask patches [
20.
21.   ifelse ((random-float 1.0 ) <= Sha) [set pcolor red] [set pcolor green]
22.   ifelse (pcolor = red)[set x 0][set x 1]
23.   ;set B (40 + random-float 40 )
24.   set B (random-normal 50 5 )
25.   set a (random-normal 5 0.5)
26.   set β0 (0.2 + random-float 0.1)
27.   set β1 (0.2 + random-float 0.3)
28.   set GAMMA1 (0.2 + random-float 0.1)
29.   set GAMMA2 (0.2 + random-float 0.1)
30.   set GAMMA3 (0.2 + random-float 0.1)
31.
32. ]
33.
34.
35. reset-ticks
36.
37.
38. end
39. ;main programm
40. to go
41. tick
42.
43. set Dt ( Dt + 0.0333 )
44.
45. set D count patches
46. ;Sha is the value of the agents that preferred the product a over the product b compared to the
   total number of products
47. set Sha (count patches with [x = 0]) / D
48. set Shb (count patches with [x = 1]) / D
49.
```

```

50.
51. ifelse CLa > CLb
52.   [set CLlockin CLa]
53.   [set CLlockin CLb]
54.
55.
56. set Sa (Sha * D)
57. set Sb (Shb * D)
58.
59. ;if none of the patches prefer one product stop the simulation
60. if Sa = 0 [ set Sb 961 ]
61. if Sa = 0 [ no-op ]
62. if Sb = 0 [ set Sa 961 ]
63. if Sb = 0 [ no-op ]
64.
65. set CLa (Dt * Sa + CLa)
66. set CLb (Dt * Sb + CLb)
67.
68. set EX ((-log (gamma) 10)/(log 2 10))
69.
70. set LFa ((CLa / CLlockin)^(EX))
71. set LFc ((CLb / CLlockin)^(EX))
72.
73. set Pa (LFa * (FCa + Sa * VCa) / Sa)
74. set Pb (Lfc * (FCb + Sb * VCb) / Sb)
75.
76. ifelse (Sha * D + (random-normal 0 sigma) < D)
77. [set Sa ( Sha * D + (random-normal 0 sigma))]
78. set Sb (D - Sa)]
79. [set Sa ( D)
80. set Sb (D - Sa)]
81.
82. set TSa Sa
83. set TSb Sb
84.
85. ask patches [ play ]
86.
87. ask patches [ update ]
88.
89.
90. end
91.
92.
93. ; NLS calculation
94. to play
95.
96. let L []
97. set L [B] of neighbors
98.
99. set Bneighbors median L
100.
101. ;XN0 is the number of neighbors with the product a as preference
102. ;XN1 is the number of neighbors with the product b as preference
103.
104. set XN0 count (neighbors with [x = 0])

```

```

105.      set XN1 (8 - XN0)
106.
107.
108.      set LNS1a ( ( XN0 / 8))
109.      set LNS1b (XN1 / 8)
110.
111.      set LNS2a  $\beta_0$ 
112.      set LNS2b  $\beta_1$ 
113.
114.      set LNS3a (B / Pa)
115.      set LNS3b (B / Pb)
116.
117.      set LNS4a (1 - (e ^ (( - a) / (C * (1 -  $\mu$ ) +  $\lambda_0$ ))))
118.      set LNS4b (1 - (e ^ (( - a) / (C * (1 -  $\mu$ ) +  $\lambda_1$ ))))
119.
120.
121.      set LNSa (LNS1a ^ GAMMA1) + (LNS2a ^ GAMMA2) + (LNS3a ^ GAMMA3) +
(LNS4a ^ (1 - GAMMA1 - GAMMA2 - GAMMA3))
122.      set LNSb (LNS1b ^ GAMMA1) + (LNS2b ^ GAMMA2) + (LNS3b ^ GAMMA3) +
(LNS4b ^ (1 - GAMMA1 - GAMMA2 - GAMMA3))
123.
124.      end
125.
126.      ;update after the results (x,C,colors)
127.      to update
128.
129.      ifelse (LNSa > LNSb)
130.          [ifelse (LNSa > LNSStol) ;control the certainty of the agents decision
131.              [set x 0] ;if LNSa > LNSb and the LNS is above the LNSStolerance the agent is
certain and chooses the product a
132.                  [
133.                      ifelse (XN0 > XN1 and ((B * (1 - Btol)) <= Bneighbors) and (Bneighbors <= (B *
(1 + Btol))))[set x 0][set x 1]
134.                  ] ;if LNSa > LNSb and the LNS is below the LNSStolerance the agent is uncertain
and he is getting infuelnced by his neighbors with similiar budget
135.                  ]
136.          [ifelse (LNSb > LNSStol)
137.              [set x 1]
138.              [
139.                  ifelse (XN0 > XN1 and ((B * (1 - Btol)) <= Bneighbors) and (Bneighbors <= (B *
(1 + Btol))))[set x 0][set x 1]
140.              ]
141.          ] ;depending on the NLS value of each agent set x = 0 for product a and x = 1 for
product b
142.
143.          ifelse (x = 0)[set pcolor red][set pcolor green] ;"repaint" the color of the agents
accordingly with the value of preference (x)
144.
145.          ;update the C value with the new data
146.          let N0 count (patches with [x = 0])
147.          let N1 count (patches with [x = 1])
148.          set C (C * (1 -  $\mu$ ) +  $\lambda_0$  * N0 +  $\lambda_1$  * N1)
149.
150.      end

```

ΠΑΡΑΡΤΗΜΑ Γ : SCRIPT R

```
1. var_csv=read.csv("C:/Users/kosta/OneDrive/Υπολογιστής/thesis19_10var.csv",sep=',',skip=6)
2.
3. colnames(var_csv) = c('runNum','var','step','Sa','Sb')
4.
5. var10Sa <- var_csv[var_csv$step == 100 & var_csv$var == 10 ,]
6. var10FSa<- mean(var10Sa$Sa)
7.
8. var10Sb <- var_csv[var_csv$step == 100 & var_csv$var == 10 ,]
9. var10FSb<- mean(var10Sb$Sb)
10.
11. var100Sa <- var_csv[var_csv$step == 100 & var_csv$var == 100 ,]
12. var100FSa<- mean(var100Sa$Sa)
13.
14. var100Sb <- var_csv[var_csv$step == 100 & var_csv$var == 100 ,]
15. var100FSb<- mean(var100Sb$Sb)
16.
17. var1000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 1000 ,]
18. var1000FSa<- mean(var1000Sa$Sa)
19.
20. var1000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 1000 ,]
21. var1000FSb<- mean(var1000Sb$Sb)
22.
23. var2000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 2000 ,]
24. var2000FSa<- mean(var2000Sa$Sa)
25.
26. var2000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 2000 ,]
27. var2000FSb<- mean(var2000Sb$Sb)
28.
29. var3000Sa <- var_csv[var_csv$step == 100 & var_csv$var == 3000 ,]
30. var3000FSa<- mean(var3000Sa$Sa)
31.
32. var3000Sb <- var_csv[var_csv$step == 100 & var_csv$var == 3000 ,]
33. var3000FSb<- mean(var3000Sb$Sb)
34.
35. varData<-
    data.frame(varVal=c(10,100,1000,2000,3000),Sa=(var10FSa,var100FSa,var1000FSa,var2000
    FSa,var3000FSa),Sb = c(var10FSb,var100FSb,var1000FSb,var2000FSb,var3000FSb))
36.
37. library(tidyverse)
38.
39. ggplot(varData , aes(varVal))+geom_line(aes(y = Sa , colour = "Sa"))+geom_line(aes(y = Sb
    , colour = "Sb"))+geom_point(aes(y = Sa , colour = "Sa"))+geom_point(aes(y = Sb , colour =
    "Sb")) +labs(title = "var 10-3000",y = "Agents' preference" ,x="var")
```

ΠΑΡΑΡΤΗΜΑ Δ : ΔΕΔΟΜΕΝΑ ΠΡΩΤΟΥ ΜΟΝΤΕΛΟΥ

mu 0.5

gamma 0.9

sigma 0.5

ta 2
Dt 0.033
FCa 10
VCa 10
FCb 10
VCb 13
D 961
Sha 0.5
Shb 0.5
CLa 1
CLb 1

ΠΑΡΑΡΤΗΜΑ Ε : ΔΕΔΟΜΕΝΑ ΤΕΛΙΚΟΥ ΜΟΝΤΕΛΟΥ

gamma 0.9
sigma 0.5
Dt 0.033
FCa 10
VCa 10
FCb 10
VCb 13
D 961
Sha 0.5
Shb 0.5
CLa 1
CLb 1
C 100
 λ_0 1
 λ_1 1
LNStol 2.7
Btol 0.2
 μ 0.5
B random-normal 50 5
a random-normal 5 0.5

β_0 0.2-0.3

β_1 0.2-0.5

GAMMA1 0.2-0.3

GAMMA2 0.2-0.3

GAMMA3 0.2-0.3